

**BÀI GIẢNG**

**TIN HỌC ĐẠI CƯƠNG**

**Biên soạn : PHAN THỊ HÀ**  
**NGUYỄN TIẾN HÙNG**



VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
Khu Đường Nguyễn Trãi, Hà Đông-Hà Tây  
Tel: (04).5541221; Fax: (04).5540587  
Website: <http://www.c-ptth.edu.vn>; E-mail: [dhkc@ptth.edu.vn](mailto:dhkc@ptth.edu.vn)

## Chương 1: CÁC KHÁI NIỆM CƠ BẢN

### 1.1. THÔNG TIN VÀ XỬ LÝ THÔNG TIN

#### 1.1.1. Khái quát

##### 1.1.1.1. Khái niệm thông tin

###### a. Khái niệm

Trong đời sống hàng ngày, chúng ta tiếp nhận và sử dụng nhiều thông tin. Thông tin đem lại cho chúng ta sự hiểu biết, giúp chúng ta nhận thức đúng đắn về các hiện tượng tự nhiên và xã hội; cũng nhờ thông tin ta có được những hành động hợp lý nhằm đạt được những mục đích trong cuộc sống.

Chúng ta ai cũng thấy được sự cần thiết của thông tin và cảm nhận được thông tin là gì. Nhưng để đưa ra một định nghĩa chính xác về thông tin thì hầu hết chúng ta đều lúng túng bởi thông tin là một khái niệm khá trừu tượng và nó được thể hiện dưới nhiều dạng thức khác nhau. Tuy nhiên, người ta có thể tạm đưa ra khái niệm sau đây:

"Thông tin thường được hiểu là nội dung chứa trong thông báo nhằm tác động vào nhận thức của một số đối tượng nào đó".

Thông báo được thể hiện bằng nhiều hình thức: văn bản, lời nói, hình ảnh, cử chỉ...; và các thông báo khác nhau có thể mang cùng một nội dung. Trong lĩnh vực tin học, thông tin có thể được phát sinh, được lưu trữ, được biến đổi trong những vật mang tin; thông tin được biến đổi bởi các dữ liệu và các dữ liệu này có thể được truyền đi, được sao chép, được xử lý hoặc bị phá hủy.

###### *Ta có thể lấy một vài ví dụ sau để minh họa*

Thông báo thể hiện dưới dạng văn bản ví dụ như "Thông tin về một mạng máy tính bị nhiễm virus" - Trong thông báo này, thành phần "Mạng máy tính" đóng vai trò là vật mang tin, còn sự kiện "nhiễm virus" là dữ liệu của thông tin.

Hoặc ví dụ "Nhiệt độ đo được ở bệnh nhân là 41°C" - Thông tin này có thể được thể hiện dưới dạng văn bản hoặc lời nói. Dữ liệu ở đây là 41°C (nếu được thông báo bằng lời nói thì dữ liệu chính là tín hiệu) và thông tin thu được thông qua dữ liệu cho thấy bệnh nhân bị sốt cao...v.v

###### b. Phân loại thông tin

Dựa trên đặc điểm liên tục hay gián đoạn về thời gian của các tín hiệu thể hiện thông tin, ta có thể chia thông tin làm hai loại cơ bản như sau:

+ Thông tin liên tục: Là thông tin mà các tín hiệu thể hiện loại thông tin này thường là các đại lượng được tiếp nhận liên tục trong miền thời gian và nó được biểu diễn bằng hàm số có biến số thời gian độc lập, liên tục.

**Ví dụ:** Thông tin về mức thủy triều của nước biển hay thông tin về các tia bức xạ từ ánh sáng mặt trời...

+ Thông tin rời rạc: Là thông tin mà các tín hiệu thể hiện loại thông tin này thường là các đại lượng được tiếp nhận có giá trị ở từng thời điểm rời rạc và nó được biểu diễn dưới dãy số.

**Ví dụ :** Thông tin các vụ tai nạn xảy ra trên đoạn đường Nguyễn Trãi.

**c. Đơn vị đo thông tin**

Các đại lượng vật lý đều có đơn vị đo chẳng hạn như đơn vị đo khối lượng (kg), đo chiều dài (m) và đo thời gian (giây)...v.v. Để lượng hoá một thông tin ta cũng cần đưa ra một đơn vị đo thông tin.

Trong tin học, đơn vị đo thông tin nhỏ nhất là Bit (viết tắt của Binary digit - số nhị phân) - được biểu diễn với 2 giá trị 0 và 1, viết tắt là b.

Trong thực tế người ta thường dùng đơn vị lớn hơn là byte. Byte là một nhóm 8 bit trong bảng mã ASCII

Ngoài ra người ta còn dùng các bội số của byte như sau:

Tên gọi	Ký hiệu	Giá trị
Byte	B	8 bit
Word	w	8,16, 32 hoặc 64 bit
KiloByte	KB	$1024b=2^{10}b$
MegaByte	MB	$1024Kb=2^{10}Kb$
GigaByte	GB	$1024Mb=2^{10}Mb$
TeraByte	TB	$1024Gb=2^{10}Gb$

**d. Mã hoá thông tin rời rạc**

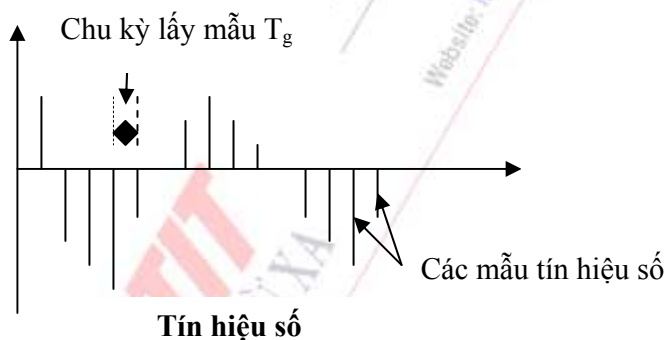
Mã hóa thông tin là quá trình biến đổi thông tin từ dạng biểu diễn thông thường sang một dạng khác theo quy ước nhất định. Quá trình biến đổi ngược lại của mã hóa thông tin được gọi là phép giải mã.

**Ví dụ:** Ta có 1 tập quản lý hồ sơ sinh viên. Nếu ta quản lý bằng tên thì sẽ xảy ra rất nhiều trường hợp tên bị trùng nhau. Nếu ta thêm các yếu tố khác kèm theo như địa chỉ, ngày sinh, quê quán...v.v thì việc quản lý trở nên rất rườm rà, phức tạp mà vẫn không loại trừ được khả năng trùng nhau. Nếu ta gán cho mỗi một sinh viên 1 mã số ID khác nhau thì việc quản lý hồ sơ sẽ trở nên thuận tiện hơn nhiều. Từ mã số ID, ta có thể tìm ra số liệu về sinh viên tương ứng. Như vậy, quá trình gán mã số ID cho mỗi hồ sơ sinh viên được gọi là mã hóa; còn quá trình dựa trên mã số ID để xác định thông tin về sinh viên gọi là giải mã.

Tất cả các thông tin ở dạng văn bản (text), chữ (character), số (number), ký hiệu (symbol), đồ họa (graphic), hình ảnh (image) hoặc âm thanh (sound)... đều được biểu diễn bằng các tín hiệu (signals). Các tín hiệu biểu diễn này có thể là liên tục hay rời rạc và nó được đưa vào xử lý thông qua các hệ thống máy tính. Đối với hệ thống máy tính tương tự (Analog Computer), thông tin được đưa vào xử lý chủ yếu là một số các tín hiệu liên tục như tín hiệu điện, âm thanh... Trong khi đó, hầu hết các dữ liệu mà chúng ta có được thường ở dạng các tín hiệu rời rạc và nó được xử lý trên các hệ thống máy tính số. Do đó, khi đưa các tín hiệu này vào máy tính, chúng được mã hóa theo các tín hiệu số (digital signal) nhằm giúp máy tính có thể hiểu được thông tin đưa vào. Đây là cơ sở thực tiễn của nguyên lý mã hoá thông tin rời rạc. Nguyên lý này tập trung các điểm chủ yếu sau:

Tín hiệu liên tục có thể xem như một chuỗi xấp xỉ các tín hiệu rời rạc với chu kỳ lấy mẫu nhỏ ở mức độ chấp nhận được.

Tín hiệu rời rạc có thể được đặc trưng qua các bộ ký hiệu hữu hạn (chữ cái, chữ số, dấu, ...) gọi là phép mã hóa (encode). Mọi phép mã hóa đều có thể xây dựng trên bộ ký hiệu các chữ số, đặc biệt chỉ cần bộ ký hiệu gồm 2 chữ số là 0 và 1. Ngược với phép mã hoá gọi là phép giải mã (decode).



Tín hiệu rời rạc là tín hiệu có trục thời gian bị rời rạc hoá với chu kỳ lấy mẫu là  $T_s = 1/F_s$ , trong đó  $F_s$  là tần số lấy mẫu. Ta có thể xét một số ví dụ như tiếng nói con người thông thường nằm trong dải âm tần từ 0,3 kHz đến 3,4 kHz; khi tiếng nói con người được truyền đưa trên mạng nó sẽ được rời rạc hóa bằng tần số lấy mẫu là 8 kHz nhưng người nghe vẫn không cảm nhận được điều này. Một ví dụ khác về thông tin rời rạc là hình trên phim khi được chiếu lên màn ảnh là các ảnh rời rạc xuất hiện với tốc độ 25 ảnh/giây. Mắt người không phân biệt sự rời rạc này nên có cảm tưởng hình ảnh là liên tục.

Mã hoá thông tin rời rạc là một khái niệm rất căn bản và ứng dụng nhiều trong kỹ thuật máy tính điện tử.

### 1.1.1.2. Xử lý thông tin

#### a. Sơ đồ tổng quát của một quá trình xử lý thông tin

Quá trình xử lý thông tin chính là sự biến đổi những dữ liệu đầu vào ở dạng rời rạc thành thông tin đầu ra ở dạng chuyên biệt phục vụ cho những mục đích nhất định. Mọi quá trình xử lý

thông tin cho dù thực hiện bằng máy tính hay bằng con người đều phải tuân thủ theo chu trình sau:

Dữ liệu (data) được nhập ở đầu vào (input). Sau đó, máy tính hay con người sẽ thực hiện những quá trình xử lý để xuất thông tin ở đầu ra (output). Quá trình nhập dữ liệu, xử lý và xuất thông tin đều có thể được lưu trữ để phục vụ cho các quá trình tiếp theo khác.



**Mô hình tổng quát quá trình xử lý thông tin**

### ***b. Xử lý thông tin bằng máy tính điện tử (MTĐT)***

Máy tính điện tử là một hệ thống xử lý thông tin tự động dựa trên nguyên tắc chung của quá trình xử lý thông tin. Mặc dù khả năng tính toán của máy tính vượt xa so với khả năng tính toán của con người và các phương tiện khác; tuy nhiên, máy tính sẽ không tự nó đưa ra quyết định khi nào phải làm gì mà nó chỉ có thể hoạt động được nhờ sự chỉ dẫn của con người - tức là con người phải cung cấp đầy đủ ngay từ đầu cho MTĐT các mệnh lệnh, chỉ thị để hướng dẫn MTĐT theo yêu cầu đề ra.

Tổng quát quá trình xử lý thông tin trên MTĐT có thể được tóm tắt như sau:

- + Trước hết đưa chương trình cần thực hiện (do con người lập sẵn) vào bộ nhớ của máy tính
- + Máy bắt đầu xử lý, dữ liệu nhập từ môi trường ngoài vào bộ nhớ (Thông qua thiết bị nhập dữ liệu).
- + Máy thực hiện thao tác dữ liệu và ghi kết quả trong bộ nhớ.
- + Đưa kết quả từ bộ nhớ ra bên ngoài nhờ các thiết bị xuất (máy in, màn hình).

Máy tính điện tử có một số đặc điểm chính như sau:

- + Tốc độ xử lý nhanh, độ tin cậy cao.
- + Khả năng nhớ rất lớn.
- + Tham số về tốc độ thường được tính bằng số phép tính thực hiện trong một giây, còn khả năng nhớ được tính theo dung lượng bộ nhớ trong đo bằng KB, MB.

### ***1.1.1.3. Tin học và các lĩnh vực nghiên cứu của tin học***

#### ***a. Tin học là gì ?***

Tin học là một ngành khoa học công nghệ nghiên cứu các phương pháp xử lý thông tin một cách tự động dựa trên các phương tiện kỹ thuật mà chủ yếu hiện tại là máy tính điện tử.

**b. Các lĩnh vực nghiên cứu của tin học :**

Từ các định nghĩa trên thấy tin học gồm hai khía cạnh nghiên cứu:

- Khía cạnh khoa học: nghiên cứu về các phương pháp xử lý thông tin tự động.

- Khía cạnh kỹ thuật: nhằm vào 2 kỹ thuật phát triển song song - đó là :

+ Kỹ thuật phần cứng (hardware engineering): nghiên cứu chế tạo các thiết bị, linh kiện điện tử, công nghệ vật liệu mới... hỗ trợ cho máy tính và mạng máy tính đẩy mạnh khả năng xử lý toán học và truyền thông thông tin.

+ Kỹ thuật phần mềm (software engineering): nghiên cứu phát triển các hệ điều hành, ngôn ngữ lập trình cho các bài toán khoa học kỹ thuật, mô phỏng, điều khiển tự động, tổ chức dữ liệu và quản lý hệ thống thông tin.

**c. Ứng dụng của tin học**

Tin học hiện đang được ứng dụng rộng rãi trong tất cả các ngành nghề khác nhau của xã hội từ khoa học kỹ thuật, y học, kinh tế, công nghệ sản xuất đến khoa học xã hội, nghệ thuật,... như:

- Tự động hóa văn phòng
- Quản trị kinh doanh
- Thống kê
- An ninh, quốc phòng
- Công nghệ thiết kế, Giáo dục
- Y học, Công nghệ in
- Nông nghiệp, Nghệ thuật, giải trí, v.v....

**1.1.2. Biểu diễn thông tin trong máy tính**

**1.1.2.1. Hệ đếm và logic mệnh đề**

**a. Hệ đếm**

Hệ đếm là tập hợp các ký hiệu và qui tắc sử dụng tập ký hiệu đó để biểu diễn và xác định các giá trị các số. Mỗi hệ đếm có một số ký số (digits) hữu hạn và tổng số ký số của mỗi hệ đếm được gọi là cơ số (base hay radix), ký hiệu là b.

Các hệ đếm phổ biến hiện nay hay dùng là hệ đếm La mã và hệ đếm thập phân, hệ đếm nhị phân, hệ đếm bát phân, hệ đếm thập lục phân. Nhưng trong lĩnh vực kỹ thuật hiện nay phổ biến 4 hệ đếm như sau :

Hệ đếm	Cơ số	Ký số và trị tuyệt đối
Hệ nhị phân	2	0, 1
Hệ bát phân	8	0, 1, 2, 3, 4, 5, 6, 7
Hệ thập phân	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Hệ thập lục phân	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

**\*/ Hệ đếm thập phân (decimal system)**

Hệ đếm thập phân hay hệ đếm cơ số 10 là một trong những phát minh của người Ả rập cổ, bao gồm 10 ký số theo ký hiệu sau:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Quy tắc tính giá trị của hệ đếm này là mỗi đơn vị ở một hàng bất kỳ có giá trị bằng 10 đơn vị của hàng kế cận bên phải. (Ở đây  $b = 10$ ). Bất kỳ số nguyên dương trong hệ thập phân được thể hiện như là một tổng các chuỗi các ký số thập phân nhân với 10 lũy thừa, trong đó số mũ lũy thừa được tăng thêm 1 đơn vị kể từ số mũ lũy thừa phía bên phải nó. Số mũ lũy thừa của hàng đơn vị trong hệ thập phân là 0.

Ví dụ: Số 5246 có thể được thể hiện như sau:

$$5246 = 5 \times 10^3 + 2 \times 10^2 + 4 \times 10^1 + 6 \times 10^0 = 5 \times 1000 + 2 \times 100 + 4 \times 10 + 6 \times 1$$

Thể hiện như trên gọi là ký hiệu mở rộng của số nguyên.

$$\text{Vì } 5246 = 5000 + 200 + 40 + 6$$

Như vậy, trong số 5246: ký số 6 trong số nguyên đại diện cho giá trị 6 đơn vị (1s), ký số 4 đại diện cho giá trị 4 chục (10s), ký số 2 đại diện cho giá trị 2 trăm (100s) và ký số 5 đại diện cho giá trị 5 ngàn (1000s). Nghĩa là, số lũy thừa của 10 tăng dần 1 đơn vị từ trái sang phải tương ứng với vị trí ký hiệu số,

$$10^0 = 1 \quad 10^1 = 10 \quad 10^2 = 100 \quad 10^3 = 1000 \quad 10^4 = 10000 \dots$$

Mỗi ký số ở thứ tự khác nhau trong số sẽ có giá trị khác nhau, ta gọi là giá trị vị trí (place value).

Phần phân số trong hệ thập phân sau dấu chấm phân cách (theo qui ước của Mỹ) thể hiện trong ký hiệu mở rộng bởi 10 lũy thừa âm tính từ phải sang trái kể từ dấu chấm phân cách

$$\begin{aligned} \text{Ví dụ: } 254.68 &= 2 \times 10^2 + 5 \times 10^1 + 4 \times 10^0 + 6 \times 10^{-1} + 8 \times 10^{-2} \\ &= 200 + 50 + 4 + \frac{6}{10} + \frac{8}{100} \end{aligned}$$

Tổng quát, hệ đếm cơ số  $b$  ( $b \geq 2$ ,  $b$  là số nguyên dương) mang tính chất sau:

- Có  $b$  ký số để thể hiện giá trị số. Ký số nhỏ nhất là 0 và lớn nhất là  $b-1$ .
- Giá trị vị trí thứ  $n$  trong một số của hệ đếm bằng cơ số  $b$  lũy thừa  $n$ :  $b^n$

Số  $N(b)$  trong hệ đếm cơ số  $(b)$  thể hiện:  $N_{(b)} = a_n a_{n-1} a_{n-2} \dots a_1 a_0 a_{-1} a_{-2} \dots a_{-m}$

trong đó, số  $N(b)$  có  $n+1$  ký số chẵn ở phần nguyên và  $m$  ký số lẻ, sẽ có giá trị là:

$$N_{(b)} = a_n \cdot b^n + a_{n-1} \cdot b^{n-1} + a_{n-2} \cdot b^{n-2} + \dots + a_1 b^1 + a_0 \cdot b^0 + a_{-1} \cdot b^{-1} + a_{-2} \cdot b^{-2} + \dots + a_{-m} \cdot b^{-m}$$

Hay

$$N_{(b)} = \sum_{i=-m}^n a_i \cdot b^i$$

**\* Hệ đếm nhị phân (binary system)**

Với  $b = 2$ , chúng ta có hệ đếm nhị phân. Đây là hệ đếm đơn giản nhất với 2 chữ số là 0 và 1. Mỗi chữ số nhị phân gọi là BIT (viết tắt từ chữ BInary digiT). Hệ nhị phân tương ứng với 2 trạng thái của các linh kiện điện tử trong máy tính - cụ thể: đóng (có điện) ký hiệu là 1 và tắt (không điện) ký hiệu là 0. Vì hệ nhị phân chỉ có 2 trị số là 0 và 1, nên khi muốn diễn tả một số lớn hơn, hoặc các ký tự phức tạp hơn thì cần kết hợp nhiều bit với nhau.

Ta có thể chuyển đổi hệ nhị phân theo hệ thập phân quen thuộc.

Ví dụ 3.6: Số  $11101.11_{(2)}$  sẽ tương đương với giá trị thập phân là :

					← vị trí dấu chấm cách		
Số nhị phân:	1	1	0	1		1	1
Số vị trí:	4	3	2	1		-1	-2
Trị vị trí:	$2^4$	$2^3$	$2^2$	$2^1$		$2^{-1}$	$2^{-2}$
Hệ 10 là:	16	8	4	2		0.5	0.25

như vậy:

$$11101.11_{(2)} = 1 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 + 1 \times 0.5 + 1 \times 0.25 = 29.75_{(10)}$$

tương tự số  $10101_{(2)}$  sang hệ thập phân sẽ là:

$$10101_{(2)} = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 0 + 4 + 0 + 1 = 13_{(10)}$$

**\*/ Hệ đếm La mã**

Hệ đếm La mã được xem như là hệ đếm có tính hệ thống đầu tiên của con người. Hệ đếm La mã sử dụng các ký hiệu ứng với các giá trị như sau:

$$I = 1 \quad V = 5 \quad X = 10 \quad L = 50 \quad C = 100 \quad D = 500 \quad M = 1000$$

Ký số La mã có một số qui tắc sau:

- Số lần  $n$  liên tiếp kế nhau của mỗi ký hiệu thể hiện giá trị ký hiệu tăng lên  $n$  lần. Số lần  $n$  chỉ là 1 hoặc 2 hoặc 3. Riêng ký hiệu M được phép xuất hiện 4 lần liên tiếp.

Ví dụ: III = 3 x 1 = 3; XX = 2 x 10 = 20; MMMM = 4000, ...

- Hai ký hiệu đứng cạnh nhau, nếu ký hiệu nhỏ hơn đứng trước thì giá trị của chúng sẽ là hiệu số của giá trị ký hiệu lớn trừ giá trị ký hiệu nhỏ hơn.

Ví dụ: IV = 5 - 1 = 4; IX = 10 - 1 = 9; CD = 500 - 100 = 400; CM = 1000 - 100 = 900

- Hai ký hiệu đứng cạnh nhau, nếu ký hiệu nhỏ đứng sau thì giá trị của chúng sẽ là tổng số của 2 giá trị ký hiệu.

Ví dụ: XI = 10 + 1 = 11; DCC = 500 + 100 + 100 = 700

Giá trị 3986 được thể hiện là: MMMCMLXXXVI

- Để biểu thị những số lớn hơn 4999 (MMMMCMXCIX), chữ số La mã giải quyết bằng cách dùng những vạch ngang đặt trên đầu ký tự. Một vạch ngang tương đương với việc nhân giá



trị của ký tự đó lên 1000 lần. Ví dụ  $M = 1000 \times 1000 = 10^6$ . Như vậy, trên nguyên tắc chữ số La mã có thể biểu thị các giá trị rất lớn. Tuy nhiên trong thực tế người ta thường sử dụng 1 đến 2 vạch ngang là nhiều.

Hệ đếm La mã hiện nay ít được sử dụng trong tính toán hiện đại.

**\* / Hệ đếm bát phân (octal number system)**

Nếu dùng 1 tập hợp 3 bit thì có thể biểu diễn 8 trị số khác nhau: 000, 001, 010, 011, 100, 101, 110, 111. Các trị số này tương đương với 8 trị số trong hệ thập phân là 0, 1, 2, 3, 4, 5, 6, 7. Tập hợp các chữ số này gọi là hệ bát phân, là hệ đếm với  $b = 8 = 2^3$ . Trong hệ bát phân, trị số vị trí là lũy thừa của 8.

Ví dụ:  $235_{(8)} = 2 \times 8^2 + 3 \times 8^1 + 5 \times 8^0 = 157_{(10)}$

**\* / Hệ đếm thập lục phân (hexa-decimal number system)**

Hệ đếm thập lục phân là hệ cơ số  $b = 16 = 2^4$  tương đương với tập hợp 4 chữ số nhị phân (4 bit). Khi thể hiện ở dạng hexa-decimal, ta có 16 ký tự gồm 10 chữ số từ 0 đến 9, và 6 chữ in A, B, C, D, E, F để biểu diễn các giá trị số tương ứng là 10, 11, 12, 13, 14, 15. Với hệ thập lục phân, trị vị trí là lũy thừa của 16.

Ví dụ:  $34F5C_{(16)} = 3 \times 16^4 + 4 \times 16^3 + 15 \times 16^2 + 5 \times 16^1 + 12 \times 16^0 = 216294_{(10)}$

Ghi chú: Một số chương trình qui định viết số hexa phải có chữ H ở cuối chữ số.

Ví dụ: Số 15 viết là FH.

Bảng qui đổi tương đương 16 chữ số đầu tiên của 4 hệ đếm

Hệ 10	Hệ 2	Hệ 8	Hệ 16
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

**\* Chuyển đổi số giữa các hệ đếm**

Chuyển một số từ hệ cơ số  $L=10$  sang hệ cơ số H:

Ta lưu ý rằng các hệ cơ số ta xét đều lấy 1 làm đơn vị, vì vậy một số bất kỳ dù biểu diễn ở hệ cơ số nào thì phần thập phân và phần nguyên đều không đổi. Nghĩa là dù biến đổi sang hệ cơ số nào đi nữa thì phần thập phân cũng chỉ chuyển sang phần thập phân, phần nguyên sang phần nguyên.

Giả sử ta có một số có phần thập phân  $b=k+d$  trong hệ cơ số  $L$  trong đó  $k$  là phần nguyên trước dấu phẩy và  $d$  là phần thập phân sau dấu phẩy. Ta sẽ chuyển đổi riêng từng phần theo quy tắc sau:

- Với phần nguyên: Lấy  $k$  chia liên tiếp cho  $H$  cho đến khi thương số bằng 0, phép chia thứ  $i$  có số dư  $b_i$  là chữ số trong hệ cơ số  $H$ ,  $i = 0, 1, 2, \dots, n$ , khi đó  $b_n b_{n-1} b_{n-2} \dots b_0$  là phần nguyên của số  $b$  trong hệ cơ số  $H$ .

- Với phần thập phân: Lấy phần thập phân của  $d$  nhân liên tiếp với  $H$  cho đến khi kết quả phép nhân không còn phần thập phân hoặc đạt được độ chính xác ta cần, mỗi lần nhân ta lấy phần nguyên của kết quả là  $c_j$  là chữ số trong hệ cơ số  $H$ ,  $j = 1, 2, \dots, m$ . Khi đó số  $c_1 c_2 \dots c_m$  chính là phần thập phân của số nhị phân cần tìm. (Chúng ta lưu ý là sau mỗi lần nhân ta chỉ lấy phần thập phân để nhân tiếp với  $H$ , phần nguyên ở đây được hiểu là phần bên trái dấu chấm thập phân).

Ví dụ: Cho số thập phân 14.125 tìm số nhị phân tương ứng.

Ta có  $k = 14$ ,  $d = 0.125$

Chuyển đổi phần nguyên 14

Chia 2	Dư
14	0
7	
3	1
1	1
0	

Chuyển đổi phần thập phân 0.125

Nhân 2	Phần nguyên
0.125	
0.25	0
0.5	0
1	1

Vậy  $14.125 = 11110.001$

Chuyển đổi 0.2 sang hệ nhị phân:

Nhân 2	Phần nguyên
0.2	
0.4	0
0.8	0
1	
1	

Ta thấy rằng số 0.2 trong hệ cơ số 2 là một số thập phân vô hạn tuần hoàn  $0.210 = 0.(0011)_2$

**Chuyển từ hệ bất kỳ sang hệ thập phân**

Giả sử ta có biểu diễn số B theo cơ số H là

$$B = b_n b_{n-1} b_{n-2} \dots b_1 b_0 . c_1 c_2 c_n \dots c_m$$

Vì ta đã quen tính toán với hệ cơ số 10 nên ta có thể chuyển đổi trực tiếp theo công thức sau:

$$B = b_n \times H_n + b_{n-1} \times H_{n-1} + b_{n-2} \times H_{n-2} + \dots + b_1 \times H + b_0 + c_1 \times H_{-1} + c_2 \times H_{-2} + \dots + c_m \times H_{-m}$$

(Ta hoàn toàn có thể áp dụng quy tắc đã nêu: chia lấy phần dư, nhân lấy phần nguyên... để tìm biểu diễn của B trong hệ thập phân)

**Chuyển từ hệ nhị phân sang bát phân (hoặc thập lục phân)**

Qui tắc: Nhóm các Bit thành từng nhóm 3 Bit (4 Bit - cho hệ thập lục phân) bắt đầu từ Bit ngoài cùng bên phải, tính giá trị số học quy luật giá trị vị trí riêng cho từng nhóm 3 (hay 4) Bit, viết các giá trị này liền nhau.

Ví dụ cho số nhị phân 11110101 chuyển số này sang dạng bát phân và thập lục phân.

(11 110 101) -> 365 trong hệ bát phân là số 365

(1111 0101) -> 15 5 -> F5 trong hệ thập lục phân là số F5

Khi cần chuyển ngược lại chúng ta làm theo các bước tương tự

**Chuyển đổi hệ thống số dựa trên hệ 8 và hệ 16**

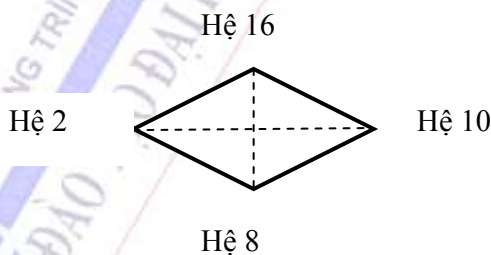
Trong phần bài giảng, chúng ta đã làm quen với cách chuyển đổi giữa hệ 2 và hệ 10. Tuy nhiên, ở những trị số lớn và dài thì làm cách trên trở nên rất phức tạp và dễ nhầm lẫn, ví dụ :

$$101110110101_{(2)} = ?_{(10)}$$

$$2997_{(10)} = ?_{(12)}$$

Trong ví dụ thứ nhất ta phải liên tiếp làm nhiều phép nhân và ở ví dụ thứ hai, ta lại thực hiện nhiều phép chia liên tiếp.

Người ta đưa ra hệ thống số trung gian là hệ 8 và hệ 16 để giải quyết:



**Thông qua hệ 8 và hệ 16 để chuyển đổi hệ 2 sang hệ 10**

Chia số nhị phân làm thành từng bộ 3 số và 4 số liên tiếp theo thứ tự tương ứng với cách thông qua hệ 8 và hệ 16 và dùng phương pháp nhân với các thừa số bên trên tương ứng rồi cộng lại.

Ví dụ:  $101110110101_{(2)} = ?_{(10)}$

THÔNG QUA HỆ 8: Chia số nhị phân từng bộ 3 số:

8 <sup>3</sup>			8 <sup>2</sup>			8 <sup>1</sup>			8 <sup>0</sup>		
2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
1	0	1	1	1	0	1	1	0	1	0	1
5			6			6			5		

Chú ý:  $5 = 1x2^2 + 0x2^1 + 1x2^0$  và  $6 = 1x2^2 + 1x2^1 + 0x2^0$

Kết quả:

$$101110110101_{(2)} = 5x8^3 + 6x8^2 + 6x8^1 + 5x8^0 = 5x512 + 6x64 + 6x8 + 5x1 = 2997_{(10)}$$

**THÔNG QUA HỆ 16: Chia số nhị phân thành bộ 4 số**

	16				16				16		
	2				1				0		
2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
1	0	1	1	1	0	1	1	0	1	0	1
	11				11				5		

Chú ý:  $11 = 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0$  và  $5 = 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0$

Kết quả:

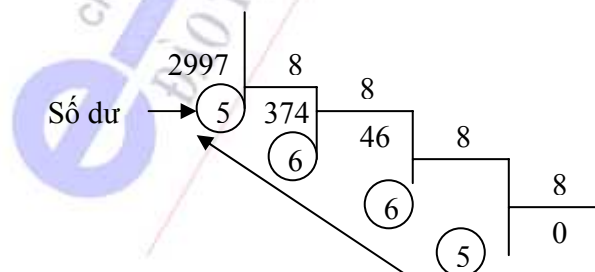
$$101110110101_{(2)} = 11x16^2 + 11x16^1 + 5x16^0 = 11x256 + 11x16 + 5x1 = 2997_{(10)}$$

**Thông qua hệ 8 và hệ 16 để chuyển hệ 10 sang hệ 2**

Cách làm tương tự như trên, nhưng thay phép nhân thành phép chia và lấy các số dư của phép chia ngược từ dưới lên trên để chuyển đổi.

Ví dụ:  $2997_{(10)} = ?_{(2)}$

**THÔNG QUA HỆ 8:**



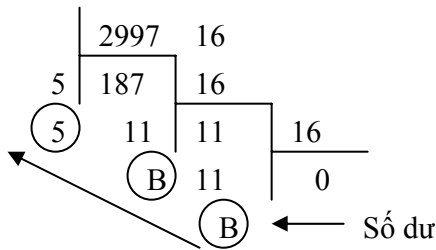
Ta có:  $(5)_{(8)} = 4 + 1 = 1x2^2 + 0x2^1 + 1x2^0 = 101_{(2)}$

Tương tự:  $(6)_{(8)} = 4 + 2 = 1x2^2 + 1x2^1 + 0x2^0 = 110_{(2)}$

Suy ra:

$$2997_{(10)} = 101\ 110\ 110\ 101_{(2)}$$

THÔNG QUA HỆ 16: \_\_\_\_\_



Ta có :  $2997_{(10)} = BB5_{(16)}$

$$B \text{ (hệ 16)} = 11 = 8 + 2 + 1 = 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0 = 1011 \text{ (hệ 2)}$$

$$5 \text{ (hệ 16)} = 4 + 1 = 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 0101 \text{ (hệ 2)}$$

Suy ra:  $2997_{(10)} = BB6_{(16)} = 1011\ 1011\ 0101_{(2)}$

**Chuyển hệ 8 sang hệ 16 và ngược lại:**

Ta có thể dùng hệ 10 hoặc hệ 2 làm trung gian để chuyển đổi hệ 8 sang hệ 16 và ngược lại. Thông thường dùng hệ 2 để trung chuyển có thuận lợi hơn.

Ví dụ:  $5665_{(8)} = ?_{(16)}$

Cách làm như sau:

**Bước 1: Chuyển hệ 8 thành hệ 2: biểu thị từng trị số trong hệ 8 thành từng nhóm 3 số và ghép các nhóm đó lại.**

$$5 \text{ (hệ 8)} = 4 + 1 + 0 = 1x2^2 + 0x2^1 + 1x2^0 = 101 \text{ (hệ 2)}$$

$$6 \text{ (hệ 8)} = 4 + 2 + 2 = 1x2^2 + 1x2^1 + 0x2^0 = 110 \text{ (hệ 2)}$$

Vậy  $5665_{(8)} = 101\ 110\ 110\ 101_{(2)}$

**Bước 2: Chia dãy số hệ 2 vừa có được thành các bộ 4 số và chuyển các bộ đó sang hệ 16**

$$5665_{(8)} = 101\ 110\ 110\ 101_{(2)} = 1011\ 1011\ 0101_{(2)}$$

$$\text{Vi: } 1011_{(2)} = 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0 = 8 + 0 + 2 + 1 = 11 = B_{(16)}$$

$$0101_{(2)} = 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 0 + 4 + 0 + 1 = 5_{(16)}$$

Nên:

1011	1011	1010
B	B	5

Vậy:  $5665_{(8)} = BB5_{(16)}$

Việc chuyển từ hệ 16 sang hệ 8 ta cũng tiến hành 2 bước như vậy.

**b. Số học nhị phân**

Chương 1: Các khái niệm cơ bản

Trong số học nhị phân chúng ta cũng có 4 phép toán cơ bản như trong số học thập phân là cộng, trừ, nhân và chia. Qui tắc của 2 phép tính cơ bản cộng và nhân:

X	Y	X + Y	X * Y
0	0	0	0
0	1	1	0
1	0	1	0
1	1	10	1

Ghi chú: Với phép cộng trong hệ nhị phân,  $1 + 1 = 10$ , số 10 (đọc là một - không) chính là số 2 tương đương trong hệ thập phân. Viết 10 có thể hiểu là viết 0 nhớ 1. Một cách tổng quát, khi cộng 2 hay nhiều chữ số nếu giá trị tổng lớn hơn cơ số b thì ta viết phần lẻ và nhớ phần lớn hơn sang bên trái cạnh nó.

Ví dụ: Cộng 2 số  $0101 + 1100 = ?$

$$\begin{array}{r} 0101 \\ + 1100 \\ \hline 10001 \end{array}$$
 tương đương số 5 trong hệ 10  
 tương đương số 12 trong hệ 10  
 tương đương số 17 trong hệ 10

Ví dụ: Nhân 2 số  $0110 \times 1011 = ?$

$$\begin{array}{r} 0110 \\ \times 1011 \\ \hline 0110 \\ 0110 \\ + 0000 \\ \hline 1000010 \end{array}$$
 tương đương số 6 trong hệ 10  
 tương đương số 11 trong hệ 10  
 tương đương số 66 trong hệ 10

Phép trừ và phép chia là các phép toán đặc biệt của phép cộng và phép nhân.

Ví dụ: Trừ hai số

$$\begin{array}{r} 101 \\ - 011 \\ \hline 010 \end{array}$$
 tương đương số 5 trong hệ 10  
 tương đương số 3 trong hệ 10  
 tương đương số 2 trong hệ 10

Ghi chú:  $0 - 1 = -1$  (viết 1 và mượn 1 ở hàng bên trái).

Ví dụ: Chia hai số

$$\begin{array}{r|l} 110 & 10 \\ - 10 & 11 \\ \hline 010 & \\ - 10 & \\ \hline 00 & \end{array}$$
 tương đương số 6 và trong hệ 10

tương đương số 3 trong hệ 10

Quy tắc 1: Khi nhân một số nhị phân với  $2^n$  ta thêm n số 0 vào bên phải số nhị phân đó.

Ví dụ :  $1011 \times 2^3 = 1011000$

Quy tắc 2: Khi chia một số nguyên nhị phân cho  $2^n$  ta đặt dấu chấm ngăn ở vị trí n chữ số bên trái kể từ số cuối của số nguyên đó.

Ví dụ :  $100111110 : 2^3 = 100111.110$

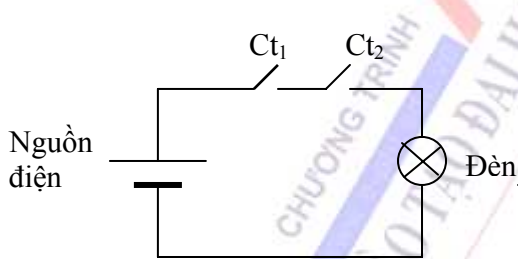
**c. Mệnh đề logic**

Mệnh đề logic là mệnh đề chỉ nhận một trong 2 giá trị : Đúng (TRUE) hoặc Sai (FALSE), tương đương với TRUE = 1 và FALSE = 0.

Quy tắc:  $TRUE = NOT FALSE$  và  $FALSE = NOT TRUE$

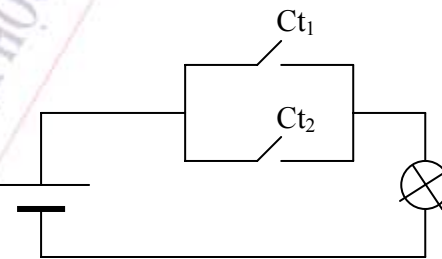
Phép toán logic áp dụng cho 2 giá trị TRUE và FALSE ứng với tổ hợp AND (và) và OR (hoặc) như sau:

x	y	x AND y	x OR y
TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE
FALSE	TRUE	FALSE	TRUE
FALSE	FALSE	FALSE	FALSE



Đèn sáng = [ct1+] AND [ct2+]

Đèn tắt = [ct1-] OR [ct2-]



Đèn sáng = [ct1+] OR [ct2+]

Đèn tắt = [ct1-] AND [ct2-]

Ký hiệu:  
 Ct: công tắc  
 + : đóng (on)  
 - : ngắt (off)

**1.1.2.2. Biểu diễn dữ liệu**

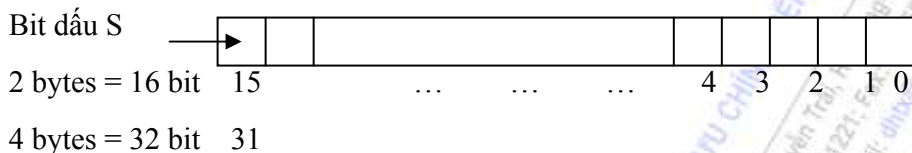
Dữ liệu số trong máy tính gồm có số nguyên và số thực.

**a/ Biểu diễn số nguyên**

Số nguyên gồm số nguyên không dấu và số nguyên có dấu.

\* Số nguyên không dấu là số không có bit dấu như 1 byte = 8 bit, có thể biểu diễn  $2^8 = 256$  số nguyên dương, cho giá trị từ 0 (0000 0000) đến 255 (1111 1111).

\* Số nguyên có dấu thể hiện trong máy tính ở dạng nhị phân là số dùng 1 bit làm bit dấu, người ta qui ước dùng bit ở hàng đầu tiên bên trái làm bit dấu (S): 0 là số dương và 1 cho số âm. Đơn vị chiều dài để chứa thay đổi từ 2 đến 4 bytes.



Ta thấy, với chiều dài 16 bit : bit đầu là bit dấu và 15 bit sau là bit số

Trị dương lớn nhất của dãy 2 bytes sẽ là:  $01111111 11111111 = 2^{15} - 1$

Trị âm lớn nhất trong dãy 2 bytes là  $-2^{15}$

Để thể hiện số âm trong hệ nhị phân ta có 2 khái niệm:

- Số bù 1: Khi đảo ngược tất cả các bit của dãy số nhị phân: 0 thành 1 và 1 thành 0, dãy số đảo đó gọi là số bù 1 của số nhị phân đó.

Ví dụ:  $N = 0101 = 5_{(10)}$

Số bù 1 của N là: 1010

- Số bù 2: Số bù 2 của số N là số đảo dấu của nó (-N). Trong hệ nhị phân, số bù 2 được xác định bằng cách lấy số bù 1 của N rồi cộng thêm 1.

Ví dụ:  $N = 0101 = 5_{(10)}$

Số bù 1 của N là: 1010

$$\text{Số bù 2 của N là: } \begin{array}{r} 1010 \\ + 0001 \\ \hline 1011 \end{array} = -5_{(10)} = -N$$

### b/ Biểu diễn số thực

Đối với các số thực (real number) là số có thể có cả phần lẻ hoặc phần thập phân. Trong máy tính, người ta biểu diễn số thực với số dấu chấm tĩnh (fixed point number) và số dấu chấm động (floating point number).

\*/ Số dấu chấm tĩnh: thực chất là số nguyên (integers) là những số không có chấm thập phân

\*/ Số dấu chấm động: là số có chữ số phần lẻ không cố định. Mỗi số như vậy có thể trữ và xử lý trong máy tính ở dạng số mũ.

Ví dụ:  $499,000,000 = 499 \times 10^6 = 49.9 \times 10^7 = 0.499 \times 10^9 = 0.499E + 09$

$0.000 123 = 123 \times 10^{-6} = 1.23 \times 10^{-4} = 0.123 \times 10^{-3} = 0.123E - 03$



Ghi chú: Dấu chấm thể hiện trong máy tính để phân biệt phần lẻ, dấu phẩy tượng trưng cho phần ngàn, được viết theo qui ước của Mỹ.

Tổng quát, số dấu chấm động được biểu diễn theo 3 phần :

- phần dấu S (sign) : 0 cho + và 1 cho -
- phần định trị m (mantissa)
- phần mũ e (exponent), có thể là số nguyên dương (+) hoặc âm (-)

với một số X bất kỳ, có thể viết :

$$X = \pm m \cdot b^e = \pm m E e$$

Trong đó, b là cơ số qui ước, trị số mũ e có thể thay đổi tùy theo số vị trí cần dịch chuyển dấu chấm để có lại trị số ban đầu. Khi dịch chuyển dấu chấm sang  $\pm n$  vị trí về phía trái (+n) hay phía phải (-n) thì số mũ e thay đổi lên  $\pm n$  đơn vị tương ứng

Để biểu diễn số có dấu chấm động, người ta dùng dãy 32 bit với hệ thống cơ số 16. Trong đó, 1 bit cho phần dấu, 7 bit cho phần mũ để biểu diễn phần đặc trị C (characteristic) và 24 bit cho phần định trị m.

S	C	m
dấu	Phần mũ	Phần định trị
1 bit	7 bit	24 bit

Phần mũ có 7 bit =  $2^7 = 128$  đặc trị C, tương ứng phần mũ e từ -64 đến +63

$$C = \text{số mũ biểu diễn} + 64$$

Phần mũ e	-64	-63	-62	...	-2	-1	0	1	...	62	63
Đặc trị C	0	1	2	...	62	63	64	65	...	126	127

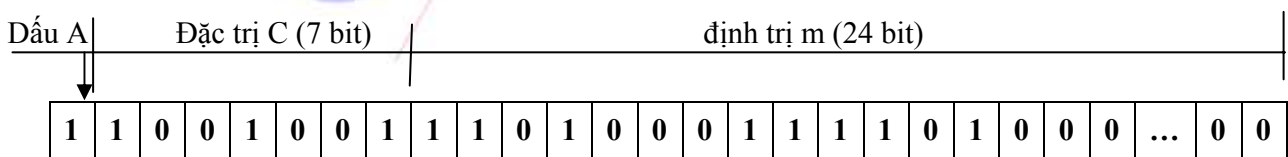
Ví dụ:  $A = -419.8125_{(10)} = -110100011.1101_{(2)} = -0.1101000111101 \times 2^9$

Số mũ của A là 9, số đặc trị C là:

$$C = 9 + 64 = 73 = 1001001_{(2)}$$

Trong máy tính, số A sẽ được trữ theo vị trí nhớ 32 bit như sau :

Dấu A      đặc trị C (7bit)                      định trị m (24 bit)



**c. Biểu diễn ký tự**

Để có thể biểu diễn các ký tự như chữ cái in và thường, các chữ số, các ký hiệu... trên máy tính và các phương tiện trao đổi thông tin khác, người ta phải lập ra các bộ mã (code system) qui ước khác nhau dựa vào việc chọn tập hợp bao nhiêu bit để diễn tả 1 ký tự tương ứng, ví dụ các hệ mã phổ biến :

- Hệ thập phân mã nhị phân BCD (Binary Coded Decima) dùng 6 bit.

- Hệ thập phân mã nhị phân mở rộng EBCDIC (Extended Binary Coded Decimal Interchange Code) dùng 8 bit tương đương 1 byte để biểu diễn 1 ký tự.

- Hệ chuyển đổi thông tin theo mã chuẩn của Mỹ ASCII (American Standard Code for Information Interchange) là hệ mã thông dụng nhất hiện nay trong kỹ thuật tin học. Hệ mã ASCII dùng nhóm 7 bit hoặc 8 bit để biểu diễn tối đa 128 hoặc 256 ký tự khác nhau và mã hóa theo ký tự liên tục theo cơ số 16.

Hệ mã ASCII 7 bit, mã hoá 128 ký tự liên tục như sau:

0	:	NUL (ký tự rỗng)
1 - 31	:	31 ký tự điều khiển
32 - 47	:	các dấu trống SP (space) ! # \$ % & ( ) * + , - . /
48 - 57	:	ký số từ 0 đến 9
58 - 64	:	các dấu : ; < = > ? @
65 - 90	:	các chữ in hoa từ A đến Z
91 - 96	:	các dấu [ \ ] _ `
97 - 122	:	các chữ thường từ a đến z
123 - 127	:	các dấu {   } ~ DEL (xóa)

Hệ mã ASCII 8 bit (ASCII mở rộng) có thêm 128 ký tự khác ngoài các ký tự nêu trên gồm các chữ cái có dấu, các hình vẽ, các đường kẻ khung đơn và khung đôi và một số ký hiệu đặc biệt (xem phụ lục).

- Hệ chuyển đổi thông tin theo bộ mã Unicode: Ngày nay máy tính đã toàn cầu hóa, mà hình ảnh cụ thể là mạng Internet, do vậy bảng mã ASCII đã bộc lộ khả năng mã hóa hạn chế của nó.

Để thống nhất bộ mã trên toàn thế giới, người ta đã đề xuất bộ mã 16 bit mang tên Unicode. Vì dùng tới 16 bit để mã hóa (mã hóa được  $2^{16}$  ký tự), vì vậy nó đủ lớn để đáp ứng cho việc mã hóa tất cả các ngôn ngữ trên toàn thế giới. Đặc điểm chính của Unicode là nó không chứa các ký tự điều khiển mà dành tất cả để mã hóa ký tự. Bảng sau đây cho chúng ta biết sơ bộ cách phân bố mã chuẩn trong Unicode:

Mã thập phân	Kí tự
0 đến 8191	Chữ cái Anh, Latin1, Châu Âu, Latin mở rộng, chữ cái phiên âm, Hy Lạp, Nga, Armerical, Do thái, Ả rập, Ethiopi, Dvanagari, bengali, Gurmukhi, Gujarati, Orya, Tamil, Telugu, Kanada, Malaixia, Thái, Lào, Miền điện, Khme, Tây tạng, Mông cổ, Georgi
	Kí hiệu

	Chữ tượng hình, chữ cái Hán, chữ Nhật, Hàn
8192-12287	Chữ tượng hình Hán, Nhật, Hàn
12288-16383	Dành cho người sử dụng
16384-59391	Vùng tương thích
59392-65024	Cho các mục đích trong tương lai
65025-65036	

8192 giá trị đầu dành cho chữ cái chuẩn;

4096 giá trị tiếp theo dành cho kí tự toán học, kỹ thuật,...

...

Unicode qui định các chữ cái có âm tiết trong tiếng Việt là các kí tự tổ hợp. Ví dụ chữ “â” là tổ hợp của hai chữ ‘a’ và ‘^’; mỗi kí tự tổ hợp bao gồm nguyên âm cơ sở được nối tiếp bởi kí tự dấu thanh. Nguyên âm cơ sở và dấu thanh được đặt vào cùng vị trí khi hiển thị. Nếu chữ cái được tổ hợp từ hai hay nhiều kí tự âm tiết (ví dụ ‘â’) thứ tự các dấu không quan trọng nếu không có luật chính tả cụ thể.

Các kí tự tổ hợp từ trước như chữ ‘đ’ chỉ dùng một mã duy nhất để mô tả.

Để biểu diễn tiếng Việt ta cần :

- 33 chữ cái hoa
- 33 chữ cái thường
- 5 dấu thanh: huyền ( ` ), ngã ( ~ ), hỏi ( ? ), nặng ( . ), sắc ( ´ )

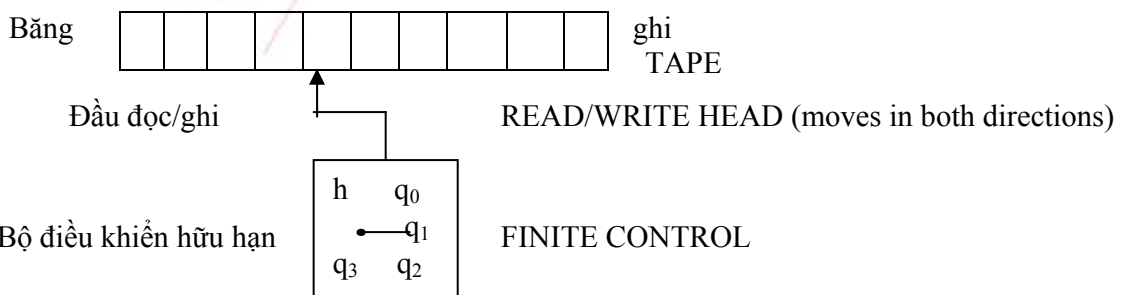
## 1.2. CẤU TRÚC TỔNG QUÁT CỦA HỆ THỐNG MÁY TÍNH

### 1.2.1. Nguyên lý thiết kế cơ bản

#### 1.2.1.1. Nguyên lý Turing

Alan Mathison Turing (1912 - 1954) là một nhà toán học người Anh đã đưa ra một thiết bị tính đơn giản gọi là máy Turing. Về lý thuyết, mọi quá trình tính toán nếu thực hiện được thì đều có thể mô phỏng lại trên máy Turing. Máy Turing gồm có (xem hình 2.1):

- Một bộ điều khiển trạng thái hữu hạn (finite control), trong đó có các trạng thái đặc biệt như trạng thái khởi đầu và trạng thái kết thúc.
- Một băng ghi (tape) chứa tín hiệu trong các ô.
- Một đầu đọc (head) và ghi có thể di chuyển theo 2 chiều trái hoặc phải một đơn vị.



## Sơ đồ máy Turing

Đầu đọc/ghi mang chức năng thông tin nối giữa Bộ điều khiển hữu hạn và băng ghi. Đầu băng cách đọc dấu hiệu từ băng và cũng dùng nó để thay đổi dấu hiệu trên băng. Bộ kiểm soát vận hành theo từng bước riêng biệt; mỗi bước nó thực hiện 2 chức năng tùy thuộc vào trạng thái hiện tại của nó và tín hiệu hiện tại của băng:

1. Đặt bộ điều khiển ở trạng thái ban đầu  $q_1$ , băng trắng và đầu đọc/ghi chỉ vào ô khởi đầu.
2. Nếu: (a) trạng thái hiện tại  $q$  trùng với trạng thái kết thúc  $q_0$  thì máy sẽ dừng.

(b) ngược lại, trạng thái  $q$  sẽ chuyển qua  $q'$ , tín hiệu trên băng  $s$  thành  $s'$  và đầu đọc dịch chuyển sang phải hoặc trái một đơn vị. Máy hoàn thành xong một bước tính toán và sẵn sàng cho bước tiếp theo.

### 1.2.1.2. Nguyên lý Von Neumann

Năm 1946, nhà toán học Mỹ John Von Neumann (1903 - 1957) đã đề ra một nguyên lý máy tính hoạt động theo một chương trình được lưu trữ và truy nhập theo địa chỉ. Nguyên lý này được trình bày ở một bài báo nổi tiếng nhan đề: Thảo luận sơ bộ về *thiết kế logic của máy tính điện tử*. Nội dung nguyên lý Von Neumann gồm:

- *Máy tính có thể hoạt động theo một chương trình đã được lưu trữ.*

Theo Von Neumann, chúng ta có thể tập hợp các lệnh cho máy thi hành theo một chương trình được thiết kế và coi đó như một tập dữ liệu. Dữ liệu này được cài vào trong máy và được truyền bằng xung điện. Đây là một cuộc cách mạng mới cho máy tính nhằm tăng tốc độ tính toán vào thời đó vì trước kia máy chỉ có thể nhận được các lệnh từ băng giấy hoặc bìa đục lỗ và nạp vào bằng tay. Nếu gặp bài toán lặp lại nhiều lần thì cũng tiếp tục bằng cách nạp lại một cách thủ công như vậy gây hạn chế trong tính toán sử dụng.

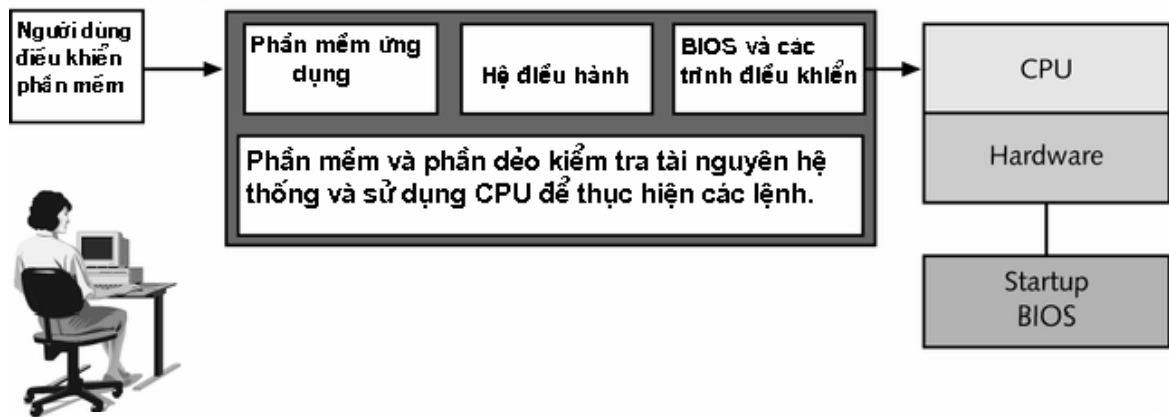
- *Bộ nhớ được địa chỉ hóa*

Mỗi dữ liệu đều có một địa chỉ của vùng nhớ chứa số liệu đó. Như vậy để truy nhập dữ liệu ta chỉ cần xác định địa chỉ của nó trên bộ nhớ.

- *Bộ đếm của chương trình*

Nếu mỗi câu lệnh phải dùng một vùng nhớ để chứa địa chỉ của câu lệnh tiếp theo thì không gian bộ nhớ sẽ bị thu hẹp. Để khắc phục hạn chế này, máy được gắn một thanh ghi để chỉ ra vị trí của lệnh tiếp theo cần được thực hiện và nội dung của nó tự động được tăng lên mỗi lần lệnh được truy cập. Muốn đổi thứ tự lệnh ta chỉ cần thay đổi nội dung thanh ghi bằng một địa chỉ của lệnh cần được thực hiện tiếp.

### 1.2.2. Quá trình xử lý thông tin



#### Máy tính là công cụ xử lý thông tin

Máy tính là công cụ xử lý thông tin. Về cơ bản, quá trình xử lý thông tin trên máy tính - cũng như quá trình xử lý thông tin của con người - có 4 giai đoạn chính :

- *Nhận thông tin (Receive input)*: thu nhận thông tin từ thế giới bên ngoài vào máy tính. Thực chất đây là quá trình chuyển đổi các thông tin ở thế giới thực sang dạng biểu diễn thông tin trong máy tính thông qua các thiết bị đầu vào.

- *Xử lý thông tin (process information)*: biến đổi, phân tích, tổng hợp, tra cứu... những thông tin ban đầu để có được những thông tin mong muốn.

- *Xuất thông tin (produce output)* : đưa các thông tin kết quả (đã qua xử lý) ra trở lại thế giới bên ngoài. Đây là quá trình ngược lại với quá trình ban đầu, máy tính sẽ chuyển đổi các thông tin trong máy tính sang dạng thông tin ở thế giới thực thông qua các thiết bị đầu ra.

- *Lưu trữ thông tin (store information)*: ghi nhớ lại các thông tin đã được ghi nhận để có thể đem ra sử dụng trong những lần xử lý về sau.

Để đáp ứng 4 thao tác đó thì một máy tính thông thường cũng gồm bốn thành phần hợp thành, mỗi thành phần có một chức năng riêng:

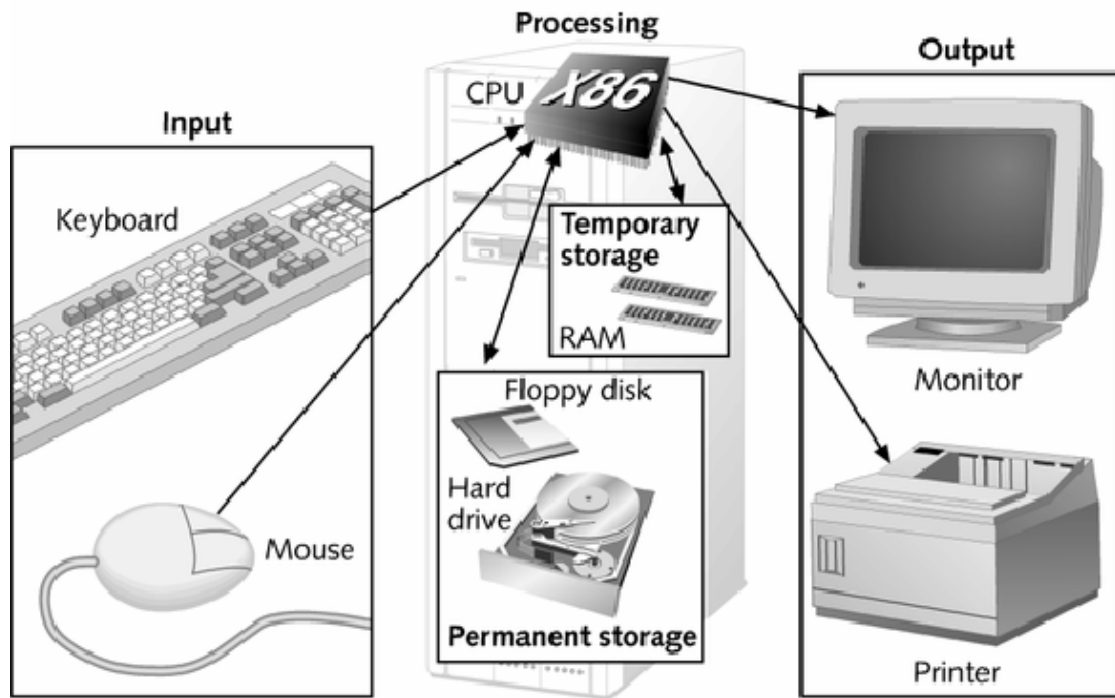
- *Thiết bị nhập (input device)* : thực hiện thao tác đưa dữ liệu từ thế giới bên ngoài vào, thường là bàn phím và con chuột, nhưng cũng có thể là các loại thiết bị khác mà ta sẽ nói rõ hơn ở những phần sau.

- *Thiết bị xử lý* : hay đơn vị xử lý trung tâm - CPU thực hiện thao tác xử lý, tính toán các kết quả, điều hành hoạt động tính toán của máy vi tính, có thể xem CPU như một bộ não của con người.

- *Thiết bị xuất (Output)* thực hiện thao tác gởi thông tin ra ngoài máy vi tính, hầu hết là dùng màn hình máy tính là thiết bị xuất chuẩn, có thể thêm một số khác như máy in.

- *Thiết bị lưu trữ (storage devices)* được dùng để cất giữ thông tin. Lưu trữ sơ cấp (primary memory) là bộ nhớ trong của máy tính dùng để lưu các tập lệnh của chương trình, các thông tin dữ liệu sẵn sàng trong tư thế chuẩn bị làm việc ty theo yêu cầu của CPU. Lưu trữ thứ cấp

(secondary storage) là cách lưu trữ đơn thuần với mục đích cất giữ dữ liệu, cách này dùng các thiết bị như đĩa cứng, đĩa mềm, CD,..



Quá trình xử lý thông tin trên máy tính

### 1.2.3. Cấu trúc tổng quát của MTĐT

Mỗi loại máy tính có thể có các hình dạng hoặc cấu trúc khác nhau tùy theo mục đích sử dụng. Tuy nhiên, một máy tính muốn hoạt động được phải hội tụ đủ các yếu tố sau:

- Phần cứng: bao gồm các thiết bị vật lý mà người dùng có thể quan sát được. Đó là các bảng mạch điện tử được lắp ghép lại với nhau và được cung cấp điện năng để hoạt động. Phần cứng máy tính thường được chia ra làm ba phần cơ bản - đó là: Thiết bị nhập, thiết bị xử lý và thiết bị xuất

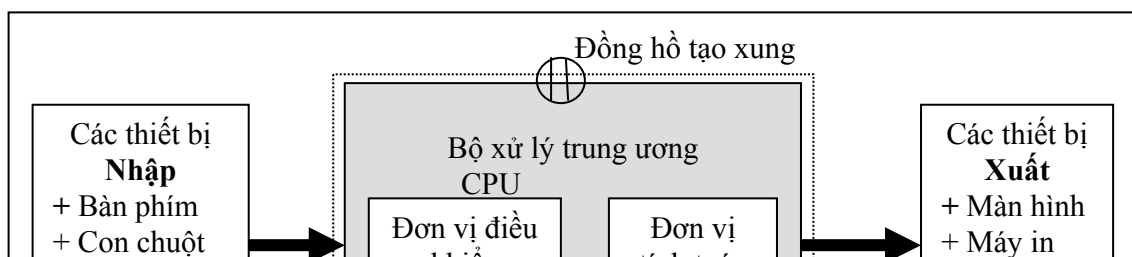
- Phần mềm: bao gồm các chương trình được viết bởi các nhà lập trình nhằm mục đích điều khiển các mạch điện tử cũng như thực hiện các phép tính toán. Phần mềm thường chia làm ba loại cơ bản - đó là: Hệ điều hành, phần mềm ứng dụng và phần mềm tiện ích.

#### 1.2.3.1. Phần cứng (Hardware)

Phần cứng có thể được hiểu đơn giản là tất cả các phần trong một hệ máy tính mà chúng ta có thể thấy hoặc sờ được. Phần cứng gồm các thiết bị máy có thể thực hiện các chức năng sau:

- \* Nhập dữ kiện vào máy (input)
- \* Xử lý dữ kiện (processing)
- \* Xuất dữ kiện/ thông tin (output)

#### Sơ đồ cấu trúc phần cứng



## Sơ đồ cấu trúc phần cứng

### *a/ Bộ xử lý trung tâm (CPU - Central Processing Unit)*

CPU là đơn vị xử lý trung tâm, hay còn gọi là bộ vi xử lý - đây là bộ phận đầu não của máy tính, nó thực hiện các lệnh, tính toán và điều khiển các phần cứng. CPU là một vi mạch làm bằng Silicon và có kích cỡ không bằng đầu ngón tay cái. Được bọc trong một lớp vỏ màu đen, CPU được gắn vào bảng mạch chính thông qua giao diện SOCKET hoặc SLOT1. Vi mạch tích hợp này chứa đựng các Transistor (các công tắc bán dẫn mini), các điện trở chỉnh lưu dòng điện và các tụ điện lưu dòng điện. Bộ CPU thực sự đầu tiên là Intel 4004 được sản xuất năm 1971. Hãng Intel được thành lập năm 1968 là hãng chuyên sản xuất các chip bộ nhớ. Sau khi bộ vi xử lý 8088 của họ được chọn sử dụng cho máy IBM-PC thì doanh thu của hãng tăng cao. Năm 1993 hãng Intel đưa ra chip Pentium với trên 3 triệu transistor và một đường truyền dữ liệu 64 bit. Giống với 80486, Pentium cũng có bộ nhớ đệm Cache dùng để chứa dữ liệu cho đến khi dữ liệu được xử lý và một chip đồng xử lý toán học. Thêm vào đó, Pentium còn chứa 2 đường dẫn lệnh cho phép nó xử lý 2 lệnh chương trình cùng một lúc với khoảng thời gian gần bằng thời gian xử lý một lệnh. Một số chương trình - đặc biệt là những chương trình video và đồ họa khi chạy trên máy Pentium sẽ nhanh gấp 2 lần so với chạy trên 80486. Tốc độ xử lý của các CPU thường được đo bằng Megahertz (= 1 triệu chu kỳ/giây). CPU có 3 bộ phận chính: khối điều khiển, khối tính toán số học và logic, và một số thanh ghi.

· *Khối điều khiển (CU: Control Unit) là trung tâm điều hành máy tính. Nó có nhiệm vụ giải mã các lệnh, tạo ra các tín hiệu điều khiển công việc của các bộ phận khác của máy tính theo yêu cầu của người sử dụng hoặc theo chương trình đã cài đặt.*

· *Khối tính toán số học và logic (ALU: Arithmetic-Logic Unit) bao gồm các thiết bị thực hiện các phép tính số học (cộng, trừ, nhân, chia, ...), các phép tính logic (AND, OR, NOT, XOR) và các phép tính quan hệ (so sánh lớn hơn, nhỏ hơn, bằng nhau, ...)*

· Các thanh ghi (registers) được gắn chặt vào CPU bằng các mạch điện tử làm nhiệm vụ bộ nhớ trung gian. Các thanh ghi mang các chức năng chuyên dụng giúp tăng tốc độ trao đổi thông tin trong máy tính.

Ngoài ra, CPU còn được gắn với một đồng hồ (clock) hay còn gọi là bộ tạo xung nhịp. Tần số đồng hồ càng cao thì tốc độ xử lý thông tin càng nhanh. Thường thì đồng hồ được gắn tương xứng với cấu hình máy và có các tần số dao động (cho các máy PC 386 DX trở lên) là 33 MHz, 66 MHz, 100 MHz, 120 MHz, 133 MHz, ... hoặc cao hơn.

#### **Các thông số quan trọng của CPU:**

1- **Tốc độ:** Tốc độ của CPU được tính bằng tần số xung điện trong lõi của CPU. Tần số xung điện được tính theo đơn vị MHz. Đây cũng chính là tốc độ tính toán của CPU dựa vào các phép toán cơ bản là +/- . Ví dụ: CPU có tốc độ 100 MHz là 100.000.000 Phép tính/giây.

2- **Dung lượng bộ nhớ đệm (Cache Size):** Quyết định tốc độ xử lý lệnh nhanh hay chậm. Đây là một loại RAM tốc độ nhanh được tích hợp sẵn trong CPU. Dung lượng Cache thường có là: 32 KB, 64 KB, 128 KB, 256 KB, 512 KB.

3- **Tốc độ BUS (Đường truyền dữ liệu):** Tính bằng MHz. Đây là tốc độ được tính toán thích hợp để CPU và Mainboard có thể giao tiếp được với nhau. Tốc độ BUS thường được gọi là CLOCK và đồng thời cũng là BUS giao tiếp giữa Mainboard và CPU. Các chỉ số CLOCK thường là: 50/ 60/ 66/ 75/ 83/ 100/ 133/ 150/ 200/ 253/ 400/ 800.

4- **Điện áp lõi của CPU:** Điện áp sử dụng cho các linh kiện bên trong CPU. Đây là điện áp có độ ổn định cao để CPU có thể hoạt động ổn định. Hiện tại các mức điện áp thường từ 1.27 và 2.9 v.

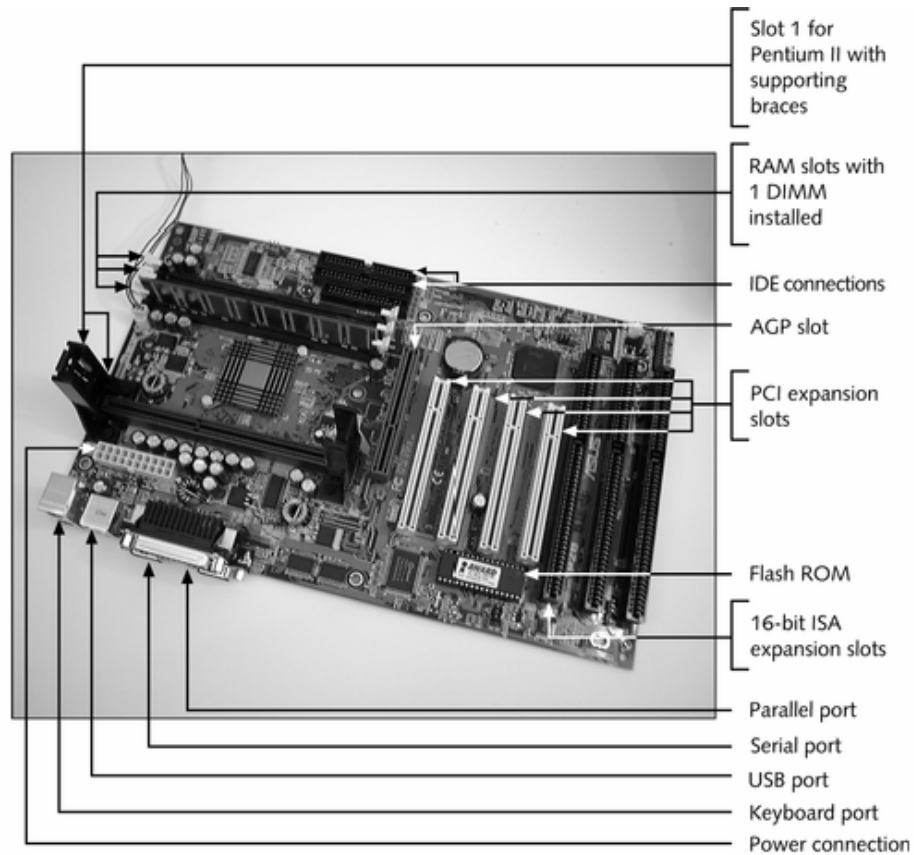
Tùy thuộc vào thời gian ra đời của các loại CPU, các nhà sản xuất Mainboard thiết kế bảng mạch cho phép người dùng lựa chọn CPU có tốc độ thích hợp với nhu cầu công việc, thích hợp với khả năng tài chính của mình. Để có thể xác định được đúng loại CPU mà Mainboard cho phép cắm, ta phải biết được một số thông số như : đế hoặc khe cắm (SOCKET/SLOT1), tốc độ CPU, tần số BUS của Mainboard (Clock), tần số BUS giao diện của CPU, qui định về thiết lập hệ số nhân xung (Radio, Jumper, Switch),... Thông thường, các đặc tính của Mainboard được hướng dẫn rất kỹ trong sách hướng dẫn đi kèm theo từng loại Mainboard, ta nên cố gắng tham khảo các thông tin của Mainboard trước khi lắp đặt một máy tính mới hoặc nâng cấp các thiết bị như CPU, RAM

#### **b/ Bo mạch chủ và bộ nhớ (Mainboard and Memory)**

##### **\*/ Mainboard (MotherBoard, System Board).**

Bảng mạch chính (MainBoard) hay còn gọi bảng mạch mẹ (MotherBoard) chứa các IC quan trọng nhất của hệ thống máy tính cá nhân bao gồm: CPU, RAM, ROM và một số IC và các mạch phụ trợ khác như khe cắm cho các vỉ mạch điều khiển ổ đĩa cứng, ổ đĩa mềm, modem, âm thanh, video...





### Mainboard và các thành phần

Trong đó các thành phần của Mainboard:

- Khe cắm RAM (RAM Slot):

SIMMs RAM Slot:

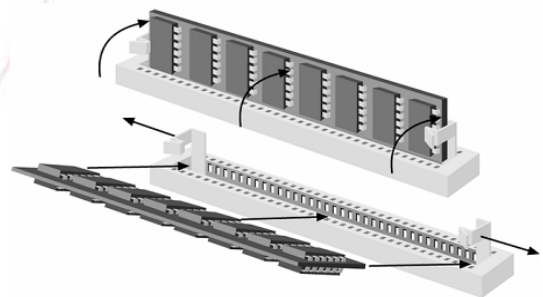
Khe cắm màu trắng có 30, 72 chân. Loại RAM này không còn trên thị trường. Khả năng nâng cấp dung lượng bộ nhớ tùy thuộc vào từng loại MAIN. Thường tối đa là (4 khe x 16 MB)

DIMMs RAM Slot:

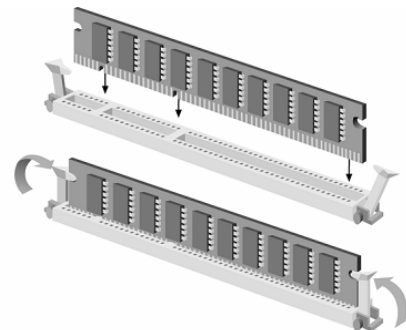
Khe cắm màu đen có 168 chân. Khả năng nâng cấp bộ nhớ tùy thuộc vào từng loại Mainboard khác nhau (số lượng khe cắm) và dung lượng tối đa thường khoảng 1GB.

- Khe cắm mở rộng:

+ ISA (Industrial Standard Architecture):



Khe cắm RAM SIMMS và cách cắm RAM



Khe cắm RAM DIMMS và cách cắm RAM



Độ rộng đường truyền 16 bits, sử dụng cho các loại thiết bị tốc độ chậm như các loại thẻ mạch (Card) màn hình, sound, modem 38k... Hiện nay các loại Mainboard mới theo tiêu chuẩn PC 99 các loại khe cắm này đã bị loại bỏ. Rất ít các loại Mainboard hỗ trợ cho loại khe cắm này.

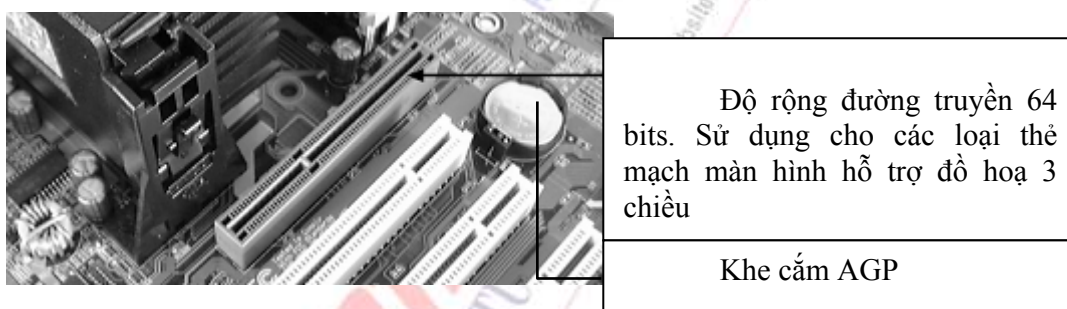
+ VESA Local Bus: Bị loại bỏ hoàn toàn vì thiết kế cồng kềnh.

+ PCI (Peripheral Component Interconnect):



Độ rộng đường truyền 32 bits. Hiện đang thông dụng trên thị trường. Sử dụng cho các thiết bị tốc độ nhanh.

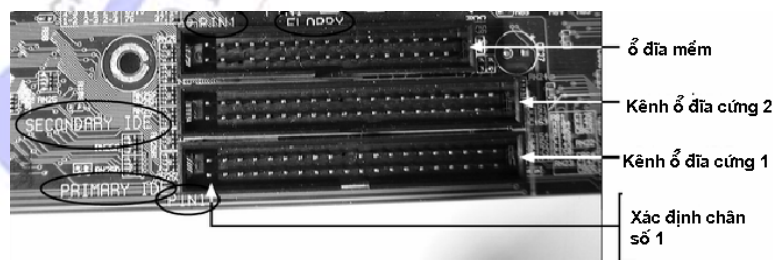
+ AGP (Accelerate Graphic Port):



+ Chipset

Mỗi hãng sản xuất đều đưa ra một loại Main với bộ chipset riêng. Chipsets quyết định khả năng tích hợp, nâng cấp các thiết bị như CPU, RAM. Thẻ mạch mở rộng. Các khả năng này được mô tả chi tiết trong sách hướng dẫn của nhà sản xuất đi kèm theo Mainboard. Các loại Chipset thông dụng là: VIA, INTEL, SIS .

+ Chân kết nối bộ nhớ ngoài



Chân kết nối cấp dữ liệu ổ đĩa mềm (Floppy) và ổ đĩa cứng, đĩa CD (IDE).

+ Chân kết nối ổ đĩa mềm:

1 bộ chân tích hợp trên thẻ mạch vào ra (I/O Card Đối với các hệ thống cũ), tích hợp trên Main (Đối với các hệ thống mới). 34 chân/bộ cho phép tích hợp 2 ổ đĩa mềm đồng thời.

+ Chân kết nối ổ đĩa cứng:

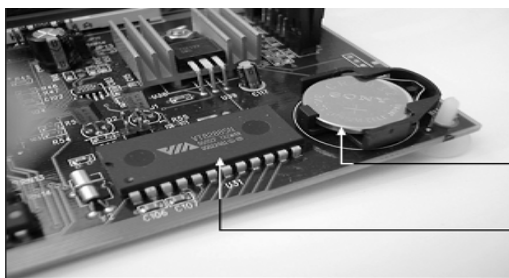
Tích hợp trên thẻ mạch vào ra ( I/O Card ) trên các hệ thống máy cũ. Trên các hệ thống máy mới, các bộ kết nối này được gọi là IDE/ EIDE. 40 chân/ bộ cho phép kết nối tối đa 4 ổ đĩa cứng/ CD ROM.

+ Chân kết nối các cổng vào ra:

- . Cổng bàn phím: kết nối bàn phím DIN 5 (AT), MiniDIN 6 (ATX)
- . Cổng truyền thông, tuần tự (COM): gồm 2 cổng COM 9 chân được sử dụng cho việc kết nối chuột và Modem.
- . Cổng song song (LPT, PRN): dành cho kết nối máy in.
- . Cổng đa năng (USB): dành cho kết nối các loại thiết bị hỗ trợ kiểu kết nối này.
- . Cổng chuột PS2: Dành riêng cho chuột PS2.

+ Chân kết nối bộ nguồn: 12 chân dành cho chuẩn AT, 20 chân dành cho chuẩn ATX.

- BIOS, CMOS RAM và Pin CMOS:



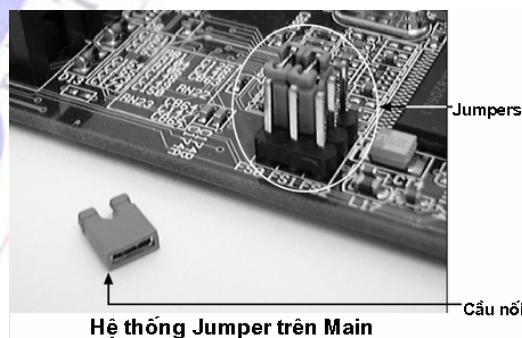
CMOS RAM Chip, Pin nuôi dữ liệu.



BIOS ROM của hãng AWARD

Thường đi kèm theo main. Là thiết bị lưu trữ bộ lệnh khởi động và điều khiển ngắt cứng của hệ thống. Khả năng nâng cấp tùy thuộc vào từng loại có thể bằng phần cứng (Thay chip BIOS) hoặc bằng phần mềm (FLASH BIOS).

+ Cầu nối (Jumper).



Dành cho người sử dụng để nâng cấp hệ thống. Xác định điện áp, tốc độ cho CPU, RAM, cấp nguồn cho CMOS RAM và một số công dụng khác. Thường có hướng dẫn chi tiết đi kèm theo sách hướng dẫn Mainboard.

+ Tốc độ BUS (BUS Speed):

Tần số xung điện trên Main sẽ quyết định tính tương thích cho các loại thiết bị, tốc độ truyền dữ liệu. Tốc độ xung cơ bản của Mainboard thường là: 50MHz, 60MHz, 66 MHz, 75 MHz(Cyrix), 83 MHz, 100 MHz, 133 MHz, 150 MHz, 200 MHz (AMD K7), 400 MHz, 800 MHz.

**Một số loại Main thông dụng trên thị trường :**

INTEL: Sử dụng chipsets Intel.

GIGABYTE: Sử dụng chipset Intel và VIA.

MSI: Sử dụng chipset Intel và VIA.

TOMATO: Phần lớn sử dụng chipset VIA

**\*/ Bộ nhớ (Memory)**

Bộ nhớ là thiết bị lưu trữ thông tin trong quá trình máy tính xử lý. Bộ nhớ bao gồm bộ nhớ trong và bộ nhớ ngoài.

- **Bộ nhớ trong gồm ROM và RAM :**

+ ROM (Read Only Memory) là bộ nhớ chỉ đọc. Thực chất đây là một Chip IC có khả năng nhớ dữ liệu. Có hai loại ROM chính: ROM và EPROM. Đối với ROM, chương trình được các nhà sản xuất ghi vào Chip và không xoá được; còn đối với EPROM thì chương trình đã ghi vào rồi vẫn có thể dùng một thiết bị đặc biệt để ghi hoặc xoá được với các chế độ điện áp và truy nhập khác nhau. Về phía người dùng, chương trình đã ghi trong ROM thì chỉ có thể đọc ra để dùng cho nên nó thường được gọi là bộ nhớ chỉ đọc. Xét trên một góc độ khác, ROM là một linh kiện IC thuộc phần cứng nhưng nó lại được ghi chương trình điều khiển trong đó nên nó lại có thể được coi là phần mềm. Và trên quan điểm đó, các nhà tin học đã xếp ROM vào loại trung gian và gọi là phần sụn (hay phần nhào)!

Trong máy tính, ROM đóng vai trò gì? Khi truy nhập máy tính có khá nhiều thủ tục và chương trình cơ sở cần phải thi hành. Có thể ví CPU là ông chủ thì ROM như là một người thư ký. Mỗi khi máy tính bắt đầu hoạt động (khởi động hệ thống) sẽ có một số động tác cần phải thực thi đầu tiên gọi là POST (Power On Self Test) để kiểm tra các cấu hình chính của hệ thống như RAM, ổ đĩa, bàn phím có hoạt động bình thường không? Để làm những công việc thường xuyên đó, các nhà thiết kế máy tính đã viết những chương trình thực hiện các thủ tục kiểm tra này. Sau đó sẽ ghi vào ROM và mỗi khi khởi động máy, chương trình đó được thực hiện tức thì. Một số chương trình khác theo logic thiết kế cũng được ghi vào ROM.

+ RAM (Random Access Memory) được gọi là bộ nhớ truy cập ngẫu nhiên. Nó cũng bao gồm các linh kiện IC có khả năng nhớ tạm các chương trình được tải xuống trong quá trình hoạt động của máy tính. Điểm khác biệt về mặt nguyên tắc giữa RAM và ROM là: ROM ghi nhớ các

chương trình cố định, không thay đổi còn RAM thì nhớ tạm các chương trình, có thể đọc, ghi đè lên từng phần đã được tải xuống và khi mất điện hoặc tắt máy, các chương trình đó sẽ bị mất.

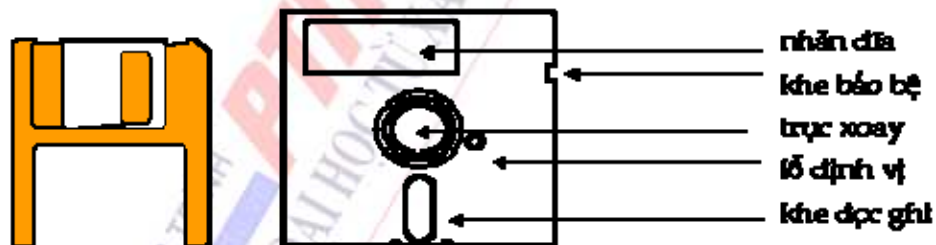
Nếu chúng ta ví ROM như người thư ký, thì có thể ví RAM như bàn làm việc của “ông trung tâm” CPU. Mỗi khi có một chương trình cần chạy, chương trình đó được nạp vào RAM và sẽ được CPU lấy ra để xử lý. Một chương trình nào đó có thể được nạp vào cư trú ở vùng khác hoặc cũng có thể bị ghi đè lên. Với vai trò như vậy, RAM cần được tăng dung lượng nhớ để có thể nạp nhiều chương trình. Trong các máy vi tính hiện nay khá phổ biến là 64/128/256 MB RAM tùy nhu cầu người sử dụng.

RAM được đóng thành mảng, mỗi mảng có thể 2, 4, 8, 16, 64, 128...MB nhớ. Trong bảng mạch chính có một số khe cắm (slot) để cắm thêm RAM. Cùng với tốc độ của CPU, dung lượng RAM càng lớn càng tăng tốc độ xử lý của hệ thống máy tính.

- *Bộ nhớ ngoài*: như đĩa từ, băng từ ... Để lưu trữ thông tin và có thể chuyển các tin này qua máy tính khác, người ta sử dụng các đĩa, băng từ như là các bộ nhớ ngoài. Các bộ nhớ này có dung lượng chứa lớn, không bị mất đi khi không có nguồn điện. Trên các máy vi tính phổ biến hiện nay có các loại đĩa từ sau:

+ Đĩa cứng (hard disk): có nhiều loại dung lượng từ vài trăm MB đến vài chục GB, hiện nay đã có đĩa cứng hơn 80 GB.

+ Đĩa mềm (floppy disk): phổ biến có 2 loại đĩa có đường kính 5.25 inches (dung lượng 360 KB hoặc 1.2 MB) và loại 3.5 inches (dung lượng 720 KB hoặc 1.44 MB)



Đĩa máy tính loại 3.5 in và 5.25 in.

+ Đĩa quang (Compact disk): loại 4.72 inches có dung lượng vào khoảng 600 MB. Đĩa quang thường chỉ được đọc và không ghi được (CD-ROM) là thiết bị phổ biến với các phần mềm phong phú mang nhiều thông tin, hình ảnh, âm thanh không thể thiếu được trong các phương tiện đa truyền thông (multimedia).

Tốc độ quay của đĩa mềm 5.25 in. khoảng 300 vòng/phút, đĩa 3.5 in. khoảng 600 vòng/phút. Tốc độ quay của đĩa cứng rất cao thường đạt trên 3600 vòng/phút. Vì vậy, thông tin chứa trên đĩa cứng sẽ được truy cập nhanh hơn trên đĩa mềm rất nhiều.

Hiện nay trên thị trường còn có loại đĩa nén, có kích thước như loại đĩa 1.44 MB, nhưng có dung lượng đến 100 MB và dễ dàng mang đi các nơi.

**c/ Các thiết bị vào/ra và lưu trữ dữ liệu**



- Màn hình (Monitor/Screen)
- > Kệ máy tính (Computer case)
- Ổ đĩa (Drive)
- Con chuột (Mouse)
- Bàn phím (Keyboard)

**Các thiết bị vào ra**

**Các thiết bị nhập thông tin đầu vào bao gồm :**

- *Bàn phím (Keyboard)*: là thiết bị nhập dữ liệu và câu lệnh, bàn phím máy vi tính phổ biến hiện nay là một bảng chứa 104 phím có các tác dụng khác nhau.

Có thể chia làm 3 nhóm phím chính:



**Bàn phím**

+ Nhóm phím đánh máy: gồm các phím chữ, phím số và phím các ký tự đặc biệt (~, !, @, #, \$, %, ^, &, ?, ...).

+ Nhóm phím chức năng (function key): gồm các phím từ F1 đến F12 và các phím khác như ←→ (phím di chuyển từng điểm), phím PgUp (lên trang màn hình), PgDn (xuống trang màn hình), Insert (chèn), Delete (xóa), Home (về đầu), End (về cuối), ...

+ Nhóm phím đệm số (numeric keypad) như NumLock (cho các ký tự số), CapsLock (tạo các chữ in), ScrollLock (chế độ cuộn màn hình) thể hiện ở các đèn chỉ thị.

Ngoài 3 phím có đèn chỉ thị trên ta còn các nút điều khiển sau:

+ Phím Shift: kèm với các phím chữ sẽ tạo ra chữ in hoa hoặc thường, đổi phím số thành các ký hiệu tương ứng trên nó.

+ Phím ←BackSpace: lùi điểm nháy đồng thời xóa ký tự đứng trước nó.

+ Phím Enter: nút thi hành lệnh hoặc xuống hàng.

+ Phím Space: thanh dài nhất, tạo ký tự rỗng.

+ Phím PrintScreen: nút in nội dung màn hình ra giấy.

+ Phím Pause: dừng thi hành chương trình.

+ Phím Ctrl (Control) và Alt (Alternate): là phím dùng để phối hợp các phím khác tùy chương trình sử dụng.

+ Phím Esc (Escape): phím thoát, được dùng khi có chỉ định rõ.

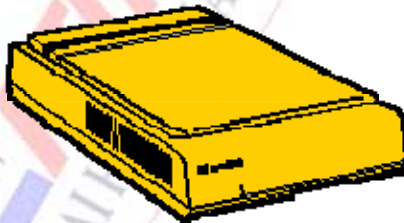
+ Phím Tab: phím nhảy cách, thường 8 khoảng (khoảng nhảy có thể khác đi tùy chương trình hay người sử dụng định).

- *Con chuột (Mouse)*: là thiết bị cần thiết phổ biến hiện nay, nhất là các máy tính chạy trong môi trường Windows. Con chuột có kích thước vừa nắm tay di chuyển trên một tấm phẳng (mouse pad) theo hướng nào thì đầu nhảy hoặc mũi tên trên màn hình sẽ di chuyển theo hướng đó tương ứng với vị trí của của viên bi hoặc tia sáng (optical mouse) nằm dưới bụng của nó. Một số máy tính có con chuột được gắn trên bàn phím.



**Con chuột**

- *Máy quét (scanner)*: là thiết bị dùng để nhập văn bản hay hình vẽ, hình chụp vào máy tính. Thông tin nguyên thủy trên giấy sẽ được quét thành các tín hiệu số tạo thành các tập tin ảnh (image file). Scanner đi kèm với phần mềm để nhận diện các tập tin ảnh hoặc văn bản.



**Máy quét Scanner**

- *Digitizer*: dùng để nhập dữ liệu đồ họa theo tọa độ X-Y vào máy tính, thường được dùng trong vẽ bản đồ.

- *Bút quang (Light pen)*: dùng nhập điểm bằng cách chấm lên màn hình.

- *Touch screen*: màn hình đặc biệt có thể dùng ngón tay để chạm lên các điểm.

**Các thiết bị xuất thông tin đầu ra bao gồm:**

- *Màn hình (Screen hay Monitor)*: là thiết bị xuất chuẩn, dùng để hiển thị thông tin cho người sử dụng xem. Thông tin được hiển thị ra màn hình bằng phương pháp ánh xạ bộ nhớ (memory mapping), với cách này màn hình chỉ việc đọc liên tục bộ nhớ và hiển thị (display) bất

kỳ thông tin nào hiện có trong vùng nhớ ra màn hình. Vì vậy để xuất thông tin ra màn hình ta chỉ cần xuất ra vùng nhớ tương ứng.



**Màn hình**

Có 2 chế độ hiển thị màn hình:

+ Trong chế độ văn bản, màn hình thể hiện 80 cột ký tự (đánh số từ 0 - 79) và 25 dòng (đánh số từ 0 - 24).

+ Trong chế độ đồ họa, màn hình được chia thành các phần tử ảnh (pixel: picture element). Độ phân giải màn hình được xác định bằng tích số kích thước chiều ngang và chiều cao tính theo phần tử ảnh. Tích số này càng lớn thì màn hình càng mịn, rõ nét.

Màn hình phổ biến hiện nay trên thị trường là màn hình màu SVGA và LCD.

Loại màn hình màu	Độ phân giải (pixel)
CCA : Color Graphics Adapter	320 x 200
EGA : Enhanced Graphics Adapter	640 x 350
VGA : Video Graphics Array	640 x 480
SVGA : Super VGA	1020 x 768

- *Máy in (printer)*: là thiết bị xuất để đưa thông tin ra giấy. Máy in phổ biến hiện nay là loại máy in ma trận điểm (dot matrix) loại 9 kim và 24 kim, máy in phun mực, máy in laser trắng đen hoặc màu. Giấy in thường dùng là loại giấy in 80 cột (in được 80 ký tự, in nén được 132 ký tự) và loại giấy in khổ rộng in được 132 cột (in 132 ký tự, in nén 256 ký tự). Cả 2 loại giấy in đều có khả năng in 66 dòng/trang.



**Máy in**

- *Máy vẽ (plotter)*: loại máy đặc biệt dùng các bút màu để vẽ đồ họa, chữ ...

- *Đĩa từ, băng từ (diskette, tape) ...*: dùng để chứa thông tin xuất



### 1.2.3.2. Phần mềm máy tính

Phần mềm là tên gọi khi nói tới các chương trình chỉ thị máy tính hoạt động xử lý dữ liệu thành những hình thái mà ta mong muốn. Chương trình là một chuỗi các chỉ thị lệnh có liên quan nhằm thực hiện từng bước tại mỗi thời điểm để hoàn thành một vài công việc nào đó dưới sự điều khiển của CPU. Các chương trình sẽ xác định việc các máy tính tiếp nhận đầu vào như thế nào và được hiển thị hoặc đưa tới đầu ra cái gì. Thông thường có 3 kiểu chương trình là : phần mềm hệ thống, trình dịch ngôn ngữ và các chương trình ứng dụng.

#### a/ Phần mềm hệ thống:

Phần mềm hệ thống là một thuật ngữ bao gồm tất cả các chương trình quản lý và điều khiển quá trình hoạt động của phần cứng máy tính. Nó có 2 loại chính - đó là:

- Phần mềm hệ điều hành.
- Các chương trình tiện ích.

#### \*/ Hệ điều hành:

Hệ điều hành là một phần mềm điều khiển quá trình hoạt động của máy tính từ khi khởi tạo hệ thống, nó được coi như là một yếu tố liên kết giữa phần cứng và phần mềm máy tính. Hệ điều hành bao gồm các chương trình quản lý điều khiển truyền thông giữa các bộ phận của phần cứng như card màn hình, card âm thanh, máy in, bảng mạch chính và các ứng dụng. Nó điều khiển tất cả đầu vào, đầu ra từ các thiết bị ngoại vi cũng như sự hoạt động của các chương trình khác. Nó cho phép người sử dụng làm việc và quản lý các tệp tin mà không cần biết cụ thể dữ liệu được lưu trữ và lấy ra trong các hệ thống đa người dùng; ngoài ra hệ điều hành còn quản lý người dùng truy nhập tới bộ xử lý, các thiết bị ngoại vi và lịch trình công tác.

Các chức năng của hệ điều hành:

- Định vị các tài nguyên hệ thống: điều khiển sự vận chuyển bên trong máy tính, có tác dụng quyết định đến tài nguyên nào sẽ được sử dụng và trong thời gian bao lâu.
- Thời gian : định thời trong CPU được phân chia thành các khoảng thời gian có đơn vị tính là ms. Đối với mỗi tác vụ được thực thi, CPU sẽ ấn định chính xác một khoảng thời gian nhất định cho nó. Khi thời gian kết thúc, các tác vụ khác sẽ được đưa vào để tiếp tục quay vòng.
- Bộ nhớ : bộ nhớ máy tính cũng được quản lý bởi hệ điều hành. Chúng được CPU sử dụng luân phiên để rời chuyển dữ liệu thông qua các bộ nhớ đệm. Chẳng hạn, các khoảng không gian của đĩa cứng cũng được coi như là một phần của bộ nhớ chính. Điều đó có nghĩa là, việc đưa dữ liệu thẳng từ đĩa cứng lên bộ nhớ chính mỗi khi CPU yêu cầu sẽ rất chậm trong khi dữ liệu đó phải thường xuyên cập nhật lên bộ nhớ. Vì vậy khi bộ nhớ chính bị đầy, một vài dữ liệu sẽ được phân trang đưa ra ngoài đĩa cứng và nó được gọi là vùng nhớ trao đổi (swapping). Trong hệ điều hành Windows, việc sử dụng tệp tin swap cũng chính là cho mục đích này.
- Quản lý vào/ra (input/output) : điều khiển luồng lưu lượng cũng là một phần khả năng đáp ứng của hệ điều hành. Hệ điều hành phải quản lý tất cả các yêu cầu như đọc dữ liệu từ ổ đĩa và băng từ hay ghi dữ liệu vào chúng hoặc đưa ra máy in.
- Giám sát hoạt động hệ thống : hệ điều hành thực hiện 2 công việc giám sát chủ yếu là thi hành hệ thống và bảo mật hệ thống. Thi hành hệ thống đưa ra các thông tin như tên các tác vụ

đang thi hành trên hệ thống, thời gian chiếm giữ CPU, bộ nhớ... hay người quản trị hệ thống cũng có thể theo dõi, kiểm tra xem các máy tính trên mạng có bị quá tải không để có những can thiệp kịp thời. Vấn đề bảo mật hệ thống khá quan trọng, đặc biệt khi sử dụng cho chế độ đa người dùng. Khi người sử dụng muốn truy nhập vào hệ thống cần phải thực hiện thủ tục đăng nhập user và mật khẩu (gọi là ID).

- Quản lý tệp tin và ổ đĩa: việc lưu trữ và bảo vệ các tệp tin trên thiết bị ổ đĩa, băng từ là một công việc chính, quan trọng trong mỗi hệ điều hành. Hiện nay, hầu hết các hệ điều hành đều sử dụng các hệ thống tệp tin như FAT, NFS. Ngoài việc quản lý các thiết bị nhớ vật lý như FDD, HDD, CDROM,... hệ điều hành còn quản lý các bộ nhớ ảo (virtual memory) được tạo ra trên cơ sở phần còn rỗi (free) của các thiết bị nhớ kể trên.

Các hệ điều hành cơ bản :

- Microsoft Windows: Windows 95, Windows 98, Windows Me (Millennium Edition), Windows NT (NT là viết tắt của New Technology), Windows CE và Windows 2000/NT,...

- Apple Macintosh.

- OS/2 của hãng IBM.

- UNIX : SCO Unix, SUN Solaris, AIX (IBM)

- LINUX được phát triển bởi Linus Torvalds tại University of Finland

#### **\*/ Các chương trình tiện ích:**

Các chương trình tiện ích thực hiện các công việc có liên quan đến bảo trì máy tính như phần cứng và dữ liệu. Các hệ điều hành hiện nay, hầu hết đều xây dựng phần mềm các chương trình tiện ích như :

- Chương trình quản lý tệp tin: Tạo ra cho người dùng dễ dàng quản lý các tệp tin của mình như : viết các chương trình trợ giúp tìm kiếm tệp tin, tạo ra và tổ chức các thư mục, sao chép, chuyển đổi tên tệp tin.

- Chương trình quản lý đĩa: Bao hàm cả định dạng và chống phân mảnh các đĩa. Chương trình chống phân mảnh thực hiện sắp xếp lại vị trí các tệp tin trên đĩa theo một dãy liên tục. Một số chương trình quản lý đĩa còn định rõ sắp đặt cho bạn khi có tệp tin thường xuyên được truy cập. Ngoài ra, có những hệ điều hành còn có thêm một số chức năng mới như cho phép chuyển đổi kiểu hệ thống tệp tin từ FAT32 sang NTFS và ngược lại...

- Phần mềm quản lý bộ nhớ: Thực hiện điều khiển bộ nhớ khi các dữ liệu hiện thời được đưa lên RAM. Chúng chuyển dịch các đối tượng nội trú bộ nhớ nào đó ra ngoài và điều này có thể làm tăng hiệu quả sử dụng bộ nhớ.

- Chương trình sao lưu (Backup) dữ liệu: Cho phép người sử dụng có thể phục hồi lại dữ liệu khi cần thiết.

- Chương trình nén dữ liệu: Cho phép người sử dụng thông qua các phần mềm nén dữ liệu trước khi lưu trữ nhằm tiết kiệm không gian nhớ của đĩa.

- Chương trình phòng chống Virus: Là các phần mềm khác nhau được cài đặt vào bộ nhớ máy tính nhằm giám sát sự hoạt động của virus trong máy tính và mạng. Nó có nhiệm vụ phát

hiện sự hoạt động của virus trong bộ nhớ máy tính và thực hiện “làm sạch” chúng. Hiện nay các phiên bản phần mềm phòng chống virus hữu hiệu được sử dụng rộng rãi như Norton AntiVirus, D2, Bkav2000,...

**b/ Trình dịch và ngôn ngữ:**

Trong quá trình hoạt động của các thế hệ máy tính điện tử, tất cả các chỉ thị lệnh được hoạt hóa bởi máy tính đều phải cung cấp tới CPU dưới dạng mã máy (machine code). Tuy nhiên, con người không phải tương tác với máy tính tại mức này mà các nhà lập trình có thể viết phần mềm bằng ngôn ngữ Assembler cho phép can thiệp trực tiếp tới các thanh ghi của CPU. Hiện nay, hầu hết các chương trình được viết trên cơ sở các tiêu chuẩn ngôn ngữ bậc cao như Pascal, Basic, C,....

Trình dịch là một chương trình được thiết kế nhằm chuyển hóa một chương trình được viết bởi ngôn ngữ bậc cao thành các chỉ thị lệnh máy của hệ thống tính toán riêng biệt. Đầu ra của một trình dịch C cho máy tính IBM hầu như không có gì chung với đầu ra của một trình dịch C cho máy chủ VAX. Mặc dù các ngôn ngữ bậc cao được dùng chung để phát triển các gói ứng dụng như GIS, nó thường được dịch cho các nền (platform) riêng biệt trước khi phân phối tới người dùng.

