Introduction to Operating System

What is an Operating System?

A program that acts as an intermediary between a user of a computer and the computer hardware

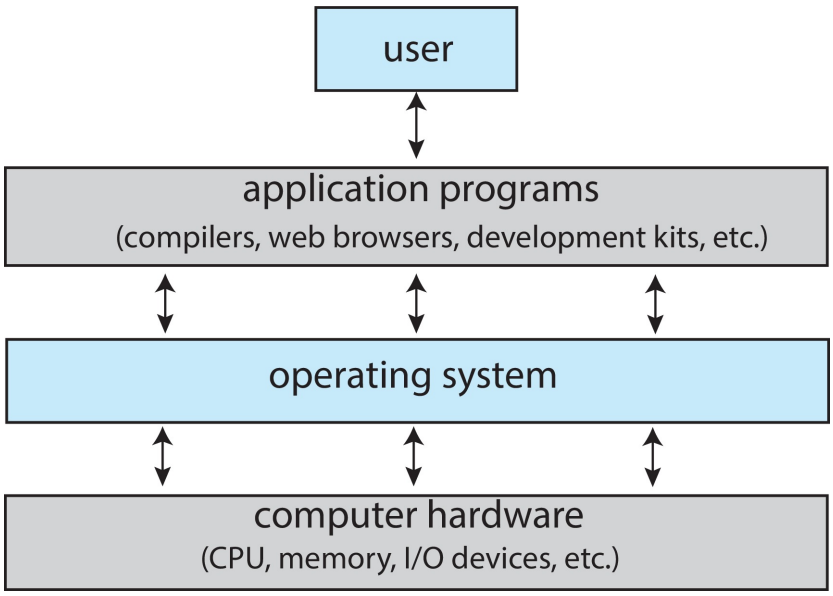
- Operating system goals:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

Computer System Structure

Computer system can be divided into four components:

- Hardware – provides basic computing resources
  - CPU, memory, I/O devices
- Operating system
  - Controls and coordinates use of hardware among various applications and users
- Application programs
  - define the ways in which the system resources are used to solve the computing problems of the users
  - Word processors, compilers, web browsers, database systems, video games
- Users
  - People, machines, other computers

Abstract View of Components of Computer



What Operating Systems Do?

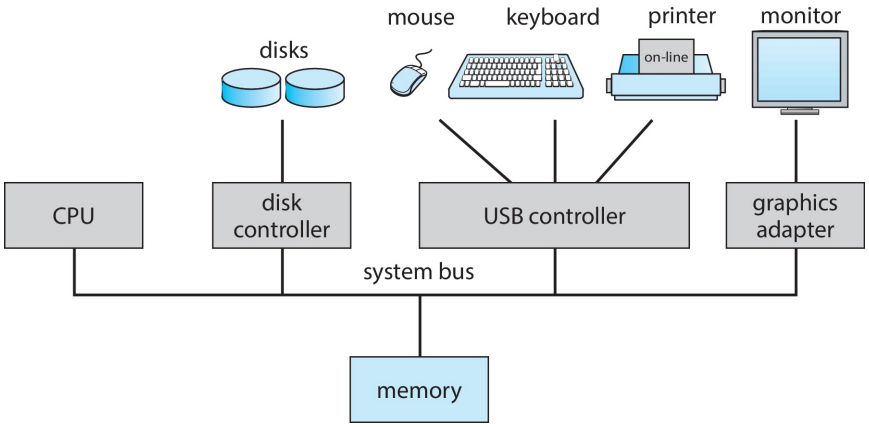
- Depends on the point of view
  - Users want convenience, ease of use and good performance
    - Don ’t care about resource utilization
  - But shared computer such as mainframe or minicomputer must keep all users happy
    - Operating system is a resource allocator and control program making efficient use of HW and managing execution of user programs

What Operating Systems Do?

