<https://www.cnblogs.com/saolv/p/7793379.html>

C语言中没有输入输出语句，所有的输入输出功能都用 ANSI C提供的一组标准库函数来实现。文件操作标准库函数有：

　　 　　 文件的打开操作 fopen 打开一个文件

　　 　　 文件的关闭操作 fclose 关闭一个文件

　　 　　 文件的读写操作 fgetc 从文件中读取一个字符  
　　 　　 　　　　　　　 fputc 写一个字符到文件中去  
　　 　　 　　　　　　　 fgets 从文件中读取一个字符串  
　　 　　 　　　　　　　 fputs 写一个字符串到文件中去  
　　 　　 　　　　　　　 fprintf 往文件中写格式化数据  
　　 　　 　　　　　　　 fscanf 格式化读取文件中数据  
　　 　　 　　　　　　　 fread 以二进制形式读取文件中的数据  
　　 　　 　　　　　　　 fwrite 以二进制形式写数据到文件中去  
　　 　　 　　　　　　　 getw 以二进制形式读取一个整数  
　　 　　 　　　　　　　 putw 以二进制形式存贮一个整数

　　 　文件状态检查函数 feof 文件结束  
　　 　　 　　　　　　　 ferror 文件读/写出错  
　　 　　 　　　　　　　 clearerr 清除文件错误标志  
　　 　　 　　　　　　　 ftell 了解文件指针的当前位置

　　　　　　文件定位函数 rewind 反绕  
　　 　　 　　　　　　　 fseek 随机定位

**# 文件的打开**  
　1．函数原型

FILE \*fopen(char \*pname,char \*mode)

　2．功能说明  
　　 按照mode 规定的方式，打开由pname指定的文件。若找不到由pname指定的相应文件，就按以下方式之一处理：  
（1） 此时如mode 规定按写方式打开文件，就按由pname指定的名字建立一个新文件；  
（2） 此时如mode 规定按读方式打开文件，就会产生一个错误。

打开文件的作用是：  
（1）分配给打开文件一个FILE 类型的文件结构体变量，并将有关信息填入文件结构体变量；  
（2）开辟一个缓冲区；  
（3）调用操作系统提供的打开文件或建立新文件功能，打开或建立指定文件；  
FILE \*：指出fopen是一个返回文件类型的指针函数；

　3．参数说明  
pname：是一个字符指针，它将指向要打开或建立的文件的文件名字符串。  
mode：是一个指向文件处理方式字符串的字符指针。所有可能的文件处理方式见表8.1

　4．返回值  
　　　正常返回：被打开文件的文件指针。  
　　　异常返回：NULL，表示打开操作不成功。

例如：

[复制代码](javascript:void(0);)

//定义一个名叫fp文件指针

FILE \*fp；

//判断按读方式打开一个名叫test的文件是否失败

if((fp=fopen（"test"，"r"）) == NULL)//打开操作不成功

{

printf("The file can not be opened.\n")；

exit(1);//结束程序的执行

}

[复制代码](javascript:void(0);)

　要说明的是：C语言将计算机的输入输出设备都看作是文件。例如，键盘文件、屏幕文件等。ANSI C标准规定，在执行程序时系统先自动打开键盘、屏幕、错误三个文件。这三个文件的文件指针分别是：标准输入stdin、标准输出stdout和标准出错 stderr。

**\* 文件的关闭**

1． 函数原型

int fclose(FILE \*fp)；

2． 功能说明  
　　关闭由fp指出的文件。此时调用操作系统提供的文件关闭功能，关闭由fp->fd指出的文件；释放由fp指出的文件类型结构体变量；返回操作结果，即0或EOF。

3． 参数说明  
　　fp：一个已打开文件的文件指针。

4． 返回值  
　　正常返回：0。  
　　异常返回：EOF，表示文件在关闭时发生错误。  
例如：

int n=fclose(fp);

**\*文件的读写操作**

A. 从文件中读取一个字符  
　　1． 函数原型

int fgetc(FILE \*fp)；

　　2． 功能说明  
　　　　从fp所指文件中读取一个字符。  
　　3． 参数说明  
　　　　fp：这是个文件指针，它指出要从中读取字符的文件。  
　　4． 返回值  
　　　　正常返回： 返回读取字符的代码。  
　　　　非正常返回：返回EOF。例如，要从"写打开"文件中读取一个字符时，会发生错误而返回一个EOF。  
　　5． 实例

【例8.1】显示指定文件的内容。

[复制代码](javascript:void(0);)

//程序名为：display.c

//执行时可用：display filename1 形式的命令行运行。显示文件filename1中的内容。例如，执行命令行display display.c将在屏幕上显示display的原代码。

//File display program.

