Operating Systems

CS220

Lecture 1

Introduction

8th March 2021

Myself

Dr. Rana Asif Rehman

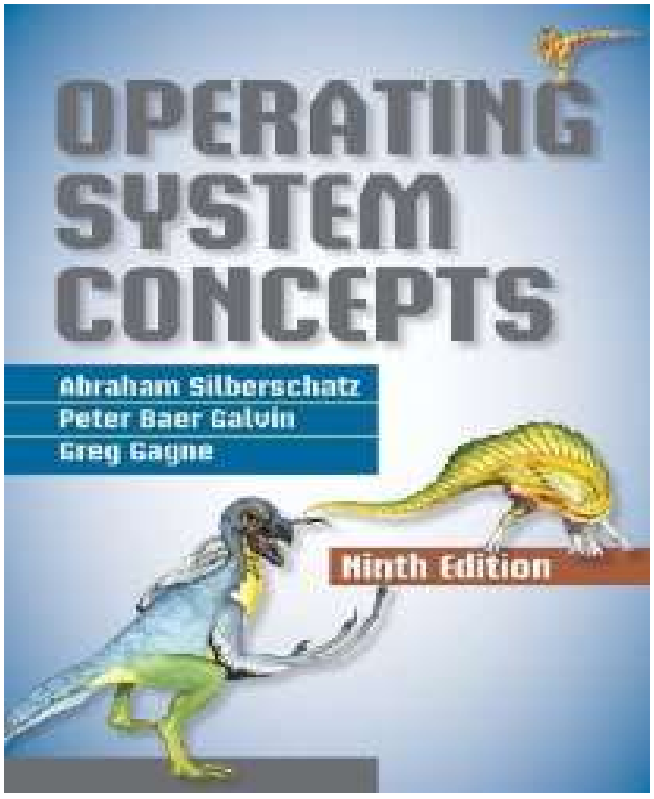
Email: **r.asif@nu.edu.pk**

Phone: (041) 111-128 -128 Ext: 172

Office: Room 127, 1st floor, Department of CS, FAST-NUCES.

Books and Resources

- **Operating System Concepts** 9th Edition by Silberschatz, Galvin and Gagne



Course outline

- Introduction
- Processes, Inter-process communication,
- Threads
- Synchronization, Semaphores,
- Deadlocks
- CPU Scheduling
- Memory management
- File-system management
- ……

Class Rules & Regulations

- **Quizzes**
  - All quizzes will be **Announced**
  - Quizzes will have more weightage as compared to Assignments
- **Assignments**
  - In case of copied, all remaining assignment will be marked **Zero**
  - Avoid late submission, **No** assignment will be entertained after deadline
  - Don't send me assignments by using email, submit them on Google Class
  - Assignments **Viva** will be conduct in the end of semester
- **Attendance**
  - Don't mark Proxy of your friend (If he/she is your real friend ☺)
    - In case of proxy, **5 Absents** will be marked to that student
  - In case of emergency, get permission personally or send an email prior to the class
  - Attend all classes regularly, no attendance issue will be entertained in the end of semester

Are you ready !!!!

Lets Begin

Operating Systems

- How is RAM managed?
- How do you find something on disk?
- How do you know where to load it in RAM?
- How do you keep track of all running programs?
- Answer
  - This is what the operating system does

What is an Operating System?

- 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

What is an Operating System?

