CHƯƠNG 7 - THAO TÁC VÀO RA ĐỐI VỚI TỆP (FILE)

7.1. Khái niệm

Ngôn ngữ C có hai loại file:

- File nhị phân: Thường dùng để ghi các cấu trúc.
- File văn bản (text): Dùng để ghi dữ liệu dạng ký tự.

7.2. Các hàm thao tác với tệp

7.2.1. Khai báo file

```
FILE  *<biến file> ;
```

7.2.2. Mở file

```
FILE *fopen(const char *filename, const char
access_mode);
```

CHƯƠNG 7 - THAO TÁC VÀO RA ĐỐI VỚI TỆP (FILE)

Trong đó:

- ***const char * file name***: Tên tệp được mở (bao gồm cả đường dẫn, trong ngôn ngữ C phải chú ý dấu gạch chéo phân cách trong đường dẫn là hai gạch \\ ).
- ***const char access_mode***: Kiểu truy nhập khi mở file, nó được biểu diễn bởi một xâu ký tự cho biết kiểu mở tệp như là đọc, ghi, hay cả hai. Kiểu truy nhập có thể là:

Kiểu file	Chế độ mở	Ý nghĩa	Ghi chú
Nhị phân	"rb"	Mở để đọc	File đã có
	"wb"	Mở để ghi	Tạo file mới
	"ab"	Mở để bổ sung	File đã có hoặc tạo mới
	"r+b"	Mở để đọc hoặc ghi	File đã có
	"w+b"		Tạo file mới
	"a+b"		File đã có hoặc tạo mới

CHƯƠNG 7 - THAO TÁC VÀO RA ĐỐI VỚI TỆP (FILE)

Kiểu file	Chế độ mở	Ý nghĩa	Ghi chú
Van ban	"r" "rt"	Mở để đọc	File đã có
	"w" "wt"	Mở để ghi	Tạo file mới
	"a" "at"	Mở để bổ sung	File đã có hoặc tạo mới
	"r+" "r+t"	Mở để đọc hoặc ghi	File đã có
	"w" "w+t"		Tạo file mới
	"a" "a+t"		File đã có hoặc tạo mới

Ví dụ:

```
FILE  *tepvao, *tepra;
…
tepvao = fopen("dulieu.in", "r");
tepra = fopen("ketqua.out", "w")
```

CHƯƠNG 7 - THAO TÁC VÀO RA ĐỐI VỚI TỆP (FILE)

7.2.3. Đóng file

fclose (FILE *fp);

7.2.4. Đọc file

(a) Đọc file theo kiểu nhị phân:

```
int fread(void *ptr, int  size, int  n, FILE *<biến file> );
```

Đọc nội dung từ file đưa vào vùng nhớ trỏ bởi ptr, mỗi khối đọc có kích thước size byte, và sẽ đọc n khối nếu được, không thì sẽ đọc ít hơn n khối. Hàm trả về số khối thực sự đọc được.

(b) Đọc file theo kiểu văn bản (theo khuôn dạng):

```
int fscanf(FILE *fp,const char *dk,<danh sách địa chỉ biến>);
```

Trong đó:

- ***fp:*** con trỏ tệp.
- ***const char *dk:*** địa chỉ của chuỗi điều khiển quy cách đọc.
- ***<danh sách địa chỉ biến>:*** D/sách các địa chỉ biến cần nhập dữ liệu từ file.

Ví dụ:

```
fscanf(fp, " %d%d%d", &ngay, &thang, &nam);
```

CHƯƠNG 7 - THAO TÁC VÀO RA ĐỐI VỚI TỆP (FILE)

(c) Đọc một ký tự từ file văn bản:

```
int fgetc(FILE  *fp);
```

Trong đó:

- *fp* là con trỏ tệp.
- Hàm *fgetc* thực hiện việc đọc một ký tự từ tệp, nếu thành công thì cho mã đọc được, nếu gặp cuối tệp hay có lỗi, hàm cho EOF.

