

Instituto Tecnológico de Costa Rica
Centro Académico de Alajuela
IC3101. Arquitectura de Computadoras



Laboratorio #2:
Reconociendo palíndromos en ensamblador NASM

Grupo 6:
Marco Rodriguez Vargas – Carné: 2022149445
Ion Angel Dolanescu Bravo – Carné: 2022049034
Alejandro Campos Paredes – Carné: 2022058238

Profesor:
Ing. Emmanuel Ramírez Segura

Fecha de entrega:
5/10/2022

II Semestre, 2022

Objetivo:	2
Descripción de la solución:	3
Logros o fallos	6
Bibliografía	6
ANEXO:	7

Objetivo:

Entender cómo hacer manipulación de arreglos en ensamblador como mecanismo para detectar palíndromos en ASCII.

Descripción de la solución:

Nuestro proceso de entendimiento nos llevó a primero plantearnos cuales macros podrían servirnos para la realización del programa así que usamos los siguientes(Nota:los macros fueron creados por el profesor,todo el credito a el):

“input” ya que el palíndromo iba a ser escrito en la terminal así que ocupamos guardar la entrada

“output” ya que ocupamos imprimir en la terminal si era palindromo o no

```
2      ; Macros prestados del profesor
3
4      %macro input 0
5          mov eax, 3
6          mov ebx, 0
7          mov ecx, entrada ; dirección para guardar la entrada.
8          mov edx, 255 ; tamaño de la entrada a leer.
9          int 80h
10     %endmacro
11     %macro output 2
12         mov eax, 4
13         mov ebx, 1
14         mov ecx, %1
15         mov edx, %2
16         int 80h
17     %endmacro
18
```

Luego llenamos la “section .data” y la “section. bss”

En estas secciones definimos distinta información de relevancia como por ejemplo las oraciones que van a salir en el output y sus tamaños

Además de reservar el espacio en memoria para la entrada

```
18  section .data
19
20      ; Mensajes.
21      inicio db "Bienvenido, ingrese su palabra:  "
22      lenIni equ $-inicio
23      isPal db "La palabra es palindromo.", 10,13
24      lenIs equ $-isPal
25      isNot db "La palabra no es palindromo", 10,13
26      lenNot equ $-isNot
27
28      text times 255 db 0
29      textSize equ $-text
30
31  section .bss
32      ; Longitud del formato de la entrada.
33      entrada: resb 255
34      lenEnt: resb 1
35
```

Luego en la “section .text” es donde ocurre toda la lógica del programa primero pedimos un input

luego calculamos la longitud del input por medio de un loop que recorre caracter por caracter una palabra hasta encontrar “null” la cantidad de veces que se haga el loop es el tamaño de la palabra

Una vez con la longitud hacemos un ciclo que consta de lo siguiente comparamos la primera letra de la palabra con la última (sabemos cual es la última por la longitud)

si son diferentes saltan al caso de no son palíndromos(el caso fallo)

si son iguales pasamos a la segunda letra(le sumamos 1 al valor inicial) y a la penúltima (le restamos 1 a la “longitud”) y las comparamos y así sucesivamente con todas las posiciones de la palabra, en caso de que se recorra la palabra completa y

nunca se encuentre ninguna diferencia pasamos al caso de si son palíndromos(caso done)

y con eso saldríamos del programa

```

39  section .text
40      global _start
41      extern puts
42  _start:
43      output inicio, lenIni
44      input
45  inputLen: ; Calcula la longitud de la entrada
46      mov eax, entrada
47      mov ecx, -1
48  loop:
49      inc ecx
50      cmp byte [eax+ecx],0
51      jne loop
52  dec ecx
53  dec ecx ; Retrocede hasta la ultima letra saltandose el salto de linea
54  mov edx, ecx
55  mov ebx, entrada
56  mov eax, ebx
57  revision:
58      cmp byte [ebx], 10
59      je done
60      mov dl, [ebx]
61      cmp dl, [eax+ecx] ; Se comparan 2 caracteres X distantes uno de otro.
62      jne fallo ; si la comparación dice que son distintos llama a fallo y termina.
63      inc ebx
64      dec ecx
65      jmp revision
66  fallo:
67      output isNot, lenNot
68      jmp salir
69  done:
70      output isPal, lenIs
71  salir:
72      mov eax, 1
73      mov ebx, 0
74      int 80h
75

```

Lecciones aprendidas:

Por medio de la realización del laboratorio mejoramos mucho nuestro conocimiento de cómo trabajar en ensamblador, los más destacados fueron aprender a utilizar inputs y outputs, entender cómo funcionan las palabras en ensamblador (conseguir su longitud, recorrerlas, etc), además de aprender a implementar macros en los programas

