**Instructions**

Provide concise answers to the following questions. Submission must be a Word or PDF document and must contain the question text above your answer.

**Chapter 1 Questions**

1. What are the three main purposes of an operating system?
2. We have stressed the need for an operating system to make efficient use of the computing hardware. When is it appropriate for the operating system to forsake this principle and to “waste” resources? Why is such  
   a system not really wasteful?
3. What is the main difficulty that a programmer must overcome in writing an operating system for a real-time environment?
4. Keeping in mind the various definitions of operating system, consider whether the operating system should include applications such as web browsers and mail programs. Argue both that it should and that it should not, and support your answers.
5. How does the distinction between kernel mode and user mode function as a rudimentary form of protection (security)?
6. Which of the following instructions should be privileged?
   1. Set value of timer.
   2. Read the clock.
   3. Clear memory.
   4. Issue a trap instruction.
   5. Turn off interrupts.
   6. Modify entries in device-status table.
   7. Switch from user to kernel mode.
   8. Access I/O device.
7. Some early computers protected the operating system by placing it in a memory partition that could not be modified by either the user job or the operating system itself. Describe two difficulties that you think could arise with such a scheme.
8. Some CPUs provide for more than two modes of operation. What are two possible uses of these multiple modes?
9. Timers could be used to compute the current time. Provide a short description of how this could be accomplished.
10. 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 for which it is caching (for instance, a cache as large as a disk), why not make it that large and eliminate the device?
11. Distinguish between the client–server and peer-to-peer models of distributed systems.

**Chapter 2 Questions**

1. What is the purpose of system calls?
2. What is the purpose of the command interpreter? Why is it usually separate from the kernel?
3. What system calls have to be executed by a command interpreter or shell in order to start a new process on a UNIX system?
4. What is the purpose of system programs?
5. What is the main advantage of the layered approach to system design? What are the disadvantages of the layered approach?
6. List five services provided by an operating system, and explain how each creates convenience for users. In which cases would it be impossible for user-level programs to provide these services? Explain your answer.
7. Why do some systems store the operating system in firmware, while others store it on disk?
8. How could a system be designed to allow a choice of operating systems from which to boot? What would the bootstrap program need to do?