CS 3305A: Operating Systems
Department of Computer Science
Western University
Assignment 1
Fall 2018
Due Date: October 7, 2018

Purpose

The goals of this assignment are the following:

- Get experience with *fork(), wait(), exec()* family, *pipe(), and dup2()* system functions
- Understand how a shell works
- Learn more about the operating System is structured
- Gain more experience with the C programming language

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 will be provided with skeleton code that will define the structure of the code and handle the user input. You will be tasked with implementing the code that handles interpreting those command lines and executing them. Your shell must handle the following:

- **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.
- **Pipes**: This includes multiple pipes within the same command, you should be able to support the following: cat f1.txt | sort.
- You do not have to support a command that has both I/O redirection and pipes.
- A built-in command that will allow the user to terminate shell. For consistency and use of marking the command will be ‘exit’.

Hints

The following system commands can prove to be useful when coupled with a pipe in this assignment:

- tee
- cat

Provided Files

- Your changes should only be inside the shell.c, shell.h, files and Makefile
- You are encouraged to create other files to help with modularization of your code
- There are three tests provided with the code. You are encouraged to run them and create your own tests as well

Use the following commands to compile and run the program:

Compile the program using:

```
Make
```

Run the program using:

```
./myOS shell
```

Test the program using:

```
make test
```

Marks Distribution

- Forking a new process for every command: 30 points
- Handling I/O redirection for piping: 30 points
- Handling I/O redirection: 30 points
- Gracefully terminate shell using the ‘exit’ command: 10 points

Assignment Submission Guideline

- Must run on GAUL. Otherwise, you will receive a mark of zero. Refer to the website for further information.
- Only .c, .h, and Makefiles. Marks will be deducted if other files are submitted