



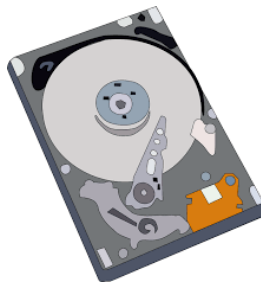
ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

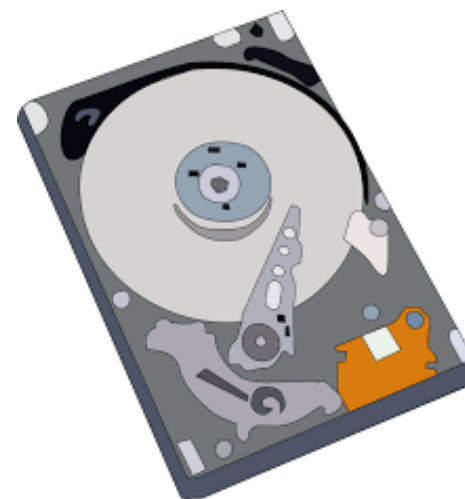
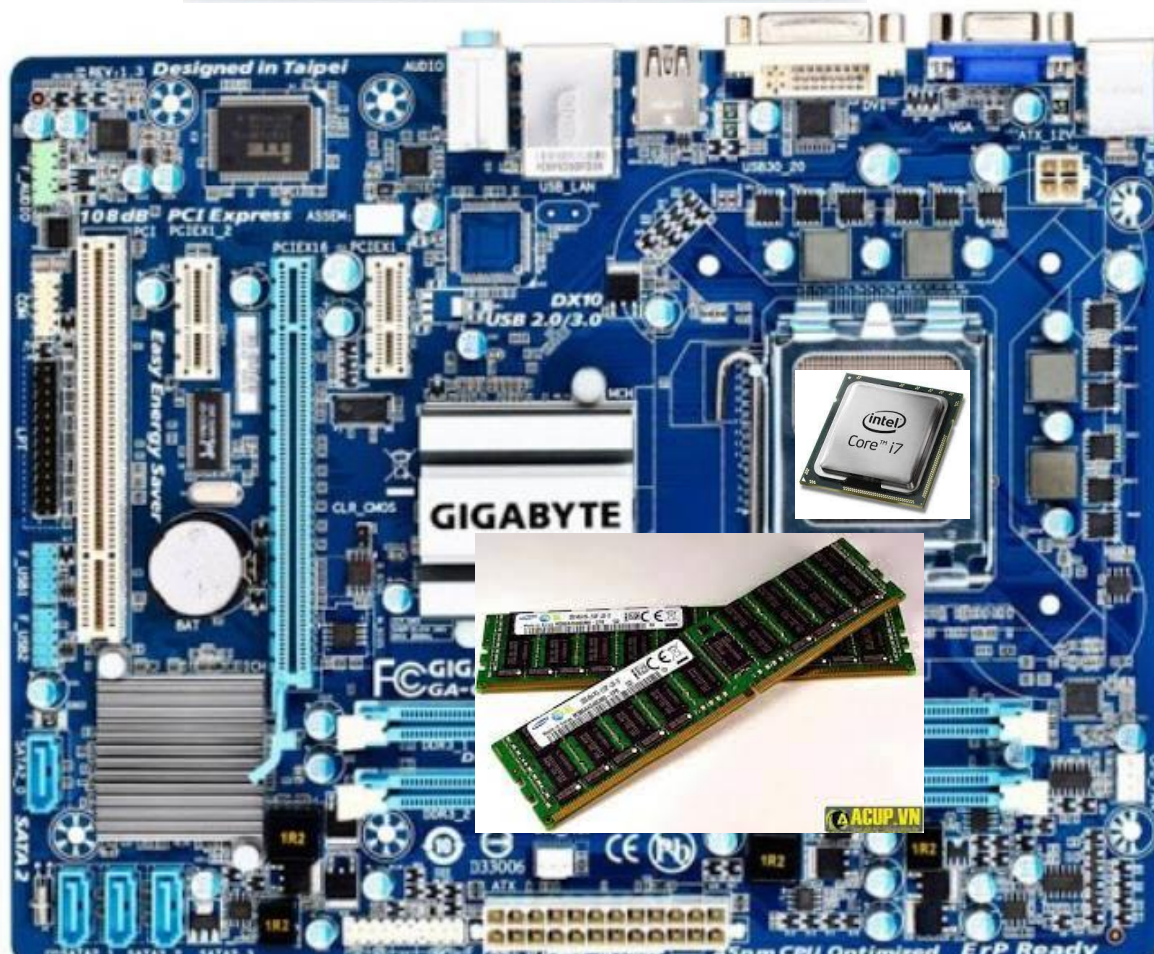
C PROGRAMMING INTRODUCTION

