1. [25 points] Which of the following instructions should be privileged for an OS to function properly? Explain.
   a. Set value of timer.
   b. Read the clock.
   c. Issue a trap instruction.
   d. Turn off interrupts.
   e. Access I/O device.

   Ans:

   In my opinion, these following instructions should be privileged

   1. Set value of timer.
   2. Turn off interrupts.
   3. Access I/O device.

   This is because to ensure proper operation, we must protect the operating system and all other programs and their data from any malfunctioning program. Protection is needed for any shared resource. The approach taken is to provide hardware support to allow us to differentiate among various modes of execution. We accomplish this protection by designating some of the machine instructions that may cause harm as privileged instructions.

   Firstly, setting value of time can count the number of clock to prevent user program from hogging the CPU and thus is privilege.
   Secondly, Turing off interrupt is privileged so that a process cannot monopolize the CPU.

   Last but not least, protection of access I/O device ensure that a user program could never gain control of the computer in monitor mode and so it is privileged.

   In conclusion, by Definition, It usually include storage protection setting, interrupt handling, timer control, input/output, and special processor status-setting instructions, that can be executed only when the computer is in a special privileged mode that is generally available to an operating or executive system, but not to user programs.

2. [25 points] (Problem 1.8) 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?

   Ans:

Well, an interrupt is change-of-flow generated by hardware in the system. Besides, the interrupt handler is summoned to cope with the result of the interrupt. Then, the control is returned to the interrupted context and instruction. On the contrary, the trap is software-generated interrupt. The interrupt can be used to signal the completion of I/O to remove the demand for the device polling. A trap can be used to call OS routines or to catch arithmetic error.

3. [25 points] (Problem 1.12) Consider an SMP system similar to the one shown in Fig. 1.6. Illustrate with an example how data residing in memory could in fact have a different value in each of the local caches. [Hint: all you need to do is to explain an example of how this can happen]

   Ans:

   Consider processor 1 reads data A with value 5 from main memory into its local cache. Meanwhile, processor 2 also reads data A into its local cache. Next, Processor 1 then updates A to 10. However, since A resides in processor 1's local cache, the update only occurs there **and not** in the local cache for processor 2 Consequently, data residing in memory could have a distinct value in each local caches.