Q.1 What are the two main functions of an operating system ?

=>> An operating system has three main functions: (1) manage the computer's resources, such as the central processing unit, memory, disk drives, and printers, (2) establish a user interface, and (3) execute and provide services for applications software.

Q.2 What is multiprogramming?

=>> Multiprogramming is a rudimentary form of parallel processing in which several programs run at the same time on a uniprocessor system. However, because there is only one processor, there is no true simultaneous execution of different programs.

Q.3 List some differences between personal computer operating systems and mainframe operating systems.

=>> Mainframes typically run on large boxes with many processors and tons of storage, as well as high-bandwidth busses. PCs are desktop or mobile devices with a single multi-core processor and typically less than 32GB of memory and a few TBs of disk space. Second, a mainframe OS usually supports many simultaneous users.

Q.4 What is the key difference between a trap and an interrupt?

=>> The difference between a trap and an interrupt is that a trap is triggered by a user program to invoke OS functionality. Still, an interrupt is triggered by a hardware device to allow the processor to execute the corresponding interrupt handler routine.

Q.5 On early computers, every byte of data read or written was directly handled by the CPU (i.e.there was no DMA. What implications does this organization have for multiprogramming ?)

=>> It makes multiprogramming less favorable since it is no longer the case that when one process does I/O the CPU is completely free to work on other processes.

Q6. Which of the following instructions should be allowed only in kernel mode?

(a) Disable all interrupts.

(b) Read the time-of-day clock.

(c) Set the time-of-day dock.

(d) Change the memory map.

=>> In kernel mode, the executing code has complete and unrestricted access to the underlying hardware. It has the ability and permission to execute any of the CPU instructions and reference any memory address. The kernel mode is reserved for the lowest-level and the most trusted functions of the operating system. Hence it has the access to all the necessary functions to disable all interrupts when needed, set the time-of day clock and change the memory map as and when required, and these instructions should be allowed only in kernel mode.

These instructions mentioned above should not be accessed in user mode, but the reading of the time-of-day clock should be allowed in both the user and kernel mode.

**the answer is (a), (c) and (d).**

Q7. Can the

count = write(fd, buffer, nbytes);

call return any value in *count* other than *nbytes*? If so, why?

=>> Can the count = write (fd, buffer, nbytes); Call return any value in count other than nbytes? If so, why? If the call fails, for example because fd is incorrect, it can return −1. It can also fail because the disk is full, and it is not possible to write the number of bytes requested.

Q8. A file whose file descriptor is *fd* contains the following sequence of bytes: 3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5. The following system calls are made:

lseek(fd, 3, SEEK\_SET);

read(fd, &buffer, 4);

where the lseek call makes a seek to byte 3 of the file. What does *buffer* contain after the read has completed?

=>> the answer is 1,5,9,2

Q9.

A computer uses the relocation scheme of Fig. 1-9(a). A program is 10,000 bytes long and is loaded at address 40,000. What values do the *base* and *limit* register get according to the scheme described in the text?

=>> 50000