

# Operating System Lab 2

Roll No.: K041	Name: Anish Sudhan Nair
Batch No.: K2/A2	Date: 20/12/2021

Aim: Learning about the basics of various operating systems

2.a.: Compare and contrast monolithic, layered and microkernel strategies for designing operating systems (at least 10 differences)

Monolithic	Layered	Microkernel
The entire operating system works in the kernel space.	The OS architecture is divided into several layers, each layer having its own functionality.	It has a kernel that provides mechanisms such as low-level address space management, thread management and interprocess communication to implement an operating system.
There are 3 main layers - application, monolithic kernel, hardware layer	There are 6 main layers - hardware, CPU layer, memory management, process management, IO buffer, user programs. It overcomes the limitations of the monolithic operating systems.	It is not layered. The microkernel provides just a few essential services, and the rest of the operating system is turned into separate user-level programs.
Every application has its own address space. The OS is in a separate address space.	Applications run in the user program layer.	Each application and service has its own address space.
Applications are more secure. The OS is secure from the normal applications and malfunctioned applications.	It is secure since it has implementation of access privilege, wherein a layer can access lower layers but not the higher ones.	Applications and OS services are secure.
The kernel handles the OS services including the file system, memory manager and CPU scheduler.	It consists of layers in place of kernel architecture.	The OS Services and the kernel are separated.

The applications request services from the kernel via system calls.	When one layer wishes to interact with another, it sends a request that must traverse all layers between the two layers to be fulfilled.	The Inter Process Communication (IPC) helps to establish the communication between the application and the OS services.
When an application requests a service, the hardware address space of the application switches to the hardware address of the operating system to execute it.	When a service is required, the request passes through the lower or equal layers	When the application requests the OS services for a service, the OS services communicate with each other to provide the required service to the application.
It is possible to add device drivers to the kernel as modules.	The hardware layer interacts with the i/o and the peripherals, the required programs run here while also isolating it from the user layer.	It is possible for things like file systems and device drivers to be user-level programs. When modules are loaded they become part of kernel address space.
Failure in one component will affect the entire system.	Failure in one layer does not affect the entire system, only the functionalities of that layer.	Failure in one component will not affect the other components.
Difficult to add new functionalities.	Code in one layer, can make calls to functions in other layers, but only if those functions are in the same or lower layers. At the same time, modification in one layer doesn't affect others, so it is easier to add functionalities.	Easier to add new functionalities. It is more customizable.
Larger in size	Has no kernel.	Smaller in size.
Fast. The monolithic kernel runs quickly because of memory management, file management, process scheduling, etc. These are all implemented in the	It is much slower since the requests need to travel through multiple layers. As a result, increasing the	Microkernels will be slower because of the message passing protocol which needs to be followed to interact with the rest of the kernel.

same address space.	number of layers may lead to a very inefficient design.	
---------------------	---	--

2. b.: Compare Multi-User, Multiprocessing, Multitasking, Multithreading and Real Time Operating systems (5 to 10 points)

Multi-User	Multiprocessing	Multitasking	Multithreading	Real Time
A multi-user operating system is an operating system that permits several users to access a single system running to a single operating system.	The availability of more than one processor per system, which can execute several sets of instructions in parallel is called multiprocessing.	The execution of more than one task simultaneously is called multitasking.	A process is divided into several different sub-processes called threads, which has its own path of execution. This concept is called multithreading.	A real-time operating system (RTOS) is an operating system (OS) for real-time applications that processes data and events that have critically defined time constraints.
Large machines have more ICS for CPU while smaller ones have a single microprocessor .	Has more than 1 CPU.	Has 1 CPU.	Can have 1 or more than 1 CPU.	Can range from multiple processors on guidance systems to single chip embedded systems.
Has several users.	Can have more than 1 user.	Has more than 1 user.	Has 1 user usually.	Has a large number of users.
Maximum efficiency	Maximum efficiency	Moderate efficiency	Moderate efficiency	Maximum efficiency
'Simultaneous Peripheral Output on Line' or spooler runs all computer processes and outputs the results at the same time.	More than one process can be executed at a time	One by one, the jobs are executed.	Various components of the same process are being executed at a time.	Very few tasks run simultaneously, and multitasking operation is accomplished by scheduling processes for

				execution independently of each other..
Less time is taken for job processing.	Less time is taken for job processing.	Moderate amount of time.	Moderate amount of time is taken for job processing.	Minimum time is taken for job processing.