1 Instalación

1. Tener actualizados los repositorios

```
sudo apt update
```

2. Setting up AArch64 toolchain

```
sudo apt install gcc-aarch64-linux-gnu
```

3. Setting up QEMU ARM (incluye AArch64)

```
sudo apt install qemu-system-arm
```

4. Fetch and build AArch64 GDB

```
sudo apt install gdb-multiarch
```

5. Configurar GDB para que haga las cosas más amigables

```
wget -P ~ https://github.com/cyrus-and/gdb-dashboard/raw/master/.gdbinit
```

Esto crea un archivo llamado .gdbinit en el directorio personal que configura el GDB para funcionar como un dashboard.

2 Ensamblado

1. Escribir el programa a simular en el template main.s
2. Compilar utilizando el Makefile en la carpeta de quemu descompimida

   Si no tiene instalado make, instalarlo con el siguiente comando

   ```
   sudo apt install make
   ```

   ```
   make clean
   ```

3. Volver a armar el Makefile

```
make
```

3 Inicio del emulador

1. Iniciar el emulador del microprocesador ARM (controle que todo este en una sola linea)

```
qemu-system-aarch64 -s -S -machine virt -cpu cortex-a53 -machine
type=virt -nographic -smp 1 -m 64 -kernel kernel.img
```

4 Inicio del debugger

1. Iniciar debugger GDB (controle que todo este en una sola linea)

```
gdb-multiarch -ex "set architecture aarch64" -ex "target remote localhost:1234"
```

2. Configurar la arquitectura a utilizar

```
set architecture aarch64
```

3. Importar al GDB los símbolos de debug en la dirección de memoria donde se encuentra el programa

```
add-symbol-file main.o 0x0000000040080000
```

5 Configuración del Dashboard

El dashboard es un interfaz visual modular que se utiliza para mostrar de forma más amigable la información relevante para realizar el debugging. Algunas de las secciones más importantes son: • Assembly • Memory • Registers • Source

6 Ejecución paso a paso con GDB

• El comando que se utiliza para ejecutar una única instrucción de assembly es:

```
stepi
```

• Si se desea ejecutar un bloque de n instrucciones:

```
stepi n
```

• Otra forma de ejecutar más de una instrucción es utilizando breakpoints:

```
break n instr
```

```
break etiqueta
```

• Luego de indicar los breakpoints, mediante el comando continue se ejecuta el programa hasta encontrar el primer breakpoint:

```
continue
```

• Para ver todos los breakpoints existentes:

```
info breakpoints
```

• Para eliminar todos breakpoints:

```
delete breakpoints
```