CS3.301: Operating Systems and Networks

P. Krishna Reddy
E-mail: pkreddy@iiit.ac.in
http://www.iiit.ac.in/~pkreddy
Office: Data Sciences and Analytics Center

Outline

- **Introduction**
  - **What is an Operating System ?**
- Course topics and grading
- History, development and concepts of Oss
- Different kinds of Computer Systems
- Concept of virtual computer

Questions

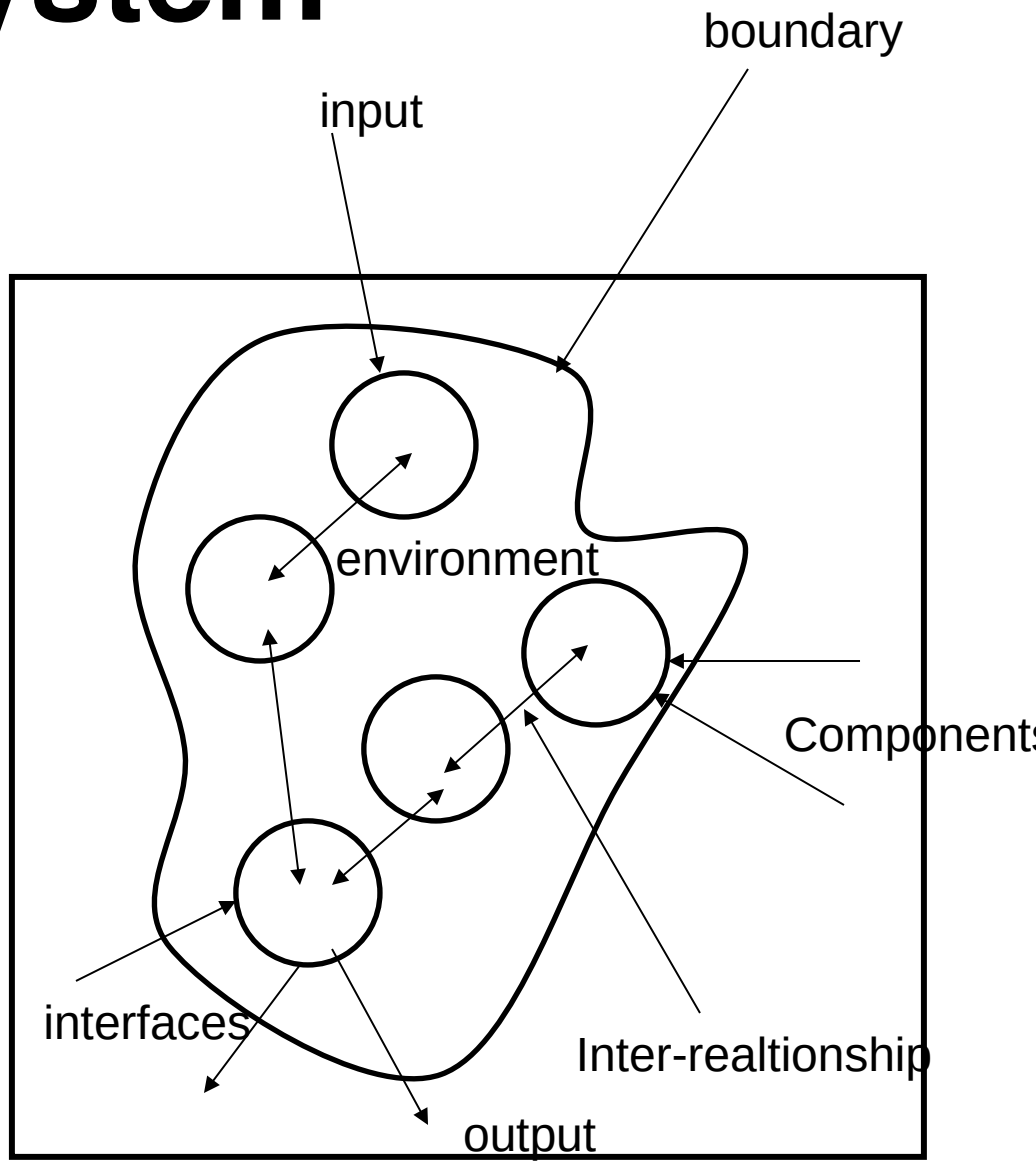
- What is a system ?
- What is an operating system ?
- What is a computer operating system ?

What is a system ?

- A system is an inter-related set of components with an identifiable boundary working together for some purpose.
- System can be natural or fabricated
  - Natural systems: human body or solar system
  - Fabricated systems: cycle, bus, computer, government, boat

System

- A system has nine characteristics
  - Components
  - Inter-related components
  - A boundary
  - A purpose
  - An environment
  - Interfaces
  - Input
  - Output
  - Constraints.



A general depiction of a system

Characteristics…

- Components:
  - A system is made up of components
  - A component is either an irreducible part or aggregation of parts that make-up a system. A component is also called a sub-system.
- Interrelated:
  - The components of interrelated
  - Dependence of one subsystem on one or more subsystems.

Characteristics…

- Boundary (Scope):
  - A system has a boundary, within which all of its components are contained and which establishes the limits of a separating the system from other systems.
- Purpose
  - The overall goal of function of a system. The system's reason for existing.

