CSE 4/521 Introduction to Operating Systems

Instructor: Jerry Ajay

Summer 2018

Table of Content

About Myself

- Name: Jerry Ajay
- Advisor: Dr. Wenyao Xu
- Research Interest: Computer Architecture (both hardware + software)
- Secondary research interest: Building novel computer systems to address neuro-degenerative diseases.

- Undergraduate: Bachelors in Engineering (CS), Bangalore University, India (2009-2013)
- Graduate: MS in UB (2013-2015) , PhD (2015-current)

High Level Logistics

Location and Time:

- Class - NSC 205 (MWF 11-12.05PM)
- Recitation - Bell 138 (MWF 12.05-12.30PM)

Office hours conducted during recitation

For specific details wanting my personal attention, please email me for an appointment: jerryant@buffalo.edu

What will you not learn in this course

- C programming – (Have to learn by yourself)
- Any sophistication algorithm.
- A one shot solution to the two programming projects.

What will you learn in this course

- To think like a systems programmer (very different from applications programmer).
- A clear understanding about the fundamentals of all existing operating systems – Linux, Windows, Solaris, Unix, etc.
- To compile/re-compile ‘a lot’ to make things work.

Grading rubrics

THEORY (60%)

- **Mid-terms – 25%**
- **Finals – 25%**
- **HW + Quizzes – 10%** (5 quizzes and 5 homeworks)

PROJECT (40%)

- **Pintos Checkpoint 1 [Thread Support] – 20%** (15% source code + 5% design doc) Objective: Understand thread synchronization.
- **Pintos Checkpoint 2 [User Programs] – 20%** (15% source code + 5% design doc) Objective: Understand user program interaction with kernel.

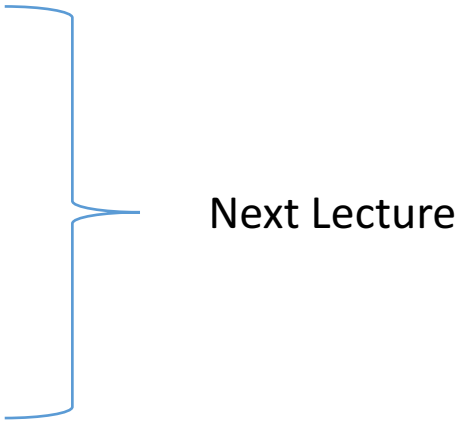
Additional Points of Note

- This is a 12 week course. May 30th – Aug 17th .
- Grads and undergrads would be graded on separate curves.
- Please note the academic integrity at: http://academicintegrity.buffalo.edu/policies
- Projects to be done in groups of 2-3. Be on the lookout for potential teammates.

Introduction

Overview

- What Operating Systems Do
- Computer-System Architecture
- Operating-System Structure
- Operating-System Operations
- Process Management
- Memory Management
- Storage Management
- Protection and Security
- Kernel Data Structures
- Computing Environments
- Open-Source Operating Systems

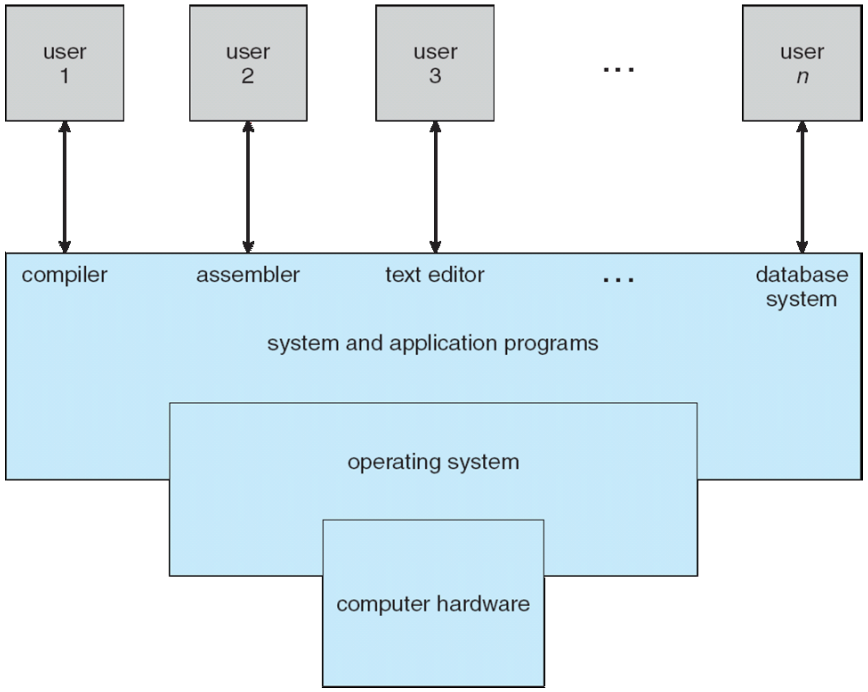


Overview

- What Operating Systems Do
- Computer-System Architecture
- Operating-System Structure
- Operating-System Operations
- Process Management
- Memory Management
- Storage Management
- Protection and Security
- Kernel Data Structures
- Computing Environments
- Open-Source Operating Systems

Next Lecture

What Operating Systems Do



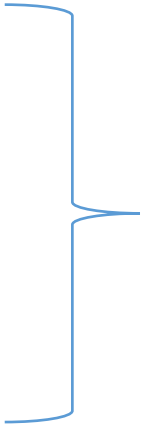
1. Could you infer the definition of an OS from this diagram?
2. What are the goals of an operating system?

