
文件属性、目录操作

主讲：大海老师

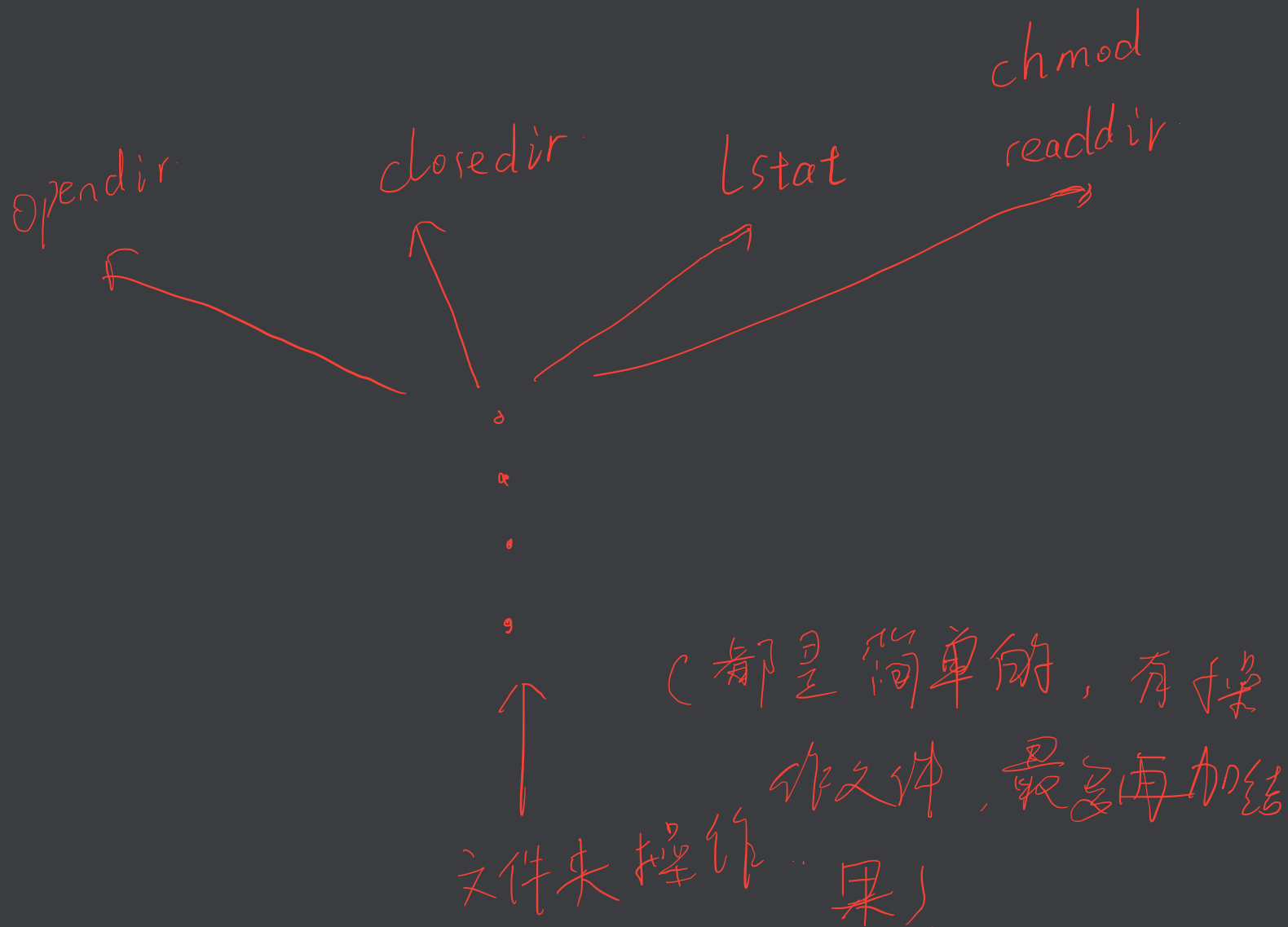
课程目标:

读取目录 (熟练)

修改文件访问权限 (熟练)

获取文件属性 (熟练)

小结



访问目录 – opendir

opendir函数用来打开一个目录文件:

```
#include <dirent.h>
```

```
DIR *opendir(const char *name);
```

DIR *fdopendir(int fd); 使用文件描述符, 要配合open函数使用

- DIR是用来描述一个打开的目录文件的结构体类型
- 成功时返回目录流指针; 出错时返回NULL

→ 路径 模式 接判

访问目录 – readdir

readdir函数用来读取目录流中的内容:

```
#include <dirent.h>
```

```
struct dirent *readdir(DIR *dirp);
```

