

CS 3305: Operating Systems
Department of Computer Science
The University of Western Ontario
Programming Assignment 1
Spring 2015

Purpose

The goals of this assignment are the following:

- Get experience with the `fork()`, `execlp()`, `pipe()`, and `dup2()`, system functions
- Understand how a shell works
- Learn more about how operating systems are structured
- Gain more experience with the C programming language

Part I: Specification for Shell Program

In this assignment you are to implement a basic shell. A shell is a command line interpreter that accepts input from the user and executes programs on behalf of the user based on the commands that the user inputs. The line that the user enters commands on is referred to as the *command line*. The shell repeatedly prints a prompt on the command line, waits for the user to enter commands and executes programs. You are to write a simple Unix-like shell in the C programming language that has these features:

- If your name is *xyz* then the prompt should be the string *xyz>*
- Your shell must support I/O redirection. This includes supporting both input and output redirection in the same command i.e., you should be able to support the following: *sort < f1.txt > f2.txt*. However, you can assume that *<* will appear at most once in the command and that *>* will appear at most once in the command.
- Your shell must support up to three pipes.
- You do not have to support a command that has both I/O redirection and pipes.
- A *built-in* command is a command that changes the state of the shell or requests information about the shell's state. You should support the following two built-in commands:
 - *exit*: This command is used to terminate the shell.
 - *history*: By default this command will display the last ten commands issued. If there are fewer than 10 commands then display all previously issued commands. The history command may take a numerical argument e.g., *history 5*. For this specific example you would display the last 5 commands. If you type *history 15* then the last 15 commands would be displayed. The most recent command should be the last command.

Part II: Hints

See the assignment web page

Part III: Tentative Mark Distribution

This section describes a tentative allocation of marks assigned for the desired features.

- Prompt is displayed properly (2 points)
- exit command works properly (3 points)
- history command works properly (15 points)
- Repeatedly be able to read and execute a line with a command that has no arguments and no pipes (10 points)
- Repeatedly be able to read and execute a line with a command that has arguments (20 points)
- Repeatedly be able to read and execute a line with a command that has pipes (35 points). To receive the full 35 points you must support up to three pipes.
- I/O Redirection (15 points)