Tarea I Lenguaje ensamblador, determinación de palindromos

Luis Carlos Quesada Rodríguez, Lucía Elizondo.

29 de abril de 2019

1. Introducción

Para la tarea I del curso Lenguaje ensamblador se lleva a cabo la creación de un algoritmo para determinar si una hilera es palíndromo. Para este problema se toman 8 hileras, se identifica si son palíndromos y se retorna el resultado a forma de *EXIT_status*, por lo que el numero decimal en el *EXIT_status* al convertirse a binario represente en cada dígito, con un 1 si la hilera fue palíndromo y con un 0 en caso de que no lo fuera.

2. Funcionamiento

La comprobación de palíndromo funciona de la siguiente manera:

1. Se toma la dirección de la ultima hilera como índice izquierdo.
2. Se busca la dirección del caracter 0 en esa hilera y se le resta 1 para tomarlo como índice derecho.
3. Se compara la dirección de los caracteres para ver si el izquierdo es mayor al derecho.
4. Si el izquierdo es menor igual al derecho se considera palíndromo.
5. Se comparan los caracteres para comprobar su igualdad.
6. Si son iguales se suma uno al índice izquierdo y resta uno al índice derecho y vuelve al paso 3.
7. Si son diferentes, se considera que esta hilera no es palíndromo.

Para la representación del dato se debe utilizar un único valor numérico entre 0 y 1023, pues debe ser representado en 8 bits por convenciones del sistema operativo. La limitación del sistema operativo nos obliga a utilizar 2 programas en vez de uno para representar las 16 hileras de caracteres por lo que los datos

se dividen en dos secciones de 8 hileras. Durante la comprobación de palíndromos los resultados fueron almacenados uno a uno en la pila, de esta forma al extraerlos se encuentran de forma inversa, de la misma forma las hileras fueron recorridas de forma inversa para obtener los datos en orden de la siguiente forma:

1. Se inicia un contador en 8
2. Se toma un registro en 0.
3. Se hace un corrimiento de bits izquierdo de un bit.
4. Se hace un *pop* en la pila.
5. Se suma el resultado encontrado en la pila.
6. Se hace un loop al paso 3
7. Finalmente se guarda el resultado de las sumas y se retorna.

Al ejecutar el programa, los bits que representan las hileras encuentran en el mismo orden que se introducen en la sección *.data* del programa.

3. Compilación

Para la compilación del programa se escribió el siguiente script, de forma que el código tanto de la parte A como la parte B es ensamblado y linkeado en una sola ejecución.

```
#Ensamblar el programa con la primer mitad de los datos
yasm -g dwarf2 -f elf64 parteA.asm -l ListaA.lst

#Linkear el programa con la primer mitad de los datos
gcc -static -o parteA parteA.o


#Ensamblar el programa con la segunda mitad de los datos
yasm -g dwarf2 -f elf64 parteB.asm -l ListaB.lst

#Linkear el programa con la primer mitad de los datos
gcc -static -o parteB parteB.o
```

El script se encuentra en el entregable con el nombre *compile.sh*, para ser ejecutado simplemente debe utilizarse el siguiente comando.

```
./compile.sh
```

4. Ejecución del programa

Para obtener los resultados del programa se utiliza la variable global de *EXIT_status* representada como $?, cada programa debe ser ejecutado independientemente para obtener un código de salida de 8 bits. esto se puede lograr de la siguiente forma:

```
#Ejecutar la parte A
./parteA

#Imprimir el valor de A
echo $?

#Ejecutar la parte B
./parteB

#Imprimir el valor de B
echo $?
```

Sin embargo, para comodidad se creó un script llamado *palin.sh* que se encuentra en el archivo entregable que ejecuta ambos programas y entrega un unico resultado.

```
#Conseguir el codigo de A
./parteA
parteA=$?

#Conseguir el codigo de B
./parteB
parteB=$?

#Hacer Bitshift de 8 en A
results=$(( parteA << 8 ))

#Sumar A con B
results=$(( results + parteB ))

#print de los resultados
echo $results
```

Como se puede ver, para el resultado final se corre A en 8 bits a la izquierda, luego se le suma B y de esta forma representa en 16 bits la totalidad de las hileras de datos.

El resultado generado por el shell script devuelve 56749, lo cual es 11011101 10101101 en binario. Esto indica que la hilera0, hilera2, hilera3, hilera5, hilera7, hilera8, hilera10, hilera11, hilera12, hilera14, hilera15 son palíndromos. Las otras hileras: hilera1, hilera4, hilera6 ,hilera9, hilera13 no son palíndromos.