Operating Systems

Introduction

- Textbooks
  - Silberschatz, A, Galvin, P.B, and Gagne, G., Operating System Principles, 9e, John Wiley & Sons, 2013.
- Reference Books
  - Stallings W., Operating Systems-Internals and Design Principles, 7e, Pearson Education, 2014.
  - Harvey M. Deital, “Operating System”, Third Edition, Pearson Education, 2013.
  - Andrew S. Tanenbaum, “Modern Operating Systems”, Second Edition, Pearson Education, 2004.
  - Gary Nutt, “Operating Systems”, Third Edition, Pearson Education, 2004.

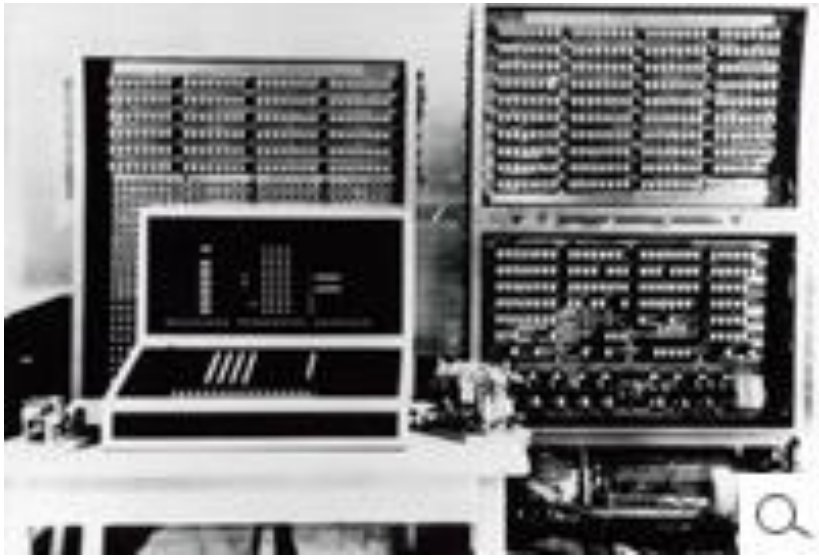
- A program that acts as an intermediary between a user of a computer and the computer hardware.
- Operating system goals:
  - ✓ Execute user programs and make solving user problems easier.
  - ✓ Make the computer system convenient to use.
  - ✓ Use the computer hardware in an efficient manner.

… OS problems parallel many problems in everyday life:
Queues, semaphores, buffers, traffic control, overlapping tasks, coordinating multiple related tasks, deadlock vs gridlock in traffic, scheduling problems …

- **Resource allocator** – manages and allocates resources
- **Control program** – controls the execution of user programs and operations of I/O devices .
- **Kernel** – the one program running always (all else being application programs).

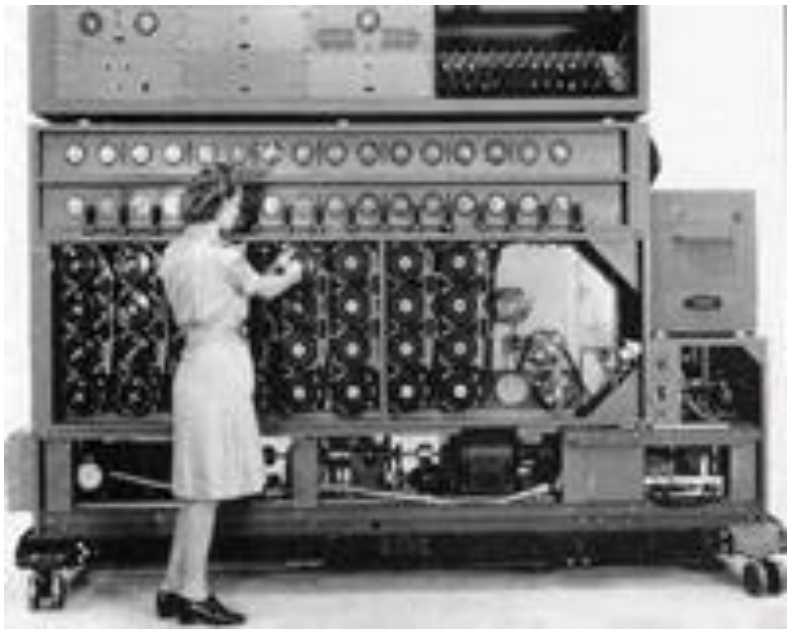
- Charles Babbage (1791-1871) designed the first true digital computer called the analytical engine.
- That was purely mechanical and intended to do math operations
- The engine was supposed to be made of brass, and steam powered
- He did not actually build the machine but inspired others in the field.
- Babbage also thought of the concept of software
- And hired the first programmer (Lady Ada) for his analytical machine

- Hewlett Packard was founded in 1939 by David Packard and Bill Hewlett,
- Their first product, the HP 200A Audio Oscillator
- Walt Disney Pictures ordered eight of the 200B model to use during the creation of the movie “Fantasia.”



Konrad Zuse: Zuse Z3

- Computer wars:
  - Germany vs Britain



Alan Turing: The bombe

- First generation 1945 – 1955
  - ✓vacuum tubes, plug boards
  - ✓Still very slow and used for scientific calculations
- No OS was needed
- Programs were entered by setting some switches

