Welcome To... The New...

18-600 "Foundations of Computer Systems" (Fall 2016)

Instructors:
John P. Shen & Zhiyi Yu

Head TAs:
Jenna MacCarley & Preeti Murthy

Electrical & Computer ENGINEERING
Carnegie Mellon University

18-600 Foundations of Computer Systems

Lecture 1: "Course Introduction & Overview"

John P. Shen & Zhiyi Yu
August 29, 2016

- 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 http://piazza.com/cmu/fall2016/18600/home
  - Complete the short survey: https://goo.gl/forms/mX2inGxYVdaqyk8b2
  - If you are still deciding on taking this course, please decide this week.

Electrical & Computer ENGINEERING
Carnegie Mellon University

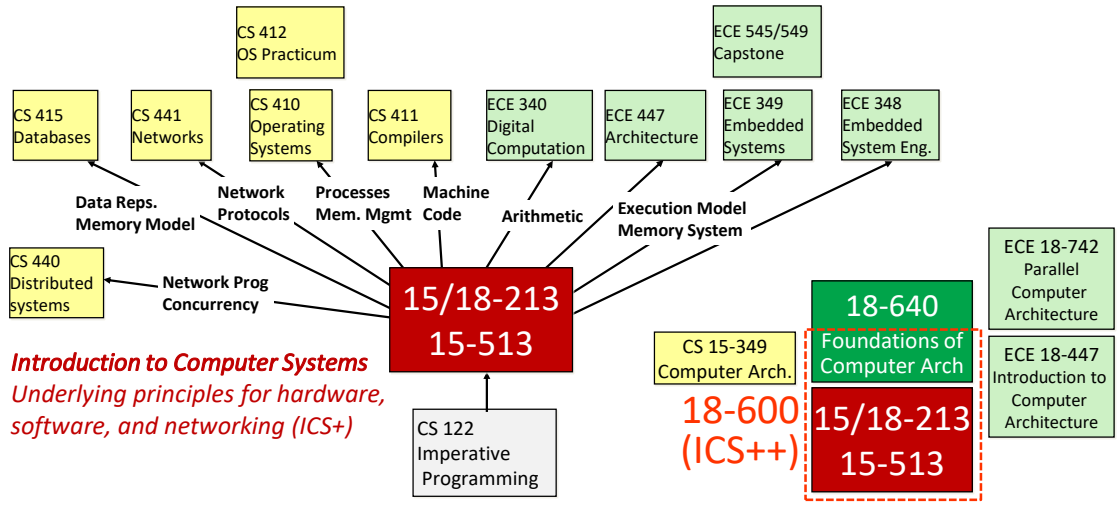
18-600 Foundations of Computer Systems

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

Electrical & Computer ENGINEERING

Why Create 18-600? What Is 18-600?



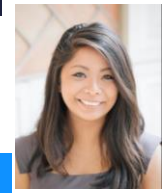
Course Assumptions and Expectations

- Who should take 18-600?
  - Graduate students (MS/PhD in ECE)
    - Applications and systems programming; Broad computing systems expertise
    - Computer systems design & development; Strong computer architect's mindset
- Assumed undergraduate background:
  - HLL programming, and some assembly language exposure
  - Digital logic design, and computer organization notions
  - C/C++ programming, and Unix operating system 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

18-600 Cast of Characters:

- Instructors:
  - John P. Shen (SV)
  - Zhiyi Yu (GZ)

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



- JIE Course Coordinator:
  - Xiaobai Chen (GZ) <chenxb29@mail2.sysu.edu.cn>

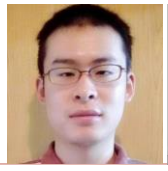
18-600 Cast of Characters:

- Head Teaching Assistants:
  - Jenna MacCarley (SV) (Sec. SB)
  - Preeti Murthy (PGH) (Sec. B)
- Teaching Assistants:
  - Mila Kankanala (PGH) (Sec. A)
  - Steffi Dsouza (PGH) (Sec. A)
  - Kevin Xu (SV) (Sec. SA)
  - Abhinav Jauhri (SV) (Sec. SA)
  - Kangyi Lu (PGH) (Sec. GZ)
  - Zhen Hu (PGH)) (Sec. GZ)