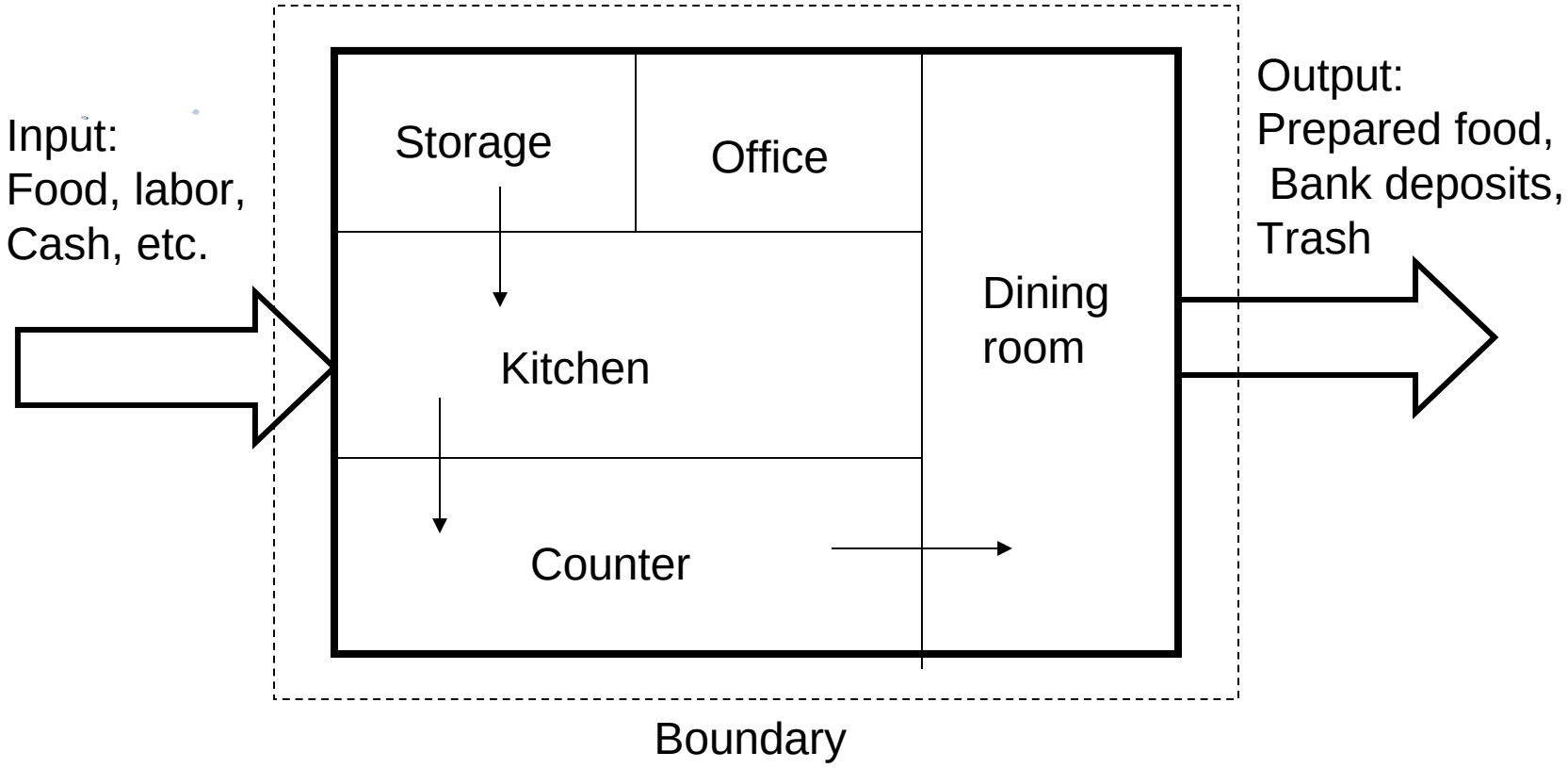
Characteristics…

- Environment
  - Everything external to the system that interacts with the system.
- Interface
  - Point of contact where a system meets its environment or subsystems meet each other.
- Constraint:
  - A limit what a system can accomplish: Capacity, speed or capabilities.

Characteristics…

- Input
  - Whatever a system takes from its environment in order to fulfill its purpose.
- Output:
  - Whatever a system returns to its environment in order to fulfill its purpose.

Example: A fast food restaurant as a system



Environment: Customers, food distributors, banks etc

Represents an inter-relationship

Constraints: Popular foods, Health dept., constraints of storage

Important System Concepts

- Decomposition
- Modularity
- Coupling
- Cohesion

Decomposition (Divide and Conquer)

- It deals with being able to break down a system into its components.
- Decomposition results in smaller and less complex pieces that are easier to understand than larger, complex pieces.
- Decomposing a system also allows to focus on one particular part of a system, making easier to think of how to modify that part independently of the entire system.

Modularity

- Modularity refers to dividing a system up into chunks or modules of a relatively uniform size.
- You can replace or add any other module (or a component) without effecting the rest of the system.
- It is a design strategy in which system is composed of relatively small and autonomous routines fit together.

Coupling

- Coupling is the extent to which subsystems are dependent on each other.
- Subsystems should be as **independent** as possible.
- If a subsystem fails and other subsystems are highly dependent on it, the others will either fail themselves or have problems in functioning.

Cohesion

- The extent to which a system or a subsystem performs a single function.

What is an operating system ?

- Operating system is a system.
- Operating system is a subsystem of any tool.
- Each tool constitutes machine part and operating part.
- The operating part of a tool is called as operating system of that tool.
- The purpose of operating system is to facilitate the operation of the underlying machine or tool.
- For some tools, operating system may not exist.
  - Example: Pen.
- For a user, the operating system abstracts the machine part in terms of simple services by hiding the details of the machine. The OS can provide services to users or other subsystems.
- Examples of typical operating systems:
  - Car operating system, Telephone operating system, TV operating system and so on.

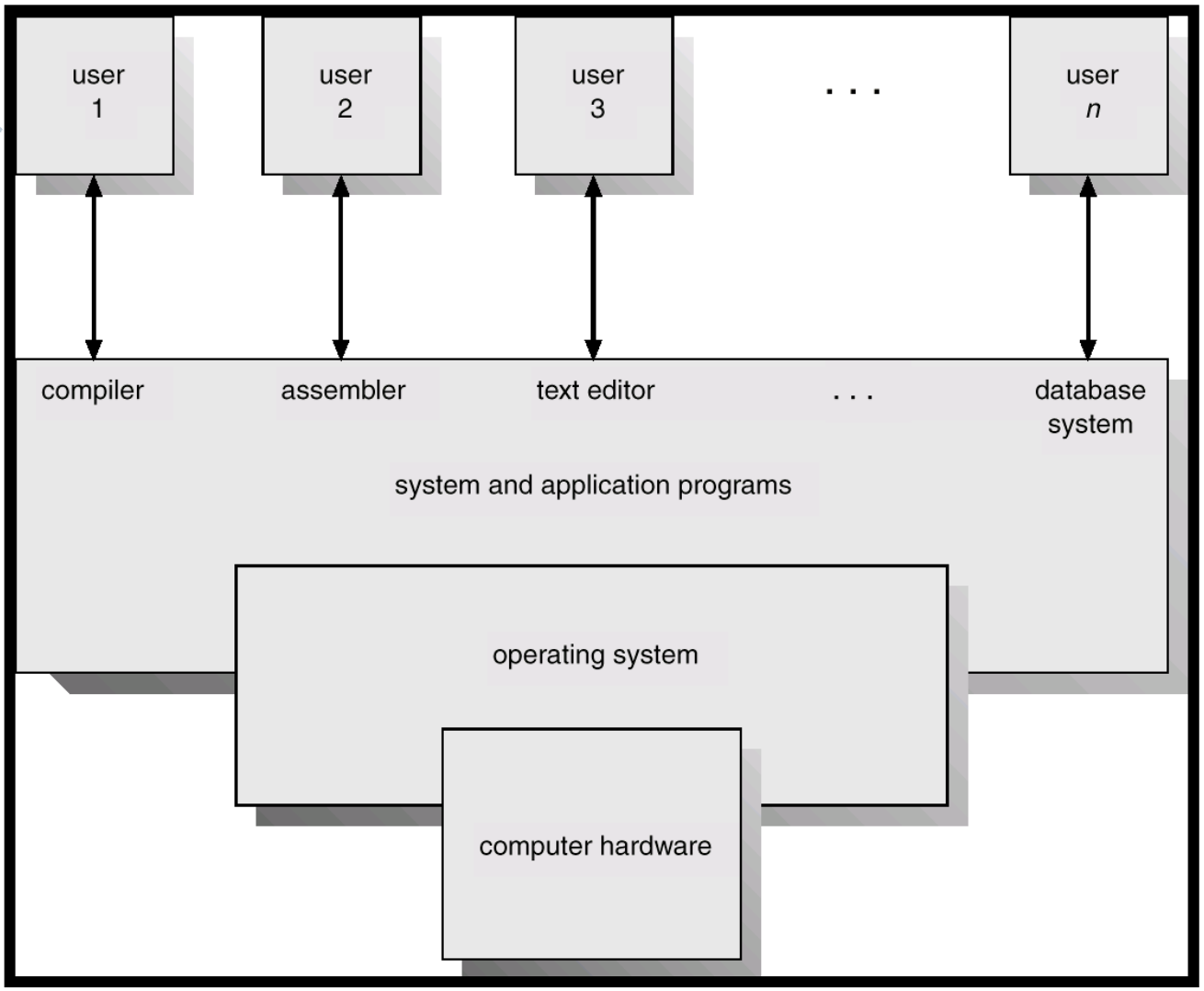
What is a computer operating system ?

