

# **NHẬP MÔN TIN HỌC**

TS. Phan Đăng Cầu

## MỤC LỤC

Phần I. Một số khái niệm cơ bản của tin học .....	1
1.1. Thông tin và xử lý thông tin .....	1
1.1.1. Khái niệm thông tin.....	1
1.1.2. Mã hoá và phân loại thông tin .....	2
1.1.3. Xử lý thông tin.....	2
1.1.4. Tin học là gì? .....	3
1.1.5. Lịch sử phát triển các máy tính số và các thế hệ máy tính. ....	3
1.1.6. Sự phát triển và ứng dụng tin học trong n- ớc và trong ngành BCVT .....	4
1.2. Cấu trúc tổng quát của máy tính điện tử.....	4
1.2.1. Những nguyên lý cơ bản.....	4
1.2.2. Cấu trúc tổng quát của máy tính cá nhân .....	5
1.3. Biểu diễn thông tin trong MTĐT .....	11
1.3.1. Các hệ đếm .....	11
1.3.2. Các phép toán cơ bản của số nhị phân .....	12
1.3.3. Mệnh đề logic .....	13
1.3.4. Biểu diễn dữ liệu.....	13
1.4. Mạng máy tính .....	16
1.4.1. Tổng quan về mạng máy tính .....	16
1.4.2. Mạng Internet/Intranet.....	19
Phần II. Hệ điều hành và các phần mềm hỗ trợ.....	25
2.1. Hệ điều hành là gì? .....	25
2.2. Hệ điều hành MS-DOS.....	26
2.2.1. Các khái niệm chính của MS-DOS.....	26
2.2.2. DOS tổ chức và l- u trữ số liệu trên đĩa nh- thế nào? .....	27
2.2.3. Các modul của MS-DOS.....	31
2.2.4. Một số lệnh thông dụng của DOS .....	35
2.2.5. Cấu hình cho máy tính.....	41
2.3. Phần mềm hỗ trợ NC (Norton Commander).....	42
2.3.1. Giới thiệu tiện ích NC.....	42
2.3.2. Các lệnh của NC.....	44
2.4. Virus máy tính và cách phòng chống .....	47
2.4.1. Khái niệm .....	47
2.4.2. Ch- ơng trình chống Virus .....	48
2.4.3. Cách phòng chống Virus .....	48
2.5. Hệ điều hành Windows 98 .....	49
2.5.1. Giới thiệu về hệ điều hành WINDOWS 98 .....	49
2.5.2. Làm việc với WINDOWS 98.....	50
Phần III. Các ch- ơng trình ứng dụng.....	59
Ch- ơng 3. Microsoft word (Winword) .....	59
3.1. Tổng quan về Winword .....	59
3.1.1. Chức năng của Winword .....	59
3.1.2. Khởi động Winword .....	59
3.1.3. In văn bản: .....	59
3.1.4. L- u cất văn bản Winword .....	60
3.1.5. Thoát khỏi Winword.....	60
3.2. Làm việc với Winword .....	60
3.2.1. Màn hình làm việc của Winword .....	60
3.2.2. Vùng văn bản (Text area) .....	63
3.2.3. Nhập và điều chỉnh văn bản .....	64

3.2.4. Trình bày trang .....	66
3.2.5. Trình bày màn hình .....	68
3.2.6. Lập bảng biểu (Table) .....	69
3.2.7. Kẻ bảng (draw table) .....	71
3.2.8. Các ứng dụng đặc biệt .....	72
3.2.9. Chèn hình ảnh vào văn bản.....	72
3.2.10. Microsoft Equation.....	73
3.2.11. Các công cụ đồ họa (Drawing) .....	73
3.2.12. Định nghĩa từ gõ tắt.....	74
3.2.13. Template (mẫu văn bản) .....	74
3.2.14. Trộn văn bản (Mail Merging).....	74
Ch- ơng 4. Microsoft excel .....	76
4.1. Tổng quan về Excel .....	76
4.1.1. Chức năng của Excel .....	76
4.1.2. Khởi động Excel.....	76
4.1.3. Mở một cửa sổ mới.....	76
4.1.4. In Workbook.....	77
4.1.5. L- u cất bảng tính Excel.....	77
4.1.6. Thoát khỏi Excel .....	77
4.2. Làm việc với Excel.....	77
4.2.1. Màn hình làm việc của Excel .....	77
4.2.2. Nhập và sửa số liệu trong bảng tính .....	80
4.2.3. Biểu đồ bảng tính.....	84
4.2.5. Cơ sở dữ liệu trên bảng tính.....	85
4.2.5. Hàm trong Excel.....	87
Phần V. Ngôn ngữ lập trình C .....	90
Ch- ơng 1. Tổng quan về ngôn ngữ C .....	90
1.1. Ngôn ngữ và thuật toán .....	90
1.1.1. Ngôn ngữ lập trình.....	90
1.1.2. Thuật toán (Algorithm) .....	91
1.1.3. Ngôn ngữ C .....	92
1.2. Khởi động và thoát khỏi C.....	92
1.2.1. Khởi động C .....	92
1.2.2. Mở một cửa sổ mới.....	93
1.2.3. L- u văn bản .....	93
1.2.4. Một vài chú ý khi soạn thảo văn bản .....	93
1.2.5. Thoát khỏi C.....	93
1.3. Các thành phần cơ bản của C.....	93
1.3.1. Bộ chữ viết của C. Từ khóa. Tên.....	93
1.3.2. Một ch- ơng trình C đơn giản .....	95
1.4. Cấu trúc chương trình trong C.....	95
1.4.1. Các thành phần của một ch- ơng trình C.....	95
1.4.2. Soạn thảo và chạy một ch- ơng trình C .....	96
Ch- ơng 2. Hằng, biến và mảng(23).....	98
2.1. Các kiểu dữ liệu cơ sở.....	98
Kiểu char .....	98
Kiểu nguyên .....	98
Kiểu dấu phẩy động.....	98
2.2. Hằng (25).....	99
2.3. Kiểu enum (30).....	100

2.4. Biến (32).....	100
2.5. Mảng (33) và chuỗi .....	102
2.6. Định nghĩa kiểu bằng typedef (36).....	102
2.7. Khối lệnh (37) .....	102
2.8. Vài nét về hàm và ch- ơng trình (41) .....	103
2.9. Biến, mảng tự động (42).....	103
2.9.1. Định nghĩa .....	103
2.9.2. Phạm vi hoạt động và thời gian tồn tại .....	103
2.9.3. Khởi đầu cho biến và mảng tự động.....	103
2.10. Biến, mảng ngoài (43) .....	103
2.10.3. Khởi đầu cho biến và mảng ngoài .....	105
2.11. Toán tử sizeof (47).....	105
2.12. Biến tĩnh, mảng tĩnh (48).....	105
Ch- ơng 3. Biểu thức(53).....	106
3.1. Khái niệm biểu thức(expression, 53).....	106
3.2. Phép toán số học(54) .....	106
3.3. Các phép thao tác bit(55).....	106
3.4. Các phép toán quan hệ và logic(57).....	106
3.5. Phép chuyển đổi kiểu giá trị(59) .....	106
3.6. Phép toán tăng giảm(61).....	106
3.7. Câu lệnh gán và biểu thức(62).....	106
3.8. Biểu thức điều kiện(63) .....	106
e1?e2:e3.....	106
3.9. Thứ tự - u tiên các phép toán(66) .....	106
Ch- ơng 4. Vào ra(70) .....	107
4.1. Hàm printf (70).....	107
4.1.1. Chuỗi điều khiển .....	107
4.1.2. Danh sách các đối.....	108
4.2. Hàm scanf (78) .....	108
4.2.1. Danh sách các đối.....	108
4.2.2. Chuỗi điều khiển .....	108
4.3. Đ- a ra máy in- hàm fprintf (83) .....	109
4.4. Dòng vào stdin và các hàm nhập dữ liệu (93) .....	109
4.5. Nhập /xuất số liệu cho chuỗi và ký tự .....	111
4.6. Một số hàm xử lý chuỗi (549) .....	111
4.7. Sự khác biệt giữa mảng và chuỗi .....	112
4.8. Các hàm vào ra và dịch chuyển vị trí trên màn hình(97).....	113
4.9. Các luồng nhập xuất cin,cout .....	114
Ch- ơng 5. Các toán tử điều khiển(102).....	115
5.1. Nhắc lại khái niệm câu lệnh và khối lệnh (102).....	115
5.2. Toán tử if (103).....	115
5.3. else if (106).....	115
5.4. Toán tử switch (103).....	115
5.5. Toán tử goto và nhãn (111).....	115
5.6. Toán tử for (114) .....	115
5.6. Toán tử while (123) .....	116
5.7. do ... while (126) .....	116
5.8. Lệnh break và lệnh continue (129).....	116
Ch- ơng 6. Hàm và cấu trúc ch- ơng trình(450 Bài tập,71) .....	117
6.1. Khai báo và định nghĩa hàm(71) .....	117

6.2. Các hàm void (73) .....	118
6.3. Lời gọi hàm .....	119
6.4. Sử dụng hàm assert() để kiểm tra điều kiện tr- ớc(76).....	119
6.5. Các đối số mặc định (77).....	119
6.6. Truyền tham số cho hàm (78).....	120
6.6.1. Truyền bằng tham trị(value parameter): .....	120
6.6.2. Truyền theo địa chỉ .....	121
6.6.3. Truyền bằng tham chiếu(reference parameter):.....	121
6.7. Hàm chồng(Overloaded function, 81) .....	123
6.8. Các mẫu (template 250).....	124
6.9. Hàm với các tham số là mảng .....	124
Nhập môn tin học - Câu hỏi và bài tập: .....	126
Một số khái niệm cơ bản của tin học.....	126
Hệ điều hành và các phần mềm hỗ trợ .....	127
Microsoft Word .....	129
Microsoft Excel .....	132
Ngôn ngữ lập trình C .....	133

- Phần I. Một số khái niệm cơ bản của tin học  
 Phần II. Hệ điều hành và các phần mềm hỗ trợ.  
 Phần III. Các chương trình ứng dụng  
 Phần IV. Ngôn ngữ lập trình C

## PHẦN I. MỘT SỐ KHÁI NIỆM CƠ BẢN CỦA TIN HỌC

### 1.1. Thông tin và xử lý thông tin

#### 1.1.1. Khái niệm thông tin

Khái niệm thông tin được sử dụng thường ngày. Thông tin giúp chúng ta nâng cao nhận thức, hiểu biết rõ hơn môi trường xung quanh để có những ứng xử thích hợp. Thông tin ngày càng trở thành nhu cầu cấp thiết của con người như cơm ăn áo mặc. Chúng ta ai cũng thấy được sự cần thiết của thông tin và cảm nhận được thông tin là gì. Mặc dầu vậy nếu phải đưa ra một định nghĩa về thông tin thì hầu hết chúng ta đều lúng túng. Quả thật thông tin là một khái niệm trừu tượng và rất khó đưa ra một định nghĩa chính xác. Sau khi tham khảo một số tài liệu chúng tôi tạm đưa ra định nghĩa 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... Đám mây đen ở cuối chân trời hay đoàn chuồn chuồn bay thấp báo hiệu cho ta cơn giông sắp đến, nghe bản sonata ánh trăng của Beethoven ta hình dung ra một đêm trăng êm đềm như buồn man mác... Các thông báo khác nhau có thể mang cùng một nội dung. Thí dụ bản tin phát đi bằng nhiều thứ tiếng. Trong nhiều trường hợp cùng 1 thông báo lại chuyển tải thông tin khác nhau đến cho từng người: khi nghe 1 bản nhạc không lời rõ ràng mức độ cảm nhận của mỗi người một khác và mỗi người có cách lý giải khác nhau về nội dung bản nhạc. Cùng một cử chỉ gật đầu như ở Bungari lại có nghĩa là từ chối. Thông tin không chỉ có ở con người mà còn tồn tại ở các loài sinh vật: con ong khi phát hiện ra điểm có hoa nó liền múa và những con ong khác có thể căn cứ vào đó mà biết được vị trí nơi có hoa. Thậm chí cây cối cũng có thể trao đổi thông tin cho nhau bằng những dấu hiệu vô hình: theo nhà sinh thái học Mỹ D.Roader, khi một cây bị sâu tấn công nó liền phát ra một dấu hiệu hóa học để báo cho cây bên cạnh, cây bên cạnh liền tăng chất chất trong lá để hạn chế sâu. Gần đây các nhà khoa học Nhật cũng đã chứng minh được sự trao đổi thông tin giữa những cây đậu.

*Thông tin phải chứa yếu tố bất ngờ.* Một thông báo mà mọi người đều biết rõ thí dụ: người rồi cũng đến lúc phải chết, ngày mai mặt trời mọc đằng đông... không chứa thông tin. Hoặc ta có thể xem đây là trường hợp đặc biệt của thông tin, trường hợp suy thoái như trong toán học. Vậy có thể xem thông báo là những biến ngẫu nhiên hoặc tổ hợp của các biến ngẫu nhiên.

Từ lâu người ta muốn định lượng thông tin như khối lượng hay trọng lượng của một vật. Trớc một bản tin hay thông báo, chúng ta đều có cảm nhận bản tin đó chứa nhiều thông tin hay không, tuy nhiên thường ta không thể định lượng và không thể so sánh lượng thông tin trong các thông báo một cách chính xác. Nhà bác học SHANNON khi xây dựng lý thuyết thông tin đã cố gắng đưa ra cách định lượng thông tin như sau:

Giả sử thông báo T về một sự kiện có n trạng thái  $\{x_1, x_2, x_3, \dots, x_n\}$

với  $P(T = x_i) = p_i$ ,  $0 \leq p_i \leq 1$ . Ta gọi độ bất định hay ENTROPI chứa trong thông báo T là đại lượng:

$$H = - \sum_{i=1}^n p_i \log_2 p_i$$

Trong tr-ờng hợp đặc biệt khi T chỉ nhận 2 giá trị 0 hoặc 1 với xác suất nh- nhau ta có  $H=1$ . Trong máy tính khả năng sử dụng các số 0 và 1 là nh- nhau nên thông báo chỉ gồm 1 số nhị phân đ-ợc xem nh- chứa đơn vị thông tin nhỏ nhất và đ-ợc gọi là 1 bit (**Binary digit**).

Trong tin học ng-ời ta th-ờng dùng một số đơn vị bội của BIT nh- sau:

Tên gọi	Viết tắt	Giá trị
Byte	b	8 bit
Word		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	$1024Gb=2^{10}Gb$

### 1.1.2. Mã hoá và phân loại thông tin

Mã hóa thông tin là biểu diễn thông tin d-ới một dạng khác theo quy -ớc. Quá trình ng-ợc lại đ-ợc gọi là phép giải mã. Thí dụ ta có 1 tập hồ sơ các thí sinh. Nếu ta quản lý bằng tên thì rất nhiều tr-ờng hợp bị trùng. Nếu ta thêm các yếu tố khác nh- địa chỉ, ngày sinh thì việc quản lý rất r-ờm rà phức tạp mà vẫn không loại trừ đ-ợc khả năng trùng. Nếu ta gán cho một thí sinh 1 số báo danh khác nhau thì việc quản lý thuận tiện hơn nhiều. Từ số báo danh ta có thể tìm ra thí sinh t-ơng ứng. Quá trình này gọi là giải mã.

Trong quốc phòng để bảo đảm bí mật khi truyền tin ng-ời ta cũng mã hóa thông tin. Chúng ta th-ờng nghe nói đến các bảng mật mã, đó chính là bảng qui -ớc cách biểu diễn các ký tự d-ới một dạng khác.

Trong xử lý thông tin tự động, dạng mã quan trọng đ-ợc dùng là bảng mã nhị phân, tức là ng-ời ta mã hóa bằng chữ cái và các số bằng các số nhị phân là 0 và 1. Bảng mã nhị phân đ-ợc dùng rộng rãi nhất hiện nay là bảng ASCII (American Standard Code for Information Interchange). Bảng mã này đ-ợc biên soạn năm 1968 để tăng c-ờng hiệu quả cho truyền thông dữ liệu và để làm t-ơng hợp các thiết bị máy tính với nhau. Bảng mã ASCII gồm 96 ký tự thể hiện và 32 ký tự điều khiển không thể hiện trên màn hình. Mỗi mã gồm 7 ký tự và 1 bit chẵn lẻ kiểm tra lỗi. Vì ASCII không có các ký tự đồ họa nên hầu hết các máy tính đều dùng bộ ký tự mở rộng có chứa các ký tự theo yêu cầu (eXtended Character set) gồm 254 ký tự, đặt cơ sở trong bộ nhớ ROM của máy bổ sung các ký tự kỹ thuật và ký tự đồ họa khối.

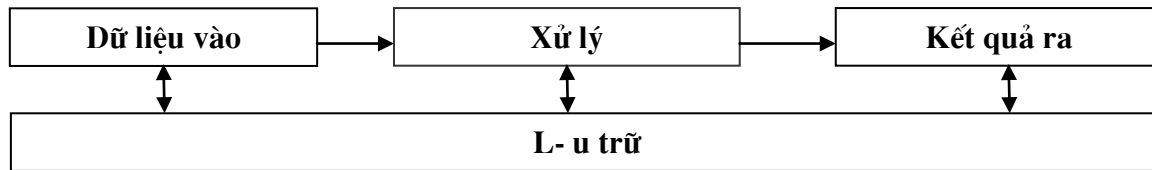
Thông tin có thể nhận các giá trị liên tục thí dụ mức n-ớc thuỷ triều, điện thế. Trong tr-ờng hợp này các giá trị có thể tiếp nhận là vô hạn và ta gọi là thông tin liên tục. Nếu miền giá trị của thông tin là rời rạc thí dụ số nhà trong dãy phố, số tai nạn giao thông trên đ-ờng 5 ta gọi là thông tin rời rạc. Trong tr-ờng hợp này số các giá trị có thể nhận là hữu hạn. Tuy nhiên trong thực tế dù bản chất thông tin là liên tục nh- ng khi ghi nhận thông tin ta đã rời rạc hóa chúng: đo nhiệt độ cơ thể, mức n-ớc thuỷ triều, cân trọng l-ợng của một vật. Mọi thông tin rời rạc có thể biểu diễn bằng số và có thể mã hóa đ-ợc.

Thí dụ về bảng ASCII:           A   có mã là 0100 0001  
  B   có mã là 0100 0010

Bảng mã nổi tiếng thứ 2 đ-ợc dùng ở Mỹ là bảng mã EBCDIC (The Extended Binary Coded Decimal Interchange Code). Bảng mã EBCDIC dùng từ 8 bit để mã hóa 256 ký tự, do hãng IBM lập để dùng cho các máy tính lớn.

### 1.1.3. Xử lý thông tin

Trong thực tế con ng-ời th-ờng phải xử lý thông tin để nhận đ-ợc những thông tin có ích và thiết thực hơn, từ đó họ có những ứng xử thích hợp. Thí dụ từ số liệu điều tra dân số ng-ời ta phân tích xử lý để tính ra thu nhập trung bình giữa miền xuôi và miền ng-ợc, phân tích so sánh đời sống văn hoá của đồng bào nông thôn và thành thị, từ đó giúp các nhà lãnh đạo đ- ra ra những chính sách hợp lý. Quá trình xử lý thông tin có thể rất phức tạp, tuy nhiên ta có thể tóm tắt bằng sơ đồ sau:



### Xử lý thông tin bằng máy tính điện tử (MTĐT):

Quá trình xử lý thông tin trên MTĐT là một quá trình xử lý tự động dựa trên nguyên tắc chung của 1 quá trình xử lý thông tin. Mặc dầu khả năng tính toán của máy tính v- ợt xa khả năng tính toán của con ng- ời và các ph- ơng tiện khác nh- ư máy tính lại chỉ có thể hoạt động theo sự chỉ dẫn của con ng- ời, dĩ nhiên là sự chỉ dẫn đó đã đ- ọc dịch ra ngôn ngữ mà máy có thể hiểu đ- ọc. Ta có thể hình dung quá trình xử lý thông tin trên MTĐT bằng l- ược đồ d- ưới đây:



Máy tính điện tử có một số tính năng cơ bản sau:

- Tốc độ xử lý nhanh, độ tin cậy cao
- Khả năng nhớ rất lớn.

#### **1.1.4. Tin học là gì?**

Tin học là một ngành khoa học công nghệ nghiên cứu các ph- ơng pháp xử lý thông tin một cách tự động dựa trên các ph- ơng tiện kỹ thuật mà chủ yếu hiện tại là MTĐT.

Nh- ư vậy khía cạnh khoa học của tin học là ph- ơng pháp, khía cạnh kỹ thuật của tin học chính là công nghệ chế tạo các thiết bị MTĐT và sản xuất các phần mềm hệ thống và ứng dụng. MTĐT vừa là ph- ơng tiện, vừa là đối t- ượng nghiên cứu của tin học. **Ng- ời ta phân chia tin học thành 2 phần: phần cứng và phần mềm.** Phần cứng là toàn bộ các thiết bị kỹ thuật của MTĐT, phần mềm là các ch- ơng trình có chức năng điều khiển, khai thác phần cứng để thực hiện yêu cầu ng- ời sử dụng.

#### **1.1.5. Lịch sử phát triển các máy tính số và các thế hệ máy tính.**

Có 2 loại máy tính:

- Máy tính xử lý các thông báo liên tục đ- ọc gọi là máy tính t- ơng tự (analog) hay máy chuyên dụng.
- Máy tính xử lý thông báo rời rạc đ- ọc gọi là máy tính số.

Nhờ những thiết bị chuyển đổi thông báo từ dạng liên tục sang số và ng- ọc lại nên máy tính số cũng có thể xử lý thông báo liên tục. Chính vì vậy ngày nay máy tính số đã đ- ọc sử dụng rộng rãi trong mọi lĩnh vực của cuộc sống.

Vì lẽ đó trong tài liệu này chúng tôi chỉ trình bày về máy tính số.

Ng- ời ta xem máy tính số đầu tiên là bàn tính gậy đ- ọc tìm thấy ở Trung quốc, sau đó đến thời tiền trung cổ đ- ọc đ- a sang châu Âu. Tuy nhiên nếu ta xem máy tính số phải có một chút tự động hóa thì chiếc máy tính cơ học đầu tiên đ- ọc chế tạo bởi giáo s- ơ Do Thái W.Schickard (1592 - 1632) phỏng theo các thao tác chuyển vận kim đồng hồ của các nghệ nhân Trung quốc. Máy của Schikard thực hiện tự động đ- ọc các phép cộng trừ và bán tự động các phép nhân chia. Độc lập với Schikard, nhà toán học nổi tiếng ng- ời Pháp Blaise Pascal (1623 - 1662) cũng chế tạo đ- ọc một máy tính cơ học vào năm 1642 lúc ông 19 tuổi. Máy tính này thực hiện đ- ọc 2 phép tính là cộng và trừ và chuyển đổi các đơn vị tiền tệ. Vào cuối thế kỷ 17 nhà toán học Leibniz ng- ời Đức (1646 -1716) phát triển máy của Pascal để có thể thực hiện đ- ọc thêm các phép tính nhân chia. Công trình của Schikard, Pascal và Leibniz đ- a ra đ- ọc ý t- ờng lớn: có thể cơ học hóa các phép tính số. Tuy nhiên,



hạn chế của các máy tính nói trên là không có khả năng nhớ các kết quả trung gian. Đến đầu thế kỷ 19 nhà toán học, thiên văn học, kinh tế học, triết học lớn người Anh Charles Babbage (1791 - 1871) đã đưa ra nguyên lý máy tính số. Hồi đó máy tính tự động tự động khá rộng rãi, tuy nhiên Babbage đã nêu được những nhược điểm của máy tự động:

- Không có hàm tự động tổng quát để tính các hàm bất kỳ.
- Dùng máy tự động tự động khó thu được độ chính xác cần thiết.
- Khó lắp ráp kết quả tạo ra trong tính toán.

Cách nhìn của Babbage về sau được Alan Turing xây dựng cơ sở vững chắc vào năm 1935.

Babbage đã đề xuất một mô hình máy hoạt động theo nguyên lý số nhị phân:

- Tự động hóa các phép toán số học trực tiếp với các số.
- Các thành phần máy được phân định chức năng rõ ràng: đơn vị số học-logic, bộ nhớ, đơn vị điều khiển, đơn vị vào ra.
- Dùng băng đục lỗ làm kênh liên lạc với máy tính.

Những ý tưởng trên đây đã được J.V.Neumann (1903 -1957) tiếp tục phát triển thành một nguyên lý cơ bản cho mọi thế hệ máy tính sau này.

Máy tính điện tử đầu tiên được chế tạo ở Mỹ năm 1946 mang tên ENIAC (Electronic Numerical Integrator And Calculator). Máy gồm 20 000 ống đèn điện tử và có thể thực hiện 300 phép tính nhân trong 1 giây. Tuy nhiên máy tính điện tử đầu tiên cho mục đích thương mại là máy UNIVAC 1, được chế tạo năm 1951 ở Mỹ. Từ đó cho đến nay có thể phân chia sự phát triển của máy tính điện tử thành các thế hệ như sau:

- **Thế hệ 1 (50-59):** Bắt đầu bằng UNIVAC-1, đỉnh cao là IBM-1401. Đặc điểm:  
Dùng bóng đèn điện tử, sử dụng ngôn ngữ máy, tốc độ : 2 000 -> 20 000 phép tính/giây
- **Thế hệ 2 (60-63):** Đặc điểm:  
Dùng bóng bán dẫn, sử dụng ngôn ngữ FORTRAN, COBOL, ALGOL, PL1, tốc độ : 200 000 phép tính/giây
- **Thế hệ 3 (64-74):** IBM-360, IRIS (Pháp), Odra-1900 (Ba Lan). Đặc điểm:  
Dùng mạch tích hợp (IC), tốc độ : 1 000 000 phép tính/giây. CPU hoạt động ở chế độ đa chương trình, thiết bị nhớ ngoài được cải tiến: đĩa từ thay cho trống từ.
- **Thế hệ 4 (75- nay):** Bắt đầu bằng sự ra đời của IBM-370 của hãng IBM, sau đó là họ XT đầu 80, rồi AT-286, AT-386, AT-486, Pentium của hãng Intel. Đặc điểm:  
Dùng mạch tích hợp cỡ lớn, tốc độ : 10 000 000 — 500 000 000 phép tính/giây.

Cũng có quan điểm cho rằng thế hệ thứ tư kết thúc vào năm 77 và từ năm 78 cho đến nay là thế hệ thứ 5.

### ***1.1.6. Sự phát triển và ứng dụng tin học trong nông nghiệp và trong ngành BCVT***

Tin học đang ngày càng được ứng dụng rộng rãi trong mọi lĩnh vực của cuộc sống. BCVT là ngành có ứng dụng tin học rất đa dạng:

- Trong Bưu chính: Các phần mềm quản lý như: phần mềm bưu chính, tính cước, phát chuyển bưu phẩm bưu kiện, quản lý danh bạ...
- Trong Viễn thông: Các phần mềm tổng đài, quản lý mạng viễn thông, truyền dẫn.
- Các phần mềm khác: Tài chính kế toán, quản lý nhân sự, vật tư, tài sản...

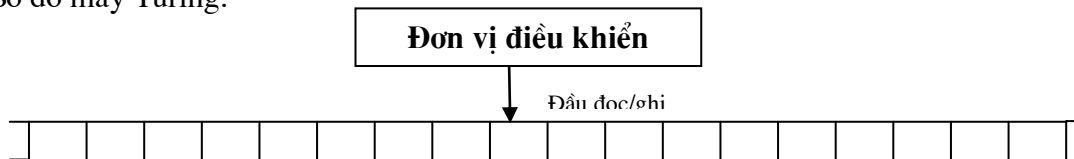
## **1.2. Cấu trúc tổng quát của máy tính điện tử**

### ***1.2.1. Những nguyên lý cơ bản***

#### ***a. Nguyên lý Turing***

Năm 1936, nhà toán học trẻ tuổi Alan Turing đã đưa ra mô hình lý thuyết đơn giản về một thiết bị tính toán mà ngày nay thường được gọi là máy Turing. Thiết bị này hết sức đơn giản nhưng có thể tính được mọi hàm được xem là tính được. Điều này có nghĩa là: mọi quá trình tính toán có thể thực hiện được thì đều có thể mô phỏng lại trên máy Turing. Chính vì vậy máy Turing ngày nay vẫn được xem là cơ sở cho lý thuyết tính toán hiện đại (và các máy tính hiện đại vẫn hoạt động theo nguyên lý này).

Sơ đồ máy Turing:



Bảng vô hạn ở 2 đầu gồm các ô, mỗi ô có thể ghi 1 ký tự của máy

Một máy Turing gồm có:

- Một bảng ký tự riêng, thí dụ với máy tính hiện nay bảng ký tự là  $\{0,1\}$
- Một đơn vị điều khiển với đầu đọc/ghi, có thể đọc ghi các ký tự vào ô hiện thời và dịch sang trái hoặc phải 1 ô. (ngày nay ĐVĐK này là CPU)
- Một bảng vô hạn ở 2 đầu gồm các ô để ghi các ký tự (Ngày nay là bộ nhớ)

Máy hoạt động như sau:

Trong quá trình tính toán máy Turing luôn ở trong một điều kiện xác định gọi là trạng thái. Sự tính toán được bắt đầu ở một trạng thái đặc biệt gọi là trạng thái khởi đầu và dừng khi máy đạt tới trạng thái kết thúc. Quá trình tính toán bao gồm một dãy các bước được thực hiện bởi ĐVĐK. Tại mỗi bước: đầu tiên máy đọc ký tự trong ô, viết một ký tự khác lên ô đó, dừng, dịch sang trái hay phải 1 ô tùy theo trạng thái hiện tại quy định và chuyển sang trạng thái mới.

### ***b. Nguyên lý Von-Newman***

Năm 1946 Von Newman với sự cộng tác của W.Burks và H.Goldstein đã công bố bài báo nổi tiếng nhan đề: "Thảo luận sơ bộ về thiết kế logic của máy tính điện tử", những tư tưởng của bài báo này sau đó đã được gọi chung là "nguyên lý Von-Newman" và đã trở thành nguyên lý cơ sở cho việc chế tạo nên MTĐT từ đó cho đến nay. Vì vậy V.Newman được dư luận đánh giá là: "Người cha tinh thần, kiến trúc sư vĩ đại của nhiều thế hệ máy tính". Nội dung chính của nguyên lý Von Newman là:

- **Máy được hoạt động theo chương trình lưu trữ trong máy**

Chương trình-một tập hợp các lệnh máy- được coi như dữ liệu và được lưu trữ trong máy. Chương trình chỉ cần nhập 1 lần, lưu trữ trên các thiết bị nhớ để bảo quản và chạy nhiều lần.

- **Bộ nhớ được địa chỉ hóa**

Để sử dụng các dữ liệu, chỉ cần thay đổi phân biểu thị địa chỉ của vùng nhớ chứa dữ liệu đó. Như vậy việc truy cập tới dữ liệu là gián tiếp và dữ liệu được xác định thông qua địa chỉ.

### ***1.2.2. Cấu trúc tổng quát của máy tính cá nhân***

Trong phần này chúng tôi chỉ hạn chế giới thiệu cấu tạo của máy tính cá nhân (Personal Computer), là loại máy đang được dùng rộng rãi ở Việt Nam và nhiều nước khác trên thế giới. Máy tính cá nhân đã có từ trước máy IBM PC, nhưng sự ra đời của IBM PC năm 1981 trên cơ sở sử dụng bộ vi xử lý 8088 của hãng Intel đã chính thức hóa nền công nghiệp máy tính cá nhân non trẻ, và đã bảo đảm cho sự chấp nhận về công nghệ trong cộng đồng doanh nghiệp. Chiếc máy 1981 là một trở ngại cho các tiêu chuẩn hiện nay: RAM 16K, ổ đĩa 160K, bus dữ liệu bên trong là 16 bit, nhưng trao đổi với các bộ phận khác của máy chỉ với 8 bit. Đến năm 1983 máy XT (eXtended Technology) ra đời cũng với bộ vi xử lý 8088 với tốc độ xử lý khoảng 5MHz và có bổ sung thêm đĩa cứng. Đến năm 1983 máy AT (Advanced Technology) ra đời với bộ vi xử lý 80286 đã cải tiến đáng kể tính năng của các máy PC và XT. Cùng với sự phát triển của các thế hệ máy tính lớn, máy tính cá nhân cũng

nhánh chóng tiếp cận với ng-ời dùng với một tốc độ chóng mặt. Sự phát triển của các họ vi xử lý, đứng đầu là của Intel với các Chip Pentium các thế hệ làm cho tốc độ máy tính cá nhân đạt tới mức gần 1 GHz (1 tỷ phép tính /s). Một cách phân chia máy tính cá nhân về mặt chức năng theo kinh điển gồm:

- Phần cứng
- Phần mềm

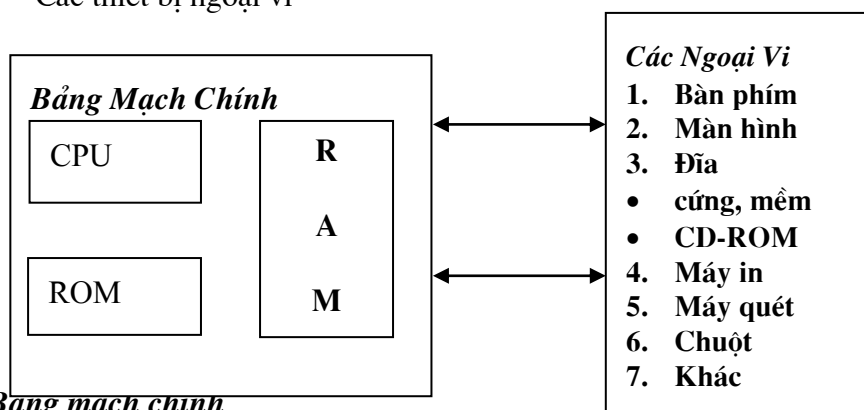
(Một số tác giả khác còn đ-a ra khái niệm phần sụn hay phần nhão, nh-ng trong phạm vi tài liệu này không đề cập sâu đến các khái niệm đó).

### a. Phần cứng

Có thể xem là các thiết bị cấu kiện mang tính vật lý có thể tiếp xúc bằng tay chân đ-ợc nh- máy in, ổ đĩa, các mạch tích hợp (IC). Một trong các thiết bị dùng cho máy tính cá nhân đáng kể là các IC lập trình đ-ợc.

Theo một cách nhìn hệ thống, chúng ta có thể phân chia phần cứng làm hai phần:

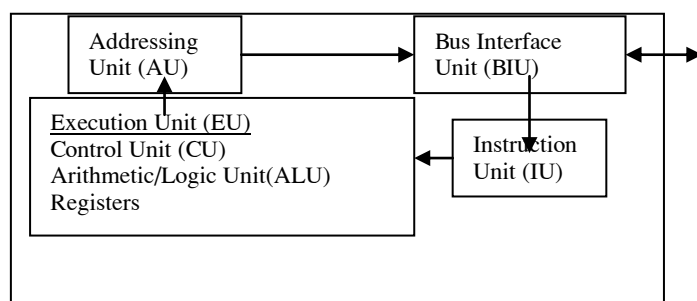
- Bảng mạch chính
- Các thiết bị ngoại vi



### Bảng mạch chính

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 thí dụ khe cắm cho các vĩ mạch điều khiển ổ đĩa cứng, ổ đĩa mềm, modem, âm thanh, video...

CPU (Central Processing Unit)



**CPU** - đơn vị xử lý trung tâm, hay còn gọi là bộ vi xử lý, là bộ phận đầu não của máy tính, nó thực hiện các lệnh, tính toán và điều khiển các phần cứng. CPU bao gồm 3 thành phần chính: Đơn vị ghép nối bus BIU, đơn vị lệnh IU và đơn vị thực hiện lệnh EU. CPU là một vi mạch làm bằng Silicon và có kích cỡ khoảng bằng đầu ngón tay cái. Đ-ợc bọc trong một lớp vỏ màu đen, CPU đ-ợc gắn vào bảng mạch chính bằng các chân chấm nhỏ. Mạch tích hợp nhỏ xíu này bao gồm các Transistor (các công tắc bán dẫn mini), các điện trở chỉnh l-u dòng điện và các tụ điện l-u dòng điện. Bộ CPU thực sự đầu tiên là Intel 4004 đ-ợc sản xuất năm 1971. Hãng Intel đ-ợc thành lập năm 1968 là hãng chuyên sản xuất các chip bộ nhớ. Sau khi bộ vi xử lý 8088 của họ đ-ợc chọn sử dụng cho máy IBM-PC thì doanh thu của hãng tăng cao. Năm 1993 hãng Intel đ-a ra chip Pentium

với 3 triệu transistor và một bộ nhớ 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 bộ nhớ dẫn lệnh cho phép nó xử lý 2 lệnh 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, chạy trên máy Pentium 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). Hãy nghĩ một chu kỳ như một tích tắc đồng hồ, bất kỳ điều gì xảy ra trong máy tính đều xảy ra trong cái tích tắc ấy. Một sự kiện kế tiếp phải chờ tích tắc tiếp theo. Các bộ CPU còn khác nhau ở lượng dữ liệu mà chúng có thể chuyển nhận trong một đơn vị thời gian. Bộ nhớ truyền dữ liệu càng rộng (tức là có thể truyền nhiều bit một lúc) thì dữ liệu đến và đi khỏi CPU trong một chu kỳ đồng hồ càng nhiều. Trong họ 80486 thì các bộ DX sử dụng BUS-32, còn các bộ SX sử dụng BUS-16 nên chậm hơn.

Các CPU điển hình có nhiều hãng sản xuất cho máy tính cá nhân như Intel, Motorola, Zilog... Trên thị trường Việt Nam thường dùng các máy tính cá nhân với CPU của Intel. Họ máy tính đầu tiên có thể kể đến

- Intel 8088/86 với máy tính cá nhân IBM Personal Computer XT-8086 (XT:extended Technology, 1983).

- Intel 80286, máy tính cá nhân IBM Personal Computer AT-80286 (AT:Advanced Technology).

- Intel 80386, máy tính cá nhân IBM Personal Computer AT-80386 (1984)...

- Intel 80486, máy tính cá nhân IBM Personal Computer AT-80486 v.v...

và hiện nay là Pentium với các tốc độ khác nhau. Có thể nói CPU là “ông chủ” của máy tính.

**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 xóa được. Đối với EPROM thì chương trình đã ghi vào có thể dùng thiết bị đặc biệt để ghi và xóa được với các chế độ điện áp và truy nhập khác nhau. Đối với ng-ời dùng, chương trình ghi trong ROM chỉ có thể đọc ra để dùng cho nên nó thường được gọi là bộ nhớ chỉ đọc. Từ một góc độ khác, ROM là một linh kiện IC thuộc phần cứng; tuy nhiên nó lại được ghi chương trình trong đó nên nó lại có thể coi là phần mềm. Các nhà tin học trên quan điểm đó xếp ROM vào loại trung gian và gọi là phần sụn (hay phần nhão)!

Trong máy tính, ROM đóng vai trò gì? Khi truy nhập máy tính có nhiều thủ tục và chương trình cơ sở cần phải tiến hành. Có thể ví CPU là ông chủ thì ROM như ng-ời thư ký. Mỗi khi máy tính bắt đầu hoạt động (khởi động hệ thống) có một số động tác cần phải làm đầu tiên được gọi là POST (Power On Self Test) như kiểm tra các cấu thành chính của hệ thống như RAM, ổ đĩa, bàn phím có hoạt động bình thường không? Để làm những công việc chần chừ đó, các nhà thiết kế máy tính viết những chương trình thực hiện các thủ tục đó, ghi vào ROM và mỗi khi khởi động máy, chương trình đó được thực hiện. 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. Đây cũng là 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 của RAM so với ROM ở chỗ: 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 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 đổ xuống RAM c- trú trong đó và được CPU lấy và xử lý. Một chương trình nào đó có thể được tải xuống c- 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 MB RAM tùy nhu cầu ng-ời 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 RAM thường được tạo nên bởi các thẻ mạch DRAM, tuy nhiên một phần của RAM là các thẻ mạch SRAM có tốc độ cực nhanh được dùng làm MEMORY CACHE (bộ nhớ đệm của bộ nhớ chính) để dùng lưu trữ dữ liệu hoặc các lệnh hay dùng đến, để làm cân bằng tốc độ làm việc của CPU quá nhanh với các bộ phận khác chậm hơn.  
(- Trong bảng mạch điều khiển đĩa cũng có những chip RAM rồi dùng làm bộ nhớ đệm đĩa và được gọi là DISK CACHE).

Ngoài ba linh kiện chính kể trên, *bảng mạch chính còn chứa các mạch khác dùng cho báo hiệu, chuyển đổi, mạch điều khiển loa... các slot để cắm các bảng mạch ghép nối...*

**BUS** Được truyền. Đây là một đường dẫn để truyền tín hiệu từ bộ phận này sang bộ phận khác trong máy tính. Có 3 loại BUS và không phải luôn đi kèm với nhau: Bus dữ liệu (8,16, hoặc 32 bit), Bus địa chỉ (20 bit) và Bus điều khiển truyền các tín hiệu điều khiển. Tại các khe cắm mở rộng đều có đường truyền gồm 62 dây dẫn bao gồm cả các loại BUS nói trên để nối với đường truyền chung và 8 đường nhận các mức điện áp nguồn khác nhau. Vậy tất cả các khe cắm mở rộng đều nhau và khi cần cắm thêm một vỉ mạch mới ta có thể cắm vào đâu cũng được. Sau đó việc nhận và truyền tín hiệu đi sẽ phụ thuộc vào vỉ mạch cắm thêm đó.

### Các thiết bị ngoại vi.

Các thiết bị ngoại vi là các phần cứng nối ghép thêm với bảng mạch chính để tạo nên một kết cấu chuẩn cho hệ thống máy tính như bàn phím, màn hình, chuột, ổ đĩa. Ngoài ra, hàng chục thiết bị khác có thể nối thêm với máy tính để mở rộng như máy in, máy vẽ, máy quét và nhiều thiết bị khác. Với cách tổ chức như vậy, các hãng sản xuất tạo cho người dùng khả năng mở rộng rất lớn, mềm dẻo và máy tính không chỉ có tác dụng tính toán như quan niệm nhiều năm trước đây mà là một phần quan trọng trong các hệ thống xử lý thông tin.

Thông thường các thiết bị ngoại vi ghép nối với bảng mạch chính thông qua các bộ biến đổi và còn gọi là các bộ thích nghi, hoặc có các tên gọi riêng cho các thiết bị ghép nối đặc biệt. Các thiết bị ghép nối đó được cắm vào các khe có sẵn trong bảng mạch chính với số lượng tùy ý theo thiết kế của bảng mạch chính.

Bảng mạch chính liên hệ với các thiết bị ngoại vi thông qua các cổng. **Cổng (Port)** là các lỗ cắm đặt phía sau máy tính được dùng để nối các thiết bị ngoại vi như máy in, màn hình, bàn phím, chuột, modem hay cần điều khiển. Các dây cáp từ mỗi thiết bị ngoại vi được cắm vào các cổng này cho phép các thiết bị nối kết gởi và nhận thông tin từ bộ vi xử lý. Có 2 loại cổng: nối tiếp và song song. Cổng nối tiếp chuyển mỗi lần một bit dữ liệu, cổng song song có thể chuyển 8 bit hay 1 byte. Cáp song song cho phép nối tốc độ cao nhưng dây càng dài thì rủi ro nhiễu âm càng lớn do ảnh hưởng lẫn nhau giữa các sợi song song. Do đó các dây cáp máy in song song thường không dài quá 10 - 15 feet (1 foot ~ 0.3m). Có thể bố trí 3 cổng song song được gọi là LPT1, LPT2, LPT3 (LPT=Line Printer). Thiết bị có tên PRN cũng tương tự như LPT1. Các cổng nối tiếp được đặt địa chỉ bên trong là COM1, COM2. Trong các máy tính thế hệ mới còn có cổng SCSI (Small Computer System Interface: giao diện hệ máy tính nhỏ) Cổng SCSI được chế tạo cho các thiết bị ngoại vi tốc độ cao như ổ CD-ROM hay một ổ cứng bên ngoài.

**Bàn phím (Keyboard):** Là thiết bị vào trong hệ thống máy tính. Đây là phần cứng để nhập dữ liệu tương đối kinh điển và được mở rộng và cải tiến nhiều. Bàn phím có thể chia làm năm khối theo bố trí.

**Khối chính** nằm giữa bàn phím là các ký tự chữ cái, số, các dấu, các phím điều khiển như Ctrl, Alt, Shift, Tab, xóa lùi (Backspace), Khóa chữ hoa (Caps Lock)... Để chuyển đổi chữ hoa thành chữ thường hoặc ngược lại ấn và giữ phím Shift (thường dùng để đổi từng ký tự). Một số phím nhập ký tự dấu bằng cách ấn và giữ phím Shift và gõ phím tương ứng để lấy ký tự dấu phía trên. Để đổi kiểu chữ thường thành chữ hoa hoặc ngược lại một cách cố định, ấn phím Caps Lock. Khi ở chế độ chữ

hoa, thông thường ở một số bàn phím có đèn tín hiệu ở phía trên bên phải bàn phím sáng, hoặc tắt để chỉ chế độ đang ở trạng thái hoạt động hay không.

**Nhóm thứ hai** ở phía trên cùng là các phím chức năng từ F1 đến F12 (Ký tự F viết tắt của chữ Function: Chức năng). Tùy thuộc chương trình đang dùng, mỗi phím chức năng được gán một chức năng (hay một lệnh) thường được sử dụng để ngừng dùng từ bàn phím xử lý nhanh các thao tác của mình. Trong nhóm này còn có phím Esc (Escape: thường dùng để hủy bỏ một lệnh hoặc một thao tác); Phím in (Print Scrn) những hiển thị trên màn hình theo truyền thống của Hệ điều hành DOS; phím cuộn (Scroll Lock); phím dừng (Pause).

**Nhóm thứ ba:** nhóm bàn phím số bao gồm các số thập phân, các dấu phép toán số học. Các phím này có chức năng kép. Muốn chuyển chế độ cần ấn phím NumLock (phím khóa số). Trong nhiều trường hợp có thể gõ ký tự số ở nhóm bàn phím số hoặc ở nhóm chính của bàn phím.

**Nhóm thứ tư :** gồm bốn phím chỉ các mũi tên lên xuống, trái, phải dùng để chỉnh vị trí con trỏ lên, xuống một dòng, hoặc sang trái, phải một ký tự.

**Nhóm cuối cùng:** có sáu phím chèn (Insert), xóa (Delete), về đầu dòng (Home), về cuối dòng (End), con trỏ lên một trang màn hình (Page Up), con trỏ xuống một trang màn hình (Page Down) thường dùng trong các thao tác soạn thảo hoặc một số chức năng khác tùy thuộc từng chương trình cụ thể.

Hiện nay, trên thị trường có thể còn có nhiều loại bàn phím khác nhau và các mô tả trên chỉ đúng cho các loại bàn phím thông thường. Phần cứng cấu trúc bàn phím nói chung ở đây không mô tả.

**Màn hình (Monitor):** là thiết bị ra chuẩn của hệ thống máy tính. Về mặt cấu trúc, màn hình máy tính giống với màn hình của TV cho loại dùng ống đèn điện tử, nhưng không có phần chọn kênh TV, cao trung tần... Các phím điều chỉnh sáng tối, tương phản, dịch dọc, dịch ngang (nếu có bố trí) giống như màn hình TV.

Màn hình máy tính có thể chia thành loại tinh thể lỏng hoặc màn hình dùng ống điện tử. Các loại màn hình dùng ống điện tử có thể khác nhau theo về thức, màu hoặc đen trắng.

**Chuột (Mouse):** là thiết bị vào được dùng thay thế nhiều thao tác bàn phím nhất là dùng cho các kỹ thuật đồ họa máy tính. Chuột cũng như bàn phím là thiết bị vào dữ liệu nối tiếp. Chuột dùng viên bi dùng để điều chỉnh vị trí của trỏ chuột trong không gian thể hiện. Chuột có thể có loại hai nút hoặc ba nút điều khiển và có rất nhiều loại khác nhau. Đi cùng với chuột cần có chương trình điều khiển chuột.

**Đĩa (Disk):** là một trong các loại thiết bị nhớ ngoài để lưu trữ thông tin. Trong máy tính xách tay hiện nay có ba loại đĩa thường dùng: Đĩa mềm (Floppy Diskette), đĩa cứng hay đĩa cố định (Hard Disk hay Fixed Disk), đĩa CD.

**Đĩa mềm:** sử dụng ô xít sắt màu đen nâu phủ trên bản nhựa tròn mỏng có thể cho quay được để đọc ghi thông tin ghi trên đó. Phía ngoài, đĩa mềm có thể có vỏ bọc bằng giấy (nhuộm loại 360 KB) hoặc bằng nhựa (nhuộm loại 1,44 MB). Đĩa mềm có thể phân loại theo dung lượng hoặc theo kích thước. Theo kích thước có loại 5,25, 3,5... inch. Kích thước của đĩa mềm ngày càng có thể thu hẹp nhưng dung lượng ngày càng tăng do công nghệ ngày càng cao. Mỗi loại đĩa mềm đều thường có một ổ đĩa đọc riêng với kích thước tương ứng. Đĩa mềm có đầu đọc trên và dưới, có một động cơ bước điều khiển quay đĩa theo chiều cố định theo bậc tự do thứ nhất. Bậc tự do thứ hai là chuyển động dịch để di chuyển đầu từ trên hoặc dưới. Với một vài đặc trưng như vậy, người dùng cần chú ý bảo quản đĩa mềm theo các tính chất cấu tạo: tránh để gần các môi trường có từ tính cao, tránh nhiệt để làm hỏng nền nhựa, tránh ẩm vì có thể gây mốc làm che khả năng đọc ghi trên đĩa, tránh va chạm cơ khí làm gãy gập đĩa mềm...

**Đĩa cứng:** Đĩa cứng là thiết bị nhớ phụ phức hợp bao gồm từ 3 — 5 đĩa làm bằng nhôm cứng có phủ vật liệu nhạy cảm từ tính chồng lên nhau; giữa chúng là các đầu từ có bọc khí mỏng (không làm mòn đĩa) và bộ giao diện điện tử điều khiển việc ghép nối ổ đĩa với máy tính. Tất cả nằm trong một hệ cơ khí khép kín. Đĩa có dung lượng nhớ khá lớn thường gắn cố định trong máy tính. Các máy tính hiện nay thông dụng dùng đến 4 GB, 8 GB hoặc cao hơn.

**Đĩa CD:** là đĩa quang có thể mang đi lại đ-ọc. Đĩa CD có loại chỉ đọc nh- CD-ROM, đĩa có thể ghi đ-ọc CD-R hoặc CD-RW với nhiều loại tốc độ khác nhau và đang dần thay thế đĩa mềm. Dung l-ợng đĩa CD cao nên có thể đ-ọc dùng để ghi các ch-ơng trình cài đặt lớn, các ch-ơng trình nhạc, các ch-ơng trình hình ảnh, trò chơi v.v...

