

系统函数 IO

★ 系统调用函数都必须考虑返回值

open函数

函数原型:
int open(const char *pathname, int flags);
int open(const char *pathname, int flags, mode_t mode);

pathname: 文件的相对或绝对路径

flags: 打开方式

必选项

O_RDONLY 只读打开

O_WRONLY 只写打开

O_RDWR 可读可写打开

互斥

可选项

O_APPEND 表示追加。

如果文件已有内容，这次打开文件所写的数据附加到文件的末尾而不覆盖原来的内容。

★ O_CREAT 若此文件不存在则创建它。

使用此选项时需要提供第三个参数mode，表示该文件的访问权限。

文件权限由open的mode参数和当前进程的umask掩码共同决定

0777 & (~0002) = 0775
111111111
111111101
&
111111101

O_EXCL 如果同时指定了O_CREAT，并且文件已存在，则出错返回。

O_TRUNC 如果文件已存在，则将其长度截断（Truncate）为0字节。

O_NONBLOCK 设置文件为非阻塞状态

头文件: fcntl.h

常见错误

1. 打开文件不存在

2. 以写方式打开只读文件(打开文件没有对应权限)

3. 以只写方式打开目录

close函数

参数: open函数的返回值

返回值

0 --> 正常关闭

-1 --> 关闭出现错误

read函数

从打开的设备或文件中读取数据。

返回值

-1 --> 错误

>0 --> 读出的字节数

=0 --> 文件读取完毕

函数原型: ssize_t read(int fd, void *buf, size_t count);

fd: 文件描述符

buf: 数据缓冲区

count: 请求读取的字节数

write函数

向打开的设备或文件中写数据。

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

fd: 文件描述符

buf: 需要输出的缓冲区

count: 最大输出字节数

返回值

-1 --> 失败

>=0 --> 写入文件的字节数

lseek函数

函数作用 修改文件偏移量(读写位置)

原型 off_t lseek(int fd, off_t offset, int whence)

int fd --> 文件描述符

off_t offset --> 偏移量

int whence --> 偏移位置

SEEK_SET - 从文件头向后偏移

SEEK_CUR - 从当前位置向后偏移

SEEK_END - 从文件尾部向后偏移

返回值

较文件起始位置向后的偏移量

允许超过文件结尾设置偏移量，文件会因此被拓展。

失败返回 -1

应用

拓展文件空间

获取文件长度 lseek(fd, 0, SEEK_END);

返回值即为文件长度