Welcome To... The New...

18-600 “Foundations of Computer Systems” (Fall 2017)

Instructors:
John P. Shen & Gregory Kesden

Head TAs:
Abhinav Jauhri & Gautam Arakalgud
Lecture 1: “Course Introduction & Overview”

John P. Shen & Gregory Kesden
August 28, 2017

- Required Reading Assignment:
  - Chapter 1 of CS:APP (3rd edition) by Randy Bryant & Dave O’Hallaron

- Assignments for This Week:
  - Check out our Piazza site: https://piazza.com/cmu/fall2017/18600/home
  - Complete the short survey: https://goo.gl/forms/vxD83w75bgyuONlg2
  - If you are still deciding on taking this course, please decide this week.
Lecture 1:
“Course Introduction & Overview”

1. Course Introduction
   a. The New 18-600 FCS
   b. Teaching & Support Staff
   c. Course Organization
   d. Course Policy

2. Course Overview
   a. Tour of Computer Systems
   b. Lab Assignments Overview
What Is 18-600 (FCS)? ... starting with 15-513 ...

Introduction to Computer Systems
Underlying principles for hardware, software, and networking (ICS+)

15/18-213
15-513

CS 122
Imperative
Programming
\[ 18-600 = 15-513 + (15-349) + 18-640/abridged \]
Course Assumptions and Expectations

➢ Who should take 18-600?
  ➢ Graduate students (MS/PhD in ECE, MS in INI)
    o Applications and systems programming; Broad computing systems expertise
    o Computer systems design and development; Computer architect’s mindset

➢ Assumed undergraduate background:
  ➢ C/C++ programming & Unix operating systems experience
  ➢ Digital logic design, and computer organization BS courses
  ➢ Assembly language (preferably x86) programming exposure

➢ Course expectations:
  ➢ Focusing on foundational principles and key insights; in-class interactions encouraged
  ➢ Emphasis on hands-on lab assignments to gain deeper understanding and personal skills
  ➢ Assume self motivated and disciplined students with professional integrity and attitude
Course Objectives and CMU Distinctives

➢ **Smarts**
  ❖ **Broad Knowledge Base:** What and how much you know.

➢ **Skills**
  ❖ **Superb Hands-on Builder:** What you can do and implement.

➢ **Sense**
  ❖ **Great Insights & Intuition:** How you think and solve problems.

➢ **Savvy**
  ❖ **High Industry Awareness:** How you come across and interact.
18-600  Cast of Characters:

➢ Instructors:
  ▪ John P. Shen (SV)
  ▪ Gregory Kesden (PGH)

➢ Academic Services Assistants:
  ▪ Michelle Mahouski (PGH)
  ▪ Brittany Jade Reyes (SV)

➢ Head Teaching Assistants:
  ▪ Abhinav Jauhri (SV)
  ▪ Gautam Arakalgud (PGH)
18-600 Cast of Characters:

Teaching Assistants (PGH):
- Jithin Yaratapalli (Sec. A)
- Sampath Chanda (Sec. A)
- Akanksha Periwal (Sec. B)
- Gautam Arakalgud (Sec. B)
- Abhiroop Kaginalkar (Sec. C)
- Prerit Rodney (Sec. C)
- Harish Dattatraya Dixit (Sec. D)
- Mani Swetha Mandava (Sec. D)

Teaching Assistants (SV):
- Daniel Min-Hao Chen (Sec. SA)
- Siyang Mai (Sec. SA)
- Abhinav Jauhri (Sec. SB)
Prof. John Paul Shen:

➢ Academia (1982-2000)
  ○ Carnegie Mellon University
    ○ Computer Aided Design
    ○ Computer Architecture

  ○ Intel, Research Lab
    ○ Superscalar/Multicore Processors
  ○ Nokia, Research Center
    ○ Mobile/Cloud Computing Systems

➢ Academia (2015-present)
  ○ Carnegie Mellon University (Silicon Valley Campus)
    ○ Human Mobility Analytics and Services (HUMANS)
Prof. Gregory Kesden:

