

## Programming Assignment 1

CS450 Fall, 2017

1. This assignment is an individual effort. It is due on 9/16/2017

### 2. Requirements:

This programming assignment is inspired by one given at MIT (<https://pdos.csail.mit.edu/6.828/2012/homework/xv6-shell.html>). The MIT assignment asks the students to implement the pipe “|” and IO indirection “>” operators and somewhat more. We ask you to implement the sequence command execution operator “;” and the parallel execution operator “&” and more. You should do this assignment by modifying the skeleton code attached. Invoke your shell from the shell that comes with xv6.

- 1) After your shell has started, it will give a prompt to the user. You should use “\$” for the prompt.
- 2) If the user types `cmd1 ; cmd2` after the prompt, `cmd1` will get executed, followed by `cmd2`. After `cmd2` is executed, your shell will give a new prompt in a new line, ready to execute the next command line.
- 3) 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.
- 4) The commands `cmd1` etc. are real commands supported by xv6 and they take arguments. An example is the command `echo "A"`.
- 5) Your shell shall support a command line with a string of 3 or more commands connected by the “;” and “&” operators. The operators are of equal rank and they will be executed from left to right. The string “`cmd1 & cmd2 ; cmd3`” will see `cmd1` and `cmd2` executed in parallel. `cmd3` after they both terminate.
- 6) A command string terminated by a “&” is illegal; but it is fine with “;”

### 3. Deliverables:

- 1) Source and executable objects with a README on how to build and execute them.
- 2) A copy of the modified skeleton code, showing the modifications that you make and with comments that explains how your code works.
- 3) A file containing the test cases and outputs. You should describe the test data that you use and explanation on why the test data is of good quality. If you use the equivalence partitioning method to generate your test data, describe your equivalence partitions.