Welcome To… The New…

18-600 "Foundations of Computer Systems" (Fall 2017)

Instructors:
John P. Shen & Gregory Kesden

Head TAs:
Abhinav Jauhri & Gautam Arakalgud

Electrical & Computer ENGINEERING

18-600 Foundations of Computer Systems

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.

Electrical & Computer ENGINEERING

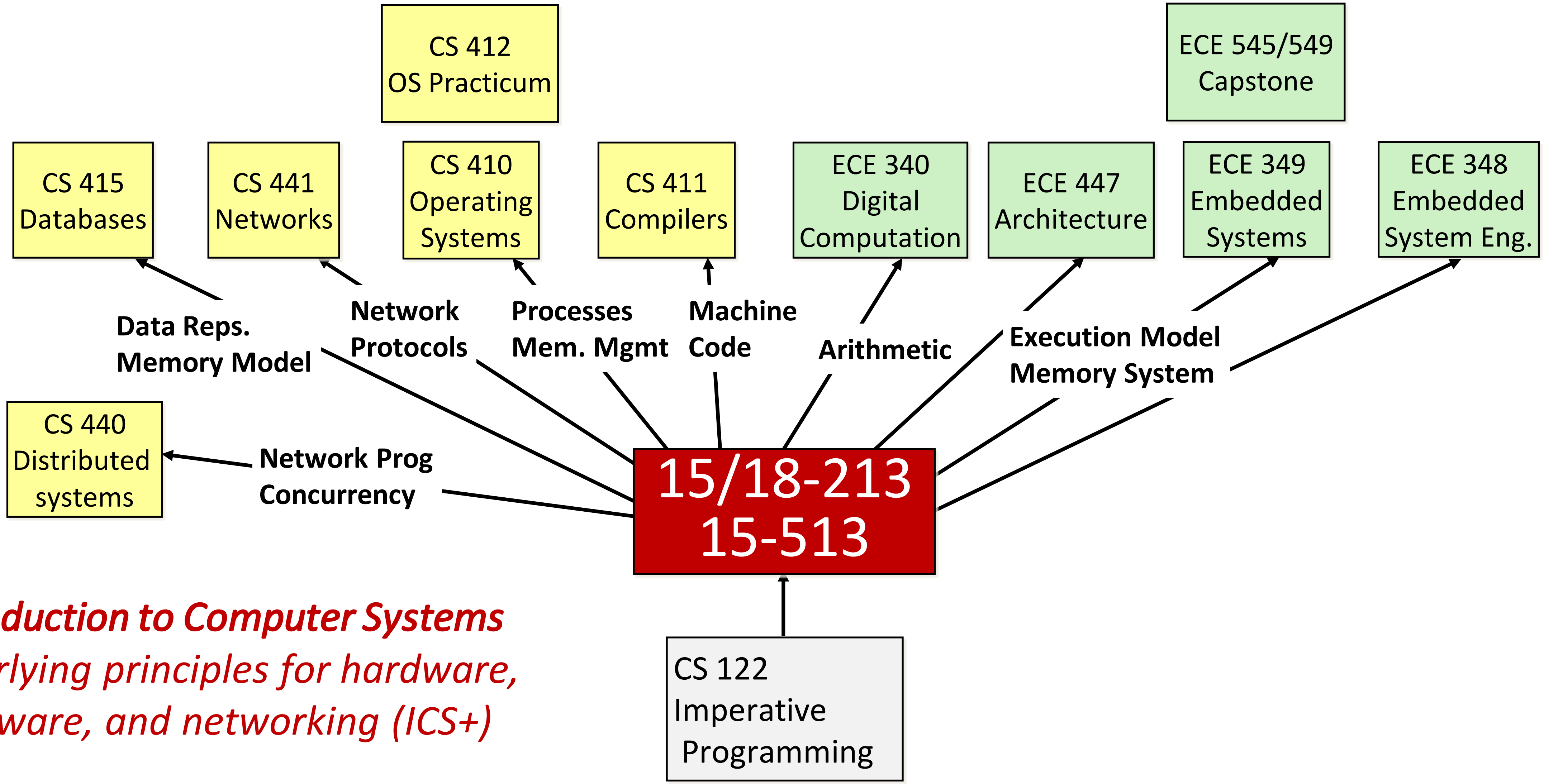
18-600 Foundations of Computer Systems

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

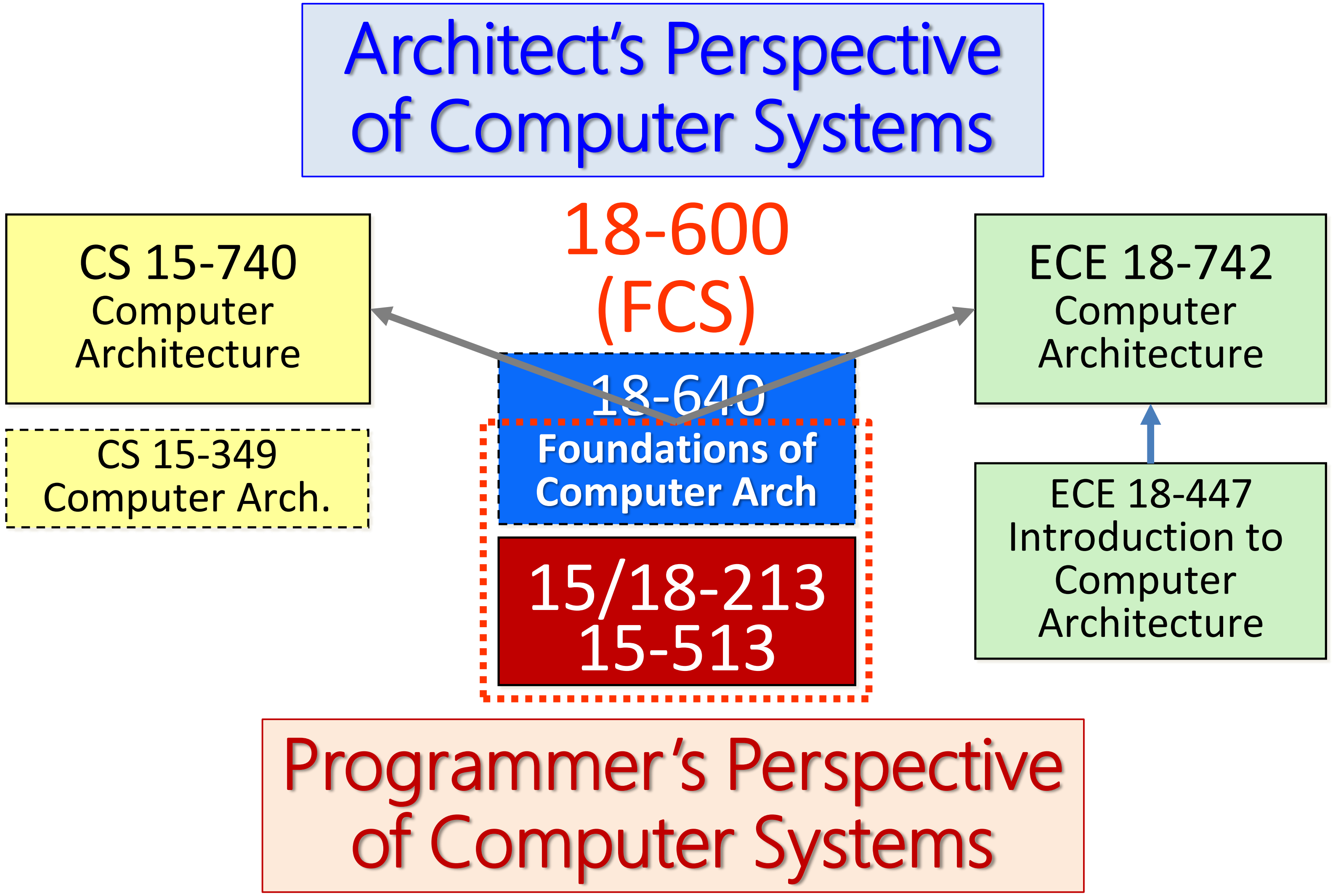
What Is 18-600 (FCS)? … starting with 15-513 …



Introduction to Computer Systems

Underlying principles for hardware, software, and networking (ICS+)

18-600 = 15-513 + (15-349) + 18-640/abridged



Architect's Perspective of Computer Systems

18-600 (FCS)

CS 15-740 Computer Architecture

CS 15-349 Computer Arch.

