Operating Systems 2023 Spring Term
Week 1

Dr. Emrah İnan (emrahinan@iyte.edu.tr)

Overview

March 9, 2023

What is an Operating System?

Figure: Gates and Crellin Laboratories, Division of Chemistry and Chemical Engineering, Caltech
(http://www.columbia.edu/cu/computinghistory/oldpunch.html)

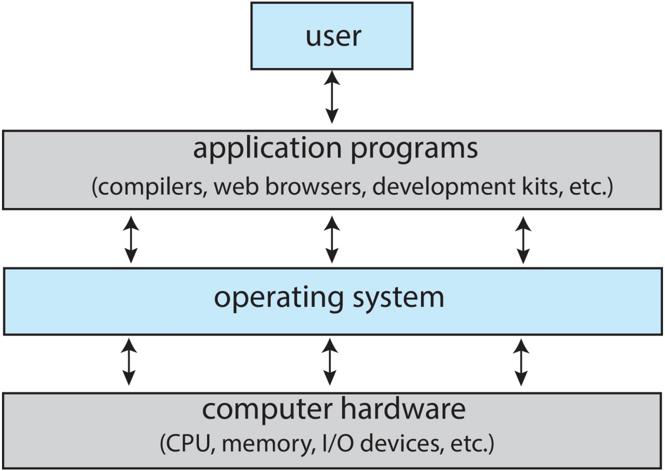
- A program that acts as an intermediary between a user of a computer and the computer hardware
- Operating system goals:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

Computer System Structure: 4 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)

Four Components of a Computer System

Figure: Abstract view of the components of a computer on page 4



What Operating Systems Do

- Users want convenience, ease of use and good performance (Don’t care about resource utilisation)
- 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
- 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, optimised for usability and battery life
- Some computers have little or no user interface, such as embedded computers in devices and automobiles

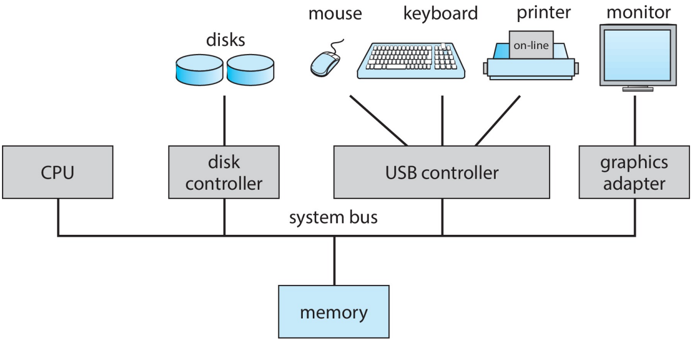
Operating System Definition

- “Everything a vendor ships when you order an operating system” is a good approximation
- **kernel:**“The one program running at all times on the computer” is the part of the operating system
- **A system program:** ships with the operating system, but not part of the kernel, or
- **An application program** all programs not associated with the operating system
- **middleware:** a set of software frameworks that provide additional services to application developers such as databases, multimedia, graphics

Computer Startup

- **Bootstrap program:**simple code to initialise the system, load the kernel
- Starts system daemons (services provided outside of the kernel)
- Kernel interrupt driven (hardware and software)

Computer System Organization



- 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
- Each device controller type has an operating system device driver to manage it
- CPU moves data from/to main memory to/from local buffers
- I/O is from the device to local buffer of controller
- Device controller informs CPU that it has finished its operation by causing an **interrupt**

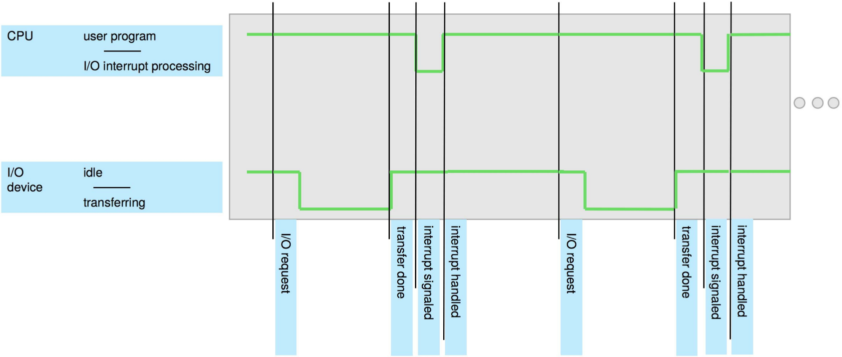
Common Functions of Interrupts

- 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**

Intel Processor Event-vector Table on Page 11

vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
13	general protection
14	page fault
15	(Intel reserved, do not use)
16	floating-point error
17	alignment check
18	machine check
19–31	(Intel reserved, do not use)
32–255	maskable interrupts

