

Types OF Operating Systems

Batch, Time Shared, Multi-Programmed



**Operating Systems**

* **What is Operating System?**

An Operating System (OS) acts as an interface between a user and computer hardware. The fundamental goal of an Operating System is to execute user programs and to make tasks easier. Various application programs along with hardware system are used to perform this work. Operating System is a software which manages and control the entire set of resources and effectively utilize every part of a computer. It performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers. Currently the well-known Operating Systems are Windows, Mac, Linux, etc.

* **History of Operating System:**

1. **The First Generation (1940's to early 1950's)**

When electronic computers were first introduced in the 1940's they were created without any operating systems. [All programming was done in absolute machine language, often by wiring up plugboards to control the machine's basic functions.](http://www.informit.com/articles/article.aspx?p=24972) During this generation computers were generally used to solve simple math calculations, operating systems were not necessarily needed.

1. **The Second Generation (1955-1965)**

The first operating system was introduced in the early 1950's, it was called GMOS and was created by General Motors for IBM's machine the 701. Operating systems in the 1950's were called single-stream batch processing systems because the data was submitted in groups. These new machines were called mainframes, and they were used by professional operators in large computer rooms. Since there was such as high price tag on these machines, only government agencies or large corporations were able to afford them.

1. **The Third Generation (1965-1980)**

By the late 1960's operating systems designers were able to develop the system of multiprogramming in which a computer program will be able to perform multiple jobs at the same time. The introduction of multiprogramming was a major part in the development of operating systems because it allowed a CPU to be busy nearly 100 percent of the time. [Another major development during the third generation was the phenomenal growth of minicomputers, starting with the DEC PDP-1 in 1961.](http://www.informit.com/articles/article.aspx?p=24972) These microcomputers help create a whole new industry and the development of more PDP's. These PDP's help lead to the creation of personal computers which are created in the fourth generation.

**Note: PDP: Programmed Data Processor**

1. **The Fourth Generation (1980-Present Day)**

The fourth generation of operating systems saw the creation of personal computing. Although these computers were very similar to the minicomputers developed in the third generation, personal computers cost a very small fraction of what minicomputers cost. A personal computer was so affordable that it made it possible for a single individual to own one for personal use while minicomputers where still at such a high price that only corporations could afford to have them. One of the major factors in the creation of personal computing was the birth of Microsoft and the Windows operating system. The windows Operating System was created in 1975 when Paul Allen and Bill Gates had a vision to take personal computing to the next level. They introduced the MS-DOS in 1981 although it was effective it created much difficulty for people who tried to understand its cryptic commands. Windows went on to become the largest operating system used in technology today with releases of Windows 95, Windows 98, Windows XP, Windows 7(Which is currently the most used operating system to this day), and their newest operating system Windows 10. Along with Microsoft, Apple is the other major operating system created in the 1980's.

* **Need Of Operating System!!**

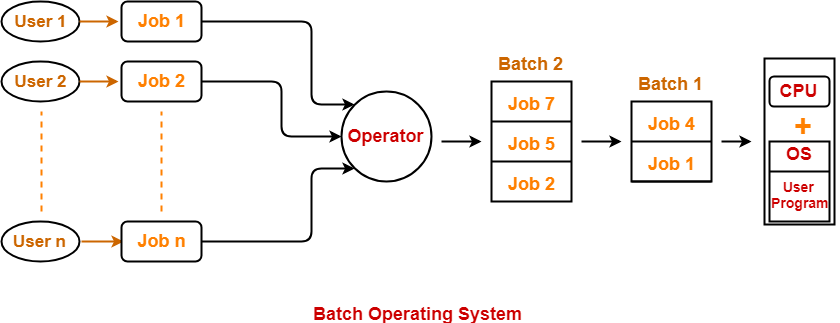
In earlier day’s user had to design the application according to the internal structure of the hardware. Operating System was needed to enable the user to design the application without concerning the details of the computer’s internal structure. Some of the important features of operating System Explains Its needs: -

* **Platform for Application Programs:** Operating system provides a platform, on top of which, other programs, called application programs can run. These application programs help the users to perform a specific task easily. It acts as an interface between the computer and the user. It is designed in such a manner that it operates, controls and executes various applications on the computer.
* **Managing Input-Output unit:** Operating System also allows the computer to manage its own resources such as memory, monitor, keyboard, printer etc. Management of these resources is required for an effective utilization. The operating system controls the various system input-output resources and allocates them to the users or programs as per their requirement.
* **Consistent user interface:** Operating System provides the user an easy-to-work user interface, so the user doesn’t have to learn a different UI every time and can focus on the content and be productive as quickly as possible. Operating System provides templates, UI components to make the working of a computer, really easy for the user.
* **Multitasking:** Operating System manages memory and allow multiple programs to run in their own space and even communicate with each other through shared memory. Multitasking gives users a good experience as they can perform several tasks on a computer at a time.
* **Types of Operating Systems:**