TUẦN 15: Tập / File

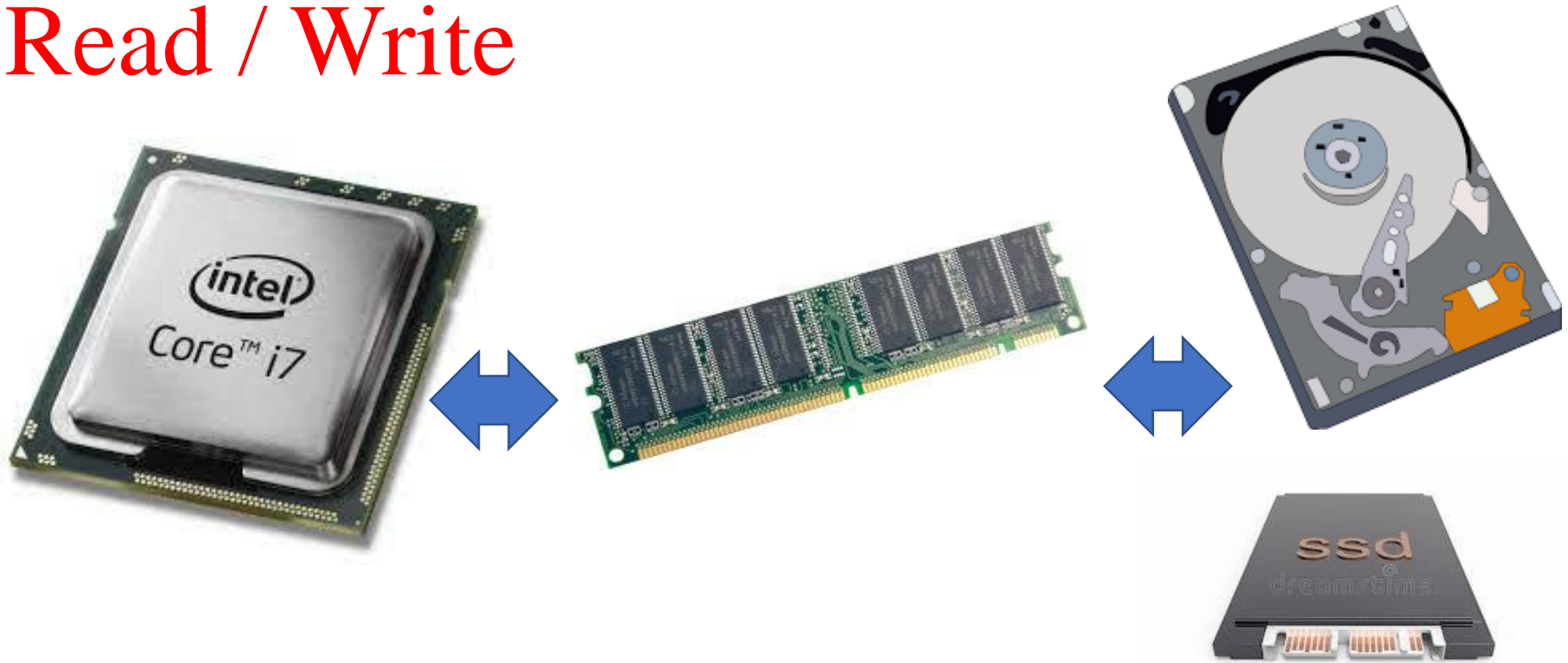
Khái niệm tệp

- Tệp dữ liệu (File) là một tập hợp các dữ liệu có liên quan với nhau và có cùng kiểu dữ liệu.
- Tệp được lưu trữ trên các **thiết bị nhớ ngoài** (đĩa cứng, USB, thẻ nhớ ...) với một tên nào đó để phân biệt với nhau.
- Vai trò của tệp: Tệp là phương tiện để cất giữ **dữ liệu lâu dài**, dữ liệu lưu trữ trong tệp **không bị mất đi** khi chương trình kết thúc hay khi tắt máy





Bộ nhớ: RAM (bộ nhớ trong) Ổ cứng (bộ nhớ ngoài) Read / Write



RAM: Random Access
Memory

My_file.txt
My_file_2.dat

Khái niệm

- Tập (Tập tin/File):
 - Tập hợp các dữ liệu cùng kiểu
 - Có liên quan tới nhau
- Lưu trữ tập
 - Lưu trữ trên thiết bị lưu trữ ngoài
 - Có tên riêng để phân biệt
- Phân thành 2 loại
 - Tập văn bản (text file)
 - Tập nhị phân (binary file)

Kiểu mảng?