Interrupt Timeline



Interrupt Handling

- The operating system preserves the state of the CPU by storing the registers and the program counter
- Determines which type of interrupt has occurred
- Separate segments of code determine what action should be taken for each type of interrupt

I/O Structure

- After I/O starts, control returns to user program only upon I/O completion
  - Wait instruction idles the CPU until the next interrupt
  - Wait loop (contention for memory access)
  - At most one I/O request is outstanding at a time, no simultaneous I/O processing
- After I/O starts, control returns to user program without waiting for I/O completion
  - System call – request to the OS to allow user to wait for I/O completion
  - Device-status table contains entry for each I/O device indicating its type, address, and state
  - OS indexes into I/O device table to determine device status and to modify table entry to include interrupt

Direct Memory Access Structure

- Used for high-speed I/O devices able to transmit information at close to memory speeds
- Device controller transfers blocks of data from buffer storage directly to main memory without CPU intervention
- Only one interrupt is generated per block, rather than the one interrupt per byte

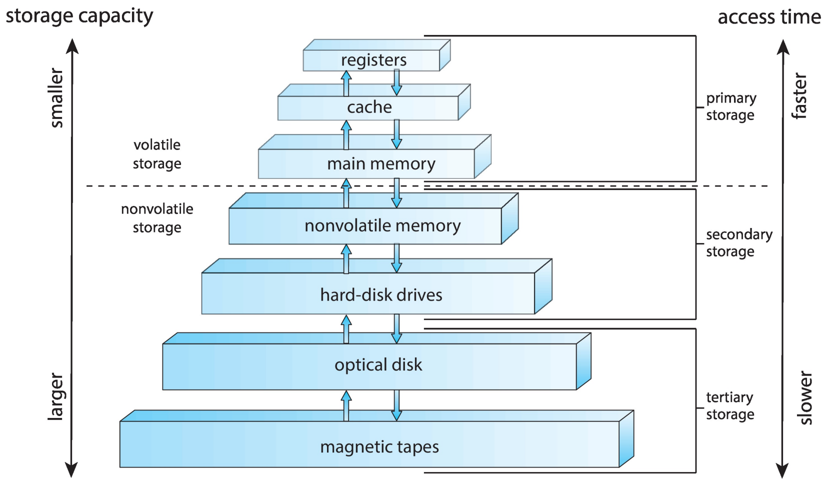
Storage Structure

- 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
- 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

Storage Hierarchy

- Storage systems organised in hierarchy (Speed, Cost, Volatility)
- Caching – copying information into faster storage system; main memory can be viewed as a cache for secondary storage
- Device Driver for each device controller to manage I/O (Provides uniform interface between controller and kernel)

