

TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT
KHOA CÔNG NGHỆ THÔNG TIN



KỸ THUẬT LẬP TRÌNH

CHƯƠNG 5
TẬP TIN

NỘI DUNG

- 1 • Khái niệm kiểu tập tin
- 2 • Các thao tác trên tập tin
- 3 • Tập tin văn bản
- 4 • Tập tin nhị phân
- 5 • Các hàm xử lý tập tin
- 6 • Bài tập

Kết quả học tập

Tạo tập tin và đọc, ghi dữ liệu vào tập tin



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 tin được chứa trong thiết bị nhớ ngoài
- Kích thước và số lượng gần như không hạn chế.



1. Khái niệm kiểu tập tin

❖ Phân loại tập tin

➤ Theo cách truy cập

- ✓ Tập tin truy cập tuần tự: việc đọc một phần tử bất kỳ của tệp 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 tệp thông qua chỉ số thứ tự phần tử trong tệp tin.



1. 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)

- ✓ 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...)



1. Khái niệm kiểu tập tin (tt)

❖ 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 tin>;

- 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 tin>=**fopen**(<Tên tập tin>,<Kiểu xử lý tập tin>);

➤ 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:\\TC\\VIDU.TXT", "**w**") ;

Mở tập tin VIDU.TXT mới để ghi

f2=fopen("C:\\TC\\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 tin, 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 tin*/  
    /* 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 tin>);

- 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 tin>);

- 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 tin>);



3. Tập tin văn bản

❖ Ghi dữ liệu lên tập tin văn bản

- **putc**(**ch**, **f**) ; ghi kí tự **ch** vào tập tin văn bản **f**, trả về **EOF** nếu gặp lỗi.
- **fputs**(**str**, **f**) ; ghi xâu **str** vào tập tin văn bản **f**, trả về 0 nếu **str** rỗng và 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**()



3. Ghi dữ liệu lên tập tin văn bản (tt)

❖ 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("C:\\sample.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("C:\\vidu.txt","w");
  fputs("Ngon ngu lap trinh C",f);
  fclose(f);
}
```



3. Ghi dữ liệu lên tập tin văn bản (tt)

❖ Các hàm đọc dữ liệu từ tập tin văn bản

- **getc**(**f**) ; **fgetc**(**f**) ; đọc một kí tự từ tập tin văn bản **f**. Hàm trả về mã ASCII của kí tự nào đó (kể cả **EOF**) trong tập tin **f**
- **fgets**(**str**, **n**, **f**) ; đọc một xâu **str** từ tập tin văn bản **f** cho đến khi gặp kí tự xuống dòng **'\n'** hoặc kí tự EOF hay đủ **n** kí 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**()



3. Ghi dữ liệu lên tập tin văn bản (tt)

❖ Ví dụ 1: Sao chép tập tin

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

```
main()
{ FILE *f1,*f2; int ch;
  f1=fopen("C:\\sample.txt","r");
  f2=fopen("C:\\sp.txt","w");
  if (f1!=NULL&&f2!=NULL)
  { ch=fgetc(f1);
    while (!feof(f1))
    { fputc(ch,f2); //ghi vao f2
      ch=fgetc(f1); //doc tu f1
    }
    fclose(f1);
    fclose(f2);
  }
}
```



3. Ghi dữ liệu lên 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("C:\\songuyen.txt","r");
 f2=fopen("C:\\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);
```



4. 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;
//Hàm nhập danh sách sinh viên lưu vào file
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)

//Hàm nhập danh sách sinh viên lưu vào file (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();
}
```



Ví dụ - Quản lý sinh viên (tt)

//Hàm in danh sách sinh viên từ file

```
void ReadFile(char *FileName)
{
    FILE *f;
    SinhVien sv;
    f=fopen(FileName,"rb");
    printf(" MSSV | Ho va ten\n");
    printf("-----\n");
    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();
}
```



Ví dụ - Quản lý sinh viên (tt)

//Hàm tìm kiếm sinh viên trong file

```
void Search(char *FileName)
{
    char MSSV[10];
    FILE *f;
    int Found=0;
    SinhVien sv;
    fflush(stdin); //Xóa bộ nhớ đệm
    printf("Ma so sinh vien can tim: ");gets(MSSV);
    f=fopen(FileName,"rb");
    while (!feof(f) && Found==0)
    {
        fread(&sv,sizeof(sv),1,f);
        if (strcmp(sv.Ma,MSSV)==0) Found=1;
    }
}
```



Ví dụ - Quản lý sinh viên (tt)

//Hàm tìm kiếm sinh viên trong file (tt)

```
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()    //Hàm 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)**
 - Xoá 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



Bài tập

1. Viết chương trình đọc từ tập tin văn bản Songuyen.inp gồm hai dòng:

- Dòng một chứa một số nguyên n .
- Dòng hai chứa n số nguyên mỗi số cách nhau 1 ký tự trắng.

Hãy tạo các file sau:

- Nguyento.txt chứa các số nguyên tố có trong file Songuyen.inp.
- Chinhphuong.txt chứa các số chính phương có trong file Songuyen.inp.
- Hoanhao.txt chứa các số hoàn hảo có trong file Songuyen.inp.



Bài tập (tt)

2. Cho tập tin Songuyen1.txt gồm 2 dòng. Dòng thứ nhất chứa số nguyên n , dòng thứ hai chứa n số nguyên (mỗi số cách nhau một khoảng trắng). Hãy:

- Đọc các số trong tập tin Songuyen1.txt vào mảng a .
- Sắp xếp các phần tử trong a tăng dần.
- Lưu mảng a ngược vào tập tin Songuyen1.txt (chỉ lưu các số trong a , không lưu số n).

3. Cho tập tin Songuyen2.txt chứa các số nguyên (mỗi số cách nhau một khoảng trắng). Hãy thực hiện các yêu cầu tương tự câu 2.

4. Giả sử đã có hai tập tin Songuyen1.txt và Songuyen2.txt. Các số trong hai tập tin đã được sắp xếp tăng. Hãy tạo tập tin Songuyen3.txt gồm các số trong cả 2 tập tin trên và cũng được sắp theo thứ tự tăng dần.



Bài tập (tt)

5. Viết chương trình đọc danh sách họ tên từ tập tin Hoten.inp sau đó sắp xếp theo thứ tự alphabet. Kết quả được ghi vào tập tin Hoten.out.

Nội dung tập tin Hoten.inp gồm $n+1$ dòng, trong đó:

- Dòng đầu ghi số phần tử của danh sách n ($n \leq 99$),
- n dòng tiếp theo mỗi dòng ghi một chuỗi họ tên.

❖ Ví dụ:

Với tập tin Hoten.inp

```
5
Le Thi Binh
Tran Van An
Tran Ro
Phan Anh
Truong Dinh Bon
```

Tập tin Hoten.out

```
Danh sach sau khi da sap xep alphabet la:
1. Tran Van An
2. Phan Anh
3. Le Thi Binh
4. Truong Dinh Bon
5. Tran Ro
```



