



GIÁO TRÌNH
TIN HỌC ĐẠI CƯƠNG

MỤC LỤC

Chương 1 : ĐIỆN TOÁN CĂN BẢN	1
1.1 Lịch sử máy tính	4
1.2 Khái niệm tin học và máy tính	5
1.2.1 Tin học là gì ?	5
1.2.2 Máy tính điện tử (Computer)	5
1.3 Các hệ đếm	5
1.4 Các bộ phận cơ bản của máy vi tính	14
1.4.1 Bộ xử lý trung tâm(CPU : Central Processor Unit).....	14
1.4.2 Bộ nhớ (Memory)	15
1.4.3 Thiết bị ngoại vi.....	16
1.5 Chương trình phần mềm	20
1.5.1 Phần mềm hệ thống	20
1.5.2 Trình dịch và ngôn ngữ	23
1.5.3 Phần mềm ứng dụng	24
1.6 Các ứng dụng tin học	24
Chương 2 : HỆ ĐIỀU HÀNH	25
2.1 Khái niệm và phân loại.....	25
2.1.1 Khái niệm.....	25
2.1.2 Phân loại hệ điều hành	26
2.2 Hệ điều hành MS-DOS	26
2.2.1 Các thành phần cơ bản	26
2.2.2 Quá trình làm việc của MS-DOS	30
2.2.3 Các lệnh cơ bản của MS-DOS	31
2.3 Hệ thống quản lý file.....	39
2.3.1 Giới thiệu	39
2.3.2 Các khái niệm	40
2.3.3 Hệ thống quản lý file	44
Thực hành chương 2.....	44
Chương 3 : HỆ ĐIỀU HÀNH WINDOWS	46
3.1 Tổng quan về Windows	46
3.2 Làm việc với Windows	47
3.2.1 Khởi động và thoát khỏi Windows XP	47
3.2.2 Một vài thuật ngữ thường sử dụng	48
3.2.3 Giới thiệu màn hình nền (Desktop) của Windows XP	49
3.2.4 Cửa sổ chương trình	52
3.2.5 Hộp thoại (Dialogue Box)	53
3.2.6 Sao chép dữ liệu trong Windows	55
3.2.7 Cách khởi động và thoát khỏi chương trình	55
3.2.8 Menu Documents.....	57

3.2.9 Tìm kiếm dữ liệu	57
3.2.10 Thay đổi cấu hình máy tính	60
3.2.11 Máy In.....	67
3.2.12 Taskbar AND Start Menu	68
3.3 Windows Explorer	71
3.3.1 Giới thiệu	71
3.3.2 Thao tác với các thư mục và tập tin	73
3.3.3 Thao tác với các lối tắt (Shortcuts).....	76
3.3.4 Định dạng đĩa	77
3.3.5 Hiển thị thông tin của đĩa.....	78
Thực hành chương 3.....	79
Chương 4 : PHÒNG VÀ CHỐNG VIRUS	85
4.1 Virus máy tính là gì?.....	85
4.2 Cách thức phá hoại của virus tin học.....	85
4.3 Các phương pháp phòng và chống virus	86
4.5 Chương trình diệt virus BKAV	86
Chương 5 : NGÔN NGỮ LẬP TRÌNH PASCAL.....	89
5.1 Giới thiệu:	89
5.1.1 Những quy định về cách viết biểu thức và chương trình.....	91
5.1.2 Cấu trúc chung của một chương trình Pascal	92
5.2 Các kiểu dữ liệu	95
5.2.1 Kiểu số nguyên.....	95
5.2.2 Kiểu số thực	95
5.2.3 Kiểu kí tự.....	97
5.2.4 Kiểu logic (boolean).....	97
5.2.5 Kiểu liệt kê (Enumerated scalar type).....	97
5.2.6 Kiểu khoảng con (Sub-range type)	98
5.3 Khai báo hằng, biến, biểu thức, câu lệnh.....	99
5.3.1 Khai báo hằng.....	99
5.3.2 Khai báo biến.....	99
5.3.3 Biểu thức.....	100
5.3.4 Câu lệnh	101
5.4 Lệnh nhập và xuất dữ liệu	101
5.4.1 Lệnh nhập dữ liệu.....	101
5.4.2 Lệnh xuất dữ liệu.....	102
5.4.3 Kết hợp read, readln và write, writeln để nhập dữ liệu.....	103
5.5 Các lệnh có cấu trúc.....	104
5.5.1 Câu lệnh điều kiện.....	104
5.5.2 Câu lệnh lựa chọn : CASE..OF	106
5.5.3 Lệnh lặp có số lần xác định.....	108
5.5.4 Lệnh lặp có số bước lặp không xác định	113
Thực hành chương 5.....	119

ĐINH VĂN ĐỆ

Chương 1 : ĐIỆN TOÁN CĂN BẢN

1.1 Lịch sử máy tính

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

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

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

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

❖ Thế hệ 3 (1965 - 1974): Máy tính được gắn các bộ xử lý bằng vi mạch điện tử cỡ nhỏ có thể có được tốc độ tính khoảng 100.000 đến 1 triệu phép/s.

Máy đã có các hệ điều hành đa chương trình, nhiều người dùng đồng thời hoặc theo kiểu chia thời gian. Kết quả từ máy tính có thể in ra trực tiếp ở máy in. Điển hình như loại IBM 360 (Mỹ) hay EC (Liên Xô cũ), ...

❖ Thế hệ 4 (1974 - 1990): Máy tính bắt đầu có các vi mạch đa xử lý có tốc độ tính hàng chục triệu đến hàng tỷ phép/giây. Giai đoạn này hình thành 2 loại máy tính chính : máy tính cá nhân để bàn (Personal Computer - PC) hoặc xách tay (Laptop hoặc Notebook computer) và các loại máy tính chuyên nghiệp thực hiện đa chương trình, đa vi xử lý ... hình thành các hệ thống mạng máy tính (Computer Networks), và các ứng dụng phong phú đa phương tiện.

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

1.2 Khái niệm tin học và máy tính

1.2.1 Tin học là gì ?

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

1.2.2 Máy tính điện tử (Computer)

Là một thiết bị điện tử dùng để lưu trữ và xử lý thông tin theo các chương trình định trước do con người định ra

1.3 Các hệ đếm

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

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

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

❖ Hệ đếm thập phân (decimal system)

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

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

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

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

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

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

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

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

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

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

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

$$\text{Ví dụ: } 254.68 = 2 \times 10^2 + 5 \times 10^1 + 4 \times 10^0 + 6 \times 10^{-1} + 8 \times 10^{-2}$$

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

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

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

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

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

❖ Hệ đếm nhị phân

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

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

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

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

như vậy:

$$11101.11_{(2)} = 1x16 + 1x8 + 1x4 + 0x2 + 1x1 + 1x0.5 + 1x0.25 = 29.75_{(10)}$$

trong tự số 10101 (hệ 2) sang hệ thập phân sẽ là:

$$10101_{(2)} = 1x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 16 + 0 + 4 + 0 + 1 = 21_{(10)}$$

❖ Hệ đếm bát phân (octal number system)

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

Ví dụ: $235.64_{(8)} = 2x8^2 + 3x8^1 + 5x8^0 + 6x8^{-1} + 4x8^{-2} = 157.8125_{(10)}$

❖ Hệ đếm thập lục phân (hexa-decimal number system)

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

Ví dụ: $34F5C_{(16)} = 3x16^4 + 4x16^3 + 15x16^2 + 5x16^1 + 12x16^0 = 216294_{(10)}$

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

Ví dụ: Số 15 viết là FH.

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

Hệ 10	Hệ 2	Hệ 8	Hệ 16
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4

5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

❖ Chuyển đổi số giữa các hệ đếm

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

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

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

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

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

phần nguyên ở đây được hiểu là phần bên trái dấu chấm thập phân).

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

Ta có $k = 14$, $d = 0.125$

Chuyển đổi phần nguyên 14

Chia 2	Dư
14	0
7	1
3	1
1	1
0	

Chuyển đổi phần thập phân 0.125

Nhân 2	Phần nguyên
0.125	
0.25	0
0.5	0
1	1

Vậy $14.125 = 1110.001$

Chuyển đổi 0.2 sang hệ nhị phân:

Nhân 2	Phần nguyên
0.2	
0.4	0
0.8	0
1	
1	

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

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

❖ 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 dư, nhân lấy phần nguyên... để tìm biểu diễn của B trong hệ thập phân)

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

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

Ví dụ cho số nhị phân 11110101 chuyển số này sang dạng bát phân và thập lục phân. (11 110 101) -> 365 trong hệ bát phân là số 365

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

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

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

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

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

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

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

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

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

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

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

8^3			8^2			8^1			8^0		
2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
1	0	1	1	1	0	1	1	0	1	0	1
5			6			6			5		

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

Kết quả:

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

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

	16^2				16^1				16^0			
2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0	
1	0	1	1	1	0	1	1	0	1	0	1	
	11				11				5			

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

Kết quả:

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

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

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

Ví dụ: $2997_{(10)} = ?_{(2)}$

THÔNG QUA HỆ 8:

Chia 8	dư
2997	5
374	6
46	6
5	5

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

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

Suy ra:

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

THÔNG QUA HỆ 16:

Chia 16	dư
2997	5
187	11
11	11

Ta có : $2997_{(10)} = BB5_{(16)}$

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

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

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

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

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

Ví dụ: $5665_{(8)} = ?_{(16)}$

Cách làm như sau:

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

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

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

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

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

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

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

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

$$\text{Vậy: } 5665_{(8)} = BB5_{(16)}$$

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

1.4 Các bộ phận cơ bản của máy vi tính

1.4.1 Bộ xử lý trung tâm (CPU : Central Processor Unit)

a) Chức năng của CPU

CPU là bộ chỉ huy của máy vi tính, có các chức năng sau:

- Thực hiện việc nhận lệnh, giải mã lệnh và điều khiển các khối khác thực hiện lệnh.

- Thực hiện các phép tính số học, logic và các phép tính khác.

- Sinh ra các tín hiệu địa chỉ trên máy.

b) Cấu tạo của CPU

CPU bao gồm các phần sau:

❖ Đơn vị điều khiển (Control unit): Thực hiện việc nhận lệnh, giải mã lệnh và điều khiển các khối khác thực hiện lệnh và sinh ra các tín hiệu địa chỉ trên máy để quản lý bộ nhớ.

❖ Đơn vị tính toán số học và logic (Arithmeric - Logic Unit): Bao gồm các vi mạch tính để thực hiện các phép tính số học, logic và các phép tính khác.

❖ Thanh ghi (Register) là một cấu trúc gồm 16 bit (hoặc 32 bit) nhớ liền kề nhau được thiết lập ngay trong mạch vi xử lý. Các thanh ghi này được phân thành 4 nhóm theo mục đích sử dụng sau:

- Nhóm 1: 4 thanh ghi đa năng kí hiệu là AX, BX, CX, DX được sử dụng cho nhiều mục đích khác nhau.

- Nhóm 2: 4 thanh ghi địa chỉ đoạn (Segment) bộ nhớ, thanh ghi CS (Code segment) chứa địa chỉ đoạn mã lệnh, thanh ghi DS (Data Segment) chứa địa chỉ đoạn dữ liệu, thanh ghi SS (Stack Segment) chứa địa chỉ đoạn ngăn xếp, thanh ghi ES (Extra Segment) chứa địa chỉ đoạn dữ liệu mở rộng.

- Nhóm 3: 5 thanh ghi địa chỉ tương đối (offset), thanh ghi IP (Instruction Pointer) trỏ tới ô chứa lệnh trong đoạn CS; thanh ghi SI (Source Index) trỏ tới ô chứa dữ liệu nguồn trong đoạn DS, Thanh ghi DI (Destination Index) trỏ tới ô chứa dữ liệu đích trong đoạn DS, Thanh ghi SP (Stack Pointer) trỏ tới ô chứa dữ liệu trong đoạn ngăn xếp SS và thanh ghi BP (Base Pointer) trỏ tới ô dữ liệu trong đoạn chứa dữ liệu mở rộng ES.

- Nhóm 4: 1 thanh ghi cờ (Flag) để ghi các trạng thái (ta gọi là cờ) xảy ra trong các phép toán số học.

1.4.2 Bộ nhớ (Memory)

Bộ nhớ hay còn gọi là bộ nhớ trong (bộ nhớ) chính của máy vi tính.

❖ Chức năng của bộ nhớ: Bộ nhớ dùng để chứa các thông tin cần thiết như chương trình, dữ liệu trong quá trình máy hoạt động.

❖ Tổ chức của bộ nhớ:

- Các thông tin được cất trong bộ nhớ dưới dạng mã nhị phân (các bit 0 hoặc 1).

- Một nhóm các bit có thể là 16 bit hoặc 32 bit,... gọi là một ô nhớ. Mỗi ô nhớ được đánh một số thứ tự trong hệ 16 gọi là địa chỉ của ô nhớ. Thông tin ghi bên trong ô nhớ là nội dung của ô nhớ. Dùng địa chỉ để truy nhập vào ô nhớ để đọc nội dung hoặc ghi nội dung vào ô nhớ.

- Kích thước của bộ nhớ hay dung lượng của bộ nhớ là số lượng thông tin mà bộ nhớ có khả năng chứa được, thường dùng đơn vị MB hoặc GB.

❖ Phân loại bộ nhớ: Theo tính chất thông tin chứa trong bộ nhớ người ta chia thành bộ nhớ ROM và RAM.

- ROM (Read Only Memory): là bộ nhớ cố định cho phép chỉ đọc thông tin mà không ghi thông tin vào được. ROM là bộ nhớ cứng do hãng chế tạo cài đặt sẵn các chương trình bên trong, bao gồm các chương trình kiểm tra và các chương trình cơ sở cốt lõi nhất của máy vi tính. Các thông tin trong ROM sẽ không bị mất đi khi ta tắt nguồn của máy.

- RAM (Random Access Memory): Là bộ nhớ mềm, có thể thay đổi, truy nhập một cách ngẫu nhiên. RAM làm từ các mạch vi mạch (gọi là Chip nhớ). RAM dùng để ghi chương trình của hệ điều hành nạp vào từ đĩa khởi động, chương trình và dữ liệu của người sử dụng. Có thể đọc và ghi thông tin vào RAM. Thông tin trong RAM sẽ bị mất đi khi ta tắt nguồn của máy, do vậy ta phải lưu trữ thông tin ra bộ nhớ ngoài.

1.4.3 Thiết bị ngoại vi

a) Bàn phím (Keyboard)

❖ Bàn phím là thiết bị vào thông dụng của máy vi tính. Dùng bàn phím có thể đưa vào máy các lệnh điều khiển, chương trình, dữ liệu.

❖ Nguyên tắc của bàn phím: ấn một phím thì gây ra sự tiếp điện giữa một dây dọc và một dây ngang tạo ra một xung điện, xung điện này qua chương trình điều khiển bàn phím sẽ đưa vào máy mã ASCII của kí tự của phím đó.

❖ Bàn phím gồm 5 khu vực sau:

- Khu phím máy chữ gồm các phím chữ, phím dấu và phím số như các phím trên bàn phím máy chữ trong đó có phím cách (Space bar)

- Khu phím chức năng F1,F2,..F12 có tác dụng thực hiện một chức năng hoàn chỉnh nào đó khi ta gõ phím đó, tác dụng của nó được cài đặt tùy theo chương trình. Thí dụ DOS dùng phím F1 để nhắc lại từng kí tự của câu lệnh vừa gõ trước, F3 để nhắc lại toàn bộ câu lệnh vừa gõ.

- Khu vực phím số ở bên phải bàn phím

- Khu phím dịch chuyển con trỏ màn hình gồm các phím dịch lên, dịch xuống, dịch trái, dịch phải, dịch về đầu màn hình (Home), dịch xuống cuối (End), dịch lên một trang (PgUp), dịch xuống một trang (PgDn), Dịch trái và xoá (Back space), phím TAB dịch con trỏ đi một khoảng 8 cột (gọi là một tab)

- Khu phím điều khiển:

ESC để thoát ra khỏi chương trình hoặc lệnh đang thực hiện

CapsLock để chuyển đổi thường xuyên kiểu chữ nhỏ sang to và ngược lại.

Shift có tác dụng chuyển tạm thời chữ to thành nhỏ hoặc ngược lại khi đè giữ phím đó và gõ chữ. Với các phím có 2 kí tự thì nếu gõ được kí tự dưới, nếu đè shift và gõ thì cho kí tự trên.

Insert - chèn kí tự vào vị trí con trỏ

Delete - xoá kí tự tại vị trí con trỏ

Print Screen - In màn hình

Scroll Lock - bật hoặc tắt trạng thái cuộn màn hình

Pause/Break - Chờ hoặc ngắt chương trình

Ctrl và Alt dùng để mở rộng chức năng của bàn phím.

Num Lock - để kích hoạt bàn phím số, bàn phím số chỉ có tác dụng khi nào đèn Num Lock sáng.

b) Màn hình (Display)

❖ Màn hình là thiết bị ra thông dụng tối thiểu của máy vi tính. Các kí hiệu ta gõ vào từ bàn phím, các kết quả xử lý, các thông báo của máy vi tính đều được hiện ra trên màn hình.

❖ Màn hình có cấu tạo vật lý giống như một màn hình vô tuyến thông thường cũng gồm có ống phóng tia điện tử, mạch quét ngang và quét dọc, tia điện tử đập vào màn hình tạo thành một điểm sáng, bộ điều khiển CRT tạo tín hiệu bật hoặc tắt tia điện tử theo tín hiệu lấy ra từ bộ nhớ màn hình 1 hay 0, tia điện tử quét từ trái sang phải tạo thành một dòng, từ dòng trên cùng xuống dòng dưới cùng tạo thành một màn hình. Tần số quét màn hình bằng 60Hz nghĩa là trong 1 giây đồng hồ màn hình được làm mới lại từ đầu 60 lần, thời gian hồi dọc để tia điện tử chuyển về góc trái trên cùng là 1,25ms.

❖ Có 2 kiểu màn hình: kiểu màn hình văn bản và màn hình đồ họa:

- Màn hình văn bản (Text mode) được phân thành 2 chế độ: 80 cột (kí tự) 25 dòng là dạng chuẩn hoặc 40 cột (kí tự) 25 dòng.

- Màn hình đồ họa (Graphic Mode) được thiết kế gồm nhiều điểm sáng (Pixel) theo hai chiều ngang và dọc. Màn hình đồ họa được phân thành nhiều chế độ khác nhau tùy theo độ phân giải, chẳng hạn chiều ngang 640 pixel chiều dọc 200 pixel (640x200), 640x350, 640x480, 1024x768. Màn hình càng có độ phân giải cao thì hình ảnh càng mịn.

- Màu của màn hình được tạo ra bởi sự pha trộn của 4 yếu tố sau: 3 màu cơ bản là đỏ (Red), xanh lá cây (Green), xanh da trời (Blue) và độ chói. Với tổ hợp của 3 màu cơ bản ở độ chói khác nhau của từng màu mà máy tính có thể tạo ra tới 256 màu khác nhau.

c) Ổ đĩa và đĩa từ

Ổ đĩa và đĩa từ là thiết bị vào / ra : Có thể đọc thông tin từ đĩa từ và ghi thông tin vào đĩa từ.

❖ Ổ đĩa là khối hình chữ nhật có chứa các thiết bị và linh kiện để làm việc với đĩa từ. Máy vi tính có 2 loại ổ đĩa từ là ổ đĩa cứng (Hard Disk/ Fixed Disk) và ổ đĩa mềm (Soft Disk/Floppy Disk).

- Ổ đĩa cứng được lắp trong thân máy, nếu máy có 1 ổ đĩa cứng thì tên ổ đĩa là C: , nếu máy có 2 ổ đĩa cứng thì tên là C: , D: . Ổ đĩa cứng chứa các đĩa cứng có dung lượng lưu trữ lớn từ 40 MB đến hàng chục GB.

- Ổ đĩa mềm lắp ngay bên ngoài vỏ máy, 2 ổ mềm có tên là A: , B: , ổ đĩa mềm chứa các đĩa mềm có dung lượng lưu trữ nhỏ hơn đĩa cứng, hiện nay thông dụng là 1,44MB.

❖ Đĩa từ có hình dạng tròn được cấu tạo bằng chất dẻo (đĩa mềm) hoặc nhôm (đĩa cứng) có phủ các hạt sắt từ trên bề mặt để lưu trữ thông tin dựa theo đặc tính nhiễm từ của các hạt sắt từ. Các thông tin lưu trữ trên đĩa từ ở dạng mã nhị phân (bit 0 hoặc 1).

- Mỗi đĩa từ mềm gồm có hai mặt từ hoá được đánh số là mặt 0/1, có 2 đầu từ tiếp xúc trên dưới để đọc và ghi thông tin, mỗi mặt phân thành nhiều rãnh (Track) đánh số từ 0 trở đi ,rãnh ngoài gần mép đĩa là rãnh số 0, mỗi mặt được phân thành nhiều cung (sector) được đánh số từ 1 trở đi, (đĩa 1,2M phân thành 80 rãnh và 15 sector, đĩa 1,44 M phân thành 80 rãnh và 18 sector), mỗi cung chứa 512 byte.

- Đĩa cứng gồm nhiều mặt đĩa cùng trục quay tạo thành khối trụ gọi là cylinder, số mặt tùy theo dung lượng của ổ đĩa, chẳng hạn ổ đĩa có 8 đầu từ (Head) đánh số từ 0 đến 7 gồm 4 mặt đĩa, mỗi mặt của đĩa cứng được phân thành 17 sector, mỗi mặt gồm nhiều rãnh (đĩa cứng XT gồm 305 rãnh, đĩa cứng AT gồm 615 rãnh), mỗi rãnh trên một cung chứa 1024 byte.

d) Máy in (Printer)

❖ Máy in là thiết bị ra của máy vi tính, dùng để in ra kết quả xử lý, dữ liệu, chương trình, thông báo của máy vi tính.

❖ Máy in dùng trong máy tính gồm nhiều loại khác nhau hoạt động theo các nguyên lý khác nhau. Ta có thể phân thành các loại cơ bản sau:

- Máy in màu (Plotter) gồm 8 bút vẽ màu khác nhau hoạt động trên nguyên tắc bút vẽ.

- Máy in Lazer hoạt động trên nguyên tắc dùng tia Laser ép nóng chảy bột mực khô bám trên lô.

- Máy in kim (9 kim hoặc 24 kim) hoạt động trên nguyên tắc kim phun bắn vào băng mực và tạo ra trên giấy các nốt chấm đen.

Hiện nay máy in Laser được sử dụng rộng rãi vì có ưu điểm là in nhanh và chất lượng tốt nhưng giá đắt hơn máy in kim. Máy in kim giá rẻ hơn và bền hơn nhưng chất lượng không cao và in chậm hơn.

e) Thiết bị chuột

Chuột là một thiết bị vào của máy vi tính. Có thể chọn lệnh, chọn các đối tượng, di chuyển và thay đổi kích thước các đối tượng bằng chuột.

Thông thường chuột có 2 nút bấm. Nút trái dùng cho phần lớn các thao tác, nút phải tùy theo từng chương trình mà có các chức năng khác nhau, thông thường nhấn nút phải để hiện ra các lệnh có thể thực hiện được tại vị trí vừa chọn của chuột. lệnh này tùy thuộc phần mềm đang sử dụng.

f) Các thiết bị khác

Các thiết bị ngoại vi của máy vi tính hiện nay rất phong phú, ngoài các thiết bị cơ bản nêu trên còn các thiết bị khác như : đĩa CD-ROM, máy chiếu (Projecter), máy quét ảnh (Scanner), ... Tất cả các thiết bị ngoại vi giúp cho việc đưa thông tin vào máy ở nhiều dạng từ các kí tự đến âm thanh, hình ảnh.

1.5 Chương trình phần mềm

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

1.5.1 Phần mềm hệ thống

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

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

❖ **Hệ điều hành:**

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

Các chức năng của hệ điều hành:

- Định vị các tài nguyên hệ thống: điều khiển sự vận chuyển bên trong máy tính, có tác dụng quyết định đến tài nguyên nào sẽ được sử dụng và trong thời gian bao lâu.

- Thời gian : định thời trong CPU được phân chia thành các khoảng thời gian có đơn vị tính là ms. Đối với mỗi tác vụ được thực thi, CPU sẽ ấn định đích xác một khoảng thời gian nhất định cho nó. Khi thời gian kết thúc, các tác vụ khác sẽ được đưa vào để tiếp tục quay vòng.

- Bộ nhớ : bộ nhớ máy tính cũng được quản lý bởi hệ điều hành. Chúng được CPU sử dụng luân phiên để rời chuyển dữ liệu thông qua các bộ nhớ đệm. Chẳng hạn, các khoảng không gian của đĩa cứng cũng được coi như là một phần của bộ nhớ chính. Điều đó có nghĩa là, việc đưa dữ liệu thẳng từ đĩa cứng lên bộ nhớ chính mỗi khi CPU yêu cầu sẽ rất chậm trong khi dữ liệu đó phải thường xuyên cập nhật lên bộ nhớ. Vì vậy khi bộ nhớ chính bị đầy, một vài dữ liệu sẽ được phân trang đưa ra ngoài đĩa cứng và nó được gọi là vùng nhớ trao đổi

(swapping). Trong hệ điều hành Windows, việc sử dụng tệp tin swap cũng chính là cho mục đích này.

- Quản lý vào/ra (input/output) : điều khiển luồng lưu lượng cũng là một phần khả năng đáp ứng của hệ điều hành. Hệ điều hành phải quản lý tất cả các yêu cầu như đọc dữ liệu từ ổ đĩa và băng từ hay ghi dữ liệu vào chúng hoặc đưa ra máy in.

- Giám sát hoạt động hệ thống : hệ điều hành thực hiện 2 công việc giám sát chủ yếu là thi hành hệ thống và bảo mật hệ thống. Thi hành hệ thống đưa ra các thông tin như tên các tác vụ đang thi hành trên hệ thống, thời gian chiếm giữ CPU, bộ nhớ... hay người quản trị hệ thống cũng có thể theo dõi, kiểm tra xem các máy tính trên mạng có bị quá tải không để có những can thiệp kịp thời. Vấn đề bảo mật hệ thống khá quan trọng, đặc biệt khi sử dụng cho chế độ đa người dùng. Khi người sử dụng muốn truy nhập vào hệ thống cần phải thực hiện thủ tục đăng nhập user và mật khẩu (gọi là ID).

