

# CSCI 3334 Program Assignment

## Computer Simulation - Phase 1

due February 16, 2018

The first phase of your semester project will be to write a command line interpreter for the computer to be described in a later assignment. This command line interpreter will function as a simple operating system for the rest of the project. This must be written in the C programming language (NOT C++) and must run in the LINUX environment (2nd floor labs) without any added compiler options.

The interpreter should do the following:

1. The command interpreter should output an appropriate welcome message and a prompt and wait for the user to type a command line. The command interpreter will process the command and call the function which carries out the command. Commands will consist of single word commands followed by zero or more parameters. If a command requires an parameter and it is missing, the interpreter should tell the user that an parameter is required and then output the prompt and wait for a command. The same holds if an parameter is provided that is not required. For now, each command (except Help, Directory, and Exit) should be a stub which simply states that it has been reached. These commands will be programmed in later phases. NOTE: Your program MUST check for the presence/absence of parameters!
2. For this phase, do not use any precompiled routines to break up the line into pieces. You may only use loops and comparisons. That is, do not use strtok (or any of its variants). You may use library functions outside of this part of the code.
3. The command interpreter should **not** be menu driven, and should recognize the following commands (only the commands indicated by \* need to be implemented, but **all** should be recognized):

**load filename** : will call the load function to load the specified file.

**execute** : will call the computer simulation program to execute the program that was previously loaded in memory.

**debug** : will allow you to execute in debug mode.

**dump start end** : will call the dump function, passing the values of start and end. Start and end will be hexadecimal values.

**\*help** : will print out a list of available commands. If an unrecognized command is typed at the command level, a message to type help should be displayed.

**assemble filename** : will assemble an SIC assembly language program into a load module and store it in a file.

**\*directory** : will list the files stored in the current directory. Do this by `system("ls");`. You will need the directive `#include <stdlib.h>`.

**\*exit** : will exit from the simulator.

NOTE: The \* is not part of the command!!!

You may want to have the interpreter recognize the first few characters of the command, e.g., enter "dir" for directory. **Do not make up your own versions of the commands** -- use what is listed above.

Turn in a well-documented source listing. There should also be a working copy of your code on the server in the CS lab. This code should be in your area in a top-level directory called 3334/phase1. If it is NOT there, I cannot grade your program, and you get a 0. The only files in this directory should be your source files.