Syllabus

- C Programming
- UNIX Shell Programming
- UNIX Systems Call
- Programming Tools
Goals

- Get familiar with UNIX programming environment
- Learn the philosophy of UNIX
Teaching Style

- Student centered learning
- No lecture notes
- Lots of demo - source code and video recording will be distributed.
Learning Style

- Nothing to remember (open book tests and exams).
- Learning by doing and referring to references.
- When in doubt, write small program to test.
- Get your hands dirty!
Time Table

- **Lecture**: Mon and Thu, 10am - 12noon, SR1
- **Lab**: Fri, 10am - 12noon, PL3
- **Office Hours**: Tue, 10am - 12noon, SOC1, 04-20
Continuous Assessments

- 50%: Programming Assignments
- 30%: Two Practical Tests
- 20%: Final Exam
I practice zero-tolerance policy for plagiarism.

School’s policy: *zero* mark for assignment and final grade lowered by *one grade point*. 
Assessment Principles

- Ability to write and debug programs in UNIX
- Understand the basic concepts of programming in UNIX
- Will NOT test on obfuscated language syntax such as `char (*(*x()))[][()]()` or "weird" statement such as `i = i++`. 
Working Environment

- Official programming environment: *sunfire*
- In class demonstration using Intel machine and Linux.
Website

- Not using IVLE.
- Use newsgroup SoC.acad.level2 on bbs instead of IVLE forum for discussion.
- Main website will be

http://www.comp.nus.edu.sg/~cs2281
Background

- Java
- Basic knowledge of using UNIX (ls, cd, mkdir etc.)
- Knowledge of a programmer’s editor, *vim* or *emacs* recommended. No *pico* please.
C and UNIX

- Invented together at Bell Labs.
- C: low-level language for implementing UNIX
- UNIX: a simple and elegant OS
- See "Bell Lab’s History of UNIX".