- A computer is also a tool that contains the machine part and the operating part.
- The operating part of a computer is called Computer Operating System.
- the operating system abstracts the underlying hardware in terms of simple services by hiding the details of the hardware. The OS can provide services to users or other subsystems.
- Examples of Computer operating systems:
  - WINDOWS 10, Macintosh, UNIX, SOLARIS, LINUX, Android, MAC IOS and so on.
- In the rest of this course, operating system means computer operating system.

Computer System Components

1. **Hardware** – provides basic computing resources (CPU, memory, I/O devices).
2. **Operating system** – controls and coordinates the use of the hardware among the various application programs for the various users.
3. **Applications programs** – define the ways in which the system resources are used to solve the computing problems of the users (compilers, database systems, video games, business programs), Internet.
4. **Users** (people, machines, other computers).

Abstract View of System Components



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:
  - Make the computer system convenient to use.
  - Use the computer hardware in an efficient manner.

Operating System Definitions…

- **Resource allocator** – manages and allocates resources.
  - Resources: CPU time, Memory Space, file storage space, I/O devices and son on.
- **Control program** – controls the execution of user programs and operations of I/O devices .
- **Kernel** – the one program running at all times (all else being application programs).
- The two goals, efficiency and convenience are sometimes contradictory
- Much of OS theory is concentrates on optimal use of resources.

Outline

- Introduction
  - What is an Operating System ?
- **Course topics and grading**
- History, development and concepts of Oss
- Different kinds of Computer Systems
- Concept of virtual computer

Objectives

- Understand the operational part of any computer.
  - learn the important concepts which have been evolved for building **modern operating systems and networking protocols.**
- Understanding the general principles of OS design.
  - Focus on general-purpose, multi-user systems.
  - Emphasis on widely applicable concepts rather than any specific features of any specific OS.
- Understanding problems, solutions and design choices.
- Understanding the structure of specific OSs: UNIX, LINUX, WINDOWS 10

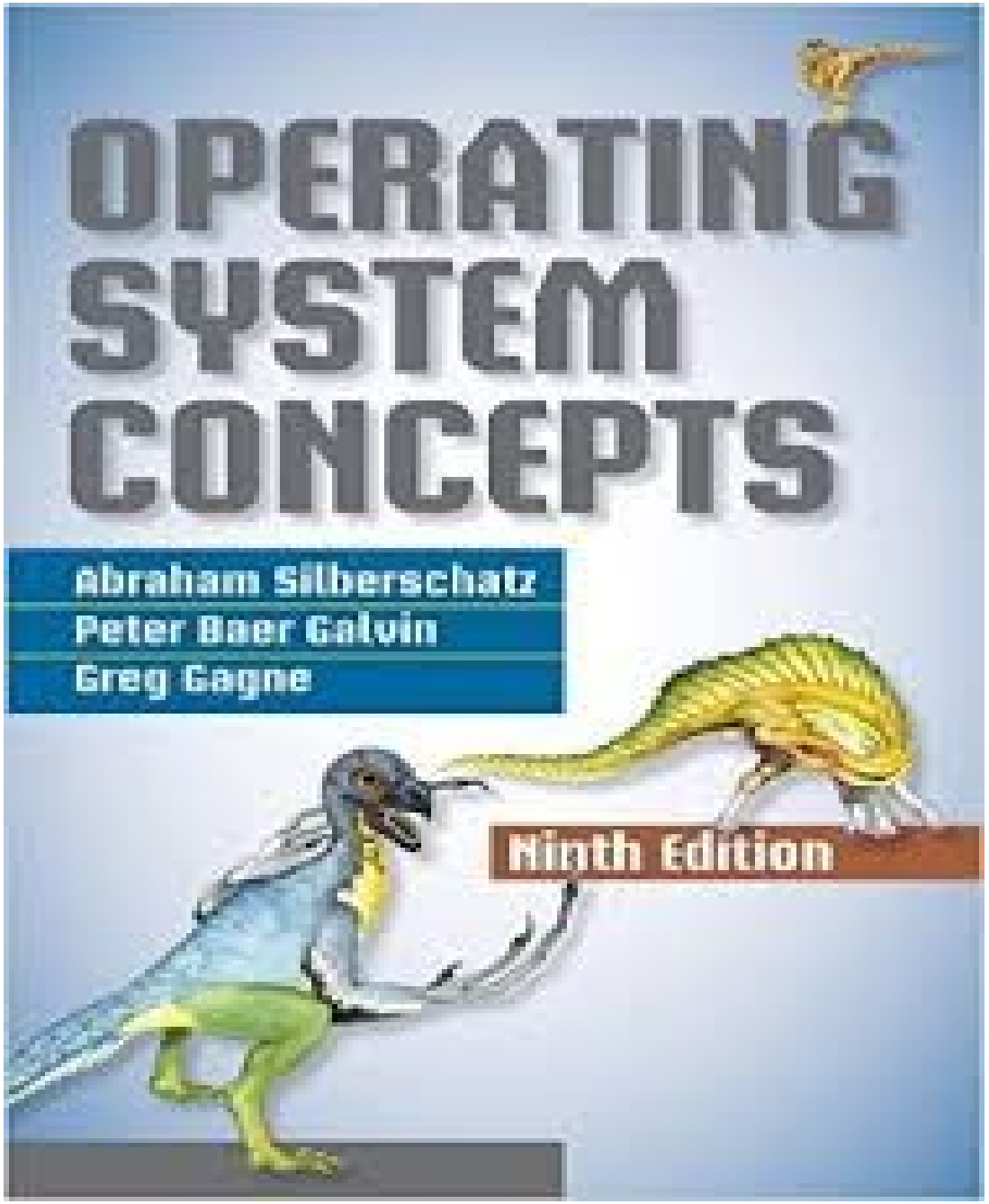
Course topics

- Introduction (1 week)
- Process and thread management (2 weeks)
- CPU Scheduling (1 week)
- Process Synchronization (2 weeks)
- Deadlocks (1 week)
- Memory management (1.5 weeks)
- Virtual Memory (1.5 weeks)
- File Systems (1 week)
- Protection and Security (1 week)
- Networking (3 weeks)