- Users of dedicate systems such as workstations have dedicated resources but frequently use shared resources from servers
  - Mobile devices like smartphones and tables are resource poor, optimized for usability and battery life
    - Mobile user interfaces such as touch screens, voice recognition
  - Some computers have little or no user interface, such as embedded computers in devices and automobiles
    - Run primarily without user intervention

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

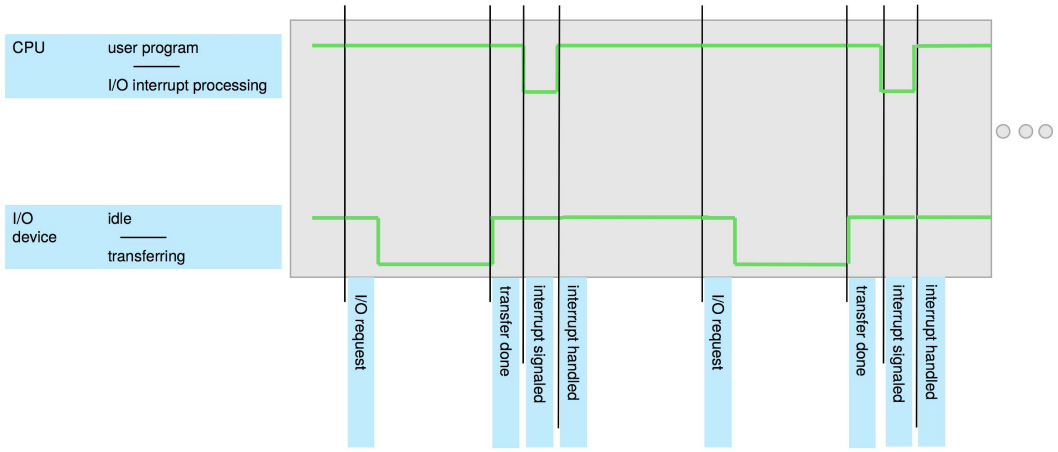


Computer-System Operation

- Common Functions of Interrupts
  - 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
– A *trap* or exception is a software-generated interrupt caused either by an error or a user request
– An operating system is interrupt driven

