2.1 OPERATING SYSTEM OBJECTIVES AND FUNCTIONS

1- The OS provides essential services like:

- Accounting: the OS provides statistics of different resources in addition to monitoring the performance parameters like response time.
- Program development: the OS provides tools and facilities that help the programmers in creating programs.
- Program execution: in order for a program to be executed, it must be loaded into the main memory, files must be initialized, and resources must be allocated. The OS handles all these tasks.
- Error detection and response
- System access
- Controlled access to files

2- A kernel is a core component in the OS that manages the system's resources. It contains the most frequently used functions of the OS. It is saved in the main memory.

3- The resources managed by the OS are:

- I\O devices
- Memory
- Processor
- Files

2.2 THE EVOLUTION OF OPERATING SYSTEMS

4- To meet the memory protection requirements and provide privileged instructions. The user program runs in a user mode in which it has no access to certain portions of the memory and in which certain instructions can’t be executed. On the other hand, the monitor executes in the kernel mode in which it has full access to all portions of the memory and can execute privileged instructions.

5- It is a technique in which the system clock generates an interrupt at a rate of one every 0.2 seconds. At each interrupt, the OS regains control and assigns the processor to another user.

6-

a) In serial processing system, each user had to sign up his name in a haed copy sheet to book the computer. Sometimes a user booked the computer for an hour and finished in less than an hour. This left the computer unused for the rest of the time. Sometimes the hour was not enough, and the user had to sign off before finishing the execution of his job.

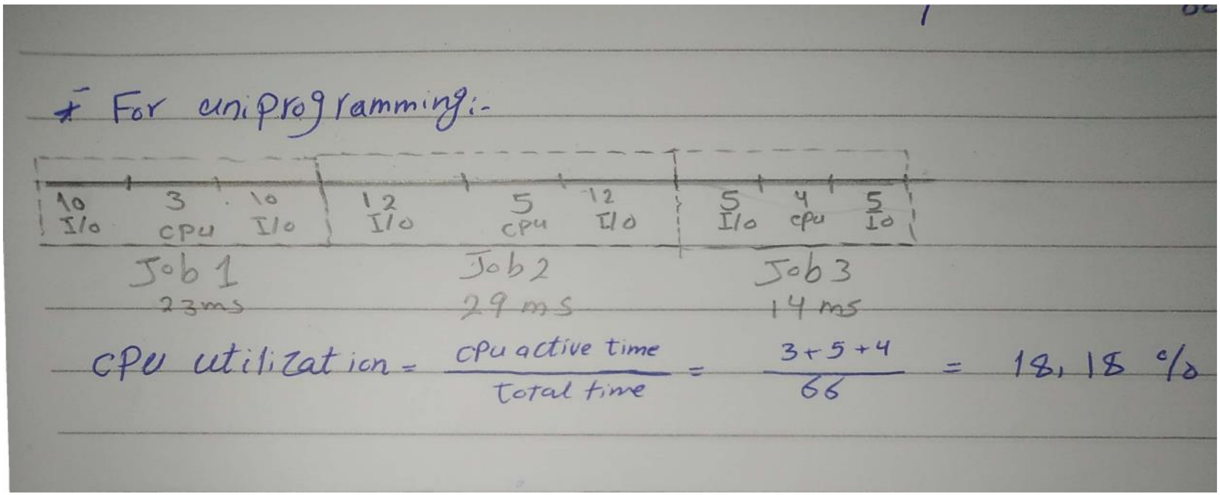
b) In simple batch systems, a piece of software called the monitor was used to control the execution of different jobs. In this system, jobs were batched together sequentially for the monitor to process them one by one and as rapidly as possible. When a job finishes execution, the control is branched back to the monitor which prepares the next job for execution.

c) In multiprogramming system, multiple programs are allowed to run concurrently with the processor switching between them. This system increases the utilization of the resources which yields higher efficiency.

d) The time-sharing technique provides a mode in which the user can interact directly with the computer. In this technique, the processor is shared among multiple users and the processor serves each user for a burst of time then switches to other user. This switch is not observable by the users.

