

Taller 1

De acuerdo al siguiente programa en ensamblador, identifica, desarrolla y describe su funcionamiento. (Qué secciones conforman al programa, qué tipo de registros se utilizan y por qué. Cambia los valores que se están asignando a los registros `eax`, `ebx`, `ecx`, `int 0x80`. Desarrolla qué pasa cuando cambias estos valores y por qué. Cambia el mensaje 'Hola Mundo' por una cadena de al menos el doble de caracteres. Documenta que pasa al imprimir el resultado. Modifica el programa para que imprima la nueva cadena en su totalidad y explica tus cambios. Programa 1:

```
section .data
    mensaje db 'Hola Mundo', 0
```

```
section .text
    global _start
```

```
_start:
```

```
    mov eax, 4
    mov ebx, 1
    mov ecx, mensaje
    mov edx, 11
    int 0x80
```

```
    mov eax, 1
    xor ebx, ebx
    int 0x80
```

Ejercicio:

Secciones del Programa:

- ☐ `.data`: En esta sección se define la variable `mensaje` que contiene la cadena "Hola Mundo" junto con un byte nulo (0) que indica el final de la cadena.
- ☐ `.text`: Esta sección contiene el código ejecutable del programa.

Registros Utilizados:

- ☐ `eax`: Se utiliza para almacenar el número de la `syscall` que se desea ejecutar.
- ☐ `ebx`: Se utiliza para almacenar el primer argumento de la `syscall`, que generalmente es el descriptor de archivo (en este caso 1 para la salida estándar).
- ☐ `ecx`: Se utiliza para almacenar el segundo argumento de la `syscall`, que en este caso es un puntero a la cadena `mensaje`.
- ☐ `edx`: Se utiliza para almacenar el tercer argumento de la `syscall`, que en este caso es la longitud de la cadena `mensaje`.

Desarrollo del Programa:

1. Se establece la syscall write para escribir en la salida estándar (eax = 4).
2. Se asigna el descriptor de archivo (ebx = 1) que representa la salida estándar.
3. Se carga el puntero a la cadena mensaje en ecx.
4. Se carga la longitud de la cadena mensaje en edx.
5. Se realiza la interrupción (int 0x80) para realizar la llamada al sistema y escribir "Hola Mundo" en la salida estándar.
6. Se establece la syscall exit para terminar el programa (eax = 1).
7. Se establece el código de salida (ebx = 0).
8. Se realiza otra interrupción (int 0x80) para finalizar el programa.

Programa propio cambiado:

```
ASM hola.asm
1  section .data
2  |   mensaje db 'Hola mundo, mi nombre es Pedro Balderrama', 0
3
4  section .text
5  |   global _start
6
7  _start:
8  |   mov eax, 4      ; syscall write
9  |   mov ebx, 1      ; file descriptor: stdout
10 |   mov ecx, mensaje
11 |   int 0x80
12 |   int 0x80
13
14 |   mov eax, 1      ; syscall exit
15 |   xor ebx, ebx    ; código de salida: 0
16 |   int 0x80
17
18
```

- Explicación de los Cambios:

- ☐ Se cambió el mensaje a una cadena más larga y se modificó edx para reflejar la nueva longitud de la cadena.
- ☐ El programa ahora imprime la cadena completa "Hola mundo, mi nombre es Pedro Balderrama".