

Linux操作系统编程

# 目录操作

# ls -l功能分析

```
root@xrxy-virtual-machine: ~  
root@xrxy-virtual-machine:~# cd /root/  
root@xrxy-virtual-machine:~# ls -l  
total 64  
-rwxr-xr-x 1 root root 7255 2013-12-31 22:43 a.out  
drwxr-xr-x 2 root root 4096 2012-02-23 10:08 Desktop  
drwxr-xr-x 2 root root 4096 2012-02-23 10:08 Documents  
drwxr-xr-x 2 root root 4096 2012-02-23 10:08 Downloads  
drwxr-xr-x 3 root root 4096 2014-01-01 03:18 etc  
-rwxr-xr-x 1 root root 7759 2014-01-01 02:33 listdirectory  
-rw-r--r-- 1 root root 1985 2014-01-01 03:42 listdirectory.c  
-rw-r--r-- 1 root root 1985 2014-01-01 02:41 listdirectory.c~  
drwxr-xr-x 2 root root 4096 2012-02-23 10:08 Music  
drwxr-xr-x 2 root root 4096 2012-02-23 10:08 Pictures  
drwxr-xr-x 2 root root 4096 2012-02-23 10:08 Public  
drwxr-xr-x 2 root root 4096 2012-02-23 10:08 Templates  
drwxr-xr-x 6 root root 4096 2014-01-01 03:18 usr  
drwxr-xr-x 2 root root 4096 2012-02-23 10:08 Videos  
root@xrxy-virtual-machine:~#
```

列出当前目录的所有文件  
(包括普通文件、目录文  
件、字符特殊文件、套接  
字等)，并显示文件类型、  
访问权限、文件大小等重  
要属性



## 文件属性

所有者 权限	其他用 户权限	文件 所有者	文件大小 (字节)	文件名
drwxr-xr-x	2	root	4096	Documents
drwxr-xr-x	2	root	4096	Downloads
drwxr-xr-x	3	root	4096	etc
-rwxr-xr-x	1	root	7759	listdirectory
-rw-r--r--	1	root	1985	listdirectory.c