Prof. Zhiyi Yu:

- Research Interests:
  - Digital VLSI and Computer Architecture
- Education and Working Experience:
  - SYSU-CMU Joint Institute of Engineering 2015 –
  - Microelectronics Department, Fudan University, China 2009-2014
  - IntellaSys Corporation, USA 2007-2008
  - Ph.D. in Electrical and Computer Engineering, University of California, Davis 2007
  - M.S. in Electrical Engineering, Fudan University, China 2003
  - B.S. in Electrical Engineering, Fudan University, China 2000

Prof. John P. Shen:

- Academia (15 years)
  - Carnegie Mellon University
    - First Half → Computer Aided Design
      - Sabbatical at Stanford
    - Second Half → Computer Architecture
      - Sabbatical at Intel
- Industry (15 years)
  - Intel, Research Lab
    - Microarchitecture Lab → Microprocessor Design
  - Nokia, Research Center
    - North America Lab → Mobile Computing Systems
- Academia (Fall 2015)
  - Carnegie Mellon University (SV Campus)

Textbooks: Two Required, One Optional

- Randal E. Bryant and David R. O'Hallaron,
  - *Computer Systems: A Programmer's Perspective*, **Third Edition** (CS:APP3e), Pearson, 2016
  - http://csapp.cs.cmu.edu
  - This book really matters for the course!
    - How to solve labs
    - Practice problems typical of exam problems
- Brian Kernighan and Dennis Ritchie,
  - *The C Programming Language*, Second Edition, Prentice Hall, 1988
  - Still the best book about C, from the originators

- [Optional] John P. Shen and Mikko Lipasti, (supplement to CS:APP Chapter 4)
  - *Modern Processor Design: Fundamentals of Superscalar Processors,* 2005; reissued by Waveland Press Inc , 2013. ISBN 10: 1478607831, ISBN 13: 9781478607830

Course Schedule – Fall 2016

Class Schedule: (CST=UTC+08, EDT=UTC–04, PDT=UTC–07, EST=UTC–05, PST=UTC–08)

Lectures :

Lectures, Section A:
MW, 8:00pm to 9:50pm (ET), DH A302

Lectures, Section B:
MW, 8:00pm to 9:50pm (ET), DH A302

Lectures, Section SA:
MW, 5:00pm to 6:50pm (PT), B23 118

Lectures, Section SB:
MW, 5:00pm to 6:50pm (PT), B23 109/110

Lectures, Section GZ:
TR, 8:00am to 9:50am (CST), JIE 214

Recitation :

Recitation, Section A:
T, 8:00pm to 9:20pm (ET), HH 1107

Recitation, Section B:
T, 8:00pm to 9:20pm (ET), DH A302

Recitation, Section SA:
T, 4:30pm to 5:50pm (PT), B23 118

Recitation, Section SB:
T, 4:30pm to 5:50pm (PT), B23 109/110

Recitation, Section GZ:
W, 8:00am to 9:20am (CST), JIE 214

Course Components

- Lectures (27)
  - Higher level concepts
- Recitations (14)
  - Applied concepts, important tools and skills for labs, clarification of lectures, exam coverage
- Labs (7)
  - The heart of the course
  - 1-2 weeks each
  - Provide in-depth understanding of an aspect of systems
  - Programming and measurement
- Exams (Midterm + Final)
  - Test your understanding of concepts & mathematical principles

Course Grading Distribution

RECITATIONS (Led by TA's)	LAB Assignments	50%	(7) Individual lab assignments with varying weights.
LECTURES (Instructors)	Mid-Term EXAM	25%	In class Exam covering Lectures 1-15, and Lab Assignments 1-4.
	Final EXAM	25%	In class Exam covering Lectures 16-27, and Lab Assignments 5-7.
EXTRA CREDITS	Class Participation Online Contribution	5%	Active participation in lectures and recitations. Active contribution in Piazza Q&A discussions.