**Máy in (Printer):** là các thiết bị đ- a tín hiệu ra. Máy in có nhiều loại theo nguyên tắc truyền song song, nối tiếp của tín hiệu. Máy in cũng có thể phân loại theo ph-ơng thức in ký tự nh- loại va đập hay không va đập. Đối với loại va đập với ma trận điểm, máy in chia thành loại có nhiều điểm với độ liên tục giữa các điểm cao, loại khổ rộng (cho khổ giấy A3), loại khổ hẹp (cho khổ giấy A4). Loại không va đập có thể kể đến loại máy in Laser hoặc in phun. Các máy in đó lại có thể in màu hoặc đen trắng với các tốc độ in khác nhau.

**Các thiết bị khác:** có thể kể đến máy quét (Scanner), máy vẽ (Plotter) cũng là các thiết bị đ- a thông tin ra cho các cỡ giấy A0 đến A4. Hàng chục thiết bị khác của truyền thông, các thiết bị đo, điều khiển khác cũng đ-ọc ghép nối với máy tính để xử lý.

### ***b. Các phần mềm***

Phần cứng máy tính là một thực thể vật lý thuần túy. Để hoạt động, máy tính cần có các phần mềm điều khiển và các ch-ơng trình ứng dụng khác nhau. Có thể nói, phần mềm là các ch-ơng trình h-ớng dẫn phần cứng thao tác. Chúng ta có thể chia các phần mềm thông dụng hiện nay thành 3 loại: ***Phần mềm hệ thống, ngôn ngữ lập trình và ch-ơng trình ứng dụng***. Cách chia này cũng chỉ t-ơng đối theo một nghĩa nào đó.

#### ***Các phần mềm hệ thống***

Là các ch-ơng trình khởi động máy, tạo môi tr-ờng làm việc thuận tiện cho ng-ời dùng, quản lý các tài nguyên và điều khiển hoạt động của máy. Phần mềm hệ thống quan trọng nhất là hệ điều hành.

#### ***Các ngôn ngữ lập trình***

Ngôn ngữ lập trình là các công cụ tạo ra các ch-ơng trình ứng dụng. Có thể ví ngôn ngữ lập trình nh- các máy cái để tạo ra các máy gia công (ch-ơng trình ứng dụng) theo một nghĩa nào đó.

Ngôn ngữ lập trình có thể chia thành các ngôn ngữ cấp thấp và ngôn ngữ cấp cao. Ngôn ngữ cấp thấp, ví dụ ngôn ngữ Assembly là ngôn ngữ gần với các ngôn ngữ máy. Các ch-ơng trình viết bằng ngôn ngữ cấp thấp chạy nhanh hơn các ngôn ngữ cấp cao, tuy nhiên phức tạp và quá chi tiết. Ng-ọc lại các ngôn ngữ cấp cao nh- Pascal, C, Fortran, Cobol, Basic... đơn giản hơn về mặt thể hiện, gần gũi hơn với tiếng nói con ng-ời. Một số ch-ơng trình đ-ọc coi là ngôn ngữ vừa cao vừa thấp ví dụ nh- ngôn ngữ C. Ch-ơng trình đ-ọc viết ra từ các ngôn ngữ cao cấp phải qua các giai đoạn liên kết (link), biên dịch (Compiler) hoặc diễn dịch (Interpreter).

#### ***Các ch-ơng trình ứng dụng thông dụng***

Từ các ngôn ngữ lập trình, ng-ời ta có thể thiết kế và xây dựng các ch-ơng trình ứng dụng. Các ch-ơng trình ứng dụng rất nhiều. Sau đây là một số phần mềm đ-ọc dùng nhiều nhất

##### ***Các phần mềm soạn thảo văn bản***

WinWord trong gói phần mềm Microsoft Office là một phần mềm đ-ọc dùng phổ biến ở n-ớc ta trong những năm gần đây. Phần mềm này chạy trên môi tr-ờng Windows với các phiên bản khác nhau và nhiều tính năng về chèn ảnh, bảng biểu, sắp chữ, trang trí.

Các phần mềm soạn thảo tiếng Việt: có thể kể đến VNI sản phẩm kết hợp của n-ớc ngoài và Việt nam đ-ọc dùng nhiều ở các tỉnh phía nam. BKED (Bách Khoa Editor) của Ts. Quách Tuấn Ngọc chủ biên. Các phần mềm khác bao gồm VIETRES, VIETKEY, FREECODE, ABC... . Trong đó ABC đang đ-ọc khuyến cáo dùng nh- một phần mềm tiếng Việt chuẩn.

##### ***Trang tính điện tử (Worksheet) Excel, Lotus***

Mục đích phần mềm này là lập trang tính thuận tiện cho công tác kế toán, quản lý nhân sự.

## 1.3. Biểu diễn thông tin trong MTĐT

### 1.3.1. Các hệ đếm

#### a. Biểu diễn số trong các hệ đếm

Nh- ta đã thấy, các ô nhớ của máy tính chỉ có 2 trạng thái 0 hoặc 1, vì vậy các thông tin đ- a vào để máy tính xử lý phải mã hóa sang dạng nhị phân. Mặc dầu khi sử dụng các phần mềm ứng dụng ta chỉ làm việc với số thập phân nh- ng thực ra các ngôn ngữ lập trình đã làm giúp ta công việc chuyển đổi buồn tẻ. Để có thể hiểu rõ hơn hoạt động của máy tính và giúp ta dễ dàng hơn trong việc lập trình ta sẽ tìm hiểu một chút cách biểu diễn các số trong các hệ cơ số khác nhau. Một số  $K = a_n a_{n-1} a_{n-2} \dots a_1 a_0$  trong hệ thập phân đ- ọc biểu diễn d- ới dạng:

$$K = a_n \times 10^n + a_{n-1} \times 10^{n-1} + a_{n-2} \times 10^{n-2} + \dots + a_1 \times 10 + a_0$$

Ta có thể thấy rằng với một số tự nhiên  $H > 1$  cho tr- ớc và một số nguyên  $K$  bất kỳ ta luôn luôn có thể biểu diễn  $K$  d- ới dạng

$$K = a_n \times H^n + a_{n-1} \times H^{n-1} + a_{n-2} \times H^{n-2} + \dots + a_1 \times H + a_0$$

Trong đó  $a_i < H, i=0,1,2,\dots,n$

Nếu  $H > 10$  ta mã hóa các số 10, 11, ...,  $H-1$  bằng những ký tự, thí dụ  $A=10, B=11, \dots$  ta gọi  $K = a_n a_{n-1} a_{n-2} \dots a_1 a_0$  là số cơ số  $H$ . T- ơng tự với  $B$  là một số thập phân, ta cũng có thể biểu diễn  $B$  d- ới dạng

$$(1) \quad B = b_n \times H^n + b_{n-1} \times H^{n-1} + b_{n-2} \times H^{n-2} + \dots + b_1 \times H + b_0 + c_1 \times H^{-1} + c_2 \times H^{-2} + \dots + c_m \times H^{-m}$$

Và ta có biểu diễn số  $B$  theo cơ số  $H$  là

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

Mặc dù máy tính chỉ làm việc với các số nhị phân nh- ng việc biểu diễn các số d- ới dạng nhị phân th- ờng quá dài. Vì vậy những ng- ời lập trình th- ờng nhóm các số nhị phân lại và biểu diễn chúng ở hệ t- ơng ứng. Thí dụ nếu ta nhóm 3 số nhị phân ta đ- ọc số ở hệ cơ số 8, nhóm 4 số nhị phân đ- ọc số ở hệ cơ số 16...

Sau đây là bảng các hệ đếm hay dùng trong tin học:

Hệ đếm	Cơ số	Dạng ký số và ký tự
Nhị phân (Binary)	2	0, 1
Bát phân (Octal)	8	0, 1, 2, 3, 4, 5, 6, 7
Thập phân (Decimal)	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Thập lục phân (Hexadecimal)	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F (10, 11, 12, 13, 14, 15)

#### b. Chuyển đổi số giữa các hệ đếm

- Chuyển một số từ hệ cơ số  $L$  sang hệ cơ số  $H$ :

Ta l- u 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 phân và phần nguyên đều không đổi. Nghĩa là dù biến đổi sang hệ cơ số nào đi nữa thì phần phân cũng chỉ chuyển sang phần phân, phần nguyên sang phần nguyên.

Giả sử ta có một số có phần phân  $b = k + d$  trong hệ cơ số  $L$  trong đó  $k$  là phần nguyên tr- ớc dấu phẩy và  $d$  là phần 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 th- ơng số bằng 0, phép chia thứ  $i$  có số d-  $b_i$  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 phân:* Lấy phần phân của  $d$  nhân liên tiếp với  $H$  cho đến khi kết quả phép nhân không còn phần phân hoặc đạt đ- ợc 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 phân của số nhị phân cần tìm. (Chúng ta l- u ý là sau mỗi lần nhân ta chỉ lấy phần phân để nhân tiếp với  $H$ , phần nguyên ở đây đ- ọc hiểu là phần bên trái dấu chấm phân).

**Ví dụ:** 1. Cho số thập phân 14.125 tìm số nhị phân t- ơng ứng.

Ta có  $k=14, d=0.125$



### Chuyển đổi phần nguyên 14

	Chia 2	D-
14		0
7		1
3		1
1		1
0		

### Chuyển đổi phần phân 0.125

	Nhân 2	Phần nguyên
0.125		
0.25		0
0.5		0
1		1

Vậy  $14.125 = 1110.001$

### 1. Chuyển đổi 0.2 sang hệ nhị phân:

	Nhân 2	Phần nguyên
0.2		
0.4		0
0.8		0
1.6		1
1.2		1

...

Ta thấy rằng số 0.2 trong hệ cơ số 2 là một số phân vô hạn tuần hoàn

$$0.2_{10} = 0.(0011)_2$$

### • 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 d-, 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 liên nhau.

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

$$(11\ 110\ 101) \rightarrow 365 \rightarrow \text{trong hệ bát phân là số } 365$$

$$(1111\ 0101) \rightarrow 15\ 5 \rightarrow F5 \rightarrow \text{trong hệ thập lục phân là số } F5$$

Khi cần chuyển ng- ợc lại chúng ta làm theo các b- ớc t- ơng tự

### 1.3.2. Các phép toán cơ bản của số nhị phân

Khi định nghĩa một kiểu số bao giờ cũng phải định nghĩa các phép toán. Đối với số nhị phân, ta có 3 loại phép toán:

- + Phép toán số học nh- cộng, trừ, nhân, chia
- + Phép toán logic nh- AND, OR, NOT, XOR
- + Phép toán quan hệ nh- =, >, <, >=, <=

Sau đây là hai phép tính số học cơ bản nhất: 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

Với phép cộng,  $1 + 1$  sẽ là 10, tương đương với việc viết 0 nhớ 1'.

Điều đó hoàn toàn hợp lý vì 10 chính là số có giá trị bằng 2 nh- ng viết trong hệ cơ số 2. Hiểu theo cách khác, khi cộng hai chữ số, nếu giá trị v- ợt quá cơ số a thì ta phải viết phần lẻ và nhớ phần lớn hơn sang hàng bên cạnh.

**Qui tắc 1:** Khi nhân một số nhị phân với  $2^n$ , ta chỉ việc thêm n số 0 vào bên phải.

Ví dụ:  $1011 \times 2^3 = 1011000$

**Qui tắc 2:** Khi chia một số nguyên nhị phân cho  $2^n$ , hãy dịch dấu chấm ngăn phần nguyên và phần phân đi n chữ số sang bên trái.

Ví dụ:  $01011101 : 2^3 = 01011.101$

Phép trừ và phép chia có thể suy ra từ phép cộng và phép nhân nói trên

**Các phép tính logic đối với số nhị phân:**

Ngoài các phép tính số học mà ta từng quen biết, chúng ta phải làm quen thêm với phép tính logic đối với số nhị phân. Các phép tính logic cơ bản là.

- + Phép AND (phép Và)
- + Phép OR (phép Hoặc)
- + Phép XOR (phép Hoặc loại trừ)
- + Phép NOT (phép Phủ định)

Bảng hay OR,XOR,AND và NOT

x	y	x AND y	x OR y	x XOR y	NOTx
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

Phép AND, OR của số nhiều bit thì làm từng bit t- ơng ứng một. Thí dụ

	00110101		00110101
AND	11110001	OR	11110001
	<hr/>		<hr/>
	00110001		11110101

### 1.3.3. Mệnh đề logic

Mệnh đề logic là mệnh đề nhận giá trị logic (biểu thức số nhận giá trị số) nghĩa là chỉ nhận 1 trong 2 giá trị đúng (TRUE) hoặc sai(FALSE). Thí dụ mệnh đề "ở Việt nam mặt trời mọc ở phía Tây" là FALSE, còn mệnh đề "Người rồi cũng đến lúc phải chết" là mệnh đề đúng (TRUE)

D- ối đây là bảng chấm giá trị của các mệnh đề logic

A	B	A AND B	A OR B	A XOR B	NOT A
FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE	TRUE	FALSE
TRUE	TRUE	TRUE	TRUE	FALSE	FALSE

### 1.3.4. Biểu diễn dữ liệu

Máy tính số chỉ xử lý những dữ liệu ở dạng nhị phân. Do vậy các dữ liệu số, dãy ký tự, hình ảnh, âm thanh đều đ- ợc biểu diễn d- ới dạng nhị phân để máy tính có thể xử lý.

Đơn vị l- u trữ cơ bản của bộ nhớ gọi là từ (Word). Mỗi từ là một số cố định các Bit. Độ dài (số các Bit) của mỗi từ thay đổi tùy theo máy. Độ dài chung cho từ ở các máy th- ờng là 8, 16, 32, 64 Bit.



nguyên có phạm vi vượt quá giá trị của một từ của máy thì người ta phải sử dụng nhiều từ để lưu trữ một số nguyên.

### **Phương pháp dấu l-ong (sign - magnitude):**

Người ta còn dùng một cách biểu diễn khác cho số âm gọi là dấu l-ong. Theo cách biểu diễn này bit cực trái cũng được dùng làm dấu nh- ng quy - ớc lại ng- ọc lại với phương pháp dùng số bù 2: 1 là số d-ong, 0 là số âm. Để biểu diễn 1 số nhị phân với mẫu K bit theo phương pháp dấu l-ong ta lấy số cần biểu diễn cộng thêm  $2^{K-1}$  sau đó biểu diễn chúng ở hệ nhị phân.

Ví dụ với mẫu 4 bit:  $2^{K-1} = 8$

Với số + 5 ta biểu diễn bằng số  $5+8=13$  là 1101

Với số - 5 ta biểu diễn bằng số  $-5+8=3$  là 0011

### **Biểu diễn số thực**

Để biểu diễn phần phân của một số, trong hệ nhị phân ta cũng dùng một dấu chấm phân cách gọi là dấu chấm nhị phân. Ví dụ số 6.625 ở hệ thập phân ở hệ nhị phân sẽ có dạng

(1) 110.101

(2) hay  $0.110101 \times 2^3$

Cách biểu diễn (1) gọi là biểu diễn dấu phẩy cố định, còn cách biểu diễn (2) được gọi là biểu diễn dấu phẩy động.

Trong biểu diễn dấu phẩy cố định vị trí dấu phân cách phần nguyên và phần phân luôn ở một vị trí cố định, còn với biểu diễn dấu phẩy động vị trí dấu phân cách phần nguyên và phần phân không cố định.

Vì sao phải dùng cách biểu diễn dấu phẩy động? Các chương trình máy tính thường dùng các từ có độ dài 32 hoặc 64 bit lưu trữ các số thực. Ta thấy rằng giả sử ta cố định vị trí dấu chấm phân thì sẽ xảy ra trường hợp bit bị lãng phí trong khi đó số ta biểu diễn chưa đủ độ chính xác. Thí dụ nếu ta làm việc với các số lớn hàng trăm tỷ thì rõ ràng phần phân trở thành vô nghĩa. Còn nếu ta đang tính toán với những con số biểu diễn bán kính các nguyên tử thì phần bên trái của dấu chấm phân thành vô nghĩa, trong khi đó phần phân lại không đủ chính xác.

Chính vì vậy trong máy tính người ta biểu diễn số thực dưới dạng dấu chấm động (Floating Point) gồm một chuỗi các bit chia thành 3 nhóm như sau:

- Bit đầu tiên: gọi là bit dấu (qui - ớc 0 - d-ong, 1 - âm)
- Một số bit tiếp theo là phần mũ (exponent field)
- Các bit còn lại mô tả phần định trị (mantissa field).

**Phần mũ** là số nguyên và được biểu diễn bằng phương pháp dấu l-ong. Ta thấy ngay nếu quy - ớc 1 là d-ong thì sẽ tiện hơn cho việc so sánh 2 số mũ. Vì chữ số bên trái nhất của phần định trị khác 0, ở đây là 1, do đó số nào có số mũ lớn hơn sẽ lớn hơn. Dĩ nhiên là số d-ong (1) thì sẽ lớn hơn số âm (0).

**Phần định trị** có bit bên trái nhất là 1. Với phần định trị ta ghi ra và đặt dấu chấm ở phía bên trái nhất. Sau đó tùy thuộc vào số mũ ta có thể dịch sang phải hoặc sang trái một số bit để biết được số cần tìm. (Số mũ âm dịch sang trái, d-ong dịch sang phải).

Nếu độ dài từ là 32 hoặc 64 bit thì phần mũ thường là 8 bit. Độ dài các nhóm số mũ và định trị tùy thuộc vào từng máy và phần mềm tương ứng.

Thực ra ta có thể biểu diễn một số B bất kỳ dưới dạng dấu phẩy động trong hệ cơ số H :

Ta có thể viết  $B = mH^e$

Trong đó  $B^{-1} \leq m < 1$ , vậy m là một số chỉ có phần phân và chữ số đầu tiên bên phải dấu chấm phải khác 0.

Ví dụ:  $123.5678 = 0.12345678E+03$

↑                    ↑  
Phần định trị    Phần số mũ

m gọi là phần định trị, e là phần mũ.

## **Biểu diễn ký tự**

Nh- trong phần tr- ớc chúng tôi đã nhắc đến, hiện nay có hai bảng mã đ- ọc dùng phổ biến để mã hóa các ký tự (số, chữ, dấu), đó là bảng mã EBCDIC (The Extended Binary Coded Decimal Interchange Code) và bảng mã ASCII (The American Standard Code for Information Interchange). Bảng mã EBCDIC dùng từ 8 bit để mã hóa một số ký tự, do hãng IBM lập để dùng cho các máy tính lớn.

Bảng mã ASCII chuẩn dùng từ 7 bit để mã hóa một ký tự, ví dụ ký tự “A” trong bảng mã này có mã 1000 001. Tuy nhiên, với các máy IBM/PC hoặc t- ơng thích ng- ời ta dùng bản mã ASCII mở rộng dùng từ 8 Bit để mã hóa các ký tự.

Có sự khác nhau giữa mã của dữ liệu số và mã của dữ liệu ký tự. Một ký tự đ- ọc mã hóa bởi một và chỉ một mã nhị phân, các mã này là độc lập với nhau. Vì thế trong máy tính để chứa một ký tự ng- ời ta chỉ cần dùng một Byte của bộ nhớ nh- ng để l- u giữ một từ hay một câu hay một đoạn văn bản v.v... là một chuỗi liên tiếp các ký tự thì ta phải dùng một chuỗi liên tiếp các byte liên nhau.

Nh- vậy, mã của một ký tự hay của một chuỗi các ký tự đều ở dạng nhị phân nh- ng qui luật về giá trị vị trí của các mã này là không có ý nghĩa.

Với dữ liệu kiểu số với mỗi số phải dùng số byte trong phạm vi của số đó để mã hoá, thì mỗi bit của mã đều có một giá trị vị trí nhất định là: 1, 2, 4, 8... Khi đó, mã của mỗi số đều ở dạng nhị phân xác định giá trị số học của mỗi số ta phải dùng qui luật giá trị vị trí.

## **1.4. Mạng máy tính**

### **1.4.1. Tổng quan về mạng máy tính**

#### **a. Mạng (Network) là gì?**

**Mạng máy tính là tập hợp các máy tính nối lại với nhau (bằng các đ- ờng truyền vật lý theo một kiến trúc nào đó) sao cho các máy tính có thể khai thác tài nguyên của nhau nh- ch- ơng trình, dữ liệu, thiết bị.** Máy tính có thể nối lại với nhau theo nhiều cách, nh- ng điểm khác biệt chính của cách nối tạo thành mạng với cách nối khác là tạo môi tr- ờng làm việc thuận tiện giữa những ng- ời sử dụng, họ có thể trao đổi thông tin qua mạng, dùng chung dữ liệu, phần mềm. Thí dụ ta có thể dùng cáp nối cổng 2 máy tính và dùng phần mềm tiện ích NORTON để liên kết 2 máy. Lúc này ta có thể truyền các file dữ liệu từ máy này sang máy khác, nh- ng đó không phải là mạng. Bởi vì ta bắt buộc phải làm việc trong môi tr- ờng của NORTON. Ta không thể dùng chung máy in, dùng chung tài nguyên khi chạy một ch- ơng trình ứng dụng nào đó. Ta cũng có thể định nghĩa mạng thông tin theo quan điểm của lý thuyết tập hợp nh- sau:

Ta gọi  $E = \{e_1, e_2, \dots, e_n\}$  là tập hợp các phần tử xử lý thông tin,  $B = \{b_1, b_2, \dots, b_m\}$  là tập hợp các đ- ờng truyền. Mạng thông tin đ- ọc định nghĩa nh- sau:

$$M = \{E, B\}$$

Tùy theo đặc tính của E và B mà ta có các mạng thông tin khác nhau nh- mạng điện thoại, mạng truyền số liệu, mạng truyền hình. Mạng máy tính là tr- ờng hợp E là các máy tính còn B là các đ- ờng truyền số liệu.

Lợi ích của mạng:

- Mọi ng- ời có thể dùng chung thiết bị nh- máy in, ổ đĩa, ch- ơng trình: giảm chi phí mua các thiết bị chung.
- Dữ liệu chỉ cần nhập và l- u trữ trên một máy, nh- vậy sẽ tiện cho việc cập nhật sửa đổi, số liệu sẽ thống nhất và chính xác hơn.
- Rất tiện cho các nhóm lập trình viên cùng hoàn thành một công việc, phải th- ờng xuyên trao đổi thông tin cho nhau.
- Chỉ cần tập trung nâng cấp máy chủ, giảm tối đa cấu hình các máy trạm

#### **b. Lịch sử phát triển mạng**

- Từ những năm 60 ở Mỹ xuất hiện mạng xử lý trong đó các trạm cuối (Terminal) đ- ọc nối vào một máy xử lý trung tâm (MXLTT). Máy xử lý trung tâm làm tất cả mọi việc: quản lý các thủ tục truyền dữ liệu, quản lý sự đồng bộ của các trạm cuối, các hàng đợi cho đến quản lý các ngắt từ trạm cuối... Để giảm nhẹ nhiệm vụ của MXLTT, ng- ời ta thêm vào các bộ tiền xử lý (preprocessor hoặc còn gọi là frontal) để nối thành một mạng truyền tin trong đó có các thiết bị tập trung (concentrator) và dồn kênh (multiplexor) dùng để tập trung trên cùng 1 đ- ờng truyền các tín hiệu gửi tới từ trạm cuối. Sự khác nhau ở 2 thiết bị này là ở chỗ: bộ dồn kênh có khả năng chuyển song song các tín hiệu từ các trạm cuối gửi tới, còn bộ tập trung phải dùng bộ nhớ đệm (buffer) để l- u trữ tạm thời các thông tin.
- Từ đầu những năm 70 các máy tính đã đ- ọc nối với nhau trực tiếp để tạo thành mạng máy tính nhằm phân tán tải của hệ thống và tăng độ tin cậy.
- Cũng trong những năm 70 bắt đầu xuất hiện khái niệm mạng truyền thông (communication network), trong đó các thành phần chính của nó là các nút mạng đ- ọc gọi là các bộ chuyển mạch (switching unit) dùng để h- ớng thông tin tới đích của nó. Các nút mạng đ- ọc nối với nhau bằng đ- ờng truyền (transmission line) còn các máy tính của ng- ời dùng (host) hay các trạm cuối (terminal) đ- ọc nối trực tiếp vào các nút mạng để khi cần thì trao đổi thông tin qua mạng. Bản thân các nút mạng cũng th- ờng là các máy tính nên có thể đồng thời đóng cả vai trò máy của ng- ời sử dụng. Vậy ta có thể xem mạng truyền thông cũng là mạng máy tính. Mạng rất có ích về nhiều mặt nh- ta đã thấy ở trên. Tuy vậy lúc đầu giá của máy tính đắt hơn rất nhiều so với chi phí truyền tin nên ng- ời ta vẫn còn ít kết nối mạng. Phải từ thập kỷ 80 trở đi khi họ các máy tính cá nhân ra đời với giá thành hạ thì việc nối mạng mới đ- ọc thực hiện rộng rãi.

### c. Cấu thành của mạng máy tính

Mạng máy tính có thể chia thành hai yếu tố cơ bản nh- sau:

- **Đ- ờng truyền vật lý:** là môi tr- ờng chuyển các tín hiệu giữa các máy tính với nhau d- ới dạng xung nhị phân và th- ờng đ- ọc điều chế thành sóng điện từ nhất là khi truyền tải đi xa. Các xung điện từ đ- ọc truyền theo đ- ờng truyền hữu tuyến nh- : dây cáp đồng trục (coaxial), cáp đôi xoắn (twisted-pair) hay cáp sợi quang (fiber-optic) hoặc vô tuyến: Radio, sóng cực ngắn (microwave, viba), tia hồng ngoại (infrared)
- **Kiến trúc mạng:** cách thức kết nối các máy tính với nhau đ- ọc gọi là hình trạng (topology) của mạng và các quy tắc, quy - ớc mà các thành phần trên mạng phải tuân theo đ- ọc gọi là giao thức (protocol).

### d. Phân loại mạng máy tính

Có nhiều cách phân loại mạng khác nhau tùy thuộc vào yếu tố chính đ- ọc chọn để làm chỉ tiêu phân loại, chẳng hạn đó là "khoảng cách địa lý", "kỹ thuật chuyển mạch" hay "kiến trúc mạng".

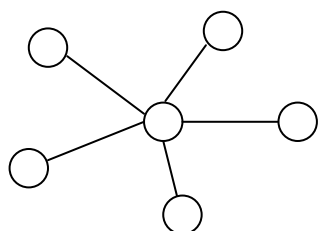
- **Phân loại theo khoảng cách địa lý, mạng máy tính gồm có:**
  - Mạng cục bộ (Local Area Networks viết tắt là **LAN**) là mạng máy tính đ- ọc tổ chức trong phạm vi **vài chục ki lô mét** trở lại, ví dụ mạng nội bộ cơ quan, tr- ờng học, xí nghiệp, văn phòng ...
  - Mạng đô thị (Metropolitan Area Networks viết tắt là **MAN**) là mạng máy tính đ- ọc 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 đ- ọc tổ chức trong phạm vi rộng, nh- mạng quốc gia, liên bang, châu lục .
  - Mạng toàn cầu (Global Area Network viết tắt là **GAN**) là mạng máy tính đ- ọc 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 đ-ợc chỉ định đích đến. Căn cứ vào thông tin đích đến các thông báo có thể đ-ợc truyền qua nhiều con đ-ờng khác nhau để đến đích.
- Mạng chuyển mạch gói (Packet-Switched Networks) là mạng trong đó thông báo cần gửi đi đ-ợc chia nhỏ thành các gói (packet) có số l-ợng bytes cố định. Mỗi gói tin có địa chỉ đích và đánh dấu thứ tự và có thể đi theo nhiều đ-ờng khác nhau để tới đích. Khi tới đích, chúng đ-ợc kết nối lại với nhau theo theo thứ tự đã đ-ợc đánh số.

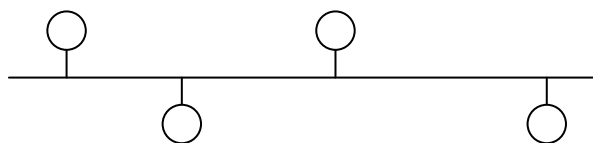
• *Phân loại theo kiến trúc mạng:*

Chúng ta th-ờng nghe nói đến mạng SNA của IBM, ISO của ISO hay mạng TCP/IP...

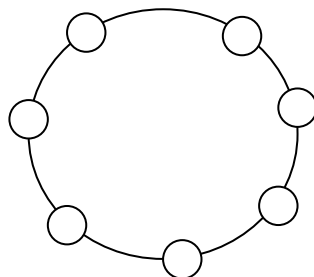
Trong mạng LAN và thậm chí cả mạng WAN ng-ời ta còn phân chia mạng theo TOPO mạng: Mạng hình sao(STAR), mạng hình BUS, mạng hình vòng. Ngoài ra còn một số dạng biến t-ơng khác nh- mạng hình cây (TREE), mạng hình sao vòng, mạng hỗn hợp...



**Hình sao (STAR)**



**Hình BUS**



**Hình vòng (RING)**

**Mạng hình sao:** **Ưu điểm:** Một máy trạm hỏng các máy khác vẫn hoạt động, cấu trúc đơn giản, thuật toán điều khiển ổn định, mở rộng thu hẹp dễ.

**Nh-ợc điểm:** Trung tâm có sự cố thì ng- ng hoàn toàn, tốn giấy.

**Mạng hình bus:** **Ưu điểm:** Một máy trạm hỏng các máy khác vẫn hoạt động, mở rộng thu hẹp dễ.

**Nh-ợc điểm:** Một điểm trên BUS hỏng thì ng- ng hoàn toàn, ph-ơng pháp thâm nhập mạng phức tạp, dễ chồng chéo số liệu.

**Mạng hình vòng:** **Ưu điểm:** Điều khiển đơn giản, ít khả năng chồng chéo thông tin.

**Nh-ợc điểm:** Một nút có sự cố thì ng- ng hoàn toàn.

***e. Hệ điều hành mạng***

Cùng với việc ghép nối các máy tính thành mạng, cần thiết phải có hệ điều hành trên phạm vi toàn mạng có chức năng quản lý dữ liệu, tính toán và xử lý một cách thống nhất. Các hệ thống nh- vậy đ-ợc gọi chung là hệ điều hành mạng (Network Operating System — viết tắt là NOS).

Một số hệ điều hành mạng đ-ợc dùng phổ biến hiện nay đ-ợc kể đến là hệ điều hành mạng Novel/Netware, Windows NT, Unix.

### **1.4.2. Mạng Internet/Intranet**

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

Năm 1968 Cục các dự án nghiên cứu tiên tiến (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 đ-ợc 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 đ-ợc 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 đ-ợc tách làm 2 : một gọi là MILNET dành cho các địa điểm quân sự, 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 đ-ợc số hóa và phân chia thành hàng loạt các gói để truyền đi, sau đó các gói này đ-ợc lắp ráp lại tại nơi nhận.*** Hệ thống các gói đ-ợc tạo ra nh- vậy nhằm đảm bảo cho thông tin đ-ợc 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 ng-ời nhận*** . Nh- 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 đ-ợc nh- 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 đ-ờng truyền là 1.5 Mb/s thay vì 560Kb/s trong ARPANet). Sự xuất hiện mạng x-ơng sống NSFNet đã thúc đẩy sự tăng tr-ởng của Internet. Một xa lộ thông tin mới hình thành và nhiều tr-ờng 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 v-ợt ra khỏi n-ớc Mỹ và trở thành mạng toàn cầu với vài chục triệu ng-ời dùng nh- 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 x-ơng sống của mạng Internet.

Về kiến trúc, Internet cũng đã có những thay đổi. Tr-ớc đây ng-ời 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 đ-ợc 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 đ-ờng (router) với giao thức sử dụng chủ yếu là TCP/IP.*** Hiện nay Internet bao gồm trên 15 000 mạng với khoảng từ 25 đến 30 triệu ng-ời sử dụng. Các dịch vụ chính trên mạng Internet là : th- 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),...

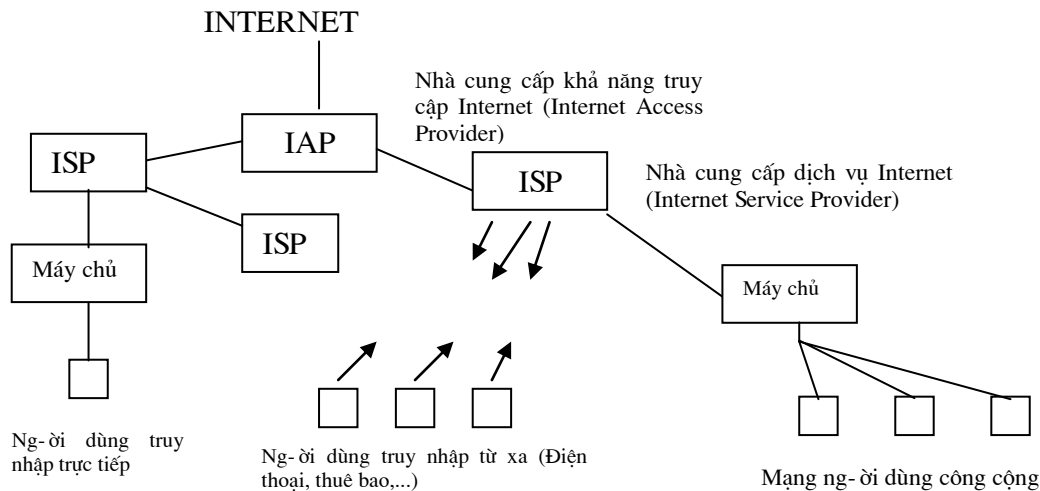
Bên cạnh khái niệm liên mạng Internet), khái niệm ***mạng nội bộ (Intranet)*** cũng xuất hiện. Intranet là mạng dùng riêng cho các yêu cầu hoạt động nội bộ của một đơn vị xã hội, có thể là mạng cục bộ hay mạng diện rộng, trong đó ***sử dụng các công nghệ cốt lõi của Internet.***



Một câu hỏi đặt ra: Ai là ng-ời 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 ng-ời sử dụng (host) nối vào Internet ban đầu do chính ISOC trực tiếp làm. Nh-ng từ năm 1992 do sự tăng tr-ởng quá nhanh của Internet nên công việc đó đ-ợc phân cấp cho các Trung tâm thông tin mạng viết tắt là NIC (Network Information Center). NIC của khu vực Châu □ - Thái Bình d-ơng - gọi là APNIC có trụ sở tại Tokyo, Nhật Bản. Hiện nay ở khu vực Châu á - Thái Bình D-ơng chỉ có 2 NIC quốc gia của Nhật bản và của Hàn Quốc. Hiện tại APNIC ở Tokio vẫn chịu trách nhiệm điều hành và phân phối địa chỉ cho các Host ở Việt nam.

### c. Sơ đồ cung cấp dịch vụ truy nhập Internet



### d. Các dịch vụ của Internet

#### ▪ Th- điện tử (E-mail)

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

Tên ng-ời sử dụng @ Tên đầy đủ của vùng (domain)

Ví dụ 1: [namng@hanoi.vnn.vn](mailto:namng@hanoi.vnn.vn)

Trong đó từ phải sang trái theo thứ tự

. vn: chỉ vùng địa lí; vn viết tắt của Việt nam. Phần cuối của tên vùng th-ờng định nghĩa cho hoạt động: EDU, NET, COM (đối với n-ớc Mỹ) hoặc từ viết tắt cho quốc gia.

. vnn: chỉ tên mạng dịch vụ Internet quốc gia đặt tại VDC (VietNam Net viết tắt là Vnn)

. hanoi: chỉ tên một máy chủ trong VDC

.namng: chỉ tên hộp th- của một Account truy nhập của thuê bao.

Ví dụ 2: [hungcntt@hotmail.com](mailto:hungcntt@hotmail.com)

Là dịch vụ th- điện tử miễn phí có máy chủ ở Mỹ.

Để gửi th- điện tử máy gửi th- và máy nhận th- không cần phải liên kết trực tiếp. Email là một dịch vụ kiểu l-u và chuyển tiếp. Th- điện tử đ-ợc chuyển từ máy này qua máy khác cho tới máy đích (giống nh- trong hệ thống b-u chính thông th-ờng: th- đ-ợc chuyển đến tay ng-ời nhận sau khi đã đi qua một số b-u cục trung chuyển). Mỗi ng-ời dùng (client) đều phải kết nối với một E-mail Server gần nhất (đóng vai trò b-u cục địa ph-ơng). Sau khi soạn thảo xong th- và đề rõ địa chỉ đích (ng-ời nhận), ng-ời sử dụng sẽ gửi th- tới E-mail Server của mình. E-mail Server này có nhiệm vụ chuyển th- đến đích hoặc đến một E-mail Server trung gian khác. Th- sẽ chuyển đến E-mail Server của ng-ời nhận và đ-ợc l-u tại đó. Đến khi ng-ời nhận thiết lập một cuộc gọi nối tới máy E-mail Server đó thì th- sẽ đ-ợc chuyển về máy của ng-ời nhận và đồng thời vẫn còn lại trên máy chủ cho đến khi máy chủ đ-ợc dọn dẹp. Giao thức truyền thống sử dụng cho E-mail của Internet là SMTP (Simple Mail Transfer Protocol). Cấu trúc của một th- điện tử th-ờng gồm 2 phần: phần đầu th- (header) và phần thân th- (body). Ví dụ

Date: Sat. 16 Nov. 1997

From: nguyen @ it-hut.edu.vn
Subject: Seminar on IT
To: join @ ux1.sco.uiuc.edu.us
Dear Join
I Would like to inform you about seminar on IT...

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

Đây là dịch vụ mới và mạnh nhất trên Internet. WWW đ-ợc 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 nh- văn bản, ảnh hay âm thanh. Để xây dựng các trang dữ liệu với các kiểu dữ liệu khác nhau nh- 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 đ-ợc đánh dấu bằng các thẻ (tag) HTML cho phép ng-ời sử dụng có thể đọc đ-ợc 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 th-ờng bao gồm các thành phần: Giao thức Internet đ-ợc 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 nh- sau:

<http://www.iist.unu.edu/home/unuiist/newrh/III/1/page.html>

<http://web.msu.edu/vincent/index.html>

Trong đó:

**http:**                **//web.msu.edu**                **/vincent/index.html**

Giao thức        Domain name của server        Tài liệu trên server

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

<gopher://gopher.msu.edu/...>

Nh- 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, ng-ời 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. □ 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 cả 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 tương 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, chẳng 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 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 ng-ời 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 nh- : nhóm tin nhạc cổ điển, nhóm tin về hội họa, nhóm tin về thể thao...Tên (địa chỉ) của các nhóm tin đ-ợc cấu trúc theo kiểu phân cấp. Nhóm rộng nhất sẽ đ-ợc 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 đ-ợc thu thập từ nhiều nguồn khác nhau. Ng-ời sử dụng t-ương tác với một server tin thông qua một ch-ương trình đ-ợc 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, t-ương 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 đ-ợc các thông tin mà mình quan tâm của nhiều ng-ời từ khắp mọi nơi, đồng thời có thể gửi thông tin của mình đi cho những ng-ời có cùng mối quan tâm.

## PHẦN II. HỆ ĐIỀU HÀNH VÀ CÁC PHẦN MỀM HỖ TRỢ

### 2.1. Hệ điều hành là gì?

Nh- ta đã biết trong phần tr- ớc, máy tính muốn hoạt động đ- ọc phải có ch- ơng trình hay còn gọi là phần mềm. Phần mềm chúng ta 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 đ- ọc 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 th- ờng 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 đ- a hàng loạt thiết bị nh- bộ nhớ, màn hình, máy in, ổ đĩa, bàn phím... vào trạng thái sẵn sàng làm việc. Ngay cả khi đang hoạt động các ch- ơng trình ứng dụng vẫn th- ờng xuyên cần đến sự trợ giúp của OS. OS nh- một ng- ời đầy tớ cần mẫn: chỉ cần ch- ơng trình ứng dụng rung chuông là anh ta có mặt ngay. OS có rất nhiều dịch vụ có đánh số thứ tự, các ch- ơng trình ứng dụng có thể gọi các dịch vụ này thông qua các lệnh ngắt t- ơng ứng. Những lệnh nh- mở file, đóng file, copy file... các ch- ơng trình ứng dụng đều gọi đến dịch vụ OS. Có rất nhiều thao tác của ng- ời sử dụng cần làm bên ngoài các ch- ơng trình ứng dụng: tạo các th- 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 ng- ời sử dụng với máy. *Nhiệm vụ của hệ điều hành là đảm bảo sự hoạt động trôi chảy của máy tính.* Dĩ nhiên là các nhà thiết kế phần mềm có thể làm theo cách là chỉ giữ lại một vài chức năng tối thiểu cho OS nh- kiểm tra máy và các thiết bị khi khởi động và đ- a hết các chức năng cơ bản của OS vào các phần mềm khác. Nh- ng nếu nh- thế thì các phần mềm ứng dụng sẽ rất cồng kềnh và phần mềm nào cũng chứa một phần khá lớn các chức năng giống nhau. Các nhà thiết kế đã chọn cách giải quyết thông minh hơn: những chức năng nào mà hầu hết các phần mềm đều phải dùng đến thì sẽ tập trung vào một phần mềm hệ thống chung. Nền tảng này sẽ là “sân chơi” cho các phần mềm khác. Và công việc của hệ điều hành là che dấu những chi tiết đáng chán và phức tạp, th- ờng xuyên dùng đến trong hoạt động của máy tính tr- ớc mắt ng- ời sử dụng.

Tóm lại chúng ta có thể hiểu hệ điều hành nh- 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 ng- ời 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 ch- ơng trình lo việc điều khiển hoạt động của máy tính và tạo môi trường để các phần mềm khác chạy đ- ọc.* Nếu một hệ điều hành sử dụng môi tr- ờng đĩa làm ph- ơng tiện l- u trữ thông tin thì đ- ọc gọi là hệ điều hành đĩa (DOS —Disk Operating System). Những chức năng chính của DOS là:

- Đọc và cho thi hành các dòng lệnh.
- Cung cấp toàn bộ các dịch vụ cần cho các ch- ơng trình ứng dụng nh- : quản lý đĩa, các tệp, bộ nhớ, các thiết bị ngoại vi.

**Phân loại hệ điều hành:** Ng- ời ta phân loại hệ điều hành theo khả năng thực hiện cùng lúc một hay nhiều ch- ơng trình hoặc khả năng quản lý một hay nhiều máy tính. Theo tiêu chuẩn thứ nhất ta có 2 loại hệ điều hành:

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

- HĐH đa nhiệm: tại một thời điểm có thể thực hiện nhiều ch-ơng trình (multitasking) và các ch-ơng 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 nh- 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 chính của MS-DOS

**a. Tập tin (File) :** Một tập hợp thông tin đ-ợc đặt tên và đ-ợc l-u trữ trên thiết bị nhớ phụ nh- đĩa cứng, đĩa mềm, đĩa quang, băng từ... đ-ợc gọi là tập. Thông tin chứa trong tập có thể là một ch-ơng trình hoàn hảo có thể chạy đ-ợc, có thể là một ch-ơng trình con, một cơ sở dữ liệu hay một văn bản do ng-ời dùng soạn ra nh- đơn từ, bản báo cáo. Thông tin trên đĩa để máy tính có thể gọi vào bộ nhớ thao tác luôn luôn đ-ợc l-u trữ d-ới dạng các tập. Tên tập gồm 2 phần: phần tên nhiều nhất là 8 ký tự và phần mở rộng nhiều nhất là 3 ký tự. Phần tên là bắt buộc phải có. Nếu có phần mở rộng thì 2 phần ngăn cách bởi dấu chấm (.). Tên tập gồm các chữ, số hoặc dấu gạch nối d-ới (\_). Ví dụ: THCS.DOC

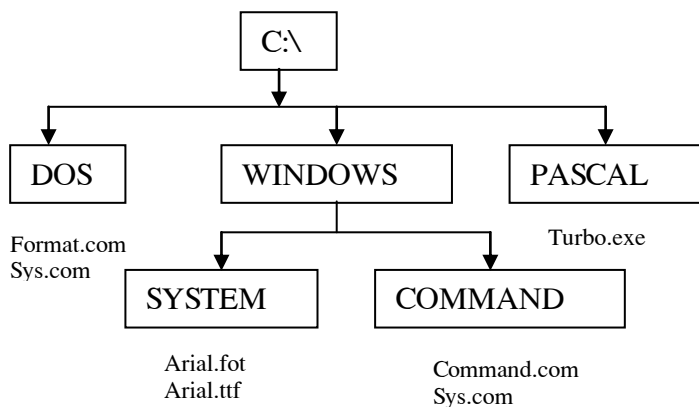
Trong các lệnh với tập DOS cho phép dùng 2 loại ký tự đại diện \* đại diện cho một nhóm ký tự, ? đại diện cho 1 ký tự. Ví dụ lệnh DIR C:\FOXPRO2\FOX\*.\* sẽ cho hiện lên màn hình tất cả các tập có 3 ký tự đầu tiên là FOX, còn lệnh DIR C:\FOXPRO2\VIDU?.PRG lại cho hiện lên màn hình các tập ch-ơng trình Foxpro có 4 ký tự đầu là VIDU, còn ký tự thứ 5 là bất kỳ.

**b. Th- mục:** Trong thực tế khi quản lý các hồ sơ tài liệu ng-ời ta th-ờng tổ chức sắp xếp chúng thành từng khu vực để tiện cho việc tìm kiếm hoặc tra cứu. Ví dụ trong th- viện ng-ời ta phân ra khu vực sách n-ớc ngoài hoặc trong n-ớc, sách khoa học kỹ thuật hoặc văn học, văn học dân gian hay văn học hiện thực... Ng-ời ta cũng áp dụng mô hình này trong việc quản lý các tập trên thiết bị nhớ. Thiết bị nhớ th-ờng đ-ợc phân thành các vùng, trong mỗi vùng có thể chứa các vùng con khác hoặc các tập. Mỗi vùng nh- vậy đ-ợc gọi là 1 th- mục. Tên th- mục cũng tuân theo quy tắc nh- tên tập.

Th- mục gốc là th- mục lớn nhất chứa toàn bộ ổ đĩa, tên của nó trùng với tên ổ đĩa và thêm dấu “\”. Ví dụ C:\

Các th- mục có thể lồng vào nhau. Ng-ời ta th-ờng tổ chức th- mục theo cấu trúc cây để tiện việc tra cứu. Các tập ở trong các th- mục khác nhau có thể trùng tên.

Ví dụ về cây th- mục:



**c. Đường dẫn:** Mỗi tệp được xác định bằng tên tệp và “con đường” dẫn đến tệp đó. Thí dụ:

Ta có tệp THCS.DOC nằm trong th- mục 99CM2, th- mục 99CM2 lại nằm trong th- mục TTDT1 ở ổ C:, “con đường” dẫn đến tệp thcs.doc là C:\TTDT1\99CM2\

Trong tr- ờng hợp trên đây ta gọi C:\TTDT1\99CM2\ là đ- ờng dẫn của THCS.DOC. Giả sử <tên tệp> đ- ọc chỉ ra nh- sau

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

trong đó th- mục sau là th- mục con của th- mục tr- ớc. Khi đó

[Tên ổ đĩa]\[tên th- mục 1]\[tên th- mục 2]\...\[Tên th- mục N]\ đ- ọc gọi là đường dẫn của <tên tệp>.

**d. Ổ đĩa và th- mục hiện thời**

Trong môi tr- ờng DOS ta sẽ thấy trên màn hình một dấu nháy sáng. Nếu ta gõ các ký tự từ bàn phím thì ta sẽ thấy các ký tự đó xuất hiện trên màn hình. Điểm sáng đó đ- ọc gọi là con trỏ DOS. Trên màn hình ta th- ờng thấy tên ổ đĩa và th- mục chứa con trỏ. Đó chính là ổ đĩa hiện thời và th- mục hiện thời của DOS.

**e. Khởi động và thoát khỏi DOS**

Trong phần sau chúng ta sẽ tìm hiểu cách setup để máy đọc DOS từ đâu. Th- ờng th- ờng máy sẽ tìm DOS theo thứ tự sau: đĩa mềm, đĩa CD-ROM, đĩa cứng. Sau khi DOS đã đ- ọc đ- a vào bộ nhớ để sẵn sàng làm việc dấu nháy sẽ xuất hiện trên màn hình chờ ng- ời sử dụng đánh lệnh vào. Những phần chính của DOS sẽ l- u trú trong bộ nhớ cho đến khi ta tắt máy.

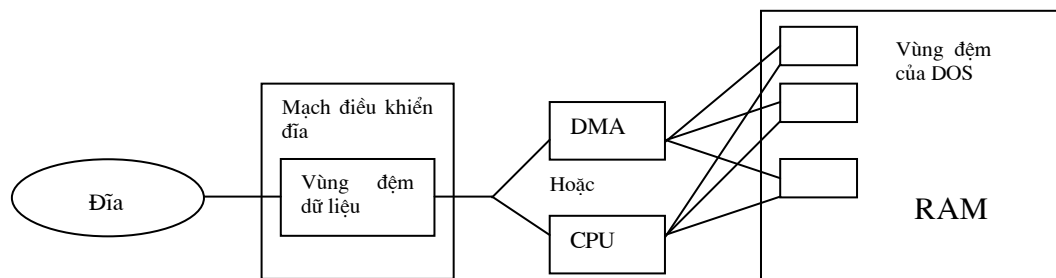
### 2.2.2. DOS tổ chức và l- u trữ số liệu trên đĩa nh- thế nào?

Mỗi HĐH có cách tổ chức và l- u trữ số liệu riêng. Thông th- ờng đĩa cứng mới sản xuất ra từ nhà máy ch- a thể dùng ngay đ- ọc mà phải trải qua 2 lần định dạng: định dạng cấp thấp và định dạng cấp cao. Với đĩa mềm thì đơn giản hơn: hoặc đĩa đã đ- ọc định dạng hoặc ta có thể dùng lệnh FORMAT để định dạng đĩa.

#### Cách tổ chức của đĩa mềm:

Đĩa mềm là một tấm nhựa hình tròn phủ ôxit sắt nâu và đ- ọc bảo vệ trong một vỏ giấy, trên vỏ giấy có khe hở để đầu từ đọc số liệu, một lỗ nhỏ để đánh dấu đ- ờng h- ớng tâm nơi bắt đầu của các từ đạo. Sau khi định dạng, mặt đĩa chia thành những vòng tròn đồng tâm (40 vòng hoặc hơn), mỗi vòng đ- ọc gọi là 1 từ đạo (track), mỗi từ đạo lại chia thành các cung từ (sector), mỗi cung từ gồm 512 byte trong đó có 2 byte dùng để ghi địa chỉ cung từ. Th- ờng từ đạo chứa 8 sector. Lệnh FORMAT của DOS là tạo ra các sector và đánh dấu địa chỉ lên chúng. Th- ờng ng- ời ta ghép 1,2,4 hoặc 8 cung từ thành 1 liên cung (cluster). DOS th- ờng dùng một số các liên cung để chứa các tệp tin, và khi đọc số liệu từ đĩa cũng đọc từng liên cung một (Nhiều máy còn đọc luôn cả track đ- a vào vùng đệm DOS trong RAM). Quá trình đọc số liệu từ đĩa mềm hay đĩa cứng đều theo cách sau đây:





Dữ liệu qua đầu từ đ-ọc truyền đến mạch điều khiển rồi đ-a vào vùng đệm (của mạch điều khiển). Khi số liệu đầy vùng đệm mạch điều khiển gửi tín hiệu đến CPU, CPU lúc đó mới đọc dữ liệu vào RAM để xử lý (Các máy AT đều tuân theo quy trình này, nh-ng các máy XT lại đọc dữ liệu trực tiếp thông qua DMA (Direct Memory Access)).

