



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

IIC2343 - Arquitectura de Computadores (II/2019)

Tarea 3 x86 + I/O

Fecha límite de entrega: Miércoles 30 de Octubre a las 21:00

Motivación

x86 es la arquitectura que utilizamos en nuestros computadores todos los días. Una diferencia con otras arquitecturas es el tipo de ISA, CISC (como por ejemplo ARM, presente en nuestros teléfonos celulares, cuyo set de instrucciones es RISC). ¿Podemos programar directamente en x86 y conseguir un programa que se ejecute? ¡Claro que sí!¹

Entre las aplicaciones de esta habilidad, hay casos muy puntuales en los que, para conseguir un cierto rendimiento o funcionalidad, no hay más opción que programar en assembly. Además entender cómo funciona la arquitectura cambiará la forma en que entienden los programas, sin que necesariamente se oriente a un alto rendimiento.

Sin embargo, **lo más importante es que eres una persona maravillosa e infinitamente capaz. Confiamos plenamente en ti. ¡Mucho ánimo con todo!**

Objetivos

1. Practicar la programación en el assembly x86.
2. Practicar el uso de dispositivos I/O reales en x86.
3. Hacer un programa con interacción de usuario en x86.
4. Conseguir un mayor entendimiento sobre cómo funcionan los computadores modernos, a un bajo nivel.

¹Es gracias a AMD que continuamos con la retro-compatibilidad con el x86 de 16 bits, teniendo computadores de 64. Intel lo intentó, pero fracasó y se vio forzado a adoptar lo desarrollado por AMD.

Instrucciones generales

- Para todas las preguntas de la tarea debe ocupar el emulador de x86 **emu8086** cuyo instalador puede encontrar entre los archivos de la Tarea². Cuando instale el emulador, hágalo en el path por defecto (C:\emu8086) para evitar complicaciones.
- Luego de instalar el emulador, usted debe modificar la línea 36 del archivo C:\emu8086\emu8086.ini
 - Línea original: "HW_INTERRUPT_FILE=c:\emu8086.hw"
 - Línea modificada: "HW_INTERRUPT_FILE=c:\Arqui\emu8086.hw"
- Para desarrollar la tarea debe ocupar el assembly x86, cuyo set de instrucciones se incluye entre los archivos entregados.
- Además deberá utilizar el programa Interruptor.exe que permite gatillar interrupciones de hardware en el emulador. Este archivo ejecutable está incluido en los archivos entregados para la tarea y debe ser copiado dentro de la carpeta DEVICES ubicada en la carpeta de instalación del emulador emu8086.

Pueden encontrar la documentación del uso del emulador en
C:/emu8086/documentation/_documentation_index.html

Junto al enunciado se les entregará el archivo base con el cual trabajarán para realizar las tres partes de esta tarea. Tener en cuenta que al momento de realizar la tercera parte, deben descomentar la cuarta línea del archivo base (*start-robot.exe*). Este cambio no afectará el desarrollo de las otras partes.

Programa (1.5 ptos)

Su tarea es conseguir que el programa **espere** a que el usuario presione uno de los tres botones de interrupción. Una vez atendida la interrupción por la ISR correspondiente, su programa debe llamar la subrutina apropiada de acuerdo a la última interrupción atendida. Estas subrutinas deben seguir la convención de llamada *stdcall*. El programa debe almacenar el retorno de las subrutinas y ser capaz de usarlo como parámetro de ser necesario. Para esto deberá definir variables, escribir subrutinas, completar correctamente las ISRs y escribir el código del programa.

La documentación necesaria para las dos primeras partes puede encontrarla en
C:/emu8086/documentation/8086_bios_and_dos_interrupts.html

A continuación el código base para la tarea:

²Solo tiene versión para Windows

```

1  name "Tarea3"
2
3  #start=Interrupcion.exe#
4  ;#start=robot.exe#
5
6  org 100h
7
8
9
10 start:
11     MOV AX, 0
12     MOV ES, AX
13     MOV AL, 90H ; EN LA DIRECCION 0X90 DEL VECTOR DE INTERRUPCIONES
14     MOV BL, 4H
15     MUL BL
16     MOV BX, AX
17     MOV SI, OFFSET [boton_1] ;REFERENCIAMOS LA ISR BOTON_1
18     MOV ES:[BX], SI
19     ADD BX, 2
20     MOV AX, CS
21     MOV ES:[BX], AX
22
23     MOV AX, 0
24     MOV ES, AX
25     MOV AL, 91H ; EN LA DIRECCION 0X91 DEL VECTOR DE INTERRUPCIONES
26     MOV BL, 4H
27     MUL BL
28     MOV BX, AX
29     MOV SI, OFFSET [boton_2] ;REFERENCIAMOS LA ISR BOTON_2
30     MOV ES:[BX], SI
31     ADD BX, 2
32     MOV AX, CS
33     MOV ES:[BX], AX
34
35     MOV AX, 0
36     MOV ES, AX
37     MOV AL, 92H ; EN LA DIRECCION 0X92 DEL VECTOR DE INTERRUPCIONES
38     MOV BL, 4H
39     MUL BL
40     MOV BX, AX
41     MOV SI, OFFSET [boton_3] ;REFERENCIAMOS LA ISR BOTON_3
42     MOV ES:[BX], SI
43     ADD BX, 2
44     MOV AX, CS
45     MOV ES:[BX], AX
46
47     ;Completar Programa
48
49     RET
50
51
52     ;Agregar Subrutinas
53
54
55     boton_1:
56         ;CompletarCodigoISR
57         IRET
58
59     boton_2:
60         ;CompletarCodigoISR
61         IRET
62
63     boton_3:
64         ;CompletarCodigoISR
65         IRET

