

标准I/O (五)

主讲：大海老师

课程目标

流的刷新（熟练）

流的定位（熟练）

检测流结束和出错（了解）

小结

标准I/O - 刷新流

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

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

→ fp

刷新流三时机

缓冲区满 / \n

流关闭

fflush函数

(行)
(块)

(行)

➤ 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

✓ 定位流 – ftell/fseek/rewind

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

```
long ftell(FILE *stream);
```

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

```
void rewind(FILE *stream);
```

→ stream 接

→ stream whence offset

- ✓ > ftell() 成功时返回流的当前读写位置，出错时返回EOF
- ✓ > fseek()定位一个流，成功时返回0，出错时返回EOF
- ✓ > 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

(视有例，但此无，觉练完这里，再找视中的)

定位流 – ftell/fseek/rewind

- 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

- 打开a模式 fseek无效

~~✓~~ rewind()将流定位到文件开始位置

文件

- 读写流时, 当前读写位置自动后移

fopen

- 示例一(在文件末尾追加字符 't') →

fseek

```
FILE *fp = fopen( "test.txt" , "r+" );
```

fputc

```
fseek(fp, 0, SEEK_END);
```

```
fputc( 't' , fp);
```

定位流 – ftell/fseek/rewind

示例二(获取文件长度)

```
FILE *fp;  
if ((fp = fopen( "test.txt" , "r+" )) == NULL) {  
    perror( "fopen" );  
    return -1;  
}
```

文件

fopen

→

到尾部

```
fseek(fp, 0, SEEK_END);
```

让当前位置

```
printf( "length is %d\n" , ftell(fp));
```

标准I/O – 判断流是否出错和结束

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

```
int ferror(FILE *stream);
```

→ stream 判

```
int feof(FILE *stream);
```

→ stream 判

✓ ➤ ferror()返回1表示流出错；否则返回0

✓ ➤ feof()返回1表示文件已到末尾；否则返回0 (此处返回真值，非成功)

标准I/O – 小结

fflush

ftell

fseek

