



This Fall, on a tiny blue marble on the edge of the Milky Way...

SYSTEM PROGRAMMING

Episode IV: A New Hope

System Programming

After the epic battles in 1010, 1729 and 2050, we are faced with a period of heightened challenges. Rebel forces managed to uncover an ancient secret, the C language of lore, that can turn into an ultimate weapon against the evil bug Empire!

It is our duty to learn this venerated skill, snatch keyboards, control systems and harness the power of this multi-core force to our advantage.

The time to escape the Empire's grip and join forces has come. Have no fear, as fear leads to the dark side. Join us and master the one true Language to restore freedom to our galaxy....



Overview

- Learning Objectives
- Organization
- Resources
- Evaluation



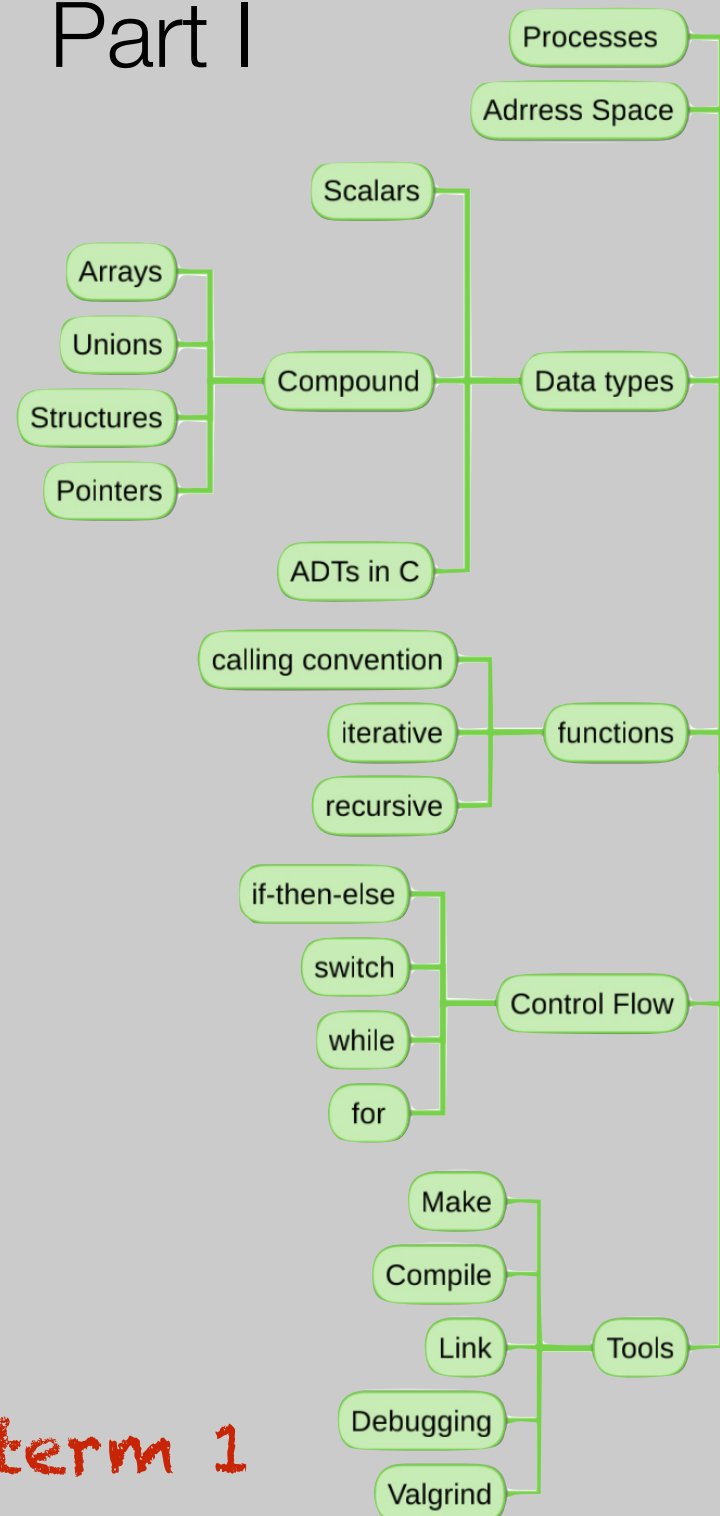
Learning Objectives

- **Twofold**
 - Develop System Skills
 - Develop programming skills on multi-core machines