References

- Text book:
  - **Silberschatz, A, Galvin, P, Gagne, G. Operating System Concepts, Addison-Wesley (8th or latest edition).**
  - **Computer Networks (5th Edition) Andrew S. Tanenbaum, David J. Wetherall Prentice Hall.**

- Other Reference BOOKS:
  - William Stallings, Operating systems, Prentice-Hall, 1998.
  - Operating Systems, Gary Nutt, Pearson Education
  - Charles Crowley, Operating Systems: A design-oriented approach, Tata McGraw-Hill, 1997.
  - Operating Systems: Concepts and Design, Milan Milenkovic, TATA McGRAW-HILL
  - Tanenbaum, A., Modern Operating Systems, Prentice-Hall, second edition, 2000.


OPERATING
SYSTEM
CONCEPTS
Abraham Silberschatz
Peter Baer Galvin
Greg Gagne
Ninth Edition

FIFTH EDITION

COMPUTER NETWORKS

TANENBAUM | WETHERALL

LAB WORK

- Experiments will be on the exposing the working of several system calls of LINUX OS:
  - Installation
  - reversing a file
  - Shell writing
  - Process communication
  - Bounded buffer,
  - semaphores,
  - shared memory,
  - threads;
  - Replace "ls" with lookup;
  - Command line for /proc;
  - Memory management

- The lab is very intensive. Please do not ask for the extension of deadline. Each experiment will be evaluated.

Related Research Papers

1. H. M. Levy and P. H. Lipman. Virtual Memory Management in the VAX/VMS Operating Systems. IEEE Computer, 15(3), March 1982, pp. 35-41.

2. Thompson, K., "UNIX Implementation," The Bell System Technical Journal, Vol. 57, No. 6 (July-August 1978), Part 2, pp. 1931-1946.

3. F. J. Corbato and V. A. Vyssotsky, "Introduction and overview of the Multics system," In Proceedings AFIPS 1965 Fall Joint Computer Conference (FJCC), Vol. 27, No. 1, 1965, Spartan Books: New York, pp. 185-196

4. Windows NT and VMS: The Rest of the Story, by Mark Russinovic

5. E. W. Dijkstra, "The Structure of the THE ##Multiprogramming System," Communications of the ACM, Vol. 11, No. 5, May 1968, pp. 341-346

6. C. Daley and J. B. Dennis. Virtual Memory, Processes, and Sharing in MULTICS. Communications of the ACM, 11(5), May 1968, pp. 306-312.

7. Ritchie, D.M., and Thompson, K., "The UNIX Time-Sharing System,"The Bell System Technical Journal, Vol. 57, No. 6 (July-August 1978), Part 2, pp. 1905-1929.

Reading/Practicing Assignments

- Problems will be given
- You have to solve on your own

OUTCOME

- After completing the course, the students will understand
  - fundamental concepts of several computer operating systems (SOLARIS, LINUX, WINDOWS, MAC, Adroid,…) and network based services (Skype, Google Hangouts,..)
  - the solutions/options to interesting problems which have been encountered by the designers of the preceding operating systems and networking protocols, and
  - the critical role of the operation system in designing several computer and network based systems like database systems, expert systems, web based information systems, multi-media systems, embedded systems, internet services and so on.

GRADING

Type of Evaluation	Weightage (in %)
Quizes (about 10)	40%
End Sem Exam	30%
Assignments (programming)	30%

Outline

- Introduction
  - What is an Operating System ?
- Course topics and grading
- **History, development and concepts of Oss**
- Different kinds of Computer Systems
- Concept of virtual computer

Stored Program Computer

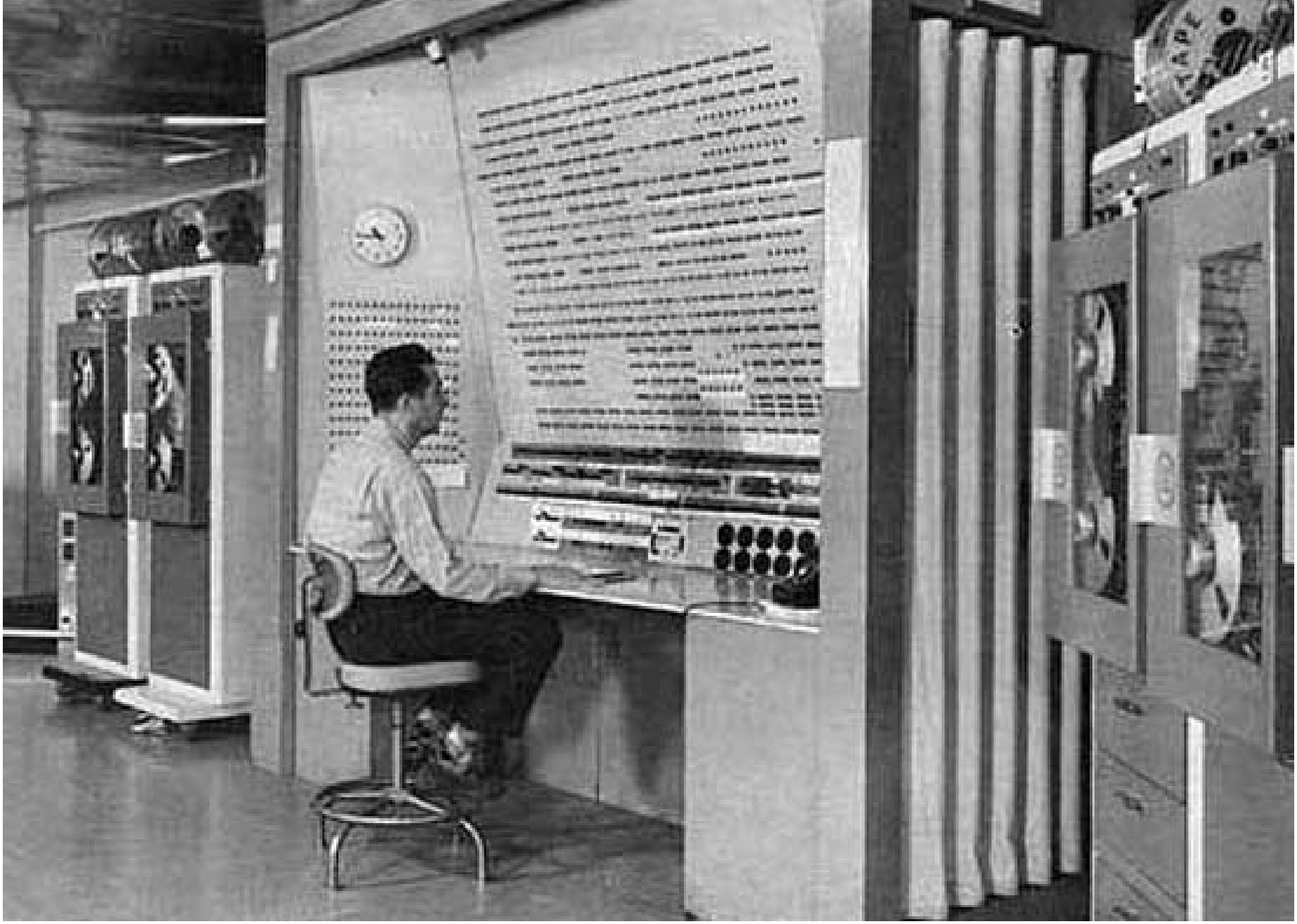
- 1940
  - Computers were designed to perform specific tasks.
  - Modification of the tasks required a great deal of effort and manual labour
- Alan Turing and John von Neumann
  - Proposed the concept of stored program computer.
  - **Machine has both program store and data store and program store provides instructions about what to do on data.**
  - This concept has generated the concept of general purpose computer.
- *Watch the movie*
  - *The Imitation Game*
- *1951*
  - *First general purpose computer*
    - *Machester Mark 1 (ran duriing 1940 to 1949)*
  - *First general purpose commercial computer was available in the market*
    - *Ferranti Mark1*

