

上机作业 2：遍历目录

088 于海鑫

2017211240

name1e5s@bupt.edu.cn

版本：8

更新：April 9, 2020

1 作业要求

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。

2 作业实现

2.1 获取选项

因为在这次作业中，使用 `getopt` 是不被允许的，因此我们首先需要一个用于处理选项的函数。在我们的实现中，我们决定基于 C 标准库函数 `strchr` 模拟实现一个 `getopt` 以支持我们的选项分析。核心代码如下：

```
1     optdecl = strchr(optstring, optchar);  
2     if (optdecl) {  
3         if (optdecl[1] == ':') {
```

```

4         optarg = ++optcursor;
5         if (*optarg == '\\0') {
6             if (++optind < argc) {
7                 optarg = argv[optind];
8             } else {
9                 optarg = NULL;
10                optchar = (optstring[0] == ':') ? ':' : '?';
11            }
12        }
13        optcursor = NULL;
14    }
15    } else {
16        optchar = '?';
17    }

```

2.2 输出单个文件

在输出单个文件时，我们使用 `stat` 函数来获取单个文件的状态信息，之后根据这些状态信息打印出文件。核心代码如下：

```

1 static void list_node(const char *prefix, const char *file_name) {
2     // 一些拼接代码
3
4     struct stat status;
5     if (stat(node_name, &status)) {
6         fprintf(stderr, "%s_ _Can't access_ \"%s\":_ %s\n", elf_name,
7             node_name, strerror(errno));
8         return;
9     }
10
11     mode_t stat_mode = status.st_mode;
12     off_t stat_size = status.st_size;
13     time_t stat_mtime = status.st_mtime;
14
15     // 一些格式化代码...
16
17     int filter_flag = (!modify_time || time (NULL) - stat_mtime <=
18         modify_time) &&

```

```

17         (!lo_size || stat_size >= lo_size) &&
18         (!hi_size || stat_size <= hi_size);
19     if((filter_flag && !S_ISDIR(stat_mode)) || (S_ISDIR(stat_mode) &&
20         recursive_flag)) {
21         char stat_time_str[64];
22         strftime(stat_time_str, 64,
23             "%Y-%m-%d_%H:%M", localtime(&stat_mtime));
24         printf ("%s_%8ld_%s_%s\n", mode_text, stat_size,
25             stat_time_str, file_name);
26     }
27
28     // 一些收尾代码
29 }

```

2.3 输出文件夹下的全部文件

基于上面输出单个文件的代码，我们可以实现输出某文件夹下全部文件的函数。其核心在于对 `readdir` 函数的使用。代码如下：

```

1 static void list_dir(const char *name) {
2     if(!init) {
3         printf("\n");
4     }
5     init = 0;
6
7     DIR *dir = opendir (name);
8     if (dir == NULL) {
9         fprintf(stderr, "%s-Can't access dir- \"%s\": %s\n",
10             elf_name, name, strerror(errno));
11         return;
12     }
13
14     printf("%s:\n", name);
15
16     struct dirent *entry;
17     int count = 0;
18     while ((entry = readdir(dir)) != NULL) {
19         const char *entry_name = entry->d_name;

```

```

19         if(!all_flag && entry_name[0] == '.')
20             continue;
21
22         list_node(name, entry_name);
23         count++;
24     }
25     printf("%d_files_in_total.\n", count);
26 }

```

2.4 递归输出

有时我们需要实现对于某文件夹的递归输出，此时我们使用广度优先搜索的方式，使用一个链表模拟的队列进行文件夹的递归输出。核心代码如下：

```

1 static void list_main(const char *name) {
2     struct stat status;
3     if(stat(name, &status)) {
4         fprintf(stderr, "%s-Can't access \"%s\": %s\n", elf_name,
5             name, strerror(errno));
6         return;
7     }
8     if(S_ISDIR(status.st_mode)) {
9         list_start = malloc(sizeof(list_node_t));
10        list_start->node = name;
11        list_start->next = NULL;
12        list_end = list_start;
13        while (list_start) {
14            list_dir(list_start->node);
15            list_start = list_start->next;
16        }
17    } else {
18        list_node("", name);
19    }
20 }

```

3 运行效果

3.1 基本使用

输出某一文件夹的内容时结果如下：

```
1 name1e5s@sumeru:~/Homework-2/build$ ./list ..
2 ...:
3 -rw-rw-r--      1719 2020-04-09 17:31 getopt.c
4 -rw-rw-r--      5141 2020-04-09 17:31 list.c
5 -rw-rw-r--       123 2020-04-09 17:31 CMakeLists.txt
6 -rw-rw-r--       559 2020-04-09 17:31 homemade_getopt.h
7 5 files in total.
```

