CSE150
Operating Systems
Lecture 1

Course Introduction
What is an Operating System?

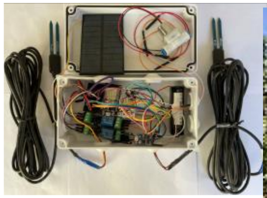
January 22, 2025

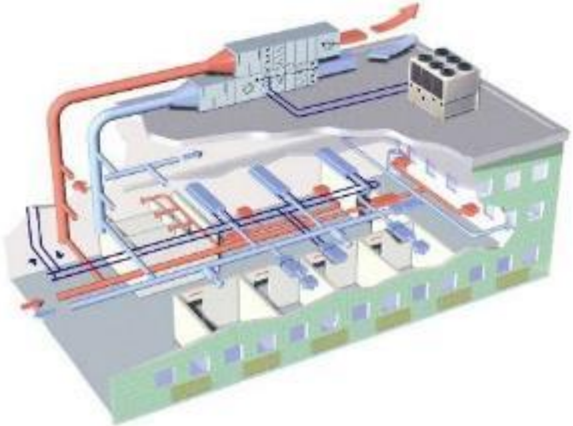
Wan Du, PhD

Who are we?

- Wan Du, Associate Professor
  - Office hours:
    - » Wednesday: 1:30-2:45 PM
    - » SE2 #208
  - Email: wdu3@ucmerced.edu
  - Preference: Meet during my office hours, rather than sending emails.
- Research areas:
  - Internet of Things
    - » Mobile Computing
    - » Networked Embedded Systems
    - » Data Analytics
    - » Deep Reinforcement Learning

[Wireless Systems Developed in Du's Lab]

