

# Home work - Set1

August 13, 2015

**There are two sections in this problem set.**

Section 1 is for IOPC 332C while Section 2 for IOSY 332C

## 1 Section1 : IOPS 332C

1. In a multiprogramming and time-sharing environment, several users share the system simultaneously. This situation can result in various security problems.
  - (a) What are two such problems?
  - (b) Can we ensure the same degree of security in a time-shared machine as in a dedicated machine? Explain your answer.

**Answer:**

- (a) Stealing or copying one's programs or data; using system resources (CPU, memory, disk space, peripherals) without proper accounting.
  - (b) Probably not, since any protection scheme devised by humans can inevitably be broken by a human, and the more complex the scheme, the more difficult it is to feel confident of its correct implementation.
2. What is the main difficulty that a programmer overcomes in writing an operating system for real time environment?
3. Which of the following instructions should be privileged?
  - (a) Set value of timer
  - (b) Read the clock
  - (c) Clear memory
  - (d) Issue a trap instruction
  - (e) Access I/O device

**Answer:**

- Access I/O device - privileged

4. What is the purpose of interrupts? What are the differences between a trap and an interrupt? Can traps be generated intentionally by a user program? If so, for what purpose?

**Answer:** An interrupt is a hardware-generated change-of-flow within the system. An interrupt handler is summoned to deal with the cause of the interrupt; control is then returned to the interrupted context and instruction. A trap is a software-generated interrupt. An interrupt can be used to signal the completion of an I/O to obviate the need for device polling. A trap can be used to call operating system routines or to catch arithmetic errors.

## 2 Section2 : IOSY 332C

1. What are the main functions of Operating System?
2. What is multi-programming?
3. Which of the following instructions should be allowed only in kernel mode?
  - (a) Disable all interrupts
  - (b) Read the time of day-clock
  - (c) Set the time of day-clock
  - (d) Change the memory map
  - (e) Switch user from user to kernel mode
4. Direct memory access is used for high-speed I/O devices in order to avoid increasing the CPU execution load.
  - (a) How does the CPU interface with the device to coordinate the transfer?
  - (b) How does the CPU know when the memory operations are complete?
  - (c) The CPU is allowed to execute other programs while the DMA controller is transferring data. Does this process interfere with the execution of the user programs? If so, describe what forms of interference are caused.

**Answer:** The CPU can initiate a DMA operation by writing values into special registers that can be independently accessed by the device. The device initiates the corresponding operation once it receives a command from the CPU. When the device is finished with its operation, it interrupts the CPU to indicate the completion of the operation. Both the device and the CPU can be accessing memory simultaneously. The memory controller provides access to the memory bus in a fair manner to these two entities. A CPU might therefore be unable to issue memory operations at peak speeds since it has to compete with the device in order to obtain access to the memory bus.