Technology Tarena 达内科技

Unix系统高级编程

元数据和访问测试 Unit11

文件元数据

Technology Tarena 达内科技

获取文件元数据

获取文件元数据

- 从i节点中提取文件的元数据，即文件的属性信息

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

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

                                成功返回0，失败返回-1
```

  - *path*：文件路径
  - *buf*：文件元数据结构
  - *fd*：文件描述符
- 将指定文件的元数据填到*buf*参数所指向的元数据结构中
- lstat与另两个函数的区别仅在于它不跟踪符号链接

知识讲解

Technology Tarena 达内科技

文件元数据结构

文件元数据结构

- stat函数族通过stat结构体，向调用者输出文件的元数据
  - struct stat {

```
    dev_t     st_dev;     // 设备ID
    ino_t     st_ino;     // i节点号
    mode_t    st_mode;    // 文件类型和权限
    nlink_t   st_nlink;   // 硬链接数
    uid_t     st_uid;     // 用户ID
    gid_t     st_gid;     // 组ID
    dev_t     st_rdev;    // 特殊设备ID
    off_t     st_size;    // 总字节数
    blksize_t st_blksize; // I/O块字节数
    blkcnt_t  st_blocks;  // 占用块(512字节)数
    time_t    st_atime;   // 最后访问时间
    time_t    st_mtime;   // 最后修改时间
    time_t    st_ctime;   // 最后状态改变时间
};
```

Technology Tarena 达内科技

文件类型和权限

文件类型和权限

- stat结构的st_mode成员表示文件的类型和权限，该成员在stat结构中被声明为mode_t类型，其原始类型在32位系统中被定义为unsigned int，即32位无符号整数，但到目前为止，只有其中的低18位有意义
- 用18位二进制数($B_{17}...B_0$)表示的文件类型和权限，从高到低可被分为五组
  - $B_{17}B_{16}B_{15}B_{14}B_{13}B_{12}$：文件类型
  - $B_{11}B_{10}B_9$：设置用户ID、设置组ID和粘滞
  - $B_8B_7B_6$：拥有者用户的读、写和执行权限
  - $B_5B_4B_3$：拥有者组的读、写和执行权限
  - $B_2B_1B_0$：其它用户的读、写和执行权限

知识讲解

文件类型和权限（续1）

- 文件类型：$B_{17}...B_{12}$

mode_t st_mode; // unsigned int