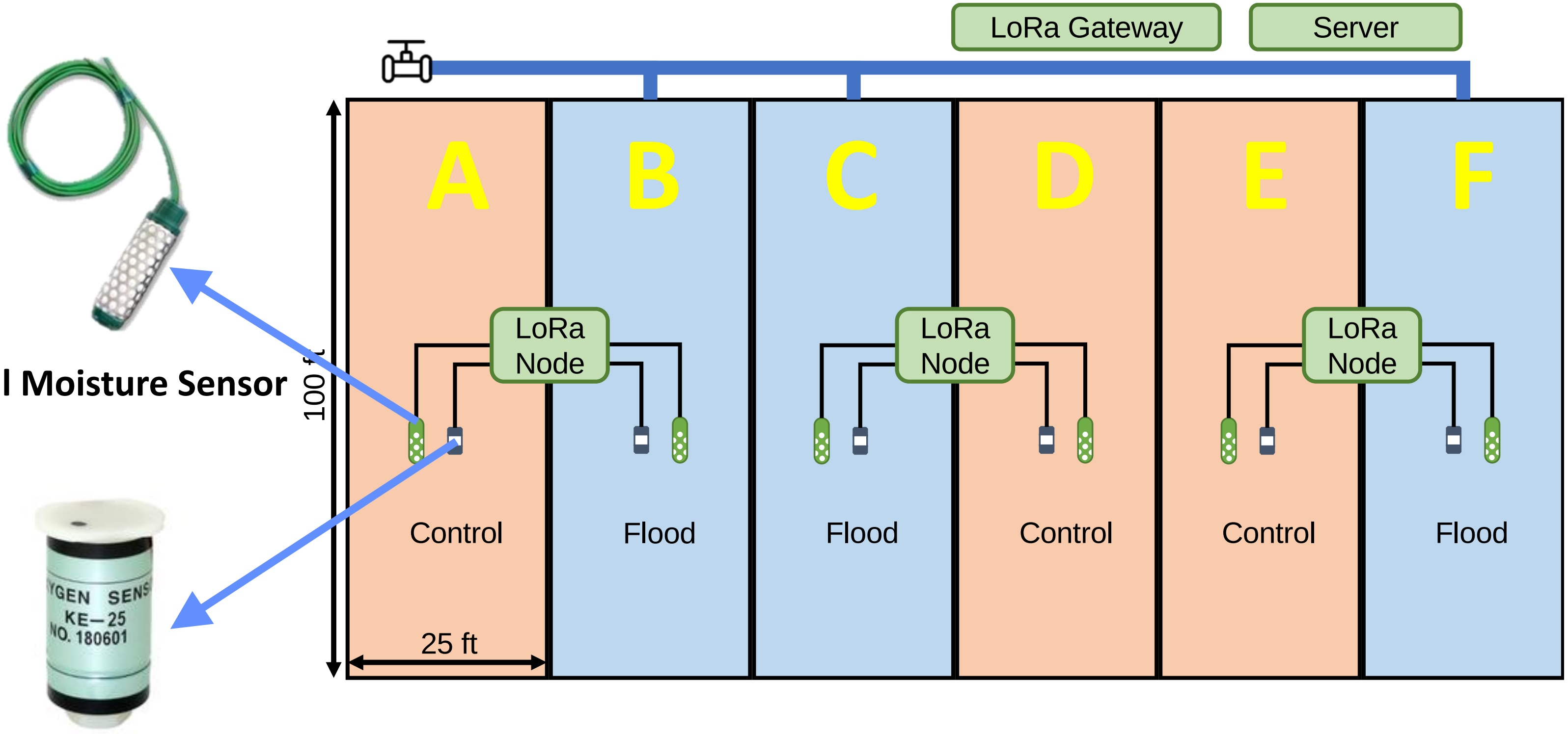




UC ANR Kearney Research & Extension Center

LoRa Networking in Orchards

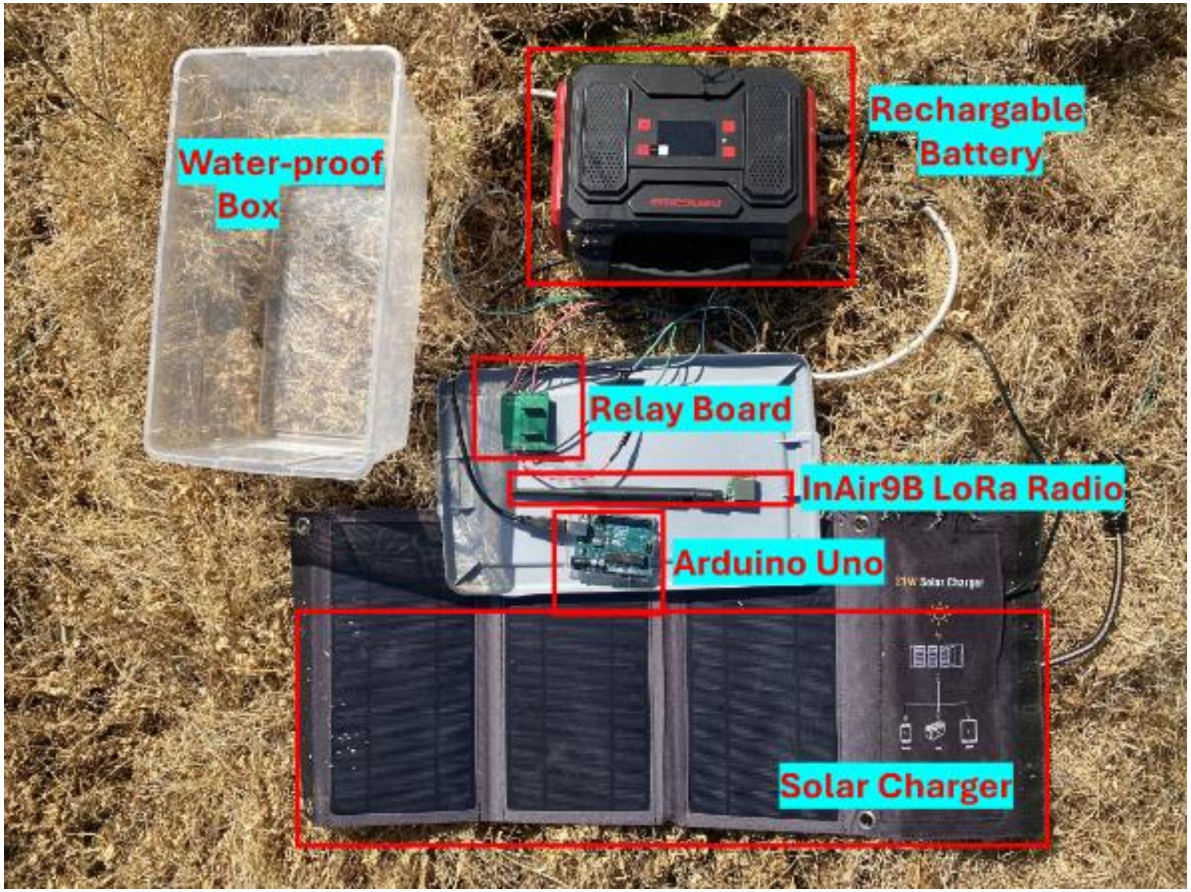
Testbed at Kearney



Soil Moisture Sensor

Oxygen Sensor

Continuous Monitoring Experience

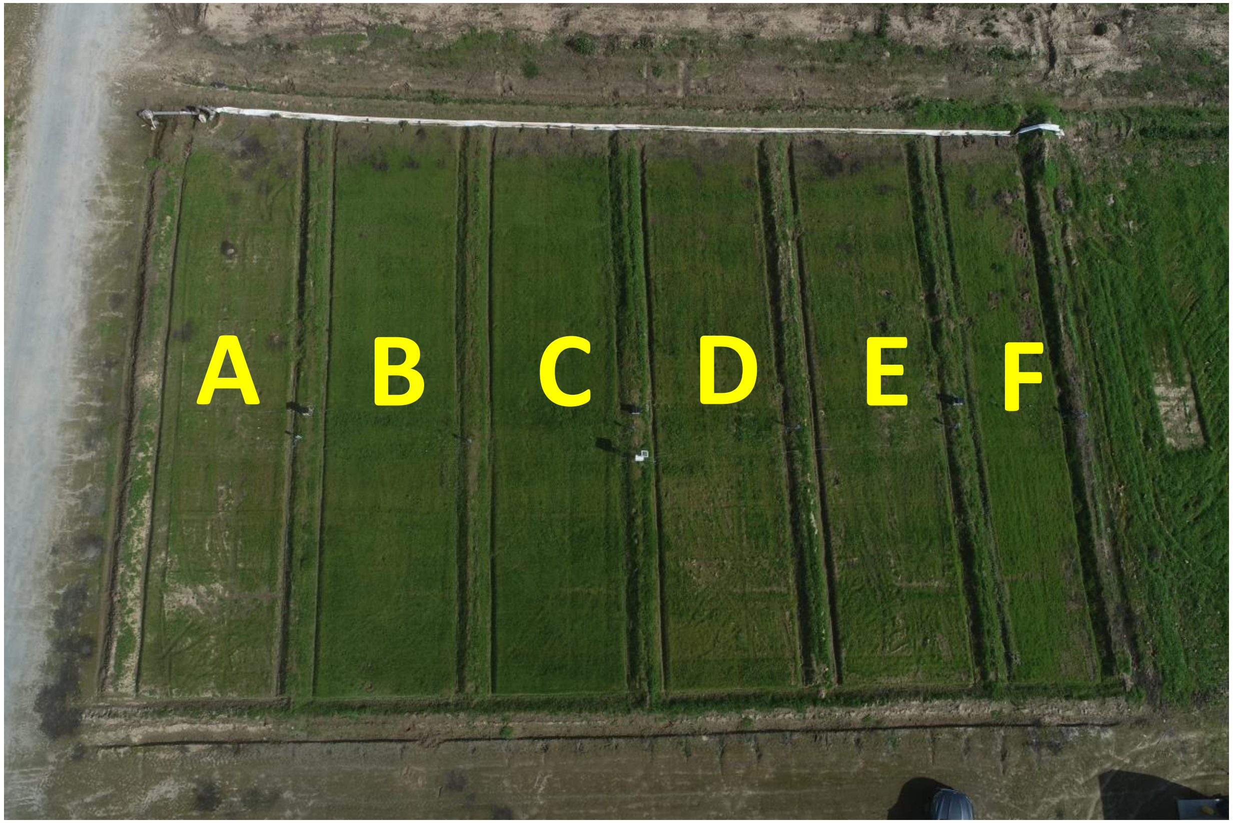


Water-proof

Free of changing battery

Realtime remote access of data flow and gateways

Testbed at Kearney



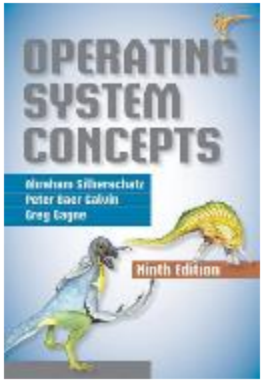
Flooding trials

Today

- Course details
- Introduction of OS

Textbook & Readings

- Textbook
  - Operating Systems Concepts, 9th Edition Silbershatz, Galvin, Gagne
  - 8th edition or 10th edition is acceptable.
- Chapter reading assignment posted on Lecture slides



- Recommended:
  - Andrew S. Tanenbaum and Herbert Bos. Modern Operating Systems, 4th Edition. Prentice-Hall, 2014.