→ ~~fp~~ size 判

(简化):
(不一样按)
接

- struct dirent是用来描述目录流中一个目录项的结构体类型
- 包含成员char d_name[256] 参考帮助文档
- 成功时返回目录流dirp中下一个目录项;
- 出错或到末尾时时返回NULL

访问目录 – closedir

closedir函数用来关闭一个目录文件:

```
#include <dirent.h>
```

```
int closedir(DIR *dirp);
```

→ fp

。

➤ 成功时返回0; 出错时返回EOF

访问目录 - 示例代码

XF 觉打开本原去那路径上拿对象
或根据之拿

打印指定的目录下所有文件名称

```
int main(int argc, char *argv[]) {  
{
```

```
    DIR *dirp;
```

```
    struct dirent *dp;
```

```
    if (argc < 2) {
```

```
        printf( "Usage : %s <directory>\n" , argv[0]); return -1;
```

```
    }
```

```
    if ((dirp = opendir(argv[1])) == NULL) {
```

```
        perror( "opendir" ); return -1;
```

```
    }
```

```
    while ((dp = readdir(dirp)) != NULL) {
```

```
        printf( "%s\n" , dp->d_name);
```

```
    }
```

```
    closedir(dirp);
```

[argc]

opendir

→ readdir printf
(while)

col)

(re)

修改文件访问权限 – chmod/fchmod

chmod/fchmod函数用来修改文件的访问权限:

```
#include <sys/stat.h>
```

```
int chmod(const char *path, mode_t mode);
```

→ " " 路径 权限

```
int fchmod(int fd, mode_t mode);
```

- 成功时返回0; 出错时返回EOF
- root和文件所有者能修改文件的访问权限

示例: chmod("test.txt", 0666);

→ chmod
(re)

获取文件属性 – stat/lstat/fstat

stat/lstat/fstat函数用来获取文件属性:

```
#include <sys/stat.h>
```

```
int stat(const char *path, struct stat *buf);
```

```
int lstat(const char *path, struct stat *buf);
```

```
int fstat(int fd, struct stat *buf);
```

→ 路径属性

- 成功时返回0; 出错时返回EOF
- 如果path是符号链接stat获取的是目标文件的属性; 而lstat获取的是链接文件的属性

文件属性 – struct stat

struct stat是存放文件属性的结构体类型:

- mode_t st_mode; 类型和访问权限
- uid_t st_uid; 所有者id
- uid_t st_gid; 用户组id
- off_t st_size; 文件大小
- time_t st_mtime; 最后修改时间

Stat 结构体

- 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)
- unsigned long st_blksize; //块大小(文件系统的I/O 缓冲区大小)
- unsigned long 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)
- };

文件类型 – st_mode

通过系统提供的宏来判断文件类型:

S_IFMT	0170000	文件类型的位遮罩
S_ISREG(st_mode)	0100000	是否常规文件
S_ISDIR(st_mode)	0040000	是否目录
S_ISCHR(st_mode)	0020000	是否字符设备
S_ISBLK(st_mode)	0060000	是否块设备
S_ISFIFO(st_mode)	0010000	是否FIFO文件
S_ISLNK(st_mode)	0120000	是否链接文件
S_ISSOCK(st_mode)	0140000	是否SOCKET文件

✓ switch 找就行
(可参考man)

文件访问权限 – st_mode

通过系统提供的宏来获取文件访问权限:

S_IRUSR	00400	bit:8	所有者有读权限
S_IWUSR	00200	7	所有者拥有写权限
S_IXUSR	00100	6	所有者拥有执行权限
S_IRGRP	00040	5	群组拥有读权限
S_IWGRP	00020	4	群组拥有写权限
S_IXGRP	00010	3	群组拥有执行权限
S_IROTH	00004	2	其他用户拥有读权限
S_IWOTH	00002	1	其他用户拥有写权限
S_IXOTH	00001	0	其他用户拥有执行权限

✓ 0 ⇒ 写 '-'
1 ⇒ %3 来
由

练习 - 获取并显示文件属性

以下面格式打印指定文件的主要信息:

```
$ ./a.out test.c
```

```
-rw-r--r-- 317 2014-11-08 test.c
```

- 调用lstat函数获取文件的属性
- 从stat结构体中获取相应信息并输出

(argc) ...
(次)
打开
获取信息
打印
(提到)
就进行

mask switch 是 [-d 2] 种)
for 8
if (switch % 3)
else { (p -)
317 ln 空格
2014 mtime → localtime → .

test.c 空格 \n .

(虽乱点, 但如真理解对, 实用
一段时间后看也能懂的)

小结

opendir

readdir

closedir

lstat/stat

+1900+1 空格02

cd)

cre) ..

