Concurrent Programming and Foundations of Operating Systems
COMPSCI 386

For this assignment, you will implement custom versions of three standard UNIX commands. This will give you a chance to refresh your C programming skills. More importantly, it will help you to understand the basic strategy for implementing a UNIX command-line interface, which you will do for the second assignment.

The commands to implement are wcat, wgrep, and wzip/wunzip as described by our textbook here. As you will see, the authors provide a testing framework. You should use it to test your code — that is how I will test it.

When you are done, try updating the configuration file for your shell by adding the path to the binaries for your new commands. Instructions for doing this can easily be found online. Now you can open a shell and execute your commands just like any other commands. This illustrates the basic strategy mentioned above: a shell really has no idea how to execute commands entered by the user (with a few exceptions to be discussed later); it is just invokes other programs to do the work.

Remember, your code must compile with gcc to be eligible for partial credit (see syllabus for details).

Grading criteria: functionality, clarity, coding style, and documentation. This applies to all assignments. Review our C Style Guide for a short list of the most important elements of good coding style.

What to submit: Place your source files (*wcat.c*, *wgrep.c*, *wzip.c*, *wunzip.c*) in a folder named **hw1-Smith** with your last name in place of *Smith*. Zip that folder and upload it to BOLT. Please be careful to follow these instructions. Deviations require me to make adjustments to my grading procedures, which slows down the grading process. That might make me angry, and as we used to say back in the 1970s, Mr. McGee, don't make me angry.