CS 343 - Operating Systems

Module-1A

Course Overview & Introduction to Operating Systems

Dr. John Jose

Associate Professor

Department of Computer Science & Engineering

Indian Institute of Technology Guwahati

Few Important Information

- Instructor: John Jose (Pre-Mid Sem) & T. Venkatesh (Post Mid-Sem)
  - Office Room: H-201, Second Floor, CSE dept
  - Personal webpage: http://www.iitg.ac.in/johnjose/
  - email: johnjose@iitg.ac.in: Phone: 0361-2583256
- Lead Teaching Assistant - Rajeswari Suance [Theory] & Amit Puri [Lab]
- Microsoft Teams [Slides] & Course WhatsApp group [Announcements]
- Lecture slots: C1 slot (Tue 3 PM, Wed 3 PM, Thu 3 PM) @ L4

Grading

- **Grading Scheme**
  - **Quiz-1 [27.08.2022] - 15%**
  - **Mid-Sem Exam [21.09.2022] - 35%**
  - **Quiz-2 [22.10.2022] - 15%**
  - **End Sem Exam [25.11.2022] - 35%**

There might be slight changes in the weightage in unavoidable cases.

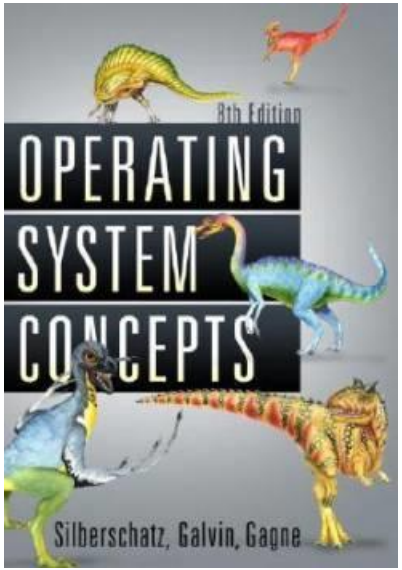
General Policies

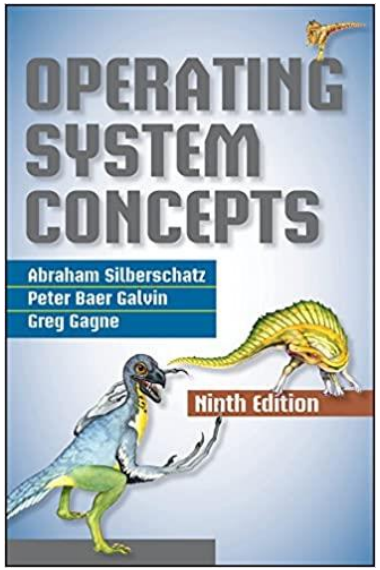
- 100% attendance is preferred. Once you miss the class you will lose the connectivity between topics
- Be on time in attending lecture class. Introductory 5 minutes is very important for the day's discussion.
- Academic dishonesty cannot be tolerated.
- I know everybody cannot score AA/AS. Do your best, Be sincere, Be open.
- It is not the marks but the effort that matters.
- I promise that you will enjoy this course.

Reference Books

Operating System Concepts (6th to 9th edition)

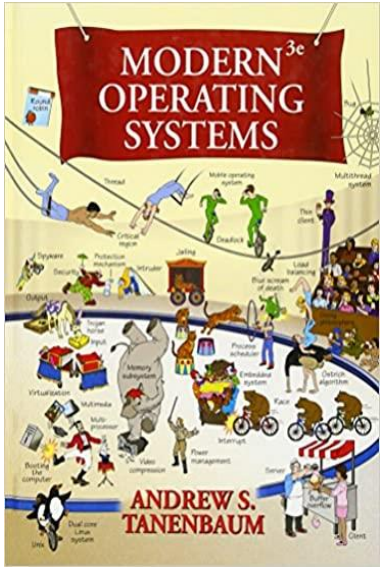
Abraham Silberschatz, Peter Baer Galvin, Greg Gagne,