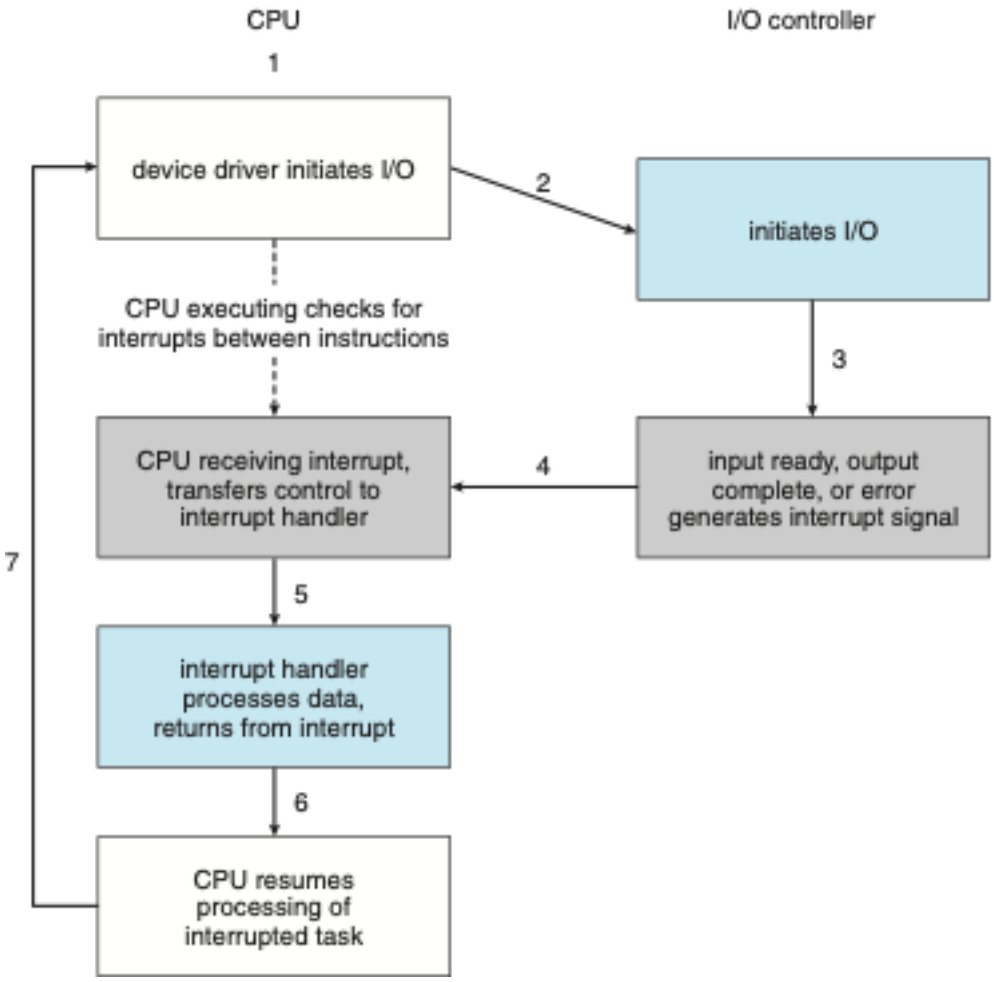
Interrupt Timeline



Interrupt Handling

- The CPU hardware has a wire called the interrupt-request line that the CPU senses after executing every instruction.
- When the CPU detects (an asynchronous event) that a controller has asserted a signal on the interrupt-request line.
- The operating system preserves the state of the CPU by storing the registers and the program counter
- It reads the interrupt number and jumps to the interrupt-handler routine by using that interrupt number as an index into the interrupt vector.
- It then starts execution at the address associated with that index.
- It performs a state restore and executes a return from interrupt instruction to return the CPU to the execution state prior to the interrupt.

Interrupt-driven I/O Cycle



Interrupt-driven I/O Cycle

- Most CPUs have two interrupt request lines.
- One is the non-maskable interrupt, which is reserved for events such as unrecoverable memory errors.
- The second interrupt line is maskable: it can be turned off by the CPU before the execution of critical instruction sequences that must not be interrupted.
- The maskable interrupt is used by device controllers to request service.
- The interrupt mechanism also implements a system of interrupt priority levels.
- These levels enable the CPU to defer the handling of low-priority interrupts without masking all interrupts and makes it possible for a high-priority interrupt to preempt the execution of a low-priority interrupt.

Checking Interrupts on Linux OS

- The file / proc /interrupts contains information about the hardware interrupts in use and how many times processor has been interrupted
  - watch -n1 “cat / proc /interrupts”

  - The first column indicates the IRQ (Interrupt Request Line) number. Subsequent columns indicate how many interrupts have been generated for the IRQ number on different CPU cores. The last column provides information on the programmable interrupt controller that handles the interrupt.
  - A small IRQ number value means higher priority.

Checking Interrupts on Linux OS

- The software interrupts are generated when the CPU executes an instruction which can cause an exception condition in the CPU [ALU unit] itself.
  - watch -n1 “cat / proc /stat”
  - The ” intr ” line gives counts of interrupts serviced since boot time, for each of the possible system interrupts.

Storage Structure

- Main memory – only large storage media that the CPU can access directly
  - Random access
  - Typically, volatile
  - Typically, random-access memory in the form of Dynamic Random-access Memory (DRAM)
- Secondary storage – extension of main memory that provides large nonvolatile storage capacity

Storage Structure (Cont.)