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

Các hệ điều hành cơ bản :

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

- Apple Macintosh.

- OS/2 của hãng IBM.

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

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

❖ Các chương trình tiện ích

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

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

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

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

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

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

- Chương trình phòng chống Virus: Là các phần mềm khác nhau được cài đặt vào bộ nhớ máy tính nhằm giám sát sự hoạt động của virus trong máy tính và mạng. Nó có nhiệm vụ phát hiện sự hoạt động của virus trong bộ nhớ máy tính và thực hiện “làm sạch” chúng. Hiện nay các phiên bản phần mềm phòng chống virus hữu hiệu được sử dụng rộng rãi như Norton AntiVirus, D2, Bkav2000,...

1.5.2 Trình dịch và ngôn ngữ

Trong quá trình hoạt động của các thể hệ máy tính điện tử, tất cả các chỉ thị lệnh được hoạt hóa bởi máy tính đều phải cung cấp tới CPU dưới dạng mã máy (machine code). Tuy nhiên, con người không phải tương tác với máy tính tại mức này mà các nhà lập trình có thể viết phần mềm bằng ngôn ngữ Assembler cho

phép can thiệp trực tiếp tới các thanh ghi của CPU. Hiện nay, hầu hết các chương trình được viết trên cơ sở các tiêu chuẩn ngôn ngữ bậc cao như Pascal, Basic, C,...

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

1.5.3 Phần mềm ứng dụng

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

1.6 Các ứng dụng tin học

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

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

Chương 2 : HỆ ĐIỀU HÀNH

2.1 Khái niệm và phân loại

2.1.1 Khái niệm

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

chương trình ứng dụng: tạo các thư mục, copy các file, xóa file trên đĩa để lấy chỗ trống... nghĩa là OS không những là cầu nối giữa phần cứng và các phần mềm ứng dụng, mà còn là cầu nối giữa người sử dụng với máy.

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

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

2.1.2 Phân loại hệ điều hành

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

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

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

Theo tiêu chuẩn thứ 2 ta có:

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

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

2.2 Hệ điều hành MS-DOS

2.2.1 Các thành phần cơ bản

❖ **Boot record** : Môđun này chính là thẻ khởi động sẽ khởi động hệ điều hành. Nhiệm vụ của Boot Record là kiểm tra trong thư mục của đĩa xem có các tệp hệ thống IO.SYS, MSDOS.SYS không? Hai tệp hệ thống này đều được

bảo vệ chống xoá, chống ghi đè, mang dấu hiệu hệ thống và được ấn dấu. Nếu DOS tìm thấy thì nạp hai tệp IO.SYS, MSDOS.SYS vào bộ nhớ trong, sau đó chuyển điều khiển cho IO.SYS

❖ **Tệp hệ thống IO.sys :**

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

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

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

- Nhiệm vụ thứ 3: Điều khiển các ngoại vi mới được nối với máy tính PC, máy vẽ, máy quét. Việc quản lý và điều khiển các thiết bị mới để giải quyết nếu các nhà lập trình hệ thống sửa IO.SYS hơn là sửa chương trình trong ROM-BIOS. Cùng với việc xuất hiện các thiết bị vào ra mới có trên thị trường nối với máy tính PC, người ta phải liên tục sửa các tệp IO.SYS (thậm chí sửa cả chương trình khác của DOS) cho nên tệp IO.SYS (và cả MSDOS.SYS) luôn thay đổi về nội dung (thể hiện bởi số Byte và ngày tháng tạo tệp). Sự sửa đổi này của hãng làm phần mềm hệ thống Microsoft hay IBM không có gì khó cả nhưng với người sử dụng thì không thể làm được vì khả năng và vì sự thiếu thông tin hệ thống thường được giữ bí mật bởi các nhà sản xuất.

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

❖ Tệp hệ thống MSDOS.sys :

Trong hai tệp hệ thống thì tệp thứ hai này đóng vai trò trung tâm bao gồm các chương trình con dịch vụ còn lại của DOS. Người ta có thể ghép IO.SYS

với MSDOS.SYS thành một tệp nhưng để đảm bảo môđun hoá nên chúng được chia riêng để dễ sửa đổi.

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

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

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

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

❖ **Chương trình COMMAND.COM và các lệnh nội trú**

COMMAND.COM có một vài chức năng:

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

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

đặt đường dẫn. Ba loại đuôi chương trình mà DOS có thể thực hiện theo thứ tự ưu tiên là: .COM, .EXE và .BAT.

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

❖ **Lệnh ngoại trú :**

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

2.2.2 Quá trình làm việc của MS-DOS

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

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

Khi khởi động máy tính thường thường máy sẽ tìm đĩa hệ thống theo thứ tự sau: đĩa mềm, đĩa CD-ROM, đĩa cứng. Khi đó MS-DOS sẽ nạp ba tệp tin COMMAND.COM, IO.SYS và MSDOS.SYS vào bộ nhớ RAM của máy tính.

Sau khi DOS đã được đưa vào bộ nhớ để sẵn sàng làm việc đầu tiên sẽ xuất hiện trên màn hình chờ để người sử dụng đánh lệnh vào. Những phần chính của DOS sẽ lưu trữ trong bộ nhớ cho đến khi ta tắt máy.

2.2.3 Các lệnh cơ bản của MS-DOS

a) Lệnh nội trú (internal command)

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

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

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

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

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

Các lệnh nội trú cơ bản:

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

Drive : Tên ổ đĩa

Path : Đường dẫn

↵: Phím Enter

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

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

❖ **Chuyển ổ đĩa:**

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

Với drive: là ổ đĩa cần chuyển đến.

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

❖ **Liệt kê thư mục (DIR)**

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

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

Ghi chú:

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

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

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

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

N : theo alphabetic tên file;

E : theo alphabetic tên phần mở rộng

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

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

G : theo nhóm thư mục trước

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

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

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

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

Cú pháp: CD [drive :] [path]

Ví dụ : C:\>CD PASCAL sẽ có C:\PASCAL>_

Ghi chú:

- Từ thư mục con, muốn trở về thư mục cha, ta gõ: CD..
- Nếu muốn về thẳng thư mục gốc, ta gõ: CD\
- Để hiển thị đường dẫn hiện hành, ta gõ: CD

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

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

Ghi chú:

- Lệnh TYPE dùng hiển thị một tập tin văn bản chứa mã ASCII mới đọc được.
- Các file chứa mã nhị phân của chương trình như các file *.EXE, *.COM,*.BIN,... thì khi gõ lệnh TYPE sẽ không đọc bình thường được.

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

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

Cú pháp: RD [drive :] <path>

❖ **Sao chép tập tin (COPY)**

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

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

Ghi chú:

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

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

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

Cú pháp 2: cho phép nối các tập tin có sẵn thành một tập tin mới chung.
COPY <file 1> + <file 2> [+ ... + <file n>] [<new_file>]

Ghi chú:

- Nếu không đặt tên tập tin mới thì tất cả các tập tin sẽ ghép chung vào <file 1>

- Nếu <new_file > đã có thì nội dung cũ sẽ được thay bằng nội dung mới.

- Tên <new_file > không được trùng với tên các tập tin cần ghép.

Cú pháp 3: dùng lệnh COPY để tạo ra một tập tin văn bản đơn giản:
COPY CON path]<filename>

Ghi chú:

- Sau khi gõ Enter, ta có thể đánh vào một vài đoạn văn trên bàn phím tùy ý. Muốn kết thúc, đánh tổ hợp phím Ctrl+Z hay F6 và Enter để lưu trữ.

- Khi gõ Enter để xuống dòng, ta không di chuyển con trỏ trở lên dòng trên được.

- Lỗi văn bản sai không thể sửa nội dung trực tiếp khi đã lưu, ngoại trừ phải dùng các trình soạn thảo văn bản khác.

❖ Xoá tập tin (Delete - DEL)

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

Ghi chú:

- Có thể xoá một loạt <tên file> nếu dùng các ký tự * và ?

- [/P] tạo nhắc để xác định từng tập tin muốn xóa

- Xóa tất cả các tập tin thì dùng lệnh DEL *.* khi đó máy sẽ hỏi lại :

All files in directory will be deleted! (Tất cả các tập tin sẽ bị xóa !)

Are you sure (y/n) ? _(Bạn có chắc không (y/n) ?)

Gõ y (yes) nếu muốn xóa tất cả và n (no) khi không muốn dùng lệnh xóa tất cả.

- Lệnh DEL không xóa các tập tin ẩn (hidden) và tập tin chỉ đọc (read only).

Muốn xóa các tập tin này ta phải dùng lệnh ATTRIB ngoại trừ để thay đổi thuộc tính của nó.

- Nếu DEL một thư mục thì xóa hết các tập tin trong thư mục đó.

❖ **Đổi tên tập tin (Rename - REN)**

Cú pháp: REN [drive:][path]<old_file> <new_file>

Ghi chú:

- <old_file> : tên tập tin cũ cần đổi <new_file> : tên tập tin mới

- Trường hợp tập tin mới đã có tên rồi hoặc không có tập tin cũ, máy sẽ báo:

Duplicate file name or file not found

❖ **Xóa màn hình (ClearScreen - CLS)**

Cú pháp: CLS

❖ **Xem và chỉnh thời gian (TIME)**

Cú pháp: TIME [hh:mm:ss]

Ghi chú:

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

Ví dụ : TIME

Current Time is 9 : 30 : 15

Enter new time : _

- Nếu không cần thay đổi thời gian, ta nhấn Enter để xác nhận.
- Muốn chỉnh thời gian, ta nhập giờ : phút : giây phù hợp tại vị trí con trỏ.

❖ Xem và chỉnh ngày tháng (DATE)

Cú pháp: DATE [mm-dd-yy]

Ghi chú:

- Sau khi gõ Enter, trên màn hình sẽ hiện ngày tháng năm hiện tại của hệ thống.
- Kiểu ngày tháng năm do người sử dụng thiết lập.

Ví dụ : DATE

Current date is Sun 09-15-96

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

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

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

Các máy tính Pentium hiện nay đều thể hiện phần năm là con số, ví dụ :
Current date is Fri 08-04-2000

b) Lệnh ngoại trú (external command) :

Như ta đã biết lệnh ngoại trú cũng là những lệnh chứa các chức năng nào đó của hệ điều hành nhưng ít được sử dụng hơn lệnh nội trú nên được để trên đĩa

hay thư mục riêng để đỡ tốn bộ nhớ. Các lệnh ngoại trú phải được nạp từ đĩa vào trong bộ nhớ mới chạy được. Khi thực hiện xong câu lệnh, vùng bộ nhớ có chứa câu lệnh ngoại trú đó sẽ bị thu hồi. Các tập lệnh ngoại trú có phần mở rộng là EXE hay COM hoặc BAT. Khi gọi lệnh mà máy không tìm thấy trên màn hình hình sẽ xuất hiện câu báo lỗi Bad command or file name (Sai lệnh hoặc không có tên tập tin). Các lệnh ngoại trú như:

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

Một số lệnh ngoại trú thường dùng:

❖ Tạo khuôn đĩa (FORMAT)

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

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

Ghi chú:

- Tham số /S dùng để tạo ra đĩa hệ thống. DOS sẽ tạo dạng và chép vào đĩa 3 file hệ thống là COMMAND.COM (file hiện), IO.SYS và MSDOS.SYS (file ẩn).

- Tham số /U thực hiện format không điều kiện (Unconditional)

- Tham số /Q thực hiện format nhanh (Quick)

- Tham số /V[:label] để chỉ định tên nhãn (label) của đĩa

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

C:\>format A:

Insert new diskette for drive A : (Đưa đĩa mới vào ổ A: and press ENTER when ready ... và ấn phím Enter khi sẵn sàng ..)

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

❖ Thay đổi thuộc tính của file (ATTRIB)

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

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

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

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

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

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

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

❖ Liệt kê cây thư mục TREE

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

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

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

❖ Xóa cây thư mục DELTREE

Lệnh này xóa thư mục được chỉ định và tất cả các thư mục con, tập tin của nó. Cú pháp: DELTREE [/y] [drive :] [path] <directory name>

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

❖ **Phục hồi file đã bị xóa (UNDELETE)**

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

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

❖ **Di chuyển file MOVE**

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

Cú pháp 1: Di chuyển tập tin

MOVE[drive:][path]<filename><destination>

Với destination là nơi di chuyển các files đến.

Cú pháp 2: Đổi tên thư mục

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

2.3 Hệ thống quản lý file

2.3.1 Giới thiệu

Khi sử dụng máy tính, một trong những nhu cầu cơ bản & cần thiết là hệ thống máy tính phải có khả năng lưu lại được các thông tin, dữ liệu. Hầu như người sử dụng máy tính nào cũng có nhu cầu lưu lại dữ liệu, sao cho sau khi tắt máy thì dữ liệu vẫn không mất. Ví dụ như các tài liệu quan trọng, các văn bản chương trình, thậm chí các phần mềm cần thiết trên máy tính như hệ điều hành & các chương trình ứng dụng cũng cần phải được lưu giữ trên máy tính. Nếu tất cả mọi thứ trên máy đều mất hết khi máy tính bị tắt hoặc mất điện thì chúng ta khó mà sử dụng được máy tính. Dĩ nhiên không thể nói rằng khi đó ai cần dùng máy tính để làm việc gì thì sẽ ngồi làm ra chương trình để thực hiện việc đó. Vì có rất nhiều người không biết lập trình, hoặc biết lập trình thì không đủ khả năng lập trình trong trường hợp đó, còn với một số rất ít người đủ khả năng lập trình thì

cũng khó mà làm ra được chương trình do trong thời gian đang làm – rất lâu – thì máy tính có thể bị tắt và phần chương trình đang làm dang dở lại mất hết!

(Dữ Liệu hoặc Thông Tin trên máy tính ở đây cần được hiểu theo nghĩa tổng quát, bao gồm tất cả mọi thông tin, dữ liệu cần lưu trữ trên máy tính như hình ảnh, âm thanh, văn bản, tài liệu, trò chơi, nhạc, phim, chương trình ứng dụng, hệ điều hành, ... Và máy tính được xét ở đây là các máy tính đa năng bình thường, không phải các máy đơn giản chuyên biệt như con chuột, màn hình, radio, tivi, ..)

Để thông tin được lưu trữ không bị mất khi tắt máy thì chúng cần lưu vào phần bộ nhớ ngoài. Việc lưu trữ thông tin, tổ chức /sắp xếp thông tin trên bộ nhớ ngoài cũng như cung cấp cho người sử dụng các thao tác cần thiết (chép, xóa, xem, sửa,...) không phải do máy tính hỗ trợ mà được thực hiện bởi hệ điều hành, cụ thể là bởi thành phần **hệ thống quản lý tập tin** – một trong các thành phần chính của hệ điều hành.

2.3.2 Các khái niệm

a) Tập tin :

Trên mỗi hệ thống bộ nhớ ngoài ta có thể lưu trữ được rất nhiều thông tin (vì dung lượng bộ nhớ ngoài thường rất lớn). Do đó để có thể dễ dàng truy xuất, quản lý các thông tin trên bộ nhớ ngoài ta cần đặt cho mỗi thông tin 1 cái tên tương ứng, đồng thời khi lưu trữ nội dung thông tin ta cũng cần gắn kèm những thuộc tính cần thiết cho sự quản lý của người sử dụng hoặc của hệ điều hành như kích thước của thông tin, ngày giờ tạo ra thông tin,... Tập hợp những thứ đó được gọi là tập tin. Như vậy tập tin là đơn vị lưu trữ thông tin trên bộ nhớ ngoài, mỗi tập tin bao gồm các thành phần: tên thông tin, nội dung của thông tin, kích thước của phần nội dung đó, và các thuộc tính cần thiết cho người sử dụng hoặc cho hệ điều hành.

Tóm lại, khái niệm tập tin không phải có sẵn trên máy tính mà được đưa ra bởi hệ điều hành (chính xác hơn là bởi những người làm ra phần mềm hệ điều hành), nhằm giúp cho việc truy xuất /quản lý thông tin trên bộ nhớ ngoài được dễ dàng thuận lợi hơn.

❖ Tên tập tin thường có 2 phần: phần tên (name) và phần mở rộng (extension). Phần tên là bắt buộc phải có của một tập tin, còn phần mở rộng thì có thể có hoặc không.

- Phần tên : Được đặt theo quy tắc : gồm chữ, số và một số ký tự đặc biệt như : \$, #, !, ~, (), {}, ^, & ...

+ Tên không được chứa các ký tự sau : , /, :, *, ?, ", < >, |

+ Tên nên mang tính gợi nhớ nội dung tập tin

- Phần mở rộng : Là phần đặc trưng cho từng loại chương trình (Word có phần mở rộng là .Doc, Excel có phần mở rộng là .Xls...)

◆ Đối với hệ điều hành MSDOS:

Phần tên không quá 8 ký tự

Phần mở rộng nếu có không quá 3 ký tự

Tên tập tin không chứa dấu cách (khoảng trắng)

Ví dụ : BaiTho.txt (tên đúng) ; VanBan.doc (tên đúng); BangLuong1.Xls (tên sai); BangLuong2.Xls (tên sai) ; BLuong1.Xls (tên đúng)

◆ Đối với hệ điều hành Windows:

Tên tập tin không quá 255 ký tự

Ví dụ: Bang Luong So 1.Xls; bang luong thang 2.xls

❖ Các thuộc tính của tập tin :

Read Only: Chỉ cho phép đọc, không cho phép sửa

Archive: Thuộc tính lưu trữ

System: Tập tin hệ thống

Hidden: tập tin ẩn

b) Thư mục :

Một khi số lượng tập tin được lưu trữ trên bộ nhớ ngoài đã lên tới 1 con số khá lớn thì nhất thiết phải đưa ra khái niệm thư mục. Khái niệm này cũng gần giống như hệ thống thư mục trong thư viện (khi đó mỗi tập tin có thể ví như 1 cuốn sách).

Nếu số sách không lộn trong thư viện không được tổ chức theo một trật tự hợp lý mà cứ để chung vào một chỗ thì khi muốn tìm 1 cuốn sách (theo 1 chủ đề nào đó mà ta chưa biết tên của cuốn sách đó chẳng hạn), ta phải nhìn vào từng cuốn một, và không phải thì nhìn tiếp qua cuốn khác cho đến khi tìm được. Cách làm đó có thể khiến ta phải mất 1 khoảng thời gian rất kinh khủng, nhất là khi cuốn sách muốn tìm không có trong thư viện đó thì ta vẫn phải xem hết tất cả mọi cuốn sách trong thư viện mới biết được là không có! Để khắc phục trở ngại đó người ta đã tổ chức hệ thống thư mục: danh mục tên sách được liệt kê theo từng chủ đề, trong mỗi chủ đề lại có thể có nhiều chủ đề con. Việc tổ chức phân cấp dạng cây như vậy chắc chắn sẽ giúp cho thời gian tìm kiếm 1 cuốn sách theo 1 chủ đề nào đó dù chưa biết tên sách cũng sẽ rất nhanh.

Nội dung các tập tin còn đa dạng hơn nội dung của các quyển sách (không chỉ là tài liệu văn bản bình thường mà còn có thể là hình ảnh, âm thanh, nhạc, phim, trò chơi, chương trình các loại...), và số lượng tập tin trên bộ nhớ ngoài cũng rất nhiều khiến cho việc tìm kiếm & sử dụng chúng cũng sẽ rất khó khăn nếu như ta cũng để chung vào 1 chỗ mà không tổ chức theo một trật tự hợp lý. Do đó tổ chức hệ thống thư mục phân cấp trên bộ nhớ ngoài là rất cần thiết, để giải quyết nhu cầu trên.

Tóm lại, khái niệm thư mục cũng được đưa ra bởi hệ điều hành, để việc tìm kiếm & sử dụng tập tin được hiệu quả. Mỗi thư mục có thể chứa các tập tin và các thư mục con bên trong (dĩ nhiên trong mỗi thư mục cũng có thể chỉ chứa toàn tập tin, hoặc chỉ chứa toàn thư mục con, hoặc đang là thư mục rỗng – không chứa gì cả).

Mỗi thư mục phải có tên như tên tập tin:

- Thư mục hiện hành (Current Directory): là thư mục đang làm việc tại đó.

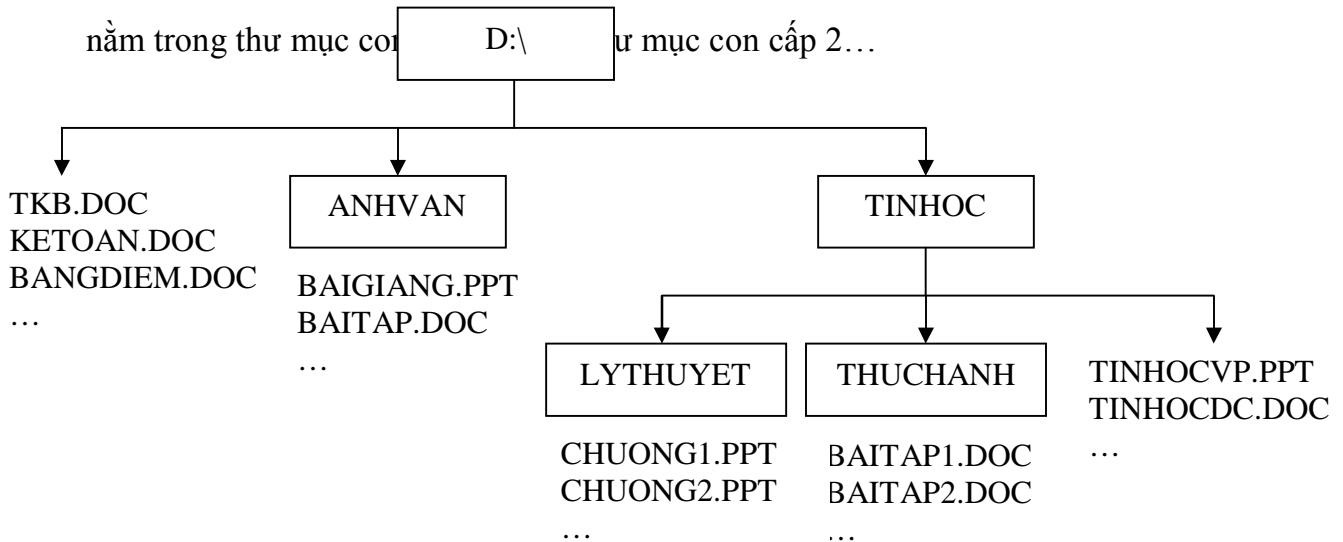
Tên viết tắt của thư mục hiện hành là “.”.

- Thư mục cha (Parent Directory): là thư mục trên thư mục hiện hành một cấp. Tên tắt của thư mục cha là “..”.

- Thư mục gốc (Root Directory): là thư mục chứa tất cả các tập tin và thư mục khác trên đĩa. Thư mục gốc có tên qui ước là dấu “\”. Thư mục gốc không được đổi tên hoặc xóa.

Tổ chức các thư mục trên một ổ đĩa có hình tượng như một cây gọi là cây thư mục (directory tree).

Các thư mục có thể trùng tên , nhưng phải ở khác thư mục mẹ hay thư mục gốc. Trong cùng một thư mục không có hai thư mục con có cùng tên , hoặc không có hai tập tin có cùng tên và cùng phần mở rộng. Nguyên tắc đặt tên thư mục giống như đặt tên tập tin , nhưng không có phần mở rộng. Các thư mục được phân cấp bậc : Thư mục nằm trong thư mục gốc gọi là thư mục con cấp 1 , các thư mục nằm trong thư mục con cấp 1 gọi là thư mục con cấp 2...



c) Đường dẫn :

Đường dẫn dùng để chỉ ra vị trí của tập tin hoặc thư mục trong một cây thư mục.

Đường dẫn là một dãy tên của các thư mục cách nhau bởi dấu “\”.

Ví dụ:

D:\TINHOC\TINHOCDAICUONG
 \THUCHANH\TINHOCDAICUONG
 ..\ANHVAN\BAITAP

Đường dẫn không bắt đầu bằng tên ổ đĩa thì được ngầm định là ổ đĩa hiện hành.

Đường dẫn bắt đầu bằng tên thư mục con thì được tính tương đối từ thư mục hiện hành.

Tên tập tin hoặc thư mục, nếu có chỉ đường dẫn thì cũng đặt cách nhau dấu “\”, nếu không chỉ đường dẫn thì được ngầm định là thư mục hiện hành.

2.3.3 Hệ thống quản lý file

Có nhiệm vụ tổ chức thông tin trên đĩa , cung cấp các phương tiện để người sử dụng có thể đọc ghi thông tin trên đĩa.

Tác dụng của hệ thống quản lý tập tin :

❖ Đối với Hệ điều hành:

- + Đảm bảo độc lập giữa phương pháp lưu trữ và phương pháp xử lý
- + Sử dụng bộ nhớ trên đĩa một cách hiệu quả
- + Tổ chức bảo vệ thông tin ở nhiều mức

❖ Đối với người sử dụng:

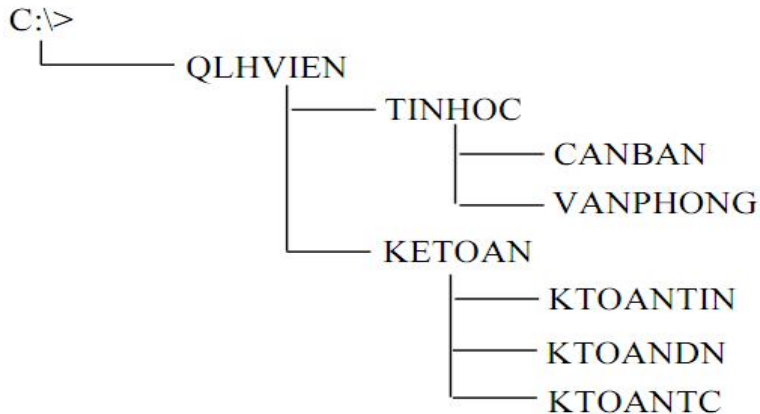
- + Xem nội dung thư mục, tập tin
- + Sao chép thư mục, tập tin
- + Xoá, đổi tên thư mục, tập tin

Đặc trưng của hệ thống quản lý tập tin :

- +Đảm bảo tốc độ truy xuất cao
- +Độc lập giữ thông tin và phương tiện mang thông tin , giữa phương pháp lưu trữ và phương pháp xử lý
- +Sử dụng bộ nhớ ngoài một cách có hiệu quả
- +Tổ chức bảo vệ thông tin . Hạn chế ảnh hưởng các lỗi kỹ thuật hoặc chương trình.

