Chương 10

KIỂU TẬP TIN

Học xong chương này, sinh viên sẽ nắm rõ các vấn đề sau:

- *Một số khái niệm về tập tin?*
- *Các bước thao tác với tập tin.*
- *Một số hàm truy xuất tập tin văn bản.*
- *Một số hàm truy xuất tập tin nhị phân.*

I. MỘT SỐ KHÁI NIỆM VỀ TẬP TIN

Đối với các kiểu dữ liệu ta đã biết như kiểu số, kiểu mảng, kiểu cấu trúc thì dữ liệu được tổ chức trong bộ nhớ trong (RAM) của máy tính nên khi kết thúc việc thực hiện chương trình thì dữ liệu cũng bị mất; khi cần chúng ta bắt buộc phải nhập lại từ bàn phím. Điều đó vừa mất thời gian vừa không giải quyết được các bài toán với số liệu lớn. Để giải quyết vấn đề, người ta đưa ra kiểu tập tin (file) cho phép lưu trữ dữ liệu ở bộ nhớ ngoài (đĩa). Khi kết thúc chương trình thì dữ liệu vẫn còn do đó chúng ta có thể sử dụng nhiều lần. Một đặc điểm khác của kiểu tập tin là kích thước lớn với số lượng các phần tử không hạn chế (chỉ bị hạn chế bởi dung lượng của bộ nhớ ngoài).

Có 3 loại dữ liệu kiểu tập tin:

- Tập tin văn bản (Text File): là loại tập tin dùng để ghi các ký tự lên đĩa, các ký tự này được lưu trữ dưới dạng mã Ascii. Điểm đặc biệt là dữ liệu của tập tin được lưu trữ thành các dòng, mỗi dòng được kết thúc bằng ký tự xuống dòng (new line), ký hiệu '\n'; ký tự này là sự kết hợp của 2 ký tự CR (Carriage Return - Về đầu dòng, mã Ascii là 13) và LF (Line Feed - Xuống dòng, mã Ascii là 10). Mỗi tập tin được kết thúc bởi ký tự EOF (End Of File) có mã Ascii là 26 (xác định bởi tổ hợp phím Ctrl + Z).

  Tập tin văn bản chỉ có thể truy xuất theo kiểu tuần tự.

- Tập tin định kiểu (Typed File): là loại tập tin bao gồm nhiều phần tử có cùng kiểu: char, int, long, cấu trúc… và được lưu trữ trên đĩa dưới dạng một chuỗi các byte liên tục.

- Tập tin không định kiểu (Untyped File): là loại tập tin mà dữ liệu của chúng gồm các cấu trúc dữ liệu mà người ta không quan tâm đến nội dung hoặc kiểu của nó, chỉ lưu ý đến các yếu tố vật lý của tập tin như độ lớn và các yếu tố tác động lên tập tin mà thôi.

Biến tập tin: là một biến thuộc kiểu dữ liệu tập tin dùng để đại diện cho một tập tin. Dữ liệu chứa trong một tập tin được truy xuất qua các thao tác với thông số là biến tập tin đại diện cho tập tin đó.

Con trỏ tập tin: Khi một tập tin được mở ra để làm việc, tại mỗi thời điểm, sẽ có một vị trí của tập tin mà tại đó việc đọc/ghi thông tin sẽ xảy ra. Người ta hình dung có một con trỏ đang chỉ đến vị trí đó và đặt tên nó là con trỏ tập tin.

Sau khi đọc/ghi xong dữ liệu, con trỏ sẽ chuyển dịch thêm một phần tử về phía cuối tập tin. Sau phần tử dữ liệu cuối cùng của tập tin là dấu kết thúc tập tin EOF (End Of File).

II. CÁC THAO TÁC TRÊN TẬP TIN

Muốn thao tác trên tập tin, ta phải lần lượt làm theo các bước:

- Khai báo biến tập tin.
- Mở tập tin bằng hàm fopen().
- Thực hiện các thao tác xử lý dữ liệu của tập tin bằng các hàm đọc/ghi dữ liệu.
- Đóng tập tin bằng hàm fclose().

Ở đây, ta thao tác với tập tin nhờ các hàm được định nghĩa trong thư viện stdio.h.

