Universidad Autónoma de Baja California

Facultad de Ciencias químicas e Ingeniería

Ingeniero en computación

Materia: Organización y Arquitectura de Computadoras

Tema: Identificar los modos de direccionamiento adecuados para manejo de memoria en aplicaciones de sistemas basados en microprocesador mediante la distinción de su funcionamiento, de forma lógica y responsable.

Alumno:

-

Maestro: Garcia Rocha Jose Isabel

Grupo:

Fecha de entrega: 23 de Septiembre de 2022

1. Responda los siguientes cuestionamientos sobre los programas en lenguaje ensamblador del 80386 usando NASM .

a) ¿Qué es la sección .data?

La sección de datos se utiliza para

b) ¿Qué es la sección .bss?

La sección bss se usa para

c) ¿Qué es la sección .text?

La sección de text se utiliza para

d) ¿Qué es la directiva global?

Es la que le comunica al

2. Copie el código del Listado 1 en un archivo llamado saludo.asm. Abra una terminal en Linux y ensamble el código con NASM por medio del comando:

nasm -f elf saludo.asm

El cual generará el archivo objeto saludo.o.

Encadene el archivo por medio de uno de los siguientes comandos:

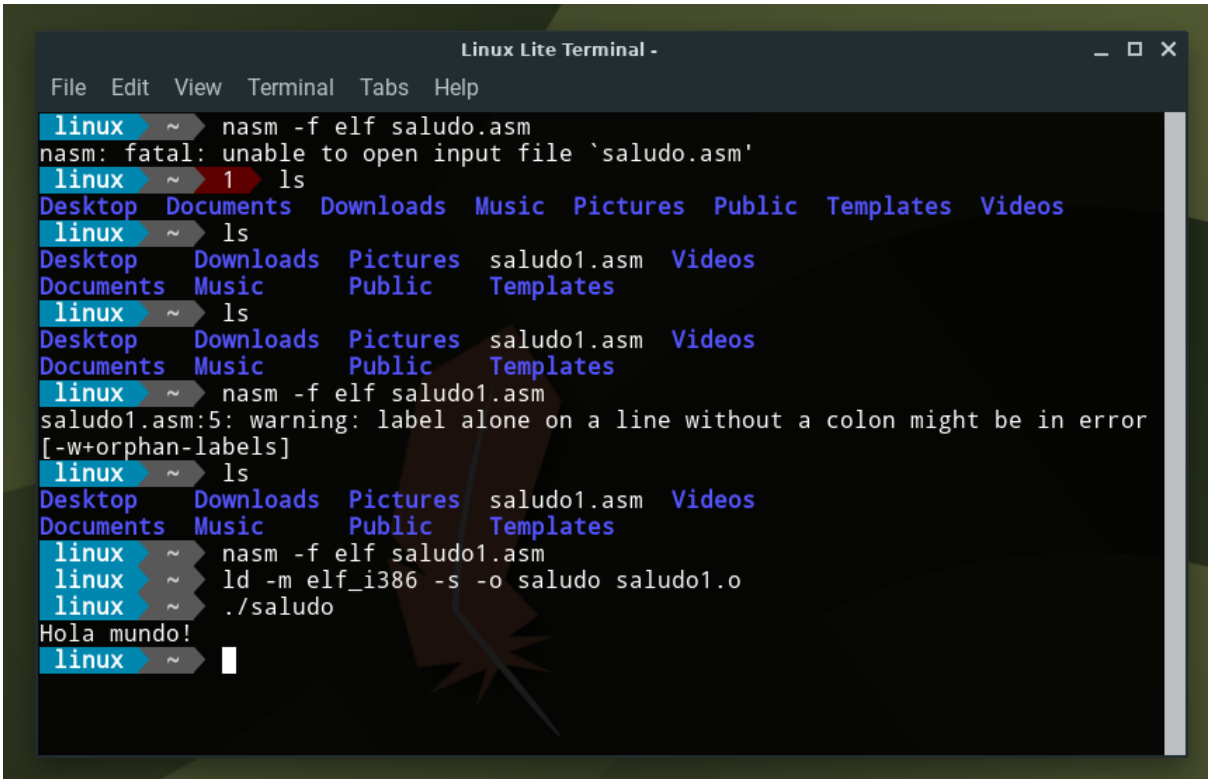
a) En un sistema operativo de 32 bits: ld -s -o saludo saludo.o

b) En un sistema operativo de 64 bits: ld -m elf_i386 -s -o saludo saludo.o

El cual generará el archivo ejecutable saludo. Ejecute el archivo por medio del comando:

./saludo

El programa desplegará en pantalla el mensaje “Hola mundo!”.



3. Copie el código del Listado 2 en un archivo llamado P5.asm.

a) Almacenar en la variable M

```
mov dword
mov eax,
mov esi, cad
call printHex
```

```
A): 01273699
```

b) Almacenar ACF2359A en

```
mov ebx,
mov eax,
mov esi, cad
call printHex
```

```
B): ACF2359A
```

c) Almacenar 0FF395C2 en

```
mov dword
mov eax,
mov esi, cad
call printHex
```

```
C): 0FF395C2
```

d) Copiar BX a

```
mov cx,
mov ax,
mov esi, cad
call printHex
```

```
D): 0FF3359A
```

e) Almacenar el byte

```
mov byte
mov al,
mov esi, cad
call printHex
```

```
E): 0FF33510
```

f) Almacenar 0xF5C en

```
mov word
mov ax,
mov esi, cad
call printHex
```

```
F): 0FF30F5C
```

g) Copiar a ESI la

```
mov dword
mov eax,
mov esi, cad
call printHex
```

```
G): 0804A01C
```

Por cada inciso, despliegue en pantalla el nuevo valor del registro o variable modificada.

