



Universidad Autónoma de Baja California

FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA



Organización y Arquitectura de las Computadoras.

Práctica No. 1: Introducción a ensamblador.

Alumno:

Joshua Osorio O. – 1293271

Docente:

José Isabel García R.

04/02/2026

Tabla de contenido

Objetivo	2
Descripción de la practica.....	2
Lista de materiales	4
Desarrollo.....	4
Investigación previa.....	4
¿Qué son: .data, .bss, .txt y directiva global?.....	4
¿Para qué sirven?.....	4
¿Cómo se usan?.....	5
¿Qué elementos van en cada sección?	5
Comandos más populares en Linux y como se usan.	5
Que son las extensiones de archivo relacionadas con asm.?	6
Evidencia sobre el repositorio y código realizado en github.	6
Conclusiones.....	8
Bibliografía	8

Objetivo

Conocer y dominar el uso de una máquina virtual con sistema operativo Linux analizando sus recursos de hardware y software, para conocer sus capacidades y limitaciones de forma organizada y responsable.

Descripción de la practica

Realizar una pequeña investigación en la arquitectura 386 de ensamblador sobre:

- .data
- .bss
- .txt
- directiva global

De cada una responder lo siguiente:

1. ¿Qué son?

UABC_FCQI_ORGANIZACION_Y_ARQUITECTURA_DE_COMPUTADORAS.

2. ¿Para qué sirven?
3. ¿Cómo se usan?
4. ¿Qué elementos van en cada sección?

Realizar una pequeña investigación sobre:

- Los comandos más populares en Linux y como se usan.
- Sobre que son las extensiones de archivo relacionadas con asm.

Realice los siguientes pasos:

1. Entrar a GitHub

2. Crear un repositorio nuevo con un archivo con nombre “Ejemplo_OAC.txt”.

3. Para activar codespaces se requiere dar clic en <> Code luego en Codespaces

4. Instalar nasm, con la siguiente instrucción:

sudo apt update

sudo apt install nasm -y

5. Revisar la versión de nasm instalado: nasm -v

6. Crear una carpeta cuyo nombre sea “ejemplo”: mkdir ejemplo

7. Cambiar del directorio /workspaces para que la carpeta nueva sea este, con:

cd /workspaces/“sustituir_con_nombre_repositorio”/ejemplo/

8. Comprobar que este cambio se haya realizado con: pwd.

9. Crear un archivo con extensión .asm y con nombre “ejemplo”: touch ejemplo.asm

10. Abrir Notepad para trabajar en ese editor de texto sin ayudas.

11. Ingresar el siguiente código en el archivo:

```
global _start
section .text
_start:
    ; sys_write(stdout, message, length)
    mov    eax,        4
    mov    ebx,        1
    mov    ecx,        message
    mov    edx,        length
    int    80h

    ; sys_exit(return_code)
    mov    eax,        1                    ;sys_exit syscall
    mov    ebx,        0                    ;return 0 (todo correcto)
    int    80h

section .data
message:   db      'Hello,      world!',     0x0A    ;mensaje y nueva linea
length:    equ     $-message           ;Obtenemos la longitud de la cadena
```

12. Cargar el código escrito en **Notepad** sin errores en el archivo ejemplo.asm en GitHub.

13. Ensamblar el archivo creado anteriormente: nasm -f elf ejemplo.asm

14. Enlazar el objeto creado: `ld -m elf_i386 -s -o ejemplo ejemplo.o`
15. Ejecutar el programa obtenido: `./ejemplo`

Lista de materiales

- Computadora
- Conexión a internet
- Git
- Vscode

Desarrollo

Investigación previa.

¿Qué son: .data, .bss, .txt y directiva global?

.text

Es la sección donde se almacena el código ejecutable del programa.

.data

Es la sección que contiene datos inicializados.

.bss

Es la sección que contiene variables no inicializadas.

Directiva global

Una directiva que hace visible un símbolo fuera del archivo.

¿Para qué sirven?

.text

Contiene las instrucciones máquina que la CPU ejecutará.

.data

Guarda variables globales o constantes cuyo valor inicial es conocido.

.bss

Reserva memoria para asignar valores; el sistema las inicializa en cero.

Directiva global

Permite que el enlazador reconozca el punto de entrada del programa.

