高级语言程序设计

主讲：王老师

尚德机构

学习是一种信仰

第十章　文件

本章内容

- 文件概述
- 文件的打开和关闭函数
- 文件的读/写函数
- 文件的定位函数

第一节 文件概述

C语言将文件看做是由一个一个的字符（ASCII码文件）或字节（二进制文件）组成的，这种文件称为流式文件。

文件定义

文件是存储在外部存储介质上的数据集合，这些数据的类型可以是整型、实型、字符型、字符串或结构体类型等。目前，外部存储介质主要是磁盘，存储在磁盘上的文件称为“磁盘文件”。

文件命名

为了标识磁盘上的不同文件，每个磁盘文件都必须有一个文件名，其一般组成为：

盘符:路径\文件主名[.扩展名]

其中，盘符表示文件所在的磁盘，可以是A、B、C和D等。路径是由目录序列组成，目录之间的分隔符为“\”，需要注意的是，如果路径出现在字符串中，C语言规定其中的符号“\”需要写成转义字符“\\”。文件主名和扩展名均是由字母和数字等字符组成的。

C语言中，常见的扩展名：

c：C源程序文件

obj：C源程序经过编译后生成的目标文件。

exe：目标文件经过链接后生成的可执行文件。

一般地，数据文件的扩展名常用dat，文本文件的扩展名常用txt。

文件分类

1、按照文件的内容划分

可分为程序文件和数据文件，程序文件又可以分为源文件、目标文件和可执行文件。

文件分类

2、按照文件的组织形式划分

可分为顺序存取文件和随机存取文件。

（1）顺序存取文件简称为“顺序文件”，数据写入文件的方式是后输入的数据放在以前输入数据的后面，按照数据的先后次序一个接一个地存放。

（2）随机存取文件简称为“随机文件”，每个数据在磁盘中所占用的长度都相同。

文件分类

3、按照文件的存储形式划分

可分为文本文件和二进制文件。

（1）文本文件中数据转换为对应的ASCⅡ码来存放，每字节存储一个字符，因此便于对字符逐一进行处理。

（2）二进制文件中数据按照其二进制形式来存放，即把内存中的数据，原样输出到磁盘文件中。

文件分类

4、按照文件的存储介质划分

可以分为普通文件即存储介质文件（磁盘、磁带等）和设备文件即非存储介质（键盘、显示器和打印机等）文件。

文件存取方式

文件存取方式主要包括顺序存取和随机存取两种：

1、顺序存取是自上而下逐一地存取文件的内容。保存数据时，将数据附加到文件的尾部。顺序存取方式常用于文本文件，被存取的文件则称为顺序文件。

2、随机存取是以一个完整的单位进行数据的读取和写入。随机存取方式常用于二进制文件，被存取的文件则称为随机文件。

文件系统

C语言使用的文件系统分为缓冲文件系统（标准I/O）和非缓冲文件系统（系统I/O）：

1、缓冲文件系统是指系统自动地为每个正在使用的文件在内存开辟一个缓冲区。从内存向磁盘输出数据时，必须首先输出到缓冲区。待缓冲区装满后，再一起输出到磁盘文件。从磁盘文件文件向内存读入数据时，恰好相反，即首先将一批数据读入到缓冲区，再从缓冲区将数据逐一送到程序数据区。