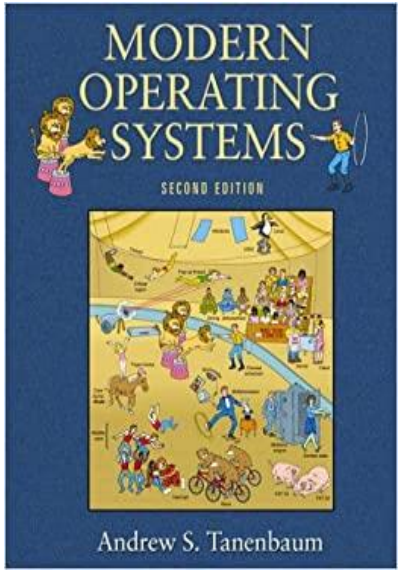




Modern Operating Systems (2nd / 3rd edition)

Andrew S. Tanenbaum,





Syllabus

- Week-1: Elementary computer architecture and introduction to operating systems. Types of OS, abstract view of OS and its functional structure.

- Week 2: Process management, process states, CPU scheduling, scheduling criteria and scheduling algorithms, Process vs threads

- Week 3: Operations on processes, inter process communication, process synchronization - critical sections, semaphores, monitors

- Week 4: Classical synchronization problems, deadlock characterization, prevention, avoidance, detection and recovery techniques.

- Week 5: Introduction to memory management, partitions & allocation technique, free space management, address mapping, segmentation and paging

- Week 6: Virtual memory concepts, page replacement strategies, working set schemes, frame allocation techniques and thrashing.

Syllabus

- Week 7: Storage Management: Hard disk structure, disk management, swap space management, disk scheduling, RAID structure.

- Week 8: File management; access and control methods, directory structure, file system structure, file system and directory implementation. Allocation methods and free space management.

- Week 9: I/O subsystem, structure and organization, polled vs interrupt-driven I/O, DMA. Classification of I/O devices, buffering, caching, scheduling, spooling.

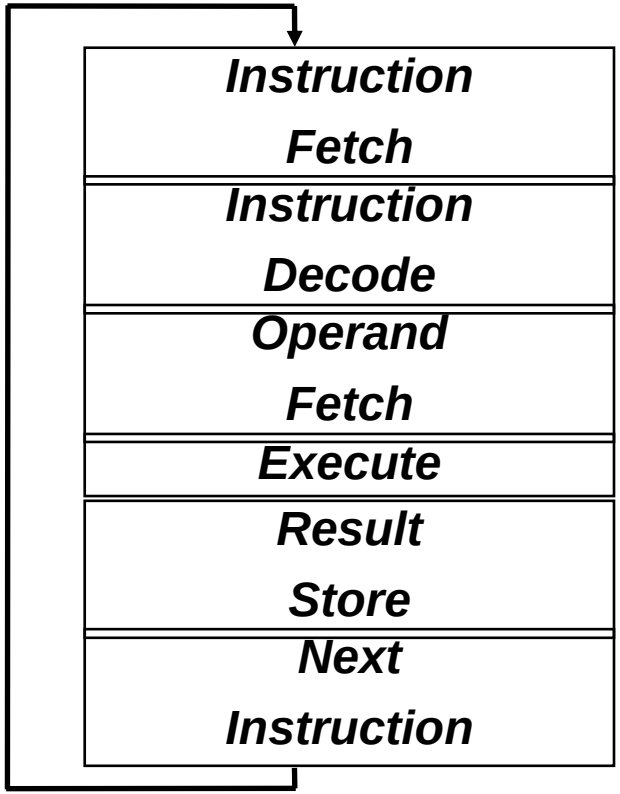
- Week 10: Protection; design principles, authentication schemes, access matrix, ACLs and capabilities, covert channels. Security and user authentication, system and network threats, security defenses and firewalls.

