

## MANUAL TECNICO

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:

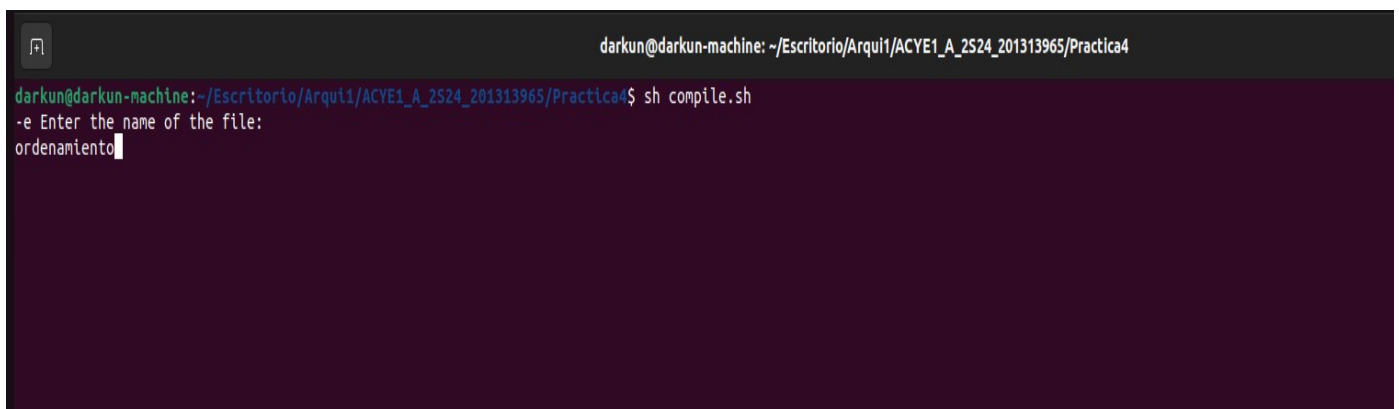
- `aarch64-linux-gnu-as -mcpu=cortex-a57 nombreArchivo.s -o nombreArchivo.o`
- `aarch64-linux-gnu-ld nombreArchivo.o -o nombreArchivo`
- `qemu-aarch64 -g 1234 ./nombreArchivo`

**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`.



The screenshot shows a terminal window with a dark background. The title bar at the top reads "darkun@darkun-machine: ~/Escritorio/Arqui1/ACYE1\_A\_2S24\_201313965/Practica4". The terminal content shows the prompt "darkun@darkun-machine:~/Escritorio/Arqui1/ACYE1\_A\_2S24\_201313965/Practica4\$ sh compile.sh". Below this, a message "-e Enter the name of the file:" is displayed, followed by the input "ordenamiento" with a cursor at the end.

El programa mostrara un encabezado con informacion

```
darkun@darkun-machine: ~/Escritorio/Arqui1/ACYE1_A_2S24_201313965/Practica3

Universidad de San Carlos de Guatemala
Facultad de Ingenieria
Escuela de Ciencias y Sistemas
Arquitectura de Computadores y Ensambladores 1
Seccion A
Luis Fernando Gonzalez Real
201313965
█
```

Presiona la tecla enter para continuar y el programa mostrara la opcion de menu:

```
darkun@darkun-machine: ~/Escritorio/Arqui1/ACYE1_A_2S24_201313965/Practica4

>> Menu Principal
1. Ingreso de lista de numeros
2. Bubble Sort
3. Quick Sort
4. Insertion Sort
5. Merge Sort
6. Salir
Ingrese Una Opcion: █
```

Elige la primera opcion para que el programa te muestre un submenu donde puedes elegir la forma en que ingresaras la lista de numeros.

```
darkun@darkun-machine: ~/Escritorio/Arqui1/ACYE1_A_2S24_201313965/Practica4

>> subMenu
1. De forma manual
2. Carga de Archivo csv
3. Regresar al menu anterior
Ingrese Una Opcion: 2 █
```

Si ingresas la opcion 2 el programa te pedira que ingreses el nombre del archivo csv que contendra los numeros a ordenar, luego presiona la tecla enter para continuar y te mostrara un mensaje, luego vuelve a presionar la tecla enter para continuar.

```
darkun@darkun-machine: ~/Escritorio/Arqui1/ACYE1_A_2524_201313965/Practica4
>> subMenu
1. De forma manual
2. Carga de Archivo csv
3. Regresar al menu anterior
Ingrese Una Opcion: 2
Ingrese el nombre del archivo: entrada.csv

El Archivo Se Ha Leido Correctamente
```

Luego de presionar la tecla enter para continuar esto te regresa al menu principal para volver a ingresar una nueva opcion.

