



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

IIC2343 – Arquitectura de Computadores

Ayudantía 6

Profesores: Hans-Albert Löbel Díaz, Jorgen Dieter Heysen Palacios

Ayudante: Germán Leandro Contreras Sagredo (glcontreras@uc.cl)

Temas a tratar

Los temas a tratar dentro de esta guía son:

- Comunicación de CPU y Memoria con I/O.

Preguntas

1. a. **(I2 - II/2015)** Explique cómo funciona la transferencia de direcciones y datos desde/hacia los dispositivos *mapeados* a memoria.
b. Describa las instrucciones de la ISA de un computador **x86** que permiten acceder a dispositivos mediante *port* I/O.
c. **(I2 - I/2017)** ¿Con qué tipo de dispositivo de I/O es preferible utilizar *mapeo* de memoria por sobre puertos?
d. ¿Por qué es mejor hacer uso de interrupciones en vez de *polling*? Mencione un ejemplo.
e. **(I2 - I/2016)** ¿Cuál es la función del vector de interrupciones? ¿Cuál es su contenido?
f. **(I2 - II/2016)** Luego de recibir la señal **INTA**, ¿qué tarea(s) debe realizar un controlador de interrupciones?
g. Detalle, paso a paso, cómo se manejaría una interrupción realizada por un dispositivo (llamémoslo IO_i) que se encuentra conectado a otro dispositivo (llamémoslo IO_j), donde la ISR de este último se encuentra almacenada en el vector de interrupciones del computador.
h. Explique la diferencia entre las interrupciones realizadas por *hardware* y *software*, dando un ejemplo de cada una.

- i. **(I2 - II/2016)** Para los siguientes ejercicios, considere la siguiente tabla, que presenta el vector de interrupciones completo de un computador con ISA x86 de 16 bits. El vector de interrupciones se encuentra almacenado a partir de la dirección de memoria 0x0000:

IRQ	Dispositivo	Pos. en vector
IRQ0	<i>Timer</i> del sistema	00
IRQ1	Disco Duro	01
IRQ2	Interfaz USB	02
IRQ3	Interrupción <i>software</i>	03

Cuadro 1: Vector de interrupciones del computador.

- I. ¿Cuántos dispositivos que generen solicitudes de interrupción pueden conectarse?
 - II. Dos dispositivos, teclado y *mouse*, están conectados a la interfaz USB. Describa un mecanismo para ejecutar la ISR correspondiente al *mouse*, cuando este genera una interrupción.
 - III. ¿Que ocurriría en este computador si se ejecuta la instrucción `MOV [0],AX`?
 - IV. Proponga un esquema para permitir el acceso (lectura y escritura) controlado y centralizado al vector de interrupciones por parte de los programas, *i.e.*, el acceso solo puede realizarse a través de una interfaz entregada por el sistema operativo (o la BIOS).
Hint: El esquema puede incluir cambios a la arquitectura del computador.
2. a. **(I3 - I/2013)** Un robot simple, conectado a un computador, es accesible mediante *mapeo* a memoria. Este robot se mueve en un espacio cuadrado infinito, en el cual cada grilla puede estar vacía o contener una muralla. El robot tiene comandos para ser prendido, apagado, avanzar 1 espacio hacia adelante, girar a la izquierda en 90° y examinar lo que hay adelante. Cada vez que el robot se encuentra desocupado, *i.e.* ha sido recién iniciado o ha terminado una acción, genera una interrupción para informar que es posible darle un nuevo comando.
- I. Describa el mapa de memoria necesario para manejar el robot.
 - II. Defina el formato de los datos que recibirá el robot como comandos y que entregará este para informar su estado.
 - III. Escriba en *Assembly x86* la ISR asociada al control del robot, siguiendo el siguiente comportamiento: el robot avanza hasta encontrar una muralla, en cuyo caso girará a la izquierda hasta encontrar un espacio vacío para avanzar, teniendo la precaución de que el robot no retroceda. Asuma que el espacio ha sido diseñado para que el robot no se quede pegado girando eternamente.
- b. **(I3 - II/2012)** Suponga que se tiene un dispositivo de adquisición de imágenes térmicas conectado a un computador que tiene una microarquitectura especializada para la adquisición de imágenes, pero con ISA compatible con x86 de 16 bits. El computador tiene una memoria principal de 64 kilobytes, con el siguiente mapa de memoria para los primeros 4096 bytes:
- Se desea escribir un programa que permita adquirir imágenes mediante la cámara y luego almacenarlas en disco. Las imágenes generadas por la cámara se encuentran en escala de grises de 8 bits, ordenadas por filas en una matriz cuadrada de 30x30.

Dirección	Función asociada
0-5	<i>Exception handlers.</i>
6	Registro de comandos de la cámara.
7	Registro de estado de la cámara.
8-14	Vectores de interrupciones de <i>hardware</i> .
15	Vector de interrupción de escritura en disco.
16	Vector de interrupción de adquisición de imagen.
17-31	Vectores de interrupciones de <i>software</i> de uso libre.
32-123	Memoria de uso libre.
124-1023	<i>Buffer</i> de adquisición de la cámara.
1024-4096	Espacio de memoria del disco.

Cuadro 2: Tabla que muestra el mapa de memoria del dispositivo.

- I. Escriba una **ISR** para alguna interrupción de *software* disponible, que permita adquirir una imagen y luego escribirla en disco.

La **ISR** de la cámara no recibe parámetros y retorna en su registro de estado información sobre la adquisición. Si la adquisición fue exitosa, el registro contendrá **0xFF** y la imagen se encontrará en el **buffer** de la cámara. En caso contrario, si la adquisición falló, el registro contendrá **0x00**. Durante la adquisición, el registro contendrá el valor **0xF0**.

La **ISR** del disco utiliza internamente el controlador de **DMA**, por lo que necesita los siguientes parámetros en los siguientes registros:

- La dirección de inicio del origen en el registro **AX**.
- La dirección de inicio del destino en disco en el registro **BX**.
- La cantidad de palabras a copiar en el registro **CX**.

Puede utilizar la cantidad de parámetros que estime conveniente para su **ISR**, pero debe dejar explícitamente escrito qué significan y dónde se almacenan.

- II. Escriba un programa que llame a la subrutina del ítem anterior para adquirir tres imágenes y almacenarlas de manera consecutiva en disco. Considere que la adquisición puede fallar y que se intentará esta un máximo de tres veces por imagen.