Ví dụ:

```
c =  fgetc(fp);      // Đọc một ký tự từ tệp và lưu vào biến c
```

(d) Đọc một dãy ký tự từ tệp văn bản:

```
char  *fgets (char  *s, int  maxchar,  FILE  *fp);
```

Trong đó:

- *fp* là con trỏ tệp.
- *s*: Con trỏ kiểu char trỏ tới một vùng nhớ chứa dãy ký tự đọc từ tệp.
- *maxchar*: Số nguyên xác định độ dài cực đại của dãy ký tự cần đọc.

Ví dụ:

- fgets(*xau*, 80, *fp*);

Thực hiện việc đọc một dãy ký tự từ tệp *fp* vào vùng nhớ *xau*

CHƯƠNG 7 - THAO TÁC VÀO RA ĐỐI VỚI TỆP (FILE)

7.2.5. Ghi file

(a) Ghi file theo kiểu nhị phân:

```
int  fwrite(void  *ptr,int size,int  n, FILE *<biến file> );
```

Hàm *fwrite* thực hiện ghi nội dung trong vùng nhớ trỏ bởi *ptr* vào file, mỗi khối ghi có kích thước *size byte*, và sẽ ghi *n* khối nếu được, không thì sẽ ghi ít hơn *n* khối. Hàm trả về số khối thực sự ghi được.

b) Ghi file theo kiểu văn bản (theo khuôn dạng):

```
int  fprintf(FILE *fp, const char *dk, <danh sách đối số>);
```

Trong đó:

- *fp*: con trỏ tệp.
- *const char *dk*: địa chỉ đầu của chuỗi điều khiển ghi.
- *<danh sách đối số>:* Danh sách các đối số cần ghi lên file.

Ví dụ:

```
fprintf(fp,  "So  duoc  ghi  len  tep  la:  %d\n",  i);
      //  Ghi số i lên tệp fp
```

CHƯƠNG 7 - THAO TÁC VÀO RA ĐỐI VỚI TỆP (FILE)

(c) Ghi một ký tự lên file văn bản:

```
int fputc(int  ch, FILE  *fp);
```

Trong đó: *fp* là con trỏ tệp.

ch là ký tự được ghi lên tệp.

Hàm *fputc* thực hiện việc ghi lên tệp *fp* một ký tự có mã bằng *ch* % 256 với *ch* là số nguyên không dấu, nếu thành công thì cho mã ký tự được ghi, trái lại hàm cho EOF.

Ví dụ: fputc('X', fp); // Ghi ký tự 'X' lên tệp fp

(d) Ghi một xâu ký tự lên tệp văn bản:

```
int  fputs (const  char  *s,  FILE  *fp);
```

Trong đó:

- *fp* là con trỏ tệp.
- *s*: Con trỏ kiểu char trỏ tới địa chỉ đầu của một xâu ký tự kết thúc bằng '\0'.

Hàm *fputs* thực hiện việc ghi một xâu ký tự lên tệp. Nếu ghi thành công hàm trả về số lượng ký tự thực sự được ghi lên file, nếu có lỗi thì hàm cho EOF.

Ví dụ: fputs("Vao du lieu", fp); // Ghi xâu "Vao du lieu" lên file fp.

CHƯƠNG 7 - THAO TÁC VÀO RA ĐỐI VỚI TỆP (FILE)

7.2.6. Làm sạch bộ đệm đọc/ghi file

(a) Hàm fflush:

```
fflush ( FILE  *fp);
```

Trong đó *fp* là con trỏ tệp.

Hàm fflush làm sạch vùng đệm của tệp *fp*. Nếu thành công hàm cho giá trị 0, ngược lại hàm cho EOF.

(b) Hàm fflushall:

```
fflushall( void) ;
```

Hàm *fflushall* làm sạch vùng đệm của các tệp đang mở. Nếu thành công hàm cho giá trị nguyên bằng số tệp đang mở, trái lại hàm cho EOF.

