


Assignment 2 for CS570, Summer 2019

This page last modified 16 May, 2019

You shall implement a Program Actuator and Microshell, “PAM”. PAM shall provide the following functions:

1. Provide a shell-like interface for launching new programs:
print a prompt "cssc9999% " (where cssc9999 is your username) and then wait for the user to type in a file name. Note, PAM will NOT have any shell built-in functions (such as cd, setenv, printenv, bg, fg, etc), with the exception of #5 (below).
2. When the user enters a filename and hits the “enter” key, read the filename entered and determine if the file is an executable file. If it isn't, then print a useful error message to the user and then return to #1 above (display a new prompt and wait for user input). If it is an executable file, then create a new process and run this program in the new process.
3. Note – you will need to determine whether the filename is a fully qualified path name or if the file needs to be searched for (fully qualified pathnames begin with a slash (i.e. "/"), use the correct variant of ‘exec’ to do this. 
4. PAM will support users creating two processes with a pipe for the two processes to use to communicate with each other. For example, the following is legal input:

```
"cssc9999% ls | sort"
```

In this example, PAM would create a pipe and two processes to use it for communication. In this example, the first process (running ls) would send its output to the second process (running sort) which would read from the pipe.

5. Well,, there is one builtin function you need to implement (actually two, counting the pipe in the previous requirement), if the user types "exit" (then hits the “enter” key), then your program shall gracefully close itself down and terminate.

NOTE - The ONLY valid input formats are the following (reject/don't accept all others):

- pam% executablefilename
- pam% executablefilename argument <- (only one argument, no more than one)
- pam% executablefilename | executablefilename
- pam% executablefilename | executablefilename | executablefilename <- (any number of pipes may be constructed on the command line)
- pam% exit

Your program will be tested by compiling it and executing it on **edoras**. Your program shall be written such that it compiles and executes cleanly when using the **gcc/g++** compiler on edoras. You shall create a sub-directory named "a2" in your home directory. In it, you shall place your source files (multiple source files are required), your header file, your Makefile, and a README file (follow instructions from assignment #1 for the README file). Additionally, identify in your README file who worked on which lines of code in this project (if you used Agile/Pair programming state who was writing and who was providing input for each function/method).

Your source files SHALL CONTAIN sufficient comments for making the source easy to read. Points will be taken off for poorly (or non) commented source. Name the executable "pam".

- Create ~/a2 by hand.
- Create multiple c/c++ source files, an include file, a Makefile, and a README file. Put them into ~/a2.
- The Makefile shall create an executable by the name of pam in the same directory (~/a2).
- The system call "system()" will NOT be allowed

The assignment is due 1800 (6:00 PM) on Monday, 24 June 2019

TURNING IN YOUR WORK:

Your project files shall be loaded onto Assignment #2 on Blackboard and into the class account.

Make sure that all of your files (all source files, Makefile, README file, test files, etc) are in the a2 sub- directory of the class account

Additionally, create a tarball (tar file) or zip file and attach that file (upload it) under Assignment Submission in Assignment #2 in Blackboard.

SUGGESTION: check out my sample source files and Makefile posted on Blackboard.