Types of operating systems

Introduction

Previously, you became familiar with the various hardware and software components that comprise a computer, and the functions they serve. To manage these elements, you would need some software that will oversee the interactions between the software, hardware, and human operator, as well as scheduling the executions of the various tasks required. This is where operating systems enter the picture.

An Operating System (OS) is a term for software that oversees the interaction between hardware and software operations and provides a means through which a human can interact with the system. Systems will have different needs and functionalities, so there are several variations of operating systems that can run the application. By the end of this reading, you'll be able to describe several different operating systems and identify their advantages and drawbacks.

Types of Operating Systems

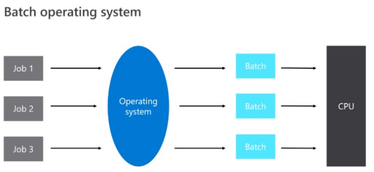
There are numerous operating system types, in this reading, you will learn about the five variations, and what distinguishes them from one another. The categories are:

1. Batch Operating Systems
2. Time-sharing Operating Systems
3. Distributed Operating Systems
4. Network Operating Systems
5. Real-Time Operating Systems

While the fundamental principles of what each OS does is the same, the way that each achieves this end is different. Let's examine what distinguishes each approach.

Batch Operating Systems (BOS)

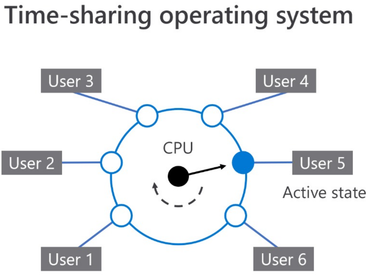
A BOS allows multiple users to work in tandem by compartmentalizing each user's actions and only allowing one user control at any given time. Users cannot communicate with one another and when the task is complete, control is passed to the next user. The advantage of this approach is that it allows many users to work on large projects at the same time.



The rationale behind this approach is that large projects are broken into smaller, more manageable chunks and hence do not require being loaded into memory all at once. Another advantage is that similar types of jobs that are queued to run will be run together. This is quicker as it reduces the overhead in loading and queuing the resources that are mutual to each task. The challenge is in coordinating many different users operating at once. This results in some tasks running slower because excessive time is spent waiting for the OS to free up the resources.

Time-Sharing Operating Systems (TOS)

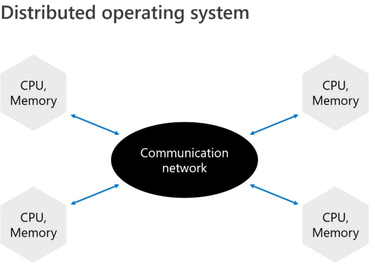
This approach is also known as a Multitasking System as it enables the execution of many different tasks. A TOS operates in a timed manner: a unit of time called a **quantum** is allocated to each task. These tasks are queued for the CPU, which will work on each one for a quantum before rotating to the next task.



This approach is beneficial when the task load is running several small tasks. The major advantage of this approach is that it minimizes CPU wait time, and every user is given a fair time location. The disadvantage to this approach is that the rapid switching between tasks can result in some communication issues, with data being mixed up.

Distributed Operating Systems (DOS)

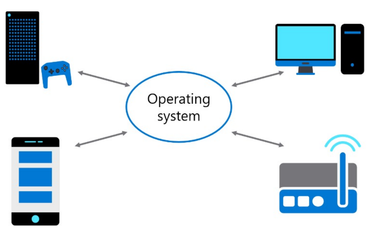
This approach to processing is a natural progression as internet communication improves. A DOS connects several dispersed CPUs to execute tasks. There is no central source and communication is achieved through using clear protocols on a communication network.



The advantage of using a distributed approach is that it can scale up or down depending on the processing need. As the network is dispersed there is greater robustness to failure. A drawback is that it is still a relatively new approach and so the communication protocols are not well defined.

Network Operating Systems (NOS)

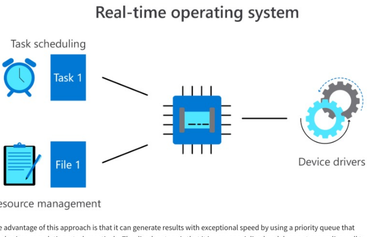
This type of operating system runs on a network and allows for the sharing of users, groups, securities, and applications. Typically, the network would contain one server-based OS that interacts with the OS of each individual device on the system. It is possible to have any number of devices on the network, and they can be of different types. For example, you could add a printer, digital thermometer, and several computers to one system and have them all interact. This type of approach would be considered **tightly-coupled**, as the devices utilize a shared network and demonstrate high interactivity with each other.



The advantage of this approach is that users can remotely log on, enabling easy upgrading as new devices and technology can be added to the existing network. It can however be costly to maintain and requires a centralized location for performing operations.

Real-time Operating Systems (RTOS)

An RTOS is a type of OS for applications that need real-time computations. While many approaches may share resources and operate under a time share, RTOS differs in that it prioritizes tasks by importance. This approach is event-driven with an emphasis on engaging and completing tasks as soon as they arise. This distinguishes RTOS from the other approaches mentioned, in which the priority of a task is determined by the time allocated. These operating systems are generally task-specific approaches and are commonly found in systems such as autonomous cars, air traffic controls, and more.



The advantage of this approach is that it can generate results with exceptional speed by using a priority queue that emphasizes completing a task reactively. The disadvantage is that it is very specialized and does not generalize well to performing a diverse range of tasks.

Conclusion

In this reading, you've analyzed various categories of operating systems, specifically batch operating, time-sharing, distributed, network, and real-time systems. You're now aware that each one has its advantages and disadvantages, and there is no single best approach. You should know that different operating systems exist for different tasks, and knowing how to distinguish between them will help with decision-making in future cases.

Go to next item ✓ Completed

Like Dislike Report an issue