

文件操作

★ stat函数

获取文件属性 (从inode上获取)

返回值

成功: 0

失败: -1

```
struct stat {
    dev_t    st_dev;    //文件的设备编号
    ino_t    st_ino;    //节点
    mode_t    st_mode;    //文件的类型和存取的权限
    nlink_t    st_nlink;    //连到该文件的硬连接数目, 刚
                        建立的文件值为1
    uid_t    st_uid;    //用户ID
    gid_t    st_gid;    //组ID
    dev_t    st_rdev;    //(设备类型)若此文件为设备文
                        件, 则为其设备编号
    off_t    st_size;    //文件字节数(文件大小)
    blksize_t st_blksize; //块大小(文件系统的I/O 缓冲
                        区大小)
    blkcnt_t st_blocks;  //块数
    time_t    st_atime;  //最后一次访问时间
    time_t    st_mtime;  //最后一次修改时间
    time_t    st_ctime;  //最后一次改变时间(指属性)
};
```

文件属性

特性

能够穿透 (跟踪) 符号链接

vi 编译器

lstat 函数

特性

不穿透 (跟踪) 符号链接

ls -l

rm

access 函数

作用

测试指定文件是否拥有某种权限

原型

int access(const char \*pathname, int mode);

参数

pathname --> 文件名

mode --> 权限类别

R\_OK 是否有读权限

W\_OK 是否有写权限

X\_OK 是否有执行权限

F\_OK 测试一个文件是否存在

返回值

0 --> 所有欲查核的权限都通过了检查

-1 --> 有权限被禁止

chmod函数

作用

改变文件的权限

原型

int chmod( const char \*filename, int pmode );

参数

filename --> 文件名

pmode --> 权限

必须是一个8进制数

返回值

0 --> 改变成功

-1 --> 失败

chown函数

作用

改变文件的所有者

原型

int chown(const char \*path, uid\_t owner, gid\_t group);

返回值

0 --> 成功

-1 --> 失败

★ truncate函数

作用

将参数path 指定的文件大小改为参数length 指定的大小。如果原来的文件大小比参数length大, 则超过的部分会被删去。

原型

int truncate(const char \*path, off\_t length);

参数

path --> 文件路径

length --> 指定的文件大小

返回值

0 --> 执行成功

-1 --> 执行失败

链接

link函数

作用

创建一个硬链接

原型

int link(const char \*oldpath, const char \*newpath);

symlink 函数

作用: 创建一个软连接

readlink 函数

作用: 读软连接对应的文件名, 不是读内容

unlink 函数

作用

删除一个文件的目录项并减少它的链接数, 若成功则返回0, 否则返回-1, 错误原因存于errno。

如果想通过调用这个函数来成功删除文件, 你就必须拥有这个文件的所属目录的写和执行权限。

使用

1. 如果是符号链接, 删除符号链接

2. 如果是硬链接, 硬链接数减1, 当减为0时, 释放数据块和inode

3. 如果文件硬链接数为0, 但有进程已打开该文件, 并持有文件描述符, 则等该进程关闭该文件时, kernel才真正去删除该文件

利用该特性创建临时文件, 先open或creat创建一个文件, 马上unlink此文件

rename 函数

作用: 文件重命名

头文件: stdio.h

函数原型: int rename(const char \*oldpath, const char \*newpath);