CS343 - Operating Systems

Module-1A

Elementary computer organization & Introduction to operating systems

Dr. John Jose

Assistant Professor

Department of Computer Science & Engineering

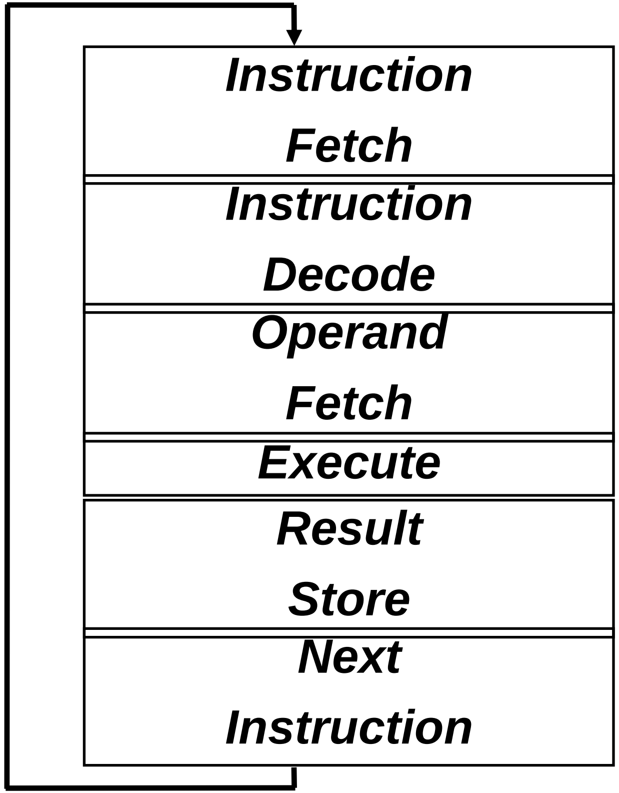
Indian Institute of Technology Guwahati, Assam.