Ta có thể hình dung mặt đĩa từ là vô số các véc tơ nam châm nhỏ xíu nằm lộn xộn. Khi đầu từ bắt đầu ghi số liệu các véc tơ nam châm này đ-ọc sắp xếp lại dọc theo từ đạo: các véc tơ ng-ọc chiều nhau sẽ biểu thị các số 0 hoặc 1. Rãnh 0 mặt 0 là rãnh quan trọng nhất. Sector đầu tiên chứa record khởi động. Record khởi động chứa các thông tin cần thiết về đĩa để DOS có thể sử dụng đ-ọc đĩa đó. Nó cho biết kích th-ớc của 1 cung từ, số cung từ trên mỗi từ đạo, số cung từ trong 1 liên cung (hay là 1 trang) kích th-ớc của FAT và th- mục gốc. Ngoài ra mẫu tin mỗi còn chứa 1 ch-ơng trình nhỏ xíu. Ch-ơng trình này sẽ tìm và cho chạy phần tiếp theo của DOS, nếu DOS đ-ọc khởi động từ đĩa mềm và đĩa đang sử dụng là đĩa khởi động DOS. Tiếp theo record khởi động là bảng FAT th-ờng chiếm 2 sector hoặc hơn. Bảng FAT điều khiển việc đặt vùng nhớ cho các file. Vì bảng FAT chứa những dữ liệu vô cùng quan trọng nên th-ờng có 2 bản FAT, một chính thức và một dự phòng trên đĩa.

Bảng định vị file (FAT) của đĩa mềm th-ờng dùng 12 bit tức là một số Hexa 3 chữ số để đánh số các liên cung.

Bảng FAT12

Số đề mục FAT	Giá trị FAT (HEXA)	Chức năng
0	FFE	Đĩa 1 mặt
1	FFF	(Hàng nạp)
2	3	Phần tiếp theo của file ở liên cung 3
3	7	Phần tiếp theo của file ở liên cung 7
4	0	Không chứa gì, có thể sử dụng
5	0	Không chứa gì, có thể sử dụng
6	FF7	Liên cung có lỗi, không dùng đ-ọc
7	FFF	Kết thúc file

Dấu hiệu kết thúc file : FF8 -> FFF

Sau bảng FAT là bảng th- mục (**DIRECTORY TABLE**) chứa 7 sector hoặc hơn.

Bảng th- mục chứa danh sách các file hoặc các th- mục con có thể tìm thấy trên đĩa. □ đây hệ thống l- u trữ mọi thông tin liên quan đến các file. Mỗi đề mục dài 32 byte, vậy mỗi sector chứa đúng 16 đề mục. Mỗi đề mục gồm 8 phần

	1	2	3	4	5	6	7	8
Tên mục	Tên tệp	Phần mở rộng	Kích th- ớc tệp	Ngày tháng	Thời gian	Số liên cung bắt đầu	Thuộc tính	Ch- a dùng
Số byte	8	3	4	2	2	2	1	10

Các mục khác các bạn có thể dễ dàng ghi nhận từ bài giảng, ở đây chúng tôi giải thích chi tiết hơn 1 chút về mục 3 (thuộc tính) và mục 7 (liên cung bắt đầu).

*Thuộc tính của tập tin* là phần chứa các thông tin về đặc tính của tập tin. Mục này DOS dùng để phân biệt 6 loại thuộc tính của 1 tệp hoặc th- mục:

- Thuộc tính “lưu trữ” (Archive): 1 nếu tệp tin ch- a đ- ọc chép dự phòng (Backup - ch- ơng trình backup của DOS), 0 nếu đã đ- ọc chép.
- Thuộc tính “ẩn dấu” (hidden): là 1 thì tệp tin không hiện lên màn hình bằng hầu hết các lệnh của DOS.
- Thuộc tính “chỉ đọc” (Read-only) là 1 thì chỉ đọc.
- Thuộc tính “hệ thống” (System): là 1 báo cho DOS biết đây là tệp tin của hệ điều hành.
- Thuộc tính “nhãn” (Label): là thuộc tính đặc biệt. Nó chỉ ra rằng mục 1 và 2 không chứa tên tập tin mà là tên ổ đĩa. Vì thế ta cũng có thể hiểu vì sao ta có thể gán cho đĩa mềm hay cứng 1 cái tên dài 11 ký tự.
- Thuộc tính “thư mục con” (Subdirectory): Cho biết mục 1 và 2 không phải tên tập tin mà là tên th- mục con. Khi này kích th- ớc tệp đ- ọc đặt bằng 0.

*Số liên cung bắt đầu* là số thứ tự của liên cung đầu tiên DOS bắt đầu l- u trữ tệp. Thí dụ tệp THCS.DOC có các mục sau:

THCS	DOC	127	08/31/00	10:23:46	104		
------	-----	-----	----------	----------	-----	--	--

Nh- vậy liên cung bắt đầu là 104, DOS sẽ đọc liên cung thứ 104, sau đó quay lại bảng FAT đọc con số ghi trên liên cung thứ 104 để biết liên cung tiếp theo, DOS đọc liên cung đó rồi lại quay lại FAT xem liên cung đứng sau đó, cứ nh- thế cho đến khi gặp một số Hexa từ FF8 -> FFF, là dấu hiệu liên cung kết thúc tệp.

Khi ta xóa tệp thực ra DOS vẫn để nguyên các thông số trong bảng FAT và bảng th- mục. DOS chỉ ghi lên byte đầu tiên của phần chứa tên tệp số Hexa E5 (ký tự 229) báo rằng vùng chứa tập tin bị xóa có thể dùng để l- u trữ các tập tin khác. Ngay sau đó ta có thể khôi phục lại tệp vừa xóa bằng lệnh UNDELETE, khi đó DOS sẽ hỏi ta ký tự đầu tiên của tệp là gì, bởi vì chỉ có nội dung byte đầu tiên là bị mất mà thôi.

Th- mục gốc chỉ nằm trên rãnh 0 do đó số l- ợng các tệp có thể tạo ra trong th- mục gốc bị hạn chế bởi số cung từ trong rãnh 0.

### **Cách tổ chức của đĩa cứng:**

Để chứa đ-ợc nhiều thông tin hầu hết đĩa cứng đều gồm nhiều đĩa chồng lên nhau có chung một trục quay và quay đều với tốc độ 60vòng/s nh- nhau nhờ một động cơ. Mỗi đĩa đều chứa thông tin cả 2 mặt và mỗi mặt có một đầu từ riêng.Hệ thống đầu từ lồng vào chồng đĩa và tủy áp sát vào bề mặt đĩa nh- ng không bao giờ chạm nhờ một lớp đệm không khí đ-ợc tạo ra khi đĩa quay. Mặt đĩa ngày nay đã đ-ợc phủ bằng một lớp crom cực mỏng nhờ sự bốc hơi crom mà không cần chất kết dính.

Đĩa cứng phải qua 2 lần định dạng mới có thể dùng đ-ợc. Lần thứ nhất là **định dạng cấp thấp** (ĐDCT) nhờ vào một ch- ơng trình chuyên dụng nh- HFORMAT hay dùng chức năng ĐDCT trong SETUP. *Khi ĐDCT bề mặt đĩa sẽ đ-ợc kiểm tra. Những phần đĩa xấu sẽ đ-ợc đánh dấu cấm sử dụng. ĐDCT sẽ tạo nên các rãnh (track) và đánh dấu các cung từ (sector), đồng thời sẽ dùng một vài byte ở mỗi cung từ để ghi thông tin xác định cung từ đó.* Phần này gọi là tiêu đề cung từ (Sector ID Header) gồm một địa chỉ tuyệt đối 3 chiều (C,H,S). C - Cylinder chỉ từ trụ, tức là các vòng tròn đồng tâm tạo nên mặt trụ. H - Head chỉ mặt đĩa hay đầu từ t- ơng ứng, S - Sector chỉ cung từ. Nh- ng trong thực tế DOS lại đánh địa chỉ cung từ bằng một số duy nhất gọi là địa chỉ t- ơng đối, đó chính là số thứ tự của cung từ. Lần l- ợt từ Cylinder thứ nhất, head 0, track 0,head 1,track 0,head 2,track 0,...

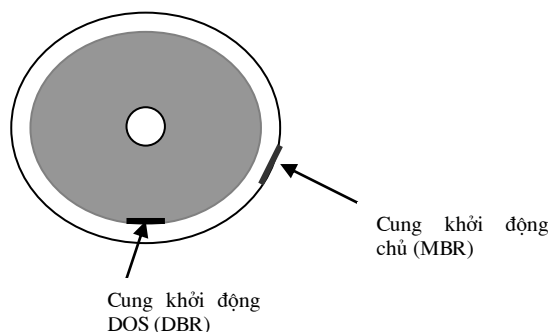
Từ đạo của đĩa cứng th- ờng chứa 17 cung từ. Nh- vậy ta có thể hiểu đ-ợc vì sao th- mục gốc của ổ đĩa cứng chỉ chứa đ-ợc khoảng 512 đề mục (Điều này không hạn chế trong th- mục con).

Khi định dạng cấp thấp máy sẽ hỏi ta nhập vào hệ số đan xen (interleave Factor) nhằm làm khớp tốc độ quay của đĩa từ với tốc độ mà đầu từ có thể xử lý dữ liệu khi chúng đi qua đầu từ. Ta nói rằng đĩa cứng có hệ số đan xen n: 1 hay n. Khi đó các sector chứa dữ liệu chỉ nối tiếp nhau về mặt logic chứ không nối tiếp về mặt vật lý.

Sau khi định dạng cấp thấp ta còn phải **chạy ch- ơng trình FDISK để phân chia đĩa cứng thành các khu vực:**

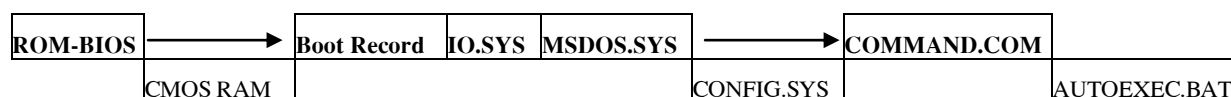
- *Phân khu DOS (Primary DOS partition)*
- *Phân khu DOS mở rộng (extended DOS partition)*
- *Phân khu phi DOS (non-DOS partition)*

Trong thực tế ở Việt nam ta chỉ dùng các khu vực DOS mà thôi. Ch- ơng trình phân khu sẽ cho ta định nghĩa các khu vực bắt đầu từ liên cung nào đến liên cung nào. Các thông tin “chia cắt” này sẽ đ-ợc ghi lại trong bảng phân khu (partition table). Cung từ khởi động chủ cũng đ-ợc tạo ra (Master Boot Record) Chỉ sau khi việc phân khu đã hoàn thành ta mới dùng lệnh FORMAT để định dạng cấp cao cho từng phân khu một. Thực ra mỗi khu gồm các từ trụ liền nhau nên chúng tạo thành những “vành đai” khác nhau quanh đĩa. **Định dạng cấp cao sẽ tạo ra cung từ khởi động DOS (DOS Boot Record), tạo bảng cấp phát tập tin (FAT), bảng th- mục gốc và sao chép các tập tin hệ thống vào đĩa cứng (IO.SYS,MSDOS.SYS và COMMAND.COM)**



### 2.2.3. Các modul của MS-DOS

Quá trình khởi động máy tính cài đặt MS-DOS gồm các bước sau:



Nếu không kể đến các chương trình trong ROM BIOS là phần được gắn với máy và Boot Record là một chương trình nhỏ chỉ đủ để đổ các chương trình cơ bản của DOS vào bộ nhớ, ta có thể xem DOS gồm 4 module lưu trên đĩa:

- IO.SYS, chứa các chương trình điều khiển thiết bị mà DOS sẽ cho nạp khi bật máy.
- MSDOS.SYS chứa các chức năng cơ bản và được gọi là DOS kernal (lõi của DOS).
- COMMAND.COM có nhiệm vụ đọc các lệnh đánh vào, diễn dịch và cho thi hành.
- Các tệp tin khác như các chương trình điều khiển thiết bị, các tệp tin lệnh ngoại trừ.

Do vậy từ đây về sau khi nói đến DOS ta sẽ hiểu là 4 module này.

### Hệ thống vào ra cơ bản ROM-BIOS (Basic Input/Output System)

Trên bảng mạch chính ta có thể thấy các IC ROM chứa **ROM-BIOS**. Ngoài ra vẫn còn một số để ROM còn bỏ trống để dùng cho sự mở rộng sau này. Các chương trình trong ROM đã được các nhà sản xuất cài sẵn bằng phương pháp đặc biệt, không phải nuôi bằng pin và không bị mất kể cả khi mất điện. ROM thường chiếm vùng địa chỉ cao nhất, dưới đó là RAM.

Môđun này chứa các chương trình vào ra cơ bản.

Ta có thể hình dung BIOS như là chất keo gắn phần cứng máy tính với hệ điều hành. Ta có thể thấy 1 phiên bản của DOS có thể chạy trên nhiều loại máy khác nhau. Đó là vì DOS được thiết kế để làm việc chủ yếu với BIOS.

BIOS gồm 3 phần:

- Một tập hợp các chương trình con cho phép DOS và các phần mềm khác liên hệ được với các thiết bị gắn với PC.
- Những chương trình con (routine) giải quyết các dịch vụ ngắt phần cứng (hard interrupt)
- Chương trình tự kiểm tra khi bật máy (POST = Power On Self Test, kiểm tra bộ nhớ, các thiết bị ghép nối với PC như ổ đĩa cứng, ổ đĩa mềm, bàn phím...).

Trong ROM-BIOS có chứa một số thông số ngầm định về các loại ổ đĩa, ổ đĩa khởi động, màn hình, ngày giờ... Tuy nhiên nếu ta muốn lắp thêm thiết bị hay muốn thay đổi ổ đĩa khởi động thì ROM-BIOS không thể tự nhận biết đ-ợc. Để giải quyết vấn đề này các nhà thiết kế máy tính đã lắp thêm trong bảng mạch chủ một IC gọi là RAM CMOS (Complementary Metal-oxide Semiconductor). CMOS đ-ợc nuôi bằng pin và chứa các thông tin về ổ đĩa, ngày giờ ... Trong BIOS th-ờng có một ch-ơng trình nhỏ có tên là SETUP có thể giúp ta sửa các thông tin trong CMOS. Khi khởi động máy tính màn hình sẽ hiện thông báo để ta biết có thể nhấn phím nào đó (thí dụ phím DEL) để chạy ch-ơng trình SETUP. BIOS sẽ đọc các thông tin trong CMOS và thay cho các thông số của nó.

Tiếp theo ROM-BIOS đọc 512 byte của thẻ khởi động (Boot Record) vào bộ nhớ và chuyển điều khiển cho Record khởi động.

### **Boot Record:**

Nhiệm vụ của Boot Record: Kiểm tra trong th- mục của đĩa xem có các tệp hệ thống IO.SYS, MSDOS.SYS không? Hai tệp hệ thống này đều đ-ợc bảo vệ chống xoá, chống ghi đè, mang dấu hiệu hệ thống và đ-ợc ẩn dấu.

Nếu DOS tìm thấy thì đổ hai tệp IO.SYS, MSDOS.SYS xuống bộ nhớ trong, sau đó chuyển điều khiển cho IO.SYS

### **Ch-ơng trình cơ sở của các ngoại vi: Tệp hệ thống IO.SYS**

Trong ba tệp hệ thống của DOS, tệp IO.SYS là sự mở rộng của ROM-BIOS. Cả ROM-BIOS và IO.SYS đều chứa các ch-ơng trình con điều khiển và xử lí các thiết bị vào ra ngoại vi. Nh-ng tại sao lại cần chia thành hai modul: một ch-ơng trình thì ghi cứng trong ROM-BIOS, ch-ơng trình kia lại để trong tệp ghi lên đĩa mềm? Đây là một đặc tr-ng khá khôn ngoan, mềm dẻo của các chuyên gia Microsoft nhằm dành một phần ch-ơng 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 quan trọng nhất: Phục vụ yêu cầu riêng đặc biệt của từng hệ điều hành. Các hệ điều hành khác DOS nh- 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 ch-ơng trình hệ điều hành khác nhau, để sửa đổi và cải tiến, nâng cấp.

- Sửa lỗi xuất hiện trong ROM-BIOS. Mặc dù ng-ời ta đã kiểm tra cực kì cẩn thận các ch-ơng 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 th-ờng 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 ch-ơng trình từ ROM-BIOS. (Hệ thống gọi các ch-ơng trình liên quan đến các ch-ơng 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 ch-ơng trình mới, nh-ng sử dụng ph-ơng pháp vừa nêu có -u điểm hơn.

- Nhiệm vụ thứ 3: Điều khiển các ngoại vi mới đ-ợc nối với máy tính PC (ví dụ các đĩa loại mới nh- đĩa mềm dung l-ợng cao, đĩa CD-ROM..), máy vẽ, máy quét. Việc quản lí và điều khiển các thiết bị mới dễ giải quyết nếu các nhà lập trình hệ thống sửa IO.SYS hơn là sửa ch-ơng trình trong ROM-BIOS. Cùng với việc xuất hiện các thiết bị vào ra mới có trên thị tr-ờng nối với máy tính PC, ng-ời ta phải liên tục sửa các tệp IO.SYS (thậm chí sửa cả ch-ơng 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 nh- ng với ng- ời sử dụng thì không thể làm đ- ợc vì khả năng và vì sự thiếu thông tin hệ thống th- ờng đ- ợc giữ bí mật bởi các nhà sản xuất.

Tuy nhiên, để đạt đ- ợc sự thành công trong việc chiếm lĩnh thị tr- ờng hệ điều hành thì các nhà sản xuất phải chiều ý khách hàng để cho ng- ời dùng khả năng tự mở rộng, thay đổi cấu hình máy tính PC ghép nối với ngoại vi. Khi nhận thức đ- ợc điều này kể từ DOS 2.0 trở đi, ng- ời 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, ng- ời dùng dễ dàng đ- a 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 ch- ơng 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 ch- ơng trình điều khiển thiết bị khi đặt trong CONFIG.SYS mà đ- ợc gọi đến sẽ đổ vào bộ nhớ trong của máy tính PC nh- là các ch- ơng trình phụ lục của IO.SYS. Theo cách đó, hệ thống máy tính có thể phát triển theo kiểu môđun với những ngoại vi mới mà không cần động vào các tệp hệ thống của DOS. Tuy nhiên, thông th- ờng bất kỳ một ch- ơng trình nào chạy trên IBM-PC đều cần đến kiểu DOS truyền thống với các thao tác cơ sở vào ra.

### **Tệp hệ thống MSDOS.SYS**

Trong hai tệp hệ thống thì tệp thứ hai này đóng vai trò trung tâm bao gồm các ch- ơng trình con dịch vụ còn lại của DOS ngoài chức năng bảo đảm vào ra. Ng- ời ta có thể ghép IO.SYS với MSDOS.SYS thành một tệp nh- ng để đảm bảo môđun hoá nên chúng đ- ợc chia riêng để dễ sửa đổi.

Việc sử dụng các ch- ơng trình con của hai nhóm này đ- ợc thực hiện bởi ngắt mềm. Mỗi một dịch vụ DOS th- ờng là một ch- ơng trình con và ứng với nó đ- ợc đặt tên thành một hàm của DOS nhằm thực hiện một số các dịch vụ trực tiếp và cơ bản nh- :

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, th- ờng chứa các thao tác con mang tính logic nh- :

Mở đóng các tệp trên đĩa,

Tìm bảng th- mục,

Xoá, tạo tệp song song với việc viết, đọc các số liệu.

☐ mức lập trình, ng- ời lập trình có thể sử dụng các dịch vụ DOS thay vì phải viết ch- ơng trình để thực hiện các thao tác hay dịch vụ đã nêu.

Các ch- ơng trình DOS ở mức cao hơn th- ờng làm nh- vậy (ví dụ các lệnh DIR, COPY của DOS thực hiện dịch vụ tìm bảng th- mục).

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

### **Ch- ơng 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. Nếu ta gõ một lệnh từ bàn phím và nhấn {ENTER} để gửi vào máy thì COMMAND.COM sẽ đọc lệnh này và quyết định sẽ làm gì với lệnh đó. Command.com sử dụng một bảng bên trong nó để chứa tên gọi của các lệnh nội trú (ta có thể quan sát bằng ch- ơng trình Debug hoặc Disk Look). Khi ta gõ lệnh tr- ớc 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 t- ơng ứng ở ổ đĩa bên ngoài trong th- mục hiện thời hoặc trong các th- mục đã đặt đ- ờng dẫn. Ba loại đuôi ch- ơng trình mà DOS có thể thực hiện theo thứ tự - u tiên là: .COM, .EXE và .BAT.

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

Các lệnh nội trú là các lệnh mà ng- ời dùng hay dùng nh- **CLS, DIR, CD, MD, RD, DEL, COPY, TYPE, DATE, TIME, VER, VOL, PROMPT, REN, ...** Chính vì lý do đ- ọc dùng th- ờng xuyên nên ng- ời ta bố trí chúng trong COMMAND.COM và l- u trú ở bộ nhớ trong để tiết kiệm thời gian tìm kiếm trên đĩa.

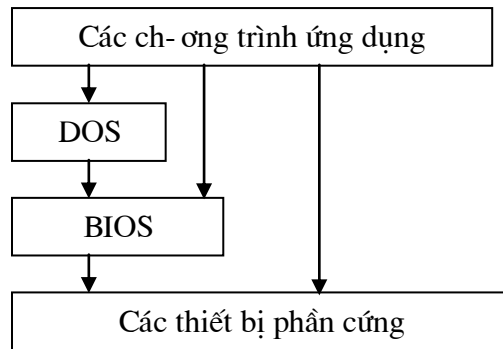
**Chức năng thứ hai** của COMMAND.COM chỉ hoạt động khi chúng ta bật máy là kiểm tra xem trên đĩa có tệp Autoexec.bat hay không? Nếu có thì nó chạy các lệnh trong tệp đó tr- ớc khi đ- a ra dấu nhắc của hệ thống. Khi nó làm xong các nhiệm vụ này, nó sẽ xóa các ch- ơng trình vừa chạy xong khỏi bộ nhớ.

**Chức năng thứ ba** của COMMAND.COM là kiểm tra sự tồn tại của COMMAND.COM trên đĩa và kiểm tra phân giải nghĩa lệnh có ở bộ nhớ hay không và nếu không có thì đổ từ đĩa xuống. Mặc dầu Command.com đ- ọc l- u trú trong bộ nhớ nh- ng khi cần thiết một phần của nó có thể bị ch- ơng trình khác ghi đè lên, do vậy Command.com phải có một phần luôn luôn l- u trú và kiểm tra sự tồn tại của bộ giải nghĩa lệnh và các lệnh nội trú.

### **Các lệnh ngoại trú**

Các lệnh còn lại của hệ điều hành đ- ọc bố trí trên đĩa ở dạng các tệp ch- ơng 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 ng- ời 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 nh- FORMAT, FDISK. Một quan niệm khác, thì các lệnh ngoại trú chỉ là các ch- ơng trình làm nhiệm vụ nh- các công cụ thông th- ờng, nh- các ch- ơng trình khác do DOS quản lý; mỗi khi gọi đến, nó đ- ọc đổ xuống bộ nhớ trong và chạy, sau đó có thể đ- ọc các ch- ơng trình gọi sau đó ghi đè lên.

Th- ờng thì các ch- ơnh trình ứng dụng phải thông qua HĐH trong mọi việc, nh- ng cũng có tr- ờng hợp chúng có thể bỏ qua DOS hoặc BIOS và thâm nhập thẳng vào các thiết bị phần cứng:



#### 2.2.4. Một số lệnh thông dụng của DOS

##### a. Tổng quan về lệnh của DOS

Lệnh DOS đ-ợc gõ vào từ bàn phím từng ký tự một, ta có thể quan sát để kiểm tra xem đã gõ đúng cú pháp của lệnh ch- a. Ta có thể dùng phím BACKSPACE để xóa ký tự vừa gõ. Khi ta gõ xong lệnh phải nhấn thì lệnh mới đ-ợc thi hành. Nếu lệnh gõ sai hoặc không tìm thấy trên đĩa DOS sẽ thông báo “ Bad command or file name”.

Nếu ta muốn gõ lại một lệnh của DOS tr-ớc đó ta chỉ cần nhấn F3.

Tập lệnh của DOS chia làm 2 loại: **lệnh nội trú và ngoại trú**

- Lệnh nội trú (Internal hoặc Resident) là những lệnh nằm trong tệp COMMAND.COM đ-ợc nạp vào bộ nhớ khi ta khởi động máy. Theo một nghĩa khác đó là những lệnh mà ng-ời dùng th-ờng sử dụng nh- DIR, CD, COPY, DEL, DATE, TIME. Vì th-ờng dùng nên cần đ-ợc tổ chức nh- đã nói ở phần trên sao cho nó có sẵn để MS-DOS chạy nó một cách nhanh nhất.

- Lệnh ngoại trú (External hay Inresident) là các lệnh ở dạng các tệp trên đĩa (đĩa cứng hoặc mềm). Khi ta gõ tên lệnh và nhấn DOS mới tìm trên đĩa, chuyển từ đĩa xuống bộ nhớ sau đó mới chạy (vì vậy lệnh ngoại trú chạy chậm hơn lệnh nội trú). Th-ờng ng-ời ta tạo th- mục C:\DOS và chứa các tệp lệnh ngoại trú tại th- mục này.

##### **Dùng trợ giúp để xem cú pháp lệnh**

MS-DOS có trợ giúp trực tuyến để tra cứu lệnh (nh- một số tay). Muốn biết cú pháp đầy đủ và cách dùng một lệnh nào đó ta gõ

HELP <tên lệnh cần trợ giúp>

##### b. Các lệnh nội trú:

Lệnh nội trú là một phần của tập tin COMMAND.COM, luôn th-ờng trú trong bộ nhớ chính (RAM), do đó chạy nhanh hơn lệnh ngoại trú.

Lệnh nội trú đ-ợc chia làm 4 nhóm liên quan tới: ổ đĩa, th- mục, tệp và hệ thống.

Sau đây chúng tôi chỉ đ- a ra cú pháp lệnh, các bạn hiểu là còn phải nhấn thì lệnh mới đ-ợc thực hiện.

##### **Lệnh về ổ đĩa:**



- Chuyển ổ đĩa: chuyển đổi ổ đĩa hiện thời.  
Cú pháp: <ổ đĩa:>  
Ví dụ: chúng ta đang ở ổ đĩa C, dấu nhắc có dạng C:\> ta muốn chuyển sang ổ A:, gõ A: ta sẽ thấy trên màn hình bây giờ là A:\>

### **Lệnh về th- mục:**

- **DIR** liệt kê danh sách các th- mục con và tệp.  
Cú pháp: **DIR** [đ- ờng dẫn] [/tham số]  
Ví dụ: chúng ta đang ở ổ đĩa C, dấu nhắc có dạng C:\> ta muốn xem danh sách các tệp và th- mục trong ổ C: gõ DIR. Lệnh DIR có nhiều tham số, sau đây là một số tham số hay dùng:  
/p xem từng trang một.  
/w xem theo hàng ngang  
/a xem cả các tệp ẩn  
Ví dụ: DIR C:\NC /w/a xem danh sách các th- mục con và các tệp trong th- mục C:\NC theo hàng ngang kể cả các tệp ẩn.  
Lệnh DIR C:\NCCNC\*.\* hiện danh sách các tệp trong th- mục C:\NC có 2 ký tự đầu tiên là NC.
- **MD** (Make Directory) tạo mới th- mục.  
Cú pháp: MD [đ- ờng dẫn] <tên th- mục cần tạo>  
□ đây ta l- u ý là phần [ổ đĩa] [đ- ờng dẫn] đã có, nghĩa là ta chỉ có thể tạo th- mục ở th- mục gốc hoặc th- mục con của một th- mục đã tồn tại. Thí dụ ta muốn tạo các th- mục C:\DANCA\QUANHO ta phải gõ lần l- ợt nh- sau  
MD C:\DANCA  
MD C:\DANCA\QUANHO  
Nếu ta đang ở th- mục nào đó trong ổ đĩa C:, ta chỉ cần gõ  
MD \DANCA  
MD \DANCA\QUANHO  
Vì dấu \ là chỉ th- mục gốc. Lệnh MD <tên th- mục cần tạo> sẽ tạo một th- mục con trong th- mục hiện thời, do đó nếu ta đang ở th- mục gốc của ổ C: ta chỉ cần gõ  
MD DANCA  
MD DANCA\QUANHO
- **CD** (Change Directory) chuyển con trở sang một th- mục khác cùng ổ đĩa.  
Cú pháp: CD [đ- ờng dẫn] <tên th- mục cần chuyển đến>  
Nếu ta muốn chuyển đến th- mục ở ổ đĩa khác tr- ớc hết ta phải chuyển ổ đĩa đã.  
Lệnh CD còn đ- ọc dùng với vài dạng đặc biệt:  
CD.. chuyển về th- mục chứa th- mục hiện thời.  
CD\ chuyển về th- mục gốc.  
Ví dụ về lệnh CD: Chúng ta đang ở một th- mục nào đó trong ổ đĩa C:, ta cần chuyển về \DANCA\QUANHO ta gõ CD \DANCA\QUANHO
- **RD** (Remove Directory) xóa th- mục rỗng.  
Cú pháp: RD [đ- ờng dẫn] <tên th- mục cần xóa>  
Chú ý: ta chỉ xóa đ- ợc th- mục khi trong nó không có tệp hoặc th- mục con. Thí dụ muốn xóa cả th- mục C:\DANCA\QUANHO tr- ớc hết ta phải bảo đảm rằng trong th- mục QUANHO không chứa gì. Sau đó ta gõ

RD C:\DANCA\QUANHO  
RD C:\DANCA

**Lệnh về tập tin:**

- **COPY CON** (Copy console — thiết bị đầu cuối gồm màn hình và bàn phím) sao chép những gì ng-òì sử dụng sắp gõ vào cho đến khi đánh dấu kết thúc tệp (Ctrl+Z) vào tệp tin văn bản DOS.  
Cú pháp: COPY CON [đ-ờng dẫn] <tên tập tin>  
*Ví dụ:* Chúng ta muốn tạo tệp tin CADA0.TXT chứa 2 câu ca giao  
Muối ba năm muối đang còn mặn  
Gừng chín tháng gừng vẫn còn cay  
Ta gõ nh- sau  
COPY CON CADA0.TXT  
Sau đó ta gõ các dòng trên, xong nhấn F6 hoặc Ctrl+Z (Nhấn giữ phím Ctrl rồi nhấn phím Z) để báo hiệu kết thúc file, nhấn {ENTER} để nghỉ tệp lên đĩa.
- **TYPE** cho hiện nội dung tệp lên màn hình.  
Cú pháp: TYPE [đ-ờng dẫn] <tên tập tin>  
*Ví dụ:* Chúng ta muốn cho hiện nội dung tệp tin CADA0.TXT vừa tạo ra ở trên, ta gõ  
TYPE CADA0.TXT
- **REN** (Rename) đổi tên tệp, không thay đổi nội dung và vị trí.  
Cú pháp: REN [đ-ờng dẫn] <tên tập tin cũ> <tên tập tin mới>  
*Ví dụ:* Chúng ta muốn đổi tên tệp tin CADA0.TXT trong th- mục C:\VANHOC  
Ta viết REN C:\VANHOC\CADA0.TXT CA\_DAO.TXT
- **COPY** Sao chép tệp tin thành tệp tin khác.  
Cú pháp: COPY [đ-ờng dẫn 1] <tên tập tin cũ> [đ-ờng dẫn 2] <tên tập tin mới>  
*Ví dụ:* Chúng ta muốn copy tệp tin CADA0.TXT trong th- mục C:\VANHOC vào th- mục C:\CADA0  
Ta viết COPY C:\VANHOC\CADA0.TXT C:\CADA0\CADA0.TXT  
Nếu [đ-ờng dẫn 1] = [đ-ờng dẫn 2] thì <tên tệp mới> phải khác <tên tệp cũ>.  
Ta cũng có thể dùng ký tự đại diện, và trong tr-ờng hợp tệp mới có tên nh- tệp cũ (đĩ nhiên là ở th- mục khác) thì ta không nhất thiết gõ vào tên tệp mới.  
Thĩ dụ nếu ta muốn copy tất cả các tệp có đuôi TXT từ th- mục \CADA0 sang th- mục \VANHOC ta dùng lệnh  
COPY \CADA0\\*.TXT \VANHOC  
Ta có thể dùng lệnh COPY để nối nhiều tệp văn bản với nhau. Thĩ dụ ta có tệp CADA01.TXT và tệp CADA02.TXT ta có thể nối 2 tệp này thành tệp CADA0.TXT bằng lệnh:  
COPY CADA01.TXT+CADA02 CADA0.TXT
- **DEL** (Delete) hoặc **ERASE** Xóa tệp.  
Cú pháp: DEL [đ-ờng dẫn] <tên tập tin>  
*Ví dụ:* Chúng ta muốn xóa tệp tin CADA0.TXT trong th- mục C:\VANHOC  
Ta viết DEL C:\VANHOC\CADA0.TXT  
Ta cũng có thể dùng ký tự đại diện  
DEL \VANHOC\\*.TXT

**Lệnh về hệ thống:**

- **CLS** (Clear Screen) Xóa màn hình.  
Cú pháp: CLS
- **DATE** Điều chỉnh ngày tháng năm của hệ thống.  
Cú pháp: DATE  
Trên màn hình sẽ hiện lên ngày tháng hiện tại, thí dụ  
Current date is Sat 08-19-2000  
Enter new date (mm-dd-yy):  
Ta có thể chỉnh lại cho đúng. Những hiệu chỉnh của ta sẽ đ- ọc ghi lại trong RAM CMOS và lần khởi động máy lần sau vẫn có hiệu lực.
- **TIME** Điều chỉnh giờ, phút, giây của hệ thống.  
Cú pháp: TIME  
Trên màn hình sẽ hiện lên giờ, phút, giây hiện tại, thí dụ  
  
Current time is 3:49:41.07p  
Enter new time:  
  
Ta chỉ cần nhập giờ và phút là đủ. Thí dụ 15:30
- **VOL** (Volume) Hiện tên ổ đĩa.  
Cú pháp: VOL [ổ đĩa]
- **VER** (Version) Hiện phiên bản của DOS.  
Cú pháp: VER  
Thí dụ khi đánh lệnh trên ta thấy trên màn hình dòng chữ: MS-DOS Version 6.22
- **PATH** Thiết lập đ- ờng dẫn để DOS có thể tìm các tệp tin lệnh ngoại trú và các ch- ơng trình ứng dụng. Khi ta gõ một lệnh ngoại trú mà không chỉ rõ đ- ờng dẫn DOS tr- ớc hết tìm trong th- mục hiện hành, nếu không thấy sẽ tìm tiếp ở các th- mục đ- ọc ghi trong lệnh PATH.  
Cú pháp: PATH để xem đ- ờng dẫn hiện hành  
PATH; xóa đ- ờng dẫn hiện hành  
PATH <đ- ờng dẫn 1>;[đ- ờng dẫn 2];[đ- ờng dẫn 3];...;[đ- ờng dẫn n]  
Thông th- ờng trên ổ đĩa C chúng ta đặt lệnh PATH trong tập tin AUTOEXEC.BAT để mỗi lần khởi động máy lệnh PATH sẽ có hiệu lực ngay mà không cần phải gõ lại.  
Ví dụ:  
PATH=C.;C:\DOS;C:\WINDOWS;C:\NC;C\VIRUS;
- **PROMPT** dùng để đổi dấu nhắc DOS  
Cú pháp: PROMPT <các tùy chọn>  
Thông th- ờng ng- ời ta th- ờng dùng PROMPT \$p\$g sẽ có dấu nhắc là th- mục hiện thời và dấu >

***c. Các lệnh ngoại trú:***

Nếu máy tính của bạn dùng hệ điều hành MS-DOS hoặc Windows các phiên bản cũ chạy trên nền DOS, thì các lệnh ngoại trú của DOS th- ờng ở trong th- mục C:\DOS.

Lệnh ngoại trú là lệnh không th- ờng trú trong bộ nhớ chính (RAM), mà ở trên đĩa cứng hoặc đĩa mềm ở dạng tệp tin có đuôi COM,EXE hoặc BAT. Sau đây là một số lệnh ngoại trú quan trọng th- ờng dùng:

- **LABEL** Đổi tên ổ đĩa.

Cú pháp: Label [ổ đĩa]

- **ATTRIB** để đặt hoặc xóa thuộc tính cho tập tin.

Cú pháp: ATTRIB [+R; -R] [+A; -A] [+S; -S] [+H; -H] [đ- ờng dẫn] <tên tập tin>

Dấu + là đặt thuộc tính, dấu - là xóa thuộc tính.

R thuộc tính chỉ đọc (**R**ead Only)

A thuộc tính l- u trữ (**A**rchive - trong phần l- u trữ thuộc tính 1 nếu ch- a đ- ọc backup, 0 nếu đã backup bằng DOS.

S thuộc tính hệ thống (**S**ystem)

H thuộc tính ẩn (**H**idden)

Thí dụ bạn muốn đặt thuộc tính chỉ đọc cho tệp C:\VANHOC\CADAO.TXT, bạn gõ  
ATTRIB +R C:\VANHOC\CADAO.TXT

- **FORMAT** để định dạng cấp cao đĩa cứng hoặc đĩa mềm.

Cú pháp: FORMAT <ổ đĩa> [/tham số]

Khi thực hiện lệnh này toàn bộ dữ liệu trên ổ đĩa đều mất sạch. Vì vậy các bạn phải thận trọng khi FORMAT đĩa mềm và không nên FORMAT đĩa cứng nếu đó không phải là máy của bạn. Vậy ta chỉ nên thực hành lệnh này với đĩa mềm mà thôi.

Thí dụ bạn muốn FORMAT đĩa mềm và muốn đĩa đó là đĩa khởi động DOS, bạn gõ

FORMAT A:/s

Rồi thực hiện nh- máy chỉ dẫn trên màn hình.

- **SYS** truyền DOS cho đĩa cứng hoặc đĩa mềm.

Cú pháp: SYS <ổ đĩa 1> <ổ đĩa 2>

Điều kiện <ổ đĩa 1> phải là đĩa khởi động DOS, nghĩa là có 3 tệp quan trọng nhất của DOS là IO.SYS, MSDOS.SYS và COMMAND.COM

Thí dụ ta muốn có một đĩa mềm khởi động ta cho đĩa vào ổ A: rồi gõ:

SYS C: A:

- **SCANDISK** Kiểm tra và sửa lỗi trên đĩa cứng và đĩa mềm, đánh dấu các cung từ hỏng để DOS về sau không dùng đến nữa.

Cú pháp: SCANDISK <ổ đĩa>

Thí dụ ta cần kiểm tra và sửa lỗi trên đĩa mềm A:, ta gõ

SCANDISK A:

Rồi làm theo h- ớng dẫn hiện ra trên màn hình.

- **DEFRAG** giải phân mảnh đĩa cứng hoặc đĩa mềm. Nh- ta biết, nếu ta ghi rồi xóa nhiều lần trên đĩa thì th- ờng các tệp không còn đ- ọc ghi trên các liên cung liên tục nữa và làm chậm tốc độ truy xuất dữ liệu. Lệnh này sẽ sắp xếp lại các tệp sao cho mỗi tệp đ- ọc ghi trên những liên cung liên tục.

Cú pháp: DEFRAG [ổ đĩa]

Thí dụ ta cần giải phân mảnh đĩa mềm A:, ta gõ

DEFRAG A:

Rồi làm theo h- ống dẫn hiện ra trên màn hình.

- **UNDELETE** Khôi phục tập tin bị xóa.

Cú pháp: UNDELETE <tên của tập tin đã bị xóa>

Nh- ta đã biết một tập tin khi bị xóa thì chỉ có ký tự đầu tiên trong phần tên tệp là đ- ọc thay bằng ký tự có mã ASCII 229, để DOS biết rằng vùng l- u trữ tệp có thể sử dụng để l- trữ các tệp mới. Nếu sau khi xóa tệp ta vẫn ch- a ghi tệp mới lên đĩa thì ta có thể dùng lệnh UNDELETE để khôi phục. Thí dụ ta cần khôi phục tệp NMTH.DOC vừa mới bị xóa, ta gõ:

UNDELETE NMTH.DOC

Máy sẽ kiểm tra và thông báo khả năng phục hồi. Nếu đ- ọc máy sẽ hỏi ta nhập vào ký tự đầu tiên của tên tệp. Ta có thể dùng ký tự đại diện cho lệnh UNDELETE

Thí dụ UNDELETE C:\MON\_HOC\QUANHON\*.\*

**d. Một số lệnh hay dùng trong tệp CONFIG.SYS**

BUFERS=30

FILES=40

DEVICE=C:\DOS\HIMEM.SYS

DEVICE=C:\DOS\EMM386.EXE

Giải thích: Nh- phần tr- ớc chúng tôi đã nói đến, để tăng tốc độ truy nhập đĩa DOS dành ra một vùng bộ nhớ RAM làm vùng đệm cho các thao tác ghi đọc đĩa. Mỗi vùng đệm nh- vậy có độ lớn 512 byte tức bằng 1 sector. Ta có thể ấn định số vùng đệm này cho DOS để đạt hiệu quả tối - u.

Một số phần mềm ứng dụng nh- FOXPRO đòi hỏi số tệp mở đồng thời rất lớn. Nếu không khai báo gì thì DOS chỉ cho phép mở tối đa 8 tệp, trong đó DOS đã sử dụng một số. Vì vậy cần phải khai báo số tệp có thể mở đồng thời đủ lớn thì các phần mềm ứng dụng mới làm việc đ- ọc. Lệnh HIMEM.SYS đ- a phân bộ nhớ mở rộng nối với bộ nhớ truyền thống (640K) để sẵn sàng làm việc, còn EMM386.EXE lại quản lý và sử dụng bộ nhớ mở rộng. Trên đây chỉ là các lệnh mẫu, các bạn có thể thay đổi tham số hoặc đ- ờng dẫn cho phù hợp với máy bạn đang sử dụng.

**e. Một số lệnh hay dùng trong tệp AUTOEXEC.BAT**

SET PROMPT \$p\$g

SET PATH = C:; C:\NC;C:\VIRUS;C:\WINDOWS;

Lệnh thứ nhất đặt hình dạng con trỏ DOS là th- mục hiện thời và dấu >. Lệnh thứ 2 chỉ ra các đ- ờng dẫn để DOS và các ch- ơng trình khác tìm kiếm các tệp lệnh hoặc dữ liệu. Và đây cũng chỉ là ví dụ mẫu, các bạn có thể thay đổi.

Bảng tóm tắt một số lệnh hay dùng của MS-DOS

	Lệnh nội trú	Ý nghĩa	Lệnh ngoại trú	Ý nghĩa
1	DATE	Sửa đổi ngày của hệ thống	LABEL	Đổi tên ổ đĩa
2	TIME	Sửa đổi giờ của hệ thống	ATTRIB	Đặt lại hoặc chỉ ra thuộc tính tệp
3	CLS	Xóa màn hình	SCANDISK	Kiểm tra đĩa
4	VER	Xem phiên bản của DOS	DEFRAG	Giải phân mảnh đĩa
5	VOL	Xem tên ổ đĩa	FORMAT	Định dạng đĩa
6	DIR	Hiện danh sách tệp tin	SYS	Chuyển các tệp hệ thống sang đĩa khác
7	MD	Tạo th- mục mới	FDISK	Tạo các phân khu trên đĩa cứng
8	CD	Chuyển sang th- mục khác	UNDELETE	Khôi phục các tệp đã bị xóa
9	RD	Xóa th- mục rỗng	DISKCOPY	Sao chép cả đĩa mềm sang đĩa mềm khác
10	COPY	Sao chép tệp		
11	DEL	Xóa tệp		
12	REN	Đổi tên tệp		
13	TYPE	Hiện lên màn hình nội dung tệp		

14	<b>PROMPT</b>	Đặt dấu nhắc hệ thống		
15	<b>PATH</b>	Đặt các đường dẫn cho DOS tìm kiếm tệp		

### 2.2.5. Cấu hình cho máy tính

DOS có rất nhiều lệnh và chức năng, tuy nhiên nhiều lệnh nay không còn thông dụng nữa, vì vậy các bạn cũng không nên tốn công để tìm hiểu tất cả các lệnh của DOS (DOS có khoảng 100 lệnh tất cả).

Sau đây chúng tôi giới thiệu một cách cấu hình máy tính để nhiều người có thể sử dụng theo cấu hình mình muốn. Giả sử bạn có một máy tính cấu hình thấp do đó bạn chỉ cài đặt Windows 3.1, Turbo Pascal, Foxpro. Pascal chạy không cần đến bộ nhớ mở rộng, Foxpro thì cần mở nhiều file một lúc, còn Windows lại cần đến bộ nhớ mở rộng. Ta có thể sửa đổi file CONFIG.SYS và AUTOEXEC.BAT nh- sau để có thể lựa chọn cấu hình thích hợp. (Trong thực tế bạn có thể sửa đổi khác cho phù hợp công việc của bạn)

Trong tệp **CONFIG.SYS**

Sau những lệnh chung bạn tạo menu nh- sau

**[Menu]**

menuitem=Pascal, Khởi động Turbo Pascal

menuitem=Foxpro, Khởi động Foxpro

menuitem=Windows, Khởi động Windows

**[Pascal]**

**[Foxpro]**

files=50

buffers=40

**[Windows]**

device=c:\dos\himem.sys

device=c:\dos\emm386.exe

files=40

buffers=30

Trong tệp **AUTOEXEC.BAT**

Sau những lệnh chung bạn gõ vào những dòng sau:

Goto %config%

: Pascal

turbo.exe

goto End

: Foxpro

Foxpro.exe

goto End

:Windows

Win

:End

## 2.3. Phần mềm hỗ trợ NC (Norton Commander)

### 2.3.1. Giới thiệu tiện ích NC

DOS đã đưa ra cách tổ chức và quản lý dữ liệu rất khoa học và cung cấp cho chúng ta một hệ thống lệnh phong phú, giúp chúng ta thực hiện được hầu hết những yêu cầu cần thiết trong quản lý dữ liệu. Tuy nhiên việc nhớ cú pháp các lệnh của DOS một cách chính xác và gõ từng ký tự từ bàn phím nhiều khi khiến chúng ta mệt mỏi, dễ nhầm lẫn. Phải nói rằng giao diện của DOS với người dùng chưa hấp dẫn, khó theo dõi. Nhận thấy những nhược điểm này một số nhà tin học đã viết những phần mềm tiện ích trên cơ sở các dịch vụ của DOS, cũng thực hiện những lệnh tương tự như DOS nhưng với giao diện đẹp hơn, người dùng không phải nhớ chi tiết các câu lệnh. Đã có nhiều phần mềm tiện ích ra đời theo hướng này và được người dùng ưa chuộng như PCTOOLS, XTREE, các tiện ích NORTON (NORTON Utilities), trong đó có thể nói các phần mềm tiện ích của PETER NORTON là ưu việt hơn hẳn. Các chương trình này có một số chức năng phong phú hơn DOS, giao diện lại đẹp, vì vậy hầu như trong nhiều trường hợp người sử dụng đã dùng các tiện ích này thay cho DOS. Sau đây chúng tôi xin giới thiệu sơ lược cách sử dụng phần quan trọng nhất của các tiện ích NORTON là NC (NORTON COMMANDER) phiên bản 5.0

Bộ chương trình NC thường được ghi trên đĩa cứng hoặc đĩa mềm, đặt trong thư mục NC với nhiều tệp chương trình. Chương trình chính có tên là NC.EXE. NC là chương trình thường trú trong bộ nhớ (RAM), nghĩa là trong khi chạy chương trình NC, ta vẫn có thể chạy những chương trình khác. Sau khi các chương trình này kết thúc, quyền điều khiển được trả lại cho NC.

Để cài đặt NC lên máy bạn, hãy chạy chương trình cài đặt INSTALL.EXE trên đĩa gốc. Nếu không có đĩa cài đặt, bạn cũng có thể copy toàn bộ các tệp của NC ở một máy khác sang máy bạn.

### Khởi động và thoát khỏi NC:

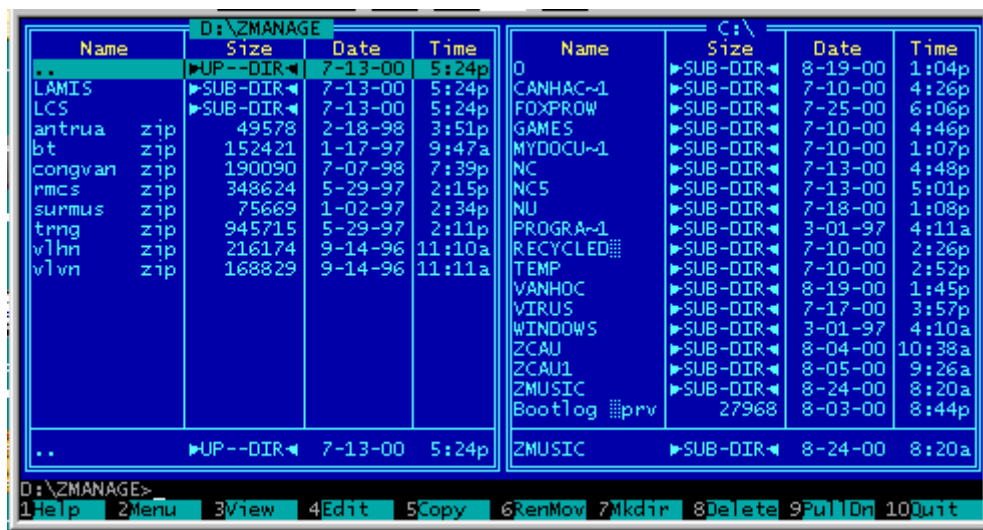
- Để khởi động NC, ta chỉ cần cho chạy chương trình NC.EXE từ dấu nhắc hệ thống (lưu ý đường dẫn thích hợp).

- Để thoát khỏi NC, ta chỉ việc nhấn phím F10, khi đó trên màn hình xuất hiện thông báo:

Nhấn phím Y để đồng ý thoát, còn nhấn phím N sẽ không thoát nữa.

### Giao diện của NC:

Sau khi khởi động, màn hình NC xuất hiện với hai khung cửa sổ:



Trong mỗi khung cửa sổ liệt kê nội dung một ổ đĩa nào đó. Ta sẽ thấy ở một trong 2 cửa sổ có một thanh màu xanh có thể dịch chuyển lên xuống bằng các phím mũi tên, và có thể dịch chuyển từ cửa sổ này sang cửa sổ khác bằng phím Tab. Thanh này đ-ợc gọi là thanh chọn và là con trỏ của NC. □ đĩa chứa thanh chọn là ổ đĩa hiện thời, còn th- mục mà thanh chọn đang ở trong đó là th- mục ng-âm định. Thí dụ tr-ờng hợp trên đây ổ đĩa ng-âm định là ổ đĩa D: và th- mục ng-âm định là th- mục \ZMANAGE

Phía d-ới các cửa sổ màn hình có hai dòng.

- Dòng thứ nhất chứa dấu nhắc hệ thống trong hình vẽ trên là C:\> để ng-ời dùng có thể nhập các lệnh của MD-DOS.
- Dòng thứ hai là h-ớng dẫn cách gõ phím chức năng từ F1 đến F10. Thí dụ gõ F1 là Help tức là hiện bảng trợ giúp, F5 Copy, F10 Quit tức là thoát ra khỏi NC.

