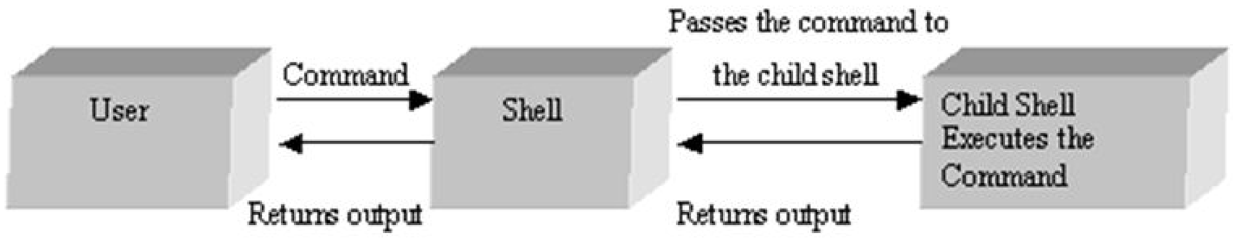
Computer Engineering Dept. **Operating Systems**
Cairo University
Fall 2016 **Linux Shell**

Cairo University

Introduction

Shell is a utility program with the Linux system that serves as an interface between the user and the kernel and also plays a very important rule as a Command Interpreter and should be able to do the following :

- Reads the command
- Locates the file in the directories containing utilities
- Loads the utility into memory
- Executes the utility as a child of its own


User
Command
Returns output
Shell
Passes the command to the child shell
Returns output
Child Shell Executes the Command

Requirement

Implement a shell that is partially similar to your default Linux Shell. In particular, your shell should do the following:

- **[20%] Display the current working directory with the format**:
  "<YOUR NAME>@<machinename>:<path>$ "
  Example:student@ubuntu:/home/student/Desktop$

Hints: you will need functions outside Labs like getpwuid()

- **[40%] Accept a Linux command and execute it when the return key is pressed. At least you have to be able to run the following commands: cd, ls, ps, kill, and running an executable file "i.e. .out file"**
  Example 1:student@ubuntu:/home/student/Desktop$ ls –l
  The files and directories on your desktop should be listed in the long listing format.

  Example 2: student@ubuntu:/home/student/Desktop$ ./infinite_loop.o
  Assuming the existence of a binary that enters in an infinite loop with the name "infinite_loop.o" on my desktop, your shell should execute this binary.

Hint: "cd" is a bit challenging command since it is not a program and you need to implement it, you also might need to use function like chdir

- **[20%] Accept CTRL+C to print "^C" then close the current process executing in your shell.**
  Example 1: if the binary infinite_loop.o from Example 3 above is executing it should be closed
  Example 2: if no binary is executing in your shell then CTRL+C prints "^C" ONLY.

This should be too easy

- **[10%] Implement the ampersand control operator "&"**, so that the user can run several processes in the background.

This should be handy as well

- **[5%] An error message should be printed on entering unsupported command**

This should be handy as well

- **[5%] Code professionalism including but not limited to comments, naming conventions , indentions, neatness and efficiency from space and time complexity and the right use of Datastructure.**

Bonus

- **[10%] Use the up arrow only on your shell to retrieve the history of the commands** the user typed before in the opposite order he first typed them and the down arrow to go back one step.

Hint: think of DS you will use to store commands

- **[+30%] Implement correct TAB completion for file and folder names only**, which means that when the user presses TAB button, you should complete the file or directory name he started writing using the files and directories in the current working directory only. If more than one file starts with the same prefix the user wrote then you should print them all sorted alphabetically.

Hints

- Your code is in c/c++ and it must compile with gcc/g++ with no errors or warnings.

- Most required functions and mechanisms to finish your assignment are given in labs 1 to 4 and this document, just read them carefully. The command “cd” may require special handling; take a look at the documentation of the function “chdir” to do the job of “cd” if needed.

- You need to execute each command in a different process. **DO NOT use the system() function**.

- Pay attention when writing the logic that updates the display of the current working directory, there are some pitfalls that you should avoid.

  Example:student@ubuntu:/home/student/Desktop$ cd .. The above command should update the current working directory to “/home/student” and thus your shell should display “student@ubuntu:/home/student$” and not student@ubuntu:..$

- Make sure that all commands are executed in the context of your working directory before and after the execution of a “cd” command.
  If we used the same example in the bullet above and executed “ls” before the “cd” command then the contents of the directory “/home/student/Desktop” should be listed but after the “cd” command the contents of the directory “/home/student” should be listed.

- You may need to use some C-string manipulation methods like “strcpy”, “strlen” and “strcat”

- Bonus may require methods that we didn’t instruct in the lab but I assume you know them.

- TAB completion bonus is a little bit hard so manage your time wisely.

- Cheating lead to a “-ve” grade not just zero for both parties.

Teams

- **Teams of 2 students** and no more than 2 are allowed per team.
- Each team will send a **“.c” file or “.cpp” on elearning** on or before **11:59 PM Monday 31/10/2016 and no later.**
- **You can send your questions on the elearning forum for this assignment**
- The “.c” file name is Student1ID_Student2ID.c, example 11234_11235.c
- Your .c file should start with a comment in the following format:
  // Name of Team Member 1 - ID of Team Member 1
  // Name of Team Member 2 - ID of Team Member 2