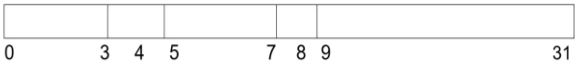
Tutorial Series No. 5

Exercise 1:

Q1) What are the common characteristics of traps, interrupts, supervisor calls, and procedural calls?
Q2) What differentiates traps, interrupts, and supervisor calls from procedural calls?
Q3) What differentiates traps and supervisor calls from interrupts?

Exercise 2:

We are studying interrupts and traps on a machine X.
The machine has only one interrupt level (triggered by the clock signal falling to 0), one single trap, and one single supervisor call.
The clock itself is a counter whose value is decremented by 1 every 5 microseconds.
A supervisor call includes several parameters, with the first indicating the reason for the call.
The processor's status word format includes the following 32 bits:
4 : Mode (Kernel mode = 1 / User mode = 0)
8 : Interrupt Mask (Masked = 1 / Unmasked = 0)
9-31: Program Counter (Instruction Pointer)



The instruction LPSW(m) allows loading the processor's status word from the value stored at the address contained in m.

a) Periodic Measurement Collection

The computer is tasked with periodically collecting measurements from an industrial installation.
The measurements must be taken every 100 milliseconds.
Since the time needed to take the measurements is much shorter than this interval, the computer spends the rest of the time performing another task P, which is periodically interrupted.
Q1) Provide the different programs needed to implement this system.
(Assume that the status word of program P is always saved at address m1 in case of an interruption.)

b) Implementation of a Job Sequencer Monitor

A job sequencer monitor reads and executes submitted tasks in sequence.
Each task has: **a** maximum execution time of 100 milliseconds, **a** start execution address
If the maximum execution time is exceeded, the monitor stops the currently executing task and moves on to execute the next task. The tasks are loaded from a fixed address, so there is no need to specify the address as a parameter. Each task must end with a supervisor call (SVC(Fin)), signaling the task's completion. This call is automatically inserted by the compiler during the compilation phase into the user program's text.
The sequencer monitor is triggered either: By the normal end of the task (SVC(Fin)), or By the expiration of the maximum execution time.
Q2) Provide the different programs necessary to implement this monitor