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

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



Instruction Fetch

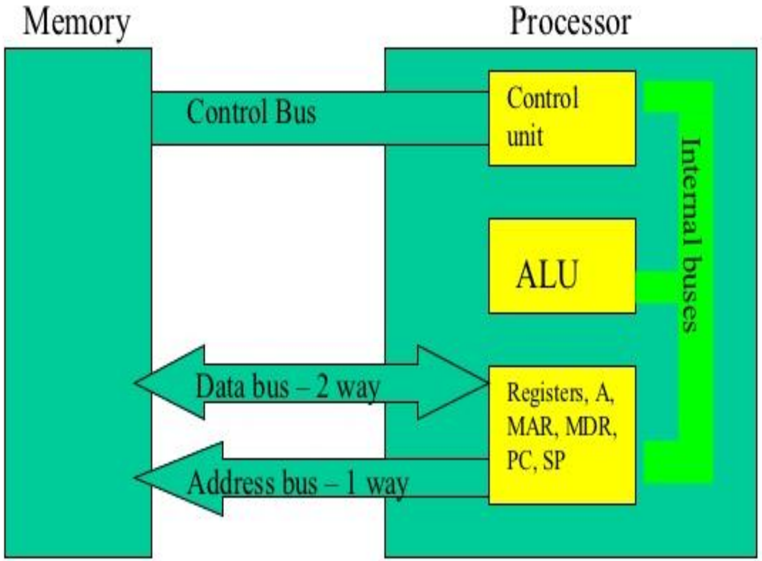
Instruction Decode

Operand Fetch

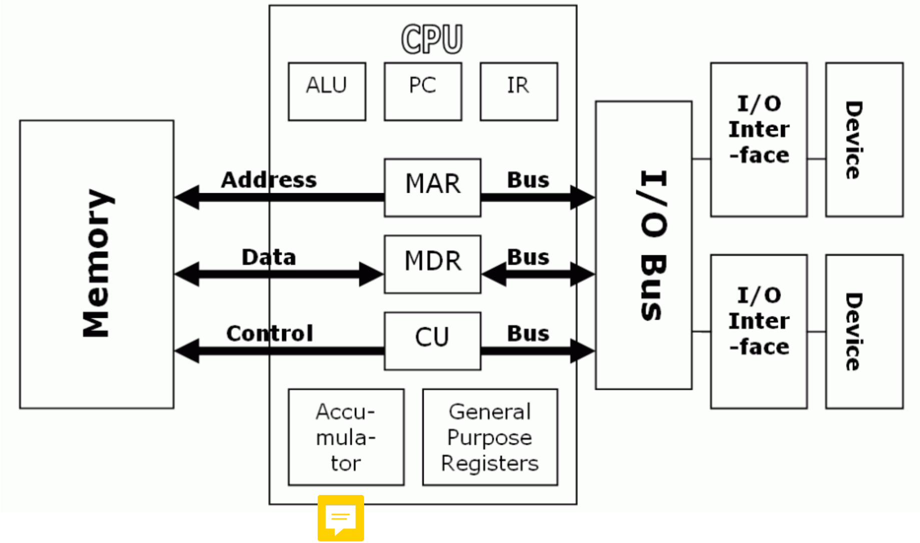
Execute

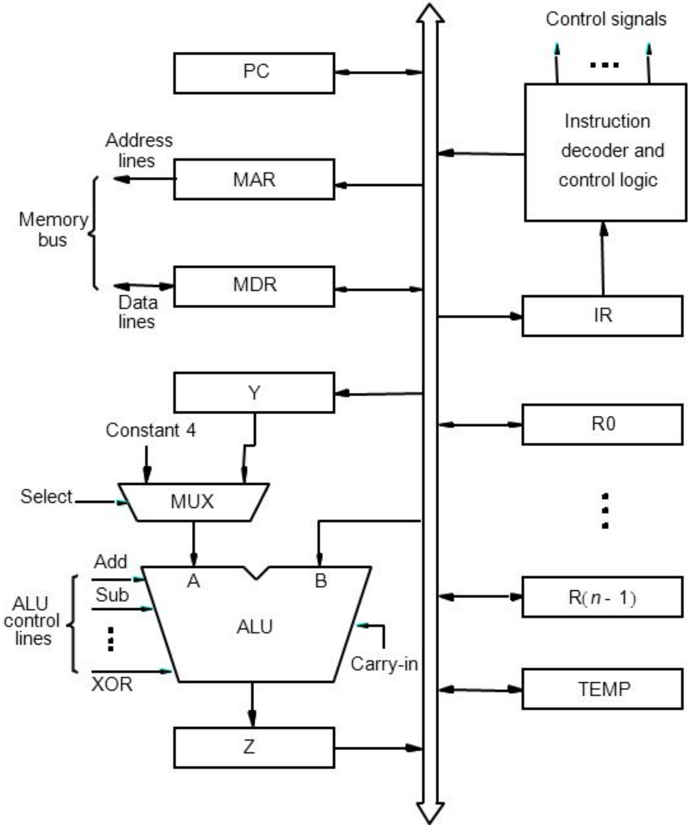
Result Store

Next Instruction

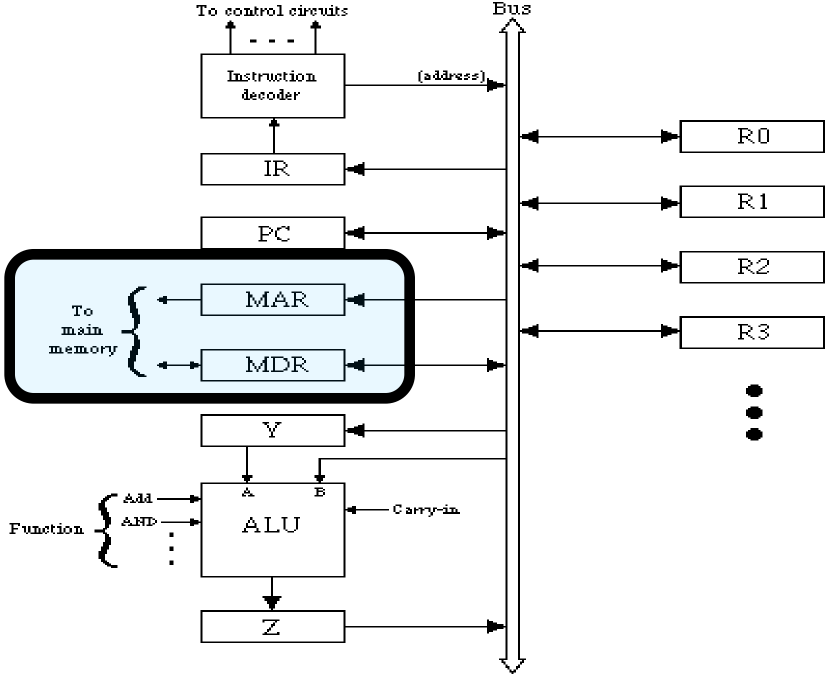


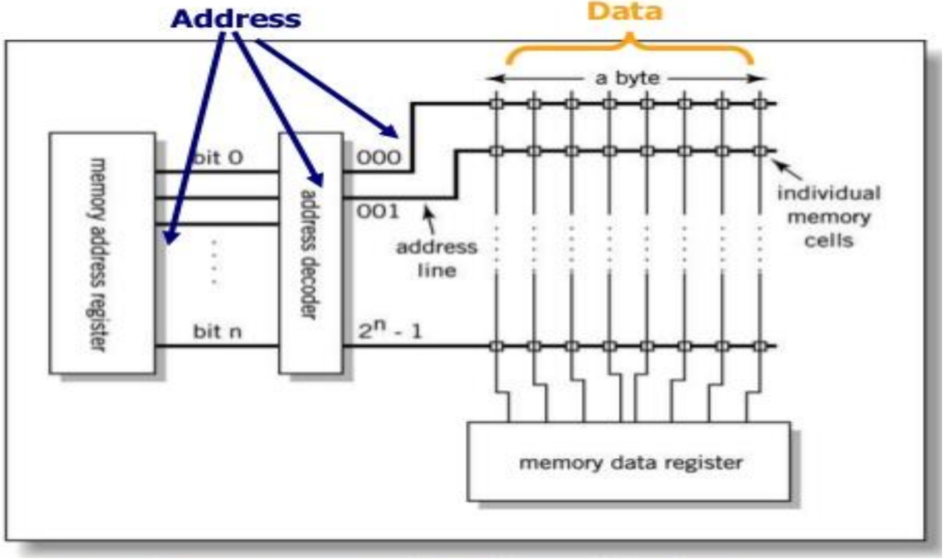
Processor Memory Interaction



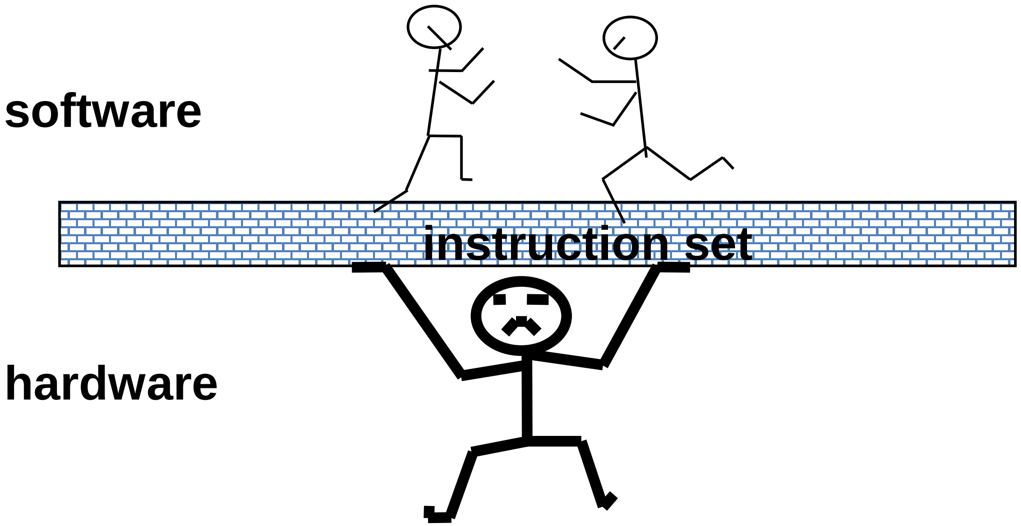


Processor Memory Interaction