Topics Covered

- OS Concepts: How to Navigate as a Systems Programmer!
- Process
  - Threads, scheduling, locks, deadlock, scalability, fairness
- File Systems
  - I/O devices, file objects, storage, naming, caching, performance, paging, transactions, databases
- Networks
  - Protocols, Scalability, multicast
- Reliability & Security
  - Fault tolerance, protection, security

Course Overview

- Prerequisites
  - CSE21 - Java and C
  - CSE31 - Computer architecture
  - CSE100 - Algorithm Design and Analysis
- CatCourses
  - Announcements and Assignments
  - Lecture slides
  - Grades for assignments will also be found there (secure).
- Three homeworks
  - One week to finish each.
- In-class exam
  - No web-exam will be provided.
  - Review will be offered before the exam.
- Final exam
  - No web-exam will be provided.
- Project 1: **00:01 AM on 02/18 (Tue) – 11:59 PM on 03/17 (Monday)**
- Project 2: **00:01 AM on 03/19 (Wed) - 11:59 PM on 04/15 (Tue)**

Grading

- 3 homework: 18% (10% penalty for each day if submitted late)
- 1 in-class tests: 15%
- Final exam: 29%
- 2 Projects: 38%
  - Participation: 8% for 8 lab sessions.
  - Design documents: 5%*2 (10% penalty for each day if submitted late)
  - Final codes: 10%*2 (10% penalty for each day if submitted late)
  - Peer review: Adjust the score individually.
  - Code graded by autograder and TA
  - Partial credit for failed tests:
    - Assigned by manual code review, comments in code, and logic (design document)

Learning by Doing: Projects

- Lab sessions start in **the week of 02/10**.
  - Server account setup and Nachos preparation
- Two Group Projects (Nachos in Java)
  - 1. Threads & Scheduling
  - 2. Multiprogramming and system calls for user-programs
- Two projects, 4 weeks/each.
  - Design document due in 2 weeks.
  - Code and final design document due in 4 weeks.
  - 2 slip days

Additional Project Details

- Projects involve two stages
  - Initial design document followed by design review with TA
  - Code and final design submission
- Submission of project
  - Both design document and code are submitted via ssh *file transfer* to klwin00.ucmerced.edu
  - Make sure …
    - » … your submissions for each project are in separate folders that you create at the server (do not over-write your project 1 submission with project 2 submission, etc.).
    - » … you only have one submission (per project) in the server at the time of code submission deadline
    - » … your design document contains your group number and member names

Group Project Simulates Industrial Environment

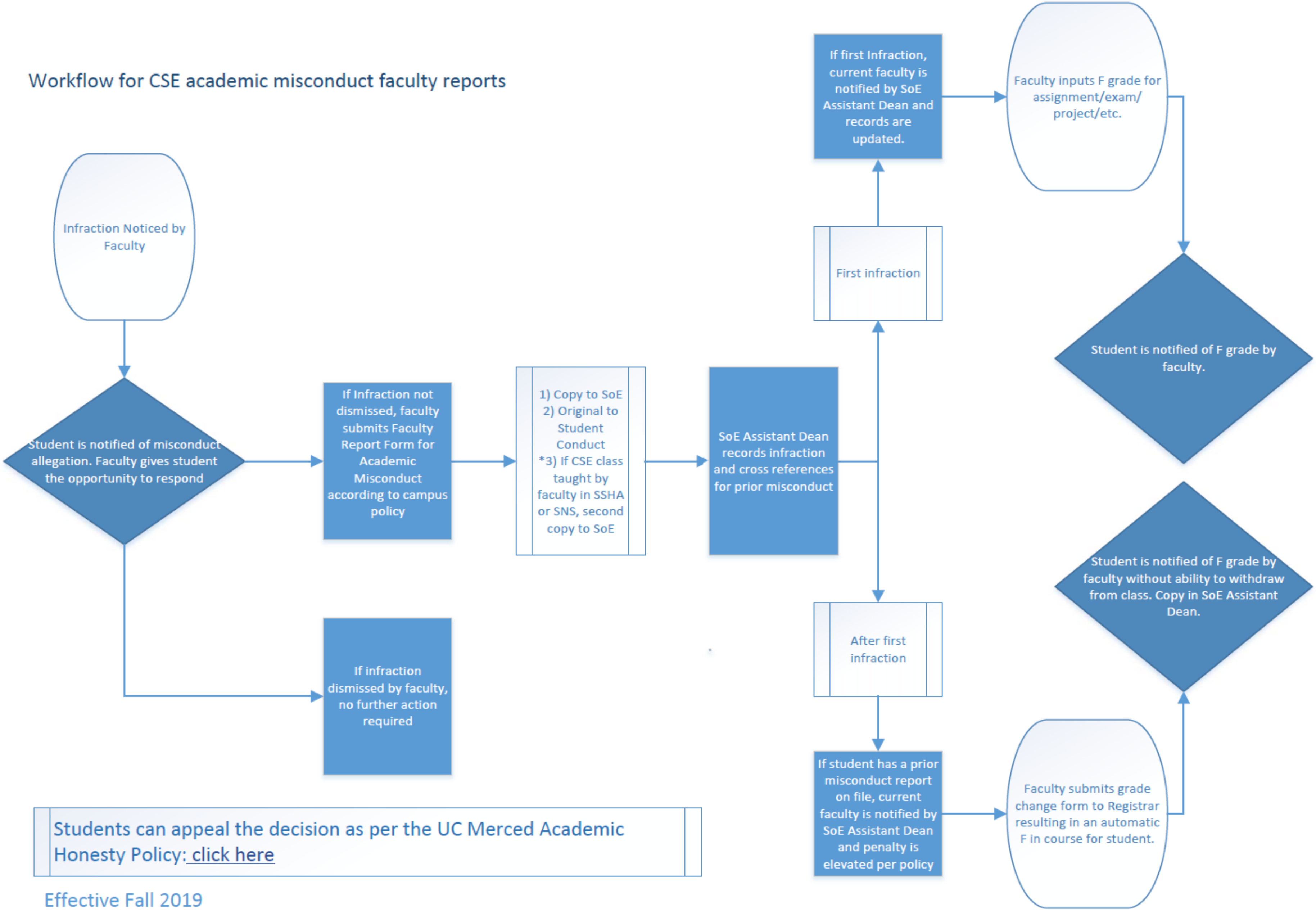
- Project teams have 5-6 members
  - Form your team by yourself by **the first lab.**
  - Must work in groups in "the real world"
- Communicate with colleagues (team members)
  - Communication problems are natural
  - What have you done?
  - What answers you need from others?
  - You must document your work!!!
- Communicate with TAs
  - What is the team's plan?
  - What is each member's responsibility?
  - Design Documents: High-level description for a manager!