Early systems (Serial processing)

- **1940-50:**
  - The programmer interacted directly with the computer hardware.
  - Display light, switches, printer, card reader.
  - No OS.
  - Error is displayed through lights.
- **Problems:**
  - Scheduling → Users spend lots of time at the computer.
    - Signup sheet was used.
  - Job Setup time
    - Loading and compiling
      - Mounting and Un-mounting of tapes
      - Setting up of card desks
  - Libraries of functions, linkers, loaders, debuggers, and I/O driver routines were available for all the users.

Early Systems…

- Early computers were (physically) large machines run from a console.
- The programmer would operate the program directly from the console.
  - The program is loaded to the memory from panel of switches, paper tape, and from punched cards.
- As time went on, additional software and hardware were developed.
  - Card readers, line printers, and magnetic tape became common place.
  - Libraries, loaders, and common functions were created.
    - Software reusability.



Early Systems…

- The routines that performed I/O were especially became important.
- Device driver: A special subroutine was written for each I/O device.
  - A device driver knows how the buffers, flags, registers, control bits, and status bits for a particular device should be used.
  - Device driver is written once and called from the library.
- Later, compilers for FORTRAN, COBOL and other languages have appeared.

Early Systems…

- Significant amount of setup time.
- Each job consisted of many separate steps:
  - Loading the FORTRAN compiler tape
  - Running the compiler
  - Unloading the compiler tape
  - Loading of assembler tape
  - Running assembler
  - Unloading the assemble tape
  - Loading the object program
  - Running the object program
- If error occurred during any step, you have to start over at the beginning.

Early Systems..

- The setup time was a real problem
- CPU is idle while tapes are being mounted or the programmer was operating the console.
- In the early days, few computers were available and they were expensive (millions of dollars).
  - +operational costs: power, cooling, programmers.
- Main question:

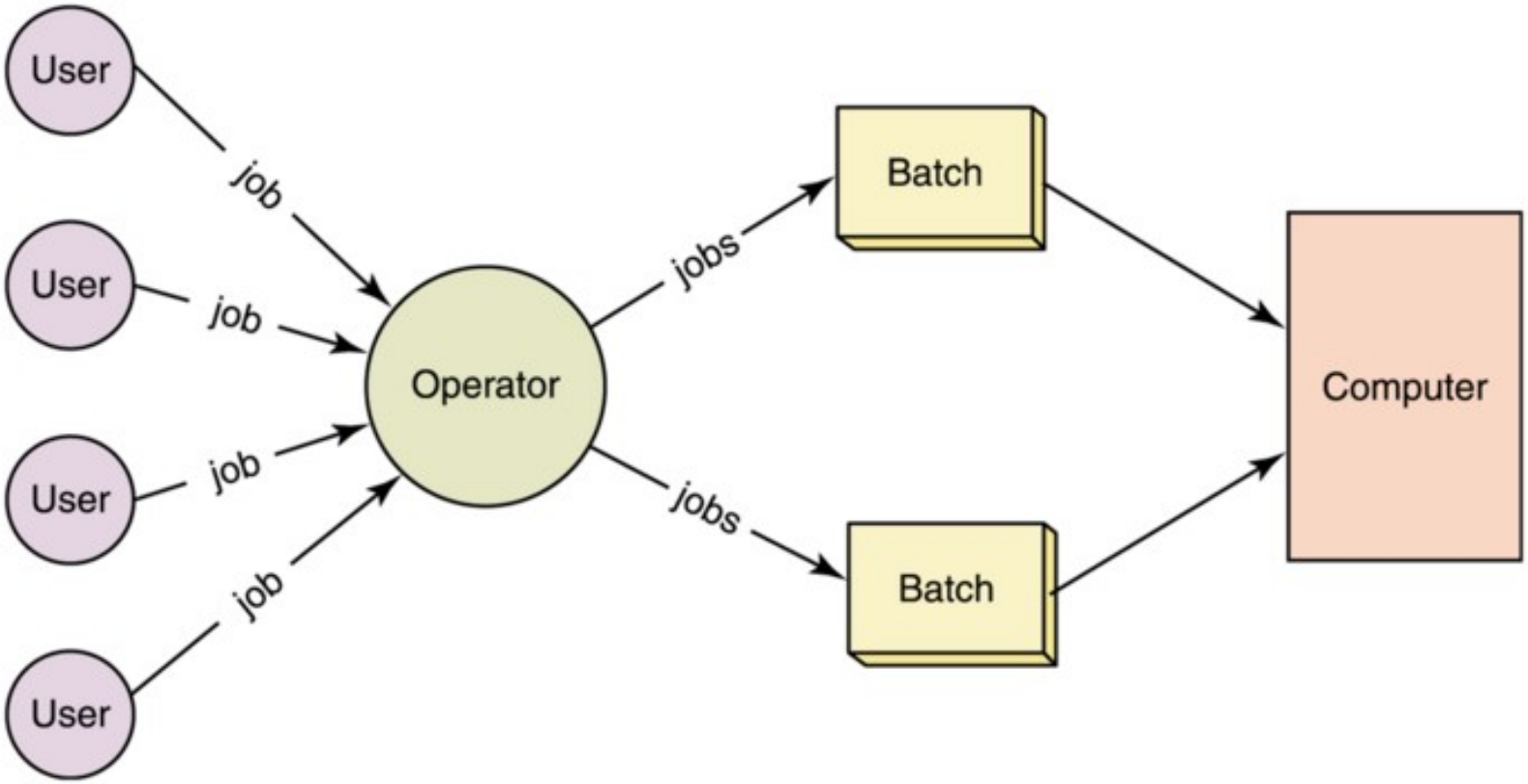
How to increase the utilization of CPU ?