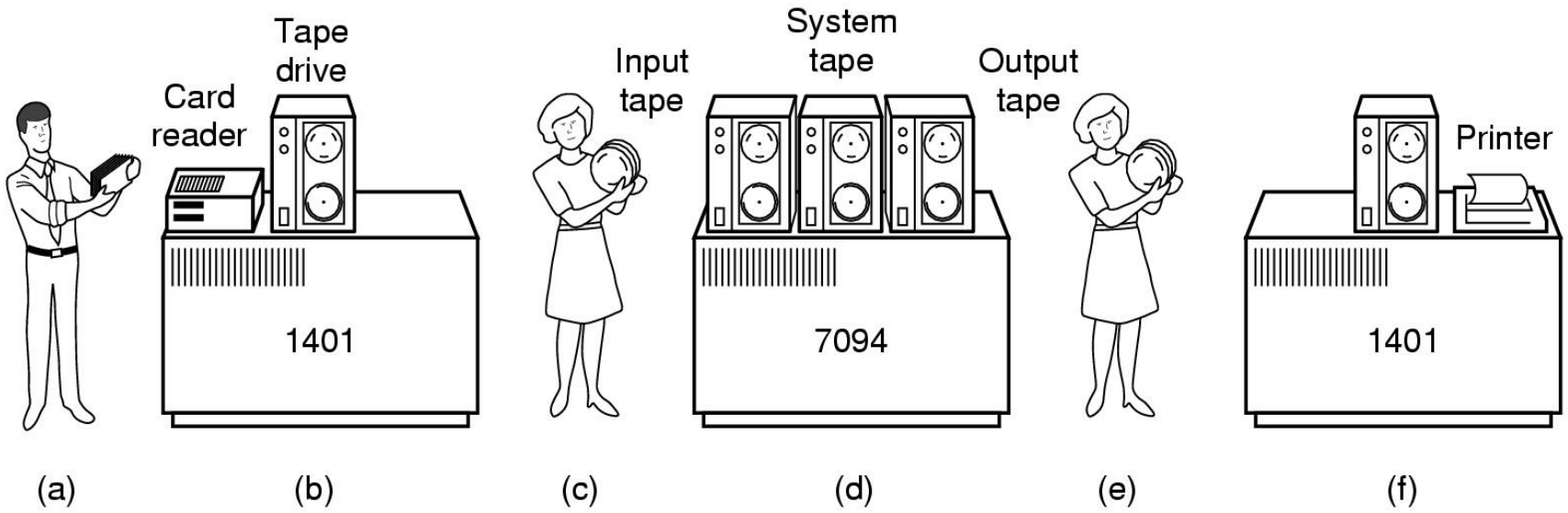
- "I think there is a world market for maybe five computers."

Thomas Watson, chairman of IBM - 1943

- Second generation 1955 – 1965
  - ✓ transistors, batch systems
  - ✓ Universities started to buy computers (spending millions of dollars)
  - ✓ Punched cards were used
  - ✓ To run a job (a program or a set of related programs) first punch it and give the deck to the operators and wait for the output (batch operation)
  - ✓ Computers were single user

