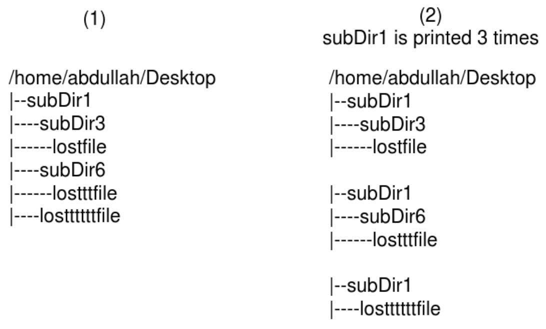
# CSE344 - HW1 Report

Abdullah Çelik

March 2021

## 1 Overview

- The set_arguments () function separates the command line arguments using the getopt() function. After doing this, the suitability of the parameters is tested (mentioned in the error handling section). In case of an error, the error message is returned. If everything goes well, NULL is returned. In this way, the user will be informed with an error message. Parameters are assigned to global variables in Srch_Basis.h.
- The findByCriteria() function calls the function (find()) that will call the recursion. If there is an I / O error, -1 is returned and the error is printed in the screen with the perror() function. With Exit (1) the program is terminated.
- The find() function makes recursion calls and starts looking for the appropriate file in the given path. Creates a tree structure from the paths of the files it finds. If it cannot find the file it is looking for, the tree will not be created and a message will be printed on the screen.
- A path tree is created containing all the files found. This allows us to write the parent of every file found over and over again. It provides a more understandable output to the user. You can see the difference in the figure below.



(1)

```
/home/abdullah/Desktop
|--subDir1
|----subDir3
|------lostfile
|----subDir6
|------lostttfile
|----losttttttfile
```

(2)
subDir1 is printed 3 times

```
/home/abdullah/Desktop
|--subDir1
|----subDir3
|------lostfile

|--subDir1
|----subDir6
|------lostttfile

|--subDir1
|----losttttttfile
```

## 2 Error Handling

- CTRL + C
  Description: The case where the user presses CTRL and C combination.
  Action: Program gives all resources to memory and terminates itself.

- The length of the given file name
  Description: The case where the length of the file name given as a command line argument is greater than 255.
  Action: "Error! The length of the file name cannot be greater than 255." message is printed to stderr and the program is terminated.

- The type of given file size
  Description: The case where the file size given as a command line argument is not integer.
  Action: "Error! The file size must be integer." message is printed to stderr and the program is terminated.

- Option of the file type
  Description: The case where the file type given as a command line argument is invalid such as -t x.
  Action: "Error! Wrong file type." message is printed to stderr and the program is terminated.

- The permissions parameter
  Description: The case where the permissions gives as a command line argument is not 9 characters or it is invalid.
  Action: "Error! Incorrect permission." message is printed to stderr and the program is terminated.

- The type of number of links
  Description: The case where the number of links given as a command line argument is not integer.
  Action: "Error! Incorrect number of links." message is printed to stderr and program is terminated.

- Not using any operator
  Description: The case where the search criteria does not contain any combination of parameters.
  Action: "Error! An operator must be used." message is printed to stderr and program is terminated.

- Not using -w parameter
  Description: The case where the mandatory path parameter does not received.
  Action: "Error! The -w parameter is mandatory." message is printed to stderr and program is terminated.

- Using undefined operation
  Description: The case where the undefined operation is used such as -x.
  Action: "Error! Undefined operation." message is printed to stderr and program is terminated.

- Using additional parameters
  Description: The case where the additional parameter is used such as -f file additionalParameter.
  Action: "Error! Additional parameter is founded." message is printed and program is terminated.

- Incorrect use of regexp
  Description: The case where the file name contains a regexp in the first character.
  Action: "Error! Incorrect usage of regexp." message is printed to stderr and program is terminated.

- Using opendir function
  Description: The case where the opendir function returns NULL (this means an error is occurred) after calling (inside of findByCriteria function).
  Action: If the value returned by the opendir is checked and an error occurs, the error is printed with the perror function and the program is terminated with exit(1).