Course Policies: Labs And Exams

- Lab work
  - You must work alone on all lab assignments
- Hand-ins
  - Labs are due at 11:59pm (PT) usually on a Thursday or Monday
  - Electronic handins using **Autolab** (no exceptions!)
- Exams
  - Exams will be held in class
- Appealing grades
  - Talk to one of the head 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:
  - Removal from course with failing grade (no exceptions!)
  - Permanent mark on your record
  - Your instructors' personal contempt
- Detection of cheating:
  - We have sophisticated tools for detecting code plagiarism
  - Last Fall, 20 students in 213/513 were caught cheating and failed the course.
  - Some were expelled from the University
- 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: http://piazza.com/cmu/fall2016/18600/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

18-600 Foundations of Computer Systems

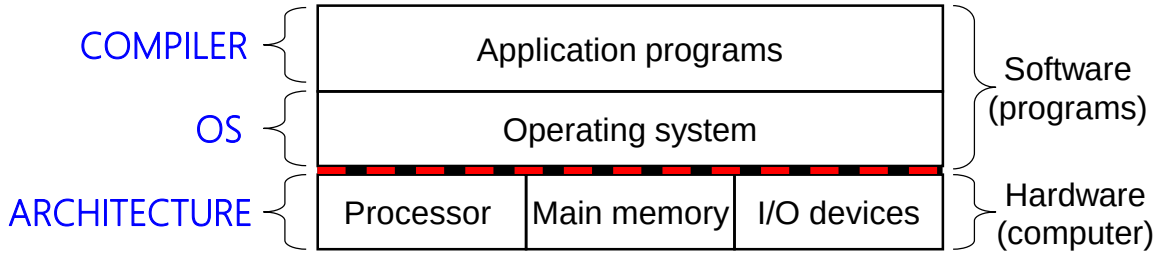
Lecture 1: "Course Introduction & Overview"

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

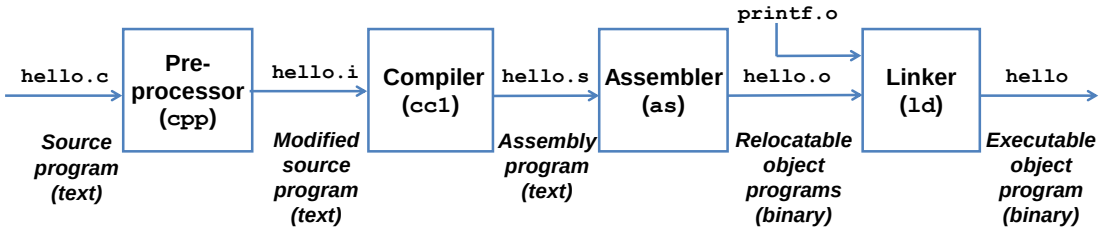
Electrical & Computer ENGINEERING

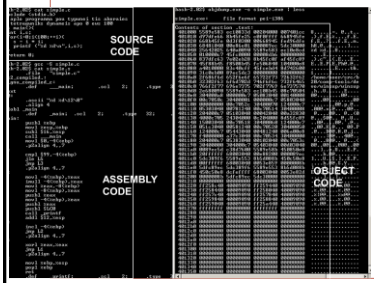
Anatomy of a Computer System: SW/HW

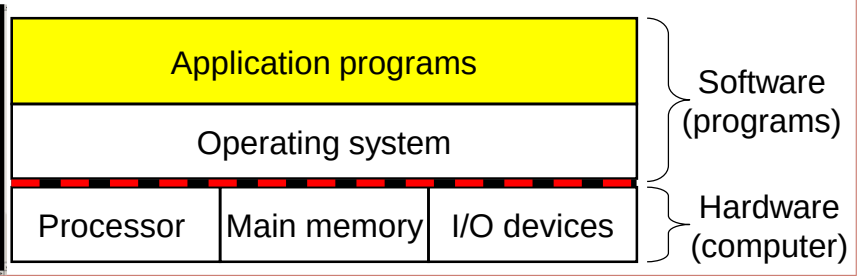
- What is a Computer System?
  - Software + Hardware
  - Programs + Computer ➔ [Application program + OS] + Computer
  - Programming Languages + Operating Systems + Computer Architecture