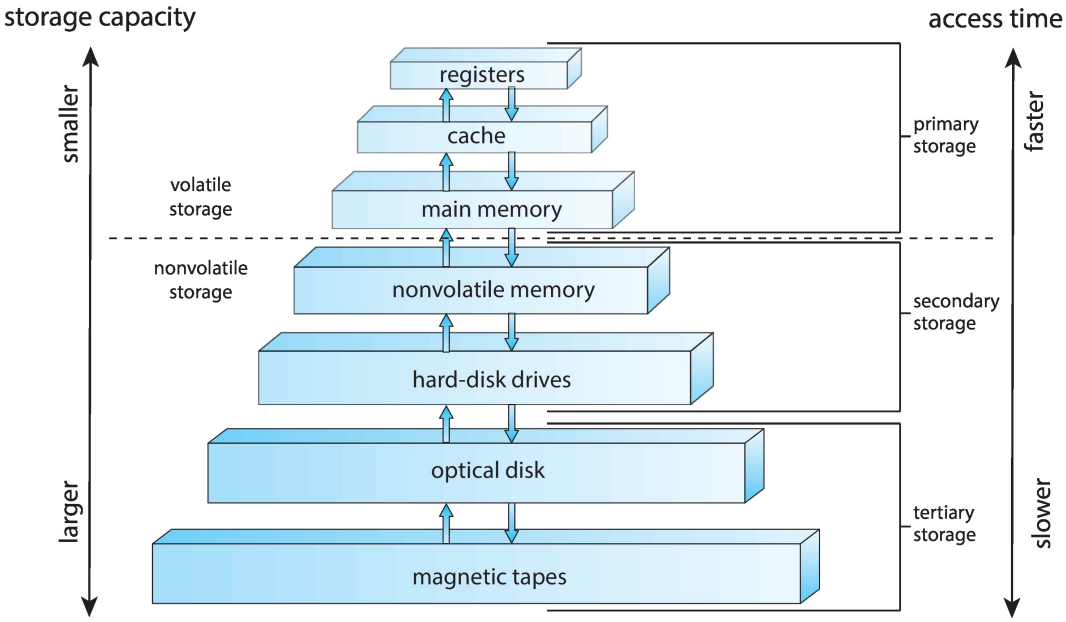
- Hard Disk Drives (HDD) – rigid metal or glass platters covered with magnetic recording material
  - Disk surface is logically divided into tracks, which are subdivided into sectors
  - The disk controller determines the logical interaction between the device and the computer
- Non-volatile memory (NVM) devices– faster than hard disks, nonvolatile
  - popular as capacity and performance increases, price drops

Storage Definitions and Notation Review

Unit	Shortened	Capacity
Bit	b	1 or 0 (on or off)
Byte	B	8 bits
Kilobyte	KB	1024 bytes
Megabyte	MB	1024 kilobytes

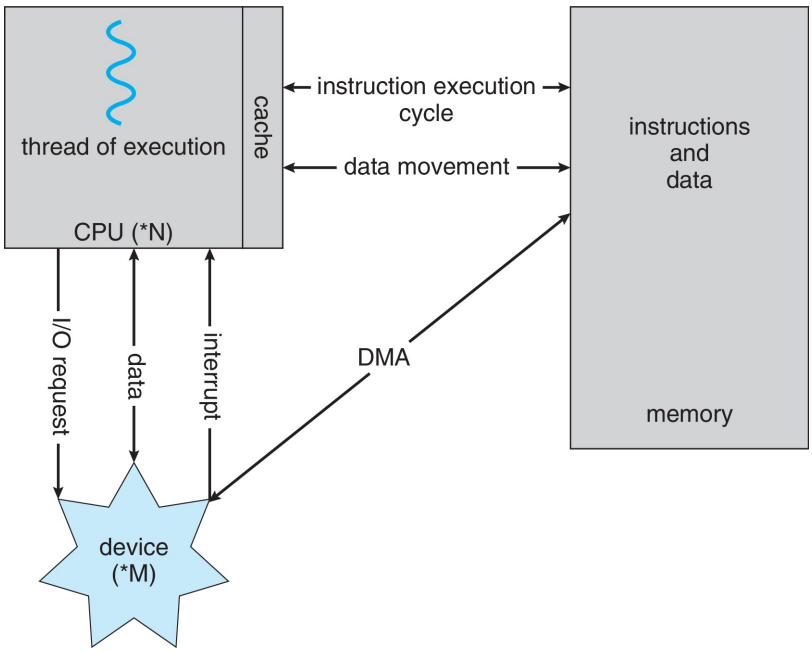
Gigabyte	GB	1024 megabytes
Terabyte	TB	1024 gigabytes
Petabyte	PB	1024 terabytes