What Operating Systems Do

- A program that acts as an intermediary between a user of a computer and the computer hardware
- Operating system goals:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

Overview

- What Operating Systems Do
- Computer-System Architecture
- Operating-System Structure
- Operating-System Operations
- Process Management
- Memory Management
- Storage Management
- Protection and Security
- Kernel Data Structures
- Computing Environments
- Open-Source Operating Systems



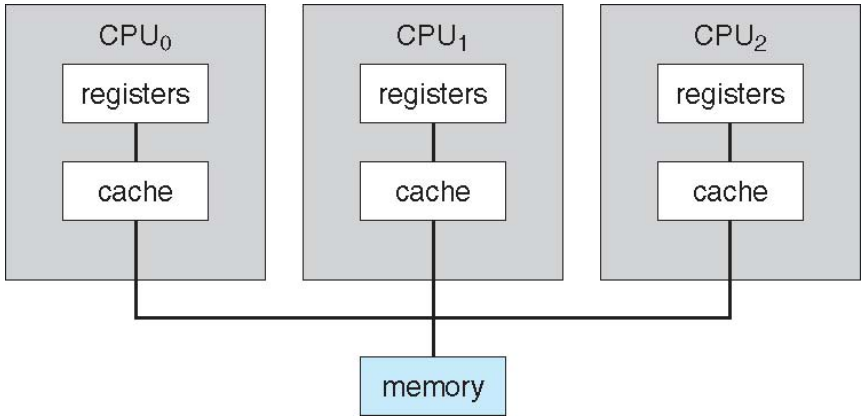
Next Lecture

Computer-System Architecture

- Most systems use a single general-purpose processor
  - Most systems have special-purpose processors as well
- **Multiprocessors** systems growing in use and importance
  - Advantages include:
    1. **Increased throughput**
    2. **Economy of scale**
    3. **Increased reliability** – graceful degradation or fault tolerance
  - Two types:
    1. **Asymmetric Multiprocessing** – each processor is assigned a specie task.
    2. **Symmetric Multiprocessing** – each processor performs all tasks

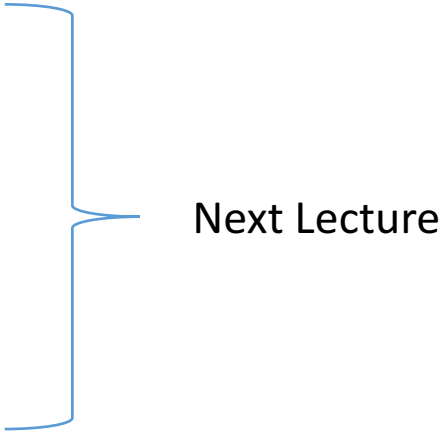
Computer-System Architecture

Symmetric Multiprocessing



Overview

- What Operating Systems Do
- Computer-System Architecture
- Operating-System Structure
- Operating-System Operations
- Process Management
- Memory Management
- Storage Management
- Protection and Security
- Kernel Data Structures
- Computing Environments
- Open-Source Operating Systems



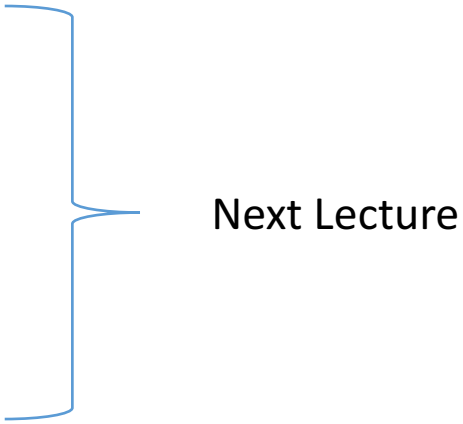
Operating-System Structure

1. Multiprogramming (Batch system) needed for efficiency
   - Single user cannot keep CPU and I/O devices busy at all times
   - One job selected and run via job scheduling
   - When it has to wait (for I/O for example), OS switches to another job

2. Timesharing (Multitasking) is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating interactive computing
   - Response time should be < 1 second
   - If several jobs ready to run at the same time ⇨ CPU scheduling
   - If processes don't fit in memory, swapping moves them in and out to run
   - Virtual memory allows execution of processes not completely in memory

Overview

- What Operating Systems Do
- Computer-System Architecture
- Operating-System Structure
- Operating-System Operations
- Process Management
- Memory Management
- Storage Management
- Protection and Security
- Kernel Data Structures
- Computing Environments
- Open-Source Operating Systems



