



Universidad Simón Bolívar
Curso: CI3815 / Organización del Computador
Trimestre: Septiembre-Diciembre, 2022
Profesor: Fernando Torre Mora
Estudiante: Junior Miguel Lara Torres - Carnet: 17-10303

Asignación 1

1. ¿Cuántas instrucciones escritas por el programador tiene este programa? Cuenta a partir de la instrucción *li \$t0 4*.

Este programa contiene un total de 17 instrucciones escritas por el usuario a partir de la instrucción *li \$t0 4*.

2. ¿Cuántas instrucciones son generadas al cargar este programa en MARS? Cuenta a partir de la instrucción *li \$t0 4*.

La cantidad de instrucciones generadas por el ensamblador en MARS es de 19.

3. ¿Cuántos bytes ocupa el programa en memoria a partir de la instrucción *li \$t0 4*? Considerar sólo el área de texto.

La cantidad de Bytes cargados en memoria al cargar el programa es de 4c en Hexadecimal, lo que equivalentemente es 76 Bytes en Decimal. Así mismo, como tenemos 19 instrucciones generadas por el ensamblador y cada instrucción pesa 4 Bytes, entonces $4 \times 19 = 76$ Bytes.

4. ¿Qué actividad realiza el programa? Indique cómo llegó a esa conclusión.

Se observa un bucle en el programa, tenemos un total de 4 iteraciones, cuyo valor es cargado en registro *t0* y que se van reduciendo una vez corre el programa (Línea de código 12). A su vez, si quitamos las líneas 19 y 20 no se carga ninguna información en la RAM al finalizar el programa. Las líneas 22, 23 y 24 permite que al final del programa se almacene la información del carnet en un orden específico. Las líneas 26 y 27 almacenan dos dígitos del carnet del estudiante. Las líneas 29, 30 y 31 permiten restar 1 al iterador, sumar 2 a los registros que almacenan la dirección del carnet e ing. En particular ese suma 2 me da a entender que estamos recorriendo el carnet de 2 dígitos en 2 dígitos. Por tanto, el programa va recorriendo el carnet de 2 en 2 y guardando dichos dígitos en la RAM.

5. ¿En qué dirección de memoria se cargó el primer y último dígito de su carnet?

Usando el lenguaje ASCII. Se almacenan en la dirección 0x10010000, para el primer dígito se encuentra en la columna VALUE (+14) en el byte número 22 y el último dígito en la columna VALUE (+18) en el byte número 28.

6. Cargue nuevamente su programa y coloque un breakpoint en la dirección donde se encuentra la etiqueta bp. Ejecute y dé una tabla indicando cómo varían los registros a los largo de la ejecución del programa. Debe incluir todos los registros cuyos valores sufran alguna variación en algún momento de la ejecución del programa.

Desactivando la vista en valores Hexadecimales tenemos la siguiente tabla

Iteración	1	2	3	4
\$at	268500992	268500992	268500992	268500992
\$v0	0	0	0	10
\$t0	3	2	1	0
\$t1	268500997	268500999	268501001	268501003
\$t2	268501014	268501016	268501018	268501020
\$s0	55	48	48	0
\$s1	49	49	51	51