Storage-Device Hierarchy

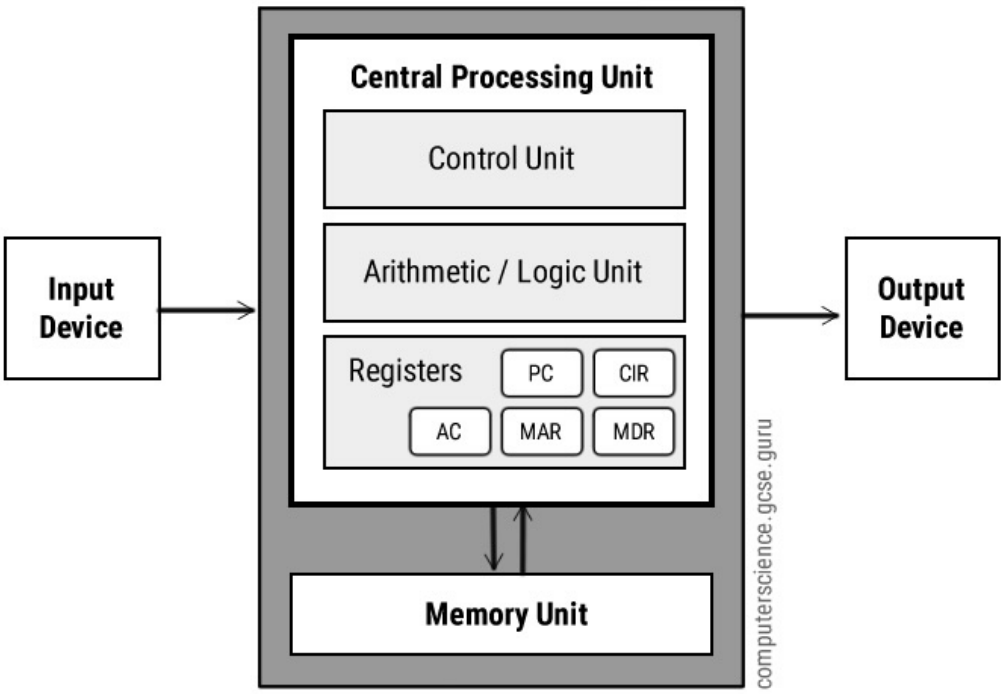


How a Modern Computer Works

- Direct Memory Access Structure



A Von Neuman's Architecture



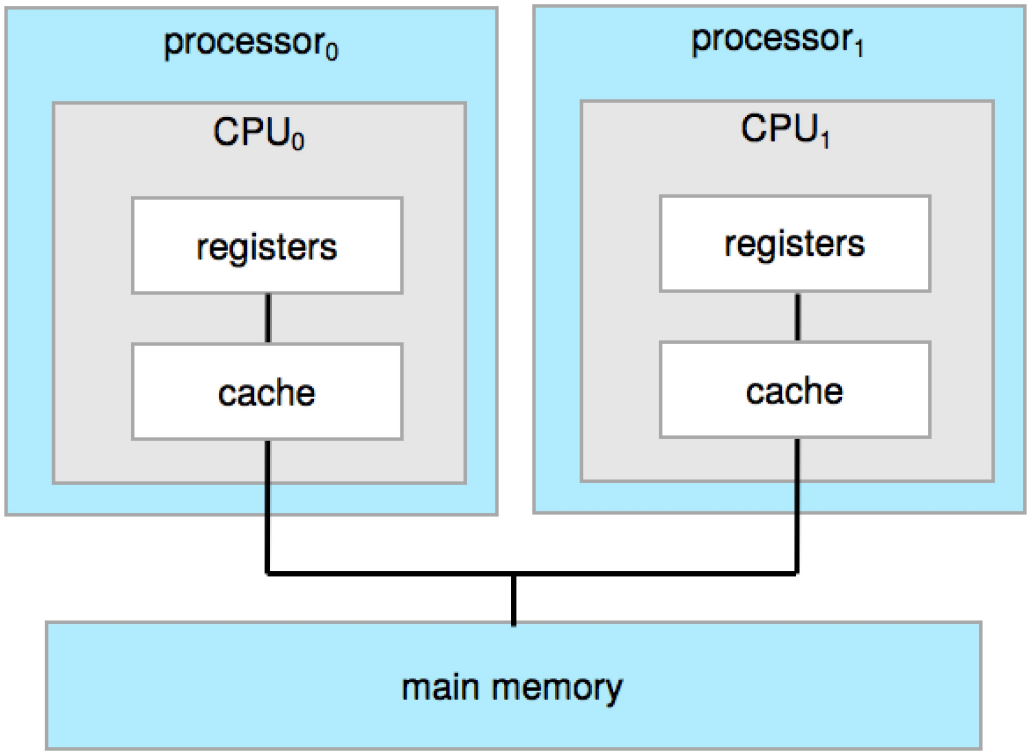
Computer-System Architecture

- Most systems use a single general-purpose processor containing one CPU with single processing core. The core can execute a general-purpose instruction set.
  - Most systems have special-purpose processors as well, but that does not make it multiprocessor system.
  - Multiprocessor's systems growing in use and importance
  - Also known as parallel systems, tightly-coupled systems