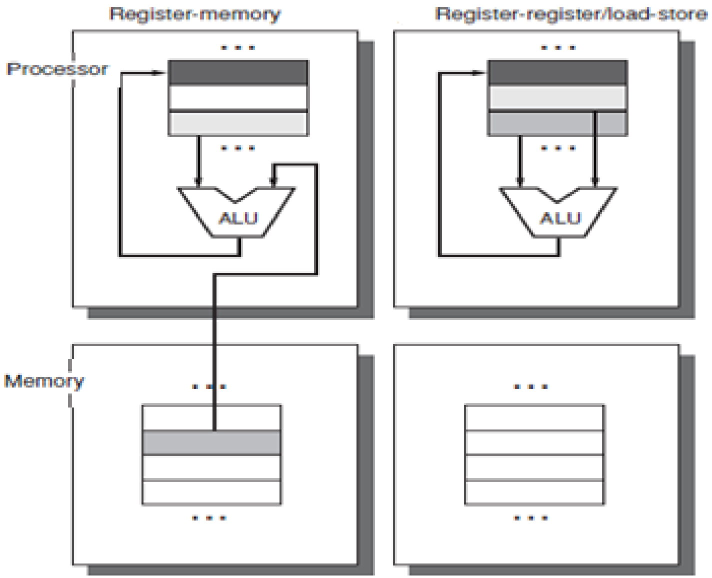
Instructions: Language of the Computer



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

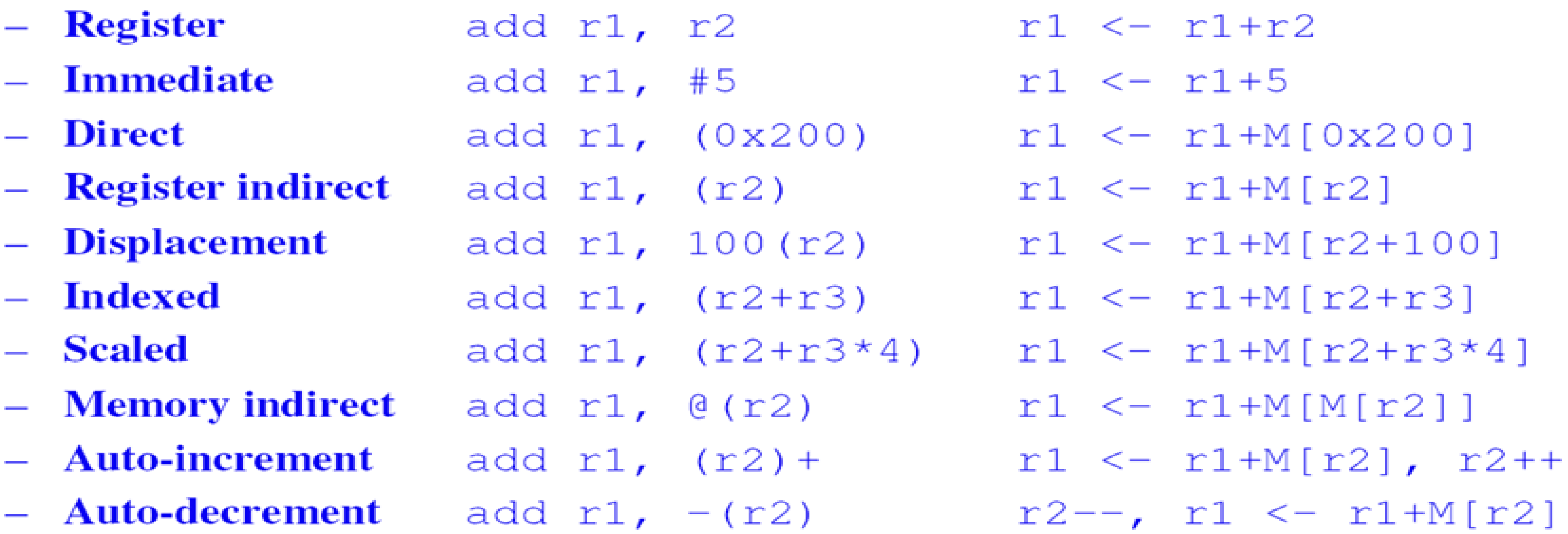
Instruction Set Architecture

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



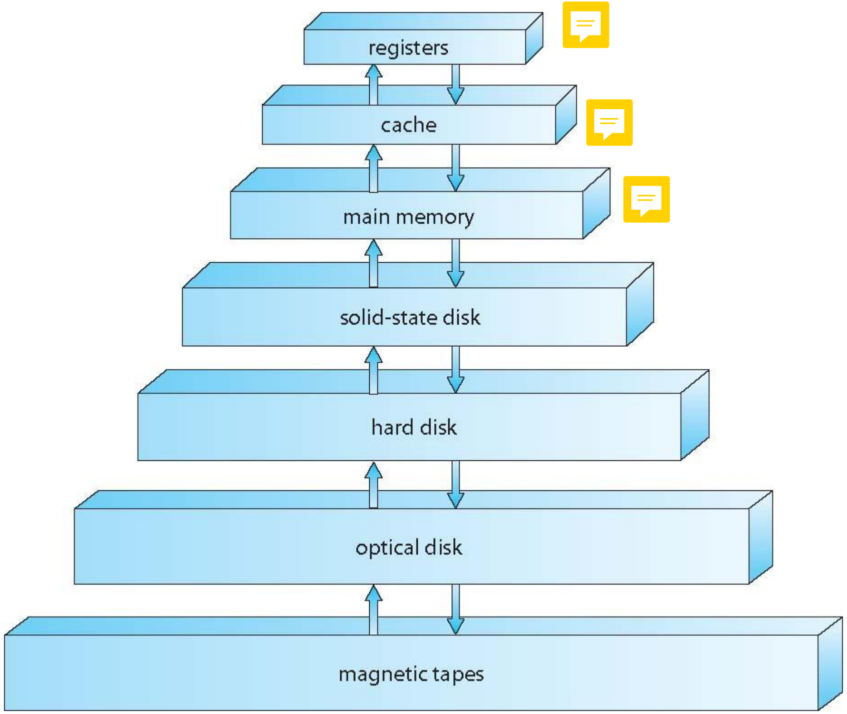
Addressing Modes

- 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

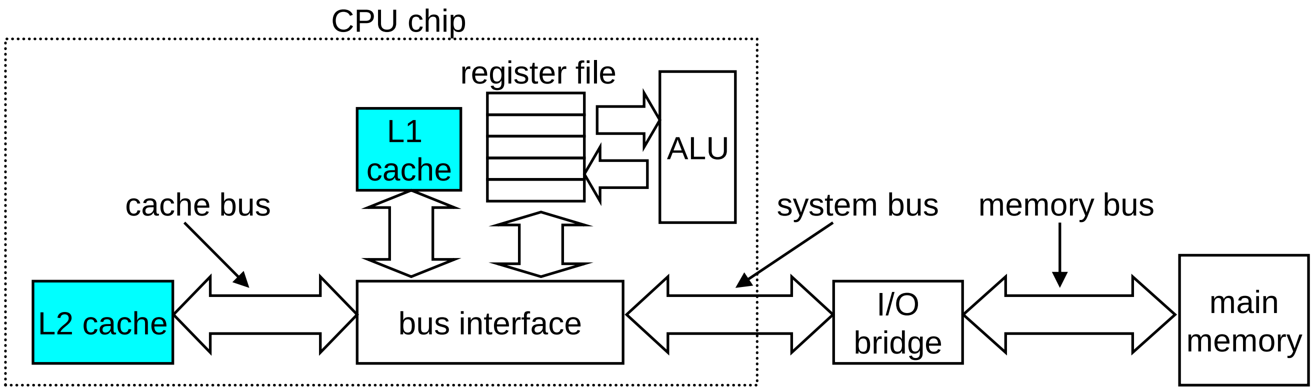


Cache Memory

- 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.

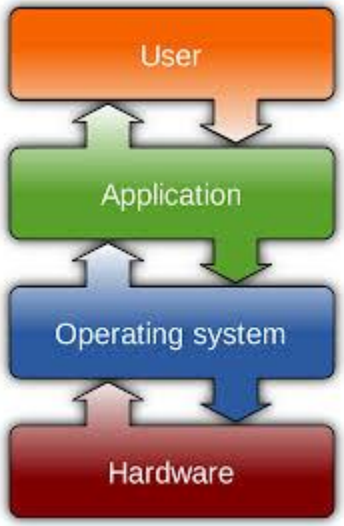


Storage Structure