Phân loại

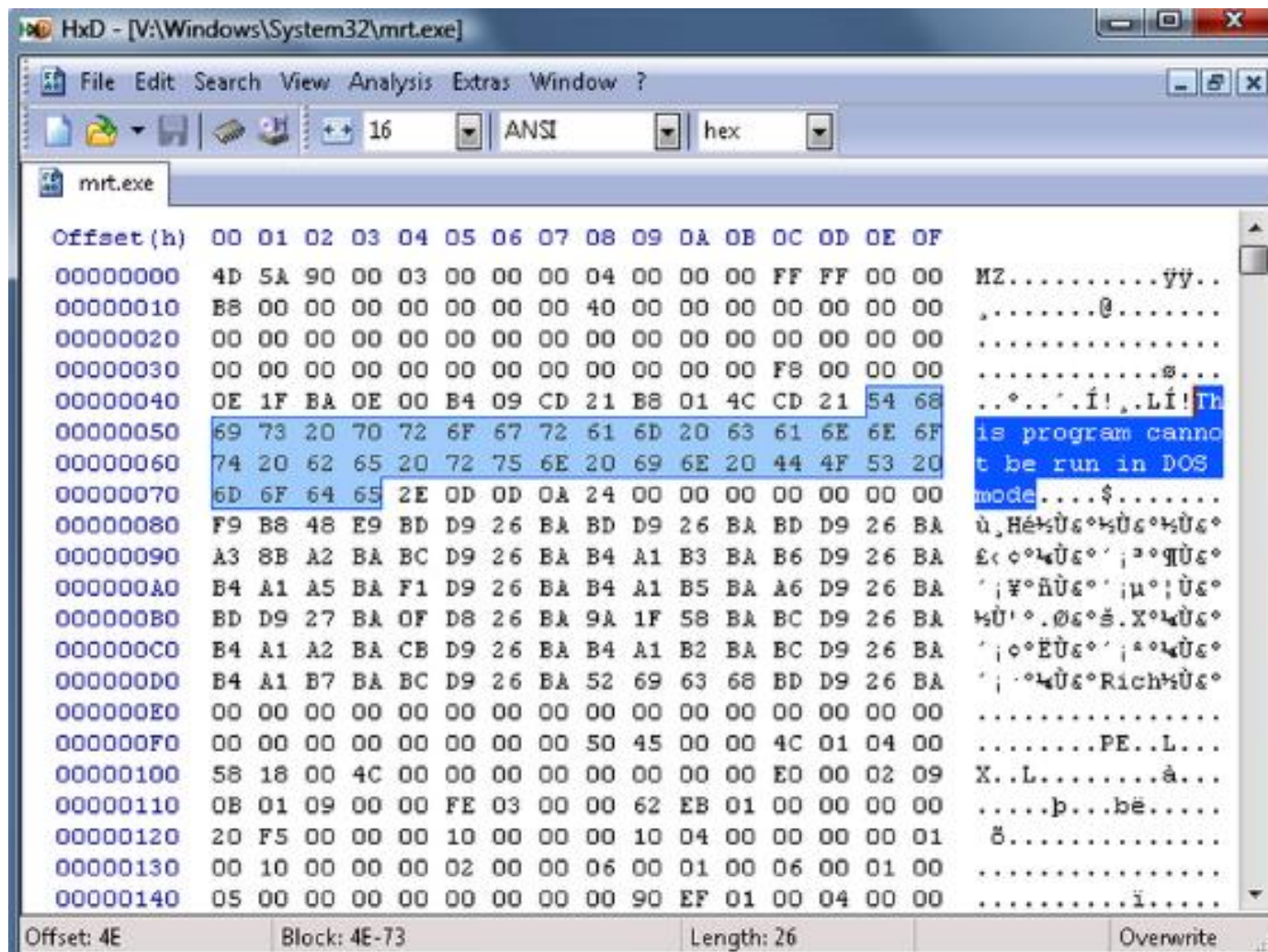
- Tập văn bản
 - Được tổ chức theo từng dòng
 - Trên mỗi dòng là **các ký tự ASCII hiển thị** được như chữ cái, chữ số, dấu câu,...
 - Cuối mỗi dòng là các ký tự điều khiển
 - CR: Carriage Return - mã ASCII 13
 - LF: Line Feed- Mã ASCII 10
- Tập nhị phân
 - Các phần tử của tập là các số nhị phân dùng mã hóa thông tin
 - Thông tin được mã hóa: số, cấu trúc dữ liệu, ...