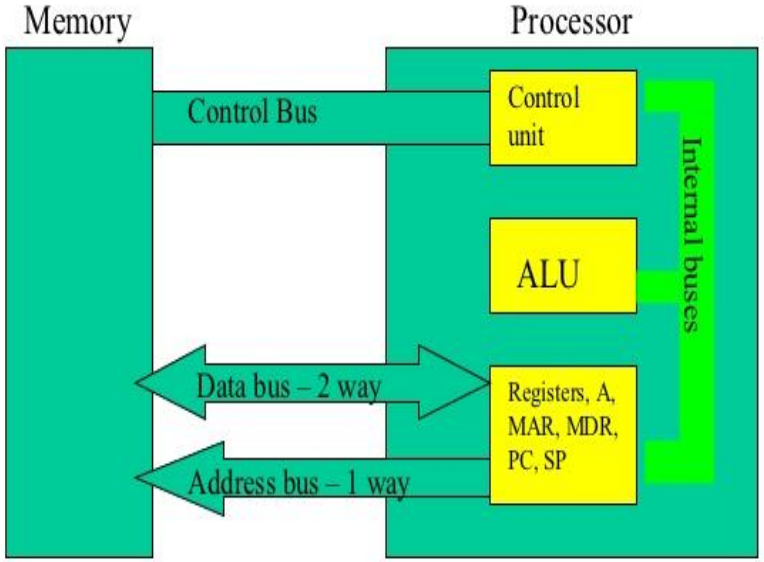
- Week 11 & 12: Introduction to distributed operating systems, design issues, distributed file systems, distributes synchronization.

Session Outline

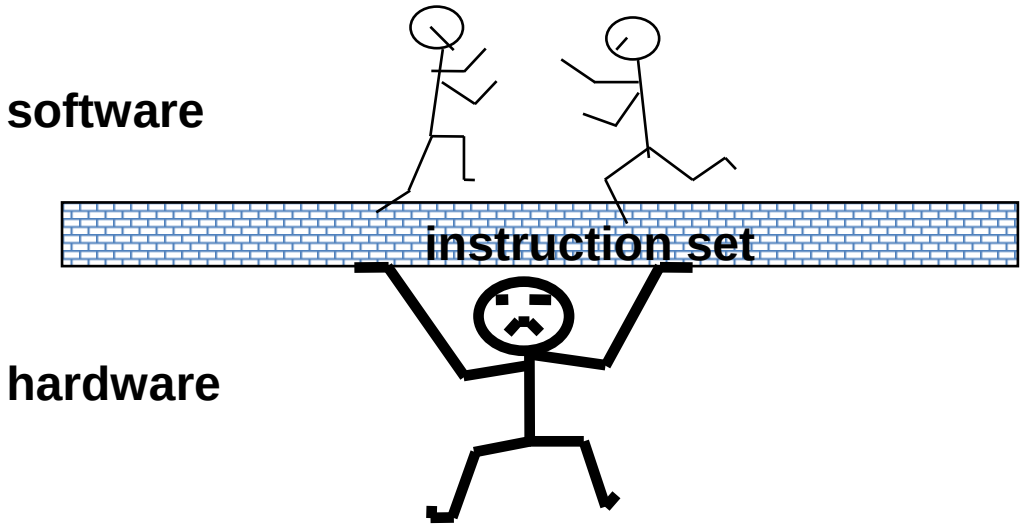
- Review of processor – memory interaction
- Instruction Set and Addressing Modes
- Storage Hierarchy – Cache, Main Memory, Disks
- Introduction to operating systems
- Functions of operating systems
- Elementary concepts in interrupts