- **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

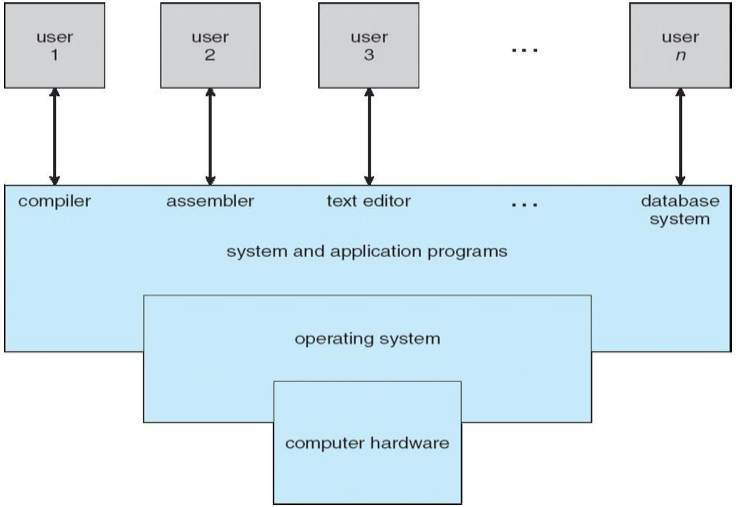
What is an Operating System?

- 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

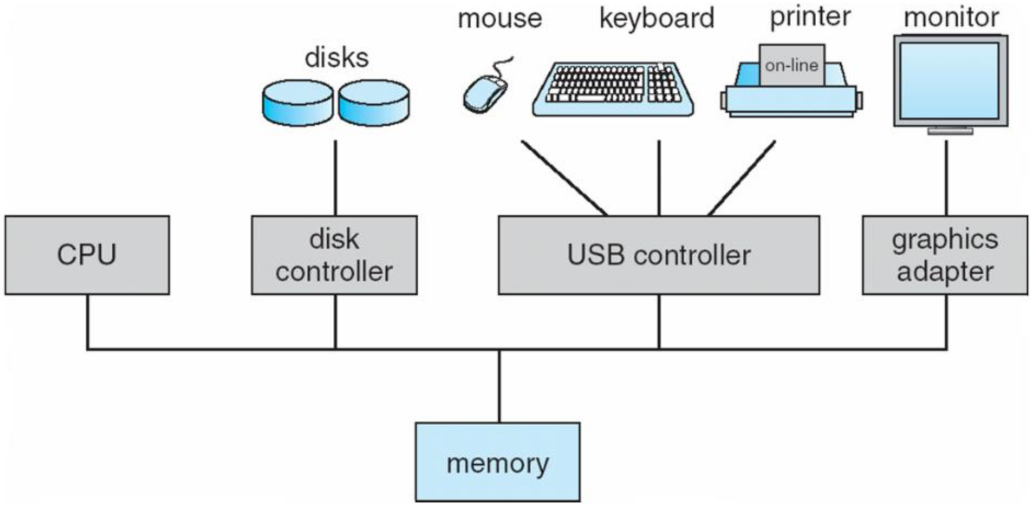


Operating System Definition

- 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 Organization

- Computer-system operation
  - One or more CPUs, device controllers connect through common bus providing access to shared memory
  - Concurrent execution of CPUs and devices competing for memory cycles



