Homework 1 – Shell

Task 1

Answer problem numbers 1, 3, 10, 12, and 17 from the end of chapter 1, on pages 81 and 82 of the old text book "Modern Operating Systems" by Tanenbaum and Bos (available on canvas).. To ensure you're doing the correct problems I have included question one (word for word) below:

1. What are the two main functions of an operating system?

Task 2

Write a c program that implements a shell on Linux. You can name the shell anything you want, but I suggest you name it after yourself so that years from now you'll be famous. Your shell should have the following basic functions:

- A "read → parse → execute" loop
- Efficient and robust buffering and parsing of user input. The user should be able to type in commands with possibly thousands of characters. Memory should not be wasted if they choose to type in a very short command.
- Built-in support for "exit", and "cd" (which use the `exit()` and `chdir()` system calls)
- You may hard-code the user name and computer name. But you should use `get_current_dir_name()` or similar to print the current folder in the prompt.
- Execution of arbitrary programs / commands via `fork()` and `exec()`
- Well structured and commented code (explaining "why" and not "what")
- Outputs some message (e.g., "unknown command") if the user types some nonsense
- Does not crash when the user enters nothing and presses enter
- You shell does not have have to handle pipes, spaces in command parameters, or any other "advanced features"

Note: Your shell program is OS specific. So, you will need to develop and test that it runs correctly on the CS department computers. It should be written in C and so it should compile using gcc. Your program should perform prudent memory management. There should not be any memory leaks (but there may be some unavoidable "uninitialized values" which you can safely ignore.)
Hint: There are several variants of `wait` and `exec` as system calls and libraries in Linux. You should read the man pages in section 2 and section 3 and select a version that seems best / easiest to use for you.

Command Examples

Some commands that should work include:

- `ls`
- `cd` (note: you don't have to update the prompt message)
- `geany`
- `sleep 4`
- `geany yourshell.c`
- `geany myfile1.c myfile2.c myfile3.c myfile4.c`
- `python3`
- `sudo apt install sl`
- `sl`

Some commands and features that don't have to be supported include:

- `ls; sleep 5; cd ../` - You don't have to handle "multi-command lines".
- `ps -e | grep kworker` - You don't have to support pipes.
- `geany your shell.c` - You don't have to support spaces in parameters.
- "tab-completion" feature
- color coding

What To Submit

On canvas, you should upload all of the following.

- A PDF with the answers to the questions from task one.
- The source code for your c shell (mine is called `novak-sh.c`)
- A compiled binary of your c shell that runs on the CS department computers (mine is called `novak-sh.bin`)
  (Note: novak-sh.exe may be necessary so canvas will allow you to upload)