Blue Print



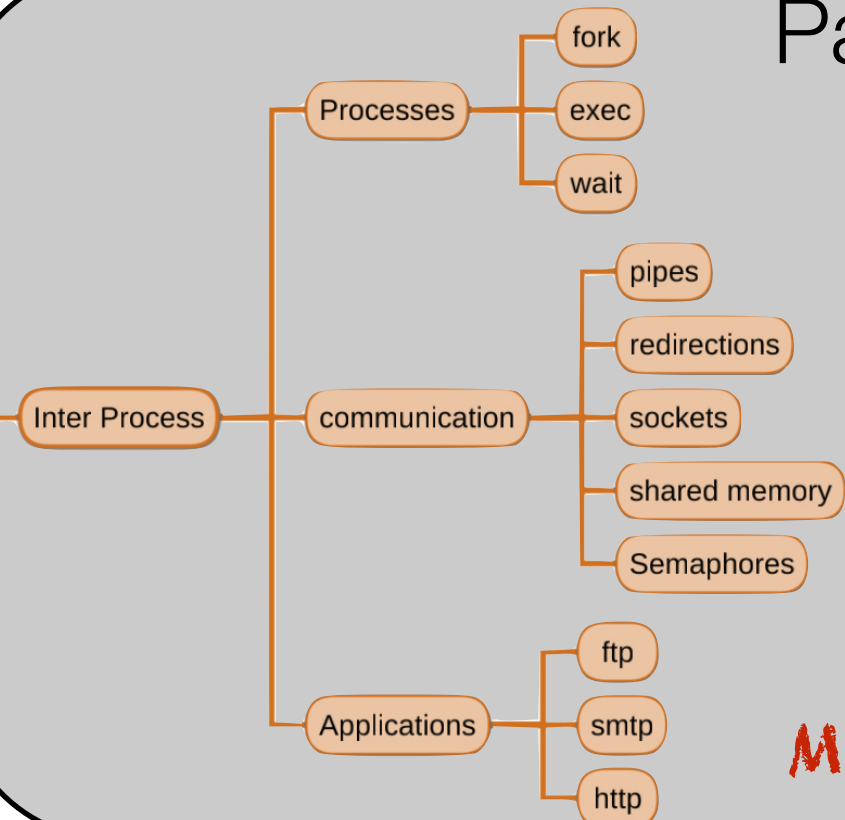
Part I



Midterm 1

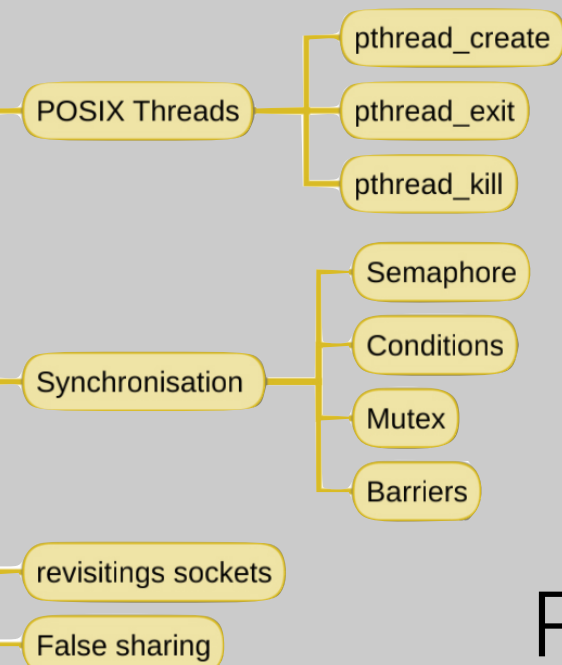
CSE3100

Part II



Midterm 2

Intra Process



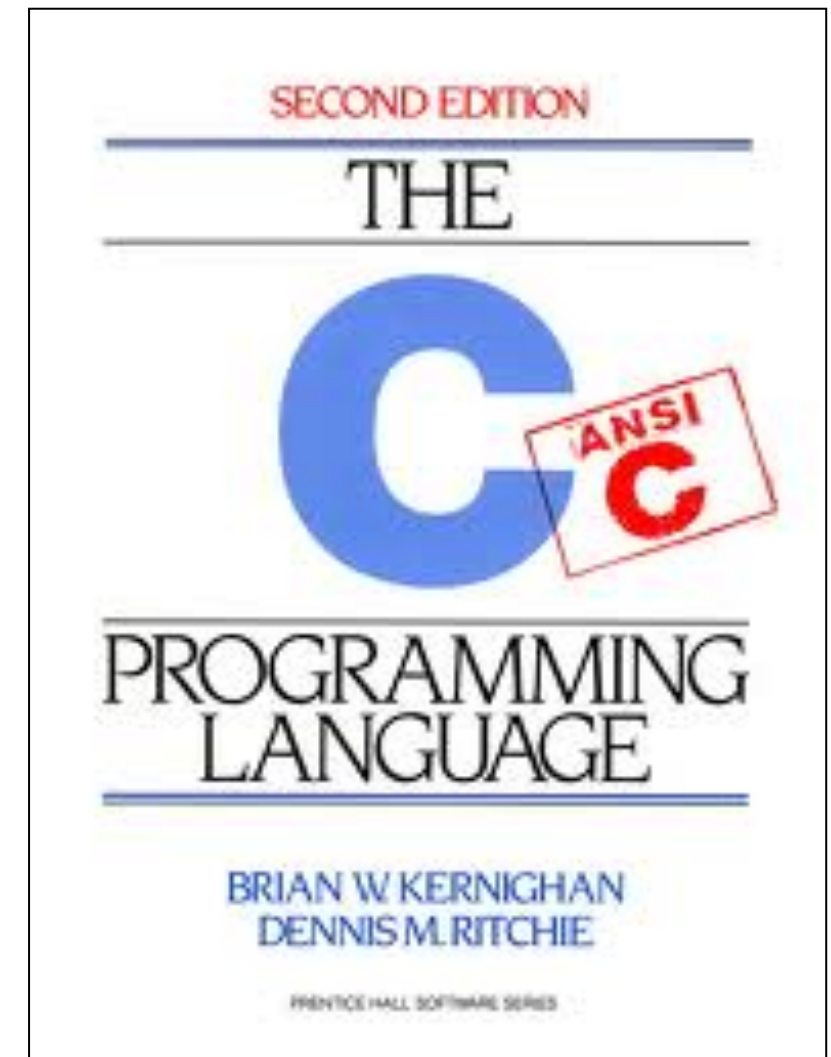
Final
or
Project

Part III



System Skills

- **C programming skills**
 - Compiled language
 - Close to O.S. layer
 - High-performance
 - Manual memory management
 - Pointers