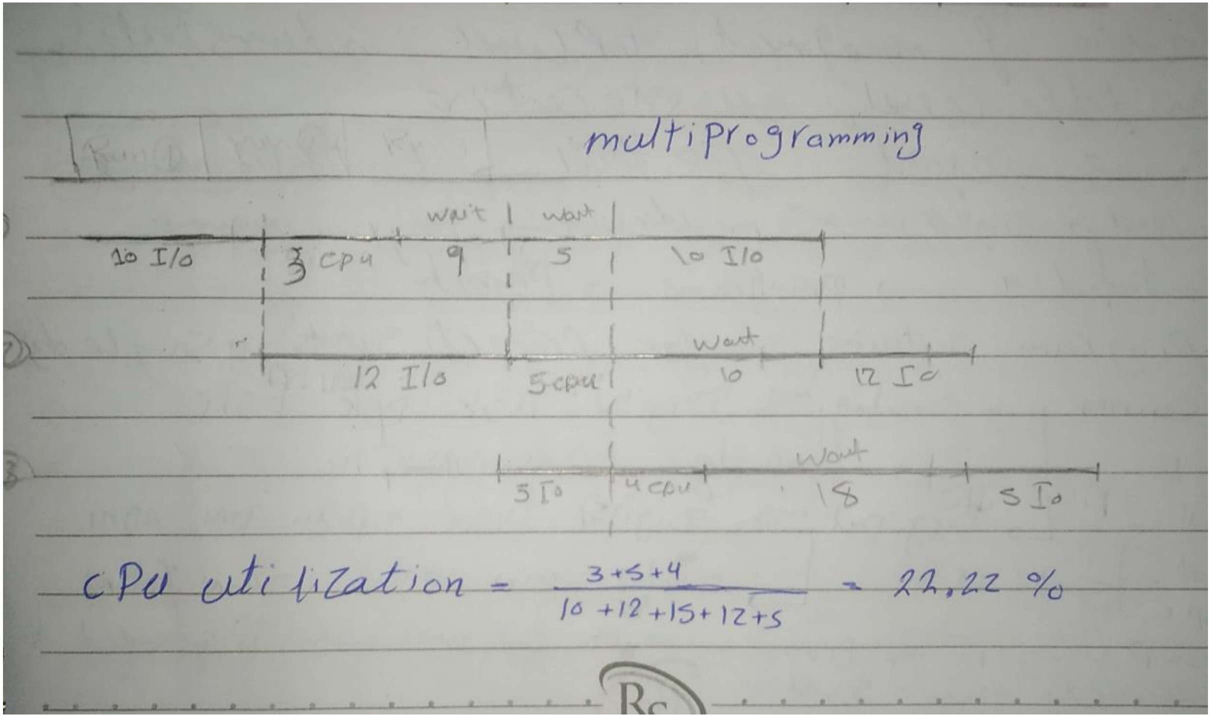
e) As an example of real-time systems is the transaction processing systems in which multiple users are entering queries or updates against a database as in the airline reservation system.

7- In simple batch systems, the three jobs are submitted to a computer operator who batches the jobs together sequentially and places the entire batch on an input devices. This batch will be controlled by the monitor. When a job finishes execution, the control is branched back to the monitor who begins loading the next job automatically.



* For uniprogramming:-

$$\text{cpu utilization} = \frac{\text{cpu active time}}{\text{total time}} = \frac{3+5+4}{66} = 18,18\ \%$$



multiprogramming

$$\text{cpu utilization} = \frac{3+5+4}{10+12+15+12+5} = 22,22\ \%$$

2.3 MAJOR ACHIEVEMENTS

8- The main types of errors include:

- Synchronization problems

- Failed mutual exclusion.
- Deadlocks
- Non-determinate program operation

9- A process consists of 3 components:

