Practica 3 - Explicaciones

1. Como funciona el MMU

   El MMU trabaja junto a una memoria y se encarga de facilitarnos el acceso a las instrucciones en la memoria en base a las direcciones lógicas. Por otra parte para correr un proceso necesitamos de dos datos en concreto la `_baseDir` (la dirección desde la cual se encuentan las instrucciones del proceso a ejecutar) y la dirección lógica o `logicalAdress` (el pc del proceso).

2. Como funcionan las clases `IoDeviceController` y `PrinterIODevice`

   `IoDeviceController` se encarga de controlar el trafico de los procesos que quieren utilizar un dispositivo dado. Es un controlador por cada dispositivo.

   `PrinterIODevice` representa a un dispositivo de IO, que tiene un estado de si esta ocupado o no.

3. Cómo se llegan a ejecutar `IoInInterruptionHandler.execute()` y `IoOutInterruptionHandler.execute()`

   Cuando el CPU detecta que la próxima instrucción es de tipo INSTRUCTION_IO, le envía al InterruptHandler la interrupción de tipo IO_IN para que esta misma lo handlee, entendiendo que es el IoInInterruptionHandler quien debe encargarse de ejecutar ese IRQ con el comando *execute*.

   Una vez que el dispositivo I/O finaliza de realizar sus instrucciones, le pide al interruptVector que handlee una nueva IRQ, esta vez es una interrupción de tipo IO_OUT, siendo handleada por la IoOutInterruptHandler.

4. 1: Que esta haciendo el CPU mientras se ejecuta una operación de I/O??

   Cuando se ejecuta una operación de I/O el CPU ejecuta el handler correspondiente en el `interruptVector` . El cual en este S.O deja el pc del CPU en -1 (no ejecuta ninguna operación) y le pide al `ioDeviceController` que ejecute la operación.

2: Si la ejecución de una operación de I/O (en un device) tarda 3 "ticks", cuantos ticks necesitamos para ejecutar el siguiente batch?? Cómo podemos mejorarlo?? (tener en cuenta que en el emulador consumimos 1 tick para mandar a ejecutar la operación a I/O)

```
prg1 = Program("prg1.exe",
    [ASM.CPU(2), ASM.IO(), ASM.CPU(3), ASM.IO(), ASM.CPU
(2)])
prg2 = Program("prg2.exe",
    [ASM.CPU(4), ASM.IO(), ASM.CPU(1)])
prg3 = Program("prg3.exe",
    [ASM.CPU(3)])
```

Ticks por programa

En *prg1* tenemos 9 instrucciones de las cuales 2 son IO (2 * 3) y sumamos la instrucción `EXIT`.

Total *prg1*: 9 + 6 + 1 = 16 ticks para ejecutar el programa completo.

En *prg2* tenemos 6 instrucciones de las cuales 1 es IO (1 * 3) y sumamos la instrucción `EXIT`.

Total *prg2*: 6 + 3 + 1 = 10 ticks para ejecutar el programa completo.

En *prg3* tenemos 3 instrucciones de las cuales ninguna es IO (0 * 3) y sumamos la instrucción `EXIT`.

Total *prg3*: 3 + 0 + 1 = 4 ticks para ejecutar el programa completo.

Total de ticks para ejecutar el batch: 16 + 10 + 4 = 30 ticks

Una posible forma de mejorarlo seria que mientras el CPU se encuentra IDLE por las instrucciones IO se ejecute las instrucciones del siguiente batch.