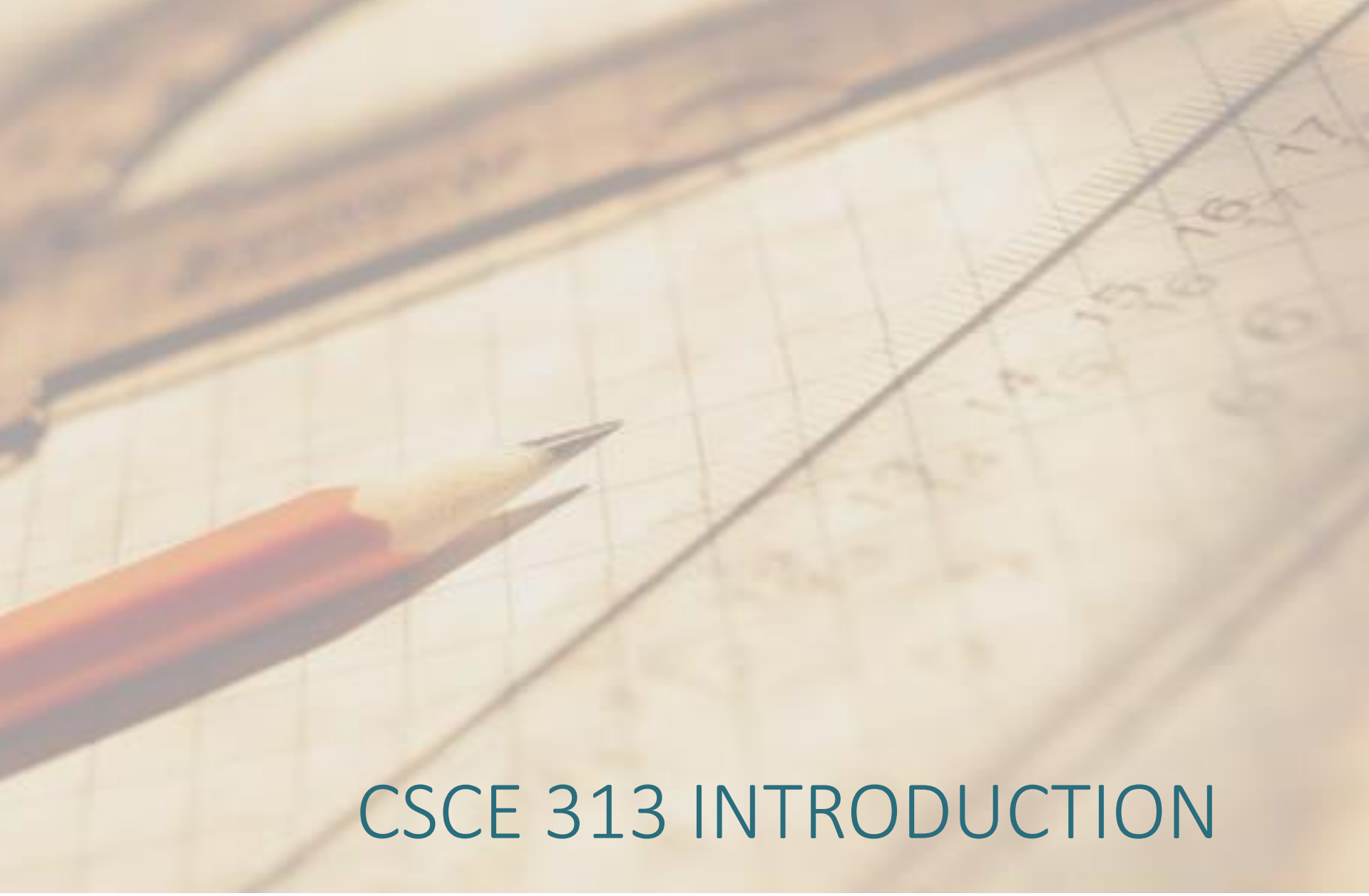


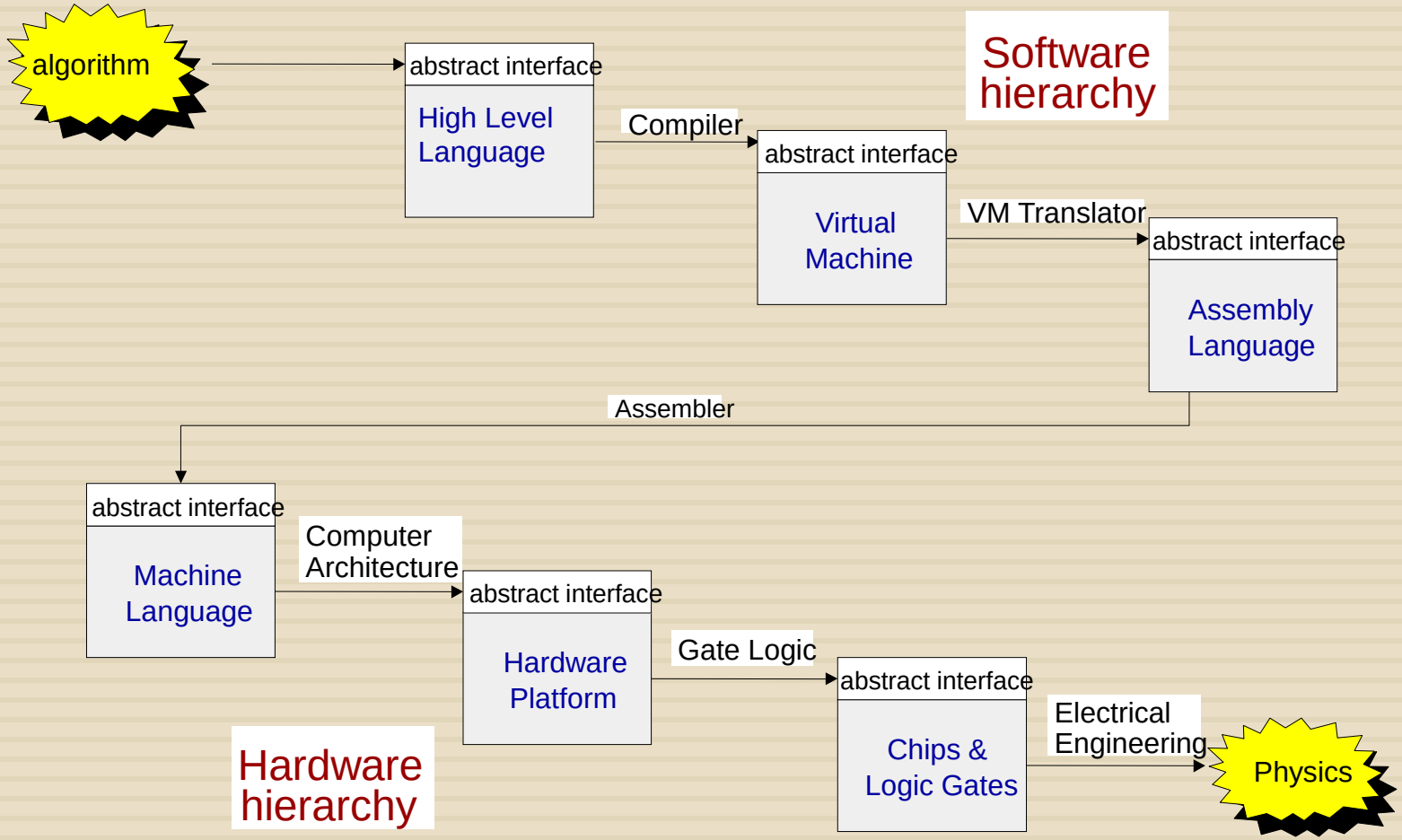
CSCE 313 INTRODUCTION

CSCE 313 Spring 2017

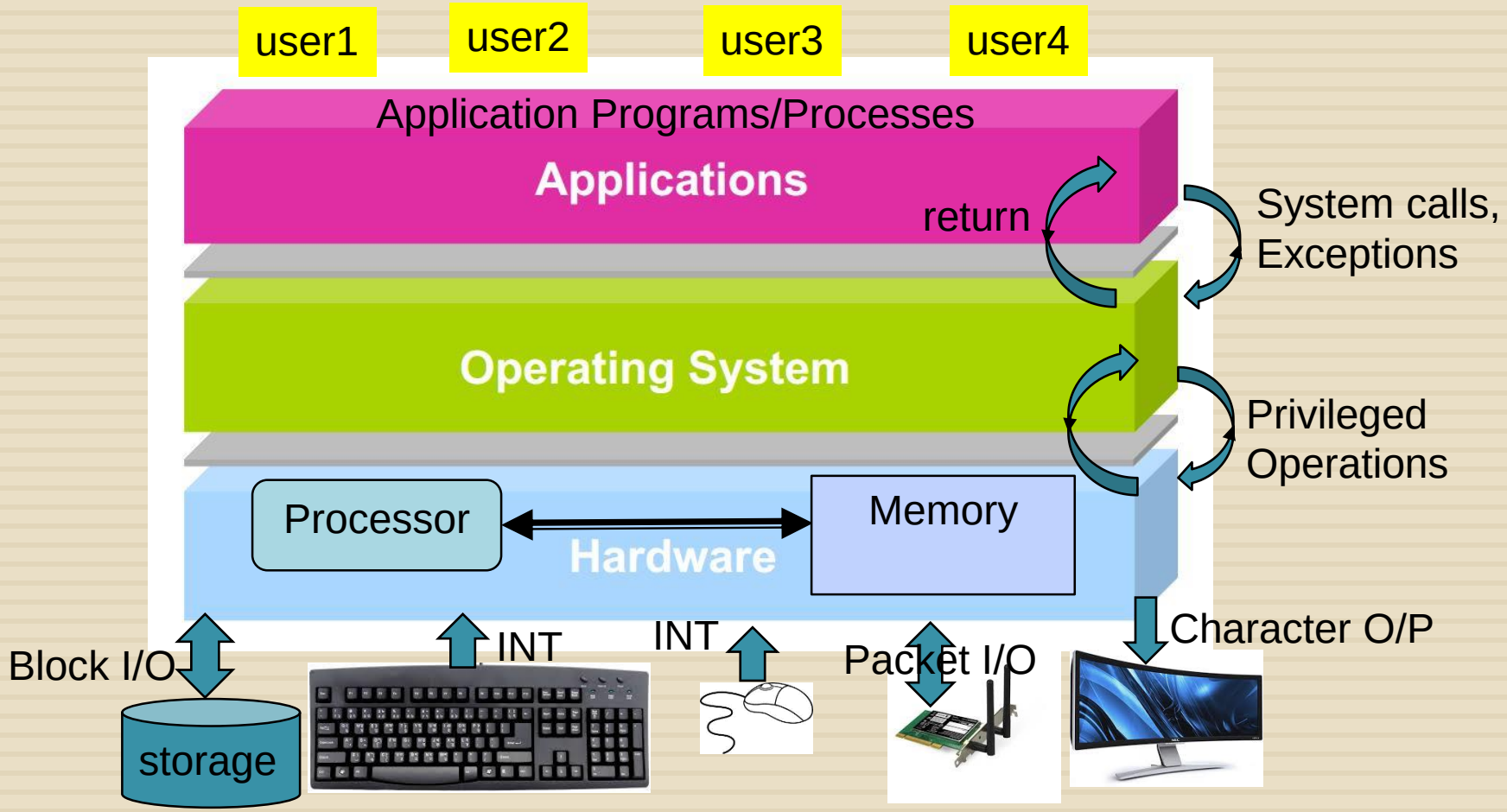
Main Points (for today)

- Part1
  - Introduction to Computer Systems
- Part2
  - Course Objectives and Outcome
  - Logistics

Learnings from CSCE-312: Hardware and Software Hierarchy



A Modern Computer System



First Generation Computer Systems (1949-1956)