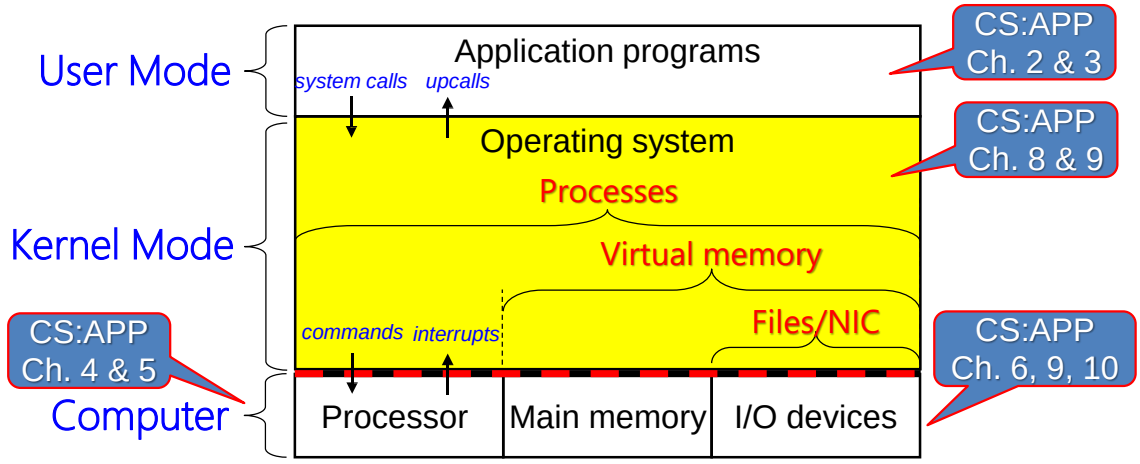
Anatomy of a Computer System: Compiler







Anatomy of a Computer System: OS



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*

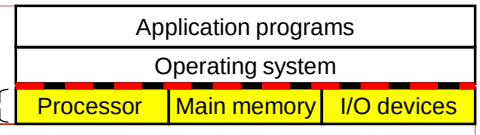
Abstraction 3: File System

application:	*copy file1 file2*
OS:	*files, directories*
hardware:	*disk*

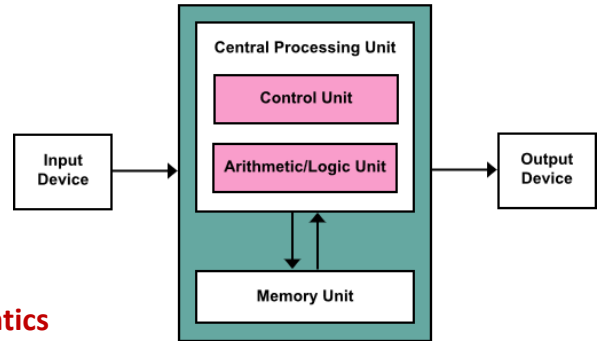
Abstraction 4: Messaging

application:	*sockets*
OS:	*TCP/IP protocols*
hardware:	*network interface*

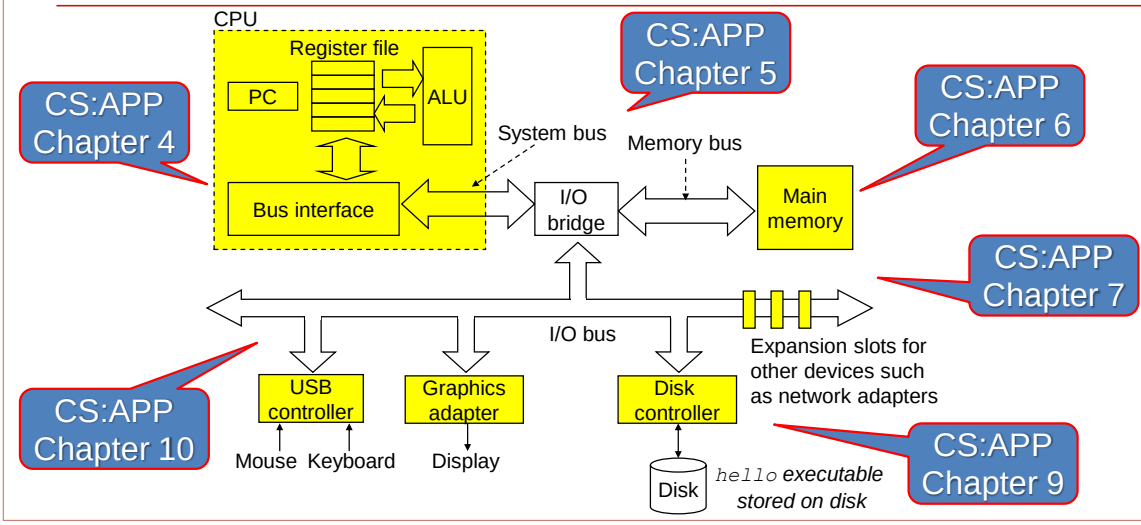
What is a Computer?