**c/ Phần mềm ứng dụng:**

Các phần mềm ứng dụng thông dụng hiện nay ngày càng phong phú và đa dạng. Nó bao gồm những chương trình được viết ra phục vụ cho một hay nhiều mục đích cụ thể như ứng dụng văn phòng, tính toán, phân tích dữ liệu, tổ chức hệ thống, bảo mật thông tin, xử lý đồ họa, trò chơi điện tử, dịch vụ thông tin mạng,...

**1.2.4. Lịch sử phát triển của hệ thống máy tính**

Do nhu cầu cần tăng độ chính xác và giảm thời gian tính toán, con người đã quan tâm chế tạo các công cụ tính toán từ xưa: bàn tính tay của người Trung quốc, máy cộng cơ học của nhà toán học Pháp Blaise Pascal (1623 - 1662), máy tính cơ học có thể cộng trừ nhân chia của nhà toán học Đức Gottfried Wilhelm von Leibniz (1646 - 1716), máy sai phân để tính các đa thức toán học, máy phân giải điều khiển bằng phiếu đục lỗ của Charles Babbage (1792 - 1871) ...

Tuy nhiên, máy tính điện tử thực sự hình thành bắt đầu vào thập niên 1950 và đến nay đã trải qua 5 thế hệ được phân loại theo sự tiến bộ về công nghệ điện tử và vi điện tử cũng như các cải tiến về nguyên lý, tính năng và loại hình của nó.

\* Thế hệ 1 (1950 - 1958): Máy tính sử dụng các bóng đèn điện tử chân không, mạch riêng rẽ, vào số liệu bằng phiếu đục lỗ, điều khiển bằng tay. Máy có kích thước rất lớn, tiêu thụ năng lượng nhiều, tốc độ tính chậm khoảng 300 - 3.000 phép tính mỗi giây. Loại máy tính điển hình thế hệ 1 như EDVAC (Mỹ) hay BESM (Liên xô cũ), ...

\* Thế hệ 2 (1958 - 1964): Máy tính dùng bộ xử lý bằng đèn bán dẫn, mạch in. Máy đã có chương trình dịch như Cobol, Fortran và hệ điều hành đơn giản. Kích thước máy còn lớn, tốc độ tính khoảng 10.000 đến 100.000 phép/s. Điển hình như loại IBM-1070 (Mỹ) hay MINSK (Liên Xô cũ), ...

\* Thế hệ 3 (1965 - 1974): Máy tính được gắn các bộ xử lý bằng vi mạch điện tử cỡ nhỏ có thể có được tốc độ tính khoảng 100.000 đến 1 triệu phép/s. Máy đã có các hệ điều hành đa chương trình, nhiều người dùng đồng thời hoặc theo kiểu chia thời gian. Kết quả từ máy tính có thể in ra trực tiếp ở máy in. Điển hình như loại IBM 360 (Mỹ) hay EC (Liên Xô cũ), ...

\* Thế hệ 4 (1974 đến nay): Máy tính bắt đầu có các vi mạch đa xử lý có tốc độ tính hàng chục triệu đến hàng tỷ phép/giây. Giai đoạn này hình thành 2 loại máy tính chính : máy tính cá nhân để bàn (Personal Computer - PC) hoặc xách tay (Laptop hoặc Notebook computer) và các loại máy tính chuyên nghiệp thực hiện đa chương trình, đa vi xử lý ... hình thành các hệ thống mạng máy tính (Computer Networks), và các ứng dụng phong phú đa phương tiện.

\* Thế hệ 5 (1990 - nay): Bắt đầu có các nghiên cứu tạo ra các máy tính mô phỏng các hoạt động của não bộ và hành vi con người, có trí khôn nhân tạo với khả năng tự suy diễn phát triển các tình huống nhận được và những hệ quản lý kiến thức cơ sở để giải quyết các bài toán đa dạng.

### 1.3. THUẬT TOÁN VÀ SƠ ĐỒ KHỐI

Muốn giải một bài toán trên máy tính điện tử người ta cần một chương trình hay phần mềm hướng dẫn máy tính thực hiện các thao tác cần thiết. Trước khi giới thiệu phần mềm ta cần hiểu các bước để giải một bài toán trên máy tính.

#### ***a/ Các giai đoạn giải một bài toán trên máy tính điện tử***

Để giải quyết một bài toán trên máy tính điện tử, cần qua các giai đoạn:

- Tìm hiểu mục tiêu chính của bài toán: số liệu nhập và kết quả xuất.
- Xây dựng một chuỗi thao tác tính toán theo tuần tự, gọi là thuật giải.
- Lập chương trình diễn tả chi tiết các bước tính theo thuật giải
- Nhập chương trình vào máy tính, thông dịch và chạy thử để sửa chữa lỗi
- Thực hiện giải bài toán với số liệu thu thập được và ghi nhận kết quả
- Thử nghiệm với nhiều trường hợp khác nhau của bài toán
- Phân tích kết quả và hoàn chỉnh chương trình

Trong các bước trên, việc thiết kế thuật toán là giai đoạn quan trọng nhất.

#### ***b/ Thuật toán (algorithm)***

**Định nghĩa:** Thuật toán là một phương pháp trình bày các bước giải quyết một hay nhiều bài toán theo một tiến trình xác định.

Thuật toán có các đặc tính sau:

- *Tính xác định:* Các thao tác của thuật toán là rõ ràng và chắc chắn thực hiện được để dẫn đến kết quả nào đó.

- *Tính hữu hạn và dừng:* thuật toán phải có một số bước giải nhất định và cuối cùng phải có kết thúc ở điểm dừng.

- *Tính kết quả:* Với dữ liệu hợp lý, thuật toán phải cho kết quả thỏa yêu cầu.

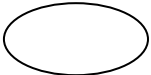
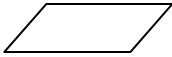
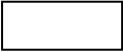
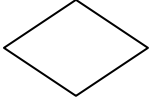

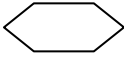



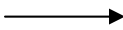
- *Tính phổ dụng:* Thuật toán phải giải được nhiều bài toán có cùng cấu trúc với các dữ liệu khác nhau và đều dẫn đến một kết quả mong muốn.

- *Tính hiệu quả*: Thuật giải phải đơn giản, dễ hiểu trong các bước giải, tối thiểu hoá bộ nhớ và thời gian thực hiện.

- *Tính hình thức*: Các bước trong thuật toán là máy móc, nghĩa là nó phải thực hiện đúng như quy định mà không cần biết đến mục tiêu cuối cùng.

Thuật toán có thể diễn giải một cách trực quan bằng lưu đồ (flowchart). Lưu đồ được sử dụng thông dụng trong việc trình bày các bước cần thiết để giải quyết vấn đề qua các hình khối khác nhau và dòng dữ liệu giữa các bước được chỉ định đi theo các đường mũi tên.

Một số qui ước ký hiệu lưu đồ:

Ký hiệu	Mô tả
	Điểm bắt đầu và kết thúc một thuật toán
	Thao tác nhập hay xuất dữ liệu
	Khối xử lý công việc
	Khối quyết định chọn lựa
	Điểm nối
	Chuẩn bị
	Tập hợp các tập tin dữ liệu
	Khối chương trình con
	Các ghi chú, giải thích
	Dòng tính toán, thao tác của chương trình

Ví dụ: Thuật toán giải phương trình bậc nhất:  $ax + b = 0$ , ta đi qua các bước:

- Bước 1: Nhập vào 2 hệ số a và b.

- Bước 2: Xét điều kiện  $a = 0$  ?

Nếu đúng là  $a = 0$ , thì đi đến bước 3. Nếu không, nghĩa là  $a \neq 0$ , thì đi đến bước 4.

- Bước 3: Xét điều kiện  $b = 0$  ?

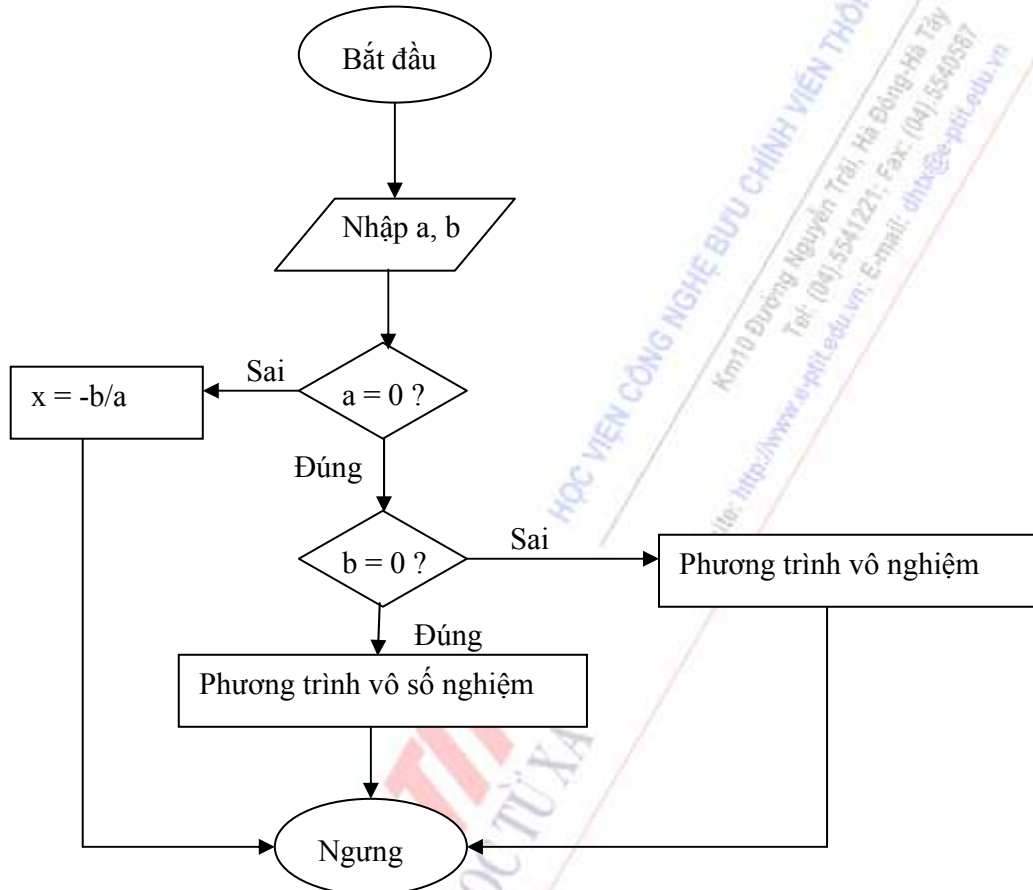
Nếu  $b = 0$ , thì báo phương trình có vô số nghiệm. Đi đến bước 5.

Nếu  $b \neq 0$ , thông báo phương trình vô nghiệm. Đi đến bước 5.

- Bước 4: Thông báo phương trình có một nghiệm duy nhất là  $x = -b/a$ .

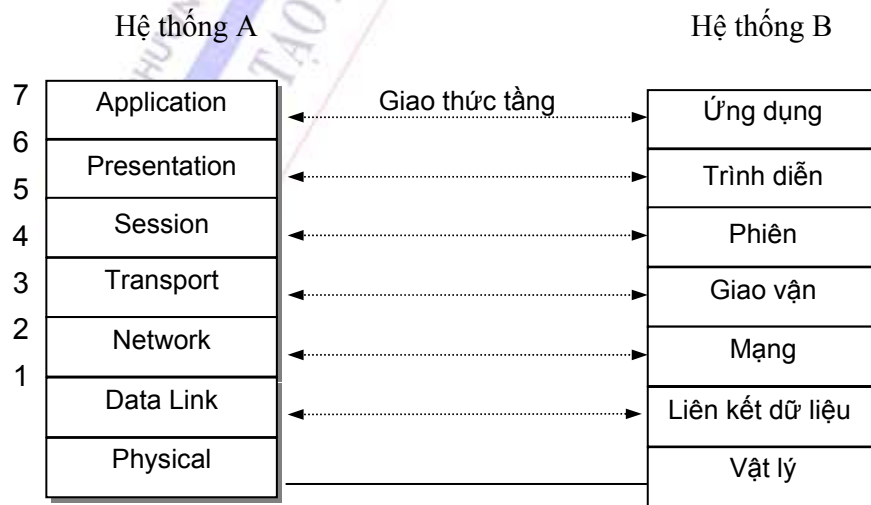
- Bước 5: Ngưng dứt thuật toán

Ví dụ: Với bài toán như trong ví dụ 2.1, ta có thể trình bày với lưu đồ sau:



## 1.4. MẠNG MÁY TÍNH (COMPUTER NETWORK)

### 1.4.1. Giới thiệu mô hình tham chiếu OSI



Mô hình OSI (Open Systems Interconnection) là một cơ sở cho việc chuẩn hóa các hệ thống truyền thông, nó được nghiên cứu và xây dựng vào năm 1971 bởi Tổ chức tiêu chuẩn Quốc tế ISO (International Standards Organization). Mục tiêu chính của mô hình OSI là nhằm tới việc kết nối giữa các sản phẩm của các hãng sản xuất khác nhau và phối hợp các hoạt động chuẩn hóa trong các lĩnh vực viễn thông và hệ thống thông tin. Theo mô hình OSI, chương trình truyền thông được chia thành 7 tầng chức năng khác nhau. Hai tầng đồng mức khi liên kết với nhau phải sử dụng một giao thức chung. Trong mô hình OSI có hai loại giao thức chính được áp dụng: giao thức có liên kết (connection-oriented) và giao thức không liên kết (connectionless)

- Giao thức có liên kết: trước khi truyền dữ liệu hai tầng đồng mức cần thiết lập một liên kết logic và các gói tin được trao đổi thông qua liên kết này; việc có liên kết logic sẽ nâng cao độ an toàn trong truyền dữ liệu.

- Giao thức không liên kết: Trước khi truyền dữ liệu không thiết lập liên kết logic và mỗi gói tin được truyền độc lập với các gói tin trước hoặc sau nó.

Chức năng của các tầng trong mô hình OSI :

• **Tầng ứng dụng (Application layer):** Tầng ứng dụng qui định giao diện giữa người sử dụng và môi trường OSI, nó cung cấp các phương tiện cho người sử dụng truy cập và sử dụng các dịch vụ của mô hình OSI.

• **Tầng trình bày (Presentation layer):** Tầng trình bày chuyển đổi các thông tin từ cú pháp người sử dụng sang cú pháp để truyền dữ liệu. Ngoài ra nó còn có thể nén dữ liệu truyền và mã hóa chúng trước khi truyền để đảm bảo tính bảo mật cho dữ liệu.

• **Tầng phiên (Session layer):** Tầng phiên qui định một giao diện ứng dụng cho tầng vận chuyển sử dụng. Nó xác lập ánh xạ giữa các tên đặt địa chỉ, tạo ra các tiếp xúc ban đầu giữa các máy tính khác nhau trên cơ sở các phiên truyền thông. Nó đặt tên nhất quán cho mọi thành phần muốn đối thoại riêng với nhau.

• **Tầng vận chuyển (Transport layer):** Tầng vận chuyển xác định địa chỉ trên mạng, cách thức chuyển giao gói tin trên cơ sở trực tiếp giữa hai điểm (end-to-end). Để đảm bảo được việc truyền ổn định trên mạng tầng vận chuyển thường đánh số các gói tin và đảm bảo chúng chuyển theo thứ tự.

• **Tầng mạng (Network layer):** Tầng mạng có nhiệm vụ xác định việc chuyển hướng, vạch đường các gói tin trong mạng, các gói tin này có thể phải đi qua nhiều chặng trước khi đến được đích cuối cùng.

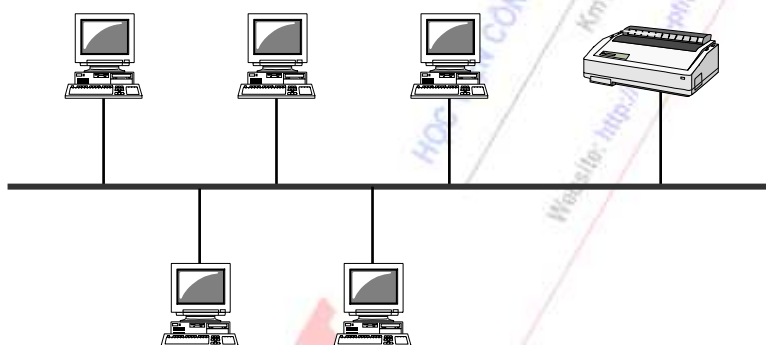
• **Tầng liên kết dữ liệu (Data link layer):** Tầng liên kết dữ liệu có nhiệm vụ xác định cơ chế truy nhập thông tin trên mạng, các dạng thức chung trong các gói tin, đóng các gói tin,...

• **Tầng vật lý (Physical layer):** Tầng vật lý cung cấp phương thức truy cập vào đường truyền vật lý để truyền các dòng Bit không cấu trúc. Ngoài ra, nó cung cấp các chuẩn về điện, dây cáp, đầu nối, kỹ thuật nối mạch điện, điện áp, tốc độ cáp truyền dẫn, giao diện kết nối và các mức kết nối,...

### 1.4.2. Khái niệm về mạng và kết nối mạng

Xét ở mức độ đơn giản nhất, mạng (network) bao gồm hai máy tính nối với nhau bằng cáp (cable) sao cho chúng có thể dùng chung dữ liệu. Mọi mạng máy tính dù có phức tạp đến mấy cũng đều bắt nguồn từ hệ thống đơn giản đó. Ý tưởng nối hai máy tính bằng cáp thoạt nghe có vẻ không có gì là phi thường, nhưng nếu nhìn lại, đó chính là một thành tựu lớn lao trong công nghệ truyền thông.

Mạng máy tính được phát sinh từ nhu cầu muốn chia sẻ và dùng chung tài nguyên dữ liệu. Máy tính cá nhân là công cụ tuyệt vời giúp ta tạo dữ liệu, bảng tính, hình ảnh và nhiều dạng thông tin khác, nhưng nó lại không cho phép ta thuận tiện chia sẻ dữ liệu ta đã tạo. Khi không có hệ thống mạng, dữ liệu phải được in ra giấy để người khác xem và hiệu chỉnh. Hay một cách tốt hơn là dùng đĩa mềm để sao chép rồi chuyển đến các máy tính khác. Tuy nhiên nếu người khác thay đổi tài liệu thì chúng ta không thể hợp nhất các thay đổi này. Phương thức làm việc kiểu này gọi là làm việc trong môi trường độc lập.



#### Một mạng đơn giản

Nếu người dùng làm việc trong môi trường mạng và nối máy tính của mình với các máy tính khác, người này có thể sử dụng dữ liệu trên các máy khác và thậm chí cả máy in. Một nhóm máy tính và những thiết bị ngoại vi kết nối với nhau được gọi là mạng, còn việc đấu nối các máy tính với nhau để sử dụng chung tài nguyên gọi là nối mạng (Networking).

Trên thực tế, hiện nay có rất nhiều mạng khác nhau với qui mô phát triển và ứng dụng khác nhau. Để phân biệt được các loại mạng này, ta có nhiều cách phân loại mạng khác nhau tùy thuộc vào yếu tố chính được chọn để làm chỉ tiêu phân loại, chẳng hạn đó là "khoảng cách địa lý", "kỹ thuật chuyển mạch" hay "kiến trúc mạng".

#### **\*/ Phân loại theo khoảng cách địa lý, mạng máy tính gồm có:**

- Mạng cục bộ (Local Area Networks viết tắt là LAN) là mạng máy tính được tổ chức trong phạm vi nhỏ khoảng vài chục ki lô mét trở lại, ví dụ mạng nội bộ cơ quan, trường học, xí nghiệp, văn phòng ...

- Mạng đô thị (Metropolitan Area Networks viết tắt là MAN) là mạng máy tính được tổ chức trong phạm vi 100 ki lô mét trở lại, ví dụ mạng thành phố, trung tâm kinh tế, khu công nghệ cao...



- Mạng diện rộng (Wide Area Network viết tắt là WAN) là mạng máy tính được tổ chức trong phạm vi rộng, như mạng quốc gia, liên bang, châu lục .

- Mạng toàn cầu (Global Area Network viết tắt là GAN) là mạng máy tính được tổ chức rộng khắp toàn cầu.

**\*/ Phân loại theo kỹ thuật chuyển mạch, mạng máy tính gồm có:**

- Mạng chuyển mạch kênh (Circuit-switched Networks) là mạng thực hiện việc kết nối hai thực thể ở hai đầu theo một kênh cố định trong thời gian truyền tin.

- Mạng chuyển mạch thông báo (Message-Switched Networks) thông tin truyền đi theo một khuôn dạng quy định, trong đó được chỉ định đích đến. Căn cứ vào thông tin đích đến các thông báo có thể được truyền qua nhiều con đường khác nhau để đến đích.

- Mạng chuyển mạch gói (Packet-Switched Networks) là mạng trong đó thông báo cần gửi đi được chia nhỏ thành các gói (packet) có số lượng bytes cố định. Mỗi gói tin có địa chỉ đích và đánh dấu thứ tự và có thể đi theo nhiều đường khác nhau để tới đích. Khi tới đích, chúng được kết nối lại với nhau theo thứ tự đã được đánh số.

**\*/ Phân loại theo kiến trúc mạng:**

Chúng ta thường nghe nói đến mạng SNA của IBM, ISO của ISO hay mạng TCP/IP... Trong mạng LAN và thậm chí cả mạng WAN người ta còn phân chia mạng theo TOPO mạng: Mạng hình sao (STAR), mạng hình BUS, mạng hình vòng. Ngoài ra còn một số dạng khác như mạng hình cây (TREE), mạng hình sao vòng, mạng hỗn hợp...

**1.4.3. Mạng cục bộ (Local Area Network - LAN)**

Mạng khởi đầu với quy mô rất nhỏ, với khoảng 10 máy tính được nối với nhau và máy in. Công nghệ tin học đã làm hạn chế qui mô phát triển mạng như số lượng máy tính kết nối với nhau, khoảng cách vật lý mà mạng có thể bao phủ. Chẳng hạn, ở những năm đầu thập kỷ 80, phương pháp lắp đặt cáp phổ biến nhất cũng chỉ cho phép chừng 30 người với chiều dài cáp tối đa khoảng 600 fit (xấp xỉ 183m).

Mạng cục bộ (LAN - Local Area Network) là một loại mạng máy tính được cài đặt trong phạm vi địa lý tương đối nhỏ, chẳng hạn một toà nhà, trường học, công sở .v.v. Khoảng cách lớn nhất giữa các máy tính trong mạng chỉ khoảng dưới vài chục ki-lô-mét.

**\*/ Lợi ích của mạng cục bộ**

Cũng giống như mạng nói chung, mạng cục bộ có một số lợi ích chính sau:

- Chia sẻ thông tin, tài nguyên:

Mạng cục bộ cho phép nhiều người dùng truy cập vào CSDL chung, được quản lý tập trung trong các thiết bị lưu trữ. Ví như trong một công ty, các bộ phận khác nhau cùng sở hữu một bộ tài liệu nào đó, và tại bất kì thời điểm nào mạng cũng cho phép chia sẻ bộ tài liệu này .

- Truyền dữ liệu với tốc độ cao:

Khi không sử dụng mạng cục bộ, người ta sử dụng con người và các dịch vụ thư tín để gửi tài liệu đến các nơi khác nhau. Nhờ có mạng cục bộ mà thư điện tử được truyền qua mạng từ một trạm ở nơi này đến một trạm ở nơi khác. Điều này làm tăng tốc độ truyền thông tin và giảm bớt chi phí công việc.

- Giảm chi phí nhờ việc chia sẻ tài nguyên, thiết bị:

Chức năng truyền thông tốc độ cao của mạng cục bộ cho phép tất cả các trạm chia sẻ các ổ đĩa (đĩa từ và đĩa quang), máy in, và các thiết bị khác, bởi vậy có thể giảm bớt chi phí.

### **\* / Phân loại mạng cục bộ**

Mạng cục bộ được chia làm 2 loại:

- Mạng ngang hàng (Peer-to-Peer).
- Mạng dựa trên máy chủ (Server-based).

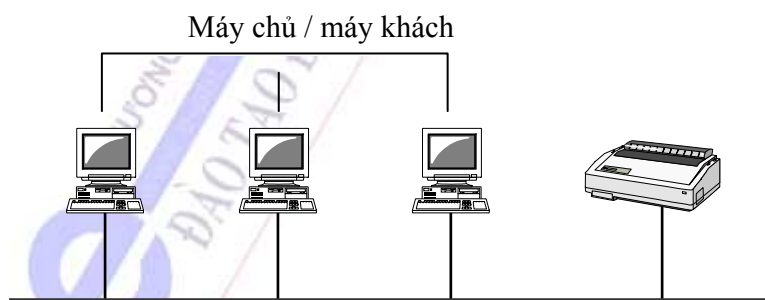
Sự phân biệt giữa 2 loại mạng nói trên là rất quan trọng do mỗi loại có những khả năng khác nhau. Loại mạng bạn sử dụng sẽ phụ thuộc vào một số yếu tố, như là:

- Quy mô tổ chức (công ty, văn phòng).
- Mức độ bảo mật cần có.
- Loại hình công việc.
- Mức độ hỗ trợ sẵn có trong công tác quản trị.
- Nhu cầu của người dùng mạng.
- Ngân sách mạng.

#### **(-) Mạng ngang hàng**

Mạng cục bộ kiểu ngang hàng không có sự phân biệt giữa máy khách và máy chủ. Các máy có mối quan hệ bình đẳng với nhau. Nói cách khác, đó là một mạng cục bộ mà mỗi máy đều có thể là máy khách hoặc máy chủ. Lưu lượng thường không bị tập trung trong hệ thống này.

Mạng ngang hàng tương đối đơn giản. Vì mỗi máy tính vừa kiêm các chức năng máy chủ và máy khách, nên không cần có một máy phục vụ trung tâm thật mạnh và cũng không bắt buộc phải có những bộ phận cần thiết cho mạng máy tính công suất cao. Việc xây dựng mạng ngang hàng có thể kinh phí sẽ thấp hơn so với mạng dựa trên máy chủ.

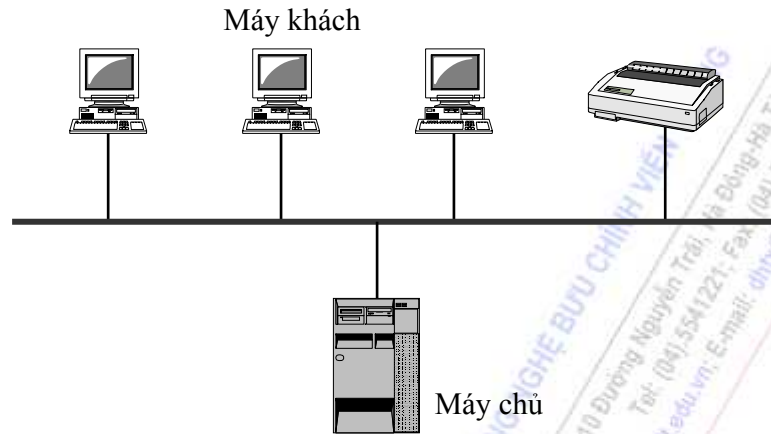


#### **Các máy tính ở mạng ngang hàng kiêm cả vai trò máy khách và máy chủ**

#### **(-) Mạng dựa trên máy chủ**

Mạng cục bộ dựa trên máy chủ bao gồm các máy khách và các máy chủ chuyên dụng. Máy chủ chuyên dụng là máy chỉ hoạt động như một máy chủ chứ không kiêm luôn vai trò máy khách hay trạm làm việc. Các máy khách nhận được các dịch vụ khác nhau do máy chủ cung cấp.

Việc truyền thông trong hệ thống này chủ yếu được tiến hành giữa nhiều máy khách và một vài máy chủ, do vậy lưu lượng thông tin sẽ tập trung vào ra các máy chủ rất lớn.



**Mạng dựa trên máy chủ**

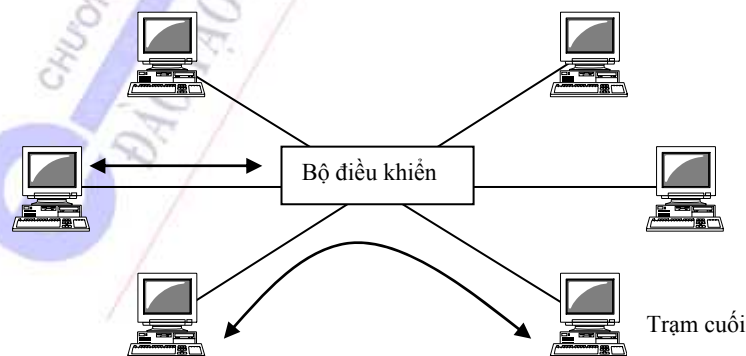
#### 1.4.4. Cấu trúc liên kết mạng (Topology)

Thuật ngữ cấu trúc liên kết (hay còn gọi là Topology) mạng máy tính chỉ sự sắp xếp các trạm cuối được gắn vào mạng. Các cấu trúc liên kết thường dùng là hình sao (star), bus, và vòng (ring).

##### a/ Cấu trúc liên kết hình sao (Star)

Mạng hình sao bao gồm một bộ điều khiển trung tâm, mỗi trạm cuối được kết nối vào bộ điều khiển trung tâm này bằng các đường truyền theo dạng hình sao.

Trong hình 4.4 ở dưới, mỗi trạm cuối trong mạng hình sao được kết nối tới bộ điều khiển trung tâm (Hub/Switch) bằng một đường truyền riêng biệt (UTP), do đó nó tạo ra dạng hình sao. Bộ điều khiển trung tâm này điều khiển việc truyền thông cho mỗi trạm cuối. Việc truyền thông ở đây bao gồm truyền thông giữa trạm cuối với bộ điều khiển trung tâm hoặc giữa trạm cuối này với trạm cuối khác thông qua bộ điều khiển trung tâm.

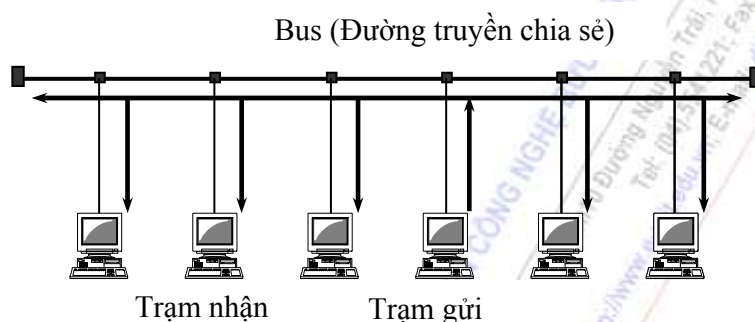


**Cấu trúc liên kết hình sao**

##### b/ Cấu trúc liên kết dạng BUS

Mạng dạng BUS bao gồm một đường truyền dữ liệu tốc độ cao duy nhất. Đường truyền này được gọi là bus và được chia sẻ bởi nhiều nút. Bất cứ khi nào muốn truyền dữ liệu, trạm truyền ấn định địa chỉ trạm đích và truyền dữ liệu lên bus.

Thông tin được truyền từ bất kì trạm cuối nào đều được gửi tới tất cả các nút. Một nút chỉ nhận dữ liệu khi nó nhận đúng địa chỉ dữ liệu gửi cho nó. Mỗi đầu của BUS được gắn 1 bộ kết cuối (terminator). Bộ kết cuối có tác dụng chặn tín hiệu để tránh tình trạng phản hồi tín hiệu. Vì trong trường hợp có tín hiệu phản hồi, đường truyền sẽ bị nhiễu và sẽ xuất hiện lỗi trong quá trình truyền.

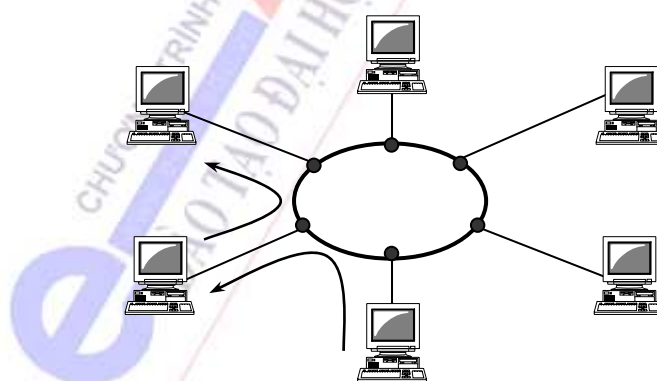


**Cấu trúc liên kết hình BUS**

**Chú ý:** Đối với Bus một chiều (có nghĩa là tín hiệu chỉ được truyền theo một hướng) thì việc phản hồi tín hiệu là cần thiết.

**c/ Cấu trúc liên kết dạng vòng (Ring)**

Mạng có cấu trúc liên kết dạng vòng có hình dạng một vòng tròn khép kín, các nút được nối với vòng tại các điểm cách nhau một khoảng nào đó. Thông tin được truyền trên vòng theo một hướng nhằm tránh xung đột. Do mỗi nút có thể tái tạo và lặp lại tín hiệu nên cấu trúc liên kết kiểu này phù hợp với các mạng có phạm vi rộng hơn so với kiến trúc kiểu bus.



**Cấu trúc liên kết dạng vòng**

### 1.4.5. Phương tiện truyền dẫn

Ngày nay, phần lớn mạng được nối bằng dây dẫn hoặc cáp thuộc loại nào đó, đóng vai trò như phương tiện truyền dẫn giữa các máy tính trong mạng. Rất nhiều loại cáp có thể đáp ứng các yêu cầu và qui mô mạng khác nhau, từ nhỏ đến lớn.

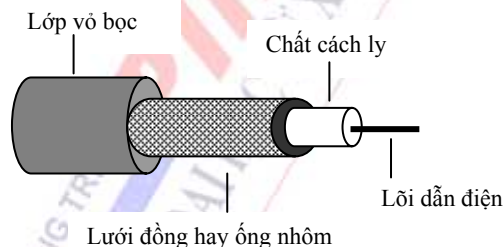
Để tránh nhầm lẫn, Belden, một hãng sản xuất cáp hàng đầu, xuất bản một catalog liệt kê hơn 2200 loại cáp. Tuy nhiên chỉ có 3 nhóm cáp chính được dùng để nối hầu hết các mạng.

- Cáp đồng trục (coaxial)
- Cáp xoắn đôi (twisted-pair)
- + Cáp xoắn đôi trần (unshielded twisted-pair)
- + Cáp xoắn đôi có bọc (shielded twisted-pair)
- Cáp sợi quang (fiber-optic)

#### a/ Cáp đồng trục

Có một thời gian, cáp đồng trục là cáp mạng thông dụng nhất. Sở dĩ cáp đồng trục được sử dụng rộng rãi là do: cáp đồng trục tương đối rẻ tiền, nhẹ, mềm và dễ kéo dây. Cáp đồng trục phổ biến đến mức nó trở thành phương tiện lắp đặt an toàn và dễ chấp nhận.

Ở dạng đơn giản nhất, cáp đồng trục gồm một lõi đồng nguyên chất được bọc chất cách ly, một lớp bảo vệ bằng lưới kim loại và một lớp vỏ bọc ngoài (hình 4.7). Lớp chất cách ly và lớp lưới kim loại được xem là lớp bọc đôi. Tuy nhiên còn có loại cáp bọc 4 lớp dành cho môi trường hay bị nhiễu.



#### b/ Cáp xoắn đôi

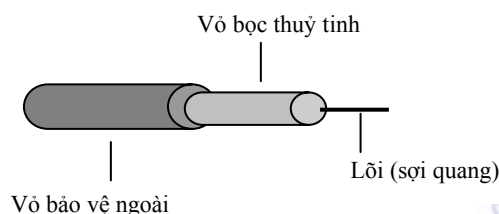
Ở dạng đơn giản, cáp xoắn đôi gồm hai sợi dây đồng cách ly cuốn vào nhau. Cáp xoắn đôi có hai loại: cáp xoắn đôi không bọc (UTP) và cáp xoắn đôi có bọc (STP).



Một số dây xoắn đôi thường được nhóm chung với nhau và được quấn kín trong vỏ bọc bảo vệ để tạo thành sợi cáp. Số lượng dây xoắn đôi trong các loại cáp là khác nhau. Sự xoắn này làm vô hiệu nhiễu điện từ dây xoắn đôi kế cận và từ những nguồn khác như mô tơ, máy biến thế.

### c/ Cáp sợi quang

Cáp sợi quang gồm một sợi thủy tinh cực mảnh, gọi là lõi, được bao bọc bởi một lớp thủy tinh đồng tâm gọi là lớp vỏ bọc. Đôi khi các sợi được làm bằng chất dẻo. Chất dẻo dễ lắp đặt hơn nhưng không thể mang xung ánh sáng đi xa như thủy tinh.



Lõi sợi thủy tinh chỉ truyền tín hiệu theo một hướng nhất định, do đó cáp có hai sợi nằm trong vỏ bọc riêng biệt. Một sợi truyền và một sợi nhận.

Cáp sợi quang có thể truyền tín hiệu đi xa hơn và với tốc độ cực nhanh (về lý thuyết cáp quang có thể truyền tín hiệu với tốc độ tối đa 200.000Mb/s).

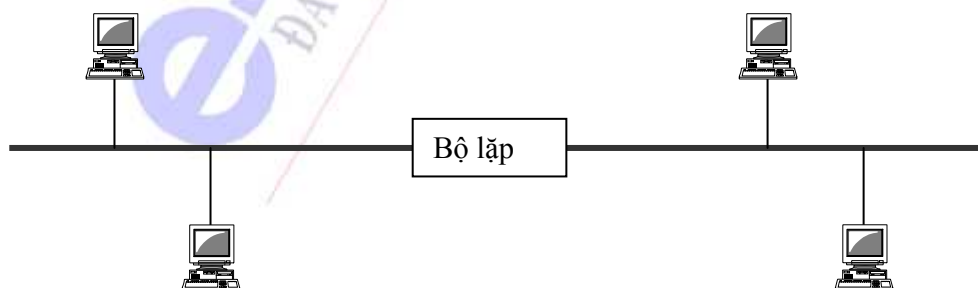
### 1.4.6. Các thiết bị liên kết mạng

Trên thực tế, mỗi chuẩn mạng cục bộ đều có những giới hạn về khoảng cách mạng, về số lượng trạm tối đa trong mạng. Để có thể mở rộng được mạng, cần có các thiết bị để kết nối các đoạn mạng lại với nhau. Các thiết bị liên kết mạng tiêu biểu - đó là: bộ lặp (repeater), HUB, cầu nối (bridge), bộ định tuyến (router) và bộ chuyển mạch LAN (LAN switch).

#### a/ Bộ lặp (repeater)

Bộ lặp là thiết bị tầng vật lý được sử dụng để kết nối các đoạn mạng lại với nhau. Bộ lặp cho phép nhiều đoạn mạng có thể hoạt động như một đoạn mạng đơn lẻ. Bộ lặp nhận tín hiệu từ một đoạn mạng, tái tạo và truyền tín hiệu này đến đoạn mạng khác. Nhờ có bộ lặp mà tín hiệu bị suy yếu do phải truyền qua một đoạn cáp dài có thể trở lại dạng ban đầu và truyền đi được xa hơn.

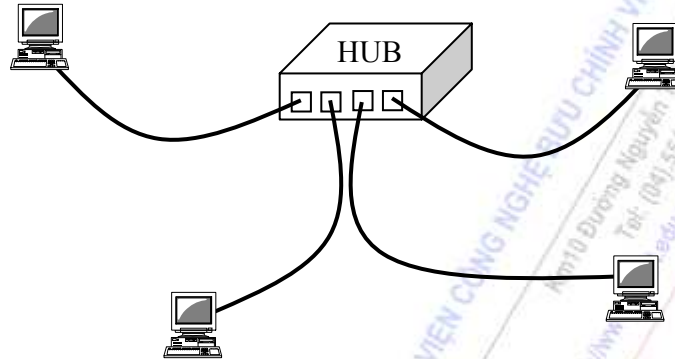
Bộ lặp không dịch hoặc lọc bất kỳ tín hiệu nào, nó không có khả năng xử lý lưu lượng. Tất cả các tín hiệu điện, bao gồm cả nhiễu điện từ và các lỗi khác cũng được lặp và khuếch đại. Để bộ lặp hoạt động, cả hai đoạn mạng nối tới bộ lặp phải sử dụng cùng một phương thức truy nhập đường truyền. Ví dụ, bộ lặp không thể nối một đoạn mạng sử dụng phương thức CSMA/CD (*Carrier Sense Multiple Access/Collision Detection: Đa truy nhập cảm nhận sóng mang có phát hiện xung đột*) và một đoạn mạng sử dụng phương thức chuyển thẻ bài (*Token Ring*).



Bộ lặp có thể di chuyển gói dữ liệu từ phương tiện truyền dẫn này sang phương tiện truyền dẫn khác. Chẳng hạn, chúng có thể nhận một gói dữ liệu từ một đoạn mạng dùng cáp đồng trục và chuyển gói đó sang đoạn mạng sử dụng

### **b/ HUB**

Một thiết bị liên kết mạng được sử dụng rất rộng rãi là HUB và việc sử dụng HUB như thành phần trung tâm trong cấu trúc hình sao.



#### **\*/ HUB chủ động**

Hầu hết các HUB đều là HUB chủ động, chúng tái tạo và truyền lại tín hiệu giống như bộ lặp. HUB thường có nhiều cổng nên thỉnh thoảng chúng còn được gọi là bộ lặp đa cổng.

#### **\*/ HUB thụ động**

Các HUB thụ động hoạt động như các điểm kết nối, chúng không tái tạo hoặc khuếch đại tín hiệu. Bảng phối dây là một ví dụ về HUB thụ động.

#### **\*/ HUB lai**

Các HUB thích ứng với nhiều loại cáp khác nhau được gọi là HUB lai.

### **c/ Cầu nối (bridge)**

Cầu nối là thiết bị hoạt động ở tầng liên kết dữ liệu. Cũng giống như bộ lặp, cầu nối có thể kết hợp nhiều đoạn mạng hoặc nhóm các mạng LAN khác nhau. Điểm hơn của cầu nối là chúng có thể phân chia mạng nhằm cô lập lưu lượng. Ví dụ, nếu lưu lượng từ một hoặc một nhóm máy tính trở nên quá tải và làm giảm hiệu suất toàn mạng thì cầu nối có thể cô lập máy tính hoặc bộ phận này.

Cầu nối được sử dụng để:

- Mở rộng khoảng cách của phân đoạn mạng, tăng số lượng máy tính trên mạng.
- Làm giảm hiện tượng tắc nghẽn do số lượng máy tính nối vào mạng quá lớn:

Cầu nối có thể tiếp nhận một mạng quá tải và chia nó thành hai mạng riêng biệt, nhằm giảm bớt lưu lượng truyền trên mỗi đoạn mạng và do đó mỗi mạng sẽ hoạt động hiệu quả hơn.

- Kết nối các phương tiện truyền dẫn khác nhau, chẳng hạn cáp xoắn đôi và cáp quang.
- Kết nối các đoạn mạng sử dụng phương thức truy nhập đường truyền khác nhau, chẳng hạn CSMA/CD và chuyển thẻ bài.

### c/ Bộ định tuyến (router)

Trong môi trường gồm nhiều đoạn mạng với giao thức và kiến trúc mạng khác nhau, cầu nối không thể đảm bảo truyền thông nhanh trong tất cả các đoạn mạng. Mạng có độ phức tạp như vậy cần một thiết bị không những biết địa chỉ của mỗi đoạn mạng, mà còn quyết định tuyến đường tốt nhất để truyền dữ liệu và lọc lưu lượng quảng bá trên các đoạn mạng cục bộ. Thiết bị như vậy được gọi là bộ định tuyến.

Bộ định tuyến là thiết bị hoạt động ở tầng mạng trong mô hình OSI. Điều này nghĩa là chúng có thể chuyển đổi và định tuyến gói dữ liệu qua nhiều mạng. Bộ định tuyến đọc thông tin địa chỉ mạng trong gói tin và vì chúng hoạt động tại tầng cao hơn so với cầu nối trong mô hình OSI, nên chúng truy cập nhiều thông tin khác.

Bộ định tuyến có thể cung cấp các chức năng của cầu nối:

- Lọc gói và cô lập lưu lượng mạng
- Kết nối nhiều đoạn mạng

Tuy nhiên do bộ định tuyến truy cập nhiều thông tin trong gói dữ liệu hơn so với cầu nối, nên chúng dùng các thông tin này để cải thiện việc phân phát gói dữ liệu. Bộ định tuyến được sử dụng trong các trường hợp mạng phức tạp vì chúng cung cấp chức năng quản lý lưu lượng tốt hơn cầu nối và không thực hiện phát quảng bá (chúng không truyền các gói tin quảng bá). Các bộ định tuyến có thể chia sẻ thông tin trạng thái và thông tin định tuyến với nhau và sử dụng thông tin này để bỏ qua các kết nối hỏng hoặc chậm.

### 1.4.7. Truyền tín hiệu

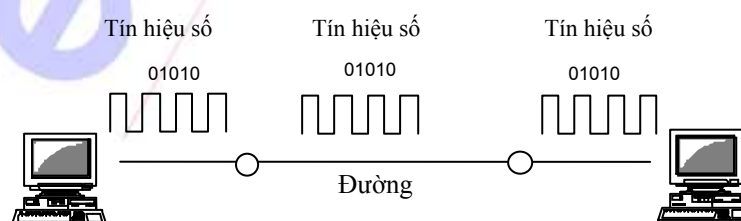
Mạng cục bộ sử dụng hai kỹ thuật truyền tín hiệu chính - đó là: truyền băng cơ sở (baseband) và truyền băng rộng (broadband). Tuy nhiên, hầu hết các mạng cục bộ đều sử dụng kỹ thuật truyền băng cơ sở.

#### a/ Truyền băng cơ sở

Hệ thống băng cơ sở truyền tín hiệu số ở một tần số đơn lẻ. Tín hiệu được truyền dưới dạng xung điện hoặc xung ánh sáng rời rạc. Đối với truyền băng cơ sở, toàn bộ dung lượng kênh truyền được dùng để truyền một tín hiệu dữ liệu. Tín hiệu số sử dụng toàn bộ băng thông của cáp, băng thông này tạo thành một kênh.

Kỹ thuật này có một số đặc điểm sau:

- Thiết bị truyền tín hiệu đơn giản và rẻ tiền.
- Vì kỹ thuật này sử dụng tín hiệu số nên nó tương thích với hệ thống máy tính.



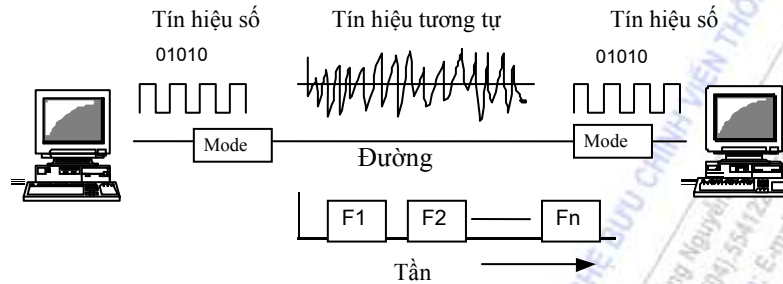
#### b/ Truyền băng rộng

Kỹ thuật truyền băng cơ sở



Hệ thống băng rộng truyền tín hiệu tương tự ở một khoảng tần số biến thiên. Với công nghệ truyền tương tự, tín hiệu được truyền liên tục. Tín hiệu được truyền qua phương tiện vật lý dưới dạng sóng điện từ hoặc sóng ánh sáng.

Khi sử dụng kỹ thuật truyền băng rộng, tín hiệu số từ máy tính phải được điều chế trước khi truyền đi.



### Kỹ thuật truyền băng rộng

Kỹ thuật này có những đặc điểm sau:

- Chi phí cao do cần một MODEM đặc biệt để điều chế tín hiệu.
- Cung cấp nhiều kênh truyền dẫn khác nhau do sử dụng các tần số sóng mang khác nhau để điều chế.

Vì kỹ thuật này sử dụng nhiều kênh nên việc truyền tín hiệu theo băng rộng cho phép truyền đồng thời nhiều dạng dữ liệu khác nhau, ngoài dữ liệu còn cho phép truyền thoại và hình ảnh cho hội nghị truyền hình và truyền hình cáp.

#### 1.4.8. Phương thức truyền dẫn

Việc truyền dữ liệu trong mạng máy tính được thực hiện một trong 3 loại sau: unicast, multicast, và broadcast. Trong các loại truyền này, một gói dữ liệu đơn được chuyển tới một hoặc nhiều nút.

Ở phương thức truyền unicast (truyền tới một điểm), một gói dữ liệu đơn được gửi từ nguồn đến một đích trên một mạng nào đó. Đầu tiên, nút nguồn đặt địa chỉ cho gói dữ liệu bằng cách sử dụng địa chỉ unicast. Sau đó gói dữ liệu này được chuyển tới mạng và cuối cùng mạng chuyển gói dữ liệu này đến đúng đích của nó.

Ở phương thức truyền multicast (truyền tới nhiều điểm), một gói dữ liệu đơn được sao chép và gửi tới một nhóm các nút trên một mạng nào đó. Đầu tiên, nút nguồn đặt địa chỉ cho gói dữ liệu bằng cách sử dụng địa chỉ multicast. Sau đó gói dữ liệu được gửi đến mạng, mạng sao chép gói dữ liệu này và gửi nó đến mỗi nút thành viên của địa chỉ multicast này.

Ở phương thức truyền broadcast (truyền quảng bá), một gói dữ liệu đơn được sao chép và gửi cho tất cả các nút trên một mạng nào đó. Đầu tiên, nút nguồn đặt địa chỉ cho gói dữ liệu bằng cách sử dụng địa chỉ broadcast. Sau đó gói dữ liệu được gửi đến mạng, mạng sao chép và gửi gói dữ liệu đến tất cả các máy tính ở mạng đích.

### 1.4.9. Mạng Internet/Intranet

#### *a/ Lịch sử phát triển của mạng Internet*

Năm 1968 Cơ quan các dự án nghiên cứu cấp cao (Advanced Research Project Agency-ARPA) của Bộ quốc phòng Mỹ đã đề ra dự án kết nối mạng giữa Bộ quốc phòng Mỹ với một số cơ sở nghiên cứu khoa học lớn ở Mỹ. Mùa thu 1969 việc kết nối giữa 4 trạm (Viện nghiên cứu Stanford, Đại học California ở Los Angeles, Đại học California ở Santa Barbara và Đại học Utah) thành công đánh dấu sự ra đời của ARPANET, tiền thân của INTERNET hôm nay. Giao thức truyền thông lúc đó được dùng là NCP (Network Control Protocol). Các nhà thiết kế ngay từ buổi ban đầu đó cũng đã nhận thức được rằng cần xây dựng “một mạng của các mạng”. Giữa những năm 70 họ giao thức TCP/IP được Vin cerf (đại học Stanford) và Robert Kahn phát triển, đến năm 1983 thì họ giao thức này hoàn toàn thay thế NCP trong ARPANET.

ARPANET thành công vang dội và đến năm 1983 thì được tách làm 2: một gọi là MILNET dành cho các địa điểm quân sự còn ARPANET mới dành cho các địa điểm phi quân sự. Tuy nhiên 2 mạng này vẫn còn liên kết với nhau nhờ vào giao thức IP (Internet Protocol - Giao thức Internet) và TCP (Transmission Control Protocol - Giao thức điều khiển truyền tin). Giao thức TCP là một giao thức trong đó các thông tin được số hóa và phân chia thành hàng loạt các gói để truyền đi, sau đó các gói này được lắp ráp lại tại nơi nhận. Hệ thống các gói được tạo ra như vậy nhằm đảm bảo cho thông tin được truyền đi kể cả khi một phần của mạng máy tính không hoạt động. Giao thức IP nhằm đảm bảo các thông tin đến đúng địa chỉ người nhận. Như vậy bất kỳ một máy tính nào tuân thủ các giao thức TCP và IP đều có thể liên hệ với nhau trong INTERNET.

Sau một thời gian, kế hoạch sử dụng ARPANet không thực hiện được như mong muốn vì một số lý do kỹ thuật và chính trị. Do vậy, năm 1986, NSF (National Science Foundation - Hội đồng khoa học Quốc gia) đã xây dựng mạng riêng lấy tên là NSFNet, mạng này hoạt động nhanh hơn nhiều và đã nối với các trung tâm tính toán lớn (tốc độ đường truyền là 1.5 Mb/s thay vì 560Kb/s trong ARPANet). Sự xuất hiện mạng xương sống NSFNet đã thúc đẩy sự tăng trưởng của Internet. Một xa lộ thông tin mới hình thành và nhiều trường đại học, viện nghiên cứu đã tham gia vào cộng đồng Internet. Và sau đó các tổ chức chính phủ và giới kinh doanh cũng vào cuộc và ngày càng chiếm tỷ trọng đáng kể trong thế giới Internet. Về mặt địa lý Internet cũng đã nhanh chóng vượt ra khỏi nước Mỹ và trở thành mạng toàn cầu với vài chục triệu người dùng như hiện nay. Đến năm 1990 thì quá trình chuyển đổi sang Internet hoàn tất và ARPANet ngừng hoạt động. NSFNet giờ đây cũng chỉ còn là một mạng xương sống của mạng Internet.

Về kiến trúc, Internet cũng đã có những thay đổi. Trước đây người ta còn có thể định nghĩa “Internet là mạng của tất cả các mạng sử dụng giao thức IP”. Nhưng ngày nay điều đó không còn chính xác nữa vì nhiều mạng với kiến trúc khác (không dùng IP) nhờ có cầu nối đa giao thức (multiprotocol gateway) nên vẫn có thể nối kết được vào Internet và sử dụng đầy đủ các dịch vụ thông tin trên Internet.

#### *b/ Internet và Intranet*

Internet là một xa lộ điện tử nối hàng ngàn mạng trên khắp thế giới lại với nhau với giao thức sử dụng chủ yếu là TCP/IP. Cũng có thể nói rằng Internet là tập hợp hàng ngàn mạng trên thế giới nối lại với nhau qua các bộ chọn đường (router) với giao thức sử dụng chủ yếu là TCP/IP. Hiện nay Internet bao gồm trên 15 000 mạng với không từ 25 đến 30 triệu người sử dụng. Các

dịch vụ chính trên mạng Internet là: thư điện tử (E-mail), tìm kiếm thông tin dựa trên siêu văn bản (WWW - World Wide Web), truyền tệp (FTP- File Transfer Protocol), truy nhập từ xa (TELNET),...

Bên cạnh khái niệm liên mạng Internet), khái niệm mạng nội bộ (Intranet) cũng xuất hiện. Intranet là mạng dùng riêng cho các yêu cầu hoạt động nội bộ của một đơn vị xã hội, có thể là mạng cục bộ hay mạng diện rộng, trong đó sử dụng các công nghệ cốt lõi của Internet.

Một câu hỏi đặt ra: Ai là người quản lý Internet?

Thực tế là không có một cơ quan quản lý tối cao cho toàn bộ mạng Internet. Một tổ chức có vai trò điều phối tối cao các hoạt động của Internet là Hiệp hội Internet viết tắt là ISOC (Internet Society), là tổ chức phi lợi nhuận tập hợp một số tổ chức và cá nhân tự nguyện tham gia vào các hoạt động nhằm khuyến khích và phát triển sử dụng Internet. Cơ quan lãnh đạo cao nhất của ISOC là ban kiến trúc Internet viết tắt là IAB (Internet Architecture board). Việc phân phối địa chỉ cho các máy tính của người sử dụng (host) nối vào Internet ban đầu do chính ISOC trực tiếp làm. Nhưng từ năm 1992 do sự tăng trưởng quá nhanh của Internet nên công việc đó được phân cấp cho các Trung tâm thông tin mạng viết tắt là NIC (Network Information Center). NIC của khu vực Châu Á - Thái Bình Dương - gọi là APNIC có trụ sở tại Tokyo, Nhật bản. Hiện nay ở khu vực Châu Á - Thái Bình Dương chỉ có 2 NIC quốc gia của Nhật bản và của Hàn Quốc. Hiện tại APNIC ở Tokyo vẫn chịu trách nhiệm điều hành và phân phối địa chỉ cho các Host ở Việt nam.

### ***c/ Một số dịch vụ cơ bản của Internet***

#### **\*/ Thư điện tử (E-mail)**

Thư điện tử (Electronic Mail gọi tắt là E-mail) là dịch vụ thông dụng nhất trên Internet. Để gửi hay nhận thư, bước đầu tiên là xác định chính xác địa chỉ của thư cần gửi đến. Cấu trúc của thư điện tử như sau:

*Tên người sử dụng @ Tên đầy đủ của vùng (domain)*

Ví dụ : *hungnt@itac.com.vn*

Trong đó từ phải sang trái theo thứ tự:

. vn: chỉ vùng địa lý, vn viết tắt của Việt Nam. Phần cuối của tên vùng thường định nghĩa cho hoạt động: EDU, NET, COM (đối với nước Mỹ) hoặc từ viết tắt cho quốc gia.

. com: chỉ mạng hoạt động dịch vụ Internet là đào tạo (EDU), kinh doanh (COM),...

. itac: chỉ tên một máy chủ của công ty iTAC

. hungnt: chỉ tên hộp thư của một Account truy nhập của thuê bao.

Để gửi thư điện tử máy gửi thư và máy nhận thư không cần phải liên kết trực tiếp. Email là một dịch vụ kiểu lưu (POP3) và chuyển tiếp (SMTP). Thư điện tử được chuyển từ máy này qua máy khác cho tới máy đích (giống như trong hệ thống bưu chính thông thường: thư được chuyển đến tay người nhận sau khi đã đi qua một số bưu cục trung chuyển). Mỗi người dùng (client) đều phải kết nối với một E-mail Server gần nhất (đóng vai trò bưu cục địa phương). Sau khi soạn xong thư và đề rõ địa chỉ đích (người nhận), người sử dụng sẽ gửi thư tới E-mail Server của mình. E-mail Server này có nhiệm vụ chuyển thư đến đích hoặc đến một E-mail Server trung gian khác.

Thư sẽ chuyển đến E-mail Server của người nhận và được lưu tại đó. Đến khi người nhận thiết lập một cuộc gọi nối tới máy E-mail Server đó thì thư sẽ được chuyển về máy của người nhận và đồng thời vẫn còn lại trên máy chủ cho đến khi máy chủ được dọn dẹp. Giao thức truyền thông sử dụng cho E-mail của Internet là SMTP (Simple Mail Transfer Protocol). Cấu trúc của một thư điện tử thường gồm 2 phần: phần đầu thư (header) và phần thân thư (body)

**\*/ Mạng thông tin toàn cầu (World Wide Web; viết tắt là WWW):**

Đây là dịch vụ mới và mạnh nhất trên Internet. WWW được xây dựng trên kỹ thuật siêu văn bản (Hypertext). Trong các trang văn bản có chứa những từ mà khi ta chọn có thể gọi ra thành một trang thông tin mới có nội dung chi tiết hơn. Trên cùng một trang thông tin có thể có nhiều dạng thông tin khác nhau như văn bản, ảnh hay âm thanh. Để xây dựng các trang dữ liệu với các kiểu dữ liệu khác nhau như vậy, WWW sử dụng một ngôn ngữ có tên là ngôn ngữ siêu văn bản HTML (HyperText Markup Language)

HTML là một ngôn ngữ định dạng (hay đánh dấu - Markup). Mỗi tệp tin văn bản được đánh dấu bằng các thẻ (tag) HTML cho phép người sử dụng có thể đọc được chúng trên máy tính của mình hay qua mạng bằng một phần mềm gọi là bộ duyệt Web (Web browser). Thẻ HTML là một đoạn mã được giới hạn bởi các dấu ngoặc nhọn "<" và ">". Chương trình Web browser sẽ đọc các thẻ khi thực hiện định dạng các tệp tin trên màn hình. Các tài liệu có trong WWW đều là các tệp tin HTML.

Để thực hiện việc truy nhập, liên kết các tài nguyên thông tin khác nhau theo kỹ thuật siêu văn bản, WWW sử dụng khái niệm URL (Uniform Resource Locator). Đây chính là một dạng tên để định danh duy nhất cho một tài liệu hoặc một dịch vụ trong Web. Cấu trúc của một URL thường bao gồm các thành phần: Giao thức Internet được sử dụng, vị trí của Server (domain name), tài liệu cụ thể trên Server (path name) và có thể có thêm các thông tin định danh khác.

Ví dụ một tài liệu Web có thể có URL như sau:

`http://www.itac.com.vn/training/library/doc/it/lan.html`

Trong đó:

- "http:" là chỉ giao thức
- "www.itac.com.vn" là chỉ tên Domain của máy chủ WWW
- "/training/library/doc/it/lan.html" là chỉ đường dẫn đến tài liệu trên server

*http (HyperText Transfer Protocol)* là một giao thức truyền thông sử dụng cho Web. Nhưng vì Web có thể chấp nhận các giao thức Internet khác như ftp, gopher, WAISE,...) nên URL có thể có một tiền tố khác với http, chẳng hạn

`gopher://gopher.msu.edu/...`

Như vậy, với địa chỉ URL, Web cho phép bạn truy nhập đến các tài nguyên thông tin từ các dịch vụ khác của Internet ở trên các Server khác nhau.

Hoạt động của Web cũng dựa trên mô hình client/server. Tại trạm client, người sử dụng sẽ dùng Web browser để gửi yêu cầu tìm kiếm các tệp tin HTML đến Web server ở xa trên mạng Internet nhờ có địa chỉ URL. Web server nhận các yêu cầu đó và thực hiện rồi gửi kết quả về cho

Web client. ở đây Web browser sẽ biên dịch các thẻ HTML và hiển thị nội dung các trang tài liệu trên màn hình.

Cho đến nay, nhiều hãng phần mềm đã chế tác các browser cho Web, thí dụ: Mosaic của NSCA, Navigator của Netscape, Internet Explorer của Microsoft, Web Access của Novell,... Trong khi đó nhiều hãng, nhiều tổ chức và c người sử dụng cũng đang rầm rộ phát triển các cơ sở dữ liệu cho Web với nhiều chủ đề khác nhau, hứa hẹn những khả năng ứng dụng ngày càng phong phú của Web trong tung lai.

### **\*/ Dịch vụ truyền tệp (File Transfer Protocol viết tắt là FTP)**

Dịch vụ truyền tệp trên Internet được đặt tên theo giao thức mà nó sử dụng là FTP (File Transfer Protocol).

FTP cho phép chuyển các tệp từ một trạm này sang một trạm khác, bất kể các trạm đó ở đâu và dùng hệ điều hành gì, chỉ cần chúng đều được nối với Internet và có cài đặt FTP.

FTP là một chương trình phức tạp vì có nhiều cách khác nhau để xử lý tệp và cấu trúc tệp, chưa nói đến có nhiều cách lưu trữ tệp khác nhau (Binary hay ASCII, nén hay không nén, v.v...)

Để khởi động FTP, từ trạm làm việc ta gõ

```
ftp <domain name or IP address>
```

ftp sẽ thiết lập liên kết với trạm xa và lúc đó bạn sẽ phải làm các thao tác quen thuộc để đăng nhập vào hệ thống (login/password).

Sau khi trên màn hình hiển thị dấu nhắc ftp> bạn có thể gõ tiếp các lệnh cho phép truyền tệp theo cả 2 chiều. Để chuyển một tệp từ máy của bạn đến trạm xa thì dùng lệnh put, ngược lại muốn lấy một tệp từ trạm xa về thì dùng lệnh get với cú pháp tương ứng như sau:

```
ftp> put <source file> <destination file>
```

```
ftp> get <source file> <destination file>
```

Nếu không chỉ ra tên tệp đích thì bản sao sẽ lấy cùng tên như tệp nguồn. Ngoài put và get, ftp còn nhiều lệnh tiện ích khác

Trong trường hợp bạn chưa đăng ký (chưa có account) trên Internet thì bạn sẽ không thể sử dụng ftp như trên được vì bạn chưa có login name/password để đăng nhập vào máy. Tuy nhiên các tác giả của ftp đã nghĩ đến điều đó và cung cấp dịch vụ gọi là “FTP vô danh” (Anonymous FTP) cho phép những người sử dụng không có login name/password có thể truy nhập tới một số tệp nhất định trên máy. Tất nhiên là có sự hạn chế: những người đó chỉ có thể sao chép các tệp chứ không thể tạo được các tệp mới hoặc biến đổi các tệp đã có. Khi dịch vụ này được cung cấp thì sẽ có một login name đặc biệt gọi là anonymous và nếu bạn dùng nó làm login name của bạn thì ftp chấp nhận một xâu ký tự bất kỳ như là password của bạn. Sau khi đã đăng nhập như một “anonymous” bạn sẽ được phép lấy các tệp mà người ta đã dành riêng cho những kẻ vô danh.

### **\*/ Dịch vụ truy nhập từ xa-TELNET**

TELNET là một giao thức để truy nhập vào một máy tính ở xa để khai thác tài nguyên ở máy đó giống như ngồi làm việc ở máy mình. Điều cần thiết là máy tính cần nối vào Internet và

có chương trình TELNET. Giống với FTP khi cần vào một máy tính nào đó cần nhập tên truy nhập và mật khẩu hoặc khai báo địa chỉ e-mail.

Để sử dụng giao thức TELNET có nhiều phần mềm, phổ biến nhất là chương trình TELNET.EXE của WINDOWS95 (hoặc WINDOWS NT). Mẫu lệnh đơn giản của chương trình TELNET có dạng:

*Telnet <domain name or IP address>*

Khi truy nhập được vào máy từ xa, trả lời ID và mật khẩu, lúc đó có thể khai thác máy chủ như là làm việc trên máy chủ vậy.

### **\*/ Dịch vụ nhóm tin-USENET (USER NETWORK)**

Đây là dịch vụ cho phép nhiều người sử dụng ở nhiều nơi khác nhau có cùng mối quan tâm có thể tham gia vào một “nhóm tin” và trao đổi các vấn đề quan tâm. Có thể có nhiều nhóm tin khác nhau như: nhóm tin nhạc cổ điển, nhóm tin về hội họa, nhóm tin về thể thao... Tên (địa chỉ) của các nhóm tin được cấu trúc theo kiểu phân cấp. Nhóm rộng nhất sẽ đứng đầu tên, theo sau là một số tùy ý các nhóm “con”, “cháu”,... thí dụ

*music.classic*

Dịch vụ các nhóm tin sử dụng giao thức NNTP (Network News Transfer Protocol) của Internet.

Trên Internet có nhiều Server tin (news server) khác nhau, trong đó tin tức được thu thập từ nhiều nguồn khác nhau. Người sử dụng tung tác với một server tin thông qua một chương trình được đặt tên là “chương trình đọc tin” (News reader). Và người sử dụng cũng chỉ biết đến một server tin duy nhất, đó là server mình kết nối vào. Mọi sự trao đổi, tung tác giữ các server tin và nhóm tin là hoàn toàn “trong suốt” đối với người sử dụng. Với dịch vụ này, một người sử dụng có thể nhận được các thông tin mà mình quan tâm của nhiều người từ khắp mọi nơi, đồng thời có thể gửi thông tin của mình đi cho những người có cùng mối quan tâm.

## Chương 2: HỆ ĐIỀU HÀNH

### 2.1. KHÁI NIỆM VÀ PHÂN LOẠI

#### 2.1.1. Khái niệm

Hệ điều hành là công cụ để tương tác giữa người sử dụng và máy tính. Máy tính muốn hoạt động được phải có chương trình hay còn gọi là phần mềm. Thông thường phần mềm chúng ta cảm thấy dùng nhiều nhất và nhìn thấy rõ hiệu quả nhất là phần mềm ứng dụng: Các phần mềm quản trị cơ sở dữ liệu viết bằng Foxpro giúp ta quản lý những cơ sở dữ liệu lớn một cách tiện lợi, Winword giúp chúng ta soạn thảo các văn bản, dùng trang tính điện tử excel chúng ta có thể tạo các báo cáo kế toán rất đẹp và chính xác... Tuy nhiên chúng ta không biết rằng các phần mềm đó đã được sự hỗ trợ rất đặc lực của một **phần mềm hệ thống là hệ điều hành (OS - Operating System)**. Thực ra các phần mềm ứng dụng thường chỉ có thể khởi động từ hệ điều hành. Nghĩa là sau khi hệ điều hành đã làm chiếc cầu nối với phần cứng máy tính, đã kiểm tra và đưa hàng loạt thiết bị như bộ nhớ, màn hình, máy in, ổ đĩa, bàn phím... vào trạng thái sẵn sàng làm việc. Ngay cả khi đang hoạt động các chương trình ứng dụng vẫn thường xuyên cần đến sự trợ giúp của OS. OS như một người đầy tớ cần mẫn: chỉ cần chương trình ứng dụng rung chuông là anh ta có mặt ngay. OS có rất nhiều dịch vụ có đánh số thứ tự, các chương trình ứng dụng có thể gọi các dịch vụ này thông qua các lệnh ngắt tương ứng. Những lệnh như mở file, đóng file, copy file... các chương trình ứng dụng đều gọi đến dịch vụ OS. Có rất nhiều thao tác của người sử dụng cần làm bên ngoài các chương trình ứng dụng: tạo các thư mục, copy các file, xóa file trên đĩa để lấy chỗ trống... nghĩa là OS không những là cầu nối giữa phần cứng và các phần mềm ứng dụng, mà còn là cầu nối giữa người sử dụng với máy.

**Tóm lại chúng ta có thể hiểu hệ điều hành như sau:**

*Hệ điều hành - theo nghĩa đầy đủ nhất của từ này - là phần mềm đầu tiên và quan trọng nhất của hệ thống máy tính, là cầu nối giữa phần cứng và các phần mềm khác, giữa người sử dụng với máy tính. Hệ điều hành quản lý các tài nguyên của máy tính: bộ nhớ, đĩa, máy in... để các phần mềm khác có thể sử dụng. Nói một cách ngắn gọn hơn, hệ điều hành là một tập hợp các chương trình lo việc điều khiển hoạt động của máy tính và tạo môi trường để các phần mềm khác chạy được.*

#### 2.1.2. Phân loại hệ điều hành.

Người ta phân loại hệ điều hành theo khả năng thực hiện cùng lúc một hay nhiều chương trình hoặc khả năng quản lý một hay nhiều máy tính. Theo tiêu chuẩn thứ nhất ta có 2 loại hệ điều hành:

HDH đơn nhiệm: tại một thời điểm chỉ có một chương trình được thực hiện. Các hệ điều hành trên máy tính cá nhân: PC-DOS của IBM và MS-DOS của Microsoft (DOS = Disk Operating System)

HDH đa nhiệm: tại một thời điểm có thể thực hiện nhiều chương trình (multitasking) và các chương trình có thể trao đổi thông tin cho nhau. Ví dụ các hệ điều hành Windows của Microsoft, UNIX...

Theo tiêu chuẩn thứ 2 ta có:

Hệ điều hành cho máy đơn lẻ như PC-DOS, MS-DOS, WINDOWS,...

Hệ điều hành mạng: NOVELL NETWARE, UNIX, WINDOWS, WINDOWS NT...

## 2.2. HỆ ĐIỀU HÀNH MS - DOS

### 2.2.1. Các khái niệm cơ bản của MS-DOS

#### 2.2.1.1. Các thành phần cơ bản của MS-DOS

**b) Boot record:** Môđun này chính là thẻ khởi động sẽ khởi động hệ điều hành. Nhiệm vụ của Boot Record là kiểm tra trong thư mục của đĩa xem có các tệp hệ thống IO.SYS, MSDOS.SYS không? Hai tệp hệ thống này đều được bảo vệ chống xoá, chống ghi đè, mang dấu hiệu hệ thống và được ẩn dấu. Nếu DOS tìm thấy thì nạp hai tệp IO.SYS, MSDOS.SYS vào bộ nhớ trong, sau đó chuyển điều khiển cho IO.SYS

**c) Tệp hệ thống IO.sys:** IO.SYS là sự mở rộng của ROM-BIOS. Cả ROM-BIOS và IO.SYS đều chứa các chương trình con điều khiển và xử lý các thiết bị vào ra ngoại vi. Nhưng tại sao lại cần chia thành hai môđun: một chương trình thì ghi cứng trong ROM-BIOS, chương trình kia lại để trong tệp ghi lên đĩa mềm? Đây là một đặc trưng khá khôn ngoan, mềm dẻo của các chuyên gia Microsoft nhằm dành một phần chương trình điều khiển ngoại vi cho vào tệp IO.SYS để dễ sửa đổi và cải cách chúng một cách đơn giản. Ba nhiệm vụ chính của IO.SYS mà ROM-BIOS không đảm nhiệm là:

- Nhiệm vụ 1: Phục vụ yêu cầu riêng đặc biệt của từng hệ điều hành. Các hệ điều hành khác DOS như CP/M-86, UCSD-p System đều có thể sử dụng ROM-BIOS trong máy tính PC. Nói cách khác, IO.SYS là một tệp điều khiển vào ra đặc thù của những nhà viết chương trình hệ điều hành khác nhau, để sửa đổi và cải tiến, nâng cấp.

- Nhiệm vụ 2: Sửa lỗi xuất hiện trong ROM-BIOS. Mặc dù người ta đã kiểm tra cực kì cẩn thận các chương trình nạp trong ROM-BIOS vì nó là cố định, tuy nhiên cũng không tránh khỏi sai sót (ví dụ sự cố Y2K). Vì thế cách thông thường nhất và đơn giản là sửa và cập nhật trên tệp IO.SYS hơn là sửa nội dung trong phần cứng ROM-BIOS. Việc sửa lỗi này bằng cách cho thực hiện đặt có hiệu chỉnh các véc tơ ngắt, còn các thao tác vào ra cơ sở thì lại gọi thực hiện các chương trình từ ROM-BIOS. (Hệ thống gọi các chương trình liên quan đến các chương trình trong ROM-BIOS luôn hoạt động thông qua sự ngắt, không thông qua địa chỉ ROM). Thực ra để sửa lỗi trong ROM-BIOS chúng ta có thể đổi ROM-BIOS với chương trình mới, nhưng sử dụng phương pháp vừa nêu có ưu điểm hơn.

- Nhiệm vụ thứ 3: Điều khiển các ngoại vi mới được nối với máy tính PC, máy vẽ, máy quét. Việc quản lý và điều khiển các thiết bị mới để giải quyết nếu các nhà lập trình hệ thống sửa IO.SYS hơn là sửa chương trình trong ROM-BIOS. Cùng với việc xuất hiện các thiết bị vào ra mới có trên thị trường nối với máy tính PC, người ta phải liên tục sửa các tệp IO.SYS (thậm chí sửa cả chương trình khác của DOS) cho nên tệp IO.SYS (và cả MSDOS.SYS) luôn thay đổi về



nội dung (thể hiện bởi số Byte và ngày tháng tạo tệp). Sự sửa đổi này của hãng làm phần mềm hệ thống Microsoft hay IBM không có gì khó cả nhưng với người sử dụng thì không thể làm được vì khả năng và vì sự thiếu thông tin hệ thống thường được giữ bí mật bởi các nhà sản xuất.

Tuy nhiên, để đạt được sự thành công trong việc chiếm lĩnh thị trường hệ điều hành thì các nhà sản xuất phải chiều ý khách hàng để cho người dùng khả năng tự mở rộng, thay đổi cấu hình máy tính PC ghép nối với ngoại vi. Khi nhận thức được điều này kể từ DOS 2.0 trở đi, người ta đã sửa đổi DOS sao cho khi IO.SYS bắt đầu làm việc đầu tiên nó cập nhật thông tin từ tệp cấu hình (CONFIG.SYS) trên đĩa (nếu có). Trong tệp CONFIG.SYS, người dùng dễ dàng đưa vào các thiết bị ngoại vi mới, các chế độ đặt, các tham số hệ thống. Ngoài những điều đã nói, tệp cấu hình còn chứa những chương trình điều khiển thiết bị mà trong quá trình thao tác vào ra cơ sở phải có sẵn để dùng đến chúng. Điều này có thể hiểu rằng những chương trình điều khiển thiết bị khi đặt trong CONFIG.SYS mà được gọi đến sẽ đổ vào bộ nhớ trong của máy tính PC như là các chương trình phụ lục của IO.SYS. Theo cách đó, hệ thống máy tính có thể phát triển theo kiểu mô đun với những ngoại vi mới mà không cần động vào các tệp hệ thống của DOS. Tuy nhiên, thông thường bất kỳ một chương trình nào chạy trên IBM-PC đều cần đến kiểu DOS truyền thống với các thao tác cơ sở vào ra.

**d) Tệp hệ thống MSDOS.SYS:** Trong hai tệp hệ thống thì tệp thứ hai này đóng vai trò trung tâm bao gồm các chương trình con dịch vụ còn lại của DOS. Người ta có thể ghép IO.SYS với MSDOS.SYS thành một tệp nhưng để đảm bảo mô đun hoá nên chúng được chia riêng để dễ sửa đổi.

Việc sử dụng các chương trình con này được thực hiện bởi ngắt mềm. Mỗi một dịch vụ DOS thường là một chương trình con và ứng với nó được đặt tên thành một hàm của DOS nhằm thực hiện một số các dịch vụ (nhiệm vụ) trực tiếp và cơ bản như: đọc đầu vào từ bàn phím, viết đầu ra tới màn hình, viết thông tin qua kênh thông tin không đồng bộ, xử lý đầu ra máy in. Trong số các dịch vụ đó, thường chứa các thao tác con mang tính logic như: mở đóng các tệp trên đĩa, tìm bảng thư mục, xoá, tạo tệp song song với việc viết, đọc các số liệu.

Ở mức lập trình, người lập trình có thể sử dụng các dịch vụ DOS thay vì phải viết chương trình để thực hiện các thao tác hay dịch vụ đã nêu.

Các chương trình DOS ở mức cao hơn thường làm như vậy (ví dụ các lệnh DIR, COPY của DOS thực hiện dịch vụ tìm bảng thư mục).

Nếu chúng ta cũng nghiên cứu các dịch vụ của ROM-BIOS thì có một số phần chức năng trùng nhau nhưng không đáng kể. Về cơ bản các dịch vụ trong ROM-BIOS và các dịch vụ chức năng trong DOS là khác nhau.

#### **e) Chương trình COMMAND.COM và các lệnh nội trú**

COMMAND.COM có một vài chức năng:

Chức năng chính: Command.com là bộ xử lý lệnh. Chương trình này đọc các lệnh đưa vào từ bàn phím và bộ xử lý lệnh của COMMAND.COM sẽ quyết định cần phải làm gì với lệnh đó. Command.com sử dụng một bảng bên trong nó để chứa tên gọi của các lệnh nội trú.

Các lệnh nội trú là các lệnh mà người dùng hay dùng như CLS, DIR, CD, MD, RD, DEL, COPY, TYPE, DATE, TIME, VER, VOL, PROMPT, REN, ... Chính vì lý do được dùng thường xuyên nên người ta bố trí chúng trong COMMAND.COM và lưu trú ở bộ nhớ trong để tiết kiệm thời gian tìm kiếm trên đĩa. Khi ta gõ lệnh thì trước hết Command.com sẽ tìm tên lệnh trong bảng lệnh nội trú, nếu có thì cho thực hiện lệnh đó ngay; nếu không thì DOS sẽ tìm tên file tương ứng ở ổ đĩa bên ngoài trong thư mục hiện thời hoặc trong các thư mục đã đặt đường dẫn. Ba loại đuôi chương trình mà DOS có thể thực hiện theo thứ tự ưu tiên là: .COM, .EXE và .BAT.

Khi chương trình COMMAND.COM tìm thấy tệp chương trình trên đĩa theo một trong ba loại định hình trên, nó sẽ đổ tệp đó xuống bộ nhớ và chuyển điều khiển cho chương trình tức là cho chương trình chạy.

**f) Lệnh ngoại trú:** Là các lệnh còn lại của hệ điều hành được bố trí trên đĩa ở dạng các tệp chương trình. Tên các lệnh ngoại trú có đuôi .COM hoặc .EXE. Ví dụ các lệnh: SYS, FORMAT, LABEL, FDISK, XCOPY, ATTRIB, SCANDISK, DEFRAG, UNDELETE, DISKCOPY... Theo một quan điểm nào đó (về phía người dùng) thì các lệnh ngoại trú là phần chủ yếu của hệ điều hành đặc biệt là những lệnh nếu thiếu nó chúng ta khó sử dụng hệ điều hành như FORMAT, FDISK. Một quan niệm khác, thì các lệnh ngoại trú chỉ là các chương trình làm nhiệm vụ như các công cụ thông thường, như các chương trình khác do DOS quản lý; mỗi khi gọi đến, nó được đổ xuống bộ nhớ trong và chạy, sau đó có thể được các chương trình gọi sau đó ghi đè lên.

#### 2.2.1.2. Tổ chức đĩa từ:

**Tên ổ đĩa:** Hệ điều hành nhận biết đĩa mềm hay ổ đĩa cứng thông qua tên của chúng. Ví dụ ổ đĩa mềm có tên A hay B, ổ đĩa cứng đặt tên là C, D, ổ đĩa CD có tên là E,...

**Tập tin (file):** Tập (tệp) tin là nơi lưu trữ thông tin bao gồm chương trình, dữ liệu, văn bản,... Mỗi tập tin có một tên riêng phân biệt. Tên tập tin thường có 2 phần: phần tên (name) và phần mở rộng (extension). Phần tên là bắt buộc phải có của một tập tin, còn phần mở rộng thì có thể có hoặc không.

- Phần tên là một dãy có từ 1 đến tối đa 8 ký tự có thể là: các ký tự chữ từ A đến Z, các chữ số từ 0 đến 9, các ký tự khác như #, \$, %, ~, ^, @, (, ), !, \_

- Phần mở rộng có từ 0 đến tối đa 3 ký tự trong số các ký tự nêu ở trên.

- Giữa phần tên và phần mở rộng có một dấu chấm (.) ngăn cách.

Tên tập tin không chấp nhận các trường hợp sau:

- Có khoảng trống trong tên file

- Trùng tên với các lệnh của DOS và lệnh điều khiển thiết bị: CON, PRN, ...

- Có chứa các ký tự như ., ?, \*, :, >, <, /, \, [, ], +, ;,

Phần mở rộng có thể được xem gần như họ trong tên người. Ta có thể căn cứ vào phần mở rộng để xác định kiểu của file: · COM, EXE, BAT : Các file khả thi và lệnh đó chạy trực tiếp được trên MS-DOS

· TXT, DOC, ... : Các file văn bản

· PAS, BAS, ... : Các file chương trình PASCAL, BASIC, ...

- WK1, XLS, ... : Các file chương trình bảng tính LOTUS, EXCEL ...
- DBF, DAT, ... : Các file dữ liệu

Các ký tự đặc biệt trên file: DOS dùng các ký tự sao (\*) và chấm hỏi (?) để mô tả một tập hợp file. Ý nghĩa như sau:

- Dấu \* dùng để đại diện cho một chuỗi ký tự bất kỳ và thay cho phần còn lại của tên file hoặc phần mở rộng của file tại vị trí nó xuất hiện trở về sau.
- Dấu ? dùng để đại diện cho một ký tự bất kỳ tại vị trí nó xuất hiện.

Ví dụ: Trong đĩa của bạn có các tập tin:

BAOCAO2.TXT, VANBAN.TXT, VANCAO.THO, SOLIEU.DAT

+ Ký hiệu BAOCAO?.TXT đại diện cho các tập tin BAOCAO1.TXT, BAOCAO2.TXT

+ Ký hiệu \*.TXT đại diện cho BAOCAO1.TXT, BAOCAO2.TXT, VANBAN.TXT

+ Ký hiệu ???CAO?.\* đại diện cho BAOCAO1.TXT, BAOCAO2.TXT, VANCAO.THO

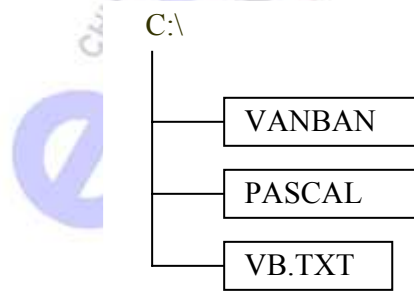
+ Ký hiệu \*.\* hoặc duy nhất một dấu chấm . đại diện cho tất cả các tập tin trên đĩa

**Thư mục (directory):** Thư mục là nơi cất giữ các tập tin theo một chủ đề nào đó theo ý người sử dụng. Đây là biện pháp giúp ta quản lý được tập tin, dễ dàng tìm kiếm chúng khi cần truy xuất. Các tập tin có liên quan với nhau có thể được xếp trong cùng một thư mục.

Bản thân mỗi đĩa mang một thư mục chung gọi là thư mục gốc (root directory). Thư mục gốc không có tên riêng và được ký hiệu là \ (dấu xô phải: backslash). Dưới mỗi thư mục gốc có các file trực thuộc và các thư mục con (sub-directory). Trong các thư mục con cũng có các file trực thuộc và thư mục con của nó. Thư mục chứa thư mục con gọi là thư mục cha (parent directory).

Thư mục đang làm việc gọi là thư mục hiện hành hay thư mục hiện thời (current directory). Ổ đĩa đang làm việc gọi là ổ đĩa hiện thời.

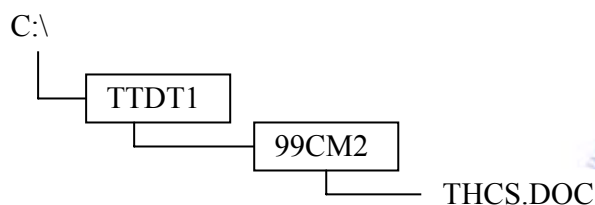
Ta có thể diễn tả cấu trúc thư mục và file chứa trong nó qua hình ảnh cây thư mục (directory tree). Ví dụ trên đĩa C, ta có 2 thư mục con của thư mục gốc là VANBAN và PASCAL, một tệp là VB.TXT



**Đường dẫn (path):** Đường dẫn dùng để chỉ dẫn lộ trình tới thư mục hoặc truy nhập tới một tệp nào đó. Đường dẫn là một dãy các thư mục cách nhau bởi dấu \, thư mục đứng sau là con của

thư mục đứng trước. Nói cách khác, đường dẫn dùng để chỉ định đường đi tới thư mục cần đến, mỗi tệp nằm trong bộ nhớ được xác định bằng tên tệp và “đường dẫn” dẫn đến tệp đó.

Ví dụ: Ta có cây thư mục sau:



Trong cây thư mục trên đây ta gọi C:\TTDT1\99CM2\ là đường dẫn của tệp THCS.DOC.

Giả sử <tên tệp> được chỉ ra như sau

[Tên ổ đĩa]\[tên thư mục 1]\[tên thư mục 2]...\[Tên thư mục N]\<Tên tệp>

trong đó thư mục sau là thư mục con của thư mục trước. Khi đó

[Tên ổ đĩa]\[tên thư mục 1]\[tên thư mục 2]...\[Tên thư mục N]\ được gọi là đường dẫn của <tên tệp>.

## 2.2.2. Quá trình làm việc của MS-DOS:

Để có thể yêu cầu máy tính thực hiện được các ứng dụng (các chương trình), thao tác đầu tiên ta cần là khởi động máy. Khi khởi động máy HĐH được nạp vào bộ nhớ của máy tính, sau đó chúng ta có thể khai thác các khả năng của hệ điều hành.

Muốn khởi động máy ta phải có đĩa hệ thống hay còn gọi là đĩa khởi động. Đĩa hệ thống là đĩa có tối thiểu 3 tệp tin **io.sys**, **msdos.sys**, **command.com**, trong đó hai tệp tin đầu cần chiếm một vị trí đặc biệt trên đĩa

Khi khởi động máy tính thường thường máy sẽ tìm đĩa hệ thống theo thứ tự sau: đĩa mềm, đĩa CD-ROM, đĩa cứng. Khi đó MS-DOS sẽ nạp ba tệp tin COMMAND.COM, IO.SYS và MSDOS.SYS vào bộ nhớ RAM của máy tính. Sau khi DOS đã được đưa vào bộ nhớ để sẵn sàng làm việc dấu mời sẽ xuất hiện trên màn hình chờ để người sử dụng đánh lệnh vào. Những phần chính của DOS sẽ lưu trú trong bộ nhớ cho đến khi ta tắt máy.

## 2.2.3. Các lệnh cơ bản của MS-DOS

### 2.2.3.1. Lệnh nội trú (internal command)

Lệnh nội trú là những lệnh nằm thường trực trong bộ nhớ máy khi đã được khởi động và sẵn sàng thực hiện lệnh khi ta gọi đến. Lệnh nội trú nằm trong phần khởi động của MS-DOS chứa trong các file COMMAND.COM, IO.SYS và MSDOS.SYS.

+ Tệp tin COMMAND.COM là tệp tin quan trọng nhất, có nhiệm vụ thông dịch lệnh và xử lý các lệnh nội trú. Khi khởi động máy, COMMAND.COM được nạp vào bộ nhớ RAM. Khi nhận được lệnh từ bàn phím, tệp tin này sẽ nhận diện và điều khiển việc thi hành các lệnh từ người sử dụng.

+ Hai tập tin IO.SYS và MSDOS.SYS là hai tập tin ẩn, nó cũng được nạp vào RAM khi khởi động máy. Chúng có nhiệm vụ quản lý và điều khiển các thiết bị ngoại vi và các tập tin trên đĩa.

Tập lệnh nội trú gồm các lệnh chính thường dùng như:

- Các lệnh liên quan đến thư mục: DIR, CD, MD, RD, PATH, TREE, ...
- Các lệnh liên quan đến file: TYPE, DEL, COPY, REN, ...
- Các lệnh thời gian: TIME, DATE
- Các lệnh khác: PROMPT, CLS, VER, VOL,...

### Các lệnh nội trú cơ bản:

Chú ý: Trong phần này chúng tôi có sử dụng một số qui ước như sau:

**drive:** Tên ổ đĩa

**path:** Đường dẫn

↵: Phím Enter

<Những thông tin nằm trong phần ngoặc nhọn>: Bắt buộc phải có

[Những thông tin nằm trong phần ngoặc vuông]: Không bắt buộc (tùy thuộc vào từng yêu cầu của câu lệnh)

#### \* Chuyển ổ đĩa:

Cú pháp: Ký tự ổ đĩa : > hoặc < drive : >

Với drive: là ổ đĩa cần chuyển đến.

Muốn chuyển từ ổ đĩa C sang ổ đĩa A ta gõ C:\> A: rồi nhấn nút Enter. Tương tự chuyển từ A: sang U: ta gõ lệnh U:

#### \* Liệt kê thư mục (DIR)

Hiện thị danh sách các tập tin và các thư mục con có trong thư mục.

Cú pháp: DIR [drive :] [path][/P][/W][/A : attribs][/O : sortorder]

Ghi chú:

/P : hiển thị từng trang màn hình (Page)

/W : hiển thị theo hàng ngang (Wide), lược bỏ bớt số liệu về kích thước byte, ngày, giờ.

/A : hiển thị thuộc tính (Attribut) của file

/O: các ý định hiển thị trật tự sắp xếp:

N : theo alphabetic tên file;

E : theo alphabetic tên phần mở rộng

S : theo kích thước bytes (từ nhỏ đến lớn)

D : theo ngày tháng và giờ (từ trước đến nay)

G : theo nhóm thư mục trước

**\*Tạo thư mục mới (Make Directory - MD)**

Tạo một thư mục mới trong ổ đĩa hoặc thư mục hiện hành.

Cú pháp: MD [drive :][path]<tên thư mục mới>

**\*Đổi thư mục (Change Directory - CD)**

Cú pháp: CD [drive :] [path]

Ví dụ C:\>CD PASCAL sẽ có C:\PASCAL>

Ghi chú:

- Từ thư mục con, muốn trở về thư mục cha, ta gõ: CD..

- Nếu muốn về thẳng thư mục gốc, ta gõ: CD\

- Để hiển thị đường dẫn hiện hành, ta gõ: CD

**\*Xem nội dung tập tin trên màn hình (TYPE)**

Cú pháp: TYPE [drive:][path]<file name>

Ghi chú:

- Lệnh TYPE dùng hiển thị một tập tin văn bản chứa mã ASCII mới đọc được.

- Các file chứa mã nhị phân của chương trình như các file \*.EXE, \*.COM,\*.BIN,... thì khi gõ lệnh TYPE sẽ không đọc bình thường được.

**\*Xóa thư mục (Remove Directory - RD)**

Xoá bỏ một thư mục con rỗng (không chứa các tập tin và thư mục con).

Cú pháp: RD [drive :] <path>

**\*Sao chép tập tin (COPY)**

**Cú pháp 1:** sao chép tập tin sang một vị trí khác.

COPY [drive1 :][path1]<filename1> [drive2:][path2][<filename2>]

Ghi chú:

- Muốn copy một nhóm tập tin, ta có thể dùng các ký tự đại diện của tập tin là dấu \* hoặc ? trong <filename>

- Nếu không viết <filename 2> thì máy sẽ hiểu là sao chép mà không đổi tên tập tin. Khi đó ổ đĩa [drive1:] phải khác [drive2:] hoặc đường dẫn <path1> khác <path2>

- Khi không chỉ rõ Ổ đĩa và đường dẫn thì máy sẽ hiểu đang thực hiện trên ổ đĩa và thư mục hiện hành.

**Cú pháp 2:** cho phép nối các tập tin có sẵn thành một tập tin mới chung.

COPY <file 1> + <file 2> [+ ... + <file n>] [<new\_file>]

Ghi chú:

- Nếu không đặt tên tập tin mới thì tất cả các tập tin sẽ ghép chung vào <file 1>
- Nếu <new\_file > đã có thì nội dung cũ sẽ được thay bằng nội dung mới.
- Tên <new\_file > không được trùng với tên các tập tin cần ghép.

**Cú pháp 3:** sao chép tập tin ra máy in.

COPY [path]<filename> PRN

**Cú pháp 4:** dùng lệnh COPY để tạo ra một tập tin văn bản đơn giản:

COPY CON [path]<filename>

Ghi chú:

- Sau khi gõ Enter, ta có thể đánh vào một vài đoạn văn trên bàn phím tùy ý. Muốn kết thúc, đánh tổ hợp phím Ctrl+Z hay F6 và Enter để lưu trữ.
- Khi gõ Enter để xuống dòng, ta không di chuyển con trỏ trở lên dòng trên được.
- Lỗi văn bản sai không thể sửa nội dung trực tiếp khi đã lưu, ngoại trừ phải dùng các trình soạn thảo văn bản khác.

**\* Xoá tập tin (Delete - DEL)**

Cú pháp: DEL [drive:][path]<file name>[/P]

Ghi chú:

- Có thể xoá một loạt <tên file> nếu dùng các ký tự \* và ?
- [/P] tạo nhắc để xác định từng tập tin muốn xoá
- Xoá tất cả các tập tin thì dùng lệnh DEL \*.\* khi đó máy sẽ hỏi lại :  
All files in directory will be deleted ! (Tất cả các tập tin sẽ bị xoá !)  
Are you sure (y/n) ? \_ (Bạn có chắc không (y/n) ?)  
Gõ y (yes) nếu muốn xoá tất cả và n (no) khi không muốn dùng lệnh xoá tất cả.
- Lệnh DEL không xoá các tập tin ẩn (hidden) và tập tin chỉ đọc (read only).
- Muốn xoá các tập tin này ta phải dùng lệnh ATTRIB ngoại trừ để thay đổi thuộc tính của nó.
- Nếu DEL một thư mục thì xoá hết các tập tin trong thư mục đó.

**\*Đổi tên tập tin (Rename - REN)**

Cú pháp: REN [drive:][path]<old\_file> <new\_file>

Ghi chú:

- <old\_file> : tên tập tin cũ cần đổi <new\_file> : tên tập tin mới
- Trường hợp tập tin mới đã có tên rồi hoặc không có tập tin cũ, máy sẽ báo:

Duplicate file name or file not found

**\* Xoá màn hình (ClearScreen - CLS)**

Cú pháp: CLS

**\* Dấu đợi lệnh (PROMPT)**

Cú pháp: PROMPT [\$text]

Ghi chú:

- Lệnh này định dạng lại dấu đợi lệnh của MS-DOS theo ý riêng của người sử dụng.

- \$text là chuỗi các ký tự liên tiếp nhau, trước mỗi ký tự có ký hiệu \$

\$P : ổ đĩa và thư mục hiện hành

\$G : dấu >

\$L : dấu <

\$Q : dấu =

\$T : giờ hiện hành

\$D : ngày hiện hành

\$N : ổ đĩa hiện hành

\$V : phiên bản của MS-DOS

\$\_ : xuống hàng

**\*Xem và chỉnh thời gian (TIME)**

Cú pháp: TIME [hh:mm:ss]

Ghi chú:

- Sau khi Enter, trên màn hình sẽ xuất hiện giờ phút giây hiện tại của hệ thống.

Ví dụ :            TIME

Current Time is 9 : 30 : 15

Enter new time : \_

- Nếu không cần thay đổi thời gian, ta nhấn Enter để xác nhận.

- Muốn chỉnh thời gian, ta nhập giờ : phút : giây phù hợp tại vị trí con trỏ.

**\*Xem và chỉnh ngày tháng (DATE)**

Cú pháp: DATE [mm-dd-yy]

Ghi chú:

- Sau khi gõ Enter, trên màn hình sẽ hiện ngày tháng năm hiện tại của hệ thống.

- Kiểu ngày tháng năm do người sử dụng thiết lập.

Ví dụ :    DATE

Current date is Sun 09-15-96



Enter new date (mm-dd-yy) : \_

- Nếu không muốn thay đổi ngày tháng của hệ thống, ta gõ Enter để xác nhận.
- Nếu muốn thay đổi ngày tháng, ta gõ theo thứ tự số : tháng-ngày-năm

Chú ý: khi gõ sai, máy sẽ báo : Invalid date (ngày tháng không hợp lệ). Khi đó ngày tháng của hệ thống vẫn giữ nguyên và ta phải dùng lệnh DATE để nhập lại ngày tháng cho hệ thống.

Các máy tính Pentium hiện nay đều thể hiện phần năm là 4 con số, ví dụ :

Current date is Fri 08-04-2000

### 2.2.3.2. Lệnh ngoại trú ( external command)

Như ta đã biết lệnh ngoại trú cũng là những lệnh chứa các chức năng nào đó của hệ điều hành nhưng ít được sử dụng hơn lệnh nội trú nên được để trên đĩa hay thư mục riêng để đỡ tốn bộ nhớ. Các lệnh ngoại trú phải được nạp từ đĩa vào trong bộ nhớ mới chạy được. Khi thực hiện xong câu lệnh, vùng bộ nhớ có chứa câu lệnh ngoại trú đó sẽ bị thu hồi. Các tập lệnh ngoại trú có phần mở rộng là EXE hay COM hoặc BAT. Khi gọi lệnh mà máy không tìm thấy trên màn hình sẽ xuất hiện câu báo lỗi Bad command or file name (Sai lệnh hoặc không có tên tập tin). Các lệnh ngoại trú như:

- Các lệnh liên quan đến ổ đĩa: FORMAT, LABEL, CHKDSK, DISKCOPY, ...
- Các lệnh liên quan đến tổ chức hệ thống: SYS, TREE, DELTREE, ...
- Các lệnh liên quan đến tập tin: UNDELETE, ATTRIB, FIND, ...
- Các lệnh khác: PRINT, GRAPHICS, SORT, ...

#### Một số lệnh ngoại trú thường dùng:

##### \* Tạo khuôn đĩa (FORMAT)

Lệnh FORMAT có tác dụng khởi tạo khuôn cho một đĩa mới hay tạo lại dạng khuôn cho đĩa cũ. Khi thực hiện lệnh FORMAT xong thì toàn bộ dữ liệu trong đĩa cũ hoàn toàn bị xoá sạch.

Cú pháp: FORMAT [drive :][/S][/U][/Q][/V[:label]]

#### Ghi chú:

- Tham số /S dùng để tạo ra đĩa hệ thống. DOS sẽ tạo dạng và chép vào đĩa 3 file hệ thống là COMMAND.COM (file hiện), IO.SYS và MSDOS.SYS (file ẩn).
- Tham số /U thực hiện format không điều kiện (Unconditional)
- Tham số /Q thực hiện format nhanh (Quick)
- Tham số /V[:label ] để chỉ định tên nhãn (label) của đĩa

**Ví dụ 4.7:** muốn format bình thường ổ đĩa A:

C:\>format A:

Insert new diskette for drive A : (Đưa đĩa mới vào ổ A:

and press ENTER when ready ... và ấn phím Enter khi sẵn sàng ..)

Sau đó là quá trình format. Ta có thể quan sát quá trình này qua số % đã thực hiện. Khi format xong, máy sẽ hiển thị dòng yêu cầu cho tên đĩa và bạn có thể cho tên đĩa với tối đa 11 ký tự và được quyền sử dụng ký tự trống, nếu không cần đặt tên, ta có thể nhấn Enter cho qua luôn.

### \*Thay đổi thuộc tính của file (ATTRIB)

Mỗi tập tin đều có 4 thuộc tính (attribute) : R, S, H, A.

Read Only(R) : Thuộc tính chỉ đọc, không thể thay đổi nội dung

System (S) : Thuộc tính hệ thống, hệ điều hành DOS sẽ chú ý đặc biệt.

Hidden (H) : Thuộc tính ẩn, khi dùng lệnh DIR sẽ không thấy.

Archive(A) : Thuộc tính lưu trữ, thường dùng với các lệnh như BACKUP, XCOPY ... để tạo ra các file dự phòng.

Cú pháp: ATTRIB [-R|+R] [-S|+S] [-H|+H] [-A|+A] [drive:][path]<file name>

Với dấu | là hoặc, hoặc chọn đặt (+) thuộc tính hoặc xóa (-) thuộc tính của tập tin.

### \*Liệt kê cây thư mục TREE

Cú pháp: TREE [drive :] [path] [/F] [/A]

Ghi chú: /F : Thể hiện tên các tập tin trong từng thư mục

/A: Dùng ASCII thay cho các ký tự mở rộng

### \*Xóa cây thư mục DELTREE

Lệnh này xóa thư mục được chỉ định và tất cả các thư mục con, tập tin của nó.

Cú pháp: DELTREE [/y] [drive :] [path] <directory name>

Ghi chú: directory name là tên thư mục đại diện cho một cây (nhánh) thư mục cần xóa. Nếu có tùy chọn /y có nghĩa là người sử dụng đã xác nhận việc xóa cây thư mục này là chắc chắn.

### \*Phục hồi file đã bị xóa (UNDELETE)

Ta có thể khôi phục lại các file đã lỡ bị xóa do dùng lệnh DEL, miễn là vùng chứa file này trong đĩa chưa bị các file khác chép chồng lên và nhất thiết ta phải nhớ ký tự đầu tiên của tên file.