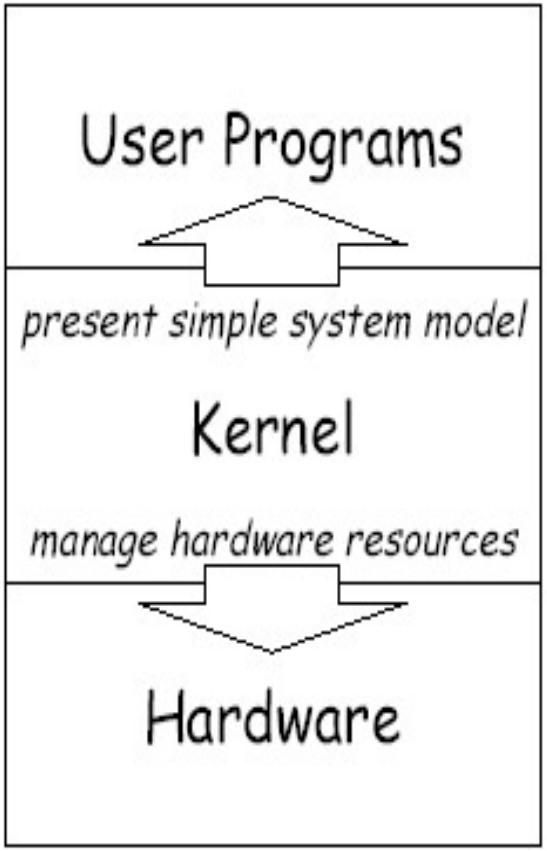
- It is a program!
- User's view
  - Provides an extended or virtual machine abstraction to user programs
    - Easier to program than the underlying hardware
    - All services are invoked and accomplished through system calls
- System view
  - Acts as a resource manager of a complex system
  - Resources consist of processors, memories, timers, disks, keyboard, network interfaces, printers etc.
  - OS manages allocation of these resources to user programs in an orderly and controlled manner

What is an Operating System?

- OS do nothing by themselves
- Similar to subroutine libraries, do nothing unless they are invoked by programs
- Act as an intermediary between users and the hardware

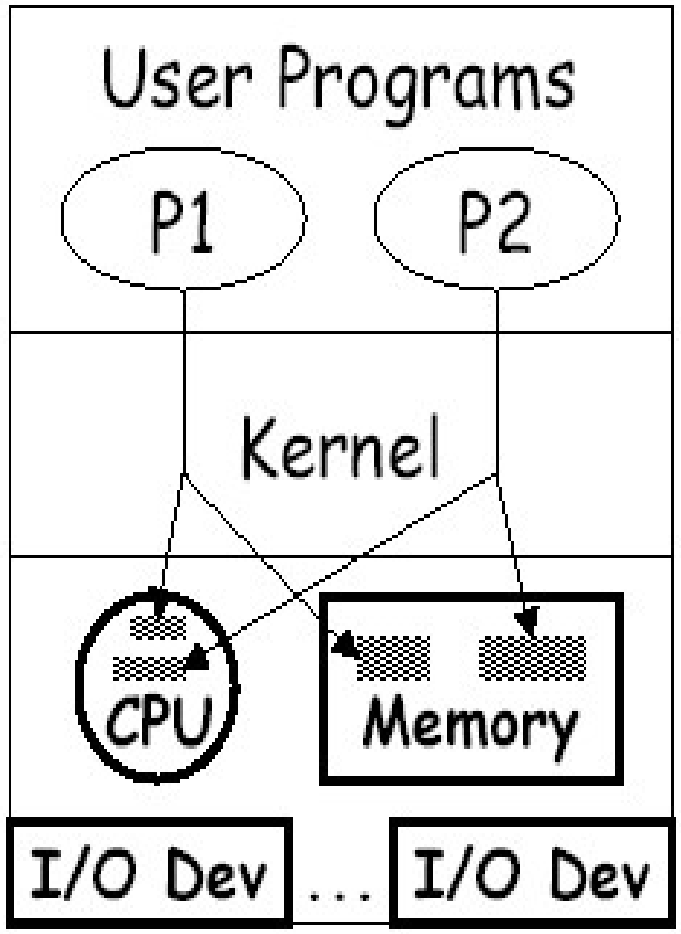
Two Goals of Operating Systems

- Manage hardware resources
  - System operates smoothly, efficiently, reliably and securely
- Present abstract system model to programmer
  - Simple and convenient access to and control of resources



Manager/Coordinator of Resources

- Coordinates who gets what
  - "who": running programs
  - "what": resources
  - "when": scheduling time
  - "where": organizing space
  - "whether": limits, rights
- Goal: smooth system operation
  - efficiency, reliability, security

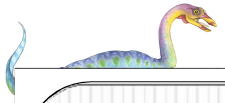


Resource multiplexing

- OS multiplexes resources in two ways:
- In time
  - Time multiplexing involves different programs taking turns in using the resource.
  - Example:
    - CPU scheduling
    - printer sharing
- In space
  - Space multiplexing involves different program getting part of the resource possibly at the same time
  - Example:
    - memory is divided into several running programs

What If No Operating System?