Thực hành chương 2

1. Tạo cây thư mục sau:



2. Trong thư mục tin học, tạo 2 tập tin danhsach1.txt và danhsach2.txt với nội dung tùy chọn
3. Nối 2 tập tin danhsach1.txt và danhsach2.txt thành danhsach.txt
4. Di chuyển tất cả các tập tin vừa tạo ở câu 2, 3 vào thư mục CANBAN
5. Đổi tên tập tin danhsach.txt trong thư mục CANBAN thành ds.txt
6. Xóa màn hình và xem nội dung thư mục CANBAN theo các cách sau:
 - Đứng tại thư mục gốc (Ổ đĩa C)
 - Đứng tại thư mục KETOAN
 - Đứng tại thư mục CANBAN
7. Tạo thêm thư mục LUUTRU trong ổ đĩa C.Sao chép tất cả các tập tin trong thư mục CANBAN vào thư mục LUUTRU
8. Xem lại ngày, giờ hệ thống nếu sai thì sửa lại
9. Xóa tập tin danhsach1.txt, danhsach2.txt trong thư mục LUUTRU
10. Xem nội dung tập tin ds.txt trong thư mục LUUTRU bằng cách đứng tại thư mục QLHVIEN
11. Xóa thư mục KTOANTC
12. Copy tất cả các tập tin có phần mở rộng là TXT từ thư mục gốc vào trong thư mục LUUTRU .
13. Xóa toàn bộ cây thư mục QLHVIEN

Chương 3 : HỆ ĐIỀU HÀNH WINDOWS

3.1 Tổng quan về Windows

Windows là một bộ chương trình do hãng Microsoft sản xuất. Từ version 3.0 ra đời vào tháng 5 năm 1990 đến nay, Microsoft đã không ngừng cải tiến làm cho môi trường này ngày càng được hoàn thiện.

Windows 95: vào cuối năm 1995, ở Việt nam đã xuất hiện một phiên bản mới của Windows mà chúng ta quen gọi là Windows 95. Những cải tiến mới của Windows 95 được liệt kê tóm tắt như sau:

- Giao diện với người sử dụng được thiết kế lại hoàn toàn nên việc khởi động các chương trình ứng dụng cùng các công việc như mở và lưu cất các tư liệu, tổ chức các tài nguyên trên đĩa và nối kết với các hệ phục vụ trên mạng - tất cả đều trở nên đơn giản và dễ dàng hơn.

- Cho phép đặt tên cho các tập tin dài đến 255 ký tự. Điều này rất quan trọng vì những tên dài sẽ giúp ta gọi nhớ đến nội dung của tập tin.

- Hỗ trợ Plug and Play, cho phép tự động nhận diện các thiết bị ngoại vi nên việc cài đặt và quản lý chúng trở nên đơn giản hơn.

- Hỗ trợ tốt hơn cho các ứng dụng Multimedia. Với sự tích hợp Audio và Video của Windows 95, máy tính cá nhân trở thành phương tiện giải trí không thể thiếu được.

- Windows 95 là hệ điều hành 32 bit, vì vậy nó tăng cường sức mạnh và khả năng vận hành lên rất nhiều.

- Trong Windows 95 có các công cụ đã được cải tiến nhằm chuẩn hóa, tối ưu hóa và điều chỉnh các sự cố. Điều này giúp bạn yên tâm hơn khi làm việc với máy vi tính trong môi trường của Windows 95.

Tóm lại, với những tính năng mới ưu việt và tích hợp cao, Windows 95 đã trở thành môi trường làm việc được người sử dụng ưa chuộng và tin dùng.

Windows 98, Windows Me: là những phiên bản tiếp theo của Windows 95, những phiên bản này tiếp tục phát huy và hoàn thiện những tính năng ưu việt của Windows 95 và tích hợp thêm những tính năng mới về Internet và Multimedia.

Windows NT 4.0, Windows 2000, Windows XP, Windows 2003: là những hệ điều hành được phát triển cao hơn, được dùng cho các cơ quan và doanh nghiệp. Giao diện của những hệ điều hành này tương tự như Windows 98/ Windows Me. Điểm khác biệt là những hệ điều hành này có tính năng bảo mật cao, vì vậy nó được sử dụng cho môi trường có nhiều người dùng.

Giáo trình này sẽ trình bày dựa vào hệ điều hành Windows XP.

3.2 Làm việc với Windows

3.2.1 Khởi động và thoát khỏi Windows XP

a) Khởi động Windows XP :

Windows XP được tự động khởi động sau khi bật máy. Sẽ có thông báo yêu cầu nhập vào tài khoản (User name) và mật khẩu (Password) của người dùng. Thao tác này gọi là đăng nhập (logging on).

Mỗi người sử dụng sẽ có một tập hợp thông tin về các lựa chọn tự thiết lập cho mình (như dáng vẽ màn hình, các chương trình tự động chạy khi khởi động

máy, tài nguyên/ chương trình được phép sử dụng, v.v...) gọi là user profile và được Windows XP lưu giữ lại để sử dụng cho những lần khởi động sau.

b) Đóng Windows XP :

Khi muốn thoát khỏi Windows XP, bạn phải đóng tất cả các cửa sổ đang mở. Tiếp theo bạn nhấn tổ hợp phím Alt + F4 hoặc chọn menu Start (nếu không nhìn thấy nút Start ở phía dưới bên góc trái màn hình thì bạn nhấn tổ hợp phím Ctrl + Esc) và chọn Turn Off Computer. Sau thao tác này một hộp thoại sẽ xuất hiện như hình bên.



Hình 3.1 : Đóng Window XP

Nếu bạn chọn Turn Off, ứng dụng đang chạy sẽ được đóng lại và máy sẽ tự động tắt.

Nếu vì một lý do nào đó mà máy tính không sẵn sàng để đóng (chưa lưu dữ liệu cho một ứng dụng hoặc sự trao đổi thông tin giữa hai máy nối mạng đang tiếp diễn v.v..) thì sẽ có thông báo để xử lý.

Chú ý: nếu không làm những thao tác đóng Windows như vừa nói ở trên mà tắt máy ngay thì có thể sẽ xảy ra việc thất lạc một phần của nội dung các tập tin dẫn đến trục trặc khi khởi động lại ở lần sử dụng tiếp theo.

3.2.2 Một vài thuật ngữ thường sử dụng

❖ Các biểu tượng (icon)

Biểu tượng là các hình vẽ nhỏ đặc trưng cho một đối tượng nào đó của Windows hoặc của các ứng dụng chạy trong môi trường Windows. Phía dưới biểu tượng là tên biểu tượng. Tên này mang một ý nghĩa nhất định, thông thường nó diễn giải cho chức năng được gán cho biểu tượng (ví dụ nó mang tên của 1 trình ứng dụng).



❖ Sử dụng chuột trong Windows :

Chuột là thiết bị không thể thiếu khi làm việc trong môi trường Windows XP. Con trỏ chuột (mouse pointer) cho biết vị trí tác động của chuột trên

màn hình. Hình dáng của con trỏ chuột trên màn hình thay đổi theo chức năng và chế độ làm việc của ứng dụng. Khi làm việc với thiết bị chuột bạn thường sử dụng các thao tác cơ bản sau :

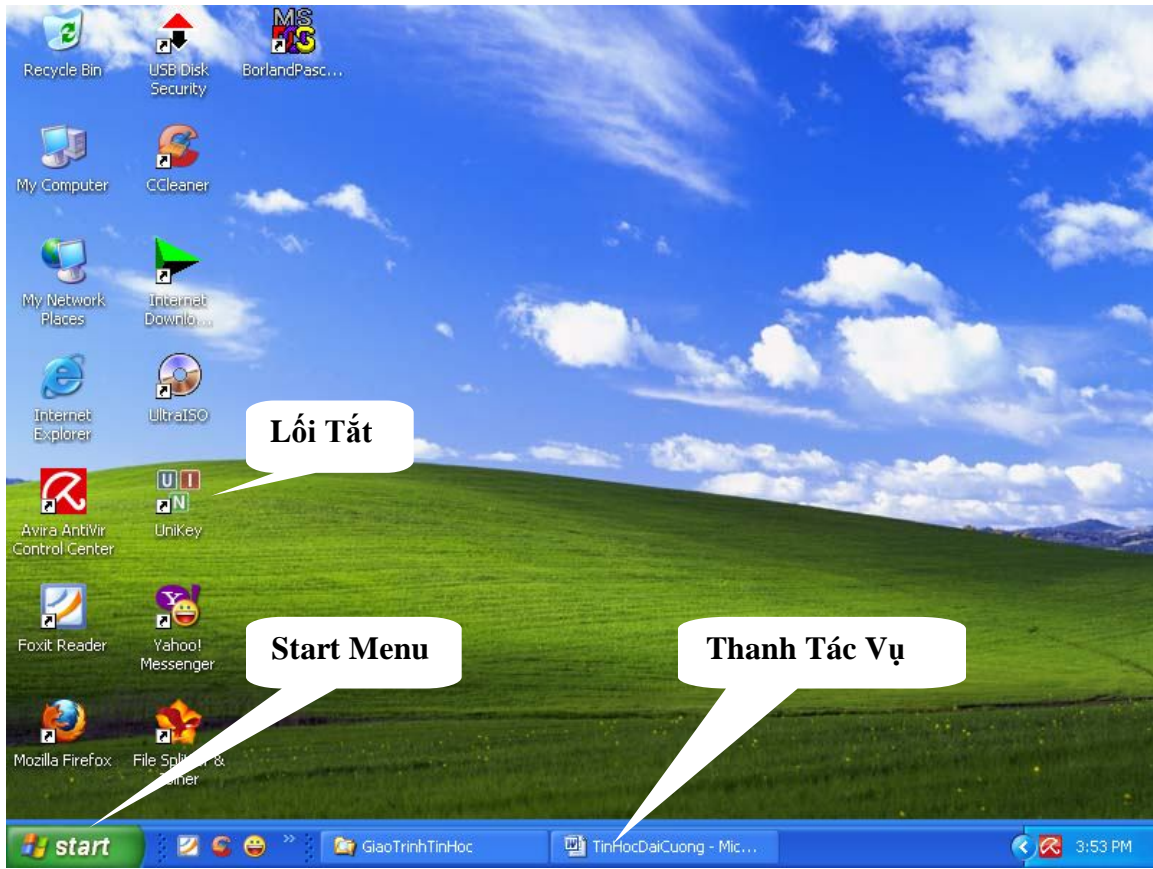
- + **Point:** trỏ chuột trên mặt phẳng mà không nhấn nút nào cả.
- + **Click:** nhấn nhanh và thả nút chuột trái. Dùng để lựa chọn thông số, đối tượng hoặc câu lệnh.
- + **Double Click (D_Click):** nhấn nhanh nút chuột trái hai lần liên tiếp. Dùng để khởi động một chương trình ứng dụng hoặc mở thư mục/ tập tin.
- + **Drag** (kéo thả): nhấn và giữ nút chuột trái khi di chuyển đến nơi khác và buông ra. Dùng để chọn một khối văn bản, để di chuyển một đối tượng trên màn hình hoặc mở rộng kích thước của cửa sổ...
- + **Right Click (R_Click):** nhấn nhanh và thả nút chuột phải. Dùng mở menu tương ứng với đối tượng để chọn các lệnh thao tác trên đối tượng đó.

Chú ý:

- Đa số chuột hiện nay có bánh xe trượt hoặc nút đẩy ở giữa dùng để cuộn màn hình làm việc được nhanh hơn và thuận tiện hơn.
- Trong Windows các thao tác được thực hiện mặc nhiên với nút chuột trái, vì vậy để tránh lặp lại, khi nói Click (nhấn chuột) hoặc D_Click (nhấn đúp chuột) thì được ngầm hiểu đó là nút chuột trái. Khi nào cần thao tác với nút chuột phải sẽ mô tả rõ ràng.

3.2.3 Giới thiệu màn hình nền (Desktop) của Windows XP

❖ **Màn hình nền :**



Hình 3.2 Màn hình nền Desktop của WinXP

Nằm cuối màn hình là thanh tác vụ (Taskbar). Bên trái màn hình là biểu tượng My Computer, My Network Places, Recycle Bin, ...

Các biểu tượng có mũi tên màu đen nhỏ (ở góc dưới bên trái) gọi là lối tắt (shortcut).

❖ Những biểu tượng trên màn hình :

– My Computer :

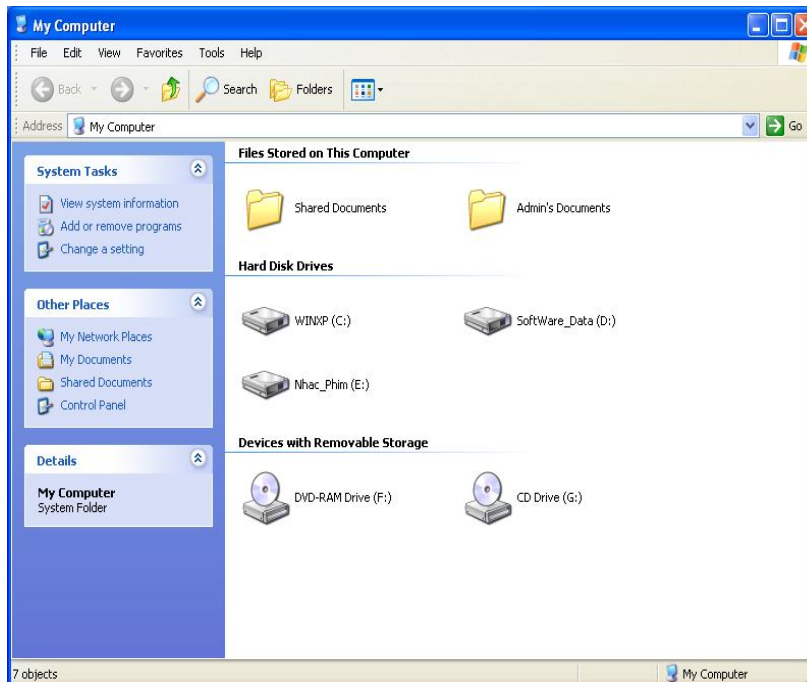
Biểu tượng này cho phép duyệt nhanh tài nguyên trên máy tính. Khi mở My Computer (bằng thao tác D_Click hoặc R_Click/ Open trên biểu tượng của nó), cửa sổ xuất hiện như hình 3.3.

+ Cửa sổ bên trái :

- *System Tasks*: cho phép chọn thực hiện một số công việc hệ thống của máy.
- *Other Places*: cho phép chọn các thành phần khác trong máy.

+ Cửa sổ bên phải :

Theo ngầm định cửa sổ này chứa biểu tượng của tất cả các ổ đĩa mềm, ổ đĩa cứng cục bộ, ổ đĩa CD, ... Khi D_Click trên các biểu tượng trong cửa sổ này sẽ có các cửa sổ cấp nhỏ hơn được mở. Do đó, bằng cách mở dần các cửa sổ từ ngoài vào trong bạn có thể duyệt tất cả tài nguyên chứa trong máy tính.



Hình 3.3 : Cửa sổ My Computer

– My Network Places :

Nếu mở cửa sổ **My Network Places** bạn sẽ thấy tên và các tài nguyên của các máy tính trong mạng máy tính cục bộ (LAN) của bạn. Từ đây bạn có thể truy cập các tài nguyên đã được chia sẻ mà bạn đã được cấp quyền sử dụng.

– Recycle Bin :

Recycle Bin là nơi lưu trữ tạm thời các tập tin và các đối tượng đã bị xóa. Những đối tượng này chỉ thật sự mất khi bạn xóa chúng trong cửa sổ Recycle Bin hoặc R_Click vào biểu tượng Recycle Bin rồi chọn Empty Recycle Bin. Nếu muốn phục hồi các tập tin hoặc các đối tượng đã bị xóa, bạn chọn đối tượng cần phục hồi trong cửa sổ Recycle Bin, sau đó R_Click/ Restore.

– Folder :

Folder được gọi là “tập hồ sơ” hay “biểu tượng nhóm” hay “thư mục”. Folder là nơi quản lý các Folder khác (cấp thấp hơn) và các tập tin.

– Menu Start :

Khi Click lên nút Start trên thanh Taskbar, thực đơn Start sẽ được mở và sẵn sàng thi hành các chương trình ứng dụng. Ngoài ra trên thực đơn này bạn còn có thể thực hiện các thao tác tìm kiếm và định cấu hình cho máy tính.

– Các lối tắt (biểu tượng chương trình - Shortcuts)

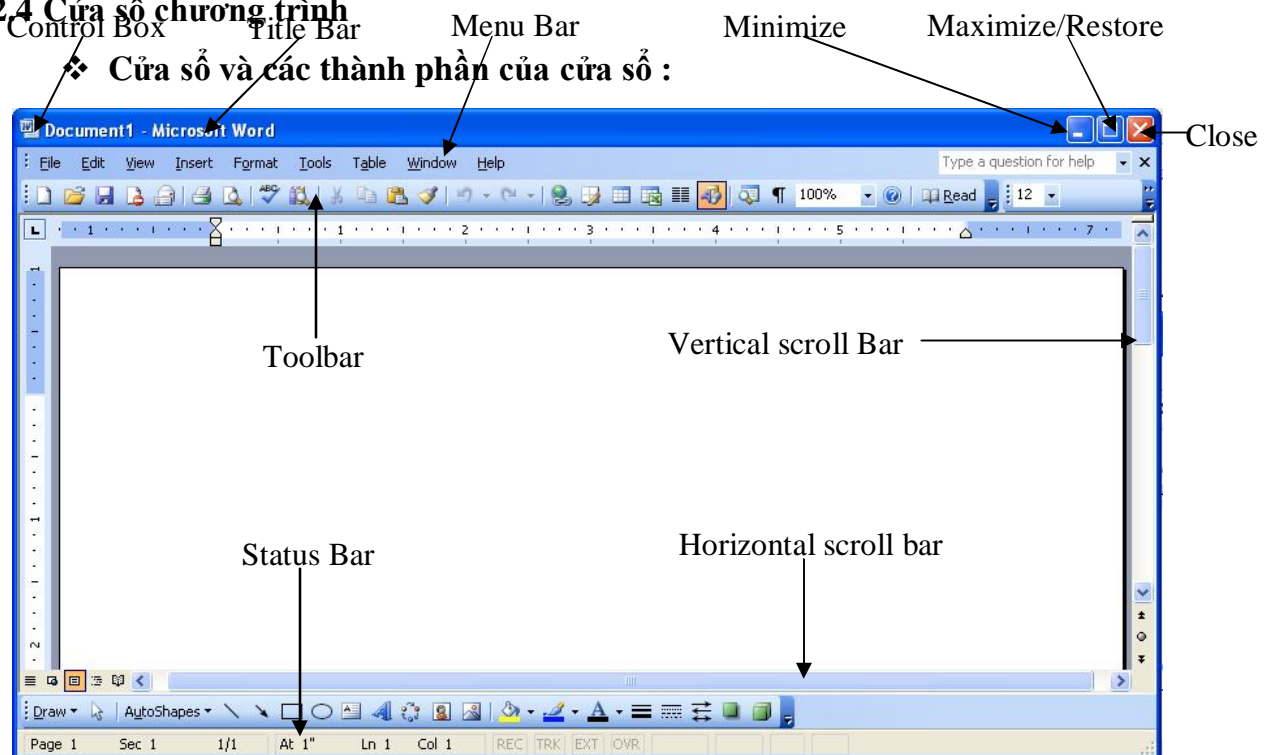
Các lối tắt giúp bạn truy nhập nhanh một đối tượng nào đó, ví dụ một chương trình, một đĩa cứng, một thư mục v.v... Để mở 1 đối tượng, bạn D_Click trên Shortcut của nó hoặc R_Click/Open.

– Menu đối tượng

Trong Windows XP khi bạn R_Click trên một biểu tượng của một đối tượng, một menu tương ứng với đối tượng đó sẽ được mở ra để bạn chọn các lệnh trên đối tượng đó. Trong các phần tiếp theo, những menu như vậy sẽ được gọi là menu đối tượng.

3.2.4 Cửa sổ chương trình

❖ Cửa sổ và các thành phần của cửa sổ :



Hình 3.4: Cửa sổ chương trình

❖ Các thao tác trên một cửa sổ :

- Di chuyển cửa sổ: Drag thanh tiêu đề cửa sổ (Title bar) đến vị trí mới.

- Thay đổi kích thước của cửa sổ: Di chuyển con trỏ chuột đến cạnh hoặc góc cửa sổ, khi con trỏ chuột biến thành hình mũi tên hai chiều thì Drag cho đến khi đạt được kích thước mong muốn.

- Phóng to cửa sổ ra toàn màn hình: Click lên nút Maximize

- Phục hồi kích thước trước đó của cửa sổ: Click lên nút Restore

- Thu nhỏ cửa sổ thành biểu tượng trên Taskbar: Click lên nút Minimize

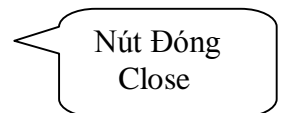
- Chuyển đổi giữa các cửa sổ của các ứng dụng đang mở: Để chuyển đổi giữa các ứng dụng nhấn tổ hợp phím Alt + Tab hoặc chọn ứng dụng tương ứng trên thanh Taskbar.

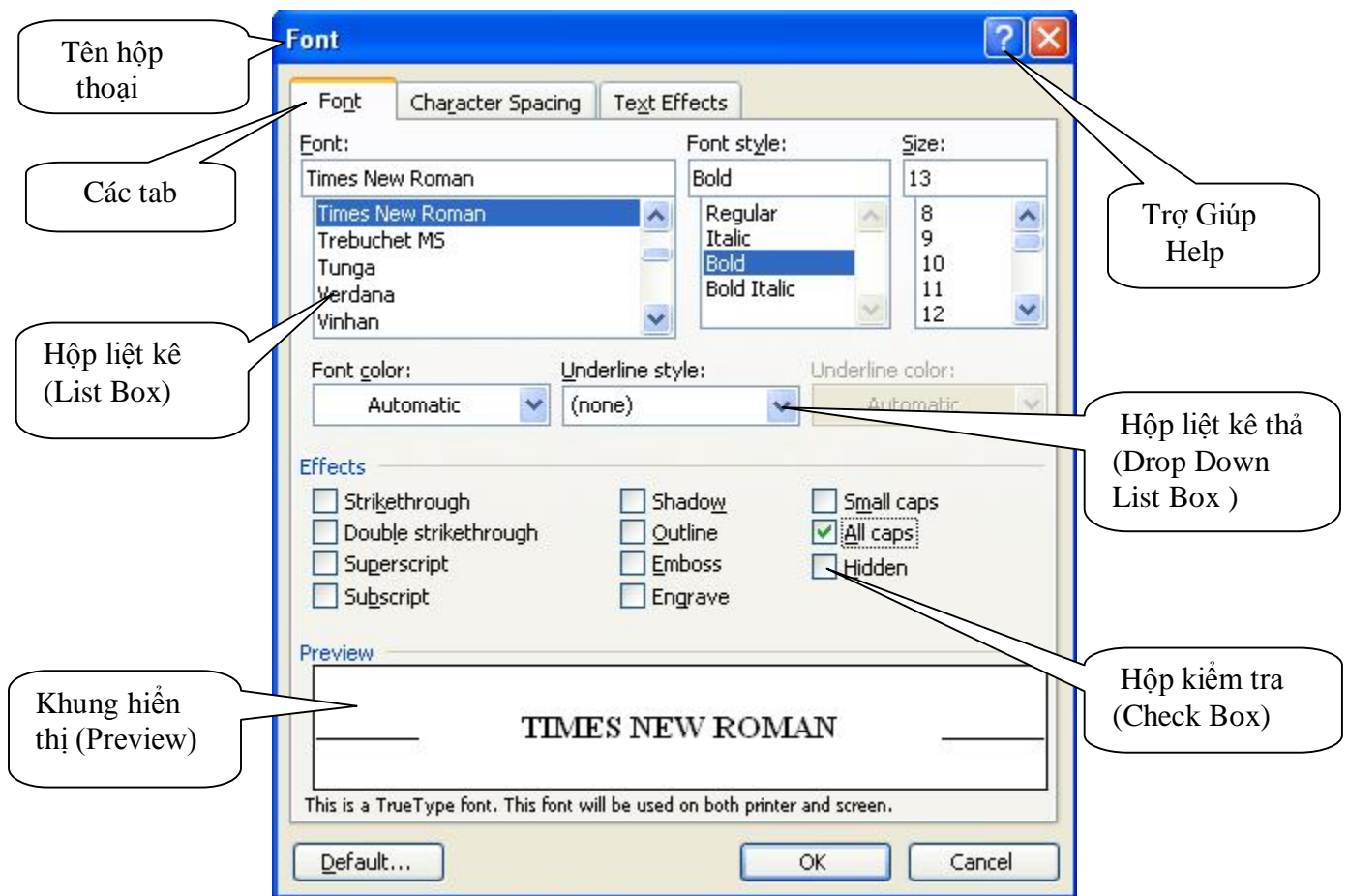
- Đóng cửa sổ: Click lên nút Close của cửa sổ hoặc nhấn tổ hợp phím Alt + F4.

3.2.5 Hộp thoại (Dialogue Box)

Trong khi làm việc với Windows và các chương trình ứng dụng chạy dưới môi trường Windows bạn thường gặp những hộp hội thoại. Các hộp thoại này xuất hiện khi nó cần thêm những thông số để thực hiện lệnh theo yêu cầu của bạn.