#include <stdio.h>

void main(int argc,char \*argv[]) //命令行参数

{

int ch;//定义文件类型指针

FILE \*fp;//判断命令行是否正确

if(argc!=2)

{

printf("Error format,Usage: display filename1\n");

return; //键入了错误的命令行，结束程序的执行

}

//按读方式打开由argv[1]指出的文件

if((fp=fopen(argv[1],"r"))==NULL)

{

printf("The file <%s> can not be opened.\n",argv[1]);//打开操作不成功

return;//结束程序的执行

}

//成功打开了argv[1]所指文件

ch=fgetc(fp); //从fp所指文件的当前指针位置读取一个字符

while(ch!=EOF) //判断刚读取的字符是否是文件结束符

{

putchar(ch); //若不是结束符，将它输出到屏幕上显示

ch=fgetc(fp); //继续从fp所指文件中读取下一个字符

} //完成将fp所指文件的内容输出到屏幕上显示

fclose(fp); //关闭fp所指文件

}

[复制代码](javascript:void(0);)

B. 写一个字符到文件中去

1． 函数原型

int fputc(int ch,FILE \*fp)

2． 功能说明  
　　把ch中的字符写入由fp指出的文件中去。  
3． 参数说明  
　　ch：是一个整型变量，内存要写到文件中的字符（C语言中整型量和字符量可以通用）。  
　　fp：这是个文件指针，指出要在其中写入字符的文件。  
4． 返回值  
　　正常返回： 要写入字符的代码。  
　　非正常返回：返回EOF。例如，要往"读打开"文件中写一个字符时，会发生错误而返回一个EOF。  
5． 实例

【例8.2】将一个文件的内容复制到另一个文件中去。

[复制代码](javascript:void(0);)

//程序名为：copyfile.c

//执行时可用：copyfile filename1 filename2形式的命令行运行，将文件filename1中的内容复制到文件filename2中去。

//file copy program.

#include <stdio.h>

void main(int argc,char \*argv[]) //命令行参数

{

int ch;

FILE \*in,\*out; //定义in和out两个文件类型指针

if(argc!=3) //判断命令行是否正确

{

printf("Error in format,Usage: copyfile filename1 filename2\n");

return; //命令行错，结束程序的执行

}

//按读方式打开由argv[1]指出的文件

if((in=fopen(argv[1],"r"))==NULL)

{

printf("The file <%s> can not be opened.\n",argv[1]);

return; //打开失败，结束程序的执行

}

//成功打开了argv[1]所指文件，再

//按写方式打开由argv[2]指出的文件

if((out=fopen(argv[2],"w"))==NULL)

{

printf("The file %s can not be opened.\n",argv[2]);

return; //打开失败，结束程序的执行

}

//成功打开了argv[2]所指文件

ch=fgetc(in); //从in所指文件的当前指针位置读取一个字符

while(ch!=EOF) //判断刚读取的字符是否是文件结束符

{

fputc(ch,out); //若不是结束符，将它写入out所指文件

ch=fgetc(in); //继续从in所指文件中读取下一个字符

} //完成将in所指文件的内容写入（复制）到out所指文件中

fclose(in); //关闭in所指文件

fclose(out); //关闭out所指文件

}

[复制代码](javascript:void(0);)

【例8.3】按十进制和字符显示文件代码，若遇不可示字符就用井号"#"字符代替之。

[复制代码](javascript:void(0);)

//程序名为：dumpf.c

//执行时可用：dumpf filename1 形式的命令行运行。

// File dump program.