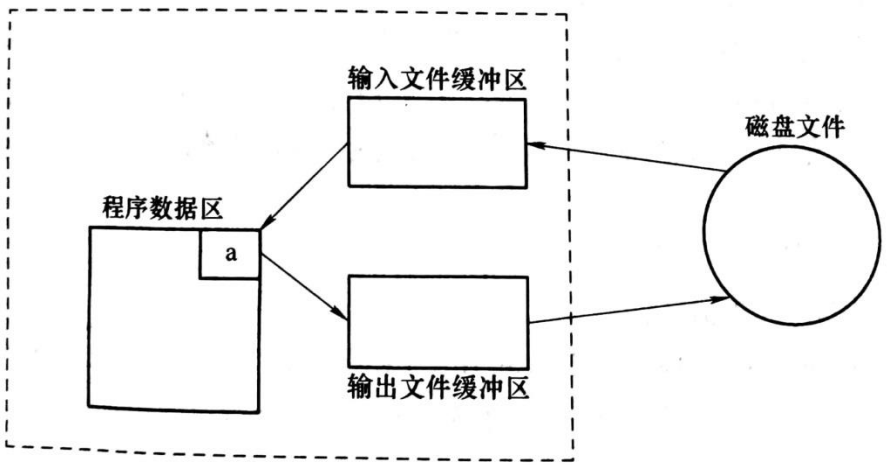

图 10-2　缓冲文件系统示意图

文件系统

2、非缓冲文件系统是指用户在程序中为每个文件设定缓冲区。

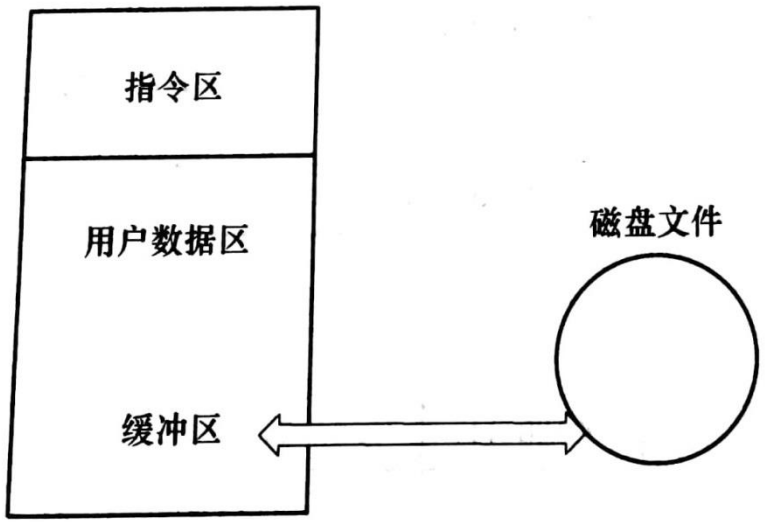

图 10-3　非缓冲文件系统示意图

课堂练习

1、C语言中，可以处理的两类文件分别是文本文件和_____。

课堂练习

1、C语言中，可以处理的两类文件分别是文本文件和_____。

答案：二进制文件

解析：按文件的存储形式划分分为文本文件和二进制文件。

第二节 文件的打开和关闭函数

文件打开函数：

【函数首部】FILE *fopen(char *filename,char *mode)

【参数】filename 字符型指针，所指向的字符串是要打开的“文件名”。

mode 字符型指针，所指向的字符串是对打开文件规定的“使用方式”。

“使用方式”是由r、w、a、t、b和+六个字符组成。如果使用方式中有b字符，则针对的是二进制文件；如果使用方式中没有b字符，则针对的是文本文件；如果使用r打开一个文件时，则该文件必须已经存在。

文件打开函数

【功能】按mode指定的“使用方式”打开filename指定的“文件名”对应的文件，同时自动地为该文件分配一个内存缓冲区。

【返回值】若打开文件正确，则返回一个“文件型”指针，程序通过该指针获得文件信息，访问文件并进行各种操作。若打开文件出错，则返回值为NULL，其中NULL是在头文件stdio.h中定义的符号常量，其值为0，含义是“空”。

库函数exit()的具体说明：

【函数首部】void exit([程序状态值])

【功能】关闭已经打开的所有文件，结束程序运行返回操作系统，并将“程序状态值”返回给操作系统。当“程序状态值”为0时，表示程序正常退出；当它为非0时，表示程序出错退出。

表 10-1　文件使用方式中各字符的含义

字　符	含　义	字　符	含　义
r(read)	读	t(text)	文本文件（可省略不写）
w(write)	写	b(binary)	二进制文件
a(append)	追加	+	读和写