C



- C is

Simple

Small

Very Popular

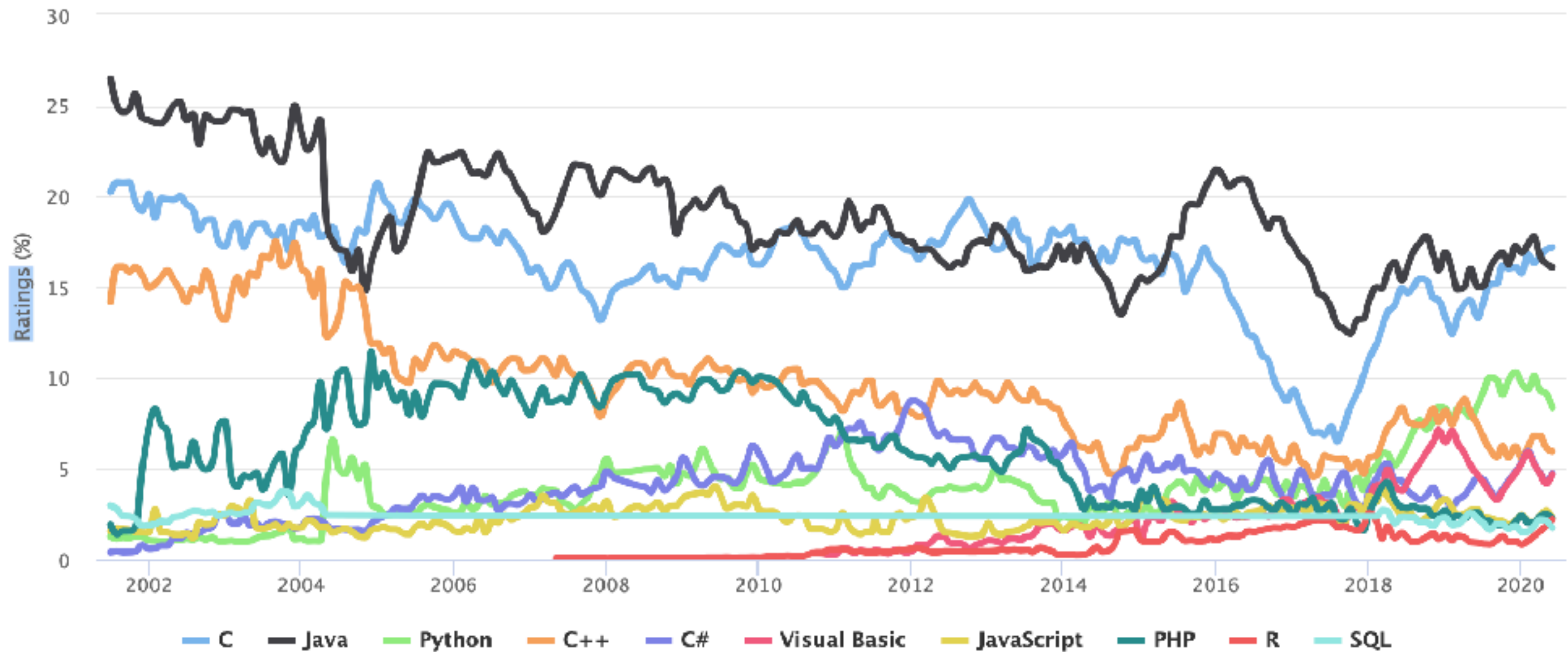
FAST

TIOBE Index



TIOBE Programming Community Index

Source: www.tiobe.com



Multi-core platforms

- In all your devices...





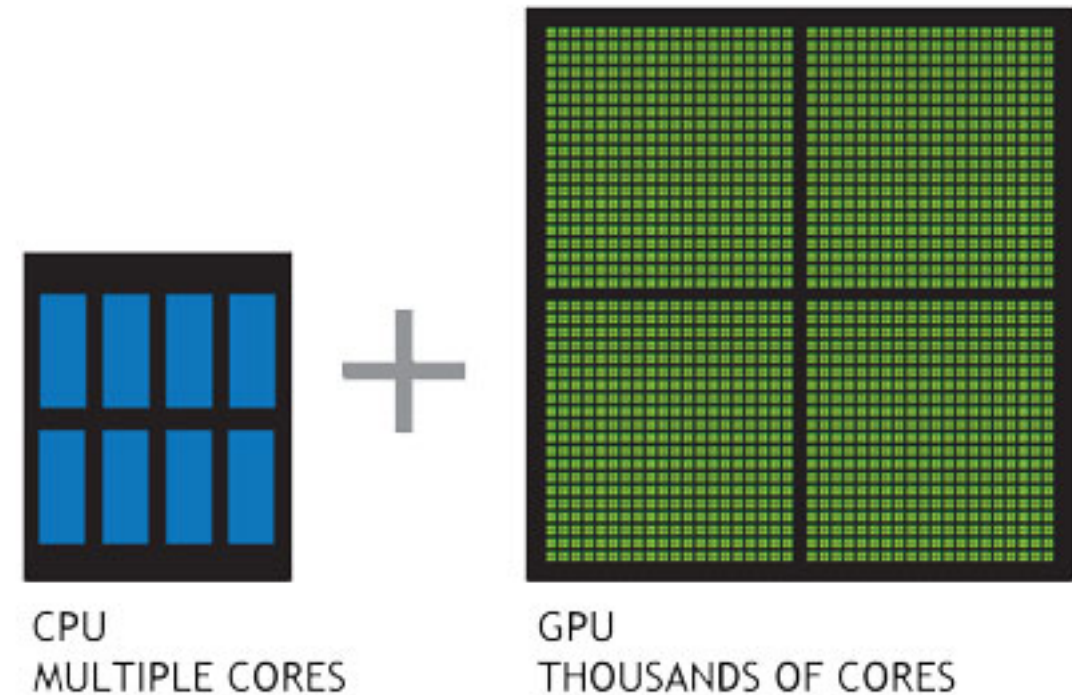
Intel / AMD / ARM ?