#include <stdio.h>

void main(int argc,char \*argv[])

{

char str[9];

int ch,count,i;

FILE \*fp;

if(argc!=2)

{

printf("Error format,Usage: dumpf filename\n");

return;

}

if((fp=fopen(argv[1],"r"))==NULL)

{

printf("The file %s can not be opened.\n",argv[1]);

return;

}

count=0;

do{

i=0;

//按八进制输出第一列，作为一行八个字节的首地址

printf("%06o: ",count\*8);

do{

// 从打开的文件中读取一个字符

ch=fgetc(fp);

// 按十进制方式输出这个字符的ASCII码

printf("%4d",ch);

// 如果是不可示字符就用"#"字符代替

if(ch<' '||ch>'~') str[i]='#';

// 如果是可示字符，就将它存入数组str以便形成字符串

else str[i]=ch;

// 保证每一行输出八个字符

if(++i==8) break;

}while(ch!=EOF); // 遇到文件尾标志，结束读文件操作

str[i]='\0'; // 在数组str加字符串结束标志

for(;i<8;i++) printf(" "); // 一行不足八个字符用空格填充

printf(" %s\n",str); // 输出字符串

count++; // 准备输出下一行

}while(ch!=EOF); // 直到文件结束

fclose(fp); // 关闭fp所指文件

}

[复制代码](javascript:void(0);)

C. 从文件中读取一个字符串

1． 函数原型  
　　char \*fgets(char \*str,int n,FILE \*fp)  
2． 功能说明  
　　从由fp指出的文件中读取n-1个字符，并把它们存放到由str指出的字符数组中去，最后加上一个字符串结束符'\0'。  
3． 参数说明  
　　str：接收字符串的内存地址，可以是数组名，也可以是指针。  
　　n： 指出要读取字符的个数。  
　　fp：这是个文件指针，指出要从中读取字符的文件。  
4． 返回值

正常返回：返回字符串的内存首地址，即str的值。  
非正常返回：返回一个NULL值，此时应当用feof()或ferror()函数来判别是读取到了文件尾，还是发生了错误。例如，要从"写打开"文件中读取字符串，将  
发生错误而返回一个NULL值。

D. 写一个字符串到文件中去

1． 函数原型

int fputs(char \*str,FILE \*fp)

2． 功能说明  
　　把由str指出的字符串写入到fp所指的文件中去。  
3． 参数说明  
　　str：指出要写到文件中去的字符串。  
　　fp：这是个文件指针，指出字符串要写入其中的文件。  
4． 返回值  
　　正常返回： 写入文件的字符个数，即字符串的长度。  
　　非正常返回：返回一个NULL值，此时应当用feof()或ferror()函数来判别是读取到了文件尾，还是发生了错误。例如，要往一个"读打开" 文件中写字符串时，  
会发生错误而返回一个NULL值。

5．实例

【例8.4】以下程序将一个文件的内容附加到另一个文件中去。

[复制代码](javascript:void(0);)

//程序名：linkfile.c

//执行时可用：linkfile filename1 filename2形式的命令行运行，将文件filename2的内容附加在文件filename1之后。

// file linked program.

#include <stdio.h>

#define SIZE 512

void main(int argc,char \*argv[])

{

char buffer[SIZE];

FILE \*fp1,\*fp2;

if(argc!=3)

{

printf("Usage: linkfile filename1 filename2\n");

return;

}

// 按追加方式打开argv[1] 所指文件

if((fp1=fopen(argv[1],"a"))==NULL)

{

printf("The file %s can not be opened.\n",argv[1]);

return;

}

if((fp2=fopen(argv[2],"r"))==NULL)

{

printf("The file %s can not be opened.\n",argv[2]);

return;

}

// 读入一行立即写出，直到文件结束

while(fgets(buffer,SIZE,fp1)!=NULL)

printf("%s\n",buffer);

while(fgets(buffer,SIZE,fp2)!=NULL)

fputs(buffer,fp1);

fclose(fp1);

fclose(fp2);

if((fp1=fopen(argv[1],"r"))==NULL)

{

printf("The file %s can not be opened.\n",argv[1]);

return;

}

while(fgets(buffer,SIZE,fp1)!=NULL)

printf("%s\n",buffer);

fclose(fp1);

}

[复制代码](javascript:void(0);)

E. 往文件中写格式化数据

1．函数原型

int fprintf(FILE \*fp,char \*format,arg\_list)

