Classification of Operating System

1. Batch Processing System
2. Interactive Operating System
3. Multiprogramming Operating System
4. Multi User Operating System
5. Multi Tasking / Time Sharing Operating System
6. Multi Threaded Operating System
7. Multi Processor Operating System
8. Distributed Operating System
9. Real Time Operating System

Batch Processing Operating System

- The users of batch processing operating system do not interact with the computer directly.
- Each user prepares his job on an offline device like punch cards and submits it to the computer operator. To speed up processing job with similar needs are batched together and run as a group.
- The programmers left their programs with operators. The operator then sorts programs into batches with similar requirements.
- The problems with batch systems are following:
    - Lack of interaction between user and job.
    - CPU is often idle because the speed of mechanical I/O devices is slower than CPU.
    - Difficult to provide the desired priority.

Interactive Operating System

- In an interactive operating system the user interacts directly with the operating system to supply commands and data as the application program executes.
- The user receives the results of processing immediately.
- The user is in direct two way communication with the computer.
- There will a user interface in place to allow this to happen. It could be CLI (Command Line Interface) or GUI (Graphical User Interface).
- Example: ATM Machine

Multiprogramming Operating System

- In multiprogramming operating system several jobs are kept in memory at the same time, & the CPU is multiplexed among them.
- It increases CPU utilization by organizing jobs so that the CPU always has one to execute.
- The operating system picks and begins to execute one of the jobs in memory. Eventually, the job may have to wait for some task, such as an I/O operation, to complete. In a non-Multiprogrammed system, the CPU would sit idle. In a Multiprogrammed system, the operating system simply switches to, and executes, another job. When that job needs to wait, the CPU is switched to another job, and so on. Eventually, the first job finishes waiting and gets the CPU back. As long as at least one job needs to execute, the CPU is never idle.

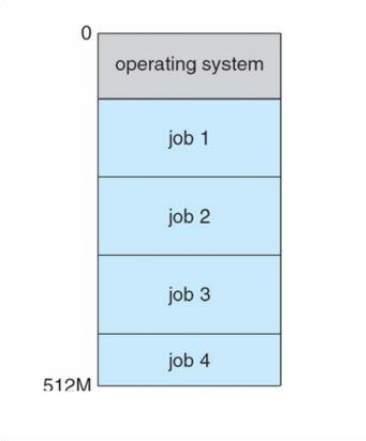

Figure: Memory Layout of Multiprogramming Operating System

- Multiprogrammed systems provide an environment in which the various system resources are utilized effectively, but do not provide for user interaction with the computer system.
- Example: Windows, Linux, etc.

Multi User Operating System

- A multi user operating system allows multiple users on different computers or terminals to access a single system with one operating system on it.
- The users will at terminals or computers that give them access to the system through a network, as well as other machines on the system such as printers.
- A multi user operating system differs from single user system on a network in that each user is accessing the same operating system at different machines.
- Example- UNIX server, where multiple remote users have access to the UNIX shell prompt at the same time.

Multitasking / Time Sharing Operating System

- Time sharing or multitasking is a logical extension of multiprogramming. It requires several jobs to be kept simultaneously in memory.
- A time-shared operating system allows many users to share the computer simultaneously.
- In time-sharing system, the CPU executes multiple jobs by switching among them, but the switches occurs so frequently that the users can interact with each program while it is running.
- As the system switches rapidly from one user to next, each user is given the impression that the entire computer system is dedicated to his use, even though it is being shared among many users.
- A time-shared operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time shared computer.
- It also provides a file system and a mechanism for protecting resources from inappropriate use.
- Example- IBM TSS/360

Multi Threaded Operating System

- A thread is a flow of control within a process.
- If a process has multiple threads of control it can perform more than one task at a time.
- Example- A word processor may have a thread for displaying graphics, another thread for responding to keystrokes from the user, and a third thread for performing spelling and grammar checking in the background.
- The benefits of multithreading include increased responsiveness to the user, resource sharing with in the process, economy, and the ability to take advantage of multiprocessor architectures.
- The difference between single threaded process and multi threaded process

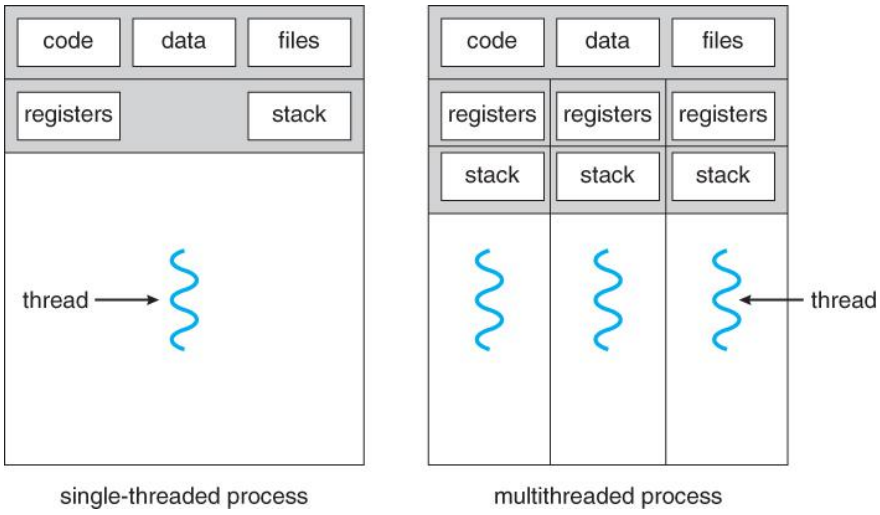

Figure: Single Threaded Process & Multithreaded Process

- An operating system which supports multiple threads are called multi threaded operating system
- Some difference between process and threads are

Process	Thread
Processes are the abstraction of running programs	Threads are the unit execution in a process
Process is heavy weight or resource intensive.	Thread is light weight taking lesser resource than a process
Process switching needs interaction with operating system	Threads switching does not need to interact with operating system

Process	Thread
In multiprocessing environment, each process executes the same code but has its own memory and file resources.	In multithreaded environment, all threads can share same set of open files, child processes.
Process are difficult to create than threads as they require separate address space	Threads are easier to create than processes as they don't require a separate address space
Each process operates independently of the others.	One thread can read, write or change another threads data.

