

Linux上机作业2

基本要求

编程实现程序 `list.c`，列表普通磁盘文件，包括文件名和文件大小。

- (1) 使用 `vi` 编辑文件，熟悉工具vi。
- (2) 使用Linux的系统调用和库函数。
- (3) 体会Shell**文件通配符**的处理方式以及命令对选项的处理方式。

对选项的处理，自行编程逐个分析命令行参数。不考虑多选项挤在一个命令行参数内的情况。

处理对象和选项

`ls` 命令类似,处理对象可以有0到多个

0个:列出当前目录下所有文件（**不含点打头的文件**，如 `.profile`）

普通文件:列出文件

目录:列出目录下所有文件

实现自定义选项 `r`，`a`，`l`，`h`，`m` 以及 `--`

`-r` 递归方式列出子目录（每项要含路径，类似 `find` 的 `-print` 输出风格，需要设计**递归程序**）

`-a` 列出文件名第一个字符为**圆点**的普通文件（默认情况下不列出文件名首字符为圆点的文件）

`-l` 后跟一整数,限定文件**大小的最小值**（字节）

`-h` 后跟一整数,限定文件**大小的最大值**（字节）

`-m` 后跟一整数 `n`，限定文件的**最近修改时间**必须在 `n` 天内

`--` 显式地**终止命令选项分析**

Linux上机作业2

一、编写 `analyse_parameter()` 函数

二、编写 `print_file_info()` 函数

1.头文件中的接口

2.获取文件状态

3.该文件是目录

4.该文件不是目录

三、编写 `read_dir()` 函数，读取目录

打开目录失败

打开目录成功

四、测试

打印错误和帮助信息

`-a` 列出原点文件

`-r` 递归打印文件

`-l` 和 `-h` 限制文件大小

```
-- 结束参数
./list *
```

一、编写 analyse_parameter() 函数

首先定义并初始化储存参数信息的结构体

```
1 struct para { //0表示未使用, 1表示使用
2     int r;
3     int a;
4     int l;
5     int h;
6     int m;
7     int min; //文件大小最小值
8     int max; //文件大小最大值
9     int day; //修改天数
10 }Parameter = {0, 0, 0, 0, 0, -1, -1, -1}; //初始化
```

实现了自定义选项 `r` , `a` , `l` , `h` , `m` 以及 `--`

从 `main` 函数中的 `int argc, char argv[]` 获得参数内容。

之后遍历并分析参数, 详见[注释](#)

```
1 //遍历参数
2 for (int i = 1; i < argc; i++) {
3     if (strcmp(argv[i], "-r") == 0) {
4         Parameter.r = 1; //有-r参数
5     }
6     else if (strcmp(argv[i], "-a") == 0) {
7         Parameter.a = 1; //有-a参数
8     }
9     else if (strcmp(argv[i], "-l") == 0) {
10         Parameter.l = 1; //有-l参数
11         i++; //下一个参数是最小值
12         Parameter.min = atoi(argv[i]); //转换为整数保存
13     }
14     else if (strcmp(argv[i], "-h") == 0) {
15         Parameter.h = 1; //有-h参数
16         i++; //下一个参数是最大值
17         Parameter.max = atoi(argv[i]); //转换为整数保存
```

```

19 else if (strcmp(argv[i], "-m") == 0) {
20     Parameter.m = 1; // 有-m参数
21     i++; // 下一个参数是天数
22     Parameter.day = atoi(argv[i]); // 转换为整数保存
23 }
24 else if (strcmp(argv[i], "--") == 0) { // -- 标记参数结束
25     i++; // 下一个参数是路径
26     if (i == argc) strcpy(path, "."); // 如果没有路径, 使用当前
    目录
27     else if (i == argc - 1) strcpy(path, argv[i]); // 如果有路
    径, 保存路径
28     else error(); // 参数错误
29     break; // 参数结束
30 }
31 else if (i == argc - 1) {
32     strcpy(path, argv[i]); // 如果没有-- 标记, 最后一个参数是路径
33 }
34 else {
35     error(); // 参数错误
36 }