CS 150 Collaboration Policy

UC Merced Academic Honesty Policy

Computer Science Department Policy on Academic Honesty on CatCourse

Workflow for CSE academic misconduct faculty reports



Infraction Noticed by Faculty

Student is notified of misconduct allegation. Faculty gives student the opportunity to respond

If Infraction not dismissed, faculty submits Faculty Report Form for Academic Misconduct according to campus policy

1) Copy to SoE
2) Original to Student Conduct
*3) If CSE class taught by faculty in SSHA or SNS, second copy to SoE

SoE Assistant Dean records infraction and cross references for prior misconduct

First infraction

If first Infraction, current faculty is notified by SoE Assistant Dean and records are updated.

Faculty inputs F grade for assignment/exam/ project/etc.

Student is notified of F grade by faculty.

After first infraction

If student has a prior misconduct report on file, current faculty is notified by SoE Assistant Dean and penalty is elevated per policy

Faculty submits grade change form to Registrar resulting in an automatic F in course for student.

Student is notified of F grade by faculty without ability to withdraw from class. Copy in SoE Assistant Dean.

If infraction dismissed by faculty, no further action required

Students can appeal the decision as per the UC Merced Academic Honesty Policy: click here

Effective Fall 2019

See full Policy HERE

CS 150 Collaboration Policy

✓

Explaining a concept to someone in another group

Discussing algorithms/testing strategies with other groups

Helping debug someone else’s code (in another group)

Searching online for generic algorithms (e.g., hash table)

✗

Sharing code or test cases with another group

Copying OR reading another group’s code or test cases

Copying OR reading online code or test cases from prior years

We compare all project submissions against prior year submissions and online solutions and will take actions (described in the syllabus) against offenders

Peer review

- Form
- Algorithm

Group Member	Partici pation	Liste ning	Feed back	Coope ration	Time	Contrib ution	Quality of Completed Work	Average	Percentage
TA									50%
Self									50%*1/3
Member A									50%*1/3
Member B									50%*1/3

Lecture Guidelines

- Understand the material being presented
- Please don’t hesitate to ask or answer questions
  - Raise your hand, or just speak up
- Please speak up if you have comments, suggestions, additional interesting points, or even disagreements to share
- Please be courteous of others
  - Turn off your cell phones

Hints for success

- Attend lecture
- Attend all lab sessions
- Preview and review my slides
- Homework and exams
- Do & understand the project YOURSELF
- Ask questions

Emailing

Ask necessary questions: Check the syllabus and other course materials first.

✓ Direct lab questions first to your TA.

Talk with me in my office during my office hours.

Expected reply latency.

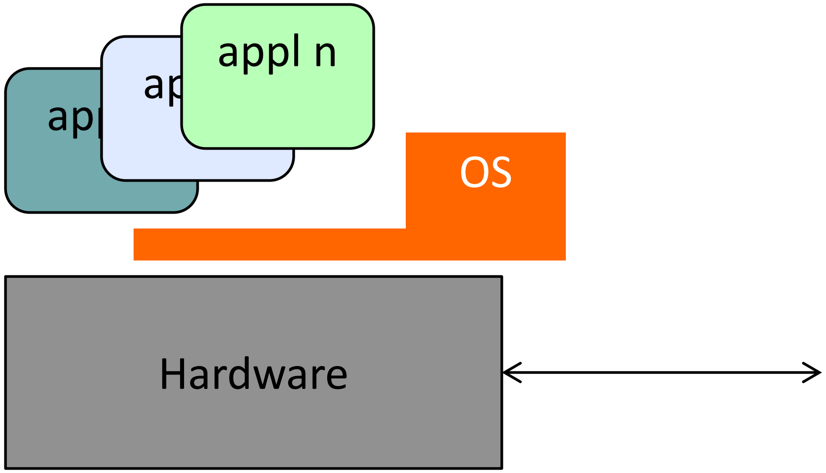
Message: Make it short and clear

Subject: CSE150 - ??

Let’s us know your FULL name.

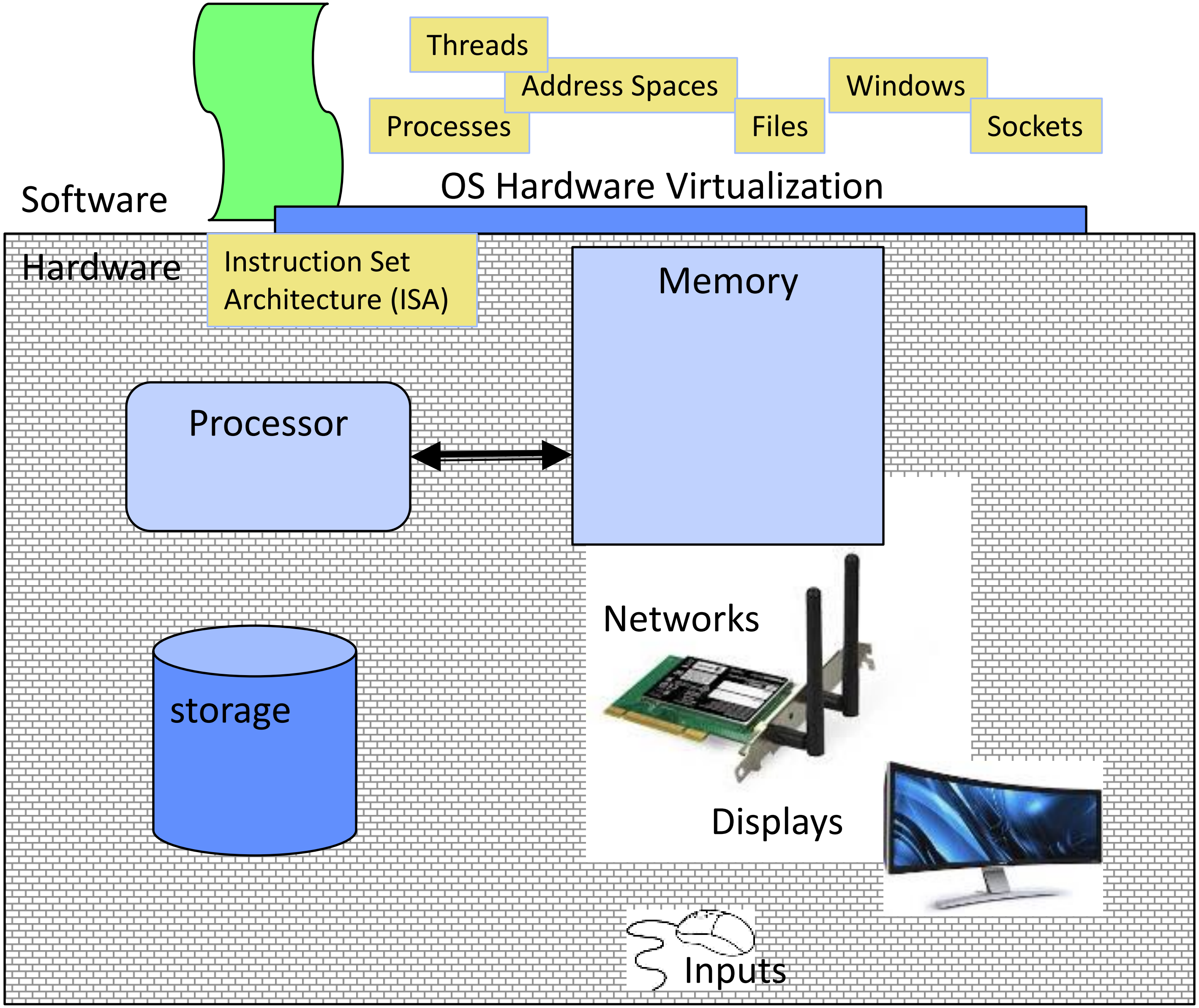
What is an operating system?

