



CƠ SỞ LẬP TRÌNH

KIỂU TẬP TIN

- ☐ Khái niệm
- ☐ Các thao tác trên tập tin
- ☐ Tập tin văn bản
- ☐ Tập tin nhị phân
- ☐ Các hàm xử lý tập tin
- ☐ Truyền tham số là tập tin cho hàm

1. Khái niệm kiểu tập tin

□ Tập tin

- Là tập hợp các dữ liệu có liên quan với nhau và có cùng kiểu được nhóm lại với nhau tạo thành một dãy.
- Tập được chứa trong thiết bị nhớ ngoài
- Kích thước và số lượng gần như không hạn chế.

□ Phân loại tập tin

■ Theo cách truy cập

- Tập tin truy cập tuần tự: việc đọc một phần tử bất kỳ của tập tin bắt buộc phải tuần tự đi qua các phần tử trước đó
- Tập tin truy cập ngẫu nhiên: có thể truy xuất phần tử bất kỳ của tập tin thông qua chỉ số thứ tự phần tử trong tập tin

Khái niệm kiểu tập tin (tt)

□ Phân loại tập tin

■ Theo bản chất dữ liệu

- Tập tin văn bản: chứa các kí tự trong bảng mã ASCII không kể đến các kí tự điều khiển. Dữ liệu được lưu thành các dòng, mỗi dòng được kết thúc bằng ký tự xuống dòng là CR (Carriage Return – về đầu dòng, mã 10) và LF (Line Feed – xuống dòng, mã 13). tập tin văn bản kết thúc bằng kí tự EOF (End Of File) có mã 26 (Ctrl + Z)

Ví dụ:

Các tập tin văn bản (text) (*.txt)

- Tập tin nhị phân: các phần tử là các số nhị phân, và chứa khá nhiều dữ liệu có mã là các kí tự điều khiển.

Ví dụ:

Các tập tin chương trình (*.exe, *.com...)

Một số khái niệm

□ Biến tập tin

- Là biến thuộc kiểu tập tin dùng đại diện cho một tập tin.
- Dữ liệu chứa trong tập tin được truy xuất thông qua các thao tác với biến tập tin.

□ Con trỏ tập tin

- Dùng để xác định vị trí của phần tử hiện tại để đọc hoặc ghi dữ liệu trên tập tin.
 - Khi tập tin được mở để đọc hoặc ghi thì con trỏ tập tin luôn ở vị trí đầu tập tin.
 - Mỗi khi đọc hoặc ghi trên tập tin thì con trỏ tập tin tự động tăng lên một khoảng theo đúng số byte vừa đọc hoặc ghi trên tập tin.
-

2. Các thao tác trên tập tin

- ❑ Các bước cơ bản để xử lý tập tin
 1. Khai báo biến tập tin
 2. Mở tập tin để ghi hoặc đọc
 3. Xử lý dữ liệu trong tập tin
 4. Đóng tập tin
- ❑ Các hàm thao tác với tập tin trong thư viện `stdio.h`

2.1 Khai báo biến tập tin

□ Cú pháp

FILE *<tên biến TẬP>;

- Trong đó, FILE là từ khoá luôn phải có và viết dạng chữ hoa
- Các biến tập tin là các biến con trỏ

□ Ví dụ:

FILE *f1, *f2;

// Khai báo 2 biến tập tin f1 và f2

2.2 Mở tập tin

□ Cú pháp

<Biến TẬP>=fopen(<Tên TẬP>,<Kiểu xử lý TẬP>);

■ Trong đó:

- Tên tập tin: đường dẫn đến tập tin trên đĩa (lưu ý, dấu \ được ghi là \\). Tên tập tin được đặt theo quy tắc đặt tên.
- Kiểu xử lý tập tin: xác định cách thức mà tập tin được mở

■ fopen trả về một con trỏ tập tin, nếu có lỗi con trỏ trả về NULL

□ Ví dụ:

f1=fopen("C:\\C-Samples\\VIDU.TXT","w");