Cú pháp: UNDELETE [drive:][path]<file name>

### \*Di chuyển file MOVE

Đề dời file từ nơi này sang nơi khác (thay vì dùng COPY và DEL)

**Cú pháp 1:** Di chuyển tập tin

MOVE [drive:][path]<file name> <destination>

Với destination là nơi di chuyển các files đến.

**Cú pháp 2:** Đổi tên thư mục

MOVE [drive:][path]<old-directory> <new-directory>

### \*In văn bản ra máy in (PRINT)

Có thể in trực tiếp các file văn bản chứa mã ASCII ra máy in.

Cú pháp: PRINT </D: device>[drive:][path]<filename>

Nếu muốn in trực tiếp các thông tin hiển thị trên màn hình ta có thể dùng tổ hợp phím Shift + PrintScreen sau khi bật máy in lên.

Tùy chọn Device dùng để khai báo cổng máy in COM1, COM2, ... hoặc LPT1, LPT2, ...

## 2.3. HỆ ĐIỀU HÀNH WINDOWS

### 2.3.1. Giới thiệu về hệ điều hành WINDOWS

Như ta thấy MS-DOS là HĐH đơn nhiệm nghĩa là tại mỗi thời điểm chỉ có một phần mềm ứng dụng chạy được, màn hình giao diện người-máy là màn hình văn bản, tức là chỉ hiển thị được các ký tự trong bảng ASCII mở rộng. Khả năng về đồ họa và âm thanh rất đơn điệu. Từ khi ra đời với phiên bản MS-DOS 1.0 đến nay với MS-DOS 6.22 đã trải qua nhiều lần cải tiến nâng cấp nhưng vẫn chưa thay đổi được một số nhược điểm nói trên. Năm 1984 hãng Apple Computer cho ra đời máy tính Macintosh có tính năng giao diện người-máy bằng đồ họa, trang bị nhiều chức năng cửa sổ, sử dụng các trình đơn kéo xuống và con chuột, có khả năng âm thanh bằng số, có các kiểu chữ đa dạng, các dụng cụ văn phòng (đồng hồ, máy tính, lịch, sổ ghi chép...). Chính hàng loạt những đổi mới của Macintosh đã truyền cảm hứng cho Microsoft xây dựng Windows sau này.

Năm 1982, IBM đã phối hợp với Microsoft xây dựng hệ điều hành đa nhiệm OS/2 (Operating System/2) cho các máy tương thích IBM PC, đã phá vỡ hàng rào 640K RAM, cung cấp sự an toàn cho các chương trình chạy đồng thời, và cho phép trao đổi động dữ liệu giữa các chương trình ứng dụng. IBM và Microsoft tin chắc rằng OS/2 sẽ thay thế MS-DOS và sẽ là hệ điều hành của tương lai. Người dùng được khuyến khích dùng OS/2 để thay thế MS-DOS. Tuy vậy phần lớn người dùng sử dụng OS/2 một cách miễn cưỡng vì họ không muốn thay hẳn những gì họ đã quen thuộc. Rất ít hãng sản xuất phần mềm chịu xây dựng các chương trình ứng dụng OS/2 và các phiên bản đầu tiên của OS/2 đã chạy các chương trình MS-DOS rất kém. Microsoft đã xây dựng Windows dựa trên những ý tưởng của Macintosh. Microsoft Windows lúc đó cũng chỉ được xem như là một giao diện người - máy bằng đồ họa cho MS-DOS. Người sử dụng vẫn có thể lựa chọn là họ có thể chạy các chương trình ứng dụng trong môi trường DOS hoặc khởi động Windows. Lúc đầu Microsoft chỉ có ý định xem Windows là biện pháp lấp chỗ trống cho đến khi nào thị trường chấp nhận OS/2. Windows 3.0 đã gây kinh ngạc cho nền công nghiệp do sự kết hợp hoàn toàn với quá trình xử lý trong chế độ được bảo vệ; đột nhiên, lý do căn bản của việc nâng cấp lên OS/2 bị xóa bỏ gần như toàn bộ, và Windows 3.0 đã bán ra được hàng triệu bản. Microsoft đã đi đến xác nhận rằng tương lai thuộc về Windows. Không đáng ngạc nhiên lắm khi những sự kiện này có vẻ như đã dội một gáo nước lạnh lên mối quan hệ vốn thân mật và gần gũi trước đây giữa IBM và Microsoft. IBM đã tuyên bố nâng cấp một cách triệt để OS/2. OS/2 phiên bản 2.0 ra đời năm 92 đã khắc phục được các nhược điểm trước đây và chạy các chương trình MS-DOS và Windows một cách hoàn hảo và đã được nhiều người hoan nghênh như là một thành tựu kỹ thuật lớn. Tuy vậy OS/2 có làm ngưng được đà phát triển đáng kinh ngạc của Windows hay không còn phải chờ xem.

Windows từ khi ra đời đến nay đã trải qua nhiều phiên bản:

Windows 3.0, 3.1, 3.11 là các hệ điều hành 16 bit (tức dữ liệu được truyền thành các chùm 16 bit) và vẫn chạy trên nền DOS. Các hệ này dùng FAT 16 bit.

Windows 95 là hệ điều hành độc lập 32 bit, nhưng vẫn dùng FAT 16 bit. So với các HĐH trước Windows 95 có nhiều ưu điểm: Giao diện thân thiện và trực quan, hỗ trợ nhiều phần cứng, Plug and Play (cắm là chạy), tương thích với DOS 16 và Win 16.

Sau Windows 95 là Windows NT (for workstation and for server): Độ bảo mật cao hơn nhưng hỗ trợ ít phần cứng hơn và chạy chậm hơn.

Windows 98 thực chất là Windows 95 cải tiến, cũng là HĐH 32 bit nhưng dùng FAT 32 và có chức năng chuyển đổi FAT 16 -> FAT 32. Windows 98 có nhiều ưu điểm:

- Hỗ trợ nhiều các màn hình (tối đa là 9), hỗ trợ đa phương tiện, tích hợp internet và các dịch vụ trực tuyến, chạy nhanh hơn Windows 95, FAT 32 giúp lưu trữ tốn ít dung lượng đĩa hơn.

- Đồ họa 3 chiều nhanh và tốt hơn nhờ vậy các trò chơi trở nên hấp dẫn hơn.

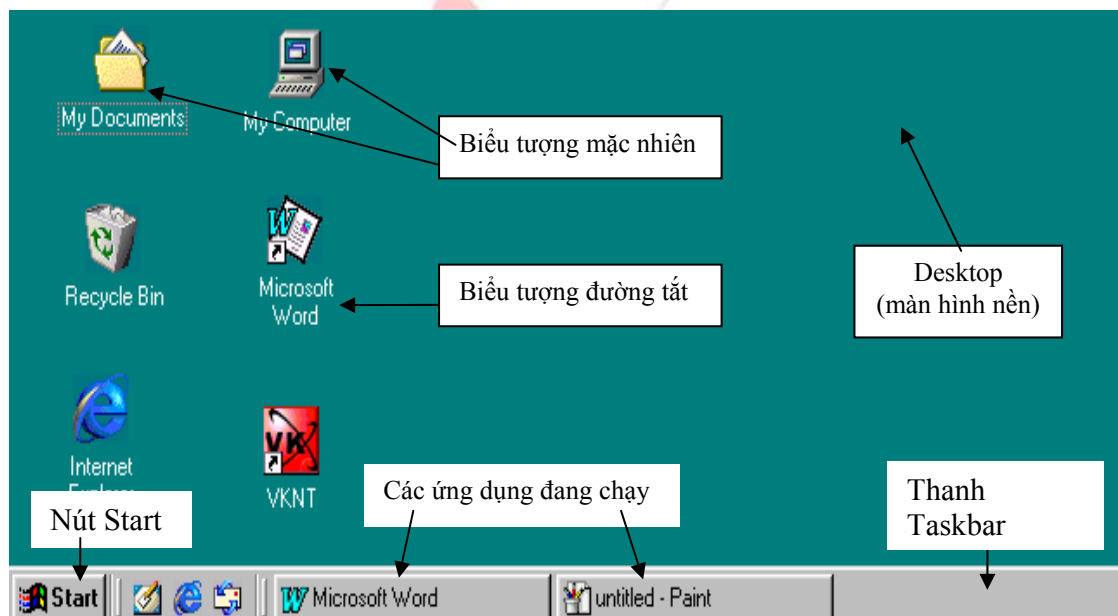
- Nâng cao khả năng giải trí của máy tính: giúp có được phim hoàn chỉnh với âm thanh nổi, chất lượng sân khấu trên máy PC trang bị DVD. Thiết bị lưu trữ mới này có thể chứa tới 17GB dữ liệu với hỗ trợ đa ngôn ngữ trên một đĩa.

Windows 2000 kết hợp các ưu điểm của Win 95, 98 và Windows NT.

Đặc điểm của Windows như tên gọi của nó là chạy tất cả các ứng dụng trong từng cửa sổ riêng biệt mà kích cỡ có thể được sửa đổi bởi người sử dụng. Trong các phần sau đây chúng tôi chỉ xin giới thiệu Windows 2000, là phiên bản đã được kiểm nghiệm và dùng rộng rãi ở Việt nam.

## 2.3.2. Các khái niệm trong Windows:

### 2.3.2.1. Màn hình giao diện:



gồm có **màn hình nền** (Desktop), trên màn hình nền là các biểu tượng. Số lượng các biểu tượng tùy theo các chương trình được cài đặt, ta có thể thêm và xóa các biểu tượng. Có hai loại biểu tượng: **biểu tượng mặc nhiên** (sẵn có ngay sau khi cài đặt Windows 2000), **biểu tượng đường tắt** (có dấu mũi tên ở góc dưới bên trái hình vẽ biểu tượng, ví dụ như biểu tượng *Norton Commander* ở màn hình trên).

Một số biểu tượng quan trọng trên màn hình nền:

- **My Computer** (máy tính của tôi): là biểu tượng cho phép xem các tài nguyên có trong máy tính đang dùng, quản lý các tệp và thư mục.
- **My Network Places**: Là biểu tượng cho phép xem tài nguyên đang có trên mạng nếu máy tính đang hay có thể gắn vào một mạng.
- **Recycle Bin** (thùng rác): Là biểu tượng cho phép lưu trữ tạm thời các tệp bị xoá, có thể dùng nó để phục hồi các tệp bị xoá nhầm.
- **Internet Explorer**: Là biểu tượng cho phép kết nối với Internet.
- **Microsoft Outlook**: Là biểu tượng cho phép gửi và nhận thư điện tử.

Dưới đáy màn hình là thanh Taskbar. Phía trái thanh Taskbar là nút Start, nơi khởi đầu của hầu hết các công việc trong Windows 2000, phía phải nút Start là các nút ứng với các ứng dụng đang chạy.

Thiết lập màn hình nền: nhấp nút phải chuột trên Desktop, hiện menu tắt, chọn Properties, hiện hộp thoại, chọn lớp Background, duyệt xem các ảnh nền trong khung Select a background picture, chọn một ảnh, OK.

Thiết lập màn hình chờ (màn hình hiện khi không dùng máy tính): nhấp nút phải chuột trên Desktop hiện menu tắt, chọn Properties, hiện hộp thoại, chọn lớp Screen Saver, duyệt xem qua các màn hình chờ trong hộp điều khiển kéo xuống Screen Saver, chọn một màn hình chờ, trong mục Wait đặt là số phút chờ, chọn OK

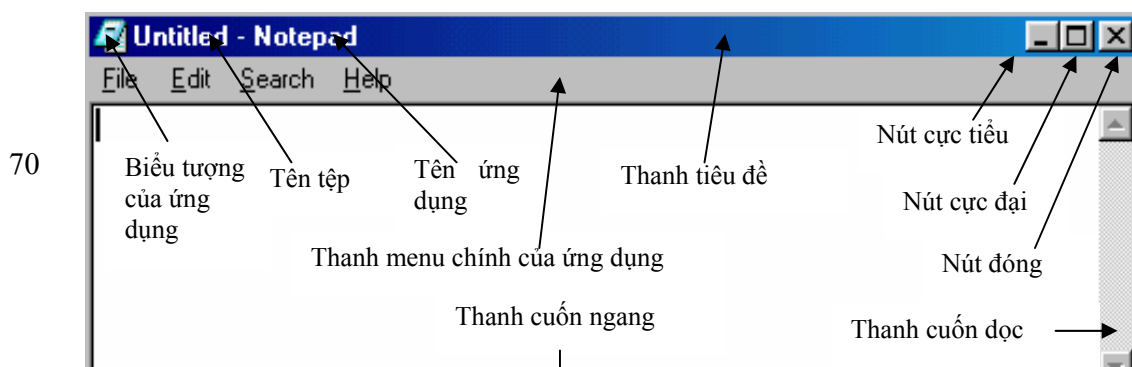
### 2.3.2.2. Chuột và cách sử dụng

Công cụ làm việc chính trong môi trường Windows là chuột và bàn phím. Trên màn hình luôn luôn nhìn thấy một mũi tên hay con trỏ nhỏ, đó chính là con trỏ chuột; sử dụng chuột để điều khiển con trỏ này; di chuyển chuột sẽ làm cho con trỏ tương ứng trên màn hình di chuyển theo. Các thao tác cơ bản với chuột:

- Nháy chuột (**Click**): bấm nút trái chuột (hoặc nút phải chuột nếu có) một lần rồi thả ra.
- Nháy đúp nút chuột (**Double Click**): bấm nhanh hai lần liên tiếp nút trái chuột.
- Rê chuột (**Drag**): bấm và giữ nguyên tay nút chuột rồi di chuyển chuột, nhả tay khi vừa ý.

### 2.3.2.3. Cửa sổ, điều khiển cửa sổ

Cửa sổ là màn hình của một ứng dụng khi được khởi động. Trên đỉnh cửa sổ là thanh tiêu đề, phía trái thanh tiêu đề là Biểu tượng của ứng dụng, Tên tệp và tên ứng dụng đang chạy. Phía phải thanh tiêu đề có 3 nút để điều khiển cửa sổ: Nút cực tiểu để cực tiểu hoá cửa sổ ứng dụng đưa về thành một nút trên thanh Taskbar. Nút cực đại để phóng to cửa sổ ra toàn màn hình (khi nút có một hình chữ nhật) hoặc thu nhỏ cửa sổ về kích thước lúc trước khi phóng (khi trên nút có 2 hình chữ nhật, nút cực đại đã trở thành nút Restore). Nút Close để đóng cửa sổ ứng dụng. Dưới thanh tiêu đề là Thanh menu chính của ứng dụng.



Có thể dùng chuột để thay đổi kích thước của cửa sổ. Di chuyển chuột tới cạnh phải, cạnh đáy hay góc dưới bên phải của cửa sổ để con trỏ chuột chuyển thành mũi tên hai đầu, kéo các cạnh để phóng to hay thu nhỏ cửa sổ theo ý muốn. Di chuyển cửa sổ trên màn hình: kéo thanh tiêu đề tới vị trí mới. Nếu cửa sổ không đủ rộng để hiện tất cả các thông tin, xuất hiện các thanh cuộn ở cạnh bên phải và cạnh đáy cửa sổ, ta có thể kéo nút cuộn hay nhấp các nút mũi tên ở trên thanh cuộn để hiện thông tin cần xem.

Toàn bộ các thao tác trên đối với cửa sổ có thể thao tác qua Menu điều khiển nhỏ ở góc trên bên trái cửa sổ. Để kích hoạt menu này ta nhấp vào Biểu tượng của ứng dụng nằm ở phía kịch trái của Thanh tiêu đề. Menu gồm các mục: Restore (trở về kích thước ban đầu của cửa sổ), Move (di chuyển cửa sổ), Size (thay đổi kích thước cửa sổ), Minimize (cực tiểu hóa cửa sổ thành một biểu tượng trên thanh Taskbar), Maximize (phóng to cửa sổ ra toàn màn hình), Close (đóng cửa sổ).

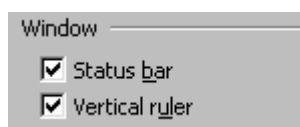
Một số chương trình dùng các cửa sổ được chia dọc hay chia ngang (ví dụ cửa sổ Windows Explorer), các bộ phận của cửa sổ gọi là các khung. Ta có thể thay đổi kích thước tương đối của các khung bằng cách kéo đường phân chia khung.

#### 2.3.2.4. Hộp thoại trong Windows

Hộp thoại trong Windows giúp cho người dùng có thể thực hiện các lựa chọn và ra quyết định hành động, hộp thoại chiếm một vùng màn hình cũng tương tự như cửa sổ. Mục 1.7 ở dưới cho hộp thoại **Shut Down Windows** xuất hiện khi tắt máy tính. Dòng trên cùng của hộp thoại là *Thanh tiêu đề* mà trên nó ghi tên hộp thoại.

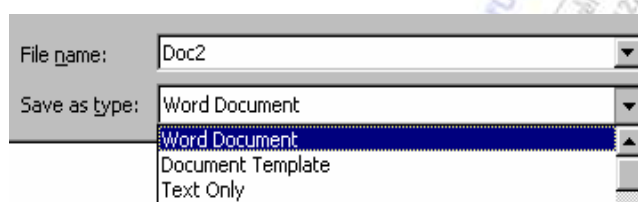
Trên hộp thoại thường có các nút sau:

- **Nút đóng hộp thoại**  ở đầu bên phải thanh tiêu đề dùng để đóng hộp thoại.
- **Nút ấn** (Push button) là những nút hình chữ nhật trên đó có đề chữ (xem hộp thoại Shut Down Windows trong mục 1.7). Các nút thường dùng: OK (khẳng định các lựa chọn), Cancel (hủy bỏ các lựa chọn và thoát khỏi hộp thoại), Help (xem hướng dẫn sử dụng cho hộp thoại), các nút có chữ với 3 dấu chấm (sinh ra hộp thoại mới).
- **Nút kiểm tra** (Check Box) là các ô nhỏ bên cạnh có chữ, khi nhấp chuột để lựa chọn thì ô được đánh dấu bởi ký tự giống chữ V. Có thể lựa chọn nhiều Nút kiểm tra trong một hộp thoại.



▪ **Nút đài** (Radio button), còn gọi là nút lựa chọn đơn, là những ô hình tròn bên cạnh có chữ, khi nhấp chuột vào ô này để lựa chọn thì ô được đánh dấu bởi một chấm to màu đen. Khi xuất hiện nhiều nút đài thì chỉ có thể được chọn một (các khả năng loại trừ nhau). Ví dụ trong hộp thoại ở mục 3.1 Chương 2 có 7 nút đài.

▪ **Nút điều khiển kéo xuống** (Control Box) là những nút mà bên phải có nút mũi tên, bên trái là xâu ký tự thuyết minh về nút. Khi nhấp chuột vào Mũi tên bên phải nút sẽ xuất hiện một menu kéo xuống cho phép ta lựa chọn một mục, hoặc ta cũng có thể nhấp chuột vào khung văn bản của nút và nhập trực tiếp văn bản như đối với Hộp văn bản.



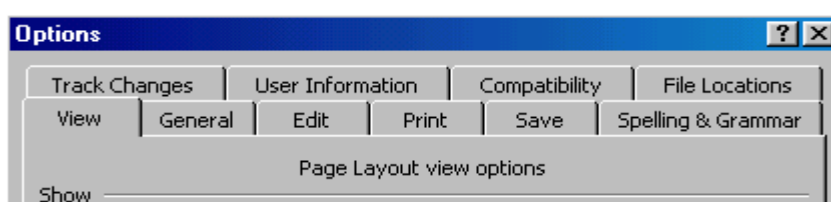
▪ **Nút tăng giảm** là những nút mà ở phía phải có 2 mũi tên lên và xuống, giữa nút chứa dữ liệu số. Nhấp vào mũi tên lên hay xuống sẽ làm tăng hay giảm giá trị của số trong ô. Ta cũng có thể nhấp chuột vào giữa ô và nhập trực tiếp giá trị số mới.



▪ **Hộp văn bản** (Text Box): khi nhấp chuột vào hộp ta có thể nhập văn bản. Ví dụ hộp văn bản dùng để nhập mật khẩu cho tài liệu:



▪ **Các lớp**. Một số hộp thoại được tổ chức thành nhiều lớp giống như các tấm bìa xếp chồng lên nhau, tên các lớp nằm ở phía trên đỉnh hộp thoại. Mỗi lớp lại tương ứng với các mục lựa chọn riêng, có thể coi mỗi lớp như là một hộp thoại con. Ví dụ hộp thoại Options bao gồm 10 lớp.



▪ **Nút trợ giúp** nằm ở phía trên bên phải cửa sổ, khi nháy nút này dấu chấm hỏi sẽ dính vào con trỏ chuột, di con trỏ chuột tới bất kỳ mục nào trong hộp thoại và nháy chuột thì ta sẽ được hướng dẫn trực tiếp của chính mục này.

#### 2.3.2.5. Các loại menu

Khi một ứng dụng được mở phía dưới Thanh tiêu đề là **Thanh menu chính** của ứng dụng. Nếu nháy vào một mục trên Thanh menu chính sẽ xuất hiện một Menu dọc gồm các mục Cut, Copy, Paste, Select All. Một mục bị mờ trên một menu là lệnh không được dùng trong ngữ cảnh hiện hành. Khi nháy vào mục có dấu mũi tên ở phía phải sẽ xuất hiện một Menu mới.

Khi nháy **nút phải chuột** vào màn hình nền, hoặc vào nền của Thanh Taskbar, hoặc vào tên các tệp đã chọn trong Windows Explorer, hoặc vào đoạn văn bản đã đánh dấu khối trong Microsoft Word, trên màn hình sẽ xuất hiện một menu nhỏ (gọi là **Menu tắt**) chứa các lệnh mà ta cần dùng trong tình huống này.

#### 2.3.2.6. Tệp và thư mục

**Tệp** (File) là tập các mã 0 và 1 được ghi trên đĩa cứng, đĩa mềm hay đĩa CD-ROM để mã hóa một văn bản, một chương trình, một bức ảnh hay một đoạn âm thanh. Mỗi tệp cần có một tên riêng để phân biệt, trong Windows 2000 tên tệp và tên thư mục có thể dài tới 250 ký tự và cho phép chứa cả dấu cách. Trong môi trường DOS trước đây tên tệp gồm 3 phần và không chứa dấu cách: tên chính không quá 8 ký tự, dấu chấm, phần mở rộng không quá 3 ký tự.

**Thư mục** (Folder): Thư mục trên đĩa là một vùng dùng để chứa các tệp và thư mục con khác. Khi khởi tạo một đĩa, một thư mục không có tên được tạo tự động gọi là thư mục gốc và được chỉ định bằng dấu gạch chéo ngược \, các thư mục con khác phải có tên. Trong cùng một thư mục không được có hai thư mục hay hai tệp trùng tên.

#### 2.3.2.7. Menu Start và thanh Taskbar

Trong Windows 2000, Windows 98 và Windows 95 menu Start đóng vai trò như Program Manager của Windows 3.1, Taskbar (Thanh công việc) đóng vai trò như cửa sổ Task List của Windows 3.1. Menu Start dùng để quản lý các ứng dụng, các chương trình đã cài đặt; Taskbar dùng để quản lý các chương trình hiện đang chạy và chuyển đổi giữa chúng.

### 2.3.3. Cơ bản về cách cài đặt và sử dụng WINDOWS

#### 2.3.3.1. Cơ bản về cài đặt:

Có thể cài đặt Windows 2000 theo hai cách: Nâng cấp từ Windows 95 hay Windows 98 lên Windows 2000 (chương trình Setup tự động chuyển các thiết lập hệ thống trước đây và các chương trình ứng dụng cũ đã cài đặt vào Windows 2000), cài Windows 2000 hoàn toàn mới (máy



tính trước đây và hệ điều hành không phải là Windows, đĩa cứng hoàn toàn mới, các chương trình ứng dụng cũng phải cài lại từ đầu).

Chương trình Setup quản lý toàn bộ tiến trình cài đặt: yêu cầu thông tin của người dùng, phát hiện cấu hình phần cứng của máy tính, sao chép các tệp và khởi động lại máy tính khi cần. Các thông báo trên màn hình giải thích Setup đang làm gì và giải thích các tùy chọn ở mỗi bước trong quá trình cài đặt.

Cài đặt phiên bản trước của Windows bằng Windows 2000 Professional CD:

- Sau khi Windows khởi động, cho đĩa gốc Windows 2000 vào ổ CD-ROM.

- Nếu thông báo xuất hiện yêu cầu ta khẳng định có nâng cấp lên Windows 2000 hay không, hãy nháy vào Yes. Nếu không có thông báo nào như thế, dùng lệnh Start / Run, nhập vào lệnh D:\I386\WINNT32 (D là tên ổ đĩa CD-ROM).

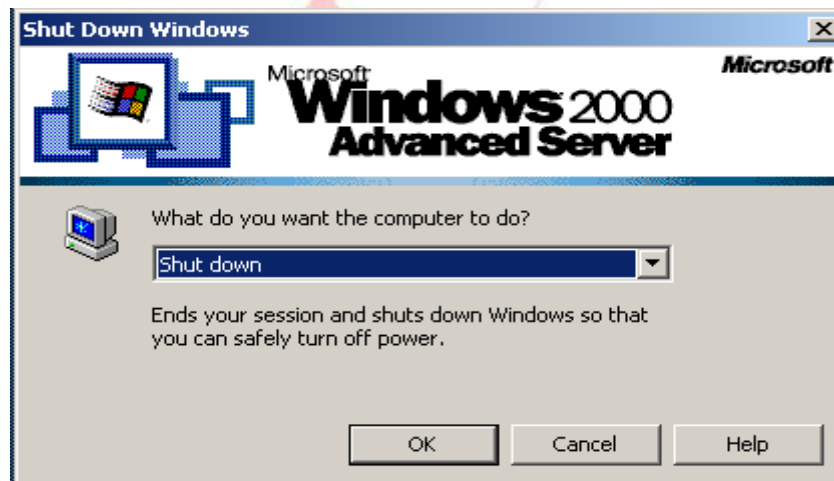
- Khi xuất hiện cửa sổ Welcome to the Windows 2000 Setup Wizard, hãy lựa tùy chọn thích hợp: Upgrade (nâng cấp) hay Install a new copy (cài đặt hoàn toàn mới). Nháy vào nút Next và theo các hướng dẫn trên màn hình.

### 2.3.3.2. Sử dụng WINDOWS:

**a) Khởi động:** Khi khởi động sẽ xuất hiện các hộp thoại "Welcome To Windows" và "Log On To Windows" yêu cầu ta nhập vào tên người dùng và mật khẩu truy nhập .

#### **b) Thoát khỏi Windows và tắt máy**

Nháy nút Start, xuất hiện menu dọc, chọn mục Shut Down, xuất hiện hộp thoại:



Nháy vào nút mũi tên của hộp điều khiển kéo xuống "What do you want the computer to do?" để lựa chọn một trong các cách thoát:

- **Shut down:** tắt máy tính (chờ cho đến khi xuất hiện câu "It is now safe to turn off your computer" xuất hiện thì tắt công tắc nguồn).

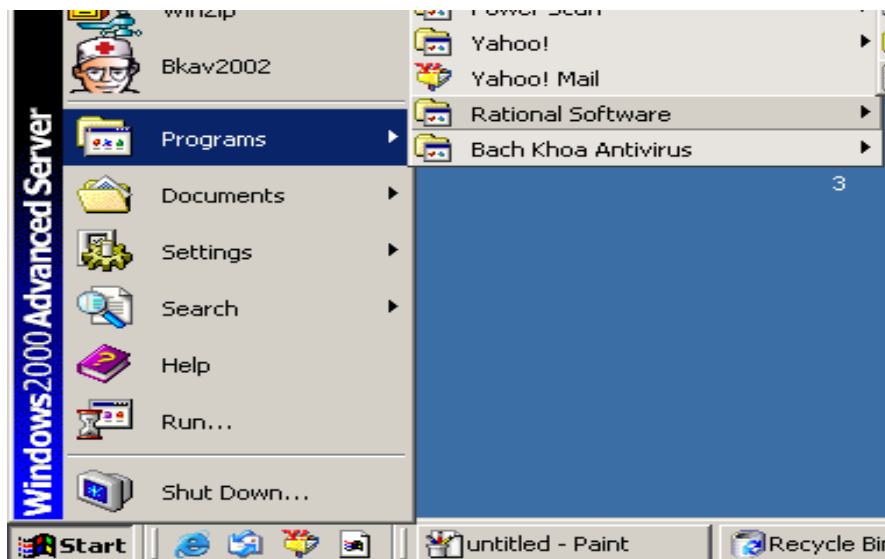
- **Restart:** khởi động lại máy tính, nạp lại hệ điều hành.

Nháy nút OK để xác nhận tùy chọn, muốn bỏ qua tùy chọn để trở lại làm việc với Windows 2000 thì nháy nút Cancel.

### c).Sử dụng MENU START và thanh TASKBAR

#### - Menu Start

Khi nháy nút Start xuất hiện menu Start:



Trong menu này có các mục:

- **Programs**: khi trỏ chuột vào mục này sẽ xuất hiện menu dọc cấp hai Programs chứa tất cả các chương trình có thể chạy.
- **Documents**: mở các tài liệu đã được mở gần đây nhất.
- **Settings**: sửa đổi màn hình nền hoặc các thiết lập hệ thống.
- **Search**: tìm tệp hoặc thư mục.
- **Help**: khi chọn mục này cửa sổ trợ giúp xuất hiện. Cửa sổ trợ giúp có 3 nút: Content (hướng dẫn phân theo chủ đề), Index (các chủ đề hướng dẫn liệt kê theo thứ tự từ vựng), Search (tìm mọi chủ đề có chứa một từ hay câu, sau đó lựa chọn chủ đề cần xem hướng dẫn).
- **Run**: mở (chạy) các chương trình
- **Shut Down**: tắt hoặc khởi động lại máy.

Trong các hệ thống menu dọc xuất phát từ nút Start, nếu ta trỏ chuột (không cần nháy) vào mục có dấu mũi tên thì sẽ xuất hiện một menu dọc mới.

Muốn chuyển menu Start sang dạng với các biểu tượng nhỏ hơn ta dùng lệnh: Start / Settings / Taskbar & Start Menu, xuất hiện hộp thoại, chọn mục Show Small Icon in Start Menu.

**Sử dụng các menu cá nhân hóa.** Với các menu cá nhân hóa, Windows chỉ hiển thị các mục menu mà ta dùng thường xuyên nhất, dấu đi các mục menu khác. Để hiện các mục bị dấu cần nháy vào mũi tên đôi ở dưới menu con, hoặc ta chờ vài giây Windows cũng tự động mở rộng

menu. Để kích hoạt hoặc vô hiệu các menu cá nhân hóa dùng lệnh: Start / Settings / Taskbar & Start Menu, xuất hiện hộp thoại, chọn hoặc xóa bỏ hộp kiểm Use Personalized Menus.

### - Khởi động một chương trình ứng dụng

**Dùng nút Start** có thể khởi động bất kỳ một chương trình ứng dụng nào, như Word, Excel, Paint, Solitaire.... Đầu tiên nháy nút Start, trỏ chuột vào mục Programs để xuất hiện menu Programs..., vào menu dọc chứa chương trình cần chạy, nháy vào mục chương trình muốn khởi động (các mục không có dấu mũi tên ở sau là các chương trình có thể chạy được). Khi một chương trình khởi động thì trên thanh Taskbar xuất hiện một nút mới chứa biểu tượng và tên chương trình vừa khởi động.

Khi đang làm việc với một chương trình ứng dụng, muốn chạy một chương trình ứng dụng khác ta lại nháy vào nút Start và làm như trên. Như vậy Windows 2000 cho phép chạy nhiều ứng dụng cùng một lúc, tên các ứng dụng hiện trên thanh Taskbar. Tất cả các cửa sổ ứng dụng khi cực tiểu hóa (không ở dạng cửa sổ) đều chuyển về thành một nút trên thanh Taskbar. Khi đang làm việc với một ứng dụng, muốn chuyển sang làm việc với một ứng dụng khác (đã được kích hoạt) ta chỉ việc nháy vào nút tương ứng trên Taskbar, khi đó cửa sổ ứng dụng với ứng dụng này sẽ xuất hiện ở trước các ứng dụng khác.

**Dùng lệnh Start / Run** để chạy một chương trình hay mở một tài liệu nếu biết trước đường dẫn cùng tên tệp: nháy nút Start, chọn mục Run, xuất hiện hộp thoại Run, gõ tên chương trình hay một tài liệu cùng đường dẫn (nếu không biết chắc chắn chương trình ở đâu thì nháy nút Browse để xác định nơi cất tệp).

**Dùng lệnh Start / Documents** để mở tài liệu mà ta mở trước đó không lâu: nháy nút Start, chọn mục Documents, xuất hiện menu dọc My Documents, nháy tên tài liệu cần mở (tài liệu có thể tạo bởi Word, Excel, Paint, Notepad...), kết quả là tài liệu được mở cùng với chương trình ứng dụng tạo ra nó cũng được khởi động theo.

**Thoát khỏi chương trình** ứng dụng đang chạy: nháy vào nút Close ở góc trên bên phải cửa sổ ứng dụng. Khi ứng dụng đóng, nút tương ứng với nó trên thanh Taskbar cũng biến mất.

### - Bổ xung một chương trình vào menu Start

**Bổ xung một chương trình vào các menu dọc** xuất phát từ nút Start tiến hành các bước sau:

- Dùng lệnh Start / Settings / Taskbar & Start menu..., xuất hiện hộp thoại Taskbar and Start Menu Properties, chọn lớp Advanced, chọn nút Add, gõ vào đường dẫn đến mục mà ta muốn thêm vào menu hoặc sử dụng nút Browse để tìm đường dẫn tới nó.

- Nháy nút Next, sử dụng hộp thoại Select Program Folder để báo cho Windows biết nơi sẽ đặt mục mới.

- Nháy nút Next, đặt tên cho mục thêm vào menu, cuối cùng nháy nút Finish.

**Loại bỏ một chương trình từ các menu dọc** tiến hành theo các bước: nháy nút Start, trỏ chuột tới mục cần xóa và nháy nút phải chuột, xuất hiện menu tắt, chọn mục Delete, nháy Yes để đưa mục này vào Thùng rác.

**Sắp xếp lại các mục trong các menu con** của menu Programs: Start / Settings / Taskbar & Start menu, chọn lớp Advanced, nhấn nút Re-Sort.

#### - Các thao tác trên thanh Taskbar

**Sắp xếp các cửa sổ ứng dụng.** Nháy nút phải chuột lên nền của Taskbar, xuất hiện menu tắt gồm các mục sau:

- *Cascade Windows*: các cửa sổ ứng dụng đang mở xếp lợp lên nhau.
- *Tile Windows Horizontally*: các cửa sổ ứng dụng xếp cạnh nhau theo chiều ngang.
- *Tile Windows Vertically*: các cửa sổ ứng dụng xếp cạnh nhau theo chiều đứng.
- *Minimize all Windows*: cực tiểu hóa tất cả các cửa sổ đang mở và đưa về Taskbar.

**Điều khiển một cửa sổ ứng dụng đã mở.** Mỗi ứng dụng đã mở sinh ra một Nút trên thanh Taskbar, khi nháy nút phải chuột vào Nút này sẽ xuất hiện một menu tắt với các mục: Restore, Move, Size, Minimize, Maximize, Close Alt+F4 (giống menu điều khiển ở góc trên bên trái cửa sổ, xem mục 1.3).

**Cho ẩn thanh Taskbar.** Nháy nút phải chuột lên trên nền của Taskbar, xuất hiện menu tắt, chọn mục *Properties*, xuất hiện hộp thoại *Taskbar and Start Menu Properties*, chọn lớp General, trong lớp này có các nút kiểm: Always on top (thanh Taskbar luôn luôn hiện), Auto hide (thanh Taskbar chỉ hiện khi con trỏ chuột nằm ở vị trí của nó), Show Clock (hiện đồng hồ trên thanh Taskbar).

**Thay đổi vị trí đặt và kích thước thanh Taskbar.** Thanh Taskbar có thể đặt ở một trong 4 cạnh của màn hình bằng cách nháy vào nền của Taskbar và kéo đi. Thông thường thanh Taskbar để ở cạnh đáy màn hình. Thay đổi độ rộng của thanh Taskbar bằng cách nháy và kéo lên các biên phía trong của nó (tại nơi có xuất hiện mũi tên hai chiều).

**Thanh công cụ Quick Launch** trên Thanh Taskbar bao gồm các biểu tượng: Show Desktop (cực tiểu các cửa sổ đang mở để hiện màn hình nền), Internet Explorer (truy nhập Internet), Outlook Express (chạy chương trình gửi thư điện tử). Bên trái Thanh Quick Launch có một vạch đứng gọi là Tay cầm của thanh, khi nháy vào tay cầm và kéo thanh vào màn hình nền thì Thanh công cụ Quick Launch chuyển thành một cửa sổ nhỏ.

#### e) WINDOWS EXPLORER

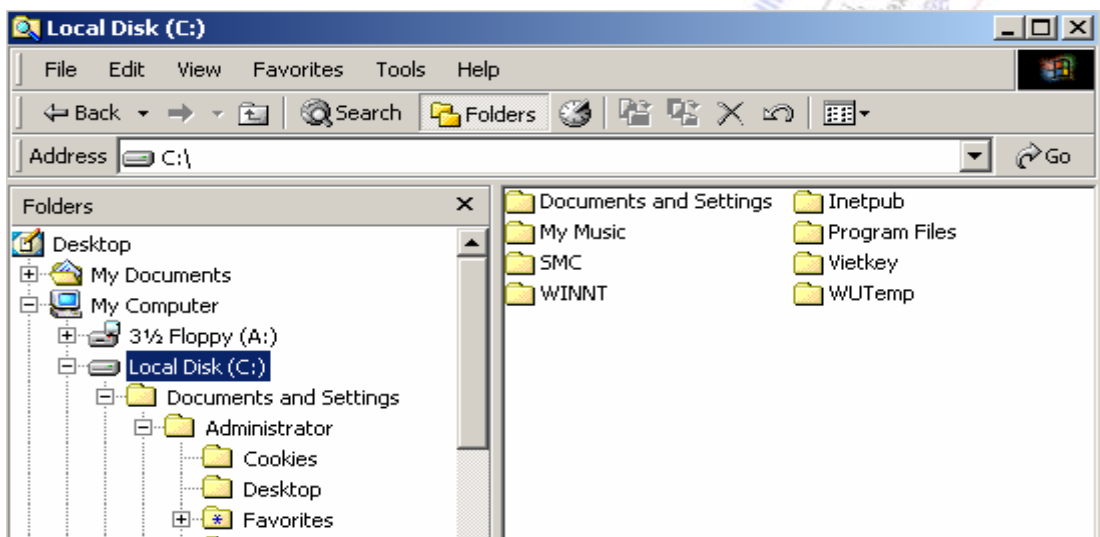
Windows Explorer là công cụ trợ giúp của Windows 2000 giúp ta biết được toàn bộ máy tính có những gì, cho phép xem nội dung máy tính như một hệ thống thứ bậc hay cấu trúc cây. Ta có thể xem dễ dàng nội dung mỗi ổ đĩa, mỗi thư mục, cũng như bất kỳ ổ đĩa nào trong mạng.

#### - Khởi động Windows Explorer

Dùng lệnh Start / Programs / Accessories / Windows Explorer để khởi động Windows Explorer, xuất hiện cửa sổ của chương trình Windows Explorer và nó được chia thành hai phần: khung bên trái và khung bên phải.

**Khung bên trái** là cây thư mục liệt kê tất cả các tài nguyên của máy: các ổ đĩa, Control Panel, thùng rác, mạng.... Trong khung này có các thao tác:

- Nháy lên đối tượng (ổ đĩa, thư mục con ...) thì nội dung bên trong của đối tượng sẽ hiện lên ở khung bên phải. Động tác này gọi là mở một thư mục.
- Nháy vào dấu (+) đứng trước một đối tượng thì trong cây thư mục sẽ hiện tiếp các thư mục con của đối tượng này.
- Nháy vào dấu (-) đứng trước một đối tượng thì trong cây thư mục sẽ không hiện các thư mục con của các đối tượng này.



**Khung bên phải** của cửa sổ liệt kê chi tiết (gồm các thư mục con và tệp) của đối tượng có hộp sáng trong Khung bên trái. Các đối tượng ở Khung bên phải có 5 cách hiện, dùng lệnh View và chọn một trong 5 khả năng:

- **List**: hiện biểu tượng và tên thư mục, tên tệp.
- **Details**: hiện biểu tượng, tên tệp, số byte, kiểu tệp, ngày giờ tạo.
- **Large Icons**: hiện biểu tượng to.
- **Small Icons**: biểu tượng nhỏ xếp hàng ngang.
- **Thumbnails**: hiển thị một hình ảnh chứa đựng các nội dung của tệp nếu có thể.

#### - Tạo một thư mục con mới

Trong khung trái cửa sổ Windows Explorer nháy vào tên thư mục cần tạo một thư mục con mới. Từ menu chính của cửa sổ Windows Explorer ra lệnh File / New / Folder, trong hộp văn bản New Folder ở Khung bên phải cửa sổ gõ vào tên thư mục mới. Tên thư mục mới xuất hiện trong cây thư mục ở Khung bên trái

#### - Lựa chọn tệp và thư mục

- Lựa chọn một đối tượng (một tệp hay một thư mục): nhấp chuột vào biểu tượng thư mục hay tệp.

- Chọn nhiều biểu tượng nằm kế tiếp nhau ở Khung bên phải cửa sổ: nhấp vào đối tượng đầu, ấn và giữ phím Shift đồng thời nhấp vào đối tượng cuối.

- Chọn nhiều đối tượng nằm rời rạc trên Khung bên phải cửa sổ: ấn và giữ phím Ctrl đồng thời nhấp vào từng đối tượng.

- Chọn tất cả các đối tượng trong Khung bên phải cửa sổ: dùng lệnh Edit / Select All.

### - Di chuyển hay sao chép các tệp và thư mục

#### Dùng bảng chọn chính của cửa sổ Windows Explorer

- Lựa chọn các đối tượng (thư mục, tệp) cần di chuyển hay sao chép.

- Nhấp Edit. Tiếp theo nhấp Cut để di chuyển các đối tượng hoặc nhấp Copy để sao chép các đối tượng.

- Trong cây thư mục ở Khung bên trái, nhấp vào tên thư mục đích là nơi mà các đối tượng sẽ được đặt vào. Nhấp Edit, rồi chọn Paste.

#### Dùng nút phải chuột để sao chép tệp hay nhóm tệp

- Lựa chọn tệp hay nhóm tệp.

- Nhấp nút phải chuột vào một tên tệp đã lựa chọn, xuất hiện một Menu tắt.

- Chọn Cut để di chuyển tệp, hoặc chọn Copy để sao chép tệp.

- Mở thư mục muốn đặt tệp vào, dùng nút phải chuột nhấp vào chỗ trống trên Khung bên phải để xuất hiện một menu dọc, chọn Paste.

- Có thể cần sắp xếp lại các biểu tượng bằng lệnh View / Arrange Icons / by Name.

#### Dùng kéo và thả

Đôi khi cách nhanh nhất để di chuyển và sao thông tin là di thả nó từ chỗ này sang chỗ khác bằng chuột. Ta có thể di chuyển hay sao chép thư mục và tệp sang một thư mục khác hay ổ đĩa khác bằng cách:

- Trong cửa sổ Windows Explorer, mở thư mục có chứa tệp cần di chuyển hay sao chép.

- Lựa chọn tệp hay nhóm tệp, dùng nút phải chuột để kéo tệp từ Khung bên phải sang tên thư mục của Khung bên trái, khi tên thư mục này được chiếu sáng thì thả nút chuột ra, xuất hiện một menu tắt.

- Chọn Move Here nếu muốn di chuyển tệp, hoặc chọn Copy Here để sao chép tệp.

### - Xóa tệp và thư mục

Có hai cách:

▪ *Cách 1.* Lựa chọn các tệp hay thư mục cần xóa, dùng lệnh File / Delete, xuất hiện hộp thoại Confirm File Delete, chọn Yes để đưa các đối tượng đã chọn vào thùng rác (Recycle Bin).

▪ *Cách 2.* Lựa chọn các tệp hay thư mục cần xóa, nháy nút phải chuột vào tên một đối tượng đã chọn, xuất hiện một Menu tắt, chọn Delete, xuất hiện hộp thoại Confirm File Delete, chọn Yes.

#### - Sao chép tệp hay thư mục vào đĩa mềm

Các bước tiến hành:

▪ Lựa chọn các tệp hay thư mục cần sao chép ra đĩa mềm.

▪ Dùng lệnh File / Send To, nháy vào tên ổ đĩa ta muốn sao tệp hay thư mục vào đó.

#### - Cho phép các máy trên mạng truy nhập vào một thư mục

Chọn tên thư mục trong Khung bên trái cửa sổ Windows Explorer, nháy nút phải chuột vào tên thư mục, hiện menu tắt, chọn Sharing, hiện hộp thoại Properties, chọn lớp Sharing, chọn mục **Do not share this folder** (các máy khác trên mạng nội bộ không vào được thư mục này) hoặc chọn **Share this folder** (cho phép các máy khác trên mạng truy nhập được vào thư mục, cho phép copy, mở tệp ...).

#### f) DESKTOP và các biểu tượng

Khởi động xong máy tính, xuất hiện màn hình nền (Desktop), trên đó có các biểu tượng. Trong khi làm việc bất kỳ lúc nào ta muốn xuất hiện Desktop thì chỉ cần cực tiểu hóa tất cả các ứng dụng đang chạy về thành các biểu tượng trên thanh Taskbar (xem Mục 2.4), hoặc nháy chuột vào biểu tượng Show Desktop trên Thanh công cụ Quick Launch.

Các biểu tượng trên Desktop chia thành hai loại: biểu tượng mặc nhiên là biểu tượng có sẵn của Windows 2000 như My Computer và Recycle Bin, biểu tượng đường tắt (Shortcut) có dấu mũi tên ở góc dưới bên trái hình vẽ biểu tượng. Biểu tượng đường tắt cung cấp cách truy nhập vào các tài liệu và chương trình thường hay sử dụng một cách dễ dàng bằng cách nháy đúp lên biểu tượng. Ví dụ ta có thể lập một biểu tượng đường tắt tới chương trình C:\NC\NC.EXE. Biểu tượng đường tắt không làm thay đổi vị trí tệp, nó chỉ là một con trỏ mà cho phép ta nhanh chóng mở tệp, không phải vào hệ thống menu. Ta có thể tạo một biểu tượng đường tắt cho một đối tượng bất kỳ như tệp chương trình, tệp văn bản, thư mục, các ổ đĩa, các máy tính khác hay máy in.

#### - Tạo một biểu tượng đường tắt

Có hai cách tạo một biểu tượng đường tắt:

##### Cách 1: dùng menu và hộp thoại

▪ Nháy nút phải chuột lên màn hình nền, xuất hiện menu tắt, chọn New, xuất hiện menu dọc thứ hai, chọn Shortcut, xuất hiện hộp thoại Create Shortcut.

▪ Trong hộp thoại văn bản *Type the Location of the Item* nhập đường dẫn cùng tên tệp chương trình hay tệp văn bản. Có thể nháy nút Browse để dò tìm chương trình trên đĩa.

▪ Nháy nút Next, nhập tên biểu tượng chương trình (do người dùng đặt tùy ý) vào hộp *Type a Name for this Shortcut*.

- Nháy nút Finish để hoàn tất việc tạo biểu tượng đường tắt.

**Cách 2:** kéo và thả

- Trong cửa sổ Windows Explorer lựa chọn mục cần tạo biểu tượng đường tắt.
- Chọn hiện màn hình nền của Windows 2000: kéo cạnh dưới của cửa sổ Windows Explorer lên phía trên để xuất hiện một phần của Desktop.
- Dùng nút phải chuột để kéo mục đã chọn vào Desktop rồi thả nút chuột, xuất hiện menu tắt.
- Nháy Create Shortcut Here.

Khi ta nháy nút phải chuột lên biểu tượng đường tắt sẽ xuất hiện một menu tắt với các mục: Delete (xóa biểu tượng và đưa vào Thùng rác, khi xóa biểu tượng tệp gốc không bị xóa), Rename (đổi tên biểu tượng), Properties (thay đổi các thuộc tính của biểu tượng, xem được nơi chứa chương trình ứng với biểu tượng, thay đổi hình ảnh của biểu tượng).

**Thiết đặt một chương trình chạy ngay hay một tệp văn bản mở ngay khi khởi động Windows:** cho hiện thư mục C:\ Documents and Settings \ Administrator \ Start Menu \ Programes \ StartUp trên Khung bên trái của cửa sổ Windows Explorer, chọn tên chương trình hay tệp văn bản trong Khung bên phải, nháy nút phải chuột vào tên tệp đã chọn và kéo tệp sang Khung bên trái sao cho thư mục StartUp được chiếu sáng, thả nút chuột, xuất hiện menu tắt, chọn mục Copy Here.

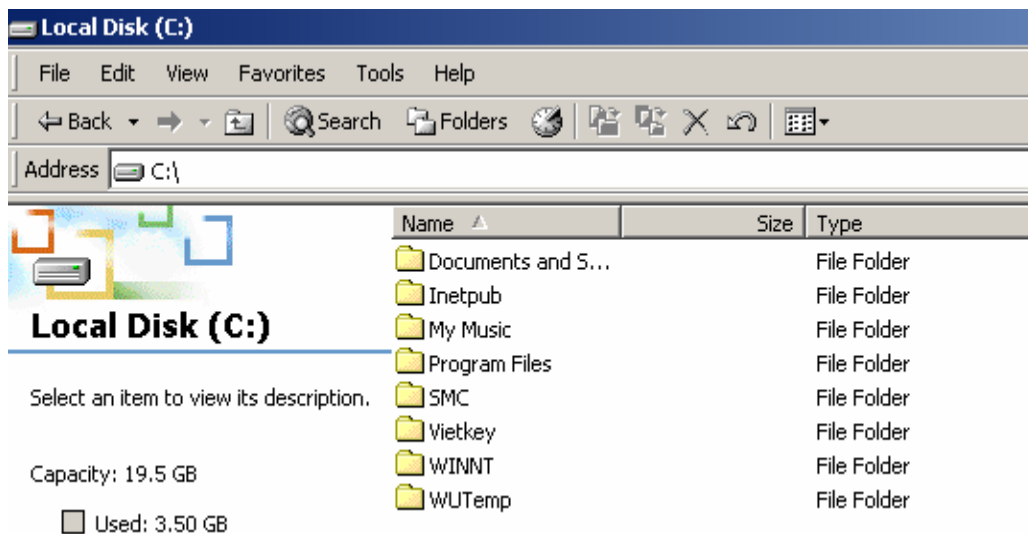
**- Thay đổi màn hình nền**

Dùng lệnh Start / Settings / Control Panel, xuất hiện cửa sổ Control Panel, nháy đúp lên biểu tượng Display, xuất hiện hộp thoại Display Properties với nhiều lớp. Lớp Background: chọn cách trang trí màn hình nền theo các mẫu sẵn, hoặc lấy một ảnh bất kỳ (dùng nút Browse để xác định nơi chứa tệp ảnh). Lớp Screen Saver: thiết lập màn hình chờ (màn hình này xuất hiện nếu sau một khoảng thời gian nhất định ta không sử dụng bàn phím hay chuột).

**- Biểu tượng My Computer**

My Computer là rất hữu dụng nếu ta muốn xem nội dung riêng một thư mục hay một ổ đĩa. Khi ta nháy đúp My Computer trên màn hình nền, các ổ đĩa đang dùng hiện lên trong cửa sổ My Computer. Nháy đúp lên biểu tượng ổ đĩa, cửa sổ sẽ hiện lên các thư mục chứa trong ổ đĩa này. Sau đó ta có thể nháy đúp vào thư mục để xem các tệp có trong thư mục này.





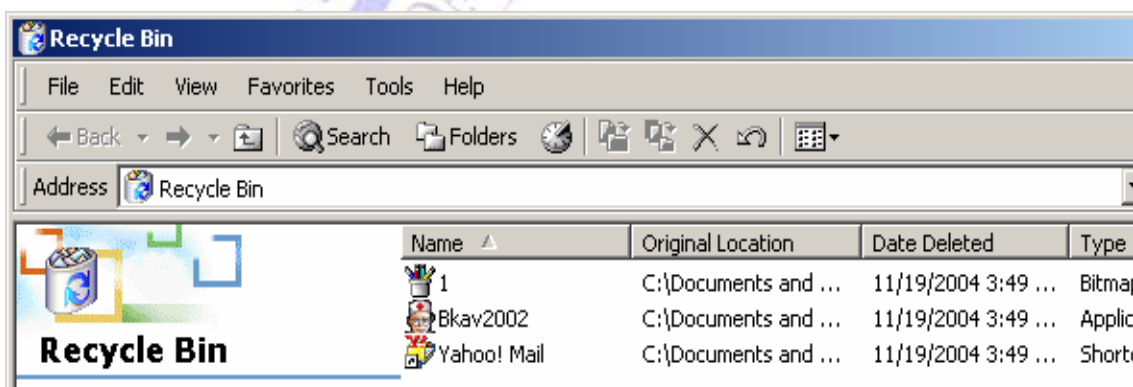
Sử dụng mục View trên menu chính ta có thể thay đổi cách trình bày các tệp và thư mục trong cửa sổ. Tất cả các thao tác đối với tệp và thư mục trong My Computer cũng tương tự như trong Windows Explorer.

Trên thanh công cụ Standard Buttons của cửa sổ My Computer nếu ta không chọn mục Folders thì cửa sổ My Computer sẽ chia thành Khung bên trái (chứa cây thư mục) và Khung bên phải hoàn toàn giống với cửa sổ Windows Explorer.

### - Biểu tượng Recycle Bin (Thùng rác)

Thùng rác là nơi chứa các tệp, các đối tượng bị xóa bỏ khi ta thao tác trong Windows 2000. Các tệp không thực sự bị loại khỏi đĩa cứng cho đến khi ta làm rỗng Thùng rác. Windows Explorer xem thùng rác như một thư mục con, do đó có thể phục hồi các tệp và biểu tượng chương trình bị xóa bằng thao tác sao chép giữa các thư mục.

**Các thao tác đối với Thùng rác.** Nháy đúp chuột lên biểu tượng Recycle Bin, xuất hiện cửa sổ Recycle Bin, trong cửa sổ này có thể làm các thao tác:



- Phục hồi: nháy tên tệp cần phục hồi, dùng lệnh File / Restore, Windows sẽ tự động khôi phục tệp hay biểu tượng chương trình theo nguồn gốc ban đầu.

- Xóa khỏi Thùng rác: nháy tên chương trình cần xóa bỏ, dùng lệnh File / Delete.
- Làm rỗng Thùng rác: dùng lệnh File / Empty Recycle Bin. Theo định kỳ cần làm rỗng Thùng rác để giải phóng không gian đĩa.

### **g) PAINT**

Paint là một công cụ của Windows 2000 cho phép tạo và hiệu chỉnh các hình ảnh đồ họa: một bức vẽ trong Paint, thông tin đồ họa lưu trong Clipboard, một bức ảnh ghi ở dạng BMP, GIF, JPEG.

#### **- Khởi động Paint**

Dùng lệnh Start / Programs / Accessories / Paint để mở cửa sổ Paint.

Vùng trắng ở giữa là vùng vẽ. 16 nút bên trái cửa sổ là hộp công cụ (Tool box), công cụ chính để vẽ là bút lông (Brush) và bút chì (Pencil). Góc dưới bên trái là Hộp màu (Color Box) dùng để chọn màu, kích trái của hộp màu là một ô cho thông tin về màu phía trước (foreground color) và màu phía sau (background color) hiện hành.

Trên vùng vẽ là các điểm ảnh (pixel) xếp theo hàng và cột, góc trên bên trái vùng vẽ có tọa độ (0, 0), trục X hướng sang phải, trục Y hướng xuống dưới. Phía dưới cửa sổ Paint là thanh trạng thái (status bar), trên đó có hai ô chứa hai cặp số. Cặp số thứ nhất có dạng (x, y) thông báo vị trí của con trỏ, hoặc là tọa độ bắt đầu của đối tượng đang vẽ (ví dụ mút đầu tiên của đường thẳng, góc trên trái của hình chữ nhật, ...). Cặp số thứ hai có dạng (dx × dy) thông báo kích thước của đối tượng đang vẽ. Ví dụ khi đang kéo chuột vẽ hình chữ nhật hai cặp số là (87, 47) và (180 × 155) sẽ có nghĩa: tọa độ góc trên trái hình chữ nhật là x = 87, y = 47, chiều ngang hình chữ nhật là dx = 180, chiều cao là dy = 155. Sử dụng các tọa độ này để vẽ chính xác hình.

**Thiết lập kích thước bức vẽ** dùng lệnh Image / Attributes, xuất hiện hộp thoại Attributes. Các mục trong hộp thoại: Width - thay đổi độ rộng bức vẽ, Height - thay đổi độ cao bức vẽ, Units - thay đổi đơn vị đo là Cm hay Inches, Color - để chọn màu của bức vẽ là trắng đen hay có màu, Default - trả lại trạng thái mặc định.

#### **Các bước chung để tạo một bức vẽ:**

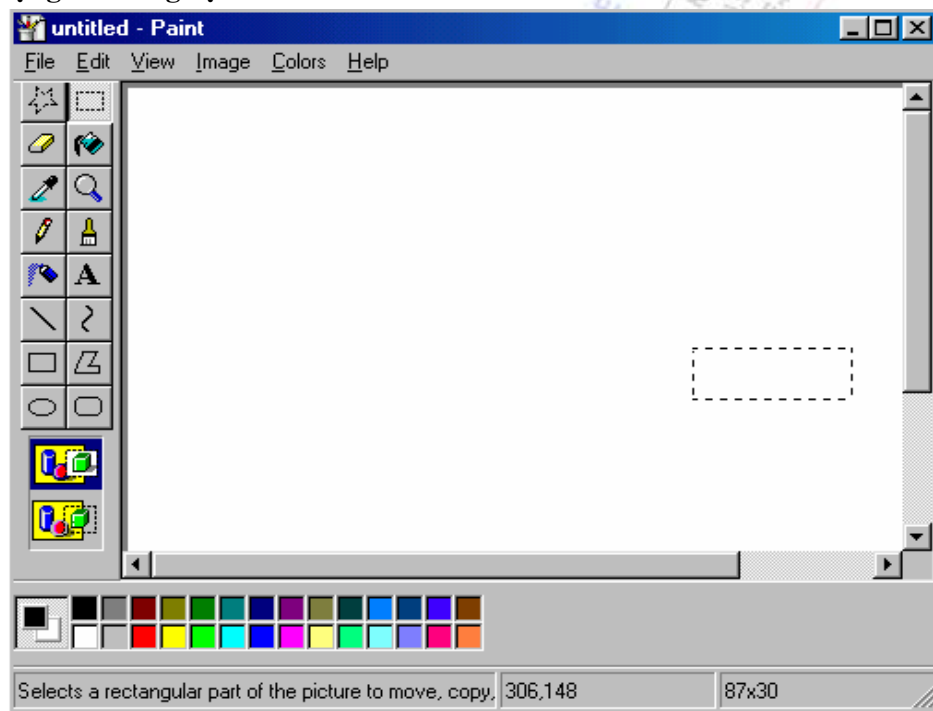
- Chọn công cụ vẽ: nháy vào nút. Hình vẽ trên nút mô tả công dụng của nút, song nếu không biết chính xác hãy để con trỏ lên nút một lát tên nút xuất hiện.
- Chọn độ rộng đường, kiểu bút lông hay loại hình chữ nhật từ nhóm các lựa chọn phía dưới hộp công cụ.
- Chọn màu phía trước: nháy nút trái chuột lên màu cần chọn trong hộp màu. Có thể tạo một màu theo ý thích bằng lệnh Colors / Edit Colors / Define Custom Colors.
- Chọn màu nền: nháy nút phải chuột vào màu cần chọn trong hộp màu.
- Tiến hành vẽ. Trong khi vẽ nếu dùng nút trái chuột sẽ lấy màu phía trước, dùng nút phải chuột sẽ lấy màu nền. Muốn hủy những gì vừa vẽ dùng lệnh Edit / Undo (có thể dùng tới 3 lần), muốn khôi phục những gì đã bị Undo dùng lệnh Edit / Repeat (có thể dùng tới 3 lần).

**Xem các bức vẽ lớn.** Với những bức vẽ lớn cần nhiều chỗ trên màn hình để xem, ta có thể không cho hiện hộp công cụ, hộp màu và thanh trạng thái bằng cách dùng lệnh View, bỏ đánh dấu vào các mục Tool box, Color box, Status bar. Muốn rộng hơn nữa dùng lệnh View / View Bitmap (ấn phím bất kỳ về trạng thái bình thường).

**Ghi một bức vẽ** dùng lệnh File / Save hay lệnh File / Save As, hộp thoại Save As xuất hiện. Trong hộp Save As Type hãy chọn một trong các dạng bitmap: Monochrome Bitmap (lưu bức vẽ trắng đen), 16 Color Bitmap (bức vẽ 16 màu), 256 Color Bitmap (bức vẽ nhiều phổ màu), 24 Bit Bitmap (lưu bức vẽ bằng phổ màu đầy đủ, chiếm nhiều không gian đĩa nhất). Gõ tên tệp vào hộp File Name, nhấn Save.

**Mở một bức vẽ đã lưu trữ** dùng lệnh File / Open.

**- Sử dụng các công cụ vẽ**



**Vẽ tự do bằng công cụ Pencil.** Pencil là công cụ vẽ mặc định khi khởi động Paint.

Nháy nút trái (nút phải) chuột vào điểm bắt đầu vẽ, giữ chuột và kéo để vẽ theo màu phía trước (màu nền). Muốn vẽ các đường hoàn toàn đứng, ngang hay xiên 45 độ thì phải giữ phím Shift khi vẽ.

**Vẽ tự do bằng công cụ Brush.** Khi đã chọn Brush, ta cần chọn kiểu của bút lông trong hộp ở liền phía dưới hộp công cụ. Nhấn giữ nút trái (nút phải) chuột và kéo để vẽ theo màu phía trước (màu nền).

**Son xịt bằng công cụ Airbrush.** Chọn Airbrush, chọn một trong ba mẫu bình xịt ở phía dưới hộp công cụ, di con trỏ vào vùng vẽ, ấn nút trái (nút phải) chuột và kéo để tạo các đám chấm hình tròn có màu phía trước (màu nền). Các đám chấm mau hay thưa phụ thuộc vào kéo chuột nhanh hay chậm.

**Vẽ các đường thẳng bằng công cụ Line.** Chọn Line, chọn độ rộng đường trong hộp ở phía dưới hộp công cụ, di chuột vào vùng vẽ, nhấn giữ nút trái (nút phải) chuột và kéo để vẽ đường thẳng theo màu phía trước (màu nền). Để vẽ các đường hoàn toàn đứng, ngang hay chéo cần ấn phím Shift khi vẽ.

**Vẽ các đường cong lượn sóng bằng công cụ Curve:**

- Chọn công cụ Curve, chọn độ rộng nét vẽ ở hộp phía dưới Hộp công cụ.
- Định vị con trỏ vào nơi bắt đầu đường cong, nhấn giữ nút trái (nút phải) chuột và kéo tới nơi kết thúc đường cong, nhả nút chuột, ta được một đường thẳng theo màu phía trước (màu nền).
- Đưa con trỏ vào một phía của đường thẳng, nhấn giữ nút chuột và kéo để đánh cong đường thẳng về một phía, nhả chuột (ta có đường với một nét cong). Đưa con trỏ sang phía ngược lại, nháy chuột và kéo để được đường cong dạng lượn sóng. Trong lúc đang uốn cong đường nếu ta ấn đồng thời cả hai nút chuột thì đường cong bị hủy bỏ.

**Vẽ hình chữ nhật và hình vuông bằng công cụ Rectangle.** Giả sử dùng nút trái chuột để vẽ, chọn công cụ Rectangle, chọn một trong ba loại hình chữ nhật ở phía dưới hộp công cụ: *Loại thứ nhất* là hình chữ nhật chỉ có viền và được vẽ theo màu mặt tiền; *Loại thứ hai* là hình chữ nhật có viền được tô bằng màu phía trước, phần bên trong được tô bằng màu mặt hậu; *Loại thứ ba* là hình chữ nhật đặc được tô đầy bằng màu mặt tiền. Di con trỏ đến vùng vẽ, nhấn giữ nút chuột và kéo để tạo hình chữ nhật.

Nếu dùng nút phải chuột để vẽ thì sẽ dùng màu nền để vẽ, các hình chữ nhật có màu ngược lại. Muốn được hình vuông thì giữ phím Shift trong khi kéo chuột.

Để xác định độ rộng đường viền của hình chữ nhật, hình chữ nhật có góc tròn, ellipse, hay đa giác, trước tiên chọn công cụ Line, chọn độ rộng đường, tiếp theo mới chọn công cụ Rectangle, Ellipse hay Polygon.

**Vẽ hình chữ nhật với các góc tròn** dùng công cụ Rounded Rectangle, thao tác giống công cụ Rectangle.

**Vẽ hình Ellipse và hình tròn bằng công cụ Ellipse:** thao tác giống công cụ Rectangle.

**Vẽ các hình đa giác bằng công cụ Polygon:** Giả sử dùng nút trái chuột để vẽ một đa giác có N cạnh, chọn công cụ Polygon, chọn một trong ba loại đa giác trong hộp phía dưới hộp công cụ: đa giác chỉ có viền và được vẽ bằng màu phía trước, đa giác có viền tô bằng màu phía trước và ruột tô bằng màu nền, đa giác tô đặc bằng màu phía trước. Nhấn giữ nút trái chuột và kéo để vẽ cạnh đầu tiên, nháy nút trái chuột vào điểm cuối của cạnh thứ hai (Paint sẽ nối điểm này với điểm cuối của cạnh đầu tiên), tiếp tục để được N-1 cạnh, nháy đúp vào điểm cuối của cạnh N-1 ta nhận được cạnh cuối cùng.

Nếu dùng nút phải chuột để vẽ thì ta sẽ dùng màu nền và đa giác có màu ngược lại.

**Tô đầy một hình đóng kín bằng Fill with Color.** Chọn công cụ Fill with Color, đưa con trỏ vào bên trong vùng đóng kín định tô, nháy nút trái chuột để tô bằng màu phía trước (hoặc nháy nút phải chuột nếu tô bằng màu nền).

**Thêm văn bản bằng công cụ Text:**

- Chọn công cụ Text, chọn một trong hai cách hiện văn bản nhờ hai biểu tượng ở phía dưới hộp công cụ. Cách ở trên: nền của khung văn bản là màu nền, màu chữ là màu phía trước. Cách ở dưới: màu chữ là màu phía trước, khung văn bản không có nền, do đó vẫn nhìn thấy những gì đã vẽ ở phía dưới khung văn bản.

- Nháy giữ nút trái chuột và kéo để tạo khung văn bản hình chữ nhật, xuất hiện con trỏ nhập văn bản và thanh công cụ Fonts:



Chọn phông chữ, cỡ chữ và kiểu chữ. Nếu không xuất hiện thanh công cụ Fonts thì dùng lệnh View / Text Toolbar.

- Nhập văn bản từ bàn phím, hoặc dán văn bản có sẵn trong Clipboard bằng lệnh Edit / Paste. Click chuột ở bên ngoài khung văn bản để chấp nhận, sau khi chấp nhận ta không thể sửa được.

**- Sử dụng các công cụ hiệu chỉnh**

**Xóa bằng công cụ Eraser:** Chọn công cụ Eraser / Color Eraser, chọn một trong 4 kích thước cục tẩy xuất hiện phía dưới hộp công cụ, chọn màu mặt hậu. Định vị con trỏ chuột vào nơi bắt đầu xóa, nhấn giữ nút trái chuột và kéo, vùng bị xóa được thay bằng màu nền.

**Chọn màu bằng công cụ Pick Color:** Chọn màu phía trước: chọn Pick Color, nháy nút trái chuột vào vị trí trên bức vẽ có màu mà ta muốn làm màu phía trước. Chọn màu nền: chọn Pick Color, nháy nút phải chuột vào vị trí trên bức vẽ có màu mà ta muốn làm màu nền.

**Công cụ Magnifier dùng để phóng một phần hay toàn bộ bức vẽ:**

- Để phóng một phần bức vẽ chọn công cụ Magnifier, di con trỏ (là khung chữ nhật lớn) lên phần của bức vẽ muốn phóng to, nháy chuột. Ta có thể xem và sửa chữa hình vẽ. Kết thúc phóng to: nháy lại công cụ Magnifier, nháy vào điểm bất kỳ trên bức vẽ.

- Phóng to toàn bộ bức vẽ: chọn công cụ Magnifier, chọn mức phóng 2 lần, 6 lần hay 8 lần ở dưới hộp công cụ. Xem hay sửa bản vẽ, trở về không phóng: nháy lại vào nút Magnifier, chọn mức phóng một lần ở dưới hộp công cụ.

**- Làm việc với các mẫu cắt**

**Tạo một mẫu cắt:**

- **Cách 1:** Tạo mẫu cắt hình chữ nhật. Chọn công cụ Select, di chuyển con trỏ chuột vào đỉnh trên trái mẫu cắt, ấn giữ nút trái và di chuyển xuống đỉnh dưới phải thì thả chuột. Mẫu cắt được đánh dấu bằng đường viền gạch hình chữ nhật.

- **Cách 2:** Tạo mẫu cắt theo hình bất kỳ. Chọn công cụ Free Form Select, di chuyển con trỏ chuột vào vùng vẽ rồi rê chuột để vẽ một đường bao quanh mẫu cắt, thả nút chuột, xuất hiện đường chữ nhật có chấm bao quanh mẫu cắt. Nếu làm sai, có thể nháy tại một vị trí bất kỳ trên màn hình để làm lại.

Lựa chọn một trong hai cách thức mẫu cắt có thể được di chuyển, sao chép hay quét nhờ hai tùy chọn ở ngay dưới hộp công cụ. Chọn cách đầu tiên (**dạng mờ đục**): phần mặt hậu của mẫu cắt cũng mang tới nơi mới, che lấp những gì đã vẽ cũ ở đây. Chọn cách thứ hai (**dạng trong suốt**): bất kỳ phần nào của mẫu cắt có màu trùng với màu mặt hậu hiện hành đều biến mất, cho phép hình ảnh nằm phía dưới hiển thị.

**Xóa mẫu cắt:** Dùng lệnh Edit / Cut (hoặc ấn Ctrl + X). Chú ý, với cách xóa này, mẫu cắt vẽ chưa bị mất hẳn, có thể khôi phục lại bằng lệnh Edit / Paste, mẫu cắt sẽ hiện lại tại đỉnh bên trái cửa sổ vẽ, dùng chức năng di chuyển mẫu cắt để đặt lại vị trí mong muốn.

**Di chuyển mẫu cắt:** Đặt con trỏ chuột vào phần trong mẫu cắt, ấn giữ nút trái chuột và di chuyển vùng ảnh đến vị trí mới thì nhả chuột.

**Sao chép mẫu cắt:** Dùng lệnh Edit / Copy để sao chép mẫu cắt vào vùng đệm. Dùng lệnh Edit / Paste dán mẫu cắt từ vùng đệm lên đỉnh bên trái cửa sổ vẽ, dùng chức năng di chuyển để đưa mẫu cắt vào vị trí mong muốn. Chú ý: thao tác dán có thể lặp lại nhiều lần. Trong trường hợp ảnh có nhiều vùng giống nhau thì ta chỉ vẽ một lần rồi thực hiện thao tác dán. Phần ảnh cắt hoặc sao chép được lưu giữ trong vùng đệm, ta có thể lợi dụng điều này để dán nó vào các cửa sổ chương trình khác.

**Quét một mẫu cắt:** Có nghĩa là tạo ra một vết các bản sao của mẫu cắt khi băng qua bức vẽ. Đầu tiên di chuyển mẫu cắt tới nơi muốn bắt đầu quét, cho con trỏ vào trong khung chữ nhật của mẫu cắt, nhấn giữ phím Shift và kéo chuột, số lượng bản sao phụ thuộc vào tốc độ kéo chuột.

**Kéo dãn và kéo nghiêng một mẫu cắt:** Dùng lệnh Image / Stretch-Skew, xuất hiện hộp thoại Stretch and Skew, nhập vào số phần trăm để thay đổi kích thước theo chiều ngang và chiều dọc, nhập vào số độ (âm hay dương) để kéo nghiêng mẫu cắt theo chiều ngang hay chiều dọc.

**Lật hay quay một mẫu cắt:** Dùng lệnh Image / Flip-Rotate, xuất hiện hộp thoại Flip and Rotate, lựa chọn lật nó theo chiều ngang hay chiều dọc, quay nó một góc 90 độ, 180 độ hay 270 độ.

**Sao chép mẫu cắt thành tệp riêng trên đĩa:** Đánh dấu vùng ảnh, dùng lệnh Edit / Copy to.... Hộp thoại xuất hiện cho phép đặt tên tệp ảnh và chọn tên thư mục sẽ lưu ảnh, kết thúc chọn OK.

**Dán ảnh từ một tệp vào bản vẽ:** Dùng lệnh Edit / Paste From..., Hộp thoại xuất hiện cho phép chọn tên tệp ảnh và thư mục chứa nó rồi chọn OK. Kết quả nội dung ảnh mới sẽ được đưa vào màn hình tại đỉnh trên trái cửa sổ vẽ. Dùng chức năng di chuyển để đặt ảnh vào vị trí mong muốn trên bản vẽ.

#### **h). Chương trình soạn thảo văn bản Notepad**

Notepad là một chương trình soạn thảo văn bản tương đối đơn giản, chạy tốc độ nhanh, nó tiện dùng trong nhiều trường hợp: soạn thảo văn bản mã ASCII thông thường, không có hình ảnh, không có định dạng phức tạp.

Muốn kích hoạt chương trình soạn thảo văn bản Notepad ta làm như sau: Dùng lệnh Start/Program/Accessories/Notepad, xuất hiện cửa sổ màn hình của Notepad. trên menu chính gồm có 3 mục cơ bản, mỗi mục ứng với 1 menu dọc:

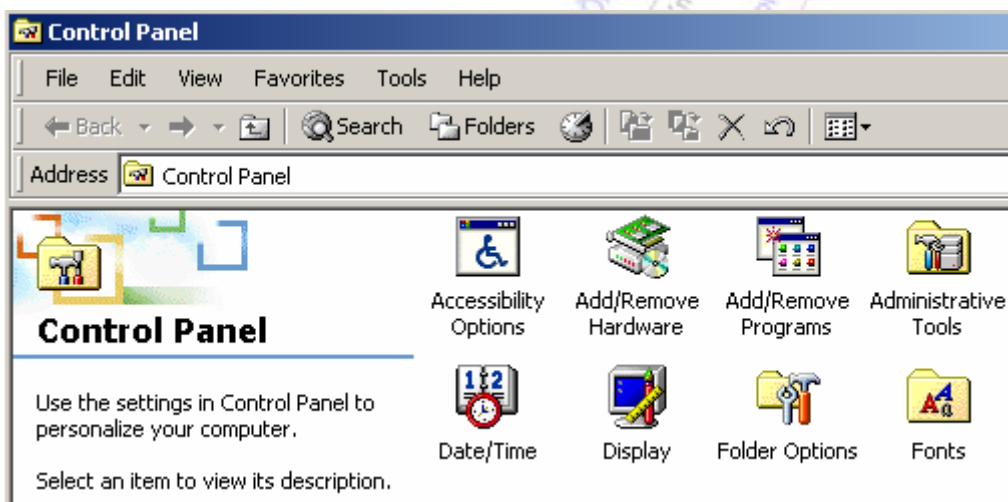
- Menu **File** gồm các mục: New(tạo tệp mới), Open( mở tệp), Save(ghi tệp vào đĩa), Save as(ghi tệp với tên khác), Page setup( chọn khổ giấy và đặt lề), Print( in văn bản), Exit( thoát).

- Menu **Edit** gồm các mục: Undo (hủy thao tác vừa làm), (Xóa khối đưa vào vùng đệm), Copy (chép khối đưa vào vùng đệm), Paste (dán khối từ vùng đệm vào chỗ con trỏ), Delete (xóa khối), Find( Tìm kiếm một chuỗi ký tự), Find Next (tìm tiếp), Replace (Tìm kiếm một chuỗi ký tự và thay thế bằng chuỗi khác), Go to (đưa con trỏ về dòng nhất định), Select All (lựa chọn khối là cả văn bản), Time/date (chèn ngày, tháng hiện tại vào văn bản).

- Menu **Format**: gồm có hai mục: WordWrap( con trỏ từ động xuống dòng khi nhập), Font ( Đăth Font chữ cho toàn văn bản, cho khối đã chọn).

### i) CONTROL PANEL

Chương trình ứng dụng Control Panel dùng để cài đặt hay thay đổi cấu hình của hệ thống. Có thể kích hoạt Control Panel bằng lệnh Start / Settings / Control Panel. Xuất hiện cửa sổ Control Panel:



Trong cửa sổ Control Panel có các biểu tượng với các chức năng sau:

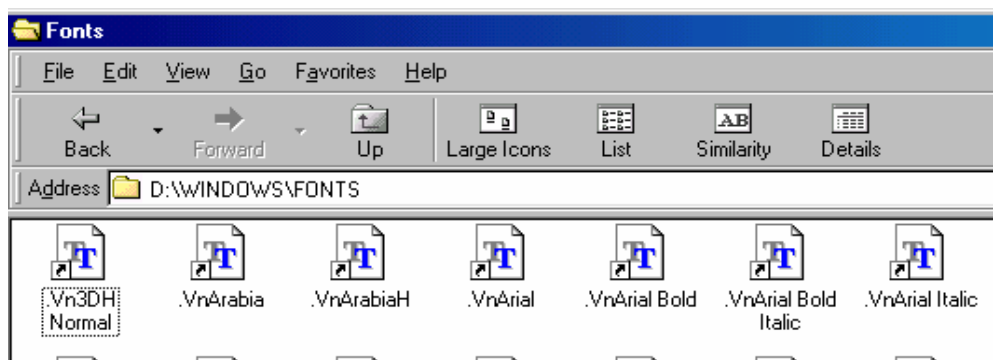
- **Accessibility Options:** Thiết lập các thông số cấu hình thiết bị như bàn phím, màn hình, chuột, âm thanh làm cho máy tính dễ dàng sử dụng với những người khuyết tật.
- **Add / Remove Hardware:** Cài đặt và đặt cấu hình phần cứng như máy in, modem, máy quét, bộ điều khiển trò chơi.
- **Add / Remove Programs:** Cài đặt hay gỡ bỏ các phần mềm ứng dụng. Chức năng này có ba mục chính: Change or Remove Programs (thay đổi hay gỡ bỏ ứng dụng), Add New Programs (bổ xung ứng dụng mới), Add / Remove Windows Components (bổ xung hay gỡ bỏ các thành phần của Windows).
- **Administrative Tools:** Các công cụ quản trị.
- **Date / Time:** Thiết lập ngày giờ hệ thống và đặt lại múi giờ.
- **Display:** Thiết lập màn hình nền và màn hình chờ.

- **Folder Options:** Thiết lập cách hiện các tệp và thư mục.
- **Fonts:** Hiện và quản lý các phông chữ.
- **Game Controllers:** Cài đặt và cấu hình bộ điều khiển trò chơi.
- **Internet Options:** Cài đặt cách hiển thị Internet và các thiết lập kết nối.
- **Keyboard:** thiết lập các thông số của bàn phím, độ nhấp nháy của con trỏ.
- **Mail:** Thiết lập cho Microsoft Outlook.
- **Mouse:** Thay đổi các thông số của chuột. Cửa sổ Mouse Properties có ba mục: Buttons (hoán đổi các nút trái và phải, qui định tốc độ thao tác và nhấp đúp chuột), Pointers (thay đổi các kiểu hình dạng của con trỏ chuột), Motion (qui định tốc độ di chuyển con trỏ chuột và bật tắt việc tạo vết của chuột).
- **Network and Dial-up connections:** Tạo các kết nối mạng.
- **Phone and modem Options:** Cài đặt và cấu hình modem.
- **Power Options:** Quản lý nguồn năng lượng.
- **Printers:** Cài đặt máy in.
- **Regional Options:** Thiết đặt cách hiển thị ngôn ngữ, số, thời gian và ngày tháng.
- **Scanners and Cameras:** Cài đặt và cấu hình một máy quét hay máy ghi hình.
- **Scheduled Tasks:** Chạy các chương trình theo lịch trình.
- **Sound and Multimedia:** Gắn âm thanh cho các sự kiện và cài đặt các thiết bị âm thanh.
- **System:** Xem các thông tin về hệ thống và thay đổi các thiết lập môi trường.
- **Users and Passwords:** Quản lý các người dùng và mật khẩu đối với máy tính đang dùng.

### Cài đặt các phông chữ

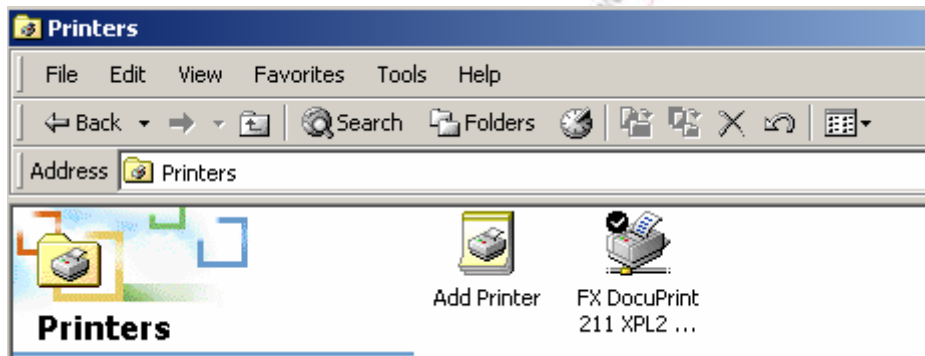
Trong cửa sổ Control Panel nhấp đúp lên biểu tượng Fonts để mở cửa sổ Fonts. Trong cửa sổ Fonts các biểu tượng có chữ **O** ứng với dạng phông chữ OpenType (là phông chữ co giãn được của Windows 2000), các biểu tượng có chữ **T** ứng với dạng phông chữ TrueType (phông chữ co giãn được của các phiên bản Windows trước đây), các biểu tượng có chữ **A** ứng với dạng phông chữ Vector (là phông vẽ có chất lượng thấp) và Raster (còn gọi là Font Bitmap được lưu trữ như các hình ảnh với các kích cỡ nhất định, có chất lượng tốt). Trong cửa sổ này có thể làm các thao tác:

- **Xem các kiểu Fonts:** nhấp đúp lên biểu tượng hoặc tên Font tương ứng, xuất hiện cửa sổ cho mẫu chữ ở các kích cỡ khác nhau.
- **Cài thêm Fonts mới:** dùng lệnh File / Install New Fonts ... Sau đó cần chỉ ra ổ đĩa và đường dẫn chỉ đến các tệp lưu trữ phông chữ, lựa chọn tên phông chữ cần thêm trong hộp List of Fonts.





- **Xóa bớt phông chữ:** nhấn tên phông chữ cần xóa, bấm phím Delete, chọn nút Yes.
  - **Thay đổi các thuộc tính của phông chữ:** nhấn nút phải chuột lên tên phông chữ, xuất hiện menu dọc nhỏ, chọn mục Properties để thay đổi.
  - **Thay đổi cách hiển thị trong cửa sổ Fonts:** dùng lệnh View, chọn các mục Large Icons, List, List Fonts by Similarity, Detail.
- Cài đặt máy in:** Trong cửa sổ Control Panel, nhấn đúp lên biểu tượng Printer, xuất hiện cửa sổ Printer, trong cửa sổ này có thể làm các thao tác:



- **Cài thêm một máy in mới:** nhấn vào biểu tượng Add Printer, nhấn nút Next, chọn cổng nối máy in (mặc nhiên là LPT1), nhấn nút Next, chọn hãng máy in trong hộp Manufactures, kiểu máy in trong hộp Printers. Khi cài đặt máy in thường cần tới đĩa gốc CD của bộ Windows 2000.
- **Hủy bỏ một máy in đã cài đặt:** nhấn biểu tượng máy in cần xóa, nhấn nút Delete.
- **Thay đổi các thông số của máy in:** nhấn nút phải chuột lên biểu tượng của máy in.

## 2.4. HỆ ĐIỀU HÀNH UNIX

Trong phần phân loại hệ điều hành chúng tôi đã đưa ra một số hệ điều hành đặc trưng cho mỗi loại, ví dụ như hệ điều hành PC-DOS, MS-DOS, WINDOWS, WINDOWNT, LINUX, UNIX, MACINTOSH.... Trước kia MS-DOS là hệ điều hành hay được sử dụng nhưng ngày nay do nhu cầu của con người cùng với sự bùng nổ của công nghệ thông tin, MS-DOS không còn đáp ứng được các yêu cầu mới nữa và WINDOWS trở thành hệ điều hành được sử dụng rộng rãi nhất. Bên cạnh đó, UNIX cũng là hệ điều hành được ưa chuộng và sử dụng rộng rãi. Phần này sẽ giới thiệu các đặc điểm và các lệnh cơ bản của hệ điều hành UNIX.

## 2.4.1. Tổng quan về hệ điều hành UNIX

### 2.4.1.1. Lịch sử

Năm 1969 Ken Thompson đã lần đầu tiên cài đặt hệ điều hành UNIX. Mục tiêu khởi đầu là cung cấp một môi trường máy tính hoá để mô phỏng trò chơi không gian. Năm 1973 Ritchie và Thompson đã viết lại hệ điều hành bằng ngôn ngữ C, khác hẳn với các hệ điều hành truyền thống ghi bằng ngôn ngữ máy, do đó UNIX rất dễ cài đặt trên các hệ máy khác. Từ đó đã khai sinh ra hệ điều hành UNIX. Năm 1974 hệ thống UNIX đã được cài đặt trên các máy DEC PDP-11 ở hơn 100 trường đại học. Mục tiêu chủ yếu là cung cấp môi trường cho các lập trình viên chuyên nghiệp.

Ngày nay, 20 năm đã qua, có hàng trăm ngàn hệ thống UNIX cài đặt trên khắp thế giới. Hầu hết các hãng sản xuất máy đều có một phiên bản cho UNIX.

Tuy nhiên hiện nay để chuẩn hoá hệ điều hành UNIX, người ta quy ước các tập lệnh chuẩn và gọi là UNIX System V Release 4. Trên máy PC hiện nay phổ biến hai hệ điều hành là SCO UNIX và SUN Solaris.

### 2.4.1.2. Các đặc điểm cơ bản

Hệ điều hành UNIX có một số đặc điểm sau:

- Đa nhiệm
- Nhiều người sử dụng
- Bảo mật
- Độc lập phần cứng
- Kết nối mở
- Dùng chung thiết bị
- Tổ chức tập tin phân cấp

#### **a/ Nhiều người sử dụng:**

Nhiều người sử dụng có thể sử dụng máy tính có cài UNIX tại một thời điểm. Ví dụ:

UNIX Server:

- User A: dùng Oracle
- User B: chương trình biên dịch
- User C: gửi thư

Hệ điều hành UNIX quản lý những người sử dụng theo cấu trúc phân cấp, người sử dụng có thể giao tiếp với nhau theo các nhóm. Người sử dụng cao nhất (super user) có thể can thiệp đến các người sử dụng khác nếu cần.

#### **b/ Đa chương trình**

Tại một thời điểm một người sử dụng có thể thực hiện đồng thời nhiều tác vụ. Với hệ điều hành đơn chương trình như MS-DOS một lệnh thực hiện sẽ chiếm toàn bộ thời gian CPU xử lý, bạn chỉ có thể thực hiện lệnh kế khi lệnh trước đó đã được thực hiện xong. Còn trong hệ điều

hành UNIX bạn có thể đặt lệnh chạy ở chế độ nền (background) đồng thời khi đó có thể thực hiện các lệnh kế.

**c/ Tổ chức tập tin phân cấp**

Các tập tin của UNIX được tổ chức theo dạng cây có chung thư mục gốc được biểu diễn bởi ký tự /.

Bên trong thư mục có thể là các thư mục con hay các tập tin . UNIX có 3 loại tập tin:

- Tập tin bình thường (ordinary file): là một tập tin chứa các dữ liệu ASCII hay nhị phân.
- Tập tin thư mục (directory file): chứa danh sách các phần tử (thư mục, tập tin, thiết bị) có thể truy xuất tới.
- Tập tin đặc biệt (special file): là các tập tin liên quan đến các thiết bị phần cứng và truyền thông.

Ví dụ:

- bàn phím là một tập tin nhập (input).
- màn hình là một tập tin xuất (output).

```

/
UMUNIX - bin - users - dev
user1
user2
(n ổ đĩa)
    
```

Đối với Unix toàn bộ hệ thống tập tin chỉ có một root. Có thể sử dụng lệnh mount để kết nối các ổ đĩa trong một hệ thống tập tin duy nhất.

```

Ổ 1 : /  LIB
        BIN
        USR
Ổ 2: /-  ETC
        ORACLE
mount => /-  LIB
        BIN
        USR-  ETC
        -  ORACLE
    
```

**d/ Độc lập phần cứng**

Vì hệ điều hành UNIX được viết bằng ngôn ngữ cấp cao cho nên nó rất dễ cài đặt trên các cấu hình phần cứng khác. Hơn nữa với cách tổ chức các thiết bị là các tập tin đặc biệt nên việc thêm vào hay loại bỏ các thiết bị rất dễ dàng.

### **e/ Dùng chung thiết bị**

Vì Unix là môi trường nhiều người sử dụng do đó các thiết bị ngoại vi như máy in, v.v... có thể được dùng chung bởi nhiều người sử dụng.

### **f/ Bảo mật**

Unix cung cấp rất nhiều cơ chế bảo mật khác. Trong đó mỗi người sử dụng có một số quyền trên các tập tin nhất định và chỉ được phép chạy một số chương trình nhất định. Ngoài ra cơ chế mã hoá và giải mã cũng là một phần của hệ điều hành.

### **g/ Kết nối mở**

Unix cung cấp các thiết bị mạng qua Ethernet, Modem, X25. Với nhiều thủ tục truyền thông khác nhau UUCP (Unix-to-Unix Copy), TCP/IP, và các ứng dụng E-mail, FTP, NFS (Network File System).

#### **2.4.1.3. Các thành phần chính của hệ điều hành UNIX:**

- Windows & Graphic User Interface
- Shell
- Lệnh và tiện ích
- Các bộ điều khiển thiết bị
- Kernel

#### **a/ Kernel:**

Là thành phần chủ yếu hay trái tim của hệ điều hành. Nó chiếm khoảng 500KB --> 2MB tùy theo tính chất phức tạp của hệ thống. Nhiệm vụ chủ yếu của Kernel là:

- Quản lý tài nguyên: như bộ nhớ, v.v...
- Quản lý hệ thống tập tin: có thể là các tập tin, thư mục cục bộ hay từ xa.
- Quản lý các quá trình thường trú (daemon).
- Quản lý bộ nhớ ảo: để thực thi nhiều quá trình đồng thời trong khi số lượng bộ nhớ có hạn, Unix tổ chức một vùng trên đĩa như là một vùng bộ nhớ (bộ nhớ ảo). Kernel phải "swap" các quá trình giữa bộ nhớ --> bộ nhớ ảo

RAM

Kernel                      Vùng làm việc                      Swap

Đĩa cứng

Swap

bộ nhớ ảo

- Quản lý quá trình:

Như đã biết vì Unix là một hệ điều hành đa chương do đó việc quản lý các quá trình đồng thời rất phức tạp. Nó phải quản lý việc khởi tạo và kết thúc các quá trình cũng như các tranh chấp có thể xảy ra.

- Quản lý các bộ điều khiển thiết bị.
- Quản lý mạng: bao gồm nhiều thiết bị phần cứng khác và các thủ tục khác.
- Quản lý việc khởi động và dừng máy.

**b/ Bộ điều khiển thiết bị:**

UNIX thể hiện các thiết bị vật lý như các tập tin đặc biệt. Một tập tin đặc biệt sẽ có 1 điểm vào trong thư mục và có 1 tên tập tin. Do đó Unix cho phép người sử dụng định nghĩa tên thiết bị.

Các thiết bị được chia làm hai loại: ký tự và khối.

- Thiết bị ký tự đọc và ghi dòng các ký tự (ví dụ các thiết bị đầu cuối).
- Thiết bị khối đọc và ghi dữ liệu trong các khối có kích thước cố định (ví dụ ổ đĩa).

Thiết bị có thể đổi tên như đổi tên tập tin. Thư mục chứa các bộ điều khiển thiết bị là /dev.

**c/ Lệnh và tiện ích:**

Các lệnh và tiện ích của Unix rất đa dạng.

Một lệnh UNIX có dạng:

\$lệnh [các chọn lựa] [các đối số]

lệnh thường là chữ nhỏ. Unix phân biệt chữ lớn, nhỏ.

Ví dụ: \$ls -c /dev

Ta có thể chia lệnh thành các nhóm sau:

**\*/ Các lệnh khởi tạo:**

- exit thoát khỏi hệ thống (Bourne-Shell)
- logout thoát khỏi hệ thống C-Shell
- id chỉ danh của người sử dụng
- logname tên người sử dụng login
- man giúp đỡ
- newgrp chuyển người sử dụng sang một nhóm mới
- psswd thay đổi password của người sử dụng
- set xác định các biến môi trường
- tty đặt các thông số terminal
- uname tên của hệ thống (host)
- who cho biết những ai đang thâm nhập hệ thống

**\*/ Trình báo màn hình:**

- echo hiển thị dòng ký tự hay biến
- setcolor đặt màu nền và chữ của màn hình

**\*/ Desktop:**

- bc tính biểu thức số học
- cal máy tính cá nhân

date	hiển thị và đặt ngày
mail	gửi - nhận thư tin điện tử
mesg	cắm/ cho phép hiển thị thông báo trên màn hình (bởi write/ hello)
spell	kiểm tra lỗi chính tả
vi	soạn thảo văn bản
write/ hello	cho phép gửi dòng thông báo đến những người sử dụng trong hệ thống

\*/ Thư mục:

cd	đổi thư mục
copy	sao chép 2 thư mục
mkdir	tạo thư mục
rmdir	loại bỏ thư mục
pwd	trình bày thư mục hiện hành

\*/ Tập tin:

cat/ more	trình bày nội dung tập tin
cp	sao chép một hay nhiều tập tin
find	tìm vị trí của tập tin
grep	tìm vị trí của chuỗi ký tự trong tập tin
ls, l, lf, lc	trình bày tên và thuộc tính của các tập tin trong thư mục
mv	chuyển/ đổi tên một tập tin
sort	sắp thứ tự nội dung tập tin
wc	đếm số từ trong tập tin

\*/ Quản lý quá trình:

kill	hủy bỏ một quá trình
ps	trình bày tình trạng của các quá trình
sleep	ngưng hoạt động một thời gian

\*/ Kiểm soát chủ quyền:

chgrp	chuyển chủ quyền tập tin, thư mục từ một nhóm sang một nhóm khác
chmod	thay đổi quyền sở hữu của tập tin hay thư mục
chown	thay đổi người sở hữu tập tin hay thư mục

\*/ Kiểm soát in:

cancel	ngưng in
lp	in tài liệu ra máy in
lpstat	trạng thái của hàng chờ in

**d/ Shell:**

Là bộ xử lý lệnh của người sử dụng, nó cho phép người sử dụng tạo các lệnh rất phức tạp từ các lệnh đơn giản. Chúng ta có thể coi shell như một ngôn ngữ lập trình cấp cao. Các chức năng chính của shell là:

UNIX Shell:

- Kiểm soát I/O và đổi hướng
- Các biến môi trường
- Thực hiện lệnh
- Thư viện lệnh nội tại
- Tên tập tin mở rộng
- Ngôn ngữ lập trình và môi trường

Thực hiện lệnh: Lệnh có thể được thực hiện ở chế độ tương tác với người sử dụng hay chế độ nền (background).

Thư viện lệnh nội tại: Các lệnh nội trú trong shell.

Ngôn ngữ lập trình và môi trường: Cho phép tạo các tập tin shell-script và các cấu trúc điều khiển như Do, While, Until, If, Case.

Tên tập tin mở rộng: Cho phép biên dịch tên tập tin ở dạng ?, \*.

Các biến môi trường: Cho phép đặt các biến môi trường. Ví dụ: PATH=/USR/BIN

Kiểm soát I/O và đổi hướng: Shell định nghĩa các thiết bị xuất/ nhập chuẩn và cho phép ta đổi hướng thiết bị xuất/ nhập của các quá trình.

Hiện nay người ta sử dụng ba loại shell, tùy theo loại mà có cú pháp khác nhau:

Bourne-Shell: là shell cơ bản nhất, nhanh, hiệu quả, nhưng ít lệnh.

C-Shell: giống như Bourne-Shell nhưng cung cấp thêm các cấu trúc điều khiển, history, bí danh.

Korn-Shell : Kết hợp cả Bourne-Shell và C-Shell.

**e/ Windows và Graphic User Interface:**

Giao tiếp đồ họa và cửa sổ là một khả năng rất mạnh của hệ điều hành UNIX, nó cho phép hệ điều hành giao tiếp thân thiện hơn với người sử dụng. Hiện nay UNIX cài đặt X-WINDOW (X11) là một môi trường quản lý đồ họa lý tưởng. Trong Sun Solaris thì sử dụng với tên gọi là OpenWin.

## Chương 3: CÁC PHẦN MỀM ỨNG DỤNG THÔNG DỤNG

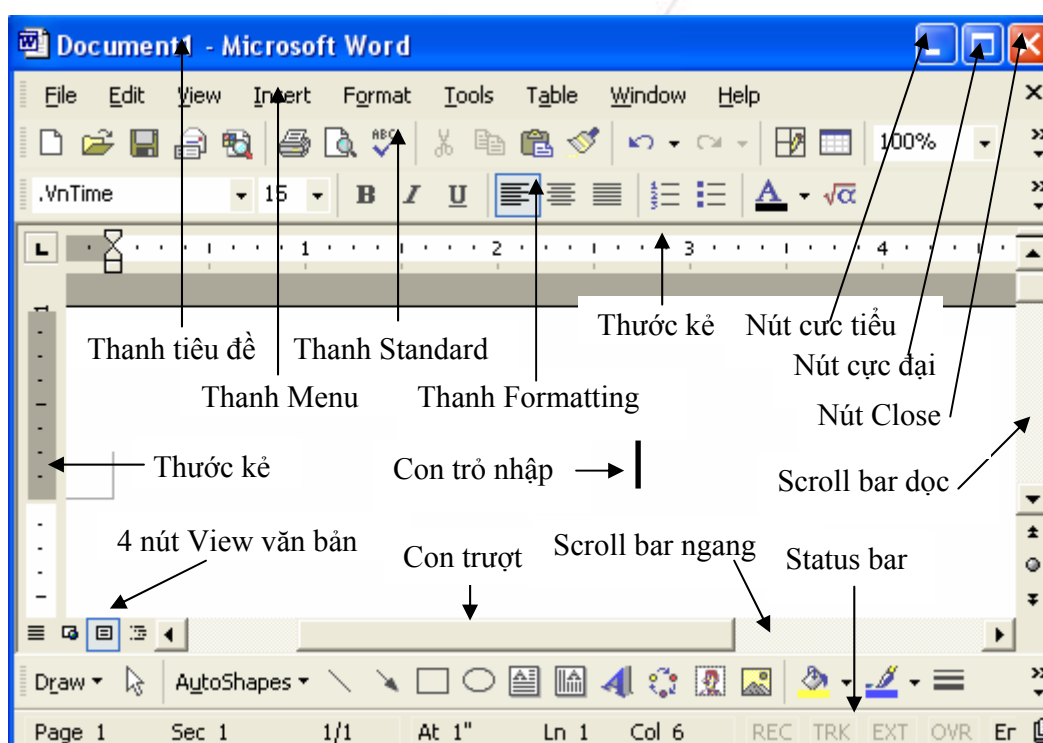
### 3.1. CHƯƠNG TRÌNH SOẠN THẢO VĂN BẢN WORD

Word 2000 là phần mềm soạn thảo văn bản cao cấp chạy dưới môi trường Windows, chuyên dùng để soạn thảo các loại văn bản, sách vở, tạp chí,... phục vụ cho công tác văn phòng. Word có các tính năng mạnh như sau: giao diện đồ họa thông qua hệ thống menu và các hộp thoại với hình thức thẩm mỹ rất cao; có khả năng giao tiếp dữ liệu với các ứng dụng khác; có các chương trình tiện ích và phụ trợ giúp tạo các văn bản dạng đặc biệt; có chương trình kiểm tra, sửa lỗi chính tả, gõ tắt, macro... giúp người sử dụng soạn thảo các văn bản khác nhau và tăng tốc độ xử lý văn bản; chức năng tạo biểu bảng mạnh và dễ dùng; soạn các công thức Toán, Lý, Hóa; dễ dàng thiết kế các trang Web.

#### 3.1.1. Các thao tác điều khiển căn bản

##### 3.1.1.1. Khởi động Word 2000 và các thành phần của màn hình

Để khởi động Word dùng lệnh Start / Programs / Microsoft Word. Màn hình chính của Word xuất hiện gồm các phần sau:



**Title bar** (thanh tiêu đề): nút ở mút trái thanh là biểu tượng của Word (nháy vào đây cho hiện menu điều khiển), tiếp theo là tên tệp văn bản và tên ứng dụng Microsoft Word, mút phải



thanh là 3 nút: Minimize, Maximize hay Restore, Close. Nháy vào nút Close sẽ làm thoát khỏi Word nếu hiện tại ta chỉ mở một tệp văn bản, nếu mở nhiều tệp văn bản thì ta chỉ đóng tệp văn bản hiện hành.

**Menu bar** (thanh menu chính): dòng chứa các lệnh của Word, mỗi mục ứng với một menu đọc. Thao tác để mở một menu đọc: nháy chuột vào tên mục hoặc gõ Alt + Ký tự đại diện của tên mục. Nếu hiện tại ta chỉ mở một tệp văn bản thì cuối thanh này có nút Close, nháy vào nút này thì ta đóng tệp văn bản duy nhất hiện tại mà không ra khỏi Word.

**Standard toolbar** (thanh công cụ chuẩn): chứa một số nút và ô điều khiển thể hiện một số lệnh thông dụng. Thay vì phải vào các menu để chọn lệnh, chỉ cần nháy chuột lên nút lệnh tương ứng. Ví dụ: muốn lưu văn bản lên đĩa thay vì vào menu File chọn Save, chỉ cần nháy chuột lên nút Save. Muốn biết tên của một nút hay một ô điều khiển nào đó trên các thanh công cụ ta chỉ cần rê mũi tên chuột lên nút hay ô đó và chờ vài giây (nếu chưa có tính chất này thì ta dùng lệnh Tools / Customize / Options, chọn mục Show ScreenTips on Taskbars).