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

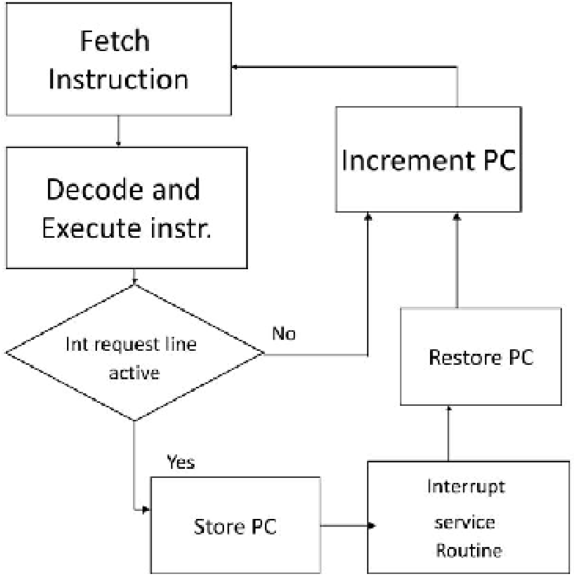
- Interrupt is an externally initiated signal to catch the attention of a processor.
- Upon an interrupt, processor may temporarily suspend the current task and run another task to service the interrupt.
- Interrupt transfers control to the interrupt service routine generally, through the interrupt vector, 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)



Thank you

johnjose@iitg.ac.in

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

CS343 - Operating Systems

Module-1B

Types of Operating Systems

Dr. John Jose

Assistant Professor

Department of Computer Science & Engineering

Indian Institute of Technology Guwahati, Assam.

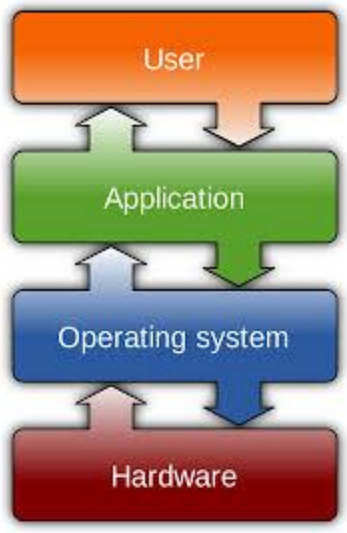
http://www.iitg.ac.in/johnjose/

Session Outline

- Review of basic operating system concepts
- Desktop PCs
- Parallel Systems
- Multiprogramming Systems
- Clustered Systems
- Real-time Systems
- Embedded Systems

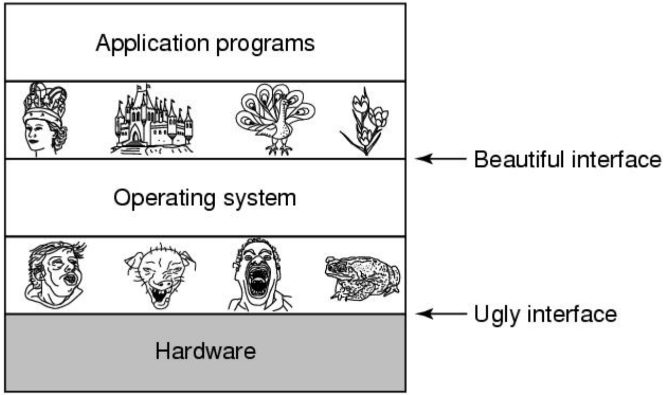
What is an Operating System?

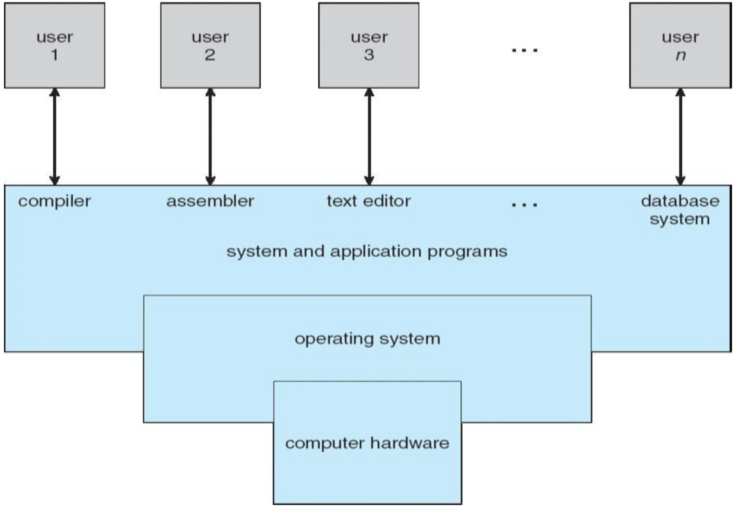
- A program that acts as an intermediary between a user of a computer and the computer hardware
- Operating system goals:
  - Execute user programs on hardware
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner



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**.

Types of Operating Systems

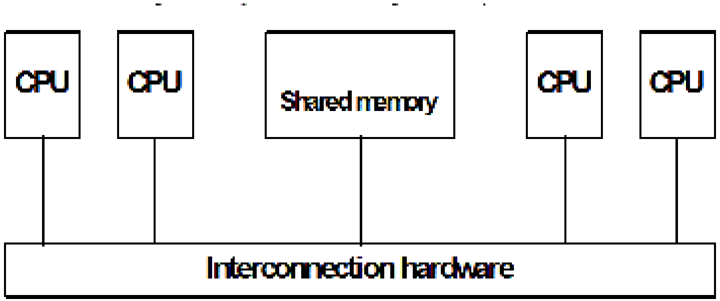
- Most systems use a single general-purpose/special purpose processor
- There are several architectures which all require a different OS:
  - Desktop PCs
  - Parallel Systems
  - Distributed Systems
  - Clustered Systems
  - Real-time Systems
  - Embedded Systems

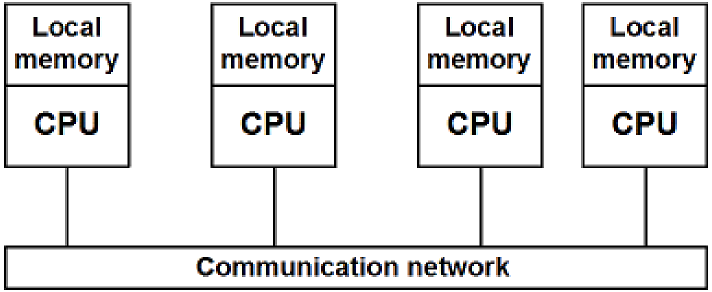
Desktop PCs

- Personal Computers – computer system dedicated to a single user.
- I/O devices – keyboards, mice, display screens, small printers.
- User convenience and responsiveness.
- Mostly single user - do not need advanced CPU utilization or protection features.
- May run several different types of operating systems (Windows, MacOS, UNIX, Linux)

Parallel Systems

- Multiprocessor systems with more than one CPU in close communication.
- **Tightly coupled system** – processors share memory and the internal clock; communication usually takes place through the shared memory.
- **Loosely coupled system** – multiple processors/computers with its own memory connected together for efficiency and throughput.