表 10-2　文件的使用方式说明

文件类型	字符	使用方式	说　明
文本文件	"r"	只读	以只读方式打开一个已经存在的文本文件，只允许读取数据，不允许写入数据
	"r+"	读/写	以读/写方式打开一个已经存在的文本文件，允许读取和写入数据
	"w"	只写	以只写方式打开一个文本文件，只允许写入数据，不允许读取数据。若文件已经存在，则删去原文件的所有内容，然后写入数据；若文件不存在，则自动创建一个新文件后写入数据
	"w+"	读/写	以读/写方式打开一个文本文件，允许读取和写入数据。若文件已经存在，则删去原文件的所有内容，然后写入数据；若文件不存在，则自动创建一个新文件后写入数据
	"a"	追加	以追加的方式打开一个已经存在的文本文件，只允许写入数据并追加到原文件尾
	"a+"	读/写	以读/写方式打开一个文本文件，允许读取数据，也允许写入数据到文件尾。若文件不存在，则自动创建一个新文件
二进制文件	"rb"	只读	以只读方式打开一个已经存在的二进制文件，只允许读取数据，不允许写入数据
	"wb"	只写	以只写方式打开一个二进制文件，只允许写入数据，不允许读取数据。若文件已经存在，则删去原文件的所有内容，然后写入数据；若文件不存在，则自动创建一个新文件后写入数据
	"ab"	追加	以追加的方式打开一个已经存在的二进制文件，只允许写入数据并追加到原文件尾
	"rb+"	读/写	以读/写方式打开一个已经存在的二进制文件，允许读取和写入数据
	"wb+"	读/写	以读/写方式打开一个二进制文件，允许读取和写入数据。若文件已经存在，则删去原文件的所有内容，然后写入数据；若文件不存在，则自动创建一个新文件后写入数据
	"ab+"	读/写	以读/写方式打开一个二进制文件，允许读取数据，也允许写入数据到文件尾。若文件不存在，则自动创建一个新文件

文件关闭函数

【函数首部】int fclose(FILE *fp)

【参数】fp 文件型指针，通过函数fopen()获得，且指向一个已经打开的文件。

【功能】关闭fp所指向的文件，同时自动释放分配给该文件的内存缓冲区。

【返回值】若正确关闭指定的文件，则返回0；否则返回非0。

标准设备文件的打开和关闭

程序开始运行时，系统自动打开三种标准设备文件，并分别定义了相应的文件型指针。其中：

1、stdin指向标准输入（通常为键盘）。如果程序指定要从stdin所指向的文件输入数据，即从键盘输入数据。

2、stdout指向标准输出（通常为显示器）。如果程序指定要向stdout所指向的文件输出数据，即向显示器输出数据。

3、stderr指向标准错误输出（通常为显示器）。

三种标准设备文件使用后，不必关闭。因为系统退出时将自动关闭。

文件的打开与关闭

- C文件操作用库函数实现,包含在stdio.h
- 文件使用方式:打开文件→文件读/写→关闭文件
- 系统自动打开和关闭三个标准文件:
  - 标准输入------键盘　　stdin
  - 标准输出------显示器　　stdout
  - 标准出错输出-----显示器　　stderr

★文件的打开（fopen函数）

- 函数原型：　FILE　*fopen(char　*name,char *mode)
- 调用方式：fopen("文件名","使用文件方式")

例：FILE *fp;

```
fp=fopen("d:\\user\\myfile.txt","r");
```

为什么 \\

Read

❖返回值：正常打开，返回指向文件结 构体的指针；打开失败，返回NULL

```
FILE   *fp;
fp=fopen("aa.c","w");
if(fp==NULL)
{
    printf("File open error!\n");
    exit(0);     /*关闭所有文件终止调用*/
}
```

r:读方式；

w:写方式；

a:追加方式；

rb/wb/ab:二进制方式；

+:既可读又可写

方式	处理方式	指定文件不存在	文件存在
r	只读	出错	正常
w	只写	建立新文件	原内容丢失
a	追加	建立新文件	在原内容后追加
r+/w+	读写	出错	正常

★文件的关闭（fclose函数）

❖作用:使文件指针变量与文件“脱钩”，释放文件结构体和文件指针

```
FILE *fp;
fp=fopen(“a.txt”,“r”);
fclose(fp);
```

❖返回值：用于表示文件是否被正确地关闭，如果文件顺利关闭，该值为0，否则为-1(EOF)。

- 返回值可以用ferror函数测试

课堂练习

2、若使用fopen函数打开一个新的二进制文件，对该文件进行读写操作，则文件使用方式字符串应该是_____。

课堂练习

2、若使用fopen函数打开一个新的二进制文件，对该文件进行读写操作，则文件使用方式字符串应该是_____。

答案：“wb+”