❖ Các thành phần của hộp thoại :

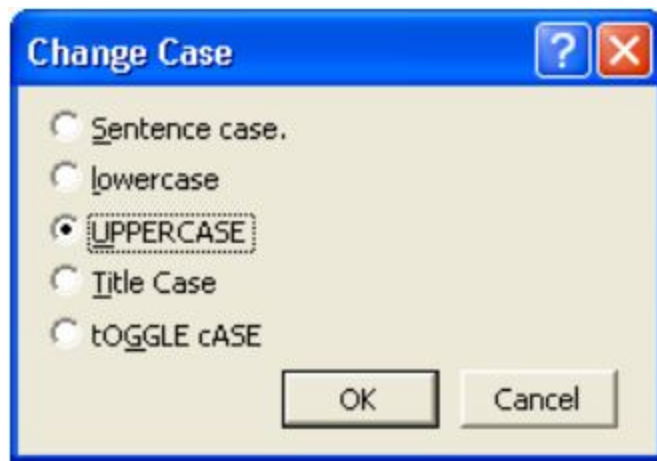




Hình 3.5 : Các thành phần của hộp thoại

Thông thường, trên một hộp hội thoại sẽ có các thành phần sau:

- Hộp văn bản (Text box): dùng để nhập thông tin.
- Hộp liệt kê (List box): liệt kê sẵn một danh sách có các mục có thể chọn lựa, nếu số mục trong danh sách nhiều không thể liệt kê hết thì sẽ xuất hiện thanh trượt để cuộn danh sách.
- Hộp liệt kê thả (Drop down list box/ Combo box): khi nhấp chuột vào nút thả thì sẽ bung xuống một danh sách, trong đó liệt kê các mục và cho chọn.
- Hộp kiểm tra (Check box): cho phép chọn một hoặc nhiều mục không loại trừ lẫn nhau.
- Nút tùy chọn (Option button): bắt buộc phải chọn một



trong số các mục. Những lựa chọn này là loại trừ lẫn nhau.

- Nút lệnh (Command button): dùng để xác nhận lệnh cần thực thi.

Các nút lệnh thông dụng:

- o **OK:** thực hiện lệnh theo thông số đã chọn.
- o **Close:** giữ lại các thông số đã chọn và đí Hình 3.6 Các tùy chọn
- o **Cancel** (hay nhấn phím Esc): không thực hiện lệnh và đóng hộp thoại.
- o **Apply:** áp dụng các thông số đã chọn.
- o **Default:** đặt mặc định theo các thông số đã chọn.

3.2.6 Sao chép dữ liệu trong Windows

Trong Windows việc sao chép dữ liệu trong một ứng dụng hoặc giữa các ứng dụng được thực hiện thông qua bộ nhớ đệm (Clipboard). Tại một thời điểm, bộ nhớ đệm chỉ chứa một thông tin mới nhất. Khi một thông tin khác được đưa vào bộ nhớ đệm thì thông tin trước đó sẽ bị xoá. Khi thoát khỏi Windows thì nội dung trong bộ nhớ đệm cũng bị xoá.

Khi muốn sao chép dữ liệu từ một vị trí nào đó để dán vào một vị trí khác, cần thực hiện bốn thao tác theo trình tự sau đây:

- Xác định đối tượng cần sao chép
- Thực hiện lệnh sao chép Edit/ Copy hoặc nhấn tổ hợp phím Ctrl + C hoặc R_Click/Copy để chép đối tượng vào bộ nhớ đệm.
- Xác định vị trí cần chép tới.
- Thực hiện lệnh dán Edit/ Paste hoặc nhấn tổ hợp phím Ctrl + V hoặc R_Click/Paste để dán dữ liệu từ bộ nhớ đệm vào vị trí cần chép.

3.2.7 Cách khởi động và thoát khỏi chương trình

❖ Khởi động chương trình ứng dụng

Có nhiều cách để khởi động một chương trình ứng dụng trong Windows:

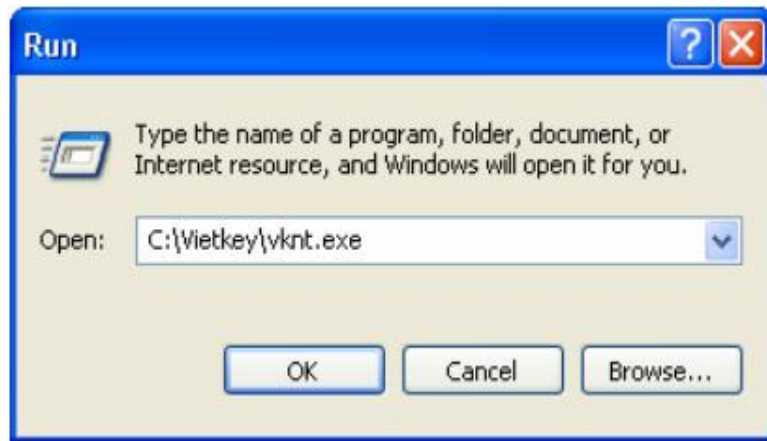
◆ Khởi động từ Start Menu :

Chọn Start/ Programs[/ Nhóm chương trình]/ Tên chương trình ứng dụng

◆ Khởi động bằng lệnh Run

Click vào nút Start trên thanh Taskbar, chọn lệnh Run sẽ xuất hiện hộp thoại Run như hình bên.

- Nhập đầy đủ đường dẫn và tên tập tin chương trình vào mục Open hoặc Click chọn nút Browse để chọn chương trình cần khởi động.



- Chọn OK để khởi động chương trình.

Hình 3.7 : Hộp thoại lệnh Start/Run

◆ Dùng Shortcut để khởi động các chương trình

D_Click hoặc R_Click/ Open vào Shortcut của các ứng dụng mà bạn muốn khởi động. Các Shortcut có thể được đặt trong một Folder hoặc ngay trên màn hình nền. Shortcut thực chất là một con trỏ đến đối tượng (hoặc có thể coi là một đường dẫn), vì thế bạn có thể xóa nó mà không ảnh hưởng đến chương trình ứng dụng.

◆ Khởi động từ các Folder

Khi tên của một chương trình ứng dụng không hiện ra trên menu Start thì cách tiện lợi nhất để bạn khởi động nó là mở từ các Folder, D_Click hoặc R_Click/ Open trên biểu tượng của chương trình ứng dụng cần mở hoặc trên biểu tượng của một tập tin tương ứng.

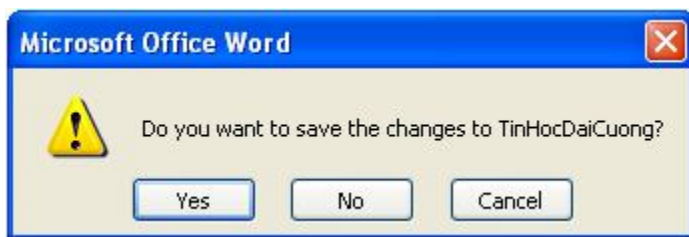
❖ Thoát khỏi chương trình ứng dụng

Để thoát khỏi một ứng dụng ta có thể dùng 1 trong các cách sau đây:

- Nhấn tổ hợp phím Alt + F4
- Click vào nút Close (ở góc trên bên phải của thanh tiêu đề).
- Chọn menu File/ Exit.

- D_Click lên nút Control Box (ở góc trên bên trái của thanh tiêu đề).
- Click lên nút Control Box. Click chọn Close.

Khi đóng 1 ứng dụng, nếu dữ liệu của ứng dụng đang làm việc chưa được lưu lại thì nó sẽ hiển thị hộp thoại nhắc nhở việc xác nhận lưu dữ liệu. Thông thường có 3 chọn lựa:



Hình 3.8 : Thông báo nhắc nhở lưu dữ liệu

- **Yes:** lưu dữ liệu và thoát khỏi chương trình ứng dụng.
- **No:** thoát khỏi chương trình ứng dụng mà không lưu dữ liệu.
- **Cancel:** hủy bỏ lệnh, trở về chương trình ứng dụng.

❖ Tự động chạy một ứng dụng khi khởi động Windows

Trên thực tế có 1 số chương trình bạn muốn nó khởi động ngay khi bắt đầu phiên làm việc với Windows. Giả sử bạn thường xuyên phải soạn thảo văn bản trong môi trường Windows XP và vì thế bạn muốn chương trình gõ tiếng Việt Vietkey phải được tự động mở ngay khi vừa khởi động Windows. Muốn vậy bạn phải tạo shortcut cho tập tin Vietkey.exe và đặt nó trong biểu tượng nhóm Startup.

3.2.8 Menu Documents

Trong quá trình thực hiện, Windows XP ghi nhận lại các tập tin vừa mới dùng gần nhất và lưu tên các tập tin này trong một menu con (Documents) của menu Start. Để mở một tập tin vừa mới dùng bạn chọn lệnh Start/ Documents và Click mở.

Để xóa nội dung trong menu Documents ta thực hiện:

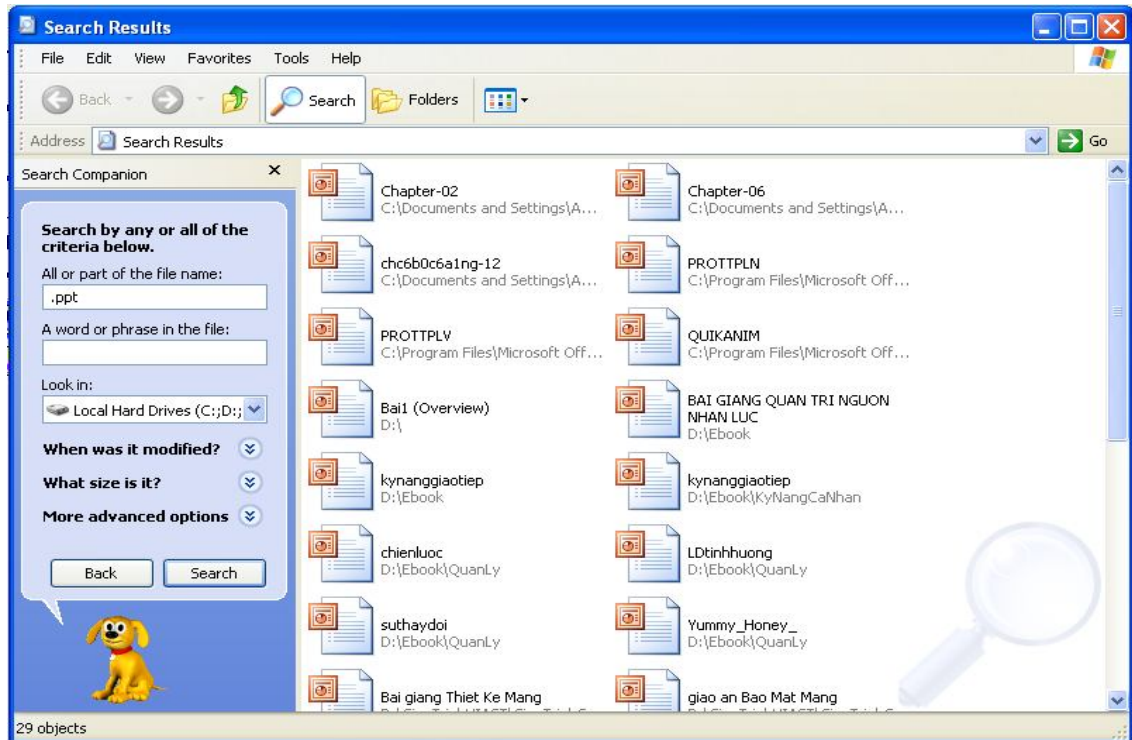
Start/ Settings/ Taskbar/ Start Menu Programs/ Clear

3.2.9 Tìm kiếm dữ liệu

Chức năng này cho phép tìm kiếm các tập tin, các thư mục và cả tên của các máy tính trên mạng LAN. Sau khi đã tìm thấy đối tượng, bạn có thể làm việc trực tiếp với kết quả tìm kiếm trong cửa sổ Search Results.

❖ Tìm kiếm tập tin và thư mục

Chọn lệnh Start/ Search/ For Files or Folders, sẽ xuất hiện cửa sổ Search Results, bạn Click chọn **All files and folders**.



Hình 3.9 : Cửa sổ Search Results

All or part of the file name: nhập tên thư mục hay tập tin cần tìm, có thể sử dụng ký tự đại diện * và ?.

A word or phrase in the file: nhập từ/ cụm từ trong nội dung tập tin cần tìm.

Look in: nơi tìm kiếm, bạn có thể nhập vào tên của ổ đĩa, đường dẫn mà từ đó việc tìm kiếm sẽ được thực hiện. Theo ngầm định, Windows tìm kiếm cả trong đĩa hoặc trong thư mục đã chỉ định trong khung Look in cùng với mọi thư mục con của nó.

Ngoài ra ta có thể thay đổi các lựa chọn để có thể tìm nhanh và chính xác hơn với các thuộc tính như sau:

❖ When was it modified?

Cho phép tìm kiếm các tập tin và thư mục dựa theo ngày sửa đổi.

Trong mục này các tùy chọn áp dụng cho những tập tin và thư mục đã được tạo ra hoặc được sửa đổi trong một khoảng thời gian nào đó. Chọn khai báo thời gian theo một trong các lựa chọn với ý nghĩa:

- Don't remember: không xác định thời gian (tìm tất cả).
- Within the last week: tìm các tập tin trong vòng 1 tuần gần đây nhất.
- Past month: tìm các tập tin trong vòng 1 tháng gần đây nhất.
- Within the past year: tìm các tập tin trong vòng 1 năm gần đây nhất.
- Specify dates: tìm các tập tin trong một

khoảng thời gian xác định. Có thể chọn theo ngày tạo (Created Date), ngày cập nhật (Modified Date), hay ngày truy cập (Accessed Date).



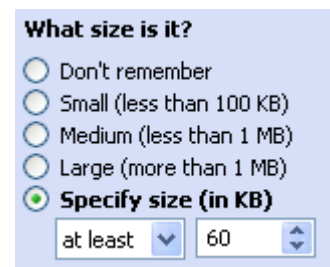
Hình 3.10: Tìm theo thời gian

❖ What size is it?

Cho phép tìm kiếm các tập tin và thư mục dựa theo kích thước tập tin.

Trong mục này các tùy chọn áp dụng tìm những tập tin và thư mục có kích thước được xác định trong một khoảng nào đó. Chọn khai báo kích thước theo một trong các lựa chọn với ý nghĩa:

- Don't remember: không xác định kích thước (tìm tất cả).
- Small (less than 100 KB): tìm các tập tin có kích thước nhỏ (dưới 100 KB).
- Midium (less than 1 MB): tìm các tập tin có kích thước vừa phải (dưới 1 MB).
- Large (more than 1 MB): tìm các tập tin có



Hình 3.11: Tìm theo kích thước

kích thước lớn (trên 1 MB).

– Specify size (in KB): tìm các tập tin có kích thước tối thiểu (at least) hay tối đa (at most) trong một giới hạn nào đó.

❖ More advanced options?

Cho phép thay đổi một số tùy chọn nâng cao khác.

– Type of file: kiểu tập tin cần tìm (tập tin văn bản, hình ảnh, bảng tính, thư mục,...).

– Search system folders: tìm/ không tìm trong thư mục hệ thống.

– Search hidden files and folders: tìm/ không tìm tập tin/ thư mục ẩn.



Hình 3.12 : Các tùy chọn khác

– Search subfolders: tìm/ không tìm trong thư mục con.

– Case Sensitive: phân biệt/ không phân biệt chữ hoa/ thường.

– Search tape backup: tìm/ không tìm trong đĩa dự phòng.

Sau khi khai báo xong các dữ liệu để tìm kiếm, Click nút Search, chương trình sẽ tiến hành tìm và thông báo kết quả.

❖ Làm việc với các kết quả tìm kiếm

Sau khi tìm kiếm, chương trình đưa ra kết quả tìm kiếm ở phía bên phải của hộp thoại Search Results. Bạn có thể làm việc với cửa sổ kết quả như với một cửa sổ tập hồ sơ thông thường (như thay đổi thứ tự sắp xếp hoặc dùng các lệnh của menu View để hiển thị các kiểu thể hiện khác nhau, kiểm tra tính chất của các đối tượng trong danh sách kết quả v.v...). Bạn cũng có thể khởi động một đối tượng vừa tìm thấy bằng cách D_Click trên tên của đối tượng đó.

3.2.10 Thay đổi cấu hình máy tính

Windows XP cho phép bạn thay đổi cấu hình cho phù hợp với công việc hoặc sở thích của bạn thông qua bảng điều khiển Control Panel (vào menu Start, chọn menu Settings/ Control Panel).

Từ bảng điều khiển Control Panel bạn có thể thiết lập cấu hình cho hệ thống, thay đổi ngày giờ, cài đặt thêm Font chữ, thiết bị phần cứng, phần mềm mới hoặc loại bỏ chúng đi khi không còn sử dụng nữa.

a) Cài đặt và loại bỏ Font chữ :

Font chữ là sự thể hiện các dạng khác nhau của ký tự. Ngoài các Font chữ có sẵn ta có thể cài đặt thêm những Font chữ khác hoặc loại bỏ các Font chữ không sử dụng. Muốn cài đặt hay loại bỏ các Font chữ, ta chọn lệnh *Start/ Settings/ Control Panel/ Fonts*, xuất hiện cửa sổ Fonts như hình 3.13.

❖ Xoá bỏ font chữ :

- Chọn những Font cần xóa bỏ.
- Chọn File/ Delete (hoặc nhấn phím Delete).



Hình 3.13 : Cửa sổ Fonts

❖ Thêm font chữ mới

Từ cửa sổ Fonts, chọn lệnh File/Install New Font, xuất hiện hộp thoại Add Fonts. Trong hộp thoại này, ta chỉ ra nơi chứa các Font nguồn muốn cài thêm bằng cách chọn tên ổ đĩa và Folders chứa các tập tin Font chữ, sau Cửa sổ Fonts đó chọn các tên Font và Click OK.

a) Thay đổi thuộc tính của màn hình

Chọn lệnh **Start/ Settings/ Control Panel/ Display** hoặc **R_Click** trên **màn hình** nền (Desktop), chọn Properties. Xuất hiện cửa sổ Display Properties (Hình 3.14) với các thành phần như sau:

❖ **Desktop:**

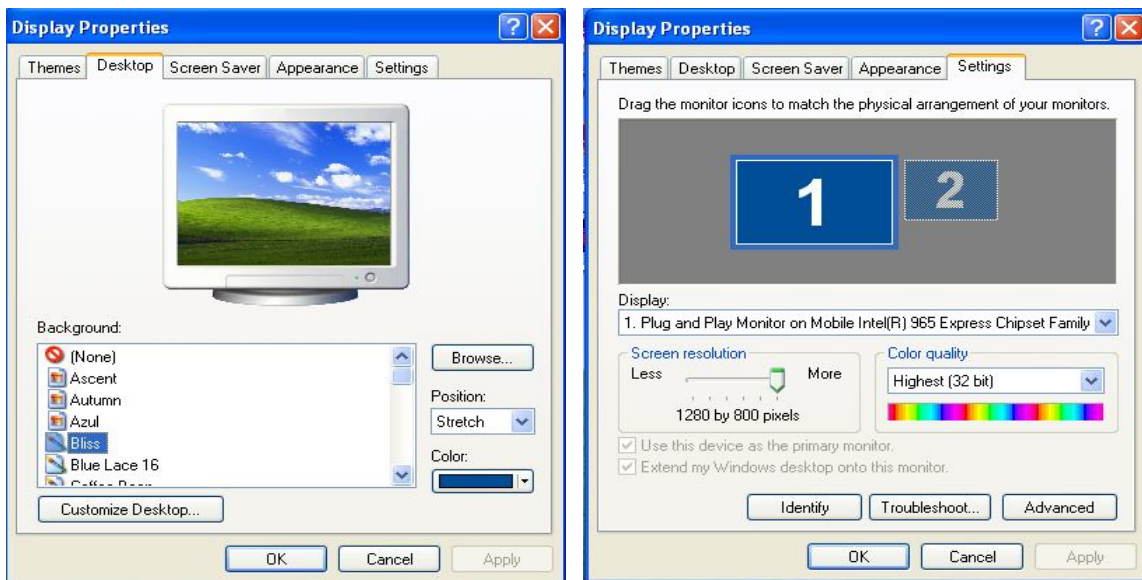
Chọn ảnh nền cho Desktop bằng cách Click chọn các ảnh nền có sẵn hoặc Click vào nút Browse để chọn tập tin ảnh không có trong danh sách những ảnh có sẵn.

❖ **Screen Saver:**

Thiết lập chế độ bảo vệ màn hình, cho phép chọn các chương trình Screen Saver đã được cài sẵn trong máy và cho phép thiết lập mật khẩu để thoát khỏi chế độ Screen Saver. Chỉ khi nào bạn nhập đúng mật khẩu thì chương trình Screen Saver mới cho phép bạn trở về chế độ làm việc bình thường.

❖ **Appearance:**

Thay đổi màu sắc, Font chữ và cỡ chữ của các Menu, Shortcut, Title bar.



Hình 3.14 : Cửa sổ Display Properties

❖ **Settings:**

Thay đổi chế độ màu và độ phân giải của màn hình.

- Chế độ màu càng cao thì hình ảnh càng đẹp và rõ nét. Các chế độ màu phổ biến là 256 màu, 64.000 màu (16 bits) , 16 triệu màu (24 bits). Chế độ màu

trên mỗi máy tính có thể khác nhau tùy thuộc vào dung lượng bộ nhớ của card màn hình.

- Độ phân giải càng lớn thì màn hình càng hiển thị được nhiều thông tin nhưng các đối tượng trên màn hình sẽ thu nhỏ lại. Các chế độ phân giải màn hình thông dụng là 640x480, 800x600, 1024x768. Tùy theo loại màn độ phân giải khác nhau.

c) Cài đặt và loại bỏ chương trình :

Để cài đặt các chương trình mới hoặc loại bỏ các chương trình không còn sử dụng bạn nhấn đúp chuột vào biểu tượng Add or Remove Programs trong cửa sổ Control Panel, xuất hiện hộp thoại như hình dưới đây:



Hình 3.15 : Nhóm Change or Remove Programs trong cửa sổ Control Panel, xuất hiện hộp thoại như hình dưới đây:

Nhóm **Change or Remove Programs**: cho phép cập nhật hay loại bỏ chương trình ứng dụng có sẵn.

- Chọn chương trình muốn cập nhật/ loại bỏ.

- Chọn Change để cập nhật chương trình hay Remove để loại bỏ khi không cần sử dụng nữa.

Nhóm **Add new Programs**: Cho phép cài đặt thêm chương trình ứng dụng



Hình 3.16 : Chọn tập tin cài đặt

hoặc cập nhật lại Windows (trực tiếp qua Internet).

- Chọn **CD or Floppy** để cài đặt chương trình ứng dụng mới, khi đó Windows sẽ yêu cầu bạn chỉ ra nơi chứa chương trình mà bạn muốn cài đặt, thường các tập tin này sẽ có tên là Setup.exe hoặc Install.exe. Bạn có thể nhập tên và đường dẫn của tập tin này vào hộp văn bản Open hoặc nhấn chuột vào nút Browse để chỉ ra. Chọn tập tin cài đặt tập tin đó (Hình 3.16).

- Chọn **Windows Update** để cập nhật lại hệ điều hành Windows (thêm, cập nhật các chức năng mới).

Nhóm **Add/ Remove**



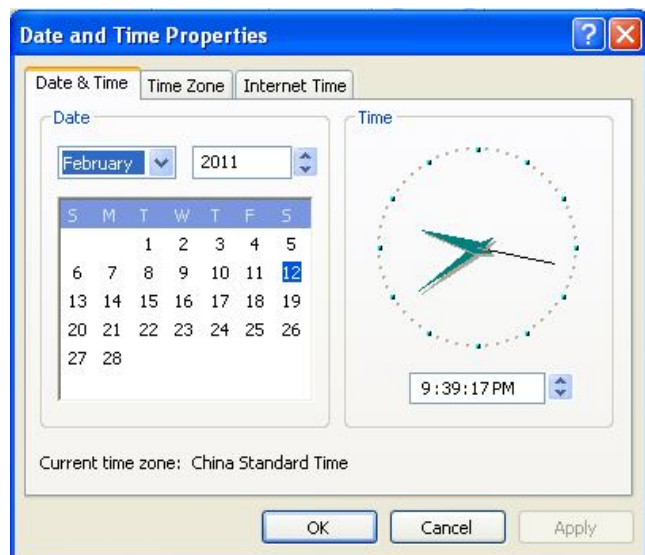
Windows components: cho phép cài đặt hoặc gỡ bỏ các thành phần trong hệ điều hành Windows (thông qua dữ liệu trong đĩa CD/ đĩa cứng).

d) Cấu hình ngày, giờ cho hệ thống

Bạn có thể thay đổi ngày giờ của hệ thống bằng cách Click lên biểu tượng đồng hồ trên thanh Taskbar hoặc chọn lệnh Start/ Settings/ Control Panel, chọn nhóm Date/Time

- **Date & Time:** thay đổi ngày, tháng, năm, giờ, phút, giây.

- **Time Zone:** thay đổi múi giờ, cho phép chỉnh lại các giá trị múi giờ theo khu vực hoặc tên các thành phố lớn.



Hình 3.18 : Định ngày giờ

- **Internet Time:** cho phép đồng bộ hoá theo giờ của máy chủ Internet.

e) Thay đổi thuộc tính của bàn phím và chuột

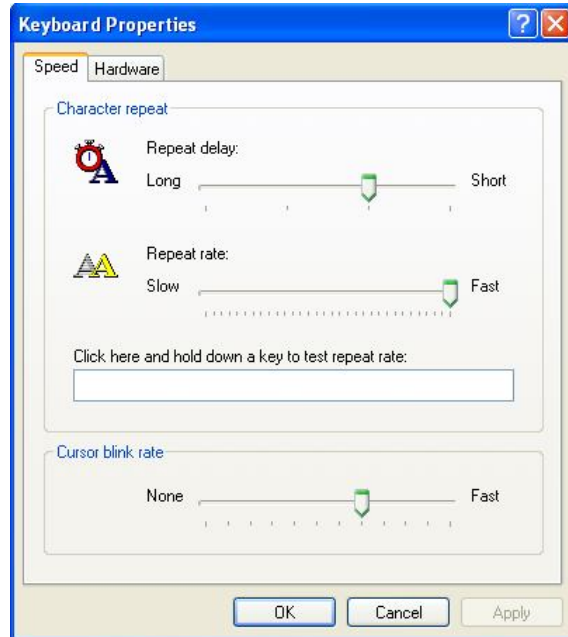
Thay đổi thuộc tính của bàn phím:

Lệnh **Start/ Settings/ Control Panel/ KeyBoard** cho phép thay đổi tốc độ bàn phím.