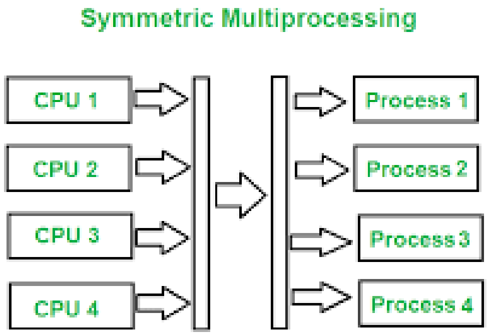


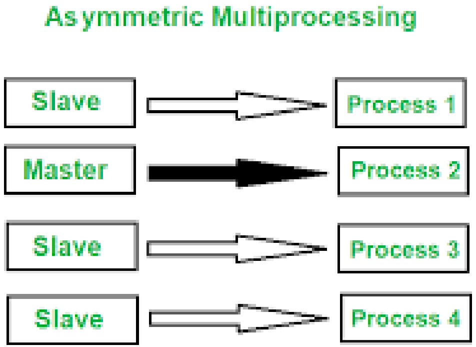
Multiprocessor Systems

- Most systems use a single general-purpose/special purpose processor
- **Multiprocessors** systems growing in use and importance
  - **Increased throughput**
  - **Economy of scale**
  - **Increased reliability** – graceful degradation or fault tolerance
  - **Asymmetric Multiprocessing**
  - **Symmetric Multiprocessing**

Multiprocessor Systems

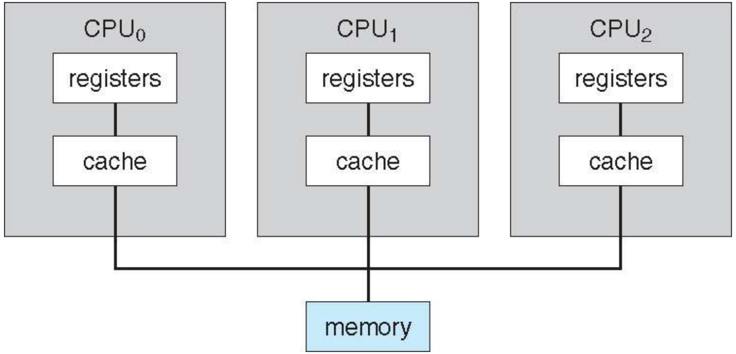
- **Asymmetric Multiprocessing**
- Each processor is assigned a specific task; master processor schedules and farms work to slave processors.
- More common in extremely large systems like mainframes with hundreds of processors.





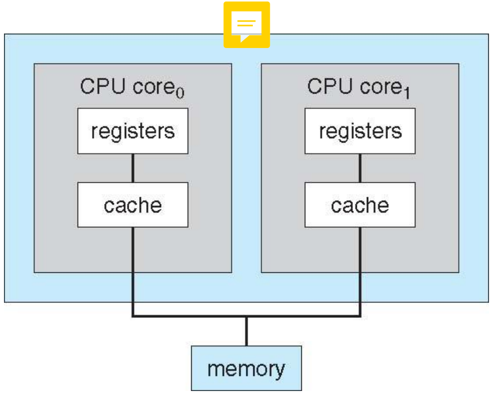
Multiprocessor Systems

- **Symmetric Multiprocessing**
- Each processor runs an identical copy of the operating system.
- The OS code is usually shared.
- Many processes can run at once without performance deterioration.
- Most modern operating systems have SMP support.
- OS has to cater for protection of data.



Multicore Design

- Multi-chip and multicore is a special case of tightly coupled parallel system.
- A single chip containing multiple separate systems

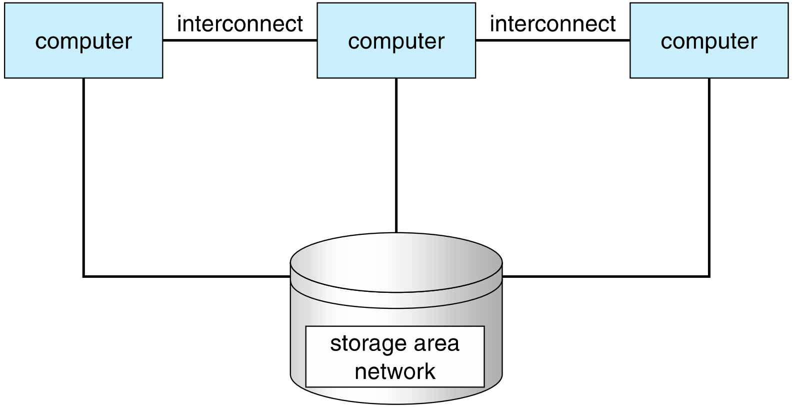


Clustered Systems

- Like multiprocessor systems working together
  - Usually sharing storage via a **storage-area network (SAN)**
  - Provides a **high-availability** service which survives failures
    - **Asymmetric clustering** has one machine in hot-standby mode
    - **Symmetric clustering** has multiple nodes running applications, monitoring each other

Clustered Systems

- Some clusters are for **high-performance computing (HPC)**
  - Applications must be written to use **parallelization**
- Some have **distributed lock manager** (**DLM**) to avoid conflicting operations



