Gebze Technical University
CSE344 Systems Programming Course
-
HOMEWORK 1 REPORT

Muhammed OZKAN 151044084

March 30, 2021

Abstract

We are expected to write an "advanced" file search program for POSIX compatible operating systems. Our program should be able to search for files that meet the given criteria and print the results in a nicely formatted tree format.

1 Overview

The homework I submitted is a complete assignment. It supports all the requirements and parameters requested in the homework document and searches the files with a recursive search according to the given search criteria. It displays the found files as output in a proper tree structure. It is POSIX compatible.

2 How did I solve this problem.

In order to solve the problem, first of all, the parameters sent to our program had to be properly separated and analyzed. I used getopt () function in unix standard c library for this. By doing this, I was able to parse without any problems regardless of the order of the parameters sent. I checked whether the values I received with the getopt function meet the criteria of our program. I have reported the values that are not suitable or incorrect to the user with the help of stderr. In this way, the first phase of our program has been completed.

Then, as the second step, we send the values we receive from the user to our search function, which will run recursively. Here, the pre order traverse function, which somehow browses the folders, goes into each folder it finds and checks with another auxiliary function that checks whether the files in it meet the search criteria given by the user. With the help of the function, files matching the criteria are displayed to the user in a properly tree structure.

2.1 Some more details

To give a little more detail about the program structure. The dirent.h library, which uses system calls to view and navigate the filesystem contents, is used. These are opendir() readdir() and closedir(). The sys / stat.h library was used for the properties of the files, the errno.h library for the informative error notes, and the signal.h libraries to capture the CTRL-C input.

3 My design decisions.

Our program consists of 4 different functions. It is responsible for sending to the next step, listDirectory() where main() parameters are checked and received. listDirectory() on the other hand, starts from the given starting position, recursively circulates all contents in it. It uses the isValid() function to check whether each content meets the given criteria. The isValid() function checks the criteria received by main() one by one and returns 0 if not even one of the criteria is appropriate. Returns 1 if it appropriate the criteria. Contents that meet the criteria are pressed on the screen to be displayed to the user in a proper tree structure in accordance with the levels calculated by the treeLevel() function, according to their order in listDirectory().

4 What requirements did I meet and which did I fail?

All requirements are fully fulfilled. Written below.

The program will additionally receive as mandatory parameter:
-w: the path in which to search recursively (i.e. across all of its subtrees)

The search criteria can be any combination of the following (at least one of them must be employed):
-f : filename (case insensitive), supporting the following regular expression: + (regex)
-b : file size (in bytes)
-t : file type (d: directory, s: socket, b: block device, c: character device f: regular file, p: pipe, l: symbolic link)
-p : permissions, as 9 characters (e.g. rwxr-xr–)

-l: number of links

In case of CTRL-C the program must stop execution, return all resources to the system and exit with an information message.

If no file satisfying the search criteria has been found, then simple output No file found. All error messages are to be printed to stderr. All system calls are to be checked for failure and the user is to be notified accordingly.

Output
In the form of nicely formatted tree

5 Tests

```
./myFind -w /home -f makefile
/home
—–cse312
——-Desktop
———spimsimulator-code-r739
———–spim
————-Makefile
———–xspim
————-Makefile
———system
———–Makefile
```

```
./myFind -w /home -f lost+file -t d
/home
—–cse312
——-Desktop
———lostttfile
```

```
./myFind -w /home -f lostfile -t d
No file found.
```

```
./myFind -w /home -f lost+file -b 4096
/home
—–cse312
——-Desktop
———lostttfile
```

```
./myFind -w /home -f rd.c -t f -l 1 -p rw-rw-r– -b 2267
/home
—cse312
—–-Desktop
——system1
——–rd.c
```

```
./myFind -w /home -f r+d.c -t f -l 1 -p rw-rw-r– -b 2267
/home
—cse312
—–-Desktop
——system1
——–rd.c
```

```
./myFind
The ’-w’ parameter is required. Program terminate
```

```
./myFind -w /
You must use at least one parameter. -f -b -t -p -l. Program terminate
```

```
./myFind -w / -l
./myFind: option requires an argument – ’l’
Option ’-l’ requires an parameter. Program terminate
```

```
./myFind -w / -f asd
Cannot open directory ’//root’: Permission denied. Program continue...
Cannot open directory ’//var/log/speech-dispatcher’: Permission denied. Program continue...
Cannot open directory ’//var/spool/rsyslog’: Permission denied. Program continue...
Cannot open directory ’//var/spool/cups’: Permission denied. Program continue...
Cannot open directory ’//var/spool/cron/crontabs’: Permission denied. Program continue...
Cannot open directory ’//var/lib/sudo’: Permission denied. Program continue...
Cannot open directory ’//var/lib/lightdm’: Permission denied. Program continue...
Cannot open directory ’//var/lib/polkit-1’: Permission denied. Program continue...
Cannot open directory ’//var/lib/udisks2’: Permission denied. Program con-
```

tinue...
Cannot open directory '//var/lib/lightdm-data/lightdm': Permission denied. Program continue...
Cannot open directory '//var/cache/lightdm/dmrc': Permission denied. Program continue...
Cannot open directory '//var/cache/ldconfig': Permission denied. Program continue...
^{C}

$Program\ terminated\ because\ CTRL - C\ keys\ were\ pressed.$