解析：“wb+”以读写方式打开一个二进制文件，允许读取和写入数据。

课堂练习

3、当打开文件出现错误时，系统函数fopen（）的返回值是（　）

A. 1

B. -1

C. 非0

D. NULL

课堂练习

3、当打开文件出现错误时，系统函数fopen（）的返回值是（　）

A. 1

B. -1

C. 非0

D. NULL

答案：D

解析：文件打开函数的返回值是若打开文件正确，则返回一个“文件型”指针，程序通过该指针获得文件信息，访问文件并进行各种操作。若打开文件出错，则返回值为NULL，其中NULL是在头文件stdio.h中定义的符号常量，其值为0，含义是“空”。

第三节 文件的读/写函数

文件尾测试函数：

文件执行读操作时，通常使用feof( )函数来判断是否到达文件尾。如果遇到文件尾，则不能继续读取数据。

【函数首部】int feof(FILE *fp)

【参数】fp文件型指针，通过fopen()函数获得，指向一个已经打开的文件。

【功能】判断fp所指向的文件是否到达文件尾。

【返回值】若遇到文件尾，则返回值是非0；否则，返回值是0。

feof()函数同时适用于文本文件和二进制文件。对二进制文件执行读操作时，必须使用feof()函数来判断是否到达文件尾；对文本文件执行读操作时，如果遇到文件尾，则返回一个文件结束标志EOF（EOF是一个符号常量，其值在头文件stdio.h中被定义为-1，含义是“文件尾”）。

写字符函数fputc( )

【函数首部】int fputc(char ch,FILE *fp)

【参数】ch写到文件中的字符，既可以是字符常量，也可以是字符变量或字符表达式。

fp 文件型指针，通过函数fopen()获得，且指向一个已经打开的文件。

【功能】将ch中的字符写到fp所指向文件的当前位置，同时将读/写位置指针后移1字节，即指向下一个读/写位置。

【返回值】若正确写入字符，则返回刚写入到文件的字符；否则，返回EOF。

本函数主要用于文本文件，也可以用于二进制文件。对于文本文件，写入的是单个字符。对于二进制文件，写入的是1字节数据。当正确地写入一个字符或1字节数据后，文件内部的读/写位置指针会后移1字节，即指向下一个读/写位置。

读字符函数fgetc( )

【函数首部】int fgetc(FILE *fp)

【参数】fp 文件型指针，通过函数fopen()获得，指向一个已经打开的文件。

【功能】从fp所指向文件的当前位置读取一个字符，同时将读/写位置指针后移1字节，即指向下一个读/写位置。

【返回值】若正确读取字符，则返回读取的单个字符；否则，返回EOF。

文件的读写

文件打开之后，就可以对它进行读与写的操作了。

★读 / 写文件中的一个字符

❖fputc函数（putc函数）

- 函数原型： int fputc(int c, FILE *fp)
- 功能：把一字节代码c写入fp指向的文件中
- 返值：正常，返回c;出错，为EOF（-1）

字符常量或变量

文件指针变量

❖fgetc函数（getc函数）

- 函数原型：int fgetc(FILE *fp)
- 功能：从fp指向的文件中读取一字节代码
- 返值：返回读到的代码值;读到文件尾或出错为EOF（-1）

文件指针变量

❖feof函数

- 调用方式：feof(fp)
- 功能：对于二进制文件读取时判断文件是否结束。
- 返值：结束-1；反之0。

例 从键盘输入一些字符，逐个把它们送到磁盘上去，直到输入一个“#”为止。

```
#include <stdio.h>
#include <stdlid.h>
void main()
{
  FILE *fp;
  char  ch , filename[10];
  scanf("%s“, filename);
  if((fp=fopen(filename,"w"))==NULL)
   { printf("cannot open file\n");
     exit(0); }     /*终止程序*/
  ch=getchar(); /*接收执行scanf时最后输入的回车符*/
  ch=getchar(); /*第一个输入的字符被赋给变量ch*/
  while(ch!='#')
    { fputc(ch,fp);  /*字符被写入filename表示的文件中*/
      putchar(ch);   /*字符被输出到显示器*/
      ch=getchar();
    }
  putchar(10);       /*向屏幕输出一个换行符*/
  fclose(fp);
 }       /*关闭文件*/
```

写字符串函数fputs( )

【函数首部】int fputs(char *str,FILE *fp)