Early Systems…

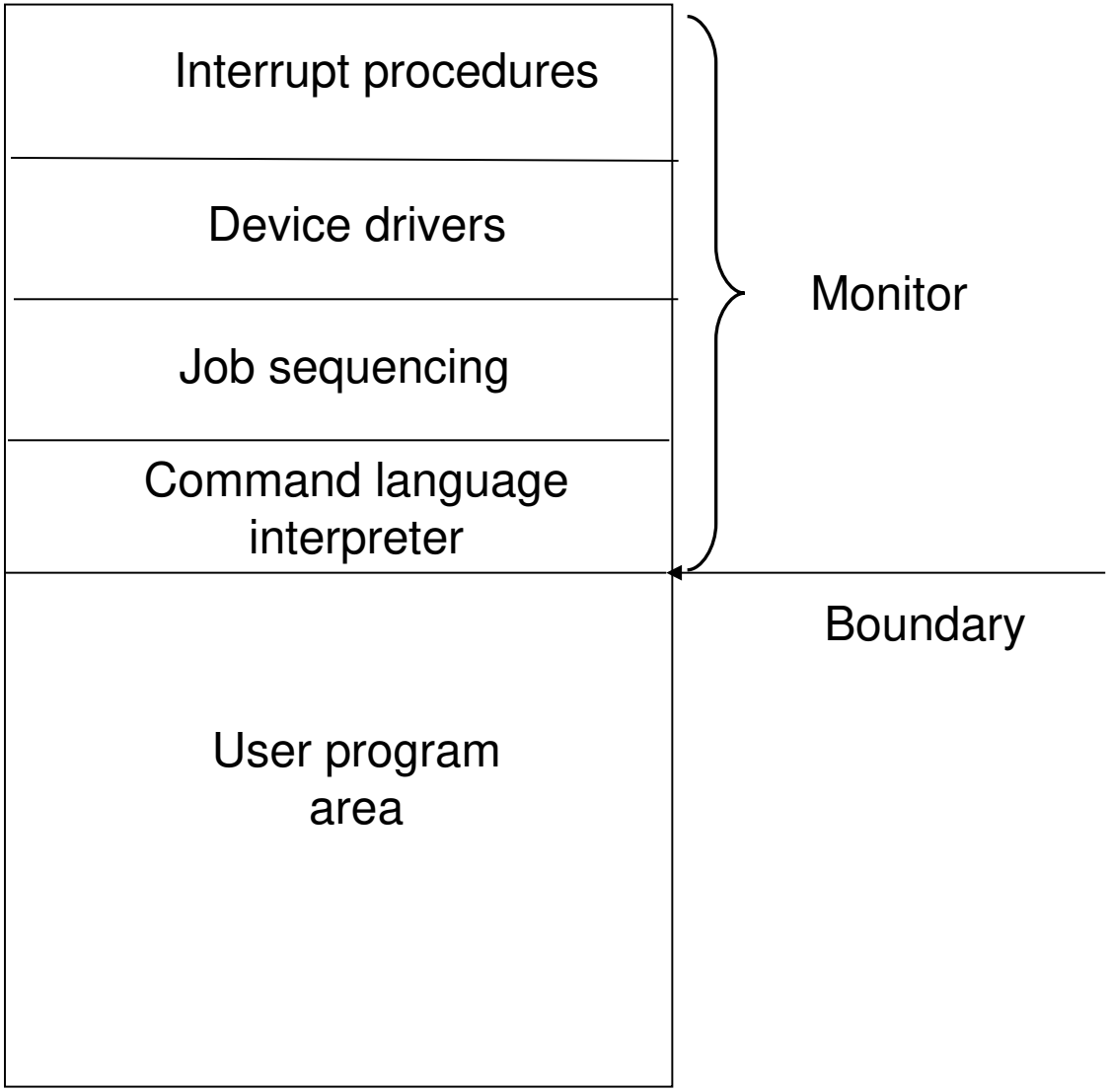
- The solution was two fold.
- First, a professional computer operator was hired.
  - Once the program was finished, he operator could start next job.
  - The operator sets up the job, produces the dump, and starts the next job.
  - The set up time was reduced due to operator's experience.
- Second, jobs with similar needs were batched together and run through the computer as a group.
  - For example, if there is a FORTRAN job, COBOL job, and FORTRAN job, two FORTRAN jobs were batched together.
- However, during transition time CPU sat idle.
- To overcome this idle time, people developed **automatic job sequencing**.
  - A first rudimentary OS was created
  - A small program called a **resident monitor** was developed.
  - The resident monitor always resided in memory.

Simple Batch Systems (early 1960s)

- In serial systems
  - Machines were very expensive
  - Wasting time was not acceptable.
- To improve usage, the concept of batch OS was developed.
- The main idea is the use of software known as **monitor**.
  - The user no longer has access to machine.
- The user submits the job (tape) to the operator.
- The operator batches the jobs together sequentially, places entire batch as an input device for use by the computer.


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

Memory Layout for a Simple Batch System



Resident monitor

Simple Batch Systems..

- At the beginning of any job, the corresponding subroutines and functions are loaded.
- The monitor reads the jobs one at a time from the input device.
- **ALGORITHM FOR MONITOR (or Operating System)**
  - The control is passed to the user's program.
    - Processor is fetching and executing user's instructions.
  - After completion, the control is returned to the monitor program
    - Processor is fetching and executing monitor instructions.

- Loop no operation
- JUMP loop

Features of Batch System

- The batch OS is simply a program.
- It relies on the ability of the processor to fetch instructions from various portions of main memory to seize and relinquish control.

- Hardware features:
  - **Memory protection**: While the user program is running, it must not alter the memory area containing the monitor.
    - If such is the case the processor hardware should detect the error and transfer control to monitor.
  - **Timer:** A timer is used to prevent the single job from monopolizing the system
  - **Privileged instructions**
    - Contains instructions that are only executed by monitor.
    - I/O instructions
    - If a program encounters them the control shifts through monitor..
  - **Interrupts:** It gives OS more flexibility.
    - Relinquishing control and regain control

Features of Batch System

- With batch OS, the machine time alters between execution of user programs and execution of monitor.
- Two overheads
  - Machine time is consumed by the monitor.
  - Memory is consumed by the monitor.
- Still, they improved the performance over serial systems.