ASCII Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SPC	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

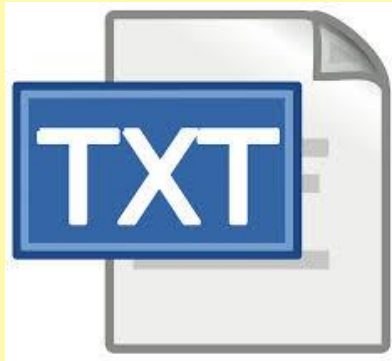
Physical Device Controls: Format Effectors

File nhị phân (Binary file)



Phân loại Tập (File)

File văn bản



ghi_chú.txt

```
1  #include <stdio.h>
2  int main() {
    FILE *fptr;
    fptr = fopen("fprintf_test.txt", "w");
    // Write to file
    fprintf(fptr, "Learning C with Guru99\n");
    close(fptr);
    return 0;
}
```

doc_file.c

File nhị phân



bai_hat.mp3



bao_cao.docx

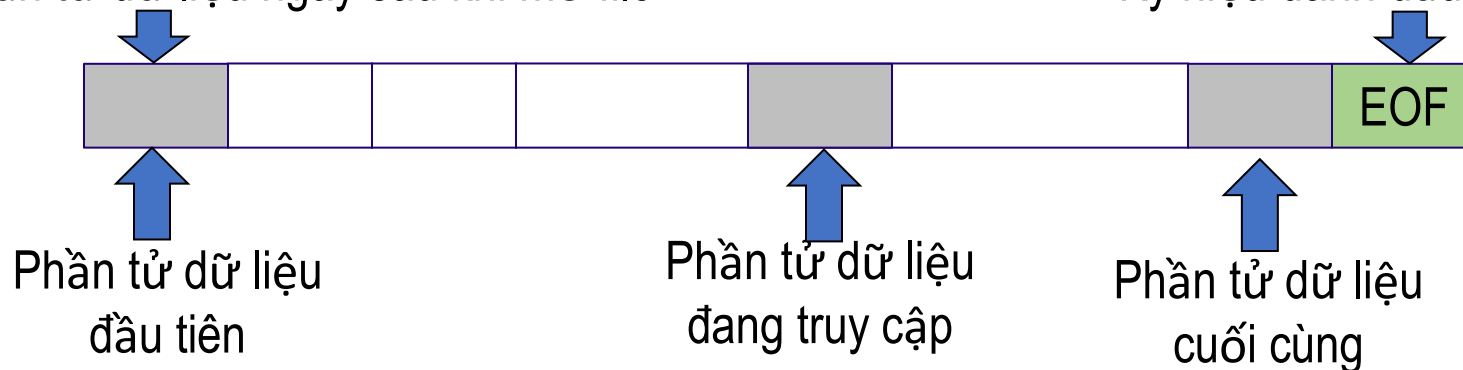


bang_diem.xls

Tổ chức tệp

Phần tử dữ liệu ngay sau khi mở file

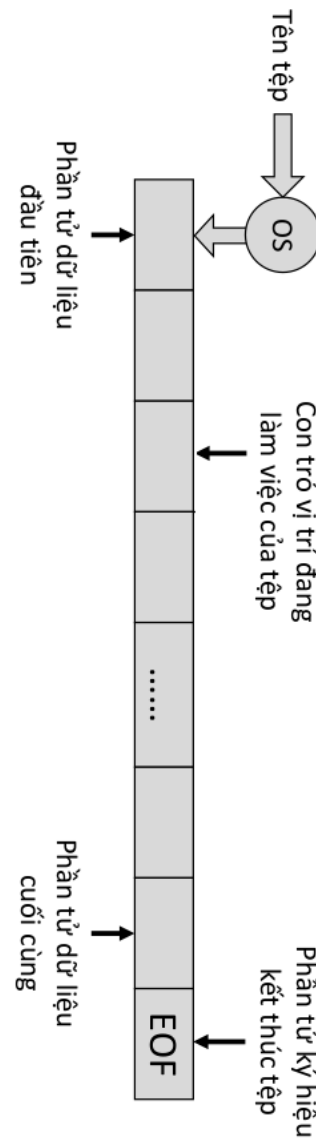
Ký hiệu đánh dấu kết thúc file



- Tệp là dãy các phần tử kế tiếp nhau
 - Sử dụng phần tử đặc biệt (EOF) để đánh dấu kết thúc tệp
- Con trỏ tệp:
 - Xác định vị trí phần tử hiện có thể truy cập
 - Khi mở file, con trỏ tệp luôn ở vị trí phần tử đầu
 - Sau các thao tác đọc/ghi tệp, con trỏ file dịch chuyển về cuối tệp một khoảng bằng số byte đã đọc/ ghi

Con trỏ tệp

- Khi **mở tệp**, con trỏ tệp sẽ luôn trở đến **phần tử đầu tiên** của tệp.
- Sau mỗi thao tác **đọc/ghi** trên tệp, con trỏ tệp sẽ **tự động dịch chuyển về phía cuối tệp**.
- Khoảng cách dịch chuyển (tính theo byte) sẽ bằng số byte đã được đọc từ tệp hoặc ghi lên tệp.