II.1. Khai báo biến tập tin

Cú pháp: FILE <Danh sách các biến con trỏ>

Các biến trong danh sách phải là các con trỏ và được phân cách bởi dấu phẩy(,).

Ví dụ: FILE *f1,*f2;

II.2. Mở tập tin

Cú pháp: FILE *fopen(char *Path, const char *Mode)

Trong đó:

- Path: chuỗi chỉ đường dẫn đến tập tin trên đĩa.
- Type: chuỗi xác định cách thức mà tập tin sẽ mở. Các giá trị có thể của *Mode*:

Chế độ	Ý nghĩa
r	Mở tập tin văn bản để đọc
w	Tạo ra tập tin văn bản mới để ghi
a	Nối vào tập tin văn bản
rb	Mở tập tin nhị phân để đọc
wb	Tạo ra tập tin nhị phân để ghi
ab	Nối vào tập tin nhị phân
r+	Mở một tập tin văn bản để đọc/ghi
w+	Tạo ra tập tin văn bản để đọc ghi
a+	Nối vào hay tạo mới tập tin văn bản để đọc/ghi
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

- Hàm fopen trả về một con trỏ tập tin. Chương trình của ta không thể thay đổi giá trị của con trỏ này. Nếu có một lỗi xuất hiện trong khi mở tập tin thì hàm này trả về con trỏ NULL.

Ví dụ: Mở một tập tin tên TEST.txt để ghi.

```
FILE *f;
f = fopen("TEST.txt", "w");
if (f!=NULL)
{
      /* Các câu lệnh để thao tác với tập tin*/
```

```
        /* Đóng tập tin*/
}
```

Trong ví dụ trên, ta có sử dụng câu lệnh kiểm tra điều kiện để xác định mở tập tin có thành công hay không?.

Nếu mở tập tin để ghi, nếu tập tin đã tồn tại rồi thì tập tin sẽ bị xóa và một tập tin mới được tạo ra. Nếu ta muốn ghi nối dữ liệu, ta phải sử dụng chế độ "a". Khi mở với chế độ đọc, tập tin phải tồn tại rồi, nếu không một lỗi sẽ xuất hiện.

II.3. Đóng tập tin

Hàm fclose() được dùng để đóng tập tin được mở bởi hàm fopen(). Hàm này sẽ ghi dữ liệu còn lại trong vùng đệm vào tập tin và đóng lại tập tin.

Cú pháp: int fclose(FILE *f)

Trong đó f là con trỏ tập tin được mở bởi hàm fopen(). Giá trị trả về của hàm là 0 báo rằng việc đóng tập tin thành công. Hàm trả về EOF nếu có xuất hiện lỗi.

Ngoài ra, ta còn có thể sử dụng hàm fcloseall() để đóng tất cả các tập tin lại.

Cú pháp: int fcloseall()

Kết quả trả về của hàm là tổng số các tập tin được đóng lại. Nếu không thành công, kết quả trả về là EOF.

II.4. Kiểm tra đến cuối tập tin hay chưa?

Cú pháp: int feof(FILE *f)

Ý nghĩa: Kiểm tra xem đã chạm tới cuối tập tin hay chưa và trả về EOF nếu cuối tập tin được chạm tới, ngược lại trả về 0.

II.5 Di chuyển con trỏ tập tin về đầu tập tin - Hàm rewind()

Khi ta đang thao tác một tập tin đang mở, con trỏ tập tin luôn di chuyển về phía cuối tập tin. Muốn cho con trỏ quay về đầu tập tin như khi mở nó, ta sử dụng hàm *rewind()*.

Cú pháp: void rewind(FILE *f)

III. TRUY CẬP TẬP TIN VĂN BẢN

III.1. Ghi dữ liệu lên tập tin văn bản

III.1.1 Hàm putc()

Hàm này được dùng để ghi một ký tự lên một tập tin văn bản đang được mở để làm việc.

Cú pháp: int putc(int c, FILE *f)

Trong đó, tham số c chứa mã Ascii của một ký tự nào đó. Mã này được ghi lên tập tin liên kết với con trỏ f. Hàm này trả về EOF nếu gặp lỗi.