➢ Academia (1998-2017)
  • Computer Science Department, Clemson University
    ▪ 1998-1999: Introductory courses, data structures, databases
  • School of Computer Science (SCS), CMU
    ▪ 1999-2015: Distributed systems, networking, operating systems, computer systems, databases, etc.
  • Computer Science and Engineering (CSE), UCSD
    ▪ 2015-2017: Operating systems, cloud computing, software engineering, introductory courses, etc.
  • Information Networking Institute (INI), CMU
    ▪ 2017-death: Cloud computing, distributed systems, networking, computer systems, etc.

➢ Trivia
  • Firearms instructor, EMT, owner/pilot 42’ ocean trawler
Textbooks: Two Required, Two Optional

**Required Textbooks:**

1. Randal E. Bryant and David R. O’Hallaron,
   - [http://csapp.cs.cmu.edu](http://csapp.cs.cmu.edu)
   - This book really matters for the course!
     - How to solve labs
     - Practice problems typical of exam problems

2. Brian Kernighan and Dennis Ritchie,
   - Still the best book about C, from the originators

**Recommended References:**

1. [Optional] John P. Shen and Mikko Lipasti, (supplement to CS:APP Chapter 4)

2. [Optional] Michel Dubois, Murali Annavaram, and Per Stenstrom
# Class Schedule – Fall 2017

<table>
<thead>
<tr>
<th>Lecture:</th>
<th>Labs/Recitation:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lectures, Section A:</strong>&lt;br&gt;MW, 6:30pm to 8:20pm (ET), DH A302</td>
<td><strong>Recitation, Section A:</strong>&lt;br&gt;T, 7:30pm to 8:50pm (ET), HH 1107</td>
</tr>
<tr>
<td><strong>Lectures, Section B:</strong>&lt;br&gt;MW, 6:30pm to 8:20pm (ET), DH A302</td>
<td><strong>Recitation, Section B:</strong>&lt;br&gt;T, 7:30pm to 8:50pm (ET), GHC 4102</td>
</tr>
<tr>
<td><strong>Lectures, Section C:</strong>&lt;br&gt;MW, 6:30pm to 8:20pm (ET), DH A302</td>
<td><strong>Recitation, Section C:</strong>&lt;br&gt;T, 5:30pm to 6:50pm (ET), WEH 4623</td>
</tr>
<tr>
<td><strong>Lectures, Section D:</strong>&lt;br&gt;MW, 6:30pm to 8:20pm (ET), DH A302</td>
<td><strong>Recitation, Section D:</strong>&lt;br&gt;T, 5:30pm to 6:50pm (ET), WEH 5320</td>
</tr>
<tr>
<td><strong>Lectures, Section SA:</strong>&lt;br&gt;MW, 3:30pm to 5:20pm (PT), B23 118</td>
<td><strong>Recitation, Section SA:</strong>&lt;br&gt;T, 4:30pm to 5:50pm (PT), B23 118</td>
</tr>
<tr>
<td><strong>Lectures, Section SB:</strong>&lt;br&gt;MW, 3:30pm to 5:20pm (PT), B23 211</td>
<td><strong>Recitation, Section SB:</strong>&lt;br&gt;T, 4:30pm to 5:50pm (PT), B23 109/110</td>
</tr>
</tbody>
</table>
Course Components

- Lectures (27)
  - Higher level and foundational concepts

- Recitations (14)
  - Applied concepts, important tools and skills for labs, clarification of lectures, exam coverage

- Labs (7)
  - The heart of the course
  - ~2 weeks for each lab assignment
  - Provide in-depth understanding of an aspect of computer systems
  - Programming, measurement, and analysis

- Exams (Midterm + Final)
  - Test your understanding of concepts, key principles, and specific techniques
# Course Grading Distribution