Ahora debes de elegir el metodo de ordenamiento para los numeros

```
darkun@darkun-machine: ~/Escritorio/Arqui1/ACYE1_A_2524_201313965/Practica4
>> Menu Principal
1. Ingreso de lista de numeros
2. Bubble Sort
3. Quick Sort
4. Insertion Sort
5. Merge Sort
6. Salir
Ingrese Una Opcion: 2
```

Si eliges la opcion 2 el programa, empezara a ordenar los numeros paso a paso, luego debes de presionar la tecla enter para continuar y te mostrara el resultado final de los numeros ordenados. Ademas, el programa creara un archivo llamado reporte.txt con los resultados. Vuelve a presionar enter para continuar y el programa te redregira automaticamente al menu principal.

```
darkun@darkun-machine: ~/Escritorio/Arqui1/ACYE1_A_2524_201313965/Practica4
>> Menu Principal
1. Ingreso de lista de numeros
2. Bubble Sort
3. Quick Sort
4. Insertion Sort
5. Merge Sort
6. Salir
Ingrese Una Opcion: 2
Conjunto Inicial: 12 1 3 23 45 32 23 2 25 6 54 6 43 4 3 4
Paso 1: 1 12 3 23 45 32 23 2 25 6 54 6 43 4 3 4
Paso 2: 1 3 12 23 45 32 23 2 25 6 54 6 43 4 3 4
Paso 3: 1 3 12 23 32 45 23 2 25 6 54 6 43 4 3 4
Paso 4: 1 3 12 23 32 23 45 2 25 6 54 6 43 4 3 4
Paso 5: 1 3 12 23 32 23 2 45 25 6 54 6 43 4 3 4
Paso 6: 1 3 12 23 32 23 2 25 45 6 54 6 43 4 3 4
Paso 7: 1 3 12 23 32 23 2 25 6 45 54 6 43 4 3 4
Paso 8: 1 3 12 23 32 23 2 25 6 45 6 54 43 4 3 4
Paso 9: 1 3 12 23 32 23 2 25 6 45 6 43 54 4 3 4
Paso 10: 1 3 12 23 32 23 2 25 6 45 6 43 4 54 3 4
```

Para poder salir del programa tienes que ingresar la opcion 6 y luego presionar la tecla enter.

```
darkun@darkun-machine: ~/Escritorio/Arqui1/ACYE1_A_2524_201313965/Practica4
>> Menu Principal
1. Ingreso de lista de numeros
2. Bubble Sort
3. Quick Sort
4. Insertion Sort
5. Merge Sort
6. Salir
Ingrese Una Opcion: 6
darkun@darkun-machine:~/Escritorio/Arqui1/ACYE1_A_2524_201313965/Practica4$
```

Este fue el codigo ensamblador que se uso para escribir el archivo.txt

```
// Macro para leer datos
.macro write buffer, lenbuffer
    // abrir archivo
    mov x0, -100          // abrir
    ldr x1, =nombreArchivo
    mov x2, 101
    mov x3, 0777          // permisos
    mov x8, 56
    svc 0
    mov x5, x0

    mov x0, x5            // file descriptor
    mov x1, 0             // offset 0
    mov x2, 2             // SEEK_END (mover al final del archivo)
    mov x8, 62            // syscall para lseek
    svc 0

    // escribir archivo
    mov x0, x5
    ldr x1, =\buffer
    mov x2, \lenbuffer
    mov x8, 64            // escribir
    svc 0

    // cerrar archivo
    mov x0, x5            // file descriptor
    mov x8, 57            // cerrar archivo
    svc 0
.endm
```

La siguiente línea de código se utilizó para obtener el primer valor de la lista de números, lsl es para aplicar un corrimiento.

```
LDRH w4, [x3, x9, LSL 1]
```

LDR: carga un valor de 64(x) o 32 bits(w)

LDRH : carga un valor de 16 bits (2 bytes)

LDRB: carga un valor de 8 bits (1 byte)

Aumenta en 1 el valor de 9, para luego obtener el siguiente valor de la lista de números.

```
ADD x9, x9, 1
```

```
LDRH w5, [x3, x9, LSL 1] // array[i + 1]
```

Cada vez que se ingrese una nueva lista de números a ordenar es necesario resetear el valor de array con la siguiente instrucción en el código.

reset\_array:

```
ldr x0, =array
```

```
mov x10, 128
```

```
mov x3, 0
```

reset\_array\_zero:

```
str x3, [x0], 8
```

```
subs x10, x10, 1
```

```
cmp x10, 0
```

```
bne reset_array_zero
```

```
ret
```

También es necesario resetear el valor de count que contiene la cantidad de números de la lista.

reset\_count:

```
ldr x0, =count
```

```
mov x3, 0
```

```
str x3, [x0]
```

```
ret
```