Operating Systems

What is OS?

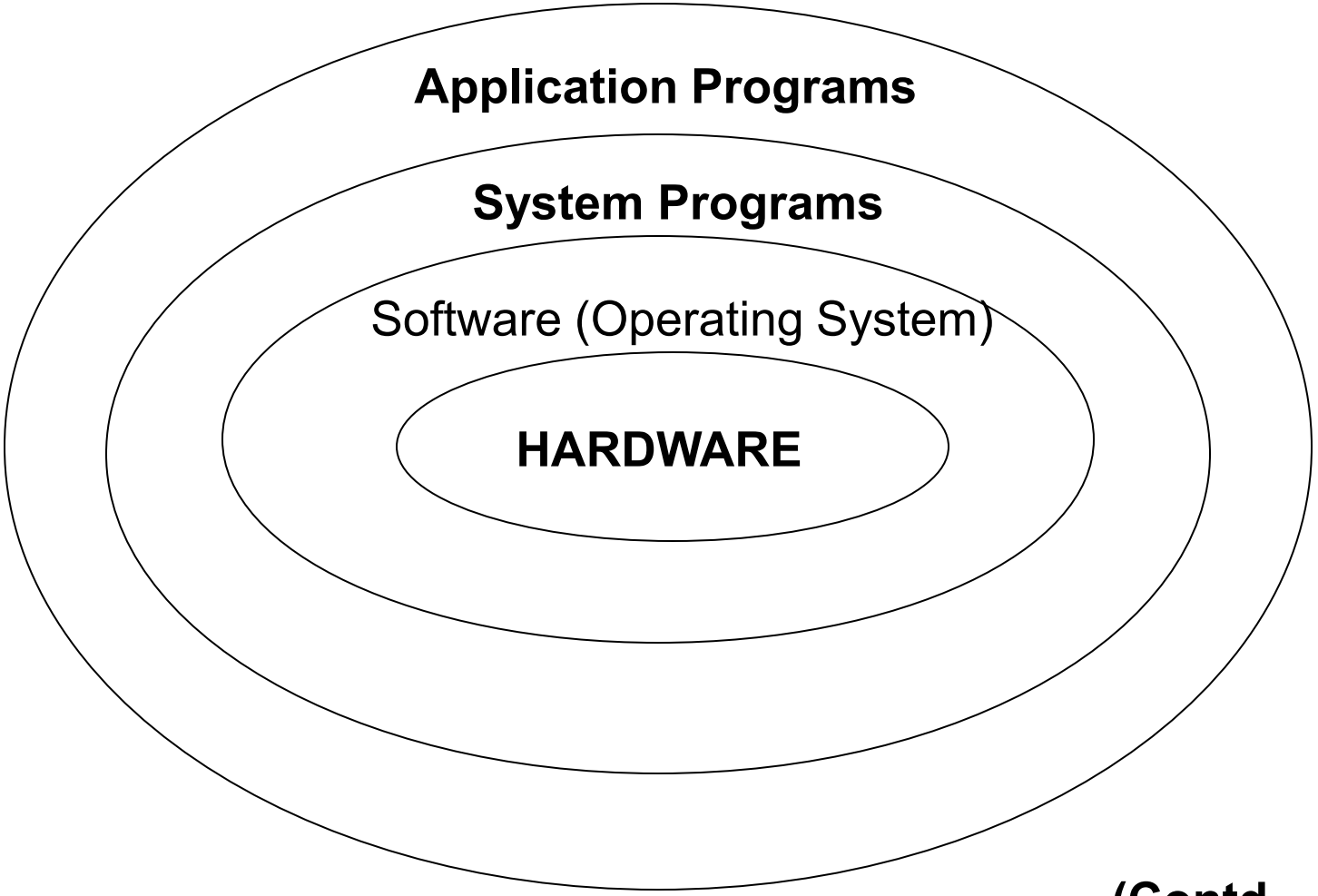
- Operating System is a software, which makes a computer to actually work.
- It is the software the enables all the programs we use.
- The OS organizes and controls the hardware.
- OS acts as an interface between the application programs and the machine hardware.
- Examples: Windows, Linux, Unix and Mac OS, etc.,

What OS does?