<table>
<thead>
<tr>
<th></th>
<th>LAB Assignments</th>
<th>50%</th>
<th>(7) Individual lab assignments with varying weights. Will allow teams of two for Lab Assignments 5-7.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RECITATIONS</strong> (Led by TA's)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LECTURES</strong> (Instructors)</td>
<td>Mid-Term EXAM</td>
<td>20%</td>
<td>In class Exam (110 minutes) covering Lectures 1-15, and Lab Assignments 1-4.</td>
</tr>
<tr>
<td></td>
<td>Final EXAM</td>
<td>30%</td>
<td>In class Exam (180 minutes) covering Lectures 16-27, and Lab Assignments 5-7.</td>
</tr>
<tr>
<td><strong>EXTRA CREDITS</strong></td>
<td>Class Participation Online Contribution</td>
<td>5%</td>
<td>Active participation in lectures and recitations. Active contribution in Piazza Q&amp;A discussions.</td>
</tr>
</tbody>
</table>
Course Policies: Labs And Exams

• Lab work
  • You must work alone on Lab Assignments.
  • Will allow teams of two for Lab Assignments 5-7.

• Hand-ins
  • Labs are due at 11:59pm (PT) usually on a Thursday or Friday
  • Electronic handins using Autolab (no exceptions!)

• Exams
  • Exams will be held in class

• Appealing grades
  • Talk to one of the TAs first with possible escalation to the instructors
Cheating: Description

• Please pay close attention, especially if this is your first semester at CMU

• What is cheating?
  • Sharing code: by copying, retyping, looking at, or supplying a file
  • Describing: verbal description of code from one person to another.
  • Coaching: helping your friend to write a lab, line by line
  • Searching the Web for solutions
  • Copying code from a previous course or online solution
    • You are only allowed to use code we supply, or from the CS:APP website

• What is NOT cheating?
  • Explaining how to use systems or tools
  • Helping others with high-level design issues

• See the course syllabus for details.
  • Ignorance is not an excuse
Cheating: Consequences

• Penalty for cheating: (No Exceptions!)
  • Any cheating on an assignment will result in zero credit for that assignment.
  • Repeated cheating will result in removal from course with failing grade.
  • Any cheating will leave a permanent negative mark on your record at CMU, results in the immediate loss of scholarship money for INI students (even for the 1st offense), and could even lead to being expelled from CMU.

• Detection of cheating:
  • We have very sophisticated tools for detecting code plagiarism; don’t test us.
  • Last Fall, a handful of students were caught cheating and failed the course.

• Just don’t do it!
  • Start early
  • Ask the staff for help when you get stuck
Getting Help

• Class Web page: http://ece.cmu.edu/~ece600/
  • Complete schedule of lectures, exams, and assignments
  • Copies of lectures, assignments, exams, solutions
  • Clarifications to assignments
  • The afs directory for 18-600 is at: /afs/ece.cmu.edu/class/ece600

• We will use Piazza in this course for communication: https://piazza.com/cmu/fall2017/18600/home

• Office Hours:
  • Recitations: other than presenting planned material there is time for Q&A
  • Each TA will have weekly office hours beyond the recitation sessions (TBA)
  • If necessary send email to your TA to arrange a special help session
Lecture 1: “Course Introduction & Overview”

1. Course Introduction
   a. Birth of the New 18-600
   b. Teaching & Support Staff
   c. Course Organization
   d. Course Policy

2. Course Overview
   a. Tour of Computer Systems
   b. Lab Assignments Overview
Anatomy of a Computer System: SW/HW

What is a Computer System?