```

二、编写 `print_file_info()` 函数

读取并打印文件信息

1. 头文件中的接口

先了解 `<sys/stat.h>` 头文件中的 `stat` 数据结构

```

1 struct stat {
2     dev_t st_dev; // 文件所在设备ID
3     ino_t st_ino; // 结点(inode)编号
4     mode_t st_mode; // 保护模式
5     nlink_t st_nlink; // 硬链接个数
6     uid_t st_uid; // 所有者用户ID
7     gid_t st_gid; // 所有者组ID
8     dev_t st_rdev; // 设备ID(如果是特殊文件)
9     off_t st_size; // 总体尺寸, 以字节为单位
10    blksize_t st_blksize; // 文件系统 I/O 块大小

```

```

12     time_t st_atime; // 上次访问时间
13     time_t st_mtime; // 上次更新时间
14     time_t st_ctime; // 上次状态更改时间
15 };

```

主要函数有

```

1  int stat(const char *restrict pathname, struct stat
   *restrict buf);
2  int fstat(int fields, struct stat *buf);
3  int lstat(const char *restrict pathname, struct stat
   *restrict buf);

```

其中 `stat` 函数的作用是获取路径名 `path` 对应的 `inode` 中的属性

`<sys/stat.h>` 中提供了宏 `S_ISDIR(info.st_mode)` 来判断文件是不是目录

2. 获取文件状态

获取文件状态并储存在 `stbuf` 结构中，采用终端转义字符 `\033[31m` 来输出红色的无法打开文件的消息

```

1  // 获取文件状态并储存在stbuf结构中
2  if (stat(name, &stbuf) == -1) { //
3      fprintf(stderr, "\033[31;41m打开%s 失败\033[0m\n", name);
   //打开文件失败
4      error();
5      return;
6  }

```

然后采用 `if (S_ISDIR(stbuf.st_mode))` 判断文件是不是目录

3. 该文件是目录

先判断是不是当前目录，不是当前目录，打印目录名

利用终端转义字符，将文件大小打印为紫色，将目录名打印为深绿色

```

1  //如果是目录，根据-r打印目录size及name
2  if (strcmp(name, path) != 0)
3      printf("\033[35m%8ld\033[0m\t\033[36m%s\033[0m\n",
   stbuf.st_size, name); //不是当前目录，打印目录名

```

如果有参数 `-r` 或者是**当前目录的话**，调用 `read_dir()`，读取目录中的内容

```
1 if (Parameter.r == 1 || strcmp(name, path) == 0) { //如果有 -
    r 遍历下一级目录
2     read_dir(name, print_result);
3 }
```

4. 该文件不是目录

不是目录，根据 `-l -h -m` 打印文件size及name

设置一个 `flag`，判断是否要输出

有参数 `-l`，`stbuf.st_size < Parameter.min`，不满足条件，不输出

有参数 `-h`，`stbuf.st_size < Parameter.min`，不满足条件，不输出

有参数 `-m`，根据修改是时间判断是否输出

如果满足条件，打印文件大小和文件名

利用**终端转义字符**，将**文件大小**打印为**紫色**，将**文件名**打印为**绿色**

```
1 //不是目录，根据-l -h -m打印文件size及name
2 int flag = 1; //判断是否要输出
3 if (Parameter.l == 1 && stbuf.st_size < Parameter.min) flag
    = 0; //不满足条件
4 if (Parameter.h == 1 && stbuf.st_size > Parameter.max) flag
    = 0;
5 if (Parameter.m == 1) { //判断文件修改时间
6     struct timeval nowTime; //当前时间
7     gettimeofday(&nowTime, NULL); //获取当前时间
8     if (nowTime.tv_sec - stbuf.st_mtim.tv_sec > (time_t)
        (Parameter.day*86400)) //判断是否超过指定天数
9         flag = 0; //不满足条件
10 }
11 if (flag == 1)
12     printf("\033[35m%8ld\033[0m\t\033[32m%s\n\033[0m",
        stbuf.st_size, name); //满足条件，打印文件大小和文件名
```

三、编写 `read_dir()` 函数，读取目录

```
1 void read_dir(char *dir, void (*func)(char *)); //读目录
```

对目录中所有文件执行 `print_result` 操作

把 `print_result` 函数作为参数传进去

打开目录失败

采用终端转义字符 `\033[31m` 来输出红色的无法打开路径的消息

```
1 if ((dfd = opendir(dir)) == NULL) {
2     fprintf(stderr, "\033[31mread_dir:无法打
3     开%s\033[0m\n", dir); // 打开目录失败
4     return;
5 }
```

