



Práctica 3

Objetivo

Distinguir las características de la organización y arquitectura del microprocesador de una computadora de propósito general, analizando sus recursos de hardware y software, para conocer capacidades y limitaciones de forma organizada y responsable.

Desarrollo

1. Se cargó en GitHub el código de la práctica anterior y se verificó su correcto funcionamiento.
2. Se descargó el archivo `lib.zip` desde Moodle, se descomprimió y se colocaron los dos archivos contenidos en la carpeta de trabajo de GitHub.
3. En la parte superior del archivo `nombre_Practica.asm` se agregó la instrucción:

```
%include "./pc_io.inc"
```

4. El código se ensambló con el comando:

```
nasm -f elf nombre_Practica.asm
```

5. Se enlazó el objeto generado con el comando:

```
ld -m elf_i386 -s -o nombre_Practica nombre_Practica.o  
libpc_io.a
```

6. Finalmente, el programa se ejecutó con:

```
./nombre_Practica
```

Explicación paso a paso

En el código se deben agregar las siguientes partes:

1. En él .data agregar una constante de texto:

```
msj: db 'Ingrese un dígito (0-9)',0x0A
len: equ $-msj
```

Esto define una cadena de texto con salto de línea (0x0A) y una constante `len` que guarda la longitud del mensaje.

2. En él .bss agregar una variable:

```
num1 resb 1
```

Aquí se reserva un byte para almacenar el valor ingresado.

3. En él .text se debe colocar el siguiente código:

¿Qué piensa usted que hará este código?

```
mov eax, 4
mov ebx, 1
mov ecx, msj
mov edx, len
int 80h
```

¿Qué hizo realmente este código?

¿Qué piensa usted que hará este código?

```
mov eax, 3
mov ebx, 0
mov ecx, num1
mov edx, 1
int 80h
```

¿Qué hizo realmente este código?

¿Qué piensa usted que hará este código?

```
mov eax, 4
mov ebx, 1
mov ecx, num1
mov edx, 1
int 80h
```

¿Qué hizo realmente este código?

4. Del paso anterior se debe explicar textualmente que se piensa que hace cada sección de código, con sus propias palabras.
5. Ahora el código anterior lo vamos a modificar:
 - a. El `msj` ahora debe quedar de la siguiente manera:

```
msj: db 'Ingrese un dígito (0-9)',0x0
```

b. El código del paso 3 se debe comentar y agregar el siguiente:

¿Qué piensa usted que hará este código?

```
mov edx, msj
```

```
call puts
```

¿Qué hizo realmente este código?

¿Qué piensa usted que hará este código?

```
call getch
```

¿Qué hizo realmente este código?

¿Qué piensa usted que hará este código?

```
call putchar
```

¿Qué hizo realmente este código?

6. En la parte inferior del código, vamos a agregar la siguiente función:

```
salto:
```

```
mov al, 13
```

```
call putchar
```

```
mov al, 10
```

```
call putchar
```

```
ret
```

7. Ahora el código se modifica con salto y debe quedar de la siguiente manera:

¿Qué piensa usted que hará todo este código?

```
mov edx, msj
```

```
call puts
```

```
call salto
```

```
call getch
```

```
call salto
```

```
call putchar
```

```
call salto
```

¿Qué hizo realmente todo este código?

8. Vamos y modificamos la función de salto:

```
salto:
```

```
pushad
```

```
mov al, 13
```

```
call putchar
```

```
mov al, 10
```

```
call putchar
```

```
popad
```

```
ret
```

¿Qué hizo realmente todo este código?

9. Ahora agregamos la funcionalidad de capturar una letra, por ejemplo la a y sumarle un 1 e imprimirla en terminal.

Agregamos el siguiente código después del último call salto

¿Qué piensa usted que hará todo este código?

```
mov ebx, num1
mov byte[ebx], al
add byte[ebx], 1
mov al, byte[ebx]
call putchar
```

¿Qué hizo realmente todo este código?

10. Ahora para la prueba capture 3 y súmele 1 como el caso anterior.
11. Ahora capture otro número, modifique lo necesario para ello, y pruebe con una suma de $3 + 4 = 7$:

¿Qué piensa usted que hará todo este código?

```
mov ebx, num1
mov al, [ebx]
mov ebx, num2
add [ebx], al
mov al, byte[ebx]
call putchar
```

¿Qué hizo realmente todo este código?

Obligatoriamente de cada paso debe tener las capturas del código y de la terminal.

Las preguntas van en relación con terminal, solo eso.

Conclusiones

Dificultades