Quiz 1

• Describe in few line, what is meant by **a multiprogramming environment**. Emphasize the task of the Operating Systemin such an environment.

• What is the main different between a **timesharing** and a **multiprocessing** environment?

• What is **batch file**?

• What is **embedded** system?

1.1

Multiprogramming is a term used to describe how a computer can run more than one application at the same time by dividing up its memory.

(one CPU, multiple processes ready for execution.)

1.2

Multiprocessing : several processors are used on a single computer system to increase the processing power of the machine.

Time sharing is a way of sharing out computer facilities between a number of people who want to use the computer at the same time. Each has a separate terminal and gets the impression that they have sole use of the computer with their own 'account'.

1.3

A batch file is a text file containing a series of commands intended to be executed by the command interpreter or sequence of commands.

1.4

An embedded system is a computer system designed for specific control functions within a larger system, often with real-time computing constraints.

What do you understand by a **Dual-Mode** processor? What would be the **REASON** for using a dual-mode processor?

Give two example each of synchronous interrupt and asynchronous interrupt.

Give the connection (relation) and different between **privileged instruction** and a **system call**.

Name **real-time operating systems** and give a few of their characteristics.

(1)

Dual-mode is the distinction between execution of user mode and kernel mode (supervisor mode, system mode, and privileged mode). A mode bit is added to the hardware to indicate the current mode: Kernel(0) or user(1).

Protection of the OS and the user processes.

(2)

Synchronous: division by 0 will create interrupt, underflow or overflow will create

interrupt, illegal reference, wait(), exit().

Asynchronous: from operating system, come from hardware, scheduler timer.

(3)

Privileged instructions are instructions that can be executed only in monitor mode. Privileged instruction will be checked by the OS.

System call is the interface between user process and OS (Kernel). OS calls the system. Any access to the main memory is system call. Everything goes through OS.

(4)

Real-Time Systems are used when there are rigid time requirements. A real-time operating system has fixed time constraints. Processing must be done within the given amount of time.

Hard real-time system: guarantees that critical task is done in time

Soft real-time system: a critical real-time task gets priority over other tasks

What is the task of a **bootstrap program** and where is this program stored? What is a **boot record**?

What are the main characteristics of **modern operating system architecture**?

Pick a version of Windows mentioned in your homework and give its characteristic.

Give some characteristics and responsibilities of the **Windows microkernel**.

(1)

when turn on the computer, a set of instructions (thats in the ROM) is executed. 1.power-on self test (POST) check the hardware.

2.BiOS seeks instruction on the master boot record.

3.Bootstrap starts executing, loads in the main memory. the kernel of the OS.

(2)

modularity:: interface with the hardware.

horizontal : communication by messages and peers together.

use of microkernel concept load from different windows will still work(HAL)

executive: have different module. peers between them. communicate by messages, control by kernel.

(3)

Win 32: interface between user mode and kernel. manages all keyboard, mouse, screen, I/O operation. starts all processes.

(4)

Microkernel represents the most used and fundamental component of the OS. The microkernel functions as a message exchange: validates messages, passes them between components, grants access to the hardware, and performs a protection function.

3.1 What is a **TSR** process? What is the **advantage (reason)** for using TSR

processes in current Operating System?

3.2 What do you understand about a Unix **external command**?

3.3 Give examples of **shells**. In what mode the UNIX shell runs?

3.4 In what **programming language** is modern UNIX written?

3.1

Terminate and Stay Resident (TSR) is a computer system call in DOS operating systems that returns control to the system as if the program has quit, but keeps the program in memory.

Efficient if run very often.

3.2

External commands are the commands that are executed by the kernel. These commands will have a process id running for it..

UNIX external commands will start a new shell and then it gets executed

eg: ls

3.3

Example : C shell, Z shell, TC shell

All the processors run in system mode and user mode.

3.4

First UNIX was written assembly language but now it’s written in C.