¿Cómo se usan?

```
.text
    section .text
.data
    section .data
    mensaje: db "Hola mundo", 0
.bss
    section .bss
    buffer: resb 64
Directiva global
    global _start
```

¿Qué elementos van en cada sección?

```
.text
    instrucciones del programa, Puntos de entrada como _start y llamadas al sistema
.data
    Cadenas, variables inicializadas y constantes.
.bss
    Buffers, arreglos y variables sin valor inicial.
Directiva global
    Nombres de funciones o etiquetas que deben ser accesibles externamente.
```

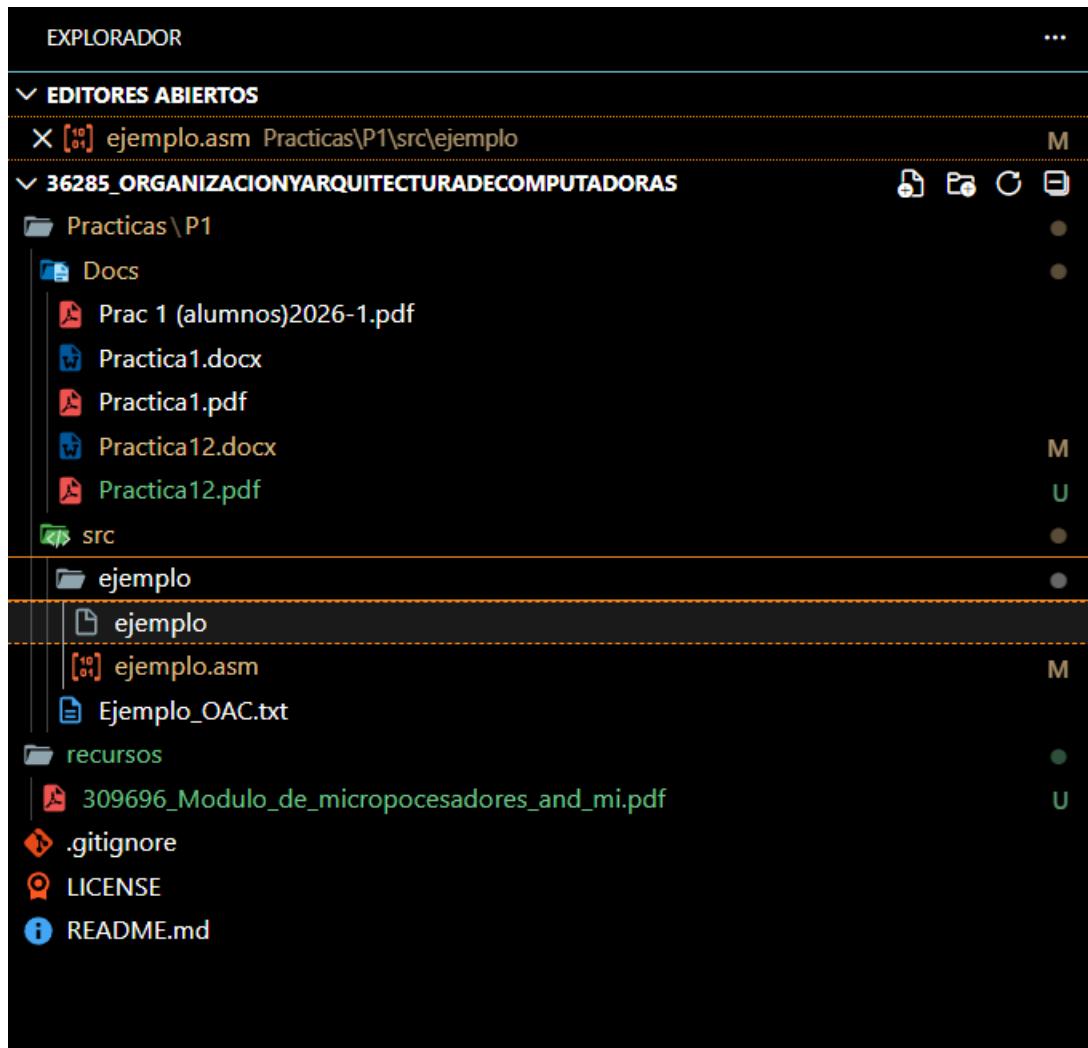
Comandos más populares en Linux y como se usan.

Comando	Función	Ejemplo
cd	Cambiar de directorio	cd carpeta
ls	Listar archivos	ls -l
pwd	Mostrar ruta actual	pwd
mkdir	Crear carpeta	mkdir ejemplo
touch	Crear archivo vacío	touch archivo.asm
rm	Eliminar archivos	rm archivo.asm
cp	Copiar archivos	cp archivo.asm b
mv	Mover / renombrar	mv archivo b
cat	Mostrar contenido	cat archivo.asm
sudo	Ejecutar como administrador	sudo apt update

Que son las extensiones de archivo relacionadas con asm.?

