

Dhvanil Shah
Professor Hakner
ECE-357
01 October 2018

Problem Set 2 – Question 3

My Find (no arguments)

```
[Dhvanils-MacBook-Pro:prog2 dhvanil$ ./find
21511341 0 drwxr-xr-x 9 dhvanil staff 288 Oct 1 15:14 .
22048813 0 -rw-r--r-- 1 dhvanil staff 0 Oct 1 15:46 ./myFind.txt
22093085 16 -rw-r--r-- 1 dhvanil staff 4780 Oct 1 15:14 ./find.o
21523998 8 -rw-r--r-- 1 dhvanil staff 188 Sep 29 22:22 ./Makefile
22048781 0 -rw-r--r-- 1 dhvanil staff 0 Oct 1 15:46 ./macFind.txt
22009407 0 drwxr-xr-x 6 dhvanil staff 192 Sep 29 14:54 ./testfs
22009439 0 drwxr-xr-x 4 dhvanil staff 128 Sep 29 12:38 ./testfs/dir2
22009634 8 -rw-r--r-- 1 dhvanil staff 18 Sep 29 12:38 ./testfs/dir2/hellodir2.txt
22017819 0 lrwxr-xr-x 1 dhvanil staff 13 Sep 29 15:14 ./testfs/dir2/linkto1 -> ./testfs/dir1
22009554 8 -rw-r--r-- 2 dhvanil staff 18 Sep 29 12:37 ./testfs/hardlinkto1
22009432 0 drwxr-xr-x 4 dhvanil staff 128 Sep 29 12:38 ./testfs/dir1
22009598 0 drwxr-xr-x 3 dhvanil staff 96 Sep 29 12:38 ./testfs/dir1/dirinhello
22009614 8 -rw-r--r-- 1 dhvanil staff 35 Sep 29 12:38 ./testfs/dir1/dirinhello/hellodirNest.txt
22009554 8 -rw-r--r-- 2 dhvanil staff 18 Sep 29 12:37 ./testfs/dir1/hellodir1.txt
22009472 8 -rw-r--r-- 1 dhvanil staff 11 Sep 29 12:37 ./testfs/hello.txt
22093086 32 -rw-r--r-- 1 dhvanil staff 13776 Oct 1 15:14 ./find
21522914 16 -rw-r--r-- 1 dhvanil staff 4319 Oct 1 15:14 ./find.c
Dhvanils-MacBook-Pro:prog2 dhvanil$
```

My Find (directory as argument)

```
[Dhvanils-MacBook-Pro:prog2 dhvanil$ ./find testfs
22009407 0 drwxr-xr-x 6 dhvanil staff 192 Sep 29 14:54 testfs
22009439 0 drwxr-xr-x 4 dhvanil staff 128 Sep 29 12:38 testfs/dir2
22009634 8 -rw-r--r-- 1 dhvanil staff 18 Sep 29 12:38 testfs/dir2/hellodir2.txt
22017819 0 lrwxr-xr-x 1 dhvanil staff 13 Sep 29 15:14 testfs/dir2/linkto1 -> ./testfs/dir1
22009554 8 -rw-r--r-- 2 dhvanil staff 18 Sep 29 12:37 testfs/hardlinkto1
22009432 0 drwxr-xr-x 4 dhvanil staff 128 Sep 29 12:38 testfs/dir1
22009598 0 drwxr-xr-x 3 dhvanil staff 96 Sep 29 12:38 testfs/dir1/dirinhello
22009614 8 -rw-r--r-- 1 dhvanil staff 35 Sep 29 12:38 testfs/dir1/dirinhello/hellodirNest.txt
22009554 8 -rw-r--r-- 2 dhvanil staff 18 Sep 29 12:37 testfs/dir1/hellodir1.txt
22009472 8 -rw-r--r-- 1 dhvanil staff 11 Sep 29 12:37 testfs/hello.txt
Dhvanils-MacBook-Pro:prog2 dhvanil$
```

My Find Error Catching

```
[Dhvanils-MacBook-Pro:prog2 dhvanil$ ./find testfs extraarg
Error: Arguments Invalid. Correct format is './find [filepath]'
[Dhvanils-MacBook-Pro:prog2 dhvanil$ ./find nulldir
Unable to open directory [nulldir]: Error code 2: No such file or directory
Dhvanils-MacBook-Pro:prog2 dhvanil$
```