Mở tập tin VIDU.TXT mới để ghi

f2=fopen("C:\\C-Samples\\VIDU.TXT","r");

Mở tập tin VIDU.TXT mới để đọc

Các chế độ xử lý tập tin

Chế độ	Ý nghĩa
r	Mở tập tin văn bản để đọc
w	Mở tập tin văn bản để ghi, ghi đè lên tập tin đã có
a	Mở tập tin văn bản và ghi nối vào cuối TẬP, chưa có tạo mới
r+	Mở tập tin văn bản để đọc/ghi
w+	Mở tập tin văn bản để ghi/đọc, ghi đè lên tập tin đã có
a+	Mở tập tin văn bản hoặc tạo mới để đọc và ghi nối vào cuối
rb	Mở tập tin nhị phân để đọc
wb	Mở tập tin nhị phân để ghi, ghi đè lên tập tin đã có
ab	Ghi nối vào tập tin nhị phân
r+b	Mở ra tập tin nhị phân để đọc/ghi
w+b	Tạo ra tập tin nhị phân để đọc/ghi
a+b	Nối vào hay tạo mới tập tin nhị phân

Ví dụ mở tập tin

❑ Mở tập tin VIDU.TXT để ghi

```
FILE *f;  
f = fopen("VIDU.txt", "w");  
if (f!=NULL)  
{  
    /* Các câu lệnh để thao tác với TẬP*/  
    /* Đóng tập tin */  
}  
else printf("Loi - Khong mo duoc tep!");
```

❑ Chú ý:

- Khi mở tập tin để ghi mà tập tin đã tồn tại rồi thì tập tin đó sẽ bị xóa và được thay bằng tập tin khác.
- Khi mở tập tin để đọc thì tập tin đó phải tồn tại, nếu không sẽ có lỗi.

2.3 Các thao tác khác

❑ Hàm đóng tập tin

fclose(<Tên biến TẬP>) ;

- Hàm trả về 0 nếu đóng tập tin thành công, trả về EOF nếu có lỗi

❑ Hàm kiểm tra kết thúc tập tin hay chưa?

fEOF(<Tên biến TẬP>) ;

- Hàm trả về EOF nếu đã hết tập tin, ngược lại trả về 0

❑ Hàm di chuyển con trỏ tập tin về đầu

rewind(<Tên biến TẬP>) ;

3. Tập tin văn bản

□ Ghi dữ liệu lên tập tin văn bản

- **putc**(*ch*, *f*) ; ghi ký tự *ch* vào tập tin văn bản *f*, trả về **EOF** nếu gặp lỗi.
- **fputs**(*str*, *f*) ; ghi chuỗi *str* vào tập tin văn bản *f*, trả về 0 nếu *str* rỗng, trả về **EOF** nếu gặp lỗi
- **fprintf**(*f*, chuỗi định dạng, danh sách biểu thức) ; ghi vào tập tin văn bản *f* các biểu thức với các định dạng được chỉ ra, tương tự hàm **printf()**

Ghi dữ liệu lên tập tin văn bản

- Ví dụ 1: Ghi một dòng các chữ hoa vào tập tin

```
#include <stdio.h>
#include <ctype.h>
main()
{
    FILE *f;    char c;
    f=fopen("C11T001.txt", "w");
    do putc(toupper(c=getchar()), f);
    while (c!='\n');
    fclose(f);
}
```

- Ví dụ 2: Ghi chuỗi ký tự vào tập tin

```
main()
{
    FILE *f;
    f=fopen("C11T002.txt", "w");
    fputs("Ngon ngu lap trinh C", f);
    fclose(f);
}
```

