

Overview

The purpose of this assignment is twofold:

1. To reinforce your understanding of file systems, file types, permissions, and recursive directory structures
2. To teach you how the unix utility "find" works so that you can benefit from using it in your studies and career

Program

The program is called as:

```
./find directory [filters]
```

Required Filters

-d

evaluates to true and enables debugging

-name filename

evaluates to true if the file name is **exactly** "filename". Wildcards are not supported.

-type filetype

evaluates to true if the file is of type "filetype". You should support the following file types:

"f" - regular file

"d" - directory

"l" - symbolic link

"b" - block special (for devices)

"c" - character special (for devices)

-size sizespec

If the first character of "filespec" is a '-', it evaluates to true if the file is smaller than the size specified. If the first character of "filespec" is a '+', it evaluates to true if the file is larger than the size specified. If the first character of "filespec" is a digit, it evaluates to true if the file is exactly the size specified.

If the filespec ends in the character 'c', then the size is in characters and the size should be compared against st_size in the file's "struct stat". Otherwise the size is in "blocks" and should be compared against the file's st_blocks.

Extra Credit Filters

-ls

Always evaluates to true and uses the long output format for all file output. The output should be identical to the real find command.

`-mtime timespec`

evaluates to true if the modified time of the file matches timespec (see the "find" manual page).

`-mmin timespec`

evaluates to true if the modified time of the file matches timespec (see the "find" manual page).

Support wildcards in file names

Anything else that seems useful to you

If you do any of the extra-credit items, be sure to provide your own `gradeFile.EXTRA` input file so I know what works.

Hints and Help

1. I've provided 2 example programs that will give you a good headstart on working with both directories (`readdir.c`) and file information (`sstat.c`)
2. I've provided a tar file "testdir.tar". You should "untar" that file into your source directory and use that for testing. It contains lots of interesting cases. To "untar" that file into your current directory, type:

```
tar xvf testdir.tar
```

3. In all cases, if you're not sure what your program is supposed to do, the answer is that it should do exactly what the real "find" program does in that case.

Example Input File

(this is from "grademe.required")

```
./find testdir/dir1/subdir1
./find testdir
./find testdir -type d
./find testdir -type f
./find testdir -type f -size +100c
./find testdir -name gf100
./find testdir -type d -name subsubdir3
./find testdir -type f -name subsubdir3
```