Operating Systems Operations

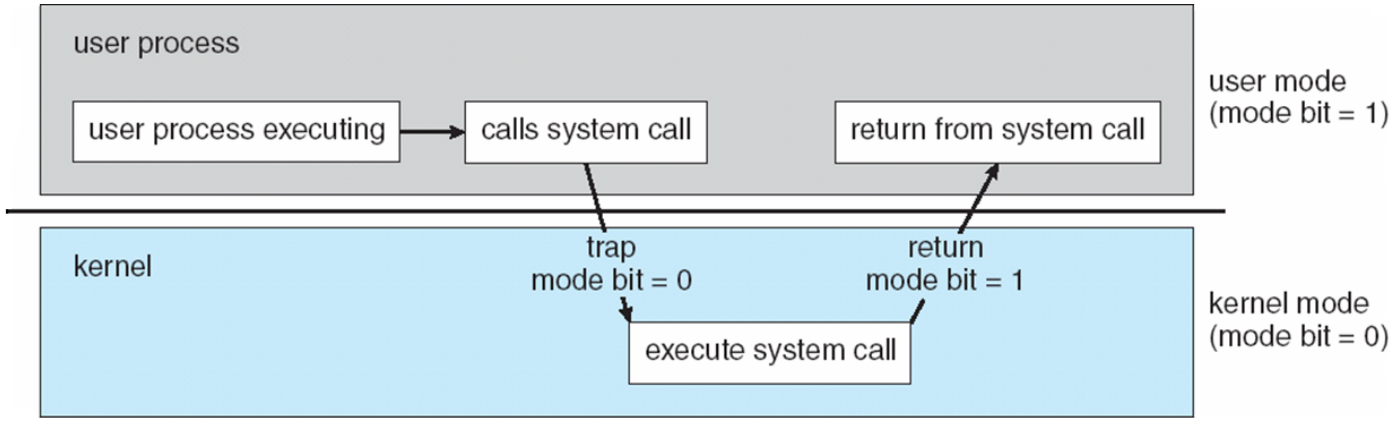
- Interrupt driven (hardware and software)
  - Hardware interrupt by one of the devices
  - Software interrupt (**exception** or **trap**):
    - Software error (e.g., division by zero)
    - Request for operating system service
    - Other process problems include infinite loop, processes modifying each other or the operating system

Operating Systems Operations

- Dual-mode operation allows OS to protect itself and other system components
  - User mode and kernel mode
  - Mode bit provided by hardware
    - Provides ability to distinguish when system is running user code or kernel code
    - Some instructions designated as privileged, only executable in kernel mode
- Increasingly CPUs support multi-mode operations
  - i.e. virtual machine manager (VMM) mode for guest VMs

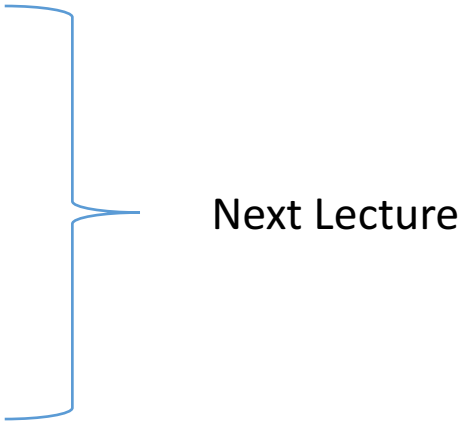
Operating Systems Operations

- Timer to prevent infinite loop / process hogging resources
  - Timer is set to interrupt the computer after some time period
  - Keep a counter that is decremented by the physical clock.
  - Operating system set the counter (privileged instruction)
  - When counter zero, generate an interrupt
  - Set up before scheduling process to regain control or terminate program that exceeds allotted time



Overview

- What Operating Systems Do
- Computer-System Architecture
- Operating-System Structure
- Operating-System Operations
- Process Management
- Memory Management
- Storage Management
- Protection and Security
- Kernel Data Structures
- Computing Environments
- Open-Source Operating Systems



Process Management

- A process is a program in execution. It is a unit of work within the system. Program is a passive entity, process is an active entity.
- Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Initialization data
- Single-threaded process has one program counter specifying location of next instruction to execute
  - Process executes instructions sequentially, one at a time, until completion
- Multi-threaded process has one program counter per thread
- Typical systems have many processes, some user, some operating system running concurrently on one or more CPUs

Overview

- What Operating Systems Do
- Computer-System Architecture
- Operating-System Structure
- Operating-System Operations
- Process Management
- Memory Management
- Storage Management
- Protection and Security
- Kernel Data Structures
- Computing Environments
- Open-Source Operating Systems

Next Lecture

Memory Management

- To execute a program all (or part) of the instructions must be in memory
- All (or part) of the data that is needed by the program must be in memory.
- Memory management determines what is in memory and when
  - Optimizing CPU utilization and computer response to users
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating memory space as needed

Overview

- What Operating Systems Do
- Computer-System Architecture
- Operating-System Structure
- Operating-System Operations
- Process Management
- Memory Management
- Storage Management
- Protection and Security
- Kernel Data Structures
- Computing Environments
- Open-Source Operating Systems

Next Lecture

Credential for slides

Silberschatz, Galvin and Gagne