7.2.7. Kiểm tra cuối tệp

Dạng hàm: `int feof (FILE  *fp);`

Trong đó *fp* là con trỏ tệp.

Hàm *feof* dùng để kiểm tra cuối tệp *fp*. Nếu thành công, hàm trả về giá trị khác 0, trái lại hàm cho giá trị bằng 0.

CHƯƠNG 7 - THAO TÁC VÀO RA ĐỐI VỚI TỆP (FILE)

VD: Lập chương trình nhập từ bàn phím một dãy số nguyên khác 0 và ghi vào file văn bản. Hiện nội dung file ra màn hình, tính tổng các số trong file.

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
char    tenfile[15];
FILE *fp;
```

CHƯƠNG 7 - THAO TÁC VÀO RA ĐỐI VỚI TỆP (FILE)

```
void Ghi(char tentep [ ])
{   int i , tg;
   fp = fopen(tentep,"w");
   if(fp==NULL)
      {       printf("Loi mo tep %s",tentep); exit(0); }
   i = 1;
   do
      {       printf("Nhap so thu %d (nhap so 0 de ket
thuc): ",i);       scanf("%d", &tg);
              if(tg != 0)
                    { fprintf(fp,"%d\n",tg); i++;  }
       } while (tg != 0);
   fclose(fp);
}
```

CHƯƠNG 7 - THAO TÁC VÀO RA ĐỐI VỚI TỆP (FILE)

```
void Hien_TinhTong(char  tentep [ ])
{  int i , tg , tong;
   fp = fopen(tentep , "r");
   if(fp == NULL)
     { printf("Loi mo tep '%s'",tentep);            exit(0);     }
   tong = 0;
   printf("\nDay so doc tu file:\n");
   while(fscanf(fp,"%d",&tg) > 0)
     { printf("%5d", tg);
        tong  +=  tg;
     }
   printf("\nTong = %d", tong);
   fclose(fp);
}
```

CHƯƠNG 7 - THAO TÁC VÀO RA ĐỐI VỚI TỆP (FILE)

```
void main()
{
   clrscr();
   printf("Nhap ten tep can luu: ");
   fflush(stdin);
   gets(tenfile);
   Ghi(tenfile);
   Hien_TinhTong(tenfile);
   getch();
}
```

CHƯƠNG 7 - THAO TÁC VÀO RA ĐỐI VỚI TỆP (FILE)

VD: Đọc dữ liệu từ một tệp có tên là "matran.vao" vào một ma trận và ghi các phần tử của ma trận này lên một tệp mới có tên là "matran.ra ".

```
#include <stdio.h>
#include <stdlib.h>
void main()
  {
  FILE   *tepvao, *tepra;
  float   a[50][50];
  int   m , n , i , j;
  float  x;
```

CHƯƠNG 7 - THAO TÁC VÀO RA ĐỐI VỚI TỆP (FILE)

```
// Nhap ma tran tu tep "matran.vao"
  tepvao = fopen("matran.vao","r");
  fscanf(tepvao,"%d",&m);
  fscanf(tepvao,"%d",&n);
  for (i=0; i<m; i++)
    for (j=0; j<n; j++)
      {
      fscanf(tepvao,"%f", &x);
      a[i][j] = x;
      }
  fclose(tepvao);
```

CHƯƠNG 7 - THAO TÁC VÀO RA ĐỐI VỚI TỆP (FILE)

```
// In ma tran ra tep “matran.ra”
  tepra = fopen(“matran.ra”,"w");
  fprintf(tepra,"Ma tran A \n");
  for (i=0; i < m; i++)
    {
    for (j=0; j < n; j++)
        fprintf(tepra," %8.2f", a[i][j]);
    fprintf(tepra,"\n");
    }
  fclose(tepra);
  }
```