```

Introducción

Señoras y señores de todo el reino, mientras escuchan esta maravillosa [música para leer el enunciado](#), me gustaría darles la bienvenida al torneo semestral los *ArquiAwards 2019-2*. Oh, valerosos concursantes, se verán enfrentados a una serie de desafíos que pondrán a prueba su valía como arquitectos de computadores y al final, ¡el ganador recibirá un jugoso premio!^a

^aEs mentira. O quizás no, ¿quién sabe?

Partes:

1. Subrutina A (1.0 pto)

Esta debe ser llamada después de la interrupción levantada al presionar el botón 1.

Deberá obtener la cantidad de ciclos de clock a partir de la media noche, desde el tiempo del sistema llamando a la interrupción ofrecida por la BIOS, y simulada por el emu8086. Luego, deberá retornar la cantidad de horas desde la medianoche (deberás calcularlo).

2. Subrutina B (1.5 ptos)

Esta debe ser llamada después de la interrupción levantada al presionar el botón 2.

Deberá recibir como parámetro lo retornado por la subrutina A y mostrarlo en decimal. Esto por medio del uso de la interrupción por software asociada a la interfaz gráfica, utilizando esta última en modo *Teletype*.

La documentación necesaria para las dos primeras partes puede encontrarla en C:/emu8086/documentation/8086_bios_and_dos_interrupts.html

3. Subrutina C (2.0 ptos)

Esta debe ser llamada después de la interrupción levantada al presionar el botón 3.

Un robot necesita tu ayuda, su ultimo dueño lo dejó abandonado y necesita de ustedes para poder "vivir". Por decisión unánime de un voto, se llamará Rigoberto.

Al descomentar el primer comentario en su programa, e iniciar la emulación, se les mostrará una pantalla con el robot dentro de un mapa, junto con distintos elementos como murallas y ampolletas.

La subrutina deberá hacer que el robot interactúe con su entorno de acuerdo a las siguientes consideraciones:

- Rigoberto comenzará su ejecución siendo diestro. Ser diestro significa que al girar, lo hace todo el tiempo a la derecha y ser zurdo, a la izquierda.

- Rigoberto avanzará indefinidamente hacia adelante hasta que encuentren un obstáculo, y dependiendo de este reaccionará de distintas formas:
 - Si es una pared, Rigoberto decidirá girar, según su mano dominante, hasta dejar de encontrarse con una pared.
 - Al descubrir una bombilla de luz encendida, se electrocutará. Esto apagará la bombilla y Rigoberto cambiará su mano dominante, y dará un giro.
 - Al encontrarse con una bombilla de luz apagada, el pequeño robot se molestará y dará media vuelta, sin cambiar su mano dominante.
- El robot deberá cumplir estos requisitos indefinidamente en el tiempo.

Pueden encontrar la documentación de su uso en <C:/emu8086/documentation/io.html>

Entrega

Debes entregar el archivo **Tarea3.asm** por medio del cuestionario correspondiente en **SIDING**. El periodo de entrega finaliza el Miércoles 30 de Octubre a las 21:00 hrs.

Descuentos

Entrega atrasada. La nota máxima se reducirá en 5 décimas por cada media hora luego del término de la entrega, **con un máximo de 3 horas**. Se aplicará de manera proporcional, esto es que si, por ejemplo, el atraso fue de 25 minutos, el descuento será de $\frac{25}{30} \cdot 5$ décimas.

Contacto

Cualquier pregunta sobre la tarea, ya sean de enunciado, contenido o sobre aspectos administrativos deben comunicarse con los ayudantes creando issues en el Syllabus del Github del curso.

Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería deben mantener un comportamiento acorde al Código de Honor de la Universidad:

“Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad.”

En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un procedimiento sumario. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica. Todo trabajo presentado por un alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en material de terceros. Por “trabajo” se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros. Si un alumno (grupo) copia un trabajo, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir reprobación del curso y un procedimiento sumario. Por “copia” se entiende incluir en el trabajo presentado como propio partes hechas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.