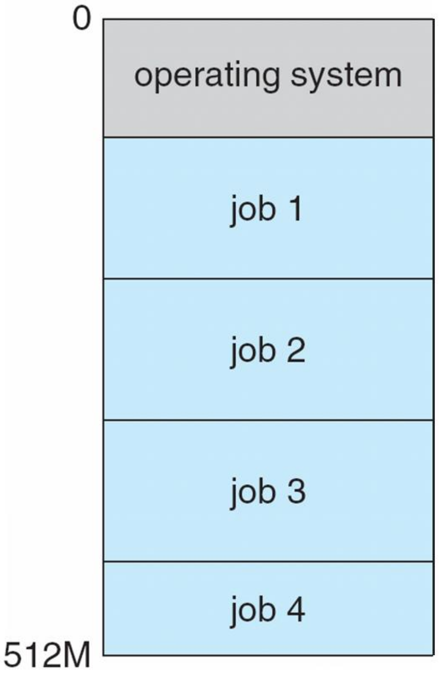
Multiprogramming Systems

- **Multiprogramming** (**Batch system**) needed for efficiency
  - Single user cannot keep CPU and I/O devices busy at all times
  - Multiprogramming organizes jobs (code and data) so CPU always has one to execute
  - A subset of total jobs in system is kept in memory
  - One job selected and run via **job scheduling**
  - When it has to wait (for I/O), OS switches to another job

Timesharing Systems

- **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
  - Each user has at least one program executing in memory
  - 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

Memory Layout for Multiprogrammed System



Real-Time Systems

- Often used as a control device in a dedicated application such as controlling scientific experiments, medical imaging systems, industrial control systems, and some display systems.
- Well-defined fixed-time constraints.
- Real-Time systems may have either hard or soft real-time.

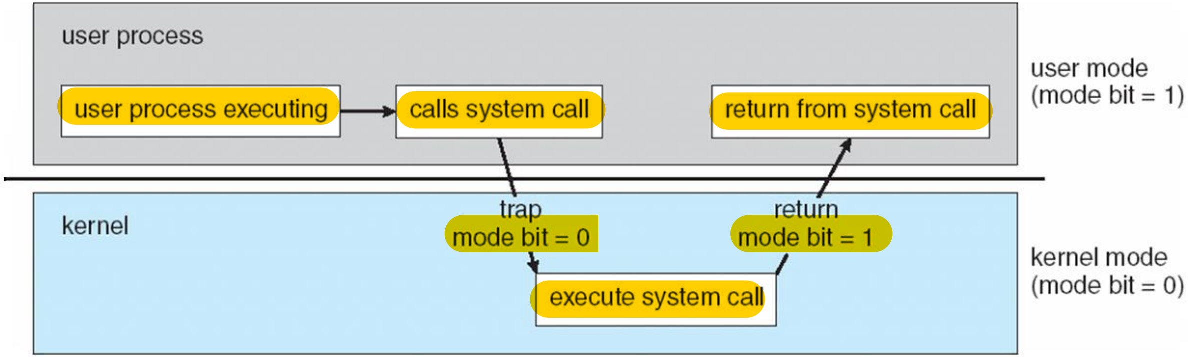
Embedded Systems

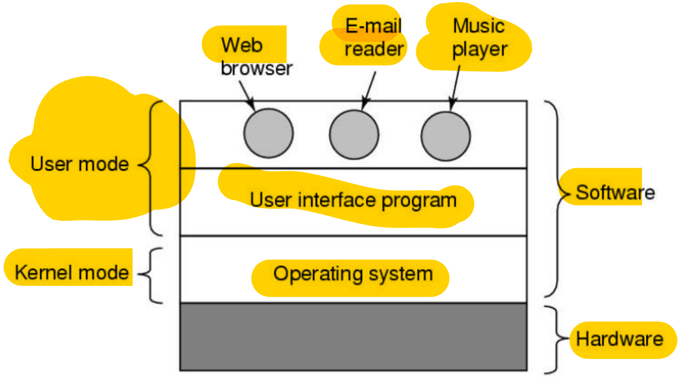
- Personal Digital Assistants (PDAs)
- Smart telephones
- Issues:
    - Limited memory, Slow processors, Small display screens.
    - Emphasis is on I/O operations.
    - Limited memory management and protection

Operating System in Dual Mode

- **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
  - System call changes mode to kernel, return from call resets it to user

Operating System in Dual Mode





Thank you

johnjose@iitg.ac.in

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

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी • Indian Institute of Technology Guwahati •

CS343 - Operating Systems

Module-1C

Operating System Services

Dr. John Jose

Assistant Professor

Department of Computer Science & Engineering

Indian Institute of Technology Guwahati, Assam.

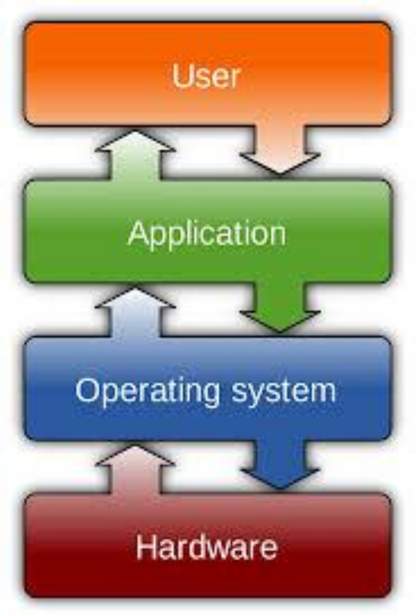
http://www.iitg.ac.in/johnjose/

Session Outline

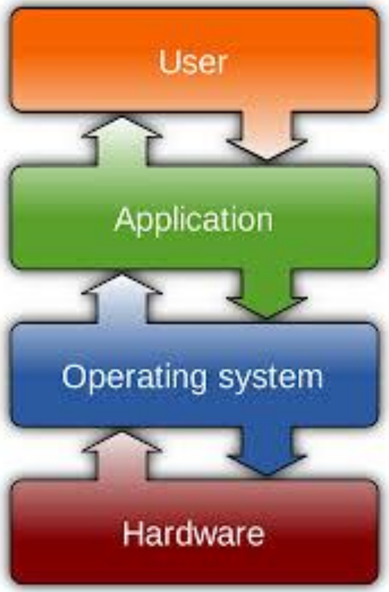
- OS structure
- Process Execution
- Process Management
- Memory Management
- File Management
- Storage Management
- I/O Sub-system Management
- Protection and Security
- User Interface