…	B_{17}	B_{16}	B_{15}	B_{14}	B_{13}	B_{12}	B_{11}	B_{10}	B_9	B_8	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0

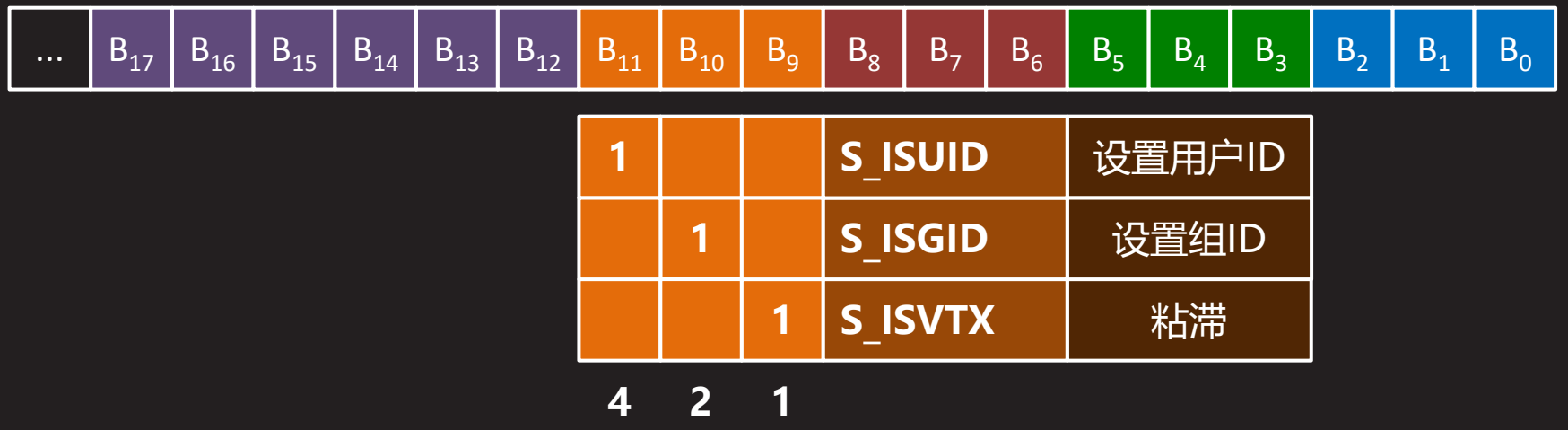
		1				S_IFREG	普通文件
			1			S_IFDIR	目录
		1	1			S_IFSOCK	本地套接字
				1		S_IFCHR	字符设备
			1	1		S_IFBLK	块设备
		1		1		S_IFLNK	符号链接
					1	S_IFIFO	有名管道
		1	1	1	1	S_IFMT	掩码

知识讲解

文件类型和权限（续2）

- 设置用户ID、设置组ID和粘滞：$B_{11}B_{10}B_9$

mode_t st_mode; // unsigned int



知识讲解

文件类型和权限（续3）

- 设置用户ID、设置组ID和粘滞：$B_{11}B_{10}B_9$
  - 带有设置用户ID位(B_{11})的可执行文件
    - 系统中的每个进程其实都有两个用户ID，一个叫实际用户ID，一个叫有效用户ID
    - 进程的实际用户ID继承自其父进程的实际用户ID。当一个用户通过合法的用户名和口令登录系统以后，系统就会为他启动一个Shell进程，Shell进程的实际用户ID就是该登录用户的用户ID。该用户在Shell下启动的任何进程都是Shell进程的子进程，自然也就继承了Shell进程的实际用户ID
    - 一个进程的用户身份决定了它可以访问哪些资源，比如读、写或者执行某个文件。但真正被用于权限验证的并不是进程的实际用户ID，而是其有效用户ID。一般情况下，进程的有效用户ID就取自其实际用户ID，可以认为二者是等价的
    - 但是，如果用于启动该进程的可执行文件带有设置用户ID位，即B_{11}位为1，那么该进程的有效用户ID就不再取自其实际用户ID，而是取自可执行文件的拥有者用户ID

文件类型和权限（续4）

- 设置用户ID、设置组ID和粘滞：$B_{11}B_{10}B_9$
  - 带有设置用户ID位(B_{11})的可执行文件
    - 系统管理员常用这种方法提升普通用户的权限，让他们有能力去完成一些本来只有root用户才能完成的任务。例如，他可以为某个拥有者用户为root的可执行文件添加设置用户ID位，这样一来无论运行这个可执行文件的实际用户是谁，启动起来的进程的有效用户ID都是root，凡是root用户可以访问的资源，该进程都可以访问。当然，具体访问哪些资源，以何种方式访问，还要由这个可执行文件的代码来决定。作为一个安全的操作系统，不可能允许一个低权限用户在高权限状态下为所欲为。如通过passwd命令修改口令
  - 带有设置组ID位(B_{10})的可执行文件
    - 与设置用户ID位的情况类似，如果一个可执行文件带有设置组ID位，即B_{10}位为1，那么运行该可执行文件所得到的进程，它的有效组ID同样不取自其实际组ID，而是取自可执行文件的拥有者组ID

知识讲解

文件类型和权限（续5）

- 设置用户ID、设置组ID和粘滞：$B_{11}B_{10}B_9$
  - 带有设置用户ID位(B_{11})的不可执行文件
