Week1 Questions

1. What are the three main purposes of an operating system?

2. What is the main difficulty that a programmer must overcome in writing an operating system for a real-time environment?

3. In a multiprogramming and time-sharing environment, several users share the system simultaneously, which can result in various security problems.

    3.1. What are two such problems?
    3.2. Can we ensure the same degree of security in a time-shared machine as in a dedicated machine? Explain your answer.

4. What is the purpose of interrupts? How does an interrupt differ from a trap? Can traps be generated intentionally by a user program? If so, for what purpose?

5. Define the following: multiprogramming, multitasking, and multiprocessing

6. Distinguish the terms interrupt, trap (system call), and mode bit.

7. Give two reasons why caches are useful. What problems do they solve? What problems do they cause? If a cache can be made as large as the device it is caching (for instance, a cache as large as a disk), why not make it that large and eliminate the device?

8. Describe the differences between symmetric and asymmetric multiprocessing. What are the three advantages and disadvantages of multiprocessor systems?

9. Many SMP systems have different levels of caches; one level is local to each processing core, and another is shared among all processing cores. Why are caching systems designed this way?

10. A three-process execution paradigm is shown in **Figure 1**. Answer the following based on the figure:

    10.1.   Is this multiprocessing or multiprogramming? Why?
    10.2. Can the processes execute without utilizing the 'wait' state? Briefly describe your reasons.

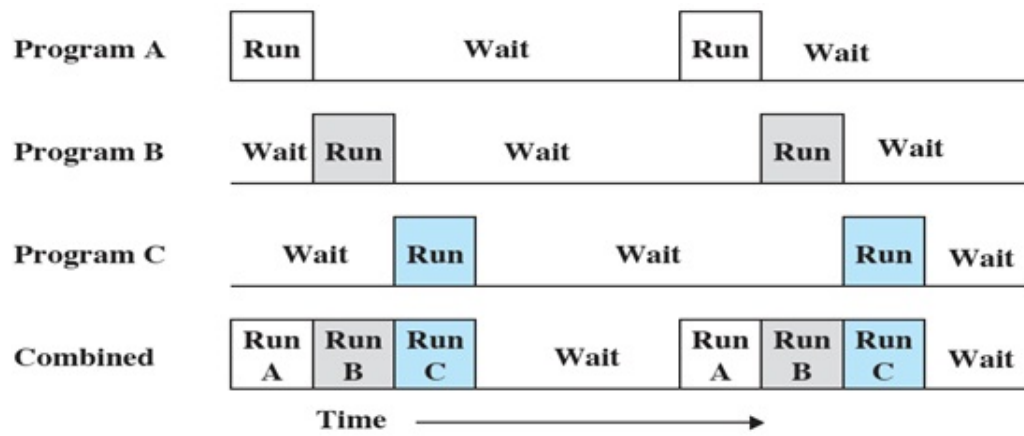10.3. Is there any system call involvement in this process's execution? Why?

Program A	Run	Wait	Run	Wait
Program B	Wait Run	Wait	Run	Wait
Program C	Wait	Run	Wait	Run Wait
Combined	Run A Run B Run C	Wait	Run A Run B Run C	Wait

Time →

Figure 1.