- All we have is bare hardware
- You want to run a program
  - How do you load it?
  - How do you run it?
  - What happens when it completes?
- Need at least some minimal OS to do these functions



…Lets go back to the “stone age” of computing…

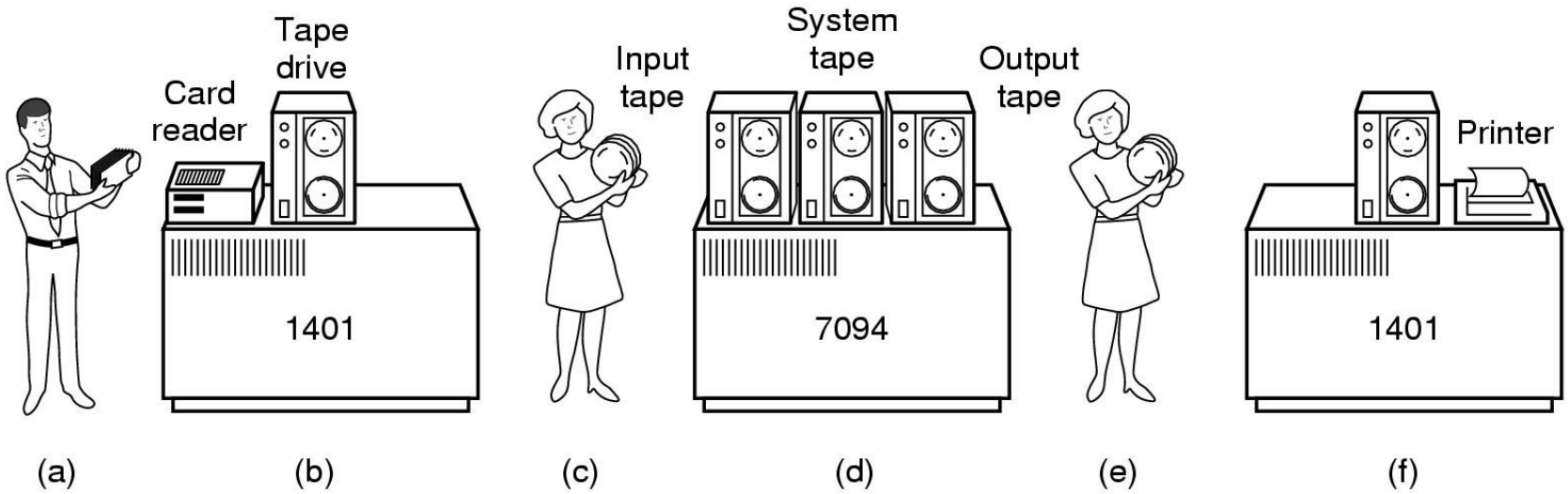
Early Systems

- No Operating systems,
- your Data and your Programs
  - X = X + 1;
- As
  - Load Register1, x
  - Load Direct Register2, 1
  - Add Register3, Register2, Register1
  - Store Register3, x
- Time slots allocated to scientists
- Enter programs using binary switches

Early Systems

- **Problem:**
  - computer remains idle while programmer sets things up
  - Poor utilization of huge investment
- **Solution:**
  - Hire a specialized person to do setup
- **Problem:**
  - Faster than programmer, but still a lot slower than the machine
- **Solution:**
  - Build a batch monitor

The Monitor



Early batch system

- bring cards to 1401
- read cards to tape
- put tape on 7094 which does computing
- put tape on 1401 which prints output

Simple Batch System

- The user submits a job (written on a card or tape) to a computer operator
- The computer operator place a batch of several jobs on a input device
- A special program, the monitor, manages the execution of each program in the batch
- Resident monitor is in the main memory and available for execution

The Resident Monitor

- Monitor reads job one at a time from the input device
- Monitor places a job in the user program area
- A monitor instruction branches to the start of the user program
- Execution of the user program continues until:
  - End of program occurs
  - Error occurs

Why study operating systems?