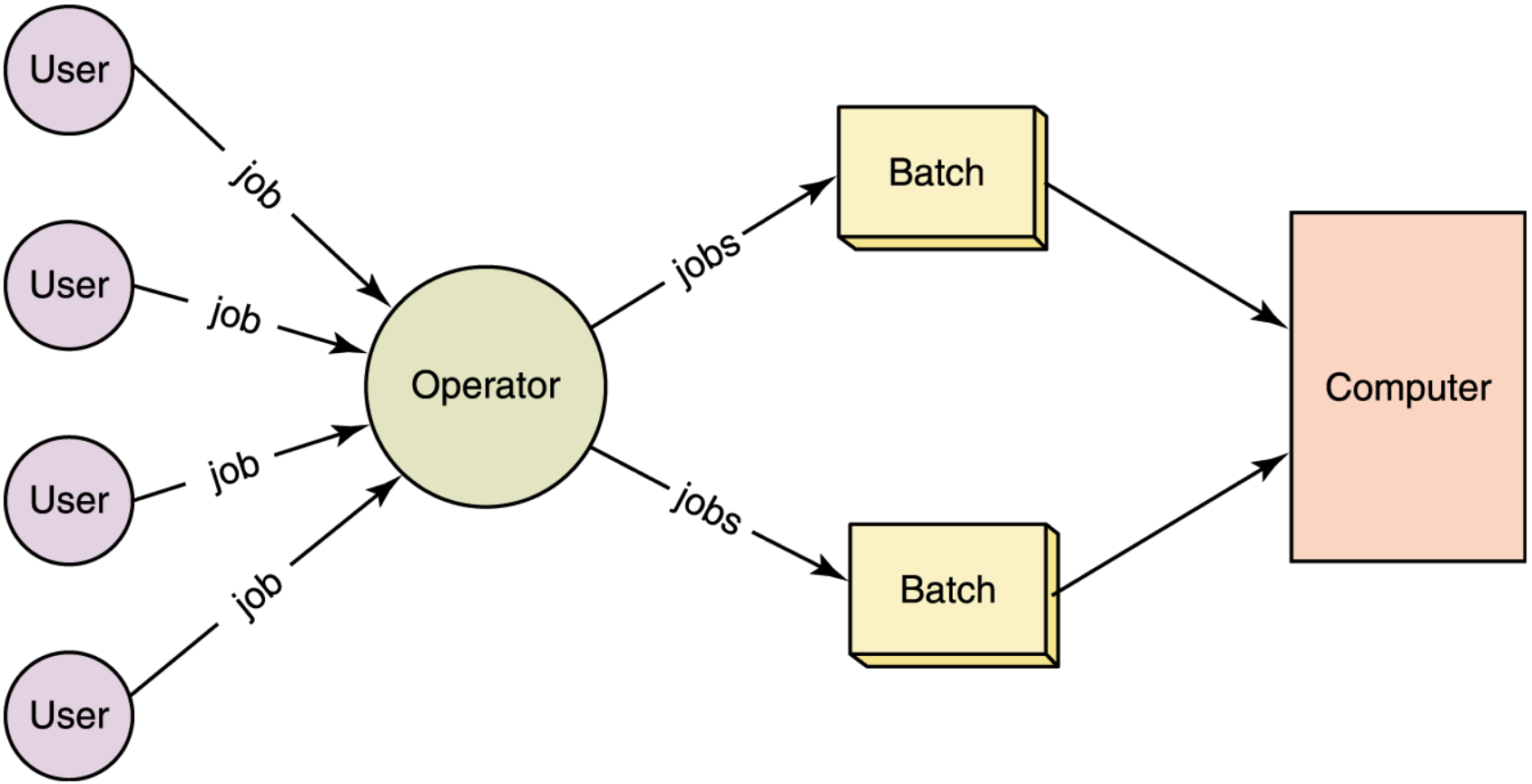
IBM STRETCH

DEC PDP-1

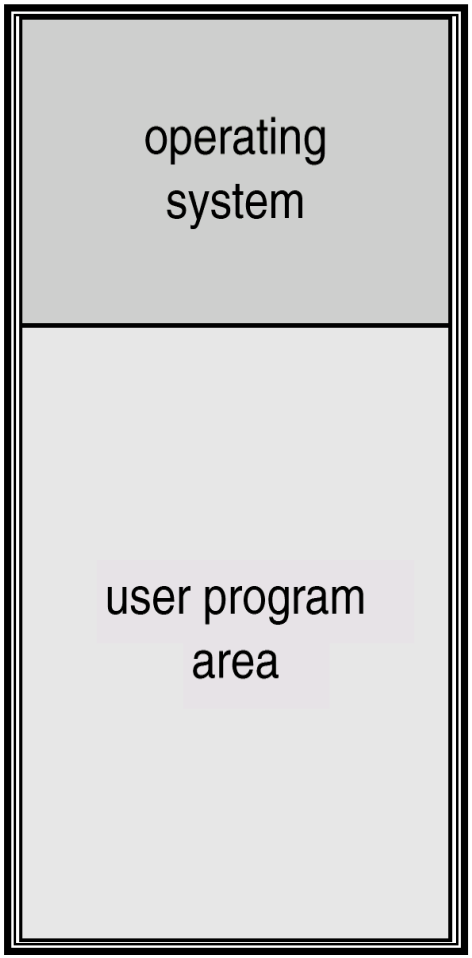


Early batch system

- bring cards to IBM 1401 machine (good at card reading)
- read cards to tape
- put tape on IBM 7094 which does computing
- put tape on IBM 1401 which prints output offline


User
User
User
User
job
job
job
job
Operator
jobs
jobs
Batch
Batch
Computer

- **1960's: Batch jobs with similar needs, leave the job with an operator, pick up results (hard copy listing) the next+ day – Hope no syntax errors! ("thin" listing) – Thick listing is a runtime error (Hex memory dump) - non-interactive**
- Reduce setup time by batching similar jobs
- Automatic job sequencing – automatically transfers control from one job to another. First rudimentary operating system.
- Resident monitor
  - ✓ initial control in monitor
  - ✓ control transfers to job
  - ✓ when job completes control transfers pack to monitor
- **Comment on use of "JCL" and how applications jobs were processed in early "batch days"**
- **Examples: THE system, IBM OS/360 etc.**



- "I have traveled the length and breadth of this country and talked with the best people, and I can assure you that data processing is a fad that won't last out the year.“
  - ✓ The editor in charge of business books for Prentice Hall - 1957

Third generation 1965 – 1980

- Integrated Circuits and multiprogramming
- IBM's System/360 mainframe computer with a standard architecture designed to handle both scientific and commercial computing

Third generation 1965 – 1980

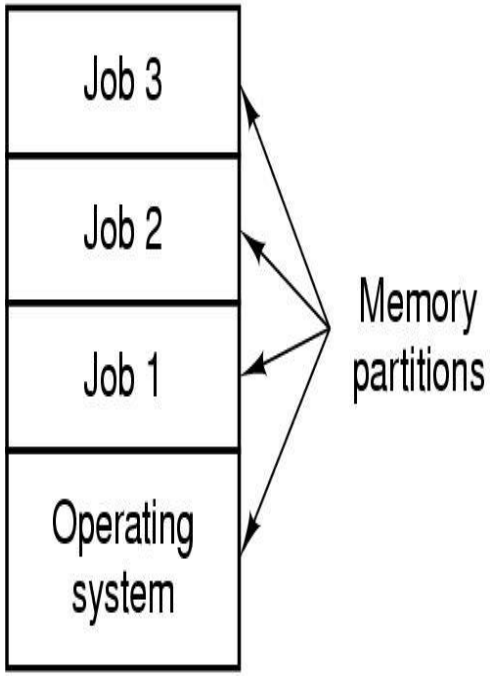
- OS/360 was the operating system of System/360
  - ✓ First the single task "Primary Control Program" was introduced
  - ✓ Then "Multiprogramming with a Fixed number of Tasks"
  - ✓ Finally "Multiprogramming with a Variable number of Tasks"

In **multiprogramming**, multiple jobs are in the memory and when a job finishes another can start immediately while the free memory partition is being reloaded

Several jobs are kept in main memory at the same time, and the CPU is multiplexed among them.

Multiprogramming (**Batch system**) needed for efficiency

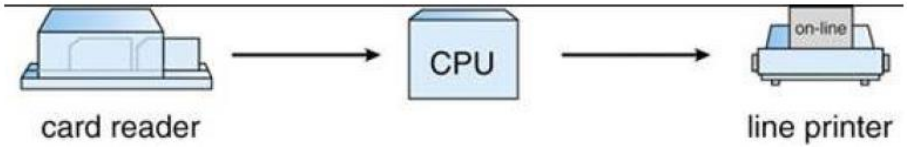
- ✓ Single user cannot always keep CPU and I/O devices busy
- ✓ Multiprogramming organizes jobs (code and data) so CPU always has one to execute
- ✓ A subset of total jobs in system is kept in memory
- ✓ One job selected and run via **job scheduling**
- ✓ When it has to wait (for I/O for example), OS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



- I/O **services provided** by the system.
  - ✓ **I/O done in "kernel mode" via device drivers**
  - ✓ **May run concurrently with application - if application must wait for I/O, then OS will allow another job to run - maximize CPU and I/O utilization.**
- Memory management – the system must allocate the memory to several jobs.
- CPU scheduling and **dispatching** – the system must choose among several jobs ready to run **- must minimize context switching time - a performance bottleneck.**
- Allocation of devices **- to "resident" jobs - danger of deadlock.**