【参数】str 字符型指针，可以是一个字符常量，或字符数组名，或字符指针变量名。

fp 文件型指针，通过函数fopen( )获得，指向一个已经打开的文件。

【功能】把str所指向的一个字符串，舍去字符串结束标志符 ‘\0’ 后写入fp所指向文件的当前位置。同时将读/写位置指针向后移动strlen（字符串长度）字节。

【返回值】若正确写入字符串，则返回写入文件的实际字符数；否则，返回EOF。

读字符串函数fgets( )

【函数首部】char *fgets(char *str,int length,FILE *fp)

【参数】str 字符型指针，可以是一个字符数组名，或字符指针变量名。

length 整型，可以是整型常量、变量或表达式。

fp 文件型指针，通过函数fopen()获得，指向一个已经打开的文件。

【功能】从fp所指向文件的当前位置读入一个字符串，即读取length-1个字符，并在尾部自动补充一个字符串结束标志符 ‘\0’ ，组成字符串后存入str所指定的内存区；同时，将读/写位置指针后移实际读出的字符个数字节。如果在读取前n-1个字符中遇到文件尾EOF或回车，则读取结束，并在读取的字符后面补充一个字符串结束标志符 ‘\0’ 组成字符串。

【返回值】若正确读取字符串，则返回str对应的地址；否则，返回NULL。

例 从键盘读入字符串存入文件，再从文件读回显示

```
#include<stdio.h>
void main()
{
   FILE  *fp;
   char  string[81];
   if((fp=fopen("file.txt","w"))==NULL)
     { printf("cann't open file");exit(0); }
   while(strlen(gets(string))>0)
     { fputs(string,fp);
        fputs("\n",fp); }
   fclose(fp);
   if((fp=fopen("file.txt","r"))==NULL)
     { printf("cann't open file");exit(0); }
   while(fgets(string,81,fp)!=NULL)
      fputs(string,stdout);
   fclose(fp);
 }
```

写数据块函数fwrite( )

【函数首部】int fwrite(void *buffer,int size,int count,FILE *fp)

【参数】buffer 字符型指针，存放写入文件数据类型的起始地址，可以是存放数据的变量地址或数组首地址，也可以是指向某个变量或数组的指针变量。

size 无符号整数，可以是整型常量、变量或表达式，表示从文件写入文件的每个数据所占用的字节数，通常使用表达式sizeof(数据类型符)。

count 无符号整数，可以是整型常量、变量或表达式，表示要写入文件的数据的个数。

fp 文件型指针，通过函数fopen()获得，指向一个已经打开的可写文件。

【功能】将从buffer开始的count个数据（每个数据占用size字节）一次性写入fp所指向的文件。同时，将读/写位置指针后移size*count字节。

【返回值】若正确写入数据块，则返回值是count；否则，返回值是0。

读数据块函数fread( )

【函数首部】int fread(void *buffer,int size,int count,FILE *fp)

【参数】buffer 字符型指针，存放从文件中读取数据的起始地址，可以是存放数据的变量地址或数组首地址，也可以是指向某个变量或数组的指针变量。

size 无符号整数，可以是整型常量、变量或表达式，表示从文件读取的每个数据所占用的字节数，通常使用表达式sizeof(数据类型符)。

count 无符号整数，可以是整型常量、变量或表达式，表示从文件中读取数据的个数。

fp 文件型指针，通过函数fopen()获得，指向一个已经打开的可读文件。

【功能】从fp所指向文件的当前位置开始，一次性读入count个数据（每个数据占用size字节），并将读入的数据存放到以buffer为首地址的内存区；同时，将读/写位置指针后移size*count字节。

【返回值】若正确读取数据块，则返回值是count；否则，返回值是0。

需要注意的是，用fread()和fwrite()函数进行数据读/写时，必须采用二进制。

格式化写函数fprintf( )

【函数首部】int fprintf(FILE *fp,char *format[,argument...])

【参数】fp 文件型指针，通过函数fopen()获得，指向一个已经打开的可写文件。

format 格式字符串。

argument 输出列表。

【功能】将argument的值以format指定的格式写入fp所指向的文件。

【返回值】若正确写入数据，则返回实际输出字符的个数；否则，返回值是负数。

格式化读函数fscanf( )

【函数首部】int fscanf(FILE *fp,char *format[,argument...])