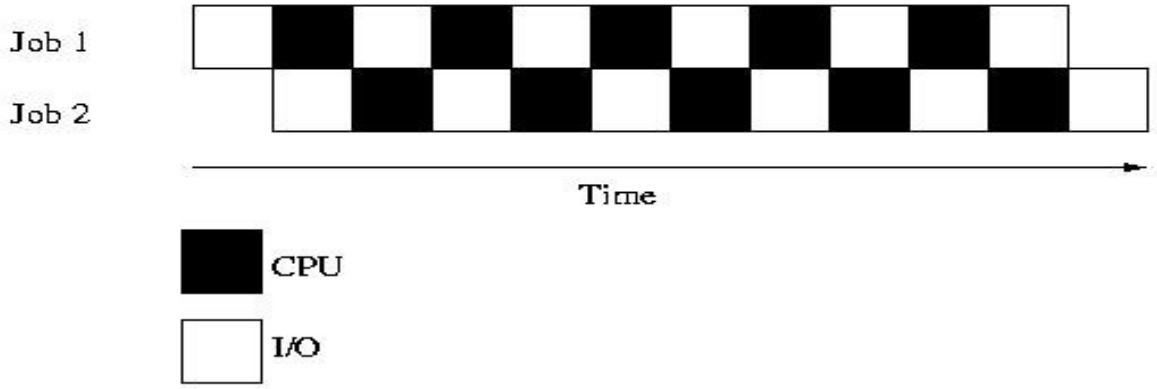
- Point of describing change isn't "Look how stupid batch processing is" – it was right for tradeoffs of the time, but not anymore
- Point is: have to change with changing technology
- Situation today is much like it was in the late 60's
- OS's today are enormous, complex things
  - 100k's of lines (or >1M lines)
  - Windows NT is 20M lines
  - Windows2000 and Windows XP are about 40M lines.
- Key aspect of this course, understand OS's so we can simplify them!

Beyond batch processing systems

- The Era of Timesharing and Multiprogramming
- Multiprogramming-Execute multiple jobs simultaneously
- Timesharing-Processor's time is shared among multiple users

Multiprogramming

- Execute multiple jobs simultaneously
- But a CPU can execute a single instruction at a time???
- One job can use the CPU while the other is waiting for I/O



- Small jobs not delayed by large jobs
- Context switching

Requirements for Multiprogramming

- Hardware support:
  - I/O interrupts and (possibly) DMA(Direct memory access)
    - OS requests I/O, goes back to computing, gets interrupt when I/O device finishes
- Memory management
  - several ready-to-run jobs must be kept in memory
  - Memory protection (data and programs)
- Software support from the OS:
  - Job Scheduling
    - Which ready jobs should be brought to memory
  - CPU Scheduling
    - Which program is to be run next

Multiprogramming

- What is the Performance criteria?

- Turn-Around Time
  - The length of time between the start of the job and when the output was done

Time Sharing System (TSS)

- Processor's time is shared among multiple users
- Use cheap terminals to let multiple users interact with the system at the same time
- OS does timesharing to give illusion of each user has own computer

Time Sharing System (TSS)

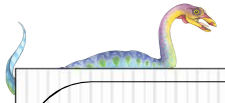
- Time sharing was achieved by resource sharing
  - e.g, the CPU among the jobs
- Performance measured in terms of Response Time
  - The length of time between the start of a job and the first output
- Utilization is still a problem. Why?
  - CPU time sacrificed to get better response time for users
  - CPU still has to wait, while a slower user is entering data on keyboard
- Other jobs must run during this time

Personal Computing

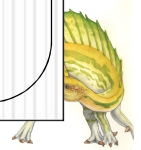
- Computers are cheap, so give everyone a computer
- Initially, OS became a subroutine library again (MSDOS)
- Since then, adding back in memory protection, multiprogramming, etc.
- Because when humans are expensive, don't waste their time by letting programs crash each other

