CS 392, Systems Programming: Assignment 4
Sorted Permission Find

Overview

For this assignment, we will be taking our past program "Permission Find" and sorting its output. Fortunately, Linux has an excellent utility that does this for us. This program is called 'sort'. Check out 'man 1 sort'.

Now that we know how to use exec(), we can leverage this tool to our advantage, because we don't want to create a whole new 'sort' utility. To utilize this tool, we will be using a combination of fork(), exec*() (choose your favorite flavor), and pipe().

Objective

The object of this assignment is as follows. You will create two child processes, one for 'pfind' (executable provided), and one for 'sort'. You will connect the standard output of 'pfind' to the standard input of 'sort' using one pipe, then you will connect a second pipe to the standard output of 'sort', which your parent process will read from.

Your parent process will read from the read-end of the 'sort' pipe until it reaches end-of-file (read() returns 0), print out all the text received, AND report how many lines were returned.

A few caveats: you will not use 'popen()', 'fread()/fwrite()', or any other functions that use FILE streams. You will be working with pipes (which are raw file descriptors), using 'fork()' and 'exec*()', using 'dup2()' to duplicate file descriptors, and making sure you close all unused file descriptors. You may also use read() and write().

You should check the return code of all the system and function calls you use in your program. If any of them fails, print an error message and return EXIT_FAILURE. We will look through your code to verify you check return codes and take the proper action.

We will, however, check for specific output for failures that occur when exec-ing 'pfind' and 'sort'. If an error occurs, we expect to see

```
Error: pfind failed.
```

or

```
Error: sort failed.
```

written to standard error.

This is a short assignment in terms of lines of code, but the lines are a bit tricky to write and hard to debug. Start early, and ask questions early and often. This is a one-week assignment.

A few common pitfalls are as follows:

- *My program hangs!*
  Make sure you have closed all file descriptors that you do not need. If the write end of a pipe is still unclosed, the process on the read end of it will think it is still open.

- *Exec on pfind fails every time!*
  Make sure you have execute permissions on the pfind executable. If you don't make sure you set execute permissions on it.

- *My pipe doesn't work!*
  Check the return value of the call to dup2, and make sure it's succeeding. It could be that you're using it wrong. Also, start small. Try a toy program with one child process which just writes some string to the pipe for the parent to read. Get comfortable working with that and then build up to the full solution.

Sample Executions

```
$ ./spfind
Usage: ./spfind -d <directory> -p <permissions string> [-h]

$ ./spfind -d ~ -p rwxr-xr-q
Error: Permissions string 'rwxr-xr-q' is invalid.
```

For the test cases below, your output may vary, depending on the files your have in your system.

```
$ ./spfind -d ~ -p rwxr-xr-x
Error: Cannot open directory '/home/user/danger_dir'. Permission
denied.
/home/user/canned_dir
/home/user/canned_dir/level1dir1
/home/user/canned_dir/level1dir1/somedirectory
/home/user/canned_dir/level1dir1/someotherdirectory
/home/user/canned_dir/level1dir2
/home/user/canned_dir/level1dir2/alice
/home/user/canned_dir/level1dir2/charles
/home/user/Documents
/home/user/Downloads
/home/user/Music
/home/user/Pictures
/home/user/Public
/home/user/Templates
/home/user/test.sh
/home/user/Videos
Total matches: 15
```

Notice the error message is written to standard error and is not part of the total matches. If we redirect standard error to /dev/null, we'll obtain the following output:

```
$ ./spfind -d ~ -p rwxr-xr-x 2>/dev/null
/home/user/canned_dir
/home/user/canned_dir/level1dir1
/home/user/canned_dir/level1dir1/somedirectory
/home/user/canned_dir/level1dir1/someotherdirectory
/home/user/canned_dir/level1dir2
/home/user/canned_dir/level1dir2/alice
/home/user/canned_dir/level1dir2/charles
/home/user/Documents
/home/user/Downloads
/home/user/Music
/home/user/Pictures
/home/user/Public
/home/user/Templates
/home/user/test.sh
/home/user/Videos
Total matches: 15
```

References

man 2 read

man 2 write

man 2 pipe

man 2 fork

man 2 wait

man 3 exec

man 2 dup2