Đọc dữ liệu từ tập tin văn bản

□ Các hàm đọc dữ liệu từ tập tin văn bản

- **getc**(*f*) ; **fgetc**(*f*) ; đọc một ký tự từ tập tin văn bản *f*. Hàm trả về mã ASCII của ký tự nào đó (kể cả EOF) trong tập tin *f*
- **fgets**(*str*, *n*, *f*) ; đọc một chuỗi *str* từ tập tin văn bản *f* cho đến khi gặp ký tự xuống dòng '\n' hoặc ký tự EOF hay đủ *n* ký tự.
- **fscanf**(*f*, chuỗi định dạng, danh sách các biến) ; đọc từ tập tin văn bản *f* các biến theo định dạng, tương tự hàm **scanf**()

Đọc dữ liệu từ tập tin văn bản (tt)

□ Ví dụ 1: Sao chép TẬP

- Sao chép nội dung tập tin sample.txt sang tập tin sp.txt

```
main()
{
    FILE *f1,*f2; int ch;
    f1=fopen("sample.txt","r");
    f2=fopen("sp.txt","w");
    if (f1!=NULL&&f2!=NULL)
    {
        ch=fgetc(f1);
        while (!feof(f1))
        {
            fputc(ch,f2); //ghi vào f2
            ch=fgetc(f1); //đọc từ f1
        }
        fclose(f1);
        fclose(f2);
    }
}
```

Đọc dữ liệu từ tập tin văn bản (tt)

□ Ví dụ 2:

Đọc từ tập tin songuyen.txt một dãy các số nguyên dương, ghi vào tập tin ketqua.txt các số nguyên tố có trong dãy đó, cuối cùng ghi ra tổng của các số nguyên tố đó.

```
int main()
{FILE *f1,*f2; int n; long tong=0;
  f1=fopen("songuyen.txt","r");
  f2=fopen("ketqua.txt","w");
  if (f1!=NULL && f2!=NULL)
  { while (!feof(f1))
    { fscanf(f1,"%d",&n);
      if (ngto(n)) { tong+=n; fprintf(f2,"%d\t",n); }
    } fprintf(f2,"\nTong la: %ld",tong);
    fclose(f1); fclose(f2);
  } }
```


4. Tập tin nhị phân

□ Ghi dữ liệu lên tập tin nhị phân

- Hàm **fwrite**(địa chỉ của khối dữ liệu, kích thước mỗi phần tử, số phần tử, **f**) ;
- Ghi vào trong tập tin **f** khối dữ liệu có địa chỉ, số lượng và kích thước của mỗi phần tử.
- Giá trị trả về là số phần tử đã được ghi vào tập tin

■ Ví dụ:

```
FILE *f;  
int i;  
f=fopen("C:\\\\SN100.txt", "wb") ;  
for (i=1; i<=100; i++)  
    fwrite(&i, sizeof(int), 1, f) ;  
fclose(f) ;
```

Tập tin nhị phân (tt)

□ Đọc dữ liệu từ tập tin nhị phân

- Hàm **fread** (địa chỉ của vùng nhớ nhận dữ liệu, kích thước mỗi phần tử, số phần tử, f) ;
- Đọc từ tập tin **f** số lượng phần tử có kích thước của mỗi phần tử được chỉ ra và lưu vào vùng nhớ nhận dữ liệu
- Giá trị trả về là số phần tử đã được đọc từ tập tin
- Ví dụ:

```
g=fopen("C:\\\\SN100.txt","rb");  
do { fread(&i,sizeof(int),1,g);  
      if (!feof(g)) printf("%d",i); }  
while (!foef(g));
```

Di chuyển con trỏ tập tin

□ Hàm fseek

fseek (f, No*Kích thước, vị trí)

Trong đó:

- f: con trỏ tập tin
- No: số thứ tự phần tử trong tập tin (phần tử đầu tiên đánh số là 0)
- Vị trí có thể là:
 - SEEK_SET hoặc 0: di chuyển từ đầu tập tin
 - SEEK_CUR hoặc 1: di chuyển từ vị trí hiện tại
 - SEEK_END hoặc 2: di chuyển từ cuối tập tin
- Hàm trả về 0 nếu di chuyển thành công, trả về khác 0 nếu ngược lại.