打开目录成功

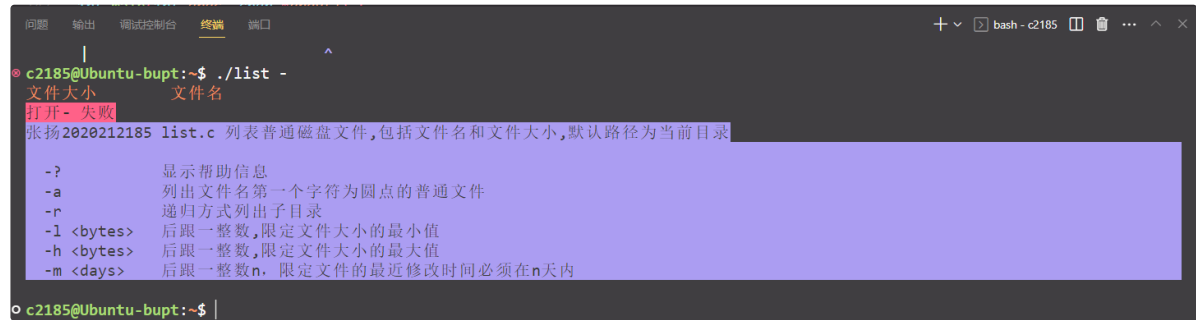
先**跳过当前目录** `.` 以及**父目录** `..` ,然后判断**文件名是否过长**,之后根据参数 `-a` 决定是否打印 `.` 开头的文件,然后**拼接文件名**,作为参数传入打印文件信息的函数 `print_result` 中。

```
1 while ((dp = readdir(dfd)) != NULL) { //读目录记录项
2     if (strcmp(dp->d_name, ".") == 0 || strcmp(dp->
3     d_name, "..") == 0) {
4         continue; //跳过当前目录以及父目录
5     }
6     if (strlen(dir) + strlen(dp->d_name) + 2 >
7     sizeof(name)) {
8         fprintf(stderr, "\033[31mread_dir : 文件名 %s %s 太
9         长\033[0m\n", dir, dp->d_name); //文件名过长
10        }
11        else {
12            if (dp->d_name[0] == '.' && Parameter.a == 0) //以.
13            开头的文件
14                continue;
15            sprintf(name, "%s/%s", dir, dp->d_name); //拼接文件
16            名
```

```
14     }  
15 }
```

四、测试

打印错误和帮助信息



```
bash - c2185  
c2185@Ubuntu-bupt:~$ ./list -  
文件大小      文件名  
打开 - 失败  
张扬2020212185 list.c 列表普通磁盘文件,包括文件名和文件大小,默认路径为当前目录  
-?      显示帮助信息  
-a      列出文件名第一个字符为圆点的普通文件  
-r      递归方式列出子目录  
-l <bytes> 后跟一整数,限定文件大小的最小值  
-h <bytes> 后跟一整数,限定文件大小的最大值  
-m <days> 后跟一整数n, 限定文件的最近修改时间必须在n天内  
c2185@Ubuntu-bupt:~$
```

-a 列出原点文件

```
1 | ./list -a
```

```

5791      ./list.c
● c2185@Ubuntu-bupt:~$ ./list -a
文件大小      文件名
4096      ./config/
4096      ./vscode-server/
183      ./wget-hsts
807      ./profile
4096      ./vscode/
5244      ./shudu.c
4096      ./vscode-remote-containers,
1      ./test.txt
220      ./bash_logout
4096      ./i_am_a_dir/
4096      ./cache/
17320     ./shudu
17368     ./list
3771     ./bashrc
288      ./123.txt
6853     ./viminfo
5481     ./bash_history
1119     ./背包+n皇后.cpp
5791     ./list.c
4096     ./dotnet/
4096     ./local/

```

-r 递归打印文件

```
1 | ./list -r
```

```

● c2185@Ubuntu-bupt:~$ ./list -r
文件大小      文件名
5244      ./shudu.c
1      ./test.txt
4096      ./i_am_a_dir/
0      ./i_am_a_dir/i_am_a_file
17320     ./shudu
17368     ./list
288      ./123.txt
1119     ./背包+n皇后.cpp
5791     ./list.c