Khi nhấn và giữ phím Alt, cũng tại dòng này một bảng h-ớng dẫn khác xuất hiện, chỉ cho ta các chức năng khi ta nhấn và giữ phím Alt sau đó gõ các phím F1, F2... Thí dụ Alt+F1 là để chọn ổ đĩa cho cửa sổ trái, Alt+F10 là hiện cây th- mục.

Nếu ta nhấn và giữ phím Ctrl, cũng tại dòng này một bảng h-ớng dẫn khác xuất hiện, là các chức năng khi ta nhấn và giữ phím Ctrl sau đó gõ các phím F1, F2... Thí dụ Ctrl+F1 là tắt hoặc bật cửa sổ trái, Ctrl+F9 là in tệp.

Nếu ta nhấn F9 thì một dòng thực đơn xuất hiện phía trên màn hình





Ta có thể chọn một mục nào đó thí dụ Files nhấn thực đơn kéo xuống sẽ hiện ra và ta có thể chọn các mục nhỏ hơn.

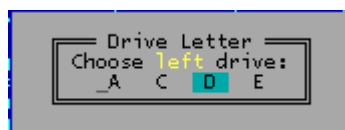
### 2.3.2. Các lệnh của NC

Mục đích của NC là hỗ trợ DOS, giúp người dùng thực hiện các lệnh DOS được dễ dàng và trực quan hơn. Bây giờ ta thử xem những lệnh DOS đã học có thể thực hiện trong NC như thế nào?

#### a. Các lệnh về ổ đĩa:

NC cho phép ta hiện 2 ổ đĩa khác nhau trên 2 cửa sổ (DOS chỉ có thể hiện một ổ đĩa). Trong đó cửa sổ có thanh chọn sáng là ổ đĩa hiện thời. Nếu ta muốn cửa sổ phía trái hiện ổ đĩa khác ta nhấn Alt+F1

Một cửa sổ nhỏ hiện lên cho phép ta chọn ổ đĩa mới:



Tương tự ta có thể nhấn Alt+F2 để chọn ổ đĩa cho cửa sổ phải.

Ta có thể **nhấn Ctrl+L để xem thông tin về bộ nhớ, dung lượng đĩa**. (Nếu không xem nữa ta nhấn Ctrl+L).

#### b. Các lệnh về thư mục:

- **Xem nội dung thư mục:** (tương ứng với lệnh DIR của DOS)

Trên màn hình ta thấy danh sách các thư mục con và tệp của thư mục hiện thời. Nếu ta muốn xem mỗi thư mục chiếm bao nhiêu không gian nhớ thì ta chuyển thanh chọn đến thư mục đó và gõ **Ctrl+Q**. Cửa sổ phía bên kia sẽ hiện cho ta thông tin về độ lớn của thư mục tính theo byte. (Nhấn Ctrl+Q để trở về màn hình cũ). Ta có thể hiện danh sách các tệp theo nhiều cách. Ta có thể chọn cách hiển thị như sau:

Thí dụ ta cần hiển thị cửa sổ trái, ta nhấn F9, chọn Left rồi nhấn, màn hình sau hiện ra:



Ta có thể chọn dạng cây (TREE), sắp xếp theo tên (NAME), hay phần mở rộng (extension)...

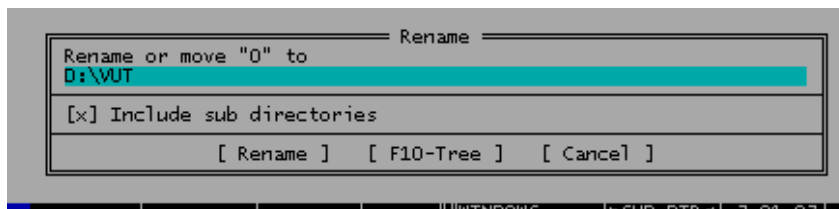
- **Chuyển sang thư mục khác:** (tương ứng với lệnh CD của DOS)

Trước hết ta tìm hiểu cách vào và ra khỏi thư mục:

Để chuyển thanh chọn vào thư mục nào đó trước hết ta chuyển thanh chọn đến tên thư mục đó, sau đó nhấn. Khi thanh chọn đang ở trong thư mục ta thấy dòng đầu tiên có 2 dấu chấm biểu thị cho thư mục mẹ chứa thư mục hiện thời. Để thanh chọn đến dòng này và nhấn ta sẽ ra bên ngoài thư mục tức là trở về thư mục chứa thư mục vừa rồi (ta cũng có thể nhấn **Ctrl+PageUp** từ

vị trí bất kỳ trong th- mục để ra th- mục mẹ). Bằng cách này ta có thể thâm nhập vào các th- mục ở trong một th- mục nào đó hoặc từ các th- mục phía trong trở dần ra ngoài. Đôi khi nếu chúng ta đang ở mức quá sâu và muốn ra ngay th- mục gốc ta có thể gõ lệnh DOS ở dòng phía d- ới: CD\

- **Copy th- mục:** (t- ơng ứng với lệnh XCOPY của DOS)  
Nếu muốn copy 1 th- mục ta chuyển thanh chọn đến th- mục đó rồi nhấn **F5** , màn hình sẽ hiện lên th- mục đích. Nếu ta không chấp nhận thì gõ đ- ờng dẫn mới vào rồi nhấn . Nếu ta muốn copy nhiều th- mục thì tr- ớc hết ta đánh dấu các th- mục bằng cách sau:  
đ- a thanh chọn đến th- mục cần chọn, nhấn phím Insert, th- mục biến thành màu vàng nghĩa là đã đ- ợc chọn (Nếu không muốn chọn nữa ta lại nhấn phím Insert lần nữa).
- **Tạo th- mục mới:** (t- ơng ứng với lệnh MD)  
Giả sử ta cần tạo một th- mục con ở th- mục hiện thời ta gõ **F7** một dòng xuất hiện trên màn hình để ta nhập vào tên th- mục cần tạo.
- **Xóa th- mục:** (t- ơng ứng với lệnh RD hoặc DELTREE)  
Ta chuyển thanh chọn đến th- mục cần xóa, gõ **F8**, máy sẽ hỏi ta tr- ớc khi xóa thực sự. Nếu th- mục có chứa th- mục con và ta muốn xóa cả các th- mục con thì ta phải đánh dấu vào ô phía tr- ớc dòng chữ [ x] Include subdirectories  
Vậy lệnh này giống lệnh DELTREE hơn.
- **Đổi tên hoặc dịch chuyển th- mục:** (lệnh này DOS không có)



Ta chuyển thanh chọn đến th- mục cần đổi tên hoặc dịch chuyển rồi nhấn **F6** , một cửa sổ sẽ hiện ra thí dụ

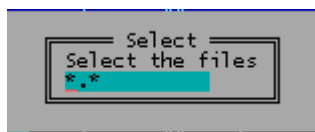
Thông th- ờng cửa sổ tích cực là th- mục nguồn, còn cửa sổ phía bên kia là th- mục đích, do đó khi ta muốn chép hoặc di chuyển file thì NC hiện th- mục của cửa sổ phía bên kia. Nếu ta muốn thay đổi thì gõ vào tên mới. Trong tr- ờng hợp trên đây nếu ta muốn đổi tên tệp thì ta đánh tên mới vào, không có đ- ờng dẫn. Còn nếu muốn chuyển file thì ta chấp nhận th- mục NC nêu ra hoặc gõ đ- ờng dẫn mới vào.

### c. Các lệnh về tệp:

Các lệnh **copy**, **xóa**, **đổi tên hay dịch chuyển tệp**: đ- ợc làm y hệt nh- với th- mục.

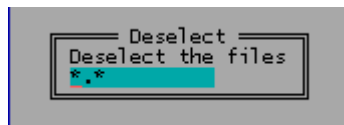
Riêng đối với tệp ta có thể dùng ký tự đại diện để chọn hoặc không chọn nhiều tệp bằng cách sau:

- Chọn nhóm tệp có cùng một đặc tr- ng nào đó: nhấn phím dấu cộng (+) ở tận cùng bên phải bàn phím, trong vùng phím số. Khi đó trên màn hình xuất hiện:



Gõ vào vị trí \*.\* các đặc tả của nhóm tệp cần chọn, chẳng hạn, \*.BAK, rồi nhấn Enter. Khi này các tệp có cùng đặc tả sẽ đổi màu.

- Nếu bạn muốn huỷ bỏ việc chọn của một số hoặc tất cả các tệp đã chọn, hãy nhấn phím dấu trừ (-) ở vùng phím số bên phải bàn phím. Trên màn hình xuất hiện:



T-ơng tự nh- trên, nhập đặc tả của các tệp muốn huỷ việc chọn rồi nhấn Enter.

- **Soạn thảo tệp**

Bạn có thể soạn thảo tệp đ-ợc chọn bằng cách nhấn **F4**.

Nội dung tệp sẽ hiện ra trên màn hình với dòng trợ giúp phía d-ới để ta có thể tiến hành soạn thảo. Chức năng này rất tiện lợi để bạn sửa đổi nhỏ các tệp có sẵn. Tuy nhiên đây không phải là một hệ soạn thảo văn bản chuyên dụng nên không mạnh nh- các hệ soạn thảo văn bản khác. Để l- u bản bản lên đĩa, nhấn phím **F2**. Để thoát ra khỏi chế độ soạn thảo mà không ghi, nhấn **Esc**.

- **Soạn thảo tệp mới**

Để soạn thảo một tệp mới, hãy gõ **SHIFT+F4**, trên màn hình xuất hiện thông báo “Edit the file:” (“Soạn thảo tệp:”), ta gõ vào tên tệp rồi nhấn Enter.

- **Tìm tệp**

Để tìm xem một hoặc một nhóm các tệp có ở trên đĩa hiện hành hay không, hãy nhấn tổ hợp phím **Alt+F7**. Hộp hội thoại Find File (tìm tệp) hiện ra:

cho phép bạn điền vào các thông tin:

Find Files (Tìm các tệp): tên tệp hoặc đặc tr- ng của nhóm tệp.

Location (s) (ở vị trí): vị trí cần tìm

Containing (có chứa ...): Đánh dấu đoạn văn bản chứa trong các tệp cần tìm.

Các tệp tìm đ-ợc sẽ đ-ợc liệt kê ra màn hình. Nếu bạn muốn tập hợp chúng lại, có thể dùng chức năng Send To Panel. Lúc này NC sẽ đ- a ra danh sách các tệp và vị trí của chúng để bạn tiếp tục xử lý.

- **In tệp**

Nhấn **Ctrl+F9** để in tệp đ-ợc chọn

- **Xem nội dung tệp**

Chọn tệp muốn xem (th- ờng là tệp văn bản), rồi gõ **F3**. Khi đó nội dung tệp sẽ đ-ợc hiện lên màn hình. Nếu nội dung tệp v- ợt quá một trang màn hình, có thể dùng các mũi tên lên, xuống và các phím **PgUp**, **PgDn** để xem những phần bị khuất.

NC còn có các tệp ch- ơng trình có tên chứa VIEW dùng để xem nội dung của các tệp là sản phẩm của các ch- ơng trình khác, chẳng hạn:

123 VIEW.EXE	Xem tệp LOTUS 123
ARCVIEW.EXE	Xem tệp nén dạng *.ZIP, *.ARC, *.LXH
BITMAP.EXE	Xem tệp ảnh .BMP, *.PIC
DBVIEW.EXE	Xem tệp cơ sở dữ liệu *.DBF

d. **Các lệnh khác:**

- **Nén tệp**

Bạn có thể tiết kiệm chỗ trên đĩa bằng cách nén các tệp không cần dùng đến th- ờng xuyên. Để nén tệp hoặc một nhóm tệp, tr- ớc hết hãy chọn chúng, sau đó ấn **Alt+F5**. Hộp hội thoại hiện ra để bạn lựa chọn các tham số nén.

- **Để gỡ nén**, chỉ cần chọn tệp \*.ZIP và nhấn **Alt+F6**.  
Bạn cũng có thể xem tệp nén chứa gì tr-ớc khi quyết định gỡ ra, bằng cách ấn ↵ sau khi chọn tệp nén.
- **Thực hiện ch-ơng trình trong NC**  
Muốn thực hiện ch-ơng trình nào, chỉ việc di thanh chọn tới tên ch-ơng trình đó, rồi nhấn Enter. Sau khi ch-ơng trình kết thúc, màn hình của NC đ-ợc khôi phục lại.
- **User menu: Tự tạo bảng chọn lệnh cho riêng bạn.**  
Tr-ớc tiên tạo ra (có thể dùng chính NC) tệp NC.MNU có nội dung thích hợp để tạo bảng chọn cho riêng bạn. Giả sử bạn th-ờng làm việc với ngôn ngữ lập trình PASCAL và hệ soạn thảo văn bản BKED chẳng hạn. Tệp NC.MNU của bạn có nội dung t-ơng ứng nh- sau:  
P PASCAL  
D:  
CD\PASCAL  
TURBO  
C:  
S Soan thao BKED  
C:  
CD\BKED  
BKED

Khi đó mỗi khi bạn chạy NC và nhấn F2, bảng chọn theo ý bạn sẽ hiện ra

Bạn chỉ cần gõ các phím tắt (P hoặc S) để khởi động các môi tr-ờng làm việc t-ơng ứng của mình một cách nhanh chóng.

- **So sánh 2 th- mục.**  
Nhiều khi ta cần so sánh 2 th- mục về cơ bản là giống nhau nh- ng có một vài tệp mới đ-ợc cập nhật. Ta muốn so sánh xem trong 2 th- mục thì th- mục nào mới hơn. Ta làm nh- sau:  
Ta cho 2 th- mục cần so sánh hiện ở 2 cửa sổ. Nhấn **F9** để hiện menu phía trên, sau đó chọn mục Command rồi chọn Compare Directories rồi những tệp mới hoặc mới hơn trong th- mục này so với th- mục kia sẽ chuyển thành màu vàng.
- **Các dịch vụ về đĩa.**  
Bạn kích hoạt menu phía trên bằng cách nhấn F9, sau đó chọn mục Disk. Mục này cung cấp các dịch vụ về đĩa (COPY, FORMAT...)  
Các tiện ích mạng đ-ợc kích hoạt khi ấn SHIFT+F2.  
Đặc biệt chức năng Disk cleanup ... (Làm sạch đĩa) là một chức năng mới của NC 5.0, cho phép bạn dọn dẹp đĩa bằng cách xóa tất cả các tệp có đặc tr-ng nào đó trong tất cả các th- mục trên đĩa cùng một lúc giúp bạn không phải tự tìm trong các th- mục khác nhau.

## 2.4. Virus máy tính và cách phòng chống

### 2.4.1. Khái niệm

Virus máy tính là một ch-ơng trình đ-ợc viết ra không phải vì mục đích ứng dụng mà là để gây cản trở thậm chí phá hoại các phần mềm khác. Ch-ơng trình loại này không đ-ợc l-ưu trữ trên đĩa nh- các ch-ơng trình thông th-ờng mà len lỏi vào các phần mềm khác. Có điều kiện (thí dụ ta cho chạy ch-ơng trình đã nhiễm virus chẳng hạn) là chúng lây lan rất nhanh, làm sai lệch các hoạt động của máy tính.

#### **2.4.2. Ch- ơng trình chống Virus**

Ch- ơng trình chống virus là ch- ơng trình đ- ọc viết ra nhằm kiểm tra và loại bỏ virus ra khỏi các phần mềm khác.

#### **2.4.3. Cách phòng chống Virus**

- Cài đặt ch- ơng trình kiểm tra virus th- ờng trú trong bộ nhớ để th- ờng xuyên kiểm tra các thao tác ghi đọc file, đặc biệt là các file trên đĩa mềm để kịp thời hiện thông báo trên màn hình mỗi khi virus xuất hiện.
- Kiểm tra máy theo định kỳ bằng những phần mềm diệt virus mới nhất.
- Khi máy bị nhiễm virus thì tắt máy, khởi động từ đĩa mềm có chứa ch- ơng trình diệt virus để kiểm tra và diệt virus.

Vì virus bao giờ cũng xuất hiện tr- ớc khi có các ch- ơng trình diệt nó. Vì vậy dù ta thực hiện kiểm tra th- ờng xuyên vẫn không thể bảo đảm là máy tính không có virus. Do vậy khi máy tính có hiện tượng hoạt động không bình thường thì ta phải thông báo cho các chuyên gia phòng chống virus để họ nghiên cứu cách giải quyết. Trong nhiều trường hợp sau khi diệt virus thì các thông tin trên đĩa cũng bị sai lệch nặng nề hoặc hỏng hoàn toàn không còn dùng đ- ọc nữa.

## 2.5. Hệ điều hành Windows 98

### 2.5.1. Giới thiệu về hệ điều hành WINDOWS 98

#### a. Hệ điều hành Windows

Nh- ta thấy MS-DOS là HĐH đơn nhiệm nghĩa là tại mỗi thời điểm chỉ có một phần mềm ứng dụng chạy đ-ợc, màn hình giao diện ng-ời-máy là màn hình văn bản, tức là chỉ hiển thị đ-ợc các ký tự trong bảng ASCII mở rộng. Khả năng vẽ 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 nh-ng vẫn ch-a thay đổi đ-ợc một số nh-ợc điểm nói trên. Năm 1984 hãng Apple Computer cho ra đời máy tính Macintosh có tính năng giao diện ng-ời-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.

IBM đã phối hợp với Microsoft xây dựng hệ điều hành đa nhiệm OS/2 (Operating System/2) cho các máy t-ơng thích IBM PC, đã phá vỡ hàng rào 640K RAM, cung cấp sự an toàn cho các ch-ơng trình chạy đồng thời, và cho phép trao đổi động dữ liệu giữa các ch-ơng trình ứng dụng. IBM và Microsoft tin chắc rằng OS/2 sẽ thay thế MS-DOS và sẽ là hệ điều hành của t-ơng lai. Ng-ời dùng đ-ợc khuyến khích dùng OS/2 để thay thế MS-DOS. Tuy vậy phần lớn ng-ời dùng sử dụng OS/2 một cách miễn c-ờng vì họ không muốn thay hẳn những gì họ đã quen thuộc. Rất ít hãng sản xuất phần mềm chịu xây dựng các ch-ơng trình ứng dụng OS/2 và các phiên bản đầu tiên của OS/2 đã chạy các ch-ơng trình MS-DOS rất kém. Microsoft đã xây dựng Windows dựa trên những ý t-ởng của Macintosh. Microsoft Windows lúc đó cũng chỉ đ-ợc xem nh- là một giao diện ng-ời - máy bằng đồ họa cho MS-DOS. Ng-ời sử dụng vẫn có thể lựa chọn là họ có thể chạy các ch-ơng trình ứng dụng trong môi tr-ờng 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 tr-ờng 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 đ-ợc bảo vệ; đột nhiên, lý do căn bản của việc nâng cấp lên OS/2 bị xóa bỏ gần nh- toàn bộ, và Windows 3.0 đã bán ra đ-ợc hàng triệu bản. Microsoft đã đi đến xác nhận rằng t-ơng lai thuộc về Windows. Không đáng ngạc nhiên lắm khi những sự kiện này có vẻ nh- đã dội một gáo n-ớc lạnh lên mối quan hệ vốn thân mật và gần gũi tr-ớc 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 đ-ợc các nh-ợc điểm tr-ớc đây và chạy các ch-ơng trình MS-DOS và Windows một cách hoàn hảo và đã đ-ợc nhiều ng-ời hoan nghênh nh- là một thành tựu kỹ thuật lớn. Tuy vậy OS/2 có làm ng-ng đ-ợc 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 đ-ợc 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, nh-ng vẫn dùng FAT 16 bit. So với các HĐH tr-ớc Windows 95 có nhiều -u điểm: Giao diện thân thiện và trực quan, hỗ trợ nhiều phần cứng, Plug and Play (cắm là chạy), t-ơng thích với DOS 16 và Win 16.

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

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

-Hỗ trợ nhiều cạc màn hình (tối đa là 9), hỗ trợ đa ph-ơng tiện, tích hợp internet và các dịch vụ trực tuyến, chạy nhanh hơn Windows 95, FAT 32 giúp l-ưu trữ tốn ít dung l-ợng 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 đ-ợc phim hoàn chỉnh với âm thanh nổi, chất l-ợng sân khấu trên máy PC trang bị DVD. Thiết bị l-u trữ mới này có thể chứa tới 17GB dữ liệu với hỗ trợ đa ngôn ngữ trên một đĩa.

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

Đặc điểm của Windows nh- tên gọi của nó là chạy tất cả các ứng dụng trong từng cửa sổ riêng biệt mà kích cỡ có thể đ-ợc sửa đổi bởi ng-ời sử dụng.

Trong các phần sau đây chúng tôi chỉ xin giới thiệu Windows 98, là phiên bản đã đ-ợc kiểm nghiệm và dùng rộng rãi ở Việt nam.

### ***c. Cài đặt hoặc gỡ bỏ Windows 98***

Yêu cầu cấu hình máy tính để cài đặt Windows 98:

Bộ vi xử lý 486DX, tốc độ 66 MHz trở lên, RAM 16 Mb hoặc hơn, đĩa cứng khoảng 400 Mb.

Có thể cài đặt nâng cấp từ Windows 3.1 hoặc Windows 95, hoặc cài đặt mới. Các bạn có thể tham khảo chi tiết các kiểu cài đặt trong trợ giúp của Windows 98. Bạn khởi động Help, chọn Getting Started book: Online Version.

## ***2.5.2. Làm việc với WINDOWS 98***

### ***a. Khởi động và ra khỏi Windows 98***

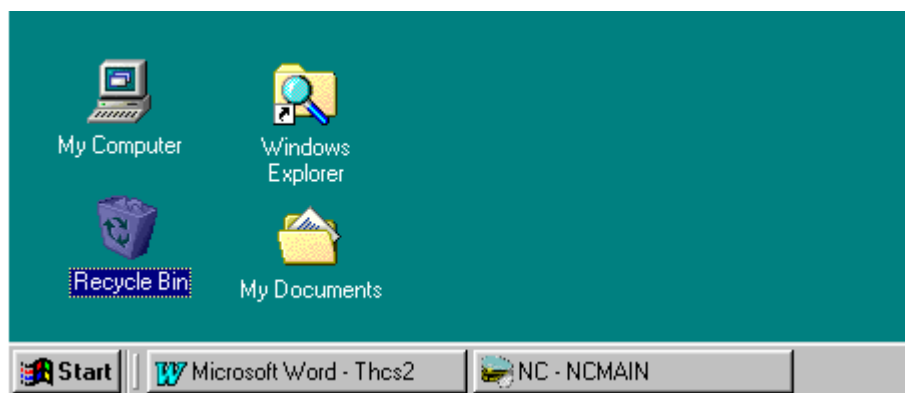
Khi máy đã đ-ợc cài đặt Microsoft Windows 98, mỗi lần khởi động máy là khởi động luôn hệ điều hành Windows 98, biểu t-ợng Windows 98 sẽ hiện ra và khi kết thúc quá trình khởi động ta thấy màn hình nền với một số biểu t-ợng bằng hình ảnh ở nửa trái của màn hình. Bạn có thể thoát khỏi Windows 98 bằng cách nhấn tổ hợp phím **Alt+F4** hoặc nhấn đúp chuột vào nút **Start** ở góc trái màn hình, chọn **Shut down...** rồi chọn tiếp **Shut down the computer**.

### ***b. Màn hình nền của Windows 98***

Từ ý t-ởng WYSIWYG (what you see is what you get) của Macintosh, Microsoft đã cố gắng xây dựng Windows với giao diện đồ họa sao cho thật gần gũi với môi tr-ờng làm việc của con ng-ời. Màn hình nền với tên gọi **Desktop** là t-ợng tr- ng cho bàn làm việc và khoảng không gian quanh ta. Trên đó chúng ta đặt những công cụ cần thiết cho công việc văn phòng nh- : bút, giấy, cặp tài liệu, điện thoại nối với thế giới bên ngoài, thậm chí cả một thùng rác để chúng ta có thể ném vào đấy những giấy tờ không cần thiết nh- ng khi cần lại có thể bới thùng rác lên để tìm kiếm những tài liệu vớt nhảm. Tóm lại màn hình nền mô phỏng bàn làm việc nh- ng với nhiều chức năng đa dạng phong phú hơn.

Vì Desktop là bàn làm việc, và mỗi ng-ời lại có cách sắp xếp theo ý mình nên Windows cho phép ng-ời dùng có thể thêm bớt các biểu t-ợng trên màn hình nền một cách thoải mái. Chính vì vậy nếu ta xem màn hình nền trên các máy khác nhau thì ta thấy chúng không nh- nhau. Thậm chí cả một số chức năng rất quan trọng với hầu hết mọi ng-ời thì lại bị một vài ng-ời sử dụng vui tính nào đó vứt bỏ.

Một màn hình nền ch- a bị thay đổi nhiều bởi ng-ời sử dụng th- ờng có những thành phần sau:



- Các **Shortcut** (lối tắt) là các biểu tượng riêng với mũi tên đen nhỏ nằm ở góc dưới bên trái - tượng trưng cho một chương trình ứng dụng, một tài liệu... nhấn đúp chuột vào Shortcut ta sẽ khởi động chương trình ứng dụng hoặc mở tài liệu tương ứng.

- Các **Folder** (cấp tài liệu) là khái niệm mở rộng của thư mục. Folder chủ yếu thay thế các thư mục của MS-DOS. Tuy nhiên, ngoài các tập tin trên đĩa Windows còn dùng Folder để tổ chức những thứ khác như: phong chữ, máy in, bàn điều khiển (control panel). Chính vì vậy khi ta dùng Windows Explorer để xem thông tin trên máy, ta sẽ thấy thư mục Recycle Bin xuất hiện ở 2 nơi: ở ổ C: với tên Recycled như là thư mục bình thường và ở bên ngoài ngang bằng với My computer, Network Neighborhood... khi đó Recycle Bin là cấp chứa các tài liệu vứt đi và vị trí của nó cũng được làm nổi bật trên màn hình như một số Folder khác.

- **Taskbar** (thanh công việc) để hiển thị các chương trình đang được mở. □ hình trên ta thấy Winword, Excel và Foxpro đang được mở đồng thời. Ta có thể kích hoạt một phần mềm đang mở bằng cách nhấn đúp chuột vào tên phần mềm đó trong thanh Taskbar.

- Nút **Start** (nút khởi động) nhấn đúp chuột vào nút Start ta sẽ khởi động **Start Menu** (thực đơn khởi động)

Start Menu thường có các mục sau:

- **Programs**: Xem danh sách các chương trình.
- **Favorites**: Mở các tệp và các trang Web - ưa thích.
- **Documents**: Mở các tài liệu đã mở gần đây nhất.
- **Settings**: Sửa đổi màn hình nền hoặc các thiết lập hệ thống.
- **Find**: Tìm tệp hoặc thư mục.
- **Help**: Trợ giúp.
- **Run**: Mở các mục như: trang Web, chương trình...
- **Shut Down**: Tắt hoặc khởi động lại máy.

Để bổ sung một mục mới vào menu Start, ta có thể kéo mục đó vào nút Start. Ta cũng có thể bổ sung mục mới vào menu Programs bằng cách sau: đầu tiên tạo biểu tượng lối tắt. Nhấn Start, Settings, Taskbar&Start Menu, Start Menu Programs, Advanced. Sau đó kéo shortcut vào cửa sổ vừa mở bên phải.

### Một số Folder và Shortcut quan trọng trên màn hình nền:



- **My Computer** (máy tính của tôi) đơn giản là một bức tranh về máy tính của bạn. Nhấp đúp biểu tượng My Computer trên màn hình nền để xem các biểu tượng khác biểu thị cho các thành phần khác nhau trên máy tính của bạn như: các ổ đĩa, các máy in, bàn điều khiển. Sau đó bạn có thể nhấp đúp bất kỳ biểu tượng nào để xem thông tin hoặc để dùng thành phần cụ thể nào đó.

- **Windows Explorer** (khám phá Windows) hoàn toàn tương tự như My Computer. Tuy nhiên Windows Explorer cho ta bức tranh có thứ bậc, để quan sát hơn nhất là khi ta làm việc với các thư mục và tệp. Cửa sổ làm việc của Windows Explorer được chia làm 2 phần: bên trái hiện thông tin theo cấu trúc hình cây, cửa sổ bên phải hiện nội dung chi tiết của cặp tài liệu hiện thời.

- **Recycle Bin** (thùng tái chế hay thùng rác). Khi bạn xóa các tệp tin hay các folder trên đĩa cứng bằng My Computer hoặc Windows Explorer thì các tệp hay folder bị xóa sẽ được chuyển đến thùng rác. Bạn có thể mở thùng rác để xem những mục đã bị xóa và bạn có thể khôi phục lại chúng bằng cách đánh dấu những mục muốn khôi phục rồi chọn File, Restore. Nếu bạn muốn xóa vĩnh viễn tất cả các mục trong thùng rác, bạn chọn File, Empty Recycle Bin.

- **My Documents** (các tài liệu của tôi). Cặp tài liệu này là nơi bạn cất giữ các tài liệu của bạn. Dĩ nhiên là bạn có thể lưu các tệp ở một thư mục khác theo ý bạn, nhất là trong trường hợp bạn không muốn cho người khác dễ dàng nhìn thấy.

- **Network Neighborhood** (người láng giềng mạng). Nếu máy tính của bạn nối mạng thì khi nhấp đúp vào biểu tượng này bạn sẽ thấy biểu tượng các máy tính nối với máy của bạn. Bạn có thể định nghĩa ổ đĩa mạng hay truy nhập các máy tính khác để khai thác tài nguyên trong phạm vi được phép. (Với các máy tính không nối mạng chức năng này trường hợp không được cài đặt)

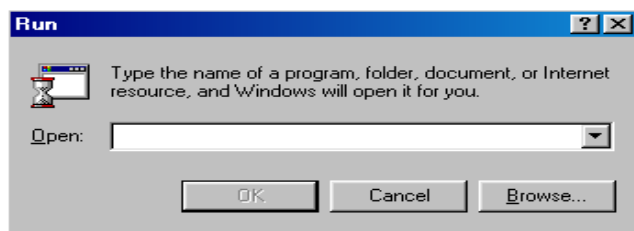
### c. Chạy một chương trình hoặc mở cặp tài liệu

- Nếu chương trình hoặc cặp tài liệu có biểu tượng, ta chỉ cần đưa chuột đến biểu tượng rồi nhấp đúp, chương trình sẽ được chạy hoặc cặp tài liệu được mở ra trong cửa sổ tương ứng. Nếu trong cửa sổ đó còn các biểu tượng ta lại có thể mở tiếp bằng cách này. Từ cặp tài liệu con ta có thể ra bên ngoài bằng cách nhấp đúp vào biểu tượng **Up** ở phần trên cửa sổ.

- Phần lớn các mục ứng dụng có thể tìm thấy trong menu Start. Do đó ta có thể chạy chúng từ menu Start. Thí dụ ta cần chạy ứng dụng vẽ Paint, ta nhấn Start, Programs, Accessories, Paint.

- Ta cũng có thể chạy chương trình một cách trực tiếp như sau:

Nhấn nút Start, Run cửa sổ sau hiện ra:







Ta có thể gõ vào tên chương trình cùng đường dẫn thích hợp hoặc nhấp đúp vào Browse để chọn tên chương trình.

**Thoát khỏi một chương trình:** nhấp vào nút Close ở góc trên bên phải của cửa sổ chương trình hoặc nhấn tổ hợp phím **Alt+F4**.

### Cửa sổ ứng dụng:

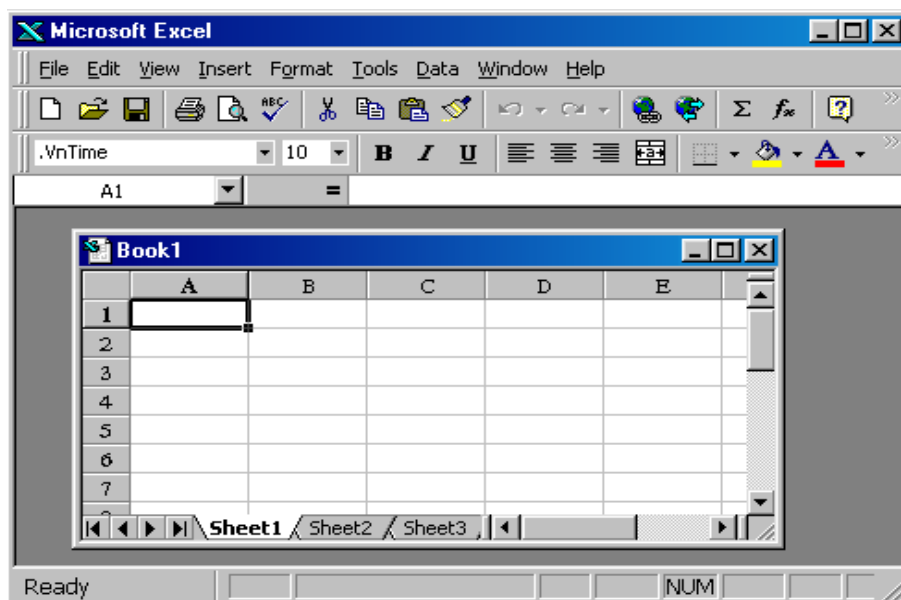
Như chúng tôi đã nói đến, mọi ứng dụng của Windows đều được thực hiện trong một cửa sổ nền. Nếu ứng dụng đó làm việc với các tài liệu thì các tài liệu đó được mở trong cửa sổ con nằm trong

cửa sổ ứng dụng. Bất cứ một cửa sổ nào cũng có những nút điều khiển sau đây nằm ở góc trên bên phải:

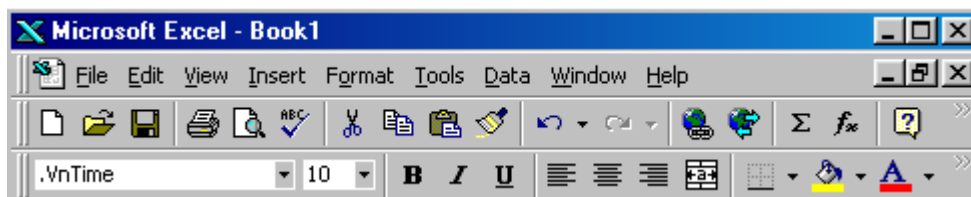
-  Thu cửa sổ thành cực nhỏ
-  Phóng cực to
-  Phục chế cửa sổ về kích cỡ tr- ớc đó
-  Đóng cửa sổ và đóng luôn ứng dụng

□ góc trên bên trái ta thấy một biểu t- ượng của ứng dụng hoặc t- liệu. Ta có thể nhấp đúp chuột vào đó để kích hoạt menu điều khiển của cửa sổ.

Thí dụ cửa sổ ứng dụng Excel (có tên là Microsoft Excel) và cửa sổ t- liệu (có tên là Book1) có thể có dạng nh- sau:



Thông th- ờng ta th- ờng tận dụng không gian và mở hết cỡ cửa sổ t- liệu và ứng dụng. Lúc này cửa sổ ứng dụng và cửa sổ t- liệu chồng khít lên nhau. Và ta thấy phần trên 2 cửa sổ nh- hình d- ưới đây:



Phía trên các cửa sổ ứng dụng th- ờng có các thanh sau đây theo thứ tự từ trên xuống:

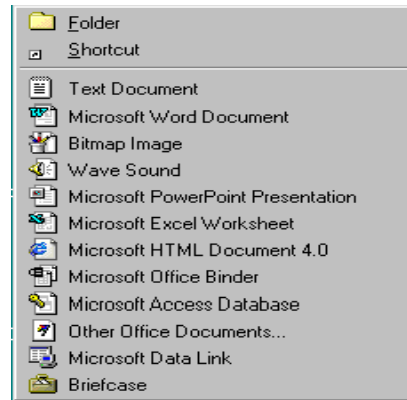
- Thanh tiêu đề (Title bar)
- Thanh thực đơn (Menu bar)
- Thanh công cụ chuẩn (Standard Toolbar)
- Thanh khuôn dạng (Formatting Toolbar)

**d. Tạo một folder, tệp t- liệu hoặc shortcut mới.**

Folder, tệp tin và Shortcut là 3 khái niệm cơ bản của Windows 98.

Trong Windows 98 tên các đối tượng dài tối đa là 255 ký tự và có thể chứa khoảng trống.

Ta có thể tạo ra các biểu tượng này trên màn hình nền hay trong các cửa sổ của My Computer hoặc Windows Explorer. Khi đang ở đâu đó nếu ta muốn biết có tạo được những biểu tượng này không ta chỉ việc nhấn chuột phải. Nếu trong menu hiện ra có mục New, thì có nghĩa là ta có thể tạo được. Ta chọn New, một cửa sổ hiện ra như sau:



ta có thể chọn Folder, Shortcut hoặc một tệp tin. Nếu ta chọn Folder thì một Folder mới hiện ra với tên New Folder, ta sửa lại tên nếu muốn và nhấn. Nếu ta chọn Shortcut thì một cửa sổ hiện ra. Ta có thể gõ vào dòng lệnh hoặc dùng Browse để tìm lệnh cần đưa vào sau đó làm theo hướng dẫn của máy cho đến khi kết thúc. Còn nếu ta chọn tệp tin thí dụ Microsoft Word Document thì 1 tệp Word được tạo ra với tên New Microsoft Word Document, ta có thể sửa lại tên cho phù hợp. Tệp tạo ra còn trống, nếu ta muốn soạn thảo thì nhấp đúp vào tên tệp để khởi động ứng dụng tương ứng cùng với tệp tin được mở.

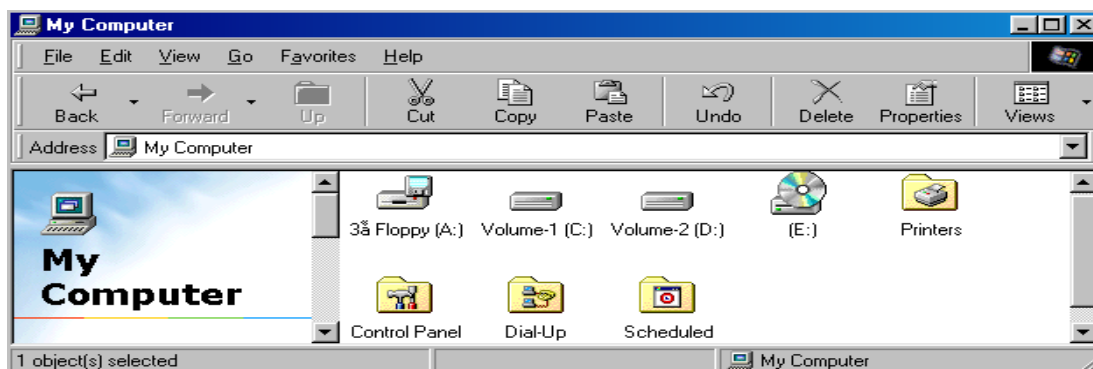
Ta cũng có thể tạo một Shortcut cho tệp trên màn hình nền bằng cách đơn giản sau: dùng nút phải chuột để kéo tệp vào màn hình nền, xuất hiện menu, nhấp vào "Create Shortcut here", Shortcut hiện lên trên màn hình nền. Ta có thể sao chép hoặc di chuyển Shortcut tới vị trí khác.

#### ***e. Một số thao tác liên quan đến ổ đĩa, thư mục và tệp***

Ta có thể thực hiện các thao tác liên quan đến ổ đĩa, thư mục hoặc tệp thông qua My Computer hoặc Windows Explorer. Ở đây chúng tôi giới thiệu cách thao tác thông qua My Computer, bạn đọc có thể tự tìm hiểu cách làm thông qua Windows Explorer.

#### **Chuyển đổi ổ đĩa:**

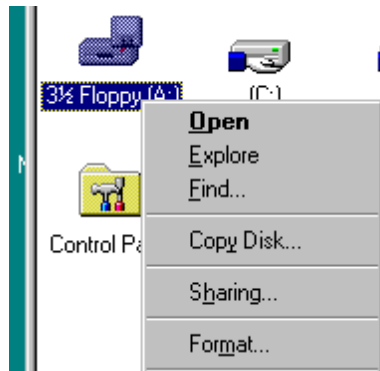
Nhấp đúp biểu tượng My Computer trên màn hình nền. Cửa sổ My Computer xuất hiện.



Trên cửa sổ ta thấy biểu tượng của các ổ đĩa có trên máy tính. Muốn chọn ổ đĩa nào ta chỉ cần nhấp đúp vào ổ đĩa đó, một cửa sổ chứa các thư mục trên đĩa đã chọn sẽ hiện ra.

### **Các thao tác liên quan đến đĩa mềm:**

Ta có thể thực hiện một số thao tác trên đĩa mềm như format, copy đĩa... Để thực hiện các thao tác này ta chuyển chuột về biểu tượng ổ đĩa A: rồi nhấp chuột phải, thực đơn sau hiện ra:



Ta có thể chọn và thực hiện các chức năng cần thiết.

### **Các thao tác liên quan đến thư mục:**

#### **Tạo thư mục:**

Ta có thể làm như ở mục **d.** hoặc bằng cách sau:

- Nhấp đúp vào tên ổ đĩa hay thư mục mà ta muốn tạo thư mục mới ở đó. Cửa sổ ổ đĩa hay thư mục xuất hiện.
- Trên menu File, trở vào mục New, sau đó nhấp Folder.
- Gõ vào tên thư mục nhấn Enter. Thư mục mới hiện lên tại vị trí đã chọn.

Để tạo một thư mục con trong thư mục nào đó ta vào thư mục và thao tác như trên.

#### **Xóa thư mục:**

- Chuyển chuột đến thư mục cần xóa, nhấp chuột phải, chọn Ddelete.

#### **Đổi tên thư mục:**

- Chuyển chuột đến thư mục cần đổi tên, nhấp chuột phải, chọn Rename.
- Gõ vào tên mới và nhấn Enter

#### **Copy thư mục:**

- Chuyển chuột đến thư mục cần chuyển, nhấp chuột phải, chọn Copy.
- Chuyển chuột đến thư mục cần chuyển đến, nhấp chuột phải, chọn Paste.

#### **Dịch chuyển thư mục:**

- Chuyển chuột đến thư mục cần chuyển, nhấp chuột phải, chọn Cut.
- Chuyển chuột đến thư mục cần chuyển đến, nhấp chuột phải, chọn Paste.

### **Các thao tác liên quan đến tệp:**

#### **Tạo tệp mới:**

Ta có thể làm nh- ở mục **d.** hoặc bằng cách sau:

- Vào th- mục mà ta muốn tạo tệp mới ở đó.

- Trên menu File, trở vào mục New, sau đó chọn loại tệp bạn muốn tạo, thí dụ bạn có thể chọn Text Document, một tệp đ- ợc tạo ra với tên New Text Document, bạn có thể sửa lại theo ý mình. Tệp tạo ra còn trống rỗng. Bạn nhấp đúp để bắt đầu soạn thảo. Soạn thảo xong chọn File, Save để cất. Sau đó chọn File, Exit hoặc nhấn vào dấu Close ở góc trái phía trên cửa sổ để thoát ra khỏi NotePad.

Các thao tác **Xóa, Đổi tên, Copy, Dịch chuyển tệp** bạn thực hiện y nh- với th- mục.

**Chú ý:** Ta có thể chọn nhiều tệp hay th- mục. Để chọn các tệp không liên nhau, hãy *giữ phím Ctrl và nhấp vào tên các tệp cần chọn*. Để chọn các tệp liên nhau, tr- ớc tiên ta *chọn tệp đầu tiên, sau đó giữ phím Shift và chọn tệp cuối cùng*. Để chọn tất cả các tệp và th- mục trong cửa sổ, trong menu Edit nhấp Select All hoặc ta nhấn tổ hợp phím Ctrl+A.

### **Tìm tệp và th- mục**

- Nháy nút Start, trở vào mục Find, nhấp mục "Files or Folders". Xuất hiện hộp thoại Find.

- Trong hộp Named, gõ vào tên tệp hay th- mục mà ta muốn tìm.

- Nháy vào mũi tên xuống của hộp "Look in" hoặc nhấp vào nút Browse để xác định nơi cần tìm kiếm.

- Nháy nút Find Now. Sau một lát, kết quả tìm kiếm đ- ợc hiện lên.

Ta có thể dùng lệnh Find trên menu Start để tìm kiếm một máy tính, một ng- ời hay một địa chỉ trên Internet. Để thu hẹp việc tìm kiếm ta có thể nhấp mục Date hoặc Advanced.

### **Làm việc với các tệp dùng th- ờng xuyên**

**Thao tác mở các tài liệu gần đây nhất:** nhấp nút Start, sau đó trở vào mục Documents. Danh sách các tài liệu đã mở gần đây nhất hiện lên. Nháy vào tài liệu trên danh sách, tài liệu đ- ợc mở.

### ***f. Sửa đổi menu Start và Taskbar***

Thao tác sửa đổi menu Start:

- Nháy nút Start, trở vào mục Settings, sau đó nhấp "Taskbar & Start menu". Hộp thoại Taskbar properties xuất hiện.

- Nháy Start Menu Programs.

- Nháy Add hoặc Remove, sau đó theo các h- ướng dẫn trên màn hình.

Ta cũng có thể sắp xếp lại các mục trên menu Start bằng cách kéo các mục tới vị trí mới trên menu Start.

Ta cũng có thể sửa đổi thanh Taskbar cho phù hợp với yêu cầu. Theo ngầm định. Taskbar nằm ở đáy của màn hình nền, nh- ng ta có thể kéo nó tới cạnh bất kỳ của màn hình nền. Ta cũng có thể làm ẩn Taskbar cho đến khi muốn dùng nó, hoặc ta có thể cho hiện Taskbar trên đỉnh của cửa sổ khác.

Cách sửa đổi Taskbar: nhấp nút Start, trở vào mục Settings, sau đó nhấp "Taskbar & Start menu", hộp thoại Taskbar Properties xuất hiện, lựa chọn các tùy chọn Taskbar mong muốn.

Cách di chuyển Taskbar: kéo Taskbar tới cạnh bất kỳ của màn hình nền.

### **g. Thay đổi màn hình nền theo ý thích**

Thao tác mở mục Display của Control Panel:

- Nháy Start, trở mục Settings, sau đó nháy Control Panel. Cửa sổ Control Panel xuất hiện.
- Nháy đúp Display. Hộp thoại Display Properties xuất hiện. Để thiết lập trang trí màn hình, chọn ảnh hay nháy nút Browse. Nháy các mục khác của hộp thoại để cài đặt màn hình chờ, thay đổi màn hình nền và màu sắc cửa sổ, bổ sung các mục của Active Desktop, thay đổi các quyết định về màn hình.

### **h. Máy in**

*Cài đặt máy in.* Trước khi bắt đầu, ta phải đảm bảo rằng máy in đã được nối đúng đắn với máy tính, đã bật điện máy in, biết loại máy in. Để xem các máy in đã cài đặt rồi ta dùng lệnh: Start / Settings / Printers.

*Thao tác cài đặt máy in:*

- Dùng lệnh Start / Settings / Add Printers.
- Theo sự chỉ dẫn trên màn hình. Nếu cài máy in nối trực tiếp với máy tính thì nên chọn LPT1 nh- là một cổng. Nếu cài một máy in mạng, ta phải gõ vào đường dẫn mạng (nếu không biết thông tin này ta nháy nút Browse để định nơi có máy in).
- Khi kết thúc, biểu tượng của máy in hiện lên trong th- mục Printers. Máy in đã sẵn sàng để dùng.

*Thao tác đặt máy in ngầm định:*

- Dùng lệnh Start / Settings / Printers
- Nháy nút phải chuột vào biểu tượng của máy in mà ta muốn đặt là ngầm định. Xuất hiện menu con.
- Chọn Set as Default.

### **i. Tạo đĩa khởi động**

Nh- ta đã thấy, trong MS-DOS lệnh FORMAT cho ta khả năng định dạng một đĩa mềm và đồng thời tạo đĩa hệ thống. Trong hệ điều hành Windows các lệnh DOS có thể tìm thấy trong th- mục C:\WINDOWS\COMMAND. Ta có thể dùng lệnh FORMAT/s A: để định dạng đĩa mềm và tạo đĩa hệ thống. Đĩa hệ thống tạo theo kiểu này cũng chứa 3 tệp: IO.SYS, MSDOS.SYS và COMMAND.COM, nh- ng là của WINDOWS. Bạn có thể dùng đĩa này khởi động máy tính nh- ng chỉ thực hiện được các lệnh trên màn hình DOS. Để có thể khởi động Windows bạn nên tạo đĩa mềm khởi động theo cách sau đây:

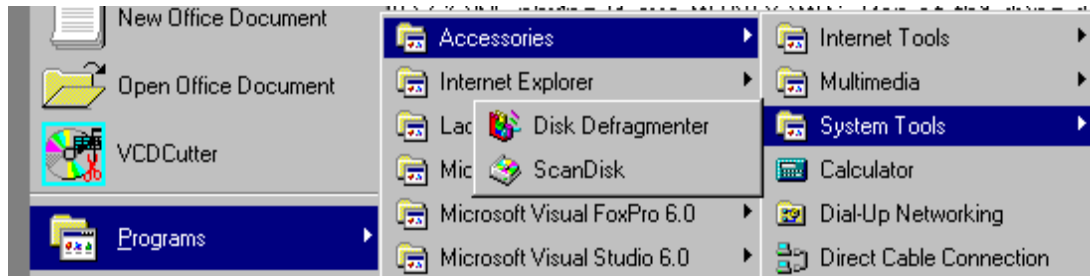
*Thao tác tạo đĩa khởi động:*

- Dùng lệnh Start, Settings, Control Panel, Add/Remove Programs, Startup Disk, Create Disk .

Sau khi tạo xong ta sẽ thấy ngoài 3 tệp trên đây còn có rất nhiều tệp khác. Tuy nhiên vẫn còn thiếu tệp SYS.COM mà bạn nên copy vào đĩa mềm khởi động để sau này khi máy tính bạn có sự cố không khởi động được bạn có thể dùng đĩa này khởi động và dùng lệnh SYS với cú pháp SYS A: C: để tạo lại các tệp hệ thống cho Windows.

***j. Dọn dẹp và kiểm tra đĩa***

Đôi khi ta cần dọn dẹp hoặc kiểm tra đĩa cứng hoặc đĩa mềm. Ta chọn Start, Program, Accessories, System tools; thực đơn sau hiện ra:



Ta chọn và thực hiện mục cần thiết.

## PHẦN III. CÁC CH- ƠNG TRÌNH ỨNG DỤNG

### CH- ƠNG 3. MICROSOFT WORD (WINWORD)

#### 3.1. Tổng quan về Winword

##### 3.1.1. Chức năng của Winword

Winword là phần mềm soạn thảo văn bản chạy trên môi tr- ờng Windows, chuyên dùng để soạn thảo các loại văn bản nh- đơn từ, tài liệu, sách, tạp chí... Winword cho phép ta cùng lúc soạn thảo nhiều văn bản trên nhiều cửa sổ và có thể copy dữ liệu từ cửa sổ này sang cửa sổ khác , thậm chí có thể copy hoặc liên kết với dữ liệu của các ứng dụng khác đang mở.

Sau đây chúng tôi giới thiệu cách sử dụng Winword 97 trên nền Windows 98. Cách sử dụng các phiên bản khác của Winword cũng gần t- ơng tự.