    - 一个不可执行的文件带有设置用户ID位是毫无意义的
  - 带有设置组ID位(B_{10})的不可执行文件
    - 一个不可执行的文件带有设置组ID位同样毫无意义，但在某些系统上，这种无意义的组合恰被用于作为强制锁的标志
  - 带有设置用户ID位(B_{11})的目录
    - 目录带有设置用户ID位没有任何意义
  - 带有设置组ID位(B_{10})的目录
    - 在该目录下创建的文件及子目录，其拥有者组取自该目录的拥有者组，而非创建者用户所隶属的组

知识讲解

文件类型和权限（续6）

- 设置用户ID、设置组ID和粘滞：$B_{11}B_{10}B_9$
  - 带有粘滞位(B_9)的可执行文件
    - 在其首次运行并结束后， 其代码区被连续地保存在磁盘交换区中，而一般磁盘文件的数据块是离散存放的。因此，下次运行该程序可以获得较快的载入速度
  - 带有粘滞位(B_9)的不可执行文件
    - 不可执行文件带有粘滞位没有任何意义
  - 带有粘滞位(B_9)的目录
    - 除root以外的任何用户在该目录下，都只能删除或者更名那些属于自己的文件或子目录，而对于其它用户的文件或子目录，既不能删除也不能改名
    - 如/tmp目录

知识讲解

文件类型和权限（续7）

- 拥有者用户的读、写和执行权限：$B_8B_7B_6$

mode_t st_mode; // unsigned int

拥有者用户可读	S_IRUSR	1		
拥有者用户可写	S_IWUSR		1	
拥有者用户可执行	S_IXUSR			1
掩码	S_IRWXU	1	1	1
		4	2	1

知识讲解

文件类型和权限（续8）

- 拥有者组的读、写和执行权限：$B_5B_4B_3$

mode_t st_mode; // unsigned int

拥有者组可读	S_IRGRP	1		
拥有者组可写	S_IWGRP		1	
拥有者组可执行	S_IXGRP			1
掩码	S_IRWXG	1	1	1
		4	2	1

知识讲解

文件类型和权限（续9）

- 其它用户的读、写和执行权限：$B_2B_1B_0$

mode_t st_mode; // unsigned int

其它用户可读	S_IROTH	1		
其它用户可写	S_IWOTH		1	
其它用户可执行	S_IXOTH			1
掩码	S_IRWXO	1	1	1
		4	2	1

知识讲解

文件类型和权限（续10）

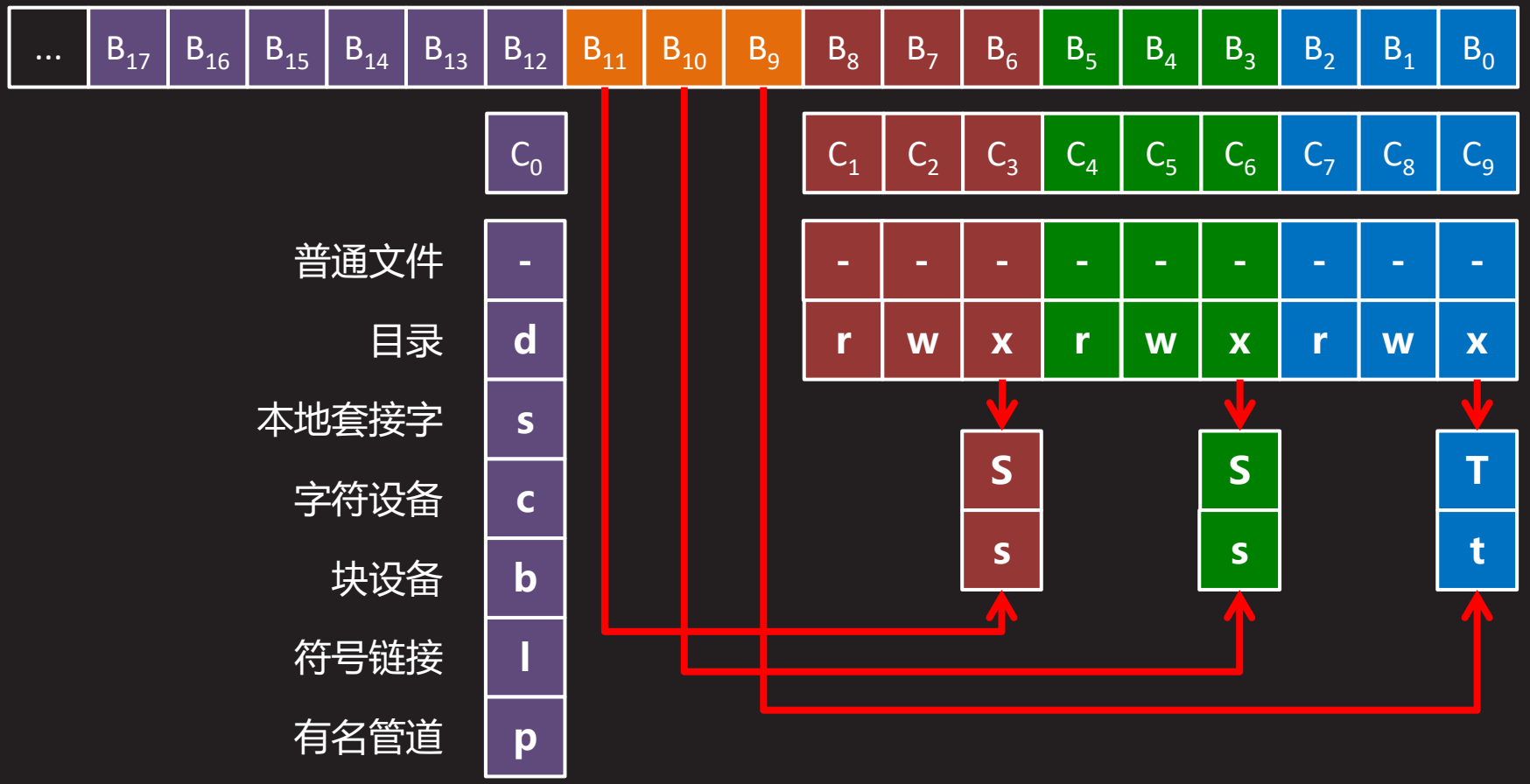
- 辅助分析文件类型的实用宏
  - S_ISREG()：是否普通文件
  - S_ISDIR()：是否目录
  - S_ISSOCK()：是否本地套接字
  - S_ISCHR()：是否字符设备
  - S_ISBLK()：是否块设备
  - S_ISLNK()：是否符号链接
  - S_ISFIFO()：是否有名管道

知识讲解

文件类型和权限（续11）

- 文件类型和权限字符串：$C_0C_1C_2C_3C_4C_5C_6C_7C_8C_9$

mode_t st_mode; // unsigned int



知识讲解

文件类型和权限（续12）

- 例如：将整数形式的文件类型和权限转换为字符串的函数