Computer-System Architecture

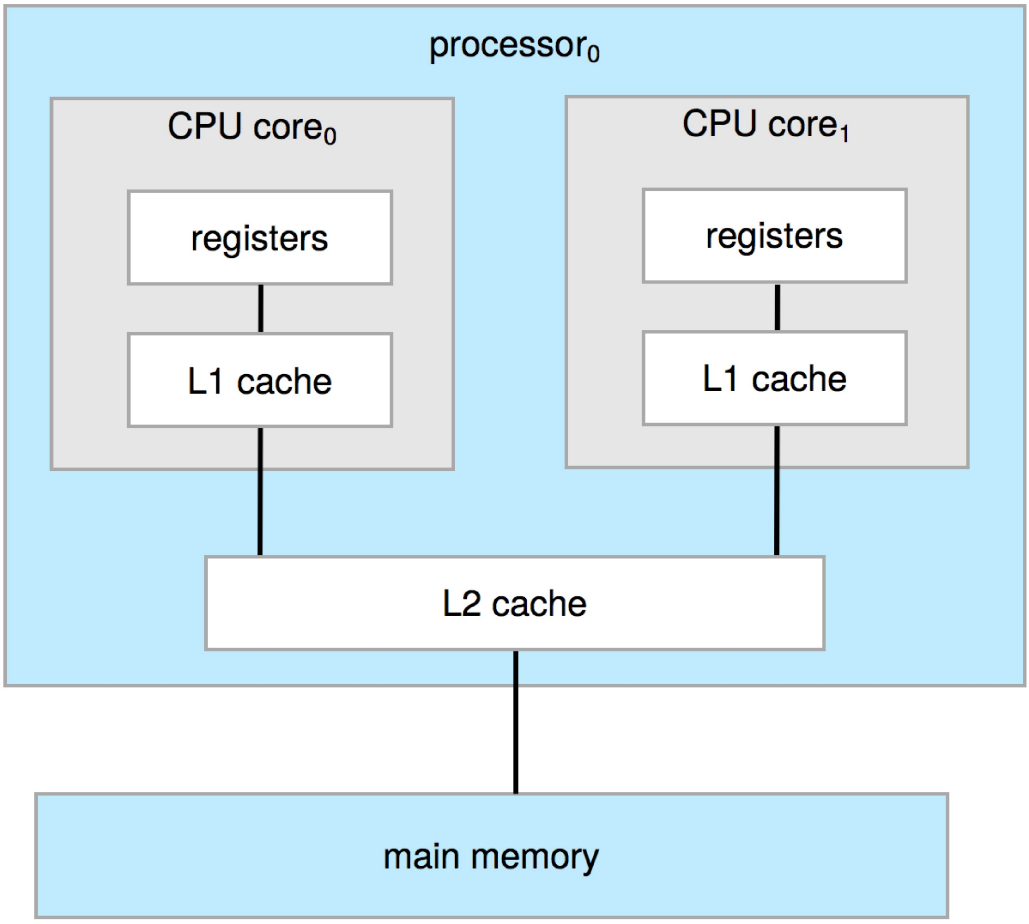
- Advantages include:
  - Increased throughput
  - Economy of scale
  - Increased reliability – graceful degradation or fault tolerance
- Two types:
  - Asymmetric Multiprocessing – each processor is assigned a specific task.
  - Symmetric Multiprocessing – each processor performs all tasks

