

1 Compilar e executar o seguinte programa escrito em Assembler (test.S)

```
.data                                #initialized data section
msg:
    .ascii "Ola Mundo\n" #ascii string
    len = . - msg    #number of bytes of the string
.text
.global _start
_start:
    movq $1, %rax    #system call number 1 is write
    movq $1, %rdi    #file descriptor of standard output is 1
    movq $msg, %rsi   #message address
    movq $len, %rdx   #number of bytes
    syscall          #raise system call interrupt write(...)

    movq $60, %rax    #system call number 60 is exit
    movq $1, %rdi    #exit status code can be caught in bash with $?
    syscall          #exit(1)
```

Compilar e executando, chamando (1) o assembler (parte integrante do cc/gcc) (2) seguido pelo linker static (tambem, parte integrante do cc/ggc mas por defeito espera duma main que neste caso não existe – por isto a chamada direito) (3) execução do programa ./test e (4) chamando no bash shell echo \$? Permite imprimir o valor de retorno do ultimo programa terminado no bash shell.

```
cc -c test.S
ld -o test test.o
./test
echo $?
```

Nota cc -c test.S é equivalente a chamar diretamente o assembler [as test.S -o test.o]

Para ver o sintaxe da chamada “write”

<https://www.di.ubi.pt/~operativos/praticos/pdf/7-fileIO.pdf>

ou

<https://man7.org/linux/man-pages/man2/write.2.html>

Para saber os números de sys calls e registos do cpu que correspondem aos parâmetros.

https://chromium.googlesource.com/chromiumos/docs/+/_master/constants/syscalls.md

Os Programas foram todos testados no Windows11 64 bit usando o bash shell e gcc (11.4) do WSL

2 Experimentar o programa seguinte e explicar o seu output.

```
#include <stdio.h>
int main(){
    printf("ola\n");
    printf("mundo");
    asm("cli");
    printf("\nfim\n");
    return 0;
}
```

- Execute o programa com a ferramenta **strace**
- Experimentar também com `printf("mundo\n");`
- Experimentar desligando o buffering do stdout, insere a linha `setvbuf(stdout, NULL, _IONBF,0);` logo a seguir q main.

Notas: https://en.wikipedia.org/wiki/Interrupt_flag