18-640 Foundations of Computer Arch

15/18-213 15-513

ECE 18-742 Computer Architecture

ECE 18-447 Introduction to Computer Architecture

Programmer's Perspective of Computer Systems

Course Assumptions and Expectations

- Who should take 18-600?
  - Graduate students (MS/PhD in ECE, MS in INI)
    - Applications and systems programming; Broad computing systems expertise
    - 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)

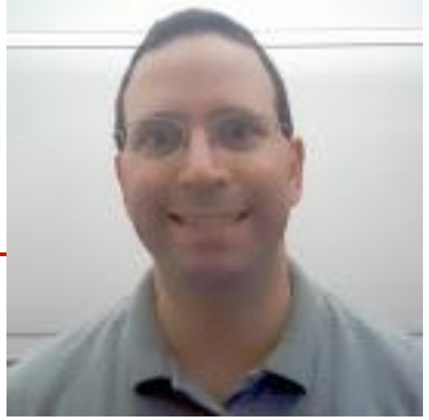
➢ Teaching Assistants (SV):

- 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
- Industry (2000-2015)
  - 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,
   - *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
2. Brian Kernighan and Dennis Ritchie,
   - *The C Programming Language*, Second Edition, Prentice Hall, 1988
   - 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)
   - *Modern Processor Design: Fundamentals of Superscalar Processors,* 2005; reissued by Waveland Press Inc , 2013. ISBN 10: 1478607831, ISBN 13: 9781478607830
2. [Optional] Michel Dubois, Murali Annavaram, and Per Stenstrom
   - Parallel Computer Organization and Design, by, Cambridge University Press, 2012. ISBN 978-0-521-88675-8.

Class Schedule – Fall 2017

- **Lecture**:	- **Labs/Recitation**:
Lectures, Section A: *MW, 6:30pm to 8:20pm (ET), DH A302*	*Recitation, Section A:* *T, 7:30pm to 8:50pm (ET), HH 1107*
Lectures, Section B: *MW, 6:30pm to 8:20pm (ET), DH A302*	*Recitation, Section B:* *T, 7:30pm to 8:50pm (ET), GHC 4102*
Lectures, Section C: *MW, 6:30pm to 8:20pm (ET), DH A302*	*Recitation, Section C:* *T, 5:30pm to 6:50pm (ET), WEH 4623*
Lectures, Section D: *MW, 6:30pm to 8:20pm (ET), DH A302*	*Recitation, Section D:* *T, 5:30pm to 6:50pm (ET), WEH 5320*
Lectures, Section SA: *MW, 3:30pm to 5:20pm (PT), B23 118*	*Recitation, Section SA:* *T, 4:30pm to 5:50pm (PT), B23 118*
Lectures, Section SB: *MW, 3:30pm to 5:20pm (PT), B23 211*	*Recitation, Section SB:* *T, 4:30pm to 5:50pm (PT), B23 109/110*

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

RECITATIONS (Led by TA's)	LAB Assignments	50%	(7) Individual lab assignments with varying weights. Will allow teams of two for Lab Assignments 5-7.
LECTURES (Instructors)	Mid-Term EXAM	20%	In class Exam (110 minutes) covering Lectures 1-15, and Lab Assignments 1-4.
	Final EXAM	30%	In class Exam (180 minutes) 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 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

18-600 Foundations of Computer Systems

Lecture 1: "Course Introduction & Overview"

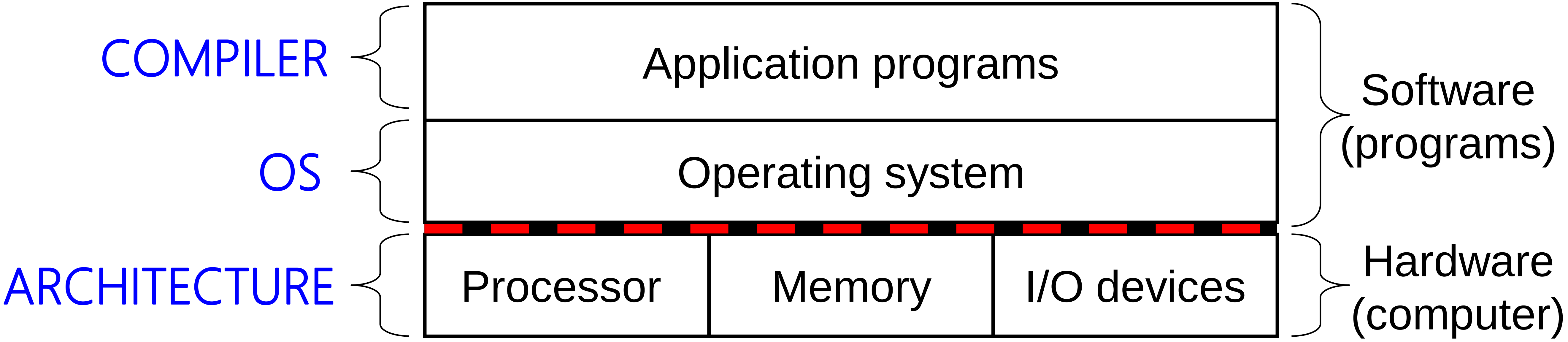
1. **Course Introduction**
   a. Birth of the New 18-600
   b. Teaching & Support Staff
   c. Course Organization
   d. Course Policy
2. **Course Overview**
   a. Tour of Computer Systems
   b. Lab Assignments Overview

Electrical & Computer ENGINEERING

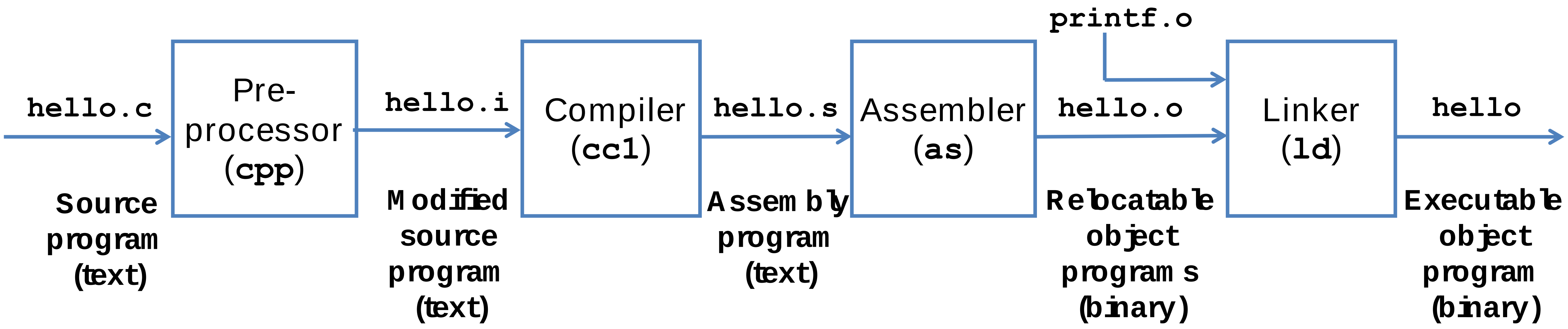
Anatomy of a Computer System: SW/HW

➢ 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**



printf.o

hello.c → Pre-processor (cpp) → hello.i → Compiler (cc1) → hello.s → Assembler (as) → hello.o → Linker (ld) → hello