##### 3.1.2. Khởi động Winword



Bạn có thể khởi động Winword bằng nhiều cách, tùy thuộc vào cách thiết lập Windows của máy tính bạn đang dùng. Có thể có các cách nh- sau:

- Nhấp đúp chuột lên biểu t- ợng Microsoft Word trên màn hình nền của Windows (nếu có).
- Nhấp chuột lên biểu t- ợng Microsoft Word trên thanh menu của Microsoft Office (nếu có).
- Nhấp nút chọn **Start**, chọn **Run**, gõ Winword vào dòng trống sau chữ **Open**: rồi chọn **Ok**

Th- ờng thì sau khi Winword đ- ọc khởi động thấy trên màn hình là cửa sổ trống với con trỏ nhấp nháy sẵn sàng để ta nhập văn bản mới.

Ta có thể thực hiện các thao tác sau đây để mở một tệp mới hoặc tệp đã có trên đĩa.

##### Mở một cửa sổ mới:


- Mở một văn bản mới: Chọn **File, New** (Ctrl + N hoặc  ).
- Mở văn bản đã có trên đĩa: Chọn **File, Open** (Ctrl + O hoặc  ).

Tại một thời điểm chỉ có một cửa sổ đ- ọc kích hoạt, muốn làm việc với một cửa sổ khác đã đ- ọc mở: gõ Alt + W, chọn số thứ tự t- ợng ứng với cửa sổ cần kích hoạt.

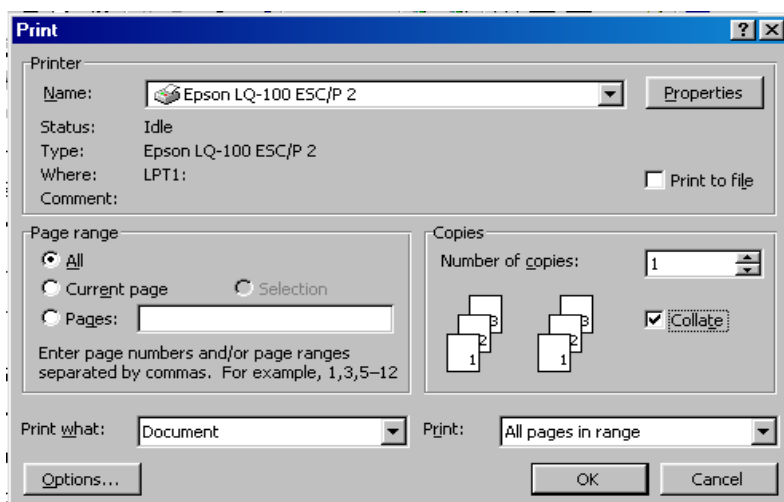
Để tiết kiệm bộ nhớ, nên đóng các cửa sổ không cần thiết (Chọn **File, Close** hoặc nhấp đúp chuột vào hộp điều khiển trên thanh Menu Bar), chỉ giữ lại các cửa sổ cần trao đổi dữ liệu với nhau.

##### 3.1.3. In văn bản:


Có thể in văn bản theo hai cách.

- Gõ **Ctrl + P** (hoặc nhấp chuột trên biểu t- ợng  ).
- Chọn **File, Print...** cửa sổ sau hiện ra






Ta có thể chọn các tham số thích hợp.

Chú ý, tr- ớc khi in có thể xem tr- ớc kết quả in trên màn hình bằng cách thực hiện lệnh **File, Print Preview** (hoặc nhấp chuột trên biểu t- ợng  ).

### 3.1.4. L- u cất văn bản Winword

Tệp văn bản Winword có đuôi \*.DOC và ta có thể l- u cất nh- sau:

Nếu tệp văn bản đã có tên và ta không muốn thay đổi ta có thể l- u văn bản đang soạn thảo bằng cách nhấp chuột vào **File, Save** hoặc nhấp vào biểu t- ợng  trên thanh công cụ. Nếu ta muốn đặt tên mới cho tệp văn bản ta nhấn **File, Save as...** rồi nhập tên mới vào.

### 3.1.5. Thoát khỏi Winword

Bạn có thể thoát khỏi Winword bằng nhiều cách nh- sau:

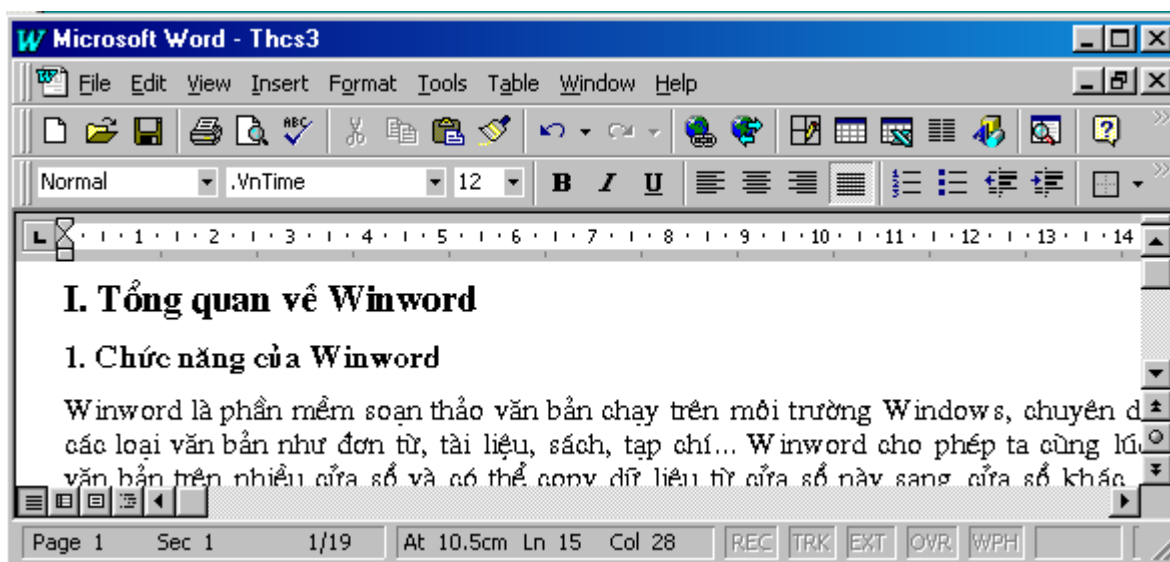
- Nhấp chuột lên mục **F**ile của thanh menu rồi chọn **E**xit .
- Nhấp chuột lên dấu **C**lose (X) ở góc phải phía trên cửa sổ Winword .
- Nhấn tổ hợp phím **Alt+F4**

Nếu Word hỏi bạn có muốn l- u cất tài liệu không, bạn sẽ chọn Yes hoặc No tùy thuộc vào ý của bạn.

## 3.2. Làm việc với Winword

### 3.2.1. Màn hình làm việc của Winword

Màn hình làm việc của Winword có thể có dạng sau:



Màn hình của Winword ở các máy có thể khác nhau chủ yếu là do ta cho hiện trên màn hình những thanh công cụ nào, đặt chúng ở đâu, có cho hiện th- ớc không. Ta có thể chọn hoặc không chọn các thanh công cụ bằng cách nhấn **View**, **Toolbars**, rồi đánh dấu hoặc thôi đánh dấu các thanh công cụ (Toolbars), chọn hoặc không chọn th- ớc (Ruler). Ta cũng có thể nhấn chuột để kéo các thanh công cụ đến các vị trí khác nhau trên màn hình. Màn hình Word th- ờng có các phần sau đây theo thứ tự từ trên xuống:

- **Title bar** (Thanh tiêu đề): dòng chứa tên của tệp văn bản.

Góc trái thanh tiêu đề ta thấy biểu t- ợng của Winword, nhấp vào đó sẽ kích hoạt hộp điều khiển. Góc phải là các nút Minimize, Restore hoặc Maximize và nút Close của cửa sổ Winword.

- **Menu bar** (thực đơn hàng ngang): dòng chứa các lệnh của Winword, mỗi lệnh ứng với một thực đơn dọc (Menu PopUp).

Góc trái thanh **Menu Bar** ta thấy biểu t- ợng của cửa sổ t- liệu, nhấp vào đó sẽ kích hoạt hộp điều khiển. Góc phải là các nút Minimize, Restore hoặc Maximize và nút Close của cửa sổ t- liệu.

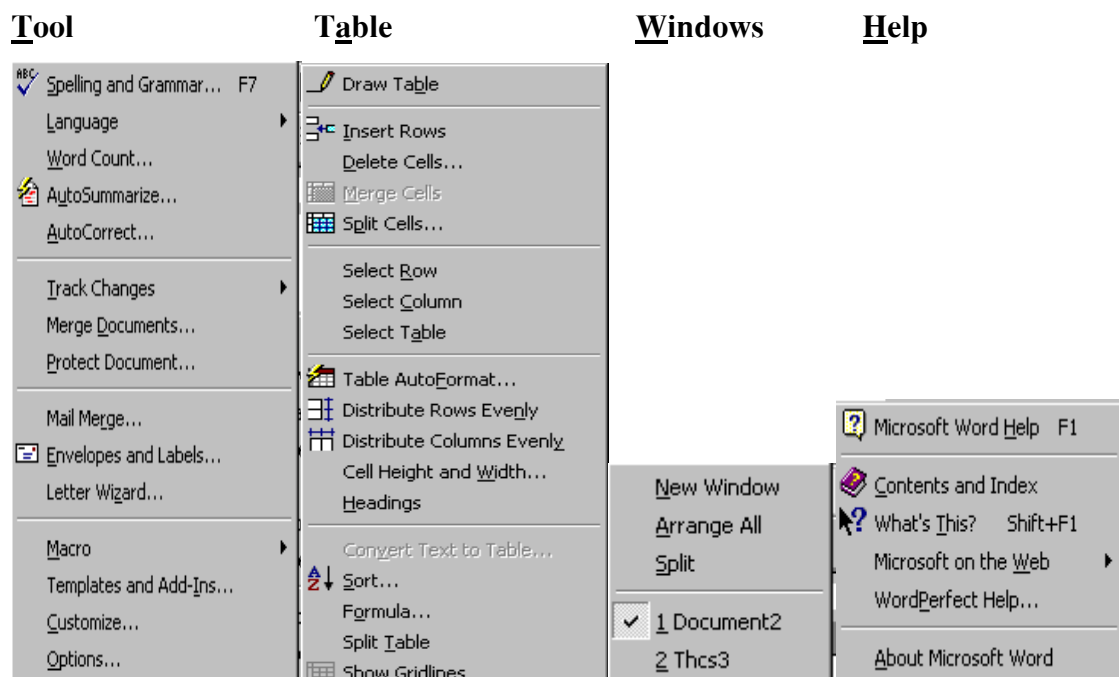
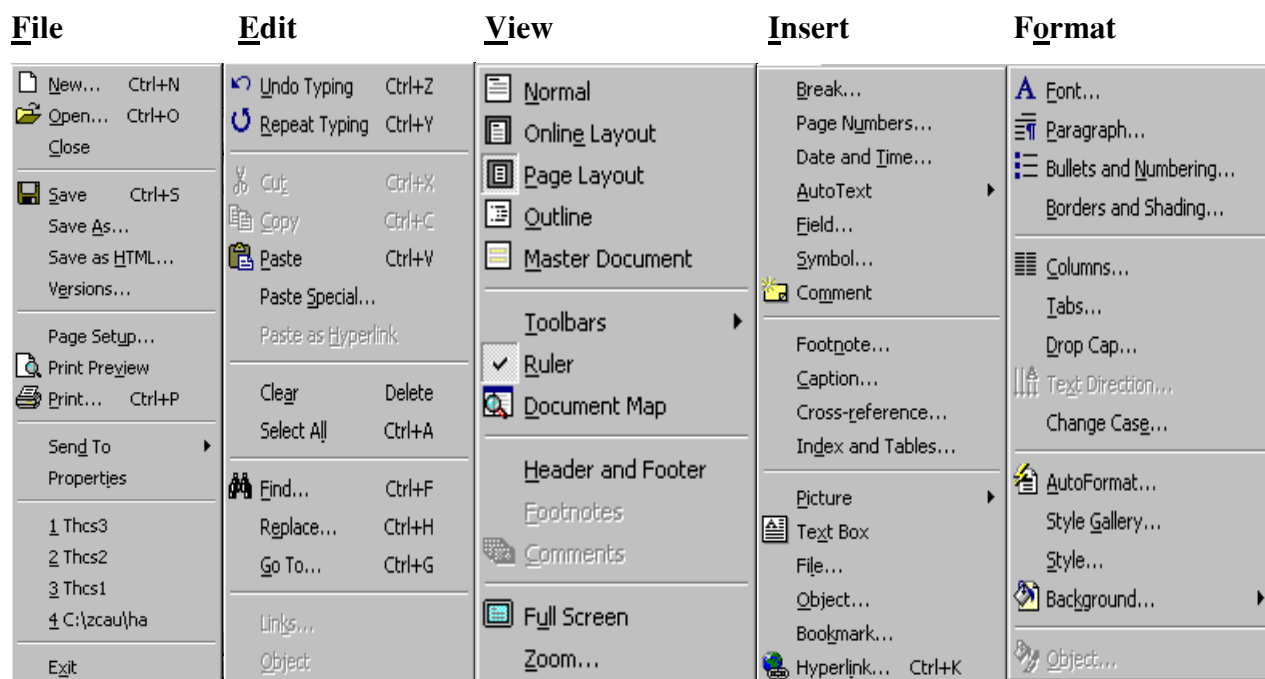
**Thao tác để mở một thực đơn dọc :**

- Chỉ con trỏ chuột vào tên thực đơn dọc trên Menu Bar rồi bấm nút trái chuột.
- Hoặc gõ phím **F10** hoặc nhấn **Alt** rồi dùng phím mũi tên di chuyển vệt sáng đến thực đơn dọc nhấn {ENTER}.


**Trong các thực đơn dọc :**

- Các lệnh có màu xám nhạt là lệnh tạm thời không thực hiện đ- ợc
- Các ký tự đ- ợc gạch d- ới thì có thể chọn lệnh t- ợng ứng bằng cách bấm ký tự đó.
- Các lệnh có ghi kèm tổ hợp phím thì có thể chọn nhanh bằng cách bấm tổ hợp phím đó (Ví dụ: có thể l- u văn bản lên đĩa bằng cách bấm **Ctrl+ S**).
- Các lệnh có dấu ba chấm (...) phía sau cho biết sẽ có một hộp thoại xuất hiện khi lệnh đ- ợc chọn.

Sau đây là các thực đơn dọc của thanh menu.



- **Tools bar** (thanh công cụ): chứa một số biểu tượng (Icon) thể hiện một số lệnh thông dụng. Thay vì phải vào các hộp menu để chọn lệnh, chỉ cần nhấp chuột lên biểu tượng của lệnh tương ứng.

Ví dụ: Muốn lưu văn bản lên đĩa thay vì vào Menu File chọn Save, chỉ cần nhấp chuột lên biểu tượng 

- **Formatting Bar** (Thanh định dạng): chứa các hộp và biểu tượng dành cho việc định dạng văn bản như kiểu, loại font, cỡ font, căn lề...

- **Ruler** (Thước): hiển thị thước theo chiều ngang, dọc văn bản. Ruler có thể tắt hay mở bằng View/Ruler. Để thay đổi đơn vị đo (Inches/Cm/Points/Picas) trên Ruler, ta thực hiện lệnh Tools/Options/General rồi vào hộp Measurements để thay đổi.

- **Text Area** (Vùng văn bản): đây là vùng lớn nhất dùng để nhập văn bản.
- **Scroll bars** (Các thanh cuộn): Có 2 thanh cuộn ở d- ưới và bên phải văn bản dùng để cuộn nhanh văn bản lên xuống hoặc sang trái, sang phải.
- **Status bar** (Thanh trạng thái): hiển thị tình trạng của cửa sổ hiện tại nh- số thứ tự trang, thứ tự cửa sổ, tổng số trang, vị trí con trỏ (dòng, cột), giờ; tình trạng của Capslock, Numlock... Thanh này nằm ở dòng cuối cùng màn hình.

Tại bất cứ vị trí nào trên màn hình ta cũng có thể kích hoạt **Shortcut Menu** bằng cách nhấp nút phải của chuột. Shortcut Menu sẽ có nội dung t- ơng ứng với vị trí con trỏ.

### 3.2.2. Vùng văn bản (Text area)

Vùng văn bản là vùng ta soạn thảo, nhập mới và sửa đổi văn bản. Văn bản th- ờng có thiết lập ngầm định lề trái, phải, trên d- ưới, kiểu chữ (Style), phong chữ (Font), cỡ phong (Font Size). Ta có thể thay đổi các thiết lập này cho phù hợp.

Nh- các hệ soạn thảo khác, trong Word cũng có các khái niệm Từ (Word), Câu (Sentence), Dòng (Line), Đoạn (**Paragraph**), Trang (Page) và Phần (**Section**). Cách đặt lề phải, trái, khổ giấy từ lệnh **File, Page Setup...** sẽ quy định độ dài của dòng và số dòng trong một trang. Khi ta gõ hết một dòng Word sẽ *tự động xuống dòng*, và khi hết một trang Word sẽ *chèn dấu phân trang mềm* ta có thể nhìn thấy trong chế độ View, Normal. Ta có thể chèn dòng phân trang cứng bằng cách nhấn tổ hợp phím **Ctrl+ENTER** hoặc nhấn **Insert, Break..., Page Break** trên thanh menu. Ta có thể xóa phân trang cứng bằng cách chuyển con trỏ đến dòng này và nhấn **DEL**. Mỗi lần ta nhấn **ENTER** là *bắt đầu một đoạn mới*. Nhiều cách định dạng nh- khoảng cách từ mép văn bản đến lề và khoảng cách giữa các dòng (Indent and Spacing), Drop cap, canh đoạn văn bản thẳng bên trái, bên phải hoặc thẳng cả 2 bên... đ- ọc đặt ngầm định cho đoạn chứa con trỏ. Chính vì vậy khi gõ văn bản ta chỉ nên gõ ENTER khi bắt đầu đoạn văn bản mới. Đoạn có thể chứa ngắt trang mềm, còn ngắt trang cứng luôn luôn bắt đầu một đoạn mới. Khi ta soạn thảo những tài liệu lớn nh- các cuốn sách dày có thể ta muốn trình bày từng phần lớn theo những khuôn dạng khác nhau. Trong tr- ờng hợp này ta chọn **Insert, Break..., Section Break** để phân tài liệu ra thành từng phân đoạn. Mục **File, Page Setup...** đ- ọc đặt ngầm định cho phân đoạn chứa con trỏ. Tuy nhiên nói chung tài liệu của chúng ta chỉ gồm một phân đoạn, do đó trong tài liệu h- ớng dẫn này chúng tôi không nói đến định dạng phân đoạn.

**Khối văn bản đ- ọc đánh dấu:** ở trên chúng tôi đã nói đến các đoạn, phân đoạn... và một số định dạng đặt ngầm định cho các vùng này. Tuy nhiên một vài định dạng nh- phong chữ, cỡ phong, chữ nghiêng, đậm, gạch d- ưới... lại chỉ có hiệu lực tại vị trí hiện thời của con trỏ và các ký tự sắp gõ vào. Nếu ta muốn định dạng có hiệu lực ở một vùng văn bản nào đó ta phải đánh dấu tức là bôi đen vùng đó tr- ớc khi thực hiện một thao tác định dạng. Khối đ- ọc đánh dấu sẽ là phần đ- ọc - u tiên nhất. Thí dụ nếu ta đánh dấu vùng văn bản rồi thực hiện chỉnh lại Indent, khi đó tất cả các đoạn có chứa phần bôi đen đều đ- ọc chỉnh Indent.

Ta có thể đánh dấu khối văn bản bằng 3 cách:

- Nhấn và giữ chuột trái ở đầu khối văn bản rồi kéo đến cuối khối.
- Nhấn và giữ phím SHIFT ở đầu khối rồi nhấn phím ↓ đến cuối khối.
- Nhấn và giữ phím SHIFT ở đầu khối rồi nhấp chuột ở cuối khối.

**Đánh dấu toàn bộ văn bản: Ctrl+A**

**Di chuyển con trỏ:**

**Home** về đầu dòng

**End** về cuối dòng

**PgUp** lên 1 trang màn hình

**PgDn** xuống 1 trang màn hình

**Ctrl+Home** về đầu văn bản

**Ctrl+End** về cuối văn bản

**F5** chuyển đến một trang nào đó

Ta có thể sử dụng cách thao tác trên để đánh dấu vùng văn bản.

**Tìm kiếm chuỗi ký tự:**

**Ctrl+F**

**3.2.3. Nhập và điều chỉnh văn bản**



**a. Thao tác với khối đã đánh dấu** (tức là đã bôi đen)

**Chép một khối:**

**Chép khối bằng thực đơn dọc:**

- Nhấn **Edit, Copy** trên thực đơn dọc (hoặc gõ **Ctrl+C**)
- Di chuyển con trỏ đến vị trí mới
- Nhấn **Edit, Paste** trên thực đơn dọc (hoặc gõ **Ctrl+V**)

**Chép khối bằng Tools Bar:**

- Nháy chuột trên biểu tượng  của Tools Bar
- Di chuyển con trỏ đến vị trí mới
- Nháy chuột trên biểu tượng  của Tools Bar

**Bằng chuột:**



- Đưa con trỏ chuột vào vùng bôi đen cho đến khi con trỏ có hình mũi tên, nhấn phím Ctrl đồng thời nhấn chuột trái kéo đến nơi cần chuyển đến rồi nhả chuột.

**Chuyển một khối:**

**Bằng thực đơn dọc:**

- Nhấn **Edit, Cut** trên thực đơn dọc (hoặc gõ **Ctrl+X**)
- Di chuyển con trỏ đến vị trí mới
- Nhấn **Edit, Paste** trên thực đơn dọc (hoặc gõ **Ctrl+V**)

**Bằng Tools Bar:**

- Nháy chuột trên biểu tượng  của Tools Bar
- Di chuyển con trỏ đến vị trí mới
- Nháy chuột trên biểu tượng  của Tools Bar

### Bảng chuột:

- Đ- a con trỏ chuột vào vùng bôi đen cho đến khi con trỏ có hình mũi tên, nhấn chuột trái kéo đến nơi cần chuyển đến rồi nhả chuột.

### Xoá một khối:

Gõ phím **Delete**

Ghi chú:

- ☐ nghĩa của Clipboard: là một vùng nhớ tạm dùng để l- u trữ dữ liệu khi thực hiện thao tác Copy hay Cut.

### **b. Định dạng văn bản**

#### Định dạng ký tự:

Nếu ta đánh dấu khối sau đó tiến hành định dạng thì định dạng này sẽ tác động lên toàn khối. Nếu không có đoạn văn bản nào đ- ợc đánh dấu thì chỉ các ký tự đ- ợc nhập sau thao tác định dạng mới bị tác động.

Thay đổi mẫu ký tự (**Font**): nháy chuột vào mũi tên xuống trong hộp tên Font, một danh sách các tên Font hiện ra, từ đó chọn Font cần sử dụng.

Thay đổi cỡ ký tự (**Font size**): nháy chuột vào mũi tên xuống trong hộp Font size, chọn số chỉ cỡ Font.

Thay đổi kiểu ký tự (**Font Style**):

Đậm (Bold) **Ctrl+B** - Nghiêng (Italic) **Ctrl+I** - Gạch d- ới (Underline) **Ctrl+U**.

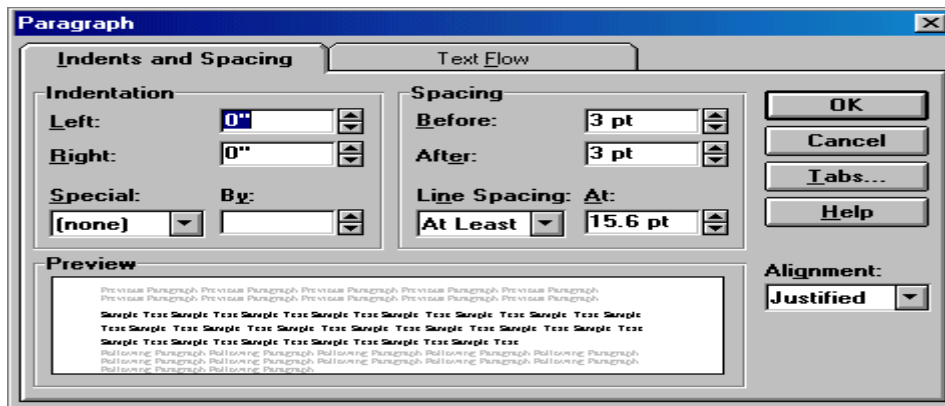
Hoặc nhấp chuột vào các biểu t- ơng trên thanh định dạng

Để định dạng đầy đủ, chọn **Format, Font** hộp đối thoại xuất hiện để ta có thể chọn các mục.

Nếu chọn nút Default... thì các thông số định dạng sẽ đ- ợc lấy làm trị mặc nhiên cho mỗi lần khởi động Word hoặc mở một tài liệu mới.

**c. Định dạng Paragraph:** cho phép thay đổi cách hiển thị của một paragraph. Nếu ta bôi đen miền văn bản rồi mới tiến hành định dạng Paragraph thì định dạng sẽ đ- ợc thực hiện trên tất cả các paragraph có chứa phần bôi đen. Nh- vậy phần văn bản bôi đen bao giờ cũng đ- ợc - u tiên tr- ớc hết trong các thao tác định dạng.

Chọn **Format, Paragraph** hộp đối thoại sau xuất hiện:



Để canh đều biên trái, phải hoặc cả 2 bên ta cũng có thể nhấp chuột vào các biểu tượng sau:



#### ***d. Các loại định dạng khác trên phần văn bản đánh dấu***

##### **Tạo khung và làm nền**

Để tạo khung và làm nền cho một đoạn văn bản, tr-ớc tiên: chọn đoạn văn bản cần tạo, sau đó vào thực đơn dọc chọn **Format, Borders and Shading**.

Hộp hội thoại xuất hiện và ta có thể chọn các kiểu khung hoặc nền.

Chú ý, có thể nhấp chuột vào biểu tượng Border  : xuất hiện thanh công cụ Border và chọn kiểu khung.

##### **Định khoảng cách Tab Stop**

Mỗi lần gõ phím Tab, con trỏ sẽ dừng 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 thay đổi chiều dài của Tab Stop có thể thực hiện bằng cách chọn **Format, Tabs...**

Hoặc nhấp chuột vào biểu tượng Tab trên thanh công cụ.

##### **Định số cột trên toàn văn bản**

Có thể trình văn bản trên chiều cột (nh- dạng bài báo bằng thao tác:



Chọn **Format, Columns** rồi gõ số cột tương ứng trong hộp Number of Columns

Hoặc nhấp chuột vào biểu tượng cột trên thanh công cụ.

##### **Bullets và Numbering**

Có thể xác định các dấu chấm đen hình tròn (Bulleted) hay đánh số (Numbering) hay đánh ký tự Alphabet ở đầu các đoạn.

Cách 1. Dùng ký tự trên Tools Bar:

- Chọn (bôi đen) các đoạn muốn đặt Bullets hay Numbering (B/N)
- Nhấp chuột lên biểu tượng  để đặt Bulleted hay biểu tượng  để đặt Numbering.

Cách 2. Dùng thực đơn dọc:

- Chọn các đoạn muốn đặt Bullets hay Numbering
- Thực hiện lệnh **Format, Bulleted and Numbering**:  
chọn Bulleted hoặc Numbered hoặc Multilevel.

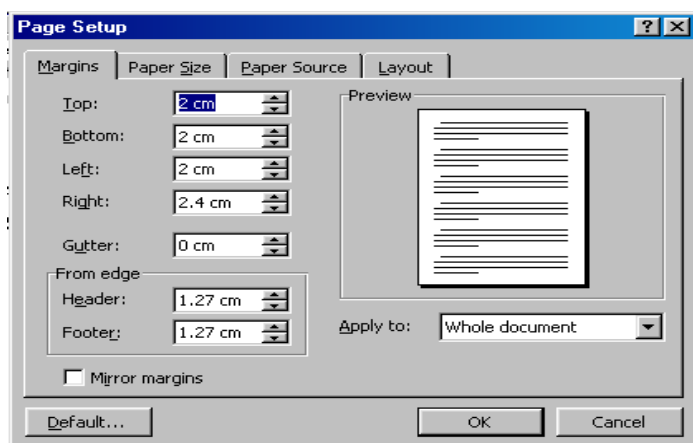
**Bật/tắt chế độ đặt Bullets hay Numbering**: Nếu ta muốn đặt B/N ngay khi gõ văn bản ta nhấp chuột vào biểu tượng trên đây. Sau đó nếu muốn tắt B/N ta gõ {ENTER} 2 lần.

#### ***3.2.4. Trình bày trang***

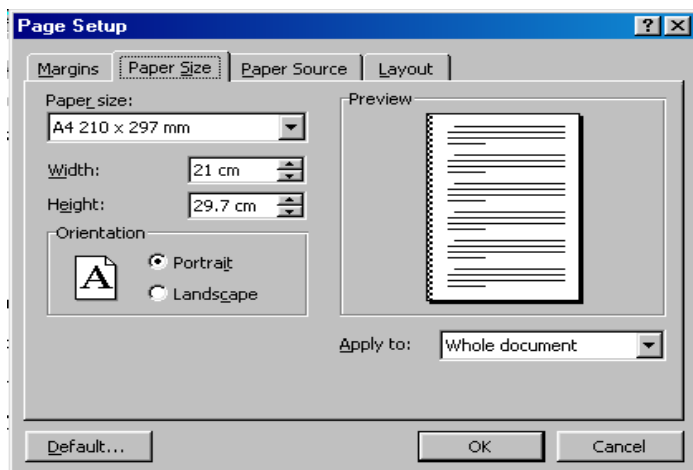
##### ***a. Định dạng trang***

Chọn **File, Page Setup** hộp đối thoại hiện ra và ta có thể chọn lề, cỡ giấy...

Cửa sổ **Margins**



### Cửa sổ Paper Size



### Cách đặt lề bằng th- ớc đo

Ph- ơng pháp này th- ờng kết hợp giữa các thao tác xử lý chuột và cây th- ớc ngang trên màn hình (việc sử dụng phím mất nhiều thời gian và phức tạp hơn). Việc đặt lề ảnh h- ưởng đến đoạn văn bản hiện thời. Trên th- ớc sẽ có các mẫu hình tam giác quy định vị trí lề hiện tại, muốn thay đổi chỉ cần rê chuột lên các mẫu t- ơng ứng, di chuyển đến vị trí mới rồi thả chuột. Phía trái th- ớc gồm ba phần: hai mẫu hình tam giác phía trên và một mẫu hình vuông bên d- ưới.



- Mẫu tam giác trên: đặt Indent trái cho dòng đầu tiên trong đoạn.



- Mẫu tam giác d- ưới: đặt Indent trái cho các dòng còn lại.

- Mẫu hình vuông: đặt Indent trái cho cả đoạn.

- Để thay đổi Indent trái cho toàn bộ văn bản, di con trỏ chuột đến phần tiếp giáp giữa hai mẫu tam giác cho đến khi xuất hiện mũi tên hai chiều ↔ rồi rê chuột để thay đổi. Mỗi khi rê chuột, sự thay đổi Indent sẽ hiện ra ngay trên màn hình.

### Đặt tiêu đề

Tiêu đề đầu trang và cuối trang là văn bản đ- ọc in trên đầu (hoặc cuối) mỗi trang của tài liệu. Th- ờng 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 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:

- Chọn **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



Chuyển đổi tiêu đề đầu/cuối trang





Xem Header (hoặc Footer) kế sau



Thực hiện chế độ chọn Header và Footer như ở các trang trước.



Chèn vào số trang  : chèn ngày hiện tại  : chèn giờ hiện tại



Cho/không cho phép hiển thị văn bản của trang hiện tại

Cuối cùng, Click nút Close để trở về chế độ soạn thảo thông thường.

Để kích hoạt chế độ sửa đổi tiêu đề, có thể nhấp đúp chuột vào phần ghi tiêu đề (phần chữ nhạ trong chế độ Page Layout).

### **Đánh số trang**

Chọn **Insert, Page Numbers**. Hộp thoại xuất hiện và ta có thể đặt các tùy chọn.

### **3.2.5. Trình bày màn hình**

#### **a. Các kiểu xem trong tài liệu**

Có thể sử dụng bốn chế độ khác nhau để xem tài liệu trên cửa sổ màn hình.

Từ thực đơn dọc **View** chọn một trong bốn kiểu sau:

- Normal: chế độ bình thường

Trong chế độ này, các trang ngăn cách nhau bởi dòng chấm chấm nằm ngang.

- Outline :chế độ tổng quan thường, dùng để xem những mục tiêu đề lớn của tài liệu, giấu đi các chi tiết nhỏ hoặc chuyển chúng đến một vị trí mới.

- Page Layout :chế độ trình bày trang.

- Master Document (chế độ tổng quan tài liệu): tương tự như chế độ Outline nhưng ở mức cao hơn. ☐ chế độ này có thể thao tác thêm cả phần tài liệu con, kết hợp nhiều tệp vào một tài liệu lớn...

#### **b. Xem toàn màn hình**

Có thể cho vùng làm việc của người dùng nhiều hơn bằng cách bỏ hết các thanh Menu, thanh công cụ... Để thực hiện việc này, chọn View-Full Screen. Trở lại màn hình cũ bằng cách gõ Esc hay nhấp chuột vào hộp Full Screen.


#### **c. Phóng to, thu nhỏ màn hình**

- Nhấp chuột vào biểu tượng Zoom Control hoặc vào thực đơn dọc View, chọn Zoom.

- Chọn một trong các con số quy định % văn bản được hiển thị hoặc tăng giảm con số tùy ý trong hộp Percent.

#### **d. Hiển thị tài liệu trước khi in**

Để xem tổng quát toàn bộ tài liệu trước khi in, có thể thực hiện bằng hai cách:

- Nháy chuột vào biểu tượng Print Preview  trên thanh công cụ.
- Vào menu File, chọn Print Preview.

Nháy chuột vào biểu tượng Close để thoát khỏi chế độ Print Preview.

#### ***e. Bật tắt việc hiển thị thước, các thanh công cụ***

- Thước: vào thực đơn dọc View, chọn Ruler.
- Các thanh công cụ: vào thực đơn dọc View, chọn Toolbars. Hộp lựa chọn hiện ra cho phép đặt/bỏ dấu X vào ô tương ứng của thanh công cụ.

### ***3.2.6. Lập bảng biểu (Table)***

#### ***a. Tạo một bảng mới***

Các bước tiến hành để tạo một bảng mới:

- Đặt con trỏ đến nơi cần tạo bảng
- Chọn **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 Column Width. Nếu để Auto, chiều rộng mỗi cột tùy thuộc vào số lượng cột và chiều rộng trang in.
- Cuối cùng, chọn OK

Ta cũng có thể nháy chuột vào biểu tượng **Insert table** trên thanh công cụ rồi rê chuột để chọn số cột và số dòng.

#### ***b. Các thao tác trong bảng***

- 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	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 của cột	Alt + PgDn

Muốn đánh dấu đoạn nào trong bảng, có thể rê chuột hoặc bấm Shift kèm với các phím mũi tên.

#### ***c. 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 trắng xuất hiện thì nháy chuột.
- Chọn một hàng: rê chuột từ ô đầu đến ô cuối dòng (hoặc vào thực đơn dọc Table, chọn Select Row).

- Chọn một cột: di chuyển con trỏ chuột lên mép trên của bảng đến khi thấy mũi tên đen chỉ xuống thì Click chuột (hoặc vào thực đơn dọc Table, chọn Select Column).
- Muốn chọn nhiều ô trong bảng thì chọn một ô đầu tiên, sau đó di chuyển chuột đến ô cuối, giữ phím Shift đồng thời nhấn chuột.
- Muốn **chọn toàn bộ bảng**: đ- a con trỏ vào trong bảng, vào thực đơn dọc Table chọn Select Table (hoặc bấm Alt và số 5 bên bàn phím số).

#### **Chèn thêm các ô, hàng, cột vào bảng**

- Chọn ô hoặc các ô muốn chèn thêm các ô mới vào kế sau đó
- Vào thực đơn dọc Table, chọn Insert Cells. Hộp hội thoại Insert Cells xuất hiện. Chọn một trong các chức năng sau:
  - Insert Entire Row: chèn toàn bộ hàng hay các hàng
  - Insert Entire Column: chèn toàn bộ cột hay các cột.
- Chọn OK.

#### **Xoá các ô, hàng hoặc cột trong bảng**

- Trước hết chọn một ô bất kỳ trong hàng hoặc cột cần xoá.
- Chọn Delete Cells từ thực đơn dọc Table.
- Chọn Delete Entire Row: xoá toàn bộ dòng
- Delete Entire Column: xoá toàn bộ cột
- Chọn OK.

#### **Di chuyển, sao chép các ô, hàng, cột trong bảng**

Thực hiện sao chép nh- với các khối văn bản.

#### **Thay đổi chiều rộng cột**

Cách 1: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.

Cách 2: Chọn cột cần thay đổi

- Vào thực đơn dọc **Table**, chọn **Cell Height And Width**
- Gõ con số chỉ chiều rộng cột vào mục Width of Column.
- Chọn <OK>

#### **Thay đổi độ cao của hàng**

- Chọn một hoặc nhiều hàng cần thay đổi
- Chọn Cell Height and Width từ thực đơn dọc Table.
- Chọn Row rồi thực hiện một trong các thao tác:
  - Chọn Auto trong mục Height of Rows: điều chỉnh độ cao theo độ cao văn bản của hàng
  - Chọn At Least trong mục Height of Rows, rồi đ- a vào giá trị số chỉ độ cao ít nhất của hàng. Nếu nội dung trong ô vượt quá con số đ- a ra thì Word sẽ điều chỉnh lại.
  - Chọn Exactly trong mục Height of Rows, rồi đ- a vào giá trị số chỉ độ cao chính xác của hàng. Nếu nội dung trong ô lớn hơn thì sẽ chỉ bị mất một phần.

- Chọn <OK>

### **Gộp, tách các cột trong bảng**

Gộp nhiều cột lại thành một:


- Chọn các cột cần gộp (có thể gồm nhiều hàng)
  - Vào thực đơn dọc Table, chọn Merge Cells
- Khi đó, các đường ngăn cách giữa các cột sẽ mất đi, tạo thành một cột duy nhất.

Tách một cột thành nhiều cột nhỏ:

- Chọn cột cần tách (có thể gồm nhiều hàng)
- Vào thực đơn dọc Table, chọn Split Cells. Hộp thoại Split Cells xuất hiện.
- Nhập vào con số chỉ số cột cần tách.

Chiều rộng mỗi cột được tạo ra bằng chiều rộng cột cũ chia cho số cột.

### **Tạo các đường kẻ bảng**

Các đường phân cách trong bảng có dạng nét đứt trên màn hình, nhưng khi in ra sẽ không có. Nếu cần có thể thêm vào các đường kẻ thay cho đường nét đứt. Trước hết cần cho hiển thị thanh công cụ kẻ khung trên màn hình bằng cách nhấp chuột vào biểu tượng Border . Thanh công cụ kẻ khung (Border) xuất hiện và ta có thể chọn các kiểu đường kẻ.

### **Sắp xếp các dữ liệu trên một bảng**

Có thể sắp xếp các hàng trong bảng theo một cột nào đó. 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.
- Vào menu **Table**, chọn **Sort**. Hộp thoại Sort xuất hiện.
- Chọn các khóa sắp xếp trong hộp Sort by và Then by.
- Chọn sắp xếp theo kiểu dữ liệu nào (Type): kiểu ký tự (Text), kiểu số (Number) hay kiểu ngày (Date).
- Chọn thứ tự sắp xếp: tăng dần (Ascending) hoặc giảm dần (Descending)
- Chọn <OK> hay gõ Enter.

### **Tính toán cuối cột số trong một bảng**

- Di chuyển con trỏ đến ô cuối cùng của cột số.
- Thực hiện lệnh **Table, Formula...**

#### **3.2.7. Kẻ bảng (draw table)**

Đây là mục mới trong Word 97. Dùng chức năng này ta có thể vẽ những bảng phức tạp. Ta nhấp chuột lên nút **Table and Border** trên thanh công cụ để bắt đầu vẽ bảng.

### 3.2.8. Các ứng dụng đặc biệt

#### a. Chèn thêm các ký tự đặc biệt

Word cho phép người sử dụng có thể chèn thêm nhiều ký hiệu, font chữ đặc biệt mà trên bàn phím không có, như: ký hiệu các con bài tây ( $\clubsuit \diamondsuit \heartsuit \spadesuit$ ), các mẫu tự Hy Lạp ( $\alpha \beta \chi \delta \varepsilon \phi \dots$ ), Các ký hiệu toán học ( $\pm \neq \equiv \approx \cap \cup \supseteq \subset \Sigma \in \dots$ )

Để chèn thêm một ký tự đặc biệt, thực hiện theo trình tự sau:

- Mở cửa sổ văn bản đến nơi cần chèn
- Vào thực đơn Insert, chọn Symbol, hộp thoại Symbol xuất hiện.
- Nháy chuột vào hộp Font để chọn bộ Font chứa các ký hiệu.
- Nháy chuột vào ký hiệu cần chèn.
- Nháy nút Insert để chèn, nháy nút Close để đóng hộp thoại Symbol.

Trong hộp thoại hiển thị sử dụng một số ký hiệu đặc biệt nào đó, nên định nghĩa cho nó một tổ hợp phím theo trình tự sau:

- Vào thực đơn Insert, chọn Symbol, chọn ký hiệu cần định nghĩa.
- Nháy nút Shortcut Key, hộp thoại Customize xuất hiện
- Cửa sổ lúc này nằm ở hộp Press New Shortcut Key, bấm tổ hợp phím cần định nghĩa.
- Nháy nút Assign, nháy nút Close để trở về hộp Symbol.
- Chọn một ký hiệu khác để định nghĩa hoặc nháy nút Close để đóng hộp Symbol.

Kể từ đó, chỉ cần bấm tổ hợp phím đã định nghĩa thì ký hiệu tương ứng sẽ xuất hiện tại vị trí con trỏ. Tất cả các định nghĩa được cất giữ trong tên \*.DOT khi thoát khỏi Word.

#### b. Tạo các hiệu ứng đặc biệt nhờ WordArt

WordArt là phần chuyên dùng để tạo các hiệu ứng đặc biệt đối với văn bản trình bày.

Cách thực hiện:

- Định vị trí con trỏ tại nơi muốn chèn.
- Vào thực đơn chọn **Insert, Picture, WordArt**. Hộp chứa các mẫu chữ xuất hiện, ta có thể chọn.

#### c. Tạo chữ cái lớn đầu dòng

Chọn **Format, Drop Cap**.

### 3.2.9. Chèn hình ảnh vào văn bản

Dùng để đưa các hình ảnh được lưu bởi các tệp trên đĩa (\*.BMP, \*.PCX, \*.DIB, \*.WPG ...) vào văn bản đang soạn.

Các bước thực hiện:

- Mở cửa sổ đến vị trí cần chèn hình ảnh.
- Vào thực đơn Insert chọn Picture, hộp thoại Insert Picture xuất hiện

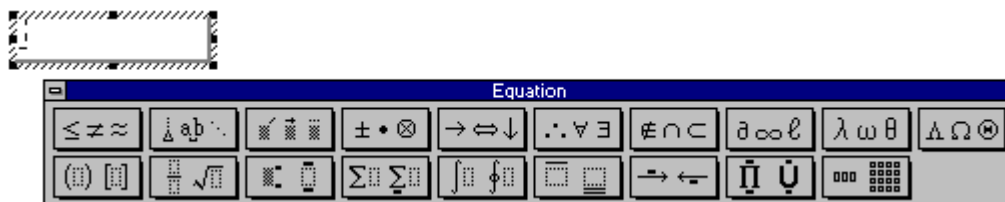
- Chọn Directory chứa hình ảnh, chọn tên tập tin ảnh, chọn <OK>

### 3.2.10. Microsoft Equation

Chuyên dùng để tạo các công thức toán và khoa học phức tạp.

Cách thực hiện:

- Đặt con trỏ nơi muốn gõ công thức.
- Vào thực đơn **Insert**, chọn **Object**, chọn **Creat New** rồi nháy đúp chuột vào mục Microsoft Equation 3.0 (hoặc nháy chuột vào biểu tượng  nếu có). Trên màn hình hiện ra:




Ta có thể nhấp vào các biểu tượng để bắt đầu gõ công thức.

### 3.2.11. Các công cụ đồ họa (Drawing)

Trước khi sử dụng các chức năng vẽ đồ họa, cần chuyển sang chế độ Page Layout (trình bày trang). Các đối tượng đồ họa có thể tồn tại độc lập với các phần văn bản, do có việc tạo các hình vẽ, ký hiệu, sơ đồ... trong văn bản dễ dàng hơn.

#### a. Tạo một đối tượng đồ họa

Nếu trên màn hình chỉ có thanh công cụ vẽ đồ họa, thì nháy đúp vào biểu tượng Drawing  để hiện thanh công cụ vẽ đồ họa.

Chuyển con trỏ chuột đến gần các biểu tượng sẽ hiện ra hộp giải thích.

#### b. Thay đổi các đối tượng đã vẽ

Trước khi thay đổi các đối tượng đã vẽ cần chọn đối tượng bằng cách nhấp vào đối tượng đó.

**Thay đổi vị trí.** Bấm các phím mũi tên để di chuyển hoặc thực hiện các bước sau:

- Di chuyển chuột vào đối tượng cho đến khi có hình mũi tên bốn hướng xuất hiện.
- rê chuột đến vị trí mới rồi thả nút chuột.

### **Thay đổi hình dạng, kích th- ớc:**

- Di trỏ chuột đến một trong các ô vuông nhỏ cho đến khi xuất hiện dấu + hoặc mũi tên hai chiều.
- Rê chuột để thay đổi hình dạng, kích th- ớc rồi thả nút chuột.

### **Sao chép, cắt, xoá đối t- ợng:**

Tr- ớc hết đánh dấu đối t- ợng

- Xóa cả đối t- ợng: Bấm phím Delete.
- Cắt một phần đối t- ợng: Cho hiện thanh **Picture** , đánh dấu đối t- ợng rồi chọn biểu t- ợng Crop.

### **3.2.12. Định nghĩa từ gõ tắt**

Gõ **Tools, AutoCorrect** để chọn cách gõ tắt

### **3.2.13. Template (mẫu văn bản)**

Template là một mẫu văn bản đã soạn sẵn. Chẳng hạn, phần văn bản cố định của một công văn, th- từ ta nên soạn sẵn và ghi vào đĩa d- ới dạng Template. Khi cần thảo một công văn mới ta chỉ cần gọi mẫu công văn này ra và điền vào những chỗ để trống và nội dung thích hợp.

#### **a. Xem một Template sẵn có của WinWord**

Dùng lệnh File/New, xuất hiện hộp thoại New, chọn nút tròn Template, d- ới hộp Template máy liệt kê rất nhiều mẫu có sẵn của Winword, muốn chọn một mẫu ta nháy vào tên mẫu nháy OK, trên màn hình xuất hiện mẫu bằng tiếng Anh.

#### **b. Tạo một Template mới**

Dùng lệnh File/New, xuất hiện cửa sổ New, chọn nút tròn Template, nháy OK. Soạn một mẫu văn bản.

Ghi văn bản vào đĩa: File/Save, vào tên tệp mẫu văn bản, ví dụ MAU1.DOT (th- mục ch- a tệp là C:\...\winwordtemplate\, nháy OK. Đóng cửa sổ vừa tạo Template.

#### **c. Sửa một Template đã có sẵn**

Dùng lệnh File/New, xuất hiện hộp thoại New, chọn nút tròn Template, nháy vào mẫu muốn chọn (chẳng hạn Mau1.dot), nháy OK. Tiến hành sửa Template theo ý muốn. Ghi lại: dùng lệnh File/Save, trong hộp File Name vào tên tệp Mau1.dot hay không, chọn Yes. Đóng cửa sổ văn bản lại.

#### **d. Dùng một Template đã tạo**

Dùng lệnh File/New, xuất hiện hộp thoại New, chọn nút tròn Document, nháy vào tên tệp Mau1.dot trong hộp Template, nháy OK. Điền nội dung thích hợp vào những chỗ trống trong Template. Ghi văn bản vào đĩa: File/Save trong hộp File Name vào tên tệp VBMAU1.dot, nháy OK. Đóng cửa sổ văn bản.

### **3.2.14. Trộn văn bản (Mail Merging)**

Trộn văn bản dùng trong tr- ờng hợp sau. Giả sử cơ quan có một công văn cần gửi đi nhiều nơi, tr- ớc tiên ta soạn mẫu công văn nh- ng để trống các chỗ: ng- ời nhận chức vụ, địa chỉ (tệp này gọi là văn bản chính). Thông tin cần điền vào các chỗ trống lấy từ một tệp dữ liệu đ- ọc liên kết với văn

bản chính. Khi ta trộn tệp dữ liệu và văn bản chính, Word sẽ chèn các thông tin thích hợp vào văn bản chính.

### ***Các bước để trộn văn bản chính và tệp dữ liệu***

Ta hãy nghiên cứu các bước để tạo một loạt các giấy mời họp, danh sách...

- Tạo văn bản chính thí dụ:

Kính gửi (Ông/bà):

Chức vụ:

Ta sẽ l-u danh sách họ tên và chức vụ ở một tệp khác để trộn với văn bản chính.



- Ta chọn **Tools, Mail Merge..., Create, Form Letter..., Active Windows, Get data, Create Data Source...** sau đó bỏ đi những trường không cần, thêm những trường mới rồi l-u lại tệp nguồn này. Khi máy hỏi Edit Data Source thì ta nhập số liệu cho từng bản ghi một.

Word sẽ hiện lại văn bản chính. Lúc này thanh công cụ Mail Merge sẽ xuất hiện ngay dưới thanh công cụ định dạng. Đặt con trỏ vào nơi cần chèn trường rồi nhấn nút **Insert Merge Field**.

Nhấp lên nút View Merge Data để xem nội dung văn bản. Có thể nhấp vào nút **Next Record** để xem bản ghi tiếp theo.

Nhấp lên nút **Print** để in toàn bộ các trường (mỗi record là một trường)

### **Cách gõ tiếng Việt bằng phông ABC:**

Nếu ABC đã được kích hoạt ta sẽ thấy biểu tượng  ở góc trên bên trái màn hình. Ta có thể bật hoặc tắt chế độ gõ tiếng Việt bằng cách nhấp chuột vào nút , hoặc nhấn Alt+Z. Cách gõ một số ký tự tiếng Việt:

Aw->ã, aa->â, dd->đ, ee->ê, oo->ô, ow ([)->ơ, uw(-,])->-

Cách gõ dấu:

s-> sắc, f-> huyền, r-> hỏi, x-> ngã, j-> nặng, z-> xóa dấu.



