CSC209: Software Tools and Systems Programming

http://www.teach.cs.toronto.edu/~csc209h/winter/

Karen Reid
reid@cs.toronto.edu
BA 4240
Introductions

• Who am I?

• Who are your fellow classmates?
  – find out the names of at least two people in the class that you didn’t know before today
  – ask them to tell you one interesting thing about themselves and offer one in return
Important tip

• Best way to start a conversation or ask a question when you don’t really know the person:

• “Hi, my name is <your name>. I’m in your CSC209 class, and I would like some advice on getting involved in research projects.”

• “Hi, my name is <your name>. I’m in your CSC209 class and I have a question about assignment 1.”
What is this Course About?

• Software Tools
  – Efficiently use the Unix Command Line
  – Understand the shell
  – Use Basic Shell Programming
  – Understand and Use Make

• Systems Programming
  – C
  – files
  – processes
  – communication
Course Information

• Check the course information sheet on the course web page for
  – Office hours
  – Contact information
  – Assignment schedule
• The course web page is the official source of announcements. (Piazza)
  [http://www.cdf.utoronto.ca/~csc209h/winter](http://www.cdf.utoronto.ca/~csc209h/winter)
• Make sure you have the prerequisites!
Communication

• **Piazza**
  – Use first for non-personal communication
  – Informative subject lines help
• **Email:** reid@cs.toronto.edu
  – Subject should include 209
  – Email is a formal method of communication:
    • Use proper English.
    • State your question clearly, with enough context.
    • **Sign it** (Name and cdf id are the most useful. Do not include your student number.)

• **Anonymous Feedback**
  – Link from the website
• **In person in Office Hours**
A typical week

- Prep (online)
- Practice (lecture)
- Exercises (lab)
Inverted/Blended Classroom

- Preparation before class (videos & exercises)
- Hands-on Activities in class
  - Need to bring computers to class (or share)
Texts

K.N. King

C PROGRAMMING
A Modern Approach
SECOND EDITION

Covers both C89 and C99

Michael Kerrisk

THE LINUX PROGRAMMING INTERFACE
A Linux and UNIX® System Programming Handbook
Assignments

• A1: Very basic C programming (5%)
• A2: Pointers, Memory Management (C) (10%)
• A3: Fork and pipes (C) (10%)
• A4: Processes and Communication(C) (10%)

• All code **must** work on the CDF (teach.cs.toronto.edu) servers to receive full marks.
• Code that does not compile on CDF will get 0.
• *Don’t wait until the last day!*
Weekly Exercises

• Starting this week
• Submitted (mostly) on PCRS
  – later in the term on MarkUs (and probably this week)
• 1% per week
• Must submit your work to get marks
• Labs are due by 6pm Friday
• Optional Attendance in BA labs (BA 3175 to start)
  – Wed 8-9 pm
  – Fridays 1, 2 or 3 pm
Submitting Assignments

• You will be using Git to manage and submit your assignments.
• The repositories will be set up this week or possibly next.

• Commit and push early and often.
• If you haven’t pushed your commits to the server, you haven’t submitted your assignment.

• *Do not wait until the last minute to try to commit your assignment for the first time.*
Plagiarism

• “The work you submit must be your own, done without participation by others. It is an academic offence to hand in anything written by someone else without acknowledgement.”

• You are not helping your friend when you give him or her a copy of your assignment.

• You are hurting your friend when you ask him or her to give you a copy of their assignment.
What is cheating?

• You are not allowed to
  – copy parts or all of another student’s assignment
  – include code from an external source without attribution
  – get someone else to do substantial parts of your assignment
  – give someone else your solution

• You are allowed to
  – help debug a friend’s program (be careful)
  – help each other understand man pages or example code.
Self Study Topics

• Using Git
• Using Unix - some initial guidance
• Learning an editor – vi, emacs, scite, nedit, ...
• Learning a debugger – ddd, gdb, eclipse
Windows users

• If you want to do some of your work locally on your own machine, you will need to install cygwin.
• Check out cdf site for help

• For general help (not just Windows users):
  – Take advantage of the help centre!
Why Unix?

• Available on a number of platforms.
• Multi-user, multi-programmed.
• Shares computer resources sensibly.
• Permits manipulation of files, processes, and programs.
• Allows inter-process and inter-machine communication.
• Permits access to its operating features.
The Unix Philosophy

• Write programs that do one thing and do it well.
• Write programs to work together.
• Write programs that handle text streams, because that is a universal interface.
Unix Principles

• **Do one basic thing well**
  – with some basic variations

• **Simple input formats**
  – plain text
  – don’t require interactive input
    • stdin to stdout/stderr

• **Simple output format**
  – expected to be input to another tool
Unix Tools Example

- sort
- sed
- standard input/ standard output
- pipes
Unix is user-friendly; it’s just choosy about who its friends are.
Basic Tools to Learn

- head, tail
- cd
- mkdir
- ls
- cp
- mv
- rm
- diff
- comm
- cut
- cat
- wc
- grep
- who
Unix manual

• man example
• don’t memorize - look it up!
• it grows on you
For Live Work in Class

- work on lab servers

- www.teach.cs.toronto.edu for help pages