Laboratory Exercise 5

cpe 357 Fall 2020

Due by (or before) 11:59pm, Friday, October 26th.
Due to the Midterm, no more than one late day may be used on this lab.
The Laboratory Exercise is to be done individually.

Problems

There are no written exercises this week. Stay tuned for next week.

Laboratory Exercise

1. There's one task for this week: Write a program, `mypwd.c`, that performs the same task as `/bin/pwd` *without* using the `getcwd(3)` function described by POSIX. I want you to traverse the directories and do it yourself. The technique is described in Stevens, Section 4.23.

   If there are multiple paths to (possible names for) any partiular directory, return the one that appears earliest in each directory that is traversed.

   Your program should produce the same results as `/bin/pwd`. In case of error, call `perror()` with the string "mypwd" to report the error and exit. If the pathname is too long (see below), simply print "path too long" and exit. Similarly, if for whatever reason, you are unable to determine the present working directory (e.g. it's been unlinked), print "cannot get current directory."

   Just to make life easier, you may assume you will get no paths deeper than `PATH_MAX`, defined in `limits.h`. If `PATH_MAX` is undefined, you may define it to be 2048 characters.

Tricks and Tools

You will probably want to look into Stevens, Chapter 4, and:

`opendir(3)`	open a directory for reading
`readdir(3)`	read a directory entry
`rewinddir(3)`	rewind a directory to the beginning
`closedir(3)`	close a directory
`stat(2)`	get file status
`lstat(2)`	get file status

Remember, in an environment where there is more than one filesystem mounted, it is possible, though unlikely, that you could discover that the i-node of the top of a mounted filesystem is the same as the one you're looking for. (Remember, inode numbers are only unique within a filesystem.) If this happens, your program might get confused. Even though the inode number may be the same on another device, the device number won't. Check both and you'll be in good shape.

For the Laboratory Exercises: Submit the program described above to the `lab05` directory of `pn-cs357` via handin.

A makefile (called `Makefile`) that will build your program when given either the target "`mypwd`" or just "`all`".

A README file that contains:

- Your name(s). In addition to your names, please include your Cal Poly login names with it, in parentheses. E.g. `(pnico)`

- Any special instructions for running your program.

- Any other thing you want me to know while I am grading it.

The README file should be **plain text,** i.e, **not a Word document**, and should be named "README", all capitals with no extension.