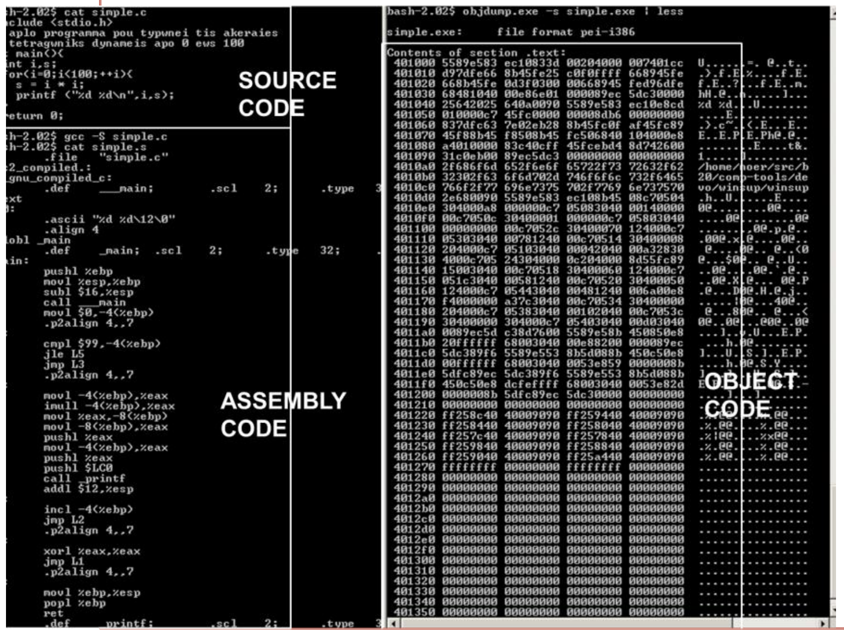
Source program (text)

Modified source program (text)

Assembly program (text)

Relocatable object programs (binary)

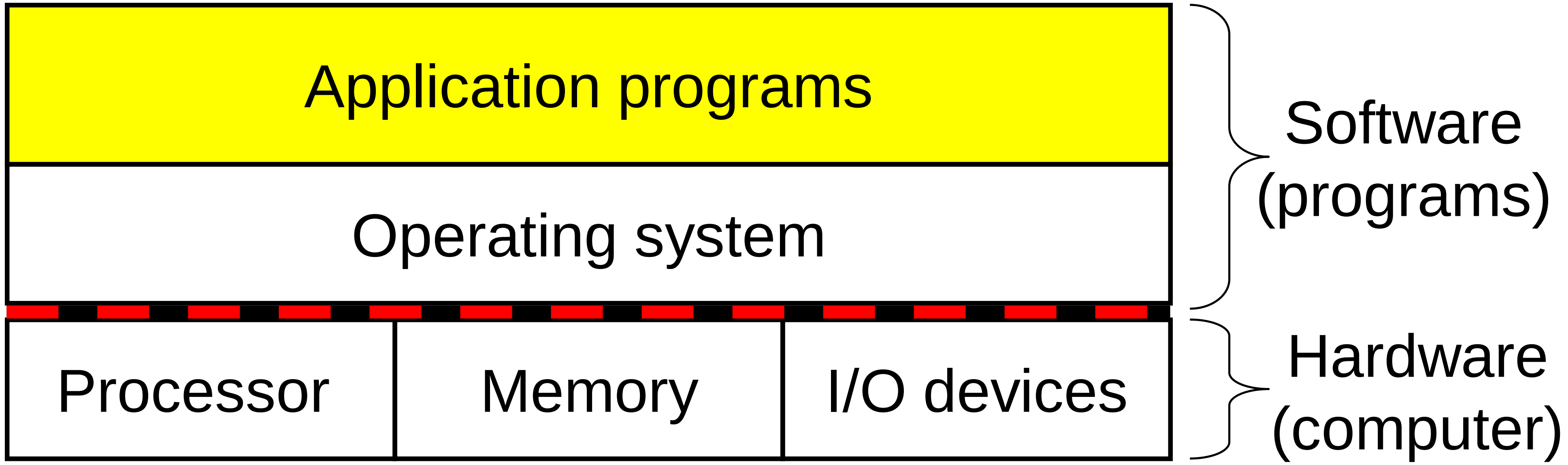
Executable object program (binary)



```
h-2.02$ cat simple.c
clude <stdio.h>
aplo programma pou typwnei tis akeraies
tetragwniks dynameis apo 0 ews 100
 main(){
int i,s;
for(i=0;i<100;++i){
  s = i * i;
  printf ("%d %d\n",i,s);

eturn 0;
```

```
h-2.02$ gcc -S simple.c
h-2.02$ cat simple.s
        .file   "simple.c"
2_compiled.:
gnu_compiled_c:
        .def    ___main;        .scl    2;      .type   3
xt
:
        .ascii "%d %d\12\0"
        .align 4
obl _main
        .def    _main;  .scl    2;      .type   32;     .
in:
        pushl %ebp
        movl %esp,%ebp
        subl $16,%esp
        call ___main
        movl $0,-4(%ebp)
        .p2align 4,,7

        cmpl $99,-4(%ebp)
        jle L5
        jmp L3
        .p2align 4,,7

        movl -4(%ebp),%eax
        imull -4(%ebp),%eax
        movl %eax,-8(%ebp)
        movl -8(%ebp),%eax
        pushl %eax
        movl -4(%ebp),%eax
        pushl %eax
        pushl $LC0
        call _printf
        addl $12,%esp

        incl -4(%ebp)
        jmp L2
        .p2align 4,,7

        xorl %eax,%eax
        jmp L1
        .p2align 4,,7

        movl %ebp,%esp
        popl %ebp
        ret
        .def    _printf;        .scl    2;      .type   3
```

```
bash-2.02$ objdump.exe -s simple.exe | less

simple.exe:     file format pei-i386

Contents of section .text:
 401000 5589e583 ec10833d 00204000 007401cc  U......=. @..t..
 401010 d97dfe66 8b45fe25 c0f0ffff 668945fe  .}.f.E.%....f.E.
 401020 668b45fe 0d3f0300 00668945 fed96dfe  f.E..?...f.E..m.
 401030 68481040 00e86e01 000089ec 5dc30000  hH.@..n.....]...
 401040 25642025 640a0090 5589e583 ec10e8cd  %d %d...U.......
 401050 010000c7 45fc0000 00008db6 00000000  ....E...........
 401060 837dfc63 7e02eb28 8b45fc0f af45fc89  .}.c~..(.E...E..
 401070 45f88b45 f8508b45 fc506840 104000e8  E..E.P.E.Ph@.@..
 401080 a4010000 83c40cff 45fcebd4 8d742600  ........E.....t&.
 401090 31c0eb00 89ec5dc3 00000000 00000000  1.....].........
 4010a0 2f686f6d 652f6e6f 65722f73 72632f62  /home/noer/src/b
 4010b0 32302f63 6f6d702d 746f6f6c 732f6465  20/comp-tools/de
 4010c0 766f2f77 696e7375 702f7769 6e737570  vo/winsup/winsup
 4010d0 2e680090 5589e583 ec108b45 08c70504  .h..U......E....
 4010e0 304000a8 000000c7 05083040 00140000  0@........0@....
 4010f0 00c7050c 30400001 000000c7 05803040  ....0@........0@
 401100 00000000 00c7052c 30400070 124000c7  .......,0@.p.@..
 401110 05303040 00781240 00c70514 30400008  .00@.x.@....0@..
 401120 204000c7 05103040 00042040 00a32830   @....0@.. @..(0
 401130 4000c705 24304000 0c204000 8d55fc89  @...$0@.. @..U..
 401140 15003040 00c70518 30400060 124000c7  ..0@....0@.`.@..
 401150 051c3040 00581240 00c70520 30400050  ..0@.X.@... 0@.P
 401160 124000c7 05443040 00481240 006a00e8  .@...D0@.H.@.j..
 401170 f4000000 a37c3040 00c70534 30400000  .....|0@...40@..
 401180 204000c7 05383040 00102040 00c7053c   @...80@.. @...<
 401190 30400000 304000c7 05403040 00d03040  0@..0@...@0@..0@
 4011a0 0089ec5d c38d7600 5589e58b 450850e8  ...]..v.U...E.P.
 4011b0 20ffffff 68003040 00e88200 000089ec   ...h.0@........
 4011c0 5dc389f6 5589e553 8b5d088b 450c50e8  ]...U..S.]..E.P.
 4011d0 00ffffff 68003040 0053e859 0000008b  ....h.0@.S.Y....
 4011e0 5dfc89ec 5dc389f6 5589e553 8b5d088b  ]...]...U..S.]..
 4011f0 450c50e8 dcfeffff 68003040 0053e82d  E.P.....h.0@.S.-
 401200 0000008b 5dfc89ec 5dc30000 00000000  ....]...].......
 401210 00000000 00000000 00000000 00000000  ................
 401220 ff258c40 40009090 ff259440 40009090  .%.@@....%.@@...
 401230 ff258440 40009090 ff258040 40009090  .%.@@....%.@@...
 401240 ff257c40 40009090 ff257840 40009090  .%|@@....%x@@...
 401250 ff259840 40009090 ff258840 40009090  .%.@@....%.@@...
 401260 ff259040 40009090 ff25a440 40009090  .%.@@....%.@@...
 401270 ffffffff 00000000 ffffffff 00000000  ................
 401280 00000000 00000000 00000000 00000000  ................
 401290 00000000 00000000 00000000 00000000  ................
 4012a0 00000000 00000000 00000000 00000000  ................
 4012b0 00000000 00000000 00000000 00000000  ................
 4012c0 00000000 00000000 00000000 00000000  ................
 4012d0 00000000 00000000 00000000 00000000  ................
 4012e0 00000000 00000000 00000000 00000000  ................
 4012f0 00000000 00000000 00000000 00000000  ................
 401300 00000000 00000000 00000000 00000000  ................
 401310 00000000 00000000 00000000 00000000  ................
 401320 00000000 00000000 00000000 00000000  ................
 401330 00000000 00000000 00000000 00000000  ................
 401340 00000000 00000000 00000000 00000000  ................
 401350 00000000 00000000 00000000 00000000  ................
```



