流的刷新

int fflush(FILE \*fp);

成功时返回0；出错时返回EOF

将流缓冲区中的数据写入实际的文件

Linux下只能刷新输出缓冲区,输入缓冲区丢弃

如果输出到屏幕使用fflush(stdout)

流的定位：

long ftell(FILE \*stream);

long fseek(FILE \*stream, long offset, int whence);

void rewind(FILE \*stream);

fseek 参数whence参数：SEEK\_SET/SEEK\_CUR/SEEK\_END

SEEK\_SET 从距文件开头 offset 位移量为新的读写位置

SEEK\_CUR：以目前的读写位置往后增加 offset 个位移量

SEEK\_END：将读写位置指向文件尾后再增加 offset 个位移量

offset参数：偏移量，可正可负

注意事项：

1.文件的打开使用a模式 fseek无效

2.rewind(fp) 相当于 fseek(fp,0,SEEK\_SET);

3.这三个函数只适用2G以下的文件

编译告警错误：

ffseek\_t.c:13:11: warning: format ‘%d’ expects argument of type ‘int’, but argument 2 has type ‘long int’ [-Wformat=]

printf("current fp=%d\n",ftell(fp));

表示参数类型不匹配

格式化输出

int fprintf(FILE \*stream, const char \*fmt, …);

int sprintf(char \*s, const char \*fmt, …);

成功时返回输出的字符个数；出错时返回EOF

格式化输入

int fscanf(FILE \*stream, const char \*format, ...);

int sscanf(const char \*str, const char \*format, ...);

重点掌握sprintf 和sscanf

**编译告警：**

wsystime.c:16:13: warning: format ‘%d’ expects argument of type ‘int’, but argument 2 has type ‘time\_t {aka long int}’ [-Wformat=]

printf("ctime=%d\n",ctime);

表示类型不匹配 期望的是int但是参数传的是time\_t

解决办法：在参数前加强制类型转换

**标准IO练习**

time()用来获取系统时间(秒数)

time\_t time(time\_t \*seconds) 1970.1.1 0:0:0

localtime()将系统时间转换成本地时间

struct tm \*localtime(const time\_t \*timer)

struct tm {

int tm\_sec; /\* 秒，范围从 0 到 59 \*/

int tm\_min; /\* 分，范围从 0 到 59 \*/

int tm\_hour; /\* 小时，范围从 0 到 23 \*/

int tm\_mday; /\* 一月中的第几天，范围从 1 到 31 \*/

int tm\_mon; /\* 月份，范围从 0 到 11 \*/

int tm\_year; /\* 自 1900 起的年数 \*/

int tm\_wday; /\* 一周中的第几天，范围从 0 到 6 \*/

int tm\_yday; /\* 一年中的第几天，范围从 0 到 365 \*/

int tm\_isdst; /\* 夏令时 \*/

};

注意：

int tm\_mon; 获取的值要加1是正确的月份

int tm\_year; 获取的值加1900是正确的年份

获取文件内的所有行数量：

while(fgets(buf,32,fp)!=NULL){

if(buf[strlen(buf)-1] =='\n'){ //注意判断是否是一行结束

linecount++;

}

}

写完文件记得fflush ，写到磁盘里面去。

标准IO磁盘文件的缓冲区一般为4096

注意和标准输出的全缓冲区别，标准输出是1024