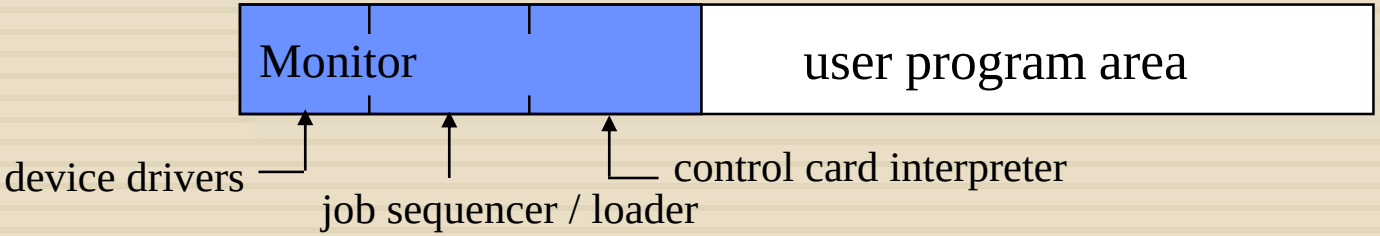
- Single user: writes program, operates computer through console or card reader / printer
- Absolute machine language

- I/O devices

- Development of libraries; device drivers
- Compilers, linkers, loaders
- Relocatable code

Second-Generation Computers (1956-1963)

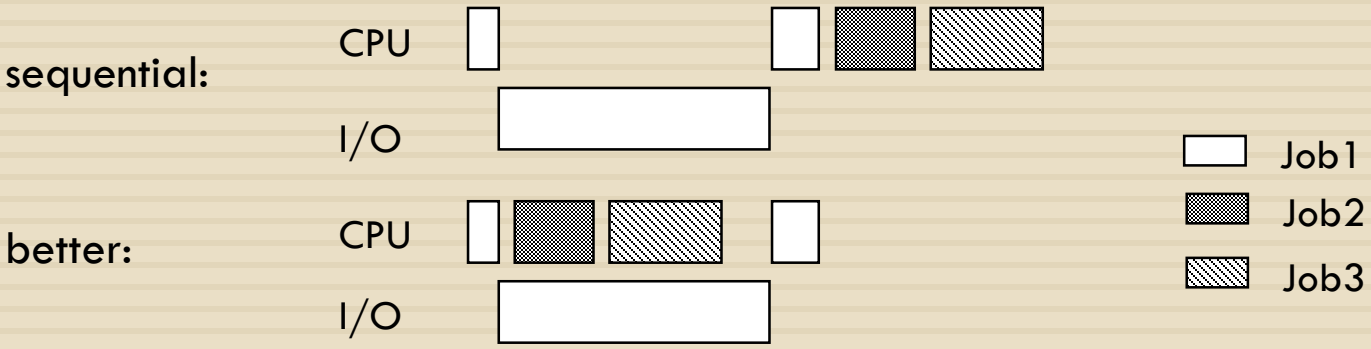
- Automation of Load/Translate/Load/Execute
  - Batch systems
  - Monitor programs



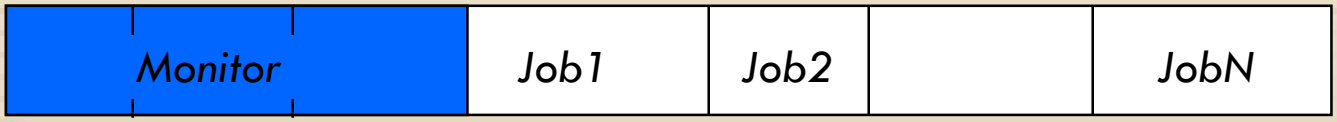
  - Job Control Language
  - Advent of operators: computers as input/output box
- Problem: Resource management and I/O still under control of programmer. Issues??
  - Memory protection
  - Timers
  - Privileged instructions

Third-Generation Computer Systems (1964-1975)

- Problem with batching: one-job-at-a-time



- Solution: Multiprogramming
  - Job pools: have several programs ready to execute
  - Keep several programs in memory



- New issues:
  - Job scheduling, Memory management, Protection

Time Sharing (mid 1960s on)

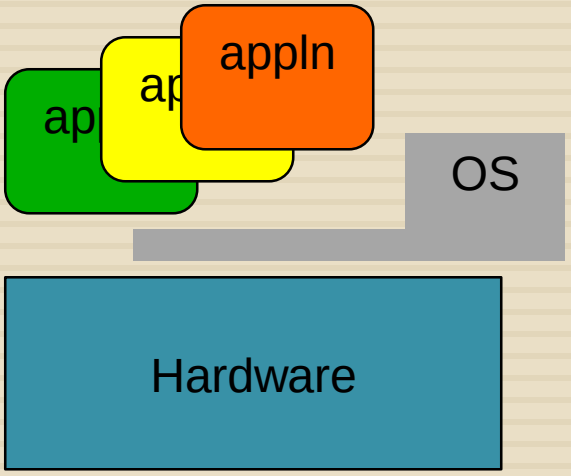
- Remote interactive access to computer: “Computing as Utility”
- OS interleaves execution of multiple user programs with time quantum, e.g. CTSS (1961): time quantum 0.2 sec
- User returns to own the machine
- New aspects and issues:
  - On-line file systems
  - resource protection
  - virtual memory
  - sophisticated process scheduling

Modern Computer Systems

- A modern computer system can be divided into the following three key components
  - **Application programs** – define the ways in which the system resources are used to solve the computing problems of the users
    - Word processors, compilers, web browsers, database systems, video games
  - **Hardware** – provides basic computing resources
    - CPU, memory, I/O devices
  - **Operating system -** controls and coordinates use of hardware among various applications and users