Thanh Standard có các nút thường dùng: New, Open, Save, Print, Print Preview, Cut, Copy, Paste, Format Painter, Undo, Can't Redo, Tables and Borders, Insert Table, Columns, Drawing, ô điều khiển Zoom...

**Formatting Bar** (thanh định dạng): chứa các hộp và biểu tượng dành cho việc định dạng văn bản như kiểu, loại font, căn lề... Thanh Formatting có các hộp điều khiển: Style, Font, Font Size và các nút: Bold, Italic, Underline, Align Left, Center, Align Right, Justify, Numbering, Bullets, Decrease Indent, Increase Indent, Border, Font Color...

**Rulers** (thước): một thước theo chiều ngang ở phía trên của vùng văn bản, một thước theo chiều dọc ở bên trái màn hình. Để thay đổi đơn vị đo (Inches / Cm / Points / Picas) trên Ruler, ta thực hiện Tools / Options / General rồi vào hộp Measurements để thay đổi. Nên chọn đơn vị là Cm.

**Text Area** (vùng văn bản): đây là vùng lớn nhất dùng để nhập văn bản.

**Hai thanh trượt** (Scroll Bars): thanh trượt dọc bên phải màn hình (di chuyển theo chiều dọc toàn bộ văn bản), thanh trượt ngang phía dưới màn hình (di chuyển theo chiều ngang của văn bản). Mút trái thanh trượt ngang có 4 nút Normal View, Web Layout View, Page Layout View, Outline View dùng để điều khiển chế độ hiển thị văn bản.

**Status bar** (thanh trạng thái, nằm ở dòng cuối của cửa sổ Word) hiển thị tình trạng của cửa sổ hiện tại: số trang của trang hiện tại, số thứ tự của Section hiện tại, trang hiện tại là trang thứ mấy trong tổng số bao nhiêu trang, vị trí con trỏ (khoảng cách so với mép trên trang giấy, vị trí dòng, cột), bàn phím ở chế độ chèn hay đè.

#### **Các loại con trỏ trên màn hình:**

- Con trỏ nhập văn bản là dấu vạch đứng |. Khi gõ một ký tự trên bàn phím, ký tự sẽ được chèn vào chỗ con trỏ nhập.

- Con trỏ chuột trong vùng văn bản có dạng dấu vạch đứng với hai vạch ngang ở hai đầu, con trỏ chuột ở menu có dạng mũi tên rỗng. Muốn chuyển con trỏ nhập tới một vị trí mới trong văn bản để nhập, ta đưa con trỏ chuột đến vị trí mới và nháy nút trái chuột.

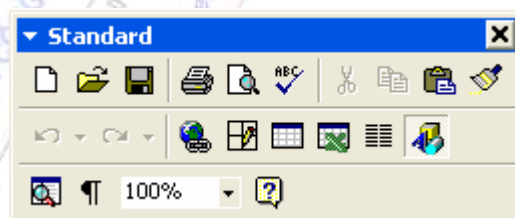
### 3.1.1.2. Sử dụng các thanh công cụ

Word có sẵn 16 thanh công cụ chuẩn, ta có thể tự làm thêm các thanh công cụ mới. Thông thường ta chỉ cần cho hiện hai thanh công cụ: thanh Standard và thanh Formatting. Để cho hiện (không cho hiện) một thanh công cụ trên màn hình ta dùng lệnh View / Toolbars, xuất hiện menu dọc thứ hai, hãy đánh dấu (bỏ đánh dấu) trước tên của thanh công cụ.

Trên một thanh công cụ có **các nút** để thực hiện một lệnh, để sử dụng một **hộp menu kéo xuống** ta chỉ cần nhấp vào hộp nhập dữ liệu mới hoặc nhấp vào mũi tên bên phải hộp và lựa chọn theo menu.

Thông thường không đủ chỗ để hiển thị tất cả các nút trên thanh công cụ, ví dụ thanh Standard. Để hiển thị nhiều nút hơn, nhấp vào nút More Buttons ở cuối thanh Standard, xuất hiện một thanh các nút khác là phần mở rộng của thanh Standard, ta có thể nhấp vào các nút này để sử dụng, trong thanh mở rộng này có nút Add or Remove Buttons, nhấp vào nút này sẽ xuất hiện một menu dọc cho phép ta lựa chọn các nút nào được hiện trên thanh Standard.

Mút trái của mỗi thanh công cụ hay thanh menu có một dấu vạch đứng gọi là tay cầm (Move handle). Nhấp vào tay cầm và kéo sang phải hay sang trái để thay đổi kích thước thanh công cụ. Để di chuyển một thanh công cụ gắn chặt vào phía trên hay phía dưới một thanh thứ hai khác, ta kéo tay cầm lên



phía trên hay phía dưới thanh thứ hai này. Ta cũng có thể để hai thanh công cụ trên cùng một dòng màn hình, mỗi thanh chiếm một nửa dòng. Nếu kéo tay cầm của thanh công cụ vào cửa sổ văn bản thì thanh công cụ chuyển thành **Thanh công cụ trôi nổi** như hình bên:

Di chuyển thanh công cụ trôi nổi trong vùng văn bản: kéo thanh tiêu đề. Thay đổi kích thước thanh trôi nổi: kéo cạnh dưới hoặc cạnh phải. Gắn thanh công cụ trôi nổi vào trên hay dưới một thanh công cụ cố định khác ở phía trên hay phía dưới cửa sổ Word: kéo thanh tiêu đề của thanh công cụ trôi nổi. Đóng một thanh công cụ trôi nổi: nhấp nút Close.

Các thanh công cụ trong Word 2000 gọi là các **thanh công cụ thích nghi**: nếu ta thường xuyên sử dụng một nút không có trên thanh công cụ thì Word sẽ tự động thêm nó vào thanh công cụ, nếu có những nút trên thanh công cụ mà ta không bao giờ sử dụng đến thì Word sẽ chuyển nó vào kho More Buttons.

### 3.1.1.3. Sử dụng các menu

Thanh menu chính nằm dưới thanh tiêu đề chứa các lệnh làm việc với Word, có thể kéo tay cầm của Thanh menu để thay đổi kích thước thanh, di chuyển thanh tới các vị trí khác hay chuyển nó thành Thanh trôi nổi.

Khi nhấp vào một mục của Thanh menu chính sẽ xuất hiện một menu dọc. Các menu dọc ứng với một mục là các **menu thích nghi**: khi mở một menu dọc ta sẽ thấy danh sách các lệnh thường xuyên dùng tới và mũi tên hai đầu ở dưới (gọi là menu ngắn), nếu nhấp vào mũi tên hai đầu (hoặc để nguyên menu trong vòng vài giây) thì menu sẽ tự mở rộng để chứa tất cả các lệnh, các lệnh mới này xuất hiện với nền nhạt hơn để phân biệt. Khi ta chọn một lệnh mà nó không có trên menu ngắn thì Word sẽ thêm nó vào menu ngắn để ta không cần phải mở rộng menu trong lần dùng tới.

Trong menu con dọc có một số mục kèm theo dấu mũi tên ở bên phải, nháy vào mục này sẽ xuất hiện một menu con khác để lựa chọn. Có một số mục ghi kèm tổ hợp phím, ví dụ mục **Copy Ctrl+C** có nghĩa: gõ phím Ctrl+C để chép khối vào Clipboard mà không cần vào menu Edit và chọn Copy. Có một số mục của menu có dấu ba chấm (...), khi chọn mục này sẽ xuất hiện một hộp thoại cung cấp cho ta nhiều lựa chọn. Có một số mục bị mờ: không thể chọn được mục này trong tình huống hiện tại.

Ngoài việc hiển thị menu ở phía trên màn hình ta còn có thể hiển thị **menu tắt** khi đang làm việc bằng cách nháy nút phải chuột, khi menu tắt đã mở ta có thể nháy vào một lệnh để chọn. Các lệnh xuất hiện trên menu tắt phụ thuộc vào nơi ta nháy nút phải chuột và tình huống.

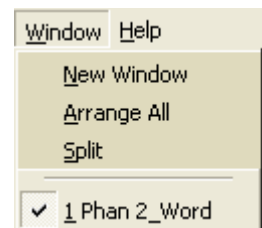
#### 3.1.1.4. Tạo một tài liệu mới, mở một tài liệu sẵn có trên đĩa

Word cho phép làm việc đồng thời với nhiều văn bản trên các cửa sổ khác nhau. Khi vừa khởi động, Word mở một cửa sổ với tên mặc định là Document1. Mỗi lần mở một tài liệu mới hay một tài liệu đã có trên đĩa Word sẽ dành một cửa sổ để hiển thị tài liệu.

- **Để mở một tài liệu mới** dùng lệnh File / New, hiện hộp thoại New, chọn lớp General, chọn Document của mục Create New, chọn Blank Document, chọn OK. Hoặc nháy nút New Blank Document trên thanh công cụ Standard.

- **Để mở một tài liệu đã có trên đĩa** dùng lệnh File / Open (hoặc nháy vào nút Open trên thanh Standard), hiện hộp thoại Open, chọn thư mục chứa tệp trong hộp Look In (nháy vào phím mũi tên bên phải hộp để hiện cây thư mục), chọn tên tệp ở phần giữa của cửa sổ Open (hoặc gõ trực tiếp tên tệp vào hộp File Name) và nháy nút Open.

Tại một thời điểm chỉ có một cửa sổ được kích hoạt, muốn làm việc với một cửa sổ khác đã được mở: nháy mục Window để hiện menu dọc Window, chọn số thứ tự tương ứng với cửa sổ cần kích hoạt. Để tiết kiệm bộ nhớ, nên đóng các cửa sổ không cần thiết, chỉ giữ lại các cửa sổ cần trao đổi dữ liệu với nhau. Muốn đóng một cửa sổ thì nó phải đang được kích hoạt, vào mục menu File, chọn Close.

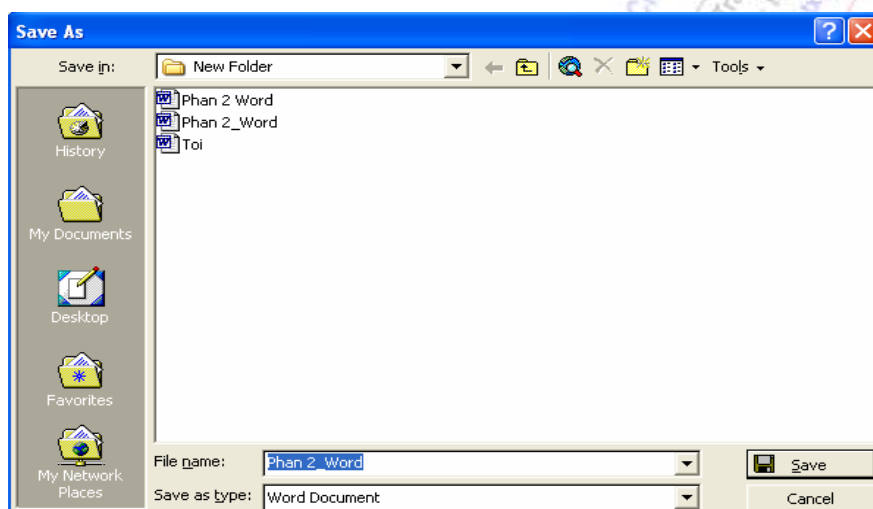


**Chia đôi cửa sổ soạn thảo:** Màn hình soạn thảo một tệp của Word có thể chia làm hai nửa theo chiều ngang, mỗi nửa có một thanh trượt dọc giúp cho việc xem lướt văn bản. Chức năng này giúp cho việc sao chép, di chuyển trong nội bộ một tệp văn bản được dễ dàng, ta có thể đồng thời nhìn thấy hai phần văn bản ở xa nhau. Thực hiện, dùng lệnh Window / Split, trên màn hình xuất hiện đường nằm ngang, kéo chuột để di chuyển đường nằm ngang đến vị trí định chia đôi màn hình, nháy nút trái chuột, màn hình chia làm hai nửa. Hủy chế độ này: dùng lệnh Window / Remove Split.

**Mở một tài liệu trên hai cửa sổ** giúp ta hiển thị cùng một tài liệu trên hai cách: một cửa sổ hiện dạng bố cục và một cửa sổ hiện dạng đầy đủ, hoặc một cửa sổ hiện dạng Print Layout và một cửa sổ hiện dạng Normal. Để mở cửa sổ thứ hai của tài liệu hiện hành dùng lệnh Window / New Window, hiện cửa sổ thứ hai, trên thanh tiêu đề sau tên tệp thêm dấu hai chấm và một con số (chỉ tệp mở ở cửa sổ nào).

### 3.1.1.5. Lưu trữ văn bản

**Ghi văn bản đã soạn lên đĩa:** dùng lệnh File / Save hoặc nhấp vào nút Save trên thanh Standard. Lần đầu ghi văn bản, hộp thoại Save As xuất hiện, yêu cầu người sử dụng nhập vào: tên tệp (hộp File Name), lựa chọn thư mục để chứa tệp (hộp Save in), kiểu loại tệp (nhấp vào hộp Save as Type, chọn Word Document), cuối cùng chọn OK để lưu. Kể từ đó, mỗi khi thực hiện thao tác lưu trữ tệp thì văn bản sẽ được lưu vào tên tệp đã đặt theo dạng của lần ghi đầu tiên. Lưu ý, muốn ghi tệp văn bản để Word 6.0 cũng đọc được thì trong hộp Save as Type ta phải chọn **Word 6.0/95** (khi đó tệp sẽ lớn hơn mức bình thường).



**Đặt mật khẩu cho tài liệu:** khi ghi tệp vào đĩa trong hộp thoại Save As nhấp vào nút Tools, hiện menu dọc, chọn General Options, hiện hộp thoại Save, nhập vào mật khẩu để mở tài liệu trong hộp Password to Open, nhập mật khẩu cho phép sửa tài liệu trong hộp Password to Modify (có thể bỏ qua).

**Ghi vào đĩa với một tên khác:** vào menu File chọn Save As, sau đó các thao tác tiến hành tương tự như lần ghi tệp đầu tiên. Khi đó tên cửa sổ soạn thảo (trên Thanh tiêu đề) sẽ là tên tệp mới.

**Ghi tệp vào đĩa và tạo bản dự phòng:** dùng lệnh Tools / Options, hiện hộp thoại Options, chọn lớp View, đánh dấu kiểm vào mục Always create backup copy. Mỗi lần mở một tệp cũ, sửa chữa, khi ghi lại Word chuyển nội dung tệp cũ thành bản Backup, nội dung mới ghi vào tệp mang tên cũ. Lựa chọn này được Word lưu lại cho các phiên làm việc sau.

### 3.1.1.6. Kết thúc làm việc với Word 2000 và trở về Windows

Muốn kết thúc làm việc với Word ta có thể dùng các cách: dùng lệnh File / Exit, hoặc nhấp vào biểu tượng của Word ở nút trái thanh tiêu đề và chọn Close, hoặc nhấp vào nút Close ở nút phải thanh tiêu đề. Nếu có thực hiện bất kỳ sửa đổi nào trong các văn bản đang mở mà chưa lưu vào đĩa (cho đến thời điểm thoát) thì Word sẽ hỏi lại "Do you want to save the changes you made to tên-tệp?", trả lời YES nếu muốn lưu, NO nếu không muốn lưu.

### 3.1.2. Các thao tác soạn thảo cơ bản

#### 3.1.2.1. Soạn thảo văn bản bằng tiếng Việt:

Trong Windows 2000 dùng bộ mã Unicode version 2 (mã hai byte) và có nhiều font cho phép hiển thị tiếng Việt có dấu như Times New Roman, Arial, Tahoma, Courier New. Muốn tạo văn bản tiếng Việt ta cần một chương trình bàn phím tiếng Việt để gõ tiếng Việt, có thể dùng VietKey. Các chương trình bàn phím tiếng Việt thường cho phép gõ tiếng Việt theo kiểu Telex như sau:

aa - â	ow [ - ơ	DD	- Đ
aw - ã	uw ] - ur	UW W }	- U
ee - ê	oo - ô	OW {	- O

Dấu gõ ở cuối từ: s - dấu sắc, f - huyền, r - hỏi, x - ngã, j - nặng, z - xóa dấu đã đánh.

#### 3.1.2.2. Các phím thường dùng khi soạn thảo

**Các phím di chuyển con trỏ:** Bốn phím mũi tên để di chuyển con trỏ sang phải hay sang trái, lên hay xuống một dòng. Home: di chuyển con trỏ về đầu dòng văn bản. End: di chuyển con trỏ về cuối dòng văn bản. PgUp: di chuyển con trỏ lên một trang màn hình. PgDn: di chuyển con trỏ xuống một trang màn hình. Clrt+Home: di chuyển con trỏ về đầu văn bản. Clrt+Home: di chuyển con trỏ về cuối văn bản. Ctrl+PgUp và Ctrl+PgDn: chuyển con trỏ lên đầu của trang trước hay đầu của trang sau. Shift+F5: tới nơi làm việc trong lần điều chỉnh con trỏ gần đây nhất.

**Các phím xóa ký tự:** Delete (xóa ký tự bên phải con trỏ), Backspace (xóa ký tự bên trái con trỏ).

**Phím Insert:** Dùng để chuyển đổi giữa chế độ chèn ký tự (Insert) và đè ký tự (Overwrite). Trong chế độ chèn, khi nhập vào một ký tự nó sẽ chèn vào tại vị trí hiện thời của con trỏ nhập và đẩy ký tự hiện thời sang bên phải. Trong chế độ đè, khi nhập vào một ký tự nó sẽ thay thế ký tự hiện thời.

**Phím Alt:** Dùng để chọn các thực đơn dọc bằng bàn phím, ví dụ Alt+E để chọn menu Edit.

**Phím Esc:** Dùng để ngắt một công việc đang thực hiện.

**Phím Enter:** Mỗi khi ấn Enter con trỏ nhập sẽ xuống dòng và tạo ra một đoạn (Paragraph) mới, đoạn mới sẽ theo phong chữ và các định dạng của đoạn trước. Việc xuống dòng là do trình soạn thảo quyết định, tùy thuộc vào lề đang định hiện thời. Muốn cố ý cho con trỏ nhập xuống dòng mà không cần tạo một đoạn mới thì ta ấn Shift+Enter. **Paragraph** là một khái niệm rất quan trọng trong Word. Muốn hiện các dấu Enter hoặc không hiện, ta nháy vào nút Show / Hide ¶ trên thanh Standard.

#### 3.1.2.3. Thao tác trên văn bản và khối văn bản

- Lựa chọn một khối văn bản

Khối là một phần văn bản được lựa chọn (trên màn hình nền của văn bản thuộc khối có màu đen). Sau khi xác định khối ta có thể di chuyển khối, sao chép khối, định dạng lại văn bản trong khối, đóng khung... Để lựa chọn một khối ta có thể dùng chuột và bàn phím.

### ***Dùng chuột***

- Nháy chuột vào vị trí đầu của khối, kéo chuột đến vị trí cuối khối.
- Nháy chuột vào vị trí đầu của khối, giữ phím Shift và đưa con trỏ chuột đến vị trí cuối rồi nháy nút trái chuột.

- Chọn một từ: nháy đúp vào từ.
- Chọn một dòng: nháy chuột vào khoảng trống bên trái dòng.
- Chọn một số dòng: rê chuột ở khoảng trống bên trái các dòng.
- Chọn một Paragraph: nháy đúp vào khoảng trống bên trái Paragraph.
- Chọn một câu: giữ phím Ctrl và nháy vào câu đó.
- Chọn toàn bộ văn bản: giữ phím Ctrl và nháy chuột vào khoảng trống bên trái văn bản.

### ***Dùng bàn phím***

- Đưa con trỏ nhập vào đầu khối, giữ phím Shift, bấm các phím ← ↑ → ↓, PgUp, PgDn, Home, End để đến vị trí cuối khối.

- Chọn toàn bộ văn bản: gõ Ctrl+A.
- Chọn từ vị trí con trỏ đến cuối văn bản: gõ Shift+Ctrl+End.
- Chọn từ vị trí con trỏ đến đầu văn bản: gõ Shift+Ctrl+Home.
- Chọn từ vị trí con trỏ đến đầu dòng: gõ Shift+Home. Chọn từ vị trí con trỏ đến cuối dòng: gõ Shift+End.
- Nhấn phím F8, trên dòng trạng thái chữ EXT chuyển sang màu đen. Nhấn F8 lần thứ hai: lựa chọn một từ chứa con trỏ chèn. Nhấn F8 lần thứ ba: lựa chọn một câu chứa con trỏ chèn. Nhấn F8 lần thứ tư: lựa chọn một đoạn... Nhấn ESC khi lựa chọn xong khối.

- Lựa chọn khối là vùng chữ nhật trên màn hình: ấn Ctrl+Shift+F8 và thả, dùng các phím mũi tên để xác định khối, sau đó có thể dùng lệnh Edit / Copy. Sau khi xác định khối nếu ấn ESC thì thôi không đánh dấu khối.

### ***Hai thao tác đơn giản sau khi chọn khối***

- Hủy đánh dấu khối: nháy chuột vào một điểm ngoài khối.
- Xóa khối: ấn phím Delete.
  - Sao chép và di chuyển khối

### ***+ Sao chép một khối***

- Sao chép khối bằng menu: chọn khối muốn chép, dùng lệnh Edit / Copy (hoặc gõ Ctrl+C) để chép khối vào vùng đệm (Clipboard), di chuyển con trỏ đến vị trí mới, thực hiện lệnh Edit / Paste (hoặc gõ Ctrl+V) để đánh dấu khối từ vùng đệm vào chỗ con trỏ.

- Sao chép khối bằng thanh Standard: chọn khối muốn chép, nháy chuột vào nút Copy của thanh Standard, di chuyển con trỏ đến vị trí mới, nháy chuột vào nút Paste của thanh Standard.

**+ Di chuyển một khối**

▪ Di chuyển khối bằng menu: chọn khối muốn chép, thực hiện lệnh Edit / Cut (hoặc gõ Ctrl+X) để xóa khối và chép nó vào vùng đệm, di chuyển con trỏ đến vị trí mới, thực hiện lệnh Edit / Paste để dán khối từ vùng đệm vào chỗ con trỏ.

▪ Di chuyển khối bằng thanh Standard: chọn khối muốn chép, nhấp chuột vào nút Cut trên thanh Standard, di chuyển con trỏ đến vị trí mới, nhấp chuột vào nút Paste trên thanh Standard.

**Clipboard (vùng đệm):** Khi cắt, sao chép hay dán văn bản, Word sử dụng một vùng nhớ đặc biệt, có tên là Clipboard chứa tới 12 phần tử và một thanh công cụ có dạng như hình bên.



Có thể tắt mở thanh công cụ này bằng lệnh View / Toolbars / Clipboard. Các thao tác trên thanh Clipboard:

- Đưa một đối tượng đã chọn (văn bản, đồ họa) vào Clipboard: nhấp nút Copy.
- Xem nội dung một phần tử: trỏ chuột vào nút, chờ vài giây, hiện thông báo.
- Dán nội dung một phần tử vào chỗ con trỏ trên văn bản: nhấp vào nút phần tử.
- Dán tất cả các phần tử trên Clipboard vào chỗ con trỏ: nhấp nút Paste All Contents.
- Xóa sạch Clipboard: nhấp nút Clear Clipboard.
- Xóa thanh công cụ Clipboard: nhấp nút Close trên thanh tiêu đề của Clipboard.

**3.1.2.4. Các thao tác định dạng văn bản**

Trong Windows loại chữ được dùng có bề rộng ký tự không cố định, do đó khi soạn thảo không được dùng dấu khoảng cách để chỉnh dòng, chỉnh đoạn. Không nên để hai dấu cách hoặc hơn liền nhau. Đầu dòng thứ nhất của một đoạn không được gõ các dấu khoảng cách mà dùng chức năng định dạng đoạn để lùi đầu dòng.

Word là một bộ xử lý từ có định dạng, tức là sau khi nhập nội dung ta có thể quy định hình thức trình bày cho các nội dung đó tùy ý. Hai yếu tố cơ bản để định dạng cho văn bản là định dạng ký tự (quy định hình dạng từng ký tự trong văn bản). Về cơ bản thuộc tính của ký tự bao gồm: font chữ, kiểu chữ (thường, đậm, nghiêng, gạch chân, chỉ số trên, chỉ số dưới), cỡ chữ tính theo chiều cao của ký tự và màu chữ. Thuộc tính của đoạn bao gồm: lề trái và lề phải của đoạn, khoảng cách lùi vào của dòng đầu tiên trong một đoạn, cách đóng đoạn (đóng trái, đóng phải, đóng giữa và đóng đều hai bên) và khoảng cách các dòng trong đoạn.

Để định dạng cho các ký tự hay nhiều đoạn, trước tiên ta cần chọn các ký tự hay các đoạn. Trong trường hợp định dạng cho một đoạn thì chỉ cần đưa con trỏ nhập vào bên trong phạm vi của đoạn đó.

**a. Định dạng ký tự**

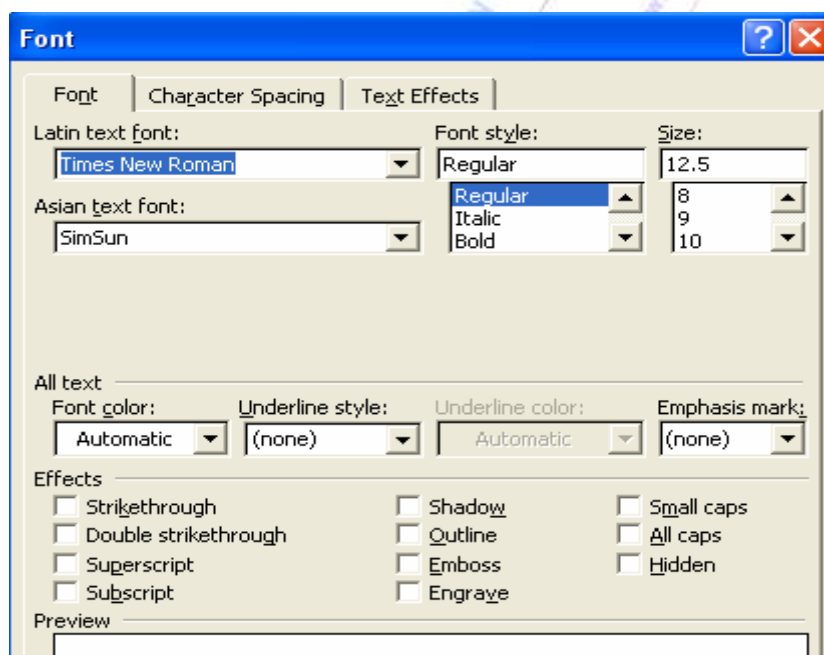
**- Định dạng bằng hộp thoại Font**

Đối với một đoạn văn bản, ta có thể định dạng lại nó theo font mới, cỡ mới, kiểu mới. Trước tiên hãy chọn khối văn bản cần định dạng lại, dùng lệnh Format / Font, xuất hiện hộp thoại

Font. Hộp thoại Font có 3 lớp.

**Lớp Font** có các lựa chọn sau:

- Font: chọn phông chữ.
- Size: chọn cỡ chữ.
- Font Style: chọn kiểu bình thường, chữ đậm, nghiêng, vừa đậm vừa nghiêng.
- Underline Style: chọn các kiểu gạch chân.
- Font Color: chọn màu cho chữ.
- Effects: chọn một số hiệu ứng đặc biệt cho font chữ. Chọn Superscript: chỉ số trên, chọn Subscript: chỉ số dưới.
- Preview: khung mẫu để quan sát sự lựa chọn.
- Nút Default: các thông số định dạng sẽ được lấy làm trị mặc nhiên cho mỗi lần khởi động Word hoặc mở một tài liệu mới.



**Lớp Character Spacing** có các lựa chọn sau:

- Scale: định tỷ lệ co giãn bề ngang của ký tự.
- Spacing dùng để chọn khoảng cách giữa các ký tự: Normal (bình thường), Expanded (các ký tự cách xa nhau thêm một khoảng), Condensed (các ký tự xít lại với nhau). Mục By bên cạnh quy định khoảng cách giãn ra hoặc xít lại của các ký tự.
- Position dùng để nâng lên hay hạ xuống ký tự so với bình thường: Normal (bình thường), Raised (nâng ký tự lên), Lowered (hạ ký tự xuống). Mục By bên cạnh quy định số điểm nâng lên hay hạ xuống.

**Lớp Text Effects** quy định tính động của ký tự, chỉ có tác dụng trên màn hình, không có tác



dụng khi in. Trong mục Text Effects có thể chọn các tính chất:

- None: chữ bình thường.
- Blinking background: nền chữ nhấp nháy
- Las Vegas Lights: chữ có những ngôi sao màu xung quanh
- Marching Black Ants: chữ có những đường gạch đen chạy xung quanh
- Shimer: chữ nhòe nét
- Sparkle Text: chữ có các ngôi sao lấp lánh.

#### - Định dạng nhanh bằng thanh công cụ

Ta có thể dùng thanh công cụ Formatting để định dạng nhanh.

**Định dạng ký tự:** chọn các ký tự cần định dạng, nhấp vào hộp điều khiển Font để chọn phông chữ, nhấp vào hộp điều khiển Font Size để chọn cỡ chữ, nhấp nút Bold chọn chữ đậm, nút Italic chọn chữ nghiêng, nút Underline chọn chữ gạch chân.

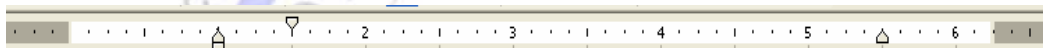


**Đóng lề trái và lề phải của đoạn:** chọn đoạn hay nhiều đoạn cần đóng, nhấp vào nút Align Left để đóng theo lề trái, nút Center để cân vào giữa, nút Align Right để đóng theo lề phải, nút Justify để đóng đều cả lề trái và lề phải.

#### Các phím gõ tắt để định dạng ký tự và đóng:

- |                             |                             |
|-----------------------------|-----------------------------|
| Ctrl+B: chữ đậm             | Ctrl+L: đóng theo lề trái   |
| Ctrl+I: chữ nghiêng         | Ctrl+E: canh giữa           |
| Ctrl+U: gạch chân           | Ctrl+R: đóng theo lề phải   |
| Ctrl+]: giảm cỡ Font đi 1   | Ctrl+J: đóng đều hai bên lề |
| Ctrl+[ : tăng cỡ Font lên 1 | Ctrl+Shift+=: chỉ số trên   |
| Ctrl+=: chỉ số dưới         |                             |

**Định lề cho đoạn bằng thanh Ruler:** chọn đoạn hay các đoạn cần định lề, kéo các nút định lề tương ứng để điều chỉnh.



#### Định dạng nhanh bằng chổi sơn

Ta có thể sao chép định dạng của một đoạn đã có sẵn (đoạn 1) sang một đoạn khác cần định dạng (đoạn 2) bằng cách: đặt con trỏ chèn vào đoạn 1, nhấp vào nút Format Painter trên thanh công cụ Standard, khi đó chổi sơn sẽ dính vào con trỏ chuột, di con trỏ chuột tới vị trí bất kỳ trong đoạn 2 và nhấp, hoặc dùng chuột kéo để xác định phạm vi văn bản cần định dạng (có thể gồm nhiều đoạn). Định dạng được sao chép gồm có: kiểu font, cỡ font, lề trái, lề phải, cách đóng, khoảng cách lùi vào của dòng đầu đoạn, khoảng trống trước đoạn và sau đoạn...

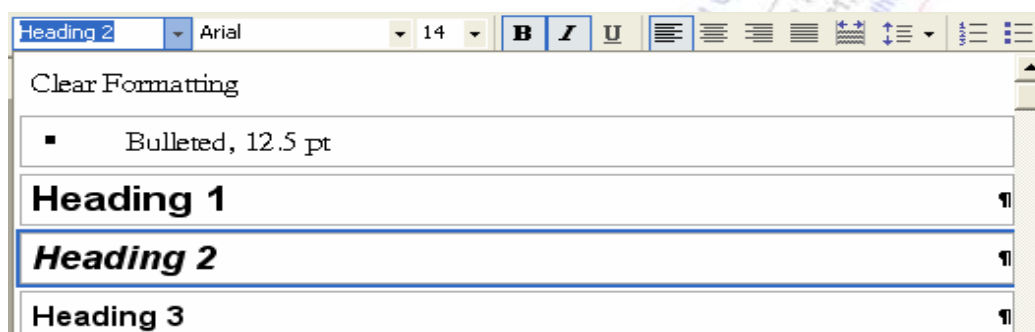
Dùng chổi sơn bằng bàn phím: để con trỏ vào đầu đoạn cần sao chép định dạng, ấn

Ctrl+Shift+C (sao định dạng), dán định dạng vào nơi khác: Ctrl+Shift+V.

Sao chép định dạng đi nhiều nơi: để con trỏ vào đoạn cần sao chép định dạng, nhấp đúp vào nút Format Painter trên thanh Standard, kéo chuột lần lượt lên các vùng muốn dán định dạng, làm xong nhấp lại vào nút Format Painter.

### **Định dạng bằng các Style có sẵn**

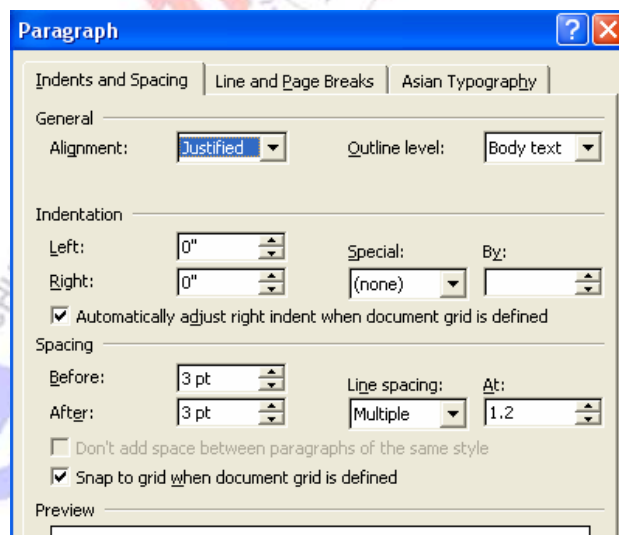
Style là một nhóm các định dạng có tên và ta có thể áp dụng cho văn bản cùng một lúc. Một tài liệu thường có một nhóm các Style đã được định nghĩa sẵn (một số do Word làm sẵn, một số ta tự làm). Để hiện danh sách Style có sẵn ta nhấp vào nút mũi tên ở bên phải hộp điều khiển Style trên thanh Formatting:



Ví dụ, Style có tên là **Heading 2** bao gồm nhóm các định dạng đoạn: phông chữ Arial, cỡ 14, in đậm và nghiêng, dóng lề trái.

Thao tác định dạng theo Style có sẵn: chọn văn bản, nhấp nút mũi tên của hộp Style trên thanh Formatting, chọn Style có sẵn.

### **b. Định dạng Paragraph**



Chức năng này cho phép thay đổi cách hiển thị của một đoạn văn bản. Chọn đoạn văn bản, dùng lệnh Format/Paragraph, xuất hiện hộp thoại sau gồm hai lớp: Indents and Spacing, Line and Page Breaks.

- **Lớp Indents and Spacing** dùng để hiệu chỉnh dóng và khoảng cách dòng trong đoạn. Các

mục trong lớp:

- **Alignment:** chọn cách dóng hai lề.

- **Indentation** dùng chỉnh lề cho đoạn. Left: chọn lề trái của đoạn so với lề trang. Right: chọn lề phải của đoạn. Mục Special dùng để chỉnh lề đặc biệt: None (bình thường), First Line (dòng đầu thụt vào), Hanging (cả đoạn thụt vào), mục By quy định khoảng thụt vào.

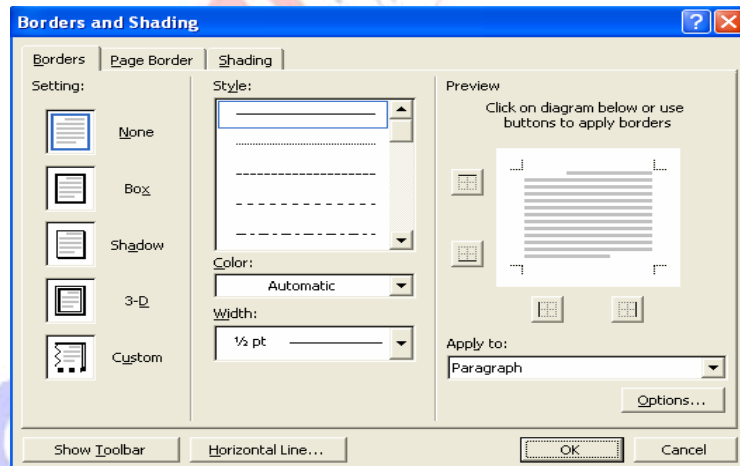
- **Spacing** quy định khoảng cách dòng: Before: khoảng trống phía trên mỗi đoạn. After: khoảng trống phía sau mỗi đoạn. Line Spacing quy định khoảng cách các dòng trong đoạn: Single (cách dòng đơn), Double (cách dòng đôi), 1.5 lines (cách 1 dòng rưỡi), At Least (cách dòng nhỏ nhất), Exactly (cách dòng chính xác theo giá trị của con số do người dùng nhập, đơn vị là **pt**), Multiple (cách dòng theo bội số của nửa dòng).

- **Lớp Line and Page Breaks** quy định cách ngắt dòng và ngắt trang trong đoạn. Ô kiểm tra **Widow/Orphan Control:** tránh dòng đầu tiên của đoạn ở cuối trang, tránh dòng cuối một đoạn ở đầu trang. Ô kiểm tra **Keep lines together:** tránh ngắt trang ở giữa đoạn. **Keep with Next:** tránh ngắt trang ở giữa đoạn đang chọn và đoạn kế tiếp. **Page Break before:** chèn dấu ngắt trang vào đầu đoạn đang chọn. **Supprress Line Number:** không in con số chỉ dòng trong đoạn (lựa chọn này áp dụng chỉ khi đánh dấu tự động dòng được định). **Don't Hefphenate:** không dùng dấu nối khi ngắt dòng trong một đoạn (lựa chọn này áp dụng chỉ khi việc dùng dấu nối được chọn)

### c. Tạo khung và làm nền

#### Đóng khung cho các đoạn văn bản

Để tạo khung cho đoạn văn bản hay một nhóm ký tự, trước tiên chọn văn bản cần tạo khung, sau đó dùng lệnh Format / Borders and Shading, hộp thoại Borders and Shading xuất hiện, chọn lớp Borders.



Các mục lựa chọn trong hộp thoại:

- **Style:** chọn kiểu đường để kẻ khung.
- **Color:** chọn màu kẻ đường.
- **Width:** chọn độ rộng đường kẻ.

▪ **Settings:** chọn các mẫu kẻ khung. None: không kẻ khung. Box: kẻ khung 4 cạnh xung quanh đoạn. Shadow: khung có bóng. Mẫu 3-D: khung nổi kiểu 3 chiều. Custom: tự tạo kiểu khung, nháy vào các nút trong mục Preview để lựa chọn cạnh cần kẻ (nháy nút lần thứ hai để xóa cạnh đã kẻ).

▪ **Nút Options:** quy định khoảng cách của văn bản với các đường kẻ khung Top, Bottom, Left, Right.

#### d. Tạo màu nền cho văn bản

Để tạo màu nền cho văn bản hay một nhóm ký tự, trước tiên chọn văn bản cần tạo nền, sau đó dùng lệnh Format / Borders and Shading, hộp thoại Border and Shading xuất hiện, chọn lớp Shading.

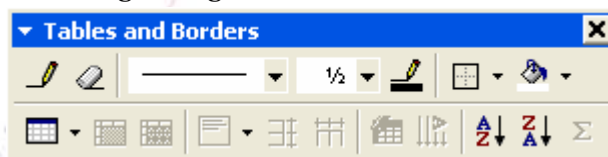
▪ Chọn màu nền của văn bản: chọn màu trong mục Fill (No Fill là không có màu nền), chọn mục More Colors để hiện bảng 120 màu hoặc tự thiết kế màu.

▪ Chọn mẫu tô cho nền: nháy vào nút mũi tên trong mục Style và chọn mẫu tô, có nhiều mẫu tô có đường kẻ. Mục Color dùng để chọn màu của đường kẻ trong mẫu tô.

e. **Tạo khung cho các trang văn bản:** dùng lệnh Format / Borders and Shading, chọn lớp Page Border. Các mục của lớp này giống với các mục của lớp Borders, song thêm vào mục mới **Art** để chọn các mẫu đóng khung nghệ thuật.

#### - Dùng thanh công cụ Tables and Borders để đóng khung văn bản

Dùng lệnh View / Toolbar / Tables and Borders để xuất hiện thanh công cụ Tables and Borders, thanh này làm nhiệm vụ như hộp thoại ở trên.



#### f. Định khoảng cách Tab Stop

Mỗi lần gõ phím Tab, con trỏ sẽ dừng lại tại một vị trí. Khoảng cách từ vị trí dừng của Tab này đến vị trí dừng của Tab khác gọi là chiều dài của Tab Stop. Khoảng cách ngầm định là 0,5 inch. Việc dùng các Tab Stop rất tiện lợi khi ta cần gõ một văn bản có đóng theo các cột, ví dụ bảng danh sách các học sinh trong lớp gồm Số thứ tự, Họ và tên, Ngày sinh, Địa chỉ, Điện thoại. Ký tự đầu tiên của một cột ứng với một Tab Stop. Khi nhập văn bản cho một cột xong, ta chỉ cần ấn phím Tab con trỏ sẽ sang cột tiếp theo.

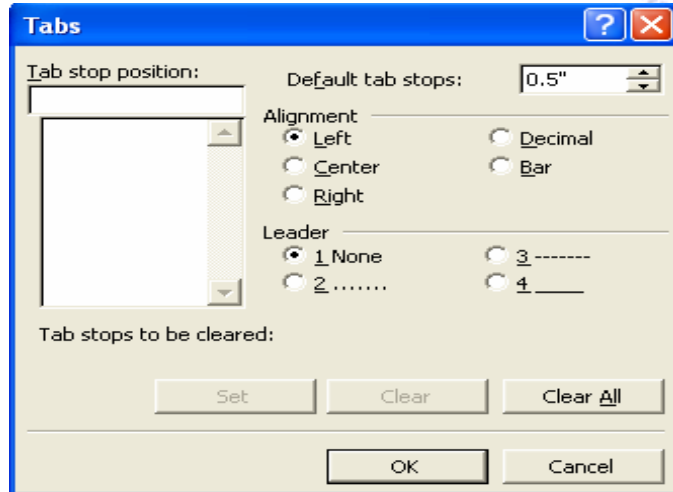
#### Dùng thanh Ruler để định các Tab Stop

Các Tab Stop nằm trên thanh Ruler, nếu thanh Ruler chưa hiện thì ta phải cho nó hiện bằng lệnh View / Ruler. Đơn vị đo của Ruler nên lấy là cm (dùng lệnh Tools / Options / General). Đầu phía trái của Ruler có Nút định dạng Tab Stop, khi nháy liên tiếp vào nút này sẽ xuất hiện 5 cách định dạng của Tab Stop: L văn bản sẽ đóng bên trái, J văn bản sẽ đóng bên phải, ⊥ văn bản đóng ở giữa, ⊥· đóng theo dấu chấm thập phân, | chèn một dấu vạch đứng.

Định một Tab Stop trên Ruler: nháy nút định dạng Tab Stop để hiện biểu tượng của kiểu định dạng mong muốn, nháy chuột lên thanh Ruler tại vị trí muốn đặt Tab Stop. Di chuyển một dấu Tab Stop trên thanh Rules: nháy và kéo. Xóa một dấu Tab Stop: nháy và kéo nó ra khỏi thanh Ruler. Xóa hết các Tab Stop đã đặt trên thanh Ruler: dùng lệnh Format / Tabs / Clear All.

### Dùng hộp thoại để định các Tab Stop

Để định các Tab Stop dùng lệnh Format/Tabs, xuất hiện hộp thoại Tabs. Xác định một Tab Stop: trong hộp Tab Stop Position gõ vào vị trí của Tab Stop tính từ lề trái, trong mục Alignment chọn một trong 5 dạng Tab Stop, nhấn nút Set. Để xóa một Tab Stop: chọn nó, nhấn nút Clear.



Để tạo các ký tự dẫn đầu của một Tab Stop, ta lựa chọn một nút dài trong mục Leader trước khi nhấn Set. Đó là cách thường dùng để tạo mục lục cuốn sách, ví dụ:

Lời nói đầu.....	3
Windows 2000-----	5
Word 2000 _____	40
Excel 2000	80

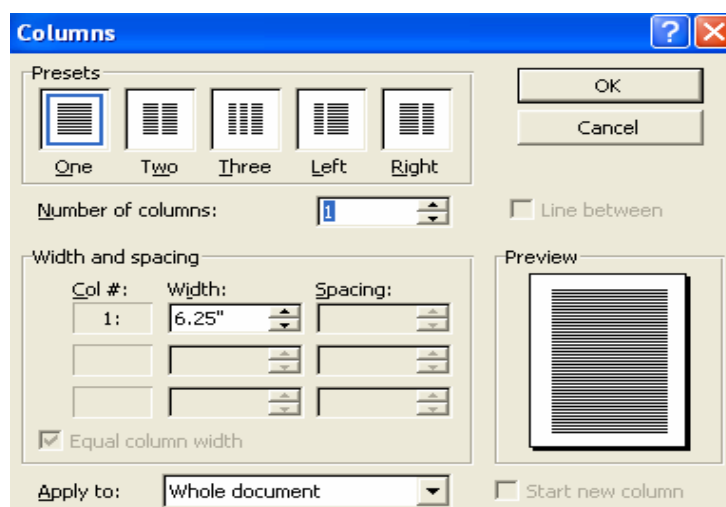
Sau khi nhấn OK để đóng hộp thoại Tabs, các Tab Stop xuất hiện trên thanh Rules.

### g. Định số cột cho văn bản

Có thể trình bày toàn bộ văn bản, một Section (phần văn bản) hay một nhóm các Paragraph trên nhiều cột (như dạng bài báo). Trước tiên cho hiện văn bản ở chế độ Page Layout bằng lệnh View / Page Layout. Sau đó dùng lệnh Format / Columns, xuất hiện hộp thoại Columns.

Xử lý hộp thoại:

- **Mục Presets** có các khả năng lựa chọn: One (trở lại một cột), Two (hai cột), Three (ba cột), Left và Right (hai cột nhưng lệch).
- **Mục number of Columns:** gõ vào số cột cần thiết.
- **Mục Width and Spacing:** xác định độ rộng từng cột và khoảng cách của cột tới cột tiếp theo.
- **Ô kiểm tra Equal Column Width:** nếu được đánh dấu thì độ rộng các cột bằng nhau.
- **Ô kiểm tra Line Between:** nếu được đánh dấu thì giữa các cột sẽ có đường kẻ dọc.



▪ **Mục Apply To:** xác định phạm vi văn bản được trình bày trên nhiều cột. Các lựa chọn: Whole Document (toàn văn bản), This Point Forward (từ vị trí con trỏ trở đi), Selected Sections (các phần văn bản được lựa chọn), Selected Text (văn bản được lựa chọn).

▪ **Mục Preview:** cho mẫu trình bày của trang in.

Việc phân văn bản thành các Setion và việc ngắt các cột văn bản được thực hiện nhờ lệnh Insert / Break (trình bày trong Mục 3.1).

#### ***h. Điền các dấu hình tròn hay số tự động ở đầu mỗi đoạn***

Ta có thể điền tự động các dấu chấm hình tròn, hình quả trám, dấu sao, dấu mũi tên, các số thường, các số La mã, các chữ cái hoa, chữ cái thường vào đầu mỗi đoạn (như ở các dòng dưới) bằng cách:

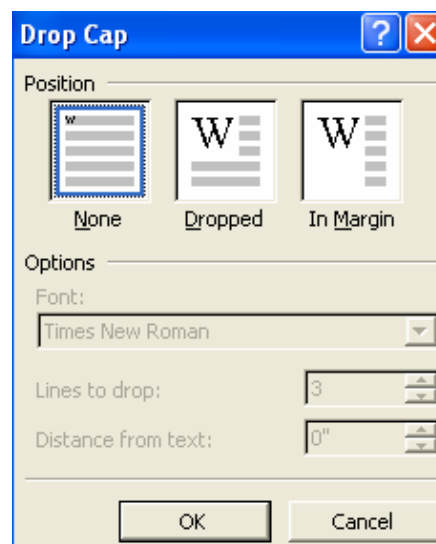
➤ Đánh dấu khối phần văn bản có các đoạn cần đánh thêm dấu tự động.

➤ Thực hiện lệnh Format / Bullets and Numbering, chọn mục Bulleted và chọn kiểu chấm, hoặc chọn mục Numbered và chọn kiểu số hay chữ. Nhờ nút Customine ta có thể chỉnh sửa các nút và số theo ý thích, hoặc chọn một kiểu mới lấy từ một bộ phong chữ, chọn khoảng cách của nút và của văn bản so với lề trái.

Khi ở chế độ tạo Bullet hay Number tự động, mỗi khi ấn Enter xuống dòng sẽ tạo tiếp một Bullet hay Number mới. Để hủy chế độ này và trở về chế độ thông thường ta sử dụng một trong hai cách: nháy vào nút Bullets hay Numbering trên thanh Formatting, hoặc trong hộp Style trên thanh Formatting chọn Normal.

#### ***i. Tạo chữ cái lớn đầu dòng***

Có thể sử dụng khả năng Drop Cap để tạo chữ cái lớn đầu dòng như trong đoạn văn bản này. Để thực hiện tạo chữ cái lớn, tiến hành các bước sau: chọn chữ cái hoặc đoạn văn bản cần



tao, dùng lệnh Format/Drop Cap, xuất hiện hộp thoại Drop Cap. Xử lý hộp thoại: trong mục Position chọn Dropped hoặc In Margin, chọn kiểu Font, số dòng văn bản cần chừa cho chữ cái (Lines to Drop), khoảng cách của văn bản so với chữ cái lớn (Distance from Text), chọn OK.

Để hủy bỏ hiệu ứng này, các bước tiến hành tương tự như trong mục Position, chọn None.

### ***i. Xóa tất cả các định dạng cũ của một khối văn bản***

Trong soạn thảo văn bản nhiều khi ta muốn xóa tất cả các định dạng cũ của một phần văn bản, nhất là khi ta muốn sửa một văn bản của người khác đã đánh. Các thao tác xóa:

- Đánh dấu khối toàn bộ phần văn bản định xóa các định dạng.
- Trên thanh Formatting: chọn phông chữ là .Vn Time cỡ 12 nếu soạn tiếng Việt bằng phông chữ ABC hoặc chọn phông chữ Times New Roman cỡ 12 nếu soạn tiếng Anh; nhấn vào các nút Bold, Italic, Underline để bỏ các kiểu chữ đậm, nghiêng và gạch chân; nhấn vào nút Justify để canh văn bản cả bên trái và bên phải.
- Dùng lệnh Format / Bullets and Numbering, chọn mục None để xóa các Bullet hay Number điền tự động ở đầu mỗi đoạn.
- Dùng lệnh Format / Paragraph, xuất hiện hộp thoại Paragraph, chọn lớp Indents and Spacing. Trong hộp thoại này ta lựa chọn Left: 0 cm, Right: 0 cm, Special: none, Before: 0 pt, After: 0 pt, Line: Single, chọn OK.

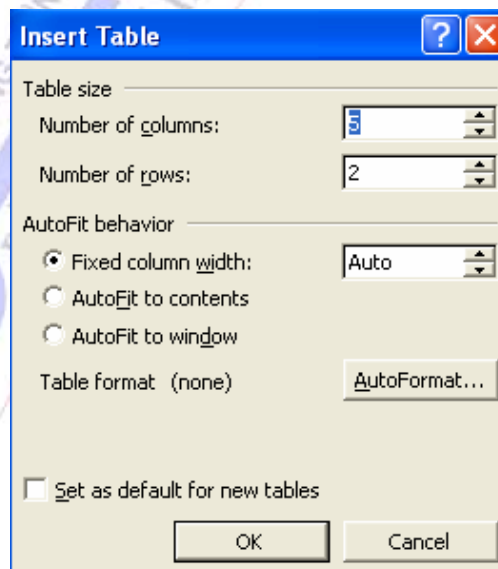
Sau đó ta có thể tiến hành định dạng hoàn toàn mới cho đoạn văn bản.

Trong các trường hợp đơn giản, có thể xóa các định dạng cũ của văn bản đã lựa chọn bằng Ctrl+Q (xóa định dạng Paragraph) và Ctrl+Spacebar (xóa định dạng Font).

## **3.1.3. Lập bảng biểu (TABLE)**

### ***3.1.3.1. Tạo một bảng mới***

***Dùng hộp thoại Insert Table tạo một mới:*** đưa con trỏ đến nơi cần tạo bảng, dùng lệnh Table / Insert / Table, hộp thoại Insert Table xuất hiện.



Nhập số cột vào mục Number of Columns, số dòng vào mục Number of Rows. Nhập chiều rộng mỗi cột vào mục Fixed column width. Nếu để Auto, chiều rộng mỗi cột tùy thuộc vào số lượng cột và chiều rộng trang in. Nút AutoFormat để chọn kiểu bảng mẫu có sẵn. Cuối cùng, chọn OK.

Khi văn bản nằm trong một ô, nó sẽ tự động xuống dòng khi chiều rộng ô không đủ. Các thao tác để di chuyển con trỏ trong bảng: đến ô kế tiếp ấn Tab, đến ô trước đó Shift+Tab, đến ô đầu tiên của dòng hiện tại Alt+Home, đến ô cuối cùng của dòng hiện tại Alt+End, đến ô đầu tiên trong cột Alt+PgUp, đến ô cuối cùng của cột Alt+PgDn.

**Tự vẽ một bảng:** chuyển màn hình sang chế độ Page Layout, dùng lệnh View / Toolbars / Tables and Borders để hiện thanh công cụ Tables and Borders, nhấp vào nút Draw Table trên thanh công cụ. Nhấp chuột và kéo để được một khung chữ nhật, nhấp chuột và kéo để vẽ các đường ngang dọc tùy ý (mỗi đường chỉ cần vẽ một đoạn ngắn là Word đã tự động hoàn tất cả đường). Nếu kẻ sai có thể dùng nút Eraser trên thanh công cụ để xóa đường. Kẻ xong ta nhập văn bản vào bảng.

### 3.1.3.2. Các thao tác sửa đổi trong bảng

#### Chọn các ô, hàng hay cột

- *Chọn một ô:* di chuyển con trỏ chuột đến mép trái ô đó cho đến khi thấy mũi tên màu đen xuất hiện thì nhấp chuột.

- *Chọn một hàng:* rê chuột từ ô đầu đến ô cuối dòng (hoặc dùng lệnh Table / Select / Row).

- *Chọn một cột:* di chuyển con trỏ chuột lên mép trên của bảng đến khi thấy mũi tên đen chỉ xuống thì nhấp chuột (hoặc dùng lệnh Table / Select / Column).

- *Chọn nhiều ô trong bảng theo một vùng chữ nhật:* đưa con trỏ nhập vào ô đầu tiên, sau đó di chuyển chuột đến ô cuối, giữ phím Shift đồng thời nhấp chuột; hoặc nhấp chuột vào ô đầu tiên và kéo; hoặc bấm Shift kèm với các phím mũi tên.

- *Muốn chọn toàn bộ bảng:* đưa con trỏ vào trong bảng, dùng lệnh Table / Select / Table.

#### Chèn thêm các ô, hàng, cột vào bảng

- *Chèn một hàng duy nhất vào cuối bảng:* đặt điểm chèn vào ô cuối cùng của bảng và ấn phím Tab.

- *Chèn nhiều hàng vào cuối bảng:* cho điểm chèn vào dưới hàng cuối cùng của bảng dùng lệnh Table / Insert Rows, xuất hiện hộp thoại Insert Rows, nhập vào số lượng hàng cần thêm.

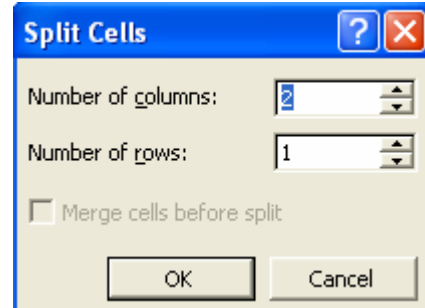
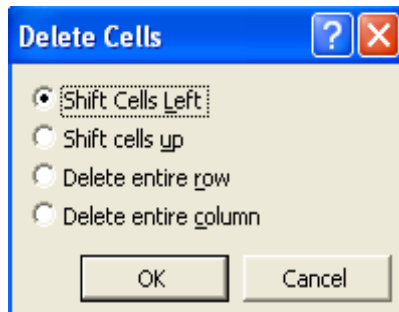
- *Chèn thêm N hàng trắng ở giữa bảng:* lựa chọn N hàng có sẵn, dùng lệnh Table / Insert, chọn Rows Above (chèn N hàng trắng vào phía trên các hàng đã chọn) hoặc Rows Below (chèn vào phía dưới).

- *Chèn thêm N cột trắng ở giữa bảng:* lựa chọn các cột đã có sẵn, dùng lệnh Table/Insert, chọn Columns to the Left (chèn N cột trắng vào bên trái các cột đã chọn) hoặc Columns to the Right (chèn bên phải).

- *Chèn thêm các ô:* muốn chèn bao nhiêu ô thì chọn bấy nhiêu ô, dùng lệnh Table/Insert/Cells, hộp thoại Insert Cells xuất hiện. Chọn một trong các chức năng sau: *Shift Cells Right*



(chèn các ô mới và đẩy các ô đã chọn sang phải), *Shift Cells Down* (chèn các ô mới và đẩy các ô đã chọn xuống dưới), *Insert Entire Row* (chèn toàn bộ các hàng ứng với các ô đã chọn), *Insert Entire Column* (chèn toàn bộ các cột).

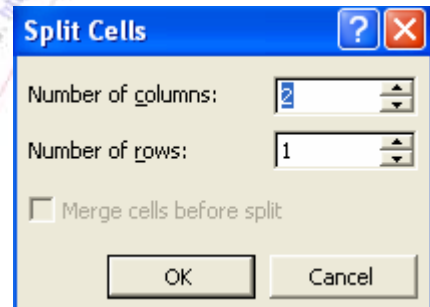


**Xóa các ô, hàng hay cột của bảng:** chọn các ô, hàng hay cột cần xóa, dùng lệnh Table / Delete / Cells (hay Columns, Rows), xuất hiện hộp thoại Delete Cells. Các lựa chọn trong hộp thoại: Shift cells left (xóa các ô đã chọn và chuyển các ô còn lại trong dòng sang trái), Shift cells up (xóa các ô đã chọn và chuyển các ô còn lại lên phía trên), Delete entire row (xóa tất cả các dòng chứa các ô đã chọn và chuyển các dòng còn lại lên trên), Delete entire column (xóa tất cả các cột chứa các ô đã chọn và chuyển các cột còn lại sang trái).

#### Gộp, tách các ô trong bảng

- Gộp nhiều ô lại thành một: chọn các ô cần gộp, dùng lệnh Table / Merge Cells. Khi đó, các đường ngăn cách giữa các ô sẽ mất đi, tạo thành một ô duy nhất.

- Tách một ô thành nhiều ô nhỏ: chọn ô cần tách, dùng lệnh Table / Split Cells, hộp thoại Split Cells xuất hiện. Các mục lựa chọn: Number of columns (số phần chia theo chiều dọc), Number of row (số phần chia theo chiều ngang), Merge cells before split (trong trường hợp chọn nhiều ô Word sẽ gộp các ô thành một ô trước khi chia nhỏ).



**Di chuyển, sao chép các ô, hàng, cột trong bảng:** Chọn các ô, hàng hay cột cần di chuyển hoặc sao chép. Đưa con trỏ chuột đến phần được chọn, giữ nút bên trái, rồi thực hiện một trong các thao tác sau:

- Rê chuột đến vị trí mới rồi thả nút trái chuột: di chuyển phần được chọn.
- Giữ phím Ctrl, đồng thời rê chuột đến vị trí mới rồi thả nút trái: sao chép phần được chọn.

Ta cũng có thể di chuyển, sao chép dữ liệu của các ô, hàng hay cột trong bảng bằng các nút Cut, Copy, Paste trên thanh Standard như văn bản thông thường.

#### Thay đổi chiều rộng cột

- Dùng chuột: di chuyển con trỏ chuột đến biên phải cột, khi trỏ chuột có dạng mũi tên hai chiều thì rê chuột để thay đổi chiều rộng cột.

- Dùng menu: chọn cột cần thay đổi, dùng lệnh Table / Table Properties, xuất hiện hộp thoại, chọn lớp Column, lựa chọn chiều rộng cột trong hộp cạnh mục Preferred width. Nháy vào các nút Previous Column và Next Column để xem và thay đổi độ rộng của các cột khác.

**Thay đổi độ cao của hàng:** Dùng chuột: trỏ chuột tới cạnh dưới của hàng để con trỏ thành mũi tên hai kép và kéo cạnh xuống phía dưới.

Dùng menu: thực hiện lệnh Table / Table properties, hiện hộp thoại, chọn lớp Row.

Lựa chọn độ cao dòng trong hộp bên cạnh mục Specify height. Các nút Previous Row và Next Row để xem và thay đổi độ cao của các dòng khác.

**Thay đổi hướng viết văn bản trong các ô:** lựa chọn các ô, nhấp nút phải chuột vào một ô đã chọn, xuất hiện menu dọc nhỏ, chọn mục Text Direction, xuất hiện hộp thoại Text Direction, lựa chọn một trong 3 hướng theo mẫu trong hộp thoại.

**Đóng văn bản trong ô:** Ta có thể đóng văn bản cho các ô đã chọn trong bảng theo chiều ngang (trái, phải, giữa) và theo chiều dọc (trên, dưới, giữa) bằng cách: nhấp nút phải chuột vào một trong các ô đã chọn, hiện menu tắt, chọn Cell Alignment, hiện một bảng các mẫu đóng văn bản, chọn một trong 9 cách.

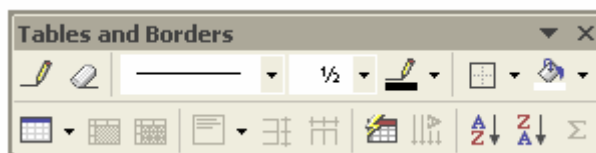
**Xác định khoảng cách của văn bản so với bốn cạnh của ô:** chọn các ô, nhấp nút phải chuột vào một trong các ô đã chọn, hiện menu tắt, chọn Table Properties, hiện hộp thoại, chọn lớp Table, nhấp nút Options, hiện hộp thoại Table Options, nhập các khoảng cách vào các mục Top (cách cạnh trên), Bottom (cách cạnh dưới), Left (cách cạnh trái), Right (cách cạnh phải), khoảng cách tối thiểu nên là 0.05 cm.

**Chọn Heading cho bảng:** lựa chọn các hàng sẽ làm Heading cho bảng (thường là các hàng đầu), dùng lệnh Table / Heading Rows Repeat. Khi bảng dài cần cắt sang trang sau thì phần bảng ở trang sau cũng có các dòng Heading. Muốn bỏ Heading thì làm lại thao tác trên một lần nữa.

**Tách bảng làm đôi:** đặt điểm chèn vào hàng muốn tách bắt đầu từ đó, dùng lệnh Table / Split Table.

**Di chuyển bảng và thay đổi kích thước bảng:** Khi đưa con trỏ chuột vào bên trong một bảng (con trỏ nhập có thể ở nơi khác), góc trên bên trái và góc dưới bên phải của bảng xuất hiện hai nút. Nếu nhấp vào nút trên bên trái (có dấu chữ thập với 4 mũi tên ở đầu) và kéo thì sẽ di chuyển toàn bộ bảng đi nơi khác (giữ nguyên kích thước). Nhấp vào nút dưới bên phải và kéo sẽ thay đổi kích thước của toàn bộ bảng.

### 3.1.3.3. Dùng thanh công cụ Tables and Borders khi tạo bảng



Từ trên xuống dưới và từ trái sang phải có các nút: Draw Table (vẽ bảng tự do), Erase (xóa các đường kẻ đã vẽ), Line Weight (chọn độ rộng đường kẻ), Boder Color (màu đường kẻ), Outside Boder (tạo các đường kẻ và xóa), Shading Color (màu nền), Insert Table (chèn một bảng), Merge Cells (gộp các ô đã chọn), Split Cells (tách các ô), Align Top Left (chọn các cách đóng văn bản), Distributed Rows Evenly (chỉnh cho chiều cao các hàng đang chọn bằng nhau), Distributed Columns Evenly (chỉnh cho bề rộng các cột đang chọn bằng nhau), Table Autoformat (chọn các mẫu bảng có sẵn), Change Text Direction (chọn hướng văn bản trong ô), Sort

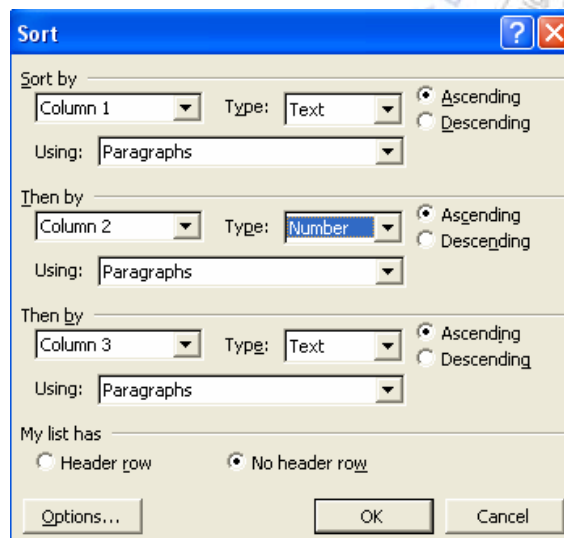
Ascending (sắp xếp theo chiều tăng), Sort Descending (sắp xếp theo chiều giảm), Autosum (tính tổng tự động).

#### 3.1.3.4. Sắp xếp dữ liệu trên một bảng

Có thể sắp xếp các hàng trong bảng theo một cột nào đó (tối đa xếp theo ba cột cùng với thứ tự ưu tiên). Dữ liệu trong cột phải có cùng kiểu gồm một trong ba kiểu sau: Text (văn bản) – Number (kiểu số) – Date (kiểu ngày tháng). Nếu dữ liệu ở các hàng trong một cột khác kiểu nhau thì chỉ được phép sắp xếp theo kiểu Text.

Các bước sắp xếp trên bảng:

- Chọn các hàng cần sắp xếp. Nếu sắp xếp toàn bộ bảng thì chỉ cần đưa con trỏ vào một ô bất kỳ trong bảng.



- Dùng lệnh Table / Sort, hộp thoại Sort xuất hiện. Trong hộp thoại này cần chọn: các khóa sắp xếp trong hộp Sort By và Then By; chọn sắp xếp theo kiểu dữ liệu nào: kiểu ký tự (Text), kiểu số (Number) hay kiểu ngày (Date); chọn thứ tự sắp xếp: tăng dần (Ascending) hoặc giảm dần (Descending).

#### 3.1.3.5. Tính toán trên một bảng:

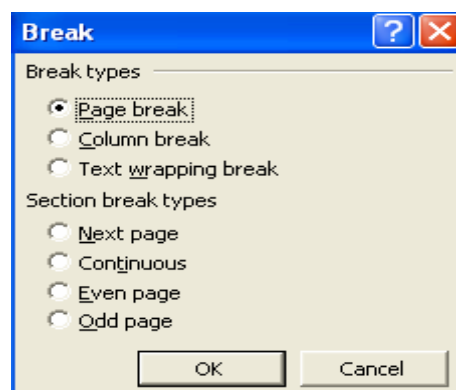
Ta đưa con trỏ vào nơi muốn đặt kết quả tính toán sau đó chọn Table / Formula rồi gõ vào công thức toán học (+, -, \*, /) hoặc các hàm đã có sẵn trong word (sum, average,.....) thông qua các số hạng (có thể là địa chỉ của các ô trong bảng đó, hoặc giá trị số...)

### 3.1.4. Định dạng trang và in ấn

#### 3.1.4.1. Khái niệm section

Bình thường một tài liệu được chia thành nhiều trang, các trang có định dạng giống nhau. Nhưng đôi khi ta muốn một số phần văn bản có định dạng trang khác (như in theo chiều ngang khổ giấy A4) hoặc trình bày theo nhiều cột, khi đó cần phải chia văn bản thành nhiều vùng (Section), mỗi Section có thể có định dạng trang khác nhau. Nếu không chia Section thì Word mặc nhiên coi toàn bộ văn bản là một Section duy nhất.

**Lệnh Insert / Break** dùng để tạo một ngắt trang, ngắt cột hay ngắt section bằng tay. Khi dùng lệnh này xuất hiện hộp thoại Break. Chọn nút Page Break để chèn một ngắt trang, chọn Column Break để chèn một ngắt cột trong trường hợp soạn văn bản theo nhiều cột. Khung Section Break Types để tạo một ngắt Section, nó có 4 lựa chọn: Next Page (section mới sẽ sang trang tiếp theo), Continuous (section mới vẫn nằm liên tục với section trước đó), Even Page (section mới sẽ bắt đầu ở một trang đánh số chẵn), Odd Page (section mới sẽ bắt đầu ở một trang đánh số lẻ).



**Xóa các ngắt trang và ngắt Section:** Đặt màn hình ở chế độ Normal View, khi đó ngắt trang là một đường gạch ngang chấm chấm ở giữa có chữ Page Break, di con trỏ tới các đường này và ấn phím Delete.

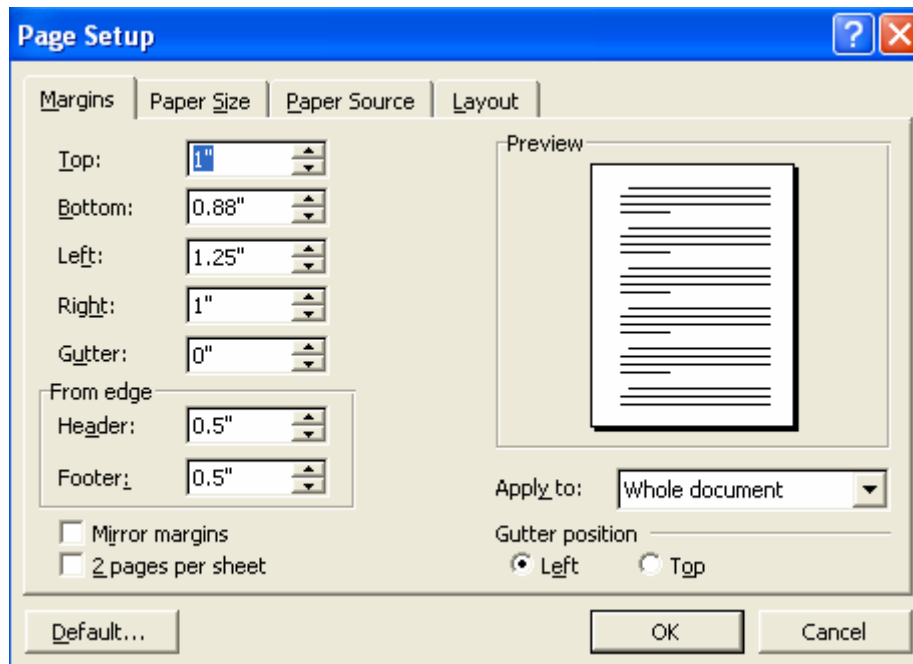
#### 3.1.4.2. Chọn cỡ giấy và đặt lề

Trước khi bắt tay vào soạn thảo một văn bản ta phải chọn cỡ giấy định in và đặt lề cho trang in bằng lệnh File / Page Setup, xuất hiện hộp thoại Page Setup.

**Chọn cỡ giấy:** trong hộp thoại Page Setup chọn lớp **Paper size**. Nháy chuột vào mũi tên xuống ở mục Paper size để chọn cỡ giấy. Có thể lựa chọn một trong các cỡ được định nghĩa sẵn. Muốn tạo trang in có kích thước tùy ý thì thay đổi các con số quy định chiều rộng (Width) và chiều cao (Height). Khi đó cỡ giấy được gọi là Custom size (cỡ của người dùng tự định nghĩa). Đối với các công văn giấy tờ ta thường chọn khổ giấy A4, 210 x 297 cm. Chọn hướng in là Portrait (hướng in dọc) hoặc Landscape (hướng in ngang). Trong mục Apply To (áp dụng cho) chọn một trong ba khả năng: This Section (để áp dụng cho Section văn bản hiện tại), This Point Forward (áp dụng cho vùng từ vị trí hiện tại trở về sau), Whole Document (áp dụng cho toàn bộ tài liệu). Chọn OK để xác nhận các thông số mới thiết lập. Có thể chọn Default để lưu các thông số vào tệp khuôn mẫu sử dụng cho nhiều tài liệu về sau.

**Đặt lề cho trang in:** Trong hộp thoại Page Setup chọn lớp **Margins**, hộp thoại Page Setup có dạng như trang sau. Thay đổi khoảng cách từ mép giấy đến phần văn bản gồm các phần: Trên (Top) - Dưới (Bottom) - Trái (Left) - Phải (Right). Con số trong hộp Gutter để chỉ khoảng cách chứa lề cho việc đóng tập tài liệu. Mục Gutter Position: chọn Gutter nằm bên trái trang (Left) hay bên trên trang (Top). Mục Header: khoảng cách từ mép trên trang giấy tới phần Tiêu đề đầu trang. Mục Footer: khoảng cách từ mép dưới trang giấy tới phần tiêu đề cuối trang. Sau khi thay đổi một trong các thông số quy định lề, phần Preview sẽ cho thấy hình dạng tương đối của văn bản so với kích thước giấy. Mục Apply To và các bước tiếp theo thao tác tương tự như phần Page size. Chọn mục Mirror Margins: lề trái và lề phải của hai trang chẵn và lề đối xứng nhau qua gáy cuốn sách, phần Preview hiện hai trang giấy.

Các số liệu trong hộp thoại Page Setup là một mẫu đặt lề cho các công văn giấy tờ soạn trên khổ giấy A4.



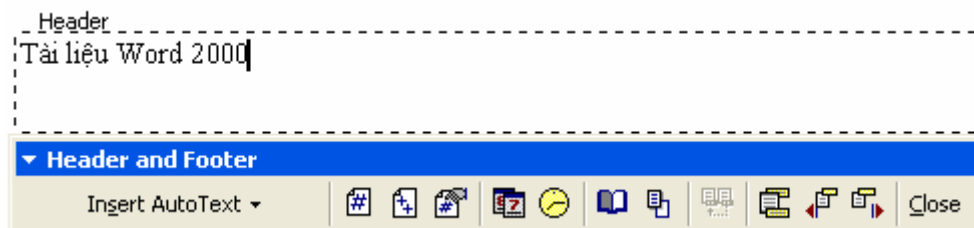
**In hai trang trên một tờ giấy:** đó là trường hợp ta muốn in hai trang trên tờ giấy A4 đặt nằm ngang, đường phân chia theo chiều dọc chính giữa (hoặc tờ giấy A4 đặt dọc, đường phân chia hai trang nằm ngang chính giữa). Khi chèn số trang máy sẽ tự động điền số trang liên tục theo các trang nhỏ.

Để in hai trang trên tờ giấy A4 nằm ngang dùng lệnh File / Page Setup, hiện hộp thoại Page Setup. Trong lớp Paper Size chọn cỡ giấy A4 29.7 x 21 cm, hướng in là Landscape, mục Apply To chọn Whole Document. Trong lớp Margins chọn Top 2 cm, Bottom 2.5 cm, Outside 1.9 cm, Inside 1.99 cm, Gutter 0 cm, Header 1.2 cm, Footer 1.5 cm, đánh dấu kiểm tra vào hộp **2 pages Sheet**, mục Gutter Position chọn Left, mục Apply To chọn Whole Document. Nháy OK.

### 3.1.4.3. Đặt tiêu đề đầu trang và cuối trang

Tiêu đề đầu trang và cuối trang là phần văn bản được in trên đầu (hoặc cuối) mỗi trang của tài liệu. Thường sử dụng mục này để thêm vào tài liệu tên tác giả, tên tài liệu, cơ quan phát hành... Có thể áp dụng các kiểu đặt tiêu đề sau: tiêu đề giống nhau trên toàn bộ các trang của tài liệu, tiêu đề của các trang đầu tiên khác với các trang còn lại, một tiêu đề dành cho các trang chẵn, tiêu đề kia dành cho các trang lẻ.

Các bước tiến hành để đặt tiêu đề: dùng lệnh View / Header and Footer. Phần trình bày tiêu đề sẽ xuất hiện cùng với thanh dụng cụ.



Gõ vào nội dung tiêu đề đầu trang ở khung Header, tiêu đề cuối trang ở khung Footer. Cuối

cùng, nhấn nút Close để trở về chế độ soạn thảo thông thường.

Từ trái sang phải thanh công cụ Header and Footer có các nút: Insert Number of pages (chèn tổng số trang của văn bản), Format Page Number, Insert Date (chèn ngày hiện tại), Insert Time (chèn giờ hiện tại), Page Setup, Show / Hide Document Text (cho hay không cho phép hiển thị văn bản của trang hiện tại), Same as Previous (thực hiện chế độ chọn Header và Footer như ở trang trước), Switch between Header and Footer (chuyển đổi con trỏ giữa tiêu đề đầu trang và cuối trang), Show Previous (xem Header hoặc Footer kế trước), Show Next (xem Header hoặc Footer kế sau), Close Header and Footer.

Để kích hoạt chế độ sửa đổi tiêu đề, có thể nhấn đúp chuột vào phần ghi tiêu đề (phần chữ nhạ trong chế độ Page Layout).

#### 3.1.4.4. Tạo các chú thích ở cuối trang hoặc cuối văn bản

Đối với một văn bản, ta có thể tạo các chú thích cho các cụm từ, chú thích này có thể ở cuối từng trang (gọi là Footnote) hoặc có thể để tất cả ở cuối văn bản (gọi là Endnote). Các chú thích này sẽ được đánh số tự động.

Khi tạo chú thích ta có thể đặt màn hình ở chế độ Normal View hay Page Layout đều được. Giả sử màn hình ở chế độ Normal View ta thao tác như sau:

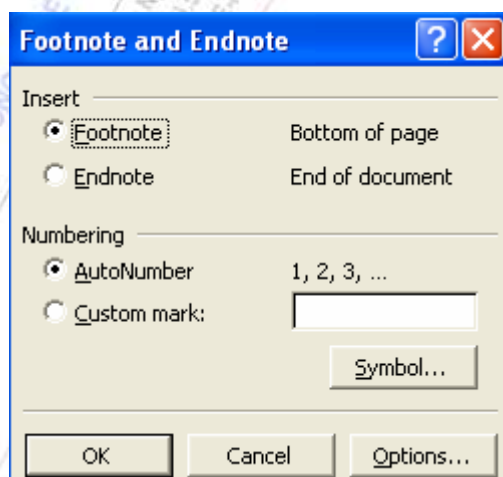
- Đặt con trỏ nhập vào ngay sát sau từ cần điền số thứ tự của chú thích.

- Dùng lệnh Insert / Footer, xuất hiện hộp thoại.

- Xử lý hộp thoại: chọn Footnote hay Endnote, chọn AutoNumber, nhấn OK, màn hình chia thành hai nửa, nửa nhỏ ở phía dưới và có thanh công cụ.

- Nhập văn bản vào nửa dưới của màn hình, nhập xong nhấn nút Close trên thanh công cụ.

**Xóa một chú thích:** lựa chọn số chú thích ở sau chữ, ấn phím Delete, chú thích bị xóa cùng với văn bản kèm theo, đồng thời các chú thích còn lại tự động đánh số thứ tự lại.

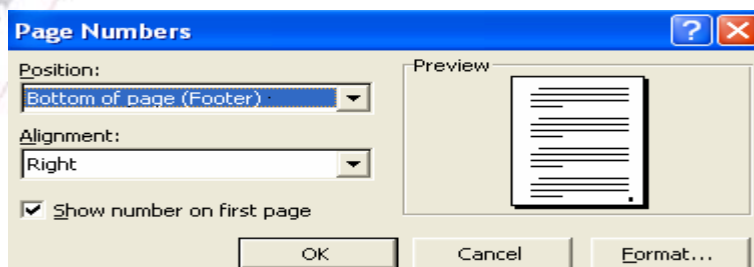


#### 3.1.4.5. Đánh số trang

Vào menu dọc Insert, chọn Page Numbers, xuất hiện hộp thoại Page Numbers. Xử lý hộp thoại:

- Hộp Position để chọn vị trí đánh số trang: Top of Page (ở đầu trang), Bottom of Page (ở cuối trang).

- Hộp Alignment chọn việc đóng số trang: bên trái, phải, giữa hoặc bên trong/ngoài mép giấy đối với trường hợp in theo hai mặt đối xứng của trang giấy.



- Nút kiểm tra Show Number on First Page: có hay không hiển thị số trang ở trang đầu tiên của tài liệu.

- Nháy chuột nút Format để thiết lập các thông số: bắt đầu đánh từ số mấy (Start At) hoặc tiếp theo số trang của phần trước (Continue from previous section), kiểu cách số (Number Format), chọn OK để xác nhận các thiết lập hoặc Cancel để bỏ qua.

#### 3.1.4.6. Xem trước trên màn hình cách bố trí các trang in

Giả sử toàn bộ văn bản đã được định dạng xong và đã đánh số trang. Để xem trước cách trình bày các trang ta dùng lệnh File / Print Preview (hoặc nháy nút Print Preview trên thanh Standard). Nếu các trang phân không hợp lý ta có thể sửa đổi. Khi màn hình soạn thảo biến mất, xuất hiện màn hình mới với các trang đã phân, đỉnh màn hình có một thanh công cụ giúp ta xem các trang một cách thuận tiện:



Chức năng của các nút:

- Nút Print: in văn bản ra máy in

- Nút Magnifier: phóng to, thu nhỏ trang. Khi nút này chìm: con trỏ chuột hình dấu cộng trong hình tròn, nháy chuột vào trang văn bản thì sẽ phóng to hay thu nhỏ trang theo tỷ lệ ghi trong ô điều khiển Zoom. Khi nút này nổi: ta có thể sửa chữa được văn bản bằng cách nháy chuột vào vị trí cần sửa để xuất hiện con trỏ nhập văn bản. Như vậy trong chế độ Print Preview ta có thể vừa xem vừa sửa các trang.

- Nút One Page: mỗi lần trên màn hình chỉ xem được một trang.

- Nút Multiple Pages: nháy nút này sẽ xuất hiện một bảng nhỏ cho phép ta chọn xem được mấy trang đồng thời trên màn hình (từ 1 đến 6).

- Ô điều khiển Zoom: nháy chuột vào phím mũi tên của ô này và chọn tỷ lệ phóng to hay thu nhỏ trang văn bản một cách thích hợp.

- Nút View Ruler: cho hay không cho hiện các thước kẻ ngang và dọc trên màn hình.

- Nút Shrink To Fit: co văn bản lại cho vừa. Dùng trong trường hợp: trang văn bản cuối cùng chỉ có một ít dòng, dùng lệnh này văn bản sẽ dồn lại vào các trang trên để bỏ đi trang cuối cùng. Đây là một nút rất nguy hiểm đối với những tài liệu dài có định dạng phức tạp, toàn bộ văn bản có thể bị định dạng lại không hợp với ý muốn ban đầu, ta phải mất nhiều thời gian để định dạng lại.

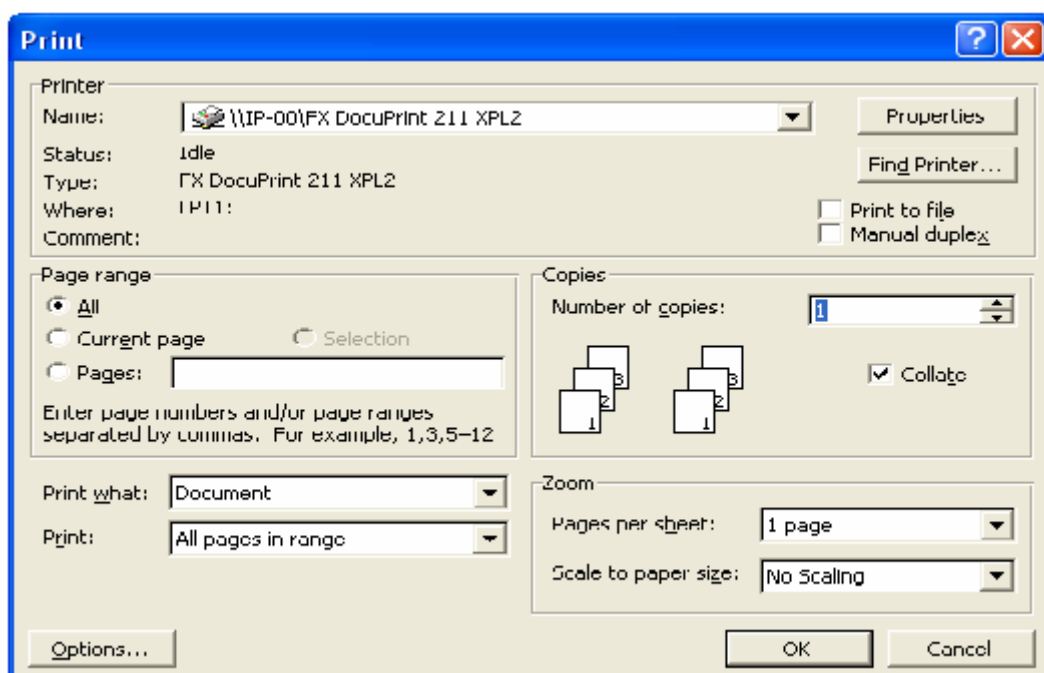
- Nút Full Screen: dành toàn màn hình để xem các trang hay không.

- Nút Close: về lại màn hình soạn thảo.

- Nút Context Sensitive Help: xem hướng dẫn sử dụng thanh công cụ này.

#### 3.1.4.7. In văn bản

Muốn in văn bản dùng lệnh File / Print (hoặc nháy chuột vào nút Print trên thanh Standard), xuất hiện hộp thoại Print:



Xử lý hộp thoại:

- Mục Printer: thông báo máy in đang dùng và lựa chọn máy in khác. Máy in này có thể là một máy in kim hay laser đang nối với máy tính, có thể là một máy in trên mạng. Trong trường hợp ta chỉ muốn in ra file PRN thì máy in trong mục này phải là tên máy in mà sau này ta sẽ dùng để in tệp PRN.

- Mục Print What: chọn loại đối tượng cần in (Document, Comments, Styles,...), chọn loại Document.

- Mục Copies: nhập vào số bản cần in, ngầm định là 1 bản.

- Mục Page Range: quy định các trang văn bản cần in. Có các lựa chọn: All (in hết), Current Page (chỉ in một trang hiện tại), Pages (quy định in các trang nào, ví dụ 1, 3, 5-12), Selection (in khối văn bản đang chọn).

- Mục Print: chọn All Page in range, Odd pages hay Even Pages.

- Ô kiểm tra Collate: trong khi in nhiều bản, in toàn bộ tài liệu mỗi lần hay là cùng một trang nhiều lần.

- Mục Pages per Sheet: in cùng một bản tài liệu trên cùng một trang giấy, chẳng hạn như Card visit.

- Mục Scale to paper size: tự động phóng to hay thu nhỏ nội dung của tài liệu để phù hợp với nội dung vài trang.

- Nút Options: xuất hiện hộp thoại Print với nhiều lựa chọn in cao cấp khác. Trong hộp thoại này cần chọn các mục: Allow A4 / Letter page resizing (định kích thước của tài liệu lại để vừa vặn trên kích cỡ A4), Background Printing (in tài liệu nhanh chóng vào vùng đệm để có thể tiếp tục công việc khác trong khi in), Drawing Objects (in đồ họa khi tài liệu có chứa nó).



- Nút Properties: khi nháy nút này sẽ xuất hiện hộp thoại ứng với máy in đã chọn, chọn lớp Graphics, chọn mật độ in cao nhất, chọn chất lượng in là Fine, di chuyển trong mục Indensity để điều chỉnh độ đậm nhạt.

- Ô kiểm tra Print to File: nếu được đánh dấu thì văn bản sẽ in ra tệp.

- Nút OK: khi nháy nút này thì văn bản bắt đầu được in. Trong trường hợp nếu lựa chọn in ra File thì Word cho xuất hiện hộp thoại mới Print To File, trong hộp thoại này ta phải lựa chọn thư mục sẽ chứa tệp, tên tệp có đuôi là PRN và nháy OK. Ta có thể ghi lại tệp PRN ra đĩa và màn sang máy khác có máy in để in với điều kiện máy in phải có đúng tên như ta đã lựa chọn, lệnh in:

COPY tệp.PRN PRN

Đây là lệnh của DOS, ta không cần Windows và Word nữa.

### 3.1.5. Chèn hình ảnh và công thức toán

#### 3.1.5.1. Chèn thêm các ký tự đặc biệt

Để chèn thêm một ký tự đặc biệt (ví dụ  $\mathcal{NR}$ ), thực hiện theo trình tự như sau: đưa con trỏ văn bản đến nơi cần chèn, dùng lệnh Insert / Symbol, hiện hộp thoại Symbol, chọn lớp Symbols, nháy chuột vào hộp Font để chọn bộ Font chứa các ký hiệu (chẳng hạn font Symbol, Webdings, Wingdings, Times New Roman, Tahoma), nháy chuột vào ký hiệu cần chèn, nháy nút Insert để chèn, nháy nút Close để đóng hộp thoại Symbol. Khi ký tự đã hiện lên trang văn bản ta có thể sao chép và dán bình thường.

#### 3.1.5.2. Đánh công thức toán – Equation

Equation chuyên dùng để đánh công thức toán và khoa học phức tạp. Để tạo một công thức toán trước tiên ta đặt con trỏ nhập vào chỗ định chèn công thức, dùng lệnh Insert / Object, hiện hộp thoại Object, chọn lớp Create New, nháy đúp chuột vào mục Microsoft Equation 3.0. Khi đó tại điểm chèn xuất hiện khung để nhập công thức và cửa sổ nhỏ Equation. Lúc này menu chính của Word xuất hiện thêm hai mục: Style và Size.

Muốn lựa chọn phong chữ cho các thành phần của công thức ta dùng lệnh Style / Define, màn hình xuất hiện hộp thoại Styles. Ta nên chọn Text (văn bản trong công thức): phong Times New Roman, Variable (các biến): Italic, L.C.Greek: Italic, Matrix-Vector: Bold.

Ta có thể lựa chọn cỡ chữ của các thành phần công thức bằng lệnh Size / Define, hộp thoại Sizes xuất hiện. trong hộp thoại này ta có thể lựa chọn các kích cỡ: chữ bình thường trong công thức (Full) là 12 pt, chỉ số mức thứ nhất (Subscript) là 9 pt, chỉ số mức thứ hai (Sub-subscript) là 5 pt, chữ Hy Lạp (Symbol) là 18 pt.

Công dụng của 19 nút trong cửa sổ Equation như sau: Nút 1: đánh các dấu bất đẳng thức. Nút 2: chèn các khoảng cách rộng và hẹp vào công thức, trong công thức ta không thể dùng phím khoảng cách để tạo khoảng cách. Nút 3: đánh các biến có dấu mũ (gõ chữ trước, chọn sau). Nút 4, 5, 6, 7, 8: các ký tự toán học. Nút 9, 10: các chữ Hy Lạp. Nút 11: các công thức có ngoặc hai bên hay một bên. Nút 12: phân số và căn. Nút 13: chỉ số trên và chỉ số dưới. Nút 14: các dấu tổng. Nút 15: các loại dấu tích phân. Nút 16: gạch trên và gạch dưới các công thức lớn. Nút 17: mũi tên có điền thêm các công thức ở trên hay ở dưới. Nút 19: đánh ma trận và các công thức có cấu trúc bảng.

Mỗi lần chọn một nút, Equation sẽ tạo cho ta một khung nhỏ để nhập dữ liệu, chuyển sang khung khác: nhấp chuột vào khung muốn chuyển đến. Muốn thoát khỏi các khung nhập chỉ số hay các khung nhập phân số ta ấn phím mũi tên phải. Kết thúc nhập công thức: nhấp đúp vào nút điều khiển của cửa sổ Equation để trở về màn hình soạn thảo, nhấp ra ngoài khung công thức để công thức hiện bình thường. Trong khung nhập công thức muốn gõ một ít văn bản vào ta phải dùng lệnh: Style /Text, gõ chữ, trở về lại chế độ gõ công thức: Style / Math.

Sửa chữa một công thức: nhấp vào công thức, xuất hiện khung với 8 nút nhỏ xung quanh, nhấp đúp vào công thức, xuất hiện cửa sổ Equation cho phép ta sửa công thức.

Khi đánh công thức phải thường xuyên kích hoạt Equation, do đó nếu dùng menu thì rất lâu, ta phải đưa nút Equation vào thanh Standard. Cách làm: nhấp nút phải chuột vào nền của thanh Standard, hiện menu tắt, chọn Customize, hiện hộp thoại Customize. Thao tác với hộp thoại Customize: chọn lớp Commands, trong hộp Categories chọn mục Insert, khi đó trong khung Commands xuất hiện các nút, nhấp chuột vào nút Equation và kéo nó lên thanh Standard.

### 3.1.5.3. Tạo một AutoShape

AutoShape (hình tự động) là những mẫu hình làm sẵn trong Word để trang trí trên văn bản. Mỗi mẫu hình vẽ gọi là một đối tượng vẽ, các đối tượng vẽ có thể nằm chồng lên nhau, có thể nằm trên hay dưới văn bản. Để vẽ ta cần bật thanh công cụ vẽ Drawing bằng lệnh View/Toolbars / Drawing:



Tên các nút từ trái sang phải: 1-Draw, 2-Select Objects, 3-AutoShape, 4-Line, 5-Arrow, 6-Rectangle, 7-Oval, 8-Text Box, 9-vertical Text box, 10-Insert WordArt, 11-Insert Diagram Organization Chart, 12- Insert Clip Art, 13-Insert Picture 14-Fill Color, 15-Line Color, 16-Font Color, 17-Line Style, 18-Dash Style, 19-Arrow Style, 20-Shadow, 21-3D.

Muốn vẽ một hình hãy nhấp chuột vào nút AutoShape, xuất hiện menu dọc có 6 mục ứng với 6 nhóm hình, lựa chọn một nhóm hình, lựa chọn mẫu hình, rê chuột trên vùng văn bản muốn vẽ hình, khi kích thước hình vẽ vừa ý thì nhả chuột. Trong nhóm Lines có nút Freeform dùng để vẽ tự do, muốn dùng máy nhấp đúp chuột. Một số mẫu hình thông dụng như đường thẳng, mũi tên, hình chữ nhật, hình ellipse có thể dùng ngay các nút Line, Arrow, Rectangle, Oval trên thanh công cụ.

Nếu ấn phím Shift đồng thời kéo chuột để vẽ đường thẳng thì sẽ vẽ được các đường thẳng góc, kéo chuột để vẽ hình chữ nhật thì sẽ được hình vuông, kéo chuột để vẽ hình elip thì sẽ vẽ được hình tròn, kéo chuột để vẽ một cung elip thì sẽ vẽ được một cung tròn.

#### Lựa chọn các đối tượng đã vẽ:

- Lựa chọn một đối tượng vẽ: nhấp chuột vào đối tượng đã vẽ, hoặc nhấp chuột vào nút Select Objects trên thanh công cụ và sau đó nhấp chuột vào đối tượng. Một đối tượng đã chọn được đánh dấu bởi 8 nút hình vuông ở xung quanh. Khi chọn một đối tượng mới thì cũng đồng thời bỏ chọn đối tượng đang chọn hiện thời.

- Lựa chọn nhiều đối tượng: ấn phím Shift đồng thời nháy chuột vào các đối tượng.

- Lựa chọn tất cả các đối tượng trong một vùng chữ nhật: nhấn nút Select Objects trên thanh công cụ, khoanh một vùng chữ nhật trên màn hình, khi đó tất cả các đối tượng nằm trong vùng này đều được chọn.

Khi một đối tượng đã được chọn, ta có thể ấn phím Delete để xóa đối tượng.

**Điều chỉnh mẫu vẽ:** Khi lựa chọn một AutoShape, ngoài 8 nút điều khiển hình vuông đôi khi còn xuất hiện các nút hình thoi màu vàng, ta có thể dùng chuột kéo các nút hình thoi này để chỉnh hình.

**Di chuyển đối tượng:** Muốn di chuyển đối tượng trước tiên ta chọn đối tượng, dùng chuột kéo đối tượng đến vị trí mới, nếu trong khi kéo ta nhấn đồng thời phím Shift thì đối tượng sẽ di chuyển theo đường thẳng góc. **Di chuyển từ từ đối tượng đã chọn:** ấn các phím mũi tên đồng thời giữ phím Ctrl. Sao chép đối tượng: chọn đối tượng, nhấn phím Ctrl đồng thời kéo chuột để di chuyển đối tượng đến vị trí mới, đối tượng ở chỗ cũ vẫn còn.

**Thay đổi kích thước đối tượng:** chọn đối tượng, nháy và kéo các nút hình vuông để co giãn đối tượng, nhả chuột.

**Viết chữ hoa vào trong một hình vẽ:** Đối với một số hình vẽ có khoảng trống bên trong (như hình chữ nhật, hình thoi, hình ellipse...) ta có thể viết thêm chữ vào trong bằng cách nháy nút phải chuột vào hình vẽ, hiện menu tắt, chọn mục Add Text, nhập văn bản.

#### 3.1.5.4. Điều chỉnh AutoShape bằng thanh công cụ Drawing

Sau khi chọn đối tượng có thể dùng nút **Line Style** để chọn lại độ rộng của nét vẽ, dùng nút **Dash Style** để chọn kiểu đường vẽ (nét liền, chấm chấm, chấm gạch...), dùng nút **Arrow Style** để chọn kiểu mũi tên (ứng với đối tượng vẽ là đường thẳng).

**Định màu tô cho đối tượng:** Chọn đối tượng, nháy vào mũi tên của nút Fill Color trên thanh Drawing, xuất hiện bảng màu như hình bên. Nháy chuột vào màu cần chọn. chọn No Fill để không tô bên trong.

Chọn mục **More Fill Colors** ..., xuất hiện hộp thoại Color, trong hộp thoại này có 2 lớp: Standard, Custom. Lớp Standard: chọn một màu chuẩn trong bảng màu. Lớp Custom: nháy chuột vào vị trí màu muốn chọn trên bảng màu, di chuyển mũi tên bên phải bảng để chọn độ sáng tối của màu vừa chọn.

Chọn mục **Fill Effects** ..., xuất hiện hộp thoại Fill Effects để quy định cách tô đặc biệt, hộp thoại này có 4 lớp: Gradient, Texture, Pattern, Picture.

- **Lớp Gradient** dùng để chọn cách tô theo kiểu dãy màu chuyển tiếp từ màu này sang màu khác: mục Colors dùng để chọn dạng màu (một màu, hai màu hay theo kiểu sẵn của Word), mục Shading styles để chọn hướng chuyển tiếp, mục Variants để chọn kiểu chuyển tiếp.

- **Lớp Texture:** tô bằng những mẫu hoa văn màu.

- **Lớp Pattern:** Chọn một mẫu tô có sẵn, với mỗi mẫu tô cần chọn mẫu của các đường kẻ trong mẫu tô và màu nền của mẫu.

- **Lớp Picture:** tô bằng một hình có sẵn, nháy vào nút Select Picture để xác định tệp chứa

hình ảnh.

**Định màu đường viền của đối tượng:** Chọn đối tượng, nhấp chuột vào mũi tên của nút Line Color trên thanh Drawing, xuất hiện một bảng màu như hình bên. Nhấp chuột vào màu cần chọn cho đường viền (có 40 màu sẵn). Chọn No Line để không có đường viền. Nếu chọn **More Line Colors** sẽ xuất hiện hộp thoại Colors gồm hai lớp Standard và Custom để lựa chọn màu ưa thích. Mục **Pattern**



**Lines:** chọn các mẫu tô đường kẻ có sẵn.

**Chọn kiểu bóng đổ:** Chọn đối tượng, nhấp chuột vào nút Shadow trên thanh công cụ, xuất hiện một bảng các kiểu bóng đổ, hãy lựa chọn một kiểu ưa thích, chọn No Shadow để bỏ bóng đổ. Nếu nhấp chuột vào mục Shadow Settings thì sẽ xuất hiện thanh công cụ Shadow Settings. Công dụng của các nút lần lượt trên thanh công cụ này là: tắt mở bóng đổ, dời bóng lên một nấc, dời bóng xuống một nấc, dịch bóng sang trái, dịch bóng sang phải, đặt lại màu cho bóng.

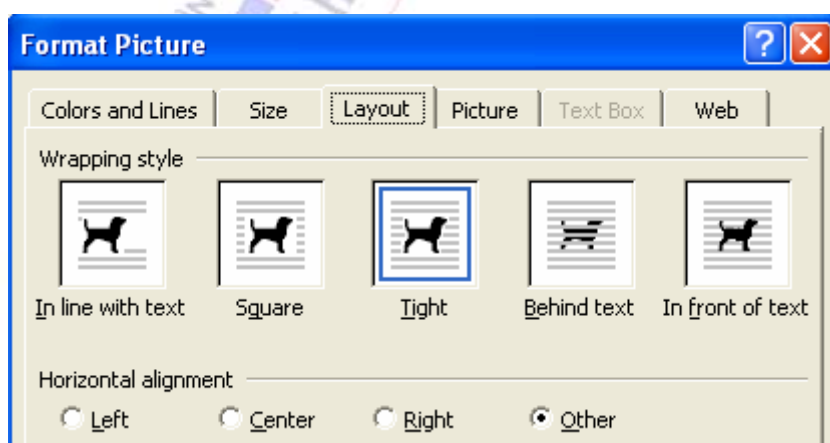


**Chuyển một hình phẳng thành hình nổi 3 chiều:** Lựa chọn hình phẳng cần chuyển, nhấp vào nút 3-D trên thanh công cụ, xuất hiện một bảng các mẫu hình 3 chiều, chọn kiểu 3 chiều thích hợp, chọn No 3-D nếu muốn hình không nổi. Nếu chọn **3-D Settings** sẽ xuất hiện một thanh công cụ cho phép điều chỉnh hình vẽ.

Các nút trên thanh công cụ 3-D Settings lần lượt từ trái sang phải có các chức năng sau: tắt mở hình nổi, xoay theo trục ngang và trục dọc (4 nút tiếp theo), định độ sâu hình nổi, chọn hướng nhìn, chọn nguồn sáng, chọn kiểu bề mặt, chọn màu phần nổi.

