CSE 344: Homework #1

Türker Tercan
turker.tercan2017@gtu.edu.tr

Gebze Technical University — March 31, 2021

Objective

You are expected to write an “advanced” file search program for POSIX compatible operating systems. Your program must be able to search for files satisfying the given criteria, and print out the results in the form of a nicely formatted tree.

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

Example

./myFind -w targetDirectoryPath -f ‘lost+file‘ -b 100 -t b

means it will search in the targetDirectoryPath path, for block device files named lost+file of size exactly 100 bytes

Output

In the form of nicely formatted tree

Command Line

```
targetDirectoryPath
|--subDirectory
|----subDirectory2
|------subDirectory3
|--------LOStttttttFile
|--subDirectory6
|----lOstFile
```

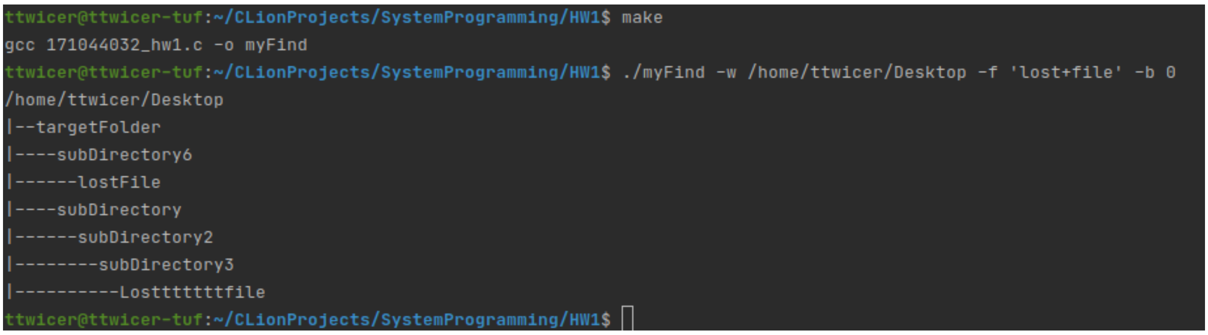
If no file satisfying the search criteria has been found, then simple output “No file found”. All error messages are to be printed to stderr. All system calls are to be checked for failure and the user is to be notified accordingly.

Implementation

In order to search for files in the directories recursively, i used opendir(), readdir() and closedir() functions in the dirent.h library which supported by POSIX. I used the perror() function as much as possible against all possible errors. Always used syscalls for write(). I was not sure are we allowed to use string.h library, so I implemented my own functions which I need inside of string.h. The program will print usage information every time if a command line argument is missing or invalid. For outputting the result, I implemented a simple n-child linked list for representing the directories and files. And I implemented my program with below libraries:

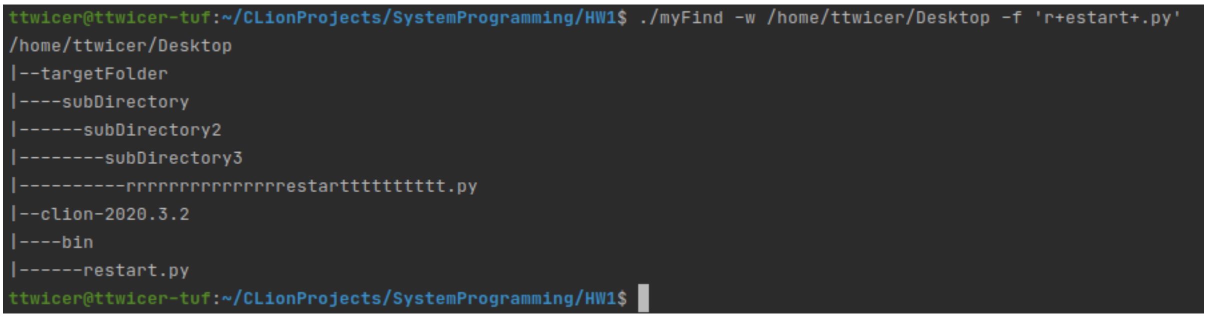
- stdio.h
- stdlib.h
- unistd.h
- sys/stat.h
- dirent.h

Output Result - 1



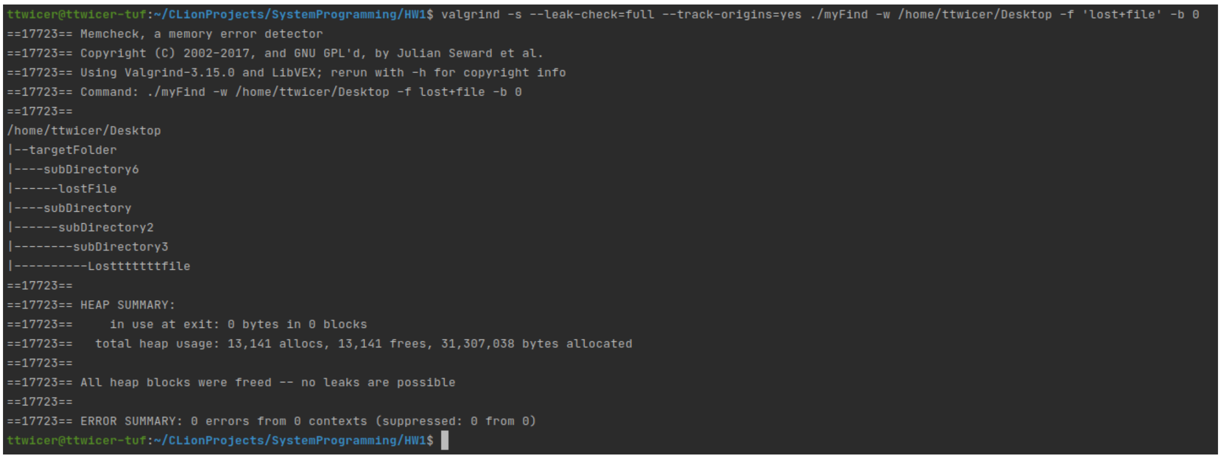
```
ttwicer@ttwicer-tuf:~/CLionProjects/SystemProgramming/HW1$ make
gcc 171044032_hw1.c -o myFind
ttwicer@ttwicer-tuf:~/CLionProjects/SystemProgramming/HW1$ ./myFind -w /home/ttwicer/Desktop -f 'lost+file' -b 0
/home/ttwicer/Desktop
|--targetFolder
|----subDirectory6
|------lostFile
|----subDirectory
|------subDirectory2
|--------subDirectory3
|----------Lostttttttfile
ttwicer@ttwicer-tuf:~/CLionProjects/SystemProgramming/HW1$ 
```

