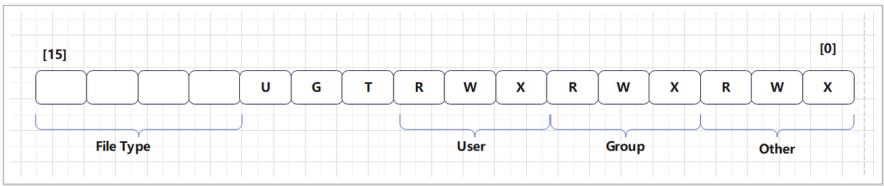


3.2 ls 命令设计与实现 (二)_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 3.2 ls 命令设计与实现 (二) 涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

- 基本思路:
 - 通过 stat 函数获取文件信息，并保存到 struct stat 结构体中
 - struct stat 结构体 st_mode 成员则可以解析到文件类型



- 注意点:
 - 对于链接文件本身需要通过 lstat 函数来获取，否则获取的链接文件所指向的文件
- 获取属性的主要接口为 get_file_attr，并调用

```
int get_file_attr(struct file_attribute *pattr,const char *path,
                  const char *filename,bool islink)
{
    int ret;

    if(islink)
        ret = lstat(path,&pattr->f_attr_stat_info);
    else
        ret = stat(path,&pattr->f_attr_stat_info);

    if(ret == -1)
    {
        perror("stat(): ");
        return -1;
    }

    get_file_type_ls(pattr);

    return 0;
}
```

- 在上面的函数中调用 get_file_type_ls 函数，用于获取文件类型
- 显示属性接口设计为 show_file_attributes 函数, 下面只是打印了文件类型

```
void show_file_attributes(struct file_attribute *pattr)
{
    printf(" %c",pattr->f_attr_type);

    putchar('\n');
}
```

- 获取文件类型 是通过解析 struct stat 结构体中的 st_mode 的第 [15:12] bit 来判断
 - 通过与 S_IFMT 进行按位与操作，则可以得到 [15:12] 的值
 - 在与定义的八进制的值进行比对即可，具体定义如下

```
#define S_IFMT 00170000
#define S_IFSOCK 0140000
```

```
#define S_IFLNK 0120000
#define S_IFREG 0100000
#define S_IFBLK 0060000
#define S_IFDIR 0040000
#define S_IFCHR 0020000
#define S_IFIFO 0010000
```

- 获取文件类型的接口设计为 `get_file_type_ls`，将获取的文件属性信息保存到结构体相应的成员中

```
int get_file_type_ls(struct file_attribute *pattr)
{
    mode_t mode = pattr->f_attr_stat_info.st_mode;

    if(pattr == NULL)
        return -1;

    switch(mode & S_IFMT)
    {
        case S_IFBLK :
            pattr->f_attr_type = 'b';
            break;

        case S_IFCHR :
            pattr->f_attr_type = 'c';
            break;

        case S_IFDIR :
            pattr->f_attr_type = 'd';
            break;

        case S_IFIFO :
            pattr->f_attr_type = 'p';
            break;

        case S_IFLNK :
            pattr->f_attr_type = 'l';
            break;

        case S_IFREG :
            pattr->f_attr_type = '-';
            break;

        case S_IFSOCK:
            pattr->f_attr_type = 's';
            break;

        default:
            break;
    }

    return 0;
}
```

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta，点击查看详细说明