2．功能说明  
　　将变量表列（arg\_list）中的数据，按照format指出的格式，写入由fp指定的文件。fprintf()函数与printf()函数的功能相同，只是printf()函数是将数据写入屏幕文件（stdout）。  
3．参数说明  
　　fp：这是个文件指针，指出要将数据写入的文件。  
　　format：这是个指向字符串的字符指针，字符串中含有要写出数据的格式，所以该字符串成为格式串。格式串描述的规则与printf()函数中的格式串相同。  
arg\_list：是要写入文件的变量表列，各变量之间用逗号分隔。  
4．返回值  
　　无。  
5． 实例

【8.5】下列程序的执行文件为display.exe，执行时键入命令行：  
　　　display [-i][-s] filename  
下面的表格列出了命令行参数的含义及其功能：

[复制代码](javascript:void(0);)

//存储文件名：save.txt

//程序代码如下：

// file display program.

#include <stdio.h>

void main()

{

char name[10];

int nAge,nClass;

long number;

FILE \*fp;

if((fp=fopen("student.txt","w"))==NULL)

{

printf("The file %s can not be opened.\n","student.txt");

return;

}

fscanf(stdin,"%s %d %d %ld",name,&nClass,&nAge,&number);

fprintf(fp,"%s %5d %4d %8ld",name,nClass,nAge,number);

fclose(fp);

if((fp=fopen("student.txt","r"))==NULL)

{

printf("The file %s can not be opened.\n","student.txt");

return;

}

fscanf(fp,"%s %d %d %ld",name,&nClass,&nAge,&number);

printf("name nClass nAge number\n");

fprintf(stdout,"%-10s%-8d%-6d%-8ld\n",name,nClass,nAge,number);

fclose(fp);

}

[复制代码](javascript:void(0);)

G. 以二进制形式读取文件中的数据

1． 函数原型

int fread(void \*buffer,unsigned sife,unsigned count,FILE \*fp)

2． 功能说明  
　　从由fp指定的文件中，按二进制形式将sife\*count个数据读到由buffer指出的数据区中。  
3． 参数说明  
buffer：这是一个void型指针，指出要将读入数据存放在其中的存储区首地址。  
sife：指出一个数据块的字节数，即一个数据块的大小尺寸。  
count：指出一次读入多少个数据块（sife）。  
fp：这是个文件指针，指出要从其中读出数据的文件。  
4．返回值  
　　正常返回：实际读取数据块的个数，即count。  
　　异常返回：如果文件中剩下的数据块个数少于参数中count指出的个数，或者发生了错误，返回0值。此时可以用feof()和ferror()来判定到底出现了什么  
情况。

H. 以二进制形式写数据到文件中去

1． 函数原型

int fwrite(void \*buffer,unsigned sife,unsigned count,FILE \*fp)

2． 功能说明  
　　按二进制形式，将由buffer指定的数据缓冲区内的sife\*count个数据写入由fp指定的文件中去。  
3． 参数说明  
buffer：这是一个void型指针，指出要将其中数据输出到文件的缓冲区首地址。  
sife：指出一个数据块的字节数，即一个数据块的大小尺寸。  
count：一次输出多少个数据块（sife）。  
fp：这是个文件指针，指出要从其中读出数据的文件。  
4．返回值  
　　正常返回：实际输出数据块的个数，即count。  
　　异常返回：返回0值，表示输出结束或发生了错误。  
5．实例

【例8.7】

[复制代码](javascript:void(0);)

#include <stdio.h>

#define SIZE 4

struct worker

{ int number;

char name[20];

int age;

};

void main()

{

struct worker wk;

int n;

FILE \*in,\*out;

if((in=fopen("file1.txt","rb"))==NULL)

{

printf("The file %s can not be opened.\n","file1.txt");

return;

}

if((out=fopen("file2.txt","wb"))==NULL)

{

printf("The file %s can not be opened.\n","file2.txt");

return;

}

while(fread(&wk,sizeof(struct worker),1,in)==1)

fwrite(&wk,sizeof(struct worker),1,out);

fclose(in);

fclose(out);

}

[复制代码](javascript:void(0);)

I. 以二进制形式读取一个整数

1． 函数原型

int getw(FILE \*fp)