Processor Memory Interaction





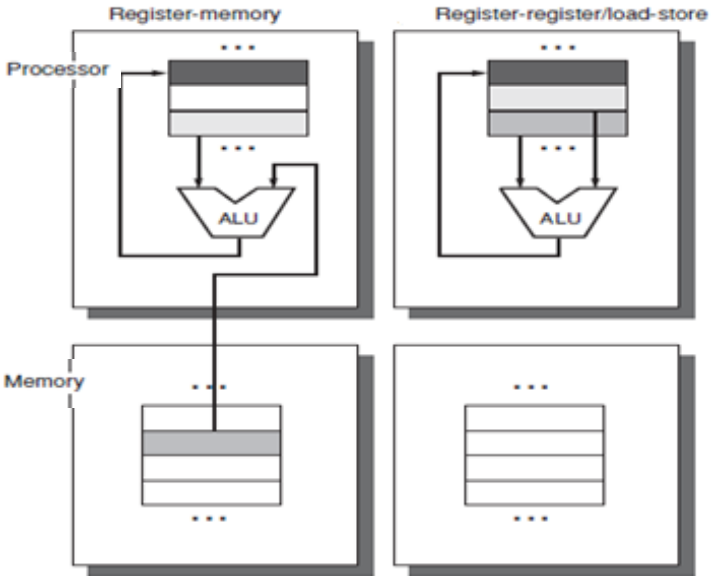
Instructions: Language of the Computer



❖Portion of the machine that is visible to the programmer or the compiler writer.

Instruction Set Architecture

- Instruction vs Program vs Software
- Opcode, Operand
- Classification of instructions
  - Arithmetic and Logical Operations
  - Data Movement Operations
  - Program Control Operations

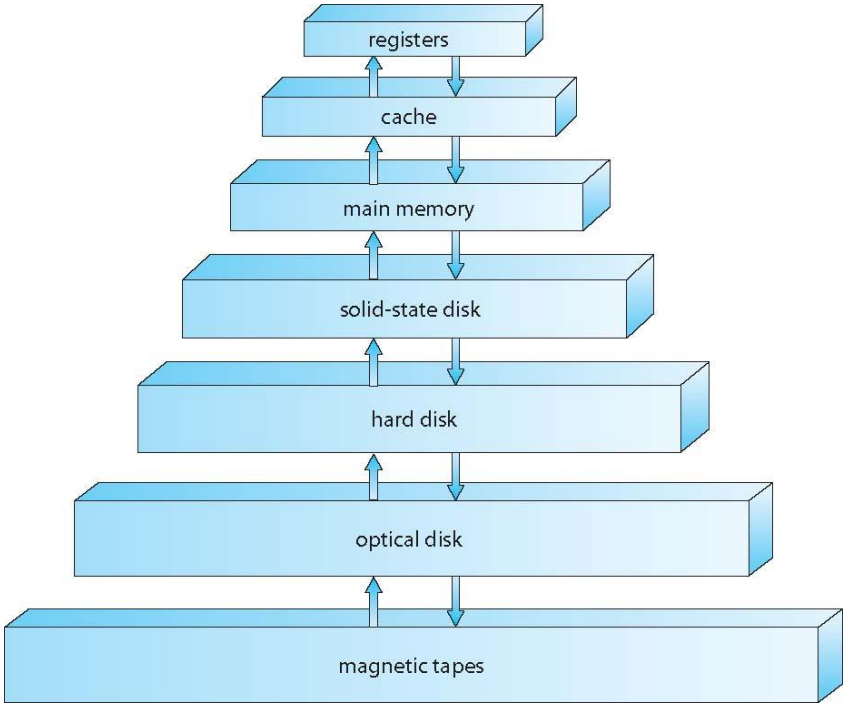


Addressing Modes

- The way by which an operand is specified in an instruction.