- The concept of spooling was introduced by 3rd generation operating systems
- Spooling-Simultaneous Peripheral Operations Online

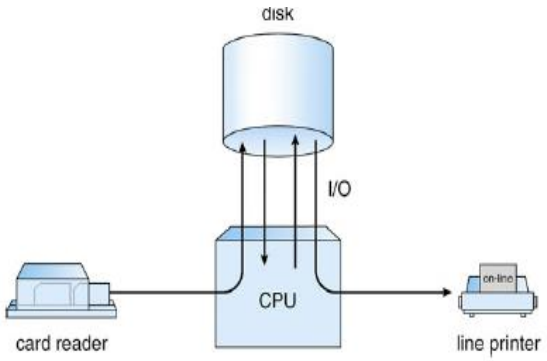
- Problem: slow I/O ties up fast CPU
  - Input ➡ Compute ➡ Output
  - Slow punch card reader and line printer



- They loaded the programs from cards to disk, and whenever a job is finished and memory partition was available, it is loaded directly to memory.

- Idea: overlap one job's I/O with other job's Compute

- OS functionality
  - ✓ buffering, DMA, interrupts
- Good: better utilization/throughput



- Bad: still not interactive i.e. whenever there is an error, programmers would realize it after a couple of hours.

- The CPU is **time** multiplexed among several jobs that are kept in memory and on disk (the CPU is allocated to a job only if the job is in memory).
  - ✓ **A "job" could now be a user ==> many users on a single machine**
- A job swapped in and out of memory to the disk **- to free up memory for a higher priority job, or if it has a long wait for I/O to complete.**
- **Comment: A program in "some state of execution" (having been already loaded in memory, but not completed) is known as a *process*.**
- On-line communication between the user and the system is provided; when the operating system finishes the execution of one command, it seeks the next "control statement" from the user's keyboard.
- On-line system must be available for users to access data and code **- presents an interactive user interface - not just batch.**
- **Examples: CTSS, Multics, Unix etc.**

- MULTICS, another third generation OS introduced the concept of client server computing and influenced other OS
- MULTICS (father of all modern OS) led to the development of UNIX
- UNIX (1970, ATT) became popular with companies and government agencies, and people started to develop their own UNIX OS
- IEEE developed a standard for UNIX, called POSIX to prevent chaos.
- POSIX defined a standard set of system call interface that conformant UNIX systems should support.
- Tanenbaum wrote a version of UNIX called MINIX with POSIX support for educational use.
- A Finnish student Linus Torvalds wrote a free production of MINIX called Linux

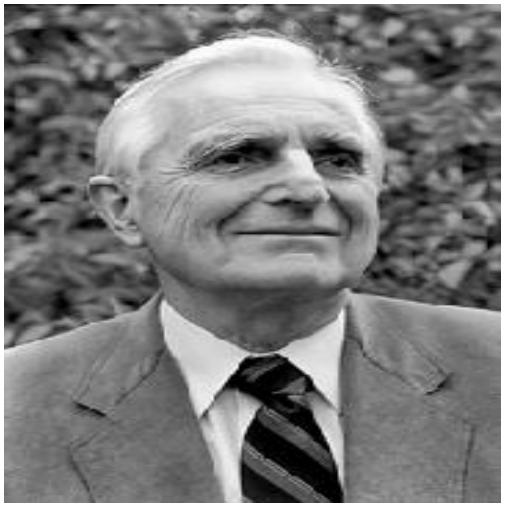
- Fourth generation 1980 – present
  - ✓ personal computers were developed after LSI (Large Scale Integration) circuits were invented.
  - ✓ First Microcomputer:
    - ❑ Intel 8080 CPU + attached 8-inch floppy disk
    - ❑ First disk-based OS CP/M (Control Program for Microcomputers)
  - ✓ In 1980s IBM designed the IBM PC and contacted Bill Gates for an operating System

- **“There is no reason anyone would want a computer in their home.“**
  - ✓Ken Olson, president, chairman and founder of Digital Equipment Corp. - 1977

- *Personal computers* – computer system dedicated to a single user.
- I/O devices – keyboards, mice, display screens, small printers.
- User convenience and responsiveness.
- Can adopt technology developed for larger operating system' often individuals have sole use of computer and do not need advanced CPU utilization of protection features.
- May run several different types of operating systems (Windows, MacOS, UNIX, Linux)

- Bill Gates suggested IBM that they should look at CP/M (one of the most successful OS for microcomputers at that time, by Gary Kildall)
- The biggest mistake of all:
  - ✓ Kindall refused to sign a non-disclosure agreement
- IBM went back to Bill Gates and signed a contract with him to write an OS for their new home computer
- MS-DOS was based on QDOS, the "Quick and Dirty Operating System" written by Tim Paterson of Seattle Computer Products,
- QDOS was based on Gary Kildall's CP/M
- Microsoft bought the rights to QDOS for $50,000

- Early MS-DOS was very primitive but later versions included advanced features taken from UNIX
- The early OS for microcomputers were based on users typing in commands from the keyboard
- Doug Engelbart from SRI invented the Graphical User Interface (GUI) with windows, icons, menus, and mouse.



- **“We don't see Windows as a long-term graphical interface for the masses.“**
  - ✓ **A Lotus Software Development official, while demonstrating a new DOS version - 1989**

- Steve Jobs saw the value of GUI in a PC and developed the Apple computer with GUI (Apple Macintosh) in his garage.
- He is also the co-founder of Pixar which has created very successful animated films: Toy Story ; A Bug's Life; Toy Story 2; Finding Nemo; Monsters.

- Microsoft developed a GUI-based system called Windows which originally ran on top of MS-DOS (just as a GUI to DOS)
- Windows 95 and 98 were real GUI based operating systems still based on 16-bit Intel assembly language
- Windows NT is 32-bit rewrite from scratch version of Windows 98.
- Windows 2000 (Windows NT version 5.0), Windows Me.