Multi-Processor Operating System

- A multi processor operating system allows a program to run on more than one CPU (processor) at a time.
- Due to more number of processors they have certain advantages like increased throughput, reliability, etc.
- They are classified into two categories:

1. Asymmetric Multi Processing (AMP)

- This scheme defines a master-slave relationship.
- In this one processor called master which controls the rest of the processors called salves.
- In this each processor is assigned a specific task.
- A master processor controls the system the other processors either look to the master for instruction or have predefined tasks. The master processor schedules and allocates work to the slave processors.

2. Symmetric Multi Processing (SMP)

- In SMP all the processors are peers.
- There is no master slave relationship exists between processors.
- In this all processors perform all tasks within the operating system.
- SMP has some serious problems like what will happen when two or more CPUs are running the operating system code at the same time.

Distributed Operating System

- A distributed OS is software over a distributed system (collection of independent, networked, communicating and physically separate computation node).
- Access to shared resource increases- computation speed, functionality, data availability and reliability.
- Distributed systems depend on networking for their functionality.
- In a distributed OS, users access remote resource in the same way they access local resources.
- Data and process migration from one site to another is under the control of the distributed OS.
- Examples: CP/M, UNIX-3

Real Time Operating System

- Primary objective of Real Time OS is to provide quick response time and thus to meet a scheduling deadline- user convenience, resource utilization are secondary concern to these systems.
- A real time system has well defined, fixed time constraints.
- Processing must be done within the defined constraints, or the system will fail. A real time system functions correctly only if it returns the correct result within its time constraints.
- A real time system is used when rigid time requirements have been placed on the operation of a processor or the flow of data.
- Systems that control scientific experiments, medical imaging systems and industrial control systems are real time systems.

Types of Real Time Systems

1. Soft Real Time Systems

- If certain deadlines are missed then system continues its working with no failure but its performance degrades.
- It is a less restrictive type of real time system, where a critical real time task of high priority gets priority over other tasks and retains the priority until it completes.
- They are risky to use for industrial control and robotics. They are useful in multimedia, virtual reality and advanced scientific projects.
- **Examples**: Many commercial OS as well as Linux provide soft real time support.

2. Hard Real Time Systems

- If any deadline is missed then system will fail to work or does not work properly. This system guarantees that critical tasks be completed on time.
- In hard real time system secondary storage is limited or absent, data is stored in short term memory or ROM.
- It is not supported by general purpose OS.
- **Examples:** OS-9, THE OS, Heart OS, Windows CE (Compact Embedded), INTEGRITY

Questions asked in semester exam:

Question: Difference between Process and Program.
[2016-2017][2 Marks][2012-2013] [5 Marks]

Question: Define multiprogramming system.
[2016-2017] [2 Marks]

Question: Write down the different types of operating system.
[2016-2017] [5 Marks]

Question: Define Multithreading.
[2015-2016] [2 Marks]

Question: Define Process.
[2015-2016] [2 Marks]

Question: What is Real Time Operating System? What is the difference between hard real time and soft real time Operating System?
[2014-2015][2013-2014] [5 Marks]

Question: What is multiprocessing operating system? Differentiate between SMP and AMP.
[2014-2015] [5 Marks]

Question: Differentiate between Real Time and Time Sharing Operating System.
[2014-2015] [5 Marks]

Question: Describe the differences between symmetric and asymmetric multiprocessing.
[2014-2015] [5 Marks]

Question: Differentiate between (with one suitable example)
(i) Interactive and Batch Processing System
(ii) Multiprogramming and Time Sharing System
[2014-2015] [5 Marks]

Question: Write down the difference between Multiprocessing and multiprogramming operating system
[2013-2014] [5 Marks]

Question: Discuss essential properties of Time sharing operating system, Real time operating system and distributed operating system.
[2012-2013] [5 Marks]

Question: Explain the following terms clearly:
(i) Process
(ii) Multiprogramming
(iii) Multithreading
[2011-2012] [5 Marks]

Question: Discuss the differences between a time sharing system and real time system.
[2010-2011] [5 Marks]

Question: Discuss the objectives of the multiprocessor systems.
[2010-2011] [5 Marks]

Question: Explain the following:
(i) Multitasking
(ii) Multithreading
[2010-2011] [5 Marks]

Question: Write down the advantages of batch processing system.
[2009-2010] [5 Marks]

Question: Explain the main features of a real time operating system.
[2009-2010] [5 Marks]

Question: Compare multitasking and multiuser operating system.
[2008-2009] [5 Marks]

Question: Explain in brief real time operating systems. Illustrate some areas where they are used.

[2008-2009] [5 Marks]

Question: Differentiate between (with one suitable example)

(i) Interactive and Batch Processing System
(ii) Multiprogramming and Multitasking

[2007-2008] [5 Marks]

Question: Differentiate between General purpose OS and Real Time OS

[2007-2008] [5 Marks]

Question: Write short-notes on the following

(i) Time Sharing System
(ii) Real Time System

[2006-2007] [5 Marks]