My Find Compared with “find” Command (with and without “diff”)

```
[Dhvanils-MacBook-Pro:prog2 dhvanil$ ./find testfs >> myfind.txt
[Dhvanils-MacBook-Pro:prog2 dhvanil$ find testfs -ls >> macFind.txt
[Dhvanils-MacBook-Pro:prog2 dhvanil$ diff myFind.txt macFind.txt
[Dhvanils-MacBook-Pro:prog2 dhvanil$
[Dhvanils-MacBook-Pro:prog2 dhvanil$
[Dhvanils-MacBook-Pro:prog2 dhvanil$ ./find testfs
22009407      0 drwxr-xr-x   6 dhvanil      staff      192 Sep 29 14:54 testfs
22009439      0 drwxr-xr-x   4 dhvanil      staff      128 Sep 29 12:38 testfs/dir2
22009634      8 -rw-r--r--   1 dhvanil      staff      18 Sep 29 12:38 testfs/dir2/hellodir2.txt
22017819      0 lrwxr-xr-x   1 dhvanil      staff      13 Sep 29 15:14 testfs/dir2/linkto1 -> ./testfs/dir1
22009554      8 -rw-r--r--   2 dhvanil      staff      18 Sep 29 12:37 testfs/hardlinkto1
22009432      0 drwxr-xr-x   4 dhvanil      staff      128 Sep 29 12:38 testfs/dir1
22009598      0 drwxr-xr-x   3 dhvanil      staff      96 Sep 29 12:38 testfs/dir1/dirinhello
22009614      8 -rw-r--r--   1 dhvanil      staff      35 Sep 29 12:38 testfs/dir1/dirinhello/hellodirNest.txt
22009554      8 -rw-r--r--   2 dhvanil      staff      18 Sep 29 12:37 testfs/dir1/hellodir1.txt
22009472      8 -rw-r--r--   1 dhvanil      staff      11 Sep 29 12:37 testfs/hello.txt
[Dhvanils-MacBook-Pro:prog2 dhvanil$ find testfs -ls
22009407      0 drwxr-xr-x   6 dhvanil      staff      192 Sep 29 14:54 testfs
22009439      0 drwxr-xr-x   4 dhvanil      staff      128 Sep 29 12:38 testfs/dir2
22009634      8 -rw-r--r--   1 dhvanil      staff      18 Sep 29 12:38 testfs/dir2/hellodir2.txt
22017819      0 lrwxr-xr-x   1 dhvanil      staff      13 Sep 29 15:14 testfs/dir2/linkto1 -> ./testfs/dir1
22009554      8 -rw-r--r--   2 dhvanil      staff      18 Sep 29 12:37 testfs/hardlinkto1
22009432      0 drwxr-xr-x   4 dhvanil      staff      128 Sep 29 12:38 testfs/dir1
22009598      0 drwxr-xr-x   3 dhvanil      staff      96 Sep 29 12:38 testfs/dir1/dirinhello
22009614      8 -rw-r--r--   1 dhvanil      staff      35 Sep 29 12:38 testfs/dir1/dirinhello/hellodirNest.txt
22009554      8 -rw-r--r--   2 dhvanil      staff      18 Sep 29 12:37 testfs/dir1/hellodir1.txt
22009472      8 -rw-r--r--   1 dhvanil      staff      11 Sep 29 12:37 testfs/hello.txt
[Dhvanils-MacBook-Pro:prog2 dhvanil$ █
```

*symbolic link used is sufficient for demonstration purposes but is an invalid link because it links a file to a directory

```

1 #include <fcntl.h>
2 #include <stdlib.h>
3 #include <stdio.h>
4 #include <unistd.h>
5 #include <string.h>
6 #include <errno.h>
7 #include <sys/types.h>
8 #include <sys/stat.h>
9 #include <dirent.h>
10 #include <pwd.h>
11 #include <grp.h>
12 #include <time.h>
13
14 void throwError(char *message, char *file)
15 {
16     if (file)
17         fprintf(stderr, "%s [%s]: Error code %i: %s\n", message, file, errno,
strerror(errno));
18     else
19         fprintf(stderr, "%s\n", message);
20     exit(-1);
21 }
22
23 char *getPermissions(mode_t mode)
24 {
25     char *p;
26     p = (char *)malloc(10);
27     if (p == NULL)
28         throwError("Error: Failure to dynamically allocate memory.", NULL);
29     p[0] = '-';
30     p[1] = (mode & S_IRUSR) ? 'r' : '-';
31     p[2] = (mode & S_IWUSR) ? 'w' : '-';
32     p[3] = (mode & S_IXUSR) ? 'x' : '-';
33     p[4] = (mode & S_IRGRP) ? 'r' : '-';
34     p[5] = (mode & S_IWGRP) ? 'w' : '-';
35     p[6] = (mode & S_IXGRP) ? 'x' : '-';
36     p[7] = (mode & S_IROTH) ? 'r' : '-';
37     p[8] = (mode & S_IWOTH) ? 'w' : '-';
38     p[9] = (mode & S_IXOTH) ? 'x' : '-';
39     return p;
40 }
41
42 void readDir(char *directory)
43 {
44     DIR *dir;
45     struct dirent *sd;
46     struct stat buf;
47     struct group *grp;
48     struct passwd *usr;
49 }

