



Universidad de Costa Rica
Escuela de Ciencias de la Computación e Informática
CI0118 Lenguaje Ensamblador, grupo 01
Fecha: 01/07/2019, I ciclo lectivo 2019
Tarea Corta # 2



Estudiante

Iván Chavarría Vega, B72097

1. Introducción

Esta tarea programada tiene como objetivo investigar acerca del llamado de funciones recursivas en ensamblador, y a su vez, el manejo adecuado de la pila en esta. Todas las instrucciones utilizadas para la programación en lenguaje ensamblador del programa fueron tomadas de [1].

En este caso, tendremos la siguiente función en lenguaje C.

```
void fun(int n){  
    if(n > 0){  
        fun(n-1);  
        printf("%d ", n);  
        fun(n-1);  
    }  
}
```

Nuestro objetivo es programarla en ensamblador, lograr que reproduzca los resultados que se obtiene en C. El n que recibe lo hace desde una función **main.c** la cual le envía 4 como argumento.

2. Descripción de la aplicación

El programa en sí, es relativamente sencillo, lo único con lo que hay que tener cuidado es el manejo correcto de la pila y los registros permitidos a usar.

Una vez iniciada la ejecución, *main.c* hace un llamado a la subrutina *fun.asm* pasándole como argumento un 4. Este argumento la subrutina en ensamblador lo recibe en el registro *rdi*. Lo primero que se hace es crear el *stack frame* de la forma estándar, seguidamente, chequeamos si *rdi* es menor o igual a 0, si lo es, entonces el programa brinca a una etiqueta *last* y termina, si no, el programa continua y volvemos a hacer el llamado a la subrutina con $rdi - 1$, el cual hará el mismo chequeo y los mismo llamados.

3. Ejecución del Programa

Los comandos utilizados para ejecutar el programa han sido los siguientes:

```
yasm -f elf64 fun.asm -o fun.o  
gcc -no-pie -o [ejecutable] fun.o main.c
```

Los resultados obtenidos al llamar el programa con la subrutina *fun* en C, los cuales deberían ser resultados a obtener cuando programamos la subrutina en ensamblador, se puede observar en Figura 1

```
ivanj@IvanJ:~/Desktop/programada4$ gcc main.c -o funmain
ivanj@IvanJ:~/Desktop/programada4$ ./funmain
1 2 1 3 1 2 1 4 1 2 1 3 1 2 1
ivanj@IvanJ:~/Desktop/programada4$
```

Figura 1: Resultados del programa en C

```
ivanj@IvanJ:~/Desktop/programada4$ yasm -f elf64 fun.asm -o fun.o
ivanj@IvanJ:~/Desktop/programada4$ gcc -no-pie -o funxd fun.o main.c
ivanj@IvanJ:~/Desktop/programada4$ ./funxd
1 2 1 3 1 2 1 4 1 2 1 3 1 2 1
ivanj@IvanJ:~/Desktop/programada4$
```

Figura 2: Resultados del programa con el llamado a ensamblador

Mientras que Figura 2 muestra los resultados que el programa retorna haciendo el llamado a la subrutina *fun* en ensamblador.

Con esto podemos notar que la subrutina fue programada correctamente.

Referencias

- [1] E. Jorgensen, *x86-64 Assembly Language Programming with Ubuntu*. 2018.