

## Programming Assignment 1

CS450 Spring, 2018

1. This assignment is an individual effort. It is due on 2/07/2018

### 2. Requirements:

This programming assignment is inspired by one given at MIT (<https://pdos.csail.mit.edu/6.828/2017/homework/xv6-shell.html>). The MIT assignment asks the students to implement the pipe “|” and IO redirection “>” operators on the shell program given in <https://pdos.csail.mit.edu/6.828/2017/homework/sh.c>. We ask you to implement the parallel command execution operator “&” and the redirection operator “>” in the same shell program. Invoke your shell from the shell that comes with xv6 or from a robust UNIX.

- 1) After your shell has started, it will give a prompt to the user. You should use “CS450\$” for the prompt.
- 2) Port a few simple commands such as “echo”, “cat” or “wc” to your environment so that you can test your results. Some of your commands will take arguments.
- 3) If the user types `cmd1>file` after the prompt, `cmd1` will get executed. Your shell will give a new prompt in a new line, ready to execute the next command line. The output of `cmd1` will appear in `file`.
- 4) If the user types `cmd1&cmd2`, both commands will get executed in parallel. After both terminate, your shell will give a new prompt in a new line, ready to execute the next command line.
- 5) Your shell shall support a command line with a string of 3 or more commands connected by the “&” operator. Two or more of the commands may use redirection.
- 6) A command string terminated by a “&” or “>” is illegal.

### 3. Deliverables:

- 1) Source and executable objects with a README on how to build and execute them.
- 2) A copy of the modified code, showing the modifications that you make and with comments that explains how your code works.
- 3) The finite state machine of the parser in UML state chart notations. (e.g. see [https://www.ibm.com/support/knowledgecenter/en/SS6RBX\\_11.4.2/com.ibm.s.a.oomethod.doc/topics/c\\_UML\\_State\\_diag.html](https://www.ibm.com/support/knowledgecenter/en/SS6RBX_11.4.2/com.ibm.s.a.oomethod.doc/topics/c_UML_State_diag.html))
- 4) The test data that you use and the reasons why the test data are of good quality. If you use the equivalence partitioning method to generate your test data, describe your equivalence partitions.