Application programs

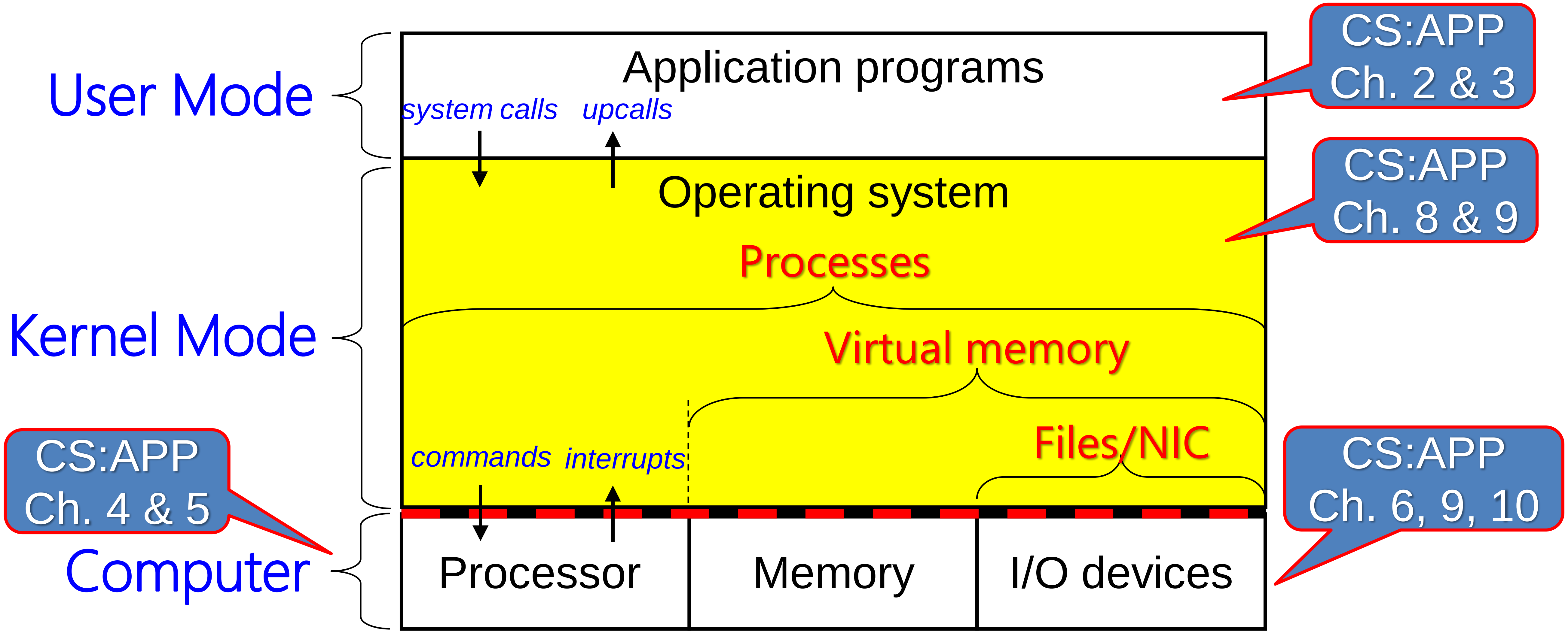
Operating system

Processor | Memory | I/O devices

Software (programs)

Hardware (computer)

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***

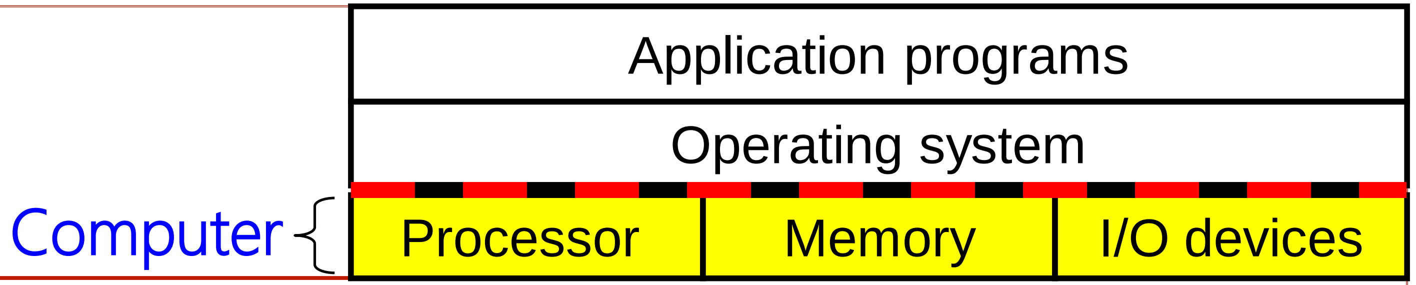
Abstraction 3: File System

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?