Haga uso de la rutina printHex, la cual recibe en EAX el valor que se quiere imprimir.

Listado 1. Primer programa.

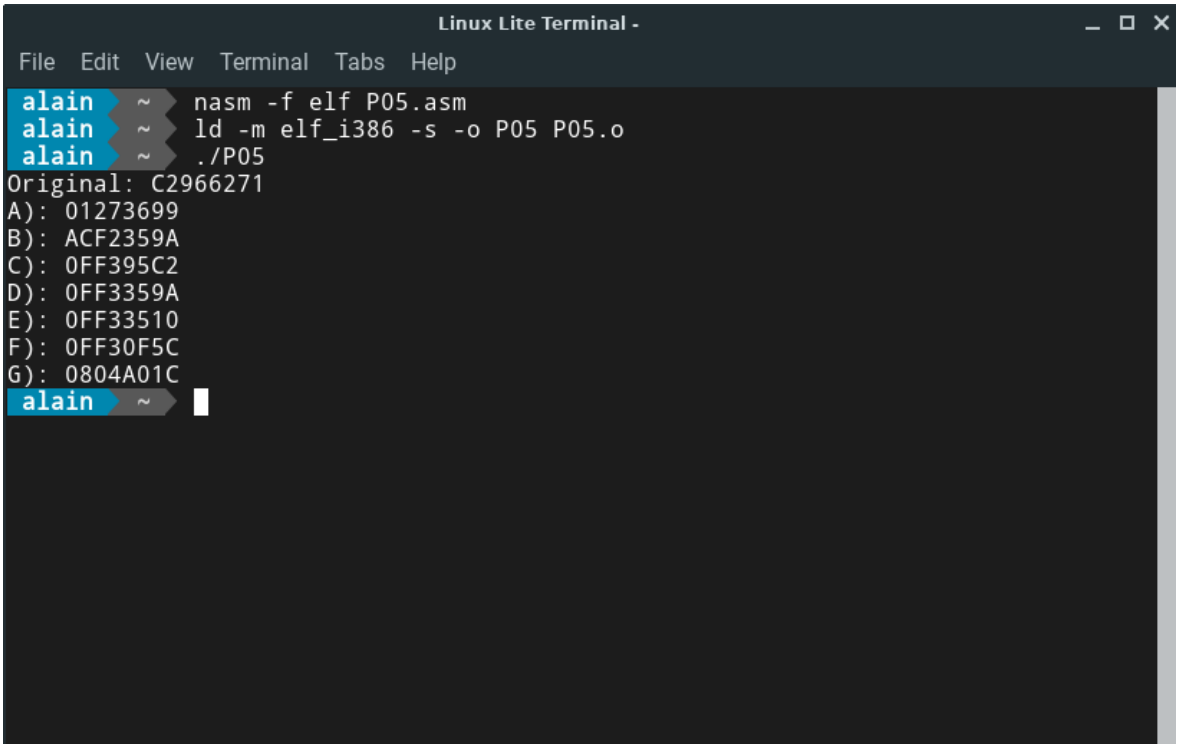
```
section .data
msg: db "Hola mundo!",10
msg_L: equ $-msg

section .text

global _start:
_start:  mov
         mo
         mo
         mo
         int
         mo
         mov e
         int
```

Listado 2. Direccionamiento.

```
section .data    ; Datos inicializados
  NL:    db  13, 10
  NL_L:  equ $-NL

section .bss     ; Datos no inicializados
N       re
M       
cad
section .text
global _start:
_start:    mov e
           mov esi
           call p
           mov e
           mov e
           mov ecx
           mov edx,
           int 80h
           mov e
           mov
           int 8
```

```
 printHex:
           
           mo
mov ebx, 0

           mo
.nxt: shr ea
.msk:                    x
           cl           9
jbe .menc
add al,7
.menor:add al,
mov byt
           in
                        x, edx

           j
           s
cmp cl, 0
ja .nxt
je .ms

.print: mov        4
           m
sub esi, 8
mov ecx, esi
mov edx, 8
           int

           pop
           ret
```

Prueba completa:



Conclusiones y comentarios:

Con respecto a la práctica realizada podemos llegar a la conclusión de que imprimir en pantalla y direccionar con cosas completamente diferentes pero si se tiene un buen entendimiento de estas es posible saber dominar los programas y escribir código correctamente.

Dificultades en el desarrollo:

Ahora no hubo muchas dificultades, lo único que se me paso fue que al almacenar datos en hexadecimal si se empiezan con letra debo poner un 0 porque luego la terminal de NASM no lo detecta y no compila, de ahí en fuera todo bien.

Referencias:

- Diseccionando un binario · Guía de exploits. (s. f.). Fundacion-Sadosky. Recuperado 14 de marzo de 2022, de https://fundacion-sadosky.github.io/guia-escritura-exploits/buffer-overflow/3-got.html
- Anónimo. (S.F). *"Diseccionando un binario*". Guia de exploits. Recuperado de: https://fundacion-sadosky.github.io/guia-escritura-exploits/buffer-overflow/3-got.html
- Anónimo. (2021). *"sección .bss y sección .data*". Programador Click. Recuperado de: https://programmerclick.com/article/41561872707/#:~:text=l%20secci%C3%B3n%20de%20datos%20
- Anónimo. (S.F). *"Capítulo 6. El programa ensamblador"*. Sin nombre. Recuperado de: https://lc.fie.umich.mx/~rochoa/Manuales/ENSAMBLADOR/ARQUITECTURA_5ium/ASM.html