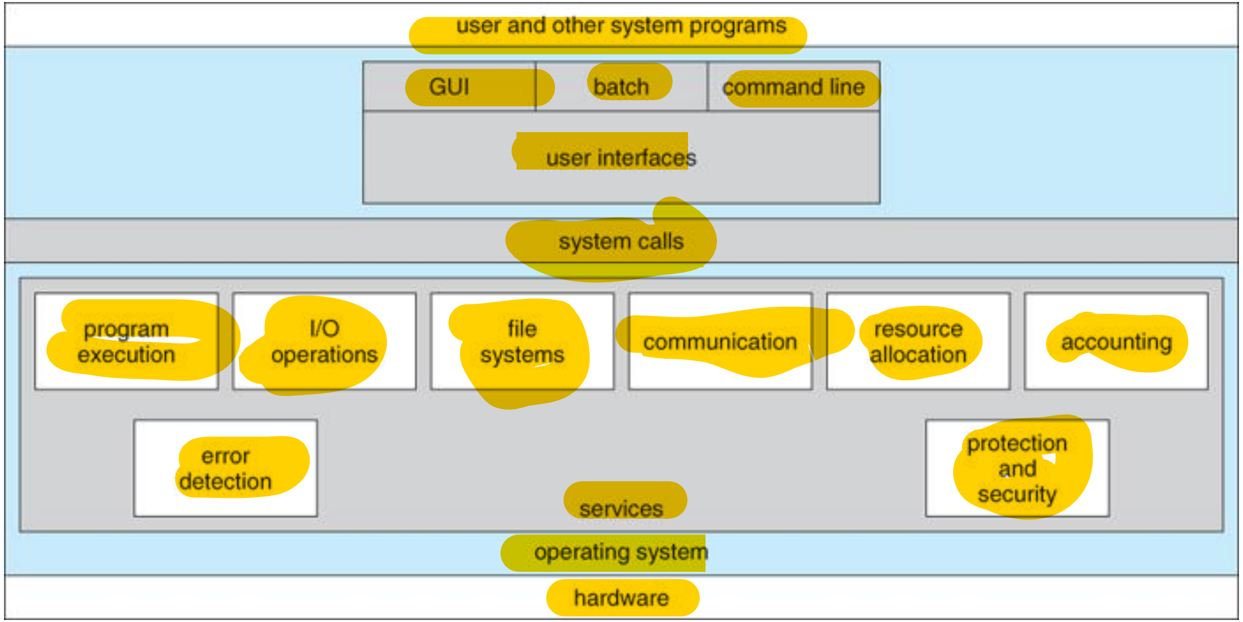
Operating System

- Operating system goals:
  - Execute user programs on hardware
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner
- OS is a **resource allocator**
- OS is a **control program**



Operating System Services





Operating System Services

- The OS structure is divided into many sub-components.
    - Process Execution
    - Process Management
    - Memory Management
    - File Management
    - Storage Management
    - I/O Sub-system Management
    - Protection and Security
    - User Interface

Process Execution

- 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.
- OS must be able to load a program into memory, run that program, and end execution
- Assign resources like CPU, memory, I/O, files, data to accomplish its task
- Process termination requires reclaim of any reusable resources

Process Execution

- Single-threaded process has one program counter specifying location of next instruction to execute
- Multi-threaded process has one program counter per thread
- Typically system has many processes, some user, some operating system running concurrently on one or more CPUs
- Concurrency by multiplexing the CPUs among the processes / threads

Process Management

- Creating and deleting both user and system processes
- Suspending and resuming processes
- Providing mechanisms for process synchronization
- Providing mechanisms for process communication
- Providing mechanisms for deadlock handling

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
- Keeping track of which parts of memory are currently being used and by whom
- Deciding which processes and data to move into and out of memory
- Allocating and deallocating memory space as needed

File Management

- OS provides uniform, logical view of information storage
- Abstracts physical properties of storage to logical storage unit - file
  - Files are usually organized into directories
  - OS determines access control on files/directories that determine who can access what
- File-System management include
  - Creating and deleting files and directories
  - Primitives to manipulate files and directories
  - Mapping files onto secondary storage
  - Backup files onto stable (non-volatile) storage media

Storage Management

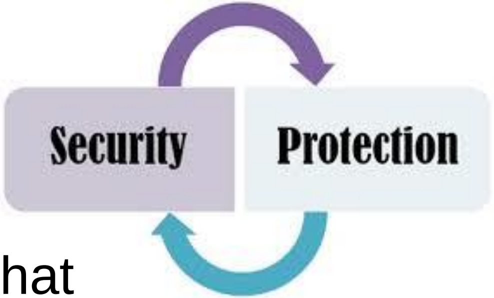
- Mass storage devices (disks/ tape drives) store data that does not fit in main memory or data that must be kept for a long period of time
- Devices vary in access speed, capacity, data-transfer rate, access method
- OS activities in disk management includes
  - Free-space management
  - Storage allocation
  - Disk scheduling

I/O Subsystem Management

- OS hides peculiarities of hardware devices from the user
- I/O subsystem responsible for
  - Buffering (storing data temporarily while it is being transferred)
  - Caching (storing parts of data in faster storage for performance)
  - Spooling (the overlapping of output of one job with input of other jobs)
  - Providing device-driver interface

Protection and Security

- **Protection** – any mechanism for controlling access of processes or users to resources defined by the OS
- **Security** – defense of the system against internal and external attacks
  - Huge range, including denial-of-service, worms, viruses, identity theft, theft of service



- Distinguish among users, to determine who can do what
  - User identities (user IDs) and associated access controls on resources
  - Group identifier (group ID) associated access controls on resources
  - Privilege escalation to change to give more rights

User Interface

- Provides a user friendly platform to initiate actions from user side.
- The UI primarily receives command from user and executes it
- Command-Line Interface (CLI) allows direct command entry
- User-friendly desktop Graphical User Interface (GUI)
  - Usually mouse, keyboard, and monitor used for giving inputs.
  - Icons represent files, programs, actions, etc
  - Various mouse buttons over objects in the interface cause various actions (provide information, options, execute function, open directory)
- Many systems (Microsoft, Apple-Mac OS, UNIX) now include both CLI and GUI interfaces

Thank you

johnjose@iitg.ac.in

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