Assignment 1

1) In general terms what are four distinct actions that a machine instruction can specify? 20%
   - In general terms, a machine instruction can specify four distinct actions:

   - Data movement: This action specifies the movement of data between different memory locations or registers within the computer.
   - Arithmetic and logical operations: This action specifies operations such as addition, subtraction, multiplication, and logical comparisons to be performed on data stored in memory or registers.
   - Control flow: This action specifies the order in which instructions are executed, including the ability to branch to different parts of the program or subroutines based on certain conditions.
   - Input/Output operations: This action specifies how the computer communicates with the outside world, such as reading data from a keyboard or writing data to a display.

2) Why was the evolution of operating systems to use interrupts important – be specific? 10%
   Give examples of interrupts 10%
   - The evolution of operating systems to use interrupts was important because it allowed for more efficient use of computer resources and better multitasking capabilities. Interrupts allow the computer to respond to external events, such as input from a user or a signal from a peripheral device, without having to constantly poll for those events. This allows the computer to perform other tasks, such as running multiple programs simultaneously, while waiting for an interrupt to occur.

   - Examples of interrupts include:

   - Hardware interrupts: These are triggered by external devices such as keyboard, mouse, network card, etc.

   - Software interrupts: These are triggered by software, such as an operating system call, system exception, or a divide-by-zero error.

   - Timer interrupts: These are generated by a hardware timer and are used to schedule tasks and maintain system time.

3) What are the basic functions of an operating system? 10%
   What does each do? 10%
   - The basic functions of an operating system include:

   - Memory management: This function manages and allocates memory resources to different programs and processes running on the system. It also ensures that each process has enough memory to run and prevents processes from interfering with each other's memory.

- Process management: This function manages and schedules the execution of different processes, ensuring that each process gets a fair share of the CPU time. It also handles process creation, termination, and synchronization.

- File management: This function manages the organization, storage, retrieval, and security of files on the system. It also provides a file system to keep track of the files and their location on the storage.

- Input/Output management: This function manages the communication between the operating system and the various input and output devices connected to the system, such as keyboard, mouse, printer, etc.

- Security: This function enforces security policies and controls access to system resources. It also ensures that only authorized users and processes can access the system and its resources.

- Networking: This function manages the communication between different systems on a network and enables the sharing of resources and information among them.

- Virtualization: This function allows multiple virtual machines to run on a single physical machine and share its resources, providing a isolated environment for each machine.

- Resource management: This function allocates resources like CPU, memory, storage and network bandwidth to different processes and services running on the system and monitors their usage.

4) What are the differences between user mode and kernel mode? 20%

- In user mode, a program runs with limited access to system resources, such as memory and CPU instructions. This mode is intended for normal, everyday applications, such as word processors and web browsers, which do not require direct access to the underlying hardware.

- In contrast, kernel mode, also known as system mode, is a privileged mode of operation in which the operating system kernel and device drivers run. Programs running in kernel mode have unrestricted access to system resources and can execute any CPU instruction. This mode is typically used for low-level, system-level tasks, such as managing memory and handling interrupt requests from hardware devices.

- The transition between user mode and kernel mode is typically controlled by the operating system, which switches the CPU between the two modes based on the needs of the running program.

5) Why multiprogramming better than uniprogramming? 20%

- Multiprogramming allows multiple programs to run simultaneously on a computer, whereas uniprogramming only allows one program to run at a time. This can lead to increased efficiency and responsiveness for the user, as the computer can switch between different tasks, rather than having to wait for a single task to complete. Also, multiprogramming can help to prevent a single malfunctioning program from bringing the entire system down.