### 3.1.5.5. Điều chỉnh AutoShape bằng menu

Dùng hệ thống menu và hộp thoại có thể định dạng đối tượng một cách chi tiết hơn. Trước tiên ta cần lựa chọn đối tượng, dùng lệnh Format / AutoShape, xuất hiện hộp thoại AutoShape, trong hộp thoại này có 6 lớp. Dưới đây xét 3 lớp đầu tiên.



**Lớp Colors and Lines.** Mục Fill: định màu tô nền. Mục Line: chọn kiểu đường viền và màu. Mục Arrows: chọn kiểu mũi tên.

**Lớp Size** quy định kích thước của đối tượng. Mục Size and Rotate quy định bề cao, bề rộng và góc xoay của đối tượng. Mục Scale: phóng to thu nhỏ đối tượng. Ô kiểm tra Lock aspect Ratio nếu được chọn: chiều rộng và chiều cao tỷ với nhau như kích thước nguồn.

**Lớp Layout.** Mục Wrapping Style: quy định cách văn bản dồn xung quanh đối tượng đồ họa, đối tượng có thể nằm ở dưới văn bản. Mục Horizontal alignment: chọn cách đóng hình so với lề trang. Mục Advanced: khi chọn xuất hiện hộp thoại Advanced Layout với hai lớp: Picture Position, Text Wrapping. Lớp thứ nhất xác định vị trí đối tượng đồ họa, trong đó có các mục: Move Object with text (nếu chọn thì đối tượng di chuyển cùng với văn bản khi văn bản di chuyển), Lock anchor (giữ cho đối tượng và đoạn chứa nó luôn in trên cùng một trang). Lớp Text Wrapping có mục Distance from Text để chọn khoảng cách giữa đối tượng và văn bản bên ngoài.

**Sắp xếp các đối tượng chồng lên nhau.** Các đối tượng vẽ có thể chồng lên nhau và chồng lên cả Text, ta có thể sắp xếp lại thứ tự chồng nhau của các đối tượng, có thể cho văn bản đè lên đối tượng. Cách thao tác: lựa chọn đối tượng, nhấp vào nút Draw trên thanh công cụ, xuất hiện menu nhỏ, chọn mục Order, xuất hiện menu nhỏ khác. Lựa chọn các mục trong menu này: Bring to Front (mang đối tượng lên trên tất cả các đối tượng khác), Bring to Back (đưa xuống dưới cùng). Bring Forward (lên trên một bậc), Send Backward (xuống dưới một bậc), Send Backward (xuống dưới một bậc), Bring in Front of Text (mang đối tượng lên trên văn bản), Send Behind Text (mang xuống dưới văn bản).

**Nhóm các đối tượng.** Để tạo được một hình vẽ hoàn chỉnh (vẽ ngôi nhà, một biểu đồ phức tạp) ta cần tạo nhiều đối tượng, lúc này ta nên nhóm các đối tượng này thành một đối tượng duy nhất để dễ dàng quản lý. Khi cần di chuyển ta có thể di chuyển đồng thời cả nhóm. Nếu sau này ta lại muốn xử lý từng đối tượng thì phải làm động tác tách nhóm. Nhóm các đối tượng: chọn tất cả các đối tượng cần nhóm, nhấp nút Draw trên thanh Drawing, xuất hiện menu dọc, chọn mục Group. Tách nhóm các đối tượng: chọn đối tượng cần tách, nhấp nút Draw, xuất hiện menu dọc, chọn mục Ungroup.

### 3.1.5.6. Text Box

Text Box là một AutoShape, đó là một khung chữ nhật mà ta có thể đặt vào bất kỳ đâu trên trang văn bản rồi nhập nội dung vào đó. Để tạo một Text Box ta dùng lệnh Insert / TextBox, hoặc nhấp vào nút Text Box trên thanh Drawing, di chuột để định độ lớn của Text Box, nhập văn bản vào trong Text Box.

**Định dạng Text Box.** Chọn Text Box, dùng lệnh Format / Text Box, xuất hiện hộp thoại Format Text Box, hộp thoại này có các lớp: Colors and Lines, Size, Layout, Picture, Text Box, Web. Xử lý hộp thoại hoàn toàn giống với một AutoShape. Mục Internal margin của lớp Text Box quy định lề cho văn bản bên trong Text Box so với 4 cạnh của nó.

**Đổi hướng văn bản bên trong Text Box.** Chọn Text Box, dùng lệnh Format / Text Direction, xuất hiện hộp thoại Text Direction – Text Box, lựa chọn một trong ba hướng thích hợp.

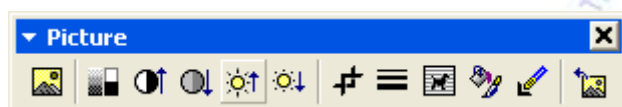
### 3.1.5.7. Chèn tệp ảnh vào văn bản đang soạn

#### Dùng lệnh Insert / Picture chèn một tệp ảnh

Ta có thể đưa các hình ảnh được lưu bởi các tệp trên đĩa (BMP, JPEG, ...) vào văn bản đang soạn thảo theo các bước: đưa con trỏ nhập đến vị trí cần chèn hình ảnh, dùng lệnh Insert /

Picture / From File, hiện hộp thoại Insert Picture, chọn thư mục chứa hình ảnh, chọn tên tệp ảnh, nhấn nút Insert để chèn hình ảnh vào chỗ con trỏ và xem nó như một ký tự (có thể nhấn vào mũi tên của nút Insert, hiện menu đọc, chọn Link to File để tạo liên kết với tệp chứa ảnh, khi tệp bị xóa thì ảnh không hiện trong văn bản).

**Định dạng một Picture.** Lựa chọn Picture, dùng lệnh View / Toolbars / Picture để xuất hiện thanh công cụ Picture:



Tên và chức năng của các nút từ trái sang phải (có thể di chuột lên trên các nút để biết tên các nút):

- **Insert Picture:** chèn thêm một ảnh mới.
- **Image Control:** Nháy nút này xuất hiện menu đọc có 4 mục: Automatic (màu theo ảnh gốc), Grayscale (ảnh xám), Black & White (ảnh đen trắng), Watermark (ảnh dạng trong suốt, có thể lấy làm trang nền cho văn bản).
- **More Contrast:** tăng độ tương phản.
- **Less Contrast:** giảm độ tương phản.
- **More Brightness:** tăng độ sáng.
- **Less Brightness:** giảm độ sáng.
- **Crop:** nhấn vào nút này và kéo các nút hình vuông xung quanh Picture để xén bớt hình, phần hình còn lại không bị thu nhỏ.
- **Line Style:** kiểu đường kẻ khung.
- **Text Wrapping:** Khi chèn một hình ảnh vào văn bản Word xử lý nó như một ký tự, để chuyển nó sang dạng hình trôi nổi ta nhấn chuột vào nút Text Wrapping, xuất hiện menu đọc, chọn mục bất kỳ. Trong menu này có 7 mục để chọn cách văn bản dồn xung quanh hình: Square, Tight, Behind Text (hình bên dưới văn bản), In front of Text, Top and Bottom, Through, Edit Wrap Points (định đường viền tùy ý bao quanh hình để văn bản dồn). Chuyển một hình ảnh từ dạng trôi nổi sang dạng như một ký tự: chọn hình ảnh, nhấn nút phải chuột vào hình ảnh, hiện menu tắt, chọn mục Format Picture, hiện hộp thoại, chọn lớp Layout, chọn dạng In Line with Text.
- **Format Picture:** khi nhấn nút này xuất hiện hộp thoại Format Picture gồm 5 lớp để định dạng hình một cách chi tiết. Lớp Picture của hộp thoại có các mục: Crop From (định một vùng cắt), Color (chọn màu cho ảnh), Brightness (chọn độ sáng), Contrast (chọn độ tương phản), Reset (trả ảnh về dạng gốc).
- **Set Transparent Color:** nhấn vào nút rồi nhấn lên một màu của bức tranh mà ta muốn màu này trở nên trong suốt.
- **Reset Picture:** trả về trạng thái ban đầu của hình.

### Chèn một hình ảnh của Microsoft Clip Gallery

Microsoft Clip Gallery có rất nhiều hình ảnh làm sẵn theo nhiều chủ đề, như Animals, Plants, Nature, Photographs, Industry, Building, ... Ví dụ như hình ảnh dưới đây lấy từ Microsoft Clip Gallery.

Muốn chèn một hình ảnh lấy từ Microsoft Clip Gallery ta dùng lệnh Insert / Picture / Clip Art, xuất hiện hộp thoại Insert ClipArt với 3 lớp, nhấp vào hình ảnh muốn chọn, hiện menu dọc, nhấp nút Insert Clip. Khi ảnh đã vào văn bản ta có thể định dạng lại như mục trên.

#### 3.1.5.8. Tạo các dòng chữ nghệ thuật nhờ WordArt

WordArt là phần chuyên dùng để tạo các hiệu ứng đặc biệt đối với văn bản tron, ví dụ có thể tạo dòng chữ ở dưới. Cách thực hiện:

định vị trí con trỏ tại nơi muốn chèn, dùng lệnh Insert / Picture / WordArt, xuất hiện hộp thoại WordArt Gallery, chọn một mẫu văn bản, nhấp OK, xuất hiện hộp thoại Edit WordArt Text, chọn Font, cỡ Font và kiểu chữ (Bold, Italic), nhập văn bản vào vùng Text, định dạng lại một đối tượng WordArt ta nhấp chuột vào phần chữ của đối tượng để chọn, dùng lệnh View / Toolbars / WordArt để xuất hiện thanh công cụ WordArt, dùng thanh công cụ này để chỉnh sửa. Muốn chỉnh sửa chi tiết hơn thì dùng hệ thống menuL chọn đối tượng WordArt, dùng lệnh Format / WordArt, xuất hiện hộp thoại Format WordArt gồm 6 lớp.



Từ trái sang phải thanh công cụ WordArt có những nút: **Insert WordArt** (hiện hộp thoại WordArt Gallery để chọn một kiểu WordArt), **Edit Text** (mở hộp thoại soạn thảo văn bản cho WordArt), **WordArt Gallery** (cho hiện hộp thoại WordArt Gallery để thay đổi kiểu WordArt), **Format WordArt** (hiện hộp thoại Format WordArt với 6 lớp để định dạng lại WordArt), **WordArt Shape** (hiện bảng các mẫu WordArt để thay đổi), **Free Rotate** (quay tự do WordArt), **Text Wrapping** (lựa chọn cách văn bản dồn xung quanh WordArt), **WordArt Same Letter Heights** (làm cho tất cả các ký tự trong văn bản có cùng độ cao), **WordArt Vertical Text** (văn bản theo chiều dọc hay ngang), **WordArt Alignment** (canh văn bản khi có nhiều dòng), **WordArt Character Spacing** (điều chỉnh khoảng cách giữa các ký tự).

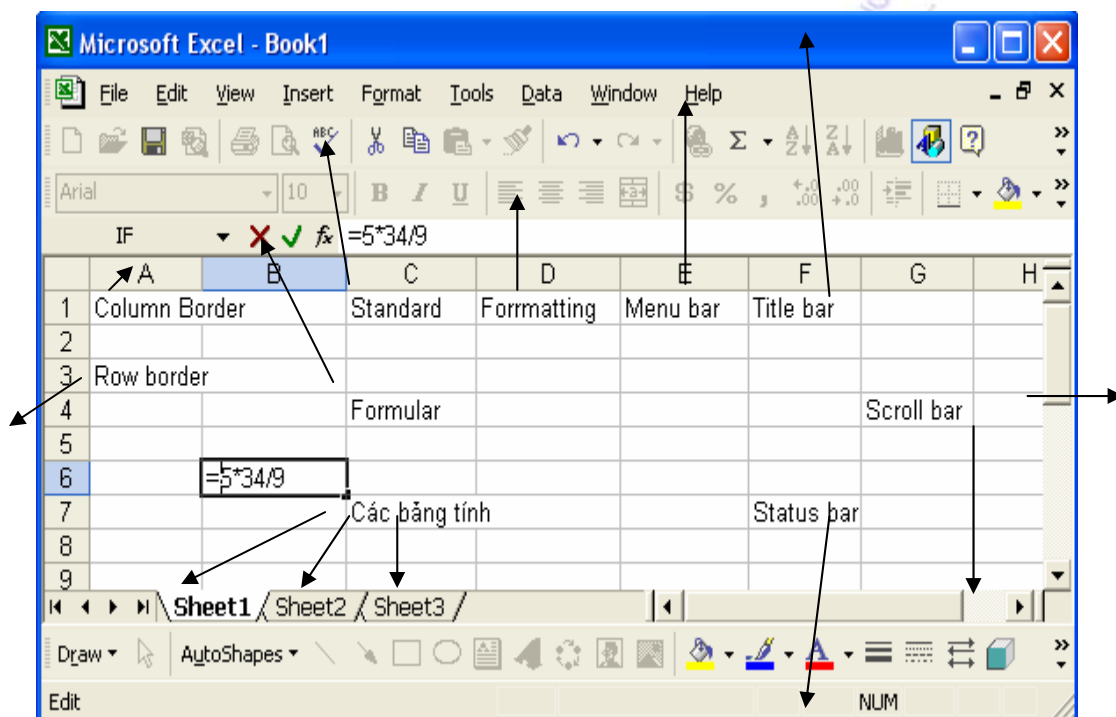
### 3.2. BẢNG TÍNH EXCEL 2000

Excel là một phần mềm chuyên dùng cho công tác kế toán, văn phòng trên môi trường Windows, thực hiện được nhiều phép tính từ đơn giản đến phức tạp. Excel tổ chức và lưu trữ thông tin dưới dạng bảng như bảng lương, bảng kế toán, bảng thanh toán, bảng thống kê, bảng dự toán, và khi có sự thay đổi dữ liệu bảng tính tự động tính toán lại theo số liệu mới. Thao tác trên bảng tính có thể tạo ra các báo cáo tổng hợp hoặc phân tích có kèm theo các ô biểu đồ, hình vẽ minh họa. Excel có nhiều công dụng mạnh để tính toán tài chính, giải các bài toán thống kê và các bài toán tối ưu.

### 3.2.1. Các khái niệm cơ bản

#### 3.2.1.1. Khởi động Excel và màn hình làm việc

Để khởi động Excel dùng lệnh Start / Programs / Microsoft Excel, màn hình làm việc có dạng như sau:



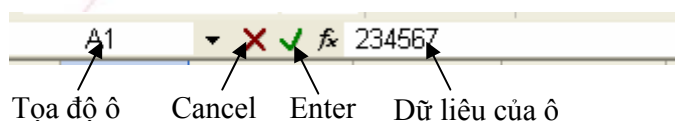
Màn hình làm việc của Excel có những thành phần chính sau:

- **Menu Bar** (menu ngang chính) liệt kê tên các mục lệnh chính của Excel. Mỗi mục ứng với một menu dọc, để mở một menu dọc ta chỉ cần nháy chuột vào tên mục.

- **Thanh Standard** (thanh chuẩn) chứa một số lệnh thông dụng của Excel dưới dạng các nút có biểu tượng, các lệnh này có thể truy xuất trực tiếp bằng chuột. Trên thanh Standard có các nút: New, Open, Save, Print, Print Preview, Cut, Copy, Paste, Format Painter... Để hiện tên nút ta chỉ cần đặt con trỏ chuột lên trên nút, không nháy chuột. Nếu chưa có khả năng này, hãy dùng lệnh Tools / Customize / Options, đánh dấu kiểm vào ô Show ScreenTips on Toolbars.

- **Thanh Formatting** (thanh định dạng) chứa các lệnh dưới dạng các nút có biểu tượng để định dạng dữ liệu của bảng tính như kiểu, loại font, cỡ font, căn lề,... Trên thanh có các ô điều khiển Font, Font Size; các nút Bold, Italic, Underline, Align left, Center, Align right...

- **Thanh Formula** (thanh công thức) gồm các ô: Name Box (hiển thị tọa độ ô hiện hành), Cancel (hủy bỏ), Enter (chấp nhận), Formula bar (nội dung dữ liệu của ô hiện hành).





• **Thanh Drawing:** các nút hoàn toàn giống của Word. Cho hiện hay không cho hiện các thanh công cụ ở trên bằng lệnh View / Toolbars, chọn tên thanh công cụ.

• **Workbook** (Sổ tính) là cửa sổ chứa nội dung tệp. Tên tệp hiện trên thanh tiêu đề cửa sổ với phần mở rộng định sẵn là XLS. Tên tệp Workbook mặc nhiên là Book# (# là số thứ tự tương ứng với những lần mở tệp). Các thành phần của Workbook là:

- **Đường viền ngang (Column Border)** ghi ký hiệu cột từ trái sang phải theo chữ cái A, B, C, ..., Y, Z, AA, AB, ..., IV. Cột (column) là một tập hợp những ô theo chiều dọc. Độ rộng mặc nhiên là 9 ký tự (có thể thay đổi trị số này từ 0 đến 255). Có tổng cột 256 cột.

- **Đường viền dọc (Row Border)** ghi số thứ tự dòng từ trên xuống dưới. Dòng (row) là một tập hợp những ô theo chiều ngang. Chiều cao mặc nhiên là 12.75 pixel (có thể thay đổi trị số này từ 0 đến 409). Có tổng cộng 65536 dòng.

- **Ô (cell)** là giao của một dòng với một cột. Địa chỉ của một ô xác định bởi cột trước dòng sau. Ví dụ: B6 là địa chỉ ô nằm trên cột B, dòng thứ 6. Ô hiện hành (Select Cell) là ô có khung viền quanh. Một ô có thể chứa tới 32.000 ký tự.

- **Sheet (Bảng tính)** là một bảng gồm có 256 cột và 65536 dòng. Tên bảng tính mặc nhiên là Sheet# (# là số thứ tự). Một tệp Workbook có nhiều Sheet, được liệt kê Sheet1, Sheet2, Sheet3, ... Trong màn hình trên đang chọn bảng tính Sheet1 của tệp Book3. Ta có thể quy định số Sheet trong một tệp Workbook bằng lệnh Tools / Options, chọn lớp General, chọn số lượng Sheet trong mục **Sheet in New Workbook** (có thể lên tới 255). Trong lớp General ta cũng có thể chọn phong chữ và cỡ chữ ngầm định cho các Sheet, ví dụ **Vn Time, 12**.

• **Scroll bar** (hai thanh trượt ở bên phải và bên dưới cửa sổ) dùng để hiển thị những phần bị che khuất của bảng tính trên màn hình.

• **Status Bar** (dưới đáy cửa sổ Microsoft Excel) dùng chứa chế độ làm việc hiện hành hay ý nghĩa lệnh hiện hành của bảng tính và các tình trạng hiện hành của hệ thống như NumLock, Capslock, ... Các chế độ làm việc thông thường gồm: Ready (sẵn sàng nhập dữ liệu), Enter (đang nhập dữ liệu), Point (đang ghi chép công thức tham chiếu đến một địa chỉ), Edit (đang điều chỉnh dữ liệu hay công thức trong ô hiện hành, chọn chế độ này bằng cách di chuyển đến ô muốn điều chỉnh và gõ phím F2).

• Trên bảng tính con trỏ chuột là hình chữ thập rộng, con trỏ ô là một khung chữ nhật bao xung quanh một ô, ô có con trỏ ô gọi là ô hiện hành.

Việc sử dụng các thanh công cụ, các menu và nhận trợ giúp trong Excel cũng giống với Word 2000.

### 3.2.1.2. Các kiểu dữ liệu

Trong một ô chỉ có thể chứa một kiểu dữ liệu. Kiểu dữ liệu phụ thuộc vào ký tự đầu tiên gõ vào.

• **Kiểu số (Number).** Ký tự đầu tiên gõ vào là các chữ số từ 0 đến 9, các dấu + - . ( \$. Một số được nhập vào mặc nhiên là dạng General, sau đó có thể định dạng trình bày số lại theo ý muốn bằng lệnh Format / Cells.

• **Kiểu chuỗi** (Text). Ký tự đầu tiên gõ vào là các ký tự từ A đến Z, các ký tự canh biên như sau ‘ “ ^ \ . Ký tự ‘ để đóng các ký tự trong ô về bên trái. Ký tự “ để đóng các ký tự trong ô về bên phải. Ký tự ^ để canh các ký tự trong ô vào giữa. Ký tự \ để lặp lại ký tự theo sau nó cho đến hết chiều rộng ô. Các ký tự canh biên chỉ có tác dụng khi người sử dụng có chỉ định: Tools / Option, chọn lớp Transition, trong đó chọn mục: Transition Navigation Keys.

• **Kiểu công thức** (Formula). Ký tự đầu tiên gõ vào là dấu = hoặc +. Kết quả trình bày trong ô không phải là các ký tự gõ vào mà chính là giá trị của công thức đó. Ví dụ: =5\*20, =A1+A2 hoặc +5\*20, +A1+A2. Trong thành phần của một công thức có thể gồm có: số, chuỗi (phải được đặt trong cặp nháy kép), tọa độ ô, tên vùng, các toán tử, các loại hàm.

Các toán tử sử dụng trong công thức:

- Toán tử tính toán: + (cộng) – (trừ) \* (nhân) / (chia) ^ (lũy thừa) % (phần trăm)
- Toán tử chuỗi: & (nối chuỗi). Ví dụ, = “Hanoi” & “Vietnam”
- Toán tử so sánh: = (bằng) <> (không bằng) > (lớn hơn) >= (lớn hơn hay bằng) < (nhỏ hơn) <= (nhỏ hơn hay bằng)

Độ ưu tiên của các toán tử trong công thức: các nhóm biểu thức trong (...), lũy thừa, nhân, chia, cộng, trừ.

Khi nhập văn bản hay công thức dài muốn xuống dòng trong ô hãy ấn Alt + Enter. Nếu trong bảng tính có công thức mà ta muốn các ô hiện công thức thì dùng lệnh Tools / Options, chọn lớp View, đánh dấu kiểm vào mục Formulas (muốn hiện lại giá trị thì không chọn mục này).

### 3.2.1.3. Cách nhập dữ liệu

Nháy chuột vào ô cần nhập, nhập dữ liệu (theo quy ước từng loại dữ liệu), kết thúc nhập bằng cách gõ phím Enter (hoặc nháy vào nút Enter trên thanh Formula, hoặc dùng các phím mũi tên di chuyển con trỏ sang ô khác). *Ghi đè dữ liệu mới* vào một ô: nháy chuột vào ô, gõ dữ liệu mới và ấn Enter. *Sửa dữ liệu một ô*: nháy chuột vào ô, gõ F, dùng phím mũi tên di chuyển con trỏ chèn (là một dấu vạch đứng) tới nơi cần để sửa, muốn xóa ký tự dùng các phím Delete và Backspace.

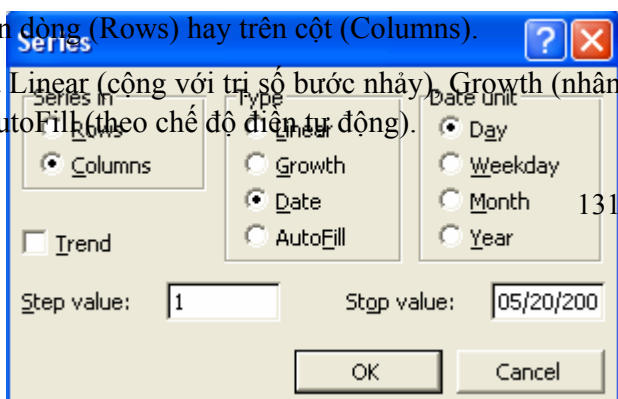
**Cách phím di chuyển con trỏ ô:** ↑ (lên một dòng), ↓ (xuống một dòng), → (sang phải một cột), ← (sang trái một cột), PgUp (lên một trang màn hình), PgDn (xuống một trang màn hình), Alt+PgUp (sang trái một trang màn hình), Alt+PgDn (sang phải một trang màn hình), Home (về đầu một dòng), Ctrl+Home (về ô A1). Ngoài ra có thể nháy chuột tại một ô để di chuyển con trỏ ô đến đó.

### 3.2.1.4. Điền dãy số tự động

Điền dãy số tự động là việc điền một dãy số vào một hàng ô hay một cột ô. Thao tác: gõ giá trị số bắt đầu vào ô đầu tiên, chọn vùng cần điền số, thực hiện lệnh: Edit / Fill / Series, hộp đối thoại Series hiện ra. Trong hộp thoại này ta cần điền các mục:

- Mục Series: chọn việc đánh chuỗi số trên dòng (Rows) hay trên cột (Columns).

- Mục Type: để chọn kiểu điền dữ kiện là Linear (công với trị số bước nhảy), Growth (nhân với trị số bước nhảy), Date (theo dạng ngày), AutoFill (theo chế độ điền tự động).



- Mục Step Value: chọn trị số bước nhảy.
- Mục Stop Value: chọn trị số kết thúc.

### 3.2.1.5. Thao tác trên tệp

Một tệp Workbook trong Excel có phần mở rộng là XLS và có thể có nhiều Sheet (bảng tính), nhiều biểu đồ.

- Lưu trữ tệp Workbook mới lên đĩa: thực hiện lệnh File / Save. Lưu trữ tệp Workbook cũ lên đĩa với tên mới : dùng File / Save As. Trong cả hai trường hợp trên máy đều hiện hộp thoại Save As (hoàn toàn giống với hộp thoại trong Word), chọn tên thư mục trong Save In, gõ vào tên tệp mới trong hộp File Name, trong hộp Save As Type chọn Microsoft Excel Workbook, nhấn nút Save.

- Lưu trữ tệp lên đĩa với tên cũ bằng cách: nhấn chuột vào biểu tượng Save trên thanh Standard, hoặc thực hiện lệnh File / Save.

- Mở một tệp Workbook đã có trên đĩa: thực hiện lệnh File / Open, hoặc nhấn chuột vào biểu tượng Open trên thanh Standard, chọn ổ đĩa và chọn thư mục, chọn tên tệp cần mở.

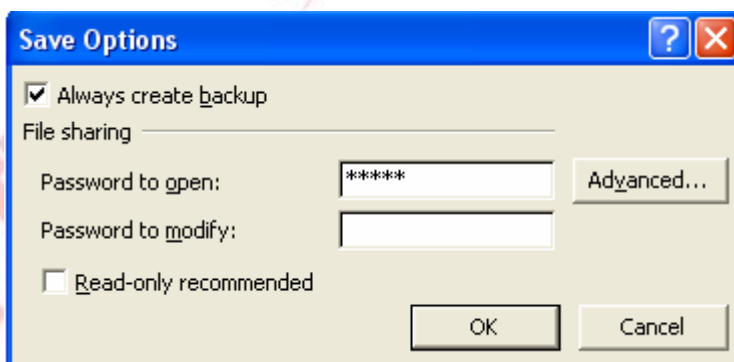
- Tạo một tệp Workbook mới: thực hiện lệnh File / New, hoặc nhấn chuột vào biểu tượng New trên thanh Standard.

- Đóng một tệp Workbook: thực hiện lệnh File / Close.

### 3.2.2. Định dạng dữ liệu

#### 3.2.2.1. Thay đổi độ rộng cột và chiều cao dòng

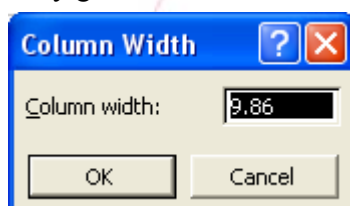
- Đổi độ rộng một cột: rê chuột trên đường gạch đứng giữa hai cột (tại đường viền ngang phía trên của bảng tính) để thay đổi độ rộng cột bên trái.



- Đổi chiều cao một dòng: rê chuột trên đường gạch ngang giữa hai dòng (tại đường viền dọc bên trái của bảng tính) để thay đổi chiều cao của dòng bên trên.

- Đổi độ rộng nhiều cột: chọn một số ô của những cột cần thay đổi độ rộng, thực hiện lệnh Format / Column / Width, hộp thoại Column Width hiện ra, gõ vào độ rộng cột cần thay đổi rồi chọn OK.

- Đổi chiều cao nhiều dòng: chọn một số ô của những dòng cần thay đổi chiều cao, thực hiện lệnh Format / Row / Height, hộp thoại Row Height hiện ra, gõ vào chiều cao dòng cần thay đổi rồi chọn OK hay gõ Enter.



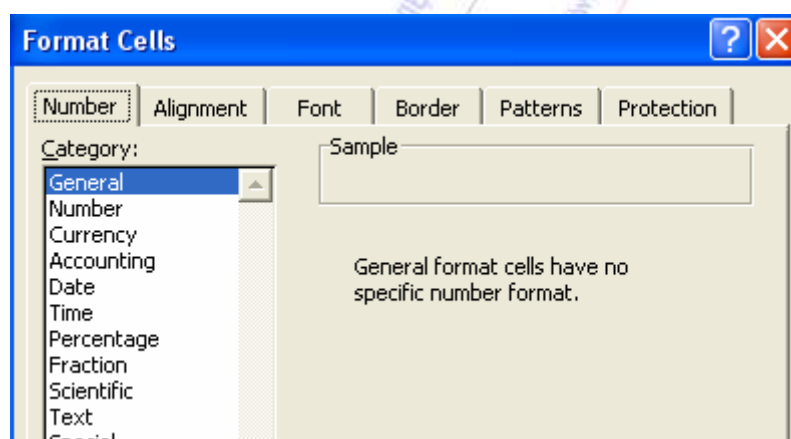
### 3.2.2.2. Định dạng dữ liệu số

Sau khi nhập dữ liệu, tính toán, ... ta có thể trình bày lại bảng tính như chọn kiểu thể hiện số liệu, chọn đơn vị tính thích hợp, thay đổi kiểu chữ, cỡ chữ, nhấn mạnh nội dung, số liệu quan trọng...

Một số khi nhập vào một ô tự động được chỉnh phải và được hiển thị phụ thuộc hai thành phần: Lớp (Category) và dạng. Các lớp gồm Number, Date, Time, Percent,... Trong một lớp có nhiều dạng. Ví dụ lớp Date có các dạng: M/D/YY, D-MMM-YY,...

Các bước thực hiện định dạng hiển thị số: chọn vùng dữ liệu để định dạng số, thực hiện lệnh Format / Cells, hộp thoại Format Cells hiện ra, chọn lớp Number, trong mục Category hãy chọn lớp thích hợp, nháy chuột tại mẫu định dạng mong muốn trong khung bên phải hộp thoại (tương ứng là ví dụ mẫu trong mục Sample), chọn OK.

Thay cho việc vào hệ thống menu ta cũng có thể nháy chuột vào các nút Currency style, Percent style, Comma style, Increase decimal, Decrease decimal trên thanh Formatting để định dạng lại dữ liệu số.



### 3.2.2.3. Định dạng dữ liệu chữ

Dữ liệu chữ đã nhập trong các ô có thể định dạng lại theo các thành phần: Font (nét chữ), Font Style (nghiêng, đậm,...), Size (kích thước chữ), Color (màu của chữ). Các bước thực hiện định dạng ký tự: chọn vùng dữ liệu để định dạng, thực hiện lệnh Format / Cells, xuất hiện hộp thoại Format Cells, chọn lớp Font (các mục cũng giống như trong Word 2000), chọn Font, Font Style, Size, Underline, Color thích hợp cho chữ.

### 3.2.2.4. Quy định vị trí của dữ liệu trong các ô

Sau khi nhập ta có thể quy định lại vị trí hiển thị dữ liệu trong ô (đóng) bằng cách: chọn vùng dữ liệu để định dạng, thực hiện lệnh Format / Cells, xuất hiện hộp thoại Format Cells, chọn lớp Alignment.

Trong hộp thoại này có các khả năng lựa chọn:

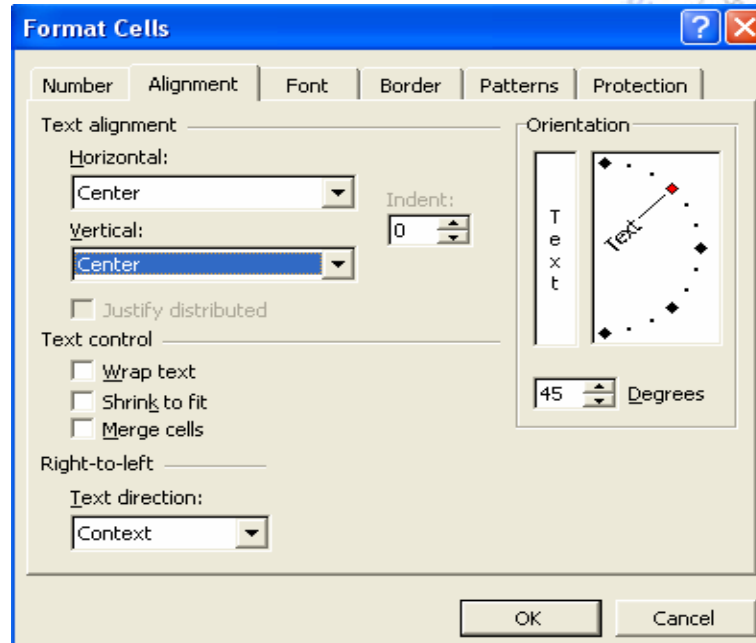
- **Khung Horizontal** để lựa chọn cách điều chỉnh ngang dữ liệu trong từng ô: General (giữ nguyên dữ liệu như khi nhập vào từ bàn phím), Left (điều chỉnh thẳng mép trái), Center (điều chỉnh giữa ô), Right (điều chỉnh thẳng mép phải), Fill (điền toàn ô bởi các ký tự có trong ô đó),

Justify (điều chỉnh thẳng hai mép), Center across selection (điều chỉnh giữa qua một dãy ô).

- **Khung Vertical** để lựa chọn điều chỉnh dọc dữ liệu trong từng ô: Top (lên phía trên), Bottom (xuống phía dưới), Center (dữ liệu cân giữa), Justify.

- **Khung Orientation** để lựa chọn hình thức trải dữ liệu trong ô: ngang, dọc hay nghiêng.

- **Ô kiểm tra Wrap Text** nếu được đánh dấu thì độ rộng cột cố định, dữ liệu nhập vào tự động dàn qua nhiều dòng.



Chú ý, việc chọn Font, Style, Size của chữ và cách dóng cũng có thể thực hiện nhờ các nút Font, Font Size, Bold, Italic, Underline, Align Left, Center, Align Right trên thanh Formatting.

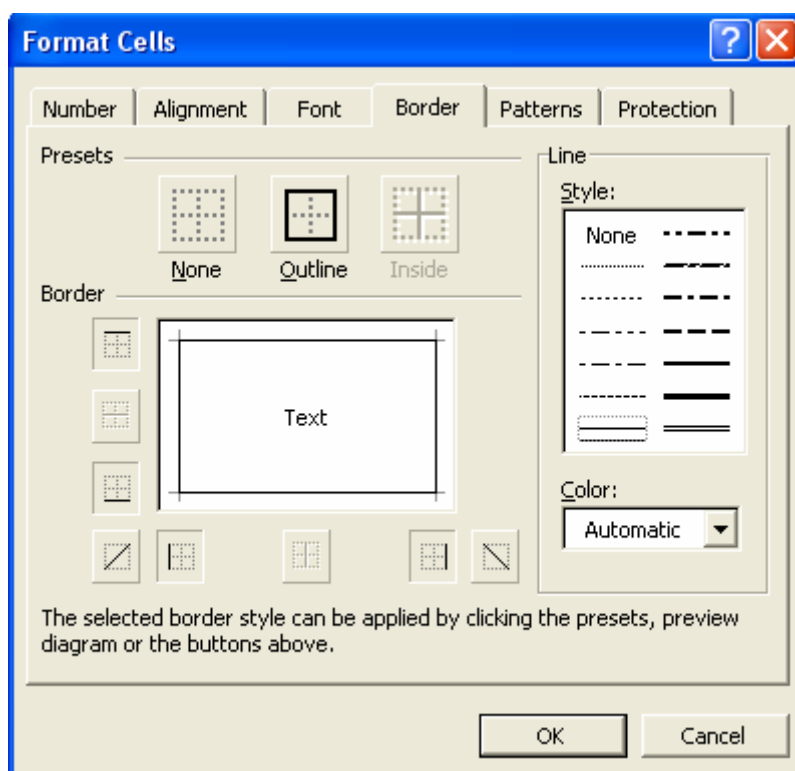
### 3.2.2.5. Tạo các đường kẻ theo vùng ô đã chọn

Muốn kẻ các đường ngang dọc trong bảng tính ta tiến hành: chọn vùng ô cần kẻ khung, dùng lệnh Format / Cells, xuất hiện hộp thoại Format Cells, chọn lớp Border.

Xử lý hộp thoại:

- Nút **Outline**: vẽ viền quanh vùng ô đã chọn. Nút **Inside**: kẻ các đường kẻ ngang dọc bên trong vùng ô đã chọn. Nút **None**: hủy bỏ các đường kẻ ô. Tám nút trong khung **Border**: kẻ các đường ngang dọc, chéo trong vùng ô, ở giữa các nút này là các mẫu tác dụng của các đường kẻ.

- Khung **Style** để chọn một trong 13 mẫu đường kẻ (kẻ đơn, đúp, nét liền hay đứt đoạn). Nút **Color**: chọn màu đường kẻ.



Ta cũng có thể tạo các đường kẻ cho vùng ô đã chọn bằng cách nhảy vào nút Borders trên thanh Formatting, khi đó xuất hiện một bảng gồm 13 mẫu kẻ (không kẻ nghĩa là xóa đường kẻ cũ), lựa chọn cách kẻ thích hợp. Các đường kẻ khung sẽ được in ra khi in bảng tính.

**Bật tắt lưới ô của bảng tính.** Lưới ô trong bảng tính giúp cho việc nhìn các ô được rõ ràng, khi in nó không được in ra. Để bật (hay tắt) lưới ô dùng lệnh: Tools / Options, xuất hiện hộp thoại Option, chọn lớp View, đánh dấu kiểm (hay bỏ chọn) mục Gridlines

#### 3.2.2.6. Định dạng nền dữ liệu

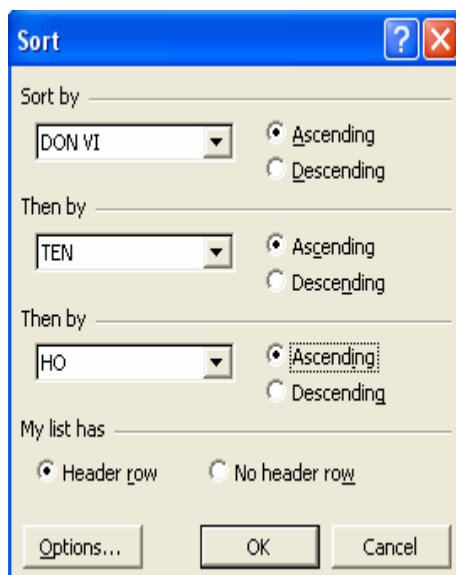
Ta có thể tô nền của các ô theo mẫu tô và màu tô nhất định. Cách tiến hành: chọn vùng để định dạng nền dữ liệu, dùng lệnh Format /Cells, hộp thoại Format Cells xuất hiện, chọn lớp Patterns. Xử lý hộp thoại: chọn mẫu tô trong mục Pattern, chọn màu tô trong mục Color, chọn OK.

#### 3.2.2.7. Sắp xếp dữ liệu

Trong Excel có thể sắp xếp số liệu trong một phạm vi được chọn một cách độc lập với các ô ngoài khu vực chọn. Việc sắp xếp có thể thực hiện trên hàng ngang hoặc cột dọc. Excel cho phép sắp xếp dữ liệu tối đa theo ba khóa. Ví dụ, nếu một danh sách lương sắp xếp theo từng đơn vị, trong mỗi đơn vị sắp xếp theo Tên, cùng tên sắp xếp theo Họ thì khóa thứ nhất là dữ liệu cột Đơn vị, khóa thứ hai là dữ liệu cột Tên, khóa thứ ba là dữ liệu cột Họ.

Các bước để sắp xếp dữ liệu: chọn vùng dữ liệu muốn sắp xếp, thực hiện lệnh Data / Sort, hộp thoại Soft hiện ra. Xử lý hộp thoại: chọn tiêu đề cho khóa sắp xếp thứ nhất và thứ tự sắp xếp (Ascending: tăng, Descending: giảm), chọn tiêu đề cho khóa sắp xếp thứ hai và thứ tự sắp xếp nếu có, chọn tiêu đề cho khóa sắp xếp thứ ba và thứ tự sắp xếp nếu có, chọn OK.

Chú ý, muốn chỉ định sắp xếp theo hàng hay sắp xếp theo cột, thì trong hộp thoại Sort chọn Options, hộp thoại Sort Options xuất hiện. Trong hộp thoại này có hai lựa chọn: Sort Top to Bottom (xếp theo hàng), Sort Left to Right (xếp theo cột).



### 3.2.3. Các hàm cơ bản

Excel có một số hàm mẫu (Function Wizard) dùng tiện lợi và đơn giản, ví dụ công thức  $=A3+A4+A5+A6+A7$  có thể thay bằng hàm SUM (A3:A7). Dạng thức tổng quát của hàm:

=<Tên hàm>(Danh sách đối số)

<Tên hàm> là tên hàm mẫu do Excel quy định. Ví dụ: SUM, AVERAGE, MAX, ... <DS đối số> có thể là các trị số, dãy các ô, địa chỉ ô, tên vùng, công thức, tên hàm. Chú ý, hàm phải bắt đầu bởi dấu bằng (=), tên hàm không phân biệt chữ thường và chữ hoa. Đối số phải đặt trong ngoặc đơn (), giữa các đối số phân cách nhau bởi dấu phẩy.

#### 3.2.3.1. Các hàm số học và đếm

- **ABS(N)**: cho giá trị tuyệt đối của biểu thức số N. Ví dụ =ABS(-25) cho 25.
- **SQRT(N)** cho trị là căn bậc hai của số N (N>0). Ví dụ =SQRT(25) cho 5.
- **INT(N)** cho giá trị là phần nguyên của biểu thức số N. Ví dụ =INT(236.26) cho 236.
- **PI()** cho trị là số Pi (3.141593).
- **MOD(N,M)** cho phần dư của phép chia nguyên N cho M. Ví dụ =MOD(10, 3) cho kết quả là 1.

• **ROUND(biểu\_thức\_số, n)**: làm tròn giá trị của biểu\_thức\_số đến n số lẻ. Nếu n>0: làm tròn về bên phải cột thập phân. Nếu n<0: làm tròn về bên trái cột thập phân.

Ví dụ: =Round(333333, -3) cho giá trị 333000 (làm tròn đến hàng ngàn)

=Round(35123.374, 2) cho giá trị 35123.37

=Round(12345.5432, 0) cho giá trị 12346

• **SUM**(danh sách các trị): tính tổng của các giá trị có trong danh sách. Ví dụ dữ liệu trong các ô B1, B2, B3, B6 lần lượt là 4, 8, 3, 6. Công thức ở ô B7 là =SUM(B1:B3, B6). Giá trị trả về trong ô B7 là 21.

• **AVERAGE**(danh sách các trị): tính trung bình cộng của các giá trị có trong danh sách. Ví dụ dữ liệu trong các ô B1, B2, B3, B4 lần lượt là 4, 8, 6, 2. Công thức ở ô B6 là =AVERAGE(B1:B4, 10). Giá trị trả về trong ô B6 là 6.

• **MAX**(danh sách các trị): tìm giá trị số học lớn nhất của các giá trị có trong danh sách. Ví dụ dữ liệu trong các ô B1, B2, B3, B4 lần lượt là 4, 8, 6, 2. Công thức MAX(B1:B4) cho giá trị là 8.

• **MIN**(danh sách các trị): tìm giá trị số học nhỏ nhất của các giá trị có trong danh sách. Ví dụ dữ liệu trong các ô B1, B2, B3, B4 lần lượt là 4, 8, 6, 2. Công thức MIN(B1:B4) cho giá trị là 2.

• **COUNTA**(danh sách các trị): cho số các ô chứa dữ liệu trong danh sách. Ví dụ, công thức =COUNTA(-2, "VTD", 5, 8) cho 4, =COUNTA(A2:F2) cho 6.

	A	B	C	D	E	F
1	4.4	1.1	9.9	3.3	2.2	5.5
2	1	AB C	2	3	XY Z	4

• **COUNT**(danh sách các trị): cho số các ô chứa giá trị số trong danh sách. Ví dụ =COUNT(-2, "VTD", 5, 8) cho kết quả 3, COUNT(A2:F2) cho kết quả 4.

• **RANK**(x, danh sách): xác định thứ hạng của trị x so với các giá trị trong Danh sách (thứ hạng xếp theo giá trị giảm dần). Trị x và danh sách phải là các giá trị số, nếu không sẽ gây ra lỗi #VALUE!. Trị x phải rơi vào một trong các giá trị của danh sách, nếu không sẽ gây ra lỗi #N/A. Ví dụ, Rank(3.3, A1:F1) cho giá trị là 4. Một trường hợp ứng dụng của hàm: trong bảng điểm cuối học kỳ của một lớp dựa vào cột Điểm trung bình các môn học tính cột Xếp thứ (học sinh xếp thứ mấy trong lớp về học lực).

### 3.2.3.2. Các hàm ngày tháng

Dữ liệu ngày (Date) là một dạng đặc biệt của dữ liệu số. Ta thường nhập dữ liệu ngày ở dạng M/D/YY, ví dụ 1/25/02 (ngày 25 tháng 1 năm 2002, năm nhập là 02 thì Excel hiểu là năm 2002). Sau khi ô đã có dữ liệu ngày, ta có thể chọn các ô, dùng lệnh Format / Cells, xuất hiện hộp thoại Format Cells, chọn lớp Number, chọn mục Date, chọn các kiểu dữ liệu ngày hay dùng như: 25-Jan-02 và March 4, 2002.

Các hàm đối với dữ liệu ngày:

• **DAY**(dữ liệu kiểu ngày) cho giá trị ngày của dữ liệu kiểu ngày. Ví dụ =DAY("1/15/02") cho 15, công thức DAY(B5) cho 25 nếu B5 chứa giá trị kiểu ngày 1/25/01.



• **MONTH**(dữ liệu kiểu ngày) cho giá trị tháng của dữ liệu kiểu ngày. Ví dụ =MONTH("11/30/03") cho kết quả 11.

• **YEAR**(dữ liệu kiểu ngày): cho giá trị năm của dữ liệu kiểu ngày. Ví dụ =YEAR("11/19/05") cho 2005.

### 3.2.3.3. Các hàm logic

• **IF**(biểu thức logic, biểu thức 1, biểu thức 2): nếu biểu thức logic là đúng thì hàm cho giá trị là <biểu thức 1>, trái lại cho giá trị là <biểu thức 2>. Các **biểu thức 1** và **biểu thức 2** cũng có thể là những hằng trị, biểu thức (chuỗi, số, logic) và cũng có thể là một hàm IF khác. Ví dụ, =IF(A1>=300, 50, 100).

• **AND**(điều kiện 1, điều kiện 2, ...): cho giá trị đúng khi mọi điều kiện nêu trong danh sách đều cho trị đúng. Ví dụ =AND(3>2, 5<8) cho giá trị TRUE, =AND(TRUE, FALSE) cho giá trị FALSE.

• **OR**(điều kiện 1, điều kiện 2, ...): cho giá trị đúng khi có bất kỳ một điều kiện nêu trong danh sách đều cho giá trị đúng. Ví dụ, =OR(3>2, 5=8) cho giá trị TRUE, =OR(1+1=3, 2+3=6) cho giá trị FALSE.

• **NOT**(điều kiện): cho trị đúng nếu điều kiện sai và cho trị sai nếu điều kiện đúng.

### 3.2.3.4. Các hàm xử lý chuỗi

• **LEFT**(TEXT, N) cho trị là chuỗi con của chuỗi TEXT được tính từ trái sang phải N ký tự. Ví dụ, =LEFT("ABCD", 2) cho "AB".

• **RIGHT**(TEXT, N) cho trị là chuỗi con của chuỗi TEXT được tính từ phải sang trái N ký tự. Ví dụ, =RIGHT("ABCD", 2) cho "CD".


• **LEN**(TEXT) cho độ dài của chuỗi TEXT. Ví dụ, =LEN("ABCD") cho 4.

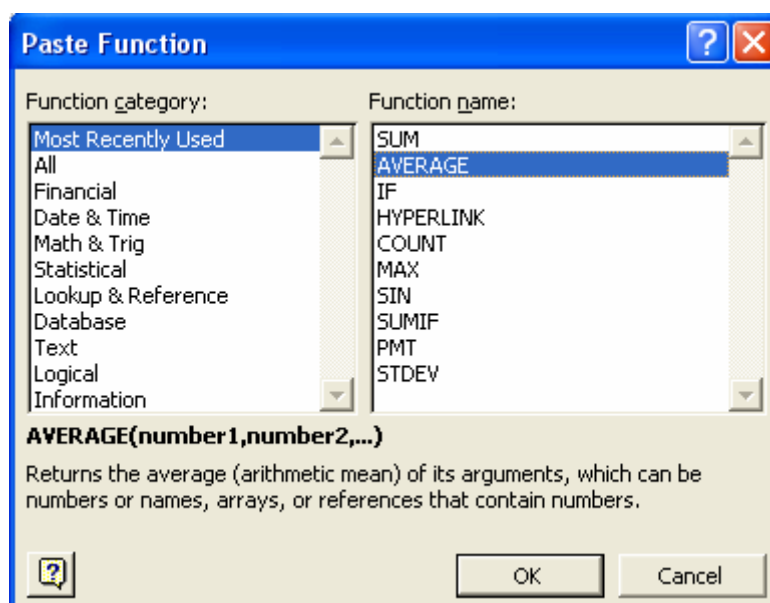
• **LOWER**(TEXT) chuyển chuỗi TEXT thành chuỗi chữ thường.

• **UPPER**(TEXT) chuyển chuỗi TEXT thành chuỗi chữ hoa.

• **PROPER**(TEXT): chuyển các ký tự đầu từ của chuỗi TEXT thành chữ hoa. Ví dụ, =PROPER("trung tam") cho kết quả là chuỗi "TRUNG TAM".

### 3.2.3.5. Chọn hàm mẫu từ hệ thống menu

Một hàm mẫu có thể nhập vào ô qua hệ thống menu và hộp thoại bằng cách: đặt con trỏ tại ô cần nhập hàm mẫu, thực hiện lệnh Insert / Function (hoặc nhấn nút trên thanh  Standard), xuất hiện hộp thoại Paste Function. Chọn nhóm hàm cần thực hiện trong khung Function Category (ví dụ All), chọn tên hàm cần thực hiện trong khung Function Name (ví dụ AVERAGE), nhấn nút OK, xuất hiện hộp thoại mới, vào địa chỉ vùng ô để xác định đối số của hàm, nhấn nút OK, kết quả tính hàm xuất hiện trên ô đã chọn.



### 3.2.4. Biểu đồ bảng tính và chèn hình ảnh

Biểu đồ là đồ thị biểu diễn dữ liệu bảng tính. Các biểu đồ làm cho dữ liệu của bảng tính phức tạp trở thành trực quan và dễ hiểu hơn. Biểu đồ biến đổi dữ liệu từ các hàng và các cột trên bảng tính thành hình khối, đồ thị,... Biểu đồ có các trục, trên đó có các giá trị tỷ lệ tương đương với giá trị dữ liệu trên bảng tính.

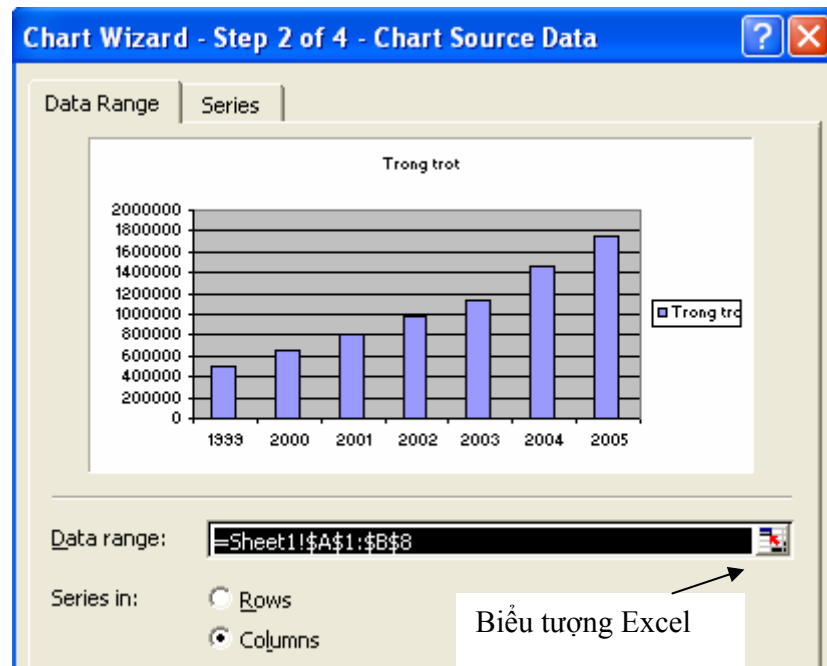
#### 3.2.4.1. Các bước tạo biểu đồ

Xét bảng tính dưới đây về thu nhập của một đơn vị sản xuất nông nghiệp trên ba ngành nghề: trồng trọt, chăn nuôi và thủ công từ năm 1999 đến năm 2005 (đơn vị là ngàn đồng).

	A	B	C	D
1		<b>Trồng trọt</b>	<b>Chăn nuôi</b>	<b>Thủ công</b>
2	1999	502000	375345	234765
3	2000	657456	532456	245785
4	2001	798453	745678	263456
5	2002	978997	845667	221678
6	2003	1134567	747897	245745
7	2004	1454788	534678	254865
8	2005	1756878	423567	276345

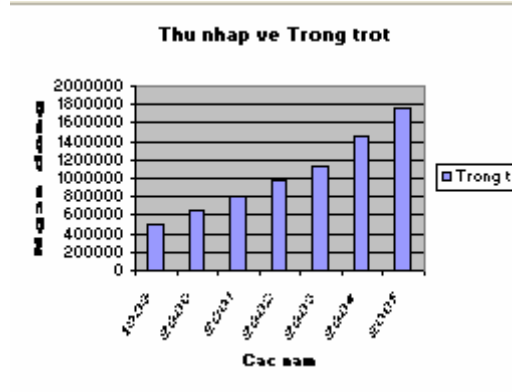
Tạo đồ thị dạng cột từ dữ liệu của Trồng trọt. Các bước thực hiện:

- Dùng chuột chọn vùng A1:B8
- Nháy vào biểu tượng Chart Wizard trên thanh công cụ để hiển thị hộp thoại **Chart Wizard - Step 1 of 4 - Chart Type**, chọn lớp Standard Types, trong mục Chart Type chọn Column, trong mục Chart sub-type chọn một trong các mẫu con (chọn mẫu đầu tiên).
- Nháy nút Next, hiện hộp thoại **Chart Wizard - Step 2 of 4 - Chart Source Data**.



Trong hộp thoại này cho dạng đồ thị ban đầu và gồm hai lớp. Lớp Data Range dùng để chọn vùng chứa dữ liệu cho biểu đồ, xác định chuỗi dữ liệu bố trí theo hàng hay theo cột. Lớp Series: chọn tên biểu đồ, chuỗi số làm biểu đồ, chọn nhãn cho trục X. Khi nhấp vào biểu tượng Excel trong các hộp văn bản của hộp thoại thì hộp thoại biến thành khung cửa sổ nhỏ, bảng tính hiện lên để ta chọn vùng thích hợp, nhấp biểu tượng Excel trong cửa sổ nhỏ để về hộp thoại.

- Nháy tiếp Next để hiện hộp thoại **Chart Wizard - Step 3 of 4 - Chart Options**. Gõ vào hộp Chart Title tiêu đề của biểu đồ (ví dụ **Thu nhập về Trồng trọt**), nhập thuyết minh cho trục X (ví dụ **Các năm**) và thuyết minh cho trục Y (ví dụ **Ngàn đồng**).



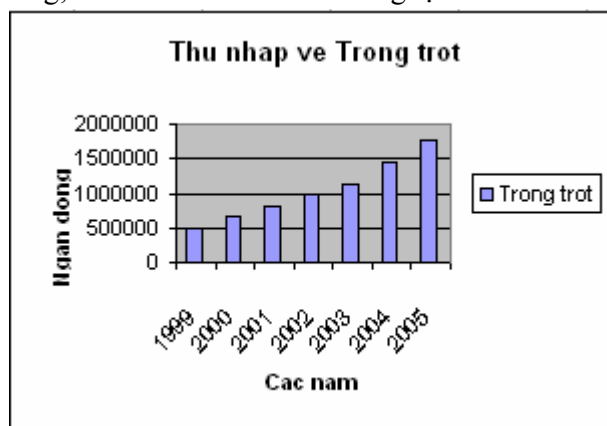
- Nháy nút Next để đến màn hình cuối cùng. **Chart Wizard - Step 4 of 4 - Chart Location**. Hộp thoại này để ta quyết định, hoặc là sẽ chèn đồ thị như là một đối tượng riêng lẻ trong một Sheet mới (mục As new sheet), hoặc để đồ thị trong một Sheet đã có thuộc Workbook hiện hành (chọn mục As Object in).

Đối với ví dụ này ta sẽ đặt đồ thị trong Sheet hiện hành, nhấp Finish để chấp nhận theo mặc định. Cuối cùng được đồ thị và có thể kéo nó tới vị trí bất kỳ trên Sheet hiện hành.

Đối với dữ liệu của cột Chăn nuôi, cột Thủ công ta cũng có thể lập các đồ thị dạng cột tương tự như trên.

Bây giờ ta muốn biểu diễn một đồ thị đặt tất cả các dữ liệu của ba nghề chung lại với nhau để dễ dàng so sánh, ta sử dụng đồ thị hợp cột. Khi đó ta cần lựa chọn dữ liệu của cả ba cột Trồng

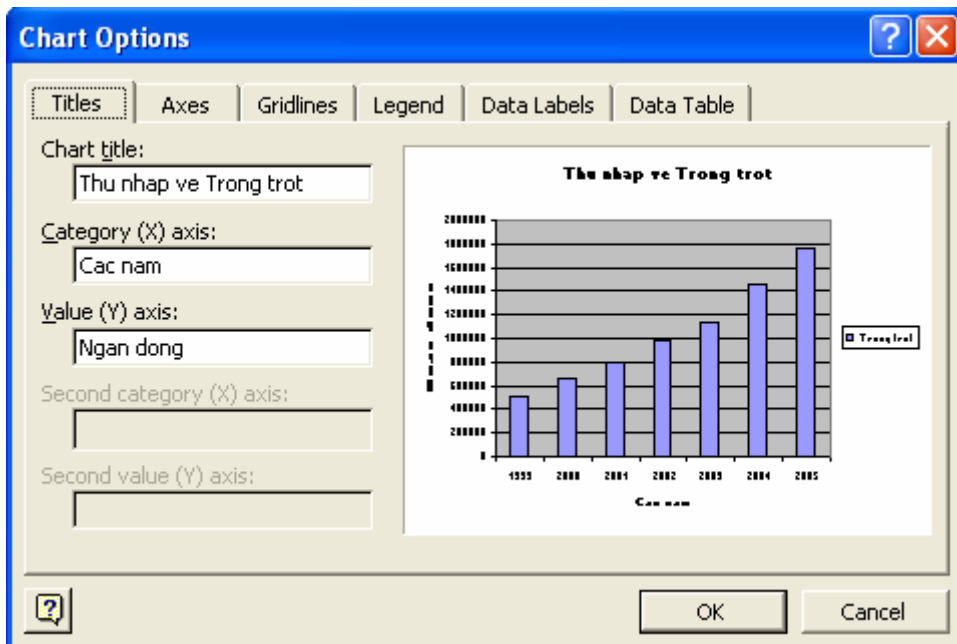
trọt, Chăn nuôi và Thủ công; sau đó làm các bước tương tự như trên. Đồ thị hoàn tất có dạng:



#### 3.2.4.2. Hiệu chỉnh biểu đồ

Để hiệu chỉnh biểu đồ trước tiên ta cần chọn biểu đồ bằng cách nhấp chuột vào biểu đồ muốn hiệu chỉnh, khung viền biểu đồ sẽ hiện 8 nút ở 4 góc và điểm giữa các cạnh. Sau đó có thể tiến hành các thao tác:

- Di chuyển biểu đồ: nhấp chuột vào biểu đồ và giữ tay cho đến khi con trỏ chuột biến thành dấu chữ thập với bốn mũi tên ở đầu thì kéo chuột để di chuyển biểu đồ đến vị trí mới.
- Thay đổi kích thước: dùng chuột rê các nút trên khung viền.
- Nhấp nút phải chuột lên biểu đồ, xuất hiện menu dọc nhỏ, chọn Chart Options, xuất hiện hộp thoại Chart Options. Hộp thoại này có nhiều lớp dùng để thay đổi tiêu đề đồ thị, thuyết minh các cột, các đường kẻ ngang dọc, chú thích, các nhãn...



• Hiệu chỉnh các thành phần bên trong biểu đồ:

- Xóa một thành phần: nháy chuột vào thành phần, ấn Delete.

- Thay đổi kích thước một thành phần: nháy chuột lên thành phần, xuất hiện 8 nút hình vuông, kéo các nút để thay đổi kích thước. Di chuyển thành phần sang vị trí khác: nháy chuột lên thành phần và kéo đi.

- Định dạng lại một thành phần: nháy đúp chuột vào thành phần, xuất hiện một hộp thoại gồm nhiều lớp, tiên hành lựa chọn theo hướng dẫn trên hộp. Ta có thể chọn mẫu tô và màu tô cho thành phần.

### 3.2.4.3. Đưa các đối tượng hình ảnh vào bảng tính

Thao tác hoàn toàn như trong Word 2000 ta có thể dùng thanh công cụ Drawing để đưa một đối tượng AutoShape, Text Box, Word Art vào trong một bảng tính như trong bảng tính sau:

	A	B	C	D	E	F
1						
2	<b>THÀNH TÍCH HỌC TẬP</b>					
3						
4						
5		12A	12B	12C		
6	Gioi	10	20	5		
7	Kha	25	20	20		
8	Trung binh	10	10	20		
9	Kem	0	1	5		
10						
11	<b>Trường Trung học Phổ thông Lê Lợi</b>					
12						
13						

Ta cũng có thể dùng lệnh *Insert / Picture / From File* để chèn một bức tranh lấy từ một tệp vào bảng tính, hoặc có thể dùng lệnh *Insert / Picture / Clip Art* để chèn một bức tranh lấy từ Clip Art vào bảng tính. Khi bức ảnh đã vào bảng tính ta có thể kéo nó tới vị trí bất kỳ trên bảng tính, có

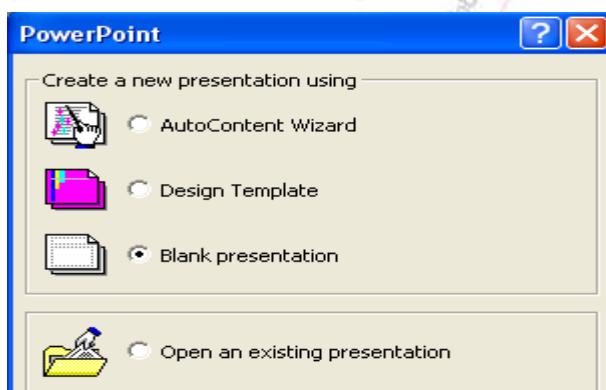
thể dùng nút Order trên Thanh công cụ Drawing để chuyển ảnh xuống dưới các đối tượng đồ họa khác.

**Ta có thể lấy một bức ảnh làm nền cho phần dữ liệu của bảng tính bằng cách:** dùng chuột lựa chọn bộ phận bảng tính muốn đặt lên phía trước của ảnh nền (đây có thể là một bộ phận của Sheet hiện hành hoặc bất kỳ bảng tính nào khác), dùng lệnh Edit / Copy để chép phần đã chọn vào vùng đệm, chuyển tới bảng tính chứa ảnh đồ họa đã nhập, dùng lệnh Edit / Paste để chép phần lựa chọn từ vùng đệm lên bảng tính, kéo phần vừa chép lên vị trí thích hợp trên bảng tính, kéo các khung viền của bộ phận bảng tính đã sao chép để định lại kích cỡ cho vừa khớp với ảnh nền.

### 3.3. POWERPOINT 2000

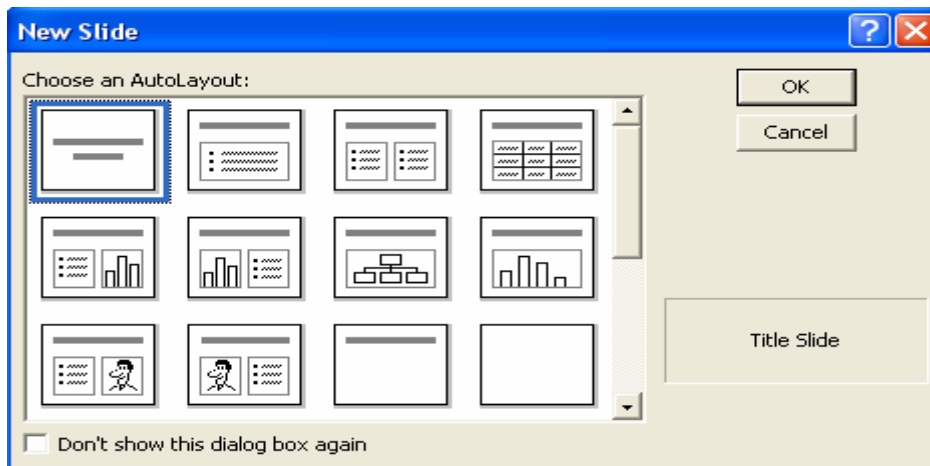
PowerPoint dùng để tạo một trình diễn (Presentation) gây ấn tượng cho một bài nói trước đông đảo người nghe trong một hội nghị, một buổi bảo vệ luận án. Cách dùng hệ thống menu, dùng các thanh công cụ Standard và Formatting, dùng Office Assistant và nhận trợ giúp, chèn các đối tượng đồ họa và hình ảnh, chèn Table trong PowerPoint được thực hiện giống như trong Word. Ta có thể dễ dàng sao chép văn bản trong Word và các biểu đồ trong Excel vào trong PowerPoint.

#### 3.3.1. Khởi động PowerPoint



Để khởi động PowerPoint dùng lệnh Start / Programs / Microsoft PowerPoint, PowerPoint được khởi động và xuất hiện hộp thoại PowerPoint:

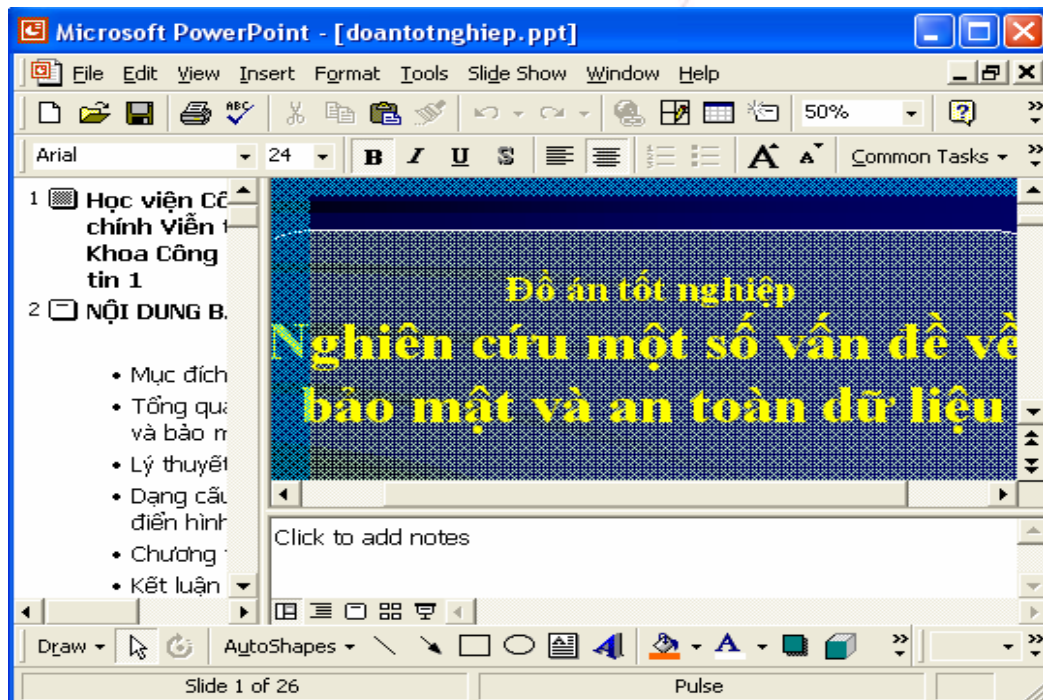
- AutoContent Wizard: bắt đầu với trình diễn chứa nội dung bằng tiếng Anh và thiết kế đề nghị sẵn, gồm nhiều trình diễn mẫu theo các chủ đề.
- Design Template: chọn các mẫu thiết kế sẵn, trong mẫu không có nội dung.
- Blank Presentation: tạo một trình diễn trắng. Khi chọn mục này xuất hiện hộp thoại New Slide cho phép ta chọn 24 mẫu Slide (bản phim) mới:



- Open an existing presentation: mở một trình diễn sẵn có.

Hai hộp thoại PowerPoint và New Slide ở trên đóng vai trò rất quan trọng, tuy nhiên ta có thể cho hiện (hay không cho hiện) hai hộp thoại này bằng lệnh Tools / Options, chọn lớp View, chọn (hay không chọn) hai mục Startup dialog (ứng với hộp thoại PowerPoint ở trên), New slide dialog.

Màn hình làm việc của PowerPoint ở chế độ Normal View có dạng sau:



Phía trên cửa sổ là Thanh tiêu đề, Thanh menu chính, Thanh công cụ Standard, Thanh Formatting. Màn hình chính chia thành ba khung: khung bên trái (khung 1) trình bày dạng Outline của tất cả các slide cùng với biểu tượng của mỗi slide (để chuyển tới một slide ta chỉ cần nhấp vào biểu tượng của nó), khung trên bên phải (khung 2) là hiển thị của slide hiện hành, khung dưới bên

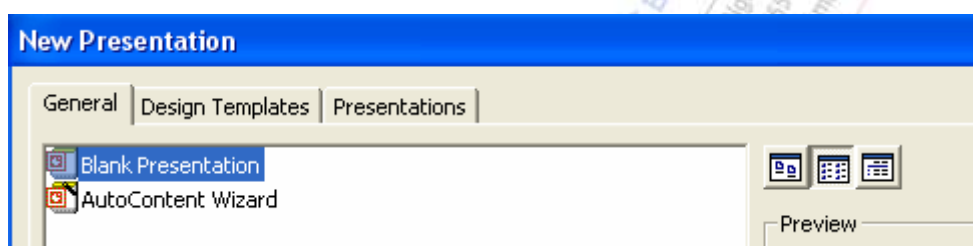
phải (khung 3) là nội dung phần chú giải của slide hiện hành.

Góc dưới bên trái có 5 nút dùng để lựa chọn cách xem slide: Normal View, Outline view (khung 1 rộng), Slide view (khung 2 rộng, khung 1 chỉ còn biểu tượng của các slide, khung 3 không có), Slide sorter view (nhiều slide cùng hiện trên màn hình), Slide show (hiện slide trên toàn màn hình). Slide còn có cách hiện Notes Pages ở dưới dạng khổ giấy A4 như khi in ra máy in, nửa trên là slide, nửa dưới là nội dung các chú giải, về cách hiện này bằng lệnh View / Notes Pages.

### 3.3.2. Tạo trình diễn dựa trên Slide trắng

Các bước để tạo một trình diễn từ đầu:

**Bước 1:** Dùng lệnh File / New, hiện hộp thoại New Presentation:



Chọn lớp General, chọn biểu tượng Blank Presentation, xuất hiện hộp thoại New Slide, chọn một slide, nháy OK. Đặt màn hình ở chế độ Normal View.

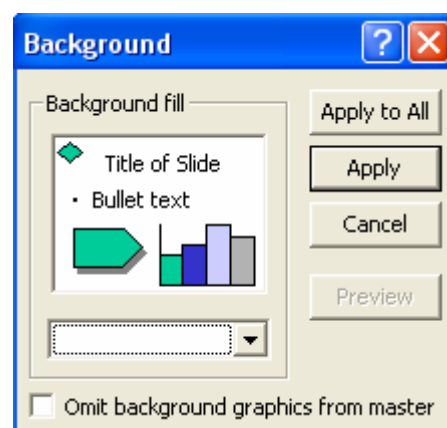
**Bước 2:** Gõ tiêu đề của Slide và nội dung của nó. Chèn một slide trắng mới: nháy nút Common Tasks trên thanh Formatting, hiện menu dọc, chọn mục New Slide (hoặc dùng lệnh Insert / New Slide, hay Ctrl+M), xuất hiện lại hộp thoại New Slide, chọn một mẫu slide, bổ xung nội dung cho nó.

**Bước 3:** Xem cách hiển thị slide đang thiết kế: nháy nút Slide Show ở góc trái bên dưới cửa sổ PowerPoint. Trong lúc xem khi nháy nút phải chuột vào màn hình, xuất hiện menu tắt gồm các mục: End Show (kết thúc xem), Next (xem trang tiếp), Previous (lùi về trang trước).

**Bước 4:** Ghi tệp vào đĩa dùng lệnh File / Save.

**Bổ xung thay đổi màu nền của Slide:** dùng lệnh Format / Background, xuất hiện hộp thoại Background. Nháy vào nút mũi tên trong khung Background fill để hiện các mẫu màu và chọn (xử lý hộp thoại giống như trong Word). Nút Apply: áp dụng thay đổi cho slide hiện tại. Nút Apply to All: áp dụng thay đổi cho tất cả các slide.

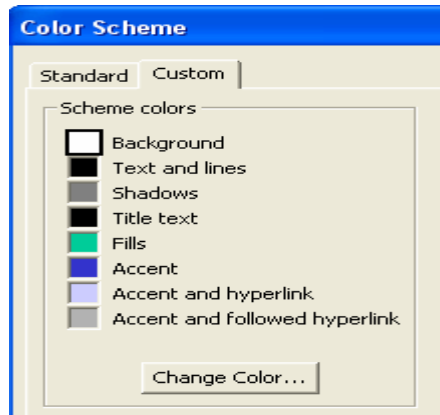
**Thay đổi màu của phong chữ:** chọn Slide, dùng lệnh Format / Slide Colour Scheme, hiện hộp thoại Color Scheme.



Chọn lớp Custom. Trong mục Scheme colors, chọn đối tượng muốn thay đổi màu (như Title Text, Text and Lines, background...) và nháy nút Change Color, hiện hộp thoại tương ứng với đối tượng đã chọn. Chọn lớp Standard để chọn màu từ bảng, hoặc chọn lớp Custom để trộn màu riêng.



Trong hộp thoại Color Scheme, nút Apply: áp dụng màu mới chỉ cho slide hiện hành, nút Apply to All: áp dụng màu cho tất cả các Slide.

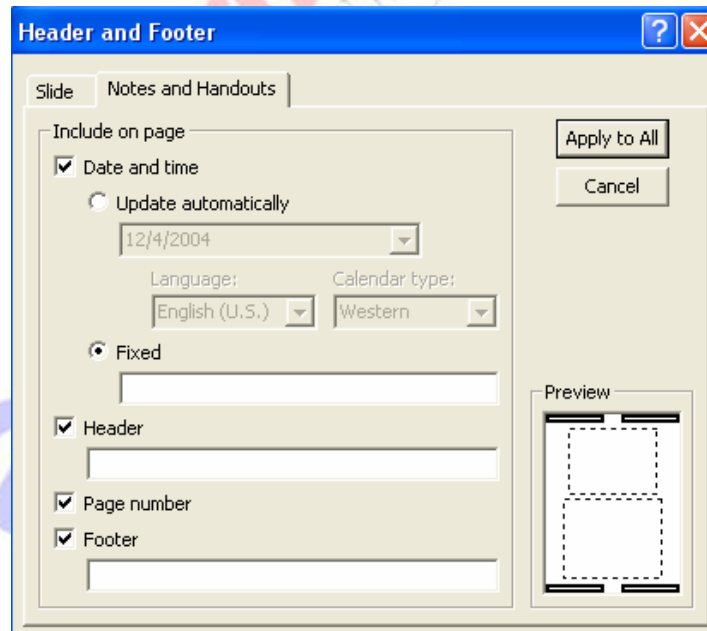


**Bổ xung và thay đổi phần đầu trang và cuối trang:** dùng lệnh View / Header and Footer, xuất hiện hộp thoại Header and Footer:

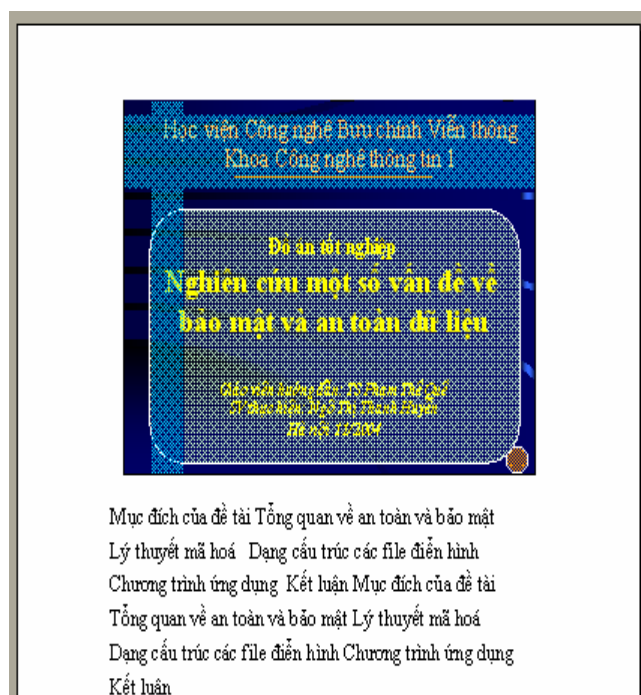
- Chọn lớp Slide. Trong lớp này mục Date and Time dùng để chèn ngày và thời gian vào cuối slide (mục Update automatically để điền tự động, mục Fixed để điền trực tiếp), mục Slide number để chèn số trang cho slide, mục Footer để chèn phần văn bản cuối mỗi slide.

- Chọn lớp Notes and Handouts: chèn ngày tháng, thời gian, phần đầu trang và cuối trang, số trang đối với tờ giấy A4 in ra trên máy in cho các dạng Notes Pages và Handouts.

Chú ý, để chèn số trang, ngày, thời gian vào bất kỳ đâu trên một Slide riêng biệt ta thực hiện: đưa con trỏ vào chỗ muốn chèn, dùng lệnh Insert / Data and Time và Insert / Slide Number.



**Tạo các chú giải cho một Slide**



Trong chế độ Normal View khi soạn một Slide, khung bên phải phía dưới là nơi nhập chú giải (Notes) cho Slide. Phần chú giải này có thể là các giải thích thêm cho các điểm trong phần slide, hoặc có thể là phần giảng chi tiết của diễn giả cho các chủ đề trong phần slide. Phần Slide và phần Notes của nó tạo nên một Trang in bằng giấy để xem thêm. Muốn xem thử hình dạng các Trang in dùng lệnh View / Notes Pages.

**Chèn hình ảnh vào Slide:** chèn một hình ảnh từ Clip Art Galery dùng lệnh Insert / Picture / Clip Art. Chèn một hình ảnh từ tệp dùng lệnh Insert / Picture / From File ...

**Tạo một biểu đồ:** Ta có thể tạo một biểu đồ đẹp trong Excel và sau đó dán vào PowerPoint qua trung gian Clipboard. Tuy nhiên ta cũng có thể tạo biểu đồ ngay trong PowerPoint bằng lệnh Insert / Chart, xuất hiện cửa sổ Datasheet giống như một bảng tính, nhập dữ liệu vào bảng tính. Để trở về PowerPoint nhảy bên ngoài biểu đồ trên Slide của PowerPoint. Định cỡ lại và chỉnh sửa biểu đồ làm giống trong Excel.

		A	B	C	D	E
		1st Qtr	2nd Qtr	3rd Qtr	4th Qtr	
1	East	20.4	27.4	90	20.4	
2	West	30.6	38.6	34.6	31.6	
3	North	45.9	46.9	45	43.9	
4						

**Xem Slide dưới dạng đen trắng:** dùng lệnh View / Black and White. Muốn chuyển lại sang xem màu thì dùng lại lệnh View, bỏ chọn mục Black and White.

**Xóa một Slide:** cho các slide hiện ở chế độ Outline View, nhấp vào biểu tượng của slide để chọn nó, dùng lệnh Edit / Delete Slide.

**Tạo một slide hoàn toàn giống với một Slide có sẵn:** cho các slide hiện ở chế độ Outline View, nhấp vào biểu tượng của slide để chọn nó, dùng lệnh Edit / Duplicate. Ta có thể sửa chữa trên Slide vừa nhận được để được một slide mới.

### 3.3.3. Thiết lập cách hiển thị Slide

#### 3.3.3.1. Xác định cách chuyển tiếp giữa các Slide

Cách chuyển tiếp của một slide bao gồm cách nó xuất hiện trên màn ảnh và nó kết thúc hiện bằng cách nào. Ta có thể quy định cách chuyển tiếp cho từng slide một, hoặc đồng thời cho toàn bộ các slide của một trình diễn. Các bước thực hiện:

- Trong chế độ Slide View hoặc Slide Sorter, chọn một Slide.
- Dùng lệnh Slide Show / Slide Transition, hiện hộp thoại Slide Transition.
- Mục Effect chọn kiểu chuyển tiếp tới slide, ví dụ Cover Right và Show (khi tới lượt, slide này chuyển động chậm từ trái sang phải vào màn hình). Ngầm định là No Transition.
- Mục Advance chọn kiểu kết thúc slide khi trình diễn. Chọn On mouse click (ngầm định): khi nhấp chuột thì kết thúc hiện slide. Chọn Automatically after N seconds: tự động kết thúc hiện sau N giây. Có thể chọn đồng thời cả hai mục này.

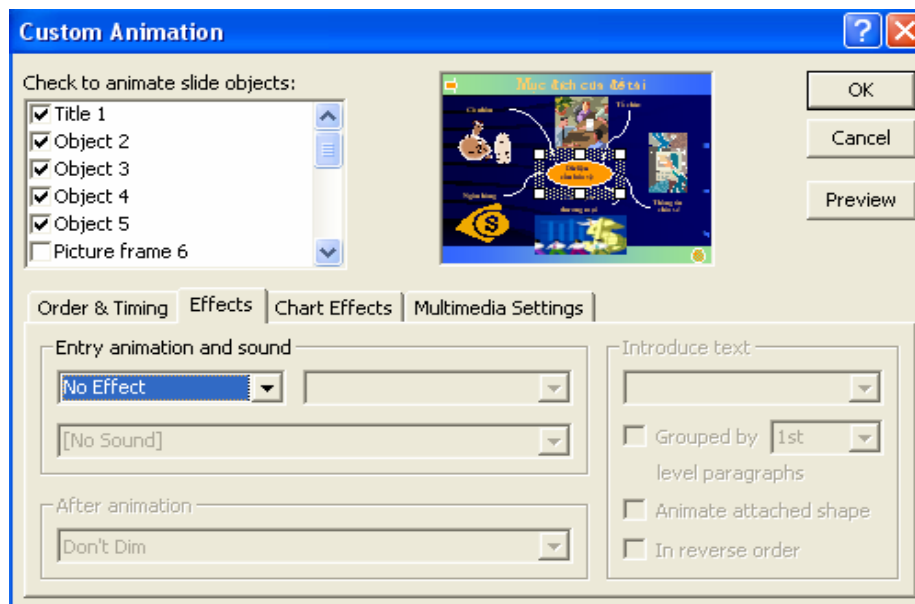


- Nhấp nút Apply để áp dụng kiểu chuyển cho slide đã chọn, hoặc nhấp nút Apply to All để áp dụng kiểu chuyển tiếp cho tất cả các slide. Để xem thử cách chuyển tiếp đã thiết đặt nhấp nút Slide Show ở góc dưới bên trái cửa sổ PowerPoint.

#### 3.3.3.2. Làm cho văn bản và đối tượng của slide trở nên sinh động

Trong mục này xét cách hiển thị của từng đối tượng trong một slide sao cho sinh động, hấp dẫn người xem. Các bước thực hiện:

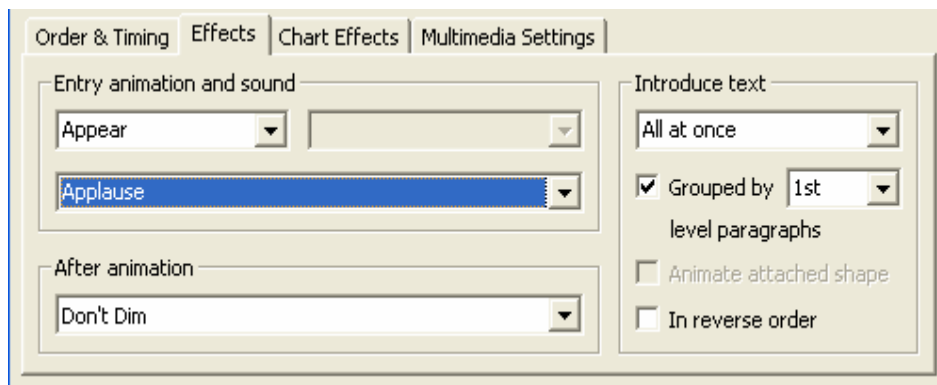
1. Trong chế độ Slide View, chọn slide có các đối tượng cần thêm hiệu ứng.
2. Dùng lệnh Slide Show / Custom Animation, hiện hộp thoại Custom Animation:



3. Trong khung Check to animation slide objects chọn đối tượng muốn tạo hiệu ứng (ví dụ Title1).
4. Chọn lớp Order & Timing. Khung Animation order: chọn đối tượng và nhấn phím mũi tên để thay đổi thứ tự xuất hiện các thành phần của Slide trên màn hình.

Khung Start animation có hai mục. Chọn mục On mouse click để quy định: muốn đối tượng xuất hiện phải nhấn chuột (nếu đối tượng văn bản có nhiều mục, muốn mỗi mục xuất hiện phải nhấn chuột). Chọn Automatically N seconds after previous event: đối tượng tự xuất hiện sau sự kiện trước N giây.

5. Chọn lớp Effects. Mục Entry Animation and Sound: chọn cách xuất hiện đối tượng, ví dụ Fly from Bottom (bay từ dưới lên). Mục Introduce text cho phép chọn đối tượng xuất hiện theo ba cách: All at once (cùng lúc), By Word (từng từ), By Letter (từng ký tự). Mục After animation: xác định tình trạng của đối tượng sau khi hiệu ứng hoàn thành (ẩn đi, đổi màu). Nhấn nút Preview để xem thử kết quả.



6. Lặp lại các bước 3, 4, 5 cho mỗi một đối tượng muốn tạo hiệu ứng. Nháy nút OK để ghi lại thiết lập.

**Chú ý:**

- Cách nhanh chóng để tạo một hiệu ứng cơ bản là chọn đối tượng trong chế độ Slide View, dùng lệnh Slide Show / Preset Animation, lựa chọn hiệu ứng mong muốn từ 15 cách. Ngâm định: các đối tượng xuất hiện khi ấn chuột (On mouse click).

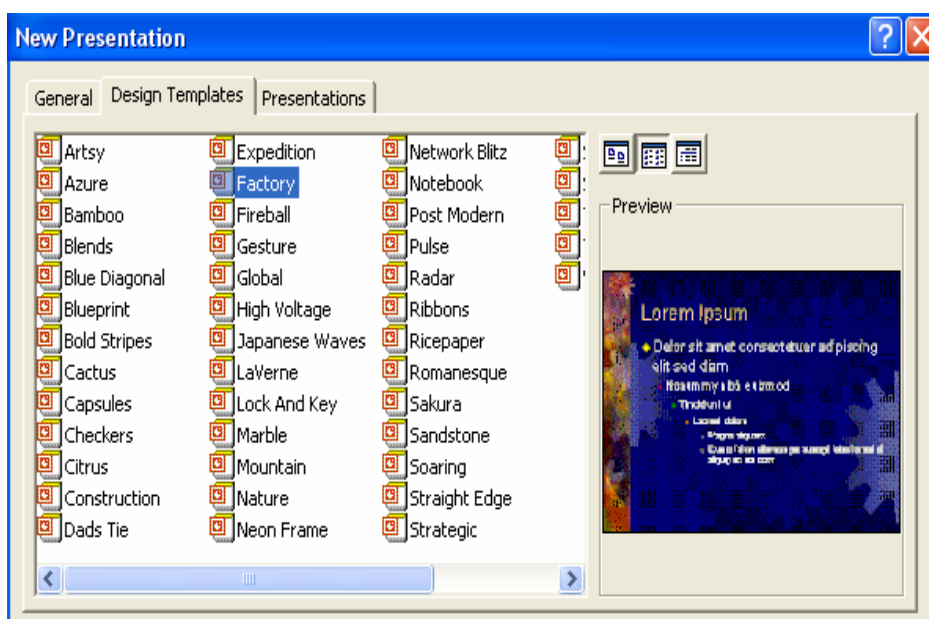
- Để xem thử hiệu ứng trong chế độ Slide View, dùng lệnh Slide Show / Animation Preview, các hiệu ứng trình diễn trong một slide thu nhỏ ở góc trên bên phải màn hình, muốn xem lại chỉ cần nháy chuột vào slide nhỏ này. Không cho hiện slide nhỏ nữa: nháy nút Close của nó.

**3.3.4. Các công cụ phụ trợ**

**3.3.4.1. Dùng các Template sẵn có**

Khi dùng lệnh File / New, xuất hiện hộp thoại New Presentation có nhiều lớp. Lớp Design Templates cho các mẫu thiết kế Slide (không có nội dung), lớp Office 97 Templates cho các mẫu thiết kế slide của Office 97. Khi cần tạo một trình diễn mới nên sử dụng các thiết kế có sẵn này. Khi đã chọn một Template, mỗi lần dùng lệnh Insert / New Slide, Slide mới vẫn theo Template này.

Lớp Presentations chứa các trình diễn có sẵn nội dung theo nhiều chủ đề bằng tiếng Anh (Marketing Plan, Company Meeting, Training,...), mỗi trình diễn có nhiều Slide (có thể có tới 20 slide sẵn). Nội dung trong slide là các gợi ý cho người dùng làm báo cáo.



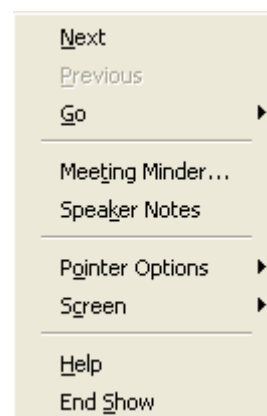
### 3.3.4.2. Chuyển đổi giữa các Template

Giả sử khi tạo một trình diễn nhờ lệnh File / New, hiện hộp thoại New Presentation, chọn lớp Design Template, chọn mẫu Bamboo và soạn 3 slide theo mẫu này. Bây giờ bạn không thích mẫu này và muốn chuyển sang mẫu khác (chẳng hạn Ribbons), hãy dùng lệnh Format / Apply Design Template, hiện hộp thoại, chọn Template mong muốn và nhấn Apply.

### 3.3.4.3. Sử dụng menu tắt khi thực hiện một trình diễn

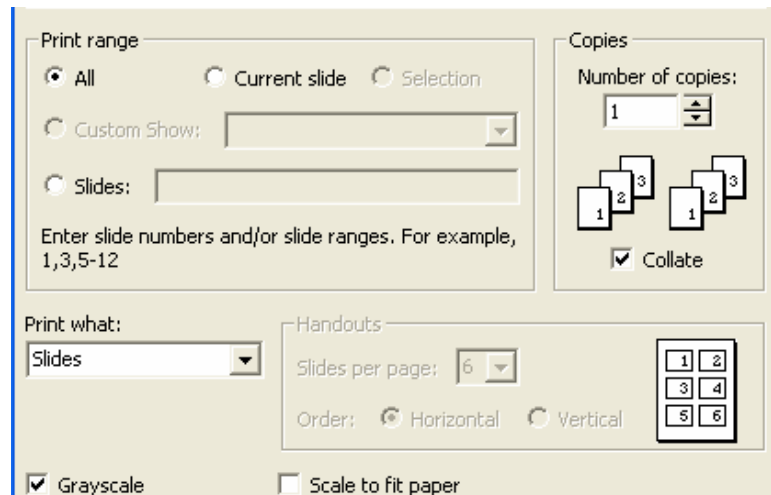
Khi trình chiếu một slide, nếu ta đưa chuột lên phía trên, góc dưới bên trái màn hình xuất hiện một nút chữ nhật nhỏ với mũi tên lên. Nháy nút trái chuột vào nút này (hoặc nháy nút phải chuột vào nền màn hình), xuất hiện menu tắt.

Ý nghĩa các mục của menu tắt như sau: Mục End Show: kết thúc xem. Mục Next: xem slide tiếp. Mục Previous: xem slide trước. Mục Go: chuyển tới slide khác. Mục Speaker Notes: cho hiện phần chú giải của slide hiện hành trong một cửa sổ nhỏ. Mục Screen: cho đen màn hình hay không.



### 3.3.4.4. In các slide và các chú giải ra giấy

Dùng lệnh File / Print, xuất hiện hộp thoại Print, trong mục Print What hãy lựa chọn đối tượng cần in: Slides (mỗi slide in trên một tờ giấy A4 theo phương án nằm ngang). Handouts (nhiều slide in trên cùng một tờ A4 theo chiều dọc khổ giấy, nhập số lượng slide in trên một trang, có in phần đầu trang và cuối trang), Notes Pages (mỗi trang giấy A4 in một slide cùng với phần chú thích của nó, có đầu trang và cuối trang của trang giấy, có cuối slide, in dọc giấy), Outline View (in nội dung toàn bộ các slide theo kiểu Outline View, in dọc giấy).



#### 3.3.4.5. Một vài lời khuyên khi chuẩn bị các slide

1. Khi chuẩn bị slide không nên đặt quá nhiều thông tin lên một slide. Mỗi slide được thiết kế để phản ánh một ý tưởng, được biểu diễn không quá 7 dòng, mỗi dòng không quá 6 từ tiếng Anh (khoảng 10 từ tiếng Việt). Không dùng vùng phía dưới của slide vì nó khó đọc.
2. Đảm bảo rằng phông chữ dùng trong slide có thể đọc được khi ở cuối phòng.
3. Hãy dùng màu cho các cụm từ để nhấn mạnh các ý tưởng chính. Làm cho slide thực sự hấp dẫn. Hình ảnh, các ký tự được nhấn mạnh có thể làm cho slide có vẻ chuyên nghiệp.
4. Hãy chiếu thử các slide trước khi thực hiện bài giảng.
5. Đừng bật máy chiếu mà không hiện slide vì ánh sáng làm cho người dự rất khó chịu.
6. Nên hiện dần các thông tin trên một slide từng bước một phù hợp với bạn đang thuyết trình từng điểm nội dung một.
7. Đừng nói khi đang chiếu một slide mới. Hãy cho người nghe thời gian để đọc. Slide sẽ thu hút chú ý của người nghe và những lời bạn nói sẽ không được chú ý.
8. Hãy luôn nhìn xuống khán giả. Muốn nhấn mạnh một điều gì đó, hãy dùng bút laser trỏ lên slide. Hết sức tránh nhìn lên màn hình và nói trong khi quay lưng lại phía người nghe.
9. Nên chuẩn bị sẵn các trang in của Slide.

### 3.4. VIRUS TIN HỌC VÀ CÁCH PHÒNG CHỐNG

#### 3.4.1. Virus tin học là gì ?

Virus tin học hay còn gọi là virus máy tính (computer virus) là một đoạn chương trình ẩn, có kích thước nhỏ, có khả năng tự sao chép, sinh sôi nảy nở như một virus sinh học, lây lan nhanh và có thể gây nguy cơ phá hoại các phần mềm hệ của hệ điều hành, xóa một phần hay toàn bộ các tập tin dữ liệu, tạo ra các tập tin vô nghĩa chiếm lấy không gian bộ nhớ của máy, hoặc chỉ đơn thuần trêu chọc hay đe dọa người sử dụng, ...

Thoạt đầu, virus tin học là vũ khí của những người viết phần mềm muốn bảo vệ sản phẩm của họ chống việc sao chép bất hợp pháp. Sau một số người tạo ra các virus khác nhằm mục đích phá hoại hay đơn thuần chỉ để tự khẳng định mình và cấy vào trong các máy khác gây lây nhiễm trên một diện rộng. Năm 1987, virus Lehigh lần đầu tiên được phát hiện trên máy IBM PC. Cùng lúc đó, virus Brain từ Pakistan xâm nhập vào các trường Đại học của Mỹ. Một virus khác từ Đại học Hebrew, Israel cũng xuất hiện. Từ đó đến nay, hàng ngàn chủng loại virus được thông báo phát hiện gây phá hoại, tê liệt nhiều hệ thống máy trên toàn thế giới gây thiệt hại nhiều triệu dollar.

Virus được chia thành 2 nhóm chính:

+ B-virus (Boot-virus) : là các virus chuyên tấn công lên các Boot sector hay Master boot, điển hình như : Brain, Nops, Clock, Stone, Sài Gòn 01/91, ...

+ F-virus (File-virus) : là các virus chỉ tấn công lên các file .COM và .EXE, điển hình như : Friday 13th, Yankee Doodle, DataLock, Keypress, VN01, ... Hay một số chuyên phá hoại các file có phần mở rộng .DB\* của các chương trình dBase, FoxBase hay chuyên phá hoại các file văn bản có phần mở rộng là .DOC như virus Concept, Macro, ....

Ngoài ra, một số virus vừa mang cả 2 tính phá hoại Boot và File như virus Dark Avenger .... Người ta còn lo ngại một số người đang cố tình tạo ra các virus còn có thể xâm nhập vào các hệ thống mạng máy tính qua đường truyền thư điện tử (E-mail) hoặc mạng Internet, mạng của các ngân hàng lớn gây đảo lộn các số liệu tài khoản, hoặc xâm nhập vào các hệ thống máy tính quân sự tạo nên những báo động giả rất nguy hiểm ...

Hiện nay, một số nước đã ra những điều luật xem những kẻ cố tình tạo ra các virus tin học phá hoại, lấy cắp mật mã, ... (gọi chung là những kẻ tin tặc - hackers) như là những tội phạm hình sự.

### 3.4.2. Cách phòng chống virus

- Không chép các chương trình mà bạn không rõ nguồn gốc và đặc biệt chưa qua thủ tục kiểm tra virus.

- Các file dữ liệu quan trọng cần phải được COPY lưu trữ dự phòng và có thể dùng các chương trình PKZIP.EXE hay ARJ.EXE.... để nén lại, tiết kiệm đĩa.

- Có thể sử dụng nhiều phần mềm BACKUP để lưu.

- Dùng lệnh BACKUP và RESTORE của DOS như sau:

### 3.4.3. Các chương trình kiểm tra và diệt virus thông dụng :

Trước hết ta cần có một số đĩa khác phòng ngừa sự mất mát hệ thống, đĩa khởi động, đĩa để lưu các chương trình diệt virus...

- Ví dụ cần có đĩa để lưu chương trình NORTON UTILITIES có file RESCUE.EXE, tệp này sẽ giúp bạn tạo một đĩa cấp cứu lưu trữ các bảng FAT, Boot Sector, CONFIG.SYS... của máy vi tính.

- Hoặc bạn phải phải tạo một đĩa khởi động máy cho mỗi loại hệ điều hành, ví dụ đối với HĐH MSDOS đĩa khởi động chứa các tập tin:



COMMAND.COM

IO.SYS

MSDOS.SYS

AUTOEXEC.BAT

CONFIG.SYS

SYS.COM

NDD.EXE

- Dùng một số đĩa chứa chương trình quét và diệt virus có phiên bản (version) mới nhất, như: SCAN.EXE và CLEAN.EXE của Công ty Mc-Afee, Mỹ

+ VSAFE.EXE của MS-DOS, Mỹ

+ MSAV.EXE của MS-DOS, Mỹ

+ NORTON ANTIVIRUS Của NORTON, Mỹ

+ ATV.EXE của Ngô Anh Vũ và Phạm Dũ Liêm, ĐH Kinh tế TP. HCM

+ TAV.EXE của Dương Hồng Tấn, Trần Thanh Sơn, Unisoft, TP. HCM

+ D2.COM của Trương Minh Nhật Quang, ĐH Tại chức TP. Cần Thơ

+ BKAV của Đại học Bách khoa Hà Nội

- v.v..

Khi máy đã bị VIRUS, bạn nên khởi động lại bằng đĩa khởi động khác không có VIRUS sau đó chạy các chương trình diệt Virus đã có ở trên đĩa đã được lưu.

## Chương 4: NGÔN NGỮ LẬP TRÌNH C

### 4.1. GIỚI THIỆU CHUNG

Năm 1970 Ken Thompson sáng tạo ra ngôn ngữ B dùng trong môi trường hệ điều hành UNIX trên máy điện toán DEC PD-7. B cũng là chữ tắt của BCPL (Basic Combined Programming Language) do Martin Richards viết. Năm 1972 Dennis Ritchie của hãng Bell Laboratories (và Ken Thompson) sáng tạo nên ngôn ngữ C nhằm tăng hiệu quả cho ngôn ngữ B. Lúc đầu ngôn ngữ C không được mọi người ưa dùng. Nhưng sau khi D.Ritchie cho xuất bản cuốn "The C Programming Language" ("Ngôn ngữ lập trình C") thì ngôn ngữ C được chú ý và được sử dụng rộng rãi. Người ta đã dùng C để viết hệ điều hành đa nhiệm UNIX, O/S 2 và ngôn ngữ Dbase. C đã được cải tiến qua nhiều phiên bản: trình biên dịch Turbo C từ phiên bản 1 đến phiên bản 5, Microsoft C từ phiên bản 1 đến phiên bản 6. Hiện nay, C lại được phát triển để thành C++ với 3 trình biên dịch: Borland C++, Visual C++ và Turbo C++.

Mặc dù hiện nay có khá nhiều ngôn ngữ lập trình mới, nhưng C vẫn là một ngôn ngữ lập trình được ưa chuộng. C được ứng dụng để viết các phần mềm trong nhiều lĩnh vực, đặc biệt là trong khoa học kỹ thuật.