Distributed Computing

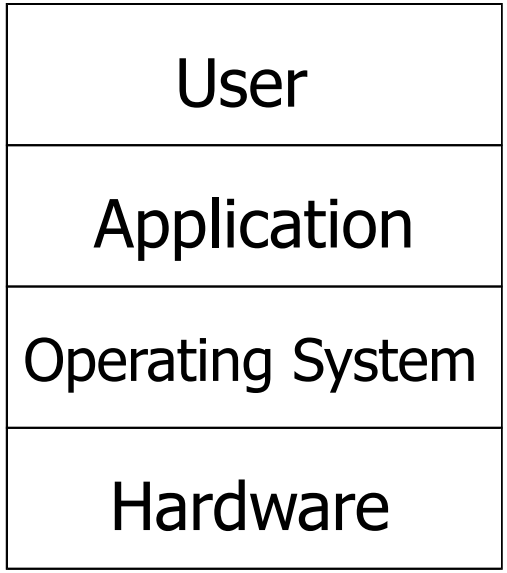
- Computers so cheap – give people a bunch of them
  - I have 2 PCs at home, 1 in my office, 3 smart phones and share some machines in a lab how do I coordinate a bunch of machines?
- Fast Networks – allow machines to share resources and data easily
- Cheap Networks – allow geographically distributed machines to interact

