

Linux操作系统编程

目录操作

- **头文件：** `sys/stat.h`
- **函数声明：** `int mkdir(const char *pathname, mode_t mode);`
- **函数说明：**
以mode方式创建一个以参数pathname命名的目录，mode定义新创建目录的权限
- **返回值：**
若目录创建成功，则返回0；否则返回-1，并将错误记录到全局变量errno中
- **注意事项：**
创建775权限的目录，需要先执行umask (0) ，然后再调用mkdir函数

新创建目录的权限

字段	权限	含义
S_IRWXU	00700	文件所有者拥有读写执行的权限
S_IRUSR(S_IREAD)	00400	文件所有者拥有可读权限
S_IWUSR(S_IWRITE)	00200	文件所有者拥有可写权限
S_IXUSR(S_IEXEC)	00100	文件所有者拥有执行权限
S_IRWXG	00070	文件用户组拥有读写执行的权限
S_IRGRP	00040	文件用户组拥有可读权限
S_IWGRP	00020	文件用户组拥有可写权限
S_IXGRP	00010	文件用户组拥有执行权限
S_IRWXO	00007	其他用户拥有读写执行权限
S_IROTH	00004	其他用户拥有可读权限
S_IWOTH	00002	其他用户拥有可写权限
S_IXOTH	00001	其他用户拥有执行权限

- **头文件：**unistd.h

- **函数声明：** int rmdir(const char *pathname);

- **函数说明：**

删除一个目录，该目录必须是空的

- **返回值：**

若目录删除成功，则返回0；否则返回-1，并将错误记录到全局变量errno中

- **注意事项：**

pathname不能超过255，不能以 “.” 开头。目录没有被其他进程占用。

- **头文件**: `unistd.h`

- **函数声明**: `int chdir(const char *path);`

- **函数说明**:

`chdir`函数用于改变当前工作目录。调用参数是指向目录的指针，调用进程需要有搜索整个目录的权限

- **返回值**:

成功，则返回0；否则返回-1，并将错误记录到全局变量`errno`中

- **错误信息**:

EFAULT: `path`指向了非法地址；**ENAMETOOLNG**:路径过长；**ENOENT**:文件不存在；**ENOMEM**:内核内存不足；**ENOTDIR**:给出路径不是目录；**EACCES**:无访问路径中某个目录的权限；**ELOOP**:解析路径中太多的符号链接；**EIO**:发生I/O错误

- **函数说明：**

- 每个进程都具有一个当前工作目录。在解析相对目录引用时，该目录是搜索路径的开始之处
- 如果调用进程更改了目录，则它只对该进程有效，而不能影响调用它的那个进程
- 在退出程序时，shell还会返回开始时的那个工作目录
- 内核解析参数中的路径名，并确保这个路径名有效。内核定位该目录的索引节点，并检查它的文件类型和权限位，确保目标文件是目录以及进程的所有者可以访问该目录
- 内核用新目标目录的路径名和索引节点替换u区中当前目录路径名和索引节点号

- **头文件：** `sys/types.h` `dirent.h`
- **函数声明：** `void rewinddir(DIR *dir);`
- **函数说明：**
设置参数dir指向的目录流目前的读取位置为原来开头的读取位置
- **返回值：**
无返回值，函数执行失败后会将错误记录到全局变量errno中
- **错误信息：**
EBADF： 表示dir指向的目录流无效

- **头文件：** dirent.h
- **函数声明：** void seekdir(DIR * dir, off_t offset);
- **函数说明：**

设置参数dir指向的目录流目前的读取位置为offset，在调用readdir()时便从此新位置开始读取。参数offset 代表距离目录文件开头的偏移量。

- **返回值：**

无返回值，函数执行失败后会将错误记录到全局变量errno中

- **错误信息：**

EBADF： 表示dir指向的目录流无效

- **头文件：** dirent.h

- **函数声明：** off_t telldir(DIR * dir);

- **函数说明：**

取得目录流dir的读取位置。

- **返回值：**

返回参数dir目录流目前的读取位置. 此返回值代表距离目录文件开头的偏移量，有错误发生时返回-1.

- **错误信息：**

EBADF： 表示dir指向的目录流无效

- **头文件：** dirent.h
- **函数声明：** int scandir(const char *dir, struct dirent **namelist, int (*select) (const struct dirent *), int(*compar) (const struct dirent **, const struct dirent**));
- **函数说明：**

scandir()会扫描参数dir指定的目录文件，经由参数select指定的函数来挑选目录结构至参数namelist数组中，最后再调用参数compar指定的函数来排序namelist 数组中的目录数据。每次从目录文件中读取一个目录结构后便将此结构传给参数select所指的函数，select函数若不想要将此目录结构复制到namelist数组就返回0，若select为空指针则代表选择所有的目录结构。scandir()会调用qsort()来排序数据，参数compar则为qsort()的参数，若是要排列目录名称字母则可使用alphasort()。
- **返回值：** 成功则返回复制到namelist数组中的数据结构数目，有错误发生则返回-1.
- **错误信息：** ENOMEM 核心内存不足

