

Arquitectura de computadoras

Entrada Salida con Lenguaje Ensamblador

Ejemplos resueltos

Los siguientes son ejemplos resueltos de programas escritos en lenguaje ensamblador con entrada salida utilizando el simulador

<https://parraman.github.io/asm-simulator/>

Ejemplo 1

Escriba un programa que escriba la frase "Hola Mundo!!!" en el display del simulador.

Ejemplo 2

Escriba un programa que dibuje la bandera de Boca Juniors en la pantalla del simulador.

Ejemplo 3

Escriba un programa que por cada tecla presionada del teclado del simulador, indique que tecla se presionó en el display. Las posiciones del display no deben sobreescribirse. Cada tecla presionada debe ser leída y escrita en el display una sola vez.

Ejemplos resueltos

Ejemplo 1

Cada instrucción mueve el código ASCII de las diferentes letras a cada posición del display. Por ejemplo:

El código ASCII de la letra H (mayúscula) es 0x48. La primera posición del display está mapeada en la dirección de 0x02E0.

El código ASCII de la letra o (minúscula) es 0x6F. La segunda posición del display está mapeada en la dirección de 0x02E1.

Y así sucesivamente.

```
MOVB [0x02E0],0x48 ;H
MOVB [0x02E1],0x6F ;o
MOVB [0x02E2],0x6C ;l
MOVB [0x02E3],0x61 ;a
MOVB [0x02E4],0x20 ;Espacio
MOVB [0x02E5],0x4D ;M
MOVB [0x02E6],0x75 ;u
MOVB [0x02E7],0x6E ;n
MOVB [0x02E8],0x64 ;d
MOVB [0x02E9],0x6F ;o
MOVB [0x02EA],0x21 ;!
MOVB [0x02EB],0x21 ;!
MOVB [0x02EC],0x21 ;!
HLT
```

Ejemplo 2

La pantalla tiene 256 píxeles. Comienza en el pixel 0x0300 y termina en el pixel 0x0400. Está organizada en 16x16 píxeles. La primera fila va de 0x0300 a 0x030F, la segunda fila va de 0x0310 a 0x031F, la tercera fila va de 0x0320 a 0x032F y así sucesivamente.

Para este ejemplo, dividimos la pantalla en 3 rangos:

- Primer rango: Tiene 5 filas. Va desde la dirección 0x0300 a la dirección 0x034FF. Pintamos dichos píxeles en color azul (0x43).
- Segundo rango: Tiene 6 filas y va desde la dirección 0x0350 a la dirección 0x03AFF. Pintamos dichos píxeles en color amarillo (0xF4).
- Tercer rango: Tiene 5 filas y va desde la dirección 0x03B0 a la dirección 0x03FFF. Pintamos dichos píxeles en color azul (0x43).

Utilizamos el registro A para direccionamiento indirecto.

```
MOV A,0x0300 ;Usamos A para direccionamiento indirecto.
azul1: ;Primer lazo. Pinta de azul los píxeles del primer rango.
```

```
MOVB [A],0x43 ;Pintamos de azul.
```

```
INC A ;Incrementamos para apuntar al siguiente pixel.
```

```
CMP A,0x0350 ;Vemos si ya se llegó al primer píxel del siguiente rango.
```

```

JZ amarillo
JMP azul1
amarillo: ;Segundo lazo. Pinta de amarillo los píxeles del segundo rango.
MOV B,[A],0xF4 ;Pintamos de amarillo.
INC A
CMP A,0x03B0 ;Vemos si ya se llegó al primer píxel del siguiente rango.
JZ azul2
JMP amarillo
azul2: ;Tercer lazo. Pinta de azul los píxeles del tercer rango.
MOV B,[A],0x43
INC A
CMP A,0x0400 ;Vemos si ya se llegó la primera dirección fuera de la pantalla.
JZ fin
JMP azul2
fin:
HLT

```

Ejemplo 3

```

MOV B,0x02E0 ;0x02E0 es la primera posición del display.
lazo:
IN 0x0005 ;Copia el contenido de KPDSTATUS en el registro A.
CMP A,0x00 ;Si KPDSTATUS tiene un cero, no se presionó ninguna tecla.
JNZ se_presiono_tecla ;Si el resultado no es cero, si se presionó una tecla.
JMP lazo
se_presiono_tecla:
IN 0x0006 ;Copia el contenido de KPDDATA al registro A.
MOV B,[B],AL ;Copia el contenido de la parte baja de A al display (apuntado por
               ;el registro B)
INC B ;Incrementa B para que apunte a la siguiente posición del display.
CMP B,0x0300 ;Verificamos si el display ya se llenó.
JZ fin
JMP lazo
fin:
HLT

```