Quy trình làm việc với tập tin

Read : đọc dữ liệu từ tập tin

Write : ghi dữ liệu vào tập tin

- Bước 1. Khai báo biến tệp: `FILE *Con_Trò_Tệp;`
- Bước 2. Mở tệp để làm việc
 - Phân biệt các loại tệp và các mục đích mở tệp
`Con_Trò_Tệp = fopen(Tên_Tệp, Chế_độ_mở);`
- Bước 3. Truy nhập tệp
 - Truy nhập để đọc/ ghi/thêm mới
 - Phân biệt giữa các loại tệp
 - Đọc dữ liệu từ tệp : `fscanf()` / `fgets()` / `getc()`
 - Ghi dữ liệu ra tệp : `fprintf()` / `fputs()` / `putc()`
 - Dịch chuyển con trỏ tệp : `fseek()` / `rewind()`
 - Kiểm tra kết thúc tệp : `feof()`
- Bước 4. Đóng tệp: `fclose(f);`

Ví dụ 1. File văn bản

```
#include <stdio.h>
```

- Tạo một file văn bản mới có tên “hello.txt”
- Ghi vào tập tin này dòng chữ “Hello World”

```
int main()
```

```
{
```

```
    FILE *out;
```

```
    out = fopen("hello.txt", "w");
```

```
    if (out == NULL)
```

```
    {
```

```
        perror("Khong the mo tep de ghi.\n");
```

```
        return 1;
```

```
    }
```

```
    fprintf(out, "Hello world");
```

```
    fclose(out);
```

```
    return 0;
```

```
}
```


Ví dụ 2. Đếm số từ của một tệp văn bản

```
#include <stdio.h>

int main()
{
    int dem = 0;
    char s[80];
    FILE *f = fopen("vanban.txt", "r");
    if (f == NULL)
    {
        perror("Loi mo tep vanban.txt\n");
        return 1;
    }
    while (!feof(f))
        dem += fscanf(f, "%s", s);
    fclose(f);
    printf("Tong so tu: %d", dem);
    return 0;
}
```

Ví dụ 3a. fgetc() và fputc()

```
FILE *input, *output;  
input = fopen( "tmp.c", "r" );  
output = fopen( "tmpCopy.c", "w+" );  
char ch;  
ch = fgetc( input );  
while( ch != EOF ) {  
    fputc( ch, output );  
    ch = fgetc( input );  
}  
  
fclose(input);  
fclose(output);
```

Ví dụ 3b: Thống kê dữ liệu trong file văn bản

- Đọc dữ liệu từ file văn bản “vanban.txt” bằng cách đọc từng ký tự.
- Thống kê :
 - Tổng số ký tự là chữ cái thường ‘a’ đến ‘z’.
 - Tổng số ký tự là chữ cái hoa ‘A’ đến ‘Z’.
 - Tổng số ký tự số : từ ‘0’ đến ‘9’.
 - Tổng số khoảng trống : ký tự khoảng trống, tab, xuống dòng (isspace(ch)).

```
FILE *fInput, *fOutput;
fInput = fopen("hello.txt", "r");
fOutput = fopen("hello_copy.txt", "w");
char ch;
int demKyTuThuong = 0, demKyTuHoa = 0;
ch = fgetc(fInput);
while (ch != EOF) {
    printf("ch: %c %d\n", ch, ch);
    if ((ch >= 'a') && (ch <= 'z'))
        demKyTuThuong += 1;
    if (isalpha(ch) && isupper(ch))
        demKyTuHoa += 1;
    fputc(ch, fOutput);
    ch = fgetc(fInput);

    if (ch == EOF) {
        printf("Ma EOF: %d\n", ch);
    }
}
fclose(fOutput);
fclose(fInput);
printf("Tong so ky tu chu cai thuong: %d\n", demKyTuThuong);
printf("Tong so ky tu chu cai hoa: %d\n", demKyTuHoa);
```

Ví dụ 4. fgets()

```
#include <stdio.h>
#define LINE_LENGTH 80
int main()
{
    FILE *fp;
    char line[LINE_LENGTH];
    int count = 0;
    fp = fopen("input.txt", "r");
    while ( fgets(line, LINE_LENGTH, fp) != NULL)
        count++;
    printf("File contains %d lines.\n", count);
    fclose(fp);
    return 0;
}
```


Ví dụ 5: thao tác với tệp văn bản

- Nhập thông tin điểm thi toán, lý, hóa từ bàn phím
- Ghi các điểm này vào tệp văn bản có tên “my_score.txt”.
- Đọc từ tệp trên, tính điểm trung bình và in kết quả ra màn hình.