Application programs

Operating system

Computer

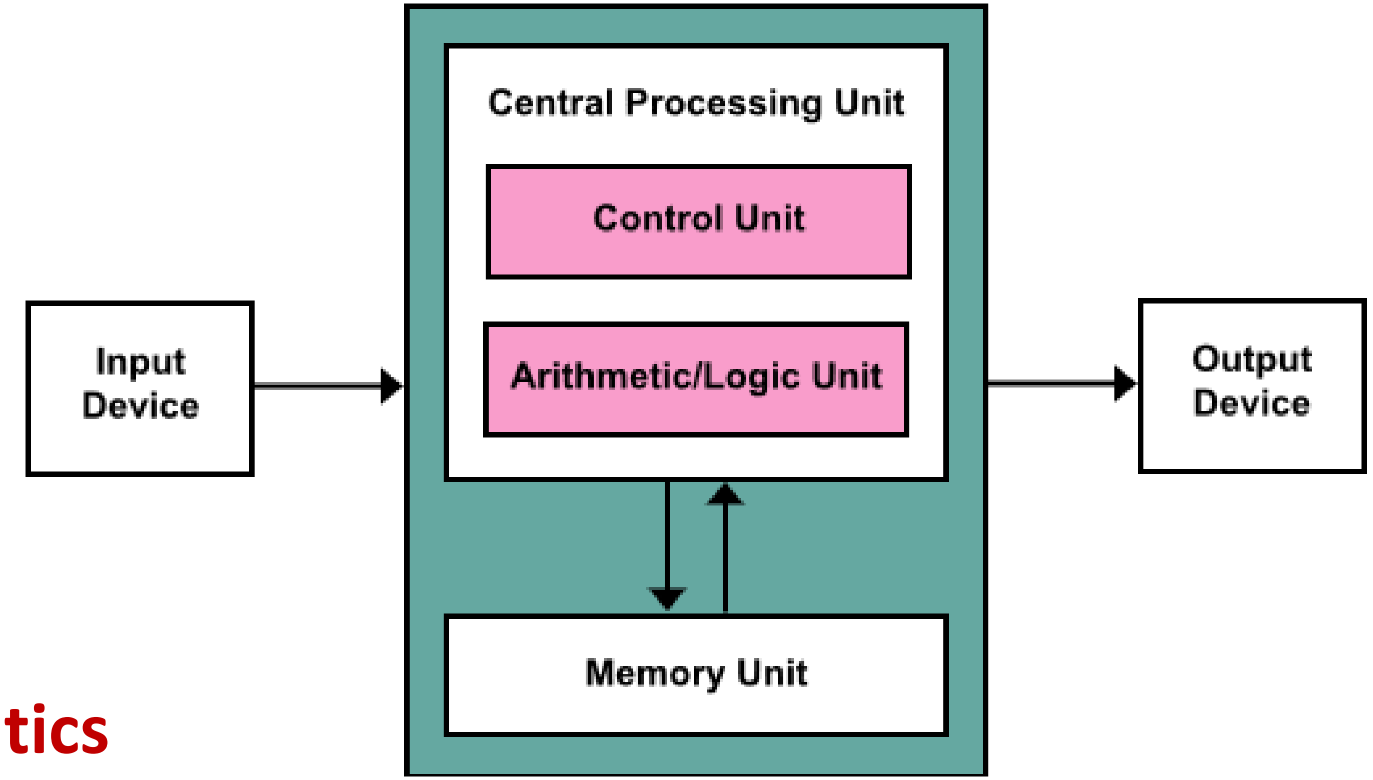
Processor

Memory

I/O devices

Bryant and O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition

- **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



Central Processing Unit

Control Unit

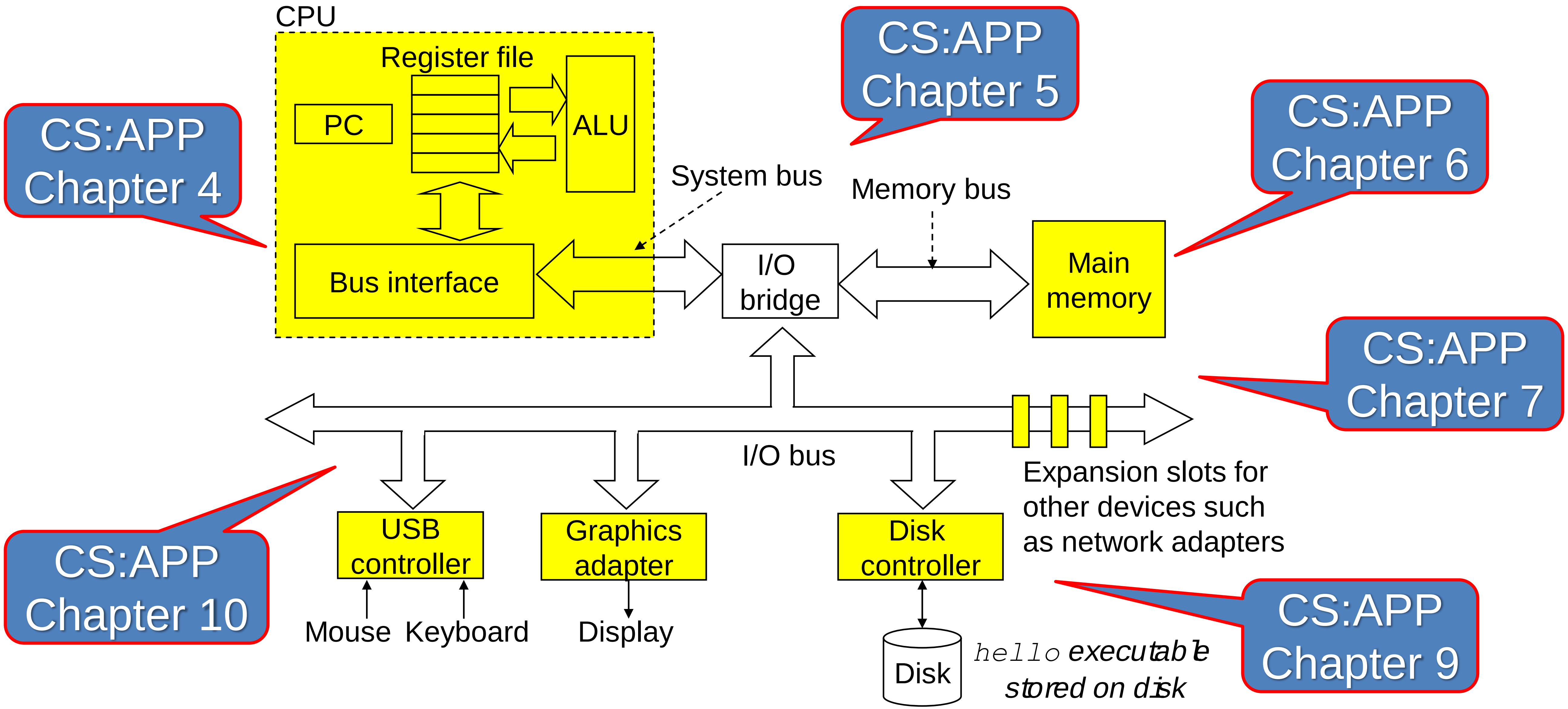
Arithmetic/Logic Unit

Memory Unit

Input Device

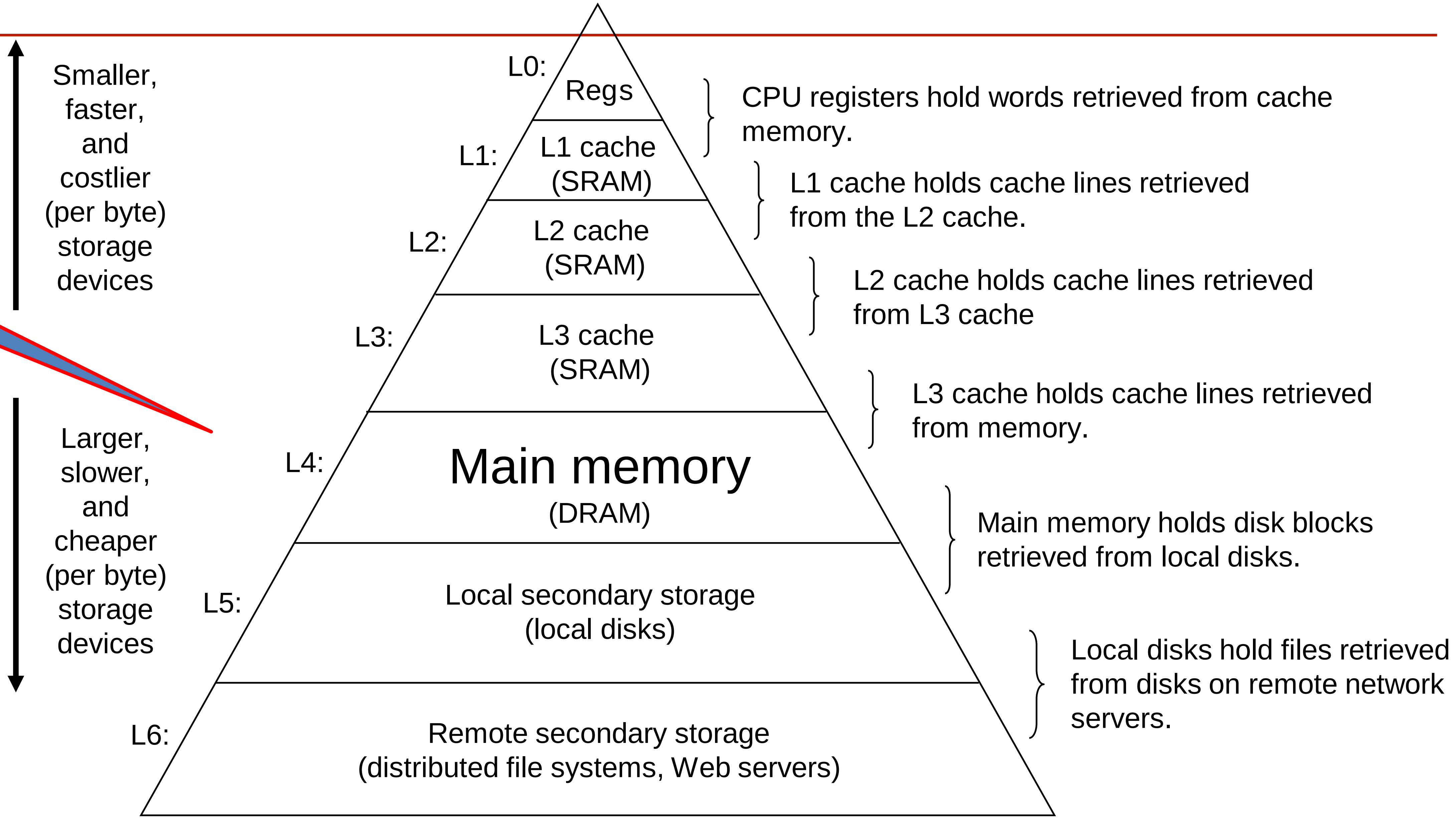
Output Device

Typical Computer (PC) Today: HW Organization