- Using readdir function
  Description: The case where the readdir function returns NULL (The readdir function return NULL in two cases. First case is If the end of the directory stream is reached, NULL is returned and errno is not changed. The second case if If an error occurs, NULL is returned an errno is set appropriately.) after calling (inside of findByCriteria function).
  Action: Before calling readdir function, errno is saved. Then after calling readdir function, it is checked whether errno is change or not. If there is a change, error is printed with perror function and program is terminated with exit(1).

- Using lstat function
  Description: The case where the lstat function returns -1 (this means an error is occurred) after calling (inside of findByCriteria function).
  Action: If the value returned by the lstat is checked and an error occurs, the error is printed with the perror function and the program is terminated with exit(1).

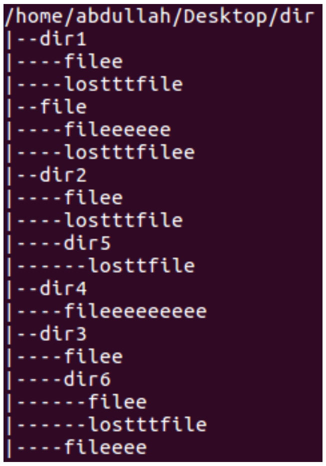
- Using closedir function
  Description: The case where the closedir function returns -1 (this means an error is occurred) after calling (inside of findByCriteria function).
  Action: If the value returned by the closedir is checked and an error occurs, the error is printed with the perror function and the program is terminated with exit(1).

## 3 Compile and Run

- make → Compiles the whole program
  Type make in the file contains the Makefile
- make clean → Cleans all objects files

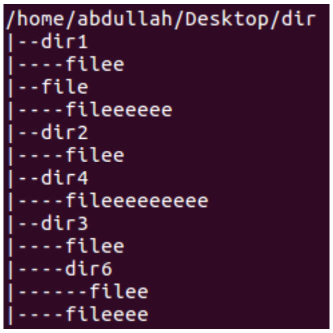
## 4 Testing

The file structure is as follows.



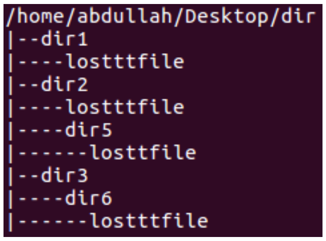
```
/home/abdullah/Desktop/dir
|--dir1
|----filee
|----lostttfile
|--file
|----fileeeeee
|----lostttfilee
|--dir2
|----filee
|----lostttfile
|----dir5
|------losttfile
|--dir4
|----fileeeeeeeee
|--dir3
|----filee
|----dir6
|------filee
|------lostttfile
|----fileeee
```

./myFind -w /home/abdullah/Desktop/dir1/ -f file+



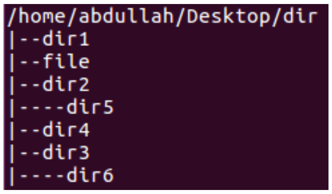
```
/home/abdullah/Desktop/dir
|--dir1
|----filee
|--file
|----fileeeeee
|--dir2
|----filee
|--dir4
|----fileeeeeeeee
|--dir3
|----filee
|----dir6
|------filee
|----fileeee
```

```
./myFind -w /home/abdullah/Desktop/dir -f lost+file
```



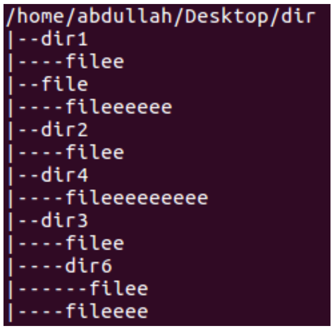
```
/home/abdullah/Desktop/dir
|--dir1
|----lostttfile
|--dir2
|----lostttfile
|----dir5
|------losttfile
|--dir3
|----dir6
|------lostttfile
```

```
./myFind -w /home/abdullah/Desktop/dir -t d
```



```
/home/abdullah/Desktop/dir
|--dir1
|--file
|--dir2
|----dir5
|--dir4
|--dir3
|----dir6
```

```
./myFind -w /home/abdullah/Desktop/dir -f file+ -b 0 -t f -p rw-rw-r– -l 1
```



```
/home/abdullah/Desktop/dir
|--dir1
|----filee
|--file
|----fileeeeee
|--dir2
|----filee
|--dir4
|----fileeeeeeeee
|--dir3
|----filee
|----dir6
|------filee
|----fileeee
```