# CSCI4730/6730 – Operating Systems

# Project #1: Multi-process and IPC

**Due date: 11:59pm, 2/7/2017**

## Description

In this project, you will design and implement a multi-process word counting program. The code of the single-process word counting[1] is provided in ELC. You will convert it into the multi-process architectures.

## Multi-process Word Counting Program

The main problem of a single-process program is the scalability. It cannot scale up to large numbers of input files.

To address the problem, you will convert the word counting program into the multi-process model. The main process creates the child processes and each child process reads and counts a single input file. The child process sends the result to the main process via Inter-process communication channel (e.g., pipe or shared memory). The main process waits all children processes and reads the result via IPC channel, and prints out the total on the screen.

- You will modify "wc_mul.c" to build a multi-process model.
- The program receives the number of input files through the command-line argument.
- You can use "time ./wc 10" and "time ./wc_mul 10" to see the performance of single and multi-process models.
- Explain your program structure and IPC in README.pdf file. Only "pdf" format will be accepted.
- Test input files are located in /tmp/CSCI4730/books/ in nike server. The file path is hardcoded in the project file. If you are using your own machine, you can copy /tmp/CSCI4730/books.tar.gz files from the nike server, unzip it, and modify the file path (#define FILEPATH "xxx") in wc.c and wc_mul.c file.

## Submission

Submit a tarball file using the following command

%tar czvf p1.tar.gz README.pdf Makefile wc_multi.c

1. README.pdf file with:
    a. Your name
    b. Explain your design of multi-process structure and IPC.
2. Your code should be compiled in **nike** machine.
3. Submit a tarball through ELC.

[1] It is slightly modified from the code in http://www.opentechguides.com/how-to/article/c/72/c-file-counts.html