An operating system performs basic tasks such as,

- controlling and allocating memory,
- prioritizing system requests,
- controlling input and output devices,
- facilitating networking and
- managing file systems.

Structure of Operating System:



(Contd…)

Structure of Operating System (Contd…):

- The structure of OS consists of 4 layers:
  1. **Hardware**

     Hardware consists of CPU, Main memory, I/O Devices, etc,

  2. **Software (Operating System)**

     Software includes process management routines, memory management routines, I/O control routines, file management routines.

(Contd…)

Structure of Operating System (Contd…):

3. **System programs**

   This layer consists of compilers, Assemblers, linker etc.

4. **Application programs**

   This is dependent on users need. Ex. Railway reservation system, Bank database management etc.,

Evolution of OS:

- The evolution of operating systems went through seven *major phases.*
- Six of them significantly changed the ways in which users accessed computers through the open shop, batch processing, multiprogramming, timesharing, personal computing, and distributed systems.
- In the seventh phase the foundations of concurrent programming were developed and demonstrated in model operating systems.

(Contd…)

Evolution of OS (contd..):

Major Phases	Technical Innovations	Operating Systems
Open Shop	**The idea of OS**	**IBM 701 open shop (1954)**
Batch Processing	**Tape batching,** **First-in, first-out scheduling.**	**BKS system (1961)**
Multi-programming	**Processor multiplexing, Indivisible operations, Demand paging, Input/output spooling, Priority scheduling, Remote job entry**	**Atlas supervisor (1961),** **Exec II system (1966)**

(Contd…)

Evolution of OS (contd..):

Timesharing	**Simultaneous user interaction,** **On-line file systems**	**Multics file system (1965),** **Unix (1974)**
Concurrent Programming	**Hierarchical systems, Extensible kernels, Parallel programming concepts, Secure parallel languages**	**RC 4000 system (1969),** **13 Venus system (1972),** **14 Boss 2 system (1975).**
Personal Computing	**Graphic user interfaces**	**OS 6 (1972)** **Pilot system (1980)**
Distributed Systems	**Remote servers**	**WFS file server (1979) Unix United RPC (1982)** **24 Amoeba system (1990)**

Batch Processing:

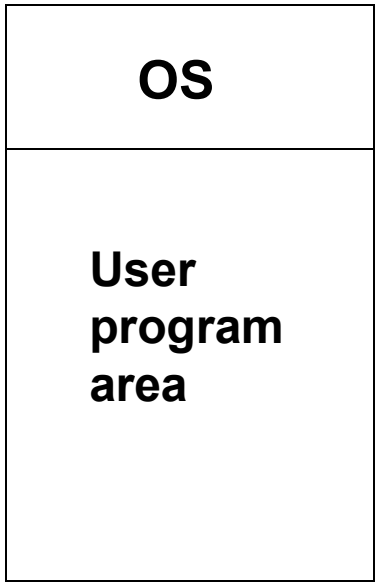
- In Batch processing same type of jobs batch *(BATCH- a set of jobs with similar needs)* together and execute at a time.
- The OS was simple, its major task was to transfer control from one job to the next.
- The job was submitted to the computer operator in form of punch cards. At some later time the output appeared.
- The OS was always resident in memory. (Ref. Fig. next slide)
- To avoid the problems of early systems the batch processing systems were introduced. The problem of early systems was more setup time. So the problem of more set up time was reduced by processing the jobs in batches, known as *batch processing system.* In this approach similar jobs were submitted to the CPU for processing and were run together.

