Unit 6. Intput/output systems
Exercises

Exercise 1. A disk drive has an average access time of 4 ms, 15,000 rpm and 500 sectors per track. Let be a file of 2500 sectors with a size of 1.22 MB. Compute the time needed to read the file in the following scenarios:

a) The file is stored in sequential way, and occupies 5 consecutive tracks.
b) The sectors of the file are randomly stored in the disk

Exercise 2. A disk with 600 sectors per track, and 7,200 rpm, has an average seek time of 2 ms. Calculate the average access time to a sector.

Exercise 3. We want to develop a controller for a traffic light. The controller has a 32 bit processor, isolated I/O address space. The processor provides the MIPS32 assembly language. This processor is connected to two I/O modules. The first one is a timer and the second one is an I/O module that controls the traffic light behavior. The time provides three registers:

- Register with address 1000. In this register the value with the time to decrease is loaded.
- Register with address 1004. When the value stored in this register is 1, the countdown of the timer starts.
- Register with address 1008. When the timer finishes the countdown, the value stored in this register is 1. During the countdown, the value stored is 0.

The I/O module that controls the traffic light has one register with address 1012. This register is used to encode the color of the traffic light: 100 for red, 010 for yellow, and 001 for green.

Write an assembly program to control the behavior of this traffic light. The traffic light always starts in red. 90 seconds are used for red, and green, and 20 seconds are used for yellow.

Exercise 4. A computer has connected a mouse device that can be checked 30 times per second, in order to update the position in the terminal. The routine used to check the position requires 2,000 cycles for the execution. The computer has a clock frequency of 2.7 GHz. What is the percentage of time that the computer is used to execute this routine?

Exercise 5. A computer is connected to a sensor that measures the temperature of an oven. The computer is also connected to an alarm. The temperature sensor is connected using an I/O module with three registers:

- Control register (address: `ST_REG_CONTROL`). This register is used to control the operation to do. There are two possible operations:
  - Device initialization. The device is initialized with a “0” value.
  - Read a temperature. When a “1” is loaded in the register.
- Data register (address: `ST_REG_DATA`). This register is used to store the temperature read from the oven.
- Status register (address: `ST_REG_STATUS`). This register can store two values:
  - Ready (value 1): when the device has been initialized and a valid temperature is stored in the data register.
  - Busy (value 0): When the device is being initialized or reading a temperature from the oven.

The alarm is connected to an I/O module with two registers:

- Control register (address: `A_REG_CONTROL`). This register is used to activate or deactivate the alarm. A “1” value allows to activate the alarm, and a “0” value allows to deactivate the alarm.
- Status register (address: `A_REG_STATUS`). This register allows knowing the status of the alarm. If the value stored in this register is “0”, the alarm is off. If the value is “1” the alarm is on.

The computer provides an isolated I/O address space with two I/O instructions:

- `in RegProcesador, RegE/S` Load in processor register `RegProcesador` the value stored in the I/O register `RegE/S`.
- `out RegProcesador, RegE/S` Load in the I/O register the value stored in the processor register `RegProcesador`.

All I/O registers have 32 bits. Write an assembly program to read the temperature of the oven. If the temperature read is over 100º C, the alarm must be activated. When the temperature is reduced below 100º the alarm must be deactivated.

Exercise 6. A 32 bit computer with byte addressing and isolated memory space and I/O space, uses the MIPS32 instructions set and two I/O machine instructions:

```
in rdest, address
out rsrc, address
```

This computer is connected to a sensor that measures the water level with the following features:

- An I/O module with three registers:

Name	**Adress**	**Number of bits**
Control reg.	`R_CONTROL_Px`	32 bits
Status reg.	`R_ESTADO_Px`	32 bits
Data reg.	`R_DATOS_Px`	32 bits

- There are two operations: `ON` to activate the sensor, and `OFF` to deactivate the sensor.
- The status register can allow three values: `MEASURING`, `NEW`, and `ERROR`.
- The behavior is the following: when the sensor is activated, the initial status is `MEASURING`. When a new water level is detected, the new value will be loaded in the Data register. When a new data is ready, the value stored in status register is `NEW`. When the value is read from the data register, the value stored in the status register is `MEASURING`. In case of failure, the value `ERROR` is loaded in the status register.

Reply:

a) Write an assembly program to control this peripheral device using programmed I/O. the program behavior is:
   1. The program must activate the sensor.
   2. The program has to read 100 different measures. When a new level is read from the data register, this value has to be written in a memory array, which starts in `M_STORAGE` address.
   3. When an error is detected, the program stops and writes the value `PROBLEM` in the memory address `M_CODE`.

b) Describe the main problems of using programmed I/O for this sensor. Describe the advantages of using interrupts.