【参数】fp 文件型指针，通过函数fopen()获得，指向一个已经打开的可读文件。

format 格式字符串。

argument 输入列表。

【功能】根据format指定的格式从fp所指向文件中读取数据保存至argument所指向的内存单元。

【返回值】若正确读入数据，则返回值是已读入的数据个数；否则，返回值是EOF。

课堂练习

4、若文件型指针fp已指向某文件的末尾，则函数feof(fp)的返回值是_____。

课堂练习

4、若文件型指针fp已指向某文件的末尾，则函数feof(fp)的返回值是_____。

答案：非0

解析：文件尾测试函数feof（）的功能是若遇到文件尾，则返回值是非0；否则，返回值是0。

课堂练习

5、在文件使用中，EOF的值是_____。

课堂练习

5、在文件使用中，EOF的值是_____。

答案：-1

解析：对文本文件执行读操作时，如果遇到文件尾，则返回一个文件结束标志EOF（EOF是一个符号常量，其值在文件头stdio.h中被定义为-1，含义是“文件尾”）。

课堂练习

5、在文件使用中，EOF的值是_____。

答案：-1

解析：对文本文件执行读操作时，如果遇到文件尾，则返回一个文件结束标志EOF（EOF是一个符号常量，其值在文件头stdio.h中被定义为-1，含义是“文件尾”）。

第四节 文件的定位函数

文件指针复位函数：

【函数首部】int rewind(FILE *fp)

【参数】fp 文件型指针，通过函数fopen()获得，指向一个已经打开的文件。

【功能】将fp所指向文件的位置指针重置到文件头，即把文件的位置指针重新定位到fp所指向文件的起始位置。

【返回值】若执行正常，返回值是0；否则，返回值是非0。

★rewind函数

- ❖函数原型： void rewind(FILE *fp)
- ❖功能：重置文件位置指针到文件开头
- ❖返值：无

例 对一个磁盘文件进行显示和复制两次操作

```
#include <stdio.h>
void main()
{
   FILE *fp1,*fp2;
   fp1=fopen("c:\\c\\user\\ch4.c","r");
   fp2=fopen("d:\\c\\user\\ch41.c","w");
   while(!feof(fp1))  putchar(getc(fp1));
   rewind(fp1);
   while(!feof(fp1))  putc(getc(fp1),fp2);
   fclose(fp1);   fclose(fp2);
}
```

文件随机定位函数

文件随机定位函数：

文件随机定位函数fseek( )可以将文件的位置指针移动到文件中的任何一个地方，一般用于二进制。

【函数首部】int fseek(FILE *fp,long offset,int origin)

【参数】fp 文件型指针，通过函数fopen()获得，指向一个已经打开的文件。

offset 长整型数据，表示位移量，是从origin为起始位置，向后（当位移量>0时）或向前（当位移量<0时）移动的字节数。

origin 表示起始位置。

文件随机定位函数

表 10-4　起始位置可选取的参数值

整　　数	0	1	2
符号常量	SEEK_SET	SEEK_CUR	SEEK_END
对应的起始位置	文件头	位置指针的当前位置	文件尾

上述的符号常量是在头文件 stdio.h 中定义的。

【功能】将fp所指向文件的位置指针从origin指定的起始位置移动offset所指定的字节数，改变文件位置指针的位置，即指向新的位置。

【返回值】若定位成功，则返回值是0；否则，返回值是非0。

课堂练习

6、系统函数rewind的作用是（ ）。

A. 将文件内部指针指向文件末尾

B. 将文件内部指针指向文件开头

C. 将文件内部指针下移一个字符位置

D. 将文件内部指针随机指向文件任意位置

课堂练习

6、系统函数rewind的作用是（ ）。

A. 将文件内部指针指向文件末尾

B. 将文件内部指针指向文件开头

C. 将文件内部指针下移一个字符位置

D. 将文件内部指针随机指向文件任意位置

答案：B

解析：函数rewind（）的功能是将指向文件的位置的指针重置到文件头，即把文件的位置指针重新定位到文件的起始位置。

课堂练习

7、系统函数 int rewind（FILE *fp）的功能是_____。

课堂练习

7、系统函数 int rewind（FILE *fp）的功能是_____。

答案：文件内部的指针重新指向一个流的开头

解析：系统函数 int rewind（FILE *fp）的功能是将fp所指向文件的位置指针重置到文件头，即把文件的位置指针重新定位到fp所指向文件的起始位置。

祝大家顺利通过考试！