

LINUX下的函数调用

- stat():获取C文件状态

```
1 int stat(const char * file_name, struct stat *buf);
2 函数说明: stat()用来将参数file_name 所指的文件状态, 复制到参数buf 所指的结构中。
```

下面是struct stat的参数说明

```
1 struct stat
2 {
3     dev_t st_dev; //device 文件的设备编号
4     ino_t st_ino; //inode 文件的i-node
5     mode_t st_mode; //protection 文件的类型和存取的权限
6     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.
7     uid_t st_uid; //user ID of owner 文件所有者的用户识别码
8     gid_t st_gid; //group ID of owner 文件所有者的组织识别码
9     dev_t st_rdev; //device type 若此文件为装置设备文件, 则为其设备编号
10    off_t st_size; //total size, in bytes 文件大小, 以字节计算
11    unsigned long st_blksize; //blocksize for filesystem I/O 文件系统的I/O 缓冲区大小.
12    unsigned long st_blocks; //number of blocks allocated 占用文件区块的个数, 每一区块大小为
13    time_t st_atime; //time of last access 文件最近一次被存取或被执行的时间, 一般只有在用mknod
14    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
15    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
16 };
```

st_mode 定义了以下数种情况

```
1 S_IFMT 0170000 文件类型的位遮罩
2 S_IFSOCK 0140000 socket
3 S_IFLNK 0120000 符号连接
4 S_IFREG 0100000 一般文件
5 S_IFBLK 0060000 区块装置
6 S_IFDIR 0040000 目录
7 S_IFCHR 0020000 字符装置
8 S_IFIFO 0010000 先进先出
9 S_ISUID 04000 文件的 (set user-id on execution)位
10 S_ISGID 02000 文件的 (set group-id on execution)位
11 S_ISVTX 01000 文件的sticky 位

12 S_IRUSR (S_IREAD) 00400 文件所有者具可读取权限
```

- 13 S_IWUSR (S_IWRITE) 00200 文件所有者具可写入权限
- 14 S_IXUSR (S_IEXEC) 00100 文件所有者具可执行权限
- 15 S_IRGRP 00040 用户组具可读取权限
- 16 S_IWGRP 00020 用户组具可写入权限
- 17 S_IXGRP 00010 用户组具可执行权限
- 18 S_IROTH 00004 其他用户具可读取权限
- 19 S_IWOTH 00002 其他用户具可写入权限
- 20 S_IXOTH 00001 其他用户具可执行权限 上述的文件类型在 POSIX 中定义了检查这些类型的宏定义
- 21 S_ISLNK (st_mode) 判断是否为符号连接
- 22 S_ISREG (st_mode) 是否为一般文件
- 23 S_ISDIR (st_mode) 是否为目录
- 24 S_ISCHR (st_mode) 是否为字符装置文件
- 25 S_ISBLK (s3e) 是否为先进先出
- 26 S_ISSOCK (st_mode) 是否为socket 若一目录具有sticky 位 (S_ISVTX)，则表示在此目录下的文件只能被

st_mode 可以使用如下的宏来判断状态

- 1 S_ISREG是否是一个常规文件。
- 2 S_ISDIR是否是一个目录
- 3 S_ISCHR是否是一个字符设备。
- 4 S_ISBLK是否是一个块设备
- 5 S_ISFIFO是否是一个FIFO文件。
- 6 S_ISSOCK是否是一个SOCKET文件。

- sprintf()

函数用于将格式化的数据写入字符串

函数原型

- ```
1 int sprintf(char *str, char * format [, argument, ...]);
2 str为要写入的字符串; format为格式化字符串, 与printf()函数相同; argument为变量。
3 sprintf(s, "%d", 123); //把整数123打印成一个字符串保存在s中
```

- recv()和send()
- 测试数组问题 比如1024个数组中填充5个 strlen(buf)是5 还是1024
- 正则表达式
- strcmp()函数