- Software + Hardware
- Programs + Computer \(\Rightarrow\) [Application program + OS] + Computer
- Programming Languages + Operating Systems + Computer Architecture

Diagram:

- COMPILER
- OS
- ARCHITECTURE

Table:

<table>
<thead>
<tr>
<th>Application programs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
</tr>
<tr>
<td>Processor</td>
</tr>
<tr>
<td>Memory</td>
</tr>
<tr>
<td>I/O devices</td>
</tr>
</tbody>
</table>

Software (programs)

Hardware (computer)
Anatomy of a Computer System: Compiler

Application programs

Operating system

Processor | Memory | I/O devices

Software (programs)

Hardware (computer)
Anatomy of a Computer System: OS

User Mode

Kernel Mode

Computer

Application programs

Operating system

Processes

Virtual memory

Files/NIC

Processor

Memory

I/O devices

System calls

Upcalls

Commands

Interrupts

CS:APP Ch. 2 & 3

CS:APP Ch. 8 & 9

CS:APP Ch. 4 & 5

CS:APP Ch. 6, 9, 10

User Mode

Kernel Mode

Computer

Application programs

Operating system

Processes

Virtual memory

Files/NIC

Processor

Memory

I/O devices

System calls

Upcalls

Commands

Interrupts

CS:APP Ch. 2 & 3

CS:APP Ch. 8 & 9

CS:APP Ch. 4 & 5

CS:APP Ch. 6, 9, 10

8/28/2017 (©J.P. Shen)
Operating System Abstractions

Abstraction 1: Processes
- application: application
- OS: process
- hardware: computer

Abstraction 2: Virtual memory
- application: address space
- OS: virtual memory
- hardware: physical memory

Abstraction 3: File System
- application: copy file1 file2
- OS: files, directories
- hardware: disk

Abstraction 4: Messaging
- application: sockets
- OS: TCP/IP protocols
- hardware: network interface
What is a Computer?

- **The Classic Von Neumann Computation Model:** Proposed in 1945 by John Von Neumann and others (Alan Turing, J. Presper Eckert and John Mauchly).

- **A “Stored Program Computer”**
  1. **One CPU**
     - One Control Unit
     - Program Counter
     - Instruction Register
     - One ALU (Data Path)
  2. **Monolithic Memory**
     - Data Store
     - Instruction Store
  3. **Sequential Execution Semantics**
     - Instructions from an Instruction Set
Typical Computer (PC) Today: HW Organization

CPU
- Register file
- ALU
- Bus interface

Main memory

I/O bridge
- System bus
- Memory bus

I/O bus
- Expansion slots for other devices such as network adapters

Disk controller
- Disk
  - hello executable stored on disk

Other components:
- USB controller
- Graphics adapter
- Mouse
- Keyboard
- Display

CS:APP Chapter 4
CS:APP Chapter 5
CS:APP Chapter 6
CS:APP Chapter 7
CS:APP Chapter 9
CS:APP Chapter 10

8/28/2017 (©J.P. Shen)
Example Memory Hierarchy

- **CPU registers** hold words retrieved from cache memory.
- **L1 cache** holds cache lines retrieved from the L2 cache.
- **L2 cache** holds cache lines retrieved from L3 cache.
- **L3 cache** holds cache lines retrieved from memory.
- **Main memory** holds disk blocks retrieved from local disks.
- **Local disks** hold files retrieved from disks on remote network servers.
- **Remote secondary storage** (distributed file systems, Web servers).
- **Local secondary storage** (local disks).
- **Main memory** (DRAM).
- **L2 cache** (SRAM).
- **L1 cache** (SRAM).
- **CPU registers** (SRAM).

- Smaller, faster, and costlier (per byte) storage devices
- Larger, slower, and cheaper (per byte) storage devices

8/28/2017 (©J.P. Shen)
Reading “hello” command from the keyboard

Canvas diagram showing the flow of information from the user (keyboard) to the main memory. The diagram includes components such as the CPU, Register file, ALU, System bus, Memory bus, I/O bridge, Main memory, I/O bus, and various peripherals like the USB controller, graphics adapter, disk controller, mouse, keyboard, and display. The user types “hello”, which is read from the keyboard and stored in main memory.
Loading executable from disk to main memory

Main memory

I/O bus

Expansion slots for other devices such as network adapters

Disk controller

Disk

"hello, world"