There are various Type of Operating System. Some of them are as follows:

1. Batch Operating System
2. Time Shared Operating System
3. Multi-Programmed Operating System

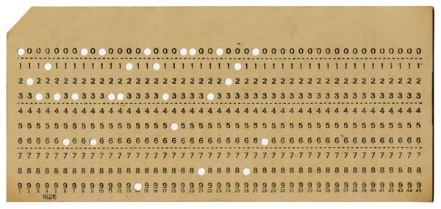
**Batch Operating System**



* **Introduction:**

Back in 1950s and 1960s, there were no well-developed [operating systems](http://www.padakuu.com/article/83-operating-system-services), advanced computer architecture and secondary memory devices. Instead there were large mainframe computers for the processing work and punched cards or magnetic tapes were used to provide the input and receive the output. The major problem was the lack of hard disk in those times. By the early 1950’s, the General Motors Research Laboratories implemented the first Single-Stream batch processing systems. In early 1960’s company’s like Nasa, Isro, etc used to use this technology for the purpose of heavy computation and research. In the era of 1970s, the Batch processing was very popular. People were used to have a single computer which was called mainframe.

In Batch Operating System similar kind of jobs were grouped together and a batch was formed. This batches were assigned to the computer for execution. Programs were written using punch card also known as Magnetic tape or Paper tape. The process of Batch System was as follows:

1. User use to prepare his job using punch card. All programming instructions were loaded into the punch card. Later The user used to submit his job to the operator.
2. The operator used to collect the jobs from different user and sort them into different batches based on the similar need
3. Later the operator used to submit the batches to the processor one by one. Thus, all the jobs of one batch were executed simultaneously.

**Punch Card**

The jobs were classified as long and short based on their processing time. The punched card used to contain the instruction in encoded form i.e. binary form. If any Input/Output operation was required then the processor used to wait for the operation to complete as the speed of processor was high if compared to speed of I/O devices. This resulted in poor utilization of processor. The processor used to remain Idle most of the time. In this scenario only one program was kept in the memory at the time of execution.

The Batch Operating Systems didn’t permit the user to interact with the jobs while they were running. Also, Job scheduling in this type of OS was based on first come first serve basis. Therefore, file and memory management are simple. The batch processing system are mainly used to update the data related to any transactions or any record. Batch Operating system was the first Operating System for Second Generation Computers. Batch operating systems load very less stress on CPU and involve lesser user interaction so that’s why we can use batch system in now a day also. Another advantage of batch operating system is that the large repeated jobs can done without interacting with computer to tell the system that you have to do that job after finishing that job. Old batch operating systems were not interactive i.e. the user interaction was not involved when the program is running. Now in modern batch operating systems we can have interactions also. For example, we can schedule the job and when the specific time comes then computer acknowledge the processor that time is over. This helps us in avoiding too many errors and makes debugging easier.

* **Advantages:**

1. Time wasted earlier by individual process in context switching from one environment to another is saved.
2. No manual intervention is needed.
3. To speed up the processing speed, the batch process can partition into the number of processes.
4. Multiple users can share the batch systems.
5. It is easy to manage large work repeatedly in batch systems.
6. Batch processing reduces a lot of work of the operator.
7. They don’t require any special hardware and system support to input data.
8. [CPU](https://ecomputernotes.com/fundamental/introduction-to-computer/what-is-cpu) utilization gets improved.

* **Dis-Advantages:**

1. Priority cannot be set for jobs.
2. It may lead to starvation

(The jobs of a particular batch might take long time for their execution. This might lead to starvation to other jobs in other batches.)