Mã nguồn: ghi dữ liệu vào file văn bản

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
    FILE *f;
```

```
    float dToan, dLy, dHoa;
```

```
    // Nhập điểm từ bàn phím
```

```
    printf("Nhập điểm thi Toán, Lý Hóa: ");
```

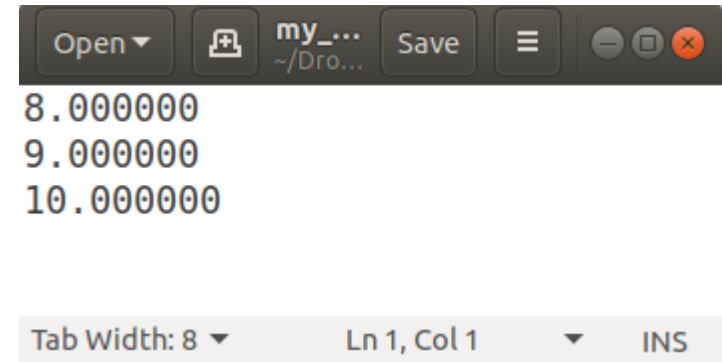
```
    scanf("%f%f%f", &dToan, &dLy, &dHoa);
```

```
    // Lưu vào file văn bản
```

```
    f = fopen("my_score.txt", "w");
```

```
    fprintf(f, "%f\n%f\n%f\n", dToan, dLy, dHoa);
```

```
    fclose(f);
```



Mã nguồn: đọc dữ liệu từ file văn bản

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
    FILE *f;
```

```
    float dToan, dLy, dHoa, dTb;
```

```
    // Doc du lieu tu file van ban
```

```
    f = fopen("my_score.txt", "r");
```

```
    fscanf(f, "%f%f%f", &dToan, &dLy, &dHoa);
```

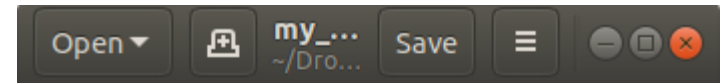
```
    fclose(f);
```

```
    // Tinh diem trung binh
```

```
    dTb = (dToan + dLy + dHoa) / 3;
```

```
    printf("Diem trung binh: %f", dTb);
```

```
}
```



```
8.000000
9.000000
10.000000
```

Tab Width: 8 Ln 1, Col 1 INS

```
Diem trung binh: 9.000000
-----
(program exited with code: 0)
Press return to continue
```

Ví dụ 7a. File nhị phân: ghi mảng số nguyên vào file

```
#include <stdio.h>
```

```
int main() {
```

```
    FILE *fp;
```

```
    int c[3] = {3, 4, 5};
```

```
    int i;
```

```
    fp = fopen("file_nhi_phan_2.dat", "wb");
```

```
    fwrite(c, 3, sizeof(int), fp);
```

```
    fclose(fp);
```

```
}
```

Name:

file_nhi_phan_2.dat

Type:

Binary (application/octet-stream)

Size:

12 bytes

Ví dụ 7b. File nhị phân: đọc mảng số nguyên từ file

```
#include <stdio.h>
```

```
int main() {
```

```
    FILE *fp;
```

```
    int buffer[3];
```

```
    int i;
```

```
    fp = fopen("file_nhi_phan_2.dat", "rb");
```

```
    fread(buffer, 3, sizeof(int), fp);
```

```
    for (i=0; i<3; i++) {
```

```
        printf("\nPhan tu thu %d : %d", i, buffer[i]);
```

```
    }
```

```
    fclose(fp);
```

```
Phan tu thu 0 : 3
```

```
Phan tu thu 1 : 4
```

```
Phan tu thu 2 : 5
```

```
-----
```

```
(program exited with code: 0)
```

```
Press return to continue
```


Ví dụ 7b2: Đọc dữ liệu từ file với khả năng xác định số phần tử trong file

```
FILE *fp;  
int buffer[100];  
int i;  
fp = fopen("fb2.dat", "rb");  
  
fseek(fp, 0, SEEK_END);  
int filesize = ftell(fp);  
printf("Kich thuoc file: %d\n", filesize);  
  
int so_phan_tu = 0;  
so_phan_tu = filesize / sizeof(int);  
printf("So phan tu trong file: %d\n", so_phan_tu);
```

Ví dụ 7b2: Đọc dữ liệu từ file với khả năng xác định số phần tử trong file

```
fseek(fp, 0, SEEK_SET);
```

```
fread(buffer, so_phan_tu, sizeof(int), fp);
```

```
for (i = 0; i < 4; i++) {
```

```
    printf("\n Phan tu thu %d: %d", i, buffer[i]);
```

```
}
```

```
fclose(fp);
```