- **Repeat delay:** thay đổi thời gian trễ cho lần lặp lại đầu tiên khi nhấn và giữ phím.

- **Repeat rate:** thay đổi tốc độ lặp lại khi nhấn và giữ một phím.

- **Cursor blink rate:** thay đổi tốc độ của dấu nháy.



Hình 3.19 : Thay đổi thuộc tính của bàn phím

Thay đổi thuộc tính của thiết bị chuột:

Lệnh **Start/ Settings/ Control Panel/ Mouse** cho phép thay đổi thuộc tính của thiết bị chuột như tốc độ, hình dáng con trỏ chuột...

- **Lớp Buttons:** thay đổi chức năng giữa chuột trái và chuột phải (thuận tay trái hay phải) và tốc độ nhấp đúp chuột.

- **Lớp Pointers:** cho phép chọn hình dạng trỏ chuột trong các trạng thái làm việc.

- **Lớp Pointer Options:**

cho phép thay đổi tốc độ và hình dạng trỏ chuột khi rê hoặc kéo chuột.



Hình 3.20 : Thay đổi thuộc tính của thiết bị chuột

f) Thay đổi thuộc tính vùng (Regional Settings)

Bạn có thể thay đổi các thuộc tính như định dạng tiền tệ, đơn vị đo lường... bằng cách chọn lệnh Start/ Settings/ Control Panel/ Regional and Language Options.

- Lớp **Regional Options**: Thay đổi thuộc tính vùng (Mỹ, Pháp,...), việc chọn một vùng nào đó sẽ kéo theo sự thay đổi các thuộc tính của Windows.

- Click chọn **Customize**, cửa sổ Customize Regional Options xuất hiện cho phép thay đổi quy ước về số, tiền tệ, thời gian, ngày tháng.

➤ **Number**: Thay đổi định dạng số, cho phép định dạng việc hiển thị giá trị số:

- Decimal symbol: Thay đổi ký hiệu phân cách hàng thập phân.

- No. of digits after decimal: Thay đổi số các số lẻ ở phần thập phân.

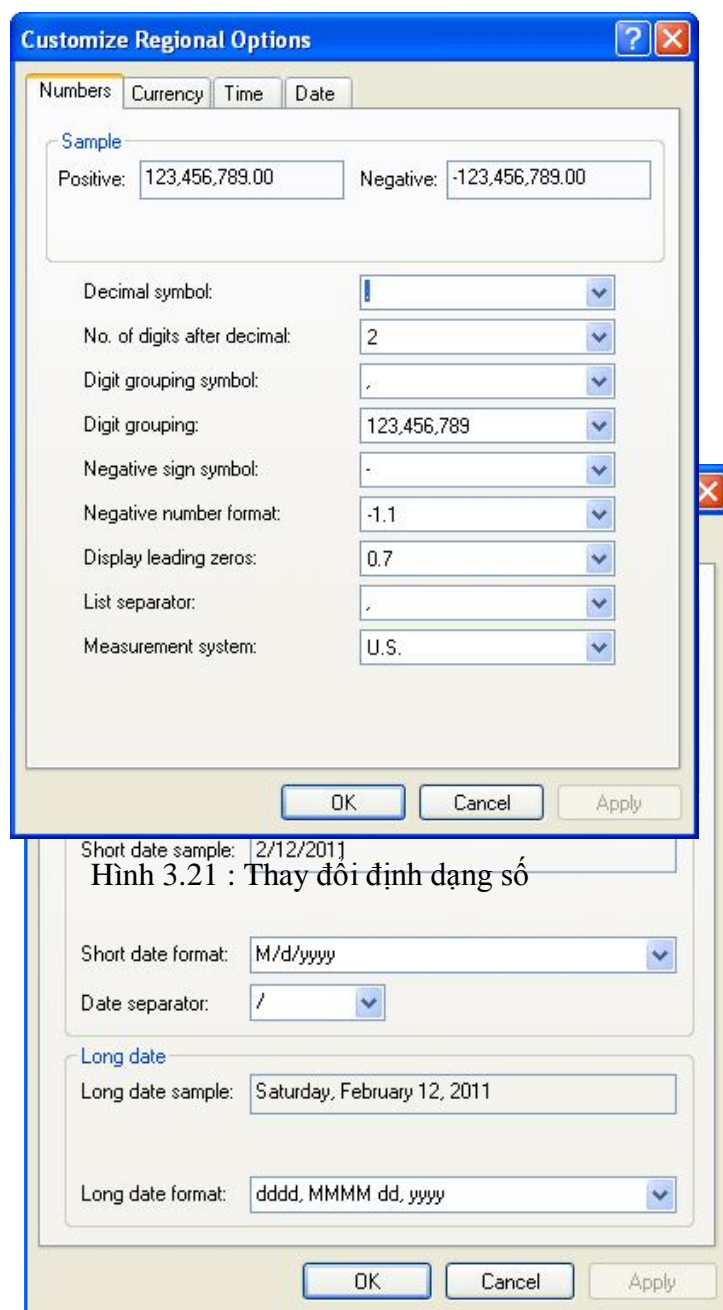
- Digit grouping symbol: Thay đổi ký hiệu phân nhóm hàng ngàn.

- Digit grouping: Thay đổi số ký số trong một nhóm (mặc nhiên là 3).

- Negative sign symbol: Thay đổi ký hiệu của số âm.

- Negative number format: Thay đổi dạng thể hiện của số âm.

- Display leading zeroes:



Hiển thị hay không hiển thị số 0

trong các số chỉ có phần thập phân: 0.7 hay .7.

- Measurement system: Chọn hệ thống đo lường như cm, inch, ...

- List separator: Chọn dấu phân cách giữa các mục trong một danh sách.

➤ **Currency:** Thay đổi định dạng tiền tệ (\$, VND, ...)

➤ **Time:** Thay đổi định dạng thời gian, cho phép bạn định dạng thể hiện giờ trong ngày theo chế độ 12 giờ hay 24 giờ

➤ **Date:** Thay đổi định dạng ngày tháng (Date), cho phép bạn chọn cách thể hiện dữ liệu ngày theo 1 tiêu chuẩn nào đó. Hình 3.22 : Thay đổi định dạng ngày tháng

Short date format: cho phép chọn quy ước nhập dữ liệu ngày tháng.

Ví dụ: ngày/tháng/năm (d/m/yy) hoặc tháng/ngày/năm (m/d/yy)

Quy ước:

d/ D (date): dùng để chỉ ngày

m/ M (month): dùng để chỉ tháng

y/ Y (year): dùng để chỉ năm

3.2.11 Máy In

Trong phần này sẽ giới thiệu một số thao tác cơ bản trên máy in như cài đặt máy in mới, định cấu hình cho máy in, quản lý việc in ấn, ...

a) Cài đặt thêm máy in :

Với một số máy in thông dụng Windows đã tích hợp sẵn chương trình điều khiển (driver) của các máy in, tuy nhiên cũng có những máy in mà trong Windows không có sẵn driver, trong trường hợp này ta cần phải có đĩa driver đi kèm với máy in.

Các bước cài đặt máy in:

- Chọn lệnh **Start/ Settings/ Printers and Faxes**

- Click chọn Add a Printer, xuất hiện hộp thoại Add Printer Wizard

- Làm theo các bước hướng dẫn của trình Wizard.

b) Loại bỏ máy in đã cài đặt

- Chọn lệnh **Start/ Settings/ Printers and Faxes**
- Click chuột chọn máy in muốn loại bỏ
- Nhấn phím Delete, sau đó chọn Yes để bỏ, ngược lại thì chọn No.

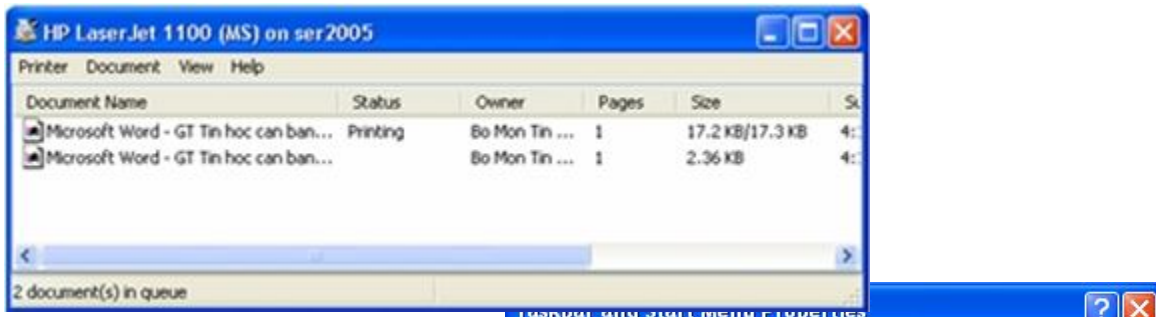
c) Thiết lập máy in mặc định

Nếu máy tính được cài nhiều máy in, ta có thể một máy in làm đặt máy in mặc định bằng cách chọn máy in đó rồi chọn **File/ Set as Default Printer** hoặc **R_Click/ Set as Default Printer**.

d) Cửa sổ hàng đợi in (Print Queue)

Nhấn đúp chuột lên biểu tượng máy in trong cửa sổ Printers and Faxes hoặc biểu tượng máy in ở thanh Taskbar. Khi đó xuất hiện hộp thoại liệt kê hàng đợi các tài liệu đang in.

Muốn loại bỏ một tài liệu trong hàng đợi in thì nhấn chuột chọn tài liệu đó và nhấn phím Delete.



Hình 3.23 : Cửa sổ hàng đợi in

3.2.12 Taskbar AND Start Menu

Chọn lệnh **Start/ Settings/ Taskbar and Start Menu**

❖ Lớp Taskbar



- **Lock the Taskbar:** khoá thanh Taskbar.
- **Auto hide:** cho tự động ẩn thanh taskbar khi không sử dụng.
- **Keep the Taskbar on top of other windows:** cho thanh Taskbar hiện lên phía trước các cửa sổ.
- **Group similar taskbar buttons:** cho hiện các chương trình cùng loại theo nhóm.
- **Show Quick Launch:** cho hiển thị các biểu tượng trong Start menu với kích thước nhỏ trên thanh Taskbar.
- **Show the Clock:** cho hiển thị đồng hồ trên thanh Taskbar
- **Hide inactive icons:** cho ẩn biểu tượng các chương trình không được kích hoạt.

Hình 3.24 : Lớp Taskbar

❖ **Lớp Start Menu**

Cho phép chọn hiển thị Menu Start theo dạng cũ (Classic Start Menu) hay dạng mới (Start Menu).

Click chọn lệnh **Customize**, cửa sổ Customize xuất hiện theo dạng cũ như hình 3.26 cho phép thực hiện một số thay đổi cho Menu Start.

- **Nút Add:** thêm một biểu tượng chương trình (Shortcut) vào menu Start.
- **Nút Remove:** xóa bỏ các biểu tượng nhóm (Folder) và các biểu tượng chương trình trong menu Start.
- **Nút Clear:** xóa các tên tập tin trong nhóm Documents trong menu Start.
- **Nút Advanced:** thêm, xóa, sửa, tạo các biểu tượng nhóm (Folder) và các biểu tượng chương trình (Shortcut) trong menu Start



Hình 3.25 : Lớp Start Menu



Hình 3.26 : Cửa sổ Customize

Khi Click vào nút **Advanced**, xuất hiện cửa sổ **Start Menu** cho phép tạo, xoá, đổi tên thư mục/ lối tắt trong **menu Start**.

✓ **Tạo Shortcut/ Folder trên menu Start**

- Click chọn Folder cha.
- Nếu tạo Folder, chọn File hoặc R_Click, chọn New/ Folder.
- Nếu tạo Shortcut, chọn File hoặc R_Click, chọn New/ Shortcut.
- Sau đó đặt tên cho Folder/ Shortcut.



Hình 3.27 : Cửa sổ Start Menu

✓ **Xóa Folder/ Shortcut khỏi menu Start**

Chọn Folder/ Shortcut cần xóa và nhấn phím Delete.

✓ **Đổi tên Folder/ Shortcut**

R_Click vào Folder/ Shortcut cần đổi tên, chọn Rename, nhập tên mới.

3.3 Windows Explorer

3.3.1 Giới thiệu

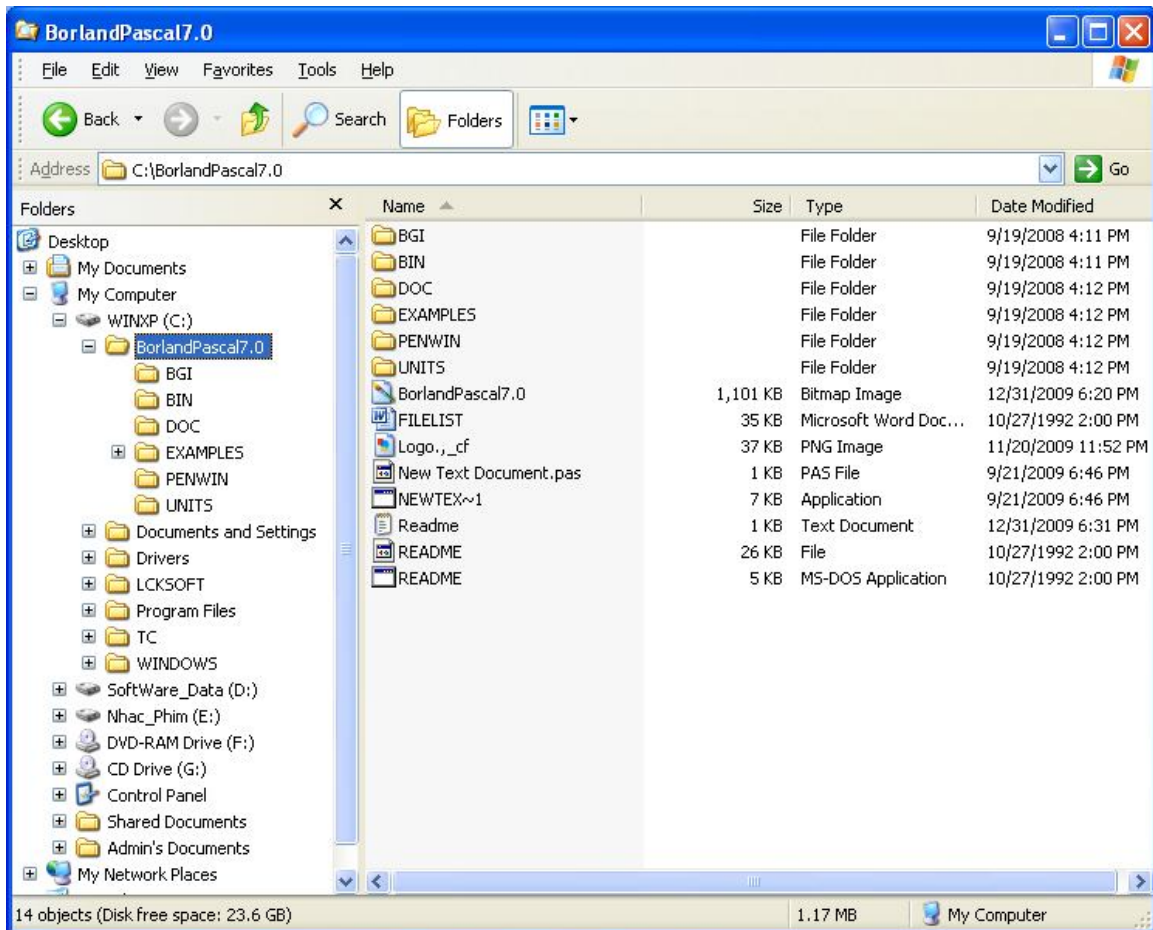
Windows Explorer là một chương trình được hỗ trợ từ phiên bản Windows 95 cho phép người sử dụng thao tác với các tài nguyên có trong máy tính như tập tin, thư mục, ổ đĩa và những tài nguyên khác có trong máy của bạn cũng như các máy tính trong hệ thống mạng (nếu máy tính của bạn có nối mạng).

Với Windows Explorer, các thao tác như sao chép, xóa, đổi tên thư mục và tập tin,... được thực hiện một cách thuận tiện và dễ dàng.

❖ **Khởi động chương trình Windows Explorer:** bạn có thể thực hiện một trong những cách sau:

- Chọn lệnh Start/ Programs/ Accessories/ Windows Explorer
- R_Click lên Start, sau đó chọn Explore
- R_Click lên biểu tượng My Computer, sau đó chọn Explore ...

❖ **Cửa sổ làm việc của Windows Explorer:**



Hình 3.28 : Cửa sổ Explorer

- **Cửa sổ trái (Folder)** là cấu trúc cây thư mục. Nó trình bày cấu trúc thư mục của các đĩa cứng và các tài nguyên kèm theo máy tính, bao gồm ổ đĩa mềm, ổ đĩa cứng, ổ đĩa CD...




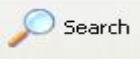

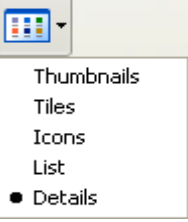
Những đối tượng có dấu cộng (+) ở phía trước cho biết đối tượng đó còn chứa những đối tượng khác trong nó nhưng không được hiển thị. Nếu Click vào dấu + thì Windows Explorer sẽ hiển thị các đối tượng chứa trong đối tượng đó. Khi đó, dấu + sẽ đổi thành dấu -, và nếu Click vào dấu - thì đối tượng sẽ được thu gọn trở lại.

- **Cửa sổ phải** liệt kê nội dung của đối tượng được chọn tương ứng bên cửa sổ trái.

❖ **Thanh địa chỉ (Address):**

Cho phép nhập đường dẫn thư mục/ tập tin cần tới hoặc để xác định đường dẫn hiện hành.

❖ **Các nút công cụ trên thanh Toolbar:**

	- Back : Chuyển về thư mục trước đó
	- Up : Chuyển lên thư mục cha
	- Forward : Chuyển tới thư mục vừa quay về (Back)
	- Search : Tìm kiếm tập tin/ thư mục
	- Folder : Cho phép ẩn/ hiện cửa sổ Folder bên trái
	- Views : Các chế độ hiển thị các đối tượng (tập tin/ thư mục/ ổ đĩa)

Nội dung trong cửa sổ có thể được sắp xếp thể hiện theo thứ tự. Đối với kiểu thể hiện Details, bạn có thể thực hiện bằng cách luân phiên nhấn chuột lên cột tương ứng (Name, Size, Type, Date Modified).

Trong các kiểu thể hiện khác bạn có thể thực hiện bằng lệnh **View/ Arrange Icons By** và lựa chọn tiếp một trong các khóa sắp xếp (theo tên, kích cỡ tập tin, kiểu tập tin, hoặc ngày tháng cập nhật).

Trong kiểu thể hiện bằng các biểu tượng lớn và biểu tượng nhỏ bạn có thể để Windows sắp xếp tự động bằng lệnh **View/ Arrange Icons By / Auto Arrange**. Tùy chọn Auto Arrange chỉ áp dụng cho cửa sổ của thư mục hiện hành.

3.3.2 Thao tác với các thư mục và tập tin

a) Mở tập tin/ thư mục :

Có ba cách thực hiện :

- Cách 1: D_Click lên biểu tượng của tập tin/ thư mục.

- Cách 2: R_Click lên biểu tượng của tập tin/ thư mục và chọn mục Open.

- Cách 3: Chọn tập tin/ thư mục và nhấn phím Enter.

Nếu tập tin thuộc loại tập tin văn bản thì chương trình ứng dụng kết hợp sẽ được khởi động và tài liệu sẽ được nạp vào.

Trong trường hợp chương trình ứng dụng không được cài đặt trong máy tính thì Windows sẽ mở hộp thoại Open With và cho chọn chương trình kết hợp. Nếu tập tin thuộc dạng chương trình ứng dụng thì chương trình tương ứng sẽ được khởi động.



Hình 3.29 : Chọn lệnh mở tập tin

b) Chọn tập tin/ thư mục :

- Chọn một tập tin/ thư mục: Click lên biểu tượng tập tin/ thư mục.

- Chọn một nhóm tập tin/ thư mục: có thể thực hiện theo 2 cách:

+ Các đối tượng cần chọn là một danh sách gồm các đối tượng liên tục: Click lên đối tượng đầu danh sách để chọn, sau đó nhấn giữ phím Shift và Click lên đối tượng ở cuối danh sách.

+ Các đối tượng cần chọn nằm rời rạc nhau: nhấn giữ phím Ctrl và Click chọn các đối tượng tương ứng.

c) Tạo thư mục :

- Chọn nơi chứa thư mục cần tạo (thư mục/ ổ đĩa ở cửa sổ bên trái).

- Chọn menu **File/ New/ Folder** hoặc **R_Click/ New/ Folder**.

- Nhập tên thư mục mới, sau đó gõ Enter để kết thúc.

d) Sao chép thư mục và tập tin :

Chọn các thư mục và tập tin cần sao chép. Sau đó có thể thực hiện theo một trong hai cách sau:

+ Cách 1: Nhấn giữ phím Ctrl và Drag đối tượng đã chọn đến nơi cần chép.

+ Cách 2: Nhấn tổ hợp phím Ctrl + C (hoặc Edit/ Copy hoặc R_Click và chọn Copy) để chép vào Clipboard, sau đó chọn nơi cần chép đến và nhấn tổ hợp phím Ctrl + V (hoặc Edit/ Paste hoặc R_Click và chọn Paste).

e) Di chuyển thư mục và tập tin :

Chọn các thư mục và tập tin cần di chuyển. Sau đó có thể thực hiện theo một trong hai cách sau:

+ Cách 1: Drag đối tượng đã chọn đến nơi cần di chuyển.

+ Cách 2: Nhấn tổ hợp phím Ctrl + X (hoặc Edit/ Cut hoặc R_Click và chọn Cut) để chép vào Clipboard, sau đó chọn nơi cần di chuyển đến và nhấn tổ hợp phím Ctrl + V (hoặc Edit/ Paste hoặc R_Click và chọn Paste).

f) Xóa thư mục và tập tin :

- Chọn các thư mục và tập tin cần xóa.

- Chọn File/ Delete

hoặc: Nhấn phím Delete

hoặc: R_Click và chọn mục Delete.

- Xác nhận có thực sự muốn xóa hay không (Yes/ No)

g) Phục hồi thư mục và tập tin :

Các đối tượng bị xóa sẽ được đưa vào Recycle Bin. Nếu muốn phục hồi các đối tượng đã xóa, bạn thực hiện các thao tác sau đây:

- D_Click lên biểu tượng Recycle Bin

- Chọn tên đối tượng cần phục hồi.

- Thực hiện lệnh File/ Restore hoặc R_Click và chọn mục Restore.

Ghi chú: Nếu muốn xóa hẳn các đối tượng, ta thực hiện thao tác xóa một lần nữa đối với các đối tượng ở trong Recycle Bin. Nếu muốn xoá hẳn tất cả các đối tượng trong Recycle Bin, R_Click lên mục Recycle Bin và chọn mục Empty Recycle Bin.

h) Đổi tên thư mục và tập tin :

- Chọn đối tượng muốn đổi tên

- Thực hiện lệnh **File/ Rename** hoặc nhấn phím F2 hoặc R_Click trên đối tượng và chọn mục Rename

- Nhập tên mới, sau đó gõ Enter để kết thúc.

Ghi chú: với tập tin đang sử dụng thì các thao tác di chuyển, xoá, đổi tên không thể thực hiện được.

i) Thay đổi thuộc tính tập tin và thư mục :

- Nhấn chuột phải lên đối tượng muốn thay đổi thuộc tính và chọn mục

Properties

- Thay đổi các thuộc tính.

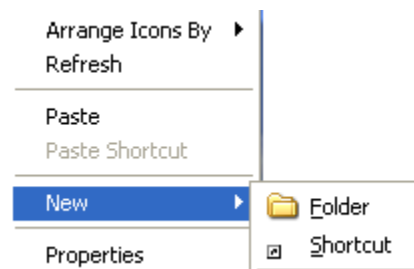
- Chọn Apply để xác nhận thay đổi, ngược lại thì nhấn Cancel.

3.3.3 Thao tác với các lối tắt (Shortcuts)

a) Tạo lối tắt trên màn hình nền :

- R_Click lên màn hình nền, chọn New/ Shortcut.

- Trong mục **Type the location of the item**, nhập đường dẫn của đối tượng cần tạo lối tắt (ổ đĩa/ thư mục/ tập tin, ...) hoặc Click lên nút Brown để chọn đường dẫn cho đối tượng. Click Next để qua bước kế tiếp.



- Nhập tên cho lối tắt cần tạo.

Hình 3.30 : Tạo lối tắt

– Click Finish để hoàn thành .

b) Các thao tác với lỗi tắt :

❖ Đổi tên cho lỗi tắt.

- R_Click lên lỗi tắt, chọn Rename.

- Nhập tên mới cho lỗi tắt.

- Nhấn Enter.

❖ Xoá bỏ lỗi tắt

- Chọn lỗi tắt cần xoá.

- Nhấn phím Delete hoặc R_Click lên lỗi tắt, chọn Delete.

- Xác nhận có thực sự muốn xoá hay không (Yes/ No).

❖ Thay đổi thuộc tính cho lỗi tắt

R_Click lên lỗi tắt, chọn Properties.

- Lớp **General**: cho phép chọn thuộc tính chỉ đọc (Read-only), hay ẩn (Hidden).

- Lớp **Shortcut**: cho phép chọn thay đổi một số lựa chọn sau:

Shortcut key: gán phím nóng cho lỗi tắt. Ví dụ: nhấn phím A (nếu muốn đặt phím nóng cho lỗi tắt là Ctrl + Alt + A, mặc nhiên phải có Ctrl + Alt). Khi muốn mở đối tượng ta chỉ cần nhấn tổ hợp phím vừa gán.

Run: chọn chế độ hiển thị màn hình khi mở (bình thường/ thu nhỏ/ phóng to).

Change Icon: thay đổi biểu tượng của lỗi tắt.

3.3.4 Định dạng đĩa

- R_Click vào tên của ổ đĩa (có thể đĩa mềm hoặc đĩa cứng) cần định dạng, sau đó chọn mục **Format**.