Problems with the Batch System

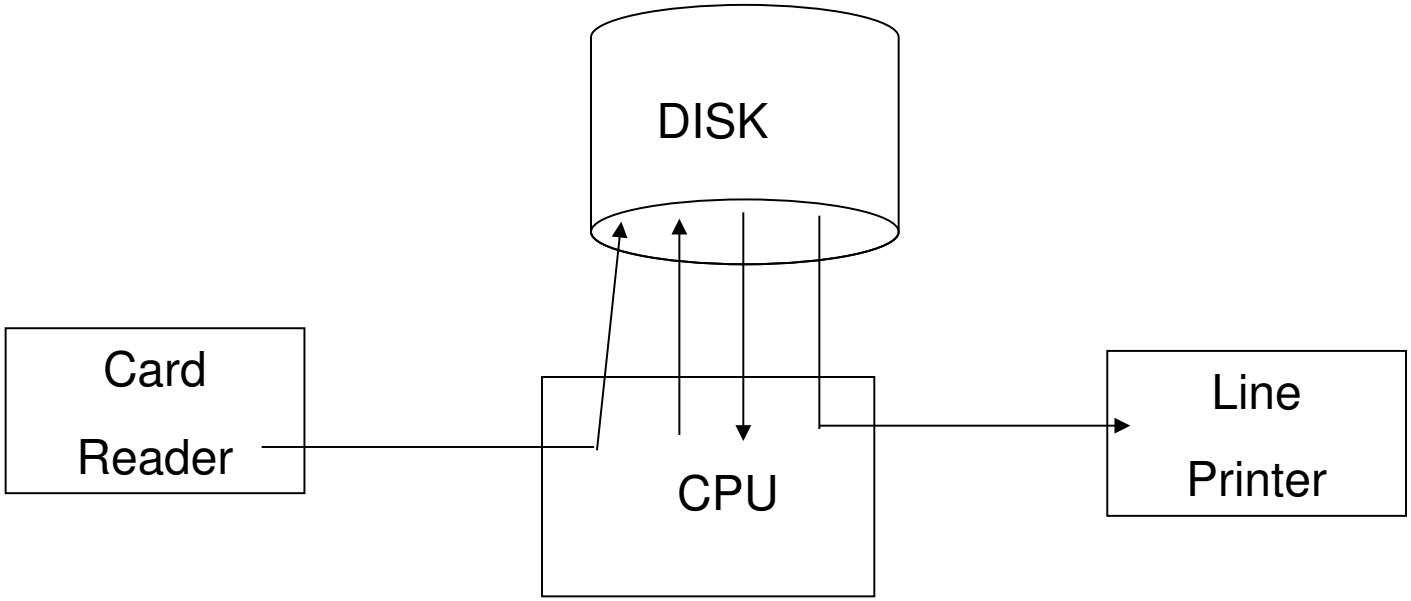
- CPU is idle
  - Speed of mechanical devices is very slower than those of electronic devices.
- CPU works in a microsecond range
  - Thousands of instructions/second
- A card reader may read 1200 cards per minute
  - (20 cards per second)
- CPU speed has increased at a faster rate.
- Tape technology improved the performance little-bit.
- Main perceived problem
  - Turn-around time: up to two days
  - CPU often **underutilized**
    - Most of the time was spent reading and writing from tape.

Resident monitor: summary

- Automatic job sequencing
  - Use of control cards
- Job control language
  - Commands
    - Mount this tape
    - Compile
    - Run
- OSs begin to be important.
  - IBM: Fortran monitor system
- Main perceived problems
  - Turn-around time
  - Inexpensive use of expensive hardware
  - CPU is still mostly idle.

Spooling

- The introduction of disk technology helped in this regard.
- Disk technology introduced the **SPOOLing (Simultaneous Peripheral Operations On-Line)**
- Considers disk as a huge buffer.
- Input comes from the disk
- Output goes to the disk.

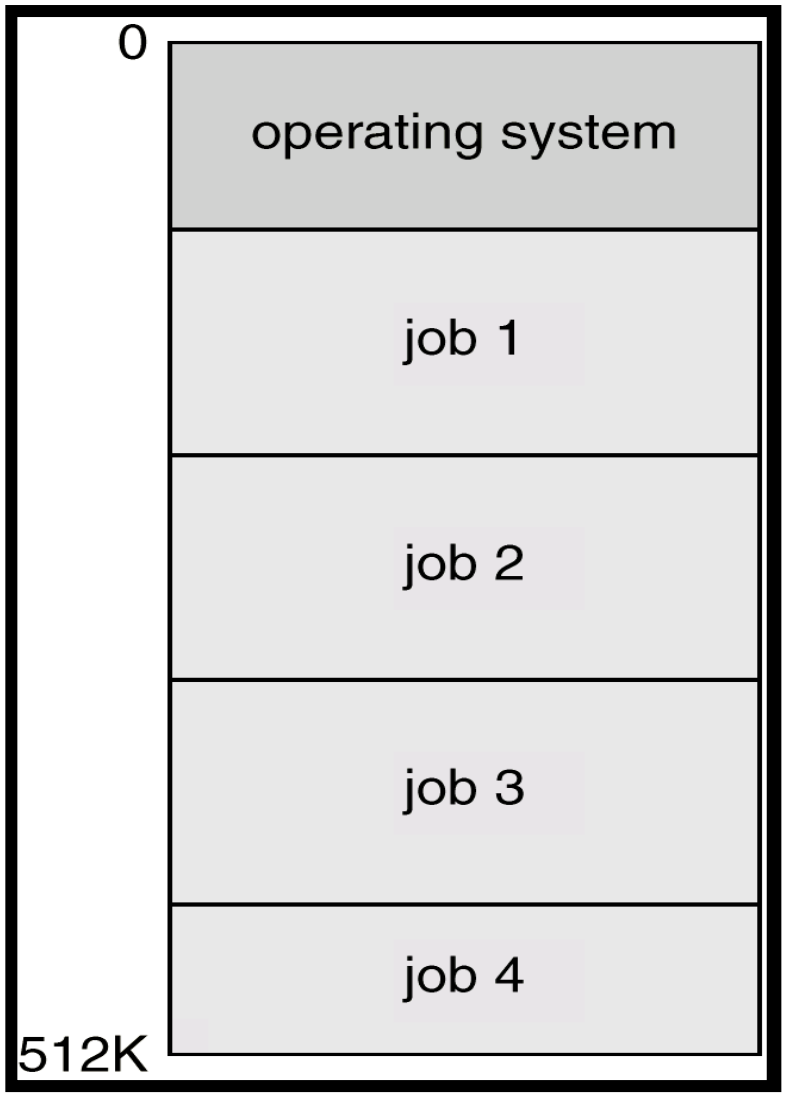


Advantage of Spooling

- Reading can be done in advance.
- Output can be stored on the disk.
- Spooling is also used for processing data at remote sites.
- Spooling overlaps the I/O of one job and computation of other jobs.
  - Even printing and reading can overlap.
- Spooling can keep both the CPU and the I/O devices working at higher rates.
- Disk is a random access device.

Multi-programmed Batched Systems (1960s) (or Multi tasking)

- A single user can not keep either CPU or I/O busy.
- Multiprogramming increases CPU utilization by organizing jobs such that the CPU always has one to execute.
- The OS keeps several jobs in memory at a time and CPU is multiplexed among them



Multi-programmed Batch Systems

- If CPU is executing a job and requires a tape to be mounted
  - In a non multi-programmed system
    - CPU sits idle.
  - In a Multi-programmed system
    - CPU takes up another job.
- **Multiprogramming is the first instance when the OS started taking decisions.**
- Job scheduling is done by OS.
- Having several programs in the memory requires memory management.