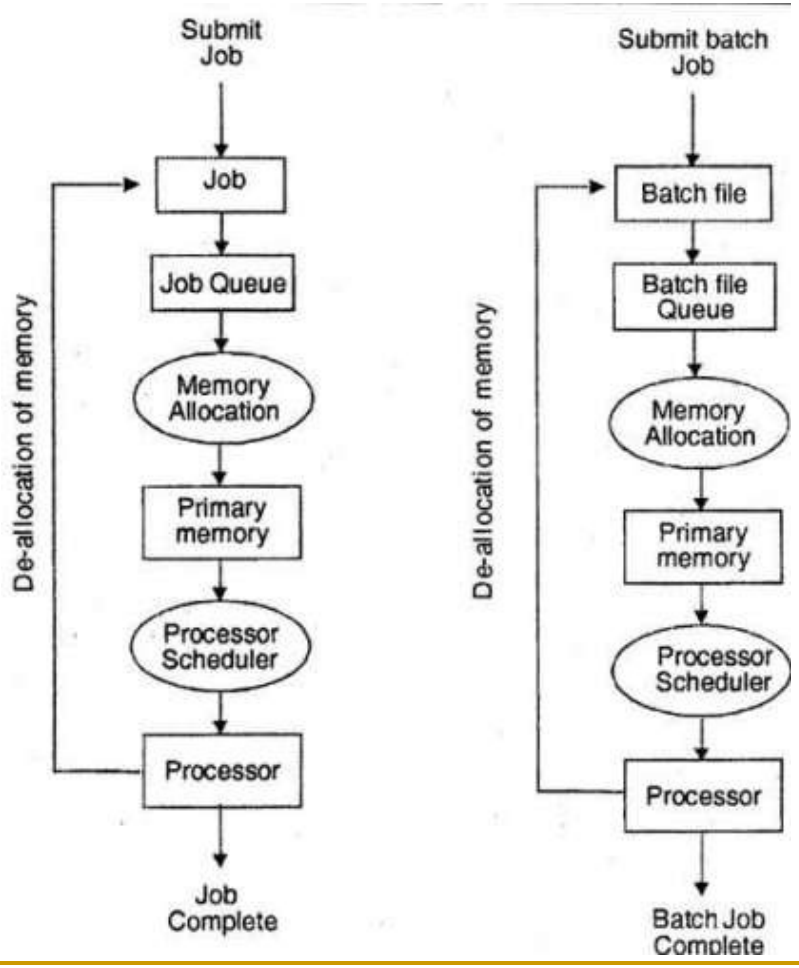
Cont..

- Common Input devices were card readers and tape drives.
- The jobs could be stored on the disk to create the pool of jobs for its execution as a batch.
- First the pooled jobs are read and executed by the batch monitor, and then these jobs are grouped; placing the identical jobs (jobs with the similar needs) in the same batch, So, in the batch processing system, the batched jobs were executed automatically one after another saving its time by performing the activities (like loading of compiler) only for once.
- It resulted in improved system utilization due to reduced turn around time.

Batch Processing (Contd…):

- Common output devices were line printers, tape drives, and card punches.
- Users did not interact directly with the computer systems, but he prepared a job (comprising of the program, the data, & some control information).

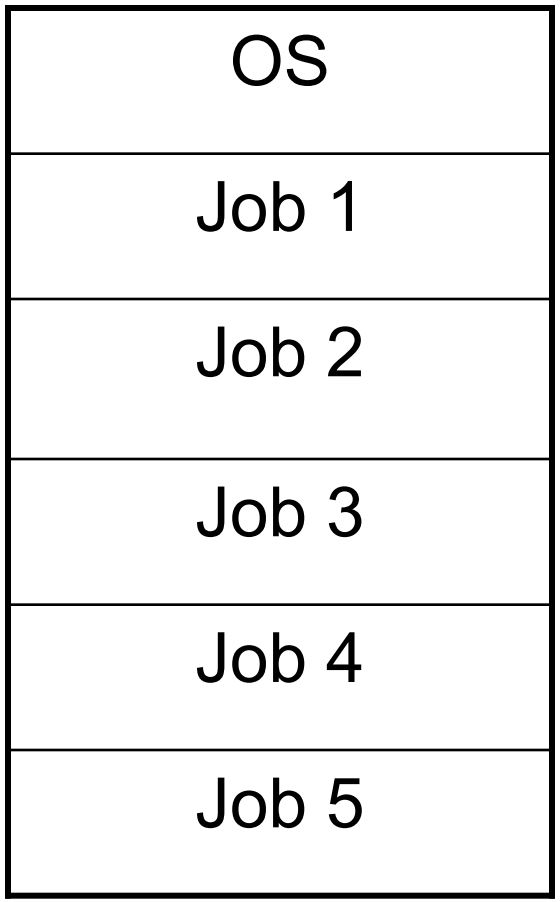



Submit Job
Job
Job Queue
Memory Allocation
Primary memory
Processor Scheduler
Processor
Job Complete
De-allocation of memory
Submit batch Job
Batch file
Batch file Queue
Memory Allocation
Primary memory
Processor Scheduler
Processor
Batch Job Complete
De-allocation of memory

Multiprogramming:

- Multiprogramming is a technique to execute number of programs simultaneously by a single processor.
- In Multiprogramming, number of processes reside in main memory at a time.
- The OS picks and begins to executes one of the jobs in the main memory.
- If any I/O wait happened in a process, then CPU switches from that job to another job.
- Hence CPU in not idle at any time.

