



Práctica 1

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.

Desarrollo

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?
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:

```

1 global _start
2 section .text
3 _start:
4
5     ; sys_write(stdout, message, length)
6     mov eax, 4
7     mov ebx, 1
8     mov ecx, message
9     mov edx, length
10    int 80h
11
12    ; sys_exit(return_code)
13    mov eax, 1          ; sys_exit syscall
14    mov ebx, 0          ; return 0 (todo correcto)
15    int 80h
16
17 section .data
18
19    message: db 'Hello, world!',0x0A      ; mensaje y nueva linea
20    length: equ $-message                ; obtenemos la longitud de la cadena
21

```

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 exemplo.o`
15. Ejecutar el programa obtenido: `./ejemplo`

Conclusiones

Dificultades