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

An operating system provides the users with an extended (i.e., virtual) machine, and it must manage the I/O devices and other system resources.

Q.2 What is multiprogramming?

Multiprogramming is the rapid switching of the CPU between multiple processes in memory. It is commonly used to keep the CPU busy while one or more processes are doing I/O. It also allows two or more people to use the same computer simultaneously. Furthermore, it permits a single user to start up multiple, independent processed at the same time.

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

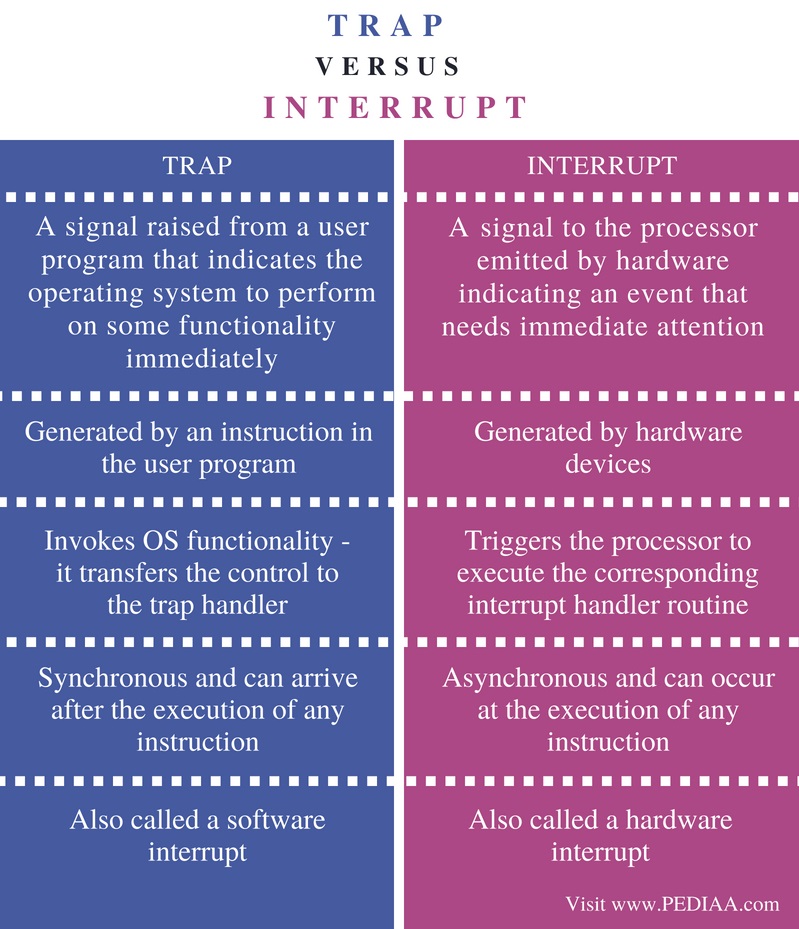
**Mainframe computers :**

* **Mainframes are typically big boxes containing a large number of processors and a large amount of storage, as well as high-bandwidth buses.**
* **A mainframe operating system is meant to handle a huge number of calculations in a sequential manner**, whereas a PC operating system is designed to process interactive transactions.

**Personal computers (PCs) :**

* **The hardware of a personal computer is typically designed to deliver speedy responses to the user.**
* These are usually graphical or console interfaces on top of an operating system's kernel and are designed for a lot of user interaction.

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



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.

a. Disable all interrupts.  
c. Set the time-of-day clock.  
d. Change the memory map.

Q7. Can the

count = write(fd, buffer, nbytes);

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

If it fails it will return -1.  
Possible if bad fd (file descriptor)  
Otherwise, will always return nbytes

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?

It contains the bytes: 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?