III.1.2 Hàm fputs()

Hàm này dùng để ghi một chuỗi ký tự chứa trong vùng đệm lên tập tin văn bản.

Cú pháp: int puts(const char *buffer, FILE *f)

Trong đó, buffer là con trỏ có kiểu char chỉ đến vị trí đầu tiên của chuỗi ký tự được ghi vào. Hàm này trả về giá trị 0 nếu buffer chứa chuỗi rỗng và trả về EOF nếu gặp lỗi.

III.1.3 Hàm fprintf()

Hàm này dùng để ghi dữ liệu có định dạng lên tập tin văn bản.

Cú pháp: fprintf(FILE *f, const char *format, varexpr)

Trong đó: format: chuỗi định dạng (giống với các định dạng của hàm printf()), varexpr: danh sách các biểu thức, mỗi biểu thức cách nhau dấu phẩy (,).

Định dạng	Ý nghĩa
%d	Ghi số nguyên
%[.số chữ số thập phân] f	Ghi số thực có <số chữ số thập phân> theo quy tắc làm tròn số.
%o	Ghi số nguyên hệ bát phân
%x	Ghi số nguyên hệ thập lục phân
%c	Ghi một ký tự
%s	Ghi chuỗi ký tự
%e hoặc %E hoặc %g hoặc %G	Ghi số thực dạng khoa học (nhân 10 mũ x)

Ví dụ: Viết chương trình ghi chuỗi ký tự lên tập tin văn bản D:\\Baihat.txt

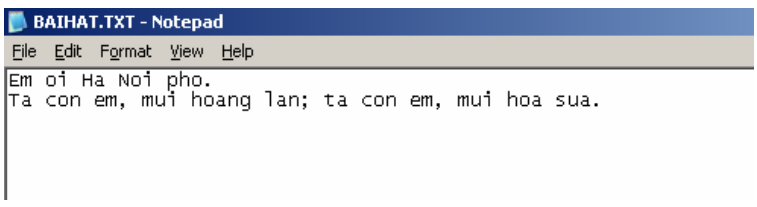
```
#include<stdio.h>
#include<conio.h>

int main()
{
   FILE *f;
   clrscr();
   f=fopen("D:\\Baihat.txt","r+");

   if (f!=NULL)
   {
     fputs("Em oi Ha Noi pho.\n",f);
     fputs("Ta con em, mui hoang lan; ta con em, mui hoa sua.",f);
     fclose(f);
   }
   getch();

   return 0;
}
```

Nội dung tập tin Baihat.txt khi được mở bằng trình soạn thảo văn bản Notepad.



III.2. Đọc dữ liệu từ tập tin văn bản

III.2.1 Hàm getc()

Hàm này dùng để đọc dữ liệu từ tập tin văn bản đang được mở để làm việc.

Cú pháp: int getc(FILE *f)

Hàm này trả về mã Ascii của một ký tự nào đó (kể cả EOF) trong tập tin liên kết với con trỏ f.

III.2.2 Hàm fgets()

Cú pháp: char *fgets(char *buffer, int n, FILE *f)

Hàm này được dùng để đọc một chuỗi ký tự từ tập tin văn bản đang được mở ra và liên kết với con trỏ f cho đến khi đọc đủ n ký tự hoặc gặp ký tự xuống dòng '\n' (ký tự này cũng được đưa vào chuỗi kết quả) hay gặp ký tự kết thúc EOF (ký tự này không được đưa vào chuỗi kết quả).

Trong đó:

- buffer (vùng đệm): con trỏ có kiểu char chỉ đến cùng nhớ đủ lớn chứa các ký tự nhận được.

- n: giá trị nguyên chỉ độ dài lớn nhất của chuỗi ký tự nhận được.

- f: con trỏ liên kết với một tập tin nào đó.

- Ký tự NULL ('\0') tự động được thêm vào cuối chuỗi kết quả lưu trong vùng đêm.

- Hàm trả về địa chỉ đầu tiên của vùng đệm khi không gặp lỗi và chưa gặp ký tự kết thúc EOF. Ngược lại, hàm trả về giá trị NULL.

III.2.3 Hàm fscanf()

Hàm này dùng để đọc dữ liệu từ tập tin văn bản vào danh sách các biến theo định dạng.