Extensión	Significado	Uso
.asm	Archivo de código ensamblador	Código fuente NASM
.s	Ensamblador AT&T	Usado por GCC
.o	Objeto compilado	Resultado de nasm -f elf
.bin	Binario puro	Código sin encabezados
.elf	Formato ejecutable Linux	Resultado de ld

Evidencia sobre el repositorio y código realizado en github.



Estructura básica del repositorio.

UABC_FCQI_ORGANIZACION_Y_ARQUITECTURA_DE_COMPUTADORAS.

The screenshot shows a code editor window titled "36285_OrganizacionYArquitecturaDeComputadoras". The left pane displays an assembly file named "ejemplo.asm" with the following content:

```
1 global _start
2 section .text
3 _start:
4     ; sys_write(stdout, message, length)
5     mov    eax,    4
6     mov    ebx,    1
7     mov    ecx,    message
8     mov    edx,    length
9     int    80h
10    ; sys_exit(return_code)
11    mov    eax,    1           ;sys_exit syscall
12    mov    ebx,    0           ;return 0 (todo correcto)
13    int    80h
14
15    section .data
16    message: db 'Hello,      world!', 0xA ;mensaje y nueva linea
17    length: equ $-message             ;obtenemos la longitud de la cadena
```

The right pane shows an "EXPLORADOR" (File Explorer) with the following directory structure:

- Prácticas > P1 > src > ejemplo > exemplo.asm
- 36285_ORGANIZACIONYARQUITECTURADECOPUTADORAS
- vscode
- tasks.json
- Prácticas/P1
- Docs
- Prac 1 (alumnos)2026-1.pdf
- Practica1.docx
- Practical.pdf
- Practica12.docx
- Practica12.pdf
- src
- ejemplo
- ejemplo
- ejemplo.asm
- Ejemplo_OAC.txt
- recursos
- 309696_Modulo_de_micropocesadores_and_mi.pdf
- .gitignore
- LICENSE
- README.md

At the bottom, the status bar shows: Joshua Osorio (Hace 11 horas) Lin. 18, col. 91 Tamaño de tabulación: 4 UTF-8 LF {} MIPS Assembly Indents: 0 Prettier ESP LAA 9:39 AM 2/5/2026

Archivo .asm con código para imprimir Hola mundo.

The screenshot shows a code editor window titled "ejemplo.asm - 36285_OrganizacionYArquitecturaDeComputadoras [Codespaces: expert space system]". The left pane displays the same assembly file content as the previous screenshot. The right pane has a "TERMINAL" tab open, showing the following command history:

```
joeOsorio ~/workspaces/36285_OrganizacionYArquitecturaDeComputadoras (main) $ cd Prácticas/P1/src/ejemplo
joeOsorio ~/workspaces/36285_OrganizacionYArquitecturaDeComputadoras (main) $ nasm -f elf ejemplo.asm
joeOsorio ~/workspaces/36285_OrganizacionYArquitecturaDeComputadoras (main) $ ld -m elf_i386 -s -o ejemplo exemplo.o
joeOsorio ~/workspaces/36285_OrganizacionYArquitecturaDeComputadoras (main) $ ./ejemplo
Hello, world!
joeOsorio ~/workspaces/36285_OrganizacionYArquitecturaDeComputadoras (main) $
```

At the bottom, the status bar shows: joeOsorio (Hace 5 días) Lin. 18, col. 79 Espacios: 4 UTF-8 LF {} Texto sin formato Finish Setup Dispositivo: US 10:30 AM 2/15/2026

Ejecución de instrucciones para ensamblar y enlazar programa.

Conclusiones

En esta práctica nos introducimos al mundo de ensamblador, creando un repositorio en GitHub con una estructura simple, pero con la que trabajaremos a lo largo del curso. Además, con la investigación nos ayuda a familiarizarnos un poco mas sobre la estructura básica del lenguaje ensamblador x86 realizando el primer hola mundo, relacionamos algunos comandos básicos.

Bibliografía

- [1] Roger Baig I Viñas, *Sistema operativo GNU/Linux básico*, 1.^a ed. 2003. Disponible en:
<https://redua.wordpress.com/wp-content/uploads/2008/04/868.pdf>