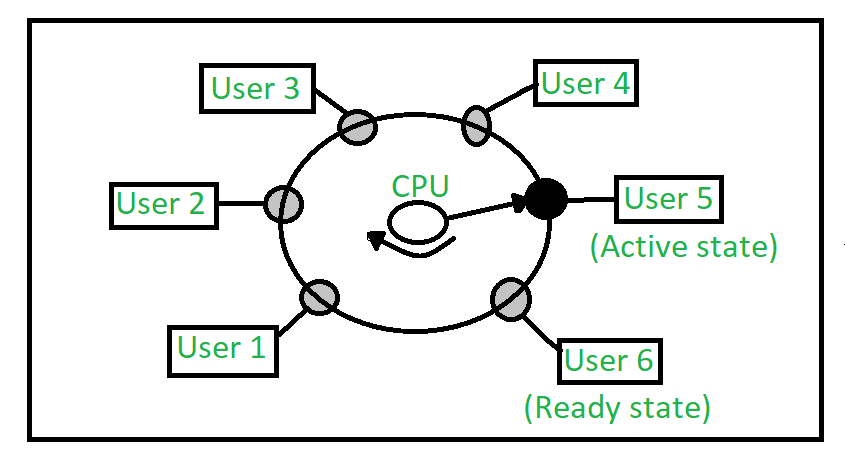
1. CPU remains idle for long time

(In batch operating system, If the jobs of a batch require some I/O operation, then CPU must wait till the I/O operation gets completed. Since I/O devices are very slow, CPU remains idle for a long time. CPU cannot take any other job and execute it.)

1. There is lack of interaction between user and his job.
2. This OS is Difficult to debug.
3. If a job gets to enter in an infinite loop/ fails, other jobs wait for unknown time.
4. Batch systems are costly.
5. Computer operator must be trained for using batch Systems.

* **Examples:**

1. FMS (the Fortran Monitor System)
2. IBSYS, IBM's operating system for the 7094.
3. GM-NAA I/O [input/output](https://en.wikipedia.org/wiki/Input/output) system of [General Motors](https://en.wikipedia.org/wiki/General_Motors) and [North American Aviation](https://en.wikipedia.org/wiki/North_American_Aviation) was the first [operating system](https://en.wikipedia.org/wiki/Operating_system) for the [IBM 704](https://en.wikipedia.org/wiki/IBM_704) computer.
4. GMOS by General Motors for IBM's machine the 701

**Time Shared Operating System**

* **Introduction:**

The implementation of first time-sharing project was by John McCarthy in the last months of 1957 using a modified IBM 704 and later a modified IBM 7090. The first successful large-scale time-sharing system which was commercially launched was the Dartmouth Time Sharing System.

Time-sharing system is a type of operating system which allows multiple users to access resources of a particular system from different locations. These systems were introduced in the 1960s as one of the perceptible models and represented a major part of the technological shift.

The architecture of a time-sharing system is also known as multi-tasking system which is a logical extension of the concept of multiprogramming. In this method, the time of a single processor is shared among different users over a network simultaneously.

Time-sharing system performs multiple tasks over a single Central Processing Unit(CPU). The processor switches between the different assigned tasks to complete them individually. This switch occurs so frequently that the overall throughput remains unaffected from this approach. For instance, the processor executes each of the user programs in a short burst during the transaction process. This short burst is also referred to as a single quantum of computing. This means, when ‘n’ users want to use the processor, then each one of them can get a dedicated time quantum to complete their tasks.

The time-sharing operating system uses CPU schedulingas well as multiprogramming concepts to offer each of the users a small part of the operating time. The concepts of the time-sharing operating system align with those of the batch processing systems. It is also said that computers which were designed primarily as batch systems have been changed to operate as time-sharing systems.

Time shared operating systems are more complex than multiprogrammed operating systems. In both, multiple jobs must be kept in memory simultaneously, so the system must have memory management and security. To achieve a good response time, jobs may have to swap in and out of disk from main memory which now serves as a backing store for main memory. A common method to achieve this goal is virtual memory, a technique that allows the execution of a job that may not be completely in memory.

In above figure the user 5 is active state but user 1, user 2, user 3, and user 4 are in waiting statewhereas user 6 is in ready state**.**

1. **Active State:**  
   The user’s program is under the control of CPU. Only one program is available in this state.
2. **Ready State:**  
   The user program is ready to execute but it is waiting for its turn to get the CPU. More than one user can be in ready state at a time.
3. **Waiting State:**  
   The user’s program is waiting for some input/output operation. More than one user can be in a waiting state at a time.

* **Basic features**
  + Every user gets a dedicated time for the operation.
  + Simultaneous tasks are carried out at once.
  + Tasks no longer have to wait for the previous task to finish to get the processor.
  + Quick processing of multiple tasks.
  + Equal time given to all the processes so that they operate smoothly without any significant delay.
* **Advantages:**

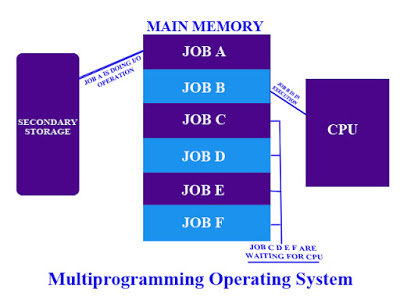
1. Quick response.
2. This type of operating system avoids duplication of software.
3. It reduces CPU idle time.
4. No starvation is there and all the processes can access the CPU for a fair amount of time.

