



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

IIC2343 – Arquitectura de Computadores

Ayudantía 5

Profesor: Yadran Francisco Eterovic Solano

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

Temas a tratar

Los temas a tratar dentro de esta guía son:

- Arquitectura x86
- Comunicación de CPU y Memoria con I/O.

Preguntas

1. a. Indique la ISA del computador básico visto en el curso y la correspondiente a la arquitectura x86, explicando sus diferencias.
- b. **(I2 - I/2017)** ¿Cuántas palabras de la memoria son modificadas al ejecutar la instrucción `ADD [BH], AX`?
- c. **(I2 - I/2016)** ¿Por qué es necesaria la existencia de la instrucción `RET Lit` en un computador x86?
- d. **(I2 - I/2013)** Al iniciar el cuerpo de una subrutina, ¿por qué es necesario ejecutar las instrucciones `PUSH BP` y `MOV BP, SP`? ¿Qué pasa si no se ejecutan?
- e. **(I2 - I/2017)** ¿Cuántas llamadas recursivas a una función es posible hacer como máximo en un computador x86 de 16 bits? Indique claramente sus supuestos.
- f. **(I2 - II/2016)** Describa un mecanismo para, en tiempo de ejecución, escribir el código de una subrutina y luego llamarla, utilizando el Assembly x86 de 16 bits. Asuma que tiene disponible la especificación completa de la ISA.

- g. **(I2 - I/2017)** Para el siguiente programa escrito en Assembly x86-16, indique los valores de los registros SP y BP y del *stack* completo, al momento de ingresar al label **set**.

```
MOV BL,3
PUSH BX
CALL func
RET
func: PUSH BP
      MOV BP,SP
      MOV BL,[BP + 4]
      CMP BL,0
      JE set
      MOV CL,BL
      DEC CL
      PUSH CX
      CALL func
      MOV BL,[BP + 4]
      MUL BL
      JMP end

set:  MOV AX,1

end:  POP BP
      RET 2
```

2. **(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.
- Describa el mapa de memoria necesario para manejar el robot.
 - Defina el formato de los datos que recibirá el robot como comandos y que entregará este para informar su estado.
 - 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.