## CH- ƠNG 4. MICROSOFT EXCEL

### 4.1. Tổng quan về Excel

#### 4.1.1. Chức năng của Excel

Excel là phần mềm chuyên dùng cho công tác kế toán, văn phòng chạy trên môi tr- ờng Windows. Trong ch- ơng tr- ớc chúng ta đã tìm hiểu về Winword. Đó là một phần mềm soạn thảo văn bản cao cấp. Ngoài các văn bản ký tự, ta có thể chèn vào các biểu bảng, một số công thức, hình ảnh. Tuy nhiên khả năng tính toán trong Winword rất hạn chế. Chỉ trong các bảng ta mới có thể tiến hành một vài tính toán đơn giản nh- tính tổng các số trong một cột. Còn ở vùng văn bản ngoài bảng rất khó thiết lập những công thức liên kết các con số trong vùng văn bản. Điều này cũng dễ hiểu vì các vùng văn bản không đ- ợc địa chỉ hóa do đó ta không thể đ- a các con số vào công thức một cách hình thức (nghĩa là không cần ghi rõ giá trị của chúng). Các nhà thiết kế phần mềm đã khắc phục điều này khi xây dựng phần mềm bảng tính Excel. Để tiện cho việc tính toán bằng công thức vùng làm việc trong Excel đ- ợc chia thành các ô có địa chỉ. Vùng làm việc vì vậy đ- ợc gọi là bảng tính điện tử hay bảng công tác (worksheet hay spreadsheet). Bảng công tác giống nh- một trang sổ cái dùng trong công tác kế toán. Bảng gồm 65 536 hàng và 256 cột (nh- vậy bảng tính thực ra là một bảng có diện tích khoảng 46 X 6 m). Các hàng và cột cắt nhau thành các ô, ta có thể gõ dữ liệu vào các ô đó. Ta có thể tiến hành các tính toán từ đơn giản đến phức tạp trên bảng tính để tạo ra các báo cáo tổng hợp hoặc phân tích có kèm theo các biểu đồ, hình vẽ minh họa... Excel còn có chức năng quản trị CSDL nếu số liệu trong bảng đ- ợc sắp xếp có trật tự hơn: dữ liệu trong một cột phải có cùng tính chất.

Excel cho phép ta cùng lúc soạn thảo nhiều bảng trên nhiều cửa sổ và có thể copy dữ liệu từ cửa sổ này sang cửa sổ khác , thậm chí có thể copy hoặc liên kết với dữ liệu của các ứng dụng khác đang mở.

Sau đây chúng tôi giới thiệu cách sử dụng Excel 97 trên nền Windows 98. Cách sử dụng các phiên bản khác của Excel cũng gần t- ơng tự.



#### 4.1.2. Khởi động Excel

Bạn có thể khởi động Excel bằng nhiều cách, tùy thuộc vào cách thiết lập Windows của máy tính bạn đang dùng. Có thể có các cách nh- sau:

- Nhấp đúp chuột lên biểu t- ợng Microsoft Excel trên màn hình nền của Windows (nếu có).
- Nhấp chuột lên biểu t- ợng Microsoft Excel trên thanh menu của Microsoft Office (nếu có).
- Nhấp nút chọn **Start**, chọn **Run**, gõ Excel vào dòng trống sau chữ **Open**: rồi chọn **Ok**

Sau khi khởi động ta thấy trên màn hình là cửa sổ Workbook (Workbook Window) gồm 3 bảng chồng lên nhau với tên gọi là sheet1, sheet2, sheet3 (có thể mở tới 255 sheet). Ta có thể chọn bảng làm việc bằng cách nhấp chuột vào nút chứa tên hoặc nhấn **Ctrl+PageUp** hoặc **Ctrl+PageDown**.

#### 4.1.3. Mở một cửa sổ mới


- Mở một Workbook mới: Chọn **File, New** (Ctrl + N hoặc  ).
- Mở Workbook đã có trên đĩa: Chọn **File, Open** (Ctrl + O hoặc  ).


Tại một thời điểm chỉ có một Workbook đ- ợc kích hoạt, muốn làm việc với một Workbook khác đã đ- ợc mở: gõ **Alt + W**, chọn Workbook cần kích hoạt.

Để tiết kiệm bộ nhớ, nên đóng các cửa sổ không cần thiết (Chọn File, Close hoặc nhấp đúp chuột vào hộp điều khiển trên thanh Menu Bar), chỉ giữ lại các Workbook cần trao đổi dữ liệu với nhau.

#### 4.1.4. In Workbook

Có thể in Workbook theo hai cách:


- Gõ **Ctrl + P** (hoặc nhấp chuột trên biểu tượng ) ta sẽ in được sheet hiện thời.
- Chọn **File, Print...** ta có thể chọn để in toàn workbook hoặc active sheet.

Chú ý, trước khi in có thể xem trước kết quả in trên màn hình bằng cách thực hiện lệnh **File, Print Preview** (hoặc nhấp chuột trên biểu tượng ).

#### 4.1.5. Lưu cất bảng tính Excel

Nếu ta lưu 1 Workbook thì tệp sẽ có đuôi \*.XLS

và ta có thể lưu cất như sau:

Nếu tệp Excel đã có tên và ta không muốn thay đổi ta có thể lưu văn bản đang soạn thảo bằng cách nhấp chuột vào **File, Save** hoặc nhấp vào biểu tượng  trên thanh công cụ. Nếu ta muốn đặt tên mới cho tệp văn bản ta nhấn **File, Save as...** rồi nhập tên mới vào.

Nếu ta lưu tất cả các Workbook đang mở thì tệp sẽ có đuôi là \*.XLW (Workspace) và làm như sau:

Chọn **File, Save Workspace...**

#### 4.1.6. Thoát khỏi Excel

Bạn có thể thoát khỏi Excel bằng nhiều cách:

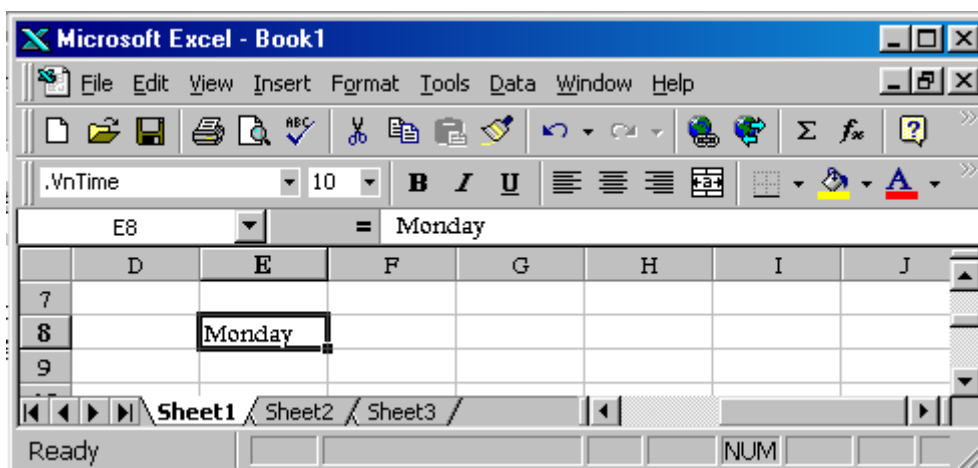
- Nhấp chuột lên mục **File** của thanh menu rồi chọn **Exit**.
- Nhấp chuột lên dấu **Close (X)** ở góc phải phía trên cửa sổ Excel.
- Nhấn tổ hợp phím **Alt+F4**

Nếu Excel hỏi bạn có muốn lưu cất tài liệu không, bạn sẽ chọn Yes hoặc No tùy thuộc vào ý của bạn.

## 4.2. Làm việc với Excel

### 4.2.1. Màn hình làm việc của Excel

Màn hình làm việc của Excel có thể có dạng sau:



Màn hình của Excel ở các máy có thể khác nhau chủ yếu là do ta cho hiện trên màn hình những thanh công cụ nào, đặt chúng ở đâu. Ta có thể chọn hoặc không chọn các thanh công cụ bằng cách nhấn **View**, **Toolbars**, rồi đánh dấu hoặc thôi đánh dấu các thanh công cụ (Toolbars), chọn hoặc không chọn thước (Ruler). Ta cũng có thể nhấn chuột để kéo các thanh công cụ đến các vị trí khác nhau trên màn hình. Màn hình Excel thường có các phần sau đây theo thứ tự từ trên xuống:

- **Title bar** (Thanh tiêu đề): dòng chứa tên của tệp văn bản.

Góc trái thanh tiêu đề ta thấy biểu tượng của Excel, nhấp vào đó sẽ kích hoạt hộp điều khiển. Góc phải là các nút Minimize, Restore hoặc Maximize và nút Close của cửa sổ Excel.

- **Menu bar** (thực đơn hàng ngang): dòng chứa các lệnh của Excel, mỗi lệnh ứng với một thực đơn dọc (Menu PopUp).

Góc trái thanh **Menu Bar** ta thấy biểu tượng của cửa sổ tệp, nhấp vào đó sẽ kích hoạt hộp điều khiển. Góc phải là các nút Minimize, Restore hoặc Maximize và nút Close của cửa sổ tệp.

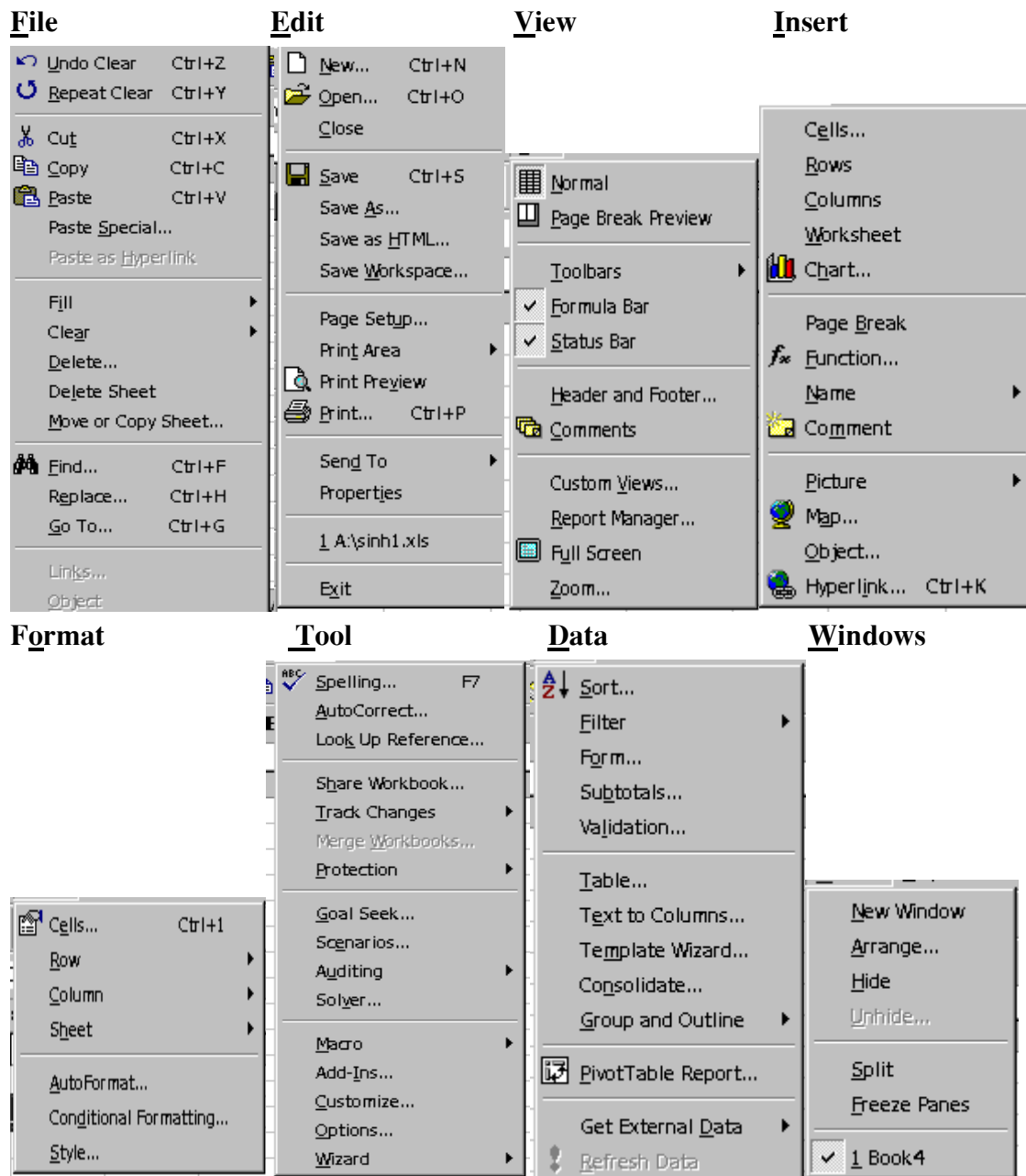
**Thao tác để mở một thực đơn dọc :**

- Chỉ con trỏ chuột vào tên thực đơn dọc trên Menu Bar rồi bấm nút trái chuột.
- Gõ phím **F10**, hoặc **Alt** rồi dùng phím mũi tên di chuyển vết sáng đến thực đơn dọc cần mở nhấn {ENTER}.

**Trong các thực đơn dọc :**

- Các lệnh có màu xám nhạt là lệnh tạm thời không thực hiện được,
- Các ký tự được gạch dưới thì chọn lệnh tương ứng bằng cách bấm ký tự đó (thay vì phải di vết sáng đến lệnh và bấm ENTER).
- Các lệnh có ghi kèm tổ hợp phím thì có thể chọn nhanh bằng cách bấm tổ hợp phím đó (Ví dụ: có thể lưu văn bản lên đĩa bằng cách bấm **Ctrl+ S**).
- Các lệnh có dấu ba chấm (...) phía sau cho biết sẽ có một hộp thoại xuất hiện khi lệnh được chọn.

Sau đây là các thực đơn dọc của thanh menu.



- **Tools bar** (thanh công cụ): chứa một số biểu tượng (Icon) thể hiện một số lệnh thông dụng. Thay vì phải vào các hộp menu để chọn lệnh, chỉ cần nhấp chuột lên biểu tượng của lệnh tương ứng.

Ví dụ: Muốn lưu Workbook lên đĩa thay vì vào Menu File chọn Save, chỉ cần nhấp chuột lên biểu tượng



- **Formating Bar** (Thanh định dạng): chứa các hộp và biểu tượng dành cho việc định dạng văn bản như kiểu, loại font, cỡ font, căn lề...

- **Thanh công thức (Formula Bar)**

Thanh công thức thường bố trí dưới thanh định dạng. Đây là thanh chủ yếu để nhập dữ liệu và có đặt một số công cụ liên quan đến việc nhập, sửa dữ liệu.

- **Phần bảng tính (Workbook Window)**

Phần nằm giữa cửa sổ Excel là phần bảng tính. Trong bảng tính một số thành phần được định nghĩa như sau:

#### **Tiêu đề cột (Column Headings)**

Dòng đầu tiên của bảng tính gồm 256 ô được kí hiệu (chuẩn) bằng các chữ cái hoa A, B, ..., AA, AB, ..., IV gọi là các tiêu đề cột (Column Headings). Tiêu đề cột dùng để kí hiệu tên cột, địa chỉ cột và đảm nhiệm một số chức năng khác.

#### **Tiêu đề dòng (Row Headings)**

Cột đầu tiên (bên trái) của bảng tính gồm 65 536 hàng tiêu đề, được kí hiệu bằng các số 1, 2, 3 ..., 65 536 gọi là các tiêu đề dòng. Tiêu đề dòng dùng để kí hiệu tên dòng, địa chỉ dòng và đảm nhiệm một số chức năng khác.

#### **Một số định nghĩa khi làm việc với các ô**

- **Ô (Cell).** Giao của một dòng và một cột gọi là ô, nhằm xác định một vị trí trong bảng tính. Thí dụ: ô A1: (cột A, dòng 1), tên cột viết trước, tên dòng viết sau. Ô được ghi như trên gọi là địa chỉ (Reference).

- **Miền (Range).** Giao của một số dòng và một số cột liên tiếp tạo thành một miền. Như vậy, một miền là vùng chữ nhật của bảng tính, gồm một số ô liên tiếp với nhau. Thí dụ miền A1:B3 là một vùng chữ nhật gồm các ô A1, A2, A3, B1, B2, B3.

- **Ô hoạt động (Active Cell).** Ô hoạt động còn gọi là ô hiện hành, là ô có con trỏ ô định vị. Dấu hiệu ở ô hoạt động thể hiện bằng một khung viền đen, đậm ở góc dưới bên phải có một nút vuông đen nhỏ gọi là nút điền (fill handle). Nút điền có nhiều tác dụng, sẽ được trình bày trong các thao tác có liên quan.

- **Scroll bars** (Các thanh cuộn): Có 2 thanh cuộn ở dưới và bên phải màn hình dùng để cuộn nhanh văn bản lên xuống hoặc sang trái, sang phải.

- **Status bar** (Thanh trạng thái): hiển thị chế độ làm việc hiện hành, tình trạng của Capslock, Numlock... Thanh này nằm ở dòng cuối cùng màn hình.

Tại bất cứ vị trí nào trên màn hình ta cũng có thể kích hoạt **Shortcut Menu** bằng cách nhấp nút phải của chuột. Shortcut Menu sẽ có nội dung tương ứng với vị trí con trỏ.

### **4.2.2. Nhập và sửa số liệu trong bảng tính**

#### **a. Di chuyển con trỏ giữa các ô**

Có nhiều cách để di chuyển con trỏ khắp bảng tính kể từ ô hiện hành:

##### **-Dùng chuột**

Nhấp chuột vào ô cần tới. Có thể dùng thanh cuộn ngang hoặc dọc để lướt nhanh về các ô cần chuyển đến nếu các ô đó đang bị khuất.

##### **-Dùng bàn phím**

- Dùng một trong các phím mũi tên lên, xuống, phải, trái để di chuyển con trỏ ô lên, xuống, sang phải, sang trái đối với ô hiện hành.

- **PgDown, PgUp** để xuống hoặc lên một màn hình bảng tính.

- **Ctrl + ->** để về cột cuối cùng (cột IV).

- **Ctrl + ↓** về dòng cuối cùng (dòng thứ 65 536).

- **Ctrl + Home** về ô đầu tiên (A1).

- **Nhập một ô vào hộp tên** (Name Box) ở trên thanh công thức để chuyển nhanh con trỏ tới ô đó.

- **Dùng lệnh Edit, Go to** sau đó đánh địa chỉ ô, miền ô hoặc tên ô, tên miền cần chuyển tới.

### **b. Chọn ô, miền**

Tr- ớc khi thực hiện một thao tác hoặc một lệnh đối với ô, miền ô, thông th- ờng phải chọn ô. Các thao tác chọn nh- sau:

- Chọn một ô: di chuyển con trỏ về ô đó.
- Chọn nhiều ô không liên kế: chọn ô đầu tiên nh- cách chọn một ô thông th- ờng. Kể từ ô thứ 2 trở đi, ấn và giữ phím Ctrl và nhấp chuột vào ô cần chọn.
- Chọn một miền ô: kéo chuột từ ô đầu ở một trong các góc chữ nhật của miền ô đến ô đối diện.
- Chọn nhiều miền ô: chọn miền đầu tiên theo cách thông th- ờng. Kể từ miền ô thứ 2 trở đi, ấn và giữ phím Ctrl rồi chọn miền đó.
- Chọn toàn dòng hoặc toàn cột: nhấp chuột vào tiêu đề dòng, hoặc tiêu đề cột cần chọn (dòng đ- ọc chọn chứa 256 ô chọn, cột đ- ọc chọn chứa 65 536 ô đ- ọc chọn).
- Chọn một miền dòng hoặc một miền cột: rê chuột xuyên suốt các tiêu đề dòng hoặc tiêu đề cột cần chọn.
- Chọn nhiều miền dòng hoặc nhiều miền cột: chọn miền dòng hoặc miền cột đầu tiên theo cách đã nêu. Kể từ miền thứ 2 trở đi cũng làm t- ơng tự nh- ng nhấn thêm phím Ctrl trong khi chọn.
- Chọn toàn bảng tính: nhấp vào ô đầu tiên của bảng tính tức là ô giao giữa các tiêu đề dòng và cột.

Khi ô hoặc miền ô đ- ọc chọn, những ô hoặc miền ô đó đ- ọc bôi đen. Muốn huỷ bỏ việc lựa chọn ô hoặc miền ô (bỏ bôi đen), hãy nhấp chuột vào vùng ch- a chọn.

### **c. Nhập số liệu vào các ô**

Khi mới khởi động Excel, một bảng tính mới, rỗng hiện ra với tên Book1 ở tiêu đề bảng tính. Việc nhập số liệu kiểu ký tự và kiểu số vào bảng tính có một số điểm khác nhau cần chú ý.

#### **- Nhập dữ liệu kiểu ký tự**

Chọn ô, sau đó đánh ký tự theo quy tắc soạn thảo chung

Ngầm định là đánh các chữ từ trái sang phải (căn thẳng lề trái)

Một ô có thể chứa tối đa 255 ký tự.

Nhìn trên màn hình có thể thấy chữ đánh vào ô đó tràn sang ô bên phải, nh- ng nội dung vẫn vẫn đ- ọc giữ tại ô đã chọn ban đầu.

#### **Tạo những chuỗi kí tự có chu kỳ**

Excel có khả năng tạo chuỗi kí tự có tính chu kỳ nh- tên tháng, tên ngày trong tuần, tên quý v.v... Để giúp ng- ời sử dụng tạo nhanh chuỗi chữ có tính chất lặp có chu kỳ, trong các thí dụ sau đây trình bày cụ thể cách thực hiện:

**Thí dụ 1:** Tại ô bất kỳ, gõ Monday (hoặc Mon) và thực hiện tiếp các thao tác:

- Đ- a trỏ chuột vào nút điền ở góc d- ới, bên phải ô hiện hành; con trỏ chuột lúc đó đổi dạng thành dấu thập mảnh.
- Giữ và kéo chuột sang phải, hoặc trái, hoặc lên hoặc xuống d- ới so với ô hiện hành, sẽ đ- ọc một chuỗi chữ kế tiếp, lặp lại có chu kỳ.

Có thể dùng hình thức này để tạo một chuỗi chữ kế tiếp nhau không theo một số từ chuẩn nh- trên của Excel.

**Thí dụ 2:** Tạo một hàng chữ tháng 1 ... tháng 12 liên tục tại các ô liền kề theo dòng hoặc cột:

- Đánh chữ Tháng 1 vào ô bất kỳ,
- Đ- a trỏ chuột vào hộp điền, con trỏ đổi thành dấu thập mảnh,
- Kéo thêm 11 ô nữa sang phải để tạo chuỗi theo dòng hoặc xuống d- ới để tạo chuỗi theo cột, khi đó sẽ có nhanh đ- ọc hàng chữ mong muốn nh- trên.

#### - Nhập dữ liệu kiểu số

Nhập số vào ô có một vài đặc thù:

- Mặc định là căn thẳng lề phải,
- số liệu ở ô nào, nằm gọn ở ô đó (không tràn qua ô khác nh- nhập chữ). Nếu độ rộng của cột không đủ vị trí để chứa số đánh vào, dữ liệu sẽ thể hiện ở dạng khoa học (thí dụ 7.89E + 02) hoặc dấu số #####. Cần mở rộng cột chứa số đó để hiển thị đầy đủ các số trong ô.





#### **Nhập những số có tính chu kỳ**


- Đánh 2 số đầu của chuỗi vào 2 ô liên tiếp kề nhau.
- Chọn (bôi đen) 2 ô đó.
- Dùng hộp điền kéo chuột sang trái, hoặc phải, hoặc lên, hoặc xuống để tạo chuỗi theo h- ướng t- rong ứng.

Có thể tạo nhiều chuỗi số có b- ớc nhảy (giá trị cách) khác nhau nh- cách trên.

#### **d. Một số quy tắc khi nhập số liệu.**

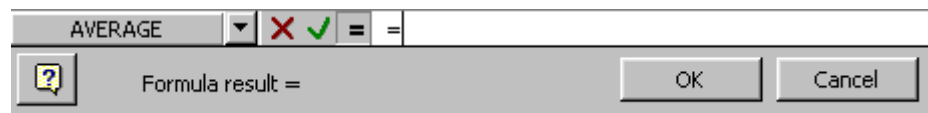
##### Nhập số hoặc ký tự:

Khi ta chuyển con trỏ đến một ô nào đó nh- ng ch- a nhập số liệu thì thanh công thức có dạng . Nếu ta bắt đầu gõ một ký tự nào đó (thí dụ: Cộng hòa) không phải là dấu = lúc đó thanh công thức sẽ có dạng   = Cộng hòa. Nhập xong cần nhấn Enter hoặc nhấp và hộp nhập  để ghi số liệu vào ô (nếu không sẽ không thực hiện đ- ọc các thao tác tiếp theo).


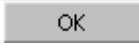
Muốn bỏ số liệu đã đánh xong, hãy ấn Esc hoặc nhấp chuột ở hộp huỷ .


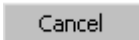
##### Nhập hàm hoặc công thức:

Nếu ta nhấp chuột vào dấu = trên thanh công thức thì thanh công thức sẽ chuyển sang dạng sau:



Ta có thể nhấp vào nút kéo xuống để chọn hàm hoặc gõ vào công thức. Công thức có thể là các hàm đ- ọc định nghĩa sẵn nh- AVERAGE(),SUM(),... hoặc do ta gõ vào, thí dụ =A1+B1\*C1.

Nhập xong cần nhấn Enter hoặc nhấp và hộp nhập  hoặc nhấp vào nút  để ghi số liệu vào ô (nếu không sẽ không thực hiện đ- ọc các thao tác tiếp theo).

Muốn bỏ số liệu đã đánh xong, hãy ấn Esc hoặc nhấp chuột ở hộp huỷ  hoặc nhấp vào nút .

### **Sửa số liệu :**

- Đặt trỏ chuột tại ô cần sửa.
- Nhấn **F2**, sau đó sử dụng các thao tác soạn thảo nh- xoá, gõ, đè, chèn v.v... để sửa dữ liệu.

### **Xoá dữ liệu:**

- Chọn ô, hoặc miền cần xoá sau đó.
- Nhấn phím **Del** hoặc chọn lệnh **Edit, Clear**, hoặc từ Menu ngắn (Short Cut Menu) chọn lệnh Clear Contents (xoá nội dung).

### ***Tìm kiếm chuỗi ký tự***

Nhấn **Ctrl+F**

### ***e. Copy, xoá hoặc dịch chuyển miền đã đánh dấu***

Các thao tác giống nh- với Winword. Ngoài ra có thêm một số thao tác sau:

**Dịch chuyển miền:** Đ- a con trỏ chuột đến mép vùng đã đánh dấu cho đến khi chuột có dạng mũi tên. Nhấn chuột trái và chuyển đến vị trí mới rồi nhả chuột. Còn nếu ta muốn copy thì ta nhấn phím Ctrl trong khi rê chuột.

### ***Địa chỉ t- ơng đối, địa chỉ tuyệt đối***

Giả sử công thức tại ô (i,j) có chứa một số địa chỉ tham chiếu, thì khi copy sang ô i+K,j+L ta sẽ thấy các địa chỉ tham chiếu sẽ đ- ọc tăng lên K dòng và L cột. Thí dụ công thức ở ô C1 là A1\*B1, chứa các địa chỉ tham chiếu là A1 và B1. Nếu ta copy nội dung ô C1 vào ô C2 thì ta có công thức tại ô này là A2\*B2. Địa chỉ A1, B1 trong công thức trên đ- ọc gọi là địa chỉ t- ơng đối. Nếu ta muốn cố định dòng ta phải thêm dấu \$ tr- ớc địa chỉ dòng, còn nếu ta muốn cố định địa chỉ cột thì ta phải thêm dấu \$ tr- ớc địa chỉ cột.

Thí dụ: địa chỉ tuyệt đối dòng A\$2  
địa chỉ tuyệt đối cột \$A2  
địa chỉ tuyệt đối \$A\$2

Khi ta copy công thức địa chỉ tuyệt đối không bị thay đổi.

### ***f. Chèn các ô, dòng, cột hoặc ngắt trang***

Chèn ô, dòng hoặc cột: Nhấn **Insert, Cells**



Chèn hoặc (thời chèn) dấu phân trang:

- Đặt con trỏ tại ô nằm d- ới và/hoặc bên phải các dấu phân trang
- Nhấn **Insert, Page Break** (Remove Page Break).

**g. Thay đổi độ rộng cột, chiều cao dòng**

**Format, Column, Width**

**Format, Row, Height**

**h. Định dạng dữ liệu số hoặc ký tự**

**Format, Cell, Number**

**Format, Cell, Font**

**i. Canh trái, phải, giữa trong từng ô**

Đánh dấu miền, chọn **Format, Cell, Alignment** hoặc nhấn vào biểu t- ợng trên thanh định dạng.

**j. Tạo các đ- ờng kẻ trong vùng đã chọn**

Ngâm định Excel in tất cả các đ- ờng phân cách giữa các ô có dữ liệu. Nếu ta không muốn in thì chọn

**File, Page Setup...,Sheet** rồi bỏ dấu chọn ở ô Gridlines.

Nếu ta muốn kẻ các đ- ờng cho miền đã chọn nhấn **Format, Cells..., Border**

Chọn màu nền dữ liệu: **Format, Cells..., Patterns**

**k. Sắp xếp dữ liệu**

Thực hiện lệnh **Data, Sort...**


Chú ý: Muốn sắp xếp bảng theo cột nào thì ta đặt con trỏ vào cột đó và không bôi đen cột hoặc một phần của cột. Vì nếu ta chỉ bôi đen một phần bảng thì chỉ phần đó đ- ợc sắp xếp làm cho số liệu bị lệch lạc, “râu ông nọ cắm cằm bà kia”.

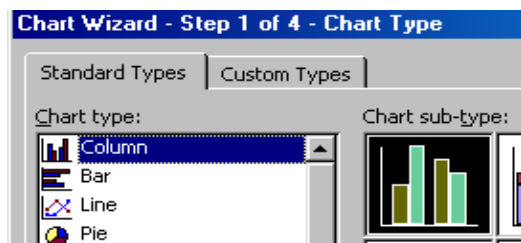
**4.2.3. Biểu đồ bảng tính**

Giả sử ta có bảng số liệu sau đây:

**BẢNG THỐNG KÊ GIÁ TRỊ XUẤT KHẨU NĂM 1995**

LOẠI	QUÍ 1	QUÍ 2	QUÍ 3	QUÍ 4
Lâm sản	180	252	155	195
Nông sản	165	225	165	122
Hải sản	210	241	185	164

Ta muốn lập biểu đồ để so sánh kết quả sản xuất giữa các loại trong từng quý và của từng loại qua 4 quý. Trước hết ta đưa con trỏ chuột vào bảng số liệu trên đây, sau đó chọn **Insert**,  Chart... bảng Chart Wizard sẽ hiện ra và ta có thể thực hiện các thao tác theo hướng dẫn của Excel



#### 4.2.5. Cơ sở dữ liệu trên bảng tính

##### a. Khái niệm cơ sở dữ liệu

Khái niệm CSDL: CSDL là một bộ s-u tập các thông tin có liên quan với nhau về một chủ đề, được tổ chức một cách tiện dụng. Bộ s-u tập này tạo nên một cơ sở cho các thủ tục như tìm kiếm thông tin, rút ra các kết luận và đưa ra các quyết định. Trong Excel CSDL được định nghĩa một cách cụ thể hơn như sau:

**Vùng CSDL (Database)** là vùng dữ liệu có ít nhất 2 dòng. Dòng đầu tiên là các tiêu đề cột, gọi là tên trường (Field Name). Tên trường phải là dữ liệu kiểu chuỗi và không được trùng lặp. Các dòng còn lại chứa dữ liệu. Dữ liệu trong cùng một cột phải có cùng tính chất.

Excel còn định nghĩa các vùng sau đây liên quan đến CSDL:

**Vùng tiêu chuẩn (Criteria)** là vùng chứa điều kiện để tìm kiếm, xóa, rút trích. Vùng này cũng có ít nhất 2 dòng. Dòng đầu là tiêu đề, các dòng còn lại chứa điều kiện.

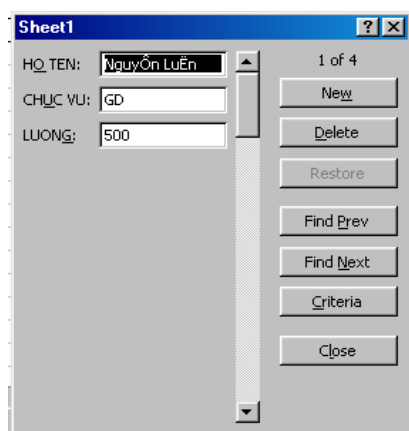
**Vùng rút trích (Extract)** là vùng chứa các mẫu tin thỏa mãn tiêu chuẩn. Vùng này cũng có dòng đầu tiên là tiêu đề. Vùng này chỉ dùng đến trong thao tác rút trích, còn các thao tác khác như tìm kiếm, xóa... không cần đến.

Các bạn có thể thấy các vùng trên qua bảng sau đây:

HO TÊN	CHỨC VỤ	LUONG		CHỨC VỤ
Hồ Văn Dũng	TK	300		TK
Phùng Văn Giá	NV	200		GD
Lê Thị Hoa	NV	220		
Nguyễn Luận	GD	500		
HO TÊN	CHỨC VỤ	LUONG		
Hồ Văn Dũng	TK	300		
Nguyễn Luận	GD	500		

##### b. Sử dụng hộp thoại Data Form

Đặt con trỏ vào vùng CSDL rồi thực hiện lệnh **Data, Form** cửa sổ sau hiện ra

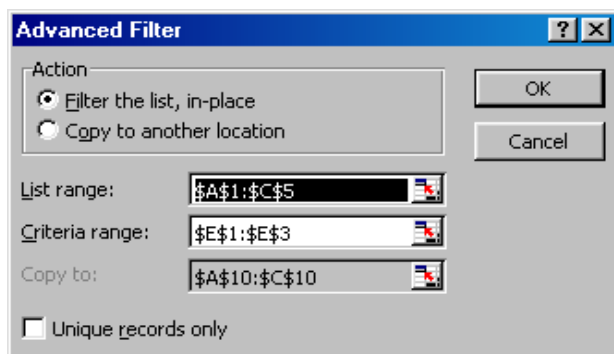


Ta có thể nhập mới, tìm kiếm, sửa đổi các bản ghi.

### c. Tìm kiếm, rút trích và xóa

Trước hết ta tạo vùng Criteria và đưa con trỏ về ô nào đó trong CSDL

Thực hiện lệnh **Data, Filter, Advance Filter...**



Ngầm định Excel sẽ chỉ cho hiện những bản ghi thỏa mãn điều kiện tiêu chuẩn, do đó nếu ta muốn xóa thì dùng lệnh Edit, Delete Row để xóa. Nếu ta muốn giữ nguyên CSDL copy các bản ghi thỏa điều kiện lọc ra một vùng khác ta chọn *Copy to another location*.

Vùng tiêu chuẩn có thể là hợp của nhiều điều kiện, thí dụ

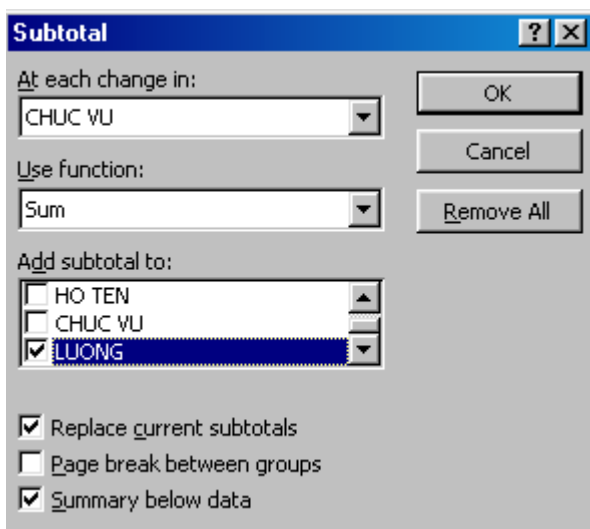
HO TEN	CHUC VU	LUONG		CHUC VU	LUONG
Nguyễn Luận	GD	500		NV	>200
Phùng Văn Giá	NV	200		GD	500
Lê Thị Hoa	NV	220			
Hồ Văn Dũng	TK	300			
HO TEN	CHUC VU	LUONG			
Nguyễn Luận	GD	500			
Lê Thị Hoa	NV	220			

Điều kiện là những NV l-ong hơn 200 hoặc GD có l-ong bằng 500.

### d. Tổng hợp số liệu theo nhóm

Giả sử ta muốn tổng hợp số liệu theo chức vụ. Ta đưa con trỏ chuột vào cột CHUC VU, sau đó thực hiện các thao tác sau

- Tr- ớc hết ta sắp xếp CSDL theo chức vụ bằng lệnh **Data, Sort...**
- Đặt con trỏ vào vùng CSDL thực hiện lệnh **Data, Subtotals...** cửa sổ sau hiện ra



Thí dụ ta chấp nhận các tham số trên đây, ta sẽ nhận đ- ợc bảng thống kê sau:

STT	HO TEN	CHUC VU	LUONG
1	Nguyễn Luận	GD	500
		<b>GD Total</b>	500
2	Phùng Văn Giá	NV	200
3	Lê Thị Hoa	NV	220
5	Phan Bá	NV	250
		<b>NV Total</b>	670
4	Hồ Văn Dũng	TK	300
6	Trịnh Quốc Anh	TK	290
		<b>TK Total</b>	590
		<b>Grand Tot</b>	1760

#### 4.2.5. Hàm trong Excel.

**Hàm là một ch- ơng trình con cho một kết quả xác định.**

Cách viết một hàm có thể thấy qua thí dụ của hàm tổng (SUM) sau đây:

**=SUM(A1:A4; B1:B4)**

- Bắt đầu của hàm là dấu bằng.
- Tiếp theo là tên hàm.
- Sau đó là mở ngoặc đơn, cuối cùng là đóng ngoặc đơn.
- Giữa các dấu ngoặc đơn là đối số.

Đối số có thể là không phải số, hoặc nhiều miền cách nhau bằng dấu phẩy (hoặc chấm phẩy). Hàm có thể có đối số hoặc không có đối số. Hàm không có đối số nh- hàm =PI(), hàm =NOW(). Có thể tạo các hàm lồng nhau.

#### Cách gọi một hàm

Dùng một trong các cách sau đây để đ- a một hàm vào xử lý:

- Đánh tên hàm ở thanh công thức;

- Nhấp hộp hàm trên thanh công cụ (fx)
- Từ menu Insert, chọn lệnh Functions.

Khi đi- ợc gọi, Excel sẽ hiện hộp đối thoại trong đó tên các hàm đi- ợc phân loại thành từng nhóm th- ờng dùng nh- trong bảng nhóm hàm (Function Category).

- Hàm Toán học và L- ợng giác (Math & Trig);
- Hàm thống kê (Statistical);
- Hàm tài chính (Financial);
- Hàm ngày và Thời gian (date and time);
- Hàm dùng với cơ sở dữ liệu (Database) v.v...

**Một số hàm thông dụng:**

**SUM**(danh sách trị): tính tổng các giá trị có trong danh sách. Thí dụ: =SUM(B1:B3,B6)

**AVERAGE**(danh sách trị): tính trung bình các giá trị có trong danh sách. Thí dụ: =AVERAGE(B1:B3,B6)

**ROUND**(biểu thức số,n): làm tròn đến n số lẻ. Thí dụ =ROUND(12.12345,3)=12.123

**MAX**(danh sách trị): tính giá trị lớn nhất trong các giá trị có trong danh sách. Thí dụ: =MAX(B1:B3,B6)

**MIN**(danh sách trị): tính giá trị bé nhất các giá trị có trong danh sách. Thí dụ: =MIN(B1:B3,B6)

**IF**(<biểu thức logic>, <trị 1>,<trị 2> ) nếu <biểu thức logic> đúng nhận <trị 1> sai nhận <trị 2>

**AND**(<điều kiện 1>,<điều kiện 2>,...,<điều kiện n>) cho giá trị đúng khi mọi điều kiện đều đúng.

**OR**(<điều kiện 1>,<điều kiện 2>,...,<điều kiện n>) cho giá trị đúng khi có ít nhất 1 điều kiện đúng.

**NOT**(<điều kiện>) cho giá trị đúng nếu điều kiện sai.

**ABS**(<biểu thức số>)

**SQRT**(<biểu thức số>) với điều kiện <biểu thức số> không âm

**INT**(<biểu thức số>) cho phần nguyên của <biểu thức số> . Thí dụ: =INT(12.7)=12, =INT(-3.2)=-4

**PI**() Không có đối số, là số PI

**MOD**(M,N) phần d- của phép chia M cho N . Thí dụ =MOD(10,3)=1

**COUNT**(danh sách trị): cho số các giá trị số có trong danh sách. Thí dụ =COUNT(-1,'AA',2)=2

**COUNTA**(danh sách trị): cho số các trị có trong danh sách. Thí dụ =COUNTA(-1,'AA',2)=3

**RANK**(X,danh sách)=n nghĩa là nếu ta xếp danh sách theo thứ tự giảm dần thì X đứng thứ n.

**LEFT**(<chuỗi>,N)

**RIGHT**(<chuỗi>,N)

**LEN**(<chuỗi>)

**LOWER**(<chuỗi>)

**UPPER**(<chuỗi>)

**PROPER**(<chuỗi>)

**DAY**(<biểu thức ngày>)

**DAY**(<biểu thức ngày>)

**MONTH**(<biểu thức ngày>)

**YEAR**(<biểu thức ngày>)

**Các hàm tìm kiếm:**

**VLOOK**(x,bảng,cột tham chiếu,cách dò)

**HLOOK**(x,bảng,cột tham chiếu,cách dò)

Cách dò=0 cột bên trái không phải sắp xếp, cách dò =1 cột bên trái phải sắp theo thứ tự tăng dần

**Các hàm trong CSDL:**

**DSUM**(Vùng CSDL, N, Vùng tiêu chuẩn) Tổng các số trên cột thứ N của Vùng CSDL của những mẫu tin thỏa mãn điều kiện của vùng tiêu chuẩn.

**DMAX**(Vùng CSDL, N, Vùng tiêu chuẩn)

**DMIN**(Vùng CSDL, N, Vùng tiêu chuẩn)

**DMAX**(Vùng CSDL, N, Vùng tiêu chuẩn)

**DCOUNT**(Vùng CSDL, N, Vùng tiêu chuẩn)

**DCOUNTA**(Vùng CSDL, N, Vùng tiêu chuẩn)

**DAVERAGE**(Vùng CSDL, N, Vùng tiêu chuẩn)

## PHẦN V. NGÔN NGỮ LẬP TRÌNH C

### CH- ONG 1. TỔNG QUAN VỀ NGÔN NGỮ C

#### 1.1. Ngôn ngữ và thuật toán

##### 1.1.1. Ngôn ngữ lập trình

Trong các phần tr- ớc chúng ta đã tìm hiểu Winword và Excel, là các phần mềm ứng dụng trong công việc soạn thảo văn bản và làm các bảng kế toán. Đặc điểm của các phần mềm ứng dụng là luôn định rõ phạm vi ứng dụng và cung cấp càng nhiều càng tốt các công cụ để hoàn thành chức năng đó. Tuy nhiên ng- ời sử dụng cũng hầu nh- bị bó buộc trong phạm vi quy định của ch- ơng trình. Chẳng hạn ta khó có thể dùng Excel để giải một bài toán gồm nhiều b- ớc tính toán nh- tính nghiệm gần đúng một ph- ơng trình vi phân hay giải một hệ ph- ơng trình tuyến tính. Mặc dầu các phần mềm ứng dụng ngày càng nhiều và thuộc đủ các lĩnh vực nh- xây dựng, thiết kế, hội họa, âm nhạc...nh- ng không thể bao trùm hết các vấn đề nảy sinh trong thực tế vô cùng phong phú. Rõ ràng không chỉ những chuyên gia tin học mà ngay cả những ng- ời sử dụng, nhất là các cán bộ kỹ thuật, rất cần đến những phần mềm uyển chuyển và mềm dẻo hơn, có khả năng thực hiện đ- ọc nhiều hơn các chỉ thị của ng- ời sử dụng để giúp họ giải quyết những công việc đa dạng bằng máy tính. Phần mềm có tính chất nh- thế đ- ọc gọi là ngôn ngữ lập trình. Chính xác hơn ***ngôn ngữ lập trình là một ngôn ngữ nhân tạo bao gồm một tập các từ vựng*** (mà ta sẽ gọi là từ khóa để phân biệt với ngôn ngữ thông th- ờng) ***và một tập các quy tắc*** (gọi là Syntax - cú pháp) ***mà ta có thể sử dụng để biên soạn các lệnh cho máy tính thực hiện.***

Nh- ta đã biết các ô nhớ của máy tính chỉ có thể biểu diễn các số 0 và 1. Vì vậy ngôn ngữ mà máy có thể hiểu trực tiếp là ngôn ngữ trong đó các lệnh là các dãy số nhị phân và do đó đ- ọc gọi là ngôn ngữ máy (machine language) . Mọi ngôn ngữ khác đều phải thông dịch hoặc biên dịch sang ngôn ngữ máy (Interpreter - thông dịch và cho chạy từng lệnh. Compiler - biên dịch thành 1 ch- ơng trình ngôn ngữ máy hoàn chỉnh, do vậy chạy nhanh hơn thông dịch).

Có nhiều loại ngôn ngữ lập trình, và hầu hết các nhà khoa học về máy tính đều cho rằng không có một ngôn ngữ độc nhất nào có đủ khả năng phục vụ cho các yêu cầu của tất cả các lập trình viên. Theo truyền thống, các ngôn ngữ lập trình đ- ọc phân làm 2 loại: các ngôn ngữ bậc thấp và ngôn ngữ bậc cao.

##### Ngôn ngữ lập trình bậc thấp (low-level programming language):

Ngôn ngữ máy, hợp ngữ (assembler: ch- ơng trình dịch hợp ngữ, assembly language: ngôn ngữ hợp ngữ). Hợp ngữ là ngôn ngữ một bậc từ ngôn ngữ máy. Nó chỉ khác với ngôn ngữ máy trong việc sử dụng các mã biểu thị các chức năng chính mà máy thực hiện.

Lập trình bằng hợp ngữ rất phiền toái: có đến vài tá dòng mã cần thiết chỉ để thực hiện phép cộng 2 con số. Các ch- ơng trình hợp ngữ rất khó viết; chúng không có cấu trúc hoặc modun hóa rõ ràng. Ch- ơng trình hợp ngữ cũng không dễ chuyển từ loại máy tính này sang loại máy tính khác. Các ch- ơng trình này đ- ọc viết theo các tập lệnh đặc thù của loại bộ vi xử lý nhất định. Lập trình bằng hợp ngữ thì mã gọn và chạy nhanh. Do đó hầu hết các ch- ơng trình điều hành hệ thống đều đ- ọc viết bằng hợp ngữ. Tuy nhiên do sự phức tạp của công việc lập trình nên các hãng sản xuất phần mềm chuyên dụng thích viết ch- ơng trình bằng ngôn ngữ C (do Bell Laboratories của hãng AT&T xây dựng) là loại ngôn ngữ kết hợp đ- ọc cấu trúc của ngôn ngữ bậc cao hiện đại với tốc độ và tính hiệu quả của hợp ngữ bằng cách cho phép nhúng các lệnh hợp ngữ vào ch- ơng trình.

##### Ngôn ngữ lập trình bậc cao:

Các ngôn ngữ lập trình bậc cao nh- Basic,C,C... cho phép các lập trình viên có thể diễn đạt ch- ơng trình bằng các từ khóa và các câu lệnh gần giống với ngôn ngữ tự nhiên. Các ngôn ngữ này đ- ọc

gọi là “bậc cao” vì chúng giải phóng các lập trình viên khỏi những quan tâm về từng lệnh sẽ đ-ợc máy tính tiến hành nh- thế nào, bộ phận thông dịch hoặc biên dịch của ch-ơng trình sẽ giải quyết các chi tiết này khi mã nguồn đ-ợc biến đổi thành ngôn ngữ máy. Một câu lệnh trong ngôn ngữ bậc cao t-ơng ứng với một số lệnh ngôn ngữ máy, cho nên bạn có thể thảo ch-ơng theo ngôn ngữ bậc cao nhanh hơn so với bậc thấp. Tuy nhiên bạn cũng phải trả giá cho việc này. Ch-ơng trình ngôn ngữ máy đ-ợc dịch ra từ mã nguồn đ-ợc viết bằng ngôn ngữ bậc cao chứa rất nhiều chi tiết thừa, do đó tốc độ chạy sẽ chậm hơn nhiều so với ch-ơng trình viết bằng hợp ngữ. Thông th-ờng một trình biên dịch đặc tr-ng th-ờng sinh ra số lệnh mã máy gấp 2 lần hay nhiều hơn số lệnh cần thiết nếu viết bằng mã máy.

#### Một cách phân loại khác của các ngôn ngữ lập trình:

Ngôn ngữ thủ tục (Procedural Language) và ngôn ngữ khai báo (Declarative Language)

Ngôn ngữ thủ tục: Lập trình viên phải xác định một thủ tục mà máy tính sẽ tuân theo để hoàn thành một công việc định tr-ớc. Thí dụ: Basic, C, Fortran, C...

Ngôn ngữ khai báo: Ngôn ngữ sẽ định nghĩa một loạt các yếu tố và các quan hệ, đồng thời cho phép bạn có thể tiến hành xếp hàng đối với những kết quả xác định. Thí dụ: Prolog, SQL (Structured Query Language)

Điều then chốt trong việc lập trình chuyên dụng là môđun hóa ngôn ngữ - đó là sự phát triển sao cho nhiệm vụ lập trình có thể phân phối đ-ợc cho các thành viên của một nhóm lập trình, và kết quả đạt đ-ợc là các bộ phận khác nhau sẽ hoạt động phù hợp với nhau khi nhiệm vụ của từng ng-ời hoàn thành. Ngôn ngữ lập trình môđun, nh- Module-2 hoặc ngôn ngữ h-ớng đối t-ợng nh- C++, sẽ cho phép từng lập trình viên có thể tập trung vào việc lập mã, biên dịch và gỡ rối các module ch-ơng trình riêng biệt, đồng thời có thể cho chạy (kiểm tra thử) riêng từng module của mình. Khi từng module riêng đã chạy tốt chúng sẽ đ-ợc liên kết với nhau mà không gây trục trặc nào.

#### **1.1.2. Thuật toán (Algorithm)**

Thuật ngữ Algorithm đ-ợc dịch ra tiếng Việt là thuật toán, thuật giải hoặc giải thuật. □ đây chúng tôi dùng từ thuật toán là cách gọi quen thuộc với nhiều ng-ời.

*Thuật toán là một **dãy hữu hạn** các b-ớc, mỗi b-ớc mô tả chính xác các phép toán hoặc hành động cần thực hiện, để giải quyết một vấn đề.*

Để hiểu đầy đủ ý nghĩa của khái niệm thuật toán, chúng ta nêu ra 5 đặc tr-ng sau đây của thuật toán:

1. **Input**                      Mỗi thuật toán th-ờng có một số dữ liệu vào.
2. **Ouput**                     Mỗi thuật toán th-ờng có một số dữ liệu ra.
3. **Tính xác định**          Mỗi b-ớc đ-ợc mô tả chính xác, chỉ có một cách hiểu duy nhất và đủ đơn giản để có thể thực hiện đ-ợc.
4. **Tính dừng**                Thuật toán phải dừng sau một số hữu hạn b-ớc thực hiện
5. **Tính hiệu quả**          Các phép toán trong các b-ớc phải đủ đơn giản để có thể thực hiện đ-ợc.
6. **Tính phổ dụng**          Thuật toán phải có tính tổng quát, có thể áp dụng cho một lớp đối t-ợng.