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

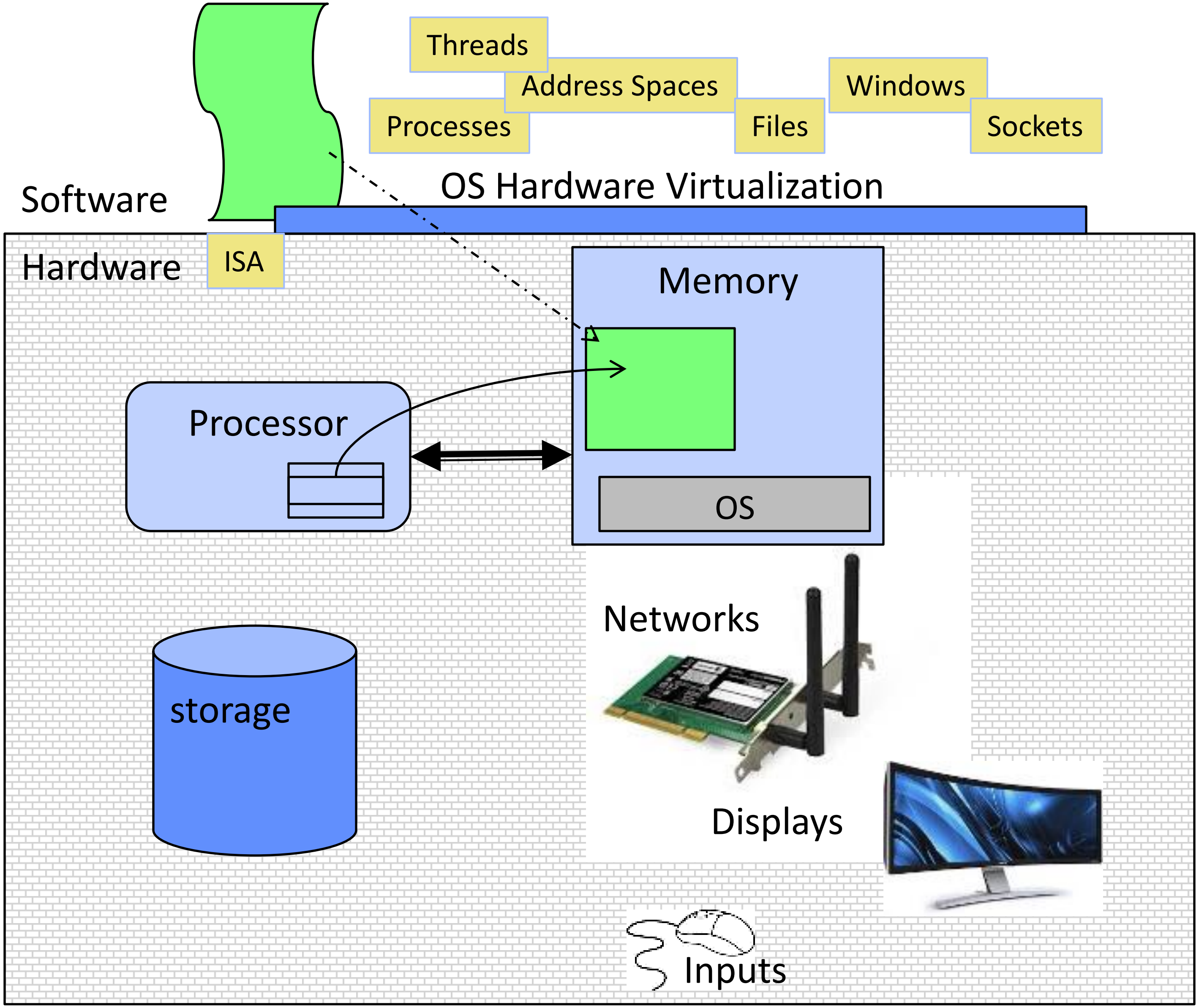


Slides adapted from CS162 Course Material at UC Berkeley

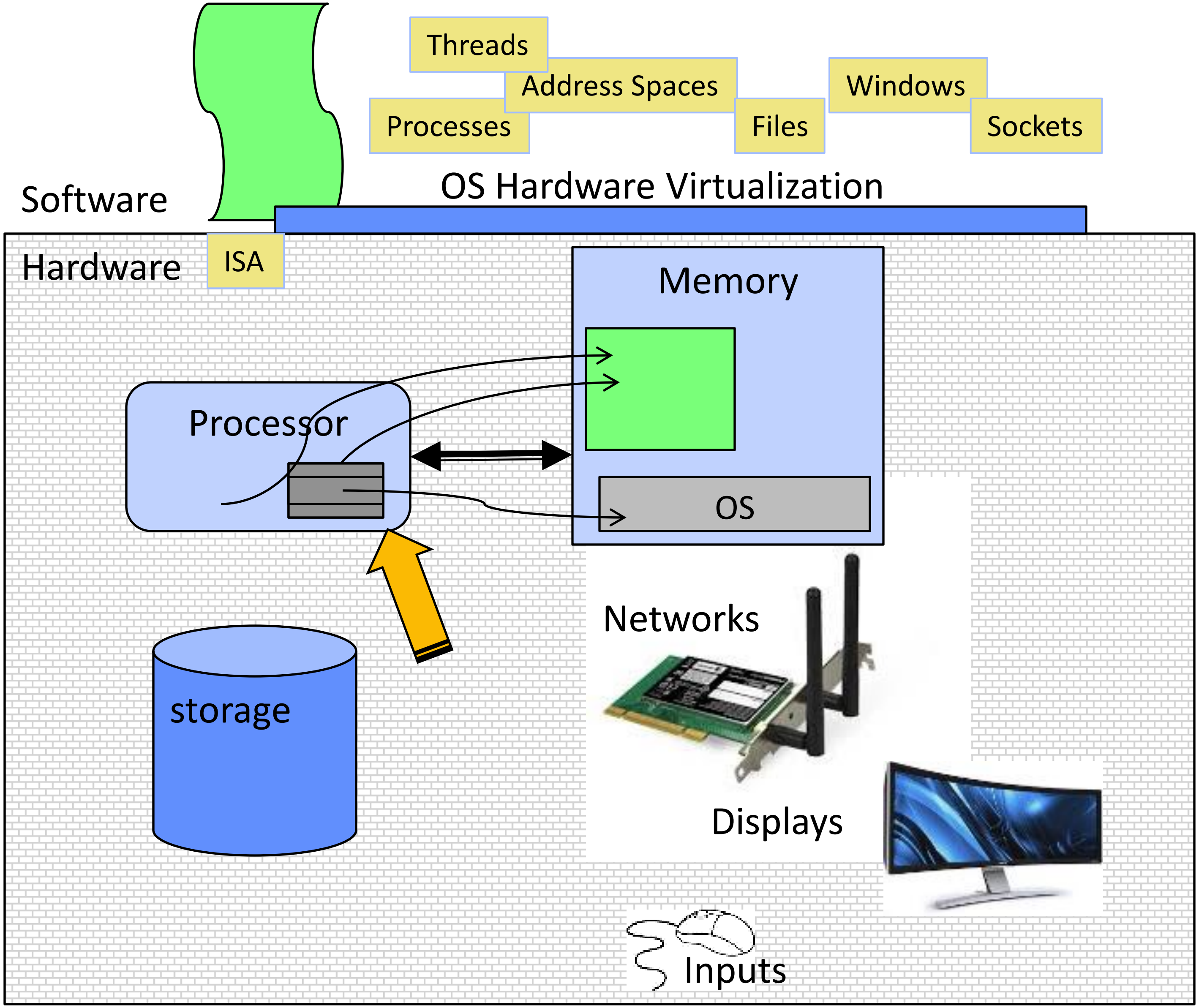