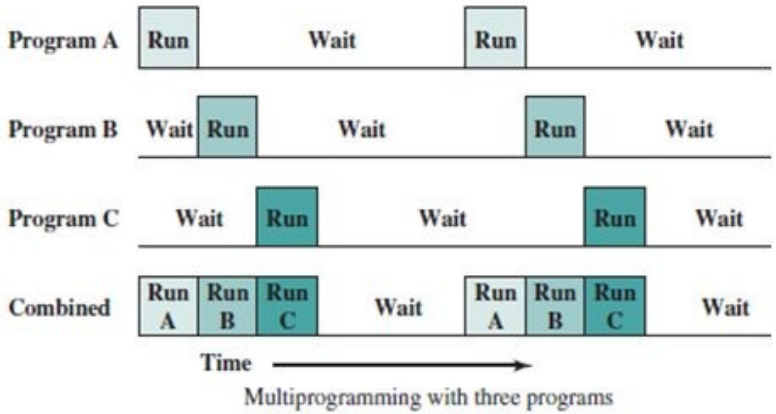
Multi-programmed Batch Systems

- I/O devices very slow.
- When one program is waiting for I/O, another can use the CPU.

Example:

Read one record from file	15 μs
Execute 100 instructions	1 μs
Write one record to file	15 μs
TOTAL	31 μs

Percent CPU utilization $= \frac{1}{31} = 0.032 = 3.2\%$



Multiprogramming with three programs

OS Features Needed for Multiprogramming

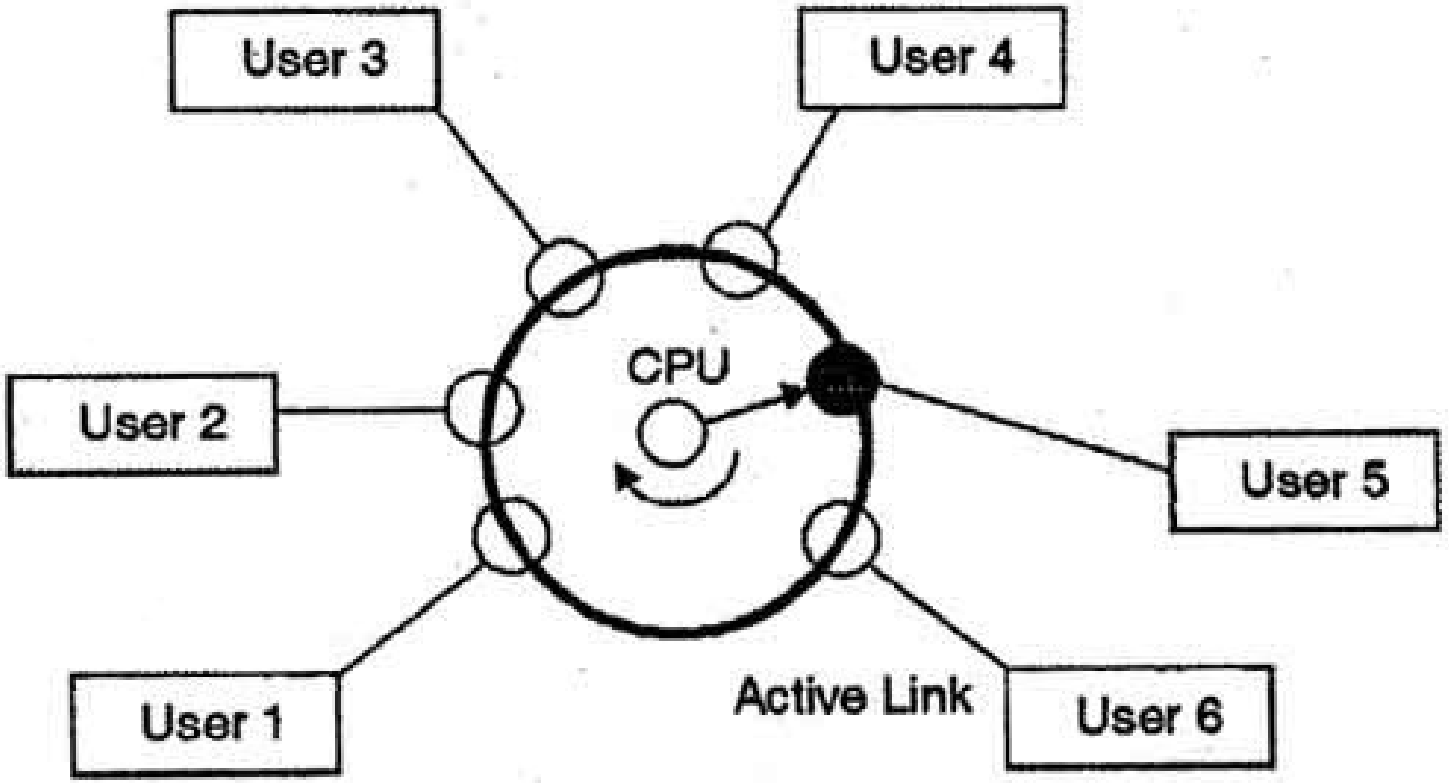
- I/O routine supplied by the system.
- Memory management –
  - the system must allocate the memory to several jobs.
- CPU scheduling –
  - the system must choose among several jobs ready to run.
- Allocation of devices.

Time-Sharing Systems–Interactive Computing

- With multiprogramming
  - Utilization is OK
  - Response time was a problem.
- Timesharing:
  - Programs could interact with user.
- Programs
  - Could wait for I/O for arbitrary time
    - CPU switched to another job.
    - However, resident jobs took up valuable memory
      - Needed to be swapped out to disk
      - Virtual memory.
- Time-sharing systems were developed to provide **interactive use of a computer system** at a reasonable cost.

Time-Sharing Systems–Interactive Computing

- A time sharing system uses CPU scheduling and multi-programming to provide each user with a small portion of a time shared computer.
- A program that is loaded into a memory and is executing is commonly known as a process.
- In timesharing system, a process executes for only a short time.
  - I/O is at people speeds, but OS can switch rapidly.
- A time-shared OS system allows the many users to share the computer simultaneously.
- It gives the impression that the user has own computer, whereas actually a computer is shared among many users.



Time-sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time.

TIME SHARING SYSTEMS

About Modern OSs

- Multiprogramming and timesharing are the central themes of modern OSs.
- Multiprogramming and timesharing requires
  - CPU scheduling
  - Process synchronization and communication
  - Deadlock detection
  - Memory management and protection
  - Virtual memory: A program is bigger than physical memory
  - Online file systems.
  - Disk management
  - Security and protection
  - Real-time and multimedia support
- In this course, we will discuss key concepts/algorithms /ideas developed since 1950 on the preceding aspects/issues.
- We also discuss networking protocols.