Windows has constantly evolved over time, changing the way we interact with our computers, and each other.
To celebrate the release of Windows 10, we're highlighting some of best parts of the most familiar versions of Windows. Enjoy!
Scroll to start

- SISD – Single Instruction, Single data stream
  - ✓ **Basic Uniprocessor – single program counter**

- SIMD - Single Instruction, Multiple data stream
  - ✓ **A logically single stream of instructions operating on different units of data in parallel – ex. A vector processor**
  - ✓ **Example of an implementation: a single stream of SIMD instructions from a single program counter in a special SISD host processor are *broadcasted* to many parallel SIMD processors each with its own registers and cache memory. Each of the SIMD processors now executes the same instruction on a different unit of data in parallel lock step synchronism. Example: the CM-2 "Supercomputer" with 65,563 processors, each having a 1 bit ALU (32 way bit slicing?)**

- MISD - Multiple Instruction, Single data stream – sequence of different data broadcasted to different parallel processors, each executing a different instruction sequence.
  - ✓ **Not ever implemented**.

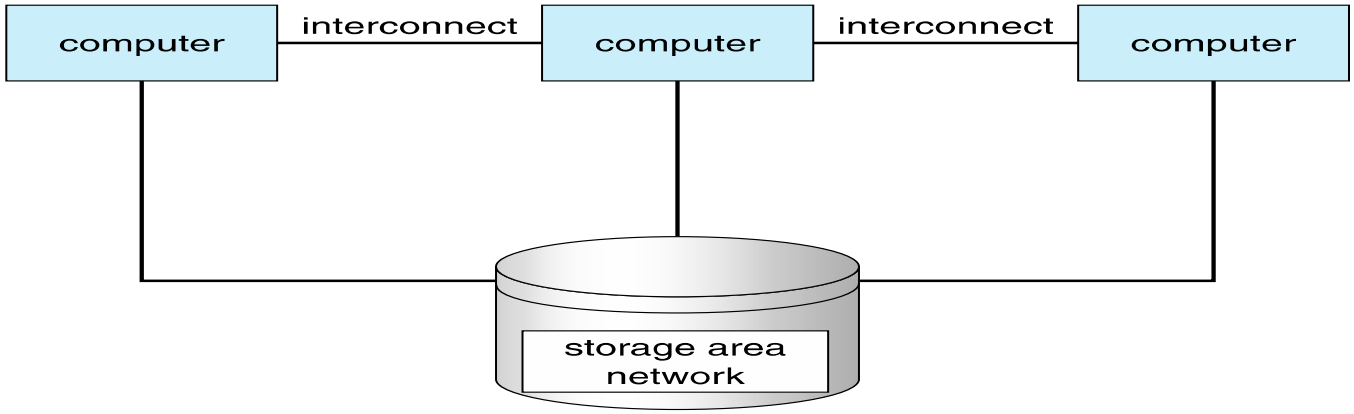
- MIMD - Multiple Instruction, Multiple data stream – many parallel processors executing different instruction streams on different data items.
  - ✓ **Commonly implemented with "loosely couples" clusters of general-purpose computers on a network and also tightly coupled SMP.**

- Multiprocessor systems with more than on CPU in *close* communication.
- ***Tightly coupled system*** – processors share memory and a clock; communication usually takes place through the shared memory **– via a bus -** .
- Advantages of parallel **(*Tightly Coupled*** **)** systems:
  - ✓ Increased *throughput*
  - ✓ **Increased computing power (speed-up factor)**
  - ✓ Economical
  - ✓ Increased reliability
    - ❑ graceful degradation
    - ❑ fail-soft systems
- **OS functions for multiprocessor systems are significantly more complex**

- ***Shared memory – tightly coupled schemes: SMP and Asymmetric***
- *Symmetric multiprocessing (SMP)*
  - ✓ **The common kernel in shared memory could operate on any processor – process/thread parallism on kernel execution possible – processors are peers – no master/slave.**
  - ✓ Many processes can run at once without performance deterioration **- true parallelism vs pseudo parallelism of a multitasking system on a uniprocessor.**
  - ✓ Most modern operating systems support SMP
- *Asymmetric multiprocessing*
  - ✓ Each processor is assigned a specific task; master processor schedules and allocated work to slave processors.
  - ✓ More common in extremely large systems

    Problem with parallelism: how do you distribute a problem across multiple processes to capture the max potential of the system? Can all algorithms be "parallelized"? Are there theoretical limits to parallelizing?
  - ✓ **Example logic simulators - a natural for parallelism.**

- **Alternative to SMP**
- **Goal is high reliability, availability, and performance.**
- **A group of interconnected, "whole" computers working together as a unified computing resource that can create the illusion of being a single machine.**
- Clustering allows two or more systems to share **(secondary?)** storage – example RAID disks.
- *Asymmetric clustering*: **multiple servers runs the application while one server stands by - monitor.**
- *Symmetric clustering*: all N hosts are running the application **mutual monitoring - no single monitor**



- Network OS is a collection of software and associated protocols which runs on set of interconnected autonomous computers.
- Loosely coupled system – each computer has its own local memory; processors communicate with one another through various communications lines, such as high-speed buses , cross-bar switches, LANS, or telephone lines.
  - ✓ **Could be a heterogeneous mixture of independent computers having different characteristics etc. all connected on by some network *fabric*.**
- Advantages of Network systems.
  - ✓ Resources Sharing
  - ✓ Communications
- Disadvantages of Network systems.
  - ✓ Lack of transparency
  - ✓ No fault tolerance
  - ✓ Requires complete path and remote login

- Examples: BSD,UNIX, Windows-NT, MS-LAN Manager etc.

- Distribute the computation among several physical processors.
- *Loosely coupled system* **(clusters?)** – each processor has its own local memory; processors communicate with one another through various communications lines, such as high-speed buses **, cross-bar switches, LANS,** or telephone lines.
  - ✓ **Could be a heterogeneous mixture of independent computers having different characteristics etc. all connected on by some network *fabric*.**
- Advantages of distributed systems.
  - ✓ Resources Sharing
  - ✓ Computation speed up – load sharing
  - ✓ Reliability
  - ✓ Communications
- **Disadvantages: control and OS functions complicated and distributing an algorithm over the nodes is difficult**.