- From 1 core to Multi-Core
 - Core 2, Duo, Athlon X2, Cortex A9,....
- Many Core (8-30)
 - Xeon, EPIC, Ryzen, Phi, Vulcan



GPU ?

- Huge potential...
 - In C 'variant'
 - CUDA / OpenCL
 - Requires code reorganization
 - Not trivial
- Yet....
 - KEY PREMISE
 - Understand C
 - Multi-threaded programming





Expectations: Learned Outcomes

- C programming skills
- Inter-process concurrency
 - UNIX APIs
 - Pipes
 - Sockets
 - Shared memory
- Intra-process concurrency
 - POSIX threading model
 - Synchronization

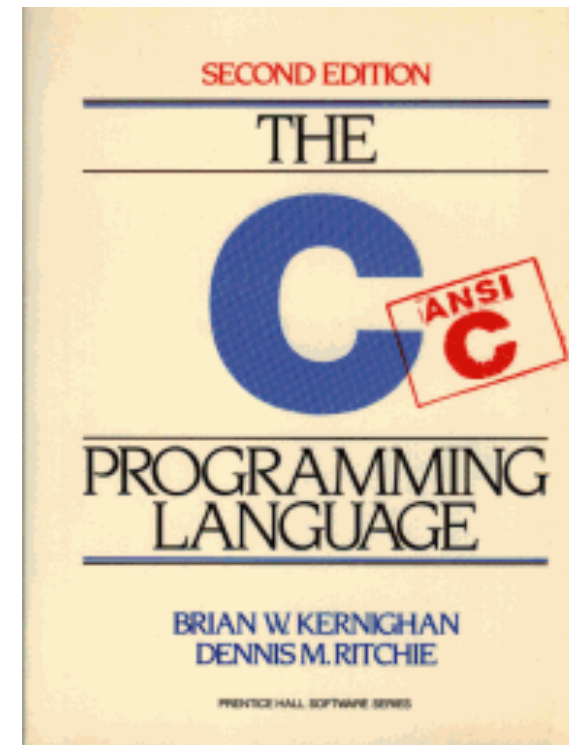
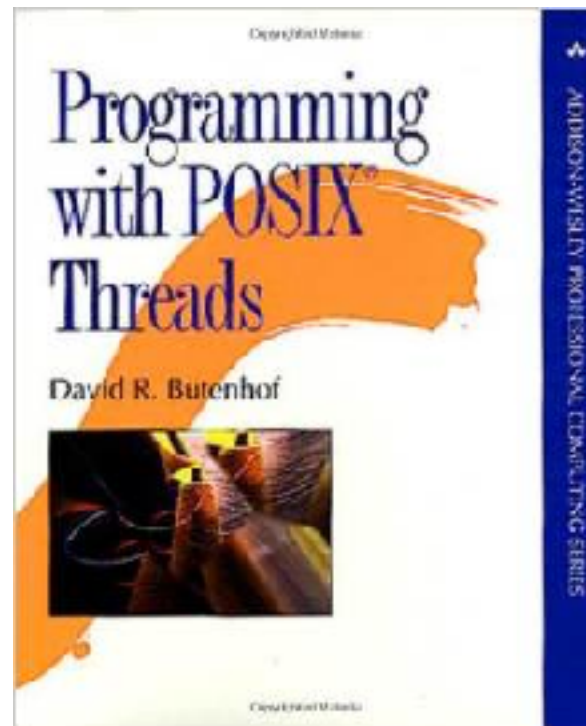