Cú pháp: fscanf(FILE *f, const char *format, varlist)

Trong đó: format: chuỗi định dạng (giống hàm scanf()); varlist: danh sách các biến mỗi biến cách nhau dấu phẩy (,).

Ví dụ: Viết chương trình chép tập tin D:\Baihat.txt ở trên sang tập tin D:\Baica.txt.

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

int main()
{
 FILE *f1,*f2;
 clrscr();
 f1=fopen("D:\\Baihat.txt","rt");
 f2=fopen("D:\\Baica.txt","wt");
 if (f1!=NULL && f2!=NULL)
 {
       int ch=fgetc(f1);
       while (! feof(f1))
       {
             fputc(ch,f2);
             ch=fgetc(f1);
       }
```

```
            fcloseall();
        }
        getch();
        return 0;
    }
```

IV. TRUY CẬP TẬP TIN NHỊ PHÂN

IV.1 Ghi dữ liệu lên tập tin nhị phân - Hàm fwrite()

Cú pháp: size_t fwrite(const void *ptr, size_t size, size_t n, FILE *f)

Trong đó:

- ptr: con trỏ chỉ đến vùng nhớ chứa thông tin cần ghi lên tập tin.
- n: số phần tử sẽ ghi lên tập tin.
- size: kích thước của mỗi phần tử.
- f: con trỏ tập tin đã được mở.
- Giá trị trả về của hàm này là số phần tử được ghi lên tập tin. Giá trị này bằng n trừ khi xuất hiện lỗi.

IV.2 Đọc dữ liệu từ tập tin nhị phân - Hàm fread()

Cú pháp: size_t fread(const void *ptr, size_t size, size_t n, FILE *f)

Trong đó:

- ptr: con trỏ chỉ đến vùng nhớ sẽ nhận dữ liệu từ tập tin.
- n: số phần tử được đọc từ tập tin.
- size: kích thước của mỗi phần tử.
- f: con trỏ tập tin đã được mở.
- Giá trị trả về của hàm này là số phần tử đã đọc được từ tập tin. Giá trị này bằng n hay nhỏ hơn n nếu đã chạm đến cuối tập tin hoặc có lỗi xuất hiện..

IV.3 Di chuyển con trỏ tập tin - Hàm fseek()

Việc ghi hay đọc dữ liệu từ tập tin sẽ làm cho con trỏ tập tin dịch chuyển một số byte, đây chính là kích thước của kiểu dữ liệu của mỗi phần tử của tập tin.

Khi đóng tập tin rồi mở lại nó, con trỏ luôn ở vị trí ngay đầu tập tin. Nhưng nếu ta sử dụng kiểu mở tập tin là "a" để ghi nối dữ liệu, con trỏ tập tin sẽ di chuyển đến vị trí cuối cùng của tập tin này.

Ta cũng có thể điều khiển việc di chuyển con trỏ tập tin đến vị trí chỉ định bằng hàm fseek().

Cú pháp: int fseek(FILE *f, long offset, int whence)

Trong đó:

- f: con trỏ tập tin đang thao tác.
- offset: số byte cần dịch chuyển con trỏ tập tin kể từ vị trí trước đó. Phần tử đầu tiên là vị trí 0.
- whence: vị trí bắt đầu để tính offset, ta có thể chọn điểm xuất phát là:

0	**SEEK_SET**	Vị trí đầu tập tin

1	**SEEK_CUR**	Vị trí hiện tại của con trỏ tập tin
2	**SEEK_END**	Vị trí cuối tập tin

- Kết quả trả về của hàm là 0 nếu việc di chuyển thành công. Nếu không thành công, 1 giá trị khác 0 (đó là 1 mã lỗi) được trả về.

IV.4 Ví dụ

Ví dụ 1: Viết chương trình ghi lên tập tin CacSo.Dat 3 giá trị số (thực, nguyên, nguyên dài). Sau đó đọc các số từ tập tin vừa ghi và hiển thị lên màn hình.

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

int main()
{
   FILE *f;
   clrscr();
   f=fopen("D:\\CacSo.txt","wb");
   if (f!=NULL)
   {
         double d=3.14;
         int i=101;
         long l=54321;
         fwrite(&d,sizeof(double),1,f);
         fwrite(&i,sizeof(int),1,f);
         fwrite(&l,sizeof(long),1,f);
         /* Doc tu tap tin*/
         rewind(f);
         fread(&d,sizeof(double),1,f);
         fread(&i,sizeof(int),1,f);
         fread(&l,sizeof(long),1,f);
         printf("Cac ket qua la: %f %d    %ld",d,i,l);
         fclose(f);
   }
   getch();
   return 0;
}
```

Ví dụ 2: 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.

