

CSE344 - System Programming - Homework #1

Fatih Kaan Salgır - 171044009

1 Problem Solution Approach

I have defined 2 structs (`criteria` and `fileInfo`) and constructed by the command line parameters. `criteria` struct holds the data of whether user wants to filter by the property or not. Similarly `fileInfo` holds the corresponding data of the parameters.

```
struct criteria {
    bool filename;
    bool size;
    bool type;
    bool permissions;
    bool noLinks;
};

struct fileInfo {
    char *filename;
    int size;
    char type;
    char *permissions;
    int noLinks;
};
```

When desired file has been found, it's path should be printed. Therefore, when it is entered a directory, parent directories need to be saved. In order to keep track of parent directories, double-ended queue data structure has been used. Function of the queue will be as follows:

1. New directories will be added to front of the queue
2. If file has been found, then it's path will be printed starting from queue's rear, by removing one by one.
3. If file have not been found after traversing a directory, directory will be removed from queue which is located at the front of the queue.

-
- `getopt()` is used to parse command line arguments.
 - For symbolic links `lstat` is used. That means link itself is used to check if matched by search criteria, not the file that it refers to.
 - `dirent` pointer returned by `readdir` is not `free()`'d, because it is stated on the man page of the `readdir` that it may be statically allocated.

1.1 SIGINT Signal

To handle `Control-C` which is `SIGINT` signal, simple signal handler function implemented. Signal handler sets the global variable `sigint` to true. Condition of the while loop which searches files, is

also dependant to this variable. Therefore in case of **SIGINT**, searching stops, resources returned to system and error message is printed.

1.2 Error Handling

- In case of any errors while getting the file stats (calling **stat** or **lstat**), error message will be send to **stderr** and the search will continue with the next file instead of terminating the whole program.

2 System Calls

Following system calls are used:

- **opendir**: To open a directory, in order to get its contents.
- **readdir**: To read the contents of a directory, and getting the filenames.
- **stat/lstat**: To get file information: size, type, permissions and number of hard links. **stat** is used for regular files, **lstat** is used for links.

3 Regex Matching

1. Both **filename** and **regexString** converted to lower case to ensure case insensitivity.
2. + character converted into \+ in **filename** to be able to compare escaped + character. If both strings are equal then it is a match.
3. Strings are compared char by char. In **regexString** when it is encountered with + repated character is consumed from **filename**.
4. In order to handle situation where **regexString** includes preceding character right after + (e.g regex string is **lost+tile** where file name is **losttile**), the number of consumed character is counted. If it is less than minimum required repetation, then it is not a match.

This method fails to match when there are multiple occurances of the **same** regex one after another. (e.g. **regexString** is **lost+t+tile**)

4 Test Cases

Program tested with different cases and also with the **valgrind** to check memory leaks. According the output of the **valgrind**, "All heap blocks were freed – no leaks are possible".

```
valgrind --leak-check=full \  
         --show-leak-kinds=all \  
         --track-origins=yes \  
         --verbose \  
         --log-file=valgrind-out.txt \  
         ...
```

4.1 Invalid option

```
$ ./output -w ./test -x rwxrwxrwx
```

```
./program: invalid option -- 'x'
```

```
Usage: ./untitled2 -w targetDirectoryPath [-f filename] [-b file size] [-t file  
↪ type] [-p permissions] [-l number of links]
```

```
-f filename      Filename, case insensitive, supporting the following regular  
↪ expressin: +
```

```
-b file size     File size in bytes
```

```
-t file type     File type (d: directory, s: socket, b: block device, c:  
↪ character device f: regular file, p: pipe, l: symbolic link)
```

```
-p rwxr-xr--     Permissions, as 9 characters
```

```
-l number of links
```

```
-w path          The path in which to search recursively (i.e. across all of  
↪ its subtrees)
```

At least one of the search criteria must be employed.

4.2 Target directory not specified

Prints error message to `stderr`, and usage to `stdout`.

4.3 No search criteria specified

Prints error message to `stderr`, and usage to `stdout`.

4.4 With interrupt signal

Control-C (SIGINT) while listing all the directories. No memory leaks.

```
$ ./output -w / -t d
```

```
...(truncated)
```

```
|-----LC_MESSAGES
```

```
|-----zh_TW
```

```
|-----LC_MESSAGES
```

```
|-----mime
```

```
|-----application
```

```
^CProgram interrupted by signal: SIGINT
```