2． 功能说明  
　　从由fp指定的文件中，以二进制形式读取一个整数。  
3． 参数说明  
　　fp：是文件指针。  
4． 返回值  
　　正常返回：所读取整数的值。  
　　异常返回：返回EOF，即-1。由于读取的整数值有可能是-1，所以必须用feof()或ferror()来判断是到了文件结束，还是出现了一个出错。  
5． 实例

【例8.8】

[复制代码](javascript:void(0);)

#include <stdio.h>

void main(int argc,char \*argv[])

{

int i,sum=0;

FILE \*fp;

if(argc!=2)

{

printf("Command error,Usage: readfile filename\n");

exit(1);

}

if(!(fp=fopen(argv[1],"rb")))

{

printf("The file %s can not be opened.\n",argv[1]);

exit(1);

}

for(i=1;i<=10;i++) sum+=getw(fp);

printf("The sum is %d\n",sum);

fclose(fp);

}

[复制代码](javascript:void(0);)

J. 以二进制形式存贮一个整数

1．函数原型

int putw(int n,FILE \*fp)

2． 功能说明  
　以二进制形式把由变量n指出的整数值存放到由fp指定的文件中。  
3． 参数说明  
　n：要存入文件的整数。  
　fp：是文件指针。  
4． 返回值  
　正常返回：所输出的整数值。  
　异常返回：返回EOF，即-1。由于输出的整数值有可能是-1，所以必须用feof()或ferror()来判断是到了文件结束，还是出现了一个出错。  
5． 实例

【例8.9】

[复制代码](javascript:void(0);)

#include <stdio.h>

void main(int argc,char \*argv[])

{

int i;

FILE \*fp;

if(argc!=2)

{

printf("Command error,Usage: writefile filename\n");

return;

}

if(!(fp=fopen(argv[1],"wb")))

{

printf("The file %s can not be opened.\n",argv[1]);

return;

}

for(i=1;i<=10;i++) printf("%d\n", putw(i,fp));

fclose(fp);

}

[复制代码](javascript:void(0);)

**\* 文件状态检查**

A. 文件结束  
（1） 函数原型

int feof(FILE \*fp)

（2） 功能说明  
　　　该函数用来判断文件是否结束。  
（3） 参数说明  
　　　fp：文件指针。  
（4） 返回值  
　　　0：假值，表示文件未结束。  
　　　1：真值，表示文件结束。  
（5） 实例

【例8.10】

[复制代码](javascript:void(0);)

#include <stdio.h>

void main(int argc,char \*argv[])

{

FILE \*in,\*out;

char ch;

if(argc!=3)

{

printf("Usage: copyfile filename1 filename2\n");

return;

}

if((in=fopen(argv[1],"rb"))==NULL)

{

printf("The file %s can not be opened.\n",argv[1]);

return;

}

if((out=fopen(argv[2],"wb"))==NULL)

{

printf("The file %s can not be opened.\n",argv[2]);

return;

}

while(!feof(in))

{

ch=fgetc(in);

if(ferror(in))

{

printf("read error!\n");

clearerr(in);

}

else

{

fputc(ch,out);

if(ferror(out))

{

printf("write error!\n");

clearerr(out);

}

}

}

fclose(in);

fclose(out);

}

[复制代码](javascript:void(0);)

B. 文件读/写出错  
（1） 函数原型

int ferror(FILE \*fp)

（2） 功能说明  
　　　检查由fp指定的文件在读写时是否出错。  
（3） 参数说明  
　　　fp：文件指针。  
（4） 返回值  
　　　0：假值，表示无错误。  
　　　1：真值，表示出错。

C. 清除文件错误标志

（1） 函数原型

void clearerr(FILE \*fp)

（2） 功能说明  
　　　清除由fp指定文件的错误标志。  
（3） 参数说明  
　　　fp：文件指针。  
（4） 返回值  
　　　无。  
（5） 实例

【例8.12】

[复制代码](javascript:void(0);)

#include <stdio.h>

void main(int argc,char \*argv[])

{

FILE \*in,\*out;

char ch;

if(argc!=3)

{

printf("Usage: copyfile filename1 filename2\n");

return;

}

if((in=fopen(argv[1],"rb"))==NULL)

{

printf("The file %s can not be opened.\n",argv[1]);

return;

}

if((out=fopen(argv[2],"wb"))==NULL)

{

printf("The file %s can not be opened.\n",argv[2]);

return;