- An executable program
- Program data
- The execution context

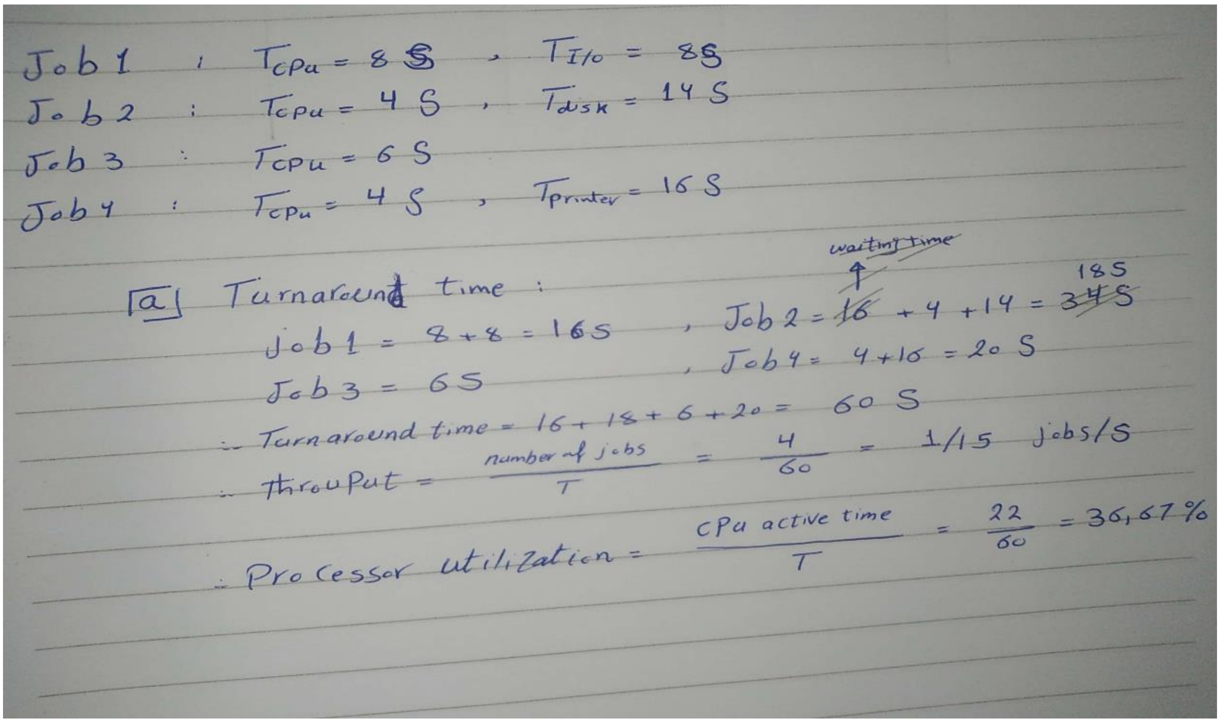
10 –

- Process isolation: the OS must prevent independent process form interfering with each other.
- Long-term storage: many application programs require means for storing information for a long time even if after the computer has powered down.
- Protection and access control: in systems that share the same memory, a program shouldn’t be able to address the memory space of another one. The OS makes sure this never happens.
- Support for modular programming: programmers should be able to define modules and to dynamically allocate, destroy or alter the size of the modules.
- Automatic allocation and management: programs should be dynamically allocated. This allocation is transparent to the user. The OS is responsible for assigning memory to jobs only as needed.

11 –

- Fairness
- Efficiency
- Differential responsiveness

12-



Job 1 : $T_{CPu} = 8$ S , $T_{I/O} = 8$S

Job 2 : $T_{CPu} = 4$ S , $T_{disk} = 14$ S

Job 3 : $T_{CPu} = 6$ S

Job 4 : $T_{CPu} = 4$ S , $T_{printer} = 16$ S

[a] Turnaround time :

Job 1 = 8 + 8 = 16S , Job 2 = $\overset{\text{waiting time}}{\cancel{16}} + 4 + 14 = \overset{18S}{\cancel{34}}$S

Job 3 = 6S , Job 4 = 4 + 16 = 20 S

∴ Turnaround time = 16 + 18 + 6 + 20 = 60 S

∴ throuPut = $\frac{\text{number of jobs}}{T} = \frac{4}{60} = 1/15$ jobs/S

∴ Processor utilization = $\frac{\text{CPu active time}}{T} = \frac{22}{60} = 36,67\%$

2.4 DEVELOPMENTS LEADING TO MODERN OPERATING SYSTEMS

13-

- Monolithic kernel: in this architecture, all the services provided by the kernel are stored in the same memory space. In other words, it’s a large process running in a single space where all the services are stored. These services are stored as a single binary file. This architecture is hard to maintain, and new features or upgrades can’t be added easily. On the other hand, since all the services are in the same memory space, communication between them is very fast and efficient.
- Microkernel: in this architecture, only a few essential functions are assigned to the kernel. Other services are provided by processes that run in the user mode.

14-

- In multiprogramming, multiple programs are allowed to run concurrently, but only one process can be served by the processor at a time. Then after a short period it switches to another process and so on until all the processes are completely served. This mode was developed to maximize the utilization of different resources and increase the efficiency of the system.
- In multiprocessing mode, simultaneous execution of multiple programs at the same time is possible with each processor executing a different program.

2.5 FAULT TOLERANCE

15-

- Process isolation: processes are isolated from each other in terms of memory, file access, and flow of control. This allows a faulty process to run in isolation from other processes.
- Checkpoints and rollbacks: a checkpoint is a copy of an application's state save in a memory space that is immune to failures. When a fault occurs, the OS makes a rollback to a previously saved checkpoint.

2.6 OS DESIGN CONSIDERATIONS FOR MULTIPROCESSOR AND MULTICORE

16-

- Parallelism within applications: most applications can be subdivided into multiple processes, perhaps each with multiple threads.
- Virtual machines approach: in this approach, one or more cores are to be dedicated to one process. In this way, much of the

overhead consumed in the switching between processes is avoided.