Output Result - 2

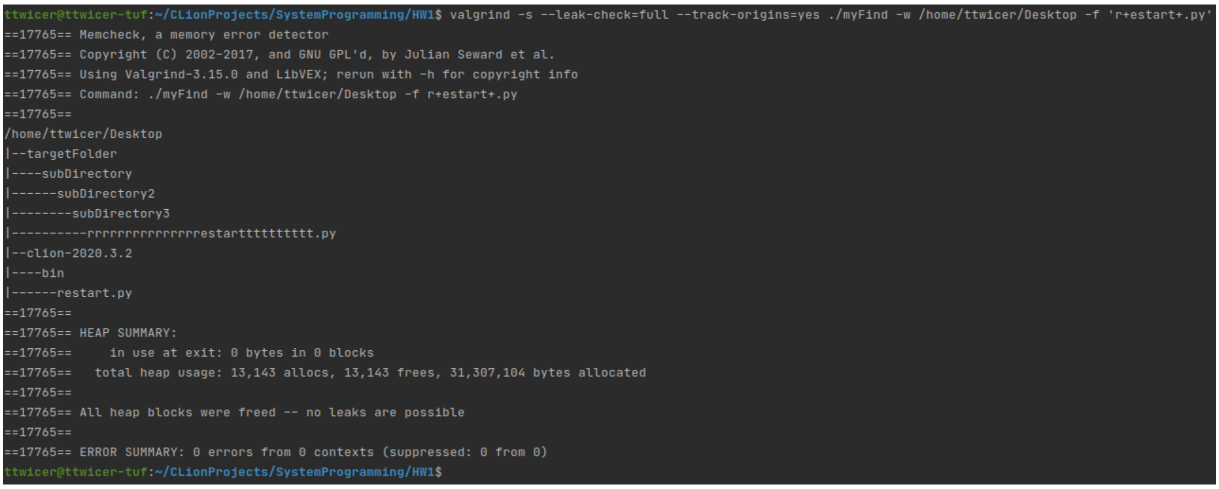


```
ttwicer@ttwicer-tuf:~/CLionProjects/SystemProgramming/HW1$ ./myFind -w /home/ttwicer/Desktop -f 'r+estart+.py'
/home/ttwicer/Desktop
|--targetFolder
|----subDirectory
|------subDirectory2
|--------subDirectory3
|----------rrrrrrrrrrrrrrrestartttttttttt.py
|--clion-2020.3.2
|----bin
|------restart.py
ttwicer@ttwicer-tuf:~/CLionProjects/SystemProgramming/HW1$ 
```

Checking For Memory Leaks



```
ttwicer@ttwicer-tuf:~/CLionProjects/SystemProgramming/HW1$ valgrind -s --leak-check=full --track-origins=yes ./myFind -w /home/ttwicer/Desktop -f 'lost+file' -b 0
==17723== Memcheck, a memory error detector
==17723== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==17723== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==17723== Command: ./myFind -w /home/ttwicer/Desktop -f lost+file -b 0
==17723==
/home/ttwicer/Desktop
|--targetFolder
|----subDirectory6
|------lostFile
|----subDirectory
|------subDirectory2
|--------subDirectory3
|----------Lostttttttfile
==17723==
==17723== HEAP SUMMARY:
==17723==     in use at exit: 0 bytes in 0 blocks
==17723==   total heap usage: 13,141 allocs, 13,141 frees, 31,307,038 bytes allocated
==17723==
==17723== All heap blocks were freed -- no leaks are possible
==17723==
==17723== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
ttwicer@ttwicer-tuf:~/CLionProjects/SystemProgramming/HW1$
```



```
ttwicer@ttwicer-tuf:~/CLionProjects/SystemProgramming/HW1$ valgrind -s --leak-check=full --track-origins=yes ./myFind -w /home/ttwicer/Desktop -f 'r+estart+.py'
==17765== Memcheck, a memory error detector
==17765== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==17765== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==17765== Command: ./myFind -w /home/ttwicer/Desktop -f r+estart+.py
==17765==
/home/ttwicer/Desktop
|--targetFolder
|----subDirectory
|------subDirectory2
|--------subDirectory3
|----------rrrrrrrrrrrrrrestarttttttttttt.py
|--clion-2020.3.2
|----bin
|------restart.py
==17765==
==17765== HEAP SUMMARY:
==17765==     in use at exit: 0 bytes in 0 blocks
==17765==   total heap usage: 13,143 allocs, 13,143 frees, 31,307,104 bytes allocated
==17765==
==17765== All heap blocks were freed -- no leaks are possible
==17765==
==17765== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
ttwicer@ttwicer-tuf:~/CLionProjects/SystemProgramming/HW1$
```