### 4.2. MỘT SỐ KIẾN THỨC CƠ SỞ

#### 4.2.1. Bộ kí tự, từ khóa, tên

##### 4.2.1.1. Bộ kí tự trong C

Mọi ngôn ngữ đều được xây dựng trên một bộ kí tự (các chữ, các kí hiệu). Đối với ngôn ngữ C sử dụng bộ kí tự sau:

Tập các chữ cái in hoa: A, B, C, D, ..., Z

Tập các chữ cái in thường: a, b, c, d, ..., z

Tập các chữ số: 0, 1, 2, 3, ..., 9

Các dấu chấm câu: , . ; : / ? [ ] { } ! @ # \$ ^ & \* ( ) + = - < > "

Các kí tự không nhìn thấy: dấu trống (Space), dấu Tab, dấu xuống dòng (Enter),

Dấu gạch dưới \_

##### 4.2.1.2. Các từ khoá (Keywords)

Từ khoá là tập các từ dùng riêng của ngôn ngữ, mỗi từ khoá mang theo một ý nghĩa và tác dụng riêng. Từ khoá không thể định nghĩa lại và cũng không thể lấy từ khoá đặt tên cho các đối tượng. Dưới đây là bảng liệt kê các từ khoá thông dụng trong C.

auto	break	base	char	continue	default
do	double	else	extern	float	for

goto	if	int	long	register	return
short	sizeof	static	struct	switch	typedef
union	unsigned	void	public	while	volatile

#### 4.2.1.3. Tên và cách đặt tên

Tên hay còn gọi là định danh (identifier) dùng để gọi các biến, hằng hoặc hàm. Đối với ngôn ngữ C, mọi tên phải được khai báo trước khi sử dụng. Tên là dãy các kí tự liền nhau gồm các chữ cái, a . . z, A . . Z, các chữ số 0 . . 9 và dấu gạch dưới (dấu gạch dưới thường dùng để liên kết các thành phần của tên). Tuy nhiên, tên không được bắt đầu bằng chữ số và không chứa các kí tự đặc biệt như dấu cách, dấu tab và dấu chấm câu. Không được lấy từ khoá của C để đặt tên cho đối tượng.

Ví dụ về cách đặt tên đúng: Delta, E\_Mu\_X, Function1 . . .

Ví dụ về cách đặt tên sai:

2Delta: bắt đầu bằng kí tự số

E Mu\_X: chứa dấu ngăn cách

Ngôn ngữ C phân biệt chữ in hoa và chữ in thường do vậy những tên sau đây là khác nhau: x <> X, While <> while, For <> for. Do vậy, chúng ta cần lưu ý trong khi viết chương trình. Thông thường tên các biến, hàm được đặt bằng chữ in thường, tên các hằng được đặt bằng chữ in hoa.

#### 4.2.1.4. Lời giải thích

Trong khi viết chương trình, đôi khi chúng ta cần ghi thêm một số lời ghi chú hoặc giải thích để chương trình trở nên dễ hiểu và dễ đọc. Lời giải thích không có tác dụng tạo nên mã chương trình và sẽ được trình dịch bỏ qua trong khi dịch chương trình. Phần ghi chú có thể biểu hiện trên nhiều dòng và được đánh dấu bởi cặp kí hiệu */\* đoạn văn bản ghi chú \*/*.

### 4.2.2. Cấu trúc chương trình trong C

#### 4.2.2.1. Cấu trúc tổng quát của chương trình trong C

Chương trình tổng quát viết bằng ngôn ngữ C được chia thành 6 phần, trong đó có một số phần có thể có hoặc không có tùy thuộc vào nội dung chương trình và ý đồ của mỗi lập trình viên.

**Phần 1:** Khai báo các chỉ thị đầu tệp và định nghĩa các hằng sử dụng trong chương trình.

**Phần 2:** Định nghĩa nên các cấu trúc dữ liệu mới (user type) để sử dụng trong khi viết chương trình.

**Phần 3:** Khai báo các biến ngoài (biến toàn cục) được sử dụng trong chương trình.

**Phần 4:** Khai báo nguyên mẫu cho hàm (Function Prototype). Nếu khai báo qui cách xây dựng và chuyển tham biến cho hàm, compiler sẽ tự động kiểm tra giữa nguyên mẫu của hàm có phù hợp với phương thức xây dựng hàm hay không trong văn bản chương trình.

**Phần 5:** Mô tả chi tiết các hàm, các hàm được mô tả phải phù hợp với nguyên mẫu đã được khai báo cho hàm.

**Phần 6:** Hàm *main()*, hàm xác định điểm bắt đầu thực hiện chương trình và điểm kết thúc thực hiện chương trình.

#### 4.2.2.2. Các bước cơ bản khi viết chương trình

##### Bước 1: Soạn thảo chương trình (dùng Turbo C)

Soạn thảo chương trình là giai đoạn dùng chương trình soạn thảo để viết văn bản chương trình. Turbo C trang bị một chương trình soạn thảo, dịch và thực hiện chương trình ngay trong môi trường của C, đó là chương trình có tên TC.EXE. Bản thân TC.EXE cũng trang bị cho người sử dụng một số phím chức năng giống như TURBO.EXE để soạn thảo. Khi ghi file văn bản chương trình lên đĩa, TC.EXE ngầm định đặt phần mở rộng của file là \*.C mà ta thường gọi là chương trình nguồn (source program). Sau đây là một số phím chức năng cơ bản nhất của TC.EXE.

- F1 (help) : Thực hiện chương trình trợ giúp trong khi viết chương trình.
- CTRL + F1 : Thực hiện trợ giúp nhanh trong khi soạn thảo
- F2 (save) : Ghi file văn bản chương trình lên đĩa với phần mở rộng là \*.c
- CTRL + F2 : Ghi file văn bản chương trình lên đĩa với một tên khác có phần mở rộng là \*.c
- F3 : Mở file mới để thực hiện soạn thảo.
- Alt + F3 : Đóng file văn bản đang trong cửa sổ soạn thảo hiện thời
- F4 : Dịch và thực hiện chương trình cho tới khi gặp dòng lệnh mà tại vị trí đó chúng ta bấm F4
- F5 : Mở rộng hoặc thu nhỏ vùng soạn thảo trên màn hình
- Alt+F5 : Nhìn lại kết quả thực hiện chương trình của lần chạy trước đó
- F6 : Thay đổi cửa sổ màn hình soạn thảo
- Alt + 1, 2, 3 : Qui định các cửa sổ màn hình 1, 2, 3 trên cùng một trang màn hình
- F7 : Thực hiện chương trình theo chế độ từng dòng lệnh kể cả các lệnh trong thân của hàm
- F8 : Thực hiện chương trình theo chế độ từng dòng lệnh nhưng coi các lời gọi hàm là một lệnh.
- F9 : Dịch chương trình nguồn thành file \*.OBJ
- CTRL + F9 : Dịch và thực hiện chương trình (đồng thời tạo file \*.OBJ sau đó tạo file mã máy \*.OBJ)
- F10 : Thực hiện trong chế độ thực đơn
- Home: Đưa con trỏ về đầu dòng
- End : Đưa con trỏ về cuối dòng
- PgUp : Đưa con trỏ lên phía trên một trang màn hình

PgDn : Đưa con trỏ xuống phía dưới một trang màn hình

Del : Xoá kí tự tại vị trí con trỏ

Back Space : Xoá kí tự nằm bên trái con trỏ

CTRL+ PgUp : Đưa con trỏ về đầu văn bản

CTRL+ PgDn : Đưa con trỏ về cuối văn bản

Shift + →: Đánh dấu khối văn bản sang bên trái

Shift + ←: Đánh dấu khối văn bản sang bên phải

Shift + ↑ : Đánh dấu khối văn bản theo từng dòng lên phía trên

Shift + ↓ : Đánh dấu khối văn bản theo từng dòng lên phía dưới

CTRL + Y : Xoá cả dòng chứa con trỏ

CTRL+Q+Y: Xoá tới cuối dòng kể từ vị trí con trỏ

CTRL +K+Y : Xoá khối văn bản đã được đánh dấu trước đó

CTRL+K+C : Copy khối văn bản đã được đánh dấu tới vị trí hiện tại của con trỏ

CTRL+K+V : Chuyển khối văn bản đã được đánh dấu tới vị trí hiện tại của con trỏ

CTRL+K+W : Ghi khối đã được đánh dấu lên đĩa. Nếu tên tệp là PRN thì nội dung của nó sẽ được chuyển qua máy in

CTRL+K+R : Đọc một tệp khác từ đĩa vào, phần được đọc coi như một khối được đánh dấu

CTRL +Q+F : Tìm cụm từ đầu tiên xuất hiện trong văn bản

CTRL+Q+A : Tìm và thay thế cụm từ xuất hiện đầu tiên trong văn bản

CTRL+L : Tìm hoặc thay thế từ tiếp theo xuất hiện trong văn bản

## **Bước 2: Dịch và hiệu chỉnh chương trình (dùng turbo c)**

Chúng ta có thể gọi chương trình dịch của C trực tiếp trong chế độ soạn thảo bằng cách bấm phím F9. Chương trình dịch có nhiệm vụ dịch chương trình của người sử dụng từ file chương trình nguồn có phần mở rộng là \*.C thành tệp có phần mở rộng là \*.OBJ, sau đó liên kết các tệp \*.OBJ lại với nhau để tạo nên file chương trình mã máy có dạng \*.COM (chương trình mã máy đã được nén) hoặc \*.EXE (chương trình mã máy chưa được nén). Quá trình liên kết được thực hiện thông qua trình liên kết Linker.

Trong quá trình dịch, chương trình có thể gặp lỗi, có ba loại lỗi chính (không kể tới lỗi do giải thuật). Đó là:

- Lỗi được thông báo bởi từ khoá error (lỗi cú pháp): Loại lỗi này thường xảy ra trong khi soạn thảo chương trình, chúng ta có thể viết sai các từ khoá ví dụ thay vì viết là int chúng ta soạn thảo sai thành Int (lỗi chữ in thường thành in hoa), hoặc viết sai cú pháp các biểu thức như thiếu các dấu ngoặc đơn, ngoặc kép hoặc dấu chấm phẩy khi kết thúc một lệnh, hoặc chưa khai báo nguyên mẫu cho hàm.

- Lỗi được thông báo bởi từ khoá Warning (lỗi cảnh báo): Lỗi này thường xảy ra khi ta khai báo biến trong chương trình nhưng lại không sử dụng tới chúng, hoặc lỗi trong các biểu thức kiểm tra khi biến được kiểm tra không xác định được giá trị của nó, hoặc lỗi do thứ tự ưu tiên các phép toán trong biểu thức. Hai loại lỗi error và warning được thông báo ngay khi dịch chương trình thành file \*.OBJ. Quá trình liên kết các file \*.OBJ để tạo nên file chương trình mã máy \*.EXE chỉ được tiếp tục khi chúng ta hiệu đính và khử bỏ mọi lỗi error, còn lỗi warning chỉ là các cảnh báo, chương trình vẫn có thể được dịch và chạy mà không cần sửa các lỗi này. Tuy nhiên, người viết chương trình cũng nên sửa các lỗi warning.

- Loại lỗi thứ ba có thể xảy ra trong quá trình liên kết: Lỗi này thường xuất hiện khi ta sử dụng tới các lời gọi hàm nhưng những hàm đó mới chỉ tồn tại dưới dạng nguyên mẫu (function prototype) mà chưa được mô tả chi tiết các hàm, hoặc những lời hàm gọi chưa đúng với tên của nó. Lỗi này được khắc phục khi ta bổ sung đoạn chương trình con mô tả chi tiết cho hàm hoặc sửa đổi lại những lời gọi hàm tương ứng.

### Bước 3: Thực hiện chương trình

Chương trình được thực hiện bằng cách bấm tổ hợp phím CTRL+F9 (thực hiện trong môi trường soạn thảo TC.EXE) hoặc trở về môi trường DOS thực hiện như các chương trình bình thường khác. Nếu kết quả nhận được là sai thì lỗi đó thuộc lỗi thuật toán mà máy tính không thể phát hiện được loại lỗi kiểu này. Để kiểm tra tính đúng đắn của thuật toán, người lập trình thường sử dụng một số bộ giá trị đặc biệt của thông tin vào.

#### 4.2.2.3. Chương trình đơn giản nhất trong C

Ví dụ: Viết chương trình in ra dòng thông báo "Ngôn ngữ lập trình C".

Trước hết, ta phải tạo ra văn bản chương trình bằng cách soạn thảo sử dụng trình soạn thảo của Turbo C đó là TC.EXE, thông thường được đặt trong thư mục C:\TC\BIN. Trình soạn thảo của Turbo C gần giống với trình soạn thảo của Pascal chỉ khác nhau ở chỗ văn bản chương trình của được ngầm định phần mở rộng là \*.C. Trong khi soạn thảo cần chú ý ghi lại chương trình bằng phím F2 hoặc chọn thực đơn File/Save. Trong ví dụ này chúng ta đặt tên là *SmallPrg.c*

Dịch chương trình thành file *SmallPrg.EXE* bằng phím F9, nếu chúng ta muốn vừa dịch và thực hiện chương trình chỉ cần bấm tổ hợp phím CTRL + F9 và xem kết quả đưa ra màn hình. Trong trường hợp gặp lỗi, trình dịch của C sẽ thông báo lỗi để chúng ta chỉnh sửa và hiệu đính lại chương trình. Chương trình hiển thị lên màn hình dòng "Ngôn ngữ lập trình C" được viết đơn giản như sau:

**Ví dụ:**

```
#include <conio.h>
/* khai báo việc sử dụng các hàm printf(), getch() trong conio.h*/
void main(void)
{
printf("Ngôn ngữ lập trình C\n");/* in ra màn hình*/
getch();/* lệnh này chờ nhận một kí tự gõ vào*/
}
```

### **Kết quả thực hiện chương trình:** Dòng chữ được in ra

Ngôn ngữ lập trình C

Để tiếp tục hãy bấm tiếp một phím bất kì ta sẽ trở về với trình soạn thảo trong Turbo C.

Chỉ thị `#include` được gọi là chỉ thị tiền xử lý, có nhiệm vụ liên kết với tệp tương ứng được đặt trong hai kí tự < tên file đầu tệp >. File có dạng \*.h được C qui định là các file chứa nguyên mẫu của các hàm và thường được đặt trong thư mục C:\TC\INCLUDE. Như vậy, chỉ thị `#include <conio.h>` khai báo việc sử dụng các hàm trong file `conio.h`, trong trường hợp này ta sử dụng hàm `printf()` và `getch()`.

Một chương trình C, với bất kì kích thước nào, cũng đều bao gồm một hoặc nhiều "hàm", trong thân của hàm có thể là các lệnh hoặc lời gọi hàm, kết thúc một lệnh là kí tự ';'. Các lời gọi hàm sẽ xác định các thao tác tính toán thực tế cần phải thực hiện. Các hàm của C cũng tương tự như các hàm và chương trình con của một chương trình FORTRAN hoặc một thủ tục PASCAL. Trong ví dụ trên `main` cũng là một hàm như vậy. Thông thường chúng ta được tự do chọn lấy bất kì tên nào để đặt cho hàm, nhưng `main` là một tên đặc biệt, chương trình sẽ được thực hiện tại điểm đầu của `main`. Điều này có nghĩa là mọi chương trình trong C phải có một `main` ở đâu đó. `Main` sẽ khởi động các hàm khác để thực hiện công việc của nó, một số hàm nằm ở trong văn bản chương trình, một số khác nằm ở các thư viện của các hàm đã viết trước.

Một phương pháp trao đổi dữ liệu giữa các hàm được thực hiện thông qua đối của hàm. Các dấu ngoặc theo sau tên hàm bao quanh danh sách đối. Thông thường, mỗi hàm khi thực hiện đều trả về một giá trị, tuy nhiên cũng có hàm không có giá trị trả về. Kiểu giá trị trả về của hàm được viết đằng trước tên hàm. Nếu không có giá trị trả về thì từ khóa `void` được dùng để thay thế (`main` là hàm không có giá trị trả về). Dấu ngoặc nhọn `{ }` bao quanh các câu lệnh tạo nên thân của hàm, chúng tương tự như `Begin . . End` trong Pascal. Mọi chương trình trong C đều phải được bao trong `{ }` và không có dấu chấm phẩy ở cuối văn bản chương trình. Hàm được khởi động thông qua tên của nó, theo sau là danh sách các đối trong ngoặc. Nếu hàm không có đối thì phải viết các dấu ngoặc tròn cho dù trong ngoặc tròn để trống.

Dòng được viết

```
printf("Ngôn ngữ lập trình C\n");
```

Là một lời gọi tới hàm có tên `printf` với đối là một hằng xâu kí tự `"Ngôn ngữ lập trình C\n"`. `printf` là hàm thư viện để đưa kết quả ra trên màn hình (trừ khi xác định rõ thiết bị nhận là loại gì khác). Trong trường hợp này hàm sẽ cho hiển thị trên màn hình dãy kí tự tạo nên đối.

Dãy các kí tự bất kì nằm trong hai ngoặc kép `"...."` được gọi là một xâu kí tự hoặc một hằng kí tự. Hiện tại chúng ta chỉ dùng xâu kí tự như là đối của `printf` và một số hàm khác.

Dãy `\n` trong xâu kí tự trên là cú pháp của C để chỉ kí tự xuống dòng, báo hiệu lần đưa ra sau sẽ được thực hiện ở đầu dòng mới. Ngoài ra C còn cho phép dùng `\t` để chỉ dấu tab, `\b` cho việc lùi lại (backspace), `\"` cho dấu ngoặc kép, và `\\` cho bản thân dấu sổ chéo.

#### **4.2.3. Các kiểu dữ liệu cơ sở**

Một kiểu dữ liệu (Data Type) được hiểu là tập hợp các giá trị mà một biến thuộc kiểu đó có thể nhận được làm giá trị của biến cùng với các phép toán trên nó. Các kiểu dữ liệu cơ sở trong C

bao gồm kiểu các số nguyên (int, long), kiểu số thực (float, double), kiểu kí tự (char). Khác với Pascal, C không xây dựng nên kiểu Boolean, vì bản chất kiểu Boolean là kiểu nguyên chỉ nhận một trong hai giá trị khác 0 hoặc bằng 0.

Biến kiểu char có kích cỡ 1 byte dùng để biểu diễn 1 kí tự trong bảng mã ASCII, thực chất là số nguyên không dấu có giá trị từ 0 đến 255. Chúng ta sẽ còn thảo luận kỹ hơn về kiểu dữ liệu char trong những phần tiếp theo.

Biến kiểu số nguyên có giá trị là các số nguyên và các số nguyên có dấu (âm, dương) int, long int (có thể sử dụng từ khoá signed int, signed long), kiểu số nguyên không dấu unsigned int, unsigned long. Sự khác biệt cơ bản giữa int và long chỉ là sự khác biệt về kích cỡ.

Biến có kiểu float biểu diễn các số thực có độ chính xác đơn.

Biến có kiểu double biểu diễn các số thực có độ chính xác kép.

Sau đây là bảng các giá trị có thể của các kiểu dữ liệu cơ bản của C:

Kiểu	Miền xác định	Kích thước
char	0.. 255	1 byte
int	-32767 .. 32767	2 byte
long	-2147483648..2147483647	4 byte
unsigned int	0 .. 65535	2 byte
unsigned long	0 .. 2147483647*2=4294967295	4 byte
float	3. 4e-38 .. 3.4e + 38	4 byte
double	1.7e-308 .. 1.7e + 308	8 byte

Toán tử sizeof(tên\_kiểu) sẽ cho ta chính xác kích cỡ của kiểu tính theo byte. Chương trình sau sẽ in ra kích cỡ của từng kiểu dữ liệu cơ bản.

**Ví dụ:**

```

/* Chương trình kiểm tra kích cỡ các kiểu dữ liệu cơ bản*/
#include <stdio.h>
#include <conio.h>
void main(void){
    clrscr(); /* hàm xoá toàn bộ màn hình được khai báo trong stdio.h*/
    printf("\n Kích cỡ kiểu kí tự: %d", sizeof(char));
    printf("\n Kích cỡ kiểu số nguyên: %d", sizeof(int));
    printf("\n Kích cỡ kiểu số nguyên dài: %d", sizeof(long));
}
    
```



```
printf("\n Kích cỡ kiểu số thực: %d", sizeof(float));  
printf("\n Kích cỡ kiểu số thực có độ chính xác kép: %d", sizeof(double));  
getch();  
}
```

#### 4.2.4. Biến, hằng, câu lệnh và các phép toán

##### 4.2.4.1. Biến và hằng

- **Biến:** Biến là một đại lượng có giá trị thay đổi trong khi thực hiện chương trình. Mỗi biến có một tên và một địa chỉ của vùng nhớ dành riêng cho biến. Mọi biến đều phải khai báo trước khi sử dụng nó. Quy tắc khai báo một biến được thực hiện như sau:

Tên\_kiểu\_dữ\_liệu tên\_biến; trong trường hợp có nhiều biến có cùng kiểu, chúng ta có thể khai báo chung trên một dòng trong đó mỗi biến được phân biệt với nhau bởi một dấu phẩy và có thể gán giá trị ban đầu biến trong khi khai báo.

**Ví dụ :**

```
int a, b, c=0;  
/* khai báo 3 biến a, b, c có kiểu int trong đó c được gán là 0*/  
float e, f, g= 1.5; /* khai báo 3 biến e, f, g có kiểu float*/  
long i, j; /* khai báo i, j có kiểu long*/  
unsigned k,m; /* khai báo k,m có kiểu số nguyên dương*/  
char key; /* khai báo key có kiểu char*/
```

- **Hằng:** Hằng là đại lượng mà giá trị của nó không thay đổi trong thời gian thực hiện chương trình. C sử dụng chỉ thị #define để định nghĩa các hằng.

- Hằng có giá trị trong miền xác định của kiểu int là hằng kiểu nguyên (nếu không có l ở cuối).
- Hằng có giá trị trong miền xác định của kiểu int và có kí hiệu 0x ở đầu là hằng kiểu nguyên biểu diễn theo cơ số hệ 16 (0xFF).
- Hằng có giá trị trong miền xác định của kiểu long và có kí hiệu L (l) ở cuối cũng được coi là hằng kiểu long (135L).
- Hằng có giá trị trong miền xác định của kiểu long là hằng kiểu long
- Hằng có giá trị trong miền xác định của kiểu float là hằng kiểu số thực (3.14).
- Hằng có giá trị là một kí tự được bao trong dấu nháy đơn được gọi là hằng kí tự ('A').
- Hằng có giá trị là một dãy các kí tự được bao trong dấu nháy kép được gọi là hằng chuỗi kí tự "Hằng String".

**Ví dụ:**

```
#define MAX 100 /* định nghĩa hằng kiểu nguyên*/  
#define MIN 0xFF /* hằng nguyên biểu diễn theo cơ số hệ 16*/
```

```
#define N 123L /* hằng long*/
#define PI 3.414 /* hằng thực*/
#define KITU 'A' /* hằng kí tự */
#define STR "XAU KI TU" /*hằng xâu kí tự*/
```

#### 4.2.4.2. Câu lệnh

Câu lệnh là phần xác định công việc mà chương trình phải thực hiện để xử lý các dữ liệu đã được mô tả và khai báo. Trong C các câu lệnh cách nhau bởi dấu chấm phẩy. Câu lệnh được chia ra làm hai loại: câu lệnh đơn giản và câu lệnh có cấu trúc

Câu lệnh đơn giản là lệnh không chứa các lệnh khác như lệnh gán, lệnh gán được dùng để gán giá trị của biểu thức, một hằng vào một biến, phép gán được viết tổng quát như sau: biến = biểu thức. Hay lệnh gọi hàm void, hàm void là hàm không nhận giá trị quay trả lại vào tên hàm.

Câu lệnh có cấu trúc: Bao gồm nhiều lệnh đơn giản và có khi có cả lệnh cấu trúc khác bên trong. Các lệnh loại này như:

- + Cấu trúc lệnh khối ( lệnh ghép hay lệnh hợp)
- + Lệnh if
- + Lệnh switch
- + Các lệnh lặp: for, while, do... while

#### 4.2.4.3. Các phép toán

- Các phép toán số học: Gồm có: +, -, \*, / (cộng, trừ, nhân, chia), % (lấy phần dư). Phép chia (/) sẽ cho lại một số nguyên giống như phép chia nguyên nếu chúng ta thực hiện chia hai đối tượng kiểu nguyên.

##### Ví dụ:

```
int a=, b=5, c; /* khai báo ba biến nguyên*/
float d=3, e=2, f; /* khai báo ba biến thực*/
c = a + b; /* c có giá trị là 8*/
c = a - b; /* c có giá trị là -2*/
c = a / b ; /* c có giá trị là 0*/
c = a % b; /* c có giá trị là 3*/
f = d / e; /* f có giá trị là 1.5*/
```

Để tiện lợi trong viết chương trình cũng như giảm thiểu các kí hiệu sử dụng trong các biểu thức số học. C trang bị một số phép toán tăng và giảm mở rộng cho các số nguyên như sau:

```
a++ ⇔ a = a + 1
a-- ⇔ a = a - 1
++a ⇔ a = a + 1
```

```
--a  ⇔  a = a - 1
a+=n ⇔  a = a + n
a-=n ⇔  a = a - n
a/=n ⇔  a = a / n
a*=n ⇔  a = a * n
a%=n ⇔  a = a % n
```

Chú ý: Mặc dù ++a và a++ đều tăng a lên một đơn vị, nhưng khi thực hiện các biểu thức so sánh, ++a sẽ tăng a trước rồi thực hiện so sánh, còn a++ sẽ so sánh trước sau đó mới tăng a. Tình huống sẽ xảy ra tương tự đối với --a và a--.

**Ví dụ 4.3:** Kiểm tra lại các phép toán số học trên hai số nguyên a và b;

```
#include <stdio.h>
#include <conio.h>
void main(void)
{
    int a=5, b=2;
    clrscr();
    printf("\ tổng a + b = %d", a + b);
    printf("\ hiệu a - b = %d", a - b);
    printf("\ tích a * b = %d", a * b);
    printf("\ thương a / b = %d", a / b);
    /* thương hai số nguyên sẽ là một số nguyên*/
    printf("\ phần dư a % b = %d", a % b);
    a++; b--; /* a = a + 1; b = b - 1; */
    printf("\n giá trị của a, b sau khi tăng (giảm): a = %d b = %d", a, b);
    a+=b; /* a=a+b;*/
    printf("\n giá trị của a sau khi tăng b đơn vị: a = %d", a);
    a-=b; /* a = a b*/
    printf("\n giá trị của a sau khi trừ b đơn vị: a = %d", a);
    a*=b; /* a = a*b;*/
    printf("\n giá trị của a sau khi nhân b đơn vị: a = %d", a);
    a/=b; /* a = a/b; */
```

```
printf("\n giá trị của a sau khi chia b đơn vị: a =%d", a);
a %=b;    /* a = a %b; */
printf("\n giá trị của a sau khi lấy modul b : a =%d", a);
getch();
}
```

- Các phép toán so sánh: Gồm có các phép  $>$ ,  $<$ ,  $>=$ ,  $<=$ ,  $==$ ,  $!=$  ( lớn hơn, nhỏ hơn, lớn hơn hoặc bằng, nhỏ hơn hoặc bằng, đúng bằng, khác).\

**Ví dụ:**

```
if ( a>b) { . . } /* nếu a lớn hơn b*/
if ( a>=b) { . . } /* nếu a lớn hơn hoặc bằng b*/
if ( a==b) { . . } /* nếu a đúng bằng b*/
if ( a!=b) { . . } /* nếu a khác b*/
```

- Các phép toán logic

$&&$  : Phép và logic chỉ cho giá trị đúng khi hai biểu thức tham gia đều có giá trị đúng (giá trị đúng của một biểu thức trong C được hiểu là biểu thức có giá trị khác 0).

$\|$  : Phép hoặc logic chỉ cho giá trị sai khi cả hai biểu thức tham gia đều có giá trị sai.

$^$  : Phép phủ định cho giá trị đúng nếu biểu thức có giá trị sai và ngược lại cho giá trị sai khi biểu thức có giá trị đúng.

**Ví dụ:**

```
int a =3, b =5;
if ( ( a !=0) && (b!=0) ) /* nếu a khác 0 và b khác 0*/
if ((a!=0) || (b!=0)) /* nếu a khác 0 hoặc b khác 0*/
if ( !(a) ) /* phủ định a khác 0*/
```

- Các toán tử thao tác bit

Các toán tử thao tác bit không sử dụng cho float và double:

$\&$  : Phép hội các bit.

$|$  : Phép tuyển các bit.

$^$  : Phép tuyển các bit có loại trừ.

$<<$  : Phép dịch trái (dịch sang trái n bit giá trị 0)

$>>$  : Phép dịch phải (dịch sang phải n bit có giá trị 0)

$\sim$  : Phép lấy phần bù.

**Ví dụ:**

Giả sử  $a=3$ ,  $b=5$  khi đó  $c = a \& b$  cho ta kết quả là 1:

0000.0000.0000.0011  $a=3$

& 000.0000.0000.0101  $b=5$

0000.0000.0000.0001  $c=1$

$c = a | b$ ; cho ta kết quả là 7;

0000.0000.0000.0011  $a=3$

| 0000.0000.0000.0101  $b=5$

0000.0000.0000.0111  $c=7$

$c = a \wedge b$ ; cho ta kết quả là 6;

0000.0000.0000.0011  $a=3$

^ 0000.0000.0000.0101  $b=5$

0000.0000.0000.0110  $c=6$

$c = \sim a$ ; cho ta kết quả là 65532;

~ 0000.0000.0000.0011  $a=3$

1111.1111.1111.1100  $c=65532$

$c = a \ll b$ ; cho ta kết quả là  $3*3*3*3*3=243$  ;

$c = a \gg b$ ; cho ta kết quả là 0;

- Toán tử chuyển đổi kiểu

Ta có thể dùng toán tử chuyển đổi kiểu để nhận được kết quả tính toán như mong muốn. Qui tắc chuyển đổi kiểu được thực hiện theo qui tắc: (kiểu) biến

**Ví dụ:** Tính giá trị phép chia hai số nguyên  $a$  và  $b$ .

```
#include <stdio.h>
{
int a=3, b=5;
float c;
c = (float) a / (float) b;
printf("\n thương c = a / b =%.2f", c);
getch();
}
```

- Thứ tự ưu tiên các phép toán

Khi viết một biểu thức, chúng ta cần lưu ý tới thứ tự ưu tiên tính toán các phép toán, các bảng tổng hợp sau đây phản ánh trật tự ưu tiên tính toán của các phép toán số học và phép toán so sánh.

Bảng tổng hợp thứ tự ưu tiên tính toán các phép toán số học và so sánh

Tên toán tử	Chiều tính toán
( ), [ ], ->	Trái -> Phải
-, ++, --, !, ~, sizeof()	Phải -> Trái
*, /, %	Trái -> Phải
+, -	Trái -> Phải
>>, <<	1.2.4.2.1    Trái    -> Phải
<, <=, >, >=,	Trái -> Phải
== !=	Trái -> Phải
&	Trái -> Phải
^	Trái -> Phải
	Trái -> Phải
&&	Trái -> Phải
	Trái -> Phải
?:	Phải -> Trái
=, +=, -=, *=, /=, %=, &=, ^=,  =, <<=, >>=	Phải -> Trái

## 4.2.5. Thủ tục vào và ra chuẩn

### 4.2.5.1. Vào ra bằng `getchar()`, `putchar()`

Cơ chế vào đơn giản nhất là đọc từng kí tự từ thiết bị vào chuẩn (bàn phím, màn hình) bằng `getchar`. Mỗi khi được gọi tới `getchar()` sẽ cho kí tự đọc vào tiếp theo. `getchar` cho giá trị EOF khi nó gặp cuối tệp trên bất cứ cái vào nào đang được đọc. Thư viện chuẩn định nghĩa hằng kí hiệu EOF là -1 (với `#define` trong tệp `stdio.h`) nhưng các phép kiểm tra phải viết dưới dạng EOF chứ không là -1 để cho độc lập với giá trị cụ thể.

Để đưa ra, `putchar(c)` sẽ đặt kí tự trên “thiết bị ra chuẩn”, cũng có giá trị mặc định là màn hình.

Việc đưa ra cho `printf` cũng chuyển tới thiết bị ra chuẩn, các lời gọi tới `putchar` và `printf` có thể chen lẫn nhau.

Nhiều chương trình chỉ đọc trên một thiết bị vào và viết trên một thiết bị ra; với việc vào và ra của các chương trình thì sử dụng `getchar`, `putchar` kết hợp với `printf` là hoàn toàn thích hợp và đầy đủ. Điều này đặc biệt đúng khi làm việc với tệp và sử dụng công cụ đường ống nối đầu ra của

chương trình này thành đầu vào của chương trình tiếp. Chẳng hạn, xét chương trình lower, chuyển kí tự vào của nó thành chữ thường:

```
#include <stdio.h>
void main() /*chuyển kí tự vào thành chữ thường*/
{
    int c;
    while ((c = getchar()) != EOF)
        putchar(isupper(c) ? tolower(c) : c);
}
```

Các “hàm” isupper và tolower thực tế là các macro được xác định trong stdio.h, macro isupper kiểm tra xem đối của nó là chữ hoa hay không, cho giá trị khác 0 nếu đúng như vậy và cho 0 nếu nó là chữ thường. marco tolower chuyển chữ hoa thành chữ thường. Ta không cần biết tới cách các hàm này được cài đặt thế nào trên máy cụ thể, hành vi bên ngoài của chúng như nhau cho nên các chương trình sử dụng chúng không cần để ý tới tập kí tự.

Ngoài ra, trong thư viện vào/ ra chuẩn “các hàm” getchar và putchar là các macro và do vậy tránh được tổn phí về lời gọi hàm cho mỗi kí tự.

#### 4.2.5.2. In ra theo khuôn dạng - Printf

Hai hàm printf để đưa ra và scanf để nhập vào cho phép chuyển ra các biểu diễn kí tự và số. Chúng cũng cho phép sinh ra hoặc thông dịch các dòng có khuôn dạng. Trong các chương trước, chúng ta đã dùng printf một cách hình thức mà chưa có những giải thích đầy đủ về nó. Bây giờ chúng ta sẽ mô tả đầy đủ và chính xác hơn cho hàm này.

```
printf (control, arg1, arg2,...)
```

printf chuyển, tạo khuôn dạng, và in các đối của nó ra thiết bị ra chuẩn dưới sự điều khiển của xâu control. Xâu điều khiển của control đều được đưa vào bằng kí tự % và kết thúc bởi một kí tự chuyển dạng. Giữa % và kí tự chuyển dạng có thể có. Dấu trừ (-), xác định việc dòn trái cho đối được chuyển dạng trong trường.

Xâu chữ số xác định chiều ngang tối thiểu của trường. Số được chuyển dạng sẽ được in ra trong trường tối thiểu với chiều ngang này, và sẽ rộng hơn nếu cần thiết. Nếu đối được chuyển có ít kí tự hơn là chiều ngang của trường thì nó sẽ được bổ sung thêm kí tự vào bên trái (hoặc phải, nếu có cả chỉ báo dòn trái) để cho đủ chiều rộng trường. Kí tự bổ xung thêm sẽ là dấu trống thông thường hoặc số 0 nếu chiều ngang trường được xác định với số 0 đứng đầu.

Dấu chấm phân tách chiều ngang trường với xâu chữ số tiếp.

Xâu chữ số (độ chính xác) xác định ra số cực đại các kí tự cần phải in ra từ một xâu, hoặc số các chữ số cần phải in ra ở bên phải dấu chấm thập phân của float hay double.

Bộ thay đổi chiều dài l (chữ ell) chỉ ra rằng phần dữ liệu tương ứng là long chứ không phải là int.

Sau đây là các kí tự chuyển dạng và nghĩa của nó là:

- d Đối được chuyển sang kí pháp thập phân.
- o Đối được chuyển sang kí pháp hệ tám

x Đối được chuyển sang cú pháp hệ mười sáu không dấu(không có 0x đứng trước).

u Đối được chuyển sang kí pháp thập phân không dấu

c Đối được coi là một kí tự riêng biệt.

s Đối là xâu kí tự; các kí tự trong xâu được in cho tới khi gặp kí tự không hoặc cho tới khi đủ số lượng kí tự được xác định bởi đặc tả về độ chính xác.

e Đối được xem là float hoặc double và được chuyển sang kí pháp thập phân có dạng[-]m.nnnnnE[+]xx với độ dài của xâu chứa n do độ chính xác xác định. Độ chính xác mặc định là 6.

f Đối được xem là float hoặc double và được chuyển sang kí pháp thập phân có dạng [-]mmm.nnnnn với độ dài của xâu các n do độ chính xác xác định. Độ chính xác mặc định là 6. Lưu ý rằng độ chính xác không xác định ra số các chữ số có nghĩa phải in theo khuôn dạng f.

g Dùng %e hoặc %f, tùy theo loại nào ngắn hơn; không in các số không vô nghĩa.

Nếu kí tự đứng sau % không phải là kí tự chuyển dạng thì kí tự đó sẽ được in ra; vậy % sẽ được in ra bởi %%.

Phần lớn các chuyển dạng là hiển nhiên, và đã được minh họa ở các chương trước. Một biệt lệ là độ chính xác liên quan tới các xâu. Bảng sau đây sẽ chỉ ra hiệu quả của các loại đặc tả trong việc in “hello, world” (12 kí tự). Chúng ta đặt dấu hai chấm xung quanh chuỗi kí tự in ra để có thể thấy sự trải rộng của nó

```

: %10s:      :hello, world:
: %-10s:     :hello, world:
: %20s:      :hello, world :
: %-20s:     :hello, world:
: %20.10s:   :      hello, world:
: %-20.10s:  :hello, word:
: %.10s      :hello,word:
    
```

Lưu ý: printf dùng đối thứ nhất của nó để quyết định xem có bao nhiêu đối theo sau và kiểu của chúng là gì. Hàm sẽ bị lẫn lộn và ta sẽ nhận được câu trả lời vô nghĩa nếu không đủ số đối hoặc đối có kiểu sai.

#### 4.2.5.3. Nhập vào có khuôn dạng - Scanf

Hàm scanf là hàm tương tự printf, đưa ra nhiều chức năng chuyển dạng như của printf nhưng theo chiều ngược lại.

```
scanf(control, arg1, arg2,...)
```

scanf đọc các kí tự từ thiết bị vào chuẩn, thông dịch chúng tương ứng theo khuôn dạng được xác định trong control, rồi lưu trữ kết quả trong các đối còn lại. Đối điều khiển sẽ được mô tả sau đây; các đối khác đều phải là con trỏ để chỉ ra nơi dữ liệu chuyển dạng tương ứng cần được lưu trữ.



Xâu điều khiển thường chứa các đặc tả chuyển dạng, được dùng để thông dịch trực tiếp dãy vào. Xâu điều khiển có thể chứa:

Dấu cách, dấu tab hoặc dấu xuống dòng (“các kí tự khoảng trắng”), thường bị bỏ qua.

Các kí tự thông thường (khác%) được xem là ứng với kí tự khác khoảng trắng trong dòng vào.

Các đặc tả chuyển dạng, bao gồm kí tự %, kí tự cắt bỏ gán \*(tùy chọn), một số tùy chọn xác định ra chiều ngang cực đại của trường, và một kí tự chuyển dạng.

Đặc tả chuyển dạng điều khiển việc chuyển dạng cho trường vào tiếp theo. Thông thường kết quả sẽ được đặt vào biến được trỏ tới bởi đối tượng ứng. Tuy nhiên, nếu việc cắt bỏ gán được nêu ra bởi kí tự \* thì trường vào sẽ bị bỏ qua. Trường vào được xác định như một xâu các kí tự khác khoảng trắng và được kéo dài hoặc tới kí tự khoảng trắng tiếp hoặc cho tới khi chiều ngang của trường.

Kí tự chuyển dạng chỉ ra việc thông dịch cho trường vào; đối tượng xứng phải là một con trỏ theo yêu cầu của lời gọi bởi ngữ nghĩa giá trị của C. Các kí tự chuyển dạng sau đây là hợp pháp:

d nhận một số nguyên trong cái vào; đối tượng ứng phải là con trỏ nguyên.

o nhận số nguyên hệ tám trong cái vào (có hoặc không có số không đứng trước) đối tượng ứng phải là con trỏ nguyên.

x nhận số nguyên hệ mười sáu trong cái vào (có hoặc không có 0x đứng trước); đối tượng ứng phải là con trỏ nguyên.

h nhận số nguyên short trong cái vào; đối tượng ứng phải là con trỏ nguyên short.

c nhận một kí tự; đối tượng ứng phải là con trỏ kí tự; kí tự vào tiếp được đặt vào chỗ chỉ ra. Trong trường hợp này không bỏ qua các kí tự khoảng trắng; để đọc kí tự khác khoảng trắng tiếp tục dùng %1s.

s nhận một xâu kí tự; đối tượng ứng phải là con trỏ kí tự trỏ tới bảng các kí tự đủ lớn để nhận được xâu và dấu kết thúc \0 sẽ được thêm vào.

f nhận số dấu phẩy động; đối tượng ứng phải là con trỏ tới float. Kí tự chuyển dạng e đồng nghĩa với f. Khuôn dạng vào cho float là một dấu tùy chọn, một xâu các số có thể chứa dấu chấm thập phân và một trường mũ tùy chọn chứa E hoặc e theo sau là một số nguyên có dấu.

Có thể viết thêm l (chữ ell) vào trước kí tự chuyển dạng d, o và x để chỉ ra rằng con trỏ tới long chứ không phải là int sẽ xuất hiện trong danh sách đối. Tương tự, có thể đặt l vào trước các kí tự chuyển dạng e hoặc f để chỉ ra rằng con trỏ trỏ tới double chứ không trỏ tới float trong danh sách đối.

Chẳng hạn, lời gọi

```
int i;
```

```
float x;
```

```
char name[50];
```

```
scanf ("%d %f %s", &i, &x, name);
```

Với dòng vào

### 25 54.32 E-1 Thompson

Sẽ gán giá trị 25 cho i, giá trị 5.432 cho x và chuỗi "Thompson" có cả dấu kết thúc \0, cho name. Ba trường vào có thể cách nhau bởi nhiều dấu cách, dấu tab và dấu xuống dòng.

Lời gọi

```
int i;
float x;
char name[50];
scanf ("%2d %f %*d %2s", &i, &x, name);
```

Với đầu vào

### 56789 0123 45a72

Sẽ gán 56 cho i, gán 789.0 cho x, nhảy qua 0123 và đặt chuỗi "45" vào name. Lời gọi tiếp tới bất kì trình vào nào cũng sẽ bắt đầu tìm tới chữ a. Trong hai ví dụ trên, name là con trỏ và do vậy phải không được đứng sau &.

Xét ví dụ khác, chương trình bàn tính thô sơ có thể được viết với scanf để thực hiện chuyển dạng cái vào

```
#include <stdio.h>
main() /*bàn tính thô sơ*/
{
    double sum, v;
    sum = 0;
    while(scanf ("%1f", &v) != EOF)
        printf ("\t%.2f\n", sum + = v);
}
```

scanf dừng lại khi nó vét hết chuỗi điều khiển hoặc khi dữ liệu vào nào đó không sánh đúng với đặc tả điều khiển. Hàm này cho giá trị là số các khoản mục vào đã được đối sánh đúng và được gán. Do vậy có thể dùng giá trị của hàm để xác định xem đã tìm thấy bao nhiêu dữ liệu vào. Khi gặp dấu hiệu cuối tệp hàm sẽ cho giá trị EOF, lưu ý rằng giá trị này khác 0, giá trị 0 có nghĩa là kí tự vào tiếp sẽ không đối sánh đúng với đặc tả đầu tiên trong chuỗi điều khiển. Lời gọi tiếp tới scanf sẽ tìm đọc tiếp ngay sau kí tự cuối cùng vừa cho.

Điều cần lưu ý nhất là: các đối của scanf phải là con trỏ. Lỗi hay mắc nhất là viết

```
scanf ("%d", n);
```

Đúng ra phải là `scanf ("%d", &n);`

Tương tự như các hàm scanf và printf là sscanf và sprintf, thực hiện các phép chuyển dạng tương ứng nhưng làm việc trên các chuỗi chứ không trên tệp. Khuôn dạng tổng quát của chúng là:

```
sprintf(string, control, arg1, arg2,...)
sscanf(string, control, arg1, arg2,...)
```

sprintf tạo khuôn dạng cho các đối trong arg1, arg2, v.v... tương ứng theo control như trước, nhưng đặt kết quả vào string chứ không đưa ra thiết bị chuẩn. Tất nhiên string phải đủ lớn để nhận kết quả. Ví dụ: nếu name là một mảng kí tự và n là nguyên thì

```
sprintf (name, "temp%d", n);
```

Tạo ra một xâu có dạng tempnnn trong name với nnn là giá trị của n.

sscanf làm công việc ngược lại - nó nhòm vào string tương ứng theo khuôn dạng trong control và đặt các giá trị kết quả vào arg1, arg2... Các đối phải là con trỏ. Lời gọi

```
sscanf(name, "temp%d", n);
```

đặt cho n giá trị của xâu các chữ số đứng sau temp trong name.

#### 4.2.5.4. *Thâm nhập vào thư viện chuẩn*

Mỗi tệp gốc có tham trỏ tới hàm thư viện chuẩn đều phải chứa dòng khai báo

```
#include < tên_tệp_thư_viện >
```

Ở đầu tệp văn bản chương trình. Dấu ngoặc nhọn < tên\_tệp\_thư\_viện > thay cho dấu nhảy thông thường để chỉ thị cho trình biên dịch tìm kiếm tệp trong danh mục chứa thông tin tiêu đề chuẩn được lưu trữ trong thư mục include. Trong trường hợp chúng ta sử dụng kí tự "tên\_tệp\_thư\_viện" trình biên dịch sẽ tìm kiếm tệp thư viện tại thư mục hiện tại.

Chỉ thị #include "tên\_tệp\_thư\_viện" còn có nhiệm vụ chèn thư viện hàm của người sử dụng vào vị trí của khai báo. Trong ví dụ sau, chúng ta sẽ viết chương trình thành 3 tệp, tệp define.h dùng để khai báo tất cả các hằng sử dụng trong chương trình, tệp songuyen.h dùng khai báo nên nguyên mẫu của hàm và mô tả chi tiết các hàm giống như một thư viện của người sử dụng, tệp mainprg.c là chương trình chính ở đó có những lời gọi hàm từ tệp songuyen.h.

**Ví dụ:** Xây dựng một thư viện đơn giản mô tả tập thao tác với số nguyên bao gồm: tính tổng, hiệu, tích, thương, phần dư, ước số chung lớn nhất của hai số nguyên dương a, b.

```
/* Nội dung tệp define.h */
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#define F1 59
#define F2 60
#define F3 61
#define F4 62
#define F5 63
#define F6 64
#define F1 65
#define F10 68

/* Nội dung tệp songuyen.h */
void Init_Int( int *, int *);
```

```

/* Nội dung tệp mainprg.c */
#include "define.h"
#include "songuyen.c"
void thuchien(void){
int a, b, control = 0; char c; textmode(0);
do {
    clrscr();
    printf("\n Tập thao tác với số nguyên");
    printf("\n F1- Nhập hai số nguyên");
    printf("\n F2- Tổng hai số nguyên");
    printf("\n F3- Hiệu hai số nguyên");
    printf("\n F4- Tích hai số nguyên");
    printf("\n F5- Thương hai số nguyên");
    printf("\n F6- Phần dư hai số nguyên");
    printf("\n F7- UCLN hai số nguyên");
    printf("\n F10- Trở về");
    c = getch();
    switch(c) {
        case F1: Init_Int(&a, &b); control =1; break;
        case F2:
            if (control) Tong_Int(a, b);
            break;
        case F3:
            if (control) Hieu_Int(a, b);
            break;
        case F4:
            if (control) Tich_Int(a, b);
            break;
        case F5:
            if (control) Thuong_Int(a, b);
            break;
        case F6:
            if (control) Mod_Int(a, b);
            break;
        case F7:
            if (control) UCLN_Int(a, b);
            break;
    }
} while (c!=F10);
}
void main(void) {
    thuchien();
}

```

### 4.3. CÁC CẤU TRÚC LỆNH ĐIỀU KHIỂN

#### 4.3.1. Câu lệnh khối

Tập các câu lệnh được bao bởi hai dấu { . . . } được gọi là một câu lệnh khối. Về cú pháp, ta có thể đặt câu lệnh khối ở một vị trí bất kì trong chương trình. Tuy nhiên, nên đặt các câu lệnh khối ứng với các chu trình điều khiển lệnh như for, while, do . . . while, if . . . else, switch để hiển thị rõ cấu trúc của chương trình.

Ví dụ:

```
if (a > b) {  
    câu_lệnh;  
}
```

#### 4.3.2. Cấu trúc lệnh if

##### Dạng 1:

```
if ( biểu_thức )  
    câu_lệnh;
```

Nếu biểu thức có giá trị đúng thì thực hiện câu\_lệnh; Câu lệnh có thể hiểu là câu lệnh đơn hoặc câu lệnh khối, nếu câu lệnh là lệnh khối thì nó phải được bao trong { . . . }.

##### Dạng 2:

```
if (biểu_thức)  
    câu_lệnh_A;  
else  
    câu_lệnh_B;
```

Nếu biểu thức có giá trị đúng thì câu\_lệnh\_A sẽ được thực hiện, nếu biểu thức có giá trị sai thì câu\_lệnh\_B sẽ được thực hiện.

**Dạng 3:** Được sử dụng khi có nhiều lệnh if lồng nhau hoặc phải kiểm tra nhiều biểu thức khác nhau.

```
if (biểu_thức_1)  
    câu_lệnh_1;  
else if (biểu_thức_2)  
    câu_lệnh_2;  
.....  
else if (biểu_thức_k)  
    Câu_lệnh_k;  
else  
    câu_lệnh_k+1;
```

Nếu biểu thức thứ i có giá trị đúng ( $0 < i \leq k$ ) thì câu\_lệnh\_i sẽ được thực hiện, nếu không biểu thức nào có giá trị đúng thì câu\_lệnh\_k+1 sẽ được thực hiện.

**Ví dụ:** Tìm số lớn nhất trong hai số a và b:

```
#include <stdio.h>  
void main(void){
```

```

float a, b, max;
printf("\n Nhập a="); scanf("%f", &a); /* nhập giá trị cho biến a*/
printf("\n Nhập b="); scanf("%f", &b); /* nhập giá trị cho biến b*/
if (a>b) max=a;
else max= b;
printf("\n Max(a,b)=%6.2f",max);
getch();
}

```

**Ghi chú:**

Toán tử: &(tên\_biến) lấy địa chỉ của biến. Câu lệnh scanf("%f",&a) có nghĩa là nhập một số thực vào địa chỉ ô nhớ dành cho biến a.

**Ví dụ:** Viết chương trình giải phương trình bậc 2 :  $ax^2 + bx + c = 0$

```

#include <stdio.h>
#include <conio.h>
#include <math.h>
void main(void){
    float a, b, c, x1, x2, delta;
    clrscr();
    printf("\n Giải phương trình bậc 2:");
    /*đọc các hệ số a, b, c từ bàn phím*/
    printf("\n Nhập hệ số a="); scanf("%f",&a);
    printf("\n Nhập hệ số b="); scanf("%f",&b);
    printf("\n Nhập hệ số c="); scanf("%f",&b);
    /* tính delta = b2 - 4ac*/
    delta=b*b-4*a*c;
    if (delta==0){
        printf("\n phương trình có 2 nghiệm kép x1=x2=%f", -b/(2*a));
    }
    else if(delta>0){
        printf("\n Phương trình có hai nghiệm");
        x1= (-b + sqrt(delta) ) / (2*a);
        x2= (-b - sqrt(delta) ) / (2*a);
        printf(" x1 = %6.2f x2=%6.2f", x1,x2);
    }
    else {
        printf("\n Phương trình có nghiệm phức:");
        x1 = -b / (2 a); / phần thực */
        x2 = ( sqrt( -delta) ) / (2 * a);
        printf(" Real = %6.2f Im = %6.2f", x1, x2);
    }
    getch();
}

```

### 4.3.3. Cấu trúc lệnh switch

Cấu trúc lệnh if thực hiện một phương án đúng trong hai phương án có thể xảy ra. Cấu trúc lệnh switch dùng để lựa chọn và thực hiện các phương án đúng có thể xảy ra.

Cú pháp

```
switch(biểu_thức_nguyên){
    case hằng_nguyên_1: câu_lệnh_1; break;
    case hằng_nguyên_2: câu_lệnh_2; break;
    case hằng_nguyên_3: câu_lệnh_3; break;
    .....
    case hằng_nguyên_n: câu_lệnh_n; break;
    default: câu_lệnh_n+1; break;
}
```

Thực hiện: Nếu biểu\_thức\_nguyên có giá trị trùng với hằng\_nguyên\_i thì câu\_lệnh\_i trở đi sẽ được thực hiện cho tới khi nào gặp từ khoá break để thoát khỏi chu trình.

**Ví dụ:** Nhập một số nguyên dương từ bàn phím và xác định xem số nguyên đó có phải là các số từ 1..10 hay không? Trong trường hợp không phải là các số nguyên từ 1..10 hãy đưa ra thông báo "số lớn hơn 10".

```
#include <stdio.h>
#include <conio.h>
void main(void){
    int n; clrscr();
    printf("\n Nhập n=");scanf("%d",&n);
    switch(n){
        case 1:
        case 2:
        case 3:
        case 4:
        case 5:
        case 6:
        case 7:
        case 8:
        case 9:
        case 10: printf("\n Số từ 1..10"); break;
        default : printf("\n Số lớn hơn 10"); break;
    }
}
```

### 4.3.4. Vòng lặp for

**Cú pháp:**

```
for(biểu_thức_1; biểu_thức_2; biểu_thức_3)
    Câu_lệnh;
```

Câu\_lệnh: Có thể là lệnh đơn hoặc lệnh khối, nếu là lệnh đơn thì câu lệnh trong thân chu trình for không cần thiết phải bao trong hai kí hiệu {, }. Nếu là lệnh khối (thân chu trình for có

hơn một lệnh) thì nó phải được bao trong hai kí hiệu {, }.

Thực hiện:

**Biểu\_thức\_1:** Được gọi là biểu thức khởi đầu có nhiệm vụ khởi đầu các biến sử dụng trong chu trình, biểu\_thức\_1 chỉ được thực hiện duy nhất một lần khi bắt đầu bước vào chu trình. Nếu trong chu trình phải khởi đầu nhiều biểu thức thì mỗi biểu thức được phân biệt với nhau bởi một kí tự '!'.

**Biểu\_thức\_2:** Được gọi là biểu thức kiểm tra và được thực hiện ngay sau khi thực hiện xong biểu\_thức\_1, nếu biểu thức kiểm tra có giá trị đúng (khác 0) thì câu lệnh trong thân của chu trình for sẽ được thực hiện, nếu biểu thức kiểm tra có giá trị sai thì điều khiển của chương trình chuyển về lệnh kế tiếp ngay sau thân của chu trình for.

**Biểu\_thức\_3:** Được gọi là biểu thức khởi đầu lại có nhiệm vụ khởi đầu lại các biến trong chu trình và được thực hiện ngay sau khi thực hiện xong câu\_lệnh. Chu trình sẽ được lặp lại bằng việc thực hiện biểu thức kiểm tra.

**Ví dụ:** Viết chương trình in ra màn hình dãy các kí tự theo dạng sau:

```
A   B   C   D . . . Z
a   b   c   d . . . z
Z   Y   X   W . . . A
z   y   X   w . . . a
```

/\* chương trình in dãy kí tự \*/

```
#include <stdio.h>
void main(void){
    char ch; clrscr();
    for(ch ='A'; ch<='Z'; ch++)
        printf("%3c",ch);
    printf("\n");
    for(ch ='a'; ch<='z'; ch++)
        printf("%3c",ch);
    printf("\n");
    for(ch ='Z'; ch>='A'; ch--)
        printf("%3c",ch);
    printf("\n");
    for(ch ='z'; ch>='a'; ch--)
        printf("%3c",ch);
    printf("\n");getch();
}
```

**Ghi chú:** Đối với ngôn ngữ C, kiểu dữ liệu char thực chất là một số nguyên có kích cỡ 1 byte, giá trị của byte là vị trí của kí tự trong bảng mã ASCII. Do vậy, chương trình trên có thể viết lại bằng cách sau:

**Ví dụ:**

```
/* chương trình in dãy kí tự */
#include <stdio.h>
void main(void){
```



```
int ch; clrscr();
for(ch =65; ch<=90; ch++)
printf("%3c",ch);
printf("\n");
for(ch =97; ch<=122; ch++)
printf("%3c",ch);
printf("\n");
for(ch ='Z'; ch>='A'; ch--)
    printf("%3c",ch);
printf("\n");
for(ch ='z'; ch>='a'; ch--)
    printf("%3c",ch);
printf("\n");getch();
}
```

**Ví dụ:** Viết chương trình giải bài toán cô "Trăm trâu trăm cỏ".

```
#include <stdio.h>
#include <conio.h>
void main(void) {
    unsigned int x, y, z; /* khai báo số trâu đứng, trâu nằm, trâu già*/
    for( x=0;x<=20;x++){
        for(y=0;y<=33;y++){
            for(z=0;z<100;z+=3){
                if(x + y + z ==100 && (5*x + 3 *y + ( z / 3))==100){
                    printf("\n Trâu đứng:%5d",x);
                    printf(" Trâu nằm:%5d ",y);
                    printf(" Trâu già:%5d", z);
                }
            }
        }
    }
}
```

#### 4.3.5. Vòng lặp không xác định while

**Cú pháp:**

```
while(biểu_thức)
    câu_lệnh;
```

Trong khi biểu thức còn đúng thì câu lệnh sẽ được thực hiện, nếu biểu thức có giá trị sai điều kiện của chương trình chuyển về lệnh kế tiếp ngay sau thân của while. Nếu trong thân của while có nhiều hơn một lệnh thì nó phải được bao trong hai ký tự { . . }.

**Ví dụ:** Đếm số chữ số, khoảng trắng (space), dấu tab, dấu về đầu dòng và những ký tự khác được

nhập từ bàn phím.

```
#include <conio.h>
#include <stdio.h>
#define ESC 27 /* mã của phím ESC*/
#define ENTER 13
void main(void){
    int number=0, space=0, tab=0, enter=0, other=0;
    char ch;

    clrscr();
    while( ( ch=getch() ) != ESC){ /* thực hiện nếu không phải là ESC*/
        if(ch>='0' && ch <='9')
            number++;
        else if(ch == ' ') space++;
        else if(ch == '\t') tab++;
        else if(ch == ENTER) enter ++;
        else other++;
    }
    printf("\n Số chữ số là: %d", number);
    printf("\n Số dấu trống là: %d", space);
    printf("\n Số dấu tab là: %d", tab);
    printf("\n Số dấu xuống dòng là: %d", enter);
    printf("\n Các kí tự khác: %d", other);
}
```

**Ví dụ:** Tìm tổng  $S = 1 + 1/3 + 1/5 + \dots + 1/(2n-1)$  với độ chính xác  $e$  ( $1/n \geq e$ );

```
#include <stdio.h>
#include <conio.h>
void main(void){
    int i=1;
    float s=0, epsilon;
    clrscr();
    printf("\n Nhập độ chính xác epsilon="); scanf("%f",&epsilon);
    while( ( (float) 1 / (float) i ) >= epsilon) {
        s+=(float) 1 / (float) i;
        i+=2;
    }
    printf("\n Tổng s=%6.2f", s);
    getch();
}
```

**Ví dụ:** Tính  $e^x$  theo công thức xấp xỉ chuỗi taylor với  $e = x^n/n!$ .

$$e^x = 1 + x/1! + x^2/2! + x^3/3! + \dots + x^n/n!$$

```
#include <stdio.h>
#include <conio.h>
void main(void){
    float e_mu_x, epsilon, x, t;
    int n; clrscr();
    printf("\n Nhập x="); scanf("%f", &x);
    printf("\n Nhập độ chính xác epsilon="); scanf("%f", &epsilon);
    e_mu_x = 1; n = 1; t = x;
    while ( t >=epsilon) {
        e_mu_x += t;
        n++; t = t * (x/n);
    }
    printf("\n e mũ %6.3f = %6.3f", x, e_mu_x);
    getch();
}
```

#### 4.3.6. Vòng lặp không xác định do .. while

##### Cú pháp:

```
do {
    câu_lệnh;
} while(biểu_thức);
```

Thực hiện câu lệnh trong khi biểu\_thức vẫn còn đúng, nếu biểu thức có giá trị sai, điều khiển chương trình chuyển về lệnh kế tiếp ngay sau while(biểu\_thức).

**Ví dụ:** Viết chương trình xây dựng tập thao tác cộng, trừ, nhân, chia, lấy phần dư của hai số nguyên a,b.

```
#include <stdio.h>
#include <conio.h>
#include <dos.h> /* sử dụng hàm delay()*/
#define F1 59 /* định nghĩa phím F1 */
#define F2 60 /* định nghĩa phím F2 */
#define F3 61 /* định nghĩa phím F3 */
#define F4 62 /* định nghĩa phím F4 */
#define F5 63 /* định nghĩa phím F5 */
#define F6 64 /* định nghĩa phím F6 */
#define F10 68 /* định nghĩa phím F10 */
void main(void){
    int a, b, control=0; char key;
    clrscr();
    do {
        printf("\n Tập thao tác với hai số nguyên a, b");
        printf("\n F1- Nhập hai số nguyên a,b");
        printf("\n F2-Tổng hai số nguyên");
        printf("\n F3-Hiệu hai số nguyên");
```

```

printf("\n F4- Tích hai số nguyên");
printf("\n F5- Thương hai số nguyên");
printf("\n F6- Modul hai số nguyên");
printf("\n F10- Trở về");
key = getch();
switch(key) {
    case F1:
        printf("\n Nhập a="); scanf("%d", &a);
        printf("\n Nhập b="); scanf("%d", &b);
        control =1; break;
    case F2:
        if( control !=0 )
            printf("\n Tổng a + b =%d", a+b);
        break;
    case F3:
        if( control !=0 )
            printf("\n Hiệu a - b =%d", a - b);
        break;
    case F4:
        if( control !=0 )
            printf("\n Tích a * b =%d", a * b);
        break;
    case F5:
        if( control !=0 )
            printf("\nThương a*b=%6.2f",(float)a/ (float)b);
        break;
}
clrscr();
} while(key!=F10);
}

```

#### 4.4. HÀM VÀ PHẠM VI HOẠT ĐỘNG CỦA BIẾN

##### 4.4.1. Tính chất của hàm

Hàm (function) hay nói đúng hơn là chương trình con (sub\_program) chia cắt các nhiệm vụ tính toán lớn thành các công việc nhỏ hơn và có thể sử dụng nó ở mọi lúc trong chương trình, đồng thời hàm cũng có thể được cung cấp cho nhiều người khác sử dụng dưới dạng thư viện mà không cần phải bắt đầu xây dựng lại từ đầu. Các hàm thích hợp còn có thể che dấu những chi tiết thực hiện đối với các phần khác trong chương trình, vì những phần này không cần biết hàm đó thực hiện như thế nào.

Một chương trình C nói chung bao gồm nhiều hàm nhỏ chứ không phải là một vài hàm lớn. Chương trình có thể nằm trên một hoặc nhiều tệp gốc theo mọi cách thuận tiện; các tệp gốc có thể được dịch tách bạch và nạp vào cùng nhau, cùng với các hàm đã được dịch từ trước trong thư viện. Sau đây là một số tính chất cơ bản của hàm:

- Hàm có thể có kiểu hoặc vô kiểu, kiểu ở đây được hiểu là kiểu giá trị trở về của hàm. Kiểu giá trị trở về của hàm có thể là kiểu cơ bản (base type) hoặc có kiểu do người dùng định nghĩa (user type). Trong trường hợp hàm vô kiểu C sử dụng từ khoá void để chỉ lớp các hàm kiểu này.

- Hàm có thể có biến hoặc không có biến. Trong trường hợp hàm không có biến C sử dụng từ khoá void để chỉ lớp hàm dạng này. Một lời gọi hàm có nghĩa khi và chỉ khi hàm nhận được đầy đủ giá trị các biến của nó một cách tường minh.

- Giá trị trở về của hàm được thực hiện bằng lệnh return(giá trị), giá trị trở về của hàm phải phù hợp với kiểu của hàm. Trong trường hợp hàm vô kiểu ta có thể sử dụng lệnh return hoặc bỏ qua lệnh return;

- Hàm có thể làm thay đổi nội dung của biến hoặc không làm thay đổi nội dung của biến được truyền cho hàm từ chương trình chính. Nếu ta truyền cho hàm là địa chỉ của biến thì mọi thao tác đối với biến trong hàm đều có thể dẫn tới sự thay đổi nội dung của biến trong chương trình chính, cơ chế này được gọi là cơ chế truyền tham biến cho hàm. Nếu ta truyền cho hàm là nội dung của biến thì mọi sự thay đổi nội dung của biến trong hàm không dẫn tới sự thay đổi nội dung của biến trong chương trình chính, cơ chế này được gọi là cơ chế truyền tham trị.

#### 4.4.2. Khai báo, thiết kế hàm

Mọi hàm trong C dù là nhỏ nhất cũng phải được thiết kế theo nguyên tắc sau:

Kiểu\_hàm Tên\_hàm ( Kiểu\_1 biến\_1, Kiểu\_2 biến\_2, . . . )

```
{ Khai báo biến cục bộ trong hàm;  
  Câu_lệnh_hoặc_dãy_câu_lệnh;  
  return(giá_trị);  
}
```

Ghi chú: Trước khi sử dụng hàm cần phải khai báo nguyên mẫu cho hàm (function prototype) và hàm phải phù hợp với nguyên mẫu của chính nó. Nguyên mẫu của hàm thường được khai báo ở phần đầu chương trình theo cú pháp như sau:

Kiểu\_hàm Tên\_hàm ( Kiểu\_1, Kiểu\_2 , . . . );

**Ví dụ:** Viết chương trình tìm USCLN của hai số nguyên dương a, b.

```
/* Ví dụ về hàm trả lại một số nguyên int*/
```

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
/* khai báo nguyên mẫu cho hàm; ở đây hàm USCLN trả lại một số nguyên và có hai biến kiểu nguyên */
```

```
int USCLN( int , int ); /* mô tả hàm */
```

```
int USCLN( int a, int b)
```

```
{ while(a!=b){
```

```

        if ( a >b )    a = a -b;
        else    b = b-a;    }

    return(a);}

/* chương trình chính */
void main(void) {
    unsigned int    a, b;    clrscr();
    printf("\n Nhập a ="); scanf("%d", &a);
    printf("\n Nhập b ="); scanf("%d", &b);
    printf("\n Ước số chung lớn nhất : ",USCLN(a,b));
    getch();}

```

**Ví dụ:** Viết hàm chuyển đổi kí tự in hoa thành kí tự in thường.

```

/* Ví dụ về hàm trả lại một kí tự*/
#include    <stdio.h>
#include    <conio.h>
/* khai báo nguyên mẫu cho hàm; */
char islower(char);
/* mô tả hàm */
char islower ( char c){
    if(c>='A' && c<='Z')
        c = c + 32;
    return(c);}
/* lời gọi hàm*/
void main(void){
    char c='A';
    printf("\n Kí tự được chuyển đổi : %c", islower(c));
    getch();}

```

**Ví dụ:** Viết hàm tính lũy thừa bậc n của số nguyên a.

```

/* Ví dụ về hàm trả lại một số nguyên dài*/
#include<stdio.h>
#include<conio.h>
/* khai báo nguyên mẫu cho hàm*/

```

```
long power(int , int );
/* mô tả hàm */
long power ( int a, int n )
{
    long s =1 ; int i;
    for(i=0; i<n;i++)
        s*=a;
return(s);
}
/* lời gọi hàm */
void main(void)
{
    int a = 5, i;
    for(i=0; i<50;i++)
        printf("\n %d mũ %d = %ld", a , i, power(a,i);
    getch();
}
```

**Ví dụ 4.20:** In ra số nhị phân của một số nguyên.

```
/* Ví dụ về hàm không trả lại giá trị*/
#include<stdio.h>
#include<conio.h>
/* khai báo nguyên mẫu cho hàm*/
void binary_int( int );
/* mô tả hàm */
void binary_int ( int a)
{
    int i, k=1; clrscr();
    for(i=15; i>=0; i--)
    {
        if ( a & (k<<i))
            printf("%3d", 1);
        else
            printf("printf("%3d", 0);
    }
}
/* lời gọi hàm */
```

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
Km10 Đường Nguyễn Trãi, Hà Đông-Hà Tây  
Tel: (04) 5541221; Fax: (04) 5540587  
Website: <http://www.c-ptit.edu.vn>; E-mail: [dhk@ptit.edu.vn](mailto:dhk@ptit.edu.vn)

GIƯỜNG TRÌNH PTIT  
ĐÀO TẠO ĐẠI HỌC TỪ XA

```

void main(void)
{
int a;

printf("\n Nhập a="); scanf("%d", &a);
printf("\n Số nhị phân của %d:", a);
binary_int(a);

getch();
}

```

#### 4.4.3. Phương pháp truyền tham biến cho hàm

Để thấy rõ được hai phương pháp truyền tham trị và truyền tham biến của hàm chúng ta khảo sát ví dụ sau:

**Ví dụ:** Cho hai số a, b hãy viết hàm đổi chỗ hai số a và b.

```

/* Phương pháp truyền tham trị */
#include<stdio.h>
void swap( float , float );
void swap ( float a, float b)
{
float temp;
temp = a;
a = b;
b = temp;
}
void main(void)
{
float a = 5, b = 7;
swap(a, b);
/* thực hiện đổi chỗ */
printf("\n Giá trị a = %6.2f, b = %6.2f", a, b);
}

```

#### **Kết quả thực hiện :**

Giá trị của a = 5, b = 7

Nhận xét: Hai biến a, b không được hoán vị cho nhau sau khi thực hiện hàm swap(a,b). Lý do duy nhất để dẫn đến sự kiện này là hàm swap(a,b) thực hiện trên bản sao giá trị của biến a và b. Phương pháp truyền giá trị của biến cho hàm được gọi là phương pháp truyền theo tham trị. Nếu muốn a, b thực sự hoán vị nội dung cho nhau chúng ta phải truyền cho hàm swap(a, b) địa chỉ của ô nhớ của a và địa chỉ ô nhớ của b khi đó các thao tác hoán đổi nội dung biến a và b được xử lý



trong hàm swap(a, b) thực chất là hoán đổi nội dung của ô nhớ dành cho a thành nội dung ô nhớ dành cho b và ngược lại.

Ví dụ sau sẽ minh họa cơ chế truyền tham biến cho hàm, trước khi chúng ta chưa thảo luận kỹ về con trỏ (pointer), ta tạm ngầm hiểu các qui định như sau:

Toán tử : &(tên\_biến) dùng để lấy địa chỉ của biến, chính xác hơn là địa chỉ ô nhớ dành cho biến.

Toán tử : \*(tên\_biến) dùng để lấy nội dung của ô nhớ dành cho biến.

**Ví dụ:** Cho hai số a, b hãy viết hàm đổi chỗ hai số a và b.

```
/* Phương pháp truyền tham trị */
```

```
#include <stdio.h>
void swap( float , float );
void swap ( float *a, float *b)
{
    float temp;
    temp = *a;
    *a = *b;
    *b = temp;
}
void main(void)
{
    float a = 5, b = 7;
    swap(&a, &b); /* thực hiện đổi chỗ địa trên chỉ của a và địa chỉ của b*/
    printf("\n Giá trị a = %6.2f    b = %6.2f", a, b);
}
```

**Kết quả thực hiện :**

Giá trị của a = 7 b = 5

Nhận xét: Giá trị của biến bị thay đổi sau khi hàm swap() thực hiện trên địa chỉ của hai biến a và b. Cơ chế truyền cho hàm theo địa chỉ của biến được gọi là phương pháp truyền tham biến cho hàm. Nếu hàm được truyền theo tham biến thì nội dung của biến sẽ bị thay đổi sau khi thực hiện hàm.

#### 4.4.4. Biến địa phương, biến toàn cục

##### a) Biến toàn cục

Biến toàn cục là biến được khai báo ở ngoài tất cả các hàm (kể cả hàm main()). Vùng bộ nhớ cấp phát cho biến toàn cục được xác định ngay từ khi kết nối (link) và không bị thay đổi trong suốt thời gian chương trình hoạt động. Cơ chế cấp phát bộ nhớ cho biến ngay từ khi kết nối còn được gọi là cơ chế cấp phát tĩnh.

Nội dung của biến toàn cục luôn bị thay đổi theo mỗi thao tác xử lý biến toàn cục trong chương trình con, do vậy khi sử dụng biến toàn cục ta phải quản lý chặt chẽ sự thay đổi nội dung của biến trong chương trình con.

Phạm vi hoạt động của biến toàn cục được tính từ vị trí khai báo nó cho tới cuối văn bản chương trình. Về nguyên tắc, biến toàn cục có thể khai báo ở bất kỳ vị trí nào trong chương trình,

nhưng nên khai báo tất cả các biến toàn cục lên đầu chương trình vì nó làm cho chương trình trở nên sáng sủa và dễ đọc, dễ nhìn, dễ quản lý.

**Ví dụ:** Ví dụ về biến toàn cục

```

/* Ví dụ về biến toàn cục*/
#include <stdio.h>
#include <conio.h>
/* khai báo nguyên mẫu cho hàm*/
void Tong_int( void );
/* khai báo biến toàn cục*/
int a = 5, b=7;
/* mô tả hàm */
int tong(void)
{   printf("\n Nhập a="); scanf("%d",&a);
    printf("\n Nhập b="); scanf("%d",&b);
    return(a+b);}
/* chương trình chính */
void main(void){
    printf("\n Giá trị a, b trước khi thực hiện hàm ");
    printf(" a =%5d b = %5d a + b =%5d", a, b, a + b);
    printf("\n Giá trị a, b sau khi thực hiện hàm ");
    printf(" a =%5d b = %5d a + b =%5d", a, b, a + b);
}

```

#### **Kết quả thực hiện:**

Giá trị a, b trước khi thực hiện hàm  $a=5$   $b=7$   $a+b=12$

Giá trị a, b sau khi thực hiện hàm

Nhập a = 10

Nhập b = 20

a = 10 b = 20 a + b = 30

#### **b) Biến địa phương**

Biến địa phương là các biến được khai báo trong các hàm và chỉ tồn tại trong thời gian hàm hoạt động. Tầm tác dụng của biến địa phương cũng chỉ hạn chế trong hàm mà nó được khai báo, không có mối liên hệ nào giữa biến toàn cục và biến địa phương mặc dù biến địa phương có cùng tên, cùng kiểu với biến toàn cục.

Cơ chế cấp phát không gian nhớ cho các biến địa phương được thực hiện một cách tự động,

khi nào khởi động hàm thì các biến địa phương được cấp phát bộ nhớ. Mỗi lần khởi động hàm là một lần cấp phát bộ nhớ, do vậy địa chỉ bộ nhớ dành cho các biến địa phương luôn luôn thay đổi sau mỗi lần gọi tới hàm.

Nội dung của các biến địa phương không được lưu trữ sau khi hàm thực hiện, các biến địa phương sinh ra sau mỗi lần gọi hàm và bị giải phóng ngay sau khi ra khỏi hàm. Các tham số dùng làm biến của hàm cũng là biến địa phương. Nghĩa là, biến của hàm cũng chỉ được khởi động khi gọi tới hàm.

Biến địa phương tĩnh (static): là biến địa phương đặc biệt được khai báo thêm bởi từ khoá static. Khi một biến địa phương được khai báo là static thì biến địa phương được cấp phát một vùng bộ nhớ cố định vì vậy nội dung của biến địa phương sẽ được lưu trữ lại lần sau và tồn tại ngay cả khi hàm đã kết thúc hoạt động. Mặc dù biến toàn cục và biến địa phương tồn tại trong suốt thời gian chương trình hoạt động nhưng điều khác nhau cơ bản giữa chúng là biến toàn cục có thể được truy nhập và sử dụng ở mọi lúc, mọi nơi, còn biến địa phương static chỉ có tầm hoạt động trong hàm mà nó được khai báo là static.

**Ví dụ:** Ví dụ về sử dụng biến địa phương static trong hàm

bien\_static() chứa biến tĩnh i và kiểm tra nội dung của i sau 5 lần gọi tới hàm.

```
#include<stdio.h>
/* nguyên mẫu của hàm */
void bien_static(void);
/* mô tả hàm */
void bien_static(void) {
    static int i;
    /* khai báo biến static */
    i++;
    printf("\n Lần gọi thứ %d", i);
}
void main(void){
    int n;
    for(n=1; n<=5; n++)
        bien_static();
}
```

**Kết quả thực hiện:**

Lần gọi thứ 1  
Lần gọi thứ 2  
Lần gọi thứ 3  
Lần gọi thứ 4  
Lần gọi thứ 5

Biến địa phương dạng thanh ghi (register) : Chúng ta đã biết rằng các bộ vi xử lý đều có các thanh ghi, các thanh ghi nằm ngay trong CPU và không có địa chỉ riêng biệt như các ô nhớ khác trong bộ nhớ chính nên tốc độ xử lý cực nhanh. Do vậy, để tận dụng ưu điểm về tốc độ của các thanh ghi chúng ta có thể khai báo một biến địa phương có kiểu register. Tuy nhiên, việc làm này cũng nên hạn chế vì số thanh ghi tự do không có nhiều. Nên sử dụng biến thanh ghi trong các trường hợp biến đó là biến đếm trong các vòng lặp.

**Ví dụ:** Biến địa phương có sử dụng register.

```
#include<stdio.h>

/* nguyên mẫu của hàm */
void bien_static(void);

/* mô tả hàm */
void bien_static(void) {
    static int i;

    /* khai báo biến static */
    i++;
    printf("\n Lần gọi thứ %d", i);
}

void main(void){
    register int n;

    for(n=1; n<=5; n++)
        bien_static();
}
```

#### Kết quả thực hiện

Lần gọi thứ 1

Lần gọi thứ 2

Lần gọi thứ 3

Lần gọi thứ 4

Lần gọi thứ 5

#### 4.4.5. Tính đệ qui của hàm

Một lời gọi hàm được gọi là đệ qui nếu nó gọi đến chính nó. Tính đệ qui của hàm cũng giống như phương pháp định nghĩa đệ qui của qui nạp toán học, hiểu rõ được tính đệ qui của hàm cho phép ta cài đặt rộng rãi lớp các hàm toán học được định nghĩa bằng đệ qui và giảm thiểu quá trình cài đặt chương trình.

**Ví dụ:** Nhận xét và cài đặt hàm tính n! của toán học

$$n! = \begin{cases} 1 & \text{khi } n=0; \\ (n-1)! * n & \text{khi } n \geq 1; \end{cases}$$

```
/* chương trình tính n! bằng phương pháp đệ qui */
#include<stdio.h>
#include<conio.h>
/* khai báo nguyên mẫu của hàm */
unsigned long GIAI_THUA( unsigned int );
/* mô tả hàm */
unsigned long GIAI_THUA(unsigned int n){
if (n == 0)
return(1);
    else return ( n * GIAI_THUA(n-1));
}
void main(void) {
unsigned int n;
printf("\ Nh\u00e0p n ="); scanf("%d", &n);
printf("\n n! = %ld", GIAI_THUA(n));
}
```

**Ghi chú:** Việc làm đệ qui của hàm cần sử dụng bộ nhớ theo kiểu xếp chồng LIFO (Last In, First Out để chứa các kết quả trung gian, do vậy việc xác định điểm kết thúc quá trình gọi đệ qui là hết sức quan trọng. Nếu không xác định rõ điểm kết thúc của quá trình chương trình sẽ bị treo vì lỗi tràn stack (stack overflow).

**Ví dụ:**

Viết đệ qui hàm tìm ước số chung lớn nhất của hai số nguyên dương a, b.

```
int USCLN( int a, int b){
    if( b == 0) return(a);
    else return( USCLN( y, x %y));
}
```

**Ví dụ:** Giải quyết bài toán kinh điển trong các tài liệu về ngôn ngữ lập trình "bài toán Tháp Hà Nội ". Bài toán được phát biểu như sau:

Có ba cột C1, C2, C3 dùng để xếp đĩa theo thứ tự đường kính giảm dần của các chiếc đĩa. Hãy tìm biện pháp dịch chuyển N chiếc đĩa từ cột này sang cột khác sao cho các điều kiện sau được thoả mãn:

- Mỗi lần chỉ được phép dịch chuyển một đĩa
- Mỗi đĩa có thể được dịch chuyển từ cột này sang một cột khác bất kỳ
- Không được phép để một đĩa trên một đĩa khác có đường kính nhỏ hơn

Ta nhận thấy, với  $N = 2$  chúng ta có cách làm như sau: Chuyển đĩa bé nhất sang C3, chuyển đĩa còn lại sang C2, chuyển đĩa 1 từ C2 sang C2.

Với  $N = 3$  ta lại xử lý lần lượt như sau với giả thiết đã biết cách làm với  $N = 2$  ( $N - 1$  đĩa): Chuyển đĩa 1, 2 sang C3 theo như cách làm với  $N=2$ ; chuyển đĩa 3 sang cột 2, chuyển đĩa 1 và 2 từ C3 sang C2.

**Chúng ta có thể tổng quát hoá phương pháp dịch chuyển bằng hàm sau:**

DICH\_CHUYEN(N\_đĩa, Từ\_Cột, Đến\_Cột, Cột\_Trung\_Gian);

Với  $N=2$  công việc có thể được diễn tả như sau:

DICH\_CHUYEN(1, C1, C3, C2);

DICH\_CHUYEN(1, C1, C2, C3);

DICH\_CHUYEN(1, C3, C2, C1);

Với  $N=3$  công việc dịch chuyển thực hiện như  $N = 2$  nhưng thực hiện dịch chuyển 2 đĩa

DICH\_CHUYEN(2, C1, C3, C2);

DICH\_CHUYEN(1, C1, C2, C3);

DICH\_CHUYEN(2, C3, C2, C1);

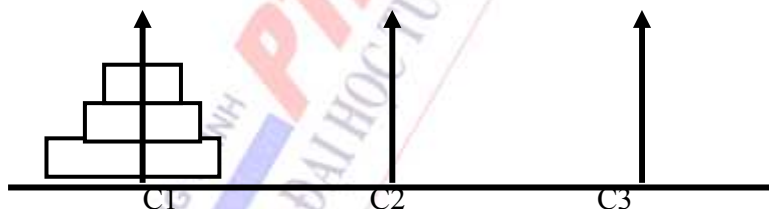
Với  $N$  tổng quát ta có :

DICH\_CHUYEN(N - 1, C1, C3, C2);

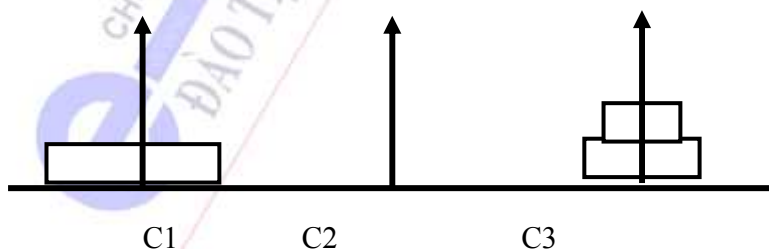
DICH\_CHUYEN(1, C1, C2, C3);

DICH\_CHUYEN(N - 1, C3, C2, C1);

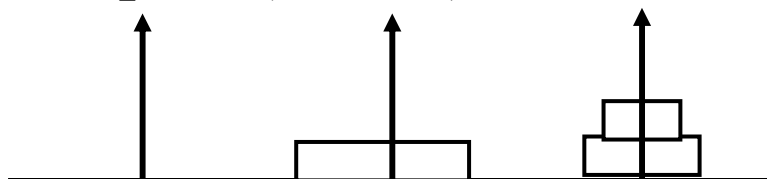
**Yêu cầu ban đầu:** dịch chuyển  $N$  đĩa từ cột C1 sang cột C2 thông qua cột trung gian C3:



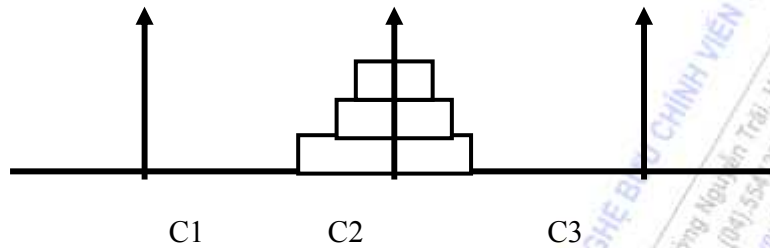
Thực hiện: DICH\_CHUYEN(N-1, C1, C3, C2);



Thực hiện: DICH\_CHUYEN(1, C1, C2, C3);



C1            C2            C3  
Thực hiện: DICH\_CHUYEN(N-1, C3, C2, C1);



Bài toán Tháp Hà Nội được thể hiện thông qua đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>
/* Khai báo nguyên mẫu cho hàm*/
void DICH_CHUYEN (int , int , int , int );
/* Mô tả hàm */
void DICH_CHUYEN (int N, int C1, int C2, int C3) {
    if ( N ==1 )

printf("\n %5d -> %5d", C1, C2);
else {
    DICH_CHUYEN ( N-1, C1, C3, C2);
    DICH_CHUYEN ( 1, C1, C2, C3);
    DICH_CHUYEN ( N-1, C3, C2, C1);
}
}
```

## 4.5. MỘT SỐ KIỂU DỮ LIỆU CÓ CẤU TRÚC

### 4.5.1. Cấu trúc dữ liệu kiểu Mảng (Array)

#### 4.5.1.1. Cấu trúc lưu trữ của mảng

##### a- Khái niệm về mảng

Mảng là một tập cố định các phần tử cùng có chung một kiểu dữ liệu với các thao tác tạo lập mảng, tìm kiếm, truy cập một phần tử của mảng, lưu trữ mảng. Ngoài giá trị, mỗi phần tử của mảng còn được đặc trưng bởi chỉ số của nó thể hiện thứ tự của phần tử đó trong mảng. Không có các thao tác bổ sung thêm phần tử hoặc loại bỏ phần tử của mảng vì số phần tử trong mảng là cố định.

Một mảng một chiều gồm n phần tử được coi như một vector n thành phần, phần tử thứ i của nó được tương ứng với một chỉ số thứ i - 1 đối với ngôn ngữ lập trình C vì phần tử đầu tiên được bắt đầu từ chỉ số 0. Chúng ta có thể mở rộng khái niệm của mảng một chiều thành khái niệm về mảng nhiều chiều.

Một mảng một chiều gồm n phần tử trong đó mỗi phần tử của nó lại là một mảng một chiều gồm m phần tử được gọi là một mảng hai chiều gồm n x m phần tử.

Tổng quát, một mảng gồm n phần tử mà mỗi phần tử của nó lại là một mảng k - 1 chiều thì nó được gọi là mảng k chiều. Số phần tử của mảng k chiều là tích số giữa số các phần tử của mỗi mảng một chiều.

Khai báo mảng một chiều được thực hiện theo qui tắc như sau:

Tên\_kiểu Tên\_biến[Số\_phần\_tử];

**Ví dụ :**

```
int A[10]; /* khai báo mảng gồm 10 phần tử nguyên*/
```

```
char str[20];
```

```
/* khai báo mảng gồm 20 kí tự */
```

```
float B[20];
```

```
/* khai báo mảng gồm 20 số thực */
```

```
long int L[20];
```

```
/* khai báo mảng gồm 20 số nguyên dài */
```

#### b- Cấu trúc lưu trữ của mảng một chiều

Cấu trúc lưu trữ của mảng: Mảng được tổ chức trong bộ nhớ như một vector, mỗi thành phần của vector được tương ứng với một ô nhớ có kích cỡ đúng bằng kích cỡ của kiểu phần tử và được lưu trữ kế tiếp nhau. Nếu chúng ta có khai báo mảng gồm n phần tử thì phần tử đầu tiên là phần tử thứ 0 và phần tử cuối cùng là phần tử thứ n - 1, đồng thời mảng được cấp phát một vùng không gian nhớ liên tục có số byte được tính theo công thức:

Kích\_cỡ\_mảng = ( Số\_phần\_tử \* sizeof(kiểu\_phần\_tử).

Ví dụ chúng ta có khai báo:

```
int A[10];
```

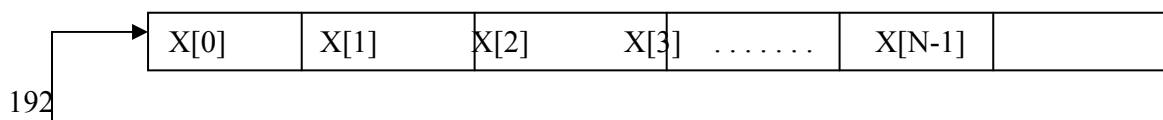
Khi đó kích cỡ tính theo byte của mảng là :

10 \* sizeof(int) = 20 byte;

float B[20]; => mảng được cấp phát: 20 \* sizeof(float) = 80byte;

Chương trình dịch của ngôn ngữ C luôn qui định tên của mảng đồng thời là địa chỉ phần tử đầu tiên của mảng trong bộ nhớ. Do vậy, nếu ta có một kiểu dữ liệu nào đó là Data\_type tên của mảng là X số phần tử của mảng là 10 thì mảng được tổ chức trong bộ nhớ như sau:

Data\_type X[N];





X - là địa chỉ đầu tiên của mảng.

$X = \&X[0] = (X + 0);$

$\&X[1] = (X + 1);$

.....

$\&X[i] = (X + i);$

**Ví dụ:** Tìm địa chỉ các phần tử của mảng gồm 10 phần tử nguyên.

```
#include<stdio.h>
#include<conio.h>
void main(void) {
    int A[10], i;
    /* khai báo mảng gồm 10 biến nguyên */
    printf("\n Địa chỉ đầu của mảng A là : %p", A);
    printf("\n Kích cỡ của mảng : %5d byte", 10 * sizeof(int));
    for ( i =0 ; i <10; i ++){
        printf("\n Địa chỉ phần tử thứ %5d : %p", i, &A[i]);
    }
    getch();
}
```

**Kết quả thực hiện chương trình:**

Địa chỉ đầu của mảng: FFE2

Kích cỡ của mảng: 20

Địa chỉ phần tử thứ 0 = FFE2

Địa chỉ phần tử thứ 1 = FFE4

Địa chỉ phần tử thứ 2 = FFE6

Địa chỉ phần tử thứ 3 = FFE8

Địa chỉ phần tử thứ 4 = FFEA

Địa chỉ phần tử thứ 5 = FFEC

Địa chỉ phần tử thứ 6 = FFEE

Địa chỉ phần tử thứ 7 = FFF0

Địa chỉ phần tử thứ 8 = FFF2

Địa chỉ phần tử thứ 9 = FFF

c- Cấu trúc lưu trữ mảng nhiều chiều

Ngôn ngữ C không hạn chế số chiều của mảng, chế độ cấp phát bộ nhớ cho mảng nhiều chiều được thực hiện theo cơ chế ưu tiên theo hàng.

Khai báo mảng nhiều chiều :

Data\_type tên\_biến[số\_chiều\_1] [số\_chiều\_2]. . . [số\_chiều\_n]

**Ví dụ:**

int A[3][3]; khai báo mảng hai chiều gồm 9 phần tử nguyên được lưu trữ liên tục từ A[0][0], A[0][1], A[0][2], A[1][0], A[1][1], A[1][2], A[2][0], A[2][1], A[2][2]

**Ví dụ:** Kiểm tra cấu trúc lưu trữ của bảng hai chiều trong bộ nhớ.

```
#include<stdio.h>
#include<conio.h>
void main(void) {
    float A[3][3];
    /* khai báo mảng hai chiều gồm 9 phần tử nguyên*/
    int i, j;
    /* Địa chỉ của các hàng*/
    for(i=0; i<3; i++)
        printf("\n Địa chỉ hàng thứ %d là :%p", i, A[i]);
    for(i=0; i<3; i++){
        printf("\n");
        for(j=0; j<3; j++)
            printf("%10p",&A[i][j]);
        }
    getch();
}
```

**Kết quả thực hiện chương trình:**

Địa chỉ hàng thứ 0 = FFD2

Địa chỉ hàng thứ 1 = FFDE

Địa chỉ hàng thứ 2 = FFEA

Địa chỉ phần tử A[0][0]= FFD2

Địa chỉ phần tử A[0][1]= FFD6

Địa chỉ phần tử A[0][2]= FFDA

Địa chỉ phần tử A[1][0]= FFDE

Địa chỉ phần tử A[1][1]= FFE2

Địa chỉ phần tử A[1][2]= FFE6

Địa chỉ phần tử A[2][0]= FFEA

Địa chỉ phần tử A[2][1]= FFEE

Địa chỉ phần tử A[2][2]= FFF2

**Ví dụ:** Kiểm tra cấu trúc lưu trữ của bảng ba chiều trong bộ nhớ.

```
#include<stdio.h>
#include<conio.h>
void main(void)
{
float      A[3][4][5] ;
/* khai báo mảng ba chiều */
    int i, j , k;
    for(i=0; i<3;i++){
        for(j=0;j<4;j++){
            printf("\n");
        }
        for( k=0; k <5; k ++)
            printf("%10p",&A[i][j]);
    }
    getch();
}
```

**Ghi chú:** Kết quả thực hiện ví dụ trên có thể cho ra kết quả khác nhau trên các máy tính khác nhau vì việc phân bổ bộ nhớ cho mảng tùy thuộc vào vùng bộ nhớ tự do của mỗi máy.

#### 4.5.1.2. Các thao tác đối với mảng

Các thao tác đối với mảng bao gồm: tạo lập mảng, tìm kiếm phần tử của mảng, lưu trữ mảng. Các thao tác này có thể được thực hiện ngay từ khi khai báo mảng. Chúng ta có thể vừa khai báo mảng vừa khởi đầu cho mảng, song cần chú ý một số kỹ thuật sau khi khởi đầu cho mảng.

**Ví dụ:**

```
int A[10] = { 5, 7, 2, 1, 9 };
```

Với cách khai báo và khởi đầu như trên, chương trình vẫn phải cấp phát cho mảng A kích cỡ  $10 * \text{sizeof}(\text{int}) = 20$  byte bộ nhớ, trong khi đó số byte cần thiết thực sự cho mảng chỉ là  $5 * \text{sizeof}(\text{int}) = 10$  byte. Để tránh lãng phí bộ nhớ chúng ta có thể vừa khai báo và khởi đầu dưới dạng sau.

```
int A[] = { 5, 7, 2, 1, 9 };
```

Trong ví dụ này, vùng bộ nhớ cấp phát cho mảng chỉ là số các số nguyên được khởi đầu trong dãy  $5 * \text{sizeof}(\text{int}) = 10$  byte.

Sau đây là một số ví dụ minh họa cho các thao tác xử lý mảng một và nhiều chiều.

**Ví dụ:** Tạo lập mảng các số thực gồm n phần tử, tìm phần tử lớn nhất và chỉ số của phần tử lớn nhất trong mảng.

```
#include<stdio.h>
#include<conio.h>
#define MAX    100
/*số phần tử tối đa trong mảng*/
void main(void) {
float A[MAX], max; int i, j, n;
/* Khởi tạo mảng số */
printf("\n Nhập số phần tử của mảng n="); scanf("%d", &n);
for(i=0; i<n; i++){
printf("\n Nhập A[%d] =",i); scanf("%f", &A[i]);
}
max = A[0]; j =0;
for(i=1; i<n; i++){
    if( A[i]>max) {
        max=A[i]; j = i;
    }
}
printf("\n Chỉ số của phần tử lớn nhất là : %d",j);
printf("\n Giá trị của phần tử lớn nhất là: %6.2f", max);
getch();
}
```

**Kết quả thực hiện chương trình:**

```
Nhập số phần tử của mảng n=7
Nhập A[0]=1
Nhập A[1]=9
Nhập A[2]=2
Nhập A[3]=8
Nhập A[4]=3
Nhập A[5]=7
Nhập A[6]=4
```

Chỉ số của phần tử lớn nhất là: 1

Giá trị của phần tử lớn nhất là: 9

**Ví dụ:** Tạo lập ma trận cấp  $m \times n$  và tìm phần tử lớn nhất, nhỏ nhất của ma trận.

```
#include<stdio.h>
#include<conio.h>
#define    M    20
#define    N    20
void main(void){
float  A[M][N], max, t; int i, j, k, p, m, n;
clrscr();
printf("\n Nhập số hàng của ma trận:"); scanf("%d", &m);
printf("\n Nhập số cột của ma trận:"); scanf("%d", &n);
for(i=0; i<m;i++){
for(j=0; j<n ; j++){
printf("\n Nhập A[%d][%d] =", i,j);
scanf("%f", &t); A[i][j]=t;
}
}
max=A[0][0]; k=0; p=0;
for(i=0; i<m; i++){
for(j=0;j<n; j++){
if(A[i][j]>max) {
max=A[i][j]; k=i ; p =j;
}
}
}
printf("\n Phần tử có giá trị max là A[%d][%d] = % 6.2f", k,p, max);
getch();
}
```

**Ghi chú:** C không hỗ trợ khuôn dạng nhập dữ liệu %f cho các mảng nhiều chiều. Do vậy, muốn nhập dữ liệu là số thực cho mảng nhiều chiều chúng ta phải nhập vào biến trung gian sau đó gán giá trị trở lại. Đây không phải là hạn chế của C++ mà hàm scanf() đã được thay thế bởi toán tử "cin". Tuy nhiên, khi sử dụng cin, cout chúng ta phải viết chương trình dưới dạng \*.cpp.

#### 4.5.1.3. Mảng và đối của hàm

Như chúng ta đã biết, khi hàm được truyền theo tham biến thì giá trị của biến có thể bị thay đổi sau mỗi lời gọi hàm. Hàm được gọi là truyền theo tham biến khi chúng ta truyền cho hàm là

địa chỉ của biến. Ngôn ngữ C qui định, tên của mảng đồng thời là địa chỉ của mảng trong bộ nhớ, do vậy nếu chúng ta truyền cho hàm là tên của một mảng thì hàm luôn thực hiện theo cơ chế truyền theo tham biến, trường hợp này giống như ta sử dụng từ khoá var trong khai báo biến của hàm trong Pascal. Trong trường hợp muốn truyền theo tham trị với đối số của hàm là một mảng, thì ta phải thực hiện trên một bản sao khác của mảng khi đó các thao tác đối với mảng thực chất đã được thực hiện trên một vùng nhớ khác dành cho bản sao của mảng.

**Ví dụ:** Tạo lập và sắp xếp dãy các số thực A1, A2, . . . An theo thứ tự tăng dần.

Để giải quyết bài toán, chúng ta thực hiện xây dựng chương trình thành 3 hàm riêng biệt, hàm Init\_Array() có nhiệm vụ tạo lập mảng số A[n], hàm Sort\_Array() thực hiện việc sắp xếp dãy các số được lưu trữ trong mảng, hàm In\_Array() in lại kết quả sau khi mảng đã được sắp xếp.

```
#include<stdio.h>
#include<conio.h>
#defineMAX    100
/* Khai báo nguyên mẫu cho hàm */
void Init_Array ( float A[], int n);
void Sort_Array( float A[], int n);
void In_Array( float A[], int n);
/* Mô tả hàm */
/* Hàm tạo lập mảng số */
void Init_Array( float  A[], int  n) {
    int  i;
    for( i = 0; i < n; i++ ) {
        printf("\n Nhập A[%d] = ", i);
        scanf("%f", &A[i]);
    }
}
/* Hàm sắp xếp mảng số */
void Sort_Array( float  A[], int  n ){
    int i , j ;
    float temp;
    for(i=0; i<n - 1 ; i ++ ) {
        for(j = i + 1; j < n ; j ++ ){
            if ( A[i]>A[j]) {
                temp = A[i];    A[i] = A[j]; A[j] = temp;
            }
        }
    }
}
```

```
    }
}
}
/* Hàm in mảng số */
void In_Array ( float A[], int n) {
    int i;
    for(i=0; i<n; i++)
        printf("\n Phần tử A[%d] = %6.2f", i, A[i]);
    getch();
}
/* Chương trình chính */
void main(void) {
    float A[MAX]; int n;
    printf("\n Nhập số phần tử của mảng n = "); scanf("%d", &n);
    Init_Array(A, n);
    Sort_Array(A,n);
    In_Array(A, n);
}
```

**Ví dụ:** Viết chương trình tính tổng của hai ma trận cùng cấp.

Chương trình được xây dựng thành 3 hàm, hàm `Init_Matrix()` : Tạo lập ma trận cấp  $m \times n$ ; hàm `Sum_Matrix()` tính tổng hai ma trận cùng cấp; hàm `Print_Matrix()` in ma trận kết quả. Tham biến được truyền vào cho hàm là tên ma trận, số hàng, số cột của ma trận.

```
#include <stdio.h>
#include <conio.h>
#include <dos.h> /* khai báo sử dụng hàm delay() trong chương trình*/
#define M 20 /* Số hàng của ma trận*/
#define N 20 /* Số cột của ma trận */
/* Khai báo nguyên mẫu cho hàm*/
void Init_Matrix(float A[M][N], int m, int n, char ten);
void Sum_Matrix(float A[M][N], float B[M][N], float C[M][N], int m, int n);
void Print_Matrix(float A[M][N], int m, int n);
/*Mô tả hàm */
void Init_Matrix(float A[M][N], int m, int n, char ten) {
    int i, j; float temp; clrscr();
    for(i=0; i<m; i++){
        for(j=0; j<n; j++){
            printf("\n Nhập %c[%d][%d] =", ten, i,j);
            scanf("%f", &temp); A[i][j]=temp;
        }
    }
}
```

```

    }
}
void Sum_Matrix(float A[M][N],float B[M][N], float C[M][N], int m,int n){
    int i, j;
    for(i=0; i<m; i++){
        for(j=0; j<n; j++){
            C[i][j]=A[i][j] + B[i][j];
        }
    }
}

void Print_Matrix(float A[M][N], int m, int n) {
int i, j , ch=179; /* 179 là mã kí tự '|' */
    for(i=0; i<m; i++){
        printf("\n %-3c", ch);
        for(j=0; j<n; j++){
            printf(" %6.2f", A[i][j]);
        }
        printf("%3c", ch);
    }
    getch();
}

/* Chương trình chính */
void main(void) {
    float A[M][N], B[M][N], C[M][N];
    int n, m; clrscr();
    printf("\n Nhập số hàng m ="); scanf("%d", &m);
    printf("\n Nhập số cột n ="); scanf("%d", &n);
    Init_Matrix(A, m, n, 'A');
    Init_Matrix(B, m, n, 'B');
    Sum_Matrix(A, B, C, m, n);
    Print_Matrix(C, m, n);
}

```

#### 4.5.2. Xâu kí tự (string)

Xâu kí tự là một mảng trong đó mỗi phần tử của nó là một kí tự, kí tự cuối cùng của xâu được dùng làm kí tự kết thúc xâu. Kí tự kết thúc xâu được ngôn ngữ C qui định là kí tự '\0', kí tự này có mã là 0 (NULL) trong bảng mã ASCII. Ví dụ trong khai báo :

```
char str[]='ABCDEF'
```

Khi đó xâu kí tự được tổ chức như sau:

```
0    1    2    3    4    5    6
```



A	B	C	D	E	F	'\0'
---	---	---	---	---	---	------

Khi đó `str[0] = 'A'`; `str[1] = 'B'`, . . ., `str[5]='F'`, `str[6]='\0'`;

Vì kí hiệu kết thúc chuỗi mã là 0 nên chúng ta có thể kiểm chứng tổ chức lưu trữ của chuỗi thông qua đoạn chương trình sau:

**Ví dụ:**

```

/* In ra từng kí tự trong chuỗi */
#include <stdio.h>
#include <string.h>
/* sử dụng hàm xử lý chuỗi kí tự gets() */
void main(void) {
    char str[20]; int i=0;
    printf("\n Nhập chuỗi kí tự:"); gets(str); /* nhập chuỗi kí tự từ bàn phím */
    while ( str[i]!='\0'){
        putchar(c); i++;
    }
}

```

*Ghi chú:* Hàm `getch()` nhận một kí tự từ bàn phím, hàm `putch(c)` đưa ra màn hình một kí tự c. Hàm `scanf("%s", str)` : nhận một chuỗi kí tự từ bàn phím nhưng không được chứa kí tự trống (space), hàm `gets(str)` : cho phép nhận từ bàn phím một chuỗi kí tự kể cả dấu trống.