}

while(!feof(in))

{

ch=fgetc(in);

if(ferror(in))

{

printf("read error!\n");

clearerr(in);

}

else

{

fputc(ch,out);

if(ferror(out))

{

printf("write error!\n");

clearerr(out);

}

}

}

fclose(in);

fclose(out);

}

[复制代码](javascript:void(0);)

D. 了解文件指针的当前位置  
（1） 函数原型

long ftell(FILE \*fp)

（2） 功能说明  
　　　取得由fp指定文件的当前读/写位置，该位置值用相对于文件开头的位移量来表示。  
（3） 参数说明  
　　　fp：文件指针。  
（4） 返回值  
　　　正常返回：位移量（这是个长整数）。  
　　　异常返回：-1，表示出错。  
（5） 实例

**\* 文件定位**

A. 反绕  
（1） 函数原型

void rewind(FILE \*fp)

（2） 功能说明  
　　　使由文件指针fp指定的文件的位置指针重新指向文件的开头位置。  
（3） 参数说明  
　　　fp：文件指针。  
（4） 返回值  
　　　无。  
（5） 实例

【例8.14】

[复制代码](javascript:void(0);)

#include <stdio.h>

void main()

{

FILE \*in,\*out;

in=fopen("filename1","r");

out=fopen("filename2","w");

while(!feof(in)) fputc(fgetc(in),out);

rewind(out);

while(!feof(in)) putchar(fgetc(in));

fclose(in);

fclose(out);

}

[复制代码](javascript:void(0);)

B. 随机定位  
（1） 函数原型

int fseek(FILE \*fp,long offset,int base)

（2） 功能说明  
　　　使文件指针fp移到基于base的相对位置offset处。  
（3）参数说明  
　　　fp：文件指针。  
　　　offset：相对base的字节位移量。这是个长整数，用以支持大于64KB的文件。  
　　　base：文件位置指针移动的基准位置，是计算文件位置指针位移的基点。ANSI C定义了base的可能取值，以及这些取值的符号常量。

（4）返回值

　　正常返回：当前指针位置。  
　　异常返回：-1，表示定位操作出错。

（5）实例  
【例8.15】

[复制代码](javascript:void(0);)

#include <stdio.h>

#include <string.h>

struct std\_type

{

int num;

char name[20];

int age;

char class;

}stud;

int cstufile()

{

int i;

FILE \*fp;

if((fp=fopen("stufile","wb"))==NULL)

{

printf("The file can't be opened for write.\n");

return 0;

}

for(i=1;i<=100;i++)

{

stud.num=i;

strcpy(stud.name,"aaaa");

stud.age=17;

stud.class='8';

fwrite(&stud,sizeof(struct std\_type),1,fp);

}

fclose(fp);

return 1;

}

void main()

{

int n;

FILE \*fp;

if(cstufile()==0) return;

if((fp=fopen("stufile","rb"))==NULL)

{

printf("The file can not be opened.\n");

return;

}

for(n=0;n<100;n+=2)

{

fseek(fp,n\*sizeof(struct std\_type),SEEK\_SET);

fread(&stud,sizeof(struct std\_type),1,fp);

printf("%10d%20s%10d%4c\n",stud.num,stud.name,stud.age,stud.class);

}

fclose(fp);

}

[复制代码](javascript:void(0);)

**\* 关于exit()函数**  
1． 函数原型

void exit(int status)

2． 功能说明  
　　exit()函数使程序立即终止执行，同时将缓冲区中剩余的数据输出并关闭所有已经打开的文件。  
3． 参数说明  
　　status：为0值表示程序正常终止，为非0值表示一个定义错误。  
4． 返回值  
　　无。

**\* 关于feof()函数**  
1． 函数原型

int feof(FILE \*fp)

2． 功能说明  
　 　在文本文件（ASCII文件）中可以用值为-1的符号常量EOF来作为文件的结束符。但是在二进制文件中-1往往可能是一个有意义的数据，因此不能用它 来作为文件的结束标志。为了能有效判别文件是否结束，ANSI C提供了标准函数feof()，用来识别文件是否结束。  
3． 参数说明  
　　fp：文件指针。  
4． 返回值  
　　返回为非0值：已到文件尾。  
　　返回为0值：表示还未到文件尾。