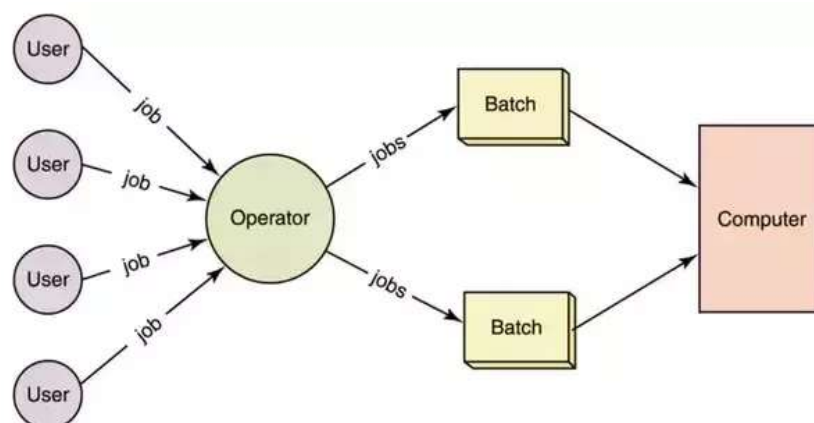


BATCH OPERATING SYSTEM

- In this type of system, there is **no direct interaction between user and the computer**.
- The user has to submit a job (written on cards or tape) to a computer operator.
- Then computer operator places a batch of several jobs on an input device.
- Jobs are batched together by type of languages and requirement.
- Then a special program, the monitor, manages the execution of each program in the batch.
- The monitor is always in the main memory and available for execution.
- Disk which contains a pool of jobs allows operating system to select job to run next for sake of CPU's proper utilization. When jobs come in direct manner on magnetic tape or cards jobs cannot run in a different order.
- **Jobs execute** on the bases of **first come, first serve** so they must run in **sequential** fashion.
- **Jobs scheduling** become possible when several jobs are on a direct access device such as disk. The main characteristic of job scheduling is the ability to multiprogram.
- In general, a single user cannot keep the I/O devices and CPU busy at all time.
- In this case jobs are organized in such a way that CPU execute one job at a time to increase the CPU utilization. Several jobs are kept by operating system in memory at a time.
- Operating system pick the one job and begin to execute it in the memory. Ultimately the job has to wait for some task such as a tape to be mounted on an I/O operation to complete.
- In case of **multiprogramming system** do not sit idle because **operating system switches to another job**. Whenever on job is on wait condition and current job finished then it gets the CPU back.
- Multiprogramming operating system is very sophisticated because it must make decisions for the user.



Advantages:

1. Suppose a job takes a very long time (1 day or so). Then, such processes can be performed even in the **absence** of humans.
2. They don't require any **special** hardware and system support to input data.

Disadvantages:

1. It is very difficult to **debug** batch systems.
2. Lack of **interaction** between user and operating system.
3. Suppose an error occurs in one of the jobs of a batch. Then, all the **remaining** jobs get affected i.e; they have to wait until the error is resolved.

Example

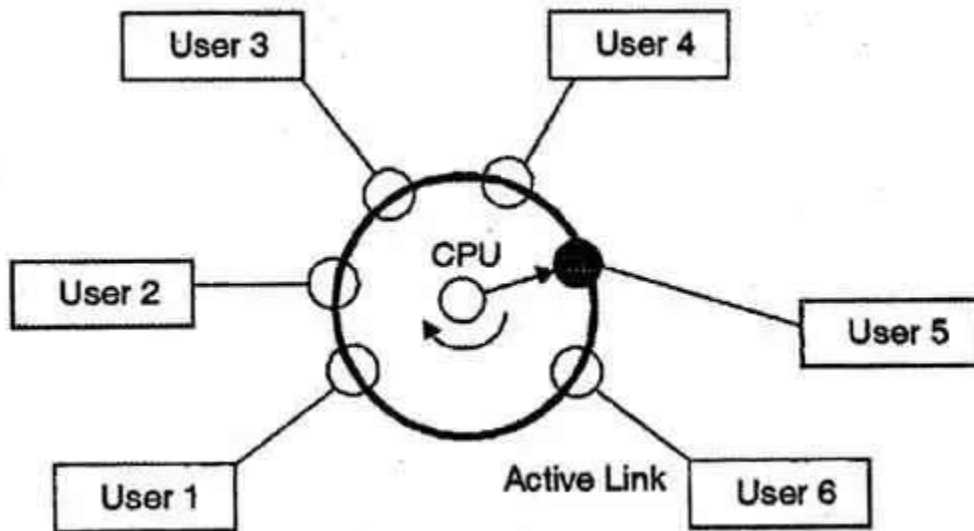
1.Payroll system - Batch systems are ideal for making payrolls. The salaries of employees can be printed at the end of month by the batch systems. So, the statements can be made easy by using batch systems.

2.Bank statements - At the end of month the bank makes statements for each account holder. So, these bank statements can be made easily by batch systems at the end of month.

TIME SHARING OPERATING SYSTEM

- Time sharing operating system is also called as multitasking which is basically a logical extension of multiprogramming.
- This system was introduced to overcome the problems of batch operating system.
- In time sharing operating system, system has to execute multiple jobs by the CPU switching between them, but the switches occur so frequently that the users may interact with each program while it is running. It provides system between user and system.
- The main objective of time-sharing systems is to **minimize response time** but not maximizing the processor use (which is the objective of multiprogramming systems).
- A time-sharing system allows many users to share the computer resources simultaneously. In other words, time sharing refers to the allocation of computer resources in time slots to several programs simultaneously.
- Multiple jobs are executed by the CPU by switching between them, but the switches occur so frequently. Thus, the user can receive an immediate response.
- A time-shared system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time-shared computer.
- It allows many users to share the computer resources simultaneously.

- As the system switches rapidly from one user to the other, a short time slot is given to each user for their executions.
- The time-sharing system provides the direct access to a large number of users where CPU time is divided among all the users on scheduled basis.
- The OS allocates a set of time to each user. When this time is expired, it passes control to the next user on the system. The time allowed is extremely small and the users are given the impression that they each have their own CPU and they are the sole owner of the CPU. This short period of time during that a user gets attention of the CPU; is known as a time slice or a quantum.



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

As soon as the time slice of user 5 is completed, the control moves on to the next ready user i.e. user 6. In this state user 2, user 3, user 4, and user 5 are in waiting state and user 1 is in ready state. The process continues in the same way and so on.

The time-shared systems are more complex than the multi-programming systems. In time-shared systems multiple processes are managed simultaneously which requires an adequate management of main memory so that the processes can be swapped in or swapped out within a short time.

Advantages:

1. They minimize the **response time** i.e; greater the response time, lesser is the efficiency of the system.
2. They reduce CPU **idle** time.

Disadvantages:

1. Security and **Synchronization** must be given a greater importance.
2. Data communication must be enabled.

Example: the Multics & Unix operating systems