

# CS 33: Computer Organization

Glenn Reinman  
4731G Boelter Hall  
[reinman@cs.ucla.edu](mailto:reinman@cs.ucla.edu)


Some notes adopted from Bryant and O'Hallaron

# Course Components




## Lectures

-  Higher level concepts

## Discussions




-  Applied concepts, important tools and skills for labs, clarification of lectures, exam coverage

## Labs

-  The heart of the course
-  Provide in-depth understanding of an aspect of systems
-  Programming and measurement


# More Info

## Web

-  Class web page hosted by CourseWeb
-  Copies of lectures, assignments, exams, solutions
-  Forum

## Office Hours

## Textbook

-  Randal E. Bryant and David R. O'Hallaron. "Computer Systems: A Programmer's Perspective", **3<sup>rd</sup> Edition**, Prentice Hall 2015.

# Grading

## 🌀 Exams (55%)

- 🌀 Midterm (20%)
- 🌀 Final (35%)
- 🌀 All exams are open book/open notes.

## 🌀 Labs (40%)

- 🌀 4 labs (10% each)
- 🌀 You must work alone on all labs

## 🌀 Homework (5%)

- 🌀 Electronic submission only

# Tentative Calendar

Week	Monday's	Wednesday's	Friday's
1	Intro + Bits and Bytes (1,2)	Integers (2)	Warmup Lab Due
2	Machine-Level Programming I: Basics (3)	Machine-Level Prog II: Control (3)	
3	Machine-Level Prog III: Procedures (3)	Machine-Level Prog IV: Data (3)	Data Lab Due
4	Machine-Level Prog V: Advanced Topics (3)	Floating Point (2)	
5	MIDTERM	Program Optimization (5)	Bomb Lab Due
6	The Memory Hierarchy (6)	Cache Memories (6)	
7	Concurrency (12+handouts)	Concurrency (12+handouts)	Attack Lab Due
8	Linking + Exceptions (7,8)	Virtual Memory (9)	
9	Holiday!	I/O (10)	
10	MIPS (handouts)	Review	Parallel Lab Due

 **Homework and Labs Due via CourseWeb by Midnight**

# Cheating

## 🌀 What is cheating?

- 🌀 Sharing code: either by copying, retyping, looking at, or supplying a copy of a file.

## 🌀 What is NOT cheating?





- 🌀 Helping others use systems or tools.
- 🌀 Helping others with high-level design issues.
- 🌀 Helping others debug their code.

## 🌀 Penalty for cheating:




- 🌀 At the discretion of the Associate Dean

# Lab Facilities

## SEAS Administered Linux Machine

-  `lnxsrv.seas.ucla.edu`
-  Remote access only
  -  Use ssh to log in with your SEAS account
-  Please direct any account issues to the SEAS help desk as they are the only ones with root access on this machine

## Alternatives (Not Recommended)

-  You may use other alternatives to develop your code
-  **BUT: We will test on the SEAS machines**
  -  **Your code must work correctly on these machines for credit**

# Course Theme

- ⌚ **Abstraction is good, but don't forget reality!**

- ⌚ **Abstractions have limits**

- ⌚ Things are more complex in hardware than they look in C/Java!!
- ⌚ Bugs are hard to track/understand if looking only from a high-level point of view

- ⌚ **Useful outcomes**

- ⌚ Become more effective programmers
  - ⌚ Able to find and eliminate bugs efficiently
  - ⌚ Able to tune program performance
- ⌚ Prepare for later “systems” classes in CS
  - ⌚ Compilers, Operating Systems, Networks, Computer Architecture, Parallel Programming



# The Compilation System

```
#include <stdio.h>

int main()
{
    printf("hello, world\n");
}
```