OS Basics: "Virtual Machine" Boundary



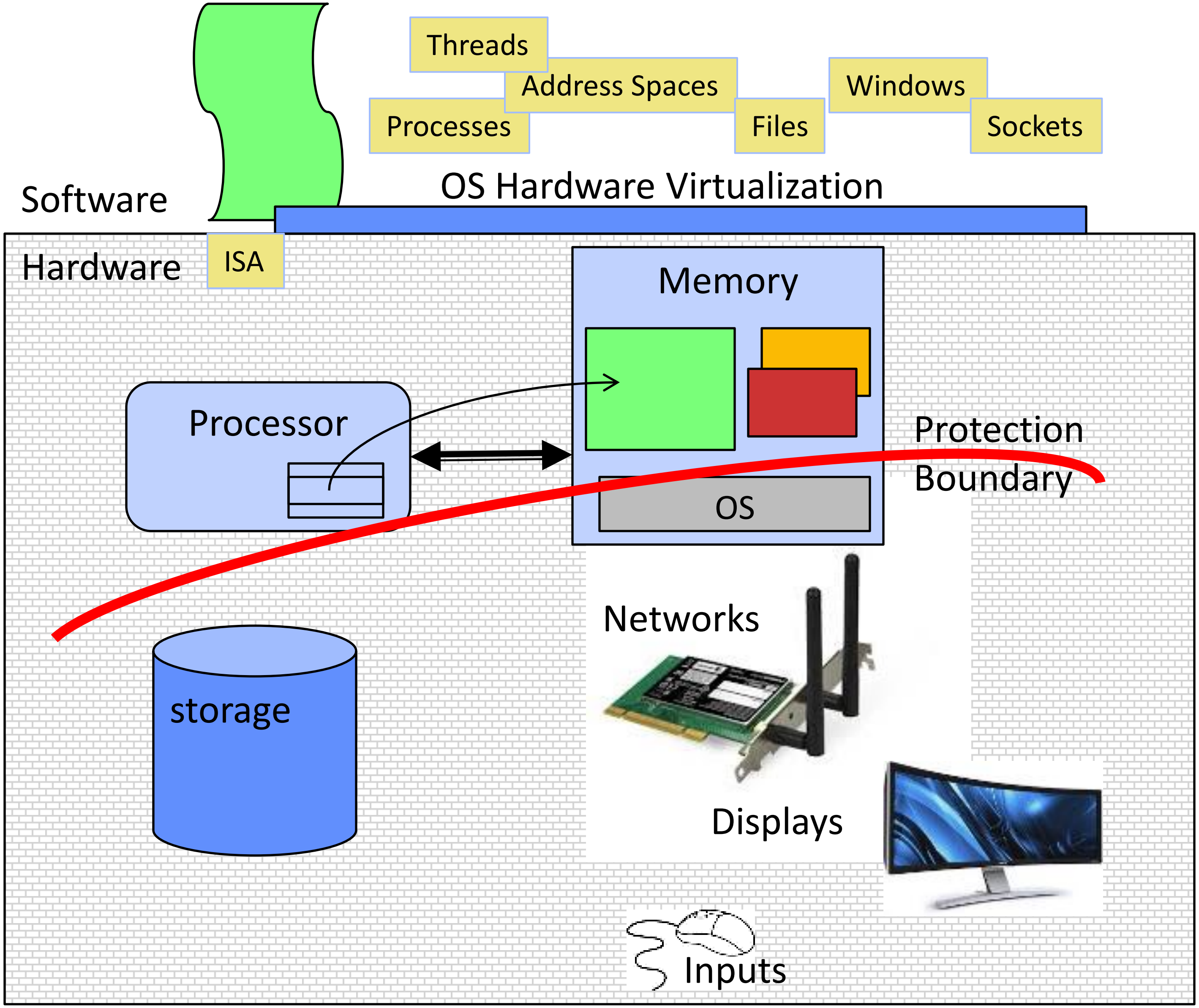
OS Basics: Program => Process



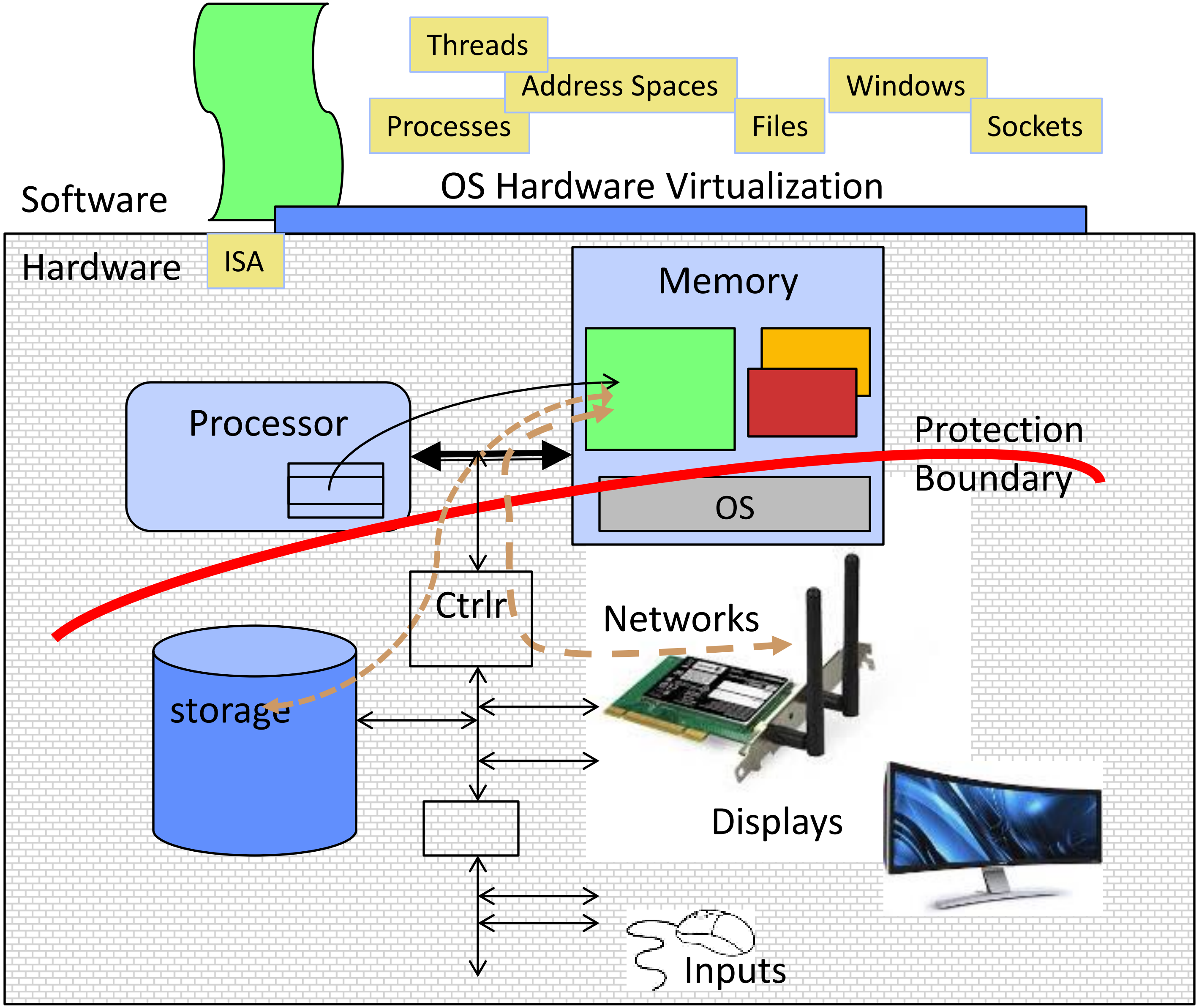
OS Basics: Context Switch