Symmetric Multiprocessing Architecture



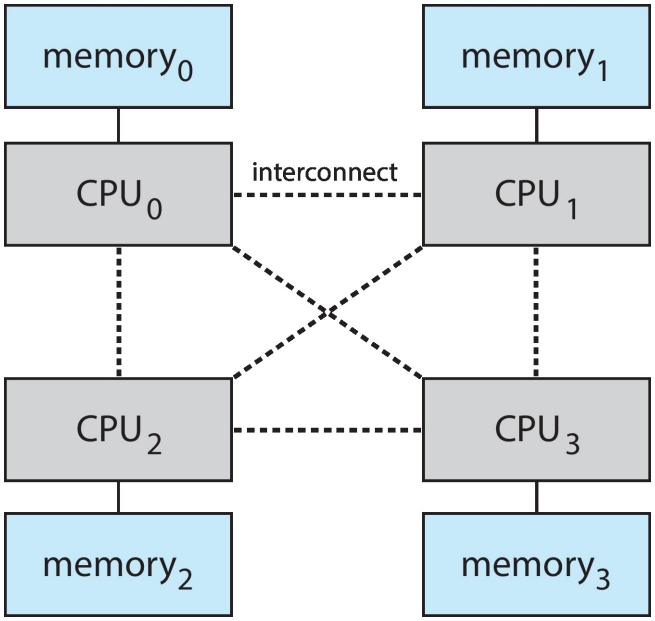
Dual-Core Design

- Multi-chip and multicore
  - Systems containing all chips
    - Chassis containing multiple separate systems



Non-Uniform Memory Access System

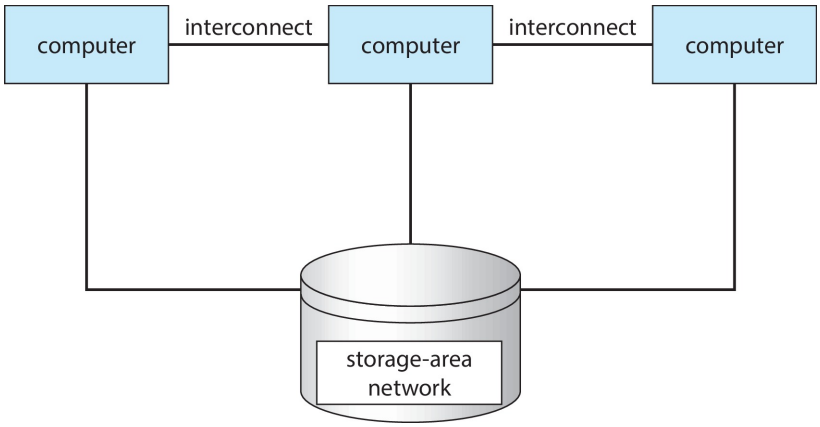
- The extent of sharing of storage can create different multi-processor systems.



Clustered Systems

- Like multiprocessor systems, but multiple 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
  - 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

Clustered Systems



Hadoop

HADOOP

Hadoop is an open-source software framework that is used for distributed processing of large data sets (known as big data) in a clustered system containing simple, low-cost hardware components. Hadoop is designed to scale from a single system to a cluster containing thousands of computing nodes. Tasks are assigned to a node in the cluster, and Hadoop arranges communication between nodes to manage parallel computations to process and coalesce results. Hadoop also detects and manages failures in nodes, providing an efficient and highly reliable distributed computing service.

Hadoop is organized around the following three components:

1. A distributed file system that manages data and files across distributed computing nodes.
2. The YARN ("Yet Another Resource Negotiator") framework, which manages resources within the cluster as well as scheduling tasks on nodes in the cluster.
3. The MapReduce system, which allows parallel processing of data across nodes in the cluster.

Hadoop is designed to run on Linux systems, and Hadoop applications can be written using several programming languages, including scripting languages such as PHP, Perl, and Python. Java is a popular choice for developing Hadoop applications, as Hadoop has several Java libraries that support MapReduce. More information on MapReduce and Hadoop can be found at https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html and https://hadoop.apache.org

Operating System Joke

- Why do astronauts prefer the Linux operating system
  - Because you can't open Window's in space.