* **Dis-Advantages:**

1. Reliability Problem
2. Question of security and integrity of user programs.
3. Problem of data communication occurs.
4. If any problem affects the operating System. Data stored may be lost.
5. This OS consumes much resources
6. Switching between tasks becomes sometimes sophisticated as there are lot of users and applications running which may hang up the system.

* **Examples:**

1. The Multics operating systems
2. Unix operating systems
3. TOPS-10 (DEC)
4. TOPS-20 (DEC)
5. Windows NT server (since 4.0)
6. CTSS (**Compatible Time-Sharing System)**

**Multi-Programmed Operating System**

* **Introduction:**

Early computer systems were extremely expensive to purchase and to operate. Unfortunately, they often sat idle as their human operators tended to their duties at a human’s pace. Even a group of highly trained computer operators could not work fast enough to keep even the earliest tape-fed batch systems CPU busy. The advent of multiprogramming broke through the utilization barrier by removing most of the human factor in CPU utilization.

In a multiprogramming system there are one or more programs loaded in main memory which are ready for execution. Only one program at a time is able to get the CPU for executing its instructions while all the others need to wait for their turn. The main idea of multiprogramming is to maximize the use of CPU time. Indeed, suppose the currently running process is performing an I/O task Then, the OS may interrupt that process and give the control to one of the other processes in-main-memory programs that are ready to execute (i.e. process context switching). In this way, no CPU time is wasted by the system waiting for the I/O task to be completed, and a running process keeps executing until either it voluntarily releases the CPU or when it blocks for an I/O operation. Therefore, the ultimate goal of multiprogramming is to keep the CPU busy as long as there are processes ready to execute.

An important concept in multiprogramming is the degree of multiprogramming. The degree of multiprogramming describes the maximum number of processes that a single-processor system can accommodate efficiently. The primary factor affecting the degree of multiprogramming is the amount of memory available to be allocated to executing processes. If the amount of memory is too limited, the degree of multiprogramming will be limited because fewer processes will fit in memory. A factor inherent in the operating system itself is the means by which resources are allocated to processes. If the operating system cannot allocate resources to executing processes in a fair and orderly fashion, the system will waste time in reallocation, or process execution could enter into a deadlock state as programs wait for allocated resources to be freed by other blocked processes. Other factors affecting the degree of multiprogramming are program I/O needs, program CPU needs, and memory and disk access speed.

Processes maintained in the computers main memory are considered to be executing concurrently even though the CPU is (usually) capable of executing only one instruction at a time. The number of processes that can be held in memory is dependent on the amount of memory available to the system. That amount may be any combination of real or virtual memory, virtual memory being a portion of on-line mass-storage allocated to hold code in the process of being executed which cannot fit into available real memory. If a process is too large to fit into the memory allocated to it, portions of its code may be stored temporarily on disk. When this code is required the operating system will load the code into memory and execution will continue. The management process by which this code is swapped into and out of memory is referred to a paging. A similar system can be used to manage data-segment memory.

As the number of processes (degree of multiprogramming) increases in a system that supports paging, the amount of memory available to execute processes in decreases and the number of paging operations required increases. At some point the amount of time the CPU spends paging code and data will drag system performance down. This phenomenon, called "thrashing," is a manifestation of exceeding the degree of multiprogramming for a system.

Efficient use of computer time was the impetus for the development of multiprogramming systems. Multiprogramming freed the CPU from relying on the inherent slowness of the operator, and permitted it to work while waiting on peripheral devices. Every computer system has a limit to the degree of multiprogramming it will support. This limit is primarily based on the amount of main memory available to the system and the efficiency of the operating systems resource allocation algorithms.

The Multi-programming operating system gives an illusion of running multiple process at once and provide users with interactive response to processes. In modern times all computer operating system support for multi programming. At the boot time itself several programs start up. Word processing ,spreadsheets, games, Internet access are some of the common use of these operating system.

* **Advantages:**

1. CPU is used most of time and never become idle, thus there is increase in processor utilization
2. The system looks fast as all the tasks runs in parallel
3. Short time jobs are completed faster than long time jobs
4. Support multiply users
5. Resources are used nicely
6. Total read time taken to execute program/job decreases
7. Response time is shorter
8. It maximizes the total job throughput of a computer.

* **Disadvantages:**

1. It is fairly sophisticated and more complex
2. CPU scheduling is required.
3. It is difficult to program a system because of complicated schedule handling
4. Tracking all tasks/processes is sometimes difficult to handle
5. Due to high load of tasks, long time jobs have to wait long
6. To accommodate several jobs in memory, memory management is essential.

* Examples:

1. Unix
2. Linux
3. Windows 7
4. Windows 8
5. Apple’s OS X