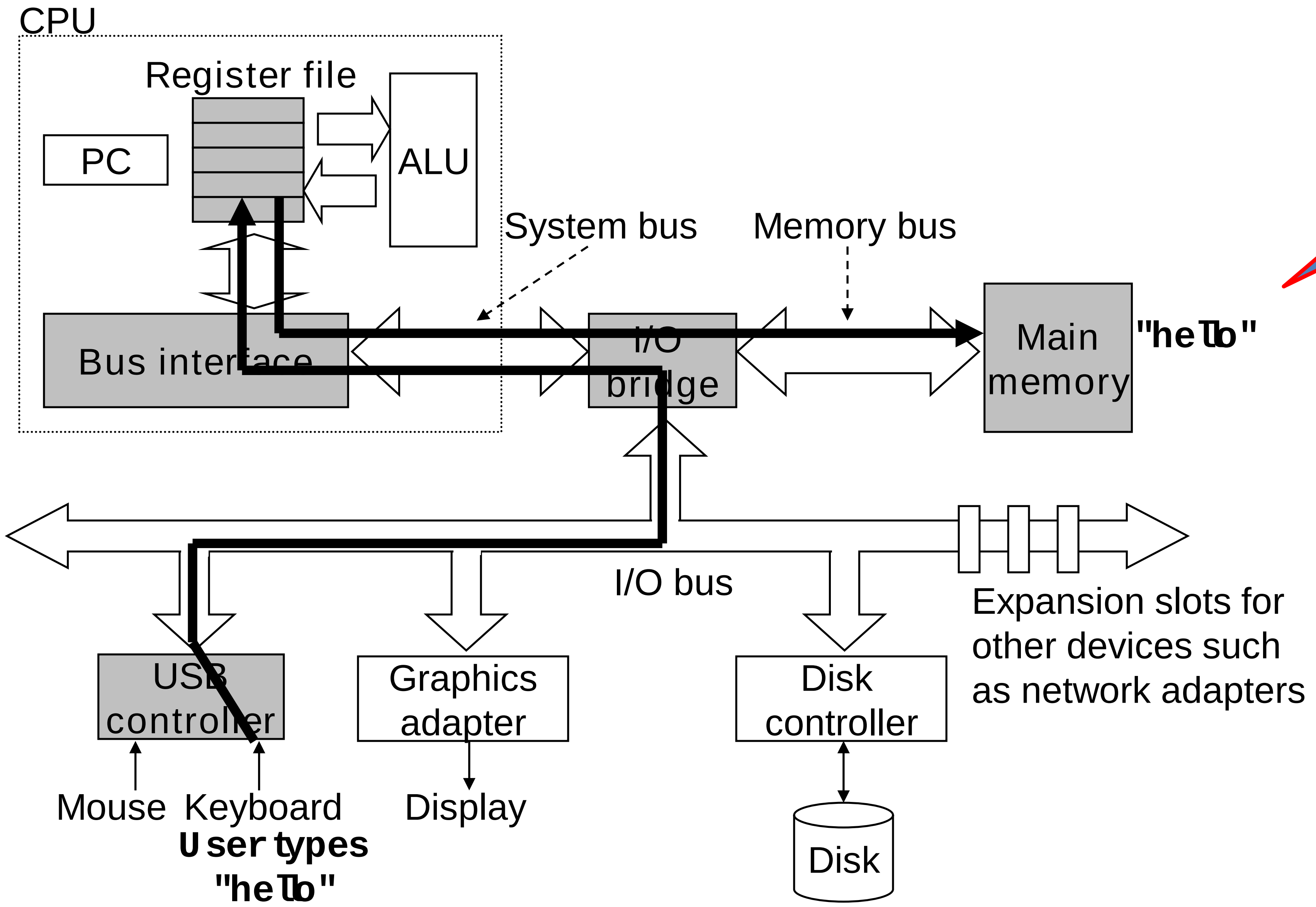


Example Memory Hierarchy

CS:APP Chapter 6

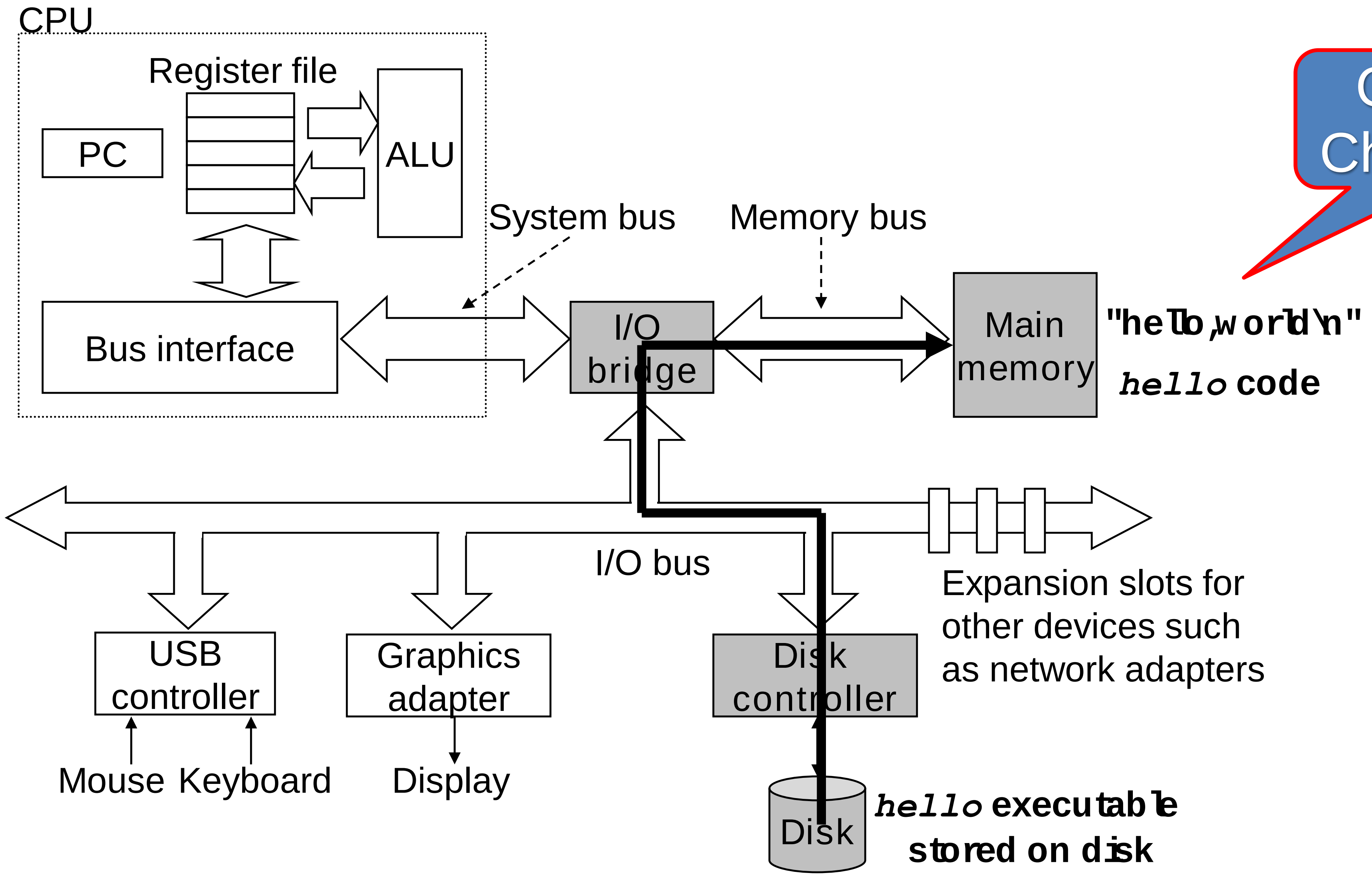

Smaller, faster, and costlier (per byte) storage devices
Larger, slower, and cheaper (per byte) storage devices
L0:
Regs
L1:
L1 cache (SRAM)
L2:
L2 cache (SRAM)
L3:
L3 cache (SRAM)
L4:
Main memory (DRAM)
L5:
Local secondary storage (local disks)
L6:
Remote secondary storage (distributed file systems, Web servers)
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.

