Instituto Superior de Engenharia de Lisboa

BEng in Computer Science and Engineering

System Virtualization Techniques, Autumn/Winter 2024/2025

First coursework assignment

For this assignment, you will develop a simplified shell program called *bach* (**b**ash **a**lternative from **ch**elas) that processes command lines as described below, where `$` is the prompt symbol.

- Execute a program without any arguments. For example:
    ```
    $ ls
    ```
- Execute a program with arguments. For example:
    ```
    $ cat text1.txt text2.txt
    ```
- Execute a program, with or without arguments, redirecting its *standard output* to a file. For example:
    ```
    $ cat text.txt > copy.txt
    ```
- Chain two program executions, connecting the *standard output* of the first to the *standard input* of the second. For example:
    ```
    $ cat text.txt | grep abc
    ```
- Chain two program executions as above, but also redirect the *standard output* of the second process to a file. For example:
    ```
    $ cat text.txt | grep abc > output.txt
    ```

The developed shell program, when executed, displays the prompt symbol (`$`) and waits for a command line (you may use `fgets` for this). The command line is split into multiple programs, separated by | (pipe), each of which is executed in its own process via fork/exec, taking care to set up the necessary piping and any final redirection. The shell waits for all executions to finish before displaying the prompt symbol (`$`) again for the next command line. The `cd` command changes the current directory. The shell terminates its own execution when the special `exit` command is issued.

Exercises

1. Implement the `bach` program as specified above and **test it** by running the provided examples, as well as **others** you find appropriate.

2. Add a new user to your Linux system with:
    ```
    $ useradd iselx -m -s full_path_to_the_new_shell
    ```
    ```
    $ passwd iselx
    Enter new UNIX password:
    Retype new UNIX password:
    ```
    Log in with the new user and verify that the shell functions correctly.

3. Analyze and understand the [documentation for the POSIX *tee* command](). Questions will be asked during the discussion. Prepare a quick demonstration using the user account *isel* and the `bash` shell.

4. [OPTIONAL] Improve your `bach` shell so that it is not limited to chains of two commands.

Delivery

Submit exercise 1 or 4 using the tag **CW1** in the GitHub repository. Exercises 2 and 3 will be demonstrated during the discussion.

Do not submit binaries and other unneeded files to the repository.

ISEL, September 20, 2024

Submission deadline: October 6, 2024