#### Thí dụ:

##### Thuật toán Euclid

Input:            m,n nguyên d-ơng



Output:  $g$  là ước số chung lớn nhất của  $m$  và  $n$

Phương pháp:

1.  $r = m \bmod n$
2. Nếu  $r=0$  thì  $g:=n$

Ngược lại ( $r>0$ )  $m:=n$ ;  $n:=r$  và quay lại bước 1.

### 1.1.3. Ngôn ngữ C

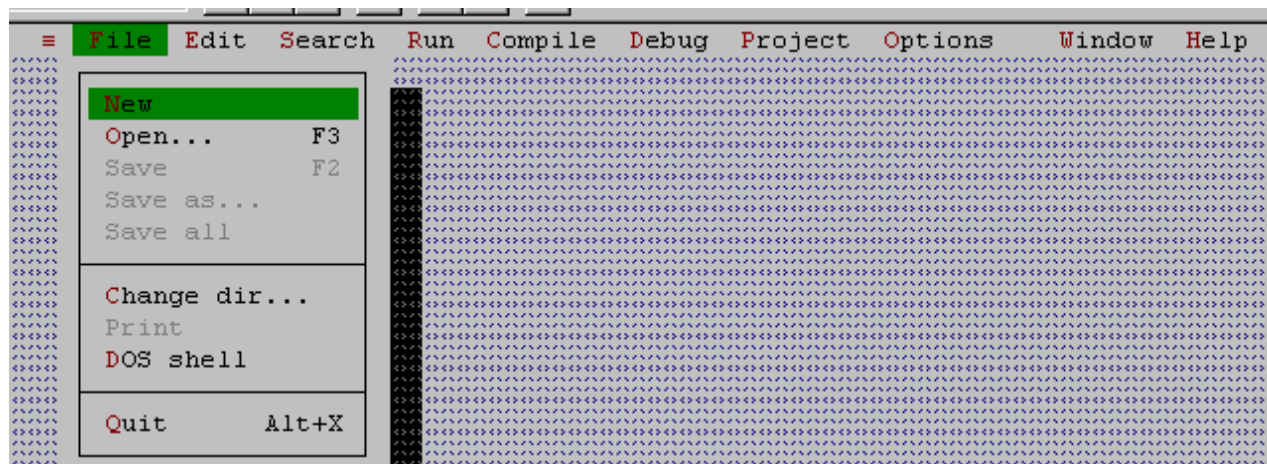
Năm 1970 Ken Thompson sáng tạo ra ngôn ngữ B dùng trong môi trường hệ điều hành UNIX trên máy điện toán DEC PD-7. B cũng là chữ tắt của BCPL (do Martin Richards viết). Năm 1972 Dennis Ritchie của hãng Bell Laboratories (và Ken Thompson) sáng tạo nên ngôn ngữ C nhằm tăng hiệu quả cho ngôn ngữ B. Lúc đầu ngôn ngữ C không được mọi người ưa dùng. Nhưng sau khi D.Ritchie cho xuất bản cuốn "The C Programming Language" 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, OS/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++ (dùng trong DOS và trong Windows), Visual C++ (dùng trong DOS và trong Windows) và Turbo C++ (dùng trong DOS). Có thể nói rằng hiện nay hầu hết các phần mềm ứng dụng đều được viết trực tiếp hoặc gián tiếp bằng C hoặc C++

Trong tài liệu này chúng tôi sẽ giới thiệu C và một số lệnh của C++.

## 1.2. Khởi động và thoát khỏi C

### 1.2.1. Khởi động C

C là phần mềm chạy trên môi trường DOS. Để khởi động C ta tìm và cho chạy chương trình TC.EXE (Turbo C). Sau khi khởi động menu chính của C sẽ xuất hiện trên màn hình cùng với màn hình màu xanh với con trỏ nhấp nháy trên góc trái là nơi ta có thể soạn thảo chương trình. Menu chính của C có các mục sau:



Để đi đến con trỏ về các mục thực đơn ta nhấn F10, sau đó di chuyển con trỏ đến mục cần thiết rồi nhấn {ENTER}, thực đơn kéo xuống sẽ xuất hiện và ta có thể chọn chức năng cần thiết.

Trong quá trình soạn thảo chương trình ta có thể nhấn F9 để dịch và phát hiện lỗi.

Khi soạn thảo xong một chương trình và đã nhấn F9 để dịch và kiểm tra lỗi, ta có thể chạy chương trình đó bằng cách nhấn Ctrl+F9 hoặc chọn mục Run trên thực đơn chính.

### 1.2.2. Mở một cửa sổ mới

- Mở một cửa sổ soạn thảo mới: Chọn **File, New**
- Mở tệp chương trình đã có trên đĩa: Chọn **File, Open** hoặc nhấn **F3**

Tại một thời điểm chỉ có một cửa sổ soạn thảo được kích hoạt, muốn chuyển sang cửa sổ khác gõ **F6**.

Để tiết kiệm bộ nhớ, nên đóng các cửa sổ không cần thiết bằng cách nhấn **Alt+F3** chỉ giữ lại các cửa sổ cần trao đổi dữ liệu với nhau.

### 1.2.3. Lưu văn bản

- Lưu với tên đã có: Chọn **File, Save** hoặc nhấn **F2**
- Lưu với tên mới: Chọn **File, Save as ...**

### 1.2.4. Một vài chú ý khi soạn thảo văn bản

Đánh dấu, cắt, copy, dán các khối:

Cách đánh dấu khối tương tự như các phần mềm soạn thảo văn bản khác, như các thao tác cắt, copy, dán có phần khác:

Xóa: Ctrl+Del      Cắt: Shift+Del

Copy: Ctrl+Ins      Dán: Shift+Ins

Thôi đánh dấu khối: Ctrl+K+B

Hoặc ta chọn các mục trong thực đơn **Edit**

### 1.2.5. Thoát khỏi C

Chọn **File, Quit**

## 1.3. Các thành phần cơ bản của C

### 1.3.1. Bộ chữ viết của C. Từ khóa. Tên

Một ngôn ngữ (viết) bất kỳ, từ ngôn ngữ giao tiếp của con người tới ngôn ngữ máy tính, đều được xây dựng dựa trên một **bộ các chữ** (ký tự) và sau đó là tập hợp các từ có nghĩa được gọi là **từ vựng** (Vocabulary = Total number of words that make up the language). Tiếp theo là phải có quy tắc viết các từ thành câu để diễn tả các hành vi, sự việc, nghĩa là phải tuân thủ cú pháp (Syntax) và Ngữ pháp (Grammar) của ngôn ngữ đó.

(Syntax = Rules for the arrangement of words into phrases and phrases into sentences.

Grammar = Rules for forming words and combining them into sentences)

Ngôn ngữ C (cũng như các ngôn ngữ lập trình khác) được xây dựng trên bộ chữ cái sau:

+ 26 chữ cái la tinh: 26 chữ cái hoa A,B,...,C

26 chữ cái thường a,b,...,c

- + Ký tự gạch nối \_
- + Các ký tự số thập phân 0,1,2,3,4,5,6,7,8,9
- + Các ký hiệu toán học thông thường +, -, \*, /, (, ), =, <, >...
- + Các ký hiệu khác nh- dấu chấm câu và các dấu khác  
. , ; : ' [ ] { } ? ! | % \$ @ # & ^

Nghĩa là gồm tất cả các ký tự trên bàn phím trừ các phím điều khiển không xuất hiện trên màn hình khi ta gõ chúng.

**Dấu cách** (Space) dùng để ngăn cách các từ.

Mọi ngôn ngữ giao tiếp của con người đều có phần từ vựng rất phong phú và đồ sộ, đủ để cho ta diễn đạt được tất cả các hiện tượng và sự kiện xảy ra trong tự nhiên và xã hội. Tuy nhiên cú pháp và ngữ pháp của các ngôn ngữ này thường không hoàn toàn chặt chẽ. Một từ có thể có nhiều nghĩa, một câu đôi khi có nhiều cách hiểu. Ví dụ khi ta nói với bà bán xôi: "Bán cho tôi 2 gói xôi 6 nghìn", thì có thể hiểu là cả hai gói xôi giá tất cả là 6 nghìn, nhưng cũng có thể hiểu là mỗi gói 6 nghìn.

Ngôn ngữ lập trình là ngôn ngữ để con người ra các chỉ thị cho máy tính thực hiện, do đó cú pháp và ngữ pháp phải chặt chẽ, mỗi câu lệnh chỉ có một cách hiểu. Phần từ vựng của ngôn ngữ lập trình vì vậy rất cô đọng để người sử dụng có thể nhớ và sử dụng chính xác. Cũng như các ngôn ngữ lập trình khác, các từ trong phần từ vựng của C được gọi là từ khóa. Các từ khóa phải được dùng đúng cú pháp và không được dùng với mục đích khác như đặt tên cho chương trình, tên các biến...

### Bảng từ khóa của C (key words, reserved words)

Các từ khóa thông dụng trong C

auto	default	float	long	static	unsigned
break	do	for	register	struct	void
base	double	goto	return	switch	public
char	else	if	short	typedef	while
continue	extern	int	sizeof	union	volatile

### Tên trong ngôn ngữ lập trình C (Identifier)

**Tên là dãy ký tự có độ dài tối đa là 127 ký tự, bắt đầu bằng chữ cái, sau đó có thể là các chữ cái hoặc các chữ số và dấu gạch dưới.** Tên không chứa khoảng trống hoặc các ký tự đặc biệt. Ví dụ tên chương trình, tên biến, tên hằng, tên chương trình con. Các tên gồm 2 loại: **tên chuẩn** do C định nghĩa sẵn, hoặc **tên do người sử dụng định nghĩa**.

Ngôn ngữ C **phân biệt chữ hoa và chữ thường**. Ví dụ các tên sau là khác nhau: VIDU, vidu, ViDu, Vidu

### Dấu chấm phẩy

Có một ký tự đặc biệt trong bộ chữ cái của C đó là dấu chấm phẩy (;). Dấu ; được dùng để ngăn cách các câu lệnh. Thông thường sau mỗi câu lệnh người ta thường gõ dấu chấm phẩy nhưng nó không mang ý nghĩa là báo hiệu kết thúc câu lệnh mà chỉ để phân cách với câu lệnh tiếp theo.

### Lời giải thích

Trong ch- ơng trình có những đoạn văn bản hoàn toàn không mang ý nghĩa gì trong quá trình thực hiện ch- ơng trình. Đó là những lời giải thích. Chúng làm cho ch- ơng trình rõ ràng hơn và dễ hiểu. C sẽ bỏ qua tất cả các ký tự trong cặp dấu /\* \*/, hoặc các ký tự nằm sau dấu // cho đến cuối dòng.

#### 1.3.2. Một ch- ơng trình C đơn giản

Ch- ơng trình hiện lên màn hình dòng chữ "Turbo C xin chào bạn"

```
#include <stdio.h>
void main()
{printf("%s", "Turbo C xin chào bạn");
}
```

*Chú ý.* Khi chạy ch- ơng trình này bạn sẽ chỉ thấy màn hình nháy một cái rồi vẫn nh- cũ, d- ờng nh- không có gì thay đổi. Sở dĩ nh- vậy là vì khi ch- ơng trình C cho hiện câu "Turbo C xin chào bạn" lên màn hình kết quả (là màn hình để hiện các kết quả, khác với màn hình soạn thảo ch- ơng trình mà bạn đang nhìn thấy) C không chờ để ta xem kết quả đó mà trở về ngay cửa sổ soạn thảo. Nếu bạn dịch ch- ơng trình này ra tệp \*.exe và chạy ở môi tr- ờng DOS thì bạn sẽ thấy kết quả hiện ra trên màn hình.

Trong môi tr- ờng của C, tức là bạn đang soạn thảo và chạy ch- ơng trình, thì bạn có thể xem màn hình kết quả bằng cách nhấn Alt+F5. Khi đang ở màn hình kết quả bạn có thể nhấn phím bất kỳ để trở về màn hình soạn thảo.

Bạn cũng có thể sửa đổi ch- ơng trình trên đây để C hiện kết quả trên màn hình và chờ cho đến lúc bạn nhấn phím bất kỳ mới chạy tiếp. Muốn vậy bạn phải đ- a thêm tệp <conio.h> là tệp chứa các thao tác vào ra trên màn hình (conio = console input-output). Sau đó bạn thêm lệnh clear(); để xóa màn hình tr- ớc khi hiển thị câu chào, và lệnh getch(); để tạm dừng ch- ơng trình sau khi hiện câu chào cho đến khi nhấn phím bất kỳ.

```
#include <stdio.h>
#include <conio.h>
void main()
{clrscr();
 printf("%s", "Turbo C xin chào bạn");
 getch();
}
```

### 1.4. Cấu trúc chương trình trong C

#### 1.4.1. Các thành phần của một ch- ơng trình C

Phần quan trọng nhất không thể thiếu của một ch- ơng trình C là hàm (hoặc thủ tục) main().

Thông th- ờng hàm main() có dạng

```
void main()
{
 //các lệnh
}
```

```
hoặc
int main()
{
 //các lệnh
 return(0);
}
```

Thông th- ờng một ch- ơng trình C đầy đủ có các phần sau:

- Chỉ dẫn tiền xử lý (preprocessor directive)

- Các bao hàm (#include)
- Các vĩ lệnh (#define)
- Khai báo toàn cục
  - Khai báo các nguyên mẫu cho hàm (function prototype)
  - Khai báo các biến toàn cục
- Hàm main()
- Định nghĩa chi tiết các hàm do người sử dụng vừa khai báo trên.

Phần "Định nghĩa chi tiết các hàm" có thể đặt ở ngay sau phần khai báo toàn cục.

Một số hàm cơ bản và các tệp bao hàm tương ứng:

Tệp bao hàm	Các hàm được định nghĩa trong tệp
stdio.h	printf,scanf,gets
conio.h	getch,clrscr
iostream.h	cin,cout
iomanip.h	setw,setprecision, (cout<<oct<<a<<hex<<b;)
math.h	sqrt,abs,fabs
stdlib.h	abs,random,randomize,calloc,malloc,atof,atoi
ctype.h	toupper,tolower,isdigit
string.h	strcmp,strcmpi,strcpy, strcat, strlen, strlenw,strupr

#### 1.4.2. Soạn thảo và chạy một chương trình C

##### 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 \ tr-ớc khi xuống dòng**.

##### b. Dịch và chạy chương trình

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 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...

##### 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. 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 mới có nguyên mẫu mà chương trình lại có khai báo chi tiết.

Dịch và chạy chương trình:

Nếu ch-ơng trình ch- a viết xong ta có thể nhấn F9 để dịch và sửa lỗi. Khi ch-ơng trình đã t-ơng đối hoàn chỉnh thì ta có thể nhấn Ctrl+F9 để dịch và chạy ch-ơng trình.

## CH- ONG 2. HÀNG, BIẾN VÀ MẢNG(23)

### 2.1. Các kiểu dữ liệu cơ sở

C có một số kiểu dữ liệu cơ sở nh- sau:

Kiểu	Miền xác định	Kích th- ớc(byte)
char (signed char)	-128 → 127	1
unsigned char	0 → 255	1
int	≈ -32 000 → 32 000	2
long	≈ -2 tỷ → 2 tỷ	4
unsigned int	≈ 0 → 65 000	2
unsigned long	≈ 0 → 4 tỷ	4
float	3.4e-38 → 3.4e+38	4
double	1.7e-308 → 1.7e+308	8
long double	3.4e-4932 → 3.4e+4932	10

#### Kiểu char

Một giá trị kiểu char chiếm một byte và đ- ọc biểu diễn một ký tự t- ơng ứng trong bảng ASCII. Khác với Pascal, trong C giá trị của biến char có thể đ- ọc hiểu là số. Thí dụ nếu ta khai báo char ch; thì có hai cách gán ch=65 hay ch='A' cùng cho một kết quả. Khi hiển thị ch ta nếu ta dùng khuôn dạng "%d" thì đ- ọc giá trị 65, còn dùng khuôn dạng "%c" thì đ- ọc ký tự A. Bảng sau đây cho sự khác biệt giữa char và unsigned char.

Mã ASCII	0	1	...	127	128	129	...	254	255
Giá trị biến kiểu char	0	1	...	127	-128	-127	...	-2	-1
Giá trị biến kiểu unsigned char	0	1	...	127	128	129	...	254	255

Theo thứ tự bảng trên đây thì sau khi khai báo char ch; nếu ta gán ch=200 thì thực chất ch chỉ nhận giá trị -56, do đó phép thử ch==200 sẽ cho kết quả sai.

Một cách tổng quát, nếu biến ch đ- ọc khai báo là char ch, thì khi gán ch=k, trong đó k là một số tự nhiên. Tr- ớc hết k đ- ọc chia cho 256 để lấy phần d- . Thí dụ k=257 thì k sẽ đ- ọc nhận giá trị là 1. Lệnh gán ch=k cho ta kết quả nh- sau: ch=k nếu  $k \in [-128, 127]$ ,  $ch = -(256-k)$  nếu  $k \in [128, 256]$ .

Phân loại ký tự(24): 0-31, 32-126, 127-254

**Chú ý:**Trong ch- ơng trình một ký tự phải đ- ọc bao trong dấu nháy đơn. Thí dụ char ch='A'; ch='1' sẽ cho ta ch=49, còn ch=1 lại cho ta ch biểu diễn ký tự coa mã ASCII là 1.

#### Kiểu nguyên

Vậy có thể xem kiểu ký tự cũng là một dạng đặc biệt của kiểu nguyên. Khi ta dùng khuôn dạng "%c" để hiện một số nguyên  $n > 255$  thì ký tự  $n \% 256$  sẽ đ- ọc hiện thí dụ nếu  $n = 304$  thì ký tự ứng với mã ASCII đ- ọc in, tức là ký tự '0'.

#### Kiểu dấu phẩy động

(25)

## 2.2. Hằng (25)

Hằng là các đại lượng (có thể có tên hoặc không) mà giá trị của nó không thể thay đổi trong quá trình chạy chương trình.

**char**

### **int (hệ 10,8,16)**

Hằng int trong hệ 8 (Octal) được bắt đầu bằng số 0, còn hằng int trong hệ 16 (Hexadecimal) bắt đầu bằng số 0 và ký tự x (in thường hoặc in hoa). Thí dụ

011 = 9 //Nh- vậy nếu chỉ viết 11, thì đó là số 11; nh- ng nếu thêm 0 phía tr- ước thì = 9  
0x11 = 17

Chú ý: Khi hiển thị một số nguyên dưới dạng bát phân bằng lệnh printf() ta dùng khuôn dạng %o và chữ 0, còn với dạng thập lục phân thì dùng %x và ký tự x (in thường hoặc in hoa). Ví dụ

printf("%o",9) hoặc printf("%x",17) cho kết quả trên màn hình là 11

printf("%o",011) hoặc printf("%x",0x11) cho kết quả trên màn hình là 11

### **float và double**

**Một số hằng có sẵn: M\_PI, M\_E,... (trong math.h)**

### **Các hằng ký tự đặc biệt**

'\n', '\t', '\a', '\0',...

Lưu ý: hằng '\0' có mã là 48, còn '\0' ứng với ký tự \0, có mã là 0.

### **Hằng chuỗi ký tự(28)**

Là một dãy ký tự bất kỳ đặt trong 2 dấu "", thí dụ "Hà Nội".

Chú ý: 'a' là hằng ký tự gồm một byte, còn "a" là hằng chuỗi ký tự gồm 2 byte, 1 byte chứa ký tự a, **byte cuối cùng chứa ký tự '\0' là ký tự báo kết thúc chuỗi.**

### **Chú ý:**

Khi ta dùng hàm strlen(st) để xác định độ dài của chuỗi ký tự thì ký tự '\0' không được tính vào độ dài này. Thí dụ 3 biến st1,st2,st3 sau đây đều có độ dài là 3

```
char st1[]="123";  
char st2[7]="123";  
char *st="123";
```

Tuy nhiên thực ra trong bộ nhớ để lưu trữ một chuỗi dài 3 thì máy cần 4 vị trí vì một vị trí để lưu trữ ký tự '\0'. Do đó khác với thao tác với mảng, khi thao tác với chuỗi ta không nên nhập số liệu cho vị trí cuối cùng. Thí dụ nếu khai báo char st[10] thì ta không nên nhập số liệu cho vị trí 9. Điều này có thể thấy được qua ví dụ sau:

Giả sử ta khai báo:

```
char st[5]="123";
```



thì khi đó chuỗi có độ dài 3, 3 vị trí chứa số liệu là `st[0]=='1', st[1]=='2', st[2]=='3'`; vị trí `st[3]=='\0'`. Còn nếu ta khai báo

```
char st[3]="123";
```

Thì tuy máy không báo lỗi nh- ng thực ra có một số điều không hợp lý. Nếu ta dùng hàm `strlen()` để kiểm tra độ dài thì thấy rằng độ dài là 4, vị trí `st[3]` không phải là `\0` mà là một ký tự nào đó, ký tự ở vị trí `st[4]` mới là `\0`. Do đó nếu ta dùng lệnh

```
while(st[i]!='\0') putchar(st[i]);
```

chẳng hạn thì sau khi máy in ra các ký tự '1', '2', '3' thì còn in tiếp một ký tự nào đó nữa rồi mới dừng.

### **Tên hằng và cách gán giá trị cho hằng(29)**

```
#define max 100
```

```
#define str "Ha Noi"
```

```
const float a=10.123;
```

```
const char str2[]="Hai Phong";
```

```
const char *str3="Tay Ho";
```

Không nên dùng `const char str2[10]="Hai Phong"`, vì có thể độ dài của hằng vượt quá độ dài định nghĩa.

Chú ý: Hằng có thể là các biến chiếm dụng bộ nhớ thí dụ `int m=10`; nh- ng cũng có thể là những giá trị tạm thời, không đ- ọc phân phối bộ nhớ thí dụ `M_PI, M_E, printf("\n%s", "Ha Noi");`

## **2.3. Kiểu enum (30)**

Giả sử ta phải tiến hành một số công việc lặp lại theo các ngày trong tuần. Trong chương trình ta viết

```
for(int i=2;i<=8;i++) {Các lệnh}
```

Tuy nhiên theo cách này thì các con số 2, 3, ... không cho ta ý nghĩa gọi nhớ đó là các ngày trong tuần. Cho ta ta định nghĩa biến kiểu liệt kê phục vụ cho mục đích này. Ta có thể làm nh- sau

```
enum ngay_tt {hai, ba, tu, nam, sau, bay, chunhat};
```

```
ngay_tt i;
```

```
for(i=hai;i<=chunhat;i++) {các lệnh}
```

Giả sử các lệnh = `cout<<i<<" "`; thì ta có kết quả là 0 1 2 3 4 5 6

Vì biến enum thực chất là biến nguyên, do đó các dòng lệnh trên có thể viết đơn giản hơn nh- sau:

```
enum {hai, ba, tu, nam, sau, bay, chunhat};
```

```
int i;
```

```
for(i=hai;i<=chunhat;i++) {các lệnh}
```

## **2.4. Biến (32)**

Mọi biến phải đ- ọc khai báo tr- ớc khi sử dụng. Việc khai báo đ- ọc thực hiện theo mẫu sau:

<Kiểu biến> <tên biến 1>,<tên biến 2>,...,<tên biến n>;

Có thể khai báo ngay khi sử dụng, thí dụ  
for(int i=0;i<n;i++) [các lệnh]

### ***Vị trí của các khai báo***

Trong C++ thì biến có thể khai báo tại vị trí bất kỳ trong chương trình.

### ***Khởi đầu cho các biến***

```
float a=10.123;  
char str2[]="Hai Phong";  
char *str3="Tay Ho";  
int x[] = {1,2,3};  
float y[4] = {1.2, 3.2, 4.3, 6.5};
```

Nh- ng ***không thể viết:***

~~float \*py = {1.2, 3.2, 4.3, 6.5};~~

Nh- ng ***với chuỗi thì lại đ- ọc:***

```
char st1[]="123";  
char st2[7]="123";  
char *st="123";
```

### ***Lấy địa chỉ của biến***

Mỗi biến đ- ọc cấp phát một vùng nhớ gồm một số byte liên tiếp. Địa chỉ của byte đầu tiên là địa chỉ của biến. Địa chỉ của biến có thể nhận đ- ọc bằng phép toán

&<tên biến>

**Chú ý:** Ta chỉ có thể lấy địa chỉ của một biến. Thí dụ các lệnh sau đây là sai

```
int *pn;  
const int m=10;  
pn=&m;  
int k;  
pn=&(k+1);
```

### ***Biến địa chỉ***

Địa chỉ của một biến là số nguyên nh- ng ta không thể thao tác chúng nh- các số nguyên thông thường. Thí dụ ta không thể gán

```
int x;  
int m=&x;
```

Địa chỉ của một biến ngoài độ lớn chỉ vị trí của biến trong bộ nhớ còn mang thông tin đó là địa chỉ của biến loại gì. Ta biết rằng địa chỉ của 2 biến nguyên liên tiếp thì cách nhau 2 byte, 2 số thực liên tiếp cách nhau 4 byte,... Để có thể l- u trữ và thực hiện các phép toán trên các giá trị địa chỉ, C định nghĩa một kiểu biến nguyên đặc biệt mà giá trị của chúng là địa chỉ của các biến và gọi biến này là con trỏ. Khi định nghĩa con trỏ ta phải chỉ rõ đó là con trỏ của biến loại gì. Thí dụ để khai báo một con trỏ kiểu nguyên ta viết

```
int *p;
```

Và khi đó ta có thể gán  $p = \&x$ ; trong đó  $x$  là biến nguyên nào đó.

### **Lấy giá trị của một biến thông qua địa chỉ**

Giả sử ta  $p$  là một con trỏ. Khi đó  $*p$  sẽ là giá trị đ-ọc chứa trong vùng bộ nhớ mà  $p$  trỏ tới.

Nh- vậy với biến  $x$  bất kỳ thì ta có  $\&x$  chính là giá trị  $x$ .

## **2.5. Mảng (33) và chuỗi**

Trong phần trên ta hiểu biến là một vùng bộ nhớ đ-ọc đặt tên và có thể l-u trữ một kiểu giá trị nào đó. Để biểu diễn nhiều giá trị ta phải dùng nhiều biến. Tuy nhiên nhiều khi số biến lại phụ thuộc một tham số nào đó. Thí dụ số thành phần của một vectơ phụ thuộc vào một số  $n$ , còn số các phần tử của một ma trận lại phụ thuộc số hàng và số cột. Trong những tr-ờng hợp này ng-ời ta dùng một kiểu dữ liệu có cấu trúc là mảng để biểu diễn số liệu. Mảng là một tập hợp nhiều phần tử có cùng một kiểu giá trị và có chung một tên. Về mặt nào đó có thể coi mảng cũng là biến, nh-ng là biến có cấu trúc phức tạp hơn.

Định nghĩa mảng

Khai báo `int a[10]` sẽ dành 10 vị trí liên tiếp trong bộ nhớ, mỗi vị trí gồm 2 byte có thể l- - trữ đ-ọc một số nguyên và các vị trí này t-ơng ứng là  $a[0], a[1], \dots, a[9]$

Khai báo `int a[8][10]` sẽ dành 100 vị trí liên tiếp trong bộ nhớ, mỗi vị trí gồm 2 byte có thể l- - trữ đ-ọc một số nguyên và các vị trí này t-ơng ứng là  $a[0][0], a[0][1], \dots, a[0][9], a[1][0], a[1][1], \dots, a[1][9], \dots, a[7][0], a[7][1], \dots, a[7][9]$ .

**Chú ý:** Khi ta khai báo một mảng có kích cỡ định tr-ớc, thí dụ

`int a[10];` thì  $a$  là một hằng địa chỉ, do đó ta không thể gán  $a = \langle \text{địa chỉ đ-ọc} \rangle$ . Tuy nhiên nếu ta định nghĩa một biến con trỏ, thí dụ `int *pn;` thì ta có thể dùng phép gán cho  $pn$ . Với một chuỗi

`char *s="123";` chẳng hạn, ta có  $s$  là địa chỉ của biến đầu tiên của chuỗi, tức là  $s[0]$ , do đó  $*s$  chính là giá trị của  $s[0]$ , còn  $*(s+1)$  là giá trị của  $s[1], \dots$

## **2.6. Định nghĩa kiểu bằng typedef (36)**

```
typedef int ng;
```

```
typedef int ma[8][10];
```

```
ng x,y;
```

```
ma a,b;
```

Các biến  $x, y$  có kiểu nguyên và  $a, b$  có kiểu là ma trận các phần tử nguyên cấp  $8 \times 10$ .

## **2.7. Khối lệnh (37)**

Khối lệnh là tập hợp các lệnh đ-ọc bao trong cặp dấu `{ }`. Các khối lệnh có thể lồng nhau. Biến có thể đ-ọc khai báo toàn cục (nh- đã nói ở trên), hoặc khai báo trong bất kỳ khối lệnh nào tr-ớc khi sử dụng. Tầm hoạt động của nó đ-ọc bắt đầu từ khi nó đ-ọc khai báo cho đến hết khối nhỏ nhất chứa khai báo nó. Nh- vậy khối  $A$  là khối nhỏ nhất chứa khai báo của một biến  $x$ , thì có thể sử dụng trong các khối con của  $A$  từ sau vị trí khai báo biến  $x$ .

## 2.8. Vài nét về hàm và chương trình (41)

Khác với Pascal, C *không cho phép các hàm lồng nhau*. Chương trình có thể có nhiều hàm trong đó hàm main() là thành phần bắt buộc.

## 2.9. Biến, mảng tự động (42)

### 2.9.1. Định nghĩa

Các biến được khai báo bên trong thân hàm (kể cả hàm main()) được gọi là biến tự động hay cục bộ.

### 2.9.2. Phạm vi hoạt động và thời gian tồn tại

Biến tự động chỉ tồn tại trong khối lệnh mà nó được định nghĩa

Ví dụ

```
void main()
{ auto int x;
}
```

Biến tự động là ngầm định cho các biến cục bộ, do đó từ khóa auto thường được bỏ qua.

Chú ý biến tự động chỉ được khai báo trong các hàm, do đó ví dụ khai báo

```
auto int i;
void main()
{ auto int x;
}
```

là sai.

### 2.9.3. Khởi đầu cho biến và mảng tự động

Biến ngoài (toàn cục) khi khai báo thì tự động được gán bằng 0 hoặc rỗng. Ngược với biến tự động khi khai báo xong mà không gán giá trị khởi đầu hoặc gán giá trị thì giá trị không xác định, là một giá trị vô nghĩa nào đó.

Khác với C, trong C++ có thể khởi đầu cho mảng hoặc biến, thí dụ

```
void main()
{ float a[4] = {2.3,1.5,2.6,3.8};
  int b[][3] = {1,2,3,4,5,6};
  int c[2][3] = {2,3,4,5,6,7};
}
```

## 2.10. Biến, mảng ngoài (43)

### 2.10.1. Định nghĩa

Các biến được khai báo bên ngoài hàm (kể cả hàm main()) được gọi là biến ngoài.

### 2.10.2. Phạm vi hoạt động và thời gian tồn tại

Biến ngoài tồn tại từ vị trí được khai báo cho đến hết chương trình.

Ví dụ:

```
#include "define_h.cpp"
void v1();
void v2();
void main()
{clrscr();
  v1();
  v2();
  getch();
}
int i;
void v1()
{ i=10;
  cout<<endl<<i;
}
void v2()
{cout<<endl<<i;
}
```

Chương trình trên cho kết quả là

10  
10

Còn nếu ta thêm lệnh

```
cout<<i;
```

trong chương trình main() thì máy sẽ báo lỗi là biến i chưa được định nghĩa.

#### Chú ý:

- Nếu trong một hàm nào đó ta khai báo một biến cùng tên với biến ngoài thì biến ngoài không có tác dụng trong hàm này. Khi ta dùng đến biến cùng tên đó thì biến luôn được hiểu là biến địa phương.

Ví dụ sau minh họa cho điều này:

```
#include "define_h.cpp"
void v1();
void v2();
void main()
{clrscr();
  v1();
  v2();
  getch();
}
int i;
void v1()
{int i=10;
  cout<<endl<<i;
}
void v2()
{cout<<endl<<i;
}
```

Kết quả chạy chương trình là

10  
0

- Nếu ch- ơng trình viết trên nhiều tệp và các tệp đ- ọc dịch độc lập, thì phạm vi sử dụng của biến, mảng ngoài có thể mở rộng từ tệp này sang tệp khác bằng từ khóa extern (xem ch- ơng 11)

### 2.10.3. Khởi đầu cho biến và mảng ngoài

Biến ngoài (toàn cục) khi khai báo thì tự đ- ọc tự động gán bằng 0 hoặc rỗng. Quy tắc khai báo và khởi đầu cũng giống nh- biến tự động.

## 2.11. Toán tử sizeof (47)

sizeof(Kiểu dữ liệu)

sizeof(Đối tượng dữ liệu)

ứng dụng:

```
int x[] = {1,2,3,4};
```

```
int n = sizeof(x)/sizeof(int); // Xác định cỡ của mảng.
```

## 2.12. Biến tĩnh, mảng tĩnh (48)

ý nghĩa của biến tĩnh có thể thấy rõ qua ví dụ sau:

```
#include "define_h.cpp"
void vv()
{static int i=1;
  cout<<i++<<" ";
}
void main()
{clrscr();
  for(int i=1;i<=3;i++) vv();
  getch();
}
```

Kết quả chạy ch- ơng trình là:

1 2 3

## CHƯƠNG 3. BIỂU THỨC(53)

### 3.1. Khái niệm biểu thức(expression, 53)

Biểu thức là một giá trị được tạo nên do sự kết hợp các phép toán với các hằng, biến, phần tử mảng, hàm... Như vậy bản thân các hằng, biến, phần tử mảng, hàm cũng là các biểu thức.

### 3.2. Phép toán số học(54)

+, -, \*, / %

### 3.3. Các phép thao tác bit(55)

&, |, ^, <<, >>, ~

### 3.4. Các phép toán quan hệ và logic(57)

>, >=, <, <=, ==, !=

&&, ||

### 3.5. Phép chuyển đổi kiểu giá trị(59)

int(a), float(b),...

sqrt(float(n));

### 3.6. Phép toán tăng giảm(61)

n++, ++n, n--, --n

Ví dụ:

```
int m,n,x,y;
```

```
m=n=5;
```

```
x = m++;
```

```
y = ++n;
```

```
cout<<x<<" "<<y<<" "<<m<<" "<<n;
```

Cho kết quả là 5 6 6 6

### 3.7. Câu lệnh gán và biểu thức(62)

```
i+=2;
```

```
i*=2;
```

```
x*=y+3 //x=x*(y+3)
```

### 3.8. Biểu thức điều kiện(63)

```
e1?e2:e3
```

e1 là biểu thức logic

Ví dụ:

```
max = a>b? a:b;
```

Câu lệnh

```
printf("\n%8.2f",a<b?a:b);
```

sẽ in ra cực tiểu của hai biến thực a và b.

### 3.9. Thứ tự - u tiên các phép toán(66)

## CH- ƠNG 4. VÀO RA(70)

### 4.1. Hàm printf (70)

int printf(const char \*dk, [danh sách các đối]);

Đối dk là con trỏ kiểu char chứa địa chỉ của chuỗi điều khiển.

#### 4.1.1. Chuỗi điều khiển

gồm 3 loại ký tự:

- Các ký tự điều khiển nh- : \n Xuống dòng, \t Dấu tab
- Các đặc tả chuyển dạng và tạo khuôn (gọi tắt là đặc tả)
- Các ký tự để hiển thị ra màn hình. Đối với một số ký tự đặc biệt nh- dấu ', ", \ thì có cách viết đặc biệt hơn: Tr- ớc các ký tự đó có thêm ký tự \

Phân quan trọng và đáng chú ý nhất trong chuỗi điều khiển là **phần đặc tả** có dạng tổng quát nh- sau:

%[-][fw][.pp]<Ký tự chuyển dạng>

Vậy một đặc tả phải bắt đầu bằng dấu % và kết thúc bằng ký tự chuyển dạng.

- a. fw là độ rộng dành cho tr- ờng ra. Nếu fw < độ rộng tr- ờng thì độ rộng in ra bằng độ rộng thực tế của tr- ờng. Nếu fw bắt đầu bằng số 0 thì vị trí trống bên trái đ- ọc thay bằng các số 0. fw có thể đ- ọc thay bằng dấu, thí dụ printf("%\*.2f ",a); Khi đó độ rộng của đối t- ơng ứng là độ rộng thực tế
- b. Dấu trừ (-): Nếu có dấu - thì dồn trái, không có thì dồn phải.
- c. pp chỉ nên dùng cho số thực. pp là số chữ số sau dấu chấm thập phân, nếu không có thì đ- ọc hiểu là pp=6.
- d. Ký tự chuyển dạng: quy định dạng in ra của đối t- ơng, thí dụ d là số nguyên, f là số thực, o là số bát phân, x là số thập lục phân...

Bảng các ký tự chuyển dạng (73)

Ký tự chuyển dạng	Kiểu của đối	Đối đ- ọc hiển thị nh-
c	char	ký tự
d hoặc i	int	số nguyên hệ 10
ld hoặc li	long	Số nguyên dài
u	int	Số nguyên không âm
o	int	Số nguyên không âm hệ 8
lo	long	Số nguyên không âm hệ 8
x	int	Số nguyên không âm hệ 16
lx	long	Số nguyên không âm hệ 16
f	float hoặc double	Số thực dạng thập phân
e	float hoặc double	Số thực dạng khoa học.
g	float hoặc double	Dạng khoa học hoặc thập phân tùy thuộc loại nào ngắn hơn. Không có các số 0 vô nghĩa.
s	Xâu ký tự	Các ký tự



Ví dụ:

Ta cần in ra dòng thông tin:

Năng suất tăng 19.25%

Trong đó giá trị 19.25 là giá trị của biến thực ns, ta viết:

```
printf("\nNang suat tang %5.2f%",ns);
```

Giả sử ta cần in ra

Năng suất tăng 19.25% trong năm 2000

Trong đó giá trị 19.25 là giá trị của biến thực ns, 2000 là giá trị của biến nguyên nam có thể thay đổi tùy thuộc từng trường hợp, nếu ta viết:

```
printf("\nNang suat tang %5.2f% trong nam %4d",ns,nam);
```

thì về mặt cú pháp là đúng nhưng lệnh không in ra điều ta mong muốn. Trong trường hợp này ta nên tách làm 2 lệnh:

```
printf("\nNang suat tang %5.2f%",ns);  
printf(" trong nam %4d",nam);
```

#### **4.1.2. Danh sách các đối**

Các đối cần được phân cách nhau bởi dấu phẩy. Đối có thể là hằng, biến, hàm hay là kết hợp của chúng bằng các phép toán. Vậy các đối chính là các biểu thức. Nói chung có bao nhiêu đặc tả thì có bấy nhiêu đối. Nếu số đối nhiều hơn số đặc tả thì các đối không có đặc tả tương ứng sẽ bị bỏ qua. Còn nếu số đối ít hơn số đặc tả thì máy sẽ bị rối và có thể dẫn đến những kết quả sai.

## **4.2. Hàm scanf (78)**

```
int scanf(const char *dk, [danh sách các đối]);
```

#### **4.2.1. Danh sách các đối**

Các đối cần được phân cách nhau bởi dấu phẩy. Đối phải là một con trỏ chứa địa chỉ của một biến nào đó. Vì tên của chuỗi ký tự là địa chỉ nên không cần dấu lấy địa chỉ đứng trước tên chuỗi.

#### **4.2.2. Chuỗi điều khiển**

Gồm các ký tự đặc tả chuyển dạng. Mỗi đặc tả thường có một đối tương ứng **phần đặc tả** có dạng tổng quát như sau:

`%[*][dr]<Ký tự chuyển dạng>`

- a. Nếu có dấu \* thì thường vào vẫn được đọc nhưng giá trị của nó không được lưu trong bộ nhớ.

Thí dụ:

```
scanf("%d%*d%d",&a,&b);
```

Và khi chương trình chạy lệnh này ta nhập các số 1 2 3 thì ta có a=1,b=3 còn giá trị 2 thì bị bỏ qua. Cách này trong thực tế ít khi dùng.

b. dr nói chung vắng mặt, nếu có mặt thì ý nghĩa của nó có thể hiểu qua ví dụ sau:

```
char a,b;
scanf("%2s%3s",&a,&b);
```

Nếu ta nhập dòng vào là 12345678 chẳng hạn thì 2 ký tự 12 đ- ọc gán cho a, 3 ký tự tiếp theo đ- ọc gán cho b, tức là 345. các ký tự còn lại bị bỏ qua.

Nói chung đặc tả chỉ gồm ký tự % và đặc tả chuyển dạng đ- ọc thống kê trong bảng sau:

Bảng các ký tự chuyển dạng (73)

Ký tự chuyển dạng	Kiểu của đối	Đối đ- ọc hiển thị nh-
c	char	ký tự
d hoặc i	int	số nguyên hệ 10
ld hoặc li	long	Số nguyên dài
u	int	Số nguyên không âm
o	int	Số nguyên không âm hệ 8
lo	long	Số nguyên không âm hệ 8
x	int	Số nguyên không âm hệ 16
lx	long	Số nguyên không âm hệ 16
<i>f hoặc e</i>	<i>float</i>	<i>Số dấu phẩy động</i>
<i>lf hoặc le</i>	<i>double</i>	<i>Số dấu phẩy động</i>
<i>g (Không có)</i>		<i>Không có dạng này</i>
s	Xâu ký tự	Các ký tự

Chú ý: Ký tự chuyển dạng của số nguyên không âm hệ 8 là chữ o, số nguyên không âm hệ 16 là ox; nh- ng biểu diễn số nguyên không âm hệ 8 bắt đầu bằng số 0, còn số hệ 16 thì bắt đầu bằng số 0 và chữ x.

Các dòng nghiêng và đậm chính là sự khác biệt giữa hàm scanf và printf.

Ta có thể nhập các tr- ờng vào liên tiếp nhau, sao cho mỗi tr- ờng cách nhau ít nhất một khoảng trắng. Trong đó khoảng trắng đ- ọc hiểu là dấu cách, dấu tab hoặc dấu xuống dòng. Nh- vậy giữa các tr- ờng vào có thể chứa một số khoảng trắng tùy ý. Nh- vậy ta không thể nhập một chuỗi ký tự có chứa khoảng trắng.

Ví dụ về hàm scanf:

```
int n;float a;double b;char st[5]
scanf("%d%f%lf%s",&n,&a,&b,st);
```

### 4.3. Đ- a ra máy in- hàm fprintf (83)

```
int fprintf(stdprn,const char *dk, [danh sách các đối]);
```

Tất cả những điều đã nói ở lệnh printf đều đúng. Chỉ có một điểm khác biệt: trong lệnh printf thiết bị ra là màn hình còn với lệnh fprintf thì thiết bị ra là máy in.

### 4.4. Dòng vào stdin và các hàm nhập dữ liệu (93)

Tr- ớc hết ta hãy chạy thử đoạn ch- ơng trình đơn giản sau:

```
char ca,cb;
ca=cb='z';
scanf("%c",&ca);
scanf("%c",&cb);
printf("\n%c %c",ca,cb);
```

Nếu ta gõ a ↵ thì lệnh scanf thứ 2 bị bỏ qua.

Lệnh printf tiếp theo chỉ hiện lên màn hình ký tự

a

mặc dù ta đã gán cb='z'. Như vậy lệnh scanf thứ 2 tuy có vẻ như không được thực hiện, nhưng trong thực tế giá trị của cb đã thay đổi.

Trước khi giải thích hiện tượng trên đây chúng tôi giới thiệu thêm một vài khái niệm liên quan đến các thao tác vào ra.

Turbo C định nghĩa một số dòng xuất nhập chuẩn, thực chất là các vùng đệm chuyên cho các thao tác vào ra. Các dòng này được đặt tên và có thể thao tác trong chương trình.

Predefined streams automatically opened when the program is started.

Name	Meaning
stdin	Standard input device
stdout	Standard output device
stderr	Standard error output device
stdaux	Standard auxiliary device
stdprn	Standard printer

Khi gặp một lệnh nhập số liệu và ta bắt đầu gõ từ bàn phím thì các ký tự được gửi vào stdin. Chương trình sẽ chờ cho đến khi ta gõ phím ↵ báo hiệu kết thúc lệnh nhập số liệu thì chương trình bắt đầu đọc các ký tự từ stdin để gán cho các biến nhập. Nếu số liệu trong stdin chưa đủ thì chương trình lại dừng để ta nhập tiếp. Còn nếu số liệu có nhiều hơn thì chương trình chỉ lấy vừa đủ để gán cho các biến vào, phần còn lại vẫn còn lại trong stdin. Khi gặp lệnh nhập số liệu tiếp theo thì trước hết chương trình xem trong stdin có số liệu không, nếu có đủ số liệu thì chương trình lấy ngay các số liệu đó để gán cho biến vào mới và lệnh nhập số liệu đó bị bỏ qua. Ta có cảm giác là lệnh không được thực hiện nhưng thực ra đã được thực hiện nhưng có thể không như ý ta muốn.

Trở lại đoạn chương trình trên đây. Khi nhập số liệu cho ca ta đã gõ a và nhấn ↵ thì cả ký tự a và ký tự ↵ (\n, LF có mã là 10) đều được đưa vào stdin. Lệnh scanf đọc ký tự a từ stdin nhưng không loại ký tự \n ra khỏi stdin. Do vậy khi tới lệnh scanf tiếp theo thì chương trình lấy ngay giá trị \n để gán cho cb mà không cần chờ người sử dụng nhập số liệu mới. Khi hiện ra màn hình thì ký tự \n không nhìn thấy do đó ta chỉ thấy chữ a mà thôi. Nếu bây giờ ta sửa lại lệnh printf thành

```
printf("\n%d %d",ca,cb);
```

Ta sẽ thấy kết quả là

97 10

97 là mã của a, còn 10 là mã của \n.

Turbo C đã khắc phục vấn đề này bằng cách cung cấp hàm làm sạch vùng đệm stdin như sau:

**fflush(stdin);**

Hàm này sẽ làm sạch vùng đệm bàn phím, nh- vậy sẽ xóa hết hậu quả của các thao tác nhập số liệu tr- ớc một thao tác nhập số liệu mới. Ch- ơng trình trên đây có thể thêm lệnh này tr- ớc khi thực hiện lệnh scanf thứ 2 nh- sau:

```
char ca,cb;
ca=cb='z';
scanf("%c",&ca);
fflush(stdin);
scanf("%c",&cb);
printf("\n%c  %c",ca,cb);
```

Và ta không còn gặp điều bất th- ờng trên đây nữa. Sau lệnh scanf thứ nhất máy sẽ chờ để ta nhập số liệu cho lệnh thứ 2. Kết quả hiện ra trên màn hình là những ký tự ta đã nhập vào.

Có một cách khác để khử ký tự '\n' là trong lệnh scanf th- nhất thêm đặc tả `%*c`

```
scanf("%c%c",&ca);
```

Khi đó ký tự '\n' cũng đ- ọc được (không l- u vào biến nào cả) và bị loại khỏi stdin.

Tuy nhiên nh- ta sẽ thấy, không phải lệnh nhập số liệu nào cũng hoạt động nh- lệnh scanf. Cũng có lệnh loại bỏ dấu \n sau khi đọc số liệu tr- ớc đó. Thí dụ trong đoạn ch- ơng trình sau đây nếu sau khi gõ a ↵ thì máy vẫn chờ để ta nhập tiếp dữ liệu cho lệnh scanf tiếp theo.

```
char a,b;
a=b='z';
cin>>a;
cin>>b;
printf("\n%c  %c",a,b);
```

## 4.5. Nhập/xuất số liệu cho chuỗi và ký tự

Một số hàm thao tác trên ký tự và chuỗi:

```
int getc(FILE *stream);
int putc(int c, FILE *stream);
int getch(void); //Nhận ký tự trực tiếp từ bàn phím, ký tự không hiện trên màn hình
int getche(void); //Nhận ký tự trực tiếp từ bàn phím, ký tự hiện trên màn hình
int putchar(int ch); //Màn tuý thuộc vào lệnh textcolor
int getchar(void); //Nhận ký tự từ stdin
int putchar(int c); //Luôn hiện màu đen trắng
char *gets(char *s);
int puts(const char *s);
```

Các hàm **scanf**, **cin** chỉ nhập đ- ọc chuỗi không có dấu cách.

Hàm **gets** nhập đ- ọc chuỗi chứa dấu cách: Nhận dãy ký tự từ stdin cho đến khi gặp ký tự '\n'. Ký tự '\n' bị loại khỏi stdin nh- ng không đ- ọc đặt vào cuối chuỗi. Chuỗi đ- ọc bổ sung ký tự kết thúc '\0'.

Các hàm **printf**, **cout** và **puts** có thể hiện đ- ọc các chuỗi có chứa dấu cách

## 4.6. Một số hàm xử lý chuỗi (549)

`strcmp(st1,st2)>0` nghĩa là `st1>st2`, phân biệt chữ hoa chữ th- ờng

`strcmpi(st1,st2)>0` nghĩa là `st1>st2`, không phân biệt chữ hoa chữ th- ờng

`strcat(st1,st2)` ghép `st2` vào `st1`

```
char* strchr(char *s, int kt);
```

Tìm lần xuất hiện đầu tiên của kt trong s. Nếu tìm thấy trả về địa chỉ của ký tự tìm được, nếu không trả về NULL.

```
strcpy(st1,st2);
```

 thực chất là gán st1=st2;

```
strlen(st);
```

 cho độ dài chuỗi st.strlwr(st) chuyển thành chữ thường (giống tolower(ch) đối với ký tự).

```
strupr(st);
```

#### 4.7. Sự khác biệt giữa mảng và chuỗi

- Trước hết ta chạy thử đoạn chương trình sau đây

```
int a[]={1,2,3};
cout<<endl<<&a;
cout<<endl<<a;
cout<<endl<<*a;
cout<<endl;
char *b="123";
cout<<endl<<&b;
cout<<endl<<b;
cout<<endl<<*b;
```

Ta có kết quả có dạng:

```
1
0x8f90fff0
0x8f90fff0

0x8f9fffee
123
1
```

Như vậy tên mảng hay tên chuỗi biểu thị địa chỉ của phần tử đầu tiên. Vì vậy \*a hay \*b đều cho giá trị của phần tử đầu tiên. Theo cách hiểu này thì đáng lẽ ra &a và a phải như nhau, &b và b cũng như nhau. Nhưng điều này chỉ đúng với mảng. Đối với chuỗi thì cout<<b lại cho toàn bộ chuỗi. Đây là điểm khác biệt giữa chuỗi và mảng. Nếu với mảng ta chỉ có thể nhập số liệu và cho hiện từng phần tử của mảng thì nếu ta cũng đòi hỏi như vậy với chuỗi thì sẽ làm cho công việc lập trình rất rắc rối. Chuỗi có thể xem như mảng, nhưng đồng thời phải được truy xuất như một phần tử, nếu không việc lập trình sẽ rất nhiều khi rắc rối. Không thể thao tác với tên người, tên địa danh mà coi như hợp thành của nhiều phần tử, mỗi phần tử là một ký tự được. Chính vì vậy các nhà sáng tạo ngôn ngữ C đã chọn một trong 2 ký hiệu &b và b một ký hiệu dùng để thao tác toàn bộ chuỗi. Vì vậy khi dùng lệnh

```
scanf("%s",b);
```

 ta có thể nhập toàn bộ chuỗi

còn lệnh

```
printf("%s",b);
```

 sẽ in toàn bộ chuỗi.