Reading "hello" command from the keyboard



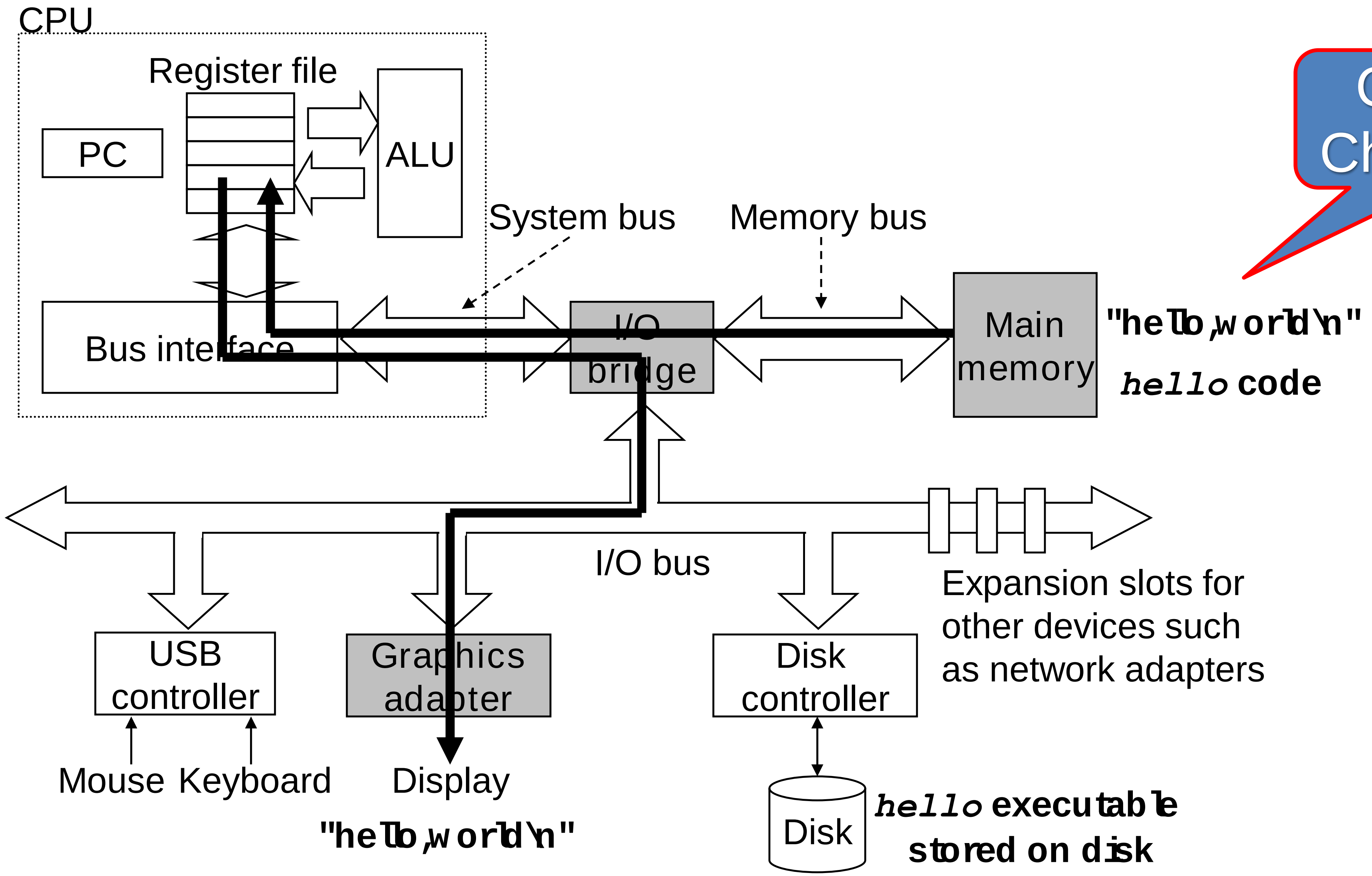
CS:APP Chapter 10

Loading executable from disk to main memory



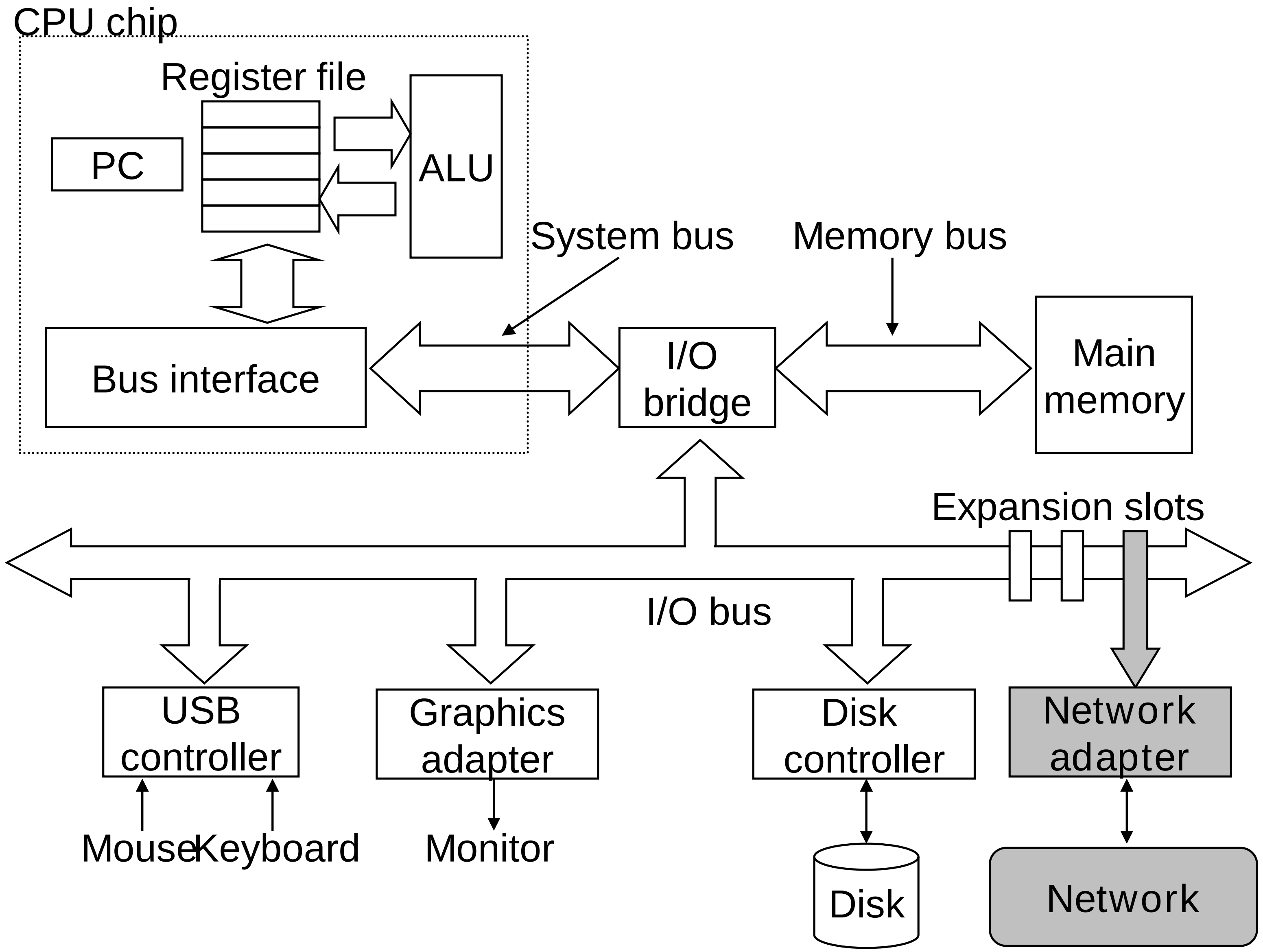
CS:APP Chapter 10

Writing output string from memory to display



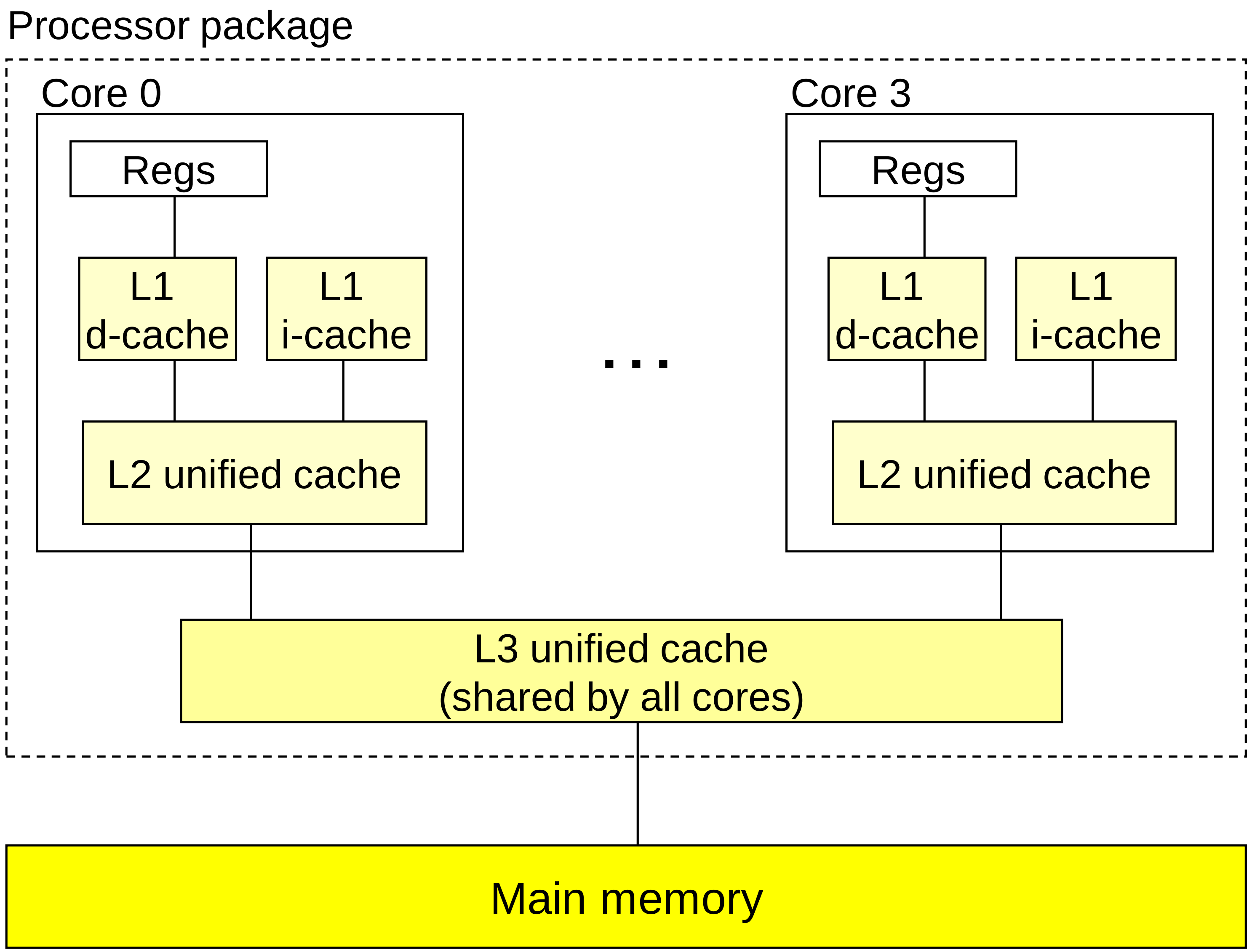
CS:APP Chapter 10

Network interface is another I/O device



CS:APP Chapter 11

Multicore Processor Organization (TLP)



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

CS:APP Ch. 2

CS:APP Ch. 3

Processor Architecture

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

CS:APP
Ch. 4 & 5

- 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

CS:APP
Ch. 8 & 10

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.

CS:APP
Ch. 5 & 6

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

CS:APP
Ch. 11 & 12

- 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!!!

18-600 Foundations of Computer Systems

Lecture 2:
"Computer Systems Big Picture"

John P. Shen & Gregory Kesden
August 30, 2017

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