- Requires networking infrastructure.
- Local area networks (LAN) or Wide area networks (WAN)
- May be either client-server or peer-to-peer systems.
  - ✓ **Clients generate requests to be satisfied by the server – server performs computation with results sent to client**
  - ✓ **Peer-to-peer example: Internet or a master and many slaves on a network or switch.**
- **Examples: Amoeba, Chorus, Angle, Alpha kernel etc.**

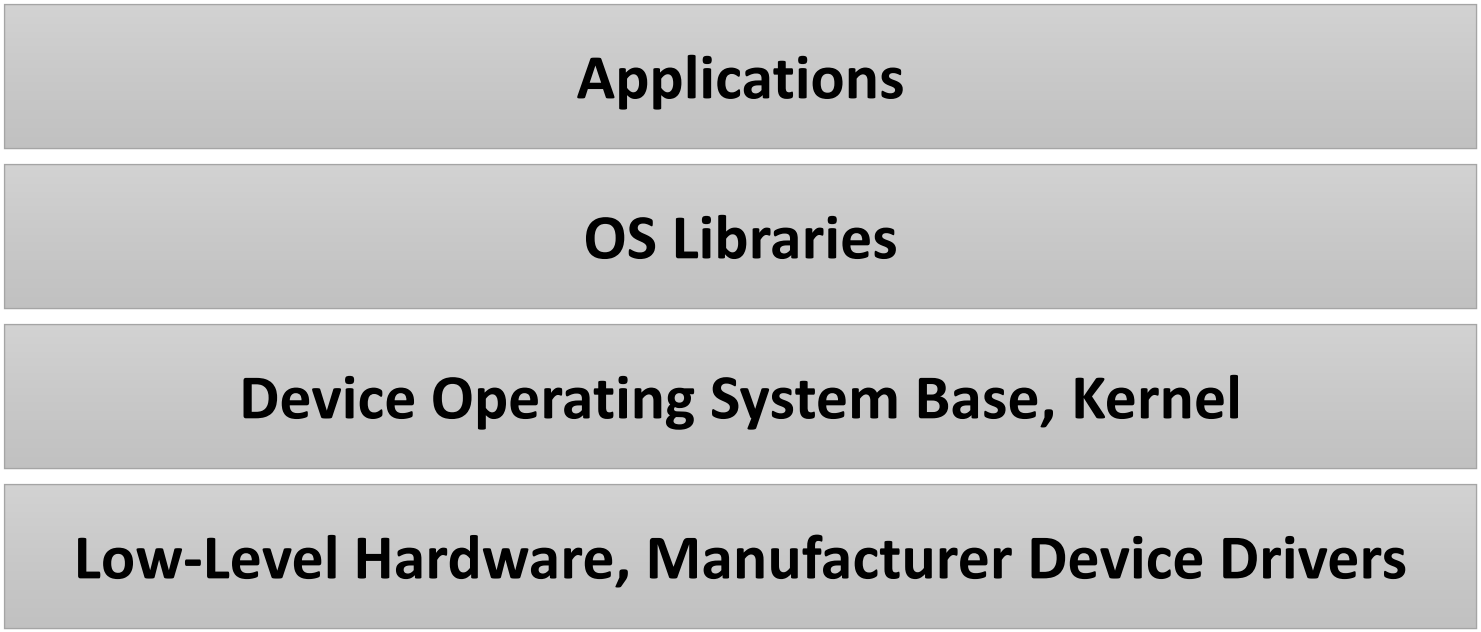
- Often used as a control device in a dedicated application such as controlling scientific experiments, medical imaging systems, industrial control systems, and some display systems.
- Well-defined fixed-time constraints.
- Real-Time systems may be either *hard* or *soft* real-time.
- **Examples: Harmony, MARUTI, HART, RTEM, VRTX etc.**

- Hard real-time:
  - ✓ Secondary storage limited or absent, data stored in short term memory, or read-only memory (ROM)
  - ✓ Conflicts with time-sharing systems, **(delays unpredictable), thus** not supported by general-purpose operating systems.
    Uses deadline scheduling of tasks
- Soft real-time
  - ✓ Limited utility in industrial control of robotics
  - ✓ Useful in applications (multimedia, virtual reality) requiring advanced operating-system features.
  - ✓ **Cannot guarantee deadlines but can guarantee highest priority for Soft RT tasks over ordinary tasks.**

- Personal Digital Assistants (PDAs)
- Cellular telephones
- Issues:
  - ✓ Limited memory
  - ✓ Slow processors
  - ✓ Small display screens.

- Design and capabilities of a Mobile OS (Operating System) is very different than a general-purpose OS running on desktop machines:
  - ✓ mobile devices have constraints and restrictions on their physical characteristic such as screen size, memory, processing power etc.
  - ✓ Scarce availability of battery power
  - ✓ Limited amount of computing and communication capabilities
- Thus, they need different types of operating systems depending on the capabilities they support. e.g. a PDA OS is different from a Smartphone OS.

- A mobile OS is a software platform on top of which other programs called application programs, can run on mobile devices such as PDA, cellular phones, smartphone and etc.

