



Assignment 4: *sfind*: A Simplified *find* Command

Task

The `find` command is a very powerful command. Its man page details all of the different ways it can be used. In this assignment, you'll write a simple version of `find` named `sfind` that implements just a few of its features.

The assignment is specified by the following man-page-like description. Your job is to write a program that satisfies the requirements of this description. The information regarding program submission is at the end of the assignment.

NAME

```
sfind -- find all files satisfying given test in given directory hierarchies
```

SYNOPSIS

```
sfind [dir1 dir2 ...] [test]
```

DESCRIPTION

The `sfind` command searches through the given directories for all files that satisfy the given test. If no directories are given, it uses the current working directory. For each file for which the test is true, it prints the relative pathname of the file with respect to the top level directory of the search. For example, if a file named `bar` in the tree rooted at `dir2` passes the test, the path that it prints starts with `dir2`. By default, this command will not follow symbolic links. When it examines a symbolic link, it tests the link itself, not its target.

Only one test can be given to `sfind`, unlike the real `find` command.

TESTS

The following tests can be applied.

```
-s filename
```

The file being examined satisfies this test if it is a link to the same file as *filename*. This means, in particular, that it has the same inode index and is in the same filesystem as *filename*.

```
-m fileglob
```

The file being examined satisfies this test if its filename, meaning the pathname stripped of all leading directories, matches the *fileglob*.

OUTPUT

The pathname relative to the starting directory of the search is printed on a single line for every matching file. All error messages are written to the standard error stream, not standard output.



EXAMPLES

The following matches all files in any subdirectory of `/usr` whose name starts with `b` and ends with `sh` and has a single character in between.

```
$ sfind /usr -m 'b?sh'  
/usr/share/lintian/overrides/bash  
/usr/share/menu/bash  
/usr/share/doc/texlive-doc/metapost/featpost/bash  
/usr/share/doc/bash  
/usr/bin/bash
```

The following matches all files in any subdirectory of `~` that are links to the same file as `sf_scene.jpg`. Here, `sf_scene.jpg` is in the current working directory.

```
$ sfind ~/pictures -s sf_scene.jpg  
/home/stewart/pictures/desktops/DSC_2026.JPG  
/home/stewart/pictures/MyPhotos/SanFrancisco_2018/DSC_2026.JPG  
/home/stewart/pictures/MyPhotos/2018_07_08/DSC_2026.JPG
```

EXIT STATUS

- 0 If it succeeded.
- 1 If it failed.

SEE ALSO

`fnmatch(3)`, `getopt(3)`

Error Handling

The program should detect incorrect usage and report this on standard error along with the correct form of usage.

Hints

You should read the man page for `fnmatch()` and read the man page for `getopt()`. Notice that the test follows the directory names in this command. The `getopt()` man page explains how to handle this.

Code Re-Use and Program Structure

You are free to use any of the code in the class's repository. There are a few demo programs in Chapter 8 that will be very useful. All of it has GNU General Public Licenses. If you do use any of that code, *the documentation should indicate which parts are not written by you*. The program should be a single source code file. **All functions other than `main()` should precede `main()` in the file.**



Instructions for Submitting the Assignment

1. Use the `submithwk_cs49366` program to submit your program. To submit, if your file is `~/sfind.c`, you'd enter

```
$ cd ~  
$ submithwk_cs49366 -t 4 sfind.c
```

The program will copy `sfind.c` into the directory

```
/data/biocs/b/student.accounts/cs493.66/hwks/hwk4/
```

and if it is successful, it will display the message, "File `hwk4_username.c` successfully submitted." where `username` is your username. You will not be able to read this file, nor will anyone else except for me. But you can double-check that the command succeeded by typing the command

```
ls -l /data/biocs/b/student.accounts/cs493.66/hwks/hwk4
```

and making sure you see a non-empty file named `hwk4_username.c` where `username` is your user name and whose date of last modification is the time at which you ran the command.

2. *You can do step 1 as many times as you want. Newer versions of the file will overwrite older ones.*

Deadline

You must complete this assignment before its *deadline*, which is *Thursday, April 18, at 7:00 PM*. *After that, you will not receive credit for completing it.*

Grading Rubric

This assignment is **18%** of your final grade. The output must be correct and the program must conform to the rules written in the *Programming Rules* document on the class's webpage. It must be thoroughly documented.