Ngôn ngữ C không cung cấp các phép toán trên chuỗi kí tự, mà mọi thao tác trên chuỗi kí tự đều phải được thực hiện thông qua các lời gọi hàm. Sau đây là một số hàm xử lý chuỗi kí tự thông dụng được khai báo trong tệp `String.h`:

**puts (string)** : Đưa ra màn hình một string.

**gets(string)** : Nhận từ bàn phím một string.

**scanf("%s", string)** : Nhận từ bàn phím một string không kể kí tự trống (space).

**strlen(string)**: Hàm trả lại một số là độ dài của string.

**strcpy(s,p)** : Hàm copy chuỗi p vào chuỗi s.

**strcat(s,p)** : Hàm nối chuỗi p vào sau chuỗi s.

**strcmp(s,p)**: Hàm trả lại giá trị dương nếu chuỗi s lớn hơn chuỗi p, trả lại giá trị âm nếu chuỗi s nhỏ hơn chuỗi p, trả lại giá trị 0 nếu chuỗi s đúng bằng chuỗi p.

**strstr(s,p)** : Hàm trả lại vị trí của chuỗi p trong chuỗi s, nếu p không có mặt trong s hàm trả lại con trỏ NULL.

**strncmp(s,p,n)** : Hàm so sánh n kí tự đầu tiên của chuỗi s và p.

**strncpy(s,p,n)** : Hàm copy n kí tự đầu tiên từ chuỗi p vào chuỗi s.

**strrev(str)** : Hàm đảo chuỗi s theo thứ tự ngược lại.

Chúng ta có thể sử dụng trực tiếp các hàm xử lý chuỗi kí tự bằng việc khai báo chỉ thị `#include<string.h>`, tuy nhiên chúng ta có thể viết lại các thao tác đó thông qua ví dụ sau:

**Ví dụ:** Xây dựng các thao tác sau cho string:

- F1- Nhập xâu kí tự từ bàn phím hàm gets(str).
- F2- Tìm độ dài xâu kí tự strlen(str).
- F3- Tìm vị trí kí tự C đầu tiên xuất hiện trong xâu kí tự.
- F4- Đảo xâu kí tự.
- F5- Đổi xâu kí tự từ in thường thành in hoa.
- F6- Sắp xếp xâu kí tự theo thứ tự tăng dần. . .

```

/* Các thao tác với xâu kí tự */
#include<stdio.h>
#include<conio.h>
#include<dos.h>
#define F1 59
#define F2 60
#define F3 61
#define F4 62
#define F5 63
#define F6 64
#define F7 65
#define F10 68
#define MAX 256
/* khai báo nguyên mẫu cho hàm */
char *gets (char str[]); /* char * được hiểu là một xâu kí tự */
int strlen(char str[]); /* hàm trả lại độ dài xâu */
int strestr(char str[], char c); /* hàm trả lại vị trí kí tự c đầu tiên trong str*/
char *strrev(char str[]); /* hàm đảo xâu str*/
char *upper(char str[]); /* hàm đổi xâu str thành chữ in hoa*/
char *sort_str(char str[]); /* hàm sắp xếp xâu theo thứ tự từ điển*/
void thuc_hien(void);
/* Mô tả hàm */
/* Hàm trả lại một xâu kí tự được nhập từ bàn phím*/
char *gets( char str[] ) {
    int i=0; char c;
    while ( ( c=getch())!='\n') { /* nhập nếu không phải phím enter*/
        str[i] = c; i++;
    }
    str[i]='\0';
    return(str);
}
/* Hàm tính độ dài xâu kí tự: */
int strlen(char str[] ) {
    int i=0;
    while(str[i]) i++;
    return(i);
}

```

```
/* Hàm trả lại vị trí đầu tiên kí tự c trong chuỗi str */
int strchr (char str[] , char c) {
    int i=0;
    while (str[i] && str[i] != c )
        i++;
    if(str[i]!='\0' ) return(-1);
    return(i);
}
/* Hàm đảo chuỗi kí tự */
char *strrev( char str[]) {
    int i , j , n=strlen(str); char c;
    i = 0; j = n-1;
    while (i < j) {
        c = str[i] ; str[i] = str [j] ; str[j] =c;
    }
    return(str);
}
/* Hàm đổi chuỗi in thường thành in hoa */
char * upper( char str[] ) {
    int i, n=strlen(str);
    for(i=0; i<n; i++){
        if( str[i]>='a' && str[i]<='z')
            str[i]=str[i] - 32;
    }
    return(str);
}
/* Hàm sắp xếp chuỗi kí tự */
char *sort_str( char str[] ) {
    int i, j , n = strlen(str); char temp;
    for (i =0; i<n-1; i++){
        for(j=0; j<n; j++) {
            if(str[i] >str[j]){
                temp = str[i]; str[i] = str[j];
                str[j] = temp;
            }
        }
    }
}
/* Hàm thực hiện chức năng */
void thuc_hien(void) {
    char c , phim , str[MAX]; int control = 0;
    textmode(0) ;
    do {
```

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Km10 Đường Nguyễn Trãi, Hà Đông-Hà Tây  
Tel: (04) 5541221; Fax: (04) 5540587

Website: <http://www.c-ptit.edu.vn>; E-mail: [dhkc@ptit.edu.vn](mailto:dhkc@ptit.edu.vn)

CHƯƠNG TRÌNH PTIT  
ĐẠI HỌC TỰ XA

```

clrscr();
printf("\n Tập thao tác với string");
printf("\n F1- Tạo lập string");
printf("\n F2- Tính độ dài xâu");
printf("\n F3- Tìm kí tự trong string");
printf("\n F4- Đảo ngược string");
printf("\n F5- Đổi thành in hoa");
printf("\n F6- Sắp xếp string");
printf("\n F10- Trở về");
phim = getch();
switch(phim){
    case F1: gets(str); control=1; break;
    case F2:
        if (control)
            printf("\n Độ dài xâu là:%d", strlen(str));
        break;
    case F3:
        if (control){
            printf("\n Kí tự cần tìm:");
            scanf("%c", &c);
            if(strestr(str, c)>=0)
                printf("\n Vị trí:%d", strestr(str,c));
        }
        break;
    case F4:
        if (control)
            printf("\n Xâu đảo:%s", strrev(str));
        break;
    case F5:
        if (control)
            printf("\n In hoa:%s", upper(str));
        break;
    case F6:
        if (control)
            printf("\n Xâu được sắp xếp:%s", sort_str(str));
        break;
}
delay(2000);
} while(phim!=F10);
}
/* chương trình chính */
void main(void) {
    thuc_hien();
}

```

**Mảng các chuỗi:** mảng các chuỗi là một mảng mà mỗi phần tử của nó là một chuỗi.

Ví dụ `char buffer[25][80]` sẽ khai báo mảng các chuỗi gồm 25 hàng trong đó mỗi hàng gồm 80 kí tự. Ví dụ sau đây sẽ minh hoạ cho các thao tác trên mảng các string.

**Ví dụ:** Hãy tạo lập mảng các chuỗi trong đó mỗi chuỗi là một từ khoá của ngôn ngữ lập trình C. Sắp xếp mảng các từ khoá theo thứ tự từ điển.

Chương trình sau sẽ được thiết kế thành 3 hàm chính: Hàm `Init_KeyWord()` thiết lập bảng từ khoá, hàm `Sort_KeyWord()` sắp xếp mảng từ khoá, hàm `In_KeyWord()` in mảng các từ khoá.

Chương trình được thực hiện như sau:

```
/* Thao tác với mảng các string */
#include<stdio.h>
#include<conio.h>
#include<dos.h>
/* Khai báo nguyên mẫu cho hàm*/
void Init_KeyWord( char key_word[][20], int n);
void Sort_KeyWord(char key_word[][20], int n);
void In_KeyWord(char key_word[][20], int n);
/* Mô tả hàm */
void Init_KeyWord( char key_word[][20], int n) {
    int i;
    for( i = 0; i < n; i++){
        printf("\n Nhập từ khoá %d :",i);
        scanf("%s", key_word[i]);
    }
}
void Sort_KeyWord(char key_word[][20], int n) {
    int i, j, char temp[20];
    for( i = 0; i < n - 1; i++){
        for(j = i + 1; j < n; j++){
            if ( strcmp(key_word[i], key_word[j]) > 0){
                strcpy(temp, key_word[i] );
                strcpy(key_word[i], key_word[j] );
                strcpy(key_word[j], temp );
            }
        }
    }
}
void In_KeyWord(char key_word[][20], int n) {
    int i;
    for ( i = 0; i < n; i++){
        printf("\n Key_Word[%d] = %s", i, key_word[i]);
    }
    getch();
}
```

```
}  
void main(void) {  
    char key_word[100][20]; int n;  
    printf("\n Nhập số từ khoá n = "); scanf("%d", &n);  
    Init_KeyWord(key_word, n);  
    Sort_KeyWord(key_word, n);  
    In_KeyWord(key_word, n); }  
}
```



## Chương 5: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU ACCESS

### 5.1. TỔNG QUAN VỀ HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

#### 5.1.1. Cơ sở dữ liệu (CSDL)

Có thể hiểu CSDL (Database) là một bộ sưu tập rất lớn về các loại dữ liệu tác nghiệp của một cơ quan, xí nghiệp... được tổ chức lưu trữ tập trung hay phân tán trên các thiết bị nhớ của mạng máy tính và được các hệ ứng dụng truy xuất, khai thác theo chế độ trực tuyến hay tương tác. Cụ thể có một số khái niệm được hiểu như sau :

- CSDL là kho lưu trữ có tổ chức về các loại dữ liệu tác nghiệp, bao gồm các loại dữ liệu về văn bản, đồ họa, âm thanh... được mã hóa dưới dạng chuỗi bit 0 và 1 và được lưu trữ trong các bộ nhớ của máy tính. CSDL phản ánh trung thực thể giới dữ liệu hiện thực khách quan.

- CSDL dùng chung cho nhiều người. CSDL là tài nguyên thông tin chung cho nhiều người cùng sử dụng. Bất kỳ người sử dụng nào trên máy tính, tại các thiết bị đầu cuối, về nguyên tắc đều có quyền truy nhập khai thác toàn bộ hay một phần dữ liệu theo chế độ trực tuyến hay tương tác mà không phụ thuộc vào vị trí địa lý của người sử dụng với các tài nguyên đó.

- CSDL được các hệ ứng dụng khai thác, để tìm kiếm, tra cứu, sửa đổi bổ sung hay loại bỏ dữ liệu. Tìm kiếm và tra cứu thông tin là chức năng phổ biến nhất, quan trọng nhất của các hệ CSDL. Hệ quản trị CSDL (DBMS - DataBase Management System) là phần mềm điều khiển các chiến lược xây dựng CSDL. Khi người sử dụng đưa ra yêu cầu truy nhập bằng một ngôn ngữ dữ liệu nào đó thì DBMS tiếp nhận, thực hiện các thao tác trên CSDL lưu trữ và trả lại kết quả (nếu có).

- Hiện nay, CSDL quan hệ (Relationship Database) được sử dụng rộng rãi nhất. Đối tượng nghiên cứu của CSDL quan hệ là thực thể và mối quan hệ giữa các thực thể. Thực thể và mối quan hệ giữa các thực thể là hai đối tượng khác nhau về căn bản. Mối quan hệ giữa các thực thể cũng là một loại thực thể đặc biệt. Trong cách tiếp cận CSDL quan hệ, người ta xây dựng cơ sở lý thuyết đại số quan hệ xây dựng các quan hệ chuẩn khi kết nối không tổn thất thông tin và khi biểu diễn dữ liệu là duy nhất. Dữ liệu được lưu trữ trong bộ nhớ của máy tính không những phải tính đến yếu tố về tối ưu không gian lưu trữ, mà phải đảm bảo tính khách quan, trung thực dữ liệu hiện thực. Nghĩa là phải đảm bảo tính nhất quán của dữ liệu và dữ liệu được sự vẹn toàn của dữ liệu.

#### 5.1.2. Hệ quản trị CSDL:

**a. Giới thiệu:** Hệ quản trị CSDL là hệ thống phần mềm điều khiển toàn bộ các chiến lược truy nhập CSDL

Các chức năng cơ bản của hệ quản trị CSDL bao gồm

- Tạo ra và duy trì cấu trúc dữ liệu

- Cập nhật dữ liệu
- Lưu trữ dữ liệu
- Tìm kiếm và xử lý các dữ liệu lưu trữ
- Cho phép nhiều người dùng truy xuất đồng thời
- Hỗ trợ tính bảo mật và riêng tư
- Cung cấp một cơ chế chỉ mục (index) hiệu quả để lấy nhanh các dữ liệu lựa chọn.
- Bảo vệ dữ liệu khỏi mất mát bằng các quá trình sao lưu và phục hồi.

**b. Phân loại hệ quản trị CSDL:** Về cơ bản, hiện nay có 3 loại hệ quản trị CSDL chính là:

• **Hệ quản trị CSDL phân cấp chứa dữ liệu trong một cấu trúc như cây**

Hệ này thiết lập một quan hệ cha-con trong dữ liệu. Ở đỉnh của cây, còn gọi là root, có thể có một số nút bất kỳ. Các nút này lại có thể có một số nút con bất kỳ, v.v... Các hệ CSDL phân cấp ngày nay đã lạc hậu.

• **Hệ quản trị CSDL mạng chứa dữ liệu ở dạng các bản ghi và các liên kết**

Hệ thống này cho phép nhiều quan hệ nhiều-nhiều hơn hệ CSDL phân cấp. Một hệ quản trị CSDL mạng lưu trữ có hiệu quả và rất nhanh, cho phép quản lý các cấu trúc dữ liệu phức tạp. Tuy nhiên, chúng lại rất cứng nhắc và yêu cầu nặng nề về thiết kế. Một hệ thống đăng ký vé máy bay là một ví dụ của loại này.

• **Hệ quản trị CSDL quan hệ (Relationship DBMS)**

Hệ quản trị CSDL quan hệ có lẽ là một hệ quản trị CSDL có cấu trúc đơn giản nhất. Trong một hệ quản trị CSDL quan hệ, dữ liệu được tổ chức thành các bảng. Các bảng lại bao gồm các bản ghi và các bản ghi chứa các trường. Mỗi trường tương ứng với một mục dữ liệu. Hai hay nhiều bảng có thể liên kết nếu chúng có một hay nhiều trường chung.

Hệ quản trị CSDL quan hệ được sử dụng khá dễ dàng và phát triển nhiều trong những năm gần đây. Hiện nay có một số hệ quản trị CSDL quan hệ được sử dụng rộng rãi trên thị trường như Oracle, Informix, Sybase, Foxpro, Access,.....

Microsoft Access là một hệ quản trị CSDL đơn giản, dễ sử dụng, rất thích hợp cho các ứng dụng vừa và nhỏ. Tài liệu này sẽ trình bày về các khái niệm và cách sử dụng một cách căn bản hệ quản trị CSDL này.

## 5.2. HỆ QUẢN TRỊ CSDL ACCESS

### 5.2.1. Tổng quan về Access

Về cơ bản, Access là một hệ quản trị CSDL. Tương tự các sản phẩm khác thuộc loại này, Access lưu trữ và thu thập các thông tin (thường gọi là dữ liệu), hiển thị các thông tin yêu cầu và tự động hoá các thao tác được lặp lại (thí dụ duy trì các tài khoản hoặc một hệ thống thông tin cá nhân, thực hiện việc kiểm soát và lập bảng kiểm kê). Với Access, ta có thể tạo ra các biểu nhập dữ liệu để sử dụng. Ta có thể hiển thị các thông tin theo ý muốn và tạo các báo cáo phức tạp.

Access cũng là một ứng dụng CSDL mạnh của Windows, nó là một sản phẩm được thiết kế



để hỗ trợ người phát triển hoặc người dùng cuối. Bên cạnh tính tiện dụng và nhất quán của Microsoft Window, Microsoft Access cũng bổ sung thêm tính hiệu quả trong quản lý CSDL. Do cả Access và Window đều là những sản phẩm của Microsoft nên cả hai phối hợp với nhau hoạt động rất tốt. Access có khả năng chạy trên nền Windows95, Windows 98, Windows ME, Windows 2000..., do đó Access có mọi lợi thế của Windows. Ta có thể cắt dán, sao chép các dữ liệu từ bất kỳ một ứng dụng nào của Windows vào Access hoặc ngược lại. Ta cũng có thể tạo ra thiết kế biểu mẫu trong Access và cắt dán vào thiết kế báo cáo.

Khi sử dụng các đối tượng *OLE* (Object Linking and Embedding - Liên kết và nhúng đối tượng) trong các sản phẩm của Windows và Microsoft Office (như Excel, Word, PowerPoint và Outlook), ta có thể mở rộng khả năng của Access để hợp nhất các đối tượng có thể xem được của các sản phẩm Microsoft này mà không cần phải sao chép nội dung của chúng và dán vào Access. Khả năng này cho phép ta có thể làm việc với các dữ liệu thực tế có sẵn trong các sản phẩm này mà không cần phải sao chép thông tin. Bằng cách sử dụng *OLE*, ta có thể thực sự thay đổi các thông tin trong đối tượng biểu cơ sở (như Excel, Word, v.v...). Với các mở rộng của Internet trong Access, ta có thể lập các biểu tương tác với các dữ liệu lấy trực tiếp từ các trang Web WWW và có thể trực tiếp chuyển các biểu ta lập thành các trang truy cập dữ liệu cho các mạng nội bộ hợp nhất làm việc trực tiếp với trình duyệt Internet của ta.

Là một hệ quản trị dữ liệu quan hệ, Access cho phép người sử dụng cùng lúc có thể truy nhập các thông tin từ hai bảng trở lên - thậm chí có thể liên kết các bảng CSDL để tạo lập bảng mới. Access có thể làm giảm tính phức tạp của các dữ liệu và làm cho công việc dễ thực hiện hơn. Có thể nối các bảng Access của ta với các dữ liệu của máy tính lớn hoặc máy chủ - thậm chí có thể sử dụng một bảng được tạo trong các chương trình dBase hoặc Excel. Có thể dễ dàng kết hợp các kết quả kết nối với một bảng tính trong Excel.

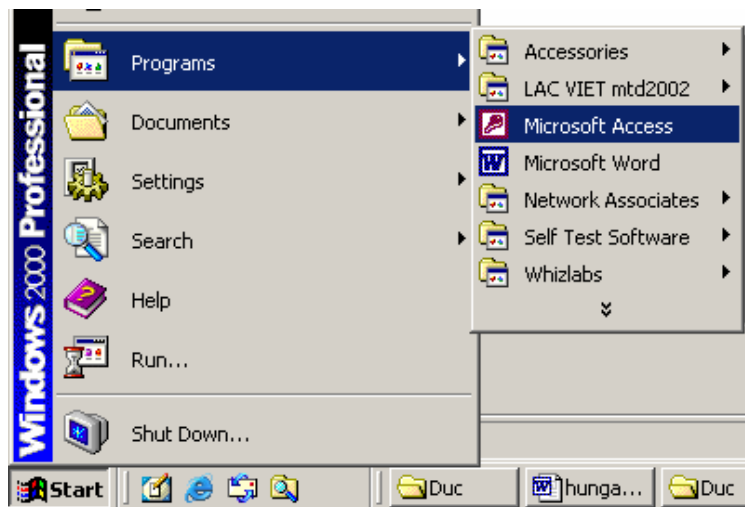
Khả năng kết nối của Access với các dữ liệu bên ngoài (lưu giữ trong các dạng CSDL khác) làm cho Access trở thành một chương trình rất thiết thực. Có thể sử dụng Access trong một môi trường mạng để liên kết với một loạt các bảng từ các CSDL khác - vừa ở dạng liên kết nội bộ (trên cùng một máy) vừa liên kết xa (thậm chí với CSDL của một máy tính lớn như Oracle hoặc DB2). Access có thể kết nối trực tiếp với các bảng bên ngoài này hoặc có thể nhập về sử dụng nội bộ. Khi đã kết nối hoặc nhập các bảng bên ngoài về, ta có thể tạo lập các biểu và các báo cáo làm việc với các thông tin - để thay đổi hoặc xem hoặc in các thông tin

## **5.2.2. Khởi động, cửa sổ làm việc của Microsoft Access**

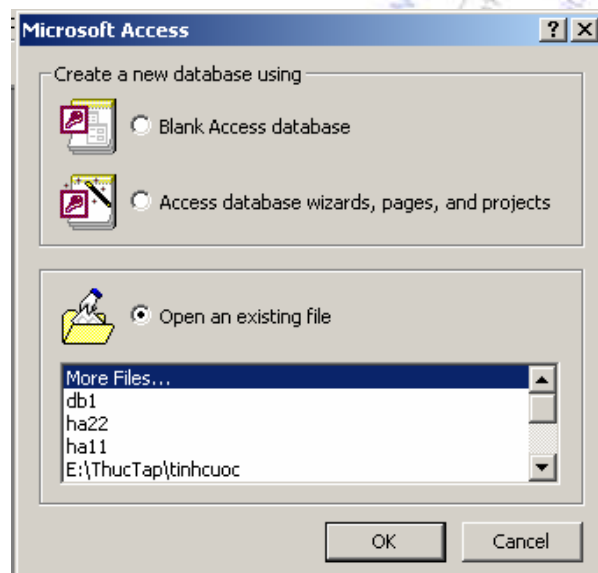
### **5.2.2.1. Khởi động Microsoft Access**

Microsoft Access là hệ quản trị CSDL được cài đặt trên môi trường Windows nên ta có thể cài đặt và khởi động Access tương tự như khởi động các chương trình ứng dụng khác như Microsoft Word, Excel... Có thể khởi động theo các bước như sau:

Trên thanh Start mở Program, sau đó nhấn vào biểu tượng Microsoft Access



Khi đó hộp thoại đầu tiên xuất hiện trên màn hình:



Hộp thoại này cho phép chúng ta chọn các hành động sau:


- **Blank Access Database** : Tạo mới hoàn toàn một tập tin CSDL Access trống (tập tin đó phân mở rộng là MDB - chưa có dữ liệu)
- **Access Database Wizards, pages, and projects**: Tạo mới một tập tin CSDL Access sử dụng đồ thuật, hoặc tạo các trang dữ liệu hay các đề án.
- **Open an Existing File**: Mở một tập tin CSDL Access đã được tạo trước đó

#### 5.2.2.2. Cửa sổ làm việc của Microsoft Access

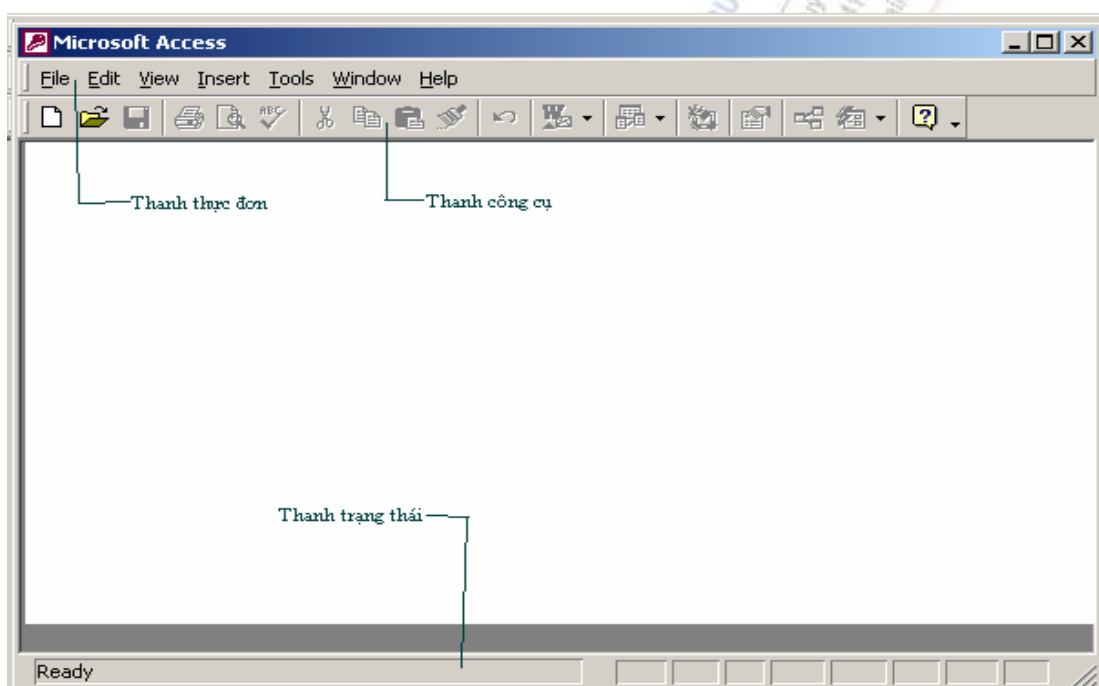
Sau khi khởi động Microsoft Access, cửa sổ của Microsoft Access xuất hiện. Các thành phần trong cửa sổ này gồm:

- a. *Thanh thực đơn (Menu bar)*: chứa các chức năng trong từng thực đơn và các chức năng này sẽ được thay đổi tùy thuộc vào đối tượng làm việc hiện hành của Microsoft Access

b. *Thanh công cụ (Tool bar)*: Chứa các biểu tượng đại diện cho một hành động nào đó mà khi người sử dụng nhấn vào thì nó sẽ thực hiện. Các thanh công cụ này cũng sẽ được thay đổi tùy ý vào đối tượng làm việc hiện hành của Microsoft Access. Mỗi một thanh công cụ đều có một tên riêng để phân biệt. Thông thường chúng ta nên nhớ ý nghĩa của các biểu tượng trên các thanh công cụ để có thể thực hiện nhanh một hành động nào đó thay vì phải vào từng cấp của thực đơn.

Ví dụ: Để mở một tập tin CSDL chúng ta phải vào File rồi chọn Open, nhanh hơn chúng ta có thể nhấn vào biểu tượng  trên thanh công cụ

c. *Thanh trạng thái (Status bar)*: Thể hiện tình trạng hiện hành của các phím đặc biệt (CapsLock, Numlock...), các thông báo, hướng dẫn sử dụng Microsoft Access.



Nói chung các thao tác với thanh công cụ giống với các thanh công cụ trong Winword

### 5.2.2.3. Các thao tác đối với tập tin CSDL

*Tạo một tệp mới*: Khi khởi động Access tại cửa sổ khởi động thì cũng có thể tạo nên một tệp CSDL mới bằng cách chọn *Blank Database* từ cửa sổ khởi động của Access.

Mặt khác tại cửa sổ Database có thể làm như sau: File -> New sau đó chọn Database, OK. Tiếp tục phải đưa tên tệp vào rồi chọn Create.

*Mở một tệp CSDL của Acces đã tồn tại trong máy tính*: Chúng ta cũng có thể mở một tệp CSDL đã tồn tại trong Access ngay từ màn hình khởi động bằng cách chọn *Open an existing file (database)*. Mặt khác tại cửa sổ Database có thể làm như sau: File -> Open, chọn Tên tệp tin, OK

*Đóng tệp CSDL*: Chọn File -> Close hoặc đóng cửa sổ

*Thoát khỏi Access*: Chọn File -> Exit.

#### 5.2.2.4. Các đối tượng bên trong tập tin CSDL Access:

Một tập CSDL gồm có 7 đối tượng cơ bản sau:

**Bảng (Table):** Là thành phần cơ sở của tập tin CSDL của Access, dùng để lưu trữ dữ liệu. Hệ thống bảng chính là cấu trúc CSDL. Do đó đây là dữ liệu đầu tiên phải được tạo ra trước. Bên trong một bảng, dữ liệu được lưu thành nhiều cột và nhiều hàng

**Truy vấn (Query):** Là công cụ cho phép người sử dụng dùng ngôn ngữ truy vấn có cấu trúc SQL (Structure Query Language) hoặc công cụ truy vấn bằng ví dụ QBE (Query by Example) để thực hiện các truy vấn rút trích, chọn lựa dữ liệu hoặc cập nhật dữ liệu (thêm, sửa, xóa) trên các bảng. Truy vấn bằng thí dụ là công cụ hỗ trợ việc thực hiện các truy vấn mà không cần phải viết lệnh SQL, mà chủ yếu chỉ dùng kỹ thuật kéo - thả trên cơ sở đồ họa. Trong tài liệu này chúng tôi chủ yếu đi vào truy vấn QBE.

**Biểu mẫu (Form):** Cho phép người sử dụng xây dựng nên các màn hình dùng để cập nhật hoặc xem dữ liệu lưu trong các bảng, ngoài ra cũng cho phép người sử dụng tạo ra các hộp thoại hỏi đáp giữa người sử dụng và hệ thống ứng dụng.

**Báo cáo (Report):** Cho phép chúng ta tạo ra kết xuất từ các dữ liệu đã lưu trong các bảng, sau đó sắp xếp lại và định dạng theo một khuôn dạng cho trước và từ đó có thể đưa kết xuất này ra màn hình hoặc máy in hoặc các dạng tập tin Word/ Excel.

**Tập lệnh (Macro):** Là công cụ cung cấp cho người sử dụng tạo ra các hành động đơn giản trong Microsoft Access như mở biểu mẫu, báo cáo, thực hiện một truy vấn... mà không cần phải biết gì về ngôn ngữ lập trình Visual Basic.

**Bộ mã lệnh (Module):** Với ngôn ngữ Visual Basic cho phép người sử dụng xây dựng các hàm hoặc thủ tục của riêng mình để thực hiện một hành động phức tạp nào đó mà không thể làm bằng công cụ tập lệnh, hoặc với mục đích cho chương trình chạy nhanh hơn.

**Các trang Web dữ liệu (Pages):** Cho phép chúng ta có thể tạo ra các trang Web dữ liệu mà trên đó có chứa dữ liệu động lấy từ một CSDL nào đó. Người sử dụng có thể cập nhật dữ liệu trực tiếp vào cơ sở dữ liệu thông qua trình duyệt Web (Microsoft Internet Explorer) .

#### 5.2.2.5. Làm việc với cửa sổ dữ liệu trong Microsoft Access:

Cửa sổ CSDL (cửa sổ Database) là cửa sổ làm việc chính của Microsoft Access, trên cửa sổ này cho phép người sử dụng tạo mới, hoặc sửa đổi hoặc xóa các đối tượng cơ bản trong tập tin CSDL Access bằng cách chọn các dãy nút dọc và ngang

##### a. Các thành phần chính trong cửa sổ CSDL của Microsoft Access

*Các nút đối tượng:* Bảng, Query, Form, Report, Macro. Modul... cho phép chúng ta lựa chọn một loại đối tượng làm việc hiện hành trong cửa sổ CSDL.

*Các nút hành động:* Open, Design, New: cho phép chúng ta mở, sửa và tạo mới các đối tượng làm việc hiện hành (là đối tượng đang được chọn trong danh sách các đối tượng)

##### b. Đóng một đối tượng đang mở

Chọn File -> Close hoặc nhấp vào nút Close trên cửa sổ chứa đối tượng.

### c. Xóa một đối tượng

Chọn đối tượng cần xóa, chọn Edit rồi ấn Delete, rồi xác nhận lại đồng ý xóa thật không (chọn YES).

### d. Đổi tên một đối tượng

Chọn đối tượng cần đổi tên, chọn Edit rồi chọn Rename, sau đó ghi vào tên mới (Bấm ESC để trả lại tên cũ trước đó)

### e. Ẩn hoặc hiện cửa sổ CSDL

Ẩn: Chọn Window -> Hide

Hiện: Chọn Window -> Unhide (hoặc nhấn phím F11)

## 5.2.3. Làm việc với các đối tượng trong Access

Trước khi bắt tay vào xây dựng các bảng, biểu mẫu, các báo biểu và các thành phần khác của một CSDL, ta cần phải bỏ thời gian thiết kế chi tiết. Một thiết kế tốt là chìa khoá cho sự thành công của mọi CSDL, nó cho phép người dùng thao tác một cách dễ dàng, chương trình chạy một cách hiệu quả và chính xác.

### 5.2.3.1. Quy trình thiết kế CSDL

Các bước cơ bản để thiết kế một CSDL là:

1. *Xác định mục đích của CSDL*: CSDL dùng để lưu trữ thông tin gì, cần rút ra những thông tin gì từ CSDL.

2. *Xác định các bảng dữ liệu, table, cần có trong CSDL*: một bảng không được chứa dữ liệu trùng lặp, và thông tin cũng không nên để bị trùng lặp giữa các bảng. Mỗi một bảng chỉ nên chứa thông tin liên quan đến một chủ đề.

3. *Xác định các trường dữ liệu, field, cần có trong mỗi bảng dữ liệu*: Mỗi bảng chứa thông tin về cùng một chủ thể, mỗi trường của một bảng lại chứa các thông tin riêng biệt về chủ thể đó. Ta chỉ nên ghi nhận các thông tin cần thiết từ thực tế, không nên có các trường tính toán - calculated field. Lưu trữ thông tin dưới dạng chi tiết nhất có thể (ví dụ: Họ, Tên chứ không nên Họ và Tên).

4. *Xác định các trường chứa giá trị duy nhất ở mỗi bản ghi*: Để Access kết nối được các thông tin lưu trữ trong các bảng riêng biệt. Ví dụ: kết nối các khách hàng với đơn đặt hàng của người đó. Mỗi bảng cần phải có một trường hoặc một nhóm các trường chứa giá trị duy nhất dùng để phân biệt giữa các bản ghi trong một bảng. Một trường như vậy gọi là trường khoá - Primary key.

5. *Xác định các mối quan hệ giữa các bảng*: Ta chỉ ra cho Access cách để tìm kiếm và đưa các thông tin liên quan lại tập hợp với nhau.

6. *Tinh chỉnh thiết kế*: Sau khi đã thiết kế, ta cần vận hành thử nghiệm CSDL để đảm bảo các chức năng cần có đều thực hiện suôn sẻ đúng như mong đợi. Nếu phát hiện những thông tin gì không cần thiết thì ta nên loại bỏ để tránh sự công kềnh.

7. *Nhập dữ liệu và tạo các đối tượng CSDL*: Khi đã hài lòng với cấu trúc các bảng biểu và

các mối liên kết là lúc phải nhập liệu. Dựa trên dữ liệu đã nhập ta có thể dễ dàng xây dựng các biểu mẫu, báo biểu, truy vấn để phục vụ công tác xử lý đầu vào và đầu ra thông tin.

8. *Sử dụng công cụ phân tích của Access*: Access cung cấp công cụ Bảng Analyzer Wizard giúp người dùng phân tích thiết kế đã có và chỉnh sửa chúng.

**5.2.3.2. Tạo và quản lý các bảng dữ liệu (Table)**

Nói chung, một bảng dữ liệu là tập hợp các dữ liệu về một chủ đề nhất định (ví dụ sản phẩm, người bán). Sử dụng mỗi bảng riêng cho mỗi đối tượng cho phép ta lưu dữ liệu một cách rõ ràng, hiệu quả, tránh lưu nhiều lần một thông tin, giảm thiểu lỗi nhập liệu.

Bảng tổ chức dữ liệu theo cột (gọi là trường - field), và theo hàng (gọi là bản ghi-record).

Ta dùng một trường chung để liên kết hai bảng. Việc này cho phép Access kết xuất hiển thị dữ liệu từ các bản ghi ở các bảng dữ liệu khác nhau có liên quan để xem, sửa, hoặc in ra báo cáo

Bảng: Sản phẩm		
Tên Sản Phẩm	Mã nhà cung cấp	Số lượng trong kho
Giấy Vĩnh Phúc	1	20
Giấy Bãi bằng	2	50
Giấy Indo	3	20

Mỗi trường trong bảng sản phẩm chứa cùng một kiểu thông tin cho tất cả các sản phẩm, như số lượng.

Mỗi một bản ghi chứa tất cả thông tin liên quan đến một sản phẩm, như là tên sản phẩm, mã nhà cung cấp, số lượng trong kho ...

Trường dữ liệu chung liên kết dữ liệu từ hai bảng

Mã nhà cung cấp	Tên nhà cung cấp
1	Nhà máy giấy Vĩnh
2	Nhà máy giấy Bãi Bằng
3	Công ty xuất nhập khẩu hải phòng

Bảng: Sản phẩm		
Tên Sản Phẩm	Nhà cung cấp	Số liệu trong kho
Giấy Vĩnh Phúc	1	20
Giấy Bãi Bằng	2	50
Giấy Indo	3	20

**a. Thiết kế một bảng dữ liệu**

**Cách 1:** Tạo bảng dữ liệu bằng cách nhập dữ liệu

1. Chọn cửa sổ Database .
2. Nhấn Tables bên dưới Objects, rồi chọn New trong cửa sổ Database.
3. Nhấn đúp Datasheet View. Một lưới dữ liệu trống được hiển thị với các tên ngầm định

của các cột là Field1, Field2,...

4. Đổi tên cột bằng cách nhấp đúp chuột vào tiêu đề các cột Field1, Field2..., rồi nhập vào tên mới..

5. Nhập dữ liệu vào lưới nhập liệu, nhớ rằng mỗi cột chỉ được nhập một kiểu dữ liệu duy nhất từ dòng thứ nhất đến dòng cuối cùng. Tức là, nếu dòng đầu dữ liệu kiểu số, kiểu ngày,... thì các dòng tiếp theo cột đó dữ liệu cũng phải là kiểu số, kiểu ngày. Access sẽ tạo kiểu dữ liệu (data type) và dạng hiển thị (display format) tương ứng cho mỗi cột. Các cột bỏ trống sẽ bị xoá khi ta ghi lại lưới nhập liệu (DataSheet).

6. Chọn Save trên thanh công cụ để ghi lại mọi thay đổi. Access sẽ hỏi nếu bạn có muốn tạo một mã khoá chính cho bảng vừa tạo (Primary key).

### Cách 2: Tự thiết kế bảng dữ liệu

#### Trình tự thiết kế

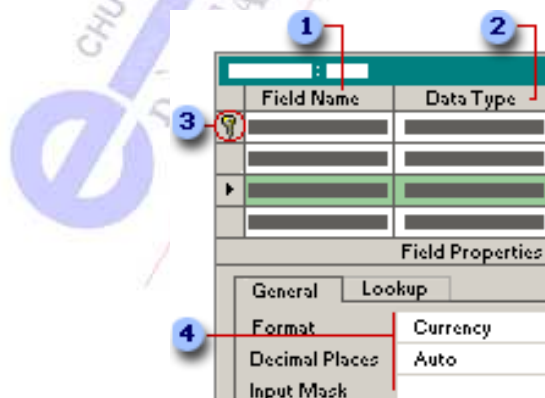
1. Chuyển sang cửa sổ CSDL. Ta có thể dùng phím F11 để chuyển đổi giữa các cửa sổ.
2. Nhấn Tables bên dưới Objects, rồi New trong cửa sổ Database.
3. Nhấn đúp chuột vào Design View để chọn chế độ thiết kế.
4. Định nghĩa các trường của bảng.
5. Thiết lập trường khóa – Primary key – trước khi ghi lại thiết kế

Lưu ý: Không nhất thiết phải xác định trường khóa, nhưng nên tạo trường khóa, vì căn cứ vào các trường khóa này, về sau khi ta thiết kế các truy vấn, biểu mẫu, báo biểu, Access sẽ tự động xác định các liên kết cần thiết để kết nối dữ liệu.

6. Nhấn chuột vào nút lệnh Save  trên thanh công cụ để ghi lại mọi thay đổi, nhập vào tên của bảng.

#### Cửa sổ thiết kế

Trong chế độ thiết kế, ta có thể thiết kế một bảng từ đầu, hoặc thêm, xoá, hoặc sửa đổi theo sở thích các trường của một bảng đã có. Trong chế độ này, cửa sổ giao diện có các khu vực chính dưới đây:



*Field Name* - Tên trường: Nếu ta muốn theo dõi thêm các thông tin, hãy bổ sung thêm các trường mới. Nếu tên của một trường đã có chưa đủ biểu đạt ý nghĩa, ta có thể đổi tên trường.

*Data types* - Kiểu dữ liệu: Việc thiết lập kiểu dữ liệu của một trường để định nghĩa kiểu dữ liệu ta sẽ nhập vào trường đó. (Kiểu dữ liệu: tính chất của một trường mà xác định kiểu dữ liệu nào sẽ được lưu trữ trong trường. Ví dụ, trường có thuộc tính Data type là Text có thể lưu dữ liệu dạng chữ hoặc số, nhưng một trường Number chỉ có thể chứa dữ liệu số).

*Primary Key* - Khoá chính: Sử dụng một mã số duy nhất, gọi là khoá chính để phân biệt mỗi một bản ghi trong bảng. Một khoá chính của bảng dùng để tham chiếu đến các bản ghi liên kết trong các bảng khác. Khóa chính có thể là một hoặc nhiều trường (cột) mà duy nhất giá trị của nó có thể dùng để phân biệt các bản ghi của một bảng. Một khoá chính không thể chứa giá trị rỗng - Null và phải là chỉ mục sắp xếp duy nhất.

*Field properties* - Thuộc tính của trường: Các thuộc tính của trường là tập hợp các

**Các kiểu dữ liệu của một trường:**

Sử dụng thuộc tính kiểu dữ liệu để chỉ định kiểu dữ liệu sẽ được lưu trong trường. Mỗi trường chỉ có thể chứa thông tin của một kiểu dữ liệu.

Kiểu dữ liệu	Mô tả	Kích thước
Text	(Ngầm định) Ký tự hoặc tổ hợp giữa ký tự và chữ số, hoặc cũng có thể chứa số không cần tính toán như số điện thoại.	Tối đa 255 ký tự, quy định bởi thuộc tính FieldSize (kích thước trường).
Memo	Những đoạn văn bản dài	Tối đa 65,535 ký tự
Number	Dữ liệu số dùng trong các phép tính toán số học.	1, 2, 4, hoặc 8 bytes (16 bytes nếu chọn kiểu số là Replication ID).
Date/Time	Các giá trị ngày tháng hoặc thời gian tính từ năm 100 đến năm 9999.	8 bytes.
Currency	Kiểu tiền tệ. Độ chính xác đến 15 số về bên trái dấu phẩy và 4 chữ số về bên phải	8 bytes.
AutoNumber	Số theo thứ tự tăng dần, mỗi lần 1 đơn vị) hoặc số ngẫu nhiên do Access sinh ra khi thêm một bản ghi mới để phân biệt giữa các bản ghi trong một bảng. Trường AutoNumber không thể bị sửa đổi..	4 bytes (16 bytes nếu thuộc tính FieldSize được thiết lập là Replication ID).
Yes/No	Trường chỉ chứa một trong hai giá trị (Yes/No, True/False, hoặc On/Off).	1 bit.
OLE Object	Một đối tượng (như trang tính Excel, văn bản Word, hình hoạ, âm thanh) được liên kết hoặc nhúng vào bảng.	Tối đa 1 gigabyte (giới hạn bởi khoảng còn trống trên ổ đĩa)




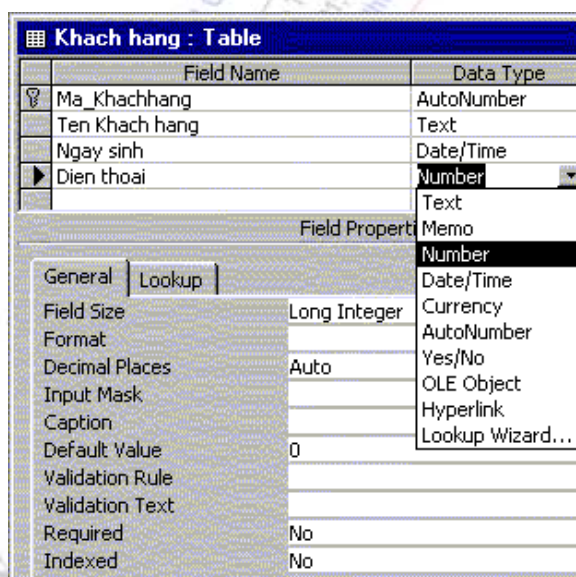
Hyperlink	Liên kết	Tối đa 2048 ký tự
Lookup Wizard	Tạo ra một trường mà cho phép ta nhập liệu bằng cách chọn giá trị từ một bảng khác, hoặc từ một danh sách các giá trị.	Cùng kích thước với trường khoá dùng để thực hiện thao tác tìm kiếm.

Ta chỉ có thể thiết lập các thuộc tính này ở phần trên của sổ thiết kế bảng dữ liệu (bảng Design view).


### b. Chỉnh sửa bảng

#### Thêm một trường vào bảng dữ liệu trong chế độ thiết kế Design view:

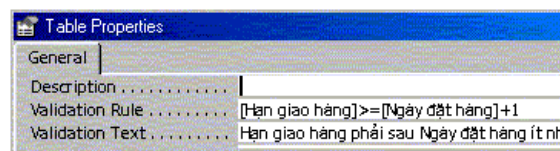
1. Mở bảng ở chế độ Design view.
2. Nhấn dòng muốn chèn thêm cột, nhấn Insert Rows  trên thanh công cụ.
3. Trong cột Field Name nhập tên cho trường.
4. Trong cột Data Type, nhấn vào mũi tên, chọn kiểu dữ liệu từ danh sách
5. Thiết lập các thuộc tính cho trường.
6. Ghi lại mọi sự thay đổi khi đóng thoát.



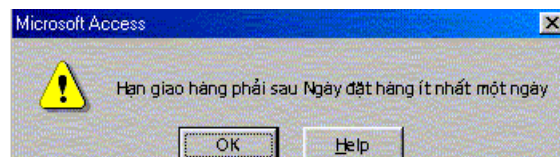
#### Quy định tính hợp lệ của dữ liệu trước khi lưu

1. Mở bảng ở chế độ Design view.
2. Nhấn chuột vào nút  trên thanh công cụ để hiển thị trang thuộc tính của bảng. Trên dòng Validation Rule, nhập vào quy tắc xác định tính hợp lệ của dữ liệu.
3. Trên dòng ValidationText, đánh vào lời thông báo khi dữ liệu nhập vào vi phạm quy tắc dữ liệu. Ví dụ, “Hàng chỉ có thể được giao sau 01 ngày tính từ ngày đặt hàng”

Ví dụ: Doanh nghiệp xác định là hàng chỉ có thể được giao đến khách sớm nhất là sau một ngày. Vậy quy tắc xác định tính hợp lệ của dữ liệu là: [Hạn giao hàng] >= [Ngày đặt hàng] + 1 (hình bên)



Nếu ta nhập vào hạn giao hàng cùng ngày hoặc trước ngày đặt hàng, Access hiển thị ngay lời cảnh báo (Validation text).



### Tạo chỉ mục sắp xếp dữ liệu

1. Mở bảng trong chế độ *Design view*.

2. Nhấn chuột vào nút lệnh Indexes  trên thanh công cụ.

3. Tại dòng còn trống đầu tiên trong cột Index Name, đánh vào tên của chỉ mục.

4. Trong cột Field Name, nhấn chuột vào mũi tên bên phải, chọn trường dùng làm chỉ mục sắp xếp.

5. Nếu muốn chọn nhiều trường để làm chỉ mục, tại dòng tiếp theo trên cột Field Name, chọn trường tiếp theo (Lưu ý để cột Index Name bỏ trống trên dòng đó).

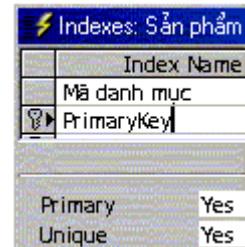


Index Name	Field Name	Sort Order
Mã danh mục	Mã Danh mục hàng	Ascending
PrimaryKey	NCC	Ascending
SupplierID	PVT	Ascending

Làm lại bước này cho đến khi đã có tất cả các trường muốn có trong chỉ mục (tối đa là 10 trường).

### Tạo khóa chính PrimaryKey

Một trường có tính chất khóa là một trường chứa giá trị duy nhất, khác nhau giữa các bản ghi, dùng để phân biệt giữa bản ghi này và bản ghi khác. Để thiết lập tính chất khóa cho trường, mở cửa sổ Indexes, nhấn chuột vào tên trường, chuyển thuộc tính Primary và Unique thành Yes.



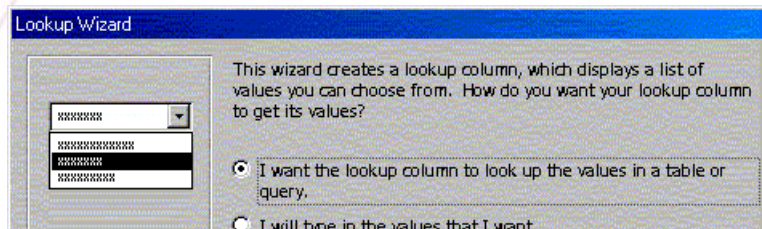
Index Name	Field Name	Sort Order	Primary	Unique
Mã danh mục			Yes	Yes
PrimaryKey				

### Tạo một trường tìm kiếm dữ liệu từ các bảng dữ liệu khác

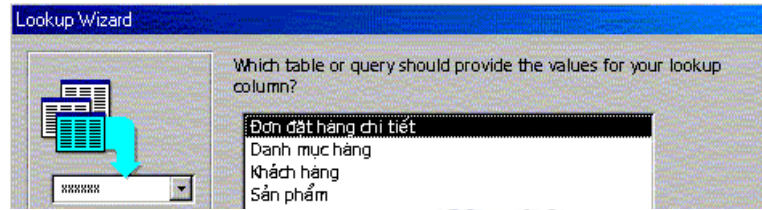
Khi nhập dữ liệu cho một đơn đặt hàng, ta phải nhập vào tên khách hàng. Nếu thông tin về khách hàng đó đã được lưu ở trong bảng Khách hàng thì tốt hơn là ta tìm mã số của khách hàng đó và nhập vào bảng đơn đặt hàng. Ta nên chọn cách làm này, vì nếu ta nhập vào bằng tay thì rất dễ sai sót. Khi nhập liệu bị lỗi, Access sẽ không thể nhận ra khách hàng.

Để làm được điều này, khi thiết kế bảng đơn đặt hàng, ta phải làm như sau :

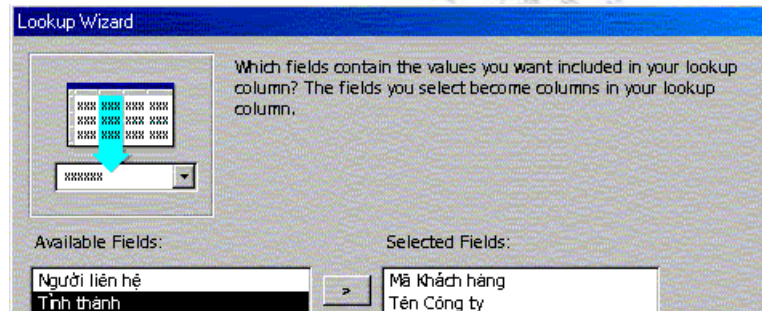
- Nhấn chuột vào trường Khách hàng, chọn Lookup Wizard trong cột Data type. Trong cửa sổ sau hiện ra tiếp theo, chọn mục thứ nhất. Nhấn Next để tiếp tục



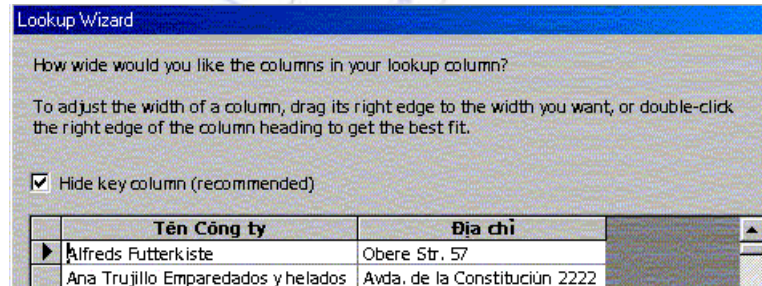
- Access hỏi chọn bảng nào chứa dữ liệu cần tìm. Chọn bảng Khách hàng. Nhấn Next để tiếp tục.



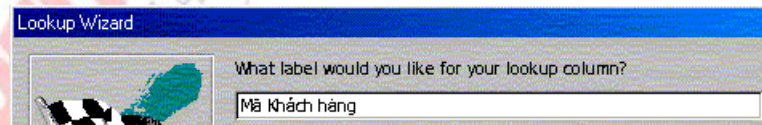
- Access hỏi giá trị của trường nào sẽ được chọn hiển thị. Nếu ta chọn cùng lúc nhiều trường, chỉ trường trên cùng sẽ lưu dữ liệu vào trong bản ghi mới. Ở đây ta chọn 3 trường: Mã khách hàng, Tên công ty, và Địa chỉ.



- Có tất cả 3 cột tương ứng với 3 trường, nhưng ta chọn mục Hide Key column để dấu không hiển thị trường khoá. Nhấn Next để tiếp tục.



- Bước cuối cùng ta đặt tên cho trường : Mã Khách hàng.



- Ghi lại mọi sự thay đổi bằng cách nhấn chuột lên nút lệnh Save trên thanh công cụ.

Mở bảng Đơn đặt hàng ở chế độ nhập liệu. Với mỗi bản ghi bất kỳ, khi ta cần nhập vào một khách hàng, nhấn chuột vào mũi tên ở bên phải của cột Khách hàng, một danh sách các khách hàng từ trước đến nay của Doanh nghiệp hiện ra gồm 2 cột, cột Tên Khách hàng và cột Địa chỉ Khách hàng. Nhấn chuột vào tên khách hàng trên danh sách này để nhập dữ liệu vào trường.

Đơn đặt hàng : Table			
	Mã Đ Đ Hg	Khách hàng	Nhân viên bán hàng
+	10248	Around the Horn	Buchanan, Steven
▶	10249	Toms Spezialitoten	Suyama, Michael
+	10250	Toms Spezialitoten Luisenstr. 48	Peacock, Margaret
+	10251	Tortuga Restauranl Avda. Azteca 123	Leverling, Janet
+	10252	Tradióo Hipermerc Av. Inòs de Castro, 41	Peacock, Margaret
+	10253	Trail's Head Gourm 722 DaVinci Blvd.	Leverling, Janet
+	10254	Vaffeljernet Smagslùget 45	Buchanan, Steven
+	10255	Victuailles en stock 2, rue du Commerce	Dodsworth, Anne
+	10256	Vins et alcools Che 59 rue de l'Abbaye	Leverling, Janet
+	10257	Wartian Herkku Torikatu 38	Peacock, Margaret

Rõ ràng với cách thiết kế trường tìm kiếm như thế này, ta có thể kiểm soát hoàn toàn quá trình nhập liệu, dữ liệu nhập vào luôn luôn đúng. Nếu bạn nhập vào tên một khách hàng mà chưa có ở bảng Khách hàng, chương trình sẽ báo lỗi. Hơn nữa, dù về mặt hiển thị, cột Khách hàng ở bảng đơn đặt hàng hiển thị tên khách hàng, nhưng thực tế cột này lại lưu giá trị Mã Khách hàng. Điều này cho phép giảm kích thước của CSDL vì mã khách hàng thì ngắn hơn tên khách hàng.

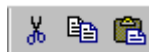
### Mở một bảng dữ liệu

1. Trong cửa sổ Database, click Tables dưới Objects.
2. Chọn tên của Bảng muốn mở.
3. Để mở bảng ở chế độ thiết kế, Design view, nhấn Design trong thanh công cụ của cửa sổ Database.
4. Để mở bảng trong khung nhìn dạng lưới, Datasheet, nhấn Open.

### Sử dụng thanh công cụ của Trang dữ liệu Datasheet

Sau khi đã thiết kế xong bảng dữ liệu, bấm  để chuyển sang chế độ nhập liệu.

Các nút lệnh thao tác chính đối với một bảng trên thanh công cụ Toolbar:



Cắt, copy, dán dữ liệu.



In, xem trước khi in một báo cáo



Sắp xếp theo thứ tự tăng dần, giảm dần



Lọc dữ liệu theo giá trị chọn, theo form.



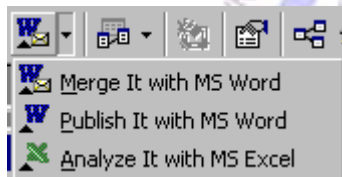
Tìm kiếm thông tin, thay thế bằng thông tin mới.



Thêm, xoá một bản ghi.



Mô hình liên kết giữa các bảng trong một CSDL.



Liên kết với các ứng dụng Office khác.



Xoá một dòng, một bản ghi



Lập Khóa chính cho một bảng.

Ngoài ra:

- Để thêm một trường dữ liệu, chọn Column trong menu Insert.
- Để đổi tên một cột dữ liệu, nhấp đúp mũi tên chuột lên tiêu đề cột, đánh vào tên mới.

- Để thay đổi thứ tự của một cột, nhấn trỏ chuột vào cột đó, giữ chuột và kéo cột đó đến vị trí mới.
- Muốn giấu đi một cột, dùng chuột chọn cột đó, chọn lệnh Hide columns trong menu Format.
- Muốn luôn cho một cột được hiển thị dù cho ta cuộn màn hình, dùng lệnh Format\Freeze columns.
- Muốn thay đổi độ rộng của một cột, một dòng:

Order ID	Customer ID
10000	Franchi S.p.A.
10001	Mère Paillarde
10002	Folk och få HB

Nháy đúp chuột vào lề của cột để thay đổi chiều rộng của cột đúng bằng chiều rộng của dữ liệu

Nhấn và giữ trỏ chuột, kéo lề của dòng, cột để thay đổi chiều cao, độ rộng của dòng hay cột đó.

### 5.2.3.3. Tạo các liên kết giữa các bảng dữ liệu

#### a. Tại sao lại phải xác định liên kết ?

Sau khi đã xây dựng và nhập xong dữ liệu vào các bảng, ta phải chỉ cho Microsoft Access làm thế nào để kết xuất các dữ liệu. Bước đầu tiên là để làm điều đó là xác định các mối liên kết, relationship, giữa các bảng.

Ví dụ ở đây, các trường từ 3 bảng khác nhau (*nhân viên, khách hàng, bán hàng*) được tập hợp lại để hiển thị nội dung của cùng một hóa đơn bán hàng. Sự tập hợp này được thực hiện nhờ có các liên kết giữa các bảng.

Liên kết sẽ tìm kiếm các bản ghi có giá trị giống nhau tại trường dữ liệu chung, thông thường là giữa trường có thuộc tính khóa chính, vì nó chứa giá trị ID duy nhất đặc trưng cho từng bản ghi, và một khóa ngoại - Foreign key - từ bảng khác.

Khóa chính

nhan vien : Table	
maNV	Ten nhan vien
A01	Thu lien
A02	lan
*	

trường maNV xuất hiện trong cả 2 bảng

banhang : Table			
maNV	makh	tenkh	tenhang
A01	Kh01	ha	may tinh
*			

Khóa ngoại

Ví dụ, mỗi nhân viên có thể liên kết đến nhiều mặt hàng mà nhân viên đó đã bán, bằng cách tạo relationship giữa bảng nhân viên và bảng bán hàng thông qua trường maNV

**b. Các loại quan hệ**

*Quan hệ Một - Nhiều*

Mỗi quan hệ một-nhiều là loại hình phổ biến nhất của liên kết. Trong liên kết này, một bản ghi ở Bảng A có thể có nhiều bản ghi tương ứng ở Bảng B, nhưng mỗi bản ghi ở Bảng B chỉ có một bản ghi tương ứng duy nhất ở Bảng A.

Ví dụ, bảng *nhân viên* có liên kết một-nhiều với bảng *bán hàng* vì mỗi nhân viên có thể bán nhiều hàng, trong khi một hàng chỉ có thể thực hiện bởi một nhân viên.

Một nhân viên

nhan vien : Table	
maNV	Ten nhan vien
A01	Thu lien
A02	lan
*	

Có thể bán nhiều hàng

banhang : Table			
maNV	makh	tenkh	tenhang
A01	kh01	ha	may tinh 1
A01	kh02	lan	may tinh 2
a02	kh03	thu	may tinh 3
*			

nhưng một hàng chỉ được bán bởi một nhân viên

*Quan hệ nhiều - nhiều*

Trong loại hình này, một bản ghi ở Bảng A có thể có nhiều bản ghi tương ứng ở Bảng B và ngược lại. Quan hệ này chỉ được thể hiện khi có một Bảng thứ 3 chứa dữ liệu từ cả hai Bảng A và Bảng B.

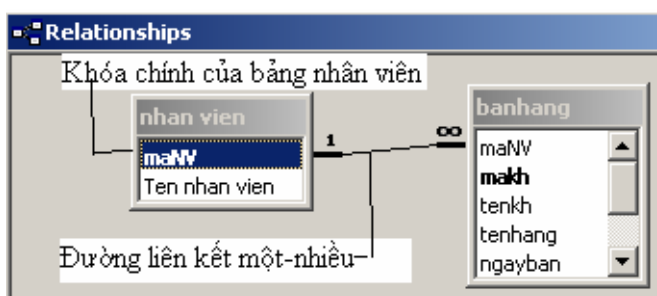
Ví dụ, mỗi liên kết giữa bảng *nhân viên* và bảng *khách hàng* là nhiều - nhiều vì nhân viên có thể bán cho nhiều khách hàng và mỗi khách hàng có thể mua nhiều mặt hàng của nhiều nhân viên.

*Quan hệ một - một*

Trong mỗi quan hệ loại này, mỗi bản ghi ở Bảng A chỉ có một bản ghi duy nhất tương ứng ở Bảng B và ngược lại. Dạng quan hệ này thường hiếm, bởi thông tin loại này có thể lưu luôn vào một bảng chung..

**c. Xác định liên kết**

Xây dựng liên kết bằng cách thêm các bảng có liên quan vào cửa sổ Relationships Window, rồi kéo trường khóa từ bảng này và thả vào trường tương ứng của bảng kia.



Kiểu liên kết Access tạo ra sẽ dựa trên các trường liên kết được xác định :

- Liên kết một - nhiều nếu chỉ có một trong các trường liên kết là khóa chính hoặc có thuộc tính chỉ mục duy nhất (unique index)
- Liên kết một - một nếu cả hai trường liên kết là khóa chính hoặc có thuộc tính unique indexes.
- Liên kết Nhiều - nhiều khi có một bảng thứ 3 có khóa chính bao gồm hai trường là khóa ngoại (foreign key: một hoặc nhiều trường mà tham chiếu đến trường khóa chính của các bảng khác) từ hai bảng kia.

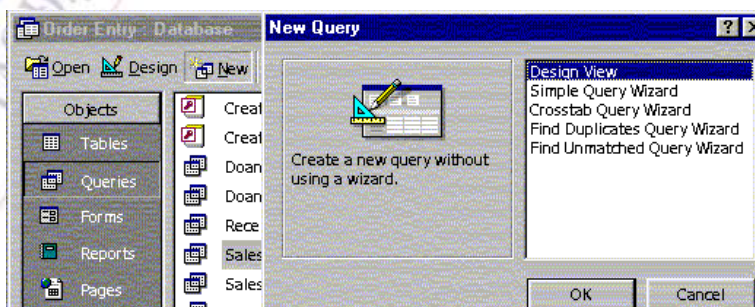
#### 5.2.3.4. Truy vấn - Query

Như đã nói ở trên, truy vấn là công cụ mạnh của Access cho phép người sử dụng dùng ngôn ngữ truy vấn có cấu trúc SQL (Structure Query Language) hoặc công cụ truy vấn bằng ví dụ QBE (Query by Example) để thực hiện việc rút trích, chọn lựa dữ liệu hoặc cập nhật dữ liệu (thêm, sửa, xóa) trên các bảng. Có nhiều loại truy vấn, thông dụng nhất là Select query, ngoài ra còn có một số loại khác như: Update query, Append query, Delete query, Crosstab query...

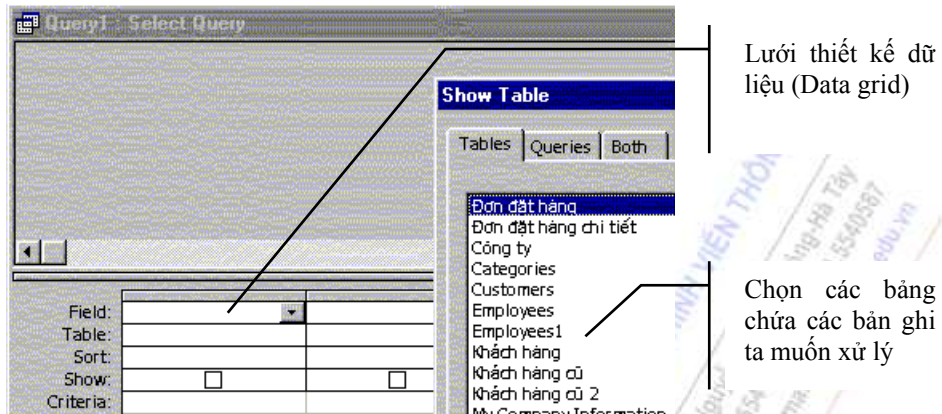
##### a. Quy trình tạo mới một query

- Trong cửa sổ Database, nhấn Queries.
- Nhấn New trên thanh công cụ.

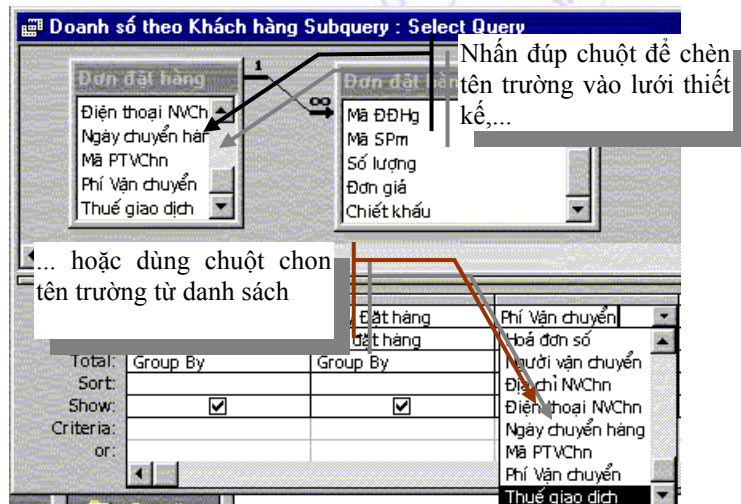
Trong hộp thoại New query vừa hiện ra, chọn Design View nếu muốn tự thiết kế truy vấn, hoặc chọn các wizard nếu muốn Access tự tạo ra các truy vấn cho mình.



Nếu ta chọn Design View, cửa sổ Show Table hiện ra yêu cầu ta chọn các bảng cần để xử lý thông tin. Nhấn đúp chuột vào tên của bảng mà ta lấy các bản ghi của nó làm cơ sở.

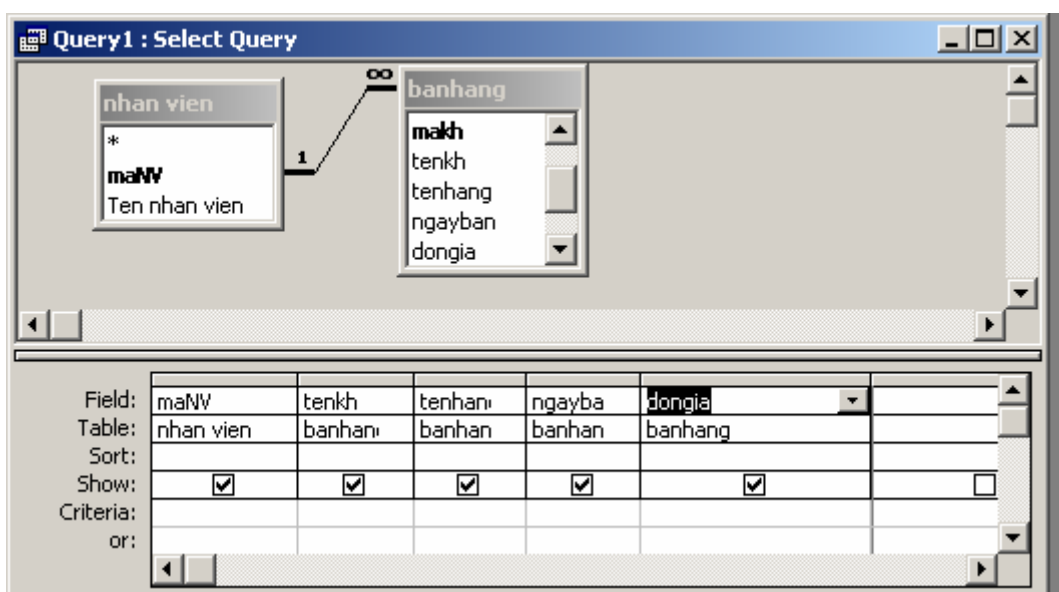


Xác định mối liên kết giữa các bảng trong truy vấn, nếu có (Dùng chuột kéo trường liên kết giữa hai bảng từ bảng này sang bảng kia).



Tại dòng Field, nhấn vào mũi tên bên phải của cột để làm hiện ra danh sách các trường dữ liệu (field list), chọn trường cần thiết từ danh sách. Hoặc ta có thể dùng chuột nhấn đúp vào tên trường hiện ra ở phía trên, Access tự động chèn trường đó vào lưới thiết kế





## b. Cách thức chỉnh sửa một query

### Thêm một bảng hoặc một truy vấn khác vào một truy vấn

1. Mở truy vấn ở chế độ Design view.

2. Nhấn Show Tables 

3. Trong hộp thoại Show Tables, nhấn tab liệt kê danh sách các đối tượng chứa dữ liệu ta cần làm việc. Nếu bảng ta cần lại ở một CSDL khác thì phải liên kết bảng đó vào CSDL hiện hành trước.

4. Nhấn vào tên của đối tượng ta muốn thêm vào truy vấn.

5. Nhấn Add, rồi Close.

#### Chú ý:

- Có thể thêm bảng hoặc truy vấn khác vào một truy vấn bằng cách dùng chuột kéo bảng hay truy vấn đó từ cửa sổ Database vào cửa sổ thiết kế của truy vấn (Design view).

- Nếu có nhiều bảng hay truy vấn trong một truy vấn, chúng cần phải được kết nối.

### Đưa dữ liệu từ nhiều bảng hoặc truy vấn vào một truy vấn

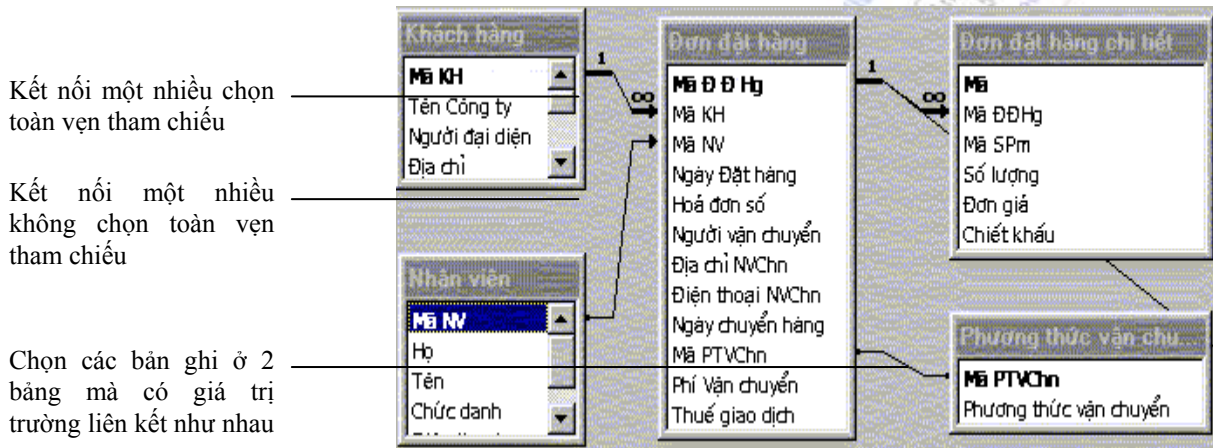
Khi thêm nhiều bảng hoặc truy vấn vào một truy vấn, ta cần kiểm tra trường chung giữa chúng phải được kết nối với nhau để Access có thể kết nối dữ liệu.

Nếu các bảng trong một truy vấn không được kết nối với nhau, trực tiếp hoặc gián tiếp, Access không thể xác định được các bản ghi nào được kết nối với các bản ghi nào, do đó nó hiển thị tất cả các tổ hợp các bản ghi (gọi là một "cross-product" hay "Cartesian product") giữa các bảng. Vì vậy, nếu mỗi bảng có 10 bản ghi, kết quả của truy vấn trên cơ sở hai bảng sẽ chứa 100 bản ghi (10\*10). Điều đó cũng có nghĩa là truy vấn cần nhiều thời gian để chạy và thậm chí sẽ tạo ra các kết quả vô nghĩa.

Nếu ta đã tạo các quan hệ giữa các bảng trong cửa sổ Relationships, Access tự động hiển thị

đường liên kết khi ta thêm các bảng vào truy vấn trong chế độ Design view. Nếu đã chọn duy trì tính toàn vẹn tham chiếu, Access cũng sẽ hiển thị "1" ở trên đường kết nối cho biết bảng nào là ở bên một trong mỗi quan hệ một - nhiều và dấu vô hạn ở bên bảng nhiều.

Nếu chưa bao giờ tạo quan hệ, Access tự động tạo các liên kết nếu ta thêm hai bảng vào một truy vấn và các bảng có một trường cùng tên, chứa kiểu dữ liệu tương thích lẫn nhau, và nếu một trong những trường liên kết có thuộc tính khóa chính. Biểu tượng một và nhiều không được hiển thị trong trường hợp này, bởi vì ta không chọn duy trì tính toàn vẹn tham chiếu.



### Cách thực hiện các tính toán trong một truy vấn

Có nhiều kiểu tính toán trong một truy vấn: tính tổng, tính trung bình giá trị của một trường, nhân giá trị của nhiều trường, ... Kết quả hiển thị dựa trên dữ liệu hiện tại được lưu trong các bảng, và Access sẽ chạy lại các tính toán khi ta mở một truy vấn, vì vậy kết quả luôn phản ánh số liệu mới nhất.

Để tính toán, ta có thể sử dụng các hàm do Access cung cấp hoặc các hàm do ta tự viết ra. Sử dụng các hàm có sẵn, được gọi là hàm tính tổng - totals calculation, nếu ta muốn tính toán trên một nhóm hoặc toàn bộ các bản ghi. Các hàm này bao gồm sum, average, count, minimum, maximum, standard deviation, hoặc variance. Lưu ý chỉ chọn một hàm tính tổng cho mỗi trường ta cần xử lý số liệu, trong dòng Total trong lưới thiết kế.

Trong lưới thiết kế, ta có thể chỉ ra các tiêu chuẩn để hạn chế nhóm các bản ghi sẽ được tính toán.

Một tính toán tùy chọn cho phép ta tính toán các giá trị dạng số, ngày tháng và ký tự dựa trên các số liệu từ một hoặc nhiều trường, ví dụ tính tiền thuế, tổng tiền trả, ... Để làm được điều này, ta phải tạo trường tính toán, calculated field, ngay trong lưới thiết kế, bằng cách nhập vào biểu thức vào một ô Field còn trống trong lưới thiết kế truy vấn.

#### Ví dụ:

Tính tổng số đơn vị sản phẩm còn ở trong kho bằng cách cộng hai trường Số lượng sản phẩm còn trong kho và Số lượng sản phẩm đã đặt hàng sắp nhập kho:

Sum([SLgKho]+[SLgĐHg]).

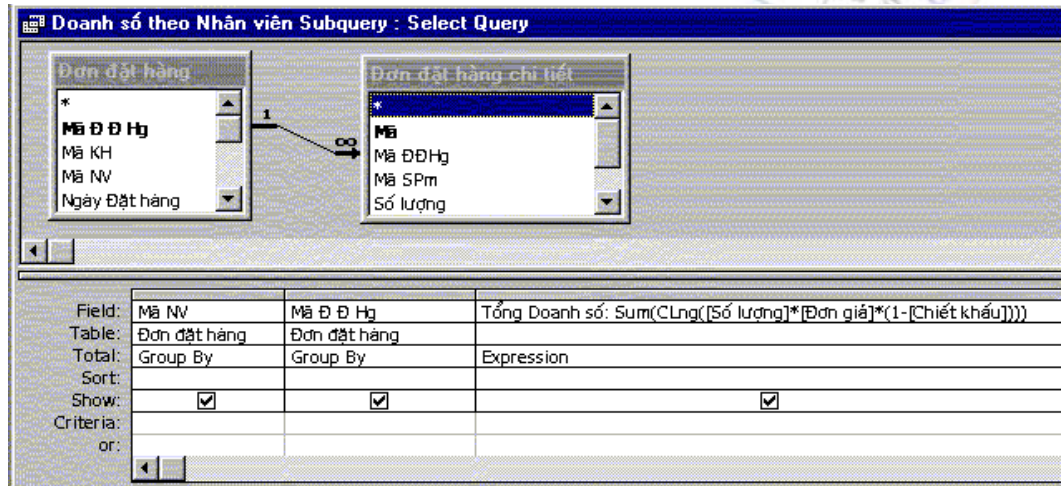
Ta cũng có thể dùng Update query để sửa đổi đồng loạt giá trị cho một trường, ví dụ tăng

đồng loạt 5% giá bán các sản phẩm.

**Ví dụ:**

Để khen thưởng nhân viên giao dịch, Giám đốc muốn biết thành tích bán hàng mà họ đã thực hiện được. Khi đó, phòng bán hàng phải liệt kê chi tiết từng hợp đồng bán hàng mà mỗi nhân viên đã thực hiện được. Thông tin thể hiện bao gồm Tên nhân viên, Mã đơn đặt hàng, tổng giá trị hàng bán trong đơn đặt hàng đó (không tính tiền vận chuyển).

Ta thiết kế như sau:



Cột thứ 3 là một trường tính toán. Trường này được đặt tên là “Tổng Doanh số”, với biểu thức tính toán là:

$$[Số\ lượng] * [Đơn\ giá] * (1 - [Chiết\ khấu])$$

Để đảm bảo thống nhất kích thước của kết quả, ta dùng hàm CLng (convert long) nhằm mục đích chuyển đổi dữ liệu kiểu số sang loại Long Integer.

$$CLng([Số\ lượng] * [Đơn\ giá] * (1 - [Chiết\ khấu]))$$

Ta muốn tính tổng giá trị của hợp đồng, vậy nên dùng hàm Sum:

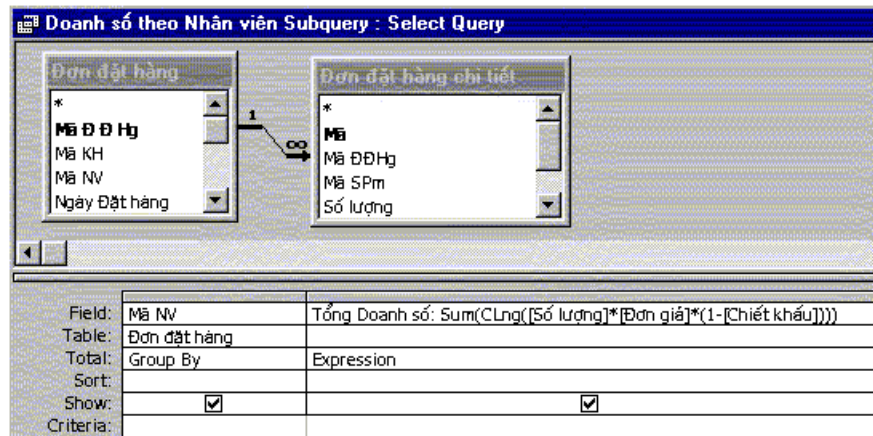
$$Sum(CLng([Số\ lượng] * [Đơn\ giá] * (1 - [Chiết\ khấu])))$$

Kết quả khi chạy truy vấn sẽ như sau:

Nhân viên nhận đặt hàng	Mã Đ Đ Hg	Tổng Doanh số
Hoàng Mạnh	2	332
Hoàng Mạnh	4	9411
Nguyễn Trọng	1	2970
Nguyễn Trọng	3	209

Ở đây ta thấy nhân viên Hoàng Mạnh bán được 2 hợp đồng, số 2 và số 4, với giá trị mỗi hợp đồng là \$332 và \$9.411.

Đến đây, Giám đốc muốn tổng kết mỗi nhân viên đặt được doanh số bán hàng là bao nhiêu. Yêu cầu này có nghĩa là, ông ta không quan tâm đến cá nhân mỗi đơn đặt hàng (mã, ngày tháng, số lượng mỗi loại sản phẩm,...), mà chỉ quan tâm đến tổng tiền hàng cộng dồn mà nhân viên đó đã bán được.



Thực hiện yêu cầu này, ta đơn giản là chỉ bỏ đi trường MãĐĐHg ở trong lưới thiết kế trong truy vấn vừa tạo phía trên: dùng con chuột nhấn vào thanh tiêu đề của cột đó, cột chuyển sang màu đen, bấm phím Delete.

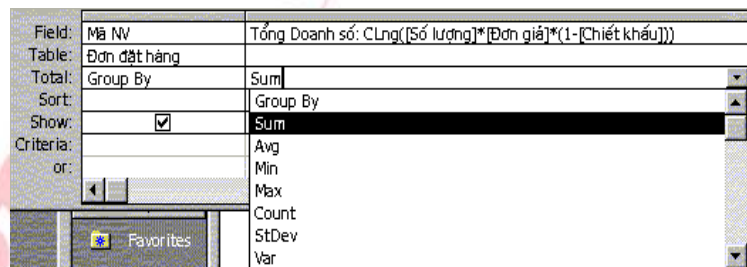
Kết quả thể hiện như hình bên:

Nhân viên nhận đặt hàng	Tổng Doanh số
Hoàng Mạnh	9743
Nguyễn Trọng	3179

Tuy vậy, ta cũng có cách như sau:

Đánh vào biểu thức:

Cln([Số lượng]\*[Đơn giá]\*(1-[Chiết khấu])) trong dòng Field, chọn Sum trong dòng Total. Kết quả cho ra giống hệt trên.



### c. Các truy vấn hành động - Action query

Truy vấn hành động là một loại truy vấn mà có tác động thay đổi đến rất nhiều bản ghi chỉ trong một lần thực hiện. Có 4 kiểu truy vấn hành động:

- Truy vấn xoá: Delete query
- Truy vấn sửa : Update query
- Truy vấn bổ sung (thêm bản ghi): Append query
- Truy vấn tạo bảng: Make-table query

#### Delete query:

Xoá một nhóm các bản ghi từ một hoặc nhiều bảng.

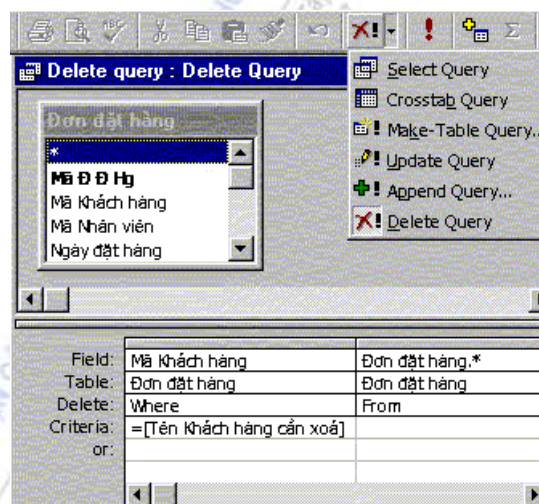
**Ví dụ:** Xoá tất cả các đơn đặt hàng của khách hàng có mã số là ALFKI.

Các bước thiết kế được mô tả như hình dưới.

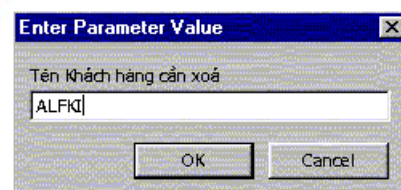
- Nhấn New, chọn Design view, chọn bảng Đơn đặt hàng vào trong lưới thiết kế. Nhấn vào biểu tượng chọn Action query, chọn Delete Query.

- Nhấn đúp vào dấu \* ở trong bảng Đơn đặt hàng. Hành động này sẽ chọn tất cả các trường của bảng vào trong lưới thiết kế.

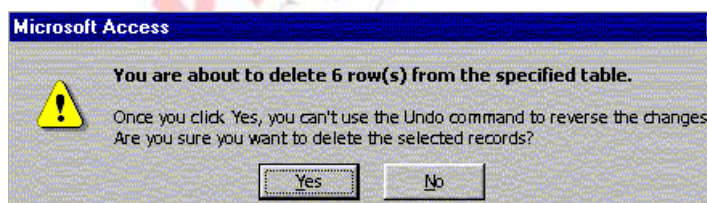
- Nhấn đúp vào dòng Mã Khách hàng trong bảng Đơn đặt hàng. Tại dòng Criteria, đánh vào công thức: =[Tên khách hàng cần xoá].



- Nhấn vào dấu ! để chạy truy vấn. Khi Access chạy truy vấn này, nó tự động đưa ra một cửa sổ hỏi ta Tên khách hàng cần xoá là gì?. Ta gõ vào "ALFKI", nhấn OK



- Để đảm bảo tránh rủi ro, Access sẽ hỏi lần cuối nếu ta muốn xoá các bản ghi này. Nhấn Yes nếu đồng ý xoá, Cancel nếu huỷ lệnh.



Đến đây, khi ta nhấn Yes, cả 6 bản ghi thỏa mãn có Mã khách hàng = ALFKI sẽ bị xoá.

### Update query:

Thực hiện các thay đổi về mặt tổng thể đến một nhóm các bản ghi của một hoặc nhiều bảng. Có nghĩa là dùng để chỉnh sửa dữ liệu của các trường trong một hoặc nhiều bảng

Các bước tiến hành:

- Chọn các bảng truy vấn/nguồn
- Chọn menu Query/Update Query
- Kéo các trường cần sửa và các trường dùng để lập điều kiện vào hàng Field.

Chú ý: cùng một trường có thể là trường cần sửa, vừa là trường điều kiện





- Đặt biểu thức vào ô Update To của các trường cần sửa. Giá trị của các biểu thức sẽ là giá trị mới của các trường này
- Đặt các biểu thức điều kiện vào các ô Criteria của các trường điều kiện
- Ghi thiết kế và đặt tên cho truy vấn
- Thực hiện truy vấn.

Kết quả: Một số trường sẽ được chuyển đổi theo như thiết kế

### **Make-table query**

Make-table query để tạo một bảng mới từ một phần hoặc tất cả dữ liệu chứa trong một hoặc nhiều bảng. Make-table query thường được dùng để tạo bản lưu số liệu, tạo bảng chứa tất cả các bản ghi cũ trước khi xoá chúng ra khỏi CSDL hiện tại.

Quy trình:

- Tạo một truy vấn, chọn các bảng hoặc truy vấn chứa các bản ghi ta muốn có trong bảng mới.
- Trong Design view, nhấn vào mũi tên bên cạnh nút lệnh Query Type  trên thanh công cụ, nhấn Make Table. Hộp thoại Make Table xuất hiện.
- Trong hộp Table Name, nhập vào tên của bảng ta muốn tạo.
- Nhấn Current Database để tạo bảng mới trong CSDL hiện thời đang mở; hoặc nhấn Another Database và đánh vào tên của CSDL ta muốn lưu bảng mới. Đánh vào tên đường dẫn nếu cần.
- Nhấn OK.
- Kéo các trường muốn có trong bảng mới từ field list vào khung thiết kế truy vấn (design grid).
- Nhập vào tiêu chuẩn trong ô Criteria.
- Xem kết quả của bảng mới trước khi tạo nó, nhấn View  trên thanh công cụ. Muốn quay lại thiết kế tiếp, nhấn View .
- Nhấn Run  trên thanh công cụ.





Lưu ý:

- Bấm CTRL+BREAK để ngắt quá trình tạo bảng mới khi đang chạy.
- Dữ liệu trong bảng mới được tạo sẽ không kế thừa các thuộc tính của bảng gốc.

### **Append query:**

Thêm các bản ghi từ một bảng vào một bảng khác

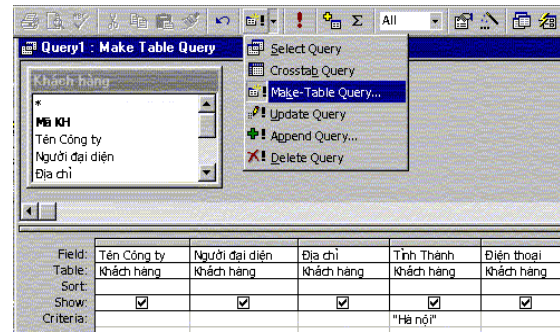
Quy trình:

- Tạo một truy vấn chứa các bản ghi mà ta muốn thêm vào bảng khác.
  - Trong cửa sổ Design view, nhấn mũi tên bên cạnh Query Type  ở trên thanh công cụ, nhấn Append. Hộp thoại Append xuất hiện.
  - Trong hộp Table Name, nhập tên của bảng mà ta muốn thêm các bản ghi vào.
  - Nhấn Current Database, nếu bảng ở trong database hiện hành; hoặc nhấn Another Database và đánh vào tên của database nơi bảng đang được lưu. Đánh vào đường dẫn nếu cần thiết.
  - Ta cũng có thể nhập đường dẫn tới một CSDL Microsoft FoxPro, Paradox, hoặc dBASE.
  - Nhấn OK.
  - Dùng chuột lôi từ danh sách các trường vào lưới thiết kế các trường mà ta muốn thêm vào bảng khác và các trường mà ta muốn lấy làm tiêu chuẩn để chọn lựa các bản ghi.
  - Nếu tất cả các trường của hai bảng có cùng tên như nhau, ta đơn giản chỉ cần lôi dấu (\*) vào lưới thiết kế.
  - Nếu các trường ta vừa chọn có cùng tên ở cả hai bảng, Access tự động điền tên tương ứng vào dòng Append To. Nếu không, ta phải tự nhập vào tên của các trường của bảng mà ta muốn thêm bản ghi tới vào trong dòng Append To.
  - Trong ô Criteria, chỉ rõ tiêu chuẩn của các bản ghi được lựa chọn.
  - Để kiểm tra các bản ghi trước khi chạy query, nhấn vào View  trên thanh công cụ.
- Để quay trở lại Design view, nhấn lần nữa View .
- Nhấn **Run**  để thực hiện truy vấn.

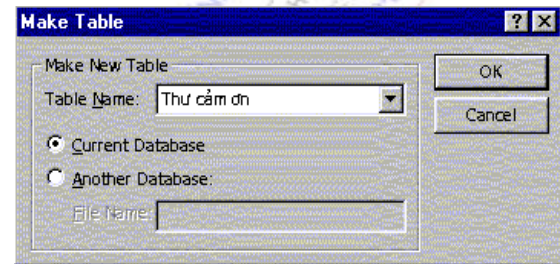
**Ví dụ:**


Nhân dịp 10 năm ngày thành lập Doanh nghiệp, Giám đốc quyết định sẽ gửi thư cảm ơn đến tất cả các khách hàng tại Hà Nội. Bộ phận Tin học nghiên cứu xem xét dữ liệu, nhận thấy tất cả các thông tin về khách hàng được lưu ở hai bảng, một tên là Khách hàng chứa các thông tin liên quan đến các khách hàng của doanh nghiệp từ năm 1999, một tên là Khách hàng cũ chứa các thông tin liên quan đến các khách hàng của Doanh nghiệp từ trước năm 1999. Thực hiện nhiệm vụ, Bộ phận Tin học khi đó sẽ phải kết xuất từ hai bảng các bản ghi mà trường Địa chỉ có giá trị là Hà Nội vào một bảng thứ ba tên là Thư khuyến mại. Họ tạo ra một bảng mới có tên là Thư khuyến mại, trong đó chứa tất cả các bản ghi của các khách hàng có trụ sở tại Hà Nội.

**Bước 1:** Tạo query Make-table dựa trên bảng Khách hàng. Chỉ ra các trường mà ta muốn có ở trong bảng mới. Xem hình.



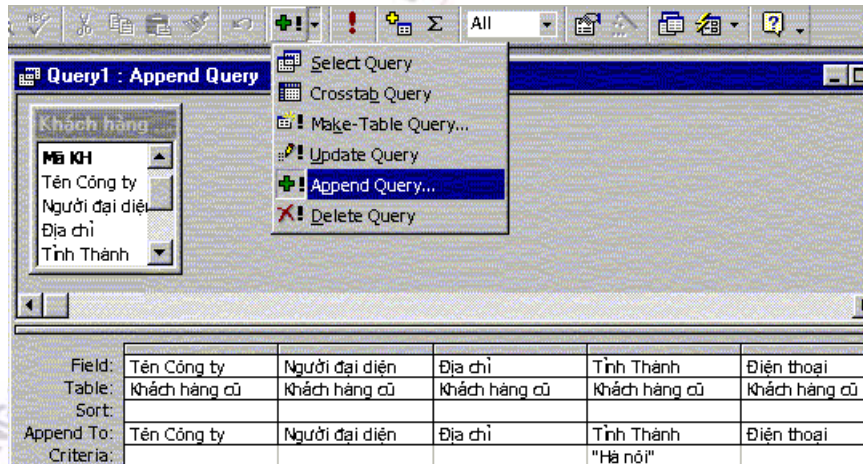
**Bước 2:** Chọn Make-table query từ trong danh sách Action Query. Trong hộp thoại Make table, đánh vào Thư cảm ơn rồi nhấn OK.



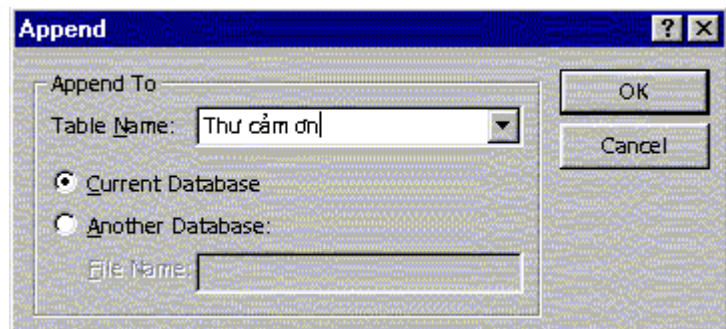
**Bước 3:** Nhấn Run  để thực hiện truy vấn. Đóng lại truy vấn này lại.

**Bước 4:** Tạo một Append query dựa trên bảng Khách hàng cũ. Nhấn New trong cửa sổ Database, chọn Design view, chọn bảng Khách hàng cũ từ danh sách các bảng.

Dùng chuột lôi các trường Tên công ty, Người đại diện, Địa chỉ, Tỉnh thành, điện thoại vào trong lưới thiết kế (xem hình). Trong cột chứa trường Tỉnh thành, tại dòng Criteria đánh vào “Hà Nội”.



**Bước 5:** Chọn kiểu Append query, trong hộp thoại Append, nhập vào Thư cảm ơn trong mục Table Name.





**Bước 6:** Kiểm tra lại trước khi chạy query. Nhấn View. Nếu các bản ghi được liệt kê đúng theo yêu cầu đặt ra, chuyển sang bước tiếp.

Query1 : Append Query					
	Tên Công ty	Người đại diện	Địa chỉ	Tỉnh thành	Điện thoại
▶	Ngân hàng Công thư	GD Nguyễn Văn M	21 Bà triệu	Hà nội	004-7841051
▶	Ngân hàng Nhà nước	Thống đốc Hà Duy	1 Tràng tiền	Hà nội	

**Bước 7:** Nhấn **Run** để thực hiện truy vấn. Đóng lại truy vấn này lại.

Đến đây bộ phận tin học đã lọc xong dữ liệu. Việc in thư, bao gồm phong bì và lá thư, cho từng khách hàng sẽ bằng cách hoặc là tự động dùng tính năng Mail-merge trong Microsoft Word, hoặc là sử dụng Report trong Microsoft Access.

#### d. Xử lý một số tình huống (thông qua ví dụ)

##### Tình huống 1

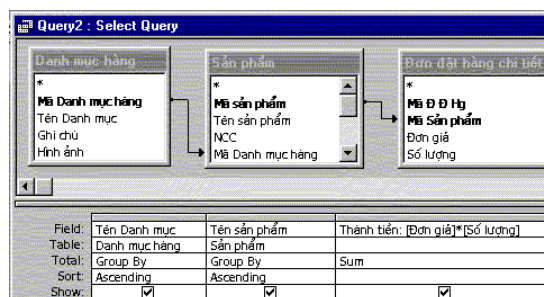
Báo cáo tổng doanh số bán của từng sản phẩm, liệt kê theo danh mục của sản phẩm đó và theo tên, thứ tự tăng dần.

Giải pháp:

- Do sắp xếp theo danh mục sản phẩm và tên của sản phẩm, vậy nên trong truy vấn ta phải có hai bảng Danh mục hàng và Sản phẩm nhằm lấy giá trị của hai trường Tên danh mục và Tên sản phẩm để hiển thị.

- Doanh số của một sản phẩm trong một đơn đặt hàng là bằng số lượng nhân với đơn giá bán. Dữ liệu này được lưu trong bảng Đơn đặt hàng chi tiết nên phải đưa nó vào lưới thiết kế. Để tính tổng doanh số, tại dòng Field nhập vào biểu thức: Đơn giá ta phải dùng hàm Sum tại dòng Total của lưới thiết kế.

- Nhằm sắp xếp, ta chọn *Ascending* tại dòng *Sort*.



- Sau khi đã thiết kế xong, nhấn *View* để xem kết quả.

Tên Danh mục	Tên sản phẩm	Thành tiền
▶ Beverages	Chai	\$14,277.60
Beverages	Chang	\$18,559.20
Beverages	Chartreuse verte	\$13,150.80
Beverages	Cyte de Blaye	\$149,984.20
Beverages	Guaranô Fantôstica	\$4,782.60
Beverages	Ipoh Coffee	\$25,079.20
Beverages	Lakkalikúuri	\$16,794.00
Beverages	Laughing Lumberjack Lager	\$2,562.00
Beverages	Outback Lager	\$11,472.00
Beverages	Rhûnbrou Klosterbier	\$8,650.55
Beverages	Sasquatch Ale	\$6,678.00
Beverages	Steeleye Stout	\$14,536.80
Condiments	Aniseed Syrup	\$3,080.00
Condiments	Chef Anton's Cajun Seasoning	\$9,424.80

### Tình huống 2

Báo cáo tổng doanh số bán ra của mỗi danh mục hàng.

Giải pháp:

- Vì cần có tên danh mục hàng nên phải đưa bảng Danh mục hàng vào trong truy vấn.
- Doanh số bán được tính từ bảng Đơn đặt hàng chi tiết, bằng cách nhân giá trị của hai trường Đơn giá và Số lượng. Ta dùng hàm Sum để tính tổng của nó.
- Vì hai bảng Danh mục hàng và bảng Đơn đặt hàng chi tiết không có trường nào chung để liên kết, vậy phải đưa vào một bảng trung gian có chứa trường chung với cả hai bảng trên, đó là bảng Sản phẩm: trường Mã sản phẩm chung với Đơn đặt hàng chi tiết và trường Mã danh mục chung với bảng Danh mục hàng.
- Xác lập mối quan hệ giữa các bảng.

- Cách thiết kế giống trường hợp trên, nhưng bỏ đi cột của trường Tên sản phẩm.

Field:	Tên Danh mục	Thành tiền: [Đơn giá]*[Số lượng]
Table:	Danh mục hàng	
Total:	Group By	Sum
Sort:	Ascending	
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

- Nhấn View để xem kết quả

Tên Danh mục	Thành tiền
Beverages	\$286,526.95
Condiments	\$113,694.75
Confections	\$177,099.10
Dairy Products	\$251,330.50
Grains/Cereals	\$100,726.80
Meat/Poultry	\$178,188.80
Produce	\$105,268.60
Seafood	\$141,623.09

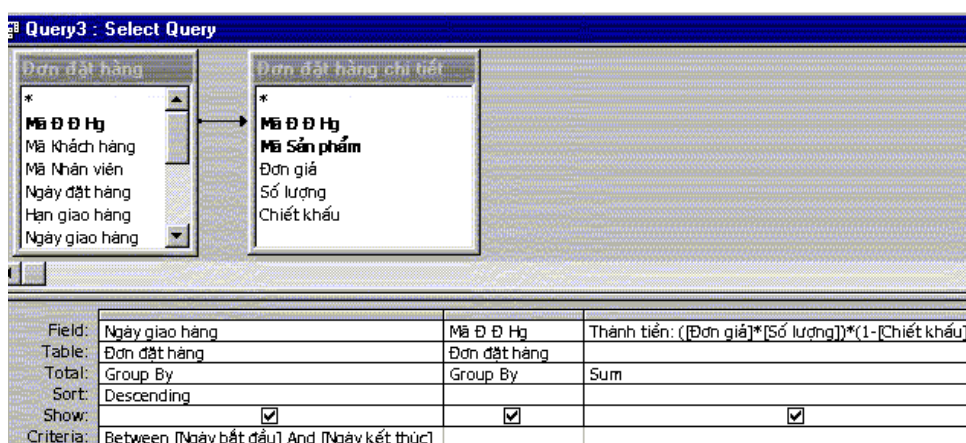
### Tình huống 3

Liệt kê doanh số tất cả các đơn đặt hàng đã hoàn thành giữa hai ngày nào đó.

Giải pháp:

- Tất cả các thông tin về mỗi một lần bán hàng được lưu trong hai bảng Đơn đặt hàng và Đơn đặt hàng chi tiết, do đó ta sẽ đưa hai bảng này vào trong lưới thiết kế. Thiết lập mối quan hệ giữa hai bảng này.