Register	`add r1, r2`	`r1 <- r1+r2`
Immediate	`add r1, #5`	`r1 <- r1+5`
Direct	`add r1, (0x200)`	`r1 <- r1+M[0x200]`
Register indirect	`add r1, (r2)`	`r1 <- r1+M[r2]`
Displacement	`add r1, 100(r2)`	`r1 <- r1+M[r2+100]`
Indexed	`add r1, (r2+r3)`	`r1 <- r1+M[r2+r3]`
Scaled	`add r1, (r2+r3*4)`	`r1 <- r1+M[r2+r3*4]`
Memory indirect	`add r1, @(r2)`	`r1 <- r1+M[M[r2]]`
Auto-increment	`add r1, (r2)+`	`r1 <- r1+M[r2], r2++`
Auto-decrement	`add r1, -(r2)`	`r2--, r1 <- r1+M[r2]`

Storage Hierarchy

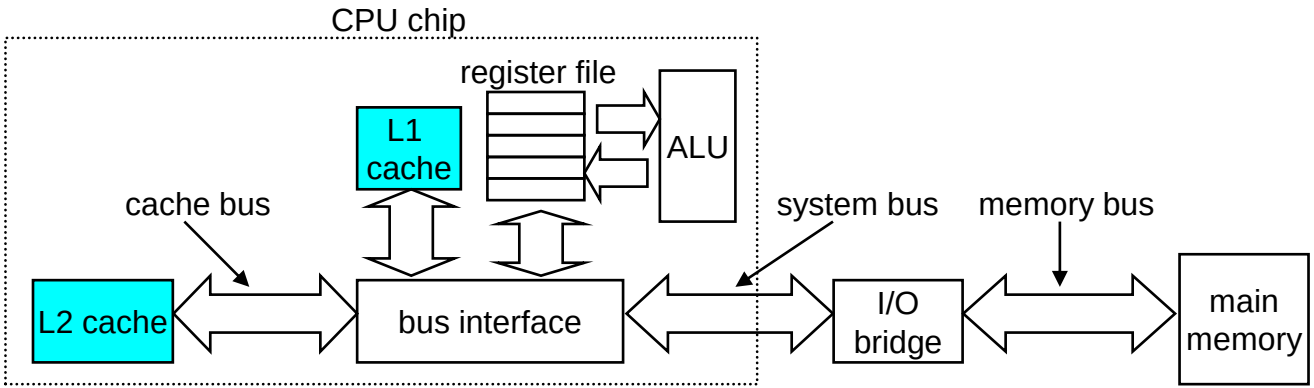


Cache Memory

- Cache is a small buffer between processor and memory
- Old values will be removed from cache to make space for new values
- **Principle of Locality :** Programs access a relatively small portion of their address space at any instant of time
- **Temporal Locality :** If an item is referenced, it will tend to be referenced again soon
- **Spatial Locality :** If an item is referenced, items whose addresses are close by will tend to be referenced soon

Cache Memory

- Cache memories are small, fast SRAM-based memories managed in hardware by cache controller.
- It hold frequently accessed blocks of main memory
- CPU looks first for data in L1, then in L2, then in main memory.

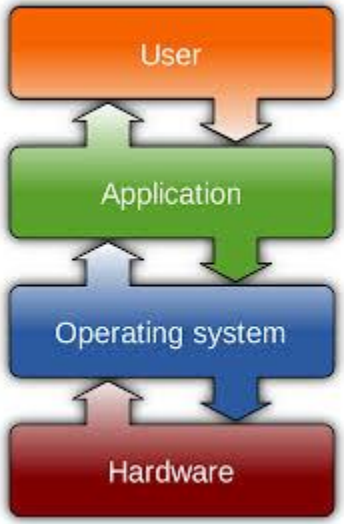


Storage Structure

- **Main memory** –large storage that the CPU can access directly
  - Random access and is typically volatile
- **Secondary storage** – extension of main memory that provides large nonvolatile storage capacity
  - Hard disks- platters covered with magnetic recording material
  - Disk surface is logically divided into tracks, which are subdivided into sectors
  - Solid-state disks – faster than hard disks, nonvolatile