Ví dụ 7c. Đọc dữ liệu tại một vị trí nhất định trong file nhị phân

- Đọc dữ liệu ở vị trí thứ 2 (index =1) trong file trên.

```
/* Doc du lieu tai mot vi tri nhat dinh trong file  
nhị phân */  
#include <stdio.h>  
int main() {  
    FILE *fp;  
    int n = 0;  
    int index = 1;  
    fp = fopen("fb2.dat", "rb");  
    if (fp == NULL) {  
        perror("Loi doc file\n");  
        return 1;  
    }  
    fseek(fp, index * sizeof(int), SEEK_SET);  
    //fread(&n, sizeof(int), 1, fp);  
    fread(&n, 1, sizeof(int), fp);  
  
    printf("Gia tri vua doc: %d\n", n);  
  
    fclose(fp);  
    return 0;  
}
```

Ví dụ 7d. File nhị phân

Bổ sung

- Bổ sung thêm phần tử thứ 4 với giá trị bằng 6 vào file.

```
#include <stdio.h>
int main() {
    FILE *fp;
    fp = fopen("fb2.dat", "ab");
    if (fp == NULL) {
        perror("Khong mo duoc file\n");
        return 1;
    }

    int n = 6;
    fwrite(&n, 1, sizeof(int), fp);

    fclose(fp);
    return 0;
}
```

Ví dụ 7e. File nhị phân

Chỉnh sửa

- Chỉnh sửa phần tử thứ 2 từ giá trị bằng 4 sang giá trị bằng 10.

```
#include <stdio.h>
int main() {
    FILE *fp;
    int n = 10;
    int index = 1;
    fp = fopen("fb2.dat", "r+b");
    if (fp == NULL) {
        perror("Khong mo duoc file\n");
        return 1;
    }

    fseek(fp, index * sizeof(int), SEEK_SET);
    fwrite(&n, 1, sizeof(int), fp);

    fclose(fp);
    return 0;
}
```


Ví dụ 8

Tạo file Songuyen.dat ghi 100 số lẻ đầu tiên.

```
#include <stdio.h>
int main() {
    FILE * f = fopen("SoNguyen.Dat", "wb");
    int i, n;
    for(i = 0; i <100; i++) {
        n = 2*i+1;
        fwrite(&n, sizeof(int), 1, f);
    }
    fclose(f);
    return 0;
}
```

Ví dụ 9

Đọc file Songuyen.dat, đưa ra màn hình các số lẻ từ vị trí số thứ 50 của file

```
#include <stdio.h>
int main() {
    FILE * f = fopen("SoNguyen.Dat", "rb");
    int n;
    fseek(f, 50*sizeof(int), SEEK_SET);
    while(!feof(f)){
        fread(&n, sizeof(int), 1, f);
        printf("%4d", n);
    }
    fclose(f);
    return 0;
}
```

Ví dụ 10: File và bản ghi

- Nhập danh sách từ bàn phím các thí sinh dự thi, mỗi thí sinh gồm họ tên, số báo danh, khoa dự thi và điểm thi.
- Dữ liệu nhập được ghi vào file ThiSinh.dat. Kết thúc nhập khi gặp một thí sinh có tên là « *** »
- Đọc từ file ThiSinh.Dat, đưa ra màn hình danh sách các thí sinh thi vào ngành CNTT có điểm thi lớn hơn 21 theo quy cách

STT	Số Báo Danh	Họ Tên	Điểm Thi
-----	-------------	--------	----------

- Từ file ThiSinh.Dat, tạo file CNTT.Dat chỉ chứa danh sách các thí sinh thi vào khoa CNTT
- Nhập vào một số báo danh, tìm trong file ThiSinh.Dat và in ra họ tên, điểm thi và khoa đăng ký của thí sinh nếu tìm thấy. Nếu không tìm thấy thí sinh thì đưa ra thông báo « không tìm thấy »

Ví dụ 10

```
#include <stdio.h>
#include <string.h>
typedef struct {
    char Ten[30];
    long SBD;
    char Khoa[10];
    float Diem;
} SinhVien;

int main() {
    FILE *f1,*f2;
    SinhVien SV;
    int i, SBD;
    //Nhap thong tin cho file ThiSinh.Dat
    //Tao file CNTT.Dat
    return 0;
}
```