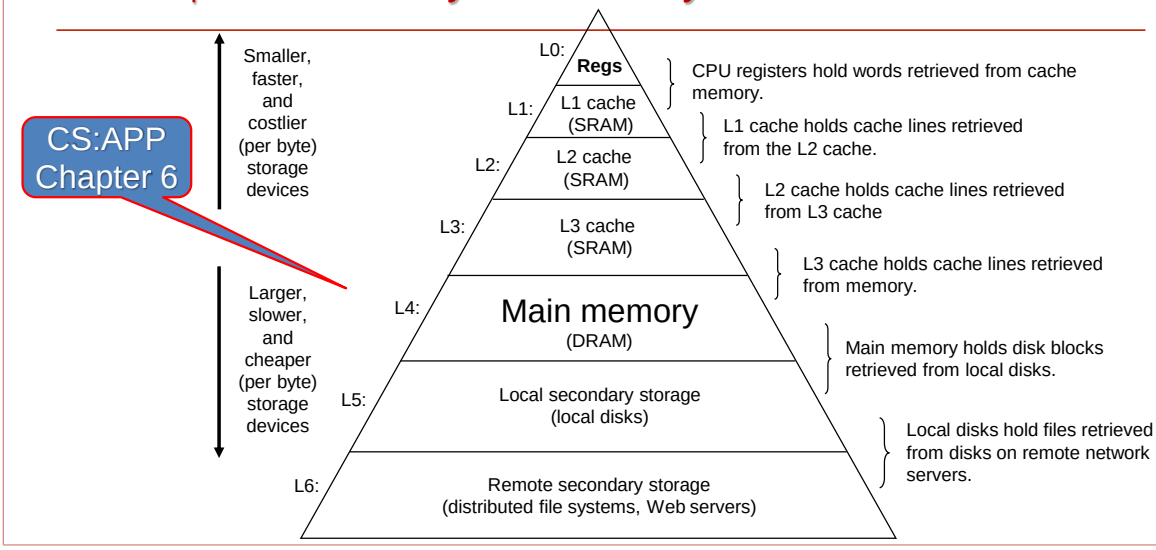
- **The Classic Von Neumann Computation Model**: Proposed in 1945 by John Von Neumann and others (Alan Turing, J. Presper Eckert and John Mauchly).
- **A "Stored Program Computer"**
  1. **One CPU**
     - One Control Unit
       - Program Counter
       - Instruction Register
     - One ALU
  2. **Monolithic Memory**
     - Data Store
     - Instruction Store
  3. **Sequential Execution Semantics**
     - Instructions from an Instruction Set