```


-l 和 -h 限制文件大小

```
1 | ./list -l 100 -h 5000
```

```
root@zhangyang-virtual-machine:/home/zhangyang/code/list# ./list -l 100 -h 5000 /bin /etc
文件大小      文件名
4096          /etc/xml/
1566          /etc/shadow
4096          /etc/pm/
23           /etc/subuid
4436          /etc/hdparm.conf
4096          /etc/rc6.d/
2945          /etc/passwd-
5217          /etc/manpath.config
681           /etc/xattr.conf
12813         /etc/services
4096          /etc/polkit-1/
4096          /etc/apparmor/
769           /etc/appstream.conf
23           /etc/subgid
106           /etc/environment
104           /etc/lsb-release
4096          /etc/bluetooth/
4096          /etc/default/
```

由于ssh学校服务器没有root权限，这个命令我在自己的虚拟机上测试

-m 限制修改日期

```
1 | ./list -a -r -l 50000 -m 2
```

递归式列出当前目录树下大小超50KB且2天内修改过的文件（包括文件名首字符为圆点的文件）

```
root@zhangyang-virtual-machine:~# ./list -a -r -l 50000 -m 2
root@zhangyang-virtual-machine:~# /home/zhangyang/code/list/list -a -r -l 50000 -m 2
文件大小      文件名
4096          ./ssh/
4096          ./ssh/java-se-8u41-ri/
4096          ./ssh/java-se-8u41-ri/lib/
4096          ./ssh/java-se-8u41-ri/lib/amd64/
4096          ./ssh/java-se-8u41-ri/lib/amd64/jli/
4096          ./ssh/java-se-8u41-ri/sample/
4096          ./ssh/java-se-8u41-ri/sample/try-with-resources/
4096          ./ssh/java-se-8u41-ri/sample/try-with-resources/src/
4096          ./ssh/java-se-8u41-ri/sample/scripting/
4096          ./ssh/java-se-8u41-ri/sample/scripting/scriptpad/
4096          ./ssh/java-se-8u41-ri/sample/scripting/scriptpad/nbproject/
4096          ./ssh/java-se-8u41-ri/sample/scripting/scriptpad/src/
4096          ./ssh/java-se-8u41-ri/sample/scripting/scriptpad/src/scripts/
4096          ./ssh/java-se-8u41-ri/sample/scripting/scriptpad/src/META-INF/
4096          ./ssh/java-se-8u41-ri/sample/scripting/scriptpad/src/com/
4096          ./ssh/java-se-8u41-ri/sample/scripting/scriptpad/src/com/sun/
4096          ./ssh/java-se-8u41-ri/sample/scripting/scriptpad/src/com/sun/sample/
4096          ./ssh/java-se-8u41-ri/sample/scripting/scriptpad/src/com/sun/sample/scriptpad/
4096          ./ssh/java-se-8u41-ri/sample/scripting/scriptpad/src/resources/
4096          ./ssh/java-se-8u41-ri/sample/nio/
```

-- 结束参数

```
1 | ./list -- -l
2 | ./list -- ./i_am_a_dir
```

```

5791          ./list.c
● c2185@Ubuntu-bupt:~$ ./list -- -l
文件大小      文件名
打开 -l 失败
张扬2020212185 list.c 列表普通磁盘文件,包括文件名和文件大小,默认路径为当前目录

-?          显示帮助信息
-a          列出文件名第一个字符为圆点的普通文件
-r          递归方式列出子目录
-l <bytes>  后跟一整数,限定文件大小的最小值
-h <bytes>  后跟一整数,限定文件大小的最大值
-m <days>  后跟一整数n, 限定文件的最近修改时间必须在n天内

● c2185@Ubuntu-bupt:~$ ./list -- ./i_am_a_dir
文件大小      文件名
          0      ./i_am_a_dir/i_am_a_file
○ c2185@Ubuntu-bupt:~$ |

```

./list *

```

1119          背包+皇后.cpp
● c2185@Ubuntu-bupt:~$ ./list *
文件大小      文件名
          1119      背包+n皇后.cpp
○ c2185@Ubuntu-bupt:~$ |

```