Operating-System Operations: Bootstrap

- Bootstrap program is loaded at power-up or reboot
  - Typically stored in ROM or EPROM, generally known as firmware
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution

Operating-System Operations:

- Kernel loads
  - Starts system daemons (services provided outside of the kernel)
  - Kernel 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 – system call
  - Other process problems include infinite loop, processes modifying each other or the operating system

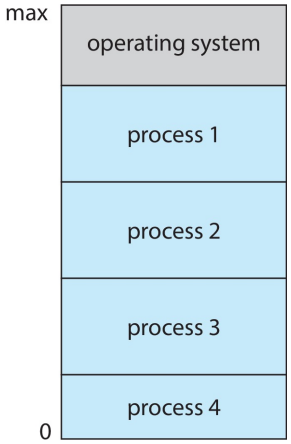
Multiprogramming (Batch system)

- Single user cannot always keep CPU and I/O devices busy
- 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 job has to wait (for I/O for example), OS switches to another job

Multitasking (Timesharing)

- A logical extension of Batch systems – the 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 process
  - If several jobs ready to run at the same time then needs 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

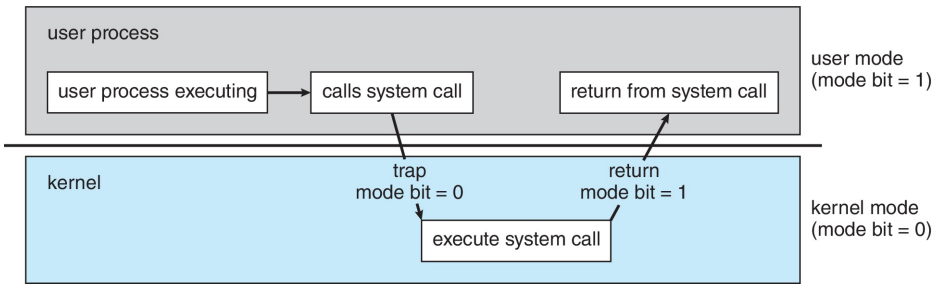


Dual-mode Operation

- Dual-mode (hardware supported) 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.
- When a user is running –> mode bit is “user”(1) When kernel code is executing –> mode bit is “kernel”(0)

Dual-mode Operation

- How do we guarantee that user does not explicitly set the mode bit to “kernel”?
- System call changes mode to kernel, return from call resets it to user
- Transition from User to Kernel Mode



Timer

- Timer to prevent infinite loop (or 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

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.

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

File-system Management

- OS provides uniform, logical view of information storage
- Abstracts physical properties to logical storage unit - file
- Each medium is controlled by device (i.e., disk drive, tape drive)

Caching

- Important principle, performed at many levels in a computer (in hardware, operating system, software)
- Information in use copied from slower to faster storage temporarily
- Faster storage (cache) checked first to determine if information is there
- If it is, information used directly from the cache (fast)
- If not, data copied to cache and used there
  - Cache smaller than storage being cached

  - Cache management important design problem
  - Cache size and replacement policy

Characteristics of Various Types of Storage

- Movement between levels of storage hierarchy can be explicit or implicit

Level	1	2	3	4	5
Name	registers	cache	main memory	solid-state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25-0.5	0.5-25	80-250	25,000-50,000	5,000,000
Bandwidth (MB/sec)	20,000-100,000	5,000-10,000	1,000-5,000	500	20-150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

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

Protection and Security

- Systems generally first distinguish among users, to determine who can do what
- User identities (user IDs, security IDs) include name and associated number, one per user
- User ID then associated with all files, processes of that user to determine access control
- Group identifier (group ID) allows set of users to be defined and controls managed, then also associated with each process, file Privilege escalation allows user to change to effective ID with more rights

Virtualization

- Allows operating systems to run applications within other OSes
- Virtualization – OS natively compiled for CPU, running guest OSes also natively compiled
- Consider VirtualBox running Linux on native Windows host OS.
- Emulation - involves simulating computer hardware in software, is typically used when the source CPU type is different from the target CPU type.

Kernel Data Structures

Many similar to standard programming data structures + Singly linked list + Doubly linked list + Circular linked list

Kernel Data Structures

- Binary search tree left <= right
  - Search performance is O(n)
  - Balanced binary search tree is O(lg n)

Kernel Data Structures

- Hash function can create a hash map
- Bitmap – string of n binary digits representing the status of n items
  - Linux data structures defined in include files <linux/list.h >, <linux/kfifo.h>, <linux/rbtree.h >

References

- A. Silberschatz, P.B. Galvin, and G. Gagne. Operating System Concepts , 10th Edition; 2018; John Wiley and Sons.
- W. Stallings. Operating Systems: Internals and Design Principles. Prentice Hall, Upper Saddle River, NJ, Eigth edition, 2014.