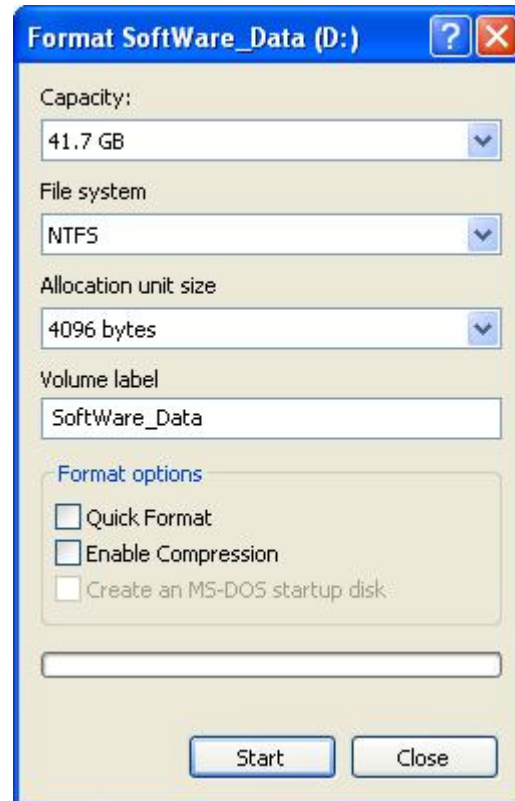
- Nếu muốn đặt tên cho đĩa thì nhập vào mục **Volume label**, muốn định dạng nhanh (chỉ xoá dữ liệu) thì chọn mục **Quick Format**, muốn dùng làm đĩa khởi động thì chọn mục **Create an MS-DOS startup disk**.

- Nhấp chọn **Start** để tiến hành định dạng.

Chú ý: dữ liệu trong đĩa sẽ bị xoá hoàn toàn, không thể phục hồi được.

3.3.5 Hiện thị thông tin của đĩa

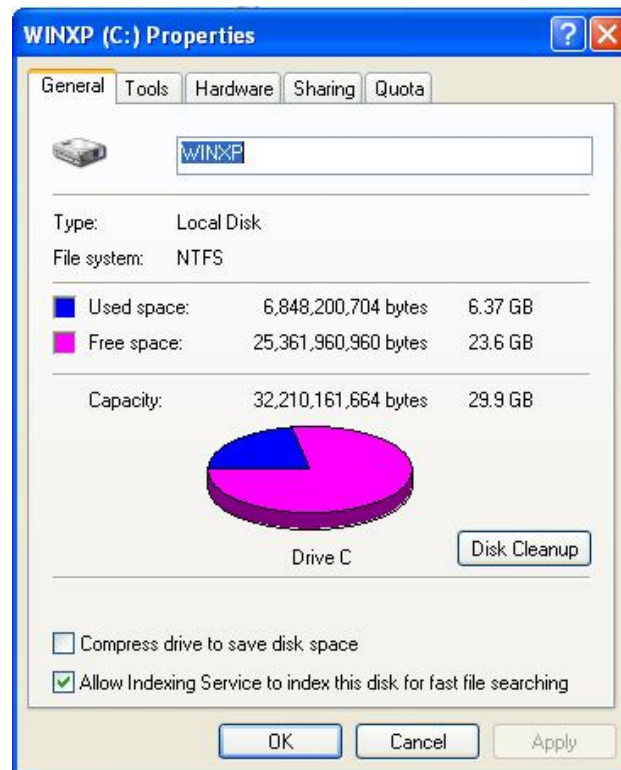
Click vào tên của ổ đĩa cần hiển thị thông tin, sau đó chọn mục Properties.



- **Lớp General:** cho biết dung lượng đĩa (Capacity), đã sử dụng bao nhiêu (Used space), còn lại bao nhiêu (Free space).

- **Lớp Tools:** cung cấp một số công cụ kiểm tra đĩa (Error-checking), tạo đĩa dự phòng (Backup), chống phân mảnh đĩa (Defragment).

- **Lớp Sharing:** cho phép đặt cách thức chia sẻ tài nguyên có trên đĩa (với máy tính có nối mạng).



Hình 3.32 : Thông tin của đĩa

Thực hành chương 3

Bài 1:

Câu 1:

- ✓ Mở cửa sổ My Computer
- ✓ Mở MSWord
- ✓ Mở MSEXcel
- ✓ Thực hiện chuyển đổi giữa các ứng dụng bằng cách nhấn tổ hợp phím Alt + Tab
- ✓ Sắp xếp các cửa sổ theo các dạng sau:
 - Mái ngói (Cascade)
 - Chiều ngang (Horizontal)
 - Chiều dọc (Vertical)

Câu 2:

- ✓ Trên màn hình Desktop tạo 2 shortcut sau :
 - Notepad biết tập tin thực hiện là Notepad.exe
 - Wordpad biết tập tin thực hiện là Wordpad.exe
- ✓ Sau khi tạo xong, hãy khởi động chương trình từ Shortcut vừa tạo
- ✓ Đóng các chương trình vừa chạy lại

Câu 3: Tiến hành khởi động chương trình Excel theo các cách sau :

- ✓ Từ Folder
- ✓ Từ lệnh Run
- ✓ Từ Menu Start
- ✓ Từ shortcut trên màn hình Desktop

Câu 4:

- ✓ Trên màn hình Desktop tạo Shortcut có tên Tap Ve biết tập tin thực hiện là MsPaint.exe
- ✓ Đổi tên Shortcut Wordpad thành Soan Thao

- ✓ Đổi biểu tượng (icon) cho Shortcut Tap Ve và Soan Thao

Câu 5: Mở một file văn bản Word và quan sát rồi thực hiện các thao tác sau :

- ✓ Phóng to, phục hồi, thu nhỏ cửa sổ chương trình
- ✓ Di chuyển cửa sổ, thay đổi kích thước cửa sổ
- ✓ Đóng cửa sổ

Câu 6: Trong Startup

- ✓ Tạo Shortcut Vietkey biết tập tin thực hiện là Vietkey.exe
- ✓ Tiến hành Restart máy và xem kết quả

Câu 7: Tìm tất cả các tập tin BaiTap với các tiêu chí sau:

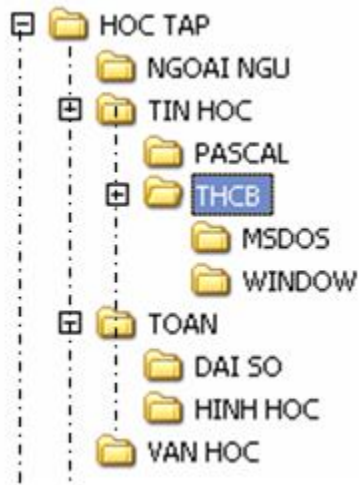
- ✓ Trong vòng một tháng gần đây nhất
- ✓ Có kích thước dưới 1MB

Bài 2:

Câu 1: Khởi động Windows Explorer và thực hiện:

- ✓ Bật / tắt thanh công cụ (Toolbar)
- ✓ Bật / tắt thanh Trạng thái (StatusBar)
- ✓ Bật / tắt cửa sổ chứa cây thư mục (Folder)
- ✓ Sắp xếp các đối tượng của cửa sổ bên phải dưới dạng:
 - Thumbnails
 - Tiles
 - Icon
 - List
 - Details
- ✓ Sắp xếp các đối tượng của cửa sổ bên phải dưới dạng:
 - By Name
 - By Type
 - By Size
 - By Modified

Câu 2: Tạo cây thư mục tại ổ đĩa D: như sau :



Câu 3: Thực hiện đổi tên các thư mục:

- ✓ Đổi tên thư mục NGOAI NGU thành ENGLISH
- ✓ Đổi tên thư mục TOAN thành TOANHOC
- ✓ Đổi tên thư mục VANHOC thành VAN
- ✓ Đổi tên thư mục MSDOS thành HDH DOS

Câu 4: Sao chép thư mục

- ✓ Chép thư mục TOANHOC vào thư mục ENGLISH
- ✓ Chép thư mục HDH DOS vào thư mục PASCAL
- ✓ Chép thư mục TINHOC vào thư mục VAN
- ✓ Chép thư mục PASCAL vào thư mục WINDOW

Câu 5: Di chuyển thư mục

- ✓ Di chuyển thư mục TOANHOC (trong thư mục ENGLISH) vào thư mục VAN
- ✓ Di chuyển thư mục TINHOC (trong thư mục VAN) vào thư mục PASCAL
- ✓ Di chuyển thư mục HDH DOS (trong thư mục PASCAL) vào thư mục HINHHOC
- ✓ Di chuyển thư mục VAN về thư mục gốc của ổ đĩa

Câu 6: Xóa và phục hồi thư mục

- ✓ Xóa thư mục TOAN
- ✓ Phục hồi thư mục TOAN vừa xóa

Câu 7: Đặt thuộc tính cho thư mục

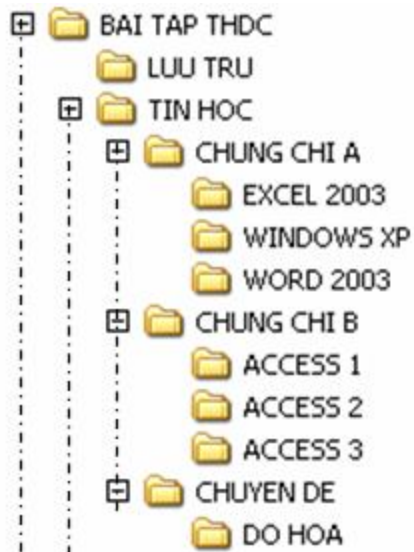
- ✓ Hãy gán thuộc tính chỉ đọc (Read Only) cho 2 thư mục TOAN và TINHOC
- ✓ Hãy gán thuộc tính ẩn cho thư mục NGOAINGU
- ✓ Gỡ bỏ các thuộc tính chỉ đọc cho thư mục TINHOC

Câu 8: Thay đổi tùy chọn (Option) của Windows Explorer

- ✓ Điều chỉnh Explorer sao cho không thể thấy các tập tin và thư mục ẩn
- ✓ Có nhận xét gì về thư mục NGOAINGU và TOAN
- ✓ Điều chỉnh Explorer để có thể xem các tập tin và thư mục ẩn
- ✓ Hãy điều chỉnh Explorer sao cho không thấy phần mở rộng của các tập tin.
Sau đó, điều chỉnh lại tình trạng như ban đầu.

Bài 3:

Câu 1: Tạo cây thư mục như sau (trên ổ đĩa D:)



Câu 2: Chuyển về thư mục ổ C:\Windows, thực hiện các yêu cầu:

- ✓ Sao chép các tập tin có phần mở rộng là .ini từ C:\Windows sang thư mục LUUTRU

- ✓ Sao chép các tập tin có phần mở rộng là .bmp từ C:\Windows sang thư mục DOHOA

Câu 3: Chuyển về ổ D:\ thực hiện các yêu cầu sau:

- ✓ Trong thư mục BAI TAP THDC tạo thêm thư mục CHUONG TRINH
- ✓ Chép tất cả các tập tin có phần mở rộng là .exe từ C:\Windows sang thư mục CHUONG TRINH vừa tạo
- ✓ Cho biết tổng dung lượng của CHUONG TRINH

Câu 4: Đổi tên, đặt thuộc tính, xóa thuộc tính, xóa thư mục

- ✓ Đổi tên thư mục thành họ tên của sinh viên
- ✓ Đặt thuộc tính chỉ đọc (Read Only) cho thư mục TIN HOC
- ✓ Gỡ (xóa các thuộc tính vừa đặt cho các thư mục
- ✓ Xóa thư mục tên của Sinh viên

Bài 4:

Câu 1: Khởi động Control Panel

Câu 2: Chỉnh các tùy chọn trong Control Panel theo các yêu cầu sau :

- ✓ Chỉnh sửa ngày và giờ của hệ thống
- ✓ Đổi hình nền cho Desktop
- ✓ Thiết lập Screen server thành dạng Marquee với dòng chữ là tên sinh viên thời gian chờ là 4 phút.
- ✓ Thực hiện thay đổi chế độ màu với các giá trị khác nhau
- ✓ Thực hiện thay đổi độ phân giải với các giá trị khác nhau

Câu 2: Thực hiện các thay đổi về mouse:

- ✓ Điều chỉnh tốc độ double click của chuột
- ✓ Thay đổi biểu tượng cho các hành động: Normal Select, Help Select, Working in background,...
- ✓ Đánh dấu vị trí chuột khi nhấp phím Control (show location)
- ✓ Tạo vệt dài (dấu / trails) khi di chuyển chuột

Câu 3: Các tùy chọn về Regional Setting:

- ✓ Chọn thẻ Date chỉnh kiểu của Short date là dd/mm/yyyy
- ✓ Chọn thẻ Time để đổi dạng hiển thị hệ thống là 24 giờ
- ✓ Chọn thẻ Currency để đổi lại ký hiệu tiền tệ là VND

Câu 4: Cài đặt Font cho máy tính

- ✓ Cài bộ Font VNI
- ✓ Xóa một số Font trong bộ Font vừa cài đặt
- ✓ Phục hồi lại các Font đã xóa

Chương 4 : PHÒNG VÀ CHỐNG VIRUS

4.1 Virus máy tính là gì?

Virus tin học (hay còn gọi virus máy tính) là các chương trình đặc biệt do con người tạo ra ẩn trong máy tính. Các chương trình này có khả năng bám vào các chương trình khác như một vật thể ký sinh. Chúng cũng tự nhân bản để tồn tại và lây lan. Do cách thức hoạt động của chúng giống virus sinh học nên người ta không ngần ngại đặt cho chúng cái tên "Virus" đầy ẩn tượng này.

Khi máy tính bị nhiễm Virus thì nó có thể có những biểu hiện không bình thường như khởi động lâu hơn, không in được văn bản ra máy in... nhưng nhiều khi không có biểu hiện khác thường nào hết.

4.2 Cách thức phá hoại của virus tin học

Người ta chia virus thành 2 loại chính là B-virus, loại lây vào các mẫu tin khởi động (Boot record) và F-virus lây vào các tập tin thực thi (Executive file). Cách phân loại này chỉ mang tính tương đối, bởi vì trên thực tế có những loại virus lưỡng tính vừa lây trên boot record, vừa trên file thi hành. Ngoài ra, ta còn phải kể đến họ virus macro nữa.

- B-virus: Nếu boot máy từ một đĩa mềm nhiễm B-virus, bộ nhớ của máy sẽ bị khống chế, kế tiếp là boot record của đĩa cứng bị lây nhiễm.

- F-virus: Nguyên tắc của F-virus là thêm đoạn mã lệnh vào file thi hành (dạng .COM và .EXE) để mỗi lần file thực hiện, đoạn mã này sẽ được kích hoạt, thường trú trong vùng nhớ, khống chế các tác vụ truy xuất file, dò tìm các file thi hành chưa bị nhiễm virus khác để tự lây vào.

- Macro virus: lây qua các tập tin văn bản, Email, chọn ngôn ngữ Macro làm phương tiện lây lan.

4.3 Các phương pháp phòng và chống virus

❖ Đề phòng F-virus: Nguyên tắc chung là không được chạy các chương trình không rõ nguồn gốc. Hãy dùng các chương trình diệt virus kiểm tra các chương trình này trước khi chép vào đĩa cứng của máy tính.

❖ Đề phòng Macro virus: họ virus này lây trên văn bản và bảng tính của Microsoft. Vì vậy, khi nhận một file DOC hay XLS nào, bạn hãy nhớ kiểm tra chúng trước khi mở ra.

❖ Diệt Virus: Để phát hiện và diệt virus, người ta viết ra những chương trình chống virus, gọi là anti-virus. Nếu nghi ngờ máy tính của mình bị virus, bạn có thể dùng các chương trình này kiểm tra các ổ đĩa của máy. Một điều cần lưu ý là nên chạy anti-virus trong tình trạng bộ nhớ tốt (khởi động máy từ đĩa mềm sạch) thì việc quét virus mới hiệu quả và an toàn, không gây lan tràn virus trên đĩa cứng. Có hai loại anti-virus, ngoại nhập và nội địa:

+ Các anti-virus ngoại đang được sử dụng phổ biến là VirusScan của McAfee, Norton Anti-virus của Symantec, Toolkit, Dr. Solomon... chúng diệt virus ngoại rất hiệu quả nhưng không diệt được virus nội địa.

+ Các anti-virus nội thông dụng là D2 và BKAV. Đây là các phần mềm miễn phí, các anti-virus nội địa chạy rất nhanh do chúng nhỏ gọn, tìm-diệt hiệu quả các virus nội địa. Nhược điểm của chúng là khả năng nhận biết các virus ngoại kém.

4.5 Chương trình diệt virus BKAV

Chương trình diệt virus BKAV được phát triển từ năm 1995 bởi nhóm nghiên cứu là giảng viên và sinh viên Trường Đại học bách khoa Hà Nội.

❖ Khởi động BKAV:

Chọn Start/ Programs/ Bách khoa Antivirus/ Bkav 2002 hoặc D_Click vào lối tắt của BKAV trên màn hình nền. Giao diện của chương trình BKAV như hình dưới:



Hình 6.1 Chương trình diệt Virus BKAV

Lớp Tùy chọn (Options):

- Chọn ổ đĩa: cho phép chọn vị trí (ổ đĩa/ thư mục) cần quét.
- Chọn kiểu File: cho phép chọn kiểu tập tin cần quét virus.

Thông thường thì chỉ có các tập tin chương trình và tập tin văn bản mới cần phải quét virus, các tập tin loại khác ít khi bị virus lây nhiễm.

- Lựa chọn khác: cho phép chọn các tùy chọn khi quét và khởi động chương trình.

❖ Lớp Nhật ký (History):

Trong khi quét: liệt kê các tập tin đã được quét virus.

Trước/ sau khi quét: liệt kê thông tin về các tập tin bị lây nhiễm virus và tình trạng của tập tin sau khi đã quét virus: diệt thành công, không diệt được

❖ Lớp Lịch quét (Schedule):

Cho phép đặt lịch quét virus tự động (hàng ngày, hàng tuần, hàng tháng, ...)

❖ Lớp Virus list: liệt kê danh sách các virus.

❖ Lệnh Quét (Scan): tiến hành quét/ ngưng quét virus theo đường dẫn được chỉ ra ở mục Chọn ổ đĩa.

❖ Lệnh Thoát (Exit): thoát khỏi chương trình.

❖ Lệnh Trợ giúp (Help): hướng dẫn sử dụng chương trình.

Tóm lại:

Việc bảo vệ dữ liệu và phòng chống virus là vấn đề chúng ta phải luôn quan tâm. Để bảo vệ dữ liệu của bạn có hiệu quả bạn cần phải luôn ghi nhớ những điều sau đây:

- Luôn tạo ra các bản sao đối với các dữ liệu quan trọng, và bản sao này phải được cất giữ ở nơi an toàn.

- Luôn luôn quét virus trên các đĩa mềm lạ trước khi mở các tập tin hoặc chạy chương trình trên đĩa đó. Các đĩa CD cũng có thể chứa các chương trình nhiễm virus.

- Thường xuyên cập nhật các chương trình diệt virus vì virus mới có thể phát sinh mỗi ngày, chương trình diệt virus cũ không thể diệt được virus mới.

Chương 5 : NGÔN NGỮ LẬP TRÌNH PASCAL

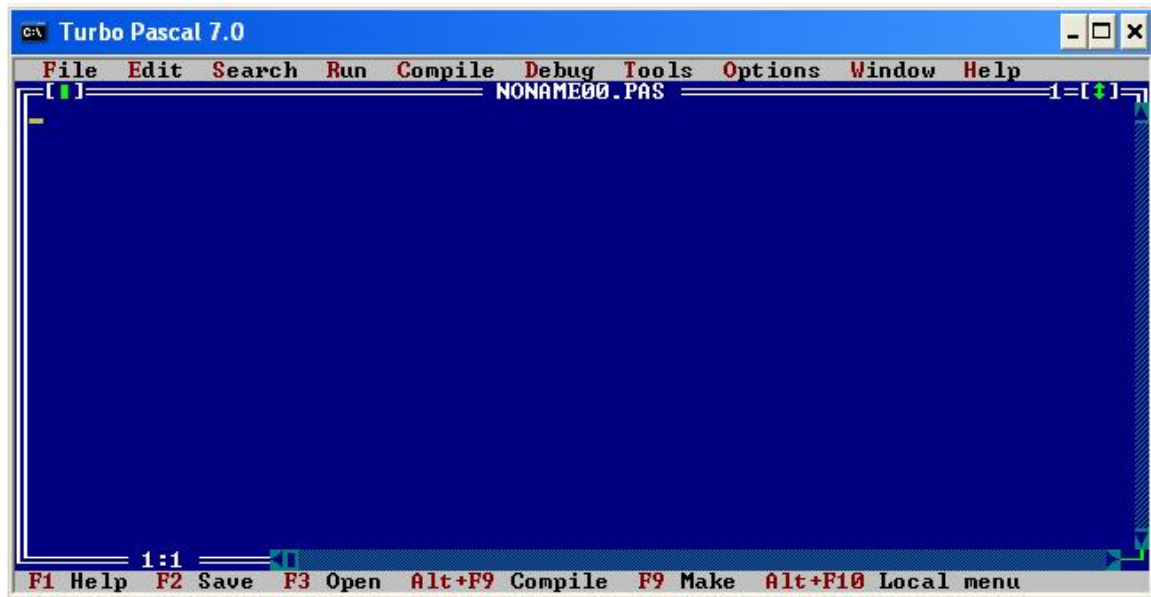
5.1 Giới thiệu

PASCAL là ngôn ngữ lập trình bậc cao của tác giả Niklaus Wirth (giáo sư người Thụy sĩ), được công bố vào đầu những năm 1970. Tên PASCAL là để kỉ niệm nhà Toán học người Pháp B. Pascal.

PASCAL là ngôn ngữ lập trình có tính cấu trúc và tính hệ thống: các kiểu dữ liệu đa dạng, các cấu trúc điều khiển chặt chẽ, các cấu trúc khối trong chương trình rõ ràng...

PASCAL là ngôn ngữ lập trình có định kiểu rõ ràng: các đại lượng (biến và hằng) đã được khai báo để sử dụng với kiểu dữ liệu này thì không thể đem dùng lẫn với kiểu khác. PASCAL ban đầu được sáng tác để làm ngôn ngữ dạy học cho những người mới học lập trình. Đặc tính sáng sủa, dễ hiểu, dễ đọc của nó giúp người mới học có thể viết một chương trình máy tính một cách dễ dàng. Sau khi ra đời, do có nhiều ưu điểm, PASCAL đã được nhiều hãng máy tính phát triển và cài đặt cho nhiều hệ thống máy tính, như: ISO PASCAL (PASCAL chuẩn), ANSI PASCAL v.v. Phổ biến nhất hiện nay ở nước ta cũng như trên thế giới là việc cài đặt TURBO PASCAL cho các hệ thống máy tính. TURBO PASCAL được hãng BORLAND INTERNATIONAL hoàn thiện với các ưu điểm là chương trình viết gọn dịch nhanh, không ngừng được cải tiến đáp ứng yêu cầu của người sử dụng. TURBO PASCAL đã được phát triển qua các phiên bản ngày càng mở rộng và phức tạp. Việc khai thác sâu các tính năng của từng phiên bản là do khả năng và nhu cầu của người lập trình.

Màn hình TURBO PASCAL 7.0



Hình 5.1: Màn hình Turbo Pascal 7.0

Dòng 1: là dòng menu chính, trên đó có ghi các mục cơ bản nhất để người sử dụng có thể chọn bằng cách ấn đồng thời Alt+ chữ cái đầu của mục. Ví dụ: Alt+F để mở mục File.

Dòng 2: tên tệp đang làm việc...

Dòng cuối cùng là dòng hướng dẫn, nhắc người sử dụng một số phím chức năng chính.

Phần màn hình rộng ở giữa là phần soạn thảo để ta nhập và chạy chương trình.

Người mới học lập trình nên ghi nhớ ngay các phím chức năng:

F2 ghi tệp đang soạn thảo vào đĩa. Nếu chưa đặt tên sẽ có thông báo để ta gõ tên vào (tên được đặt theo quy định của MS-DOS)

F3 mở tệp mới

Alt+F9 dịch chương trình

Ctrl+F9 dịch và cho chạy chương trình

Alt+X Kết thúc PASCAL trở về hệ điều hành

5.1.1 Những qui định về cách viết biểu thức và chương trình

a) Bộ kí tự :

TURBO PASCAL sử dụng bộ chữ viết như sau:

- Bộ 26 chữ cái Latinh: A, B, ..., Z ; a, b, ..., z
- Bộ các chữ số: 0, 1, ..., 9
- Các kí hiệu Toán học: +, -, * , /, =, <, >, ...
- Dấu gạch nối, dấu chấm, dấu ngoặc và các kí hiệu đặc biệt khác: _ . , ; ?

[] () \$ % ...

b) Từ khoá :

Từ khoá là một số từ dành riêng của TURBO PASCAL, những từ này đã được gán sẵn những chức năng đặc biệt, ta không được dùng vào việc đặt tên hay các công việc khác. Sau đây là một số từ quan trọng, bạn đọc sẽ lần lượt biết các từ khoá và chức năng của từng từ qua các bài học.

- Các từ khoá chung: Program, begin, end
- Các từ khoá để khai báo: uses, const, type, label, var
procedure, function
array, string, record, ...
- Các từ khoá lệnh:
if... then... else...
case ...of ...
for to ... do...
while... do...
repeat... until...
- Một số tên chuẩn:
Boolean, char, integer, real,
read, readln, write, writeln,
abs, cos, sin, sqrt, exp...

c) Tên :

Tên là một dãy kí tự dùng để chỉ tên các đại lượng như tên hằng, tên biến, tên kiểu hay tên chương trình con.

Tên được đặt theo quy tắc sau: bắt đầu bởi một chữ cái, rồi đến chữ số hay dấu gạch nối và không dài quá 127 kí tự (!). Tên thường được đặt mang tính gợi nhớ. TURBO PASCAL không phân biệt chữ hoa và chữ thường khi viết từ khoá hay tên.

Ví dụ : khi viết chương trình giải phương trình bậc 2, ta có thể đặt tên chương trình là GIAI_PHUONG_TRINH_BAC2, đặt tên biến chứa biệt thức Δ là delta (không được dùng kí hiệu Δ), đặt tên các nghiệm là X1 và X2,...