Ta nhận thấy rằng mỗi phần tử của tập tin SinhVien.Dat là một cấu trúc có 2 trường: mã và họ tên. Do đó, ta cần khai báo cấu trúc này và sử dụng các hàm đọc/ghi tập tin nhị phân với kích thước mỗi phần tử của tập tin là chính kích thước cấu trúc đó.

```
#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,"ab");
   printf("Nhap bao nhieu sinh vien? ");scanf("%d",&n);
   fflush(stdin);
   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 tuc");
   getch();
}

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

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");
   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();
}

int main()
{
    int c;
    for (;;)
    {
          clrscr();
          printf("1. 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;
}
```

Ngoài ra thư viện stdio.h còn định nghĩa một số hàm khác cho phép thao tác với tập tin, sinh viên có thể tham khảo trong phần trợ giúp.

V. BÀI TẬP

V.1 Mục đích yêu cầu

Nắm vững cách sử dụng kiểu dữ liệu tập tin. Phân biệt nó với tất cả các kiểu dữ liệu có cấu trúc đã học. Làm quen và biết cách thao tác trên tập tin. Vận dụng các kiến thức đã học viết các chương trình trong phần nội dung.

V.2 Nội dung

1. Viết chương trình quản lý một tập tin văn bản theo các yêu cầu:

a- Nhập từ bàn phím nội dung một văn bản sau đó ghi vào đĩa.

b- Đọc từ đĩa nội dung văn bản vừa nhập và in lên màn hình.

c- Đọc từ đĩa nội dung văn bản vừa nhập, in nội dung đó lên màn hình và cho phép nối thêm thông tin vào cuối tập tin đó.

2. Viết chương trình cho phép thống kê số lần xuất hiện của các ký tự là chữ (‘A’..’Z’,’a’..’z’) trong một tập tin văn bản.

3. Viết chương trình đếm số từ và số dòng trong một tập tin văn bản.

4. Viết chương trình nhập từ bàn phím và ghi vào 1 tập tin tên là DMHH.DAT với mỗi phần tử của tập tin là 1 cấu trúc bao gồm các trường: Ma (mã hàng: char[5]), Ten (Tên

hàng: char[20]).Kết thúc việc nhập bằng cách gõ ENTER vào Ma. Ta sẽ dùng tập tin này để giải mã hàng hóa cho tập tin DSHH.DAT sẽ đề cập trong bài 5.

5. Viết chương trình cho phép nhập từ bàn phím và ghi vào 1 tập tin tên DSHH.Dat với mỗi phần tử của tập tin là một cấu trúc bao gồm các trường : mh (mã hàng: char[5]), sl (số lượng : int), dg (đơn giá: float), st (Số tiền: float) theo yêu cầu:

- Mỗi lần nhập một cấu trúc
- Trước tiên nhập mã hàng (mh), đưa mh so sánh với Ma trong tập tin DMHH.DAT đã được tạo ra bởi bài tập 1, nếu mh=ma thì in tên hàng ngay bên cạnh mã hàng.
- Nhập số lượng (sl).
- Nhập đơn giá (dg).
- Tính số tiền = số lượng * đơn giá.

Kết thúc việc nhập bằng cách đánh ENTER vào mã hàng. Sau khi nhập xong yêu cầu in toàn bộ danh sách hàng hóa có sự giải mã về tên hàng theo mẫu sau:

STT	MA HANG	TEN HANG	SO LG	DON GIA	SO TIEN
1	a0101	Duong cat trang	25	10000.00	250000.00
2	b0101	Sua co gai Ha Lan	10	40000.00	400000.00