

- 1 Compare and contrast microkernel and monolithic kernel-based operating systems. Name one kernel that follows each of these models.

Ans A microkernel implements user and kernel services in different address spaces, whereas a monolithic kernel system uses the same address space. This separation makes microkernels easily extendable with new services being added to its user space. The monolithic kernel requires the entire kernel to be rebuilt after adding new services. Since only the kernel services are located in its address spaces, the microkernels are comparatively smaller than a monolithic kernel. Also, if a service fails in a microkernel, the operating system remains unaffected. Monolithic kernels, however, execute faster as they use system calls to communicate between application and hardware. Microkernels rely on the relatively slower message passing to communicate. Performance of microkernels are also affected by increased system-function overhead. Monolithic kernels also require lesser code in their design than microkernels.

- 2 What is the purpose of a system call? Give at least 3 concrete examples of potential system calls in an OS and explain why each would be a system call.

Ans

- 3 Describe what a context switch is. How does context switching differ between processes and between threads within a single process?

Ans

- 4 Name two methods of IPC (interprocess communication) and discuss the pros and cons of each.

Ans

- 5 What is the difference between a program being executed by a single thread and one that is being executed by multiple threads? Compare and contrast task parallelism and data parallelism as it relates to multithreaded programs.

Ans