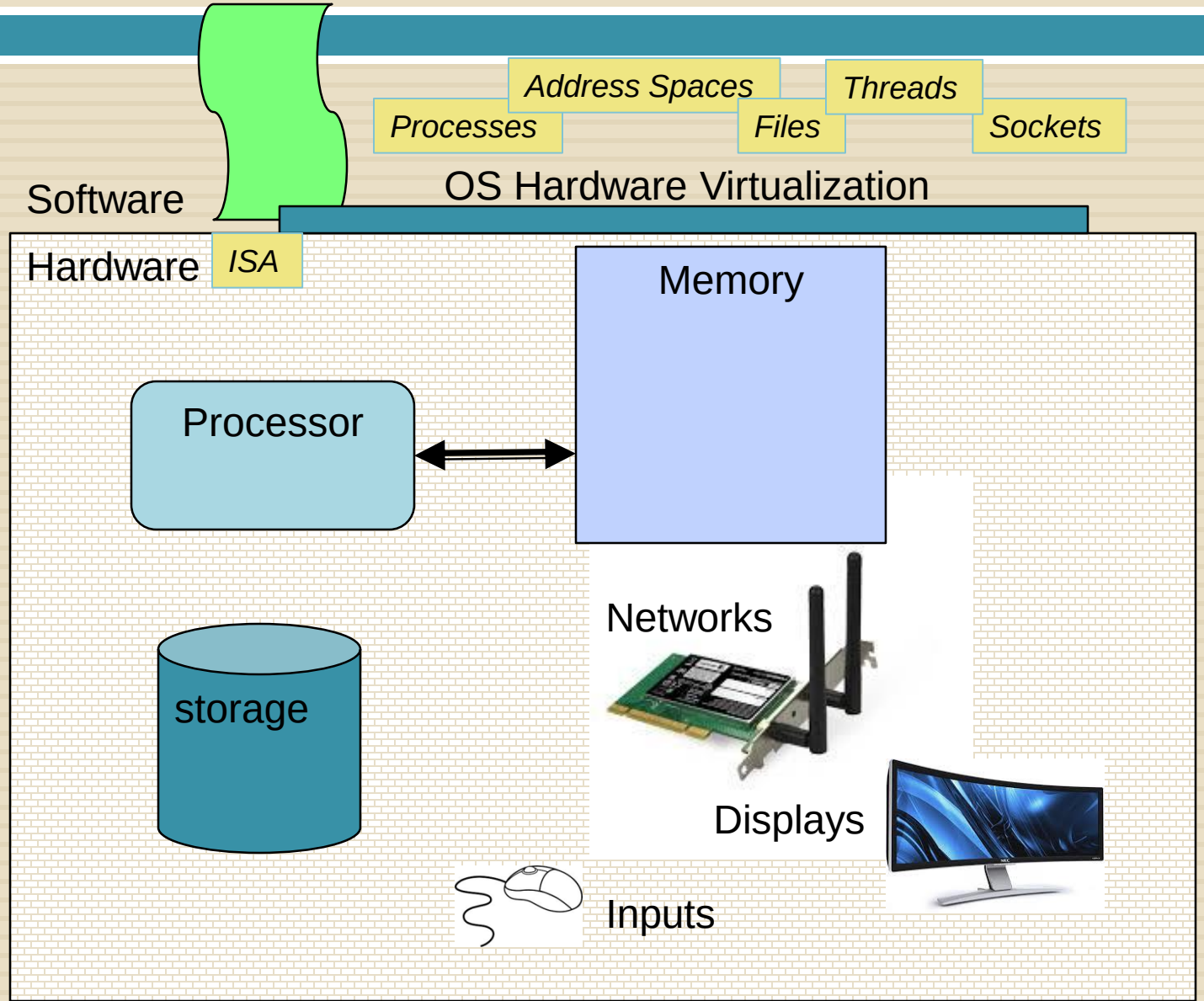
What is an operating system?

- Special layer of software that provides application software access to hardware resources
  - Convenient abstraction of complex hardware devices
  - Protected access to shared resources
  - Security and authentication
  - Communication amongst logical entities