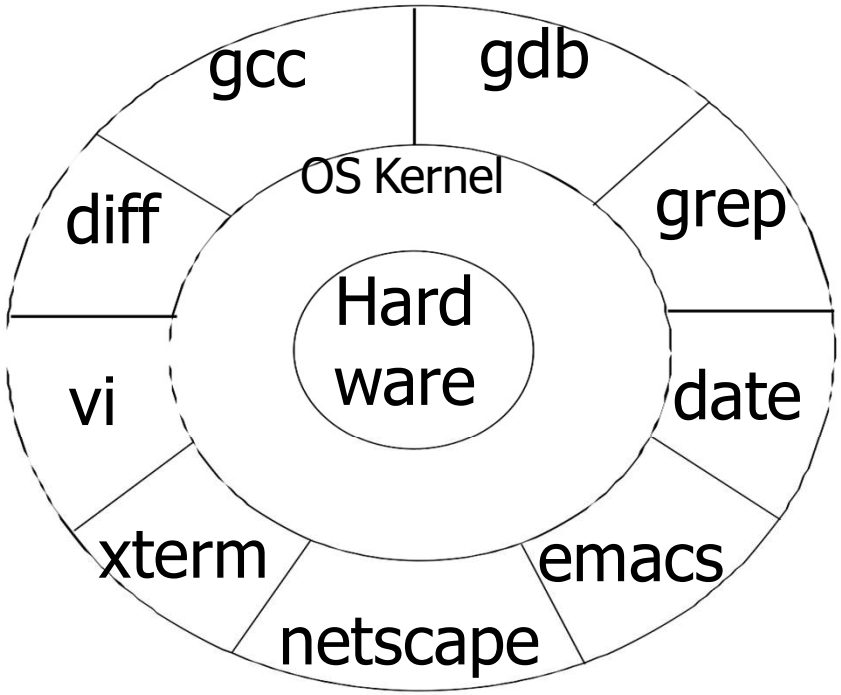


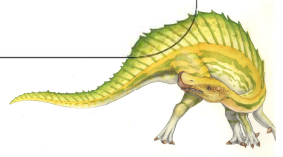
OS Overview



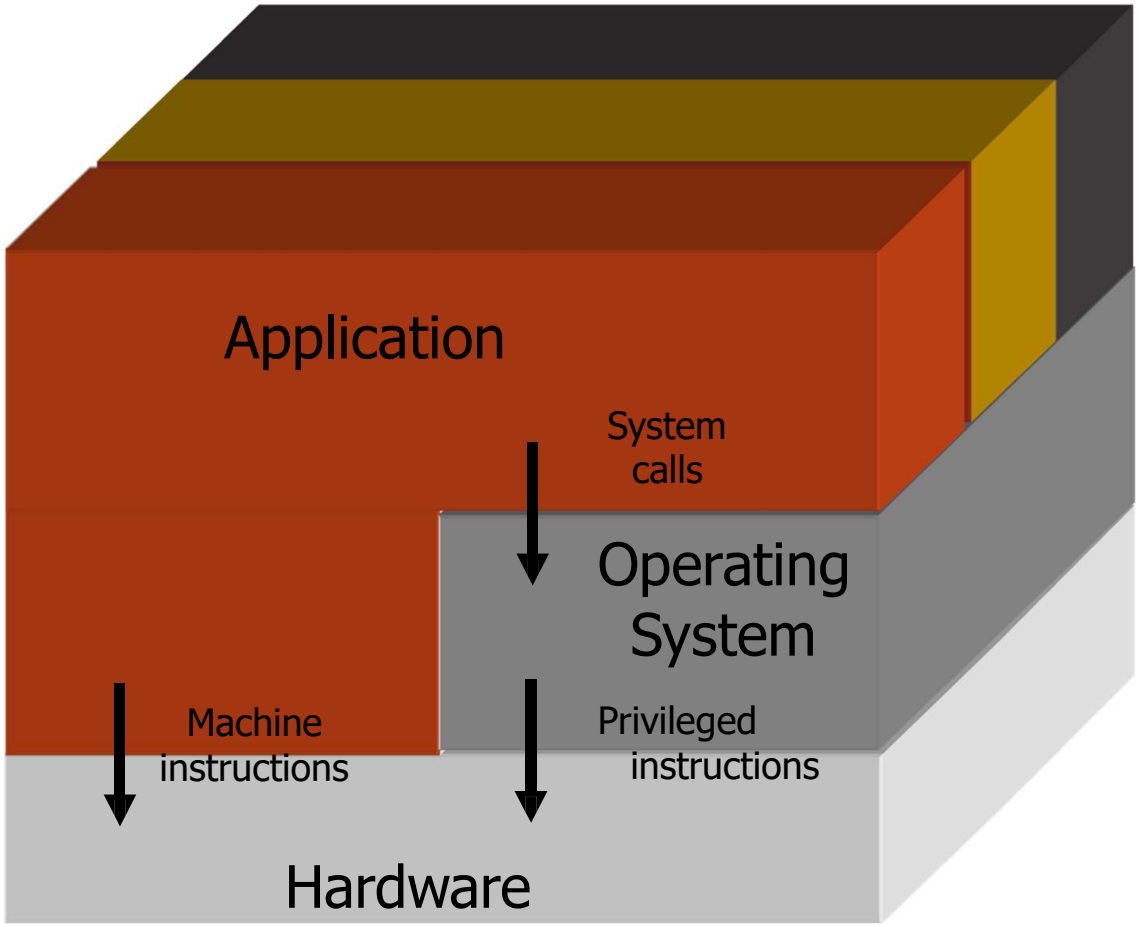
The Operating System controls the machine







A better picture



Many applications

One Operating System

One Hardware

Operating System in Action

- OS is a program, just like any other program
- When you turn power on, *bootstrap* program is loaded from ROM
- Bootstrap program
  - Examine/check machine configuration
    - # CPUs
    - How much memory
    - # and type of HW devices
  - Build configuration structure describing the HW
  - Locates and Loads the OS
  - The control transfers to the OS

Operating System in Action

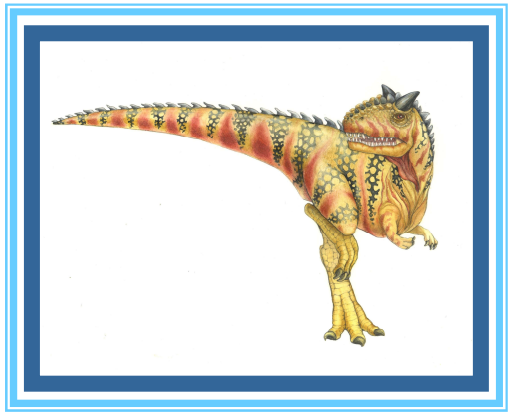
Operating System:

- Initialize kernel data structures
- Initialize state of HW devices
- Creates a number of processes to start operation

References

- Operating System Concepts 8/9th Ed Chapter 1
- Modern Operating Systems 3rd Ed Chapter 1
- Operating Systems 6th Ed Chapter 2
  - By William Stalling
- http://en.wikipedia.org/wiki/Time-sharing
- http://en.wikipedia.org/wiki/Operating_system
- http://en.wikipedia.org/wiki/Distributed_computing

End



Silberschatz, Galvin and Gagne ©2018