Logros o fallos

Logramos de manera exitosa poder identificar por medio de ensamblador si una palabra es palindromo o no. Tuvimos varias dificultades con la sintaxis pero finalmente lo pudimos conseguir.

```
mriontv@mriontv-B560-DS3H-AC-Y1:~/Desktop$ nasm -f elf64 palindromo.asm
mriontv@mriontv-B560-DS3H-AC-Y1:~/Desktop$ ld -s -o pal palindromo.o
mriontv@mriontv-B560-DS3H-AC-Y1:~/Desktop$ ./pal
Bienvenido, ingrese su palabra:  cebolla
La palabra no es palindromo
mriontv@mriontv-B560-DS3H-AC-Y1:~/Desktop$ ./pal
Bienvenido, ingrese su palabra:  amananama
La palabra es palindromo.
mriontv@mriontv-B560-DS3H-AC-Y1:~/Desktop$ ./pal
Bienvenido, ingrese su palabra:  AmananamA
La palabra es palindromo.
mriontv@mriontv-B560-DS3H-AC-Y1:~/Desktop$ ./pal
Bienvenido, ingrese su palabra:  Amananama
La palabra no es palindromo
mriontv@mriontv-B560-DS3H-AC-Y1:~/Desktop$
```

Como se ve en la imagen, el programa logra saber cuando una palabra es palindromo o cuando no lo es.

Nota: "Amanama" aunque sean las mismas letras el ascii de A mayúscula es diferente a la minúscula así que no es palindromo

Bibliografía

Ferrer. (n.d.). Programacion en lenguaje ensamblador con NASM. slideshare.

<https://www.slideshare.net/JesusAntonioFerrerSanchez/programacin-en-lenguaje-en-samblador>

RAE. (2021). Palindromo significado. Diccionario rae.

<https://dle.rae.es/pal%C3%ADndromo>

ANEXO:

IC3101. Arquitectura de Computadoras - II Semestre 2021

Laboratorio: #

Rúbrica para evaluar a los compañeros de mi grupo de trabajo (Grupo de 3 integrantes)

Estudiante evaluado:	Nombre Apellido1 Apellido2									
Trabajo Personal	Rúbrica									
	Siempre (1 punto)			A veces (0,5 puntos)			Nunca (0 puntos)			Puntos Obtenidos
	E1	E2	Autoevaluación	E1	E2	Autoevaluación	E1	E2	Autoevaluación	
Es responsable con la parte del trabajo asignada.										
Participa de las reuniones virtuales coordinadas por el grupo.										
Es respetuoso(a) con los miembros del grupo.										
Contribuye con la solución de las claves de los programas binarios.										
Contribuye en la elaboración del documento del proyecto.										
TOTAL:										

Cálculo del % (De un máximo de 10% por estudiante):

Fórmula: Puntos Obtenidos / 0,15 x 10% =

IC3101. Arquitectura de Computadoras - II Semestre 2021

Laboratorio: #

Rúbrica para evaluar a los compañeros de mi grupo de trabajo (Grupo de 3 integrantes)

Estudiante evaluado:	Nombre Apellido1 Apellido2									
Trabajo Personal	Rúbrica									
	Siempre (1 punto)			A veces (0,5 puntos)			Nunca (0 puntos)			Puntos Obtenidos
	E1	E2	Autoevaluación	E1	E2	Autoevaluación	E1	E2	Autoevaluación	
Es responsable con la parte del trabajo asignada.										
Participa de las reuniones virtuales coordinadas por el grupo.										
Es respetuoso(a) con los miembros del grupo.										
Contribuye con la solución de las claves de los programas binarios.										
Contribuye en la elaboración del documento del proyecto.										
TOTAL:										

Cálculo del % (De un máximo de 10% por estudiante):

Fórmula: Puntos Obtenidos / 0,15 x 10% =

Estudiante evaluado:	Nombre Apellido1 Apellido2									
Trabajo Personal	Rúbrica									
	Siempre (1 punto)			A veces (0,5 puntos)			Nunca (0 puntos)			Puntos Obtenidos
	E1	E2	Autoevaluación	E1	E2	Autoevaluación	E1	E2	Autoevaluación	
Es responsable con la parte del trabajo asignada.										
Participa de las reuniones virtuales coordinadas por el grupo.										
Es respetuoso(a) con los miembros del grupo.										
Contribuye con la solución de las claves de los programas binarios.										
Contribuye en la elaboración del documento del proyecto.										
TOTAL:										

Cálculo del % (De un máximo de 10% por estudiante):

Fórmula: Puntos Obtenidos / 0,15 x 10% =