

Ejemplo “Hola mundo!”

Arquitectura del Computador - LCC - FCEIA-UNR

16 de agosto de 2022

En este documento vamos a ver el típico ejemplo que imprime por pantalla la frase “Hola mundo!”. El siguiente código en lenguaje C (guardado en el archivo `hola.c`) imprime por pantalla dicha frase:

```
#include <stdio.h>
int main(){
    printf("Hola Mundo!\n");
    return 0;
}
```

Podemos compilar utilizando la opción `-g` para poder *debuggear* utilizando GDB:

```
$ gcc -g -o hola hola.c
```

Luego podemos ejecutar:

```
$ ./imprime
```

y obtenemos por pantalla

```
Hola mundo!
```

Ahora ejecutemos utilizando GDB:

```
gdb ./hola
```

Una vez que estamos dentro de la sesión de *debugging*, ponemos un *breakpoint* en `main` y ejecutamos el comando `run` pasandole los argumentos anteriores:

```
(gdb) br main
Breakpoint 1 at 0x1139: file hola.c, line 4.
(gdb) r
Starting program: /home/dferoldi/hola

Breakpoint 1, main () at hola.c:4
4         printf("Hola Mundo!\n");
```

Ahora ejecutemos el comando `s` para meternos dentro de la función `printf`. Una vez dentro de la función `printf` veamos el contenido del registro del `rdi`, que es el registro donde viene el primer argumento según la convención de llamada:

```
(gdb) i r rdi
rdi                0x555555556004      93824992239620
```

De acuerdo a la convención en `rdi` viene la dirección de memoria donde está alojada la cadena de formato de `printf`. Veamos el contenido de memoria de dicha dirección para corroborar, viendo los primeros 13 bytes a partir de dicha dirección en formato carácter¹:

¹Ver en detalle el comando `examine` en la documentación de GDB.

```
(gdb) x/13cb $rdi
0x402004:      72 'H'  111 'o' 108 'l' 97 'a' 32 ' ' 77 'M' 117 'u' 110 'n'
0x40200c:      100 'd' 111 'o' 33 '!' 0 '\000'
```

Vemos que efectivamente a partir de la dirección 0x555555556004 está almacenada la cadena de formato `Hola Mundo!`, finalizada con el caracter nulo².

Veamos ahora un ejemplo de código en Assembler x86-64 para imprimir la cadena `Hola mundo!`:

```
.data
str: .string "Hola mundo!\n"

.text
.global main
main:
    movq $str, %rdi    # En rdi cargamos la dirección de cadena de formato str
    movq $0, %rax      # Cantidad de argumentos de tipo vectorial
    call printf        # Llamamos a printf
    movq $0, %rax      # Retornamos 0
    ret
```

Compilando con la bandera `-no-pie` porque tenemos un segmento de datos:

```
gcc -g -no-pie -o hola hola.s
```

y *debuggeando* con GDB:

```
gdb ./hola
```

Una vez ejecutado GDB podemos ver nuevamente la cadena de formato:

```
(gdb) x/13cb $rdi
0x404030:      72 'H'  111 'o' 108 'l' 97 'a' 32 ' ' 109 'm' 117 'u' 110 'n'
0x404038:      100 'd' 111 'o' 33 '!' 10 '\n' 0 '\000'
```

²Aquí no se está mostrando el caracter de fin de línea.