Storage-Device Hierarchy



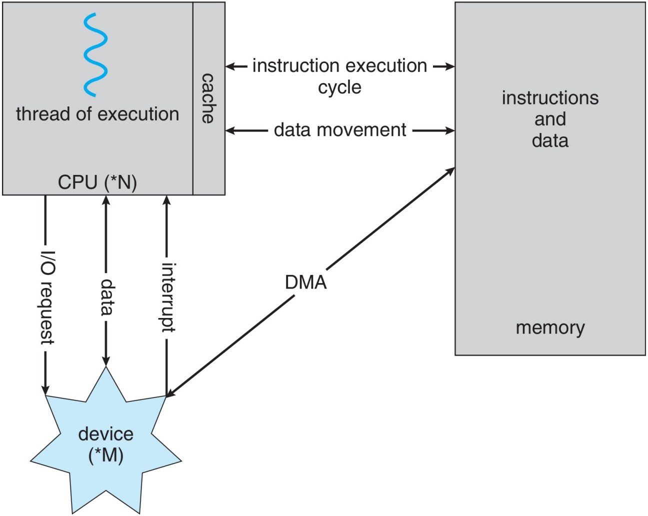
Caching

- 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

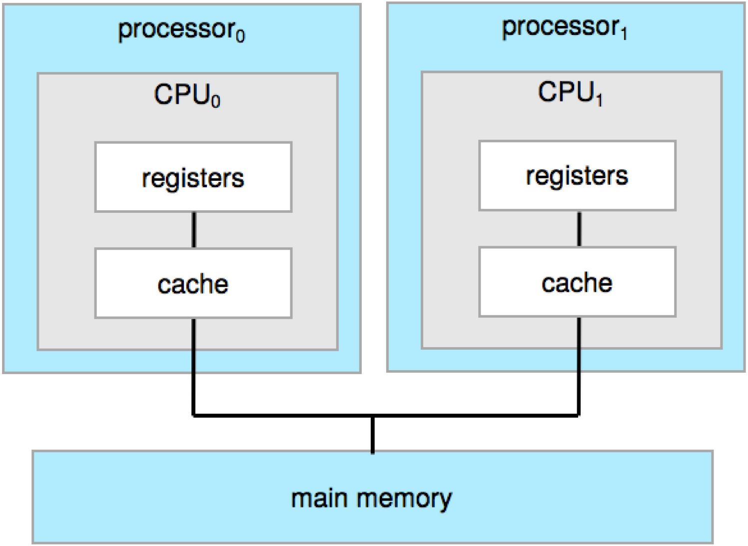
Computer-System Architecture

- Most systems use a single general-purpose processor (Most systems have special-purpose processors as well)
- Multiprocessors systems growing in use and importance
  - Also known as parallel systems, tightly-coupled systems
  - 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

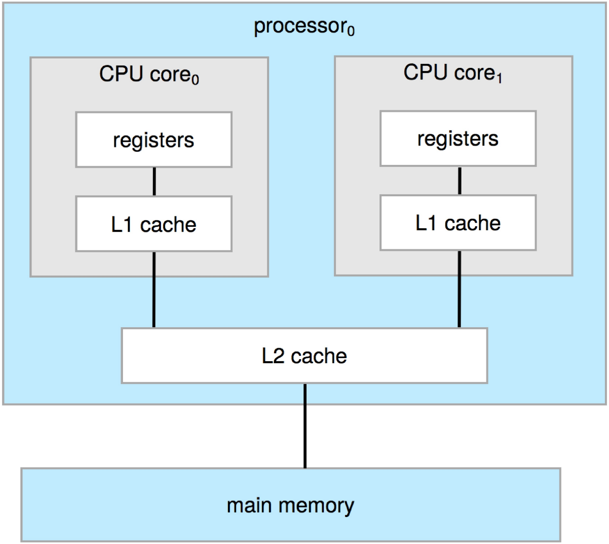
How a Modern Computer Works



Symmetric Multiprocessing Architecture



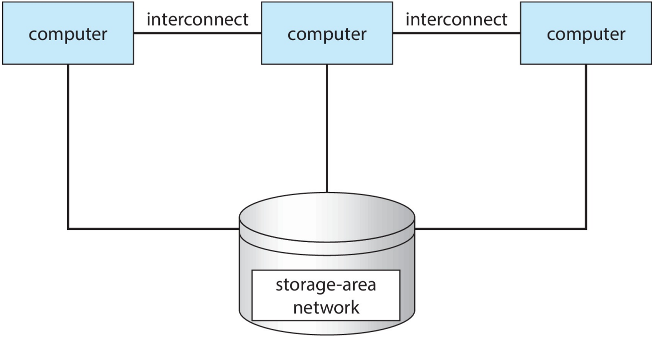
Dual-Core Design



Clustered Systems I

- 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 parallelisation
- Some have distributed lock manager (DLM) to avoid conflicting operations

Clustered Systems II



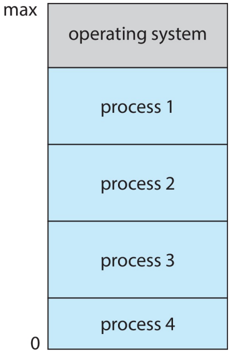
Multiprogramming (Batch system)

- 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 or 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 -> 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

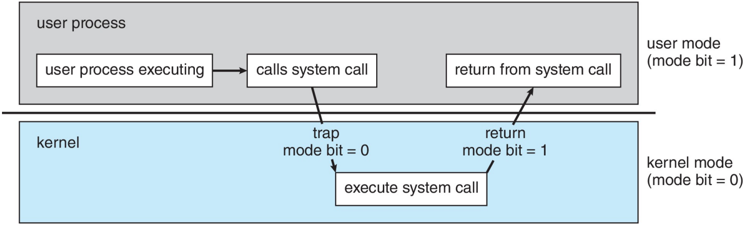


Operating-System Operations

- 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
- 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.
    - When a user is running -> mode bit is “user”
    - When kernel code is executing -> mode bit is “kernel”
    - 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
    - Some instructions designated as privileged, only executable in kernel mode

