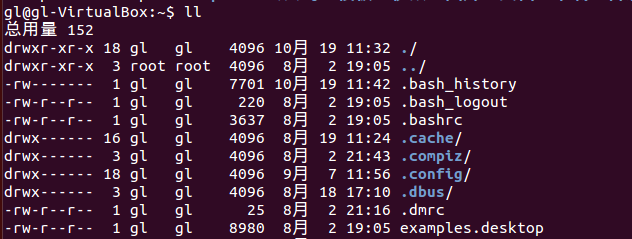
## Linux文件系统简介

Linux系统中常见的文件类型有普通文件(-)、目录(d)、字符设备文件(c)、块设备文件(b)、符号链接文件(l)等。



第一列为该文件属性，第二列为文件数量，第三列为文件的拥有者，第四列则为文件所属的用户组，第五列为文件大小，第六列为文件创建时间，最后一列为文件名。

其中，文件属性中第1个字符为文件类型标识符；随后跟着三组文件权限属性说明，每个分组的权限属性为三个字符表示。第一个分组表示文件拥有者针对文件的权限，第二个分组表示同属于一个用户组的用户对该文件的访问权限，第三个分组表示其它用户对该文件的权限。使用字符r、w、x分别表示操作文件可读、可写以及可执行。

## 相关结构体

#include <sys/types.h>

#include <sys/stat.h>

#include <unistd.h>

#include <dirent.h>

1. **目录结构体DIR**

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;

1. **目录项结构体dirent**

struct dirent

{

long d\_ino; // 索引节点号

off\_t d\_off; // 在目录文件中的偏移

unsigned short d\_reclen; // 文件名长

unsigned char d\_type; // 文件类型

har d\_name [NAME\_MAX+1]; // 文件名，最长255字符

}

1. **文件属性结构体stat**

struct stat

{

dev\_t st\_dev; //文件设备

ino\_t st\_ino; //文件节点号

mode\_t st\_mode; //文件类型与权限信息（详细信息见后）

nlink\_t st\_nlink; //文件符号链接数

uid\_t st\_uid; //文件用户ID

gid\_t st\_gid; //文件属组ID

dev\_t st\_rdev; //设备文件号

off\_t st\_size; //文件大小(单位是字节)

blksize\_t st\_blksize; //最大I/O块长度

blkcnt\_t st\_blocks; //分配给文件的块单元数量，每个单元为512字节

time\_t st\_atime; //文件最后一次访问时间

time\_t st\_mtime; //文件最后一次内容修改时间

time\_t st\_ctime; //文件结构最后一次修改时间

};

**mode\_t的定义**

*File type:*

S\_IFREG regular

S\_IFDIR directory

S\_IFCHR character special

S\_IFBLK block special

S\_IFLNK symbolic link

*File mode bits:*

S\_IRWXU read, write, execute/search by owner

S\_IRUSR read permission, owner

S\_IWUSR write permission, owner

S\_IXUSR execute/search permission, owner

S\_IRWXG read, write, execute/search by group

S\_IRGRP read permission, group

S\_IWGRP write permission, group

S\_IXGRP execute/search permission, group

S\_IRWXO read, write, execute/search by others

S\_IROTH read permission, others

S\_IWOTH write permission, others

S\_IXOTH execute/search permission, others

备注：1. 文件权限可以用3位的8进制数来设置。

2. umask命令，默认权限是775（umask=002）。

## C++处理函数

1. **创建目录：**

int mkdir(const char \*path, mode\_t mode);

参数：path：目录名；mode：目录权限，

返回值：返回0表示成功，返回-1表示错误，并且会设置errno值。

1. **改变当前工作目录：**

int chdir(const char \*path );

成功返回0，错误返回-1。

1. **删除目录：**

int rmdir(const char\* path);

1. **打开目录：**

DIR\* opendir(const char \*pathname);

1. **遍历目录内容：**

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

1. **重新回到目录的开始：**

void rewinddir(DIR \*dirp);

1. **关闭目录：**

int closedir(DIR \*dirp);

1. **读取文件属性**

int stat(const char\* file\_name, struct stat\* buf);

1. **删除文件：**

int remove(char \* filename);

1. **通过文件流（fstream）操作文件**

包含的头文件： <fstream>

fstream提供了三个类，用来实现c++对文件的操作（文件的创建，读写）。文件类型可以分为两种：文本文件和二进制文件。

ifstream 文件输入流，读文件

ofstream 文件输出流，写文件

fstream 打开文件供读写

// 写文件

ofstream file;

file.open("file.txt");

file<<"Hello file/n";

file.close();

// 读文件

ifstream file;

file.open("file.txt");

cout<<file.rdbuf();;

file.close();

open函数的参数定义了文件的打开模式，主要的模式如下：

ios::in 读

ios::out 写

ios::app 从文件末尾开始写

ios::binary 二进制模式

ios::trunc 打开一个文件，然后清空内容

ios::ate 打开一个文件时，将位置移动到文件尾

备注：默认模式是文本；默认如果文件不存在，那么创建一个新文件；多种模式可以混合，用|(按位或)。