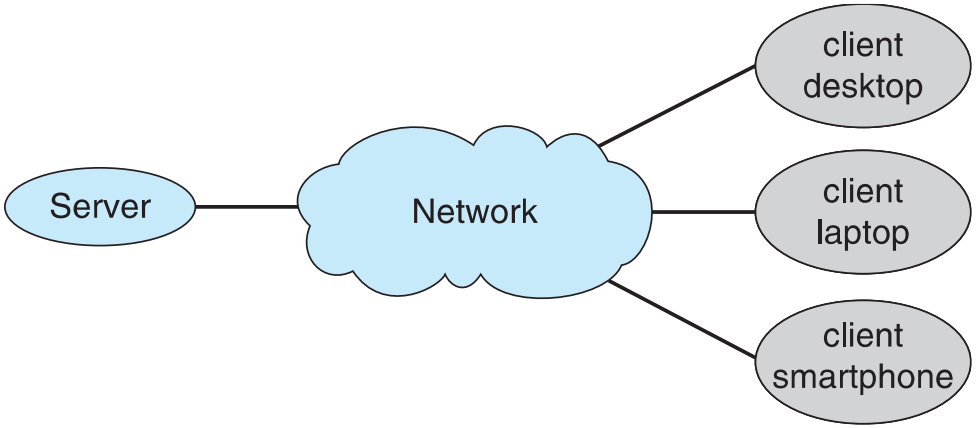


- There are many mobile operating systems. The followings demonstrate the most important ones:
  - ✓Java ME Platform
  - ✓Palm OS
  - ✓Symbian OS
  - ✓Linux OS
  - ✓Windows Mobile OS
  - ✓BlackBerry OS
  - ✓iPhone OS
  - ✓Google Android Platform

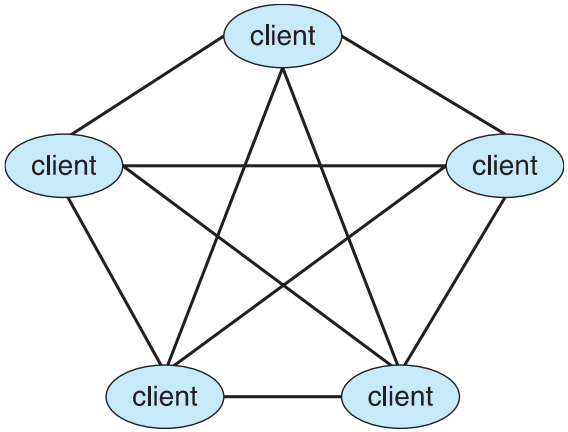
- Stand-alone general-purpose machines
- But blurred as most systems interconnect with others (i.e., the Internet)
- **Portals** provide web access to internal systems
- **Network computers** (**thin clients**) are like Web terminals
- Mobile computers interconnect via **wireless networks**
- Networking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

- Distributed computing
  - ✓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
    - ❑Communication scheme allows systems to exchange messages
    - ❑Illusion of a single system

- Client-Server Computing
  - ✓ Dumb terminals supplanted by smart PCs
  - ✓ 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

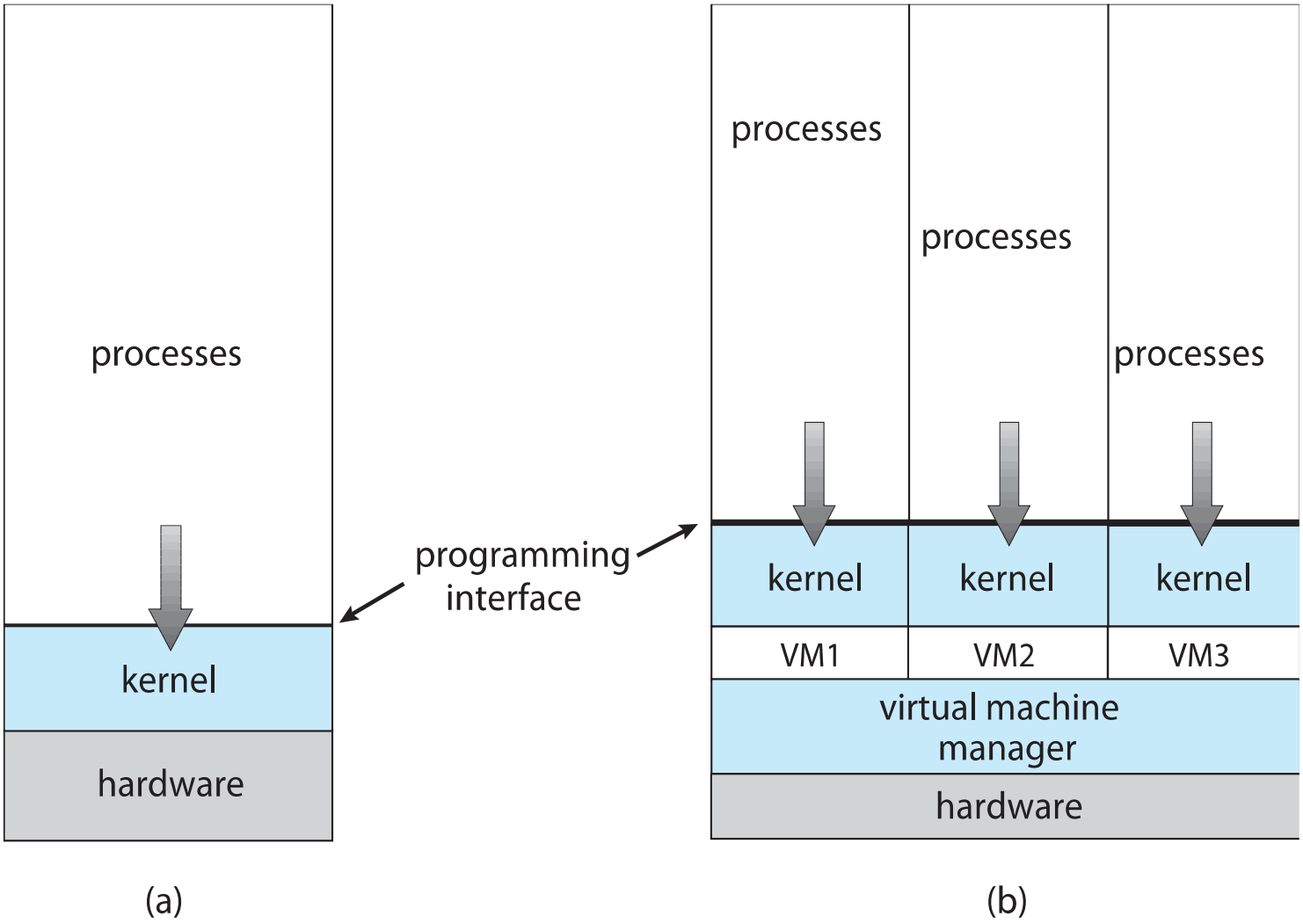


- Another model of distributed system
- P2P does not distinguish clients and servers
  - ✓Instead all nodes are considered peers
  - ✓May each act as client, server or both
  - ✓Node must join P2P network
    - ❑Registers its service with central lookup service on network, or
    - ❑Broadcast request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