文件类型

组用户  
权限

文件硬链  
接数或目  
录子目录  
数

文件所  
有者所  
在组

文件最后修改时间

# 文件类型与文件访问权限

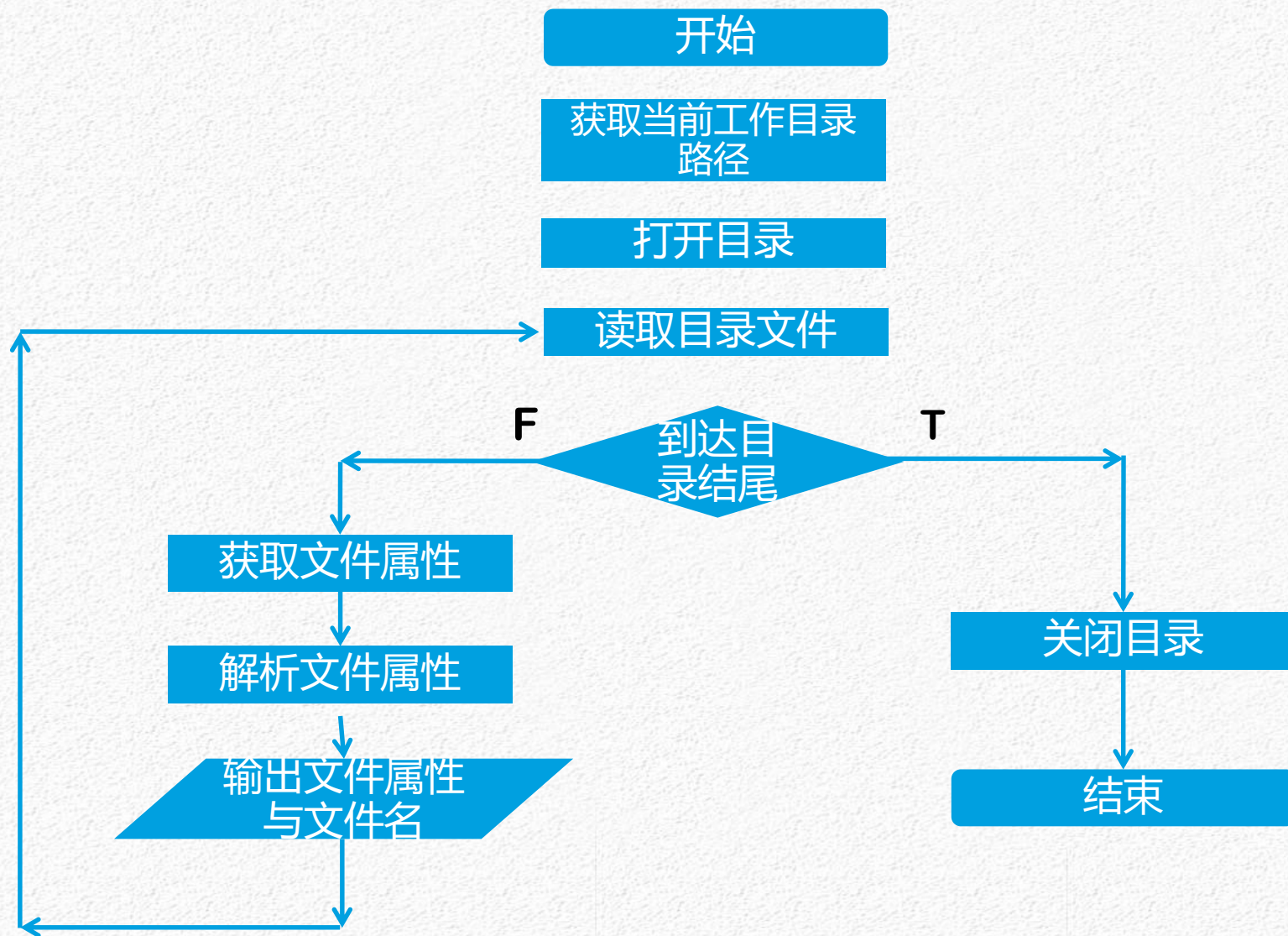
## 文件类型

标识	文件类型
-	普通文件
d	目录文件
c	字符设备文件
b	块设备文件
p	管道或FIFO
l	符号链接
s	套接字

## 文件访问权限

标识	文件访问权限
r	读权限
w	写权限
x	执行权限







- **头文件：** `unistd.h`
- **函数声明：** `char *getcwd(char *buf, size_t size);`

- **函数说明：**

将当前的工作目录绝对路径字符串复制到参数buf所指的缓冲区，参数size为buf缓冲区大小。

- **返回值：**

成功调用返回指向buf的指针，失败返回NULL

- **示例程序：**

```
#define MAX_SIZE 255
char path(MAX_SIZE);
getcwd(path, sizeof(path));
puts(path);
```



- **头文件：** `unistd.h`
- **函数声明：** `char * get_current_dir_name(void);`
- **函数说明：**  
调用后会返回一个字符串指针，指向当前工作目录绝对路径字符串
- **返回值：**  
成功则返回字符串指针，失败则返回空，错误代码存放于`errno`中



```
struct __dirstream
{
    void *__fd;
    char *__data;
    int __entry_data;
    char *__ptr;
    int __entry_ptr;
    size_t __allocation;
    size_t __size;
    __libc_lock_define(, __lock)
}; typedef struct __dirstream DIR;
```



- **头文件**: `sys/types.h` `dirent.h`
- **函数声明**: `DIR *opendir(const char *path);`
- **函数说明**: 打开参数name指定的目录
- **返回值**: 成功则返回DIR形态的目录流, 失败则返回空, 错误代码存放于errno中
- **错误代码**:

EACCESS, 代表权限不足, 也就是对这个目录没有执行权限。EMFILE表示目前同时打开这个文件的进程数目已经到达了系统上限。ENFILE代表已经达到系统可以同时打开的文件数上限。ENOTDIR代表参数name指向的不是一个真正的目录。EOENT表示参数name指向的目录不存在, 或者参数name是一个空的字符串。ENOMEM表示核心内存不足。



- **头文件：** sys/types.h dirent.h
- **函数声明：** int closedir(DIR \*dir);

- **函数说明：**

关闭参数dir指定的目录

- **返回值：**

关闭成功就返回0，失败的话就返回-1，错误原因存于errno中

- **错误代码：**

EBADF，代表参数dir是一个无效的目录流



## 重要数据结构

```
struct dirent
{
    ino_t d_ino; i节点号
    off_t d_off; 在目录文件中的偏移
    unsigned short d_reclen; 文件名长度
    unsigned char d_type; 文件类型
    char d_name[256];文件名 ★
};
```

- **头文件：** sys/types.h dirent.h
- **函数声明：** struct dirent \*readdir(DIR \*dirp);
- **函数说明：**

读取目标流dir标识的目录

- **返回值：**

每执行一次readdir，这个函数返回指向当前读取目录项结构的指针，有错误发生或读取到目录文件尾的时候就返回NULL

- **错误代码：**

EBADF，代表参数dir是一个无效的目录流