Nếu b là mảng thì các lệnh trên đây không thực hiện được và máy sẽ báo lỗi.

- Ta có thể khởi đầu các giá trị cho chuỗi ký tự như ví dụ sau

```
char a[]="123";  
char a1[4]="123";  
char *a3="123";
```

Hoặc khai báo và dùng phép gán:

```
char *a;  
a="123";
```

Tuy nhiên nếu ta viết:

```
char a[4];  
a="123";
```

Thì máy báo lỗi. Nguyên nhân là a là mảng đã đ-ợc định sẵn kích cỡ, không thể thay đổi đ-ợc, a không phải là biến con trỏ nên không thể đ-ợc phía bên trái trong phép gán.

### **Chú ý:**

Khi ta dùng hàm strlen(st) để xác định độ dài của chuỗi ký tự thì ký tự '\0' không đ-ợc tính vào độ dài này. Thí dụ 3 biến st1,st2,st3 sau đây đều có độ dài là 3

```
char st1[]="123";  
char st2[7]="123";  
char *st="123";
```

Tuy nhiên thực ra trong bộ nhớ để l-u trữ một chuỗi dài 3 thì máy cần 4 vị trí vì một vị trí để l-u trữ ký tự '\0'. Do đó khác với thao tác với mảng, khi thao tác với chuỗi ta không nên nhập số liệu cho vị trí cuối cùng. Thí dụ nếu khai báo char st[10] thì ta không nên nhập số liệu cho vị trí 9. Điều này có thể thấy đ-ợc qua ví dụ sau:

Giả sử ta khai báo:

```
char st[5]="123";
```

thì khi đó chuỗi có độ dài 3, 3 vị trí chứa số liệu là st[0]=='1', st[1]=='2', st[2]=='3'; vị trí st[3]=='\0'. Còn nếu ta khai báo

```
char st[3]="123";
```

Thì tuy máy không báo lỗi nh-ng thực ra có một số điều không hợp lý. Nếu ta dùng hàm strlen() để kiểm tra độ dài thì thấy rằng độ dài là 4, vị trí st[3] không phải là '\0' mà là một ký tự nào đó, ký tự ở vị trí st[4] mới là '\0'. Do đó nếu ta dùng lệnh

```
while(st[i]!='\0') putchar(st[i]);
```

chẳng hạn thì sau khi máy in ra các ký tự '1', '2', '3' thì còn in tiếp một ký tự nào đó nữa rồi mới dừng.

## **4.8. Các hàm vào ra và dịch chuyển vị trí trên màn hình(97)**

Hàm gotoxy(cm,rn); sẽ đ-a con trỏ đến vị trí ở cột thứ cm và hàng thứ rn trên màn hình.

Hàm kbhit() nhận giá trị 1 nếu có phím đ-ợc bấm.

#### 4.9. Các luồng nhập xuất cin,cout

C++ cung cấp các dòng (stream đ- ọc chứa trong iostream.h) cin và cout để nhập xuất số liệu. Sau đây là vài ví dụ:

```
float a;int m;  
cin>>a;  
cin>>m;  
cout<<endl<<setw(8)<<setprecision(4)<<a;  
cout<<"\n"<<oct<<m;  
cout<<"\n"<<hex<<m;
```

## CH- ƠNG 5. CÁC TOÁN TỬ ĐIỀU KHIỂN(102)

### 5.1. Nhắc lại khái niệm câu lệnh và khối lệnh (102)

### 5.2. Toán tử if (103)

if <biểu thức> <lệnh 1>

if <biểu thức> <lệnh 1> else <lệnh 2>

Thí dụ:

if(a>b) max=a; else max=b;

### 5.3. else if (106)

Khi muốn thực hiện một trong n lệnh ta có thể sử dụng if nhh- sau

if <biểu thức 1>

<lệnh 1>

else if <biểu thức 2>

<lệnh 2>

...

else if <biểu thức n-1>

<lệnh n-1>

else

<lệnh n>

### 5.4. Toán tử switch (103)

switch<biểu thức nguyên>

{ case n1: <Các lệnh>

case n2: <Các lệnh>

...

case nk: <Các lệnh>

[default: <Các lệnh>]

}

Cách hoạt động: Nếu <biểu thức nguyên> =ni thì ch- ơng trình bắt đầu thực hiện từ case ni cho đến hết khối lệnh. Nh- vậy nếu muốn thoát khỏi khối lệnh ở đoạn giữa thì ta phải dùng lệnh break. Nếu không có giá trị ni nào trùng với <biểu thức nguyên> thì ch- ơng trình nhảy đến default, hoặc ra khỏi lệnh switch nếu không có default.

### 5.5. Toán tử goto và nhãn (111)

nhãn: x+=y;

...

goto nhãn;

### 5.6. Toán tử for (114)

for([biểu thức 1]; [biểu thức 2]; [biểu thức 3]) [lệnh]

1. Nếu có [biểu thức 1] thì xác định biểu thức này.



2. Xác định [biểu thức 2], nếu biểu thức này sai thì sang bước 4 (kết thúc lệnh for), nếu đúng thì thực hiện [lệnh].
3. Xác định [biểu thức 3], trở lại bước 2. (Thông thường sau bước này thì giá trị [biểu thức 2] thay đổi).
4. Kết thúc lệnh for

Nh- vậy nếu [biểu thức 2] luôn luôn đúng thì vòng lặp sẽ vô hạn. Trong trường hợp [biểu thức 2] vắng mặt chẳng hạn, thì vòng lặp là vô hạn nếu trong [lệnh] không có lệnh break.

### 5.6. Toán tử while (123)

while<biểu thức>

<lệnh>

Nh- vậy nếu <biểu thức> sai thì <lệnh> không được thực hiện

### 5.7. do ... while (126)

do

<lệnh>

while<biểu thức>

Dù biểu thức sai hay đúng thì <lệnh> ít nhất được thực hiện một lần.

### 5.8. Lệnh break và lệnh continue (129)

## CH- ƠNG 6. HÀM VÀ CẤU TRÚC CH- ƠNG TRÌNH(450 Bài tập,71)

### 6.1. Khai báo và định nghĩa hàm(71)

T- ơng tự nh- một biến, một hàm phải khai báo tr- ớc khi nó đ- ợc gọi. Một khai báo hàm có 3 phần: Kiểu cho ra, tên và danh sách tham số.

Hàm có vai trò ngang nhau, C++ không cho phép xây dựng một hàm bên trong hàm khác.

Ví dụ: Khai báo hàm:

```
double power(double x, int n);
```

Trong đó double là kiểu cho ra, power là tên của hàm và (double x,int n) là danh sách các tham số. Hàm này có 2 tham số là x và n. Các tham số này còn đ- ợc gọi là các tham số hình thức. Danh sách các tham số có thể rỗng, nh- ng các dấu ngoặc đơn vẫn cần thiết. Một khai báo hàm còn đ- ợc gọi là các mẫu hàm (prototype). Phần này chứa các thông tin tối thiểu mà bộ biên dịch cần để biên dịch các lệnh gọi đến hàm. Vì vậy trong phần khai báo có thể bỏ bớt tên các tham số chẳng hạn nh- :

```
double max(double,double);
```

Tuy nhiên nếu biến là mảng thì không thể bỏ tên biến đ- ợc. Ta chỉ có thể viết một cách đơn giản hơn, ví dụ

```
int x[],float y[][10]
```

chẳng hạn. L- u ý là với mảng 2 chiều phải có thông tin về số cột.

**Phần khai báo phải ở vị trí tr- ớc lời gọi hàm** trong ch- ơng trình. Tuy nhiên nh- ta có thể thấy, phần khai báo ch- a đủ để xác định một hàm. Hàm còn có phần định nghĩa nằm ở vị trí bất kỳ trong ch- ơng trình.

Ví dụ: Định nghĩa hàm:

```
double max(double x,double y)
{
    return (x>y?x:y);
}
```

#### Chú ý:

- Nếu định nghĩa hàm nằm tr- ớc lời gọi nó thì không cần khai báo mẫu nữa.
- Trong thân hàm có thể khai báo các biến cục bộ. *Các biến này không đ- ợc trùng tên với các tham số.* Trong thân hàm ta có thể sử dụng các tham số mà không khai báo lại. Nếu ta khai báo lại thì ch- ơng trình sẽ báo lỗi.
- Trong phần định nghĩa hàm không đ- ợc bỏ tên các tham số nh- trong phần khai báo.
- Nếu hàm có kiểu cho ra thì cuối thân hàm phải có lệnh return <biểu thức>; trong đó <biểu thức> có cùng kiểu giá trị với hàm. Trong thân hàm có thể có một hoặc nhiều lệnh return <biểu thức>; Khi ch- ơng trình gặp lệnh này thì giá trị hàm đ- ợc trả về và ch- ơng trình thoát khỏi hàm.
- Tên hàm cũng là một identifier, do đó không đ- ợc đặt tên hàm trùng với tên biến hoặc ng- ợc ng- ợc lại.
- Khi ta khai báo tham số là mảng một chiều hay con trỏ thì khi gọi có thể gọi tham số thực là con trỏ hoặc mảng. Thí dụ khai báo char st[]; và char \*st; là hoàn toàn t- ơng đ- ơng. Tuy

nhien với mảng 2 chiều thì không còn đúng nữa. Thí dụ đoạn ch- ơng trình sau đây là **không hợp lệ**:

```
#include "define_h.cpp"
#include <graphics.h>
void v(int **mm)
{cout<<endl<<mm[0][0];
}
void main()
{clrscr();
int m[2][3]={1,2,3,4};
v(m);
getch();
}
```

Tuy nhiên cách **khai báo và gọi nh-** sau lại đúng :

```
#include "define_h.cpp"
void v(int **a,int m,int n)
{for(int i=0;i<m;i++)
{cout<<endl;
for(int j=0;j<n;j++) cout<<a[i][j]<<" ";
}
};
void main(void)
{clrscr();
int m=2,n=3,i,j;
int **b;
b=new int* [2];
for(i=0;i<2;i++) b[i]=new int [3];
for(i=0;i<m;i++)
for(j=0;j<n;j++) b[i][j]=i*3+j;
v(b,2,3);
getch();
}
```

## 6.2. Các hàm void (73)

void tiếng Anh có nghĩa là trống rỗng, khuyết. Hàm kiểu void là một hàm có kiểu cho ra là void. Điều này có nghĩa là nó không cho ra giá trị, giống nh- procedure trong Pascal. Trong hàm có thể có hoặc không có lệnh return; nh- ng lệnh này nếu có thì chỉ có ở dạng đơn giản là return; khi ch- ơng trình gặp lệnh này thì ch- ơng trình thoát khỏi hàm. Nếu không có lệnh return thì ch- ơng trình chạy đến hết thân hàm rồi mới thoát khỏi hàm.

Ví dụ:

Bây giờ ta xét một ví dụ là in ra ngày trong tuần từ thứ 2 đến thứ 7 bằng tiếng Anh khi cho ngày bằng số, thí dụ 2 là thứ 2, 3 là thứ 3, . . . và viết hàm bằng 2 cách: có kiểu cho ra và kiểu cho ra là void.

Cách 1:

```
char* weekday(int n)
{switch(n)
{case 2: return "Monday";
case 3: return "Tuesday";
case 4: return "Wednesday";
case 5: return "Thursday";
case 6: return "Friday";
```

```
    case 7: return "Saturday";  
    }  
    return '\0';  
}
```

Trong chương trình main() ta sẽ dùng lệnh `cout<<weekday(n)` để in ra các ngày.

Chú ý: Ta có thể viết `return "\0"` hay `return '\0'`. Còn nếu viết `return 'a'` chẳng hạn thì máy báo lỗi. Máy chỉ chấp nhận với trường hợp ký tự `\0` là ký tự đặc biệt.

Cách 2:

```
void weekday(int n)  
{  
    switch(n)  
    {  
        case 2: cout<<"Monday";break;  
        case 3: cout<<"Tuesday"; break;  
        case 4: cout<<"Wednesday"; break;  
        case 5: cout<<"Thursday"; break;  
        case 6: cout<<"Friday"; break;  
        case 7: cout<<"Saturday"; break;  
    }  
    return;  
}
```

### 6.3. Lời gọi hàm

Khi khai báo hàm các tham số được gọi là tham số hình thức. Khi sử dụng hàm các tham số được thay bằng các tham số có kiểu tương ứng. Khi gọi hàm ta không cần chỉ rõ các tham số có kiểu gì vì kiểu của chúng đã được khai báo trước khi gọi. Chỉ có tên của các tham số được thay vào các vị trí tương ứng của các tham số hình thức. Thí dụ để xác định max trong 2 số a,b ta gọi

```
t=max(a,b);
```

### 6.4. Sử dụng hàm `assert()` để kiểm tra điều kiện trước (76)

Hầu hết các hàm không thực thi tốt trên tất cả các giá trị của các tham số. Thí dụ hàm `double power(double x,int n)` mà chúng ta sẽ xét tới sau đây chỉ có nghĩa với  $x > 0$ . Đối với các hàm có trả giá trị khác với kiểu `void` thì đôi khi rất khó để tìm một kiểu trả tương ứng với các trường hợp không thích hợp. Trong các trường hợp này ta có thể dùng hàm `assert()` để xác định nghĩa trong `assert.h` để kiểm tra trước các điều kiện. Khi điều kiện đó không thỏa mãn thì chương trình dừng thực hiện hàm và kèm lời thông báo.

Ví dụ:

```
double power(double x,int n)  
{  
    assert(x>0);  
    double y=1.0;int m=abs(n);//Nếu là double thì dùng fabs()  
    for(int i=0;i<m;i++) y*=x;  
    if(n<0) y=1/y;  
    return y;  
}
```

### 6.5. Các đối số mặc định (77)

Nếu ta sửa lại hàm như sau

```
double power(double x=1.0,int n=2)
{ assert(x>0);
  double y=1.0;int m=abs(n);//Nếu là double thì dùng fabs()
  for(int i=0;i<m;i++) y*=x;
  if(n<0) y=1/y;
  return y;
}
```

Thì ta có thể gọi hàm power() mà không có tham số. Khi đó tham số đ-ợc hiểu là các tham số mặc định. Nh- thí dụ trên đây ta có:

```
cout<<power();
```

sẽ cho kết quả là 1.

## 6.6. Truyền tham số cho hàm (78)

### 6.6.1. Truyền bằng tham trị(value parameter):

(Chú ý rằng tên mảng là con trỏ do đó hàm có tham số là mảng luôn đ-ợc tự động truyền theo địa chỉ. Thậm chí nếu ta viết void vv(double &a[2][3]) chẳng hạn thì máy sẽ báo lỗi. Tuy nhiên nếu ta đã định nghĩa tr-ớc typedef double kmatran[2][3]; thì lại có thể viết void vv(kmatran &a);)

Ta xét đoạn ch-ơng trình sau:

```
void f1(int x)
{ x*=3;
  cout<<x<<endl;
}
void main()
{ int a=1;
  cout<<a<<endl;
  f1(a);
  cout<<a<<endl;
}
```

Kết quả trên màn hình là:

```
1
3
1
```

Mặc dầu trong thân hàm ta thay đổi giá trị của biến x nh-ng khi ra khỏi thì ch-ơng trình x lại lấy giá trị cũ. Nguyên nhân là vì hàm đã đ-ợc định nghĩa theo kiểu truyền tham trị, nên khi một tham số đ-ợc gọi thì không phải bản thân tham số đó mà bản sao của nó đ-ợc thay vào hàm. Khi ra khỏi hàm biến x lại đ-ợc hiểu là biến tr-ớc khi gọi hàm.

### **Truyền bằng tham trị hằng** (const value parameter):

Nếu hàm f1 đ-ợc sửa lại nh- sau:

```
void f1(const int x)
{ x*=3;
  cout<<x<<endl;
}
```

Thì máy sẽ báo lỗi. Vì khi khai báo kiểu tham trị hằng thì trong thân hàm không đ- ợc phép thay đổi giá trị của hàm.

### 6.6.2. Truyền theo địa chỉ

Ta xét đoạn ch- ơng trình sau:

```
void f1(int *px)
{ *px*=3;
  cout<<*px<<endl;
}
void main()
{ int a=1;
  cout<<a<<endl;
  f1(&a);
  cout<<a<<endl;
}
```

Kết quả trên màn hình là:

```
1
3
3
```

Nghĩa là sau khi ra khỏi ch- ơng trình thì x lấy giá trị đã thay đổi trong ch- ơng trình con. Đây là cách truyền tham số theo địa chỉ. Qua cách truyền này thì ch- ơng trình con làm việc trực tiếp với giá trị mà tham số địa chỉ trỏ tới, tức là giá trị thực sự của biến. Do vậy nếu trong hàm giá trị biến bị thay đổi thì khi thoát khỏi hàm những sự thay đổi đó vẫn đ- ợc giữ lại. Khi truyền tham số theo địa chỉ ta cần chú ý tới những điểm sau:

- Mảng chỉ có thể truyền cho hàm qua địa chỉ, do đó đối với dữ liệu kiểu mảng thì mọi sự thay đổi trong hàm vẫn còn tác dụng khi ra khỏi hàm.
- Có thể dùng 2 cách khai báo biến <kiểu dữ liệu>\* <tên biến con trỏ> hoặc <kiểu dữ liệu>\* <tên biến con trỏ> (thí dụ int\* pi hoặc int \*pi, trong đó pi là con trỏ kiểu int). Cả 2 cách này hoàn toàn t- ơng đ- ơng. Cách thứ nhất th- ờng đ- ợc dùng khi khai báo nguyên mẫu hàm, vì khi đó ta chỉ khai báo kiểu của biến, ch- a cần khai báo tên.
- Con trỏ không cần phải khởi tạo khi khai báo nh- ng phải khởi tạo tr- ớc khi sử dụng.
- Không đ- ợc gán giá trị khác kiểu cho con trỏ. Thí dụ trong hàm trên đây ta không thể gọi f1(2); hoặc f1(&(x+1));

### 6.6.3. Truyền bằng tham chiếu(reference parameter):

(**Chú ý** rằng tên mảng là con trỏ do đó hàm có tham số là mảng luôn đ- ọc tự động truyền theo tham chiếu. Thậm chí nếu ta viết `void vv(double &a[2][3])` chẳng hạn thì máy sẽ báo lỗi. Tuy nhiên nếu ta đã định nghĩa `tr- ớc typedef double kmatran[2][3];` thì lại có thể viết `void vv(kmatran &a);`)

Trong C++ ng- ời ta còn đ- a thêm khái niệm biến tham chiếu. Có thể xem biến tham chiếu là một b- ớc tiến của con trỏ. Ta hãy xét đoạn ch- ơng trình sau:

```
int i=10;
int& j=i;
j=20;
cout<<i;
```

Đoạn ch- ơng trình trên đây sẽ cho kết quả là

20

nghĩa là giá trị `i` đã bị thay đổi mặc dầu trong ch- ơng trình không có lệnh nào làm thay đổi giá trị `i`. Nguyên do là vì biến `j` cũng chiếm vùng bộ nhớ nh- biến `i`. Do đó mọi sự thay đổi của `i` đều t- ơng đ- ơng vớ thay đổi trên `j` và ng- ợc lại. Vậy ta có thể hình dung một biến tham chiếu giống nh- bí danh của biến khác. Ta có mấy điều chú ý sau đây về biến tham chiếu:

- Biến tham chiếu cũng chứa địa chỉ của một hực thể nh- các biến khác.
- Biến tham chiếu cần phải đ- ọc khởi tạo khi khai báo. Thí dụ nếu ta khai báo `int& i;` thì máy sẽ báo lỗi. Điều này cũng dễ hiểu, vì biến tham chiếu là biến "ăn theo", do đó phải đ- ọc khởi tạo khi khai báo. Tốt nhất là ta khởi tạo biến tham chiếu theo một biến khác đã có. Còn nếu ta khởi tạo bằng một hằng, thí dụ `int& m=2;` thì máy không báo lỗi nh- ng có dòng cảnh báo `Temporary used to initialize 'm'`.
- Không cấp phát bộ nhớ động cho biến tham chiếu.
- Không dùng các phép toán nh- tăng hoặc giảm địa chỉ đối với biến tham chiếu.

Mặc dù bản chất của biến tham chiếu là con trỏ, nh- ng ta có thể sử dụng tên biến tham chiếu để truy nhập đến giá trị của nó. Đây là điểm khác biệt giữa biến thuộc kiểu tham chiếu và biến kiểu con trỏ. Nếu trong đoạn ch- ơng trình trên đây ta sửa lại là

```
const int i=10;
int& j=i;
j=20;
cout<<i;
```

Thì kết quả là

10

Nghĩa là giá trị của `i` không bị thay đổi. Điều này có vẻ nh- mâu thuẫn với điều ta nói trên đây là biến `j` thực ra là bí danh của `i` mà thôi. Chúng tôi sẽ không đi sâu thêm vấ vấn đề này mà chỉ muốn nhắc nhở độc giả một điều là ta phải viết các dòng lệnh sao cho chúng có nghĩa. Thí dụ nếu ta đã không muốn thay đổi giá trị của `i` thì không nên gán bí danh cho nó nữa.

#### Truyền tham số theo kiểu tham chiếu:

Trở lại vấn đề truyền tham số cho hàm, nếu trong ch- ơng trình trên ta sửa lại hàm `f1` nh- sau:

```
void f1(int &x)
{ x*=3;
  cout<<x<<endl;
}
void main()
{ int a=1;
  cout<<a<<endl;
  f1(a);
  cout<<a<<endl;
}
```

Thì kết quả trên màn hình sẽ là:

```
1
3
3
```

Nh- vậy giá trị `x` đã thực sự bị thay đổi khi ra khỏi hàm. Lý do là vì khi có lời gọi hàm thì một biến tham chiếu đến biến `x` tham gia vào các tính toán trong thân hàm, và nh- vậy giá trị gốc của `x` đã bị thay đổi.

Nếu ta khai báo các tham số hàm theo kiểu tham chiếu thì khi gọi không cần gọi theo địa chỉ nh- trong hợp truyền theo địa chỉ mà chỉ cần viết tên biến nh- trong tr- ờng hợp truyền theo giá trị.

### 6.7. Hàm chồng(Overloaded function, 81)

Trong khi sắp xếp các dãy số ta th- ờng xuyên phải đổi chỗ 2 phần tử. Để khỏi phải lặp lại một số lệnh giống nhau ta định nghĩa hàm swap nh- sau:

```
void swap(int &x, int &y)
{ int tmp=x;
  x=y;
  y=tmp;
}
```

Tuy nhiên hàm này chỉ có thể đổi chỗ cho 2 số nguyên. Ta biết rằng C không cho phép định nghĩa 2 hàm cùng tên, và nếu muốn viết hàm đổi chỗ 2 số thực ta lại phải viết hàm khác với tên khác. Tuy nhiên việc định nghĩa này lại hợp lệ trong C++. Ta có thể định nghĩa thêm 2 hàm swap cho số thực float và double nh- sau:

```
void swap(float &x, float &y)
{ float tmp=x;
  x=y;
  y=tmp;
}
```

```
void swap(double &x, double &y)
{ double tmp=x;
  x=y;
  y=tmp;
}
```



Khi ta gọi hàm bằng lệnh `swap(a,b)`; chẳng hạn thì C++ sẽ xem kiểu của `a`, `b` mà chọn kiểu hàm thích hợp.

C++ sử dụng hệ thống trong đó các hàm đ-ợc nhận biết duy nhất bằng tên và loại tham số của chúng. Trong quá trình biên dịch tên hàm đ-ợc biến đổi thành một tên mới thông qua quá trình gọi là "máý các tên" (name-mangling). Trong quá trình này, tên của hàm và danh sách các kiểu tham số đ-ợc nhập chung để thành một tên duy nhất. Thí dụ có 2 hàm `f(int,int)`; và `f(float,float)`; sẽ đ-ợc bộ biên dịch đổi tên thành hai tên khác biệt nhau

```
_f_int_int  
_f_float_float
```

Nh- vật trong tên mới không có sự tham gia của kiểu giá trị hàm. ***Do đó bộ biên dịch sẽ không có khả năng phân biệt hai hàm có cùng tên danh sách tham số nh- ng có kiểu trả về khác nhau.***

## 6.8. Các mẫu (template 250)

Trên đây ta phải định nghĩa 3 hàm cho thủ tục trao đổi giá trị 2 biến. C++ có một ph-ơng pháp tốt hơn để thay đổi kiểu phần tử mà không cần ghi rõ kiểu phần tử khi khai báo hay định nghĩa. Ph-ơng pháp này đ-ợc thực hiện với mẫu.

Mẫu hàm:

Ta có thể định nghĩa hàm `swap` với mẫu nh- sau:

```
template <class T>  
void swap(T& x, T& y)  
{ T tmp=x;  
  x=y;  
  y=tmp;  
}
```

Khi ta gọi hàm `swap`, thí dụ `t=swap(a,b)`; thì kiểu của các tham số sẽ đ-ợc bộ biên dịch thay vào hàm. thí dụ nếu kiểu của `a,b` là `float` thì thực chất bộ biên dịch tạo định nghĩa hàm sau đây:

```
void swap(float &x, float &y)  
{ float tmp=x;  
  x=y;  
  y=tmp;  
}
```

Chú ý:

mẫu `template <class T>`  
phải đứng tr-ớc khai báo và cả phần định nghĩa hàm.

## 6.9. Hàm với các tham số là mảng

Trong Pascal mảng không thể xuất hiện với khai báo trực tiếp trong phần khai báo các tham số hàm, mà ta phải định nghĩa bằng lệnh `type` tr-ớc.

Trong C thì điều đó có thể đ-ợc. Việc dùng hàm có tham số là các mảng có thể đ-ợc minh họa bằng 2 ví dụ sau đây:

Mảng 1 chiều:

```
#include "define_h.cpp"
void f(int x[],int);
//=====
void main()
{clrscr();
 int n=2;
 int x[4]={1,2,3,4};
 f(x,4);
 getch();
}
//=====
void f(int x[],int n)
{for(int i=0;i<n;i++)
 cout<<x[i]<<endl;
 return;
}
```

Mảng 2 chiều:

```
#include "define h.cpp"
void f(int x[][2],int);
//=====
void main()
{clrscr();
 int n=2;
 int x[2][2]={1,2,3,4};
 f(x,n);
 getch();
}
//=====
void f(int x[][2],int n)
{for(int i=0;i<n;i++)
 for(int j=0;j<n;j++)
 cout<<x[i][j]<<endl;
 return;
}
```

## NHẬP MÔN TIN HỌC - CÂU HỎI VÀ BÀI TẬP:

### Một số khái niệm cơ bản của tin học

#### I. Thông tin và xử lý thông tin

1. Có thể hiểu thông tin là gì? Yếu tố gì đặc tr- ng cho thông tin? Hãy nêu những đơn vị đo thông tin.
2. Vì sao ng- ời ta chọn BIT làm đơn vị đo thông tin?
3. Mã hóa thông tin là gì và vì sao phải mã hoá thông tin? Hãy kể tên bảng mã hay dùng.
4. Hãy vẽ sơ đồ xử lý thông tin nói chung và trên máy tính điện tử.
5. Tin học là gì?

#### II. Cấu trúc tổng quát của các hệ thống máy tính

1. Hãy nêu 2 nguyên lý cơ bản của Von-Newman về thiết kế logic của máy tính điện tử.
2. Giải thích các khái niệm phần cứng và phần mềm trong tin học.
3. Vẽ và giải thích sơ đồ cấu trúc tổng quát của máy tính cá nhân.
4. Nêu các thành phần cơ bản của bảng mạch chính (CPU).
5. ROM, RAM là gì?
6. Có thể phân loại các phần mềm nh- thế nào?

#### III. Biểu diễn thông tin trong máy tính điện tử

1. Vì sao phải dùng các hệ đếm phi thập phân trong tin học? Hãy kể ra một số hệ đếm hay dùng và quy tắc chuyển đổi các số giữa 2 hệ đếm bất kỳ.
2. Chuyển đổi số thập phân sau sang số nhị phân (lấy 5 chữ số sau dấu chấm phân nếu gặp tr- ờng hợp phân phân quá dài):  
97, 0.69, 89.027, 132.63
3. Chuyển đổi các số nhị phân sau sang số thập phân:  
1011, .11, 1001.001
4. Chuyển các số nhị phân sau sang số bát phân  
110010, 101100111
5. Chuyển các số nhị phân sau sang số thập lục phân:  
11011001, 110010100001
6. Hãy tính giá trị biểu thức sau:  
((0 and 1) or 1) xor (1 and (not 0))
7. Bảng mã gì hay đ- ọc dùng nhất trong tin học?

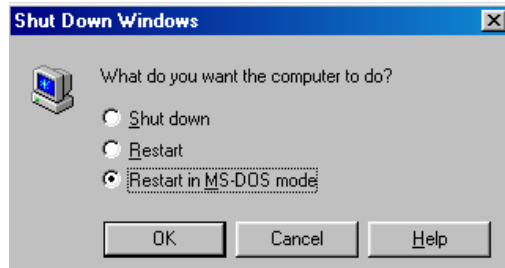
#### IV. Mạng máy tính

1. Mạng máy tính là gì? Hãy nói một vài điều về lợi ích của mạng.
2. Mạng bao gồm 2 yếu tố cơ bản gì? Định nghĩa 2 yếu tố cơ bản đó.
3. Hãy nêu cách phân loại mạng theo khoảng cách địa lý và theo TOPO mạng.
4. Hãy giải thích về hệ điều hành mạng và nêu một vài HĐHM hay dùng.
5. Internet là gì? Nêu sự khác biệt giữa Internet và Intranet.
6. Nêu một số dịch vụ cơ bản của Internet.

## Hệ điều hành và các phần mềm hỗ trợ

Trong phần này bạn sẽ thực hành trên màn hình DOS và trên Windows. Sau khi bạn khởi động máy tính bạn sẽ thấy màn hình Windows xuất hiện. Bạn sẽ thao tác nh- sau để trở về màn hình DOS:

Nhấp đúp chuột vào nút Start, chọn Shut Down... Cửa sổ sau hiện lên



Bạn chọn Restart in MS-DOS mode rồi nhấn nút OK.

Từ màn hình DOS bạn gõ EXIT {ENTER} để trở về Windows.

### I. Hệ điều hành

1. Nêu định nghĩa HĐH
2. Cách phân loại HĐH và một số HĐH hay dùng.

### II. Hệ điều hành MS-DOS

1. Nêu định nghĩa tệp tin, th- mục, đ- ờng dẫn và th- mục hiện thời.
2. Nêu các module của MS-DOS và chức năng. Hãy giải thích sơ l- ọc cách hoạt động của các module từ khi bật máy cho đến khi hiện dấu nhắc hệ thống.
3. Thế nào là lệnh nội trú và lệnh ngoại trú của DOS?
4. Hãy viết cú pháp và giải thích chức năng của các lệnh nội trú sau:  
Chuyển ổ đĩa, DIR, MD, CD, RD, COPY CON, TYPE, REN, COPY, DEL, CLS, PATH, PROMPT
5. Hãy viết cú pháp và giải thích chức năng của các lệnh ngoại trú sau:  
FORMAT, SYS, SCANDISK, DEFRAG, UNDELETE.
6. Tạo th- mục C:\HOC\_DOS chuyển con trỏ DOS vào th- mục này, copy một số tệp vào rồi thực hành các lệnh:  
MD, CD, DIR, RD, TYPE, REN, COPY, DEL, CLS, UNDELETE.
7. Xóa các cây th- mục bạn tạo ra trên đây (MON\_HOC,HOC\_DOS).
8. Hãy nêu tác dụng và cách tạo ra đĩa mềm hệ thống.

### III. Phần mềm hỗ trợ NC

1. Bạn hãy tạo một th- mục cho mình thí dụ C:\HOC\_NC sau đó tạo một số th- mục con trong đó, copy các tệp ở một số th- mục khác, chủ yếu là các tệp nhỏ để thực hành các lệnh cơ bản của NC nh- sau:  
Tạo th- mục, chuyển thanh chọn đến th- mục khác, soạn thảo tệp mới, soạn thảo tệp đã có, xem nội dung tệp không sửa, copy, xóa, đổi tên tệp hoặc th- mục, dịch chuyển tệp.
2. Cũng trong th- mục bạn tạo ra trên đây bạn thử xem các th- mục có bao nhiêu tệp, mỗi tệp chứa bao nhiêu Kb. Thử nén và gỡ nén một số tệp nào đó.
3. Dùng NC để làm bài tập số 6 trong phần II. Hệ điều hành MS-DOS trên đây.
4. Xóa các cây th- mục C:\HOC\_NC.

#### **IV. Hệ điều hành Windows 98**

1. So sánh khái niệm Directory (th- mục) trong MS-DOS và Folder trong Windows 98.
2. Bạn hãy tạo một th- mục cho mình thí dụ C:\HOC\_WIN sau đó tạo một số th- mục con trong đó. Copy các tệp ở một số th- mục khác, chủ yếu là các tệp nhỏ để thực hành các lệnh cơ bản của Windows nh- sau:  
Copy , xóa, đổi tên tệp hoặc th- mục (folder), dịch chuyển tệp.
3. Cũng trong th- mục C:\HOC\_WIN bạn thử xem các th- mục có bao nhiêu tệp, mỗi tệp chứa bao nhiêu Kb.
4. Dùng WINDOWS để làm bài tập số 6 trong phần **II. Hệ điều hành MS-DOS**. Bạn có thể copy một vài tệp nào đó vào các th- mục.
5. Bạn hãy tìm xem tệp TURBO.EXE nằm ở đâu, sau đó tạo một shortcut để khởi động Turbo C.
6. Bạn hãy chạy ch- ơng trình NC bằng nhiều cách.

## Microsoft Word

Bạn hãy tạo một th- mục (folder) có tên là C:\HOCWORD, bạn sẽ đặt tên tệp cho các bài tập sau đây và cất vào th- mục này.

1. Hãy gõ văn bản sau:

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  
Độc lập - Tự do - Hạnh phúc  
=====

Hà Nội, ngày 22 tháng 07 năm 2000

### HỢP ĐỒNG THI CÔNG XÂY DỰNG NHÀ Ở GIA ĐÌNH

Chúng tôi gồm có:

Đại diện phía chủ nhà (bên A):

Địa chỉ:.....

Điện thoại:.....

Đại diện bên thi công (bên B):

Địa chỉ:.....

Điện thoại: .....

Sau khi trao đổi bàn bạc, hai bên đã thống nhất các vấn đề sau:

**1. Diện tích khu đất:**

**2. Diện tích xây dựng:**

**3. Tầng cao:**

**4. Trách nhiệm của mỗi bên:**

**Bên A:**

- Đảm bảo mặt bằng thi công.

- Cung cấp đầy đủ thiết kế kiến trúc, kết cấu của ngôi nhà cho bên B.

**Bên B:**

- Tự lo toàn bộ trang thiết bị, máy móc dùng để xây dựng và các thiết bị bảo hiểm cá nhân.

- Phải đảm bảo tuyệt đối an toàn trong quá trình thi công và hoàn toàn chịu trách nhiệm nếu có rủi ro.

**5. Thời gian thi công:** Bắt đầu từ ngày tháng năm đến ngày tháng năm

**6. Bảo hành:** Công trình đ- ọc bảo hành ít nhất 6 tháng kể từ khi bàn giao. Nếu có sự cố xảy ra trong thời gian bảo hành, bên B phải có trách nhiệm sửa chữa.

Hai bên cam kết sẽ thực hiện đúng các điều khoản ghi trong hợp đồng. Nếu bên nào không thực hiện đúng các điều kiện hợp đồng đặt ra thì bên kia có thể hủy hợp đồng sau khi đã có thông báo tr- ớc và thanh toán xong phần công việc đã thực hiện.

Hợp đồng đ- ọc lập thành 02 bản và có hiệu lực từ ngày ký.

**Đại diện bên B**

**Đại diện bên A**

2. Hãy tạo bảng sau, l- u ý là ô tổng số tiền dùng công thức để tính:

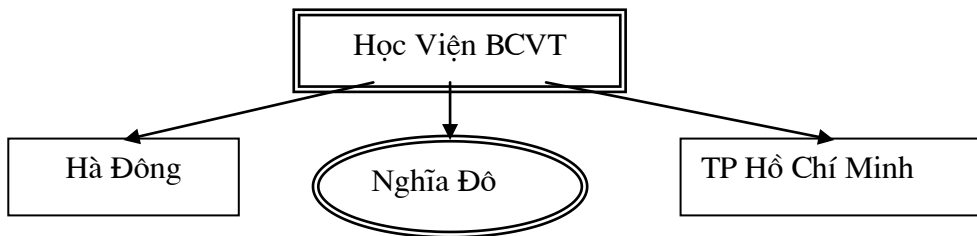
## BẢNG THANH TOÁN CỦA GỖ

Đơn vị tiền: 1000 đồng

Loại	Số m2	Đơn giá	Thành tiền
Lim đặc	6	600	3600
Dổi	7	390	2730
Kính lim	2	560	1120
Kính dổi	11	380	4218
Chóp dổi	9	400	3600
Tổng số tiền			15 268

Vậy bên A phải thanh toán cho bên B số tiền là 15 268  
000 đồng chẵn

3. Hãy vẽ sơ đồ sau:



4. Vẽ hình sau:



#### 4. Hãy nhập đoạn văn bản sau và cất thành tệp

Với mỗi ma trận vuông  $A$  cấp  $n$

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \quad (2.2)$$

tồn tại một số thực đ-ợc gọi là **định thức** của ma trận  $A$ , ký hiệu là  $\det A$ , đ-ợc xác định bởi công thức:

$$\det A = \sum_{\alpha} s(i_1, i_2, \dots, i_n) a_{1i_1} a_{2i_2} \dots a_{ni_n} \quad (2.3a)$$

với  $\alpha = (i_1, i_2, \dots, i_n)$  chạy trong tập tất cả các hoán vị của tập  $\{1, 2, \dots, n\}$ , và

$$s(i_1, i_2, \dots, i_n) = \begin{cases} 1 & \text{nếu } \alpha \text{ là hoán vị chẵn} \\ -1 & \text{nếu } \alpha \text{ là hoán vị lẻ} \end{cases}$$

#### 5. Gõ đoạn văn bản sau:

Nếu ký hiệu  $a_{kj}^{(i)}$  là phần tử ma trận ở b-ớc khử thứ  $i$ , thì thực ra chúng ta có ma trận với sự biến đổi từng b-ớc đ-ợc thể hiện nh- sau:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & \cdot & \dots & a_{1n} \\ 0 & a_{22}^{(1)} & \dots & \cdot & \dots & a_{2n}^{(1)} \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & a_{ii}^{(i-1)} & \dots & a_{in}^{(i-1)} \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \cdot & \dots & a_{nn}^{(n-1)} \end{bmatrix}$$

#### 6. Hãy nhập đoạn văn bản sau:

Ng-ời ta th-ờng dùng ba chuẩn sau:

Chuẩn cột:  $\|A\|_1 = \max_j \sum_{i=1}^m |a_{ij}|$

Chuẩn Öclit:  $\|A\|_2 = (\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2)^{1/2}$

Chuẩn hàng:  $\|A\|_\infty = \max_i \sum_{j=1}^n |a_{ij}|$

#### 7. Gõ đoạn văn bản:

Viết lại hàm tính  $\sin(x)$  và so sánh kết quả của hàm tự tạo với giá trị hàm chuẩn:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \dots$$

Với độ chính xác  $|\frac{x^{2n+1}}{(2n+1)!}| < \text{epsilon} = 1.0\text{E-}04$



## Microsoft Excel

Bạn hãy tạo một th- mục (folder) có tên là C:\HOCEXCEL, bạn sẽ đặt tên tệp cho các bài tập sau đây và cất vào th- mục này.

1. Hãy lập bảng sau và l- u với tên: TKSP.XLS

CÔNG TY ÔTÔ I DÔNG - TỔNG KẾT SẢN PHẨM NĂM 1999				
--	--	--	--	--

	Quý I	Quý II	Quý III	Quý IV
Tổ 1	60	50	60	65
Tổ 2	30	40	70	50
Tổ 3	25	30	45	45

- Lập biểu đồ trên bảng tính để biểu diễn số l- ợng sản phẩm trong quý I của từng tổ dạng hình bánh (Pie) không gian ba chiều
- Vẽ đồ thị trong không gian 2 chiều 3 đ- ờng biểu diễn theo từng quý của 3 tổ.
- Vẽ biểu đồ hình cột trong không gian 3 chiều với hoành độ là các quý, tung độ là số sản phẩm.

2. Hãy nhập bảng tính sau:

Ngày	Đơn vị	Loại	Số l- ợng	Đơn giá	Thành tiền
12/03/99	MISA	Đ- ờng	100	6000	
16/03/99	MISA	Đ- ờng	250	6000	
14/03/99	MISA	Đ- ờng	200	6000	
15/03/99	MISA	Đ- ờng	120	6000	
17/03/99	IMEX	Cà phê	200	6000	
22/03/99	IMEX	Cà phê	350	6000	
12/03/99	LOTUS	Gạo	120	4000	
12/03/99	LOTUS	Gạo	150	4000	
15/03/99	LOTUS	Gạo	240	4000	
21/03/99	LOTUS	Gạo	120	4000	
23/03/99	MISA	Sữa	150	5500	
22/03/99	IMEX	Sữa	100	5500	
15/03/99	IMEX	Sữa	600	5500	

- Lập công thức tính toán cho cột thành tiền.
- Sắp xếp CSDL theo khóa là cột cần thực hiện tổng hợp thống kê, thí dụ mặt hàng.
- Hãy lọc ra các bản ghi có số l- ợng từ 150 đến 200 và đơn giá nhỏ hơn 5000.
- Tạo dòng tổng con (Subtotal) cho tr- ờng mặt hàng và tính tổng con cho các tr- ờng Số l- ợng và Thành tiền.

## Ngôn ngữ lập trình C

- Bài tập có dấu \* bạn có thể bỏ qua trong một vài lần thực hành đầu tiên.
1. Ba số d-ong a,b,c là độ dài các cạnh của một tam giác nếu tổng của 2 số bất kỳ lớn hơn số còn lại. Lập ch-ong tình nhập 3 số a,b,c và kiểm tra xem chúng có thể là độ dài của các cạnh của một tam giác hay không. Có thông báo cho từng tr-ờng hợp.
  2. a. Đọc một số nguyên d-ong, sau đó viết ra màn hình số đó d-ới dạng nhị phân.  
b. Đọc một số thực <1, sau đó viết ra màn hình số đó d-ới dạng nhị phân.  
c. Đọc một số thực bất kỳ, sau đó viết ra màn hình số đó d-ới dạng nhị phân.
  3. Hãy nhập 4 số thực a,b,c,d. Tìm giá trị lớn nhất của chúng và gán giá trị lớn nhất đó cho biến MAX.
  4. Cho  $n < 100$  và  $x_1, x_2, \dots, x_n$  là dãy số thực. Tìm  $\text{Max}(x_1, x_2, \dots, x_n)$ .
  5. Sắp xếp dãy trên theo thứ tự tăng dần.
  6. Giải và biện luận hệ ph-ong trình tuyến tính 2 ẩn:  
 $ax+by=p$   
 $cx+dy=q$
  7. Tính giá trị của đa thức  $P(x)=a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$   
Theo cách tính của Horner để đạt đ-ợc tốc độ tính nhanh:  
$$P(x)=(((a_n x + a_{n-1})x + a_{n-2} \dots + a_1)x + a_0$$
  8. Thời gian làm một việc nào đó là X giây. Hãy đọc X từ bàn phím và viết ch-ong trình chuyển sang bao nhiêu giờ, phút, giây.
  9. Tính N! theo định nghĩa sau  
$$N! = 1 \quad \text{nếu } N=0$$
$$= 1*2*\dots*N \quad \text{nếu } N>0$$
  10. Tìm tất cả các số có 3 chữ số  $\overline{abc}$  sao cho tổng lập ph-ong của các chữ số thì bằng chính số đó.
  - 11\*. Sàng Erastosthene: Cho một số tự nhiên N, ph-ong pháp sàng Erastosthene sau đây sẽ cho tập hợp các số nguyên tố nhỏ hơn N:  
Ta viết dãy số 2,3,4,...,N. Sau đó xóa các bội số của 2 khác 2, xóa các bội số của 3 khác 3, trong các số còn lại chọn số nhỏ nhất, thực ra là số 5, xóa các bội số của 5 khác 5,... Cho đến tối đa là  $[N/2]$  thì ta đ-ợc các số còn lại đều là các số nguyên tố. Hãy viết ch-ong trình theo thuật toán trên đây để xác định các số nguyên tố nhỏ hơn 1000.
  12. Dùng while để tính số  $\pi$  theo công thức:  
$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots + (-1)^n \frac{1}{2n+1} + \dots$$
  
với độ chính xác Epsilon thỏa mãn:  
$$\frac{4}{2n+1} < \text{Epsilon} = 1.0\text{E-}04$$
  13. Tính  $n!!$   $= 1*3*5*\dots*N$  nếu N lẻ  
 $= 2*4*6*\dots*N$  nếu N chẵn
  14. Lập ch-ong trình con tính N!, viết ch-ong trình để thử và cho hiện kết quả trên màn hình.
  15. Dùng ch-ong trình con N! trên đây, tính  $C_n^k = \frac{n!}{k!(n-k)!}$
  16. Lập bảng  $C_n^k$  theo công thức truy hồi sau:

$$C_n^0 = C_n^n = 1$$

$$C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$$

Tam giác có dòng thứ k chứa các hệ số của nhị thức  $(a+b)^k$  và đ-ợc gọi là tam giác Pascal)

17. Viết ch-ơng trình con với tên AMU để tính  $a^x$  và thử lại trong ch-ơng trình chính.
18. Viết ch-ơng trình con với tên LOGA để tính  $\log_a x$  và thử lại trong ch-ơng trình chính.
19. Viết ch-ơng trình nhập 3 số thực a,b,c và giải ph-ơng trình bậc 2  
 $ax^2 + bx + c = 0$
20. Viết ch-ơng trình tìm USCLN của 2 số nguyên M,N theo thuật toán Euclid.
21. Tính

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!} + \dots$$

Với độ chính xác  $|\frac{x^n}{n!}| < \text{epsilon} = 1.0E-04$

22. a. Viết ch-ơng trình nhập số liệu cho ma trận A 2 chiều MXN trong đó M,N là các hằng cho ở phần khai báo là M=2, N=5. Sau đó tìm ma trận chuyển vị B, tức là

$$b_{ij} = a_{ji}$$

Cho hiện 2 ma trận trên màn hình để tiện so sánh.

b. Viết ch-ơng trình tính tích 2 ma trận A cấp MXN và B cấp NXK.

23. Nhập ma trận vuông A(NXN)

- Đếm các phần tử >0

- Đếm các phần tử <0

24. Cho một chuỗi ký tự có độ dài N, hãy đếm số lần xuất hiện của các ký tự 'A', 'B', 'C'.
- 25\*. Tạo tệp tep1.dat và nhập M số nguyên, tệp tep2.dat và nhập N số nguyên. Hãy viết ch-ơng trình tạo tep3.dat chứa các số nguyên trong 2 tệp tep1.dat và tep2.dat
- 26\*. Viết ch-ơng trình tạo một tệp văn bản tep1.txt và nhập một số dòng văn bản sau đó đếm số lần xuất hiện của mỗi ký tự từ A đến Z chứa trong tệp văn bản này.
- 27\*. Trong khi mở file mới để ghi, nếu trên đĩa có sẵn file cùng tên thì file trên đĩa sẽ bị xóa. Hãy viết ch-ơng trình tạo tệp tep1.dat chứa các số nguyên và kiểm tra sao cho nếu tồn tại file cùng tên thì đặt câu hỏi có muốn ghi đè lên không.

28. Lập ch-ơng trình tính tiền điện cho khách hàng giá điện tính theo KW nh- sau:

Điện tiêu thụ (KW)	Giá (đồng)
0 - 100	500 / KW
101 - 150	550 / KW
151 - 200	600 / KW
> 200	650 / KW

29. Biết rằng lãi suất gửi tiết kiệm kỳ hạn 1 năm là 0,9%. Hãy nhập vào số tiền gửi của một ng-ời và tính số tiền của ng-ời đó sau N năm (N nhập từ bàn phím), biết rằng sau một năm thì tiền lãi sẽ đ-ợc nhập vào gốc.
30. Lập ch-ơng trình tính c-ớc cho một cuộc gọi điện thoại nội tỉnh. Biết :  
 Thời gian bắt đầu, thời gian kết thúc cuộc gọi tính theo giờ, phút và đ-ợc nhập từ màn phím.  
 C-ớc dịch vụ điện thoại: 500đ / phút.
31. Hãy viết hàm tính tổng các chữ số của một số nguyên bất kỳ.  
 Ví dụ: Số 8545604 có tổng các chữ số là:  $8+5+4+5+6+0+4= 32$

32. Viết lại hàm tính  $\sin(x)$  và so sánh kết quả của hàm tự tạo với giá trị hàm chuẩn:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \dots$$

Với độ chính xác  $\left| \frac{x^{2n+1}}{(2n+1)!} \right| < \text{epsilon} = 1.0\text{E-}04$

33. Xây dựng chương trình quản lý học sinh. Mỗi học sinh quản lý các thông tin sau: Số thứ tự, Họ tên, Năm sinh, Điểm trung bình.  
Nhập danh sách cho 10 học sinh  
Sắp xếp lại danh sách theo điểm trung bình tăng dần.
34. Lập chương trình sắp xếp lại một danh sách học sinh theo họ và tên theo thứ tự tăng dần.
35. Nhập vào dãy số thực  $X_1, X_2, \dots, X_n$ . Tìm MAX của dãy đó và vị trí của chúng trong dãy.
36. Nhập vào dãy số thực  $X_1, X_2, \dots, X_n$  và 2 số  $a, b$ . Tính và in trung bình cộng các số trong khoảng  $(a, b)$ .
37. Nhập một dãy số nguyên. In các số chia hết cho 3 trên một dòng. Các số còn lại trên dòng tiếp theo. Dòng thứ 3 in tất cả các số.
38. Nhập các hệ số  $a_i$  ( $i=0, 1, 2, \dots, n$ ) của đa thức  $P(x)$  bậc  $n$  ( $a_i$  là hệ số của  $x^i$ ) và nhập 3 giá trị  $d_1, d_2, d_3$ . Tính các giá trị  $P(d_1), P(d_2), P(d_3)$ . Tìm  $M = \text{Max}$  của 3 giá trị trên.
39. Nhập các hệ số  $a_i$  ( $i=0, 1, 2, \dots, n$ ) của đa thức  $P(x)$  bậc  $n$  ( $a_i$  là hệ số của  $x^i$ ) và nhập các hệ số  $b_j$  ( $j=0, 1, 2, \dots, n$ ) của đa thức  $Q(x)$  bậc  $m$  ( $b_j$  là hệ số của  $x^j$ ). In ra các hệ số của đa thức tổng.
40. Nhập ma trận cấp  $m \times n$ . Tìm một phần tử Max của mỗi hàng. In mỗi phần tử tìm được trên một dòng (thông tin in ra càng đầy đủ càng tốt).