"hello, world"

hello code

Translation of code to instructions

Main memory

I/O bridge

System bus

Memory bus

CPU

Register file

ALU

PC

Bus interface

USB controller

Graphics adapter

Mouse Keyboard

Display

I/O bus

Disk

hello executable stored on disk

CS:APP
Chapter 10

"hello, world"

CMU
Writing output string from memory to display

CPU

Register file

ALU

System bus

Memory bus

I/O bridge

Main memory

"hello, world"

"hello, world"

hello code

I/O bus

Expansion slots for other devices such as network adapters

USB controller

Mouse, Keyboard

Graphics adapter

Display

Disk controller

Disk

hello executable stored on disk

PC

Bus interface

System bus

Memory bus
Network interface is another I/O device
Multicore Processor Organization (TLP)

Processor package

Core 0
- Regs
- L1 d-cache
- L1 i-cache
- L2 unified cache

Core 3
- Regs
- L1 d-cache
- L1 i-cache
- L2 unified cache

... (shared by all cores)

Main memory

CS:APP
Chapter 12
Lab Assignments Overview

7 Lab Assignments

• **L1 (Data Lab)**: Manipulating bits
• **L2 (Bomb Lab)**: Defusing a binary bomb
• **L3 (Arch Lab)**: Processor design & performance improvements
• **L4 (Shell Lab)**: Writing your own Unix shell.
• **L5 (Cache Lab)**: Cache optimization & cache coherence
• **L6 (Malloc Lab)**: Write your own malloc package
• **L7 (Proxy Lab)**: Write your own Web proxy
Data and Programs

• Topics
  • Bits operations, arithmetic, assembly language programs
  • Representation of C control and data structures
  • Includes aspects of architecture and compilers

• Assignments
  • L1 (Data Lab): Manipulating bits
  • L2 (Bomb Lab): Defusing a binary bomb
Processor Architecture

• Topics
  • Pipelined processor design and performance
  • Superscalar and Out-of-order processor designs
  • Performance and Power tradeoffs

• Assignments
  • L3 (Arch Lab): Processor design & performance improvements
    • Learn how to design modern processors
Exceptional Control Flow

• Topics
  • Hardware exceptions, processes, process control, Unix signals, nonlocal jumps
  • Includes aspects of compilers, OS, and architecture

• Assignments
  • L4 (Shell Lab): Writing your own Unix shell.
    • A first introduction to concurrency
Memory Hierarchy

- **Topics**
  - Memory technology, memory hierarchy, caches, disks, locality
  - Multi-core cache coherence, multi-threaded workloads
  - Includes aspects of architecture and OS

- **Assignments**
  - **L5 (Cache Lab):** Cache optimization & cache coherence
    - Learn how to exploit locality in your programs.
Virtual Memory

• Topics
  • Virtual memory, address translation, dynamic storage allocation
  • Includes aspects of architecture and OS

• Assignments
  • L6 (Malloc Lab): Writing your own malloc package
    • Get a real feel for systems-level programming
Networking and Concurrency

- **Topics**
  - High level and low-level I/O, network programming
  - Internet services, Web servers
  - Concurrency, concurrent server design, threads
  - I/O multiplexing with select
  - Includes aspects of networking, OS, and architecture

- **Assignments**
  - **L7 (Proxy Lab)**: Writing your own Web proxy
    - Learn network programming and more about concurrency and synchronization.
Timeliness on Lab Assignments

• Grace Days
  • 5 grace days total for the semester
  • Limit of 2 grace days per lab, used automatically
  • Covers scheduling crunch, out-of-town trips, illnesses, minor setbacks, etc.
  • Save them until late in the semester!

• Lateness Penalties
  • Once grace day(s) are used up, will get penalized 10% per day late
  • No hand-ins later than 3 days after due date

• Advice
  • Once you start running late, it’s really hard to catch up!!!
Lecture 2: “Computer Systems Big Picture”

John P. Shen & Gregory Kesden
August 30, 2017

Next Time ...