

Linux操作系统编程

# Linux文件权限管理



## ●函数原型

- `int chown( const char *pathname, uid_t owner, gid_t group);`
- `int fchown( int filedes, uid_t owner, gid_t group);`
- `int lchown( const char *pathname, uid_t owner, gid_t group);`

●**函数用途：**更改文件的用户ID和组ID。成功返回0，若出错则返回-1

## ●函数说明：

- 如果两个参数owner或group中的任意一个是-1，则不改变文件所有者或文件所属用户组
- lchown是改变符号链接本身的所有者，而不是该符号链接所指向的文件
- 基于BSD的系统中，只有超级用户才能更改一个文件的所有者
- 非超级用户进程调用,则在成功返回时，该文件的设置用户ID位和设置组ID位会被清除



- **函数原型**: `int link ( const char *pathname, const char *newpath);`
- **函数用途**: 任何一个文件可以有多个目录项指向其索引节点, 创建一个指向现有文件的新目录项
- **函数说明**
  - 此函数创建一个新目录项newpath, 它指向pathname指向的文件。如果newpath已经存在, 则返回出错
  - 创建新目录项以及增加连接计数是一个原子操作。
  - 大多数的系统规定, 只有超级用户可以创建指向一个目录的新连接, 目的是在文件系统中形成循环。



## ●函数原型及头文件

- `int unlink ( const char *pathname);` ([unistd.h](#))
- `int remove( const char *pathname);` ([stdio.h](#))

## ●函数用途：

- unlink删除目录项，并将由pathname所引用文件的链接计数减1，清空这个文件使用的可用的系统资源
- remove删除一个目录中的一个或多个文件或目录，也可以将某个目录及其下的所有文件及子目录均删除

## ●共同点及差异

- unlink是系统调用，而remove是库函数
- remove的参数为普通文件时等价于unlink



- **函数原型**: `int rename ( const char * oldname, const char * newname);`
- **函数用途**: 更名文件或目录
- **函数说明**
  - 如果oldname是一个文件而不是目录, 那么为该文件更名
  - 如果oldname是一个目录, 那么为该目录更名
  - 如果oldname和newname引用同一文件, 则函数不做任何更改而成功返回
  - 应对包含两个文件的目录具有写和执行许可权



- 每个文件有三个时间字段

字段	说明	例子	ls(1)选项
at_atime	文件数据的最后访问时间	read	-u
at_mtime	文件数据的最后修改时间	write	默认
st_ctime	索引节点状态的最后更改时间	chmod	-c

- 对文件进行一次读操作，它的访问时间就会改变

- 修改时间是文件内容最后一次被修改的时间

- 状态时间是该文件索引节点最后一次被修改的时间，影响索引节点的操作：

- 更改文件访问权限
- 更改文件用户ID
- 更改文件链接数



- **函数原型：** `int utime( const char *pathname, const struct utimbuf *times);`
- **函数用途：** 设置存取和修改文件的时间
- **函数说明：** 此函数的操作以及执行它所要求的优先权取决于time参数是否是NULL
  - 如果times是一个NULL，则存取时间和修改时间两者都设置为当前时间，但必须满足下面二者之一：（1）进程的有效用户ID必须等于该文件的所有者（2）进程对该文件有写许可权
  - 如果times是非空指针，则存取时间和修改时间被设置为times所指向的时间，此时进程的有效用户ID必须等于该文件的所有者ID，或者进程必须是超级用户的进程。



- **函数原型:** `int mkdir(const char *pathname, mode_t mode);`
- **函数用途:** 创建一个新的空目录, 其中的. 和 .. 两个目录项由系统自动创建。
- **函数说明:**
  - 如果在目录名的前面没有加任何路径名, 则在当前目录下创建由pathname指定的目录
  - 如果给出了一个已经存在的路径, 将会在该目录下创建一个指定的目录
  - 在创建目录时, 应保证新建的目录与它所在目录下的文件没有重名
  - 在创建文件时, 不要把所有的文件都存放在主目录中, 可以创建子目录, 来更有效地组织文件



- **函数原型:** `int rmdir( const char *pathname);`

- **函数用途:** 删除一个空目录

- **函数说明:**

- 如果此调用使目录的连接计数成为0，并且也没有其他进程打开此目录，则释放由此目录占用的空间
- 如果连接计数为0，但有进程打开了此目录，则在此函数返回前，删除. 和 .. 项。在此目录中不能再创建新文件。但是在最后一个进程关闭之前并不释放此目录