CS 1550
Introduction to Operating Systems

RECITATION – WEEK 3

PRANUT JAIN

OFFICE INFO

- Office hours: Tuesday 10:30am to 3pm AND by appointment.
- Office location: SENSQ 5501
- Recitation Materials: https://people.cs.pitt.edu/~pranut/

PLAN FOR TODAY

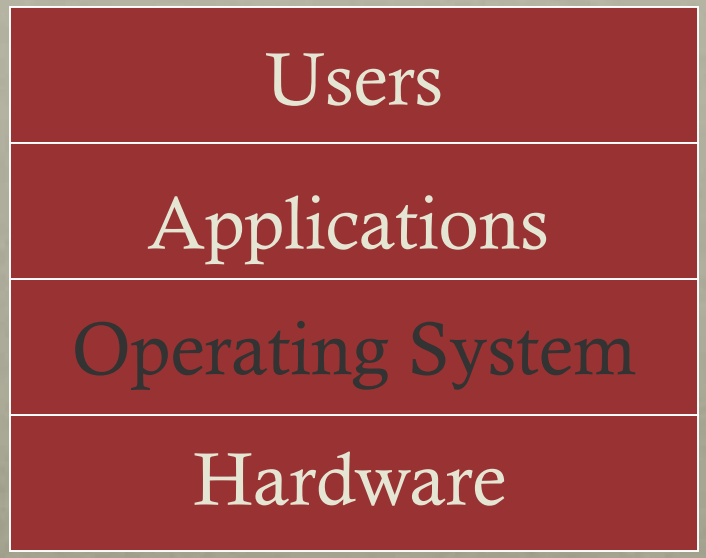
- Project 0 and Project 1 discussion
- OS basics
- Processes

DISCLAIMER

- Slides originally by Andrea for Professor Remzi H. Arpaci-Dusseau, UNIVERSITY of WISCONSIN-MADISON

WHAT IS AN OPERATING SYSTEM?

Not easy to define precisely…



Operating System (OS):

Software that converts hardware into a useful form for applications

WHAT DOES OS PROVIDE?

Role #1: Abstraction - Provide standard library for resources

What is a resource?

- Anything valuable (e.g., CPU, memory, disk)

What abstraction does modern OS typically provide for each resource?

- CPU:
  - process and/or thread
- Memory:
  - address space
- Disk:
  - files

Advantages of OS providing abstraction?

- Allow applications to reuse common facilities
- Make different devices look the same
- Provide higher-level or more useful functionality

Challenges

- What are the correct abstractions?
- How much of hardware should be exposed?

WHAT DOES OS PROVIDE?

Role #2: Resource management – Share resources well

Advantages of OS providing resource management?

- Protect applications from one another
- Provide efficient access to resources (cost, time, energy)
- Provide fair access to resources

Challenges

- What are the correct mechanisms?
- What are the correct policies?

Three Pieces: First

- Virtualization
  - Make each application believe it has each resource to itself
- Example
  - Virtualize CPU and memory

THREE PIECES: SECOND

- **Concurrency:**
  Events are occurring simultaneously and may interact with one another
- OS must be able to handle concurrent events
- Easier case
  - Hide concurrency from **independent** processes
- Trickier case
  - Manage concurrency with **interacting** processes
    - Provide abstractions (locks, semaphores, condition variables, shared memory, critical sections) to processes
    - Ensure processes do not deadlock
- Example
  - Interacting threads must coordinate access to shared data

THREE PIECES: THIRD

- **Persistence:** Access information permanently
  - Lifetime of information is longer than lifetime of any one process
  - Machine may be rebooted, machine may lose power or crash unexpectedly
- Issues:
  - Provide abstraction so applications do not know how data is stored : Files, directories (folders), links
  - Correctness with unexpected failures
  - Performance: disks are very slow; many optimizations needed!
- Example
  - File system does work to ensure data updated correctly

ADVANCED TOPICS

- Current systems
  - Multiprocessors
  - Networked and distributed systems
  - Virtual machines

Why Study Operating Systems?

Build, modify, or administer an operating system

Understand system performance

- Behavior of OS impacts entire machine
- Tune workload performance
- Apply knowledge across many layers
  - Computer architecture, programming languages, data structures and algorithms, and performance modeling

Fun and challenging to understand large, complex systems