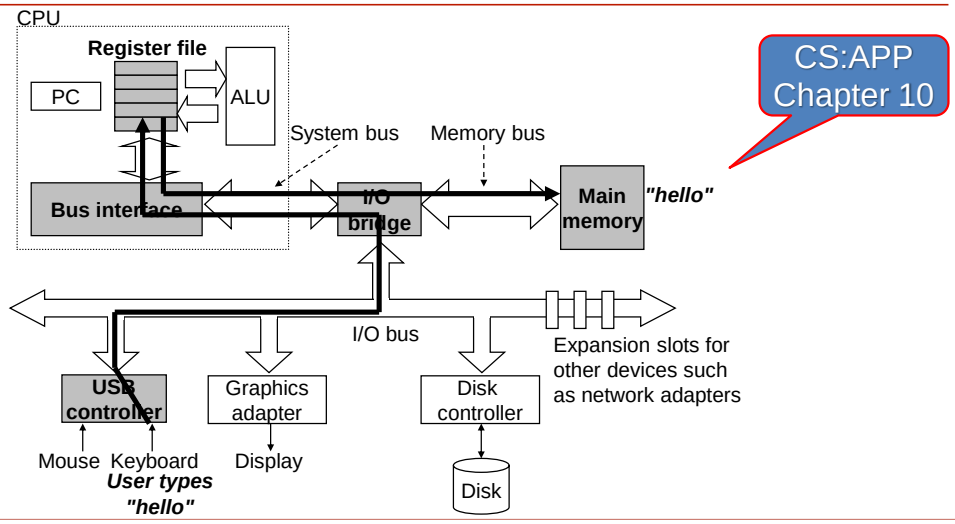
Typical Computer Today: HW Organization



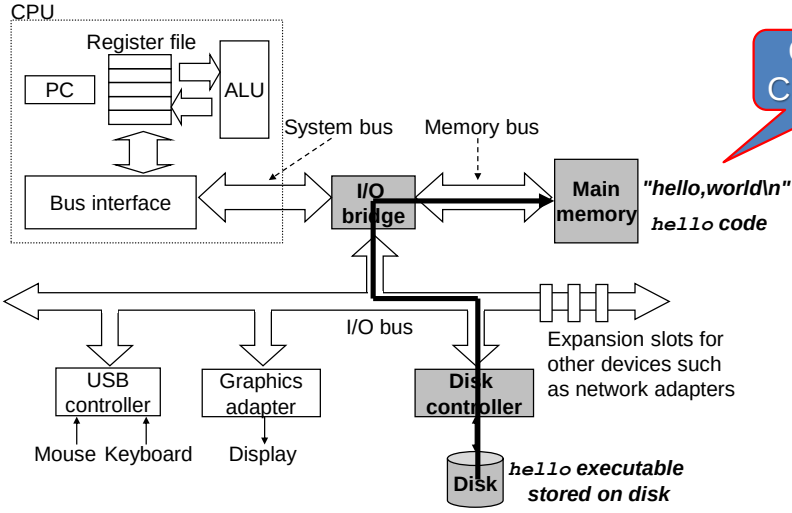
Example Memory Hierarchy



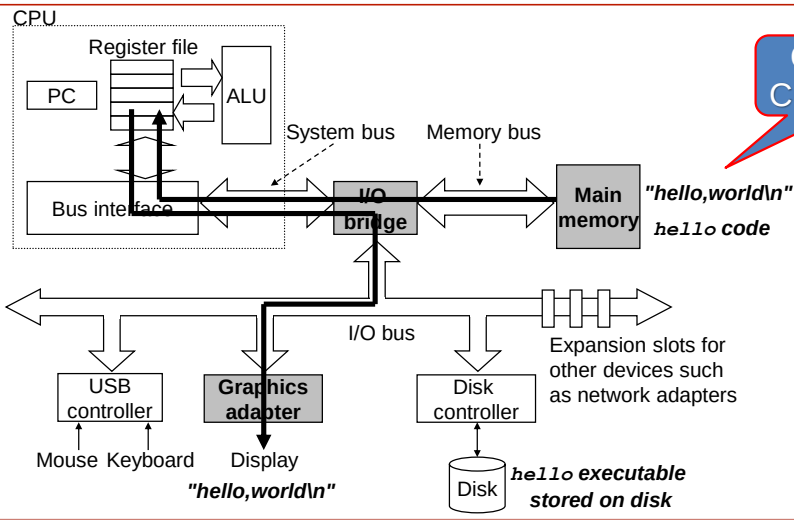
Reading "hello" command from the keyboard



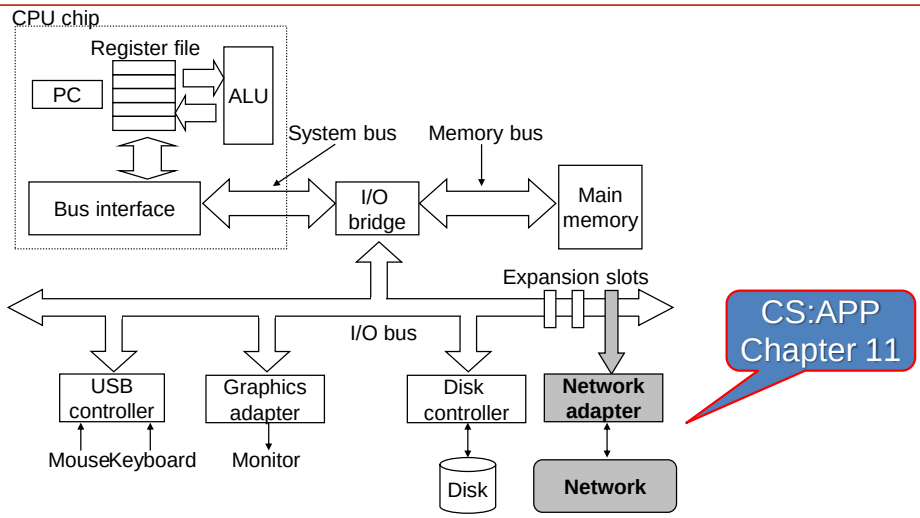
Loading executable from disk to main memory



