Q-1 What are the advantages and disadvantages of using the same systemcall interface for manipulating both files and devices?

Ans: Advantages - When we use same system call interface for manipulating file and devices, some devices can be thought of as abstract or virtual devices (for example, files). Once the device has been requested, we can read(), write(), and reposition() the device, just as we can with files. In fact, the similarity between I/O devices and files is so great that many operating systems, including UNIX, merge the two into a combined file–device structure.In this way, a set of system calls is used on both files and devices.The user interface can also make files and devices appear to be similar, even though the underlying system calls are dissimilar.

Disadvantage - The disadvantage of using the same interface is that there is a possibility of loss of functionality or loss of performance because it might be difficult to capture the functionality of certain devices within the context of the file access API.

Q-2 Would it be possible for the user to develop a new command interpreter using the system-call interface provided by the operating system?

Ans: Yes, the user should be able to develop a new command interpreter using the system-call interface provided by the operating system. This is possible in the approach used by UNIX because in this case, the command interpreter does not understand the command in any way; it merely uses the command to identify a file to be loaded into memory and executed.

Q-3 What are the two models of interprocess communication? What are the strengths and weaknesses of the two approaches?

There are two common models of interprocess communication -

Ans: Message passing model - In this model, the communicating processes exchange messages with one another to transfer information. Messages can be exchanged between the processes either directly or indirectly through a common mailbox. Message passing is useful for exchanging smaller amounts of data and also easier to implement. But the weakness of message passing is it can handle only small amounts of data.

Shared-memory model - In this model, processes use shared memory creates and shared memory attaches system calls to create and gain access to regions of memory owned by other processes.Two or more processes can exchange information by reading and writing data in the shared areas. Shared memory allows maximum speed and convenience of communication since it can be done at memory transfer speeds when it takes place within a computer.But shared memory has problems of protection and synchronization between the processes sharing memory.

Q-4 What is the main advantage of the microkernel approach to system design? How do user programs and system services interact in a microkernel architecture? What are the disadvantages of using the microkernel approach?

Ans: Advantage of the microkernel approach - Microkernel approach makes extending the operating system easier. Addition of new service does not require modification of the kernel. When the kernel does have to be modified, the changes tend to be fewer, because the microkernel is a smaller kernel. The resulting operating system is easier to port from one hardware design to another. The microkernel also provides more security and reliability because more operations are done in user mode than in kernel mode.

User programs and system services interact in a microkernel architecture through message passing.

Disadvantages of using the microkernel approach - Unfortunately, the performance of microkernels can suffer due to increased system-function overhead.