

# TCSS 422 — Computer Operating Systems

## Winter 2015 — Homework Assignment 5

**Due Date: Tuesday, Mar. 3**

### Guidelines

Homework should be electronically submitted to the instructor by the end of the day on the due date. A submission link is provided on the course Canvas page for this assignment.

### Assignment Description

This short assignment is intended to familiarize you with basic programming in MINIX. Your primary task is to set up a MINIX VM and then implement a very small C program.

### Implementation Specifications

Follow the instructions on the course Canvas site for [setting up and configuring MINIX](#).

The small C program you're to implement performs a simple operation, it reverses the data sent to its standard input stream and writes it to the standard output stream. For the sake of simplicity, the source code for your program can reside in the home directory of the `root` user, `/root`. There is no required name for the source file, but this document uses `reverse.c` in subsequent examples.

Regardless of how you edit your source file, you'll need to manually compile your program within your MINIX VM. Assuming you're in the directory with your source file, e.g., `/root`, you can compile your program with the `cc` command, e.g., “`cc reverse.c`”. By default, the compiled executable will be named `a.out` (for historical reasons), but you can change the name of the executable file the compiler produces using the `-o` command-line switch, e.g., “`cc reverse.c -o reverse`” to produce an executable file named `reverse`.

The name of the executable file is the command you issue at the command line to run your program. Unless you adjust the `PATH` environment variable, you will need to prefix that command with the path to the executable file, e.g., “`./reverse`”. If you were to execute the program so that it read input from the console and then typed “Hello”, followed by `<Enter>`, followed by `<ctrl-D>` (which indicates that you're done providing input), the program would display:

```
# ./reverse
Hello
^D
olleH#
```

The first line of the sample output above is the command prompt (`#`) and the command to execute the program. The next two lines are the echoed input, and the rest is the reverse of that input. The next command prompt appears at the end.

If you were to execute the program and type “One two”, followed by *<Enter>*, followed by “three”, followed by *<ctrl-D>*, the program would display:

```
# ./reverse
One two
three
^D
eerht
owt enO#
```

Note that this program reverses the *entire* stream, not each line in the stream. You can also redirect the standard input stream to reverse data from other sources, such as a file. For example, to display the reverse contents of the `.profile` file (located in `/root`) use the command “`./reverse < .profile`”.

### **Deliverables**

The following items should be submitted to the instructor. Failure to correctly submit assignment files will result in a 10% grade penalty.

- 1) The completed source file, e.g., `reverse.c`.

Do not include any extraneous files, such as Eclipse IDE files, object files, or subversion files.