Di chuyển con trỏ tập tin (tt)

- Ví dụ: Truy cập trực tiếp để cập nhật dữ liệu

```
main()
{
FILE *f;
int no,number;
f=fopen("C:\\\\SN100.txt","r+b");
do
{
    printf("Vi tri can cap nhat: ");
    scanf("%d",&no);
    printf("Gia tri can cap nhat: ");
    scanf("%d",&number);
    if (no>0)
    {fseek(f,sizeof(int)*(no-1),SEEK_SET);
    fwrite(&number,sizeof(int),1,f);}
} while (no!=0);
fclose(f);
```

Ví dụ - Quản lý sinh viên

- Viết chương trình quản lý sinh viên
 - Mỗi sinh viên cần quản lý ít nhất 2 thông tin: mã sinh viên và họ tên.
 - Viết chương trình cho phép lựa chọn các chức năng:
 - Nhập danh sách sinh viên từ bàn phím rồi ghi lên tập tin SinhVien.dat
 - Đọc dữ liệu từ tập tin SinhVien.dat rồi hiển thị danh sách lên màn hình
 - Tìm kiếm họ tên của một sinh viên nào đó dựa vào mã sinh viên nhập từ bàn phím.

Ví dụ - Quản lý sinh viên (tt)

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
typedef struct
{
    char Ma[10];
    char HoTen[40];
} SinhVien;
void WriteFile(char *FileName)
{
    FILE *f;
    int n,i;
    SinhVien sv;
    f=fopen(FileName,"wb");
    printf("Nhap bao nhieu sinh vien?");
    scanf("%d",&n);
    fflush(stdin);
```

Ví dụ - Quản lý sinh viên (tt)

```
for (i=1;i<=n;i++)
{
    printf("Sinh vien thu %i\n",i);
    printf(" - MSSV: ");gets(sv.Ma);
    printf(" - Ho ten: ");gets(sv.HoTen);
    fwrite(&sv,sizeof(sv),1,f);
    fflush(stdin);
}
fclose(f);
printf("Bam phim bat ky de tiep tục");
getch();
}
void ReadFile(char *FileName)
{
    FILE *f;
    SinhVien sv;
    f=fopen(FileName,"rb");
    printf(" MSSV | Ho va ten\n");
    printf("-----\n");
```

Ví dụ - Quản lý sinh viên (tt)

```
fread(&sv,sizeof(sv),1,f);
    while (!feof(f))
    {
        printf(" %s    | %s\n",sv.Ma,sv.HoTen);
        fread(&sv,sizeof(sv),1,f);
    }
    fclose(f);
    printf("Bam phim bat ky de tiep tuc!!!");
    getch();
}
void Search(char *FileName)
{
    char MSSV[10];
    FILE *f;
    int Found=0;
    SinhVien sv;
    fflush(stdin);
    printf("Ma so sinh vien can tim: ");gets(MSSV);
    f=fopen(FileName,"rb");
```


Ví dụ - Quản lý sinh viên (tt)

```
while (!feof(f) && Found==0)
{
    fread(&sv,sizeof(sv),1,f);
    if (strcmp(sv.Ma,MSSV)==0) Found=1;
}
fclose(f);
if (Found == 1)
    printf("Tim thay SV co ma %s. Ho ten la:
%s",sv.Ma,sv.HoTen);
else
    printf("Tim khong thay sinh vien co ma
%s",MSSV);
printf("\nBam phim bat ky de tiep tuc!!!");
getch();
}
```

Ví dụ - Quản lý sinh viên (tt)