- Dấu chấm phẩy (;) dùng để ngăn cách các câu lệnh trong một chương trình
- Lời giải thích được đặt trong cặp dấu (* ... *) hoặc { ... }, các nội dung được chứa trong cặp dấu này sẽ bị chương trình bỏ qua trong khi dịch

5.1.2 Cấu trúc chung của một chương trình Pascal

Ta hiểu một chương trình máy tính là một dãy các lệnh, các chỉ thị hướng dẫn máy thực hiện một nhiệm vụ, một xử lí nào đó trên tập các dữ kiện vào và cho ra kết quả. Một chương trình PASCAL có cấu trúc chung như sau:

PROGRAM	Tên_chương_trình;
USES	Danh sách đơn vị.;
LABEL	Danh sách nhãn;
CONST	Danh sách hằng;
TYPE	Danh sách kiểu;
VAR	Danh sách biến;
PROCEDURE	Khai báo thủ tục;
FUNCTION	Khai báo hàm;
BEGIN	
{ Các câu lệnh; }	
END.	

Một chương trình PASCAL điển hình gồm 3 phần: Phần tiêu đề, phần khai báo, phần thân chương trình.

a) Phần tiêu đề :

Phần này có dạng

```
PROGRAM Tên_chương_trình;
```

bắt đầu bằng từ khoá PROGRAM, ít nhất một dấu cách rồi đến tên chương trình, kết thúc bởi dấu chấm phẩy (;). Tên_chương_trình do ta đặt ra, mang ý nghĩa nội dung của chương trình, dấu ; là bắt buộc phải có.

Ví dụ : Program Giai_PT_bac2;

Phần tiêu đề là tùy chọn, có thể có hay không cũng được, nhưng nếu đã có thì phải đủ các thành phần như trên.

b) Phần khai báo

Phần này mô tả các đối tượng của bài toán mà chương trình sẽ xử lí. Các khai báo khác nhau được viết cách nhau bởi dấu ;. Gồm:

Khai báo đơn vị	uses ...
Khai báo nhãn	label
Khai báo hằng:	const
Khai báo kiểu:	Type
Khai báo biến:	Var
Khai báo chương trình con:	Procedure Function

Các khai báo của phần này tùy thuộc từng bài toán cụ thể mà có thể có hay không, hoặc có một hoặc một vài khai báo. Cách khai báo và sử dụng cụ thể sẽ được giới thiệu ở trong mỗi phần sau.

c) Phần thân chương trình

Phần này được bắt đầu bởi từ khoá BEGIN , kết thúc bởi từ khoá END và dấu chấm (.) hết. đặt giữa BEGIN và END là các lệnh của chương trình , các lệnh được viết cách nhau bởi dấu ; như sau:

BEGIN

Lệnh 1;

Lệnh 2;

Lệnh n;

END.

Phần này bắt buộc phải có đối với mọi chương trình, nó chứa các lệnh xử lý các dữ liệu đã được mô tả ở phần khai báo. đây là phần chính của chương trình để giải quyết bài toán. Người lập trình phải bỏ nhiều công sức để thiết kế thuật giải và viết chương trình cho phần này.

d) Các bước cơ bản khi lập trình

Trước khi viết một chương trình giải quyết một bài toán nào đó, người lập trình phải biết cách giải bài toán đó hay chính xác hơn phải biết thuật giải để giải bài toán và trình bày thuật giải bằng ngôn ngữ lập trình .

Bước 1: Soạn thảo chương trình

Đây là bước viết mới một chương trình, có thể viết trên giấy trước rồi nhập vào máy, sửa lỗi, cập nhật. Các chương trình này được gọi là chương trình nguồn, tên chúng được tự động gán đuôi PAS. Ta có thể dùng một hệ soạn thảo văn bản nào đó, thường là hệ soạn thảo văn bản của chính TURBO PASCAL.

Bước 2: Dịch chương trình (Alt+F9)

Gọi chương trình dịch (compiler) dịch chương trình nguồn đã viết ở bước 1 sang dạng mã máy, kết quả thường tạo ra các tệp dạng *.EXE, *.OBJ. Bước này sẽ cho phép ta phát hiện các lỗi để sửa. Thường các lỗi cú pháp như thiếu dấu ; hoặc viết sai từ khoá, sai tên sẽ được thông báo. Ta phải sửa hết các lỗi rồi chuyển sang bước 3.

Bước 3: Chạy chương trình và thử (Ctrl+F9)

Nếu các dữ kiện được cung cấp chính xác mà chương trình cho kết quả sai thì ta phải xem lại thuật giải. đây là lỗi đặc biệt nghiêm trọng vì nó không thể hiện ra ngoài qua các thông báo lỗi và có thể làm sai toàn bộ bài toán. Ta cũng phải quay lại bước 1 để sửa và chạy lại.

5.2 Các kiểu dữ liệu

5.2.1 Kiểu số nguyên

Kiểu số nguyên bao gồm tất cả các số nguyên có thể biểu diễn được trên máy tính. Từ khoá : Integer

Phạm vi biểu diễn: từ -32768 đến 32767 , tức -2^{15} đến $2^{15}-1$ (dùng 2 byte=16 bit) Như vậy kiểu integer chỉ là một đoạn của tập số nguyên

Các phép toán: Kiểu integer được thực hiện với tất cả các phép toán dành cho số nguyên thông thường

+ Các phép toán số học: + (cộng) , - (trừ) , * (nhân) , / (chia) , DIV (phép chia lấy phần nguyên, ví dụ: $7 \text{ div } 3=2$), MOD (phép chia lấy phần dư, ví dụ: $7 \text{ mod } 3=1$).

+ Các phép so sánh: = (bằng) , < (nhỏ hơn) , <= (nhỏ hơn hoặc bằng) , > (lớn hơn) , >= (lớn hơn hoặc bằng) , <> (khác)

Trong TURBO PASCAL, ngoài kiểu số nguyên được khai báo với từ khoá integer nói trên còn có các kiểu số nguyên khác (xem bảng) phục vụ cho các mục đích tính toán khác nhau.

Kiểu (từ khoá)	Phạm vi biểu diễn	Kích thước (byte)
Byte	0..255	1
Shortint	-128..127	1
Integer	-32768..32767	2
Word	0..65535	2
Longint	-2147483648..2147483647	4

5.2.2 Kiểu số thực

Kiểu số thực là tập hợp tất cả các số thực có thể biểu diễn được trên máy tính. Không phải tất cả các số thực đều có thể biểu diễn được trên máy tính mà độ lớn và độ chính xác của nó phụ thuộc vào máy và chương trình dịch.

- Kiểu số thực được khai báo với từ khoá real

- Phạm vi biểu diễn: từ 1.9×10^{-39} đến 1.7×10^{38}

- Các phép toán: Kiểu real có thể thực hiện với tất cả các phép toán của số thực thông thường: +, -, *, / và các phép so sánh (không có phép div và mod dành cho số thực).

- Cách viết số thực dạng dấu phẩy động:

Trong máy tính số thực được viết dùng dấu chấm để ngăn giữa phần nguyên và phần lẻ thập phân. Trong dạng dấu phẩy động số thực được viết theo dạng: AE+b, trong đó A gọi là phần định trị, E+b gọi là phần mũ.

Ví dụ: Ta xem xét các cách viết khác nhau của cùng 1 số $824.12345=8.2412345 \times 10^2$ hay $8.2412345E+2 = 824123.45 \times 10^{-3}$ hay $824123.45E-3$

Trong TURBO PASCAL, ngoài kiểu số thực được khai báo với từ khoá real nói trên còn có các kiểu số thực khác (xem bảng) phục vụ cho các mục đích tính toán với độ lớn và độ chính xác khác nhau

Kiểu (từ khóa)	Phạm vi biểu diễn	Chữ số có nghĩa	Kích thước (byte)
Single	1.5E-45..3.4E+38	7-8	4
Real	2.9E-39..1.7E+38	11-12	6
Double	5.0E-324..1.7E+308	15-16	8
Extended	3.4E-4932..1.1E+4932	19-20	10

Các hàm số học chuẩn dùng cho kiểu số nguyên và số thực: Đây là những hàm được định nghĩa sẵn trong TURBO PASCAL, ta có thể sử dụng các công cụ để tính toán.

Tên hàm (trong TP)	Cho kết quả(hàm)
ABS(X)	X (giá trị tuyệt đối của X)
SQR(X)	X^2 (X bình phương)
SQRT(X)	(căn bậc 2 của X)
LN(X)	lnX (Logarit Nepe)
EXP(X)	e^x
SIN(X)	sinX, X cho bằng radian
COS(X)	cosX, X cho bằng radian

ARCTAN(X)	arctangX
SUCC(n)	n+1 (số tiếp theo của n), n nguyên
PRED(n)	n-1 (số kề trước của n), n nguyên
TRUNC(X)	cắt, cho phần nguyên của X
ROUND(X)	làm tròn phần lẻ của X

5.2.3 Kiểu kí tự

Kiểu kí tự là tập hợp tất cả các kí tự của bảng mã máy tính (ASCII) Từ khoá Char. Có thể sử dụng các phép toán so sánh đối với kiểu kí tự, kí tự nào đứng trước theo thứ tự của bảng mã máy tính sẽ được coi là nhỏ hơn, kí tự đứng sau được xem là lớn hơn.

Ví dụ: 'a' < 'b'

Hàng kí tự được viết trong cặp dấu nháy ''

Các hàm dùng cho kiểu kí tự:

Tên hàm (trong TP)	Cho kết quả (hàm)
ORD(C)	cho số thứ tự của kí tự C trong bảng mã
CHR(n)	cho kí tự có số thứ tự là n
SUCC(C)	cho kí tự đứng kề sau
PRED(C)	cho kí tự đứng kề trước

5.2.4 Kiểu logic (boolean)

Từ khoá Boolean

Các phép toán logic: NOT , AND, OR, XOR

Các phép toán so sánh: =, <, >, ...

Kiểu logic boolean chỉ có 2 giá trị : True và False

Người ta định nghĩa: False < True

5.2.5 Kiểu liệt kê (Enumerated scalar type)

Là kiểu cho phép người lập trình có thể tự định nghĩa ra các kiểu vô hướng bằng cách liệt kê các giá trị của kiểu vô hướng ra thông qua các tên do người lập trình tạo ra. Danh sách các giá trị này được đặt trong ngoặc đơn và được mô tả bằng một tên kiểu trong phần mô tả (Phần TYPE).

Từ khoá: **TYPE**

Ví dụ:

TYPE

Boolean=(False, True);

Color=(Red, Blue, Green, While, Black);

Một biến vô hướng có thể định nghĩa thông qua các kiểu đã được mô tả trong phần **TYPE** như sau:

VAR

Ketqua: Boolean;

Mau1, Mau2: Color;

Hoặc khai báo trực tiếp với mô tả kiểu dữ liệu:

VAR

Gioitinh: (Nam, nu);

Ngay: (Chunhat, Hai, Ba, Tu, Nam, Sau, Bay);

Ví dụ:

Ketqua:=True;

Mau1:=Blue;

Gioitinh:=Nam;

Ngay:=Chunhat;

5.2.6 Kiểu khoảng con (Sub-range type)

Là kiểu vô hướng được dụng khi một biến chỉ được lấy giá trị trong một khoảng (xác định bởi cận trên và cận dưới).

❖ Quy tắc định nghĩa:

Hang_can_duoi..Hang_can_tren;

Trong đó: **Hang_can_duoi < Hang_can_tren**, và cùng kiểu.

❖ Tác dụng: - Tiết kiệm ô nhớ.

- Có thể kiểm tra giá trị của biến khi chạy chương trình không được vượt ra khỏi giới hạn của khoảng con.

Ví dụ:

TYPE

Ngay= (Chunhat, Hai, Ba, Tu, Nam, Sau, Bay);

Chu_cai_hoa='A'..'Z';

VAR

ch: Chu_cai_hoa;

Ngay_lam_viec: Hai..Bay;

5.3 Khai báo hằng, biến, biểu thức, câu lệnh

5.3.1 Khai báo hằng

Hằng là đại lượng không thay đổi giá trị trong quá trình xử lí. Ta khai báo hằng theo cú pháp:

```
const Tên_hằng=Giá trị ;
```

ở đó: Tên_hằng do ta đặt ra, Giá trị là giá trị mà tên hằng sẽ nhận và giữ không đổi suốt quá trình xử lí.

Ví dụ: const n=10; { khai báo hằng số nguyên n=10 }

pi=3.1416; { khai báo hằng số pi }

ki_tu='Y'; { khai báo hằng kí tự ki_tu nhận kí tự Y }

5.3.2 Khai báo biến

Biến là đại lượng có thể thay đổi giá trị trong quá trình xử lí. Biến của chương trình là tên của ô nhớ cất giữ dữ liệu. Muốn sử dụng biến nào ta phải khai báo trước biến đó bằng cách viết tên biến trong phần khai báo ở đầu chương trình sau từ khoá Var.

Cú pháp:

```
Var Tên_biến:Kiểu_dữ_liệu;
```

ở đó: Tên_biến do ta đặt ra, Kiểu_dữ_liệu là tên của một trong các kiểu dữ liệu của TURBO PASCAL, dấu hai chấm bắt buộc phải có để ngăn giữa 2 phần của khai báo, dấu chấm phẩy kết thúc dòng khai báo. Các biến có cùng kiểu có thể được khai báo cùng nhau bằng cách viết các tên biến cách nhau bởi dấu phẩy (,), nhóm các biến khác kiểu nhau được viết cách nhau bởi dấu ;

Ví dụ:

```
Var i:integer;      { khai báo một biến có tên là I, biến này có kiểu  
nguyên}  
x,y,z:real;       { khai báo 3 biến x,y,z nhận các giá trị có kiểu thực }  
t: char; { khai báo biến t có kiểu kí tự }  
kiem_tra:boolean; { khai báo biến logic }  
Tuoi: 1..300;
```

Đối với mỗi bài toán, khi phân tích thuật toán, ta phải xác định rõ ngay chương trình cần phải nhập vào các dữ kiện gì ? các kết quả trung gian nào cần phải được chứa, dữ liệu ra là những gì?... để từ đó có phần khai báo chính xác, không bị thiếu mà cũng không thừa gây lãng phí ô nhớ (!)

5.3.3 Biểu thức

Biểu thức là một công thức tính toán để có một giá trị theo một công thức toán học nào đó. Một biểu thức bao gồm các toán tử kết hợp với các toán hạng. Các toán tử trong TURBO PASCAL gồm: các phép toán số học (+, -, *, /), các phép so sánh (=, <, >, ...), các phép toán logic (NOT, AND, OR);

Các toán hạng: gồm các hằng, các biến, các hàm được định nghĩa trong TURBO PASCAL Thứ tự ưu tiên các phép toán như sau:

1. dấu ngoặc () nội dung viết trong dấu ngoặc được ưu tiên thực hiện trước nhất
2. phép lấy đối (-), phép NOT
3. các phép tính loại nhân, chia: *, /, DIV, MOD, AND
4. các phép tính loại cộng, trừ: +, -, OR
5. các phép so sánh: +, <>, <, >, ...

Chú ý: Biểu thức trong Pascal chỉ được viết trên 1 dòng do đó phải chú ý dùng các dấu () để viết cho chính xác.

5.3.4 Câu lệnh

Câu lệnh xác định công việc, thao tác mà chương trình phải thực hiện để xử lí các dữ liệu đã mô tả, khai báo. Các lệnh viết cách nhau bởi dấu (;), TURBO PASCAL phân loại các lệnh như sau:

- Lệnh đơn: là các lệnh không chứa các lệnh khác
- Lệnh có cấu trúc: thường chứa nhiều hơn một lệnh đơn, bao gồm khối lệnh, các lệnh thử và rẽ nhánh, các lệnh lặp.

- Lệnh hợp thành (còn gọi là lệnh phức hay lệnh ghép): gồm một nhóm các lệnh được đặt giữa cặp từ khoá begin ... end; (dấu chấm phẩy (;) kết thúc chứ không phải dấu chấm) như sau:

```
begin
    lệnh1;
    lệnh2;
    ...
    lệnhn;
end;
```

Dạng lệnh này cho ta thấy tính có cấu trúc của TURBO PASCAL. Một nhóm các lệnh giải quyết trọn vẹn một công việc nào đó sẽ được tổ chức trong một khối lệnh dạng này, và từng công việc của bài toán lớn sẽ được giao cho từng người làm trước khi ghép lại để giải bài toán tổng thể. Chúng ta sẽ đề cập kĩ hơn về vấn đề này ở phần chương trình con (thủ tục và hàm).

5.4 Lệnh nhập và xuất dữ liệu

5.4.1 Lệnh nhập dữ liệu

Các biến của chương trình có thể nhận giá trị thông qua phép gán ở trong chương trình, tuy nhiên, để gán giá trị cho biến thông qua các thiết bị nhập chuẩn như bàn phím ta dùng một trong 2 thủ tục sau

```
read( danh sách các tên biến);
readln(danh sách các tên biến);
```

ý nghĩa: nhập giá trị cho từng biến trong danh sách tên biến bằng cách gõ

vào từ bàn phím. Các biến trong danh sách các tên biến viết cách nhau bởi dấu phẩy (,). Khi nhập các giá trị cụ thể, các giá trị phải phù hợp với các biến trong danh sách về số lượng biến, kiểu của biến và vị trí các biến trong danh sách. Các giá trị số được viết cách nhau bởi ít nhất một dấu khoảng trắng (dấu cách).

Các biến trong thủ tục này cho phép là các kiểu sau: Nguyên, thực, kí tự, chuỗi kí tự, khoảng con,... Không được là kiểu logic.

Ví dụ: với các khai báo ở trên (bài 2.III), ta có thể viết :

```
read(x,y,z);  
readln(t);
```

khi chạy chương trình ta có thể gõ: 3 6 4 ↵

N↵

nghĩa là biến x nhận giá trị 3, biến y nhận giá trị 6, biến z nhận giá trị 4 còn biến t nhận giá trị N (kiểu char).

Hai cách viết read và readln có ý nghĩa như nhau về tác dụng nhập dữ liệu, chỉ khác là sau khi thực hiện xong chức năng này thì lệnh readln sẽ đưa con trỏ về đầu dòng tiếp theo, còn lệnh read thì không.

Ngoài 2 cách viết trên còn có thủ tục readln; không có tham số cho phép dừng chương trình chờ gõ một phím bất kì trước khi tiếp tục.

Chú ý: Khi vào dữ liệu cho biến kí tự hoặc biến chuỗi kí tự ta dùng thủ tục Readln(..) và mỗi biến trong một thủ tục.

Ví dụ: Đầu chương trình khai báo các biến

```
Var t: string; p: char; i,j,k : integer;
```

Trong thân chương trình vào dữ liệu cho các biến trên có thể viết như sau:

```
Readln(t); readln(p); readln(i,j,k);
```

5.4.2 Lệnh xuất dữ liệu

Có 2 cách viết dữ liệu ra màn hình:

```
write( các mục cần ghi ra );  
writeln( các mục cần ghi ra );
```

- ý nghĩa: đưa ra màn hình các kết quả tính toán trong chương trình, giá trị các biến, hay tất cả những gì nằm giữa cặp dấu nháy ' ... '

- Ví dụ: `write(x,y);` In ra màn hình các giá trị x và y, kết quả trên màn hình là 36

Có thể viết thành

```
write(x);
```

```
write(y);
```

hay rõ hơn

```
write('x=',x);write('y=',y);
```

 thì kết quả là `x=3 y=6`

các kí tự 'x=' và 'y=' nằm trong cặp dấu ngoặc chỉ có tác dụng trình bày, làm phân biệt rõ các giá trị được viết ra.

Hai cách viết trên có ý nghĩa như nhau về việc thể hiện dữ liệu ra màn hình, tuy nhiên thủ tục `writeln` sẽ chuyển con trỏ xuống đầu dòng tiếp theo sau khi thực hiện, con thủ tục `write` giữ nguyên vị trí con trỏ.

Ngoài 2 dạng trên còn có thủ tục `writeln`; không có tham số chỉ có tác dụng đưa ra một dòng trắng không chứa gì.

5.4.3 Kết hợp `read`, `readln` và `write`, `writeln` để nhập dữ liệu

Ta có thể kết hợp các thủ tục trên để tạo ra dạng nhập dữ liệu sáng sủa và đẹp mắt theo mẫu: `write(' câu nhắc nhập dữ liệu '); readln(biến);`

Ví dụ:

```
write(' Hãy nhập dữ liệu cho biến x:'); hay write('x=');
```

```
readln(x); readln(x);
```

kết quả sẽ là:

```
Hãy nhập dữ liệu cho biến x:( ta gõ ) 3 hay x= ( ta gõ ) 3
```

nên viết câu dẫn nhập gợi ý biến sẽ được nhập vào, không nên bỏ qua hoặc viết

```
writeln(' Hãy nhập dữ liệu cho biến x:'); hay writeln('x=');
```

```
read(x); read(x);
```

vì như thế khi chạy chương trình con trỏ nhập sẽ nhảy xuống dòng dưới dòng hướng dẫn, nhìn không đẹp mắt thế này

Hãy nhập dữ liệu cho biến x:

(ta gõ) 3

hay x=

(ta gõ) 3

5.5 Các lệnh có cấu trúc

5.5.1 Câu lệnh điều kiện

a) Dạng lệnh

❖ Dạng đơn giản

IF <điều kiện> THEN <lệnh >;

trong đó

<điều kiện> là một biểu thức điều kiện liên hệ bởi các toán tử: =, <, >, <=, >= và (hoặc) các phép toán logic NOT, AND, OR

<lệnh > có thể là một câu lệnh đơn giản hoặc phức hợp.

- ý nghĩa : khi gặp lệnh này, trước tiên máy sẽ kiểm tra <điều kiện>, nếu điều kiện> là đúng thì sẽ cho thực hiện <lệnh >, còn nếu <điều kiện> sai thì máy sẽ bỏ qua <lệnh > và chuyển sang công việc tiếp theo.

❖ Dạng tổng quát

IF <điều kiện > THEN <lệnh1 > ELSE <lệnh2 >;

- ý nghĩa : khi gặp lệnh này, trước tiên máy cũng sẽ kiểm tra <điều kiện>, nếu <điều kiện> là đúng thì sẽ cho thực hiện <lệnh1 >, còn trái lại là <điều kiện> sai thì máy sẽ thực hiện <lệnh2 >.

b) Ví dụ áp dụng

VD1: Viết chương trình nhập vào một số a tùy ý, rồi kiểm tra nếu a không âm thì in ra căn bậc 2 của a, trái lại in ra thông báo 'số âm không có căn bậc 2'

Chương trình được viết như sau:

```
Program    canbac2;
Uses      crt;
Var       a: real;
Begin
    write(' Hay nhap vao so a:');
```



```

    readln(a);
    if a >= 0 then writeln(' Can bac hai cua a la:', sqrt(a))
        else writeln(' So am khong co can bac hai');
    Readln;
End.

```

VD2: Viết chương trình giải phương trình bậc hai $ax^2+bx+c=0$

```

Program    GPTB2;
Uses crt;
Var
    a,b,c,x1,x2,dta:real;

Begin
    clrscr;
    gotoxy(5,5);
    writeln('CHUONG TRINH GIAI PHUONG TRINH BAC HAI');
    writeln;
    write('Vao cac he so:');
    readln(a,b,c);
    dta:=b*b-4*a*c;
    if dta>0 then
        begin
            writeln('Phuong trinh co 2 nghiem thuc:');
            X1:=(-b+sqrt(dta))/(2*a);
            X2:=(-b-sqrt(dta))/(2*a);
            writeln('X1=',X1:6:2);
            writeln('X2=',X2:6:2);
        end;
    if dta=0 then

```

```

        writeln('Phuong trinh co nghiem kep X=',-b/(2*a):6:2);
if dta<0 then
    begin
        writeln('Phuong trinh co 2 nghiem phuc:');
        writeln('X1=',-b/(2*a):6:2,'-',sqrt(-dta):6:2,'i');
        writeln('X2=',-b/(2*a):6:2,'+',sqrt(-dta):6:2,'i');
    end;
    readln;
End.

```

Lưu ý: - Câu lệnh phức hợp: trong chương trình có những chỗ TURBO PASCAL chỉ cho phép viết một lệnh, nhưng ta lại muốn viết nhiều hơn một lệnh thành phần thì các lệnh thành phần đó phải được đặt trong cặp từ khoá Begin ... end; (dấu chấm phẩy kết thúc). Chẳng hạn trong ví dụ trên, ứng với trường hợp $dta > 0$ máy phải thực hiện cả 5 lệnh trong cặp từ khoá begin...end; sau từ khoá then, còn ứng với trường hợp $dta = 0$ thì máy chỉ phải thực hiện đúng một lệnh, do đó không cần dùng câu lệnh phức hợp.

- Các câu lệnh điều kiện có thể viết lồng nhau. Tức là các <lệnh1> và <lệnh2> lại có thể là câu lệnh điều kiện dạng IF...THEN IF...THEN...ELSE... khi đó cần lưu ý IF nào đi với THEN nào. Chẳng hạn xem đoạn chương trình phân loại kết quả học tập sau:

```

IF diem>=5 THEN
    IF diem>=7 THEN loai:='Kha gioi' ELSE loai:='TB'
ELSE
    IF diem>=3 THEN loai:='Yeu' ELSE loai:='Kem';

```

5.5.2 Câu lệnh lựa chọn : CASE..OF

Câu lệnh điều kiện chỉ cho phép ta thực hiện rẽ 2 nhánh ứng với hai giá trị đúng hay sai của biểu thức điều kiện. Để có thể thực hiện rẽ nhiều nhánh ứng với nhiều giá trị khác nhau của một biểu thức, ta phải sử dụng câu lệnh lựa chọn.

a) Dạng lệnh

Dạng đơn giản :

```
CASE <biểu thức> OF  
    hằng1: <lệnh 1>;  
    hằng2: <lệnh 2>;  
    ....  
    hằngn: <lệnh n>;  
END;
```

Dạng tổng quát :

```
CASE <biểu thức> OF  
    hằng1: <lệnh 1>;  
    hằng2: <lệnh 2>;  
    ....  
    hằngn: <lệnh n>;  
ELSE <lệnh n+1>;  
END;
```

Trong đó <biểu thức> và các hằng phải có cùng kiểu và phải là các kiểu vô hướng đếm được (các kiểu số nguyên kiểu kí tự, kiểu Boolean đã biết).