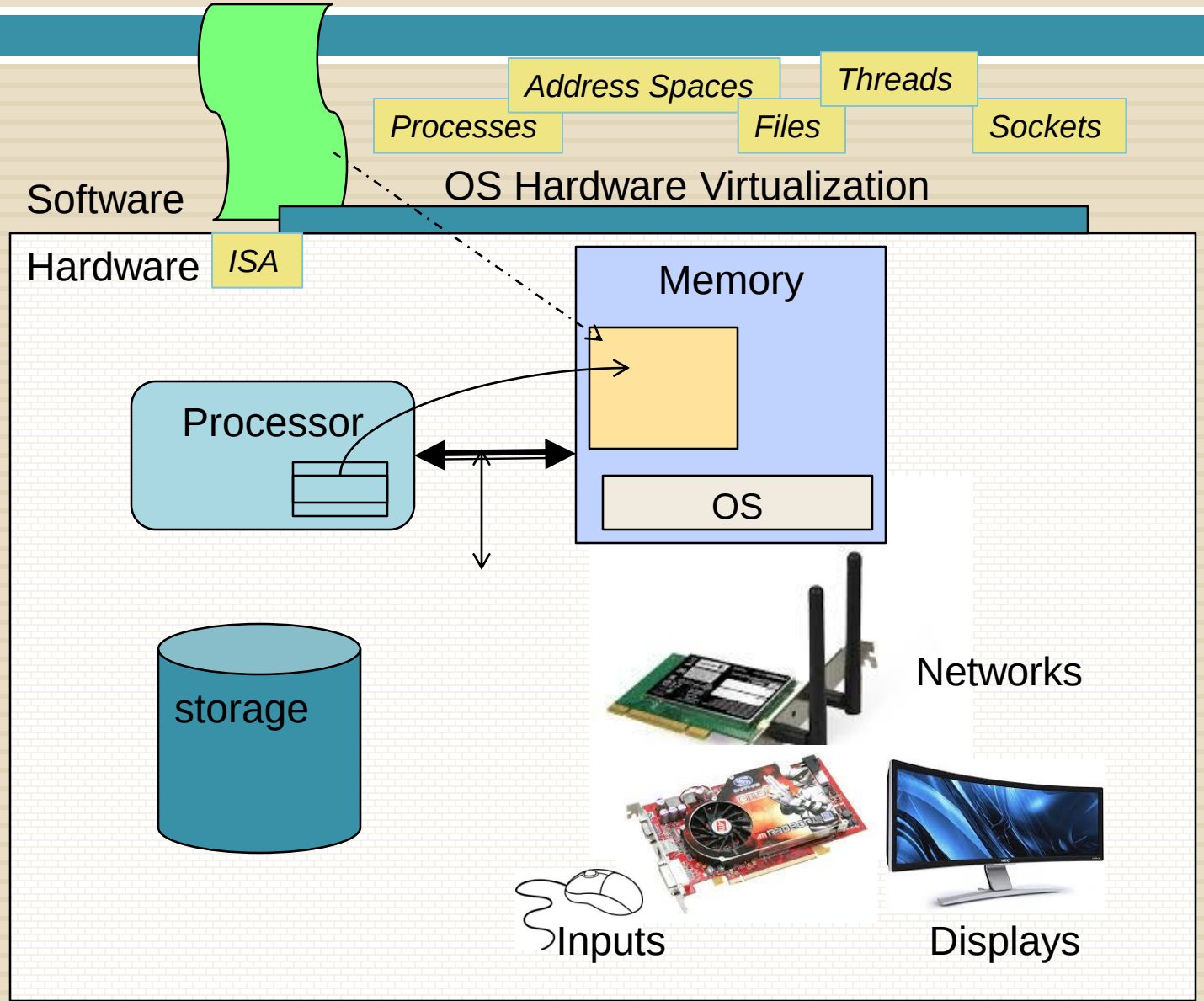


OS Basics: "Virtual Machine" Boundary

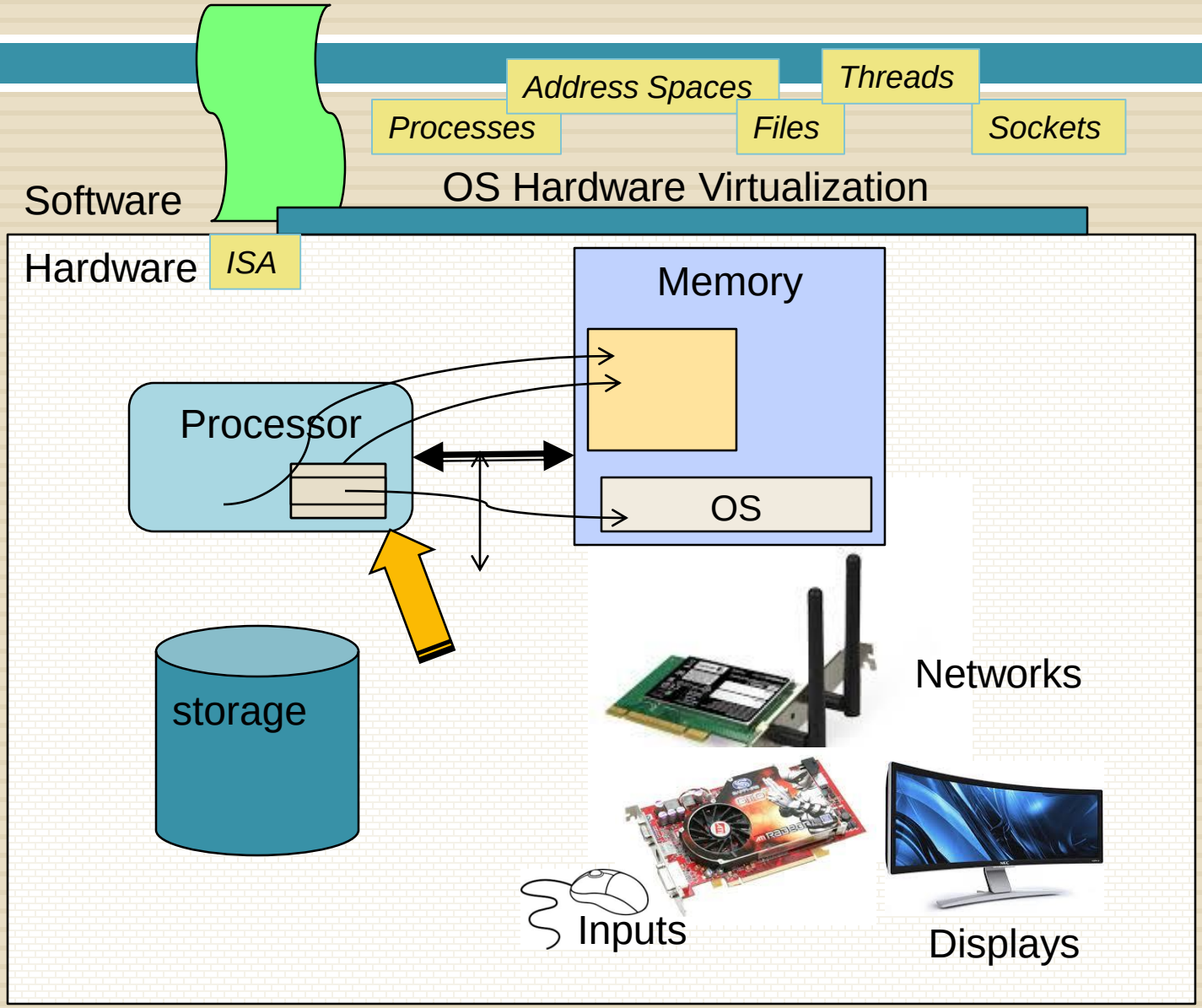
Courtesy: Prof Culler, Berkley



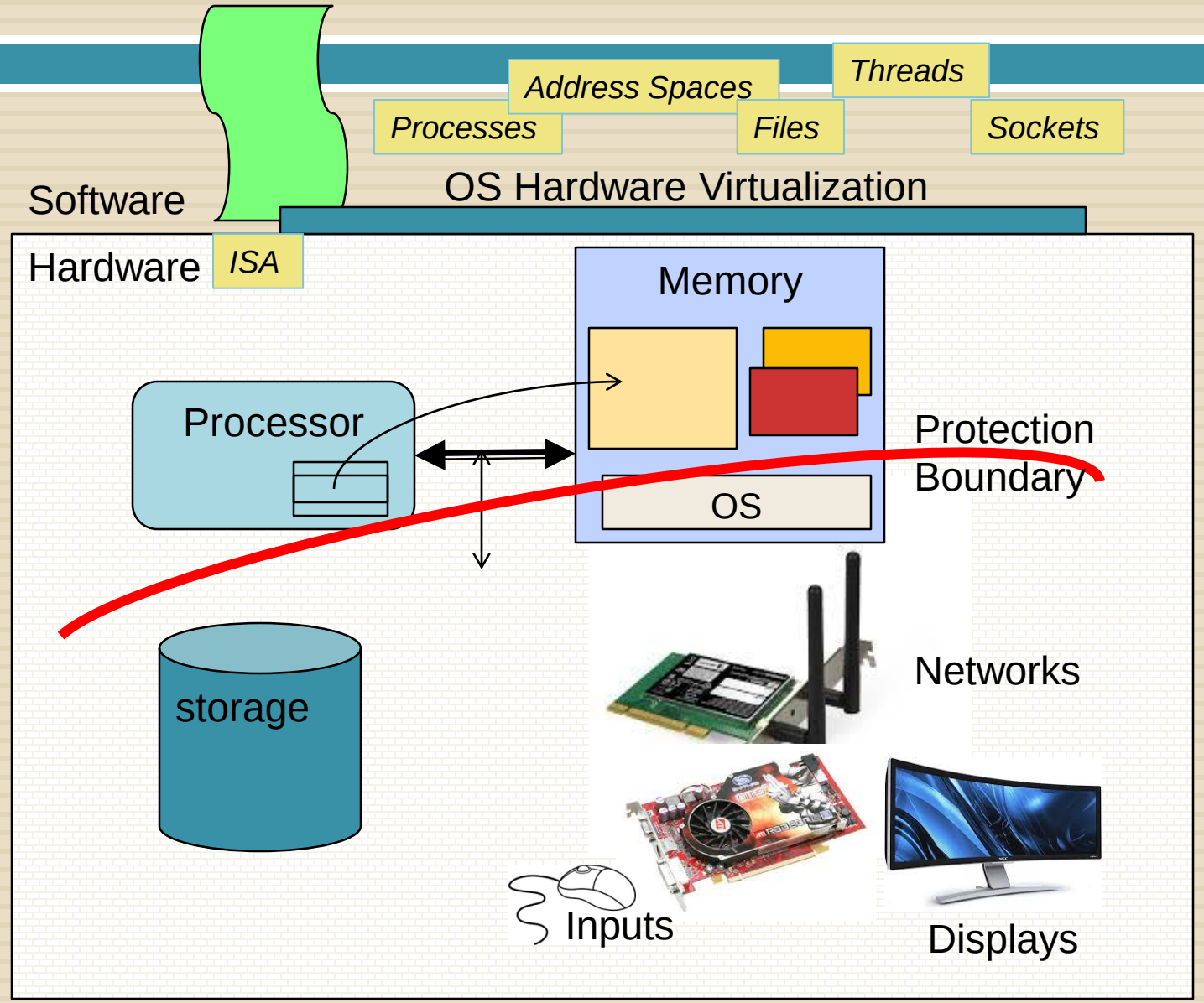
OS Basics: Program => Process



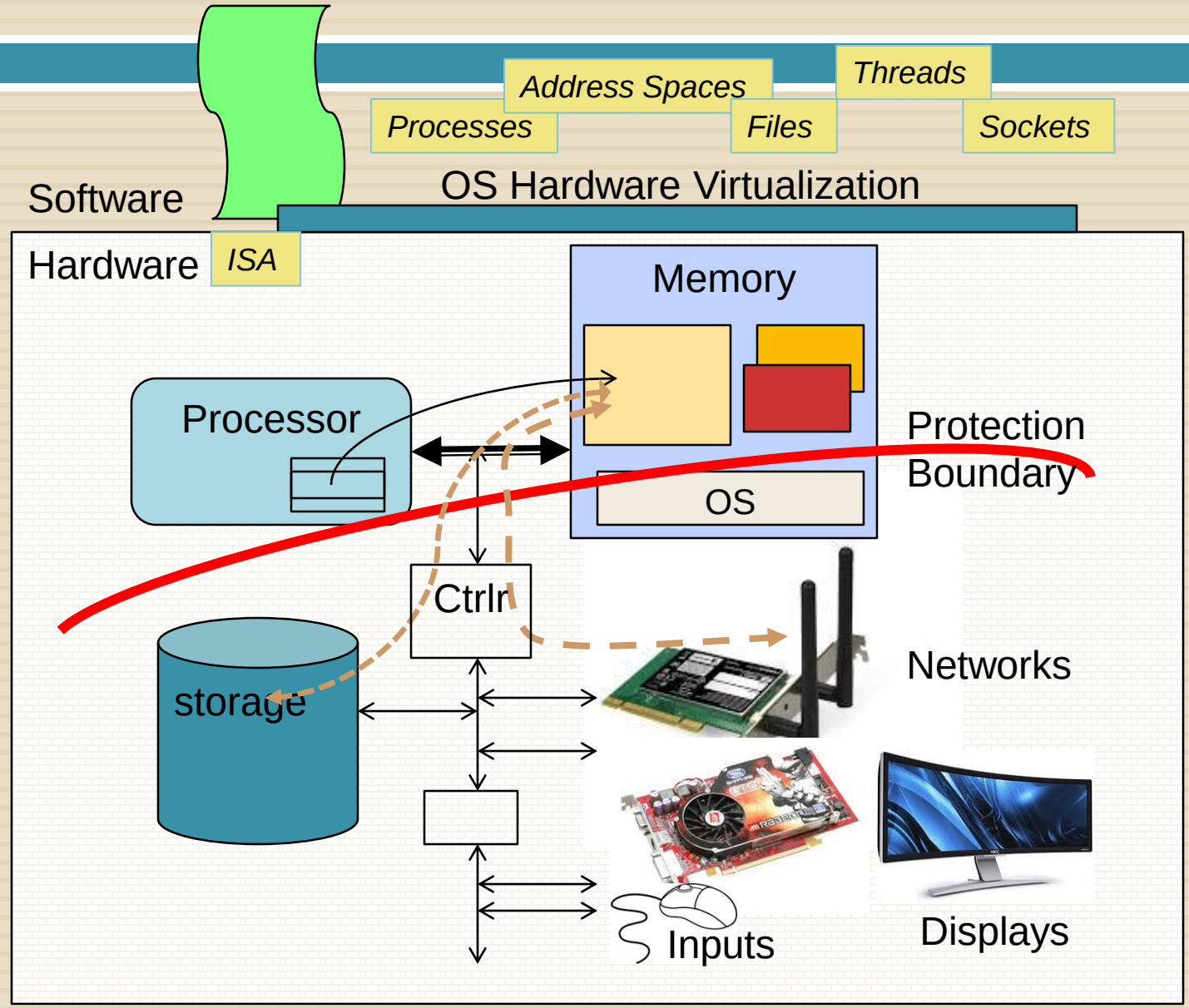
