CS433 Programming Assignment 2

Unix Shell and History Feature (100 points)

This assignment is based on the programming "Project 1 - Unix Shell" in chapter 3 (P-15) of the textbook.

This project consists of designing a C/C++ program to serve as a shell interface that accepts user commands and then executes each command in a separate process. Your implementation will support input and output redirection, as well as pipes as a form of IPC between a pair of commands. Completing this project will involve using the UNIX `fork()`, `exec()`, `wait()`, `dup2()`, and `pipe()` system calls.

Required Output

Your program should always first print your names and a short description when it starts. You must submit your source code and Makefile so that we can compile your program with the make command.

Your program should implement I - IV parts of the project, namely:

1. Creating the child process and executing the command in the child
2. Providing a history feature
3. Adding support of input and output redirection

You may earn extra credits by implementing "Part V. Communication via a Pipe" in your program.

Attention:

Always check for potential errors of system calls in your code, which might lead to creating an infinite number of processes. For example, `execvp()` may fail if the command is unknown, which should result in an error message **"`command not found`"**. Monitor your processes and make sure all processes are terminated appropriately. We don't want to have hundreds of processes use up all the computer resources.

Use command **"`ps -afu your_user_name`"** to see the processes belonging to you and kill a process of yours using "`kill -9 pid`", where pid is the id of the process.

Hint: You may use strtok function (strtok - C++ Reference (cplusplus.com)) to parse out the user command