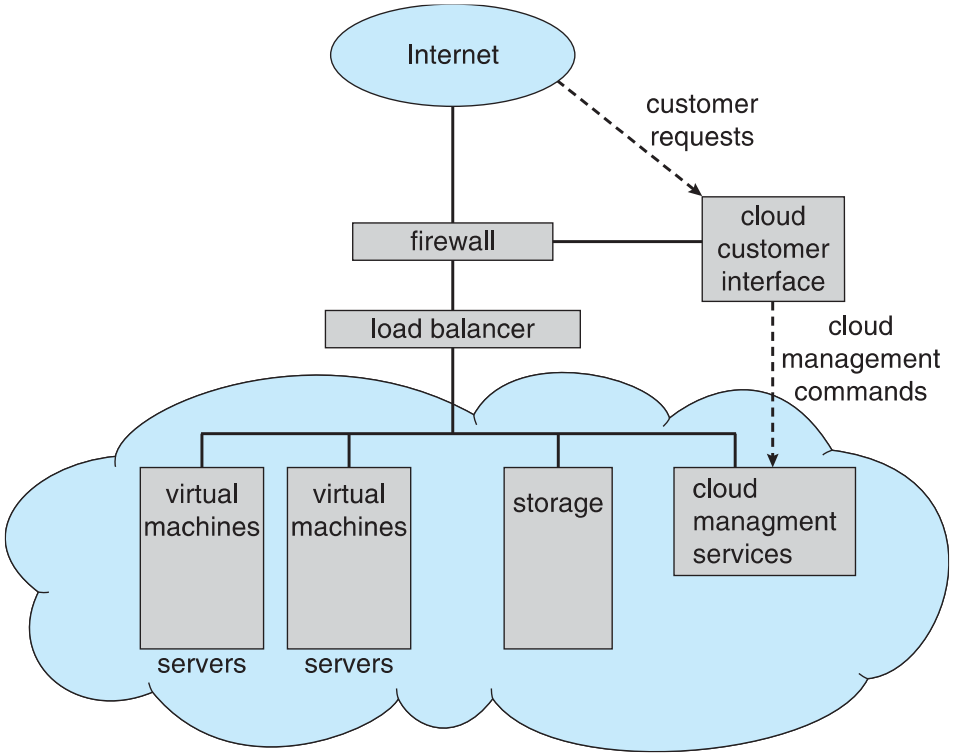
- Allows operating systems to run applications within other OSes
  - ✓Vast and growing industry
- **Emulation** used when source CPU type different from target type (i.e. PowerPC to Intel x86)
  - ✓Generally slowest method
  - ✓When computer language not compiled to native code – **Interpretation**
- **Virtualization** – OS natively compiled for CPU, running **guest** OSes also natively compiled
  - ✓Consider VMware running WinXP guests, each running applications, all on native WinXP **host** OS
  - ✓**VMM** (virtual machine Manager) provides virtualization services

- Use cases involve laptops and desktops running multiple OSes for exploration or compatibility
  - ✓Apple laptop running Mac OS X host, Windows as a guest
  - ✓Developing apps for multiple OSes without having multiple systems
  - ✓QA testing applications without having multiple systems
  - ✓Executing and managing compute environments within data centers
- VMM can run natively, in which case they are also the host
  - ✓There is no general-purpose host then (VMware ESX and Citrix XenServer)


processes
kernel
hardware
programming interface
processes
processes
processes
kernel
kernel
kernel
VM1
VM2
VM3
virtual machine manager
hardware
(a)
(b)

- Delivers computing, storage, even apps as a service across a network
- Logical extension of virtualization because it uses virtualization 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
  - ✓ **Public cloud** – available via Internet to anyone willing to pay
  - ✓ **Private cloud** – run by a company for the company's own use
  - ✓ **Hybrid cloud** – includes both public and private cloud components
  - ✓ 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)

- Cloud computing environments composed of traditional OSes, plus VMMs, plus cloud management tools
  - ✓Internet connectivity requires security like firewalls
  - ✓Load balancers spread traffic across multiple applications



- Handheld smartphones, tablets, etc
- What is the functional difference between them and a “traditional” laptop?
- Extra feature – more OS features (GPS, gyroscope)
- Allows new types of apps like ***augmented reality***
- Use IEEE 802.11 wireless, or cellular data networks for connectivity
- Leaders are **Apple iOS** and **Google Android**

Real-Time Embedded Systems

- Real-time embedded systems most prevalent form of computers
  - ✓ Vary considerable, special purpose, limited purpose OS, **real-time OS**
  - ✓ Use expanding
- Many other special computing environments as well
  - ✓ Some have OSes, some perform tasks without an OS
- Real-time OS has well-defined fixed time constraints
  - ✓ Processing ***must*** be done within constraint
  - ✓ Correct operation only if constraints met

Bibliography

- Silberschatz, A, Galvin, P.B, and Gagne, G., Operating System Principles, 9e, John Wiley & Sons, 2013.
- Stallings W., Operating Systems-Internals and Design Principles, 7e, Pearson Education, 2014.
- Harvey M. Deital, “Operating System”, Third Edition, Pearson Education, 2013.
- Andrew S. Tanenbaum, “Modern Operating Systems”, Second Edition, Pearson Education, 2004.
- Gary Nutt, “Operating Systems”, Third Edition, Pearson Education, 2004.
- https://computerhistory.org/

Acknowledgements

- I have drawn materials from various sources such as mentioned in bibliography or freely available on Internet to prepare this presentation.
- I sincerely acknowledge all sources, their contributions and extend my courtesy to use their contribution and knowledge for educational purpose.

Thank You!!

?