

3.6 ls 命令设计与实现 (六) _物联网 / 嵌入式工 程师 - 慕课网

“ 慕课网慕课教程 3.6 ls 命令设计与实现 (六) 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。

- 软链接文件比较特殊，需要进行特殊处理，具体处理如下：
 - 当需要获取链接文件的信息时，不能直接通过 stat 函数，需要调用 lstat 函数
 - 在 get_file_attr 函数中要进行分开处理

```
int get_file_attr(struct file_attribute *pattr,const char *path,const cha
{
    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;
    }
}
```

- 需要读取软链接文件所指向的文件，则需要 使用 readlink 函数进行读取，原
因为系统 ls 显示软链接是会显示具体内容

```
total 16
drwxrwxr-x 2 ben ben 4096 Oct  4 16:28 cpdir
drwxrwxr-x 2 ben ben 4096 Oct  6 09:08 demo
lrwxrwxrwx 1 ben ben   8 Oct  6 12:02 link test.txt -> test.txt
drwxrwxr-x 3 ben ben 4096 Oct  4 17:30 ls
drwxrwxr-x 2 ben ben 4096 Oct  6 12:02 project
-rw-rw-r-- 1 ben ben   0 Oct  6 12:02 test.txt
```

- readlink 函数具体信息如下：
- 函数头文件
- #include <unistd.h>
- 函数原型
- ssize_t readlink(const char *pathname, char *buf, size_t bufsiz);
- 函数功能
- 读取链接文件的内容，就是所指向的文件的文件名
- 函数参数
- pathname：路径名
- buf：缓冲区地址
- bufsiz：读取的最大大小
- 函数返回值
- 成功：读取的字节数
- 失败：-1, 并设置 errno
- 在 get_file_attr 函数中的实现如下：
- if(pattr->f_attr_type == 'l')

```

        ret = readlink(path, pattr->f_attr_link_content, sizeof(pattr->f_attr_link_content));
        if (ret == -1)
        {
            perror("readlink(): ");
            return -1;
        }
    }
}

```

- 在 show_file_attribute 函数中的实现如下:

```

oid show_file_attributes(struct file_attribute *pattr)
{
    printf(" %c", pattr->f_attr_type);
    printf(" %s ", pattr->f_attr_permission);
    printf(" %ld ", pattr->f_attr_stat_info.st_nlink);
    printf(" %s ", pattr->f_attr_uname);
    printf(" %s ", pattr->f_attr_gname);
    printf(" %ld ", pattr->f_attr_stat_info.st_size);
    printf(" %s ", pattr->f_attr_mtime);

    if (pattr->f_attr_type == 'l')
        printf(" %s -> %s \n", pattr->f_attr_name, pattr->f_attr_link_content);

    printf(" %s \n", pattr->f_attr_name);
    putchar('\n');
}

```

- 文件名的输出, 将命令输入的参数保存到相应 struct file_attribute 结构体中即可
- 在 get_file_attr 中的输出:

```

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);

    if (pattr->f_attr_type == 'l')
    {
        ret = readlink(path, pattr->f_attr_link_content, sizeof(pattr->f_attr_link_content));
        if (ret == -1)
        {
            perror("readlink(): ");
            return -1;
        }
    }

    get_file_permission(pattr);
    get_file_uname(pattr);
    get_file_gname(pattr);

    get_file_last_modify_time(pattr);

    strcpy(pattr->f_attr_name, filename);

    return 0;
}

```

- 在 show_file_attribute 函数中进行打印

```

void show_file_attributes(struct file_attribute *pattr)
{

```

```
printf(" %c",pattr->f_attr_type);
printf(" %s ",pattr->f_attr_permission);
printf(" %ld ",pattr->f_attr_stat_info.st_nlink);
printf(" %s ",pattr->f_attr_uname);
printf(" %s ",pattr->f_attr_gname);
printf(" %ld ",pattr->f_attr_stat_info.st_size);
printf(" %s ",pattr->f_attr_mtime);

if(pattr->f_attr_type == 'l')
    printf(" %s -> %s \n",pattr->f_attr_name,pattr->f_attr_link_content);

printf(" %s \n",pattr->f_attr_name);

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

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

