



Chương 9

KIỂU DỮ LIỆU TẬP (File)

Khoa Hệ thống thông tin quản lý

Hà Nội – 2015

Nội dung

- 1 Khái niệm kiểu tập
- 2 Các thao tác trên tập
- 3 Tập văn bản
- 4 Tập nhị phân
- 5 Các hàm xử lý tập
- 6 Truyền tham số là tập cho hàm

1. Khái niệm kiểu tệp

□ Tệp

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

□ Phân loại tệp

■ Theo cách truy cập

- Tệp 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
- Tệp truy cập ngẫu nhiên: có thể truy xuất phần tử bất kỳ của tệp thông qua chỉ số thứ tự phần tử trong tệp

Khái niệm kiểu tệp (tt)

□ Phân loại tệp

■ Theo bản chất dữ liệu

- Tệp 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). Tệp văn bản kết thúc bằng kí tự EOF (End Of File) có mã 26 (Ctrl + Z)

Ví dụ:

Các tệp văn bản (text) (*.txt)

- Tệp 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 tệp chương trình (*.exe, *.com...)

Một số khái niệm

□ Biến tệp

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

□ Con trỏ tệp

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

2. Các thao tác trên tệp

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

2.1 Khai báo biến tệp

□ Cú pháp

FILE *<tên biến tệp>;

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

□ Ví dụ:

FILE *f1, *f2;

// Khai báo 2 biến tệp f1 và f2

2.2 Mở tệp

□ Cú pháp

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

■ Trong đó:

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

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

□ Ví dụ:

```
f1=fopen("C:\\TC\\VIDU.TXT","w");
```

Mở tệp VIDU.TXT mới để ghi

```
f2=fopen("C:\\TC\\VIDU.TXT","r");
```

Mở tệp VIDU.TXT mới để đọc

Các chế độ xử lý tệp

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

Ví dụ mở tệp

❑ Mở tệp VIDU.TXT để ghi

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

❑ Chú ý:

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

2.3 Các thao tác khác

❑ Hàm đóng tệp

fclose(<Tên biến tệp>) ;

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

❑ Hàm kiểm tra kết thúc tệp hay chưa?

fEOF(<Tên biến tệp>) ;

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

❑ Hàm di chuyển con trỏ tệp về đầu

rewind(<Tên biến tệp>) ;

3. Tập văn bản

□ Ghi dữ liệu lên tệp văn bản

- **putc**(*ch*, *f*) ; ghi kí tự *ch* vào tệp văn bản *f*, trả về EOF nếu gặp lỗi.
- **fputs**(*str*, *f*) ; ghi chuỗi *str* vào tệp 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 tệp văn bản *f* các biểu thức với các định dạng được chỉ ra, tương tự hàm **printf**()

Ghi dữ liệu lên tệp văn bản

- ❑ Ví dụ 1: Ghi một dòng các chữ hoa vào tệp

```
#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 tệp

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

Đọc dữ liệu từ tệp văn bản

□ Các hàm đọc dữ liệu từ tệp văn bản

- **getc**(*f*) ; **fgetc**(*f*) ; đọc một kí tự từ tệp văn bản *f*. Hàm trả về mã ASCII của kí tự nào đó (kể cả EOF) trong tệp *f*
- **fgets**(*str*, *n*, *f*) ; đọc một xâu *str* từ tệp 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 tệp văn bản *f* các biến theo định dạng, tương tự hàm **scanf**()

Đọc dữ liệu từ tệp văn bản (tt)

□ Ví dụ 1: Sao chép tệp

■ Sao chép nội dung tệp sample.txt sang tệp 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 vào f2
            ch=fgetc(f1); //doc từ f1
        }
        fclose(f1);
        fclose(f2);
    }
}
```

Đọc dữ liệu từ tệp văn bản (tt)

□ Ví dụ 2:

Đọc từ tệp songuyen.txt một dãy các số nguyên dương, ghi vào tệp 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 nhị phân

□ Ghi dữ liệu lên tập nhị phân

- Hàm **fwrite** (địa chỉ của khối dữ liệu, kích thước mỗi phần tử, số phần tử, **f**) ;
- Ghi vào trong tập **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

■ 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);
```

Tệp nhị phân (tt)

❑ Đọc dữ liệu từ tệp nhị phân

- Hàm **fread** (địa chỉ của vùng nhớ nhận dữ liệu, kích thước mỗi phần tử, số phần tử, f) ;
- Đọc từ tệp **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 tệp
- 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 tệp

□ Hàm fseek

fseek (f, No*Kích thước, vị trí)

Trong đó:

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

Di chuyển con trỏ tệp (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 tệp

- ❑ `int ferror(FILE *fp)`
 - Trả về 0 nếu không có lỗi, khác 0 nếu gặp lỗi
- ❑ `int remove(char *filename)`
 - Xóa tệp 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 tệp fp

6. Truyền tham số là tệp cho hàm

□ Ví dụ 1. Sao chép 2 tệp

```
#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 tệp cho hàm (tt)

□ Ví dụ 2: Xóa tệp

```
#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 tệp cho hàm (tt)

Cách truyền tham số khi thực hiện chương trình

□ Ví dụ 1:

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

□ Ví dụ 2:

- Dịch ra tệp mydel.exe
- mydel <tên tệp 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 tệp văn bản (tệp có bao nhiêu chữ 'A', chữ 'B'...).
2. Viết chương trình đếm trong tệp 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 tệp văn bản songuyen.txt một dãy các số nguyên, hãy tạo ra các tệp:
 - 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 tệp So1.txt và So2.txt chứa các số nguyên.
 - Sắp xếp theo chiều tăng dần các số trong 2 tệp
 - Tạo tệp So3.txt gồm các số trong cả 2 tệp trên và cũng được sắp theo thứ tự tăng dần.