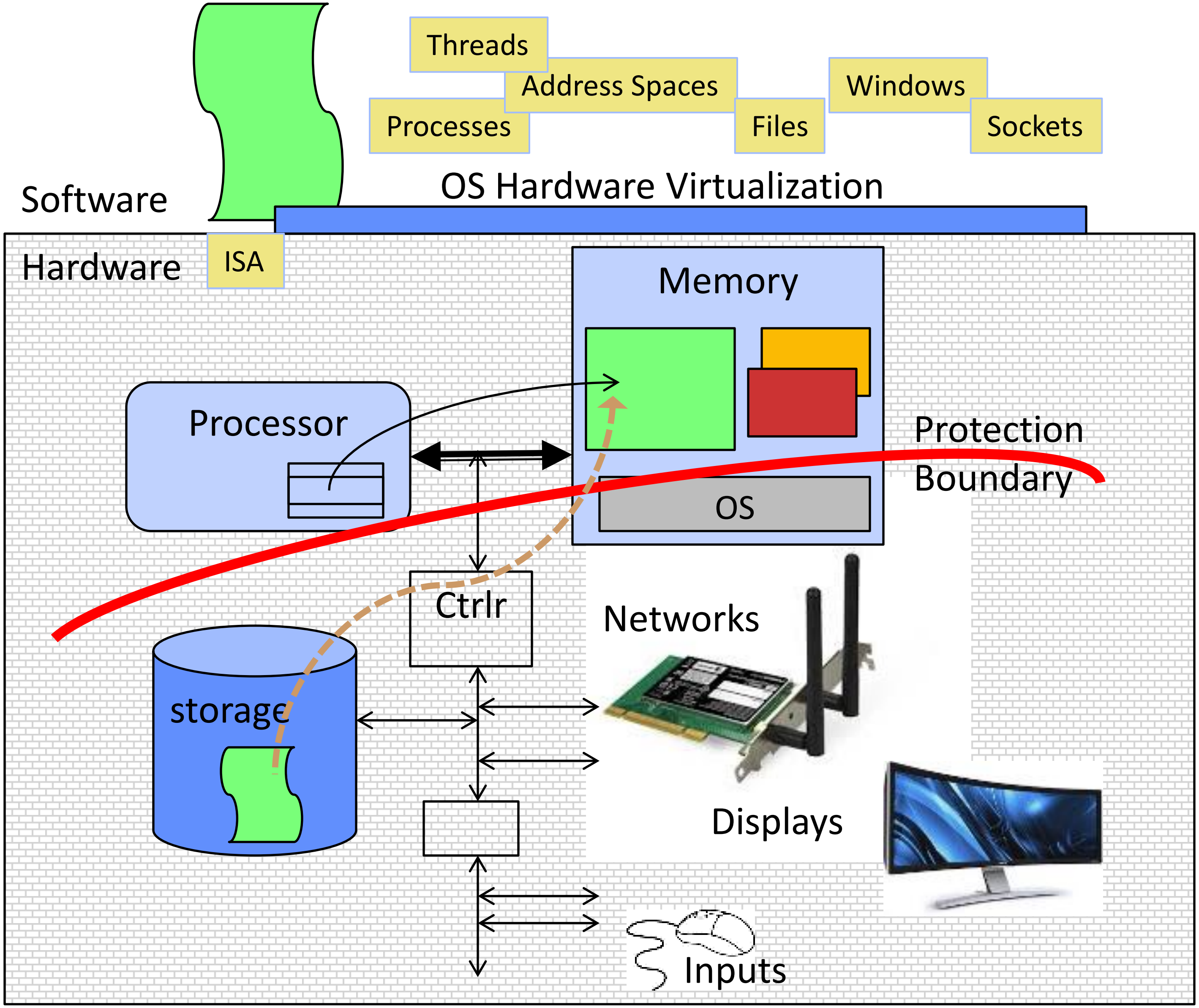
OS Basics: Scheduling, Protection



OS Basics: I/O


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

OS Basics: Loading



We need an OS.