```
const char* mtos (mode_t m) {
    static char s[11];
    if      (S_ISDIR  (m)) strcpy (s, "d");
    else if (S_ISSOCK (m)) strcpy (s, "s");
    else if (S_ISCHR  (m)) strcpy (s, "c");
    else if (S_ISBLK  (m)) strcpy (s, "b");
    else if (S_ISLNK  (m)) strcpy (s, "l");
    else if (S_ISFIFO (m)) strcpy (s, "p");
    else                   strcpy (s, "-");
    strcat (s, m & S_IRUSR ? "r" : "-"); strcat (s, m & S_IWUSR ? "w" : "-");
    strcat (s, m & S_IXUSR ? "x" : "-");
    strcat (s, m & S_IRGRP ? "r" : "-"); strcat (s, m & S_IWGRP ? "w" : "-");
    strcat (s, m & S_IXGRP ? "x" : "-");
    strcat (s, m & S_IROTH ? "r" : "-"); strcat (s, m & S_IWOTH ? "w" : "-");
    strcat (s, m & S_IXOTH ? "x" : "-");
    if (m & S_ISUID) s[3] = (s[3] == 'x' ? 's' : 'S');
    if (m & S_ISGID) s[6] = (s[6] == 'x' ? 's' : 'S');
    if (m & S_ISVTX) s[9] = (s[9] == 'x' ? 't' : 'T');
    return s; }
```

知识讲解

文件元数据

【参见：stat.c】

课堂练习

- 文件元数据

访问测试

Technology Tarena 达内科技

访问测试

访问测试

- 测试调用进程对指定文件是否拥有足够的访问权限

```
#include <unistd.h>

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

成功返回0，失败返回-1

  - *pathname*：文件路径
  - *mode*：访问权限，可取以下值

    `R_OK` - 可读否

    `W_OK` - 可写否

    `X_OK` - 可执行否

    `F_OK` - 存在否

知识讲解

访问测试（续1）

- 例如
  - printf ("文件%s", pathname);
    if (access (pathname, F_OK) == -1)
        printf ("不存在(%m)。\n");
    else {
        if (access (pathname, R_OK) == -1)
            printf ("不可读(%m)，");
        else printf ("可读，");
        if (access (pathname, W_OK) == -1)
            printf ("不可写(%m)，");
        else printf ("可写，");
        if (access (pathname, X_OK) == -1)
            printf ("不可执行(%m)。\n");
        else printf ("可执行。\n"); }

知识讲解

访问测试（续2）

- access函数在测试调用进程对指定文件的访问权限时，使用的是调用进程的实际用户ID和实际组ID，而非有效用户ID和有效组ID，因此如果调用进程的可执行文件带有设置用户ID位或者设置组ID位，access函数返回无权访问并不意味着就真的无权访问

知识讲解

访问测试

【参见：access.c】

课堂练习

- 访问测试

Technology Tarena 达内科技

总结和答疑