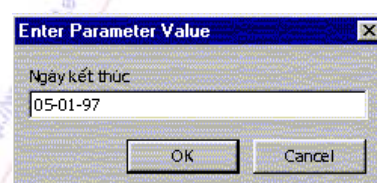
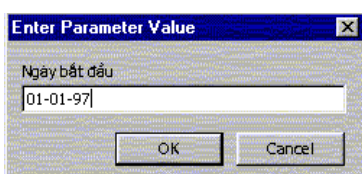
- Một khi đơn đặt hàng đã hoàn thành thì trường Ngày giao hàng sẽ phải chứa giá trị ngày tháng, bằng không sẽ là giá trị trống. Vì vậy, chọn trường Ngày giao hàng làm thông tin đầu tiên. Để giới hạn chỉ cho những đơn hàng giữa hai ngày nhất định, ta nhập vào công thức: Between [Ngày bắt đầu] And [Ngày kết thúc] trong dòng Criteria. Lấy thêm trường Mã Đơn đặt hàng. Tạo thêm thông tin về Doanh số hàng bán (xem hình). Khi chạy truy vấn này, Access sẽ không biết [Ngày bắt đầu] và [Ngày kết thúc] là gì, vì vậy nó coi đó là một biến số, và do đó sẽ hiển thị một cửa sổ để hỏi giá trị của các biến số này.



- Nhấn nút Datasheet View để xem kết quả hiển thị.

Đầu tiên, Access sẽ hỏi ngày bắt đầu

... sau đó hỏi ngày kết thúc



... kết quả truy vấn hiển thị các bản ghi như hình bên.

Ngày giao hàng	Mã Đ Đ Hg	Thành tiền
03-Jan-1997	10393	2556.95
03-Jan-1997	10394	442
03-Jan-1997	10395	2122.919997
02-Jan-1997	10397	716.719995
01-Jan-1997	10392	1440

### e. Sử dụng các biểu thức chọn lọc bản ghi

Nếu muốn truy vấn hiển thị theo điều kiện nào đó thì trên hàng **Criteria** đưa vào điều kiện hay biểu thức chọn lọc của các trường tại cột của trường đó

#### Lọc các bản ghi theo biểu thức chọn:

Ta có thể dùng các toán tử sau đây để lựa chọn các bản ghi.

Toán tử	Ví dụ	Ý nghĩa
<	< #1/1/1999#	Trước ngày 1/1/1999
>	> 102	Lớn hơn 102
<=	<= #1/1/1999#	Trước và trong ngày 1/1/1999
>=	>= 102	Lớn hơn hoặc bằng 102
<>	<> #1/1/1999#	Khác ngày 1-1-1999
=	= 1000; = #1/1/99#	Bằng
Between... and...	Between #1/1/199# and #1/1/2000#	Từ ngày 1-1-1999 đến ngày 1-1-2000
Like	Like "Hà nội"	Bằng Hà Nội

**Các ký tự thay thế (Wild character)**

Khi thiết kế một truy vấn, đôi khi ta muốn tìm kiếm theo một khuôn dạng linh hoạt chứ không phải cố định. Ví dụ Ta muốn tìm tất cả các khách hàng có tên bắt đầu bằng chữ “H”,... Khi đó, ta dùng ký tự thay thế với từ khoá Like hoặc not like.

Ký tự	Ý nghĩa
*	Thay thế một số bất kỳ các ký tự
?	Thay thế một ký tự tại vị trí ? xuất hiện
#	Thay thế một ký tự số tại vị trí # xuất hiện
[ ]	Thay các ký tự bất kỳ trong ngoặc vuông
!	Thay thế ký tự bất kỳ ngoại trừ ký tự sau dấu than trong ngoặc vuông
-	Từ ký tự đến ký tự

**Ví dụ:**

Biểu thức	Các giá trị có thể có
Like “Hu?ng”	Huong, Hưởng, Hường,...
Like “L*ng”	Long, Ling, Levering,...
Like “*th”	128 <sup>th</sup> , Smith, Breath,...
Like “*on*”	condinments, confection, congress,...
Like “10*/2000”	Tất cả ngày mùng 10 các tháng
Like “1#23”	1123, 1223, 1323,...
Like “Hu[ở]ng”	Huong, Hưởng, nhưng không có Hường,...
Like “Sm[!au]th”	Khác với Smath và khác Smuth
Like “X[1-9]BC”	X1BC, X2BC,...X9BC
Like “X[a-c]BC”	XaBC, XbBC, XcBC

**Tìm lọc các bản ghi có hoặc không có giá trị**

Nếu muốn chọn các bản ghi có giá trị của một trường nào đó là rỗng, thêm từ khoá IS NULL vào dòng điều kiện Criteria. Ngược lại ta dùng IS NOT NULL.

**Ví dụ:**

Trường	Ngày chuyển hàng	Ý nghĩa
Bảng	Đơn đặt hàng	
...		
Criteria	is null	Tìm những đơn đặt hàng nào vẫn chưa chuyển hàng đi.

Trường	Ngày chuyển hàng	Ý nghĩa
Bảng	Đơn đặt hàng	
...		
Criteria	is not null	Tìm những đơn đặt hàng nào đã giao xong hàng cho khách (đã hoàn thành hợp đồng).

**Lọc các bản ghi có chứa giá trị thuộc một tập hợp nào đó**

Dùng toán tử IN (danh sách các giá trị).

Trường	Ngày đặt hàng	Mã Khách hàng	ý nghĩa
Bảng	Đơn đặt hàng	Khách hàng	Liệt kê những ngày trong tháng 1 năm 2000 mà 3 khách hàng có tên là Long, Hùng hoặc là Dũng đã đến đặt hàng tại doanh nghiệp.
...	...	...	
Criteria	Like #*/01/2000#	IN (Long, Hùng, Dũng)	

**f) Các hàm thường dùng khi thiết kế một truy vấn**

**Left(chuỗi, n):** Hàm này dùng để cắt ra n ký tự kể từ phía bên trái của một chuỗi ký tự. Cú pháp: Left(chuỗi, số\_ký\_tự\_sẽ\_lấy)

**Ví dụ:** Khi trao đổi nghiệp vụ trên mạng, mỗi bưu cục được gán cho một tiêu đề tên. Tiêu đề này bao gồm hai phần, phần đầu là phần mã số bưu chính và phần tiếp là tên bưu cục. Ví dụ, trên các ấn phẩm nghiệp vụ, người ta tham chiếu đến Bưu cục Nguyễn Trãi thuộc cơ sở đào tạo Hà đông bằng tiêu đề 31103 Nguyễn Trãi. Quy tắc đánh số hiệu cho một bưu cục trên mạng là:

- Độ dài mã số tổng cộng: 5 chữ số.
- 2 ký tự đầu: Mã tỉnh.
- 3 ký tự sau: Số hiệu của bưu cục trong mạng nội tỉnh.

Vậy, để lọc ra các bưu cục thuộc Bưu điện Hà nội, có mã vùng là 10, từ trong bảng Túi gói đi đến, ta có thể thiết kế một truy vấn có cấu trúc như sau:

<b>Field</b>	=LEFT(Mã Bưu cục, 2)	Bưu cục nhận túi gói
<b>Bảng</b>	Túi gói đi đến	Túi gói đi đến
<b>Sort</b>		
<b>Show</b>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>Criteria</b>	= "10"	

**Right():** Dùng hàm Right(chuỗi, n) để trích ra n ký tự, tính từ phải sang, từ chuỗi ký tự cho trước.

Giả sử trong bảng Bưu cục có các mã bưu cục sau: 10103 A, 15201 B, 30101 C, 28201 D.

Dim sMaxBwucuj as string

sMaxBwucuj=Inputbox("Nhập vào mã bưu cục")

Msgbox "Mã số nội tỉnh của Bưu cục là " & Right(sMaxBwucuj,3)

**Mid (chuỗi, n, m):** Trích ra từ chuỗi m ký tự, kể từ vị trí n.

**Trim (chuỗi):** Loại bỏ dấu cách ở hai đầu chuỗi ký tự.

**DatePart (Datepart, DateExpr):**Thể hiện các giá trị kiểu ngày tháng. Trong đó các giá trị của Datepart là:

Datepart	Ý nghĩa	Ví dụ	Trả về
		[Ngày đặt hàng] = #19-07-2001# [Giờ đến] = 19:07:20"	
d	Ngày	DatePart("d", [Ngày đặt hàng])	19
m	Tháng	DatePart("m", [Ngày đặt hàng])	7
yyyy	Năm	DatePart("yyyy", [Ngày đặt hàng])	2001
q	Quý	DatePart("q", [Ngày đặt hàng])	3
ww	Tuần	DatePart("ww", [Ngày đặt hàng])	
w	Ngày trong tuần	DatePart("w", [Ngày đặt hàng])	
h	Giờ	DatePart("h", [Giờ đến])	19
n	Phút	DatePart("n", [Giờ đến])	7
s	Giây	DatePart("s", [Giờ đến])	20

**Sum (biểu thức):**Tính tổng của biểu thức. Biểu thức đây có thể chỉ là một trường, cũng có thể là công thức cộng trừ nhân chia của nhiều trường.

**Count:** Đếm các giá trị của một trường dữ liệu trong một bảng.

### 5.2.3.5. Biểu mẫu nhập liệu (FORM)

#### a. Khái niệm biểu mẫu

Một biểu mẫu - Form - là một giao diện dạng cửa sổ cho phép người sử dụng xây dựng nên các màn hình dùng để cập nhật hoặc xem dữ liệu lưu trong các bảng, ngoài ra cũng cho phép người sử dụng tạo ra các hộp thoại hỏi đáp giữa người sử dụng và hệ thống ứng dụng.

Việc chưa có một từ trong tiếng Việt đủ chính xác để biểu đạt hết ý nghĩa của khái niệm Form khiến các chuyên gia tin học khuyên người dùng nên sử dụng chính từ gốc không dịch. Kể từ đây trở đi, từ Form sẽ được sử dụng vì trong môi trường Access, tất cả các từ khoá đều bằng tiếng Anh; việc dùng từ gốc giúp người dùng làm quen với chương trình nhanh hơn là việc dùng từ Việt hoá do tránh được việc phải dịch xuôi rồi lại dịch ngược.

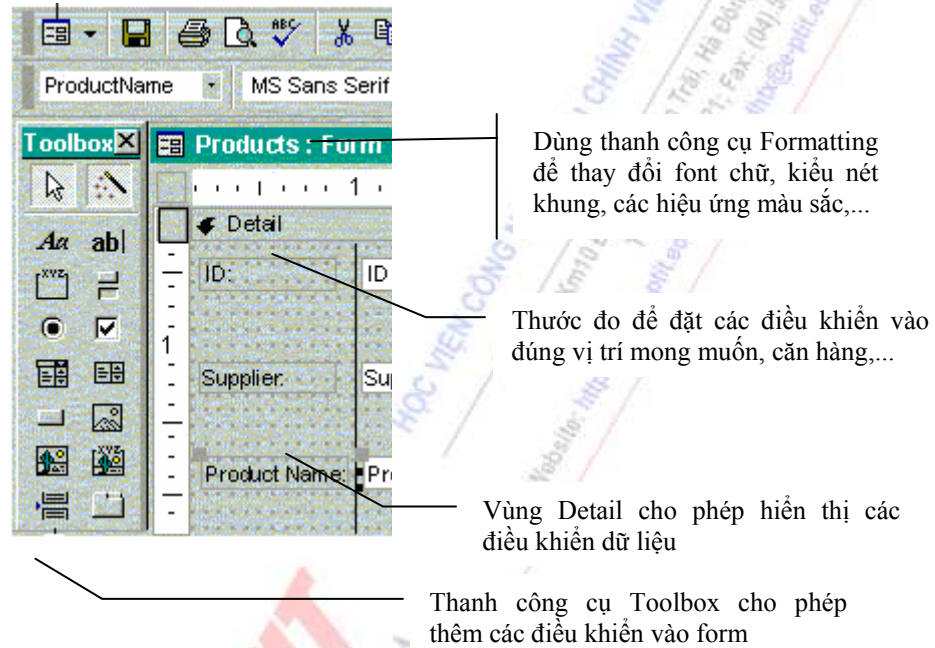
Mối liên kết giữa form và nguồn dữ liệu của nó được thể hiện bằng các đối tượng có giao

diện đồ họa, được gọi là điều khiển. Kiểu điều khiển được sử dụng thông thường nhất để hiển thị và nhập liệu là hộp văn bản.

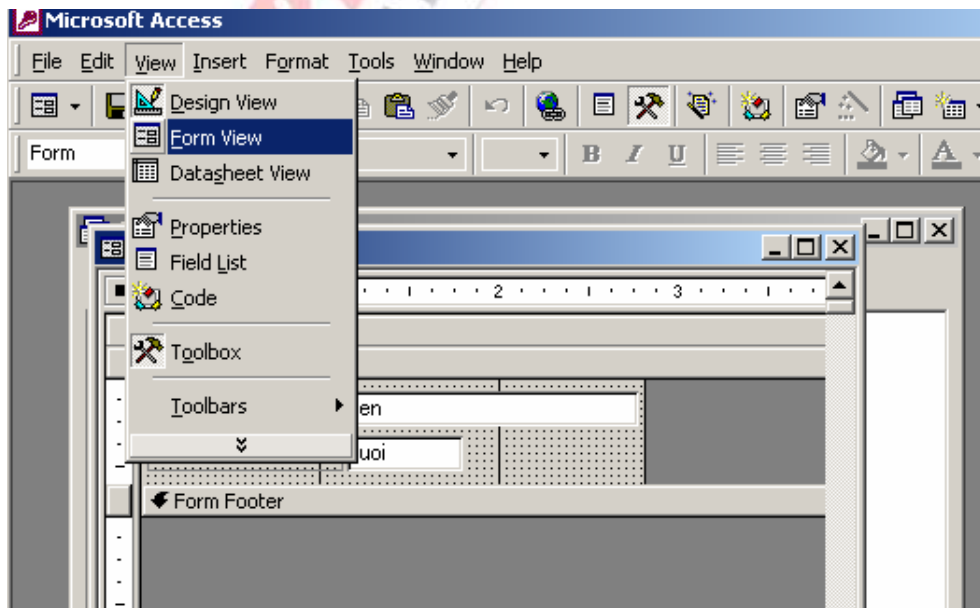
### Các khung nhìn của form

Một Form có ba khung nhìn: Design view - thiết kế, Form view - dạng cửa sổ và Datasheet view - dạng lưới dữ liệu.

Ban đầu, lúc thiết kế form ta dùng chế độ thiết kế (Design).



Thiết kế xong form dữ liệu, chuyển sang dạng Form view hoặc Datasheet dùng hiển thị dữ liệu.



Các khu vực trên một form

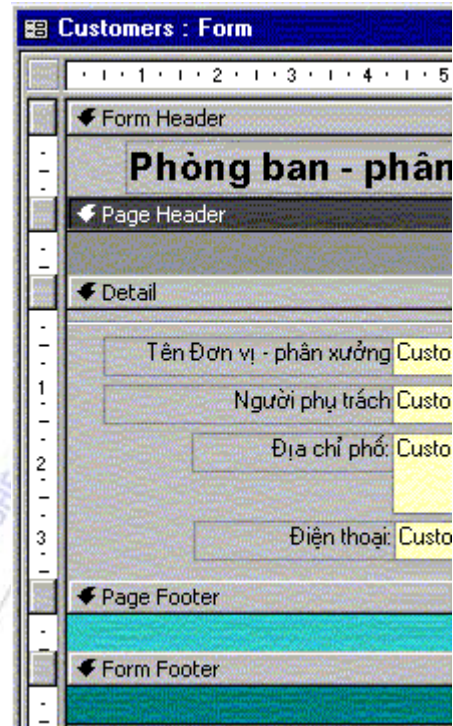
Form Header: Khu vực trên cùng của một Form. Thông thường người ta sử dụng khu vực này làm nơi hiển thị dòng tiêu đề của Form.

Form Footer: Khu vực nằm ngang dưới cùng của Form. Khu vực này thường chứa các nút lệnh hành động như In, Lọc dữ liệu, Tìm kiếm, Mở một Form mới...

Detail: Phần chi tiết của form. Nội dung của tất cả các trường (field) của một bảng dữ liệu (table) sẽ được hiển thị ở đây. Các đối tượng khác như hình vẽ, bảng tính, đối tượng liên kết OLE cũng được hiển thị ở phần này. Người ta gọi chung tất cả các đối tượng này là các điều khiển - control.

Page Header: Khi sử dụng ngắt trang - Page break - trong một form, ta sử dụng khu vực này để hiển thị một thanh tiêu đề như nhau cho tất cả các trang của một form.

Page Footer: Khi sử dụng ngắt trang - Page break - trong một form, ta sử dụng khu vực này để duy trì sự hiển thị nhất quán cho tất cả các trang của một form.




## b. Thiết kế form: Trong Access có 2 cách chính để thiết kế Form

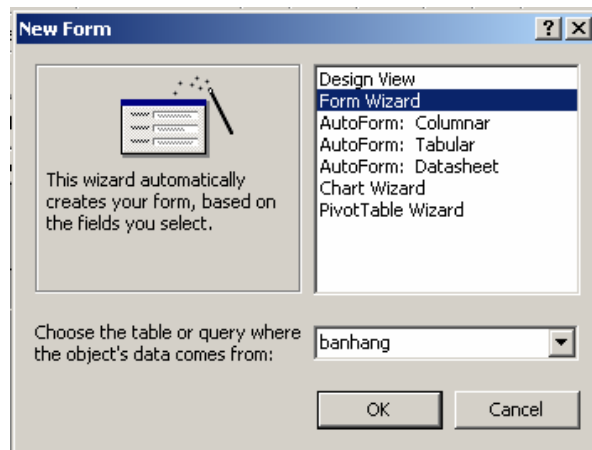
- *Tạo Form dùng Form Wizard*: Đây là cách nhanh chóng để có thể tạo được một form nhập dữ liệu cho các bảng dữ liệu trên một CSDL. Cách này dễ làm nhưng kết quả của nó không được tốt lắm, thường thì phải chỉnh sửa nhiều thì mới được Form theo như ý muốn của người sử dụng

- *Thiết kế Form dùng Design View*: Đây là cách tạo Form chuyên nghiệp. Cách này có nhiều bước phức tạp đòi hỏi nhiều đến kinh nghiệm và trình độ của người lập trình. Nhưng kết quả làm ra luôn đáp ứng được các yêu cầu khắt khe nhất của người sử dụng

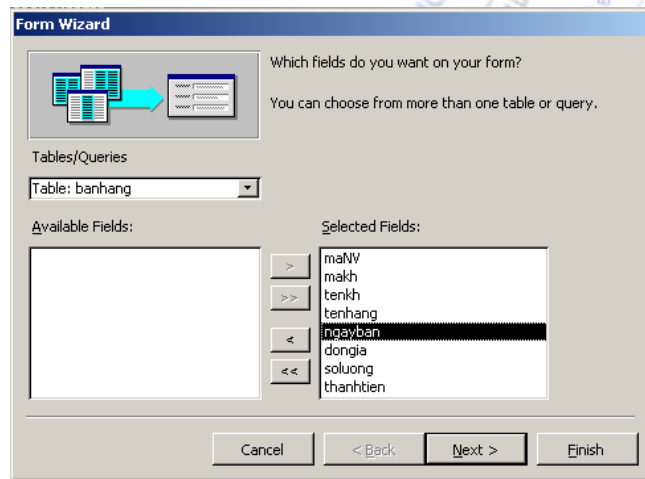
### Tạo mới form bằng wizard

1. Trong cửa sổ Database, chọn Forms .
2. Nhấn New.
3. Trong hộp thoại New Form, chọn wizard muốn dùng. Mô tả ngắn gọn về wizard được thể hiện ở bên trái của hộp thoại.

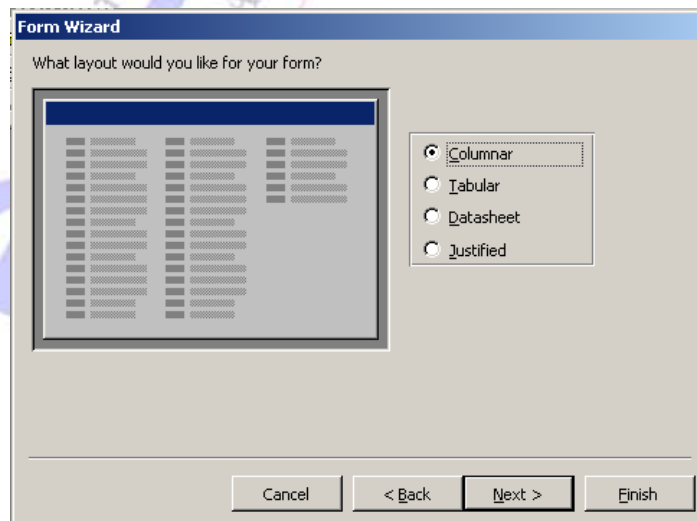




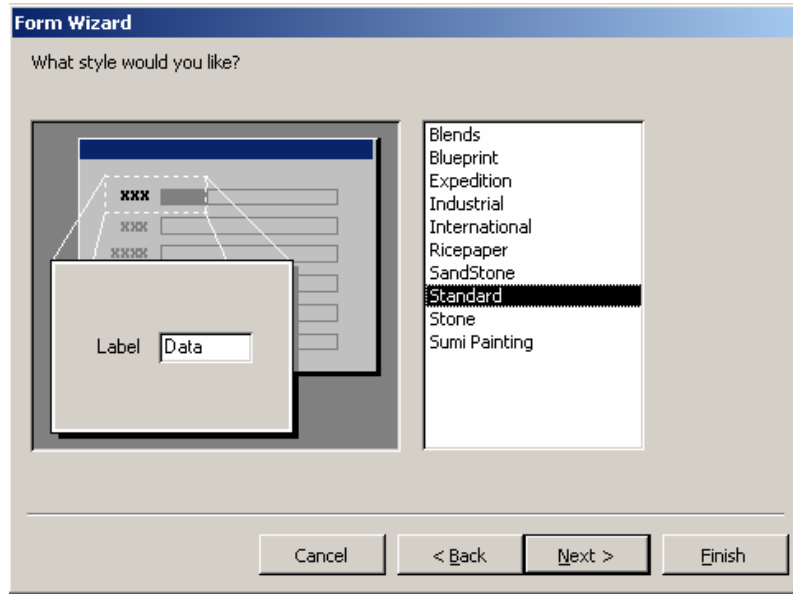
4. Chọn tên bảng cần tạo Form nhập dữ liệu ở mục *Choose the table or query...*, chọn OK. Sau đó Add (thêm) các trường cần nhập dữ liệu lên form từ danh sách *Available Fields*: bằng nút:



5. Chọn kiểu canh lề các đối tượng trên form: chúng ta có thể chọn một trong các kiểu ở dưới, nhưng tốt nhất hãy dùng kiểu *Columnar*:



6. Chọn kiểu cho các đối tượng trên form (nên chọn kiểu Standard là hợp lý nhất)



7. Chọn Finish để kết thúc quá trình tạo form:

8. Nếu ta chọn hoặc Form Wizard, Chart Wizard, hoặc PivotTable Wizard trong bước 3, đi theo các bước mà hộp thoại wizard yêu cầu. Nếu chọn AutoForm: Columnar, AutoForm: Tabular, hoặc AutoForm: Datasheet, Access sẽ tự động tạo ra form.

9. Nhấn chuột vào phím View button để xem form vừa tạo.

**Thiết kế form theo nhu cầu (Tạo Form ở chế độ Design View)**

Trước khi thực hiện thiết kế ta cũng phải thực hiện bước 1 và bước 2 ở phần “ tạo mới form bằng wizard” sau đó lựa chọn chế độ Design View rồi sử dụng các phím chức năng, hộp công cụ để tạo Form :

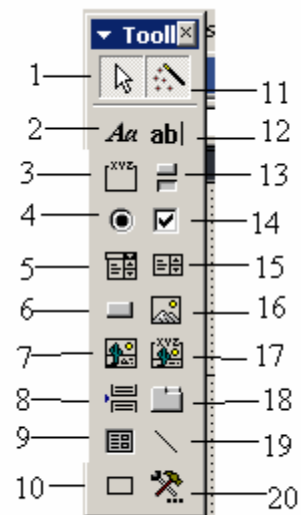
**Hộp công cụ tạo Form (ToolBox)**

Để hiển thị hộp công cụ vào View chọn Toolbox.

Hộp công cụ có dạng như sau :

Ý nghĩa các nút:

1. Select Objects: Chọn đối tượng
2. Label: Dùng để tạo một nhãn để chứa văn bản hướng dẫn hoặc tính toán
3. Option Group: Dùng để tạo một nhóm tự động
4. Option Button: Dùng để tạo một nút lựa chọn
5. Combo box: Dùng để tạo một danh sách lựa chọn
6. Command Button: Dùng để tạo nút lệnh để khi kích vào sẽ hoàn thành một nhiệm vụ



7. Unbound Objects Frame: Bổ sung vào biểu mẫu một đối tượng OLE
8. Page Break: Cho máy in biết bắt đầu một trang mới
9. Subform/Subreport: Bổ sung một biểu mẫu phụ vào biểu mẫu chính
10. Rectangle: Tạo một hình chữ nhật
11. Control Wizard: Bật tắt chế độ Control Wizard
12. Text Box: Tạo một hộp văn bản
13. Toggle Button: Tạo một nút thay đổi từ On sang Off
14. Check Box: Tạo một hộp kiểm tra
15. List Box: Tạo một hộp liệt kê qua đó có thể chọn lựa một giá trị
16. Image: Hiển thị đồ họa trên form
17. Bound Objects Frame: Hiển thị nội dung một trường OLE
18. Tab Control: Chèn một điều khiển Tab
19. Line: Vẽ một đường thẳng
20. More Controls: Kích vào nút này sẽ mở ra danh sách các điều khiển khác có thể được thêm vào hệ thống

**Thiết kế mẫu biểu bằng hộp công cụ:**

❖ - Sử dụng Textbox hoặc Fields list để thiết kế. Ta thực hiện các bước sau:

- Tại cửa sổ Database: Ta chọn Form và New
- Chọn Design View: Chọn bảng nguồn (ví dụ bảng: banhang) rồi chọn **OK**
- Tạo tiêu đề form

**Chọn View/Form Head/Footer**

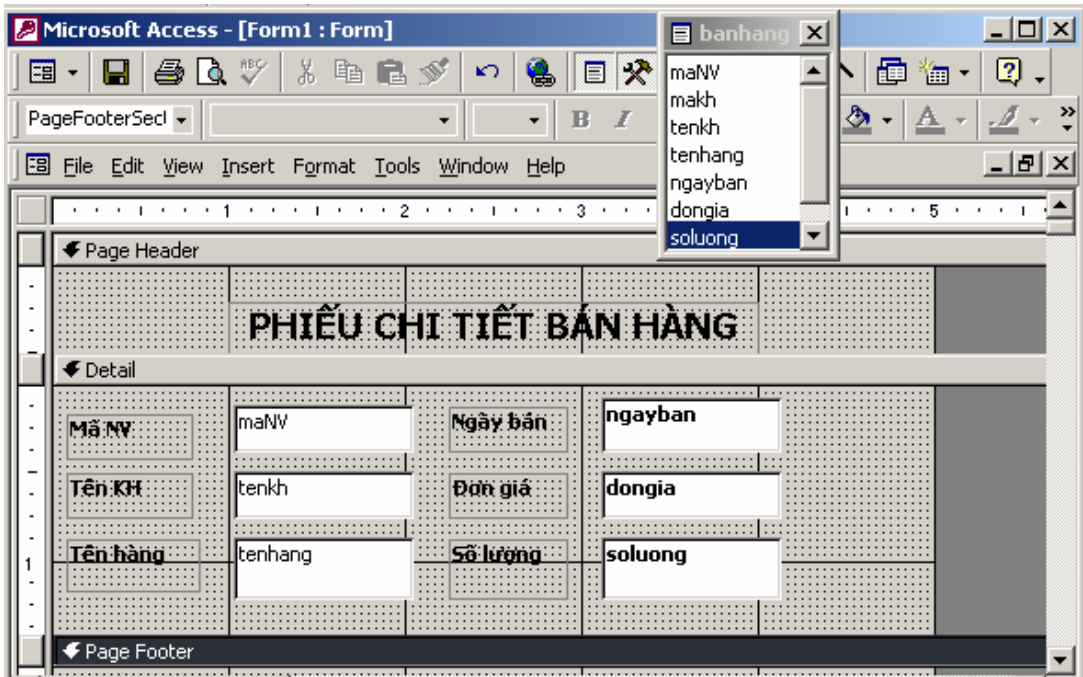
Kích chuột vào nút (lable) trên hộp công cụ (Tool Box) và đưa chuột ra vùng **Form Head**, gõ tiêu đề ( ví dụ : **Phiếu chi tiết bán hàng**)

- Thiết kế các trường

Kích vào hộp **Field List** trên thanh công cụ

Kéo các trường cần thiết từ hộp thoại vào phần **Detail** của form

- Cửa sổ thiết kế form như sau:



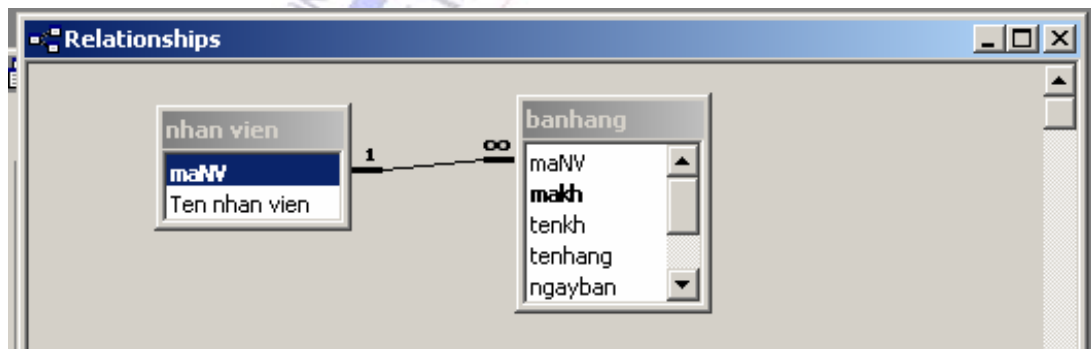
- Chọn **File/Save** để lưu biểu mẫu này

- Đóng cửa sổ thiết kế lại. Nếu muốn chạy 1 Form vừa tạo thì từ cửa sổ **Database** chọn tên Form, rồi chọn **Open** .

❖ **Sử dụng Combo Box và check Box :**

Như đã giới thiệu ở trên. Hộp **Combo Box** được dùng để tạo danh sách lựa chọn cho một dữ liệu cho 1 trường bất kỳ. Hộp **CheckBox** được dùng để chọn lựa giá trị Yes hoặc No cho trường có kiểu logic.

Giả sử chúng ta có CSDL sau:



Giả sử bảng nguồn: **Nhan vien** đã có dữ liệu

Yêu cầu:

Tạo một form cho phép nhập danh sách bán hàng như sau:

Khi nhập dữ liệu thì trường MãNV được phép lựa chọn giá trị từ một danh sách có sẵn.

#### Trình tự thiết kế như sau:

Tạo form mới, dùng bảng **banhang** làm nguồn dữ liệu cho Form

Tắt chức năng **Control Wizard** trong hộp Tool Box

Kéo các trường (không phải trường MãNV) từ hộp **Field List** đặt vào một vị trí trên form

Tạo **ComboBox** để nhập mã nhân viên cho trường MãNV bằng cách lựa chọn giá trị từ danh sách đã có từ bảng **nhanvien**:

- Chọn nút Combo Box trên thanh công cụ.

- Kéo trường MãNV từ hộp Field List đặt vào vị trí trên form. Kết quả Access tạo một combo box thuộc vào trường MãNV

- Mở bảng thuộc tính của ComboBox vừa tạo cho trường MãNV (click chuột phải vào ComboBox vừa tạo). Thiết lập một số thuộc tính như sau

**Control Source:** MãNV

**Row source type:** Table/Query

**Row source:** Nhan vien(tên bảng/truy vấn cần lấy dữ liệu)

**Column Count:** 2 (số cột trong bảng/ truy xuất được hiển thị tính từ trái sang phải)

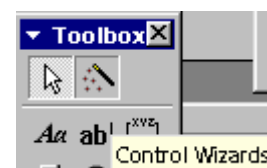
**Bound Column:** 1 (cột được chọn để nhận dữ liệu, ở đây ta chọn cột MãNV, các cột khác đóng vai trò hiển thị).

#### ❖ Sử dụng command button:

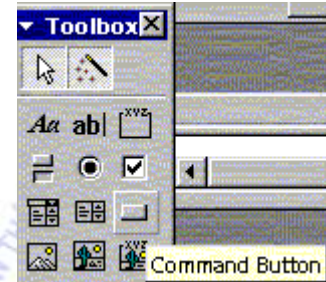
Như giới thiệu ở trên Command Button để tạo nút lệnh mà khi kích vào sẽ hoàn thành một nhiệm vụ. Cụ thể có thể sử dụng nút lệnh đó để chạy 1 query hoặc 1 form bất kỳ

#### Sử dụng Nút lệnh như sau:

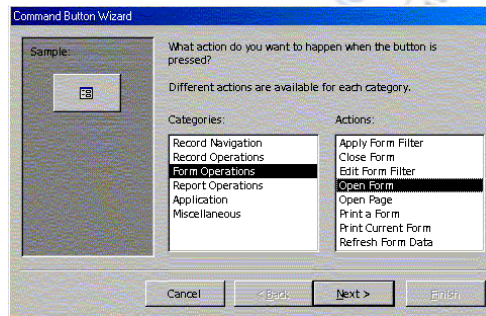
- Nhấn chuột chọn nút lệnh Control Wizard trên thanh công cụ Toolbox.



- Chọn nút Command Button trên thanh công cụ Toolbox, nhấn trở chuột vào vị trí nơi nút lệnh sẽ được định vị.



- Cửa sổ Command Button Wizard hiện ra. Trong cửa sổ này, ô Categories phân loại các nhóm hành động, ô Action liệt kê các hành động thuộc mỗi một nhóm hành động ở bên ô Categories. Ví dụ: nếu muốn nút lệnh này mở một Form (hoặc query) đã có thì ta chọn Form Operations (Miscellaneous) ở bên ô Categories, tiếp đó chọn hành động Open Form (run query) ở bên ô Action. Nhấn Next để tiếp tục. Tương tự ta có thể chọn Close Form nếu muốn đóng Form



- Chương trình hỏi: “Bạn muốn nút lệnh mở Form (hoặc query) nào?”, ta chọn Form (query) cần mở bằng cách nhấn chuột vào tên Form (query) trong ô danh sách các Form (query). Nhấn Next để tiếp tục.

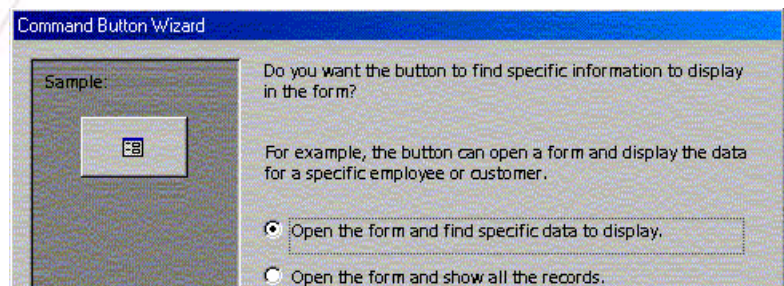


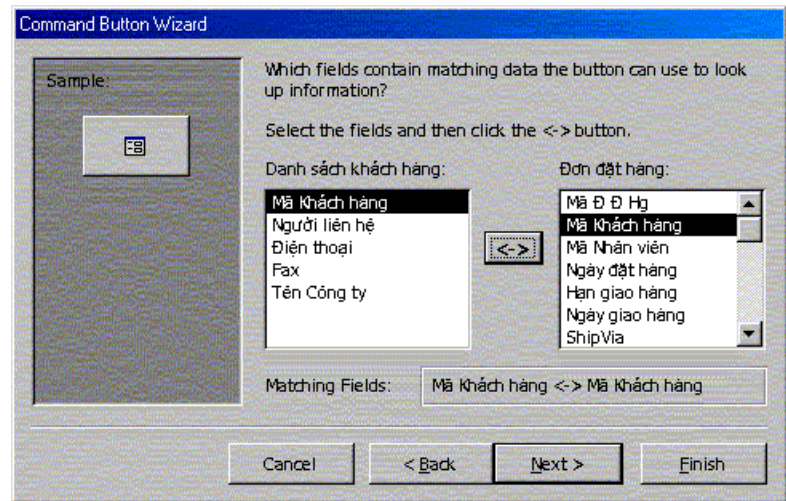
- Chọn một trong hai lựa chọn.

**Lựa chọn ←:** mở Form (query) và hiển thị chỉ một số bản ghi cụ thể;

**Lựa chọn ↑:** hiển thị tất cả các bản ghi.

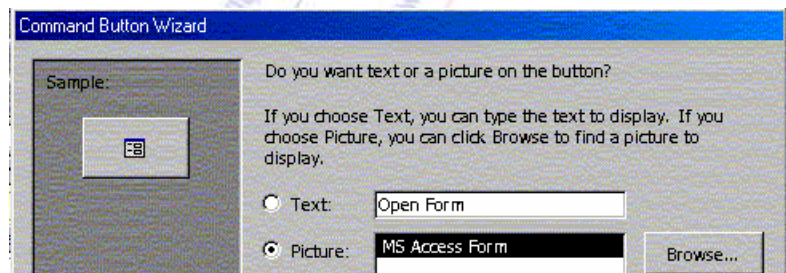
Nhấn Next để tiếp tục



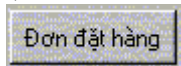


- Tại bước cuối cùng, ta có thể chọn nút lệnh có hình như (Picture) hoặc có lời chú dẫn (Text).

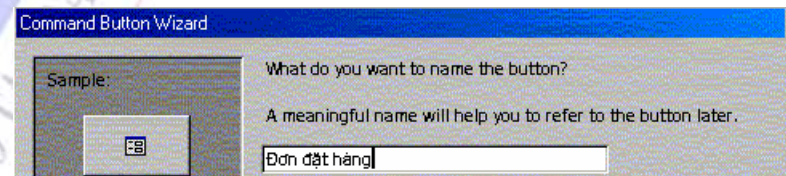
Nếu chọn Picture, chỉ ra hình như muốn có. Nút lệnh sẽ có dạng



Nếu chọn Text, đánh nhân cho nút vào dòng trống (vd là đơn đặt hàng). Nút lệnh sẽ có dạng



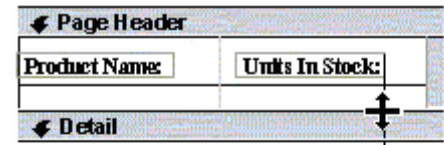
Bước tiếp theo, ta đặt tên cho nút lệnh (ví dụ đơn đặt hàng) rồi chọn Finish. Chuyển sang chế độ View để xem kết quả



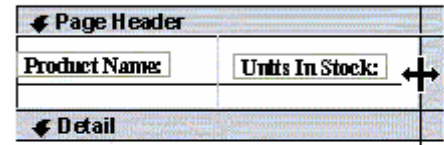
### Thay đổi kích thước các khu vực trên một Form

Có thể tăng hoặc giảm chiều cao, độ rộng từng vùng của form và report. Khi thay đổi độ rộng của một vùng, ta thay đổi độ rộng của toàn bộ form hoặc report.

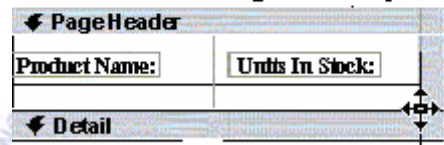
1. Mở form (hoặc report) ở chế độ Design view.
2. Thay đổi chiều cao hoặc độ rộng bằng cách đặt con trỏ vào lề dưới hoặc lề phải của vùng.
3. Nhấn giữ và kéo chuột lên xuống để thay đổi độ cao, trái - phải để thay đổi độ rộng. Để thay đổi cả độ rộng và chiều cao, đặt con trỏ chuột vào góc dưới bên phải và kéo theo đường chéo theo hướng mở rộng.



Kéo lên con trỏ lên hoặc xuống



Kéo kê con trỏ sang trái hoặc phải

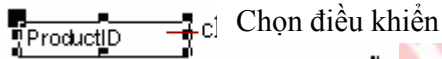


Kéo lên con trỏ chéo theo bất cứ hướng nào

### Chỉnh sửa các điều khiển trên form

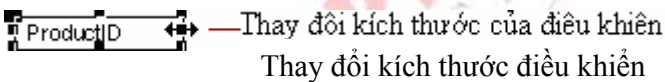
Ta có thể di chuyển, đổi kích thước, căn hàng các điều khiển trên form.

1. Để chọn một điều khiển: click chuột vào điều khiển đó. Nếu muốn chọn nhiều điều khiển cùng một lúc thì đồng thời giữ phím SHIFT và click chuột vào các điều khiển.



Chọn điều khiển

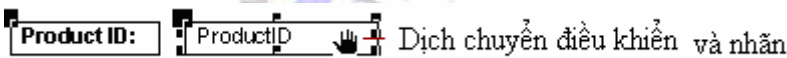
2. Khi ta chọn một điều khiển, dùng con chuột kéo rê khung của điều khiển để thay đổi kích thước của nó



Thay đổi kích thước của điều khiển

Thay đổi kích thước điều khiển

3. Di chuột để hiện ra hình bàn tay, giữ và rê chuột để chuyển vị trí cho các điều khiển.



Di chuyển điều khiển và nhấn

4. Nếu muốn chỉ di chuyển vị trí của một trong hai, điều khiển hoặc nhãn, di chuột vào đúng ô vuông góc trên bên trái của điều khiển, hoặc nhãn đó, bấm và giữ chuột để di chuyển đến chỗ mới




Chỉ di chuyển điều khiển hoặc nhãn

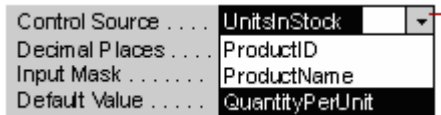
5. Để căn hàng các điều khiển, chọn chúng rồi dùng lệnh \Format\Alignment, chọn tiếp Right (trái), Left (phải), Top (trên), Bottom (dưới).

6. Để thay đổi dòng chữ của nhãn, chọn dòng chữ cũ, đánh vào dòng chữ mới.




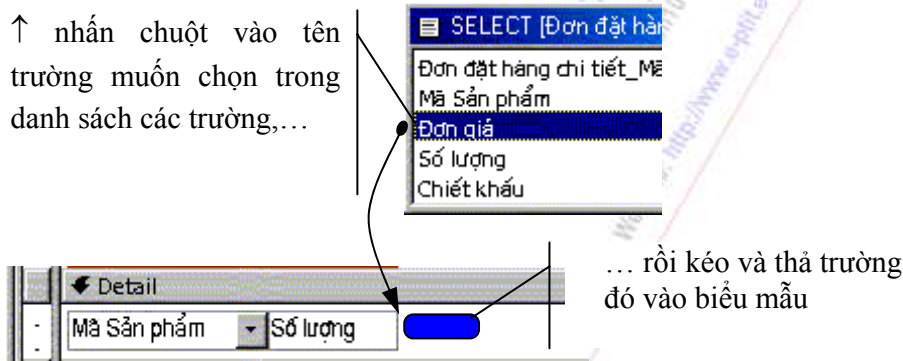


7. Để thay đổi tên trường dữ liệu gắn kết với điều khiển, trước hết chọn điều khiển đó, bấm  trên thanh công cụ, khi cửa sổ Properties hiện ra, trong phần Control Source chỉ ra tên của trường dữ liệu muốn gắn kết.





8. Thêm vào một hộp văn bản hiển thị dữ liệu của một trường:

-: Nhấn vào nút lệnh Field List  trên thanh công cụ để hiển thị danh sách các trường dữ liệu hiện có.



9. Thêm một nhãn văn bản vào Form:

Nhấn vào biểu tượng  trên thanh công cụ Toolbox, nhấn tiếp vào vị trí cần hiển thị trên form, sau đó đánh vào đoạn văn bản và bấm ENTER để kết thúc.

10. Muốn dùng Wizard để tạo các điều khiển trên form, luôn giữ nút lệnh  trên thanh công cụ Toolbox ở dạng được chọn, rồi chọn kiểu điều khiển mà ta muốn có.

### 5.2.3.6. Báo biểu (Report)

#### a. Khái niệm báo biểu

Một báo biểu là một cách trình bày hiệu quả dữ liệu dưới dạng tài liệu in. Ta có thể chỉnh sửa kích thước và hình thức hiển thị của mọi đối tượng trên một báo biểu, cho nên ta có thể hiển thị dữ liệu theo cách mà ta muốn.

Phần lớn các thông tin có trên một báo biểu được tập hợp từ các bảng dữ liệu, các truy vấn. Các thông tin còn lại có được khi thiết kế báo biểu.

#### Các khu vực của một báo biểu

Khu vực tiêu đề trên của báo biểu - Report header - chỉ xuất hiện một lần ngay ở dòng đầu tiên của trang thứ nhất. Ta có thể dùng khu vực này để hiển thị tiêu đề của báo biểu, logo, ngày giờ in báo biểu,...

Khu vực tiêu đề trên của một trang - Page header - xuất hiện ở đầu tất cả các trang của báo biểu (Chú ý rằng một báo biểu có thể có nhiều trang). Ta dùng khu vực này để hiển thị tiêu đề của cột dữ liệu.

Khu vực chi tiết - Detail - là khu vực chính của báo biểu, nó hiển thị từng bản ghi của nguồn dữ liệu của báo cáo.

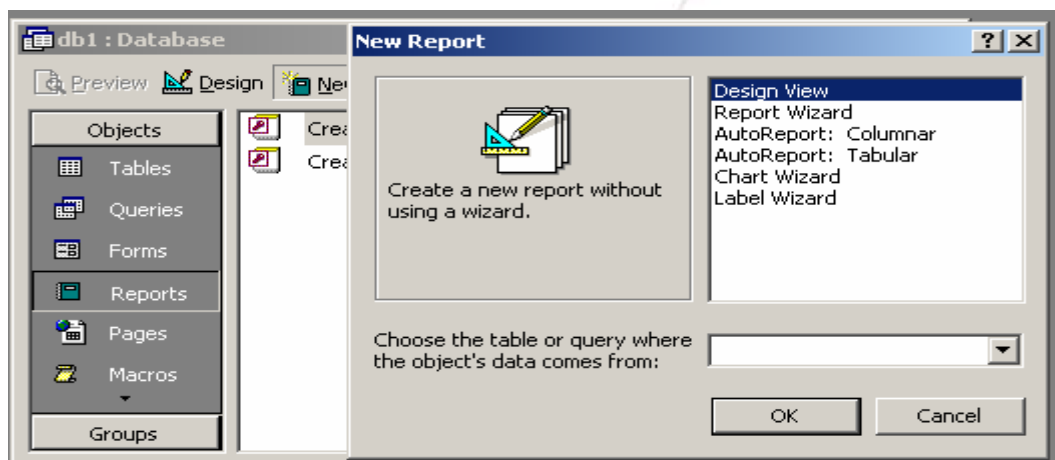
Khu vực cuối trang - Page footer - xuất hiện ở cuối tất cả các trang của báo biểu. Ta dùng khu vực này để hiển thị số thứ tự từng trang của báo biểu.

Khu vực cuối báo biểu - Report footer - xuất hiện ở dòng cuối cùng của trang cuối cùng của báo biểu. Trong khu vực này, người ta thường hiển thị con số tổng kết của các báo biểu, (ví dụ tổng số khách hàng, tổng tiền hàng).

Ngoài các khu vực trên, ta có thể thiết kế thêm tiêu đề trên cho từng nhóm các bản ghi - Group header. Ví dụ, khi in ra danh sách các hợp đồng xuất hàng trong tháng, ta muốn nhóm các hợp đồng này lại theo từng khách hàng. Khi đó, trường Khách hàng sẽ đóng vai trò tiêu đề của nhóm các bản ghi. Tổng doanh số các hợp đồng của khách hàng này sẽ được đưa vào phần tiêu đề dưới - group footer - của trường Khách hàng.

### b. Tạo mới một báo biểu

Trước hết, ta dùng chuột chọn mục Report trong cột Objects ở cửa sổ Database. Tiếp đó, nhấn chuột vào New để tạo mới một biểu mẫu.



### Các cách tạo mới một báo biểu

Có ba cách để chúng ta tạo mới một báo biểu.

- **Sử dụng AutoReport:** Với AutoReport, ta chọn một nguồn dữ liệu và tiếp đến chỉ ra cách bố trí dạng cột (columnar), hay dạng bảng (tabular), AutoReport sử dụng toàn bộ các trường trong nguồn dữ liệu và tự động định dạng (autoformat) báo biểu.

- **Sử dụng Wizard:** Wizard là một chương trình chức năng mà khi thực hiện sẽ hỏi chi tiết về nguồn dữ liệu, về các trường, về cách bố trí xếp đặt, định dạng mà người thiết kế mong muốn, từ đó sẽ tạo ra báo biểu.

### - Tự thiết kế lấy

### Chuyển đổi giữa các khung nhìn của báo biểu

Một Báo biểu có ba khung nhìn khác nhau:

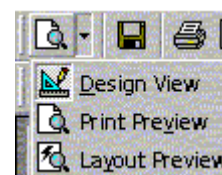
- Design view - Thiết kế: Dùng khi ta phải chỉnh sửa, thay đổi cấu trúc của một báo biểu hiện tại.

- Print Preview - Hiện thị trước khi in: dùng để xem dữ liệu hiển thị trên trang in trước khi in.


- Layout Preview - Hiện thị cách bố trí: Tương tự như Print Preview, nhưng trong cách hiển thị này, Access chỉ cho hiển thị mẫu một vài bản ghi. Hiếm khi ta dùng chức năng này, nó chỉ hữu dụng khi mà nguồn dữ liệu của báo biểu là quá lớn, nếu ta dùng Print Preview thì sẽ mất rất nhiều thời gian, nhưng nếu dùng Layout Preview thì thời gian hiển thị gần như tức thời, bởi vì lúc này Access chỉ chọn ra một vài bản ghi làm mẫu.

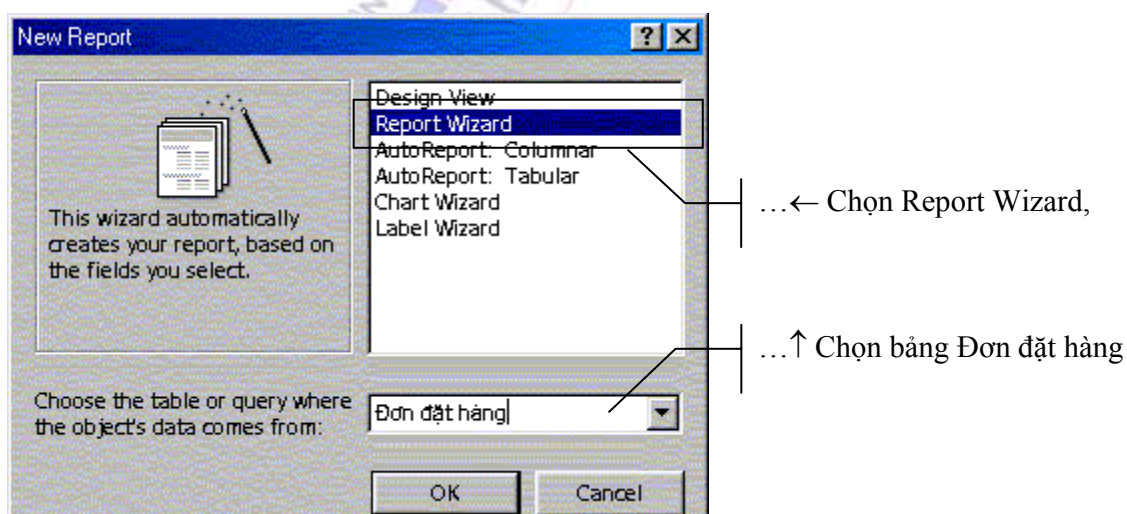
Cách chuyển đổi giữa các khung nhìn:

1. Mở báo biểu đã có.
2. Nhấn chuột vào nút lệnh View trên thanh công cụ, chọn khung nhìn cần thiết từ danh sách.

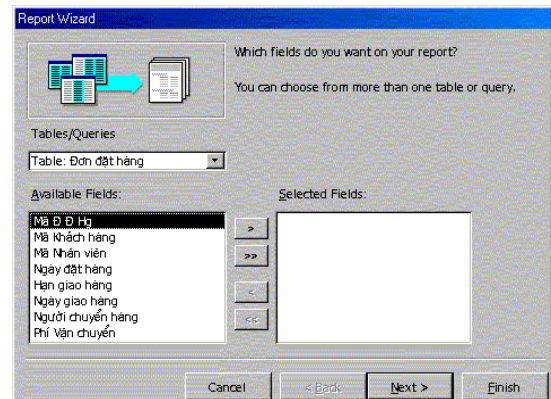


### Sử dụng Wizard để tạo mới một báo biểu

Nhấn chuột vào nút lệnh  trên cửa sổ Database. Cửa sổ dưới đây hiện ra, ...

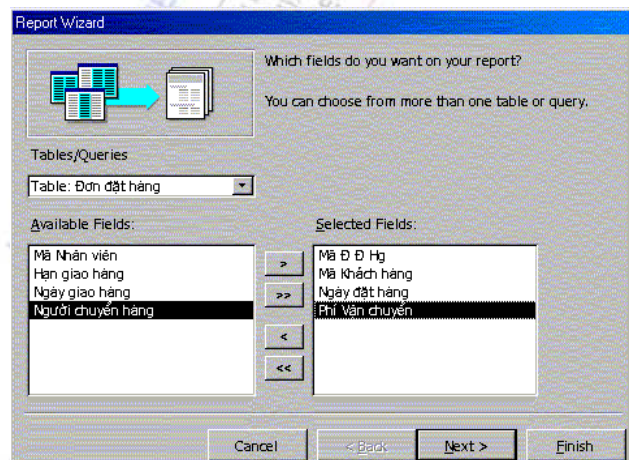


Cửa sổ Report Wizard hiện ra như hình bên. Cửa sổ gồm 2 ô chính. Ô bên trái chứa danh sách các trường của bảng dữ liệu hiện thời. Ô bên phải hiển thị danh sách các trường mà ta đã chọn để hiển thị trên báo biểu mới. Nhấn đúp vào tên trường ở ô bên trái để chọn trường sẽ được hiển thị trên báo biểu report.

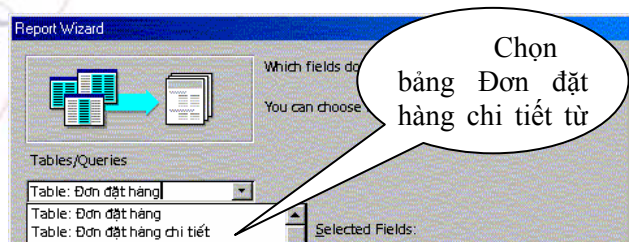


Ở đây ta chọn bốn trường: MãĐDHg, Mã Khách hàng, Ngày đặt hàng, Phí Vận chuyển.

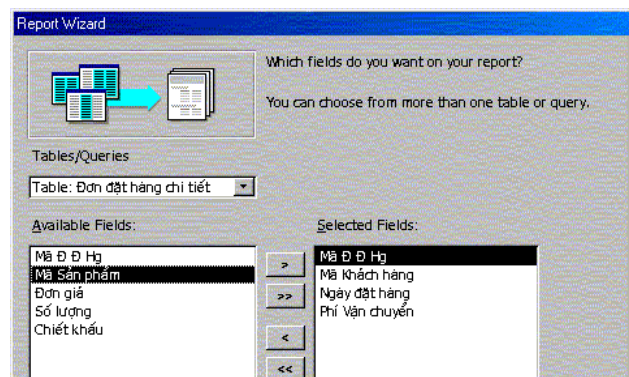
Ngoài 4 thông tin rất quan trọng này của một đơn đặt hàng, ta cũng cần phi biết chi tiết về các sản phẩm được bán trong hợp đồng: số lượng, khối lượng, đơn giá, chiết khấu. Các thông tin này được lưu ở bảng đơn đặt hàng chi tiết.




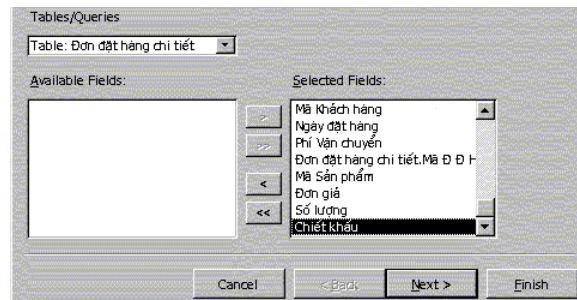
Do đó ta phải chọn ra các trường này từ bảng Đơn đặt hàng chi tiết, bằng cách chọn bảng Đơn đặt hàng chi tiết trong dòng Tables/Queries



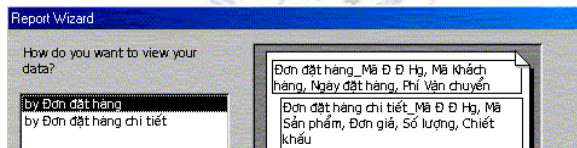
Lúc này Access chuyển sang liệt kê các trường của bảng Đơn đặt hàng chi tiết ở ô bên trái. Các trường ta đã chọn từ bảng Đơn đặt hàng trước đó vẫn còn ở ô bên phải.



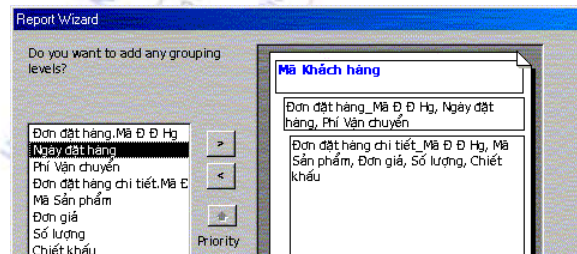
Nhấn vào nút lệnh  để chọn tất cả các trường của bảng đơn đặt hàng chi tiết. Nhấn Next để tiếp tục



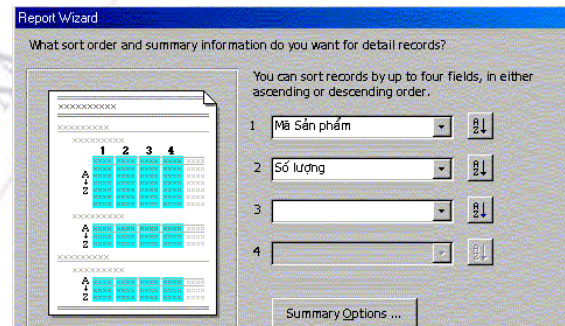
Access tiếp đó hỏi ta đâu là bảng dữ liệu chính? Ta chọn bảng đơn đặt hàng. Nhấn Next để tiếp tục.



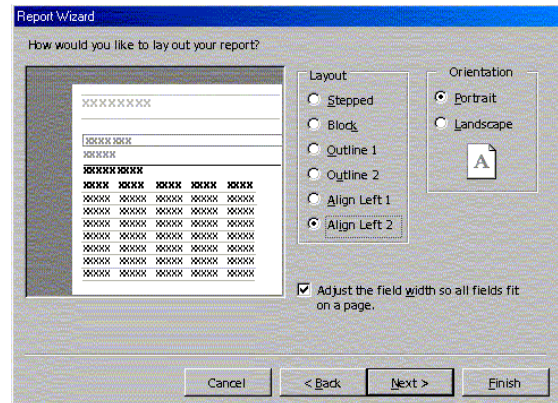
Tại bước tiếp theo, ta chọn trường Mã Khách hàng làm giá trị để tập hợp các bản ghi lại với nhau (tất cả các hợp đồng của một khách hàng sẽ được tập hợp lại cạnh nhau thành một nhóm). Nhấn chuột vào Next để tiếp tục.



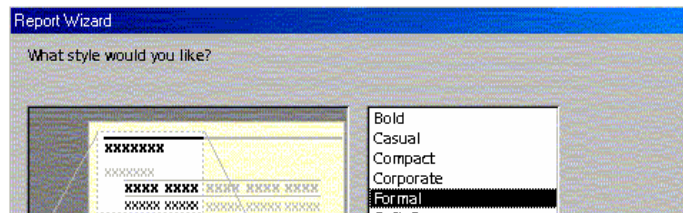
Trong cửa sổ ở bước này, ta chỉ ra báo biểu sẽ sắp xếp dữ liệu theo trường nào. Ta có thể chọn một hoặc nhiều trường, hoặc không chọn trường nào. Nếu ta muốn cứ ở cuối mỗi cột có tổng số, chọn mục Summary Options. Nhấn Next để tiếp tục.



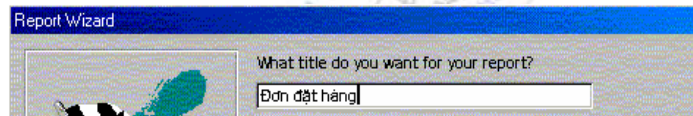
Layout là cách bố trí, bài xấp của một đối tượng. Trong Access, người ta đã thiết lập sẵn một số dạng bố trí. Ta có thể chọn bất kỳ dạng bố trí nào bằng cách nhấn chuột vào ô Layout. Ta cũng có thể chọn hướng in của báo cáo theo chiều đứng (Portrait) hay chiều nằm ngang (Landscape) tại mục Orientation. Nhấn Next để tiếp tục.



Chọn kiểu - style của báo biểu.

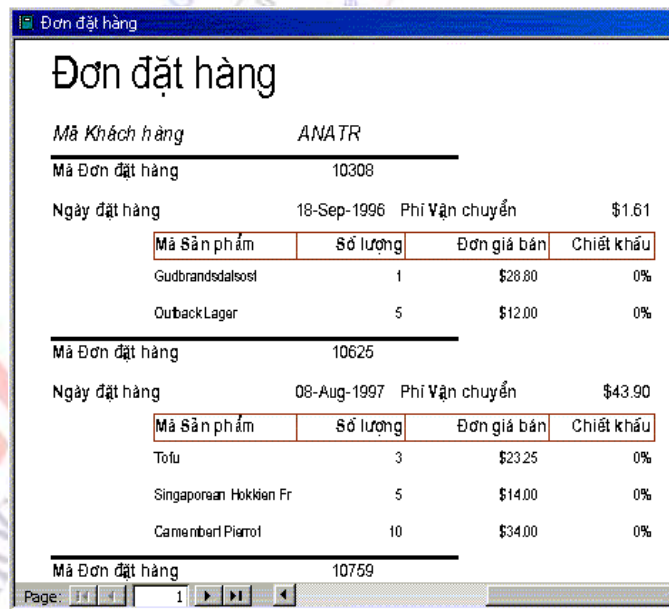


Cuối cùng, đặt tên cho báo biểu là đơn đặt hàng, nhấn Finish để kết thúc quá trình tạo báo biểu.

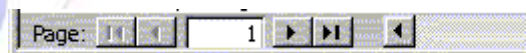


Sau khi nhấn nút lệnh Finish để kết thúc quá trình tạo lập báo biểu report bằng công cụ tự động, Access hiển thị một báo biểu như hình bên.

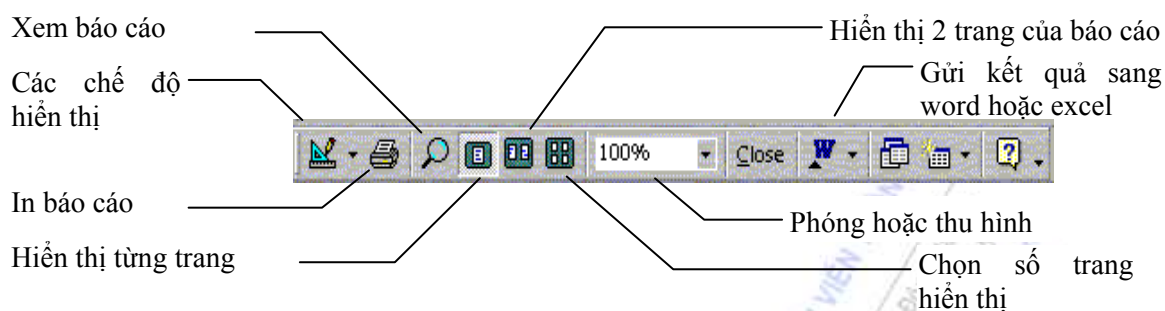
Trong biểu này, ta thấy, với mỗi một khách hàng, tất cả các hợp đồng của khách hàng đó với công ty được liệt kê lần lượt, theo thứ tự về ngày tháng ký hợp đồng.



Vì có rất nhiều khách hàng, mỗi khách hàng lại có nhiều hợp đồng cho nên một trang báo biểu không thể chứa hết, báo biểu này sẽ phải kéo dài nhiều trang. Để xem các trang tiếp theo, ta sử dụng thanh di chuyển của báo biểu, nằm ở góc dưới bên trái mỗi báo biểu.



Với mỗi báo biểu ta đều có một thanh công cụ. Ta có thể thiết kế riêng cho mỗi báo biểu một thanh công cụ khác nhau. Dưới đây là thanh công cụ ngầm định cho mỗi báo biểu.



### Thiết lập định dạng trang in cho báo biểu

Khi thiết kế các báo biểu, hoặc ngay trước khi in chúng, ta phải định dạng trang in của báo biểu cho phù hợp với các máy in mà ta hiện có. Cách thức làm như sau:

1. Trên cửa sổ các đối tượng của cơ sở dữ liệu, nhấn đúp chuột vào tên báo biểu để mở báo biểu đó.
2. Trên menu File, chọn Page Setup (Thiết lập trang giấy).
3. Nhấn vào một trong các nhãn sau để thiết lập các thuộc tính
  - *Margins*. Điều chỉnh độ rộng các lề và kiểm soát in.
  - *Page*. Điều chỉnh hướng in, kích thước giấy, chọn máy in. Thông thường ta chọn khổ A4 (khổ phổ dụng nhất), hướng giấy dọc (Portrait) hoặc ngang (Landscape).
  - *Columns*. Thiết lập số cột, kích thước và cách bố trí của các cột.
4. Nhấn OK.


### Mở xem và in một báo biểu đã có

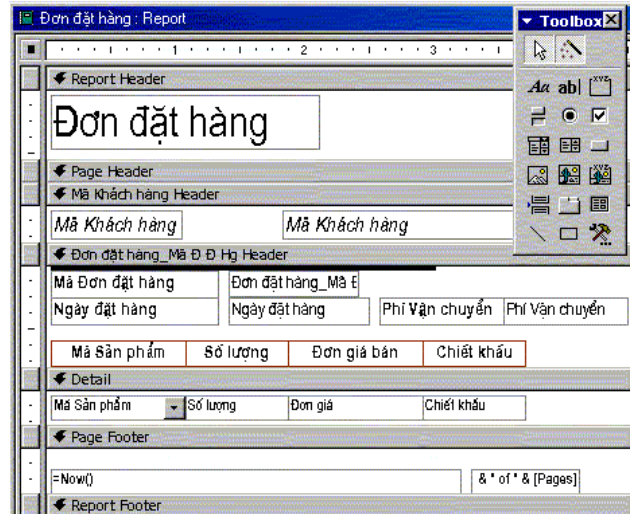
1. Trong cửa sổ Database window, nhấn chuột vào Reports trong cột Objects.
2. Nhấn chuột vào báo biểu cần mở.
3. Trên thanh công cụ của cửa sổ Database window, nhấn chuột vào nút Design để thiết kế báo biểu hiện tại, hoặc vào nút Preview để hiển thị báo biểu trước khi in.

### Hiệu chỉnh một báo biểu

Việc dùng quá trình tạo báo biểu tự động đã có thể cho phép chúng ta có được một báo biểu đẹp dựa trên các thông tin có sẵn. Tuy nhiên, để có được các kết quả xử lý phức tạp từ nhiều nguồn thông tin khác nhau, ta phải chỉnh sửa trực tiếp các báo biểu.

Cách thiết kế một báo biểu cũng tương tự như thiết kế một biểu mẫu nhập liệu. Trong báo biểu trên, ta mới có thông tin về sản phẩm, số lượng, đơn giá và chiết khấu. Tuy nhiên, ta lại không có thành tiền. Ta sẽ bổ sung thông tin này vào báo biểu.

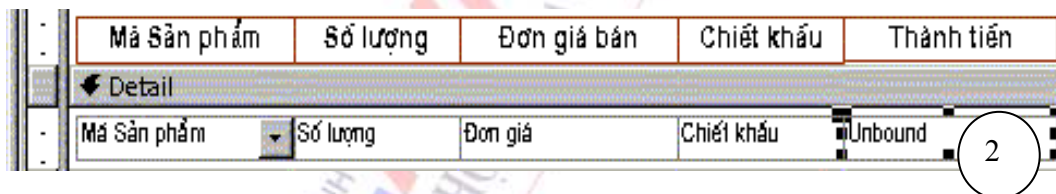
**Bước 1:** Chuyển báo biểu đơn đặt hàng sang chế độ thiết kế bằng cách nhấn chuột vào nút lệnh  (Thiết kế). Lúc này cấu trúc của báo biểu đơn đặt hàng được thể hiện như hình bên.



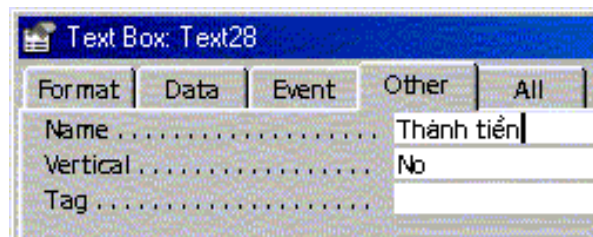
**Bước 2:** Thêm một nhãn tiêu đề (label) vào bên cạnh dòng tiêu đề hiện tại, bằng cách nhấn vào nút Label trên thanh công cụ Toolbox, sau đó nhấn chuột vào vị trí trên báo biểu. Đánh vào dòng chữ “Thành tiền”.



**Bước 3:** Thêm một hộp văn bản (text box) vào phần Detail của báo biểu, bên cạnh vị trí của trường chiết khấu (vị trí ↑), bằng cách nhấn chuột vào nút Text box trên thanh công cụ.

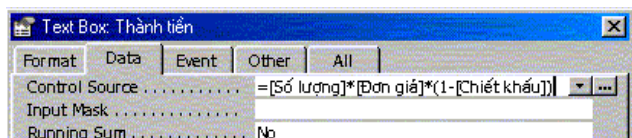


Nhấn đúp vào Text box này để hiển thị cửa sổ thuộc tính - Properties, trong mục Other \ Name, đổi tên của Text box thành : “Thành tiền”



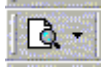
Trong mục Data\Control Source, đánh vào công thức:

$=[\text{Số lượng}] * [\text{Đơn giá}] * (1 - [\text{Chiết khấu}])$ .





Chuyển sang chế độ Nhìn - View để xem kết quả, bằng cách nhấn chuột vào nút lệnh



trên thanh công cụ.

Sau khi đã sửa đổi, báo biểu đơn đặt hàng đã có thêm cột thành tiền chứa tổng số tiền phải trả cho mỗi sản phẩm trong đơn đặt hàng.

Tuy nhiên, báo biểu còn thiếu thông tin về tổng giá trị của mỗi đơn đặt hàng.

Ta bổ sung thông tin này bằng cách sau:

**Bước 4:** Thêm cột mới chứa giá trị cộng dồn của cột thành tiền.

Dùng chuột thêm một điều khiển mới kiểu Text box vào trong báo biểu, ngay cạnh điều khiển có tên là Thành tiền, tại vị trí →, đặt tên là Tổng con.

Tại mục Data\Control Source, nhập vào công thức = [Thành tiền]. Thiết lập mục Data\Running Sum (tính tổng) thành Over group. Việc thiết lập thuộc tính tính tổng con (Over group) cho phép điều khiển này lưu giá trị cộng dồn của từng đơn đặt hàng; sang một đơn đặt hàng mới thì Tổng con này lại được tính lại từ 0.

Mã Sản phẩm	Số lượng	Đơn giá bán	Chiết khấu	Thành tiền	Cộng dồn
Gudbrandsdalsost				28.80	28.80
Outback Lager				60.00	88.80
				\$43.90	
Tofu	3	\$23.25	0%	69.75	69.75
Singaporean Hokki	5	\$14.00	0%	70.00	139.75
Camembert Pierrot	10	\$34.00	0%	340.00	479.75

Nhấn chuột vào nút lệnh trên thanh công cụ để xem kết quả. Báo biểu giờ đây đã có thêm cột mới chứa giá trị cộng dồn tiền hàng của mỗi đơn đặt hàng.

Bây giờ ta thêm dòng chứa tổng giá trị hợp đồng (bằng tổng của Tiền hàng và Tiền phí vận chuyển) vào phần kết thúc của mỗi đơn đặt hàng. Các bước tiến hành như sau:



Mã Sản phẩm	Số lượng	Đơn giá bán	Chiết khấu	Thành tiền
Gudbrandsdalsost	1	\$28.80	0%	28.80
Outback Lager	5	\$12.00	0%	60.00
Mã Đơn đặt hàng: 10625				
Ngày đặt hàng: 08-Aug-1997				Phí Vận chuyển: \$43.90
Mã Sản phẩm	Số lượng	Đơn giá bán	Chiết khấu	Thành tiền
Tofu	3	\$23.25	0%	69.75
Singaporean Hokki	5	\$14.00	0%	70.00
Camembert Pierrot	10	\$34.00	0%	340.00

**Bước 5:** Nhấn chuột vào thanh đơn đặt hàng\_Mã Đ Đ Hg Header. Nhấn phím bên phải để hiện cửa sổ Menu các lệnh, chọn Sorting and Grouping (sắp xếp và phân nhóm). Cửa sổ tiếp theo hiện ra:

Muốn thay đổi cách sắp xếp hoặc thứ tự phân nhóm các bản ghi, ta chỉ ra tên trường trong cột

Biểu tượng này chỉ ra rằng các bản ghi ở trên báo biểu đang được phân nhóm và sắp xếp theo giá trị của trường.

Thay đổi thứ tự sắp xếp của trường. Ascending: tăng, Descending: giảm

Để được phân nhóm, 1 trường phải có Group Header (tiêu đề trên nhóm) hoặc Group footer (tiêu đề dưới)

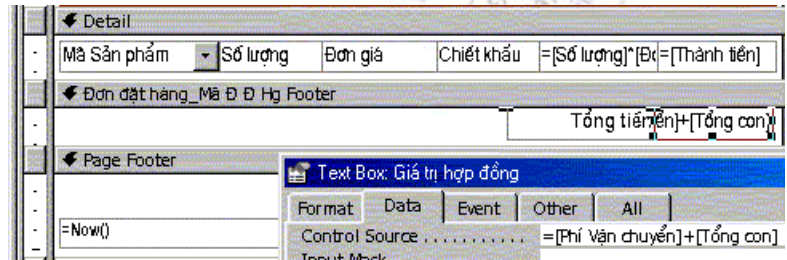
Trong cửa sổ Sorting and Grouping, chọn dòng đơn đặt hàng\_Mã Đ Đ Hg, thiết lập thuộc tính Group Footer (tiêu đề dưới) thành Yes. Việc này để nhằm tạo ra một khoảng trống phía dưới cứ mỗi khi kết thúc một đơn đặt hàng. Cửa sổ khoảng trống như sau:


Group Header	Yes
Group Footer	Yes
Group On	Yes
Group Interval	No
Keep Together	No

Mã Sản phẩm	Số lượng	Đơn giá bán	Chiết khấu	Thành tiền	Cộng dồn
Đơn đặt hàng_Mã Đ Đ Hg Header					
Đơn đặt hàng_Mã Đ Đ Hg Footer					
Page Footer					

Trong dòng trông này, ta thêm một điều khiển dạng Text box, đặt tên là Giá trị hợp đồng, nhập vào trong mục Control Source công thức:

= [Phí Vận chuyển]+[Tổng con].



Chuyển sang chế độ View (  ), kết quả báo biểu đơn đặt hàng hiển thị ra đã có ô chứa tổng giá trị đơn hàng ở cuối.

Đến đây ta đã sử dụng qua tất cả các chức năng cần thiết khi xây dựng một báo biểu.

The screenshot shows a report titled 'Đơn đặt hàng'. It displays two order records. Each record includes customer information, order number, date, and shipping fee. Below each record is a table of products with columns for product name, quantity, unit price, discount, and total price. The total price for each order is highlighted in a red box.

Mã Khách hàng	ANATR				
Mã Đơn đặt hàng	10308				
Ngày đặt hàng	18-Sep-1996				
Phí Vận chuyển	\$1.61				
Mã Sản phẩm	Số lượng	Đơn giá bán	Chiết khấu	Thành tiền	Cộng dồn
Gudbrandsdalsost	1	\$28.80	0%	28.80	28.80
Outback Lager	5	\$12.00	0%	60.00	88.80
Tổng tiền					\$90.41
Mã Đơn đặt hàng	10625				
Ngày đặt hàng	08-Aug-1997				
Phí Vận chuyển	\$43.90				
Mã Sản phẩm	Số lượng	Đơn giá bán	Chiết khấu	Thành tiền	Cộng dồn
Tofu	3	\$23.25	0%	69.75	69.75
Singaporean Hokki	5	\$14.00	0%	70.00	139.75
Camembert Pierrot	10	\$34.00	0%	340.00	479.75
Tổng tiền					\$523.65

**Phụ lục: Làm việc với dữ liệu.**

## 1. Sử dụng bộ xây dựng biểu thức - Expression Builder

Khi xây dựng một query hay một form, một report, ta thường xuyên phải sử dụng đến bộ xây dựng công thức Expression builder. Sử dụng công cụ này, chúng ta có thể nhanh chóng xây dựng được một công thức tham chiếu đến các đối tượng trong một cơ sở dữ liệu một cách chuẩn xác mà chỉ cần sử dụng con chuột.

Bộ xây dựng biểu thức có ba khu vực:

← **Ô công thức:** Khu vực phía trên của giao diện là một ô nơi ta xây dựng các biểu thức. Ta có thể trực tiếp nhập vào từ bàn phím biểu thức; hoặc sử dụng danh sách trong các ô phía dưới - bằng cách nhấn đúp chuột vào tên các đối tượng, toán tử - để dán chúng vào ô biểu thức từ đó tạo nên biểu thức.

↑ **Nút Toán tử:** Tại khu vực giữa của giao diện là các nút nhấn cho các toán tử thường được sử dụng nhất. Nhấn chuột vào nút nào thì Expression Builder sẽ chèn toán tử đó vào trong ô công thức. Muốn có danh sách đầy đủ các toán tử, nhấn chuột vào Operators ở ô phía dưới bên trái và nhóm các toán tử thích hợp. Ô bên phải sẽ hiển thị tất cả các toán tử của nhóm được chọn.

→ **Các phần tử: có ba ô ở nửa dưới của giao diện:**

- Ô phía trái là thư mục các bảng, query, form, và report, các hàm tính (functions), các hằng số - constants), các toán tử - operators.

- Ô ở giữa liệt kê các đối tượng cụ thể hoặc nhóm các đối tượng thành phần của mục đã được chọn tại ô bên trái. Ví dụ, nếu nhấn chuột vào Built-In Functions trong ô bên trái, các loại hàm của Microsoft Access sẽ được liệt kê.

- Ô bên phải liệt kê các giá trị, nếu có, của các phần tử đã được chọn ở hai ô đầu.

## 2. Tìm kiếm và thay thế dữ liệu

Có nhiều cách để tìm kiếm hoặc thay thế các dữ liệu mà ta muốn, dù dữ liệu đó là một giá trị nhất định, một bản ghi hay một nhóm các bản ghi

- Tìm bản ghi bằng cách dùng thanh cuộn ở trên lưới dữ liệu Datasheet hoặc trên biểu mẫu Form, hoặc đánh số thứ tự của bản ghi vào ô Record number trên thanh di chuyển.



- Sử dụng hộp thoại Tìm kiếm – Find, để định vị các bản ghi hoặc tìm kiếm các giá trị nhất định của các trường dữ liệu. Nếu muốn thay thế các giá trị nào đó bằng một giá trị khác, sử dụng hộp thoại Thay thế – Replace.

- Sử dụng một bộ lọc – Filter, ta có thể tạm thời cô lập và hiển thị một tập hợp các bản ghi cụ thể để xử lý trên một Form hoặc một lưới dữ liệu – Datasheet.

- Sử dụng truy vấn query, ta có thể làm việc với một tập hợp cụ thể các bản ghi trích ra từ một hoặc nhiều bảng khác nhau, thỏa mãn các tiêu chuẩn đã chỉ ra ban đầu.

## 3. Sử dụng ký tự đại diện để tìm kiếm giá trị

Khi cần tìm tập hợp các giá trị mà ta chỉ biết một phần của giá trị đó (ví dụ tìm người có tên bắt đầu bằng chữ H và kết thúc bằng chữ ng), ta sử dụng ký tự đại diện để thay vào vị trí các ký tự chưa biết. Ký tự đại diện có thể sử dụng trong hộp thoại Tìm kiếm - **Find**, hộp thoại Thay thế - **Replace**, trong truy vấn - **query**, trong các biểu thức - **expression**.

Trong ACCESS, ta có thể sử dụng ký tự đại diện trong hộp thoại Tìm kiếm, Thay thế, trong truy vấn query, trong biểu thức để tìm các giá trị, các bản ghi, hoặc các tệp.


Ký tự	Sử dụng	Ví dụ
*	Tương đương với một nhóm các chữ cái. Nó có thể được sử dụng ở đầu hoặc ở cuối chuỗi tìm kiếm.	wh* sẽ tìm <i>what, white, và why</i> .
?	Tương đương bất cứ một chữ cái nào.	B?ll sẽ tìm <i>ball, bell, và bill</i>
[ ]	Tương đương với một trong các chữ cái ở trong ngoặc vuông.	B[ae]ll sẽ tìm <i>ball hoặc bell</i> chứ không tìm <i>bill</i>
!	Loại trừ những chữ cái trong ngoặc vuông.	b[!ae]ll chỉ tìm <i>bill</i> chứ không tìm <i>bell,ball</i>
-	Tìm các chữ cái trong khoảng hai chữ cái cho trước (theo thứ tự tăng dần: A đến Z, chứ không được từ Z đến A).	b[a-c]d sẽ tìm <i>bad, bbd, và bcd</i>
#	Tương đương một số.	I#3 sẽ tìm <i>103, 113, 123,...</i>

Lưu ý:

- Ký tự đại diện được sử dụng cho dữ liệu kiểu chữ, tuy nhiên đôi khi cũng có thể sử dụng thành công với các kiểu dữ liệu khác, như kiểu ngày tháng.
- Khi sử dụng các ký tự đại diện để tìm kiếm một dấu hoa thị (\*), dấu hỏi (?), ký hiệu số (#), dấu ngoặc vuông mở ([), dấu nối (-), ta phải cho dấu đó vào trong ngoặc vuông. Ví dụ để tìm một dấu hỏi, ta phi nhập vào [?] trong hộp thoại Tìm kiếm - Find. Nếu tìm kiếm dấu cảm thán (!), dấu ngoặc vuông đóng (]) thì không cần phải cho vào trong ngoặc.
- Muốn tìm kiếm cùng lúc cặp ngoặc vuông thì ta phải nhập vào "[[ ]]" trong hộp tìm kiếm - Find.

#### 4. Tìm kiếm giá trị cụ thể trong một trường - field

1. Trên Biểu mẫu - Form hoặc lưới dữ liệu - Datasheet, chọn cột của trường - field mà ta muốn tìm kiếm (nếu muốn tìm kiếm dữ liệu ở tất cả các trường thì không cần phải chọn).

2. Nhấn nút lệnh Find  trên thanh công cụ.
3. Trên dòng Find What (tìm gì), đánh vào giá, trị muốn tìm.
4. Lựa chọn các tùy chọn (options) tìm kiếm.
5. Nhấn Find Next.



#### 5. Thay thế giá trị trong một trường field

1. Trên Biểu mẫu - Form hoặc lưới dữ liệu - Datasheet, chọn cột của trường - field mà ta muốn tìm kiếm (nếu muốn tìm kiếm dữ liệu ở tất cả các trường thì không cần phải chọn).
2. Chọn Replace trên menu Edit.
3. Đánh vào giá trị cần tìm trên dòng Find What, đánh vào giá trị mới sẽ thay thế giá trị cũ vào dòng Replace With.
4. Thiết lập các tùy chọn cần thiết.
5. Để thay thế tất cả các giá trị cũ bằng giá trị mới, chọn Replace All.
6. Để thay thế chỉ một số các giá trị cũ bằng giá trị mới, chọn Find next, sau đó nhấn Replace. Để bỏ qua không thay thế mà chuyển tiếp đến lần xuất hiện tiếp theo của giá trị đó, nhấn Find Next.

## 6. Sắp xếp dữ liệu

1. Trong khung nhìn Form View hoặc Datasheet View, nhấn chuột chọn trường cần sắp xếp. Muốn sắp xếp các bản ghi ở SubDatasheet, nhấn chuột vào dấu + để mở rộng khung nhìn, sau đó chọn trường.

2. Làm thao tác sau:

- Để sắp xếp tăng dần, chọn **Sort Ascending** .
- Để sắp xếp giảm dần, chọn **Sort Descending** .

Nếu ta chọn nhiều cột (trường) cùng một lúc để sắp xếp thì ACCESS sẽ sắp xếp lần lượt các cột từ trái qua phải

**PHỤ LỤC 1: BẢNG MÃ ASCII**

**BẢNG MÃ ASCII với 128 ký tự đầu tiên**

Hex	0	1	2	3	4	5	6	7
<b>0</b>	NUL 0	DLE 16	SP 32	0 48	@ 64	P 80	` 96	p 112
<b>1</b>	SOH 1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
<b>2</b>	STX 2	DC2 18	“ 34	2 50	B 66	R 82	b 98	r 114
<b>3</b>	♥ 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
<b>4</b>	♦ 4	DC4 20	\$ 36	4 52	D 68	T 84	d 100	t 116
<b>5</b>	♣ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
<b>6</b>	♠ 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
<b>7</b>	BEL 7	ETB 23	‘ 39	7 55	G 71	W 87	g 103	w 119
<b>8</b>	BS 8	CAN 24	( 40	8 56	H 72	X 88	h 104	x 120
<b>9</b>	HT 9	EM 25	) 41	9 57	I 73	Y 89	I 105	y 121
<b>A</b>	LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122
<b>B</b>	VT 11	ESC 27	+ 43	; 59	K 75	[ 91	k 107	{ 123
<b>C</b>	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	 124
<b>D</b>	CR 13	GS 29	- 45	= 61	M 77	] 93	m 109	} 125
<b>E</b>	SO 14	RS 30	. 46	> 62	N 78	^ 94	n 110	~ 126
<b>F</b>	SI 15	US 31	/ 47	? 63	O 79	_ 95	o 111	DEL 127

BẢNG MÃ ASCII với ký tự số 128 - số 255

Hex	8	9	A	B	C	D	E	F
<b>0</b>	Ç 128	É 144	á 160	☐ 176	Ł 192	⋈ 208	α 224	≡ 240
<b>1</b>	ü 129	æ 145	í 161	☐ 177	⌊ 193	⌋ 209	β 225	± 241
<b>2</b>	é 130	Æ 146	ó 162	☐ 178	⌈ 194	⌌ 210	Γ 226	≥ 242
<b>3</b>	â 131	ô 147	ú 163	 179	┌ 195	⋈ 211	Π 227	≤ 243
<b>4</b>	ä 132	ö 148	ñ 164	└ 180	— 196	⋈ 212	Σ 228	∫ 244
<b>5</b>	à 133	ò 149	ñ 165	┘ 181	┘ 197	⋈ 213	σ 229	∫ 245
<b>6</b>	â 134	û 150	ª 166	⋈ 182	┘ 198	⋈ 214	μ 230	÷ 246
<b>7</b>	ç 135	ù 151	º 167	⋈ 183	⋈ 199	⋈ 215	τ 231	≈ 247
<b>8</b>	ê 136	ÿ 152	¿ 168	⋈ 184	⋈ 200	⋈ 216	Φ 232	° 248
<b>9</b>	ë 137	ÿ 153	┘ 169	⋈ 185	⋈ 201	⋈ 217	Θ 233	· 249
<b>A</b>	è 138	ÿ 154	┘ 170	⋈ 186	⋈ 202	⋈ 218	Ω 234	· 250
<b>B</b>	ï 139	ç 155	½ 171	⋈ 187	⋈ 203	■ 219	δ 235	√ 251
<b>C</b>	î 140	£ 156	¼ 172	⋈ 188	⋈ 204	■ 220	∞ 236	n 252
<b>D</b>	ì 141	¥ 157	¡ 173	⋈ 189	= 205	■ 221	φ 237	² 253
<b>E</b>	Ä 142	€ 158	« 174	⋈ 190	⋈ 206	■ 222	ε 238	■ 254
<b>F</b>	Å 143	f 159	» 175	⋈ 191	⋈ 207	■ 223	∩ 239	255



## PHỤ LỤC 2: CÁC LỖI THƯỜNG GẶP TRONG C

### 1. SOẠN THẢO VÀ CHẠY CHƯƠNG TRÌNH

#### a. Soạn thảo chương trình

Mỗi câu lệnh của C có thể viết trên một dòng hay nhiều dòng nhưng phải kết thúc bằng dấu; Tuy nhiên khi nhập một chuỗi ký tự mà muốn chuyển sang dòng khác ta phải thêm dấu \ trước khi xuống dòng.

#### b. Dịch và chạy chương trình

Nếu chương trình chưa viết xong ta có thể nhấn F9 để dịch và sửa lỗi. Khi nhấn F9 thì đầu tiên chương trình được dịch sang tệp có đuôi là \*.obj, sau đó liên kết các tệp và dịch sang tệp có đuôi \*.exe có thể chạy được trong môi trường DOS. Khi chương trình đã tương đối hoàn chỉnh thì ta có thể nhấn Ctrl+F9 để dịch và chạy chương trình.

### 2. CÁC LỖI THƯỜNG GẶP

**Khi dịch chương trình có thể xuất hiện 3 loại lỗi sau đây:**

#### **Lỗi được thông báo bởi từ khóa error (lỗi cú pháp):**

Lỗi này thường xảy ra do khi ta soạn thảo chương trình không tuân theo đúng những quy tắc của C, thí dụ int thì ta gõ thành Int; hay ta gõ thiếu ngoặc đơn, ngoặc kép chẳng hạn...

Sau đây là một số thông báo lỗi thường gặp loại này:

#### **Unknown preprocessor directive**

Chỉ thị tiền xử lý không đúng. Trong trường hợp này bạn phải xem lại các lệnh #include xem bạn có viết sai không.

#### **Declaration terminated incorrectly**

Khai báo kết thúc không đúng. Ví dụ bạn đánh dấu; sau hàm main như sau:

```
void main();
```

Nên lưu ý là sau tên hàm không được đánh dấu; như trên đây. Sau tên hàm phải là dấu { và kết thúc hàm là dấu }.

#### **Unexpected }**

Thừa dấu }. Mỗi lần đánh dấu { thì bạn nên đánh dấu } rồi sau đó gõ các lệnh vào đoạn giữa, như vậy tránh được tình trạng thừa thiếu dấu { hoặc }.

Compound statement missing }

Thiếu dấu }.

### **Declaration syntax error**

Khai báo sai. Ví dụ bạn viết

```
int a,b  
printf("Chao");
```

thì máy báo lỗi ở dòng thứ 2. Sở dĩ như vậy là vì khi đọc qua dòng thứ nhất không có dấu; máy cho rằng lệnh chưa kết thúc và còn chuyển tiếp sang dòng thứ 2. Tuy nhiên sang dòng thứ 2 thì máy lại thấy lệnh không phù hợp nên báo lỗi ở dòng này. Cách viết trên đây tương đương với cách viết:

```
int a,b printf("Chao");
```

Và máy thấy rằng đây là một lệnh không đúng. Còn nếu ta sử dụng lại các lệnh trên là

```
int a,b;printf("Chao");
```

thì máy không còn báo lỗi nữa vì nó chuyển xuống dòng thứ 2 gặp dấu; và biết là lệnh `int a,b;` được khai báo đúng.

### **Undefined symbol**

Bạn đã sử dụng một biến nào đó mà chưa khai báo. Ví dụ bạn chưa khai báo biến `n` nhưng lại sử dụng trong lệnh:

```
printf("%d",n);
```

chẳng hạn thì máy báo là Undefined symbol 'n'

Function ... should have a prototype

Ví dụ trong lệnh trên bạn viết sai là

```
prinrf("%d",n);
```

thì máy báo là Function 'prinrf' should have a prototype. Nghĩa của câu này là: hàm `prinrf` cần phải có nguyên mẫu.

### **Lỗi được thông báo bởi từ khóa Warning (lỗi cảnh báo):**

Lỗi này thường xảy ra do khi ta khai báo biến nhưng không sử dụng tới.

Ví dụ

#### **... is assigned a value that is never used**

Khai báo và đã gán giá trị cho biến nhưng không sử dụng.

Ví dụ bạn viết các lệnh

```
int n; n=10;
```

nhưng trong các phần tiếp theo không sử dụng n (để hiện ra màn hình chẳng hạn, hay dùng để tính giá trị của biến khác...) thì máy báo là

'n' is assigned a value that is never used

tuy nhiên đây chỉ là thông báo (warning). Khi bạn nhấn F9 để dịch chương trình thì máy vẫn báo là success

Hai loại lỗi trên đây được thông báo ngay khi dịch chương trình thành file \*.obj

**Loại lỗi thứ 3 có thể xảy ra trong quá trình liên kết:**

Lỗi này thường xảy ra, thí dụ khi có lời gọi hàm nhưng hàm chỉ mới có nguyên mẫu mà chưa có khai báo chi tiết.



## TÀI LIỆU THAM KHẢO

1. Nguyễn Tiên Huy, *Giáo trình Tin học căn bản*, TT Tin học Đại học Tổng hợp TP. Hồ Chí Minh, NXB Đồng Nai, 1995.
2. Nguyễn Xuân Quốc Hương, *Tin học A & B*, NXB Trẻ, TP. Hồ Chí Minh, 1996.
3. Tô Văn Nam, *Giáo trình nhập môn tin học*, NXB. Giáo dục, 2004.
4. Đinh Vũ Nhân, *Tin học căn bản*, NXB TP. Hồ Chí Minh, 1995.
5. Quách Tuấn Ngọc, *Giáo trình Tin học căn bản*, NXB Giáo dục, Hà Nội, 1995.
6. Dương Trần Đức - Chu Quang Ngọc, *Mạng cục bộ*, Tài liệu dùng cho các khóa học bồi dưỡng, Trung tâm đào tạo BCVT 1, 2001
7. Bùi Thế Tâm, *Tin học văn phòng*, NXB. Giao Thông Vận Tải, 2003.
8. Phạm Văn Át, *Kỹ thuật lập trình C*, Nhà xuất bản KHKT, 1995.
9. Quách Tuấn Ngọc, *Ngôn ngữ lập trình C*, NXB Thống kê, 2003.
10. Đỗ Xuân Lôi, *Cấu trúc dữ liệu và giải thuật*, NXB KHKT, 1994.
11. Nguyễn Duy Phương, *Kỹ thuật lập trình*, Giáo trình giảng dạy tại Học viện CN-BCVT
12. Brian Kernighan, Denis Ritchie, *C Language. Norm ANSI*. Prentice Hall, 1988.
13. Bryon Gottfried, *Programming With C*. McGraw Hill, 1996.
14. Carl Townsend, *Understanding C*. SAMS, 1989.
15. Paul Davies, *The Inspensable Guide to C*. Addison Wisley, 1996.
16. Nikolus L.R. Wirth, *Program = Data Structure + Algorithms*. Prentice Hall, 1992.
17. Phạm Văn Át, *Hướng dẫn sử dụng Microsoft Access*- Nhà xuất bản khoa học và kỹ thuật, 1997.
18. Nguyễn Thiện Tâm, *Giáo trình Microsoft Access 2000* - Nhà xuất bản Đại Học Quốc Gia TP.HCM, 2003.
19. Phạm Thế Quế, *Giáo trình Cơ sở dữ liệu*- Nhà xuất bản Bưu Điện, 2004.
20. Microsoft Access 2000: Buiding Application with Form and Report

MỤC LỤC

<b>Chương 1: CÁC KHÁI NIỆM CƠ BẢN .....</b>	<b>3</b>
1.1. THÔNG TIN VÀ XỬ LÝ THÔNG TIN.....	3
1.1.1. Khái quát.....	3
1.1.2. Biểu diễn thông tin trong máy tính.....	7
1.2. CẤU TRÚC TỔNG QUÁT CỦA HỆ THỐNG MÁY TÍNH.....	20
1.2.1. Nguyên lý thiết kế cơ bản.....	20
1.2.2. Quá trình xử lý thông tin.....	22
1.2.3. Cấu trúc tổng quát của MTĐT.....	23
1.2.4. Lịch sử phát triển của hệ thống máy tính.....	36
1.3. THUẬT TOÁN VÀ SƠ ĐỒ KHỐI.....	37
1.4. MẠNG MÁY TÍNH (COMPUTER NETWORK).....	39
1.4.1. Giới thiệu mô hình tham chiếu OSI.....	39
1.4.2. Khái niệm về mạng và kết nối mạng.....	41
1.4.3. Mạng cục bộ (Local Area Network - LAN).....	42
1.4.4. Cấu trúc liên kết mạng (Topology).....	44
1.4.5. Phương tiện truyền dẫn.....	46
1.4.6. Các thiết bị liên kết mạng.....	47
1.4.7. Truyền tín hiệu.....	49
1.4.8. Phương thức truyền dẫn.....	50
1.4.9. Mạng Internet/Intranet.....	51
<b>Chương 2: HỆ ĐIỀU HÀNH .....</b>	<b>56</b>
2.1. KHÁI NIỆM VÀ PHÂN LOẠI.....	56
2.1.1. Khái niệm.....	56
2.1.2. Phân loại hệ điều hành.....	56
2.2. Hệ điều hành MS - DOS.....	57
2.2.1. Các khái niệm cơ bản của MS-DOS.....	57
2.2.2. Quá trình làm việc của MS-DOS:.....	61
2.2.3. Các lệnh cơ bản của MS-DOS.....	61
2.3. HỆ ĐIỀU HÀNH WINDOWS.....	68
2.3.1. Giới thiệu về hệ điều hành WINDOWS.....	68
2.3.2. Các khái niệm trong Windows:.....	69
2.3.3. Cơ bản về cách cài đặt và sử dụng WINDOWS.....	74
2.4. HỆ ĐIỀU HÀNH UNIX.....	90
2.4.1. Tổng quan về hệ điều hành UNIX.....	91
<b>Chương 3: CÁC PHẦN MỀM ỨNG DỤNG THÔNG DỤNG .....</b>	<b>97</b>
3.1. CHƯƠNG TRÌNH SOẠN THẢO VĂN BẢN WORD.....	97

3.1.1.	Các thao tác điều khiển căn bản.....	97
3.1.2.	Các thao tác soạn thảo cơ bản.....	102
3.1.3.	Lập bảng biểu (TABLE).....	112
3.1.4.	Định dạng trang và in ấn.....	116
3.1.5.	Chèn hình ảnh và công thức toán.....	122
3.2.	BẢNG TÍNH EXCEL 2000 .....	128
3.2.1.	Các khái niệm cơ bản.....	129
3.2.2.	Định dạng dữ liệu .....	132
3.2.3.	Các hàm cơ bản.....	136
3.2.4.	Biểu đồ bảng tính và chèn hình ảnh.....	139
3.3.	POWERPOINT 2000 .....	142
3.3.1.	Khởi động PowerPoint.....	143
3.3.2.	Tạo trình diễn dựa trên Slide trắng .....	144
3.3.3.	Thiết lập cách hiển thị Slide .....	147
3.3.4.	Các công cụ phụ trợ.....	149
3.4.	VIRUS TIN HỌC VÀ CÁCH PHÒNG CHỐNG .....	151
3.4.1.	Virus tin học là gì ?.....	151
3.4.2.	Cách phòng chống virus .....	152
3.4.3.	Các chương trình kiểm tra và diệt virus thông dụng: .....	152
<b>Chương 4: NGÔN NGỮ LẬP TRÌNH C .....</b>		<b>154</b>
4.1.	GIỚI THIỆU CHUNG .....	154
4.2.	MỘT SỐ KIẾN THỨC CƠ SỞ.....	154
4.2.1.	Bộ kí tự, từ khóa, tên .....	154
4.2.2.	Cấu trúc chương trình trong C.....	155
4.2.3.	Các kiểu dữ liệu cơ sở .....	159
4.2.4.	Biến, hằng, câu lệnh và các phép toán.....	161
4.2.5.	Thủ tục vào và ra chuẩn.....	166
4.3.	CÁC CẤU TRÚC LỆNH ĐIỀU KHIỂN .....	173
4.3.1.	Câu lệnh khối.....	173
4.3.2.	Cấu trúc lệnh if.....	173
4.3.3.	Cấu trúc lệnh switch .....	175
4.3.4.	Vòng lặp for Cú pháp: .....	175
4.3.5.	Vòng lặp không xác định while Cú pháp: .....	177
4.3.6.	Vòng lặp không xác định do... while Cú pháp: .....	179
4.4.	HÀM VÀ PHẠM VI HOẠT ĐỘNG CỦA BIẾN.....	180
4.4.1.	Tính chất của hàm.....	180
4.4.2.	Khai báo, thiết kế hàm .....	181
4.4.3.	Phương pháp truyền tham biến cho hàm .....	184
4.4.4.	Biến địa phương, biến toàn cục .....	185
4.4.5.	Tính đệ qui của hàm .....	189

4.5.	MỘT SỐ KIỂU DỮ LIỆU CÓ CẤU TRÚC.....	192
4.5.1.	Cấu trúc dữ liệu kiểu Mảng (Array).....	192
4.5.2.	Xâu kí tự (string).....	201
<b>Chương 5:</b>	<b>HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU ACCESS.....</b>	<b>207</b>
5.1.	TỔNG QUAN VỀ HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU.....	207
5.1.1.	Cơ sở dữ liệu (CSDL).....	207
5.1.2.	Hệ quản trị CSDL:.....	207
5.2.	HỆ QUẢN TRỊ CSDL ACCESS.....	208
5.2.1.	Tổng quan về Access.....	208
5.2.2.	Khởi động, cửa sổ làm việc của Microsoft Access.....	209
5.2.3.	Làm việc với các đối tượng trong Access.....	213
	<b>PHỤ LỤC 1: BẢNG MÃ ASCII.....</b>	<b>263</b>
	<b>PHỤ LỤC 2: CÁC LỖI THƯỜNG GẶP TRONG C.....</b>	<b>265</b>
	<b>TÀI LIỆU THAM KHẢO.....</b>	<b>268</b>
	<b>MỤC LỤC.....</b>	<b>269</b>