OS Basics: Context Switch



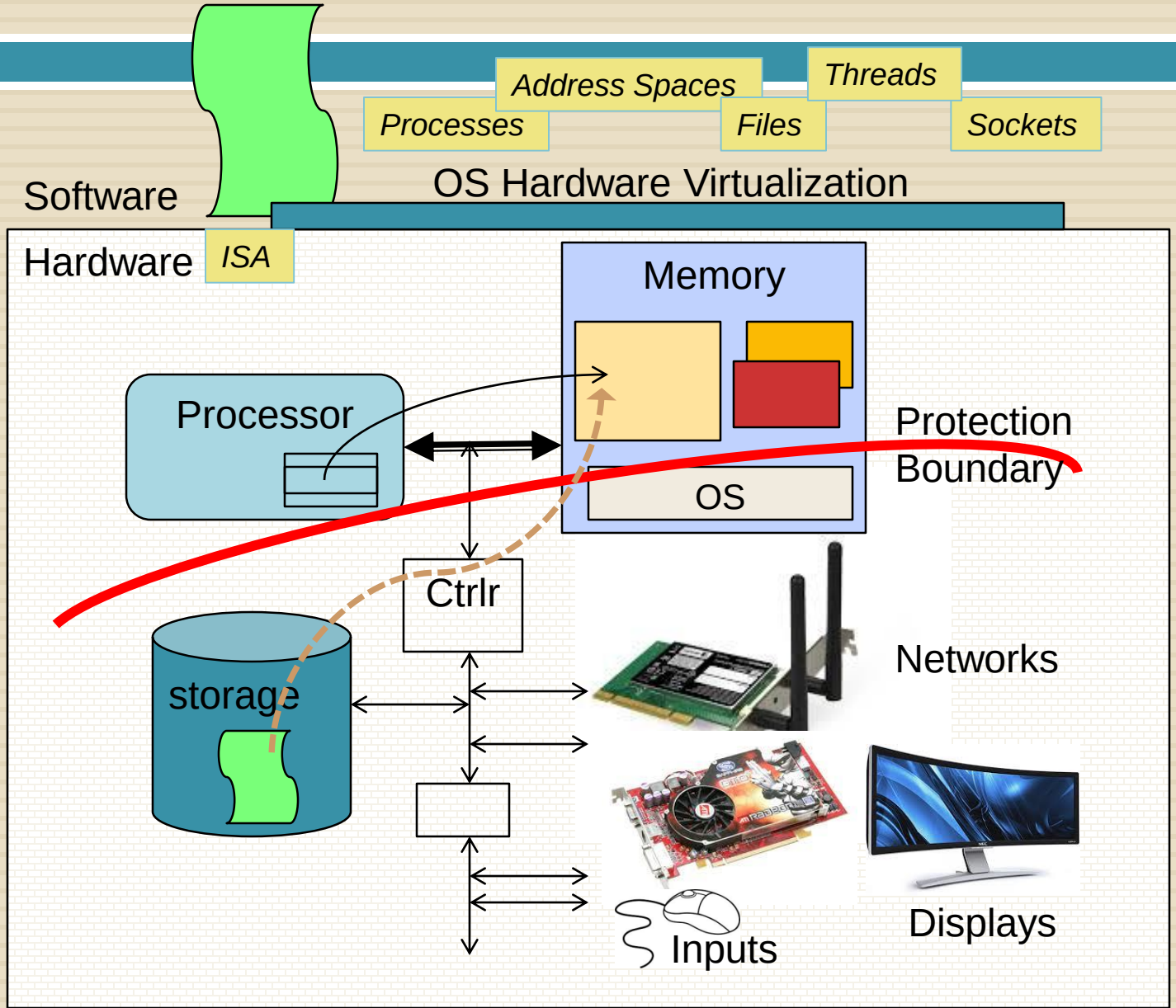
OS Basics: Scheduling, Protection



OS Basics: I/O


Address Spaces
Threads
Processes
Files
Sockets
Software
OS Hardware Virtualization
Hardware
ISA
Memory
Processor
OS
