**Operating System Theory**

**Team Activity**

|  |  |
| --- | --- |
| **Name** | **Roll Number** |
| Smit | 21BCP142 |
| Shivam | 21BCP156 |
| Saurav | 21BCP157 |
| Aditya | 21BCP166 |

Multitasking Operating System

**Q. Explain About Multitasking Operating System.**

A multi-tasking operating system is one that is capable of running multiple tasks, or processes, simultaneously. This means that the operating system can switch quickly between different tasks, giving the illusion that they are all running at the same time. This is in contrast to a single-tasking operating system, which can only run one task at a time.

Some examples of multi-tasking operating systems include Windows, Linux, and macOS. These operating systems use a technique called time-sharing to enable multiple tasks to be run simultaneously. Time-sharing works by dividing the available processing time into small slices, and allocating each slice to a different task. This way, each task gets a fair share of the processing time, and the operating system can switch between tasks quickly enough that it appears as though they are all running at the same time.

Multi-tasking also include different type of scheduling algorithms like, First In First Out(FIFO), Priority scheduling, Round Robin Scheduling, Multilevel queue Scheduling, Multilevel Feedback Queue Scheduling.

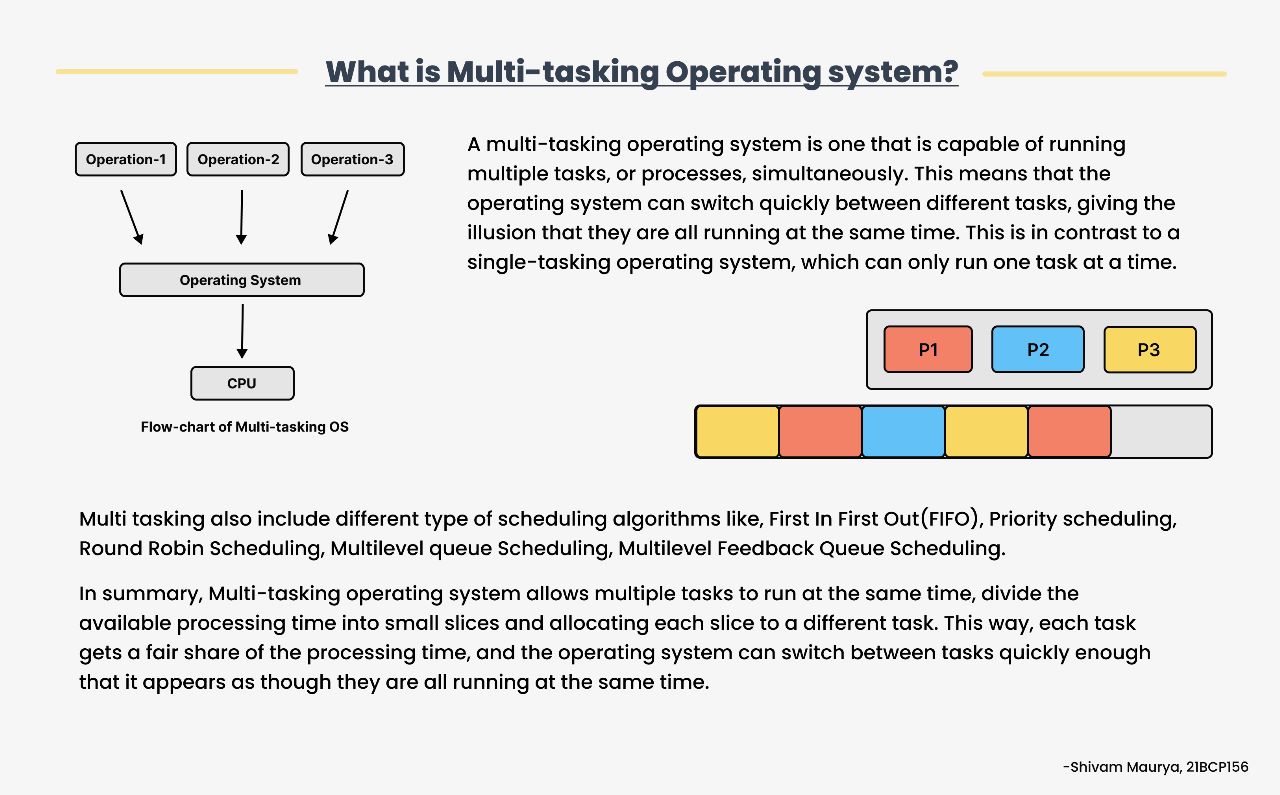


Figure : Made by Shivam Maurya (21BCP156)

Multi-tasking operating system also allows for background tasks to run, such as file indexing, software updates, virus scans, and more, without interfering with the normal operation of the system. Additionally, multi-tasking operating systems typically include memory management and process management features that help ensure that each task has the resources it needs to run efficiently.

In summary, multi-tasking operating system allows multiple tasks to run at the same time, divide the available processing time into small slices and allocating each slice to a different task. This way, each task gets a fair share of the processing time, and the operating system can switch between tasks quickly enough that it appears as though they are all running at the same time.

There are several reasons why a multi-tasking operating system is considered a must for modern computer systems:

Productivity: A multi-tasking operating system allows users to run multiple applications at the same time, which increases productivity. For example, you can write a document while also listening to music and checking your email, without having to switch back and forth between different applications.

Resource management: multi-tasking operating systems include memory management and process management features that help ensure that each task has the resources it needs to run efficiently. This improves the overall performance of the system, and prevents one task from monopolizing resources and slowing down the entire system.

Background tasks: A multi-tasking operating system allows background tasks to run, such as file indexing, software updates, virus scans, and more, without interfering with the normal operation of the system.

Real-time requirements: In some cases, certain application or tasks have real-time requirements, such as in the case of control systems, embedded systems, and other systems where timely processing is critical. Multi-tasking allows such process to be carried out efficiently with the prioritization of such task over non-real-time task.

Better User Experience: Multitasking operating system provide users with a better and more efficient experience, as they can perform multiple tasks simultaneously, rather than having to switch between different applications. This can lead to increased satisfaction and a more seamless workflow.

In short, a multi-tasking operating system is essential for modern computer systems because it increases productivity, improves resource management, allows for background tasks to run, provides Real-time requirements support and improves overall user experience.