Organization

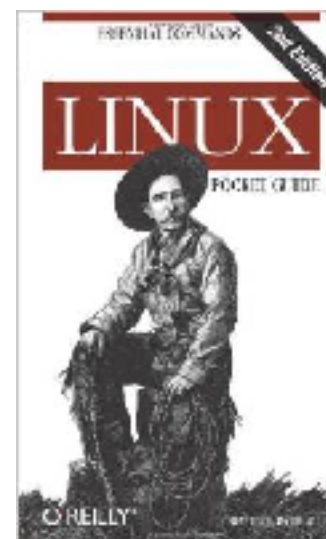
- **Weekly lectures (Asynchronous Videos)**
 - Background material. Watch *before* coming to class
- **Weekly Discussion Sessions**
 - Discussion / Q&A
 - Live coding exercises
- **Weekly labs**
 - Hands-on with concepts and tools (Squads)
 - Setup the problem sets
- **Weekly problem sets**
 - In-depth exercises

Textbooks

- Required



- Recommended





Resources

- Everything will be done on...

UNIX

- We will use virtual machines
 - Provided by UITS
 - Address: <yourNetID>.cse.uconn.edu
 - Access via ssh: ssh <yourNetID>@<yourNetID>@cse.uconn.edu



Why ?

- Simple
- Small
- Valuable
- Tutorials available online
 - <http://www.tutorialspoint.com/unix/>
 - <http://www.ee.surrey.ac.uk/Teaching/Unix/>
 - <http://people.ischool.berkeley.edu/~kevin/unix-tutorial/toc.html>
 - <http://manuals.its.virginia.edu/unixtut/>



Not only that...

- We will do most things through CLI

CLI = Command Line Interface

- Most thing means
 - Manipulate files
 - Compile
 - Debug
 - (edit)



Editing

- **Three options**

- On the guest at the command line with
 - vi <http://www.vim.org>
 - nano <https://www.nano-editor.org>
 - **emacs** <https://www.gnu.org/software/emacs>



- **Recommendations**

- Format your code properly!
- That is easy with any decent text editors
- Use syntax highlighting



IDE Be Gone!

- **Yes...**
 - You will not use any “Integrated Development Environment”
- **Reason**
 - They hide far too many things for beginners
 - Part of the process appear to be magical for newbies
 - Contribute to false sense of confidence
 - Distract from what you should focus on: **THE CODE**



Overview

- Learning Objectives
- Organization
- Resources
- Evaluation

Disclaimer



In other words.... Beware





What you will gain from the class

- **Solid understanding of**
 - Key CS programming language
 - C is the foundation for C++/C#/Objective-C/Java
 - UNIX
 - Process management and concurrency
 - POSIX APIs
 - Multi-threaded programming



What is beyond the class

- GPU Programming

- But this is a pre-requisite to go there

- Message Passing Interface (MPI)

- An often-used library for large distributed infrastructures

- Parallel Virtual Machine (PVM)

- Another contender with a different memory model for distributed infrastructures

- Languages with direct support for parallel computing

- openMP
- Cilk

All are much easier to approach
after completing this class
and can easily be summer projects!

Evaluation



	Weight
In-Lab assignments	10%
At-Home assignments	20%
Midterm 1	20%
Midterm 2	20%
Final / Project	30%
Total	100%



Tentative Schedule

Week	Topics
1	Intro, Overview of C
2	Basic Data Types
3	Control Flow, Functions
4	Compound Types (Arrays, Structures). Pointers!
5	Input / Output and some Loose Ends
6	Processes and Pipes
7	Signals and Sockets. Socket Selection/multiplexing
8	Client-server applications (e.g., firefox - apache)
9	Virtual Memory, Shared Virtual Memory and Shared Semaphores
10	Threads
11	Synchronization 1 (mutexes, spinlocks, conditions)
12	Synchronization 2 (read-write locks, semaphores and barriers)
13	Thread Local Storage, Thread Cancellation
14	Producer-consumers, Thread usage patterns.



Moodle

- Moodle: <http://courses.engr.uconn.edu/moodle>
 - You are encouraged to check it frequently for reading assignments, course handouts, and grades
 - Register on moodle now!
 - Class registration key: **SYSPROG**



Discord for Synchronous sessions

- Yes...
 - We are using Discord for discussions / live sessions
- Why?
 - You already know it.
 - It supports moderation.
- Actual video content ?
 - Live-stream via youtube





Asynchronous Video lectures

- Available on Youtube
 - It is a private resource
 - Links on Moodle/Discord



This is the way...

