**一、文件IO**

文件io是linux内核对外提供的接口。用户空间的程序通过调用这些系统调用与内核交互，最终完成对文件（包括普通文件、设备文件、网络套接字等）的读写操作。

文件io的操作的对象为“文件描述符” ，这些系统调用函数都是围绕文件描述进行的。

**1.1 理论知识**

**1.1.1 文件描述符**

文件io通过 文件描述符 标识文件，

文件描述符 是一个非负整数 0、1、2 ...

其中0,1,2对应的是标准输入、标准输出、标准错误输出

|  |  |
| --- | --- |
| 0 | 标准输入 |
| 1 | 标准输出 |
| 2 | 标准错误输出 |

**1.1.2 特点**

文件io是操作系统对外提供的底层接口，依赖操作系统环境

不带缓存

**1.1.3 POSIX**

POSIX（可移植操作系统接口）标准，由于文件io函数遵循该标准，

文件io相关函数，在所有类 Unix 系统（Linux、macOS、FreeBSD、Solaris 等）均可以使用。

**1.2 打开方式**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| flags | 说明 | 约束条件 |  | 补充 |
| O\_RDONLY | 只读权限 | 文件必须存在 | 主标志 | 三个参数互斥 |
| O\_WRONLY | 只写权限 | 文件必须存在 |
| O\_RDWR | 读写权限 | 文件必须存在 |
| O\_TRUNC | 清空 |  | 附属标志 |  |
| O\_APPEND | 追加(定位到文件末尾) |  |  |
| O\_CREAT | 创建 | 不存在则创建，需要加上权限(mode) |  |
| O\_EXCL | 文件是否存在 | 文件存在则报错返回 |  |

练习：

1.文件拷贝

2.尝试通过文件io实现 标准io 中的 “r” "w" "a"

**1.2 相关函数**

open/close //打开与关闭

read/write //读与写

lseek //定位

**二、函数备注**

//文件的打开与关闭

int open(const char \*pathname, int flags);

int open(const char \*pathname, int flags, mode\_t mode);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Description： 打开文件，关联到文件描述符(整形)

\* Input: pathname：路径名

flags：打开方式（O\_RDONLY,O\_WRONLY,O\_RDWR,O\_CREAT...)

mode：权限(当创建文件时候需要给权限 例如：0664)

\* Return: 成功返回文件描述符， 出错返回-1

\* Others:

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int close(int fd)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Description： 关闭文件

\* Input: fd：文件描述符

\* Return: 成功返0， 出错返回-1

\* Others:

#include <unistd.h>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//文件的读与写

ssize\_t read(int fd, void \*buf, size\_t count);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Description： 读文件

\* Input: fd：文件描述符，buf:内存地址(保存读取的内容)，count：希望读取的字节数

\* Return: 成功返回具体读到多少字节， 出错返回-1，读到末尾返回0

\* Others:

#include <unistd.h>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

ssize\_t write(int fd, const void \*buf, size\_t count);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Description： 写文件

\* Input: fd：文件描述符，buf：待写入数据的地址，count：希望写入的字节数

\* Return: 成功返回具体写入多少字节， 出错返回-1

\* Others:

#include <unistd.h>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//文件的定位

off\_t lseek(int fd, off\_t offset, int whence);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Description： 文件的定位

\* Input: fd：文件描述符，offset：偏移量，whence：基准值

SEEK\_SET:文件头

SEEK\_CUR：当前位置

SEEK\_END:文件尾

\* Return: 当调用成功时则返回目前的读写位置, 也就是距离文件开头多少个字节，

出错返回-1

\* Others:

#include <sys/types.h>

#include <unistd.h>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

|  |  |  |
| --- | --- | --- |
|  | 标准io | 文件io |
| 特点 | C库函数，带缓存 | 系统调用，没有缓存 |
| 操作对象 | FILE \* (流) | int （文件描述符） |
| 函数 | fopen、fclose、freopen  fgetc、fputc  fgets、fputs  fread、fwrite  fprintf、fscanf  fseek、ftell、rewind  fflush  feof | open、close  read、write  lseek |
| 移植性 | 主流操作系统 | 类Unix 系统 |