```

```

47
50 struct tm fileTime, nowTime;
51 char path[4096], link[4096], date[80], *group, *user, *perms;
52 mode_t mode;
53 time_t now = time(NULL);
54
55 dir = opendir(directory);
56 if (dir == NULL)
57 {
58     throwError("Error: Unable to open directory", directory);
59 }
60
61 while ((sd = readdir(dir)) != NULL)
62 {
63     snprintf(path, sizeof(path), "%s/%s", directory, sd->d_name);
64     if (lstat(path, &buf) < 0)
65     {
66         throwError("Error: Couldn't get stats for a path/file", path);
67     };
68
69     if ((grp = getgrgid(buf.st_gid)) == NULL)
70     {
71         throwError("Error: Could not get the group associated with a
72 file/directroy", directory);
73     };
74
75     if ((usr = getpwuid(buf.st_uid)) == NULL)
76     {
77         throwError("Error: Could not get the user associated with a
78 file/directroy", directory);
79     };
80
81     group = grp->gr_name;
82     user = usr->pw_name;
83     mode = buf.st_mode;
84
85     perms = getPermissions(mode);
86
87     localtime_r(&buf.st_mtime, &fileTime);
88     localtime_r(&now, &nowTime);
89     if (fileTime.tm_year == nowTime.tm_year)
90     {
91         strftime(date, sizeof(date), "%b %e %H:%M", &fileTime);
92     }
93     else
94     {
95         strftime(date, sizeof(date), "%b %e %Y", &fileTime);
96     }
97
98     if (S_ISDIR(mode))
99     {

```

```
97     {
98         if ((strcmp(sd->d_name, "..") != 0) && (strcmp(sd->d_name, ".")
!= 0))
99         {
100             readDir(path);
101         }
102         else if (strcmp(sd->d_name, ".") == 0)
103         {
104             perms[0] = 'd';
105             printf("%llu%9lli %s%5hu %s%15s%20lli %s %s\n", buf.st_ino,
buf.st_blocks, perms, buf.st_nlink, user, group, buf.st_size, date,
directory);
106         }
107     }
108     else if (S_ISREG(mode))
109     {
110         perms[0] = '-';
111         printf("%llu%9lli %s%5hu %s%15s%20lli %s %s\n", buf.st_ino,
buf.st_blocks, perms, buf.st_nlink, user, group, buf.st_size, date, path);
112     }
113     else if (S_ISLNK(mode))
114     {
115         perms[0] = 'l';
116         ssize_t len = readlink(path, link, 4095);
117         if (len < 0)
118         {
119             throwError("Error: Could not read path of symbolic link",
path);
120         }
121         else
122         {
123             link[len] = '\0';
124         }
125         printf("%llu%9lli %s%5hu %s%15s%20lli %s %s -> %s\n", buf.st_ino,
buf.st_blocks, perms, buf.st_nlink, user, group, buf.st_size, date, path,
link);
126     }
127     else
128     {
129         throwError("Error: This type of file is not yet supported",
NULL);
130     }
131
132     free(perms);
133 }
134 if (closedir(dir) < 0)
135 {
136     throwError("Error: Unable to close directory", directory);
137 }
138 }
```

```
139
140 int main(int argc, char *argv[])
141 {
142
143     char *directory = ".";
144
145     if (argc == 1)
146     {
147         directory = ".";
148     }
149     else if (argc == 2)
150     {
151         directory = argv[1];
152     }
153     else
154     {
155         throwError("Error: Arguments Invalid. Correct format is './find
156 [filepath]'", NULL);
157     }
158     readDir(directory);
159     return 0;
160 }
```