- Manage resources:
  - Memory, CPU, storage, …
- Provide abstractions
  - File systems
  - Processes, threads
  - VM, containers
  - Naming system
  - …

What makes Operating Systems Exciting and Challenging?

Challenge: Complexity

- Applications consisting of…
  - … a variety of software modules that …
  - … run on a variety of devices (machines) that
    - » … implement different hardware architectures
    - » … run competing applications
    - » … fail in unexpected ways
    - » … can be under a variety of attacks

- Not feasible to test software for all possible environments and combinations of components and devices
  - The question is not whether there are bugs but how serious are the bugs!

Challenge: Complexity

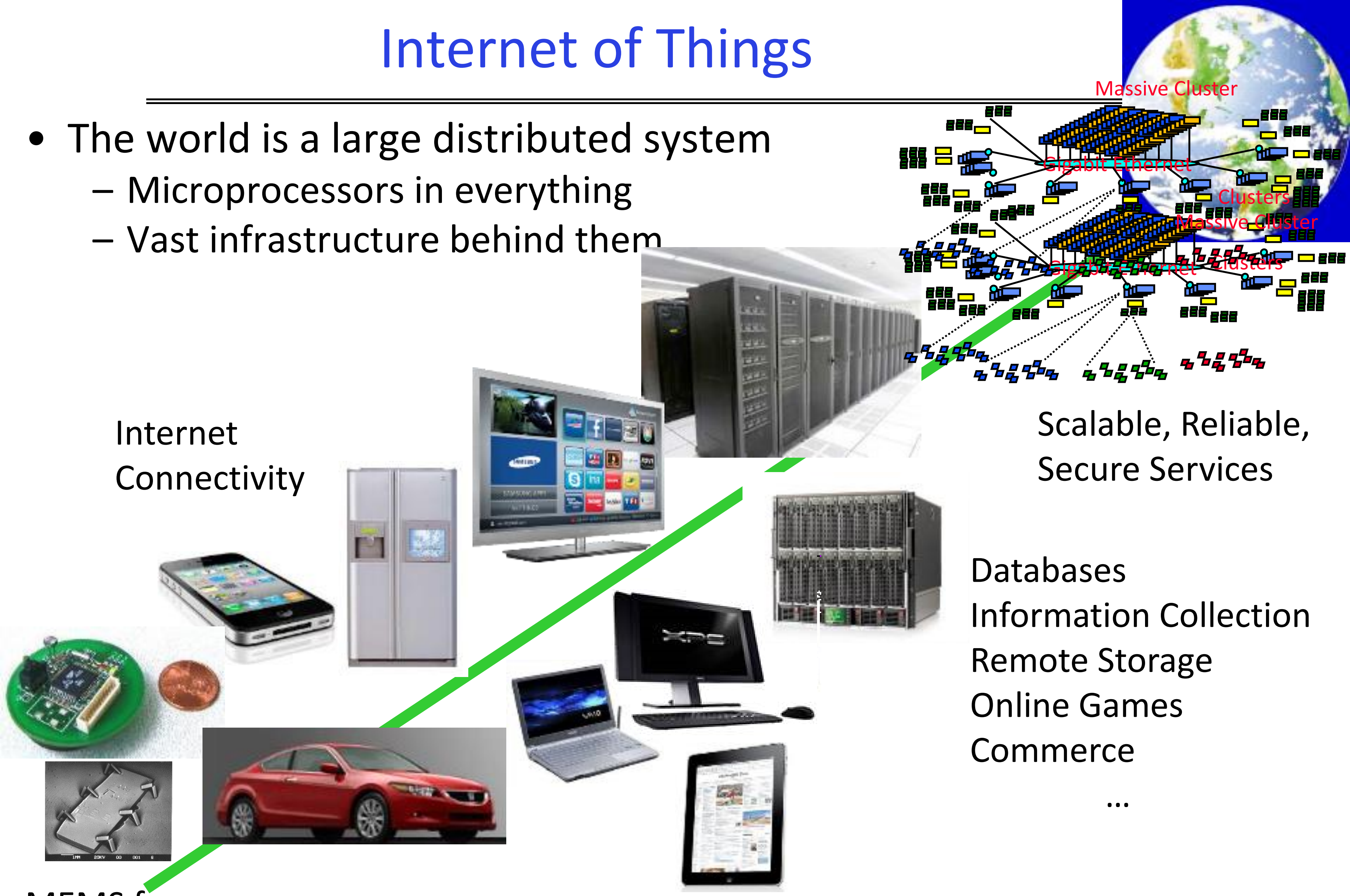
- Enormous scale, heterogeneity, and dynamic range:
  - CPU: sensors → GPUs
    - » Cores: one → 100s [2-orders of magnitude variation]
    - » Clusters: few machines → 10,000s machines [4 orders of mag.]
  - Storage: caches ⟶ disks
    - » Size: MB → TB [6 orders of mag.]
    - » Access time: few nanosecs → ms [6 orders of mag.]
  - Network: Inter-core networks → Internet
    - » Latency: nanosecs → secs (satellite) [9 orders of mag.]
    - » Bandwidth: Kbps → Gbps [6 orders of mag.]
- Interaction
  - Complex interaction between system components
  - Unexpected failure scenarios, e.g., randomly flipping a memory bit

How do we tame complexity?

- Does the programmer need to write a single program that performs many independent activities?
- Does every program have to be altered for every piece of hardware?
- Does a faulty program crash everything?
- Does every program have access to all hardware?

Internet of Things

- The world is a large distributed system
  - Microprocessors in everything
  - Vast infrastructure behind them



Internet Connectivity

Scalable, Reliable, Secure Services

Databases
Information Collection
Remote Storage
Online Games
Commerce
…

MEMS for Sensor Nets

Next Wek

- Reading assignment
  - S&G 9th/8th/7th Ch 1 - 4
- No Lab.