

a) How am I Solved This Problem

Steps: 1- First, I divide problem into chunks, 2- I started with the easy stuff such as reading command line options with getopt (), writing filename comparator with ability to support regex etc. 3- Then I wrote crucial function for homework to check are the attributes of files match between target file data and traversed file data. 4- After all these things are ready. I wrote a function to traverse all the files within specified pathname recursively, I can print all files on the screen with this function but it's hard to print the 'found' files. Because of the need of memorizing all the found files programmatically. 5- I hold the full pathnames of found files in a 2D array in mentioned above function (step 4), I have decided to write separate function to print all the tree structure with the found files. 6- Two helper functions are created to complete printing of found files on the screen. 7- Then I made minor corrections on the functions like case-insensitivity. 8- I worked on memory management for hours, because when I searched for filetype as f (-t f) in the root directory, it found 320 thousand files on my system and I wanted to print all of these on the screen in a tree structure. I believe, finally I optimized memory requirement of my program.

b) Design Decisions 1) While traversing file system, my program compare each file with the sought file, and if there is a match which we are looking for, program stored found file's pathname into a 2D char array. when the all traversing process has ended, Program starts printing those found files with the help of their pathnames in 2D char array. 2) In printing process, firstly, program prints first found file, Meanwhile, I kept the printed dir names and filenames in 3D char array. If the next found file has one of the dir names printed on the screen before, it will not print it again, it will print "-" instead. 3) I had to keep the arrays in the heap in the recursive function, since it was recursive, stack was overflowing.

c) Requirements Satisfied 1) There is no warning with respect to -Wall flag 2) If the required command line arguments are missing/invalid, my program prints usage information and exits.

```
musta@ubuntu:~/Desktop/151044028$ ./151044028 -w pathname
Usage Error:
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
• -w: the path in which to search recursively
*** -w option is mandantory
musta@ubuntu:~/Desktop/151044028$
```

3) There is no memory leak with respect to valgrind

```
musta@ubuntu:~/Desktop/151044028$ valgrind --leak-check=full --show-leak-kinds=all ./151044028 -w / -f folder2
==10946== Memcheck, a memory error detector
==10946== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==10946== Using Valgrind-3.11.0 and LibVEX; rerun with -h for copyright info
==10946== Command: ./151044028 -w / -f folder2
==10946==
1 files are found

/
|--home
|----musta
|-----Desktop
|-----folder1
|-----folder2
==10946==
==10946== HEAP SUMMARY:
==10946==      in use at exit: 0 bytes in 0 blocks
==10946==    total heap usage: 362,053 allocs, 362,053 frees, 1,465,377,136 bytes allocated
==10946==
==10946== All heap blocks were freed -- no leaks are possible
==10946==
==10946== For counts of detected and suppressed errors, rerun with: -v
==10946== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
musta@ubuntu:~/Desktop/151044028$
```

4) Shows Results as a tree structure according to specified option, green ones found files

```
musta@ubuntu:~/Desktop/151044028$ ./151044028 -w / -t s
29 files are found

/
|--run
|----NetworkManager
|-----private-dhcp
|----vmware
|-----guestServicePipe
|----avahi-daemon
|----socket
|----acpid.socket
|----cups
|-----cups.sock
|----dbus
|-----system_bus_socket
|----snapd-snap.socket
|----snapd.socket
|----uuid
|-----request
|----user
|-----1000
|-----vscode-5322c186-1.54.3-main.sock
|-----pulse
|-----native
|-----keyring
|-----pkcs11
|-----ssh
|-----control
|-----snapd-session-agent.socket
|----systemd
|-----private
|-----notify
|----systemd
|-----fsck.progress
|----journal
|-----socket
|-----stdout
|-----syslog
|-----dev-log
|-----private
|-----cgroups-agent
|-----notify
|----udev
|-----control
|--tmp
|----X11-unix
|----X0
|----ICE-unix
|----1552
|--home
|----musta
|-----gnupg
|-----5.gpg-agent
musta@ubuntu:~/Desktop/151044028$
```

5) regexp and case insensitivity works fine

```
|----LOStTFile
musta@ubuntu:~/Desktop/151044028$ ./151044028 -w .. -f lost+file+
5 files are found

..
|--lostfile
|--folder1
|----LOSTtfile
|----LOSTFILEE
|----lostfileeee
|----LOStTFile
musta@ubuntu:~/Desktop/151044028$
```

6) SIGINT handler works as expected

```
|-----libgtk-3-bin.list
|-----python3-oauthlib.list
|-----libtelepathy-glib0:amd64.md5sums
|-----hplip.preinst
|-----update-notifier-common.postrm
|-----python3.prerm
|-----base-files.list
|-----libkrb5support0:amd64.shlibs
|-----libreoffice-pdfimport.list
|-----bamfdaemon.list
|-----zlib1g:amd64.list
|-----libdevmapper1.02.1:amd64.md5sums
|-----xserver-xorg-video-intel-hwe-16.04.triggers
|-----libreoffice-avmedia-backend-gstreamer.prerm
|-----libjson-glib-1.0-common.list
|-----python-talloc.list
|-----libgeis1:amd64.symbols
|-----libnspr4:amd64.md5sums
|-----appstream.md5sums
|-----gir1.2-goa-1.0:amd64.md5sums
|-----python3-dbus.md5sums
|-----libglewmx1.13:amd64.list
|-----fonts-tlwg-typo-ttf.md5sums
|-----sysv-rc.postrm
|-----gnome-mahjongg.md5sums
^C|-----zenity.list
SIGINT arrived exiting program
musta@ubuntu:~/Desktop/151044028$
```

Unsatisfied 1- finding symbolic links with ./program -w / -t 1, I block symbolic links because it causes so much repetition of paths but when I block them I cant found soft links with above commands.