# 一、access库函数

access函数用于判断当前操作系统用户对文件或目录的存取权限。

包含头文件：

#include <unistd.h>

函数声明：

int access(const char \*pathname, int mode);

参数说明：

pathname文件名或目录名，可以是当前目录的文件或目录，也可以列出全路径。

mode 需要判断的存取权限。在头文件unistd.h中的预定义如下：

#define R\_OK 4 // R\_OK 只判断是否有读权限

#define W\_OK 2 // W\_OK 只判断是否有写权限

#define X\_OK 1 // X\_OK 判断是否有执行权限

#define F\_OK 0 // F\_OK 只判断是否存在

返回值：

当pathname满足mode的条件时候返回0，不满足返回-1。

在实际开发中，access函数主要用于判断文件或目录是否是存在。

# 二、stat库函数

## 1、stat结构体

struct stat结构体用于存放文件和目录的状态信息，如下：

struct stat

{

dev\_t st\_dev; // device 文件的设备编号

ino\_t st\_ino; // inode 文件的i-node

mode\_t st\_mode; // protection 文件的类型和存取的权限

nlink\_t st\_nlink; // number of hard links 连到该文件的硬连接数目, 刚建立的文件值为1.

uid\_t st\_uid; // user ID of owner 文件所有者的用户识别码

gid\_t st\_gid; // group ID of owner 文件所有者的组识别码

dev\_t st\_rdev; // device type 若此文件为设备文件, 则为其设备编号

off\_t st\_size; // total size, in bytes 文件大小, 以字节计算

unsigned long st\_blksize; // blocksize for filesystem I/O 文件系统的I/O 缓冲区大小.

unsigned long st\_blocks; // number of blocks allocated 占用文件区块的个数, 每一区块大小为512 个字节.

time\_t st\_atime; // time of lastaccess 文件最近一次被存取或被执行的时间, 一般只有在用mknod、 utime、read、write 与tructate 时改变.

time\_t st\_mtime; // time of last modification 文件最后一次被修改的时间, 一般只有在用mknod、 utime 和write 时才会改变

time\_t st\_ctime; // time of last change i-node 最近一次被更改的时间, 此参数会在文件所有者、组、 权限被更改时更新

};

struct stat结构体的成员变量比较多，对程序员来说，重点关注st\_mode、st\_size和st\_mtime成员就可以了。注意st\_mtime是一个整数表达的时间，需要程序员自己写代码转换格式。

st\_mode成员的取值很多，或者使用如下两个宏来判断。

S\_ISREG(st\_mode) // 是否为一般文件

S\_ISDIR(st\_mode) // 是否为目录

## 2、stat库函数

包含头文件：

#include <sys/types.h>

#include <sys/stat.h>

#include <unistd.h>

函数声明：

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

stat函数获取path指定文件或目录的信息，并将信息保存到结构体buf中，执行成功返回0，失败返回-1。

**示例（book145.c）**

/\*

\* 程序名：book145.c，此程序演示目录和文件的存取权限和状态信息

\*/

#include <stdio.h>

#include <sys/stat.h>

#include <unistd.h>

// 本程序运行要带一个参数，即文件或目录名

int main(int argc,char \*argv[])

{

if (argc != 2) { printf("请指定目录或文件名。\n"); return -1; }

if (access(argv[1],F\_OK) != 0) { printf("文件或目录%s不存在。\n",argv[1]); return -1; }

struct stat ststat;

// 获取文件的状态信息

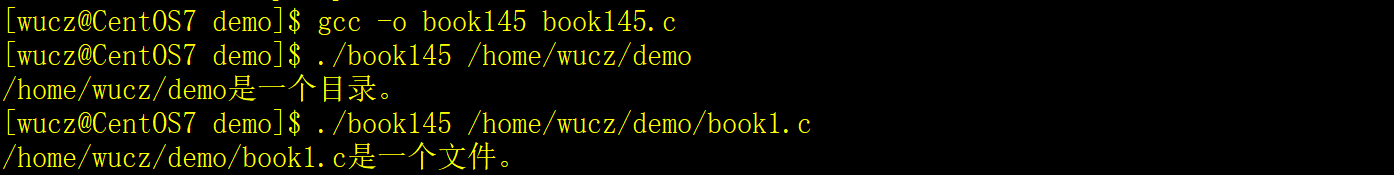
if (stat(argv[1],&ststat) != 0) return -1;

if (S\_ISREG(ststat.st\_mode)) printf("%s是一个文件。\n",argv[1]);

if (S\_ISDIR(ststat.st\_mode)) printf("%s是一个目录。\n",argv[1]);

}

**运行效果**



# 三、utime库函数

utime函数用于修改文件的存取时间和更改时间。

包含头文件：

#include <utime.h>

函数声明：

int utime(const char \*filename, const struct utimbuf \*times);

函数说明：utime()用来修改参数filename 文件所属的inode 存取时间。如果参数times为空指针(NULL), 则该文件的存取时间和更改时间全部会设为目前时间。结构utimbuf 定义如下：

struct utimbuf

{

time\_t actime;

time\_t modtime;

};

返回值：执行成功则返回0，失败返回-1。

# 四、rename库函数

rename函数用于重命名文件或目录，相当于操作系统的mv命令，对程序员来说，在程序中极少重命名目录，但重命名文件是经常用到的功能。

包含头文件：

#include <stdio.h>

函数声明：

int rename(const char \*oldpath, const char \*newpath);

参数说明：

oldpath 文件或目录的原名。

newpath 文件或目录的新的名称。

返回值：0-成功，-1-失败。

# 五、remove库函数

remove函数用于删除文件或目录，相当于操作系统的rm命令。

包含头文件：

#include <stdio.h>

函数声明：

int remove(const char \*pathname);

参数说明：

pathname 待删除的文件或目录名。

返回值：0-成功，-1-失败。