1. We need to learn the Operating System because the operating system is a very important and fundamental software in computers, which is responsible for organizing and managing computer resources such as memory, CPU, storage and input/output devices. Understanding the operating system is important to be able to create applications or programs that can run well on it, as well as to optimize the performance of the operating system itself.
2. Operating systems are very important in today's digital era because almost all digital devices that we use daily such as smartphones, laptops, and tablets use operating systems. Operating systems allow us to access and use hardware and software easily and efficiently, making it easier for us to work, study, and socialize in the digital age.
3. In the future, Operating Systems are likely to be increasingly integrated with more advanced technologies such as artificial intelligence, Internet of Things (IoT), and virtualization technologies. Operating systems will also be further optimized to improve security, privacy, and performance efficiency on increasingly complex digital devices.
4. An Operating System is a platform or environment that provides the services and APIs (Application Programming Interface) needed to create applications or programs. Programmers need to understand how the operating system manages resources and how programs are managed.
5. Windows NT has a layered architecture consisting of user mode and kernel mode. User mode programs and subsystems have limited access to system resources, while kernel mode has unrestricted access to hardware and system resources. The Executive manages I/O, object, security, and process management. The kernel provides synchronization, scheduling, and dispatching functions, and initializes device drivers at bootup. There are three main environment subsystems: Win32, OS/2, and POSIX, which can't directly access hardware and rely on kernel mode routines. Windows NT is a preemptive, reentrant multitasking operating system designed to work with uniprocessor and symmetrical multiprocessor-based computers. The Windows NT operating system has several environment subsystems, including the Win32 subsystem for running 32-bit Windows applications, the OS/2 subsystem for running 16-bit character-based OS/2 applications, and the POSIX subsystem for running applications written to the POSIX.1 standard. The security subsystem deals with user authentication, resource permissions, and network file and print sharing. Kernel mode is the privileged mode in which the operating system kernel runs. In this mode, the kernel has complete access to the hardware and system resources of the computer and controls access to scheduling, thread prioritization, memory management, and interaction with hardware. User mode processes must ask the kernel to perform operations on their behalf. The x86 architecture supports four different privilege levels, numbered 0 to 3, but only two levels, CPL 3 (user mode) and CPL 0 (kernel mode), are used in Windows. The Windows NT kernel mode includes the executive, which is made up of many modules that perform specific tasks, the kernel, which provides low-level services used by the Executive, the Hardware Abstraction Layer (HAL), and kernel drivers. The executive services deal with I/O, object management, security, and process management. These services are divided into several subsystems, including the Cache Manager, Configuration Manager, I/O Manager, Local Procedure Call (LPC), Memory Manager, Object Manager, Process Structure, and Security Reference Monitor (SRM). The Object Manager is an executive subsystem that all other executive subsystems must pass through to gain access to Windows NT resources. Handles are identifiers that represent a reference to a kernel resource through an opaque value. Similarly, opening an object through its name is subject to security checks, but acting through an existing, open handle is only limited to the level of access requested when the object was opened or created. The Cache Controller closely coordinates with the Memory Manager, I/O Manager, and I/O drivers to provide a common cache for regular file I/O. The Configuration Manager implements the system calls needed by the Windows Registry, and the I/O Manager allows devices to communicate with user-mode subsystems. The Windows NT operating system uses a hybrid kernel design that includes a collection of modules that communicate via well-known interfaces, with a small microkernel limited to core functions such as first-level interrupt handling, thread scheduling, and synchronization primitives. The Windows API is the primary operating system personality, implemented via a set of user-mode DLLs, while kernel-mode device drivers enable interaction with hardware devices. The Windows NT hardware abstraction layer (HAL) is a layer between the physical hardware of the computer and the rest of the operating system, designed to hide differences in hardware and provide a consistent platform on which the kernel is run. However, the HAL depends in some measure on the kernel or even the Executive, and hardware abstraction does not involve abstracting the instruction set, which is performed by the kernel or via hardware virtualization.

New: When a process is first created, it is in the new state. In this state, the operating system has allocated memory to the process, but the process has not yet started executing.

Running: When the CPU is executing instructions of a process, it is in the running state. At any given time, there can be only one process in the running state.

Waiting: When a process is waiting for an event to occur, such as input/output (I/O) operation or a signal from another process, it is in the waiting state. The process remains in this state until the event occurs.

Ready: When a process is waiting for CPU time to be allocated, it is in the ready state. The process is waiting to be executed and is waiting for the CPU to become available.

Terminated: When a process completes its execution, it is in the terminated state. At this stage, the process is removed from the list of processes and its resources are deallocated by the operating system.