Northeastern University
College of Engineering
Department of Electrical & Computer Engineering

EECE7376: Operating Systems: Interface and Implementation

Course Project

Instructions

In this project you can work in a group of at most 2 students. If you work in a group, only one member must submit to the project assignment on Canvas the following:

1. Your project report developed by a word processor (e.g., Microsoft Word) and submitted as one PDF file. The report cover page must include your group details (full names and student IDs), course/section/semester information. For contents that require drawing and if it is difficult on you to use a drawing application, which is preferred, you can neatly hand draw "only" these diagrams, scan them, and insert the scanned images in your report document. The report includes the following:
    - Summaries of your approach to solving the programming part.
    - Screen shots of sample runs of your project programs. The sample runs must cover all scenarios of the project requirements.
    - An extended summary of your selected topic in part 2 of the project and the link to your YouTube lecture.
2. The well-commented **source code files** (i.e., the .c or .h files) that you add or modify. At the beginning of your source code files write your group details (full names and student IDs) and any special compiling/running instruction (if any). For the xv6 programs that you only make changes in them, only add comments to the parts you change/add and no need to add any names at the beginning of these programs. Do not submit any compiled object or executable files.
3. The project part 2 PowerPoint slides file.

Do NOT submit any files (e.g., the PDF report file, the source code files, and the slides file) as a compressed (zipped) package. Rather, upload each file individually.

__Note__: You can submit multiple attempts for this project, however, only what you submit in the last attempt will be graded. This means all required files must be included in this last submission attempt.

Part 1 - Shell

The goal of this part of the project is implementing your own shell program on a Linux environment (not xv6). The shell program repeatedly reads commands from the user, interprets them, and translates them into a sequence of actions that likely involve an interaction with the operating system through system calls. Your code can rely on the string manipulation routines implemented throughout our previous homework assignments. The shell must include the following set of required features:

1. The shell should print a `$` prompt symbol followed by a space when it is ready to receive a new command from the user.
2. When a command is launched in the foreground, the shell should wait until the last of its subcommands finishes before it prints the prompt again. A command is considered to be running in the foreground when operator `&` is not included at the end of the command line.
3. When a command is launched in the background using suffix `&`, the shell should print the `pid` of the process corresponding to the last sub-command. The `pid` should be printed in square brackets. It should then immediately display the prompt and accept new commands, even if any of the child processes are still running. For example:

   ```
   $ ls -l | wc &
   [3256]
   $
   ```

4. When a sub-command is not found, a proper error message should be displayed, immediately followed by the next prompt. Example:

   ```
   $ hello
   hello: Command not found
   $
   ```

5. The input redirection operator `<` should redirect the standard input of the first sub-command of a command. If the input file is not found, a proper message should be displayed. Example:

   ```
   $ wc < valid_file.txt
   223 551 5288
   $ wc < invalid_file
   invalid_file: File not found
   ```

6. The output redirection operator `>` should redirect the standard output of the last sub-command of a command. If the output file cannot be created, a proper message should be displayed. Example:

   ```
   $ ls -l > invalid/path
   invalid/path: Cannot create file
   $ ls -l > hello
   $ cat hello
   < Contents of “hello” displayed here >
   ```

Part 2 – OS Extra Topic

The goal of this part of the project is for you to further cover an OS topic that is not covered, or partially covered, in our course. You can select such topic of your interest from our OSTEP textbook. The following are the deliverables for this part of the project:

1. PowerPoint slides where you explain in detail your selected topic with at least 15 slides following the same style as our course slides.
2. A video lecture delivered by the project group and uploaded to YouTube. The lecture should include enough details to help other students understand your topic. All members of the project group must participate in this lecture video. The video should have a duration of 15 to 20 minutes.
3. On our course Piazza, create a new discussion post for your topic. Include a link to your lecture video and a summary of your topic. You also should check the topics uploaded by other groups and ask questions or write feedback on their Piazza posts. Your engagement in discussions with other groups posts will be counted towards your class participation grade.