ý nghĩa:

Khi gặp câu lệnh rẽ nhánh, <biểu thức> đã được nhận một giá trị nào đó, giá trị này là một hằng (số hoặc kí tự). Nếu <biểu thức> nhận giá trị là hằng nào thì <lệnh> tương ứng với nó sẽ được thực hiện. Còn nếu <biểu thức> nhận giá trị không rơi vào hằng nào thì máy sẽ bỏ qua lệnh đối với dạng đơn giản, máy sẽ thực hiện <lệnh n+1> sau từ khoá ELSE đối với dạng tổng quát.

Lưu ý: Từ khoá END với dấu chấm phẩy (;) ở trong câu lệnh này để chỉ rằng kết thúc câu lệnh lựa chọn chứ không phải kết thúc chương trình con.

b) Ví dụ áp dụng

VD1: Viết chương trình xem thời khoá biểu của một ngày trong tuần. Yêu cầu: máy in ra câu hỏi 'Bạn muốn xem thời khoá biểu của thứ mấy?' ta gõ vào ngày thứ (của tuần) muốn xem và máy sẽ in ra thời khoá biểu của ngày hôm đó.

Chương trình được viết như sau: (giả sử xem thời khoá biểu của một lớp phổ thông, bạn đọc có thể hiệu chỉnh theo ý mình)

```
Program      XEM_TKB;  
Uses      crt;  
Var      thu: 2..7; {biến thu để chứa các thứ ngày trong tuần}
```

Begin

```
write(' Ban muon xem thoi khoa bieu cua thu may?');  
readln(thu);  
CASE thu OF  
    2:writeln('Toan  Ly  Van');  
    3:writeln('Sinh  Ki  Hoa');  
    4: writeln('Toan  Hoa  Ly');  
    5: writeln('Van  Sinh  The duc');  
    6: writeln('Su   Dia  Chinh tri');  
    7: writeln('Van  Toan  Sinh hoat');  
END;  
Readln;
```

End.

Bạn đọc có thể áp dụng câu lệnh dạng tổng quát để mở rộng chương trình xem thời khoá biểu này sao cho khi gõ vào một số nào đó không phải là thứ trong tuần (2..7) máy sẽ in ra thông báo rằng đó là ngày nghỉ.

Lưu ý:

- Sau mỗi hàng máy chỉ thực hiện đúng một lệnh. Do đó nếu muốn dùng nhiều hơn một lệnh thì ta phải sử dụng câu lệnh phức hợp.

- Các hàng có thể viết gộp lại với nhau. Chẳng hạn nếu thứ 3 và thứ 5 có cùng thời khoá biểu thì thay vì phải viết 2 dòng lệnh, ta có thể viết:

```
3,5: writeln('Sinh  Ki  Hoa');
```

- Các câu lệnh lựa chọn có thể lồng nhau. Tức là trong câu lệnh lựa chọn lại có thể chứa câu lệnh lựa chọn khác.

5.5.3 Lệnh lặp có số lần xác định

Đây là một câu lệnh có cấu trúc cho phép thực hiện lặp đi lặp lại một công việc (đoạn lệnh) nào đó với số lần thực hiện được xác định trước. để dễ hiểu trước hết ta xét một ví dụ đơn giản sau:

Giả sử ta phải viết ra các số từ 1 đến 10, mỗi số chiếm một dòng trên màn

hình. Ta có thể thực hiện công việc này bởi 10 lệnh writeln như sau:

```
writeln(1);
```

```
writeln(2);
```

```
writeln(10);
```

Đây là cách viết dài dòng và đơn điệu trong khi ta có thể thực hiện công việc trên bởi một dòng lệnh ngắn gọn:

```
FOR I:=1 TO 10 DO writeln(I);
```

Ý nghĩa của câu lệnh này là: cho một biến nhận giá trị nguyên I chạy từ 1 đến n (biến I lần lượt nhận các giá trị số nguyên từ 1 đến n), với mỗi giá trị của I máy sẽ thực hiện công việc sau từ khoá DO là in ra giá trị của I.

Chi tiết các bước thực hiện của vòng lặp FOR này như sau:

Đầu tiên biến I nhận giá trị khởi đầu là 1 (do lệnh gán I:=1), máy kiểm tra giá trị này không vượt quá giá trị cuối là 10 nên cho thực hiện lệnh writeln(I) viết ra giá trị 1. Sau đó biến I được tăng thêm một đơn vị, tức là I:=I+1. Bây giờ I=2, chưa vượt qua giá trị cuối là 10 nên lệnh writeln(I) lại được thực hiện để in ra giá trị của I là 2 lên màn hình. Rồi I lại được tăng lên... giá trị cuối cùng của I để lệnh writeln(I) được thực hiện là I:=10. Quá trình kết thúc khi I nhận giá trị 11 vượt quá giá trị cuối 10.

Câu lệnh trên là một ví dụ đơn giản của lệnh tổng quát sau:

a) Dạng lệnh lặp tăng

```
FOR Biến_điều_khiển:=Biểu_thức1 TO Biểu_thức2 DO <Lệnh>;
```

Trong đó Biến_điều_khiển, Biểu_thức1 và Biểu_thức2 phải cùng kiểu dữ liệu và phải là các kiểu vô hướng đếm được (như các kiểu số nguyên, kiểu kí tự...) ý nghĩa của câu lệnh này như sau: Trước tiên Biến_điều_khiển nhận giá trị khởi tạo là Biểu_thức1, máy kiểm tra nếu giá trị đó không vượt quá Biểu_thức2 thì cho thực hiện <Lệnh>. Thực hiện xong lệnh này, Biến_điều_khiển được tăng thêm một đơn vị, máy lại kiểm tra với Biểu_thức2 để thực hiện <Lệnh>... Quá trình tiếp diễn đến khi Biến_điều_khiển nhận giá trị vượt quá Biểu_thức2 thì dừng.

Số lần thực hiện lặp đi lặp lại <Lệnh> được xác định trước bởi giá trị của Biểu_thức1 và Biểu_thức2 nên câu lệnh trên còn được gọi là lệnh lặp có số lần lặp xác định.

❖ Ví dụ áp dụng

VD1: Viết chương trình tính tổng của n số tự nhiên đầu tiên $S=1+2+\dots+n$ theo phương pháp cộng dồn.

Phương pháp cộng dồn được thực hiện như sau: ban đầu ta khởi tạo $S:=0$, rồi lấy S cộng với 1, rồi cộng với 2,... cộng đến n. Như vậy ta sẽ phải sử dụng một vòng lặp FOR với một biến điều khiển I chạy từ Biểu_thức1 là 1 đến Biểu_thức2 là n. Ta sẽ dùng chính giá trị của biến điều khiển này để tính tổng S.

Chương trình được viết như sau:

```
Program    Tinh_tong;
Uses      crt;
Var       i,n:integer;
          S:real;
Begin
    Write('Cho biet gia tri cua n:'); Readln(n);
    S:=0;
    For I:=1 to n do S:=S+i;
    Writeln(' Tong tinh duoc la S:',S:10:2);
    Readln;
End.
```

VD2: Một người gửi tiền tiết kiệm với số tiền ban đầu là A đồng, thời gian gửi t tháng, lãi suất tháng là p. Hỏi sau t tháng gửi người đó thu được số tiền lãi là bao nhiêu đồng?

Phân tích bài toán: Ta biết rằng gửi tiết kiệm thì số tiền thu được sau mỗi tháng sẽ bằng số tiền tháng trước cộng với lãi của tháng đó. Nếu gọi S là số tiền thu được sau mỗi tháng, t là số tháng gửi thì ta có:

Ban đầu $S_0=A$

$t=1 \rightarrow S_1=S_0 + S_0 * p$ (gửi 1 tháng)

$t=2 \rightarrow S_2= S_1 + S_1 * p$ (gửi 2 tháng)

$t=n \rightarrow S_n= S_{n-1} + S_{n-1} * p$ (gửi n tháng)

Đây chính là quy luật để thực hiện vòng lặp tính số tiền thu được. Chương trình được viết như sau:

```
Program    Tinh_tien_lai;
Uses      crt;
Var       i,t:integer;
A,P,S:real;
Begin
    Write('Cho biet so tien gui ban dau: '); Readln(A);
    Write('Cho biet so thang gui: '); Readln(t);
    Write('Cho biet lai suat theo thang: '); Readln(P);
    S:=A;
    For I:=1 to t do S:=S+ S*P;
    Writeln(' Lai thu duoc la : ',S-A:10:2);
    Readln;
End.
```

b) Dạng lệnh lặp giảm

FOR Biến_điều_khiển:=Biểu_thức1 DOWNTO Biểu_thức2 DO <Lệnh>;

ý nghĩa hoàn toàn tương tự như dạng trên, chỉ khác là Biến_điều_khiển nhận giá trị giảm dần chứ không phải tăng dần, và điều kiện kiểm tra là Biến_điều_khiển không nhỏ quá Biểu_thức2.

Với dạng này thì câu lệnh FOR trong ví dụ 1 ở trên được viết là:

```
For I:=n downto 1 do S:=S+i;
```

Tổng S sẽ được cộng dồn theo thứ tự từ n đến 1, tức là $S=n+(n-1)+\dots+2+1$.

Ví dụ: Viết chương trình in ra các kí tự trong bảng mã ASCII theo thứ tự giảm dần.

Các kí tự trong bảng mã ASCII có mã giảm dần từ 255 đến 0. Ta dùng hàm CHR(n) để nhận được kí tự.

Chương trình được viết như sau:

```
Program In_cac_ki_tu;
    Uses crt;
    Var I:integer;
    Begin
        Clrscr;
        For i:=255 downto 0 do write(chr(i):4);
        Readln;
    End.
```

c) Các lưu ý

1) Sau từ khoá DO chỉ được viết một lệnh, do đó nếu cần thực hiện nhiều hơn một lệnh đơn thì phải sử dụng câu lệnh phức hợp

2) Các lệnh lặp có thể lồng nhau, chẳng hạn dạng:

```
FOR I:=1 TO n DO
    FOR J:=1 TO m DO
        <Lệnh>
```

Khi đó với mỗi giá trị của biến điều khiển của vòng lặp ngoài thì biến điều khiển của vòng lặp trong sẽ chạy hết các giá trị của nó. Tức là với mỗi giá trị của I thì J sẽ chạy từ 1 tới m. Xét ví dụ đoạn chương trình sau:

```
For i:=1 to 2 do
    For j:=1 to 3 do
    begin
        k:=i+j;
        writeln(k);
    end;
```

sẽ cho kết quả là:

2

3
4
3
4
5

3) <Lệnh> sau từ khoá DO không được tùy tiện thay đổi giá trị của Biến_điều_khiển, làm như vậy ta có thể không kiểm soát được giá trị của Biến_điều_khiển và có thể làm rối vòng lặp.

Ví dụ: In ra tất cả các chữ số có 3 chữ số mà tổng các chữ số chia hết cho 3. Trong bài toán này mỗi số đều có 3 chữ số, chữ số hàng trăm có thể nhận giá trị từ 1 đến 9, chữ số hàng chục, chữ số hàng đơn vị đều có thể nhận giá trị từ 0 đến 9. để quét hết các chữ số có 3 chữ số ta dùng 3 vòng lặp lồng nhau, sau đó ta kiểm tra điều kiện nếu tổng các chữ số chia hết cho 3 thì in ra số đó.

Chương trình được viết như sau:

```
Program In_cac_so;  
Uses crt;  
Var t,c,v:integer;  
Begin  
  Clrscr;  
  For t:=1 to 9 do  
    For c:=0 to 9 do  
      For v:=0 to 9 do  
        If (t+c+v) mod 3 = 0 then writeln(t,c,v);  
  Readln;  
End.
```

5.5.4 Lệnh lặp có số bước lặp không xác định

Ở trên ta đã xét lệnh lặp có số bước lặp xác định, tức là số lần thực hiện công việc lặp đi lặp lại là được định trước. Nhưng trong khi lập trình có những bài toán yêu cầu thực hiện một công việc nào đó mà số lần lặp là không thể xác định trước được, mà

phụ thuộc vào một biểu thức điều kiện nào đó. Có hai dạng lệnh lặp với số bước lặp không xác định.

a) Lệnh lặp với điều kiện trước

❖ Dạng lệnh:

WHILE <điều kiện> DO <Lệnh>;

Ý nghĩa:

Khi gặp câu lệnh này, trước tiên máy sẽ kiểm tra <điều kiện>. Nếu <điều kiện> có giá trị đúng (TRUE) thì <Lệnh> được thực hiện. Thực hiện xong lệnh này máy sẽ quay lại kiểm tra <điều kiện> để thực hiện <Lệnh>... quá trình tiếp diễn đến khi <điều kiện> sai thì dừng.

❖ Ví dụ:

VD1: Một người gửi tiết kiệm không kì hạn với số tiền ban đầu là A đồng. Hỏi sau bao nhiêu tháng người đó thu được số tiền là B đồng, biết rằng lãi suất là 1.8%?

Phân tích bài toán: Ta biết rằng gửi tiết kiệm không kì hạn thì số tiền thu được sau mỗi tháng sẽ bằng số tiền tháng trước cộng với lãi của tháng đó. Nếu gọi S là số tiền thu được sau mỗi tháng, t là số tháng gửi, ls là lãi suất thì ta có:

nếu $t=0 \rightarrow S_0=A$ (gửi 0 tháng)

nếu $t=1 \rightarrow S_1=S_0 + S_0*ls$ (gửi 1 tháng)

nếu $t=2 \rightarrow S_2= S_1 + S_1*ls$ (gửi 2 tháng)

....

Đây chính là quy luật để thực hiện vòng lặp tính số tiền thu được. Như vậy ta phải so sánh số tiền thu được S sau mỗi tháng với số tiền cần đạt B: nếu S đạt hoặc vượt B thì ta có kết luận của bài toán. đây cũng là điều kiện dừng vòng lặp. Ta sẽ sử dụng một biến đếm t để đếm số lần thực hiện lặp, cũng chính là số tháng gửi.

Chương trình được viết như sau:

Program Gui_tiet_kiem;

Const ls=0.018;

```

Var          A,B,S:real;
              t:integer;

Begin
  Clrscr;
  Write('Cho biet so tien gui ban dau:'); Readln(A);
  Write('Cho biet so tien can dat :'); Readln(B);
  S:=A;
  t:=0;
  WHILE S<B DO
  begin
    S:=S+S*Is;
    t:=t+1;
  end;
  Writeln(' Ban can gui it nhat la ', t, ' thang');
  Readln;
End.

```

VD2: Ta có thể viết lại chương trình tính tổng $S=1+2+\dots+n$ sử dụng lệnh lặp

WHILE...DO...như sau:

```

Program      Tinh_tong1;
Var          i,n:integer;
              S:real;

Begin
  Write('Cho biet gia tri cua n:'); Readln(n);
  S:=0;
  i:=1;
  While i<= n do
  begin
    S:=S+i;
    i:=i+1;
  end;

```

end;

Writeln(' Tong tinh duoc la S:',S:10:2);

End.

Như vậy ta có thể dùng lệnh lặp có số bước lặp không xác định để thay cho lệnh lặp có số bước lặp xác định. Tuy nhiên khi đó biến điều khiển vòng lặp không tự động thay đổi mà ta phải đưa nó vào thân vòng lặp (biến i trong VD trên).

b) Lệnh lặp với điều kiện sau

❖ Dạng lệnh:

REPEAT <Lệnh> UNTIL <điều kiện>;

Ý nghĩa:

Khi gặp câu lệnh này, trước tiên máy sẽ cho thực hiện <Lệnh>, sau đó kiểm tra <điều kiện>. Nếu <điều kiện> có giá trị sai (FALSE) thì quay lại thực hiện <Lệnh>. Thực hiện xong lệnh này máy sẽ lại kiểm tra <điều kiện> để thực hiện <Lệnh>... quá trình tiếp diễn đến khi <điều kiện> đúng thì dừng.

❖ Ví dụ: Lấy lại VD gửi tiết kiệm ở trên. Bây giờ ta sẽ viết lại chương trình sử dụng dạng lệnh lặp với điều kiện sau. Chương trình chỉ cần thay đổi một chút ở lệnh lặp.

Program Gui_tiet_kiem;

Const ls=0.018;

Var A,B,S:real;

t:integer;

Begin

Write('Cho biet so tien gui ban dau:'); Readln(A);

Write('Cho biet so tien can dat :'); Readln(B);

S:=A;

t:=0;

REPEAT

S:=S+S*ls;

```
t:=t+1;
UNTILS>=B;
  Writeln(' Ban can gui it nhat la ', t, ' thang');
Readln;
```

End.

Ta thấy rằng trong phần thân của vòng lặp các lệnh đặt giữa REPEAT và UNTIL không cần sử dụng câu lệnh phức hợp begin...end;

Bạn đọc có thể sử dụng lệnh này để viết lại chương trình tính tổng ở trên.

c) Các lưu ý khi sử dụng các lệnh lặp WHILE và REPEAT

1) Trong thân vòng lặp phải có ít nhất một lệnh làm thay đổi giá trị của biểu thức điều kiện nhằm dừng vòng lặp. Nếu điều kiện luôn được thoả mãn thì lệnh có thể rơi vào vòng lặp vô tận. Khi đó chỉ còn cách là tắt máy hoặc dừng chương trình.

Xem ví dụ sau:

```
I:=1;
WHILE I<2 DO write('Stop!');
```

Vòng lặp này sẽ chạy vô tận vì trong thân vòng lặp sau từ khoá DO chỉ có một lệnh in ra xâu chữ 'Stop!', không có lệnh nào làm thay đổi giá trị của I. Tức là với I được khởi tạo ban đầu là 1 luôn nhỏ hơn 2, điều kiện luôn thoả mãn.

2) Trong lệnh lặp với điều kiện trước WHILE thì thân vòng lặp (sau từ khoá DO) phải sử dụng câu lệnh phức hợp begin... end; còn trong lệnh lặp với điều kiện sau thì không cần.

3) Trong lệnh lặp với điều kiện trước, máy luôn kiểm tra điều kiện trước khi thực hiện lệnh, do đó nếu điều kiện sai ngay từ đầu thì lệnh không được thực hiện lần nào. Còn trong lệnh lặp với điều kiện sau, máy luôn thực hiện lệnh một lần rồi mới kiểm tra điều kiện, dù ban đầu điều kiện có thể đúng. Nhớ rằng trong lệnh lặp với điều kiện sau lệnh được quay lại thực hiện chỉ khi điều kiện sai.

4) Các lệnh lặp có thể viết lồng nhau

5) Người ta thường dùng các lệnh lặp có số bước lặp không xác định để quay vòng thực hiện nhiều lần cả một công việc nào đó hoặc cả chương trình, cho phép tạo sự tương tác giữa người sử dụng và máy. Xem ví dụ mẫu sau:

```
Program          VD;
Var  tieptuc:char;
      ...{ các biến của chương trình }
Begin
      Repeat
          .....
          { đoạn chương trình thực hiện công việc và thay
            đổi điều kiện kiểm tra }
          write(' Ban co tiep tuc nua hay khong (C/K)');
          readln(tieptuc);
      UNTIL (tieptuc='k') or (tieptuc='K');
End;
```

d) Lệnh nhảy vô điều kiện GOTO

Lệnh GOTO cho phép chương trình nhảy vô điều kiện tới một vị trí nào đó thông qua tên nhãn. Nhãn là một số nguyên hoặc một tên được khai báo trong phần LABEL ở đầu chương trình. Trong chương trình nhãn được đặt vào vị trí phù hợp kèm theo một dấu hai chấm (:).

Xem ví dụ sau:

```
Program          VDNHAN;
Uses  crt;
Label  N1,N2;
Var  a,b,s:real;
Begin
          a:=1; b:=2;
      N1: a:=a+1;
```

```
if a<5 then GOTO N1;
N2: s:=a+b;
if s<10 then GOTO N2;
writeln('a=', a, 'b=',b);
readln;
```

End.

Thường người ta ít dùng lệnh GOTO trong chương trình PASCAL vì nó sẽ phá vỡ tính có cấu trúc của chương trình PASCAL. Nếu có thể, ta nên tránh dùng lệnh này.

Thực hành chương 5

Câu 1: Viết chương trình nhập vào chiều dài và chiều rộng của một hình chữ nhật sau đó tính và in ra diện tích hình chữ nhật này.

Câu 2: Nhập vào 2 số nguyên sau đó in ra số lớn nhất và số nhỏ nhất.

Câu 3: Nhập vào 3 số nguyên sau đó in ra số lớn nhất và số nhỏ nhất.

Câu 4: Viết chương trình nhập vào bán kính của hình tròn. Tính và in ra chu vi, diện tích của hình tròn đó.

Câu 5: Diện tích biết trước của một hình tròn được nhập vào từ bàn phím. Tính và in ra bán kính của hình tròn đó.

Câu 6: Nhập vào số nguyên x, tính và in ra $S = 1 + x + x^3/3 + x^5/5$.

Câu 7: Viết chương trình tính tổng số nguyên n.

Câu 8: Viết chương trình tìm mã ASCII của ký tự.

Câu 9: Viết chương trình tìm ký tự khi biết mã ASCII.

Câu 10: Viết chương trình đổi chữ hoa sang chữ thường.

Câu 11: Viết chương trình đổi chữ thường sang chữ hoa.

Câu 12: Viết chương trình nhập vào điểm 3 môn Toán, Lý, Hóa của một học sinh sau đó tính và in ra điểm trung bình và xếp hạng của học sinh đó.

Câu 13: Nhập vào một số nguyên n in ra các số lẻ từ 0 đến n .

Câu 14: Nhập vào một số kiểm tra xem đó có phải là số nguyên tố.

Câu 15: Nhập vào một số nguyên dương n với $1 \leq n \leq 7$. Tùy theo $n=1, 2, 3, \dots, 7$ hãy in tương ứng các từ Sunday, Monday, Tuesday, ..., Saturday ra màn hình.

Câu 16: Nhập vào một số tiền M nguyên dương. Đổi số tiền này ra các tờ giấy bạc 10.000Đ, 500Đ, 50Đ, 20Đ và 1Đ. Với giả thiết ưu tiên cho tờ có mệnh giá lớn hơn, hãy in ra xem đổi được bao nhiêu mỗi loại.

Câu 17: Cước phí bưu điện trong nước được tính theo qui định như sau:

- Nếu trọng lượng thư trong khoảng định mức 100 gam đầu thì cước phí trung bình là 300 đồng cho mỗi gam.
- Nếu trọng lượng thư vượt định mức không quá 30 gam thì cước phí trung bình cho mỗi gam vượt định mức là 200 đồng;
- Trường hợp trọng lượng thư vượt quá 130 gam thì tính trung bình 120 đồng cho mỗi gam vượt quá 130 gam.

Viết chương trình tính cước phí phải trả khi nhập vào trọng lượng của bức thư.

Câu 18: Tính tiền điện phải trả khi biết số Kwh tiêu thụ và các thức tính theo qui định như sau:

- 100 Kwh định mức đầu tiên có đơn giá trung bình là 455 đ/Kwh.
- Các Kwh thứ 101 đến 150 có đơn giá trung bình là 591 đ/Kwh.
- Các Kwh thứ 151 đến 200 có đơn giá trung bình là 818 đ/Kwh.
- Các Kwh thứ 201 trở đi có đơn giá trung bình 1000 đ/Kwh.

Câu 19: Tính tiền thuê phòng khi biết số ngày thuê và loại phòng (một trong 3 loại A,B hoặc C) với qui định như sau:

- Loại A: 250000 đ/ngày
- Loại B: 200000 đ/ngày
- Loại C: 150000 đ/ngày

Nếu thuê quá 12 ngày thì phần trăm được giảm trên tổng số tiền (tính theo giá qui định) là: 10% cho phòng loại A, 8% cho phòng loại B hay C.

Câu 20: Tính tiền trả cho một cuộc gọi điện thoại khi biết số phút gọi và hình thức gọi (liên lạc hay hỏi thông tin). Qui định chi tiết như sau:

- Liên lạc:

3 phút đầu tiên: 750đ (không gọi hết vẫn tính 3 phút) mỗi phút sau đó: 200 đ/phút.

- Hỏi thông tin:

5 phút đầu: 3000đ (không gọi hết vẫn tính 5 phút) mỗi phút sau đó 800 đ/phút.

Câu 21: Nhập vào trọng lượng một bưu phẩm và tính cước phí của bưu phẩm đó, biết rằng:

- Nếu bưu phẩm có trọng lượng nhỏ hơn 20kg, cước phí là 20.000đ cho một bưu phẩm.

- Nếu bưu phẩm có trọng lượng từ 20kg trở lên (nhưng không vượt quá 50kg), mỗi kg bưu phẩm trên 20kg có cước phí trung bình là 1.200 đ/kg.

- Nếu bưu phẩm có trọng lượng từ 50kg trở lên, mỗi kg bưu phẩm trên 50kg có cước phí trung bình là 1.000 đ/kg.

Câu 22: Tính tiền đi xe taxi khi biết số km đã đi: 1000m (tức 1km) đầu tiên giá 5000đ; từ km thứ 2 đến km thứ 5 giá trung bình là 4500 đ/km; từ km thứ 6 trở đi, giá trung bình mỗi km là 3500. Đi hơn 120 sẽ được giảm 10% trên tổng số tiền tính theo qui định.

Câu 23: Tính tiền thuê một quyển sách nếu biết số ngày thuê (không được vượt quá 30 ngày) và đối tượng thuê (sinh viên học sinh hay đối tượng khác). Các qui định về giá cả được cho như sau:

- Nếu mượn trong 7 ngày:

Đối tượng sinh viên học sinh: 1500đ, đối tượng khác: 3000đ

- Nếu mượn hơn 7 ngày, mỗi ngày trả thêm:

500đ với đối tượng sinh viên học sinh, 700 với đối tượng khác.

TÀI LIỆU THAM KHẢO

1. Phan Thị Hà – Nguyễn Tiến Hùng. Bài giảng Tin Học Đại Cương.
2. Đỗ Thị Mơ – Dương Xuân Thành – Nguyễn Thị Thủy – Ngô Tuấn Anh.
Giáo trình Tin Học Đại Cương.
3. Đỗ Thanh Liên Ngân – Hồ Văn Tú. Giáo trình Tin Học Căn Bản.