Transition from User to Kernel Mode

- 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.
- Process needs resources to accomplish its task such as CPU, memory, I/O, files or initialisation data
- Process termination requires reclaim of any reusable resources
- 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
- 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 Activities

- The operating system is responsible for the following activities in connection with process management:
  - Creating and deleting both user and system processes
  - Suspending and resuming processes
  - Providing mechanisms for process synchronisation
  - 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 (Optimising CPU utilisation 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

Storage 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) -> Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)
- File-System management
  - Files usually organised into directories
  - Access control on most systems to determine who can access what
  - OS activities include: (1) Creating and deleting files and directories (2) Primitives to manipulate files and directories (3) Mapping files onto secondary storage (4) Backup files onto stable (non-volatile) storage media

Mass-Storage Management

- Usually disks used to store data that does not fit in main memory or data that must be kept for a “long” period of time
- Proper management is of central importance
- Entire speed of computer operation hinges on disk subsystem and its algorithms
- OS activities (1) Mounting and unmounting (2) Free-space management (3) Storage allocation (4) Disk scheduling (5) Partitioning (6) Protection

I/O Subsystem

- One purpose of OS is to hide peculiarities of hardware devices from the user
- I/O subsystem responsible for
  - Memory management of I/O including 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)
  - General device-driver interface
  - Drivers for specific hardware devices

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)
- 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

Distributed Systems

- Collection of separate, possibly heterogeneous, systems networked together
- Network is a communications path, TCP/IP most common
  - Local Area Network (LAN)
  - Wide Area Network (WAN)
  - Metropolitan Area Network (MAN)
  - Personal Area Network (PAN)
- Network Operating System provides features between systems across network (1) Communication scheme allows systems to exchange messages (2) Illusion of a single system

Special-Purpose Systems

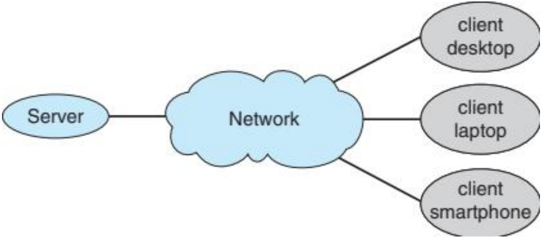
- Real-time embedded systems most prevalent form of computers (Vary considerable, special purpose, limited purpose OS, real-time OS Use expanding)
- Handheld smartphones, tablets, etc.
- Robotic systems

Computing Environments

- **Traditional:** Stand-alone general-purpose machines
- But blurred as most systems interconnect with others (i.e., the Internet)
- Portals provide web access to internal systems
- Network computers (thin clients) are like Web terminals
- Mobile computers interconnect via wireless networks
- Networking becoming ubiquitous – even home systems use firewalls to protect home computers from Internet attacks

Client-Server Computing

- Many systems now servers, responding to requests generated by clients
- Compute-server system provides an interface to client to request services (i.e., database)
- File-server system provides interface for clients to store and retrieve files



Peer-to-Peer

- P2P does not distinguish clients and servers
- Instead all nodes are considered peers
- May each act as client, server or both
- Node must join P2P network
- Registers its service with central lookup service on network, or Broadcast request for service and respond to requests for service via discovery protocol
- Voice over IP (VoIP) such as Skype

Cloud Computing

- Delivers computing, storage, even apps as a service across a network
- Logical extension of virtualisation (allows operating systems to run applications within other OSes) because it uses virtualisation as the base for it functionality.
- Amazon EC2 has thousands of servers, millions of virtual machines, petabytes of storage available across the Internet, pay based on usage
- Many types:
  - Software as a Service (SaaS) – one or more applications available via the Internet (i.e., word processor)
  - Platform as a Service (PaaS) – software stack ready for application use via the Internet (i.e., a database server)
  - Infrastructure as a Service (IaaS) – servers or storage available over Internet (i.e., storage available for backup use)

Open-Source Operating Systems

- Operating systems made available in source-code format rather than just binary closed-source and proprietary
- Counter to the copy protection and Digital Rights Management (DRM) movement
- Started by Free Software Foundation (FSF), which has "copyleft" GNU Public License (GPL)
- Examples include GNU/Linux and BSD UNIX (including core of Mac OS X), and many more

Modern Operating Systems

- Object oriented design
  - Used for adding modular extensions to a small kernel
  - Enables programmers to customise an operating system without disrupting system integrity