

MANUAL TECNICO

Requisitos del sistema para realizar el programa

- **Hardware:**
 - memoria RAM de 512 MB
 - disco duro: 80 GB
 - Resolucion de pantalla 1920 x 1080
- **Software:**
 - GDB version: GNU gdb (Ubuntu 12.1-0ubuntu1~22.04.2) 12.1
 - QEMU version: qemu-aarch64 version 6.2.0 (Debian 1:6.2+dfsg-2ubuntu6.22)
 - Visual Studio Code 1.93.1
- **Sistema operativo:** cualquier distrucion de linux, para realizar este proyecto se utilizo Ubuntu

Pasos para instalar el Debugger y el Ensamblador QEMU:

- `sudo apt-get update`
- `sudo apt-get install qemu-user qemu-user-static gdb-multiarch build-essential`

Instrucciones para ejecutar el ensamblador y el debugger

En una terminal ejecuta lo siguiente:

- `aarch64-linux-gnu-as -mcpu=cortex-a57 nombreArchivo.s -o nombreArchivo.o`
- `aarch64-linux-gnu-ld nombreArchivo.o -o nombreArchivo`
- `qemu-aarch64 -g 1234 ./nombreArchivo`

QEMU esta esperando el GNU Debugger (GDB) para ser ejecutado. Abre una segunda terminal mientras la terminal de QEMU se esta ejecutando y ejecuta el siguiente comando:

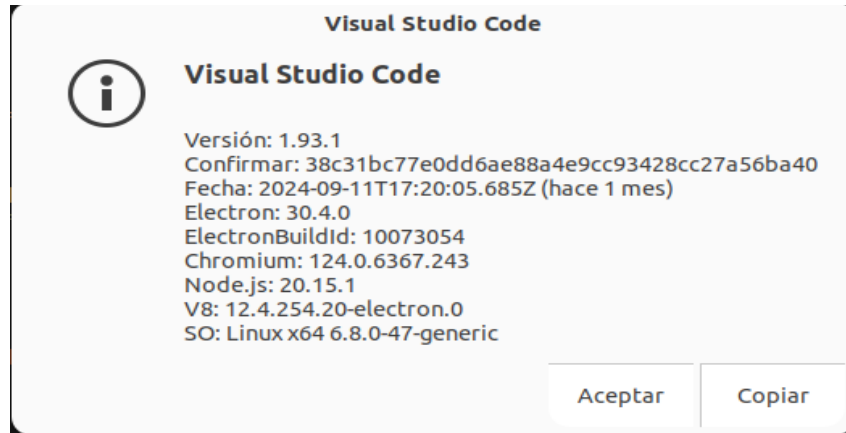
```
gdb-multiarch -q --nh \  
-ex 'set architecture aarch64' \  
-ex 'file hello_world' \  
-ex 'target remote localhost:1234' \  
-ex 'layout split' \  
-ex 'layout regs'
```

Si deseas automatizar lo anterior y ejecutar todo en un solo comando realiza lo siguiente:

- Para poder automatizar la ejecución del compilado con el emulador QEMU se creo un script llamado compile.sh
- Ejecuta el archivo compile.sh en una terminal con la siguiente instrucción: sh compile.sh
- Luego te pedira el nombre del archivo el cual es ordenamiento.

Programa y sistema operativo usados para realizar el codigo ensamblador

Version de Visual Studio Code y LINUX Ubuntu



Instrucciones AArch64 codigo ensamblador para realizar los diferentes comandos

- SUMA [Número o celda] Y [Número o celda]
add x9, x9, x10
- RESTA [Número o celda] Y [Número o celda]
sub x9, x9, x10
- MULTIPLICACION [Número o celda] Y [Número o celda]
mul x9, x9, x10
- DIVIDIR [Número o celda] ENTRE [Número o celda]
udiv x9, x9, x10

- POTENCIAR [Número o celda] A LA [Número o celda]
se utilizo un ciclo para calcular el resultados

```
ldr x8, =param1 //contiene el primer valor entero
ldr x9, [x8]
ldr x11, =param2 //contiene el segundo valor entero
ldr x10, [x11]
```

```
mov x2, 1
```

```
pot_loop:
```

```
    mul x2, x2, x9 // x2 = resultado 2, 4, 8
    sub x10, x10, 1 // exponente, 3 = 2, 1
```

```
    cmp x10, 0
    bne pot_loop // Repetir el bucle
```

- OLOGICO [Número o celda] Y [Número o celda]
orr x9, x9, x10
- YLOGICO [Número o celda] Y [Número o celda]
and x9, x9, x10
- OXLOGICO [Número o celda] Y [Número o celda]
eor x9, x9, x10
- NOLOGICO [Número o celda]
mvn x9, x9