Ví dụ 10

```
//Nhập thông tin cho file ThiSinh.Dat
f1 = fopen("ThiSinh.Dat", "wb");
i = 1;
do {
    printf("Thi sinh %d :\n", i);
    printf("  Ho Ten : "); fflush(stdin); fgets(SV.Ten, sizeof(SV.Ten), stdin);
    SV.Ten[strlen(SV.Ten) - 1] = '\0'; // Bỏ ký tự xuống dòng
    if(strcmp(SV.Ten, "***") == 0) break;
    printf("  So Bao Danh: "); scanf("%d", &SV.SBD);
    printf("  Khoa : "); fflush(stdin); fgets(SV.Khoa, sizeof(SV.Khoa), stdin);
    SV.Khoa[strlen(SV.Khoa) - 1] = '\0'; // Bỏ ký tự xuống dòng
    printf("  Diem : "); scanf("%f", &SV.Diem);
    fwrite(&SV, sizeof(SinhVien), 1, f1);
    i++;
} while(1);
fclose(f1);
```

Ví dụ 10

```
printf("\n\n DANH SACH BAN DAU \n");
f1 = fopen("ThiSinh.Dat", "rb");
i = 0;
while (fread(&SV, sizeof(SinhVien), 1, f1) > 0)
    printf("%-3d %-5d %-20s %-20s %-5.1f\n", ++i, SV.SBD,
           SV.Ten, SV.Khoa, SV.Diem);

printf("\n\n Thi Sinh thi CNTT tren 21.0\n");
i = 0;
rewind(f1);

while (fread(&SV, sizeof(SinhVien), 1, f1) > 0)
    if (strcmp(SV.Khoa, "CNTT") == 0 && SV.Diem > 21.0)
        printf("%-3d %-5d %-20s %-5.1f\n", ++i, SV.SBD,
               SV.Ten, SV.Diem);
```

Ví dụ 10

```
printf("\n\n Tao file CNTT.Dat\n");
i = 0;
rewind(f1);
f2 = fopen("CNTT.Dat", "wb");
while (fread(&SV, sizeof(SinhVien), 1, f1) > 0)
    if (strcmp(SV.Khoa, "CNTT") == 0 )
        fwrite(&SV, sizeof(SinhVien), 1, f2);
fclose(f2);

f2 = fopen("CNTT.Dat", "rb");    //doc lai file
while (fread(&SV, sizeof(SinhVien), 1, f2) > 0)
    printf("%-3d %-5d %-20s  %-5.1f\n", ++i, SV.SBD, SV.Ten,
        SV.Diem);
fclose(f2);
```

Ví dụ 10

```
printf("\n\nTim Sinh Vien\n");
printf(" So Bao Danh "); scanf("%d", &SBD);
rewind(f1);
while (fread(&SV, sizeof(SinhVien), 1, f1))
    if (SV.SBD == SBD){
        printf("Tim thay sinh vien %s", SV.Ten);
        break;
    }

if (feof(f1)) printf("Khong thay");
fclose(f1);
```


Bài tập 1. Lưu trữ dữ liệu theo cả 2 dạng: file văn bản và file nhị phân

- Nhập từ bàn phím danh sách hàng hóa, mỗi hàng hóa gồm tên hàng, số lượng, đơn giá.
- Dữ liệu nhập được ghi vào file HangHoa.dat. Kết thúc nhập khi gặp một hàng hóa có tên là « *** »
- Đọc từ file HangHoa.Dat, đưa ra màn hình danh sách các hàng hóa theo quy cách
STT Tên Hàng Số Lượng Đơn Giá Thành Tiền
Tổng kết Tổng số hàng Tổng số tiền
- Nhập vào một số lượng, tìm trong file HangHoa.Dat và in ra thông tin của những hàng hóa có số lượng như vậy. Nếu không tìm thấy thì đưa ra thông báo « không tìm thấy »

Bài tập 2

- Viết chương trình tạo ra tệp văn bản F3 từ việc ghép nội dung hai tệp văn bản F1 và F2.

Bài tập 3

- Viết một chương trình cho phép cắt hết chú thích của một chương trình C được lưu trữ trong một tệp. Tên tệp chương trình được nhập vào từ bàn phím. Giả thiết rằng chương trình không có lỗi cú pháp.

Bài tập 4

Giả thiết một tệp dữ liệu thu thập về thời tiết trong một năm có định dạng theo mỗi dòng là:

<ngày>/<tháng> <nhiệt độ thấp nhất>-<nhiệt độ cao nhất>
<độ ẩm>

1/1 11-17 70

2/1 12-17 75

...

Hãy viết chương trình đọc dữ liệu của tệp này và in ra nhiệt độ trung bình của các tháng trong năm, tháng khô hanh nhất và tháng ẩm ướt nhất.



25 YEARS ANNIVERSARY
SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you
for your
attentions!**



soict.hust.edu.vn/



fb.com/groups/soict

