

什么是文件IO

- 在Linux/Unix系统下，有着一切皆文件的定义
- 所谓文件IO就是指文件的输入和输出

修改文件的正常流程

- Windows如何手动修改文件，比如写一个word文档
- 打开/创建文档，编辑文档，保存文档，关闭文档
- Linux呢？

函数介绍

- 文件I/O介绍
- open()
- read()
- write()
- close()
- lseek

调用open()

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

int open(const char *pathname, int flags);
int open(const char *pathname, int flags, mode_t mode);
```

- 一看功能
 - 二看参数
 - 三看返回值
- flags:
 - O_RDONLY 只读打开
 - O_WRONLY 只写打开
 - O_RDWR 可读可写打开
 - O_TRUNC 文件存在，打开文件先删除文件中的数据
 - O_CREAT 创建，要用第三个参数设置其权限

调用close()

```
#include <unistd.h>

int close(int fd);
```

- 调用成功返回0，出错返回-1
- 当一个进程终止的时候，该进程打开的所有文件都有内核自动关闭

调用read()

- 从一个已经打开的可读文件中读取数据

```
#include <unistd.h>

ssize_t read(int fd, void *buf, size_t count);
```

- fd:
 - 文件描述符
 - buf:
 - 读出来的内容
 - count
 - 一次最多读count个数据，理解为大小即可
- read()
 - 调用成功返回0，并达到文件末尾
 - 失败返回-1

调用write()

- 向一个已经打开的可写文件中写入数据。

```
#include <unistd.h>

ssize_t write(int fd, const void *buf, size_t count);
```

- 成功返回已写的字节数，失败返回-1

调用lseek()

- 可以显示一个已打开文件的定位

```
#include <sys/types.h>
#include <unistd.h>

off_t lseek(int fd, off_t offset, int whence);
```

- offset:偏移量，单位是字节的数量，可正可负(向前移，向后移)
- whence:当前位置，
 - SEEK_SET 当前位置为文件的开头，新位置为大小的偏移量
 - SEEK_CUR 当前位置为文件指针的位置，新位置+偏移量
 - SEEK_END 当前位置为文件的结尾，新位置+偏移量
- lseek()
- 调用成功返回文件当前位移
- 失败返回-1

文件操作原理简述

实现cp指令

- 1.C语言参数
- 2.思路：
 - a.打开src.c
 - b.读src.c到buf
 - c.打开创建des.c
 - d.buf写入到des.c
 - e.close两个文件