Multiprogramming (Contd…):



- Figure dipicts the layout of multiprogramming system.
- The main memory consists of 5 jobs at a time, the CPU executes one by one.

Advantages:

- Efficient memory utilization
- Throughput increases
- CPU is never idle, so performance increases.

Time Sharing Systems:

- Time sharing, or multitasking, is a logical extension of multiprogramming.
- Multiple jobs are executed by switching the CPU between them.
- In this, the CPU time is shared by different processes, so it is called as “Time sharing Systems”.
- Time slice is defined by the OS, for sharing CPU time between processes.
- Examples: Multics, Unix, etc.,

Multi processor

- Also known as parallel processor or tightly coupled systems.
- Such systems have two or more processors, sharing clock, memory.
- 3 main advantages:
- Increase throughput: by increasing the number of processors, we get more work done in less time.so overall efficiency can increase.

- Economy of scale: The total cost of system with N processors will be lesser than the cost of N systems with one processor, because they can share many resources like memories, storage devices.
- Increased reliability: By coupling multiple processors into a single system, the system becomes more reliable.
- In multiprocessor the failure of a particular processor doesnot shut down the working of entire system rather it just slows it down.
- When a particular processor fails, the other processors take share of its working.

Real Time O.S

- Real time O.S are used under strict timeline or deadline boundaries. they are used in the environment where large number of events are processed within a fixed time duration.
- The operating system has to analyse all the input and has to generate the required output within a specific period or deadline.
- They are successful only if they return correct output in specified time interval.

types

- **Hard real time system**: systems that guarantees the completion of a particular job within the certain time limit. Such kind of systems have no scope for any kind of delays in the system's working.

- **Soft real time system**: systems that cannot fully guarantees the completion of a particular job within the certain time limit.these systems actually allow the concept of priorities. A process with higher priority is processed before the processing of the process with lower priority.

Operating Systems functions:

- The main functions of operating systems are:

1. Program creation
2. Program execution
3. Input/Output operations
4. Error detection
5. Resource allocation
6. Accounting
7. protection

Types of OS:

Operating System can also be classified as,-

- **Single User Systems**
- **Multi User Systems**

Single User Systems:

- Provides a platform for only one user at a time.
- They are popularly associated with Desk Top operating system which run on standalone systems where no user accounts are required.
- Example: DOS

Multi-User Systems:

- Provides regulated access for a number of users by maintaining a database of known users.
- Refers to computer systems that support two or more simultaneous users.
- Another term for *multi-user* is *time sharing*.
- Ex: All mainframes and are multi-user systems.
- Example: Unix

Operating System *(cont'd)*

- Multi-tasking
  - Same concept as multiprogramming
  - Allows 2 or more programs at the same time
  - Divide the programs into active and inactive. Active application is called *foreground application* and inactive application is *background application*
  - CPU does not run the programs simultaneously but manage the resources i.e. memory based on active/inactive applications

Operating System *(cont'd)*

- Cooperative Multi-tasking
  - Based on multi-tasking concept, whereby foreground application controls CPU until program ends
  - But if program crashes/hang, OS needs to be rebooted
- Preemptive Multi-tasking
  - Based on multi-tasking concept
  - But if program crashes/hang, OS is able to take over CPU so other programs can be maintained

Operating System *(cont'd)*

- Multithreading
  - One program may execute more than one task
  - Example *print* and *edit* at the same time
- Multiprocessing
  - Use of 2 or more processor at the same time
  - Programs run simultaneously
  - Each processor manage own resources
  - Hardware dependant esp. motherboard

- Interactive processing: The user has to be present and program cannot proceed until there is some input from the user.

- Ex - Select from a menu at ATM.