Protection Boundary
Ctrlr
storage
Networks
Inputs
Displays

OS Basics: Loading



Main Points (for today)

- Part1
  - Introduction to Computer Systems
- Part2
  - Course Objectives and Outcome
  - Logistics

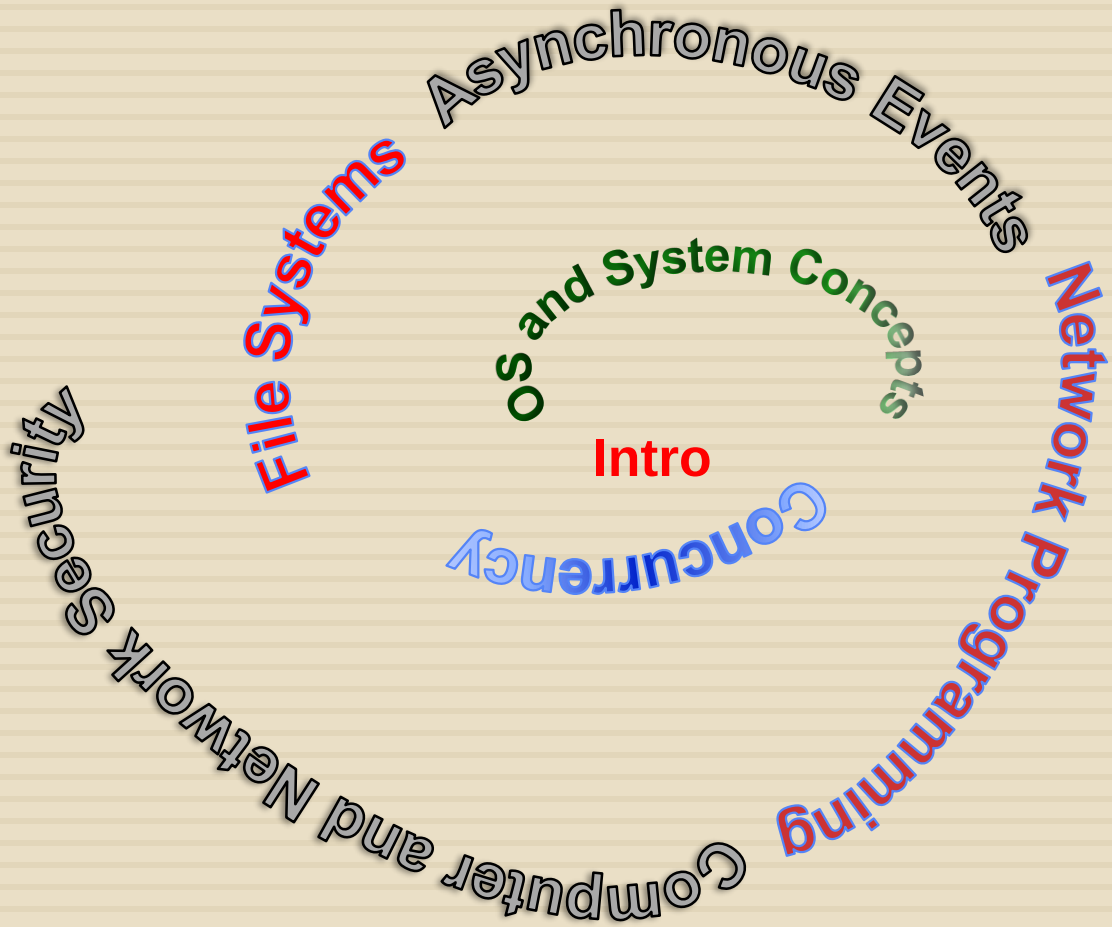
Why take CSCE 313?