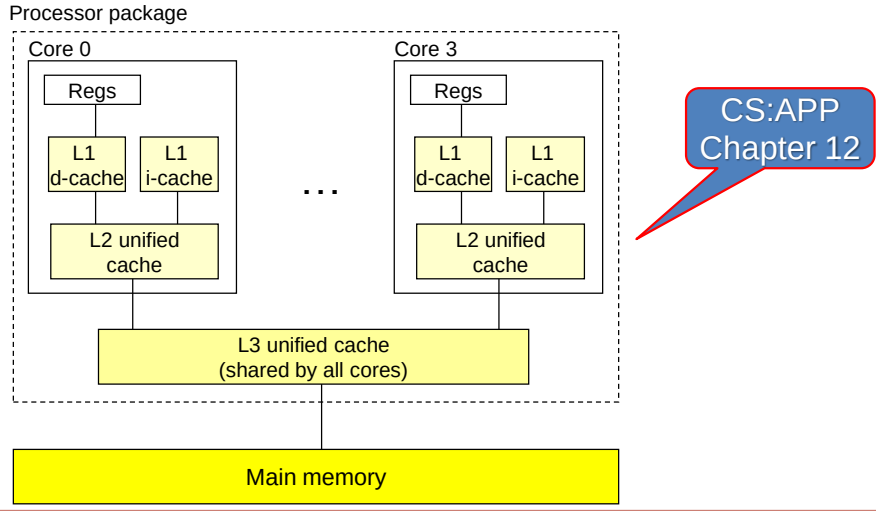
Writing output string from memory to display



Network interface is another I/O device



Multicore Processor Organization (TLP)



Lab Assignments Overview

7 Lab Assignments

- **L1 (Data Lab)**: Manipulating bits
- **L2 (Attack Lab)**: The basics of code injection attacks
- **L3 (Arch Lab)**: Pipelined processor and performance improvements
- **L4 (Cache Lab)**: Building a cache simulator and optimizing for locality
- **L5 (Shell Lab)**: Writing your own Unix shell.
- **L6 (Malloc Lab)**: Writing your own malloc package
- **L7 (Proxy Lab)**: Writing 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 (Attack Lab)**: The basics of code injection attacks

CS:APP Ch. 2

CS:APP Ch. 3

Processor and Memory Hierarchy

- Topics
  - Pipelined processor design and performance
  - Memory technology, memory hierarchy, caches, disks, locality
  - Includes aspects of architecture and OS

CS:APP Ch. 4 & 5

- Assignments
  - **L3 (Arch Lab)**: Pipeline processor design and performance improvements
  - **L4 (Cache Lab)**: Building a cache simulator and optimizing for locality.
    - Learn how to exploit locality in your programs.

CS:APP Ch. 5 & 6

Exceptional Control Flow

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

CS:APP Ch. 8 & 10

- Assignments
  - **L5 (Shell Lab)**: Writing your own Unix shell.
    - A first introduction to concurrency

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

CS:APP Ch. 9

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.

CS:APP Ch. 11 & 12

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 **15% 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!!!

18-600 Foundations of Computer Systems

Lecture 2: "Computer Systems Big Picture"

John P. Shen & Zhiyi Yu
August 31, 2016

Next Time ...

- Recommended References:
  - **Chapters 1 and 2 of Shen and Lipasti (SnL).**
  - "Amdahl's and Gustafson's Laws Revisited" by Andrzej Karbowski. (2008)

Electrical & Computer ENGINEERING