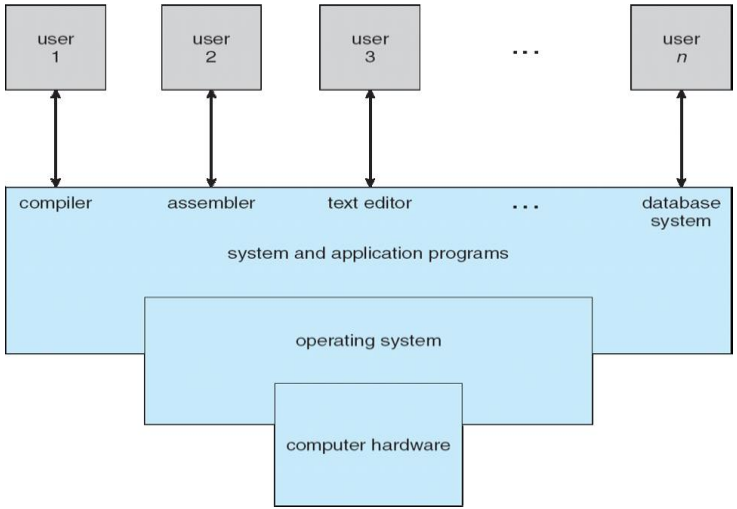
What is an Operating System?

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



Computer System Structure

- Computer system can be divided into four components:
  - **Hardware** -- CPU, memory, I/O devices
  - **Operating system** -- Controls and coordinates hardware/software
  - **Application programs** -- Word processors, compilers, web browsers, database systems, video games, apps
  - **Users** – People or devices



Operating System Definition

- OS is a **resource allocator**
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use
- OS is a **control program**
  - Controls execution of programs to prevent errors and improper use of the computer
  - The one program running at all times on the computer RAM is the **kernel of the OS**.

Computer-System Operation

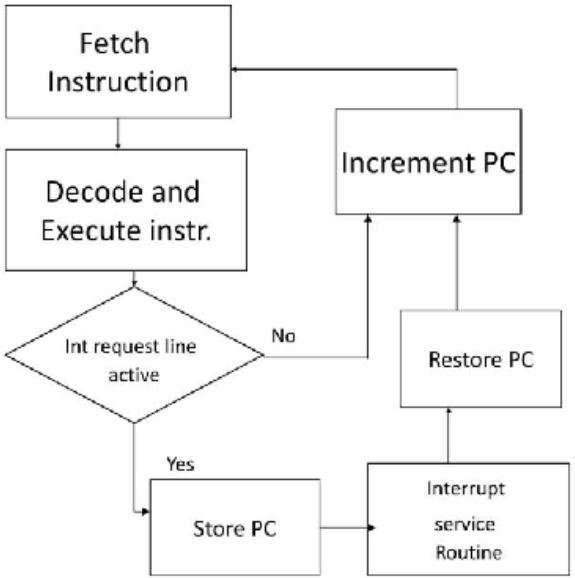
- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
- Each device controller has a local buffer
- CPU moves data from/to main memory to/from local buffers
- Addressing depends upon memory mapped I/O vs I/O mapped I/O
- I/O operation is from the device to local buffer of controller
- Device controller informs CPU that it has finished its operation by causing an interrupt
- An operating system is **interrupt driven**

Common Functions of Interrupts

- Interrupt is an externally initiated signal to catch the attention of a processor.
- Upon an interrupt, processor may temporarily suspend the current task and run another task to service the interrupt.
- Interrupt transfers control to the interrupt service routine generally, through the interrupt vector, which contains the addresses of all the service routines
- Interrupt architecture must save the address of the interrupted instruction

Interrupt Handling

- The operating system preserves the state of the CPU by storing registers and the program counter
- Determines which type of interrupt has occurred:
  - **Polling interrupt system**
  - **vectored interrupt system**
- Separate segments of code determine what action should be taken for each type of interrupt – Interrupt Service Routine (ISR)



johnjose@iitg.ac.in

http://www.iitg.ac.in/johnjose/