- Some of you will actually design and build operating systems or components of them
  - Perhaps more now than ever
- Many of you will create systems that utilize the core concepts in operating systems
  - The concept of “hierarchy” and “abstraction” are all too important
  - Whether you build software or hardware
  - The concepts and design patterns appear at many levels
- All of you will build applications, etc. that utilize operating systems
  - The better you understand their design and implementation, the better use you’ll make of them.

CSCE-313 Course Outcome

- In this course, you will learn
  - What is an operating system; its components; system calls
  - Execution of a program; function calls; interrupts
  - OS application interface; file system; process control
  - Concurrency, process and thread synchronization, inter-process communication
  - Network Programming
  - Security threats in centralized and distributed systems; authentication, authorization, confidentiality; security mechanisms

CSCE-313 Course Structure Spiral



Textbook, Reference Books

- **Text**: Operating Systems: Principles and Practice, Second Edition, Thomas Anderson and Michael Dahlin, Recursive Books, 2014.
- **Reference**:
  - **Main**: Advanced Programming in the UNIX Environment, Third Edition, W. Richard Stevens and Stephen A. Rago, Addison-Wesley Professional Computing Series, 2013.
  - **Secondary**: Understanding Unix/Linux Programming A Guide to Theory and Practice, Bruce Molay, Pearson Education Inc., 2003
- **Other Interesting Readings**
  - **Computer Systems**
    - *Computer Systems: A Programmer's Perspective*, Randal E. Bryant and David R. O'Hallaron, Prentice Hall, 2011

How Success will be Measured

- The course will have two exams, a series of machine problems and quizzes. The grade allocation is as follows:
- Total = 200 points
  - Exams = 90 points (45 points each)
  - Announced Quizzes (several) = 10 points
  - Machine Problems (approx. 8) = 100 points

- The grading scale is as follows:
  - 180 – 200: A
  - 160 - 179: B
  - 140 – 159: C
  - 110 – 139: D
  - 109 and below: F

Exams

- Two exams – Midterm and Final
- Exams are closed book
- Final Exam will be based on material covered after the midterm
- Exams will be tough but success is guaranteed if you have understood the concepts

Quizzes

- There will be several announced in-class quizzes
- These will be aligned with major topics
- These are intended to encourage class participation and incremental concepts refresh

Week	Topic	Textbook	Reference
Week 1	Introduction to CSCE-313	N/A	N/A
	Introduction to Computer Systems	Ch 1	Ch 1, 2
Week 2	Operating Systems Introduction	Ch 2	Ch 7
	Operating Systems Structure - Exceptions	Ch 2	Ch 7
Week 3	Architectural Support for Operating Systems	Ch 2	Ch 7
	Introduction to UNIX Process	Ch 2	Ch 7
Week 4	Unix Fork and Exec and Programming Interface	Ch 3	Ch 7, 8
	MP2 Presentation	N/A	N/A
Week 5	Unix Programming Interface, Process Elements	Ch 3	Ch 7, 8
	Process Scheduling	Ch 7	Ch 12
Week 6	Process Scheduling	Ch 7	Ch 12
	Concurrency and Threads	Ch 4	Ch 11
Week 7	Concurrency and Threads	Ch 4	Ch 11
	Midterm Exam	N/A	N/A
Week 8	Process and Threads Synchronization	Ch 5, 6	Ch 12
	Process and Threads Synchronization	Ch 5, 6	Ch 12
	Spring Break	N/A	N/A
	Spring Break	N/A	N/A
Week 9	Process and Threads Synchronization	Ch 5, 6	Ch 12
	Unix I/O	N/A	Ch 3
Week 10	Inter Process Communication	N/A	Ch 15
	Inter Process Communication	N/A	Ch 10, 15
Week 11	Networking Basics	N/A	Ch 16
	Network Programming	N/A	Ch 17
Week 12	Network Programming	N/A	Ch 17
	Computer Security	N/A	TBD
Week 13	Computer Security	N/A	TBD
	Files and Directories	Ch 11, 13	Ch 4
Week 14	Files and Directories	Ch 11, 13	Ch 4
	BUFFER	N/A	TBD
Week 15	REDEFINED DAY	N/A	N/A
	Final Exam LINK is HERE: http://registrar.tamu.edu/general/finalschedule.aspx#0-Spring2017	N/A	N/A

Machine Problems

- There will be 8-9 **Machine Problems** assigned weekly or bi-weekly basis. **Students will form teams of two to work on machine problems**
- Typically, machine problems will be **described in lab** on week ‘x’ and will be **graded in lab** during week ‘x+1’ or ‘x+2’
- Lab attendance is necessary to ensure you understand the problem, actively hash the problem out with your teammate, and clarify with lab instructors.
- We are deploying a new platform for MP submission. More on this in the next class

Late Policy for Machine Problems

- Quizzes: 0 marks for late submissions.
- Machine Problems Penalty: (Relative to due date and time)
  - 1 day late: 5%
  - 2 days late: 15%
  - 3 days late: 20%
  - 4 days late: 50%
  - 4 days – 30 days late: The maximum possible grade will be capped to a 50% ceiling.
  - > 30 days late: 100% penalty i.e. 0 marks in the assignment