3.2 递归输出

递归输出某一文件夹的内容时结果如下：

```
1 name1e5s@sumeru:~/Homework-2/build$ ./list ..
2 ...:
3 -rw-rw-r--      1719 2020-04-09 17:31 getopt.c
4 -rw-rw-r--      5141 2020-04-09 17:31 list.c
5 -rw-rw-r--       123 2020-04-09 17:31 CMakeLists.txt
6 -rw-rw-r--       559 2020-04-09 17:31 homemade_getopt.h
7 5 files in total.
8 name1e5s@sumeru:~/Homework-2/build$ ./list -r ..
9 ...:
10 -rw-rw-r--      1719 2020-04-09 17:31 getopt.c
11 -rw-rw-r--      5141 2020-04-09 17:31 list.c
12 -rw-rw-r--       123 2020-04-09 17:31 CMakeLists.txt
13 -rw-rw-r--       559 2020-04-09 17:31 homemade_getopt.h
14 drwxrwxr-x     4096 2020-04-09 17:31 build
15 5 files in total.
16
17 ../build:
18 -rw-rw-r--      1502 2020-04-09 17:31 cmake_install.cmake
19 -rwxrwxr-x     13832 2020-04-09 17:31 list
20 drwxrwxr-x     4096 2020-04-09 17:31 CMakeFiles
21 -rw-rw-r--      5304 2020-04-09 17:31 Makefile
22 -rw-rw-r--     12576 2020-04-09 17:31 CMakeCache.txt
23 5 files in total.
```

```

24
25 ../build/CMakeFiles:
26 -rw-rw-r--      3067 2020-04-09 17:31 Makefile2
27 -rw-rw-r--       85 2020-04-09 17:31 cmake.check_cache
28 -rwxrwxr-x    12312 2020-04-09 17:31 feature_tests.bin
29 drwxrwxr-x     4096 2020-04-09 17:31 list.dir
30 drwxrwxr-x     4096 2020-04-09 17:31 CMakeTmp
31 -rw-rw-r--       688 2020-04-09 17:31 feature_tests.c
32 drwxrwxr-x     4096 2020-04-09 17:31 3.10.2
33 -rw-rw-r--        2 2020-04-09 17:31 progress.marks
34 -rw-rw-r--    44929 2020-04-09 17:31 CMakeOutput.log
35 -rw-rw-r--       632 2020-04-09 17:31 CMakeDirectoryInformation.cmake
36 -rw-rw-r--    10011 2020-04-09 17:31 feature_tests.cxx
37 -rw-rw-r--     6639 2020-04-09 17:31 Makefile.cmake
38 -rw-rw-r--       171 2020-04-09 17:31 TargetDirectories.txt
39 13 files in total.
40
41 ../build/CMakeFiles/list.dir:
42 -rw-rw-r--       524 2020-04-09 17:31 C.includecache
43 -rw-rw-r--       261 2020-04-09 17:31 cmake_clean.cmake
44 -rw-rw-r--        94 2020-04-09 17:31 link.txt
45 -rw-rw-r--    3136 2020-04-09 17:31 getopt.c.o
46 -rw-rw-r--       290 2020-04-09 17:31 depend.make
47 -rw-rw-r--       320 2020-04-09 17:31 depend.internal
48 -rw-rw-r--       171 2020-04-09 17:31 flags.make
49 -rw-rw-r--    7424 2020-04-09 17:31 list.c.o
50 -rw-rw-r--    5633 2020-04-09 17:31 build.make
51 -rw-rw-r--        64 2020-04-09 17:31 progress.make
52 -rw-rw-r--       657 2020-04-09 17:31 DependInfo.cmake
53 11 files in total.
54
55 ../build/CMakeFiles/CMakeTmp:
56 0 files in total.
57
58 ../build/CMakeFiles/3.10.2:
59 -rw-r--r--       402 2020-04-09 17:31 CMakeSystem.cmake
60 -rwxrwxr-x    8248 2020-04-09 17:31 CMakeDetermineCompilerABI_C.bin
61 -rw-r--r--     4849 2020-04-09 17:31 CMakeCXXCompiler.cmake
62 drwxrwxr-x     4096 2020-04-09 17:31 CompilerIdC

```

```

63 -rwxrwxr-x      8264 2020-04-09 17:31 CMakeDetermineCompilerABI_CXX.
    bin
64 drwxrwxr-x      4096 2020-04-09 17:31 CompilerIdCXX
65 -rw-r--r--      2219 2020-04-09 17:31 CMakeCCompiler.cmake
66 7 files in total.
67
68 ../build/CMakeFiles/3.10.2/CompilerIdC:
69 -rwxrwxr-x      8408 2020-04-09 17:31 a.out
70 drwxrwxr-x      4096 2020-04-09 17:31 tmp
71 -rw-rw-r--     18076 2020-04-09 17:31 CMakeCCompilerId.c
72 3 files in total.
73
74 ../build/CMakeFiles/3.10.2/CompilerIdCXX:
75 -rw-rw-r--     17631 2020-04-09 17:31 CMakeCXXCompilerId.cpp
76 -rwxrwxr-x      8416 2020-04-09 17:31 a.out
77 drwxrwxr-x      4096 2020-04-09 17:31 tmp
78 3 files in total.
79
80 ../build/CMakeFiles/3.10.2/CompilerIdC/tmp:
81 0 files in total.
82
83 ../build/CMakeFiles/3.10.2/CompilerIdCXX/tmp:
84 0 files in total.

```

3.3 添加限制

有限制的输出某一文件夹的内容时结果如下：

```

1 name1e5s@sumeru:~/Homework-2/build$ ./list -l 5000 -m 2 .
2 .:
3 -rwxrwxr-x     13832 2020-04-09 17:31 list
4 -rw-rw-r--      5304 2020-04-09 17:31 Makefile
5 -rw-rw-r--     12576 2020-04-09 17:31 CMakeCache.txt
6 5 files in total.

```