```
int main()
{
    int c;
    for (;;)
    {
        printf("\n1. Nhap DSSV\n");
        printf("2. In DSSV\n");
        printf("3. Tim kiem\n");
        printf("4. Thoat\n");
        printf("Ban chon 1, 2, 3, 4: ");
        scanf("%d",&c);
        if(c==1)
            WriteFile("d:\\SinhVien.Dat");
        else if (c==2)
            ReadFile("d:\\SinhVien.Dat");
        else if (c==3)
            Search("d:\\SinhVien.Dat");
        else break;
    }
    return 0;
}
```

5. Các hàm xử lý tập tin

- ❑ **int ferror(FILE *fp)**
 - Trả về 0 nếu không có lỗi, khác 0 nếu gặp lỗi
- ❑ **int remove(char *filename)**
 - Xóa tập tin có tên là filename
 - Trả về 0 nếu thành công
- ❑ **int rename(char *old, char *new)**
 - Đổi tên cũ `old` thành tên mới `new`
 - Trả về 0 nếu đổi thành công
- ❑ **int fflush(FILE *fp)**
 - Đổ bộ nhớ đệm
- ❑ **long ftell(FILE *fp)**
 - Cho kích thước của tập tin fp

6. Truyền tham số là tập tin cho hàm

□ Ví dụ 1. Sao chép 2 tập tin

```
#include <stdio.h>
#include <stdlib.h>
main(int argc, char *argv[])
{
    FILE *in,*out; char ch;
    if (argc!=3)
        {printf("Chua co ten tep.\n"); exit(1);}
    if ((in=fopen(argv[1],"rb"))==NULL)
        {printf("Khong the mo tep nguon!\n");exit(1);}
    if ((out=fopen(argv[2],"wb"))==NULL)
        {printf("Khong the mo tep dich!\n");exit(1);}
    while (!feof(in))
    { ch=getc(in);
      if (!feof(in)) putc(ch,out);
    }
    printf("Da sao chep xong!")
    fclose(in); fclose(out);
}
```

Truyền tham số là tập tin cho hàm (tt)

□ Ví dụ 2: Xóa tập tin

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
main(int argc, char *argv[])
{ char ch;
  if (argc!=2)
  {printf("Chua co ten tep.\n"); exit(1);}
  printf("Co xoa tep %s khong? (C/K)?",argv[1]);
  ch=getchar();
  if (toupper(ch)=='C')
    if (remove(argv[1]))
      {printf("Khong the xoa!"); exit(1);}
}
```

Truyền tham số là tập tin cho hàm (tt)

Cách truyền tham số khi thực hiện chương trình

□ Ví dụ 1:

- Dịch ra tập tin mycopy.exe
- mycopy <tên tập tin nguồn> <tên tập tin đích>
Chẳng hạn: mycopy sample.txt vd.txt

□ Ví dụ 2:

- Dịch ra tập tin mydel.exe
- mydel <tên tập tin cần xóa>
Chẳng hạn: mydel sample.txt

Bài tập thực hành

1. Viết chương trình đếm số chữ của từng loại chữ trong bảng chữ cái được chứa trong một tập tin văn bản (tập tin có bao nhiêu chữ 'A', chữ 'B'...).
2. Viết chương trình đếm trong tập tin văn bản
 - Có bao nhiêu dòng?
 - Có bao nhiêu kí tự?
 - Dòng dài nhất là dòng nào?
 - Từ dài nhất trong văn bản là từ nào (từ là một nhóm các kí tự liền nhau).

Bài tập thực hành

3. Viết chương trình nhập vào từ tập tin văn bản songuyen.txt một dãy các số nguyên, hãy tạo ra các TẬP:

- prime.txt chứa các số nguyên tố có trong dãy
- cp.txt chứa các số chính phương
- dep.txt chứa các số đẹp (vd: 123, 134...)
- hoanhao.txt chứa các số hoàn hảo

4. Cho 2 tập tin So1.txt và So2.txt chứa các số nguyên.

- Sắp xếp theo chiều tăng dần các số trong 2 TẬP
 - Tạo tập tin So3.txt gồm các số trong cả 2 tập tin trên và cũng được sắp theo thứ tự tăng dần.
-