

Organización de Computadoras 66.20

Trabajo Práctico 1

Autor	Padron	Correo electrónico
Flórez Del Carpio, Christian	91011	chris.florez.d.c@gmail.com
Montenegro, Josefina	94289	mariajosefina.mont@gmail.com
Quino López, Julián	94224	julianquino2@gmail.com



Facultad de Ingeniería
Universidad de Buenos Aires
Av. Paseo Colón 850 - C1063ACV
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel: +54 (11) 4343-0893 / 4343-0092
<http://www.fi.uba.ar>

Historial de Revisiones

Fecha	Revisor	Detalle
10/10/2017	-	Entrega primera versión del TP

Resumen

El siguiente trabajo práctico tiene como objetivo familiarizarse con el conjunto de instrucciones MIPS y el concepto de ABI, para lograr dicho propósito se debe implementar la lógica de detección de palíndromos en assembly, entendiendo como palabras a aquellas compuestas por letras [A-Z], números [0-9], guiones bajos y medios, es decir, cualquier combinación posible de los anteriormente mencionados, tal como estaba enunciado en el TP0.

1. Introducción

Pueden haber tres escenarios posibles, el caso en el cual el usuario ingresa archivo de entrada y salida, el caso en el que se ingresa un archivo de entrada solamente y por último el caso donde se recibe el archivo de salida. En caso de no proporcionar un archivo de texto como entrada, se requerirá ingresar el stream por entrada standard. Si no se especifica un archivo de salida, se mostrarán los resultados por salida standard. Esto es igual a lo explicando en el TP0.

2. Desarrollo

Se desarrolló un programa C, desde el cual se invoca a la función palindrome escrita en assembly. Esta función recibe como parámetros los archivos de entrada/salida (si no se hubiesen proporcionado tales archivos se toman los streams de entrada/salida standard) y las cantidades ibytes y obytes, las cuales describen la unidad de transferencia para escribir en el buffer de entrada/salida, respectivamente. Cuando no son proporcionados estos valores, se toma por defecto el valor 1.

2.1. Comandos para compilar y ejecutar el programa

Se puede compilar el programa con el siguiente comando:

```
$ gcc isPalindrome.c -o tp0
```

Y luego ejecutarlo con el comando:

```
$ ./tp0 -i input.txt -o output.txt -ibuf-bytes numero1 -obuf-bytes numero2
```

En caso de sólo querer especificar el archivo de entrada, debe ejecutarse, por ejemplo, de la siguiente manera:

```
$ ./tp0 -i input.txt -o -
```

Análogamente si se quiere ingresar un archivo de salida:

```
$ ./tp0 -i - -o output.txt
```

Es decir que con un guión medio indicamos que no se proporcionará un archivo para entrada/-salida, acorde a lo que indica el enunciado.

2.2. Otros comandos

Pueden utilizarse comandos tales como help y version, de la siguiente forma:

```
$ ./tp0 -h
```

```
$ ./tp0 -V
```

2.3. Código fuente

3. Casos de prueba

A continuación se muestran unos casos de prueba desde la consola del GXEmul, los textos utilizados se detallarán al final.

```
chris@chris-530U3C-530U4C: ~  
chris@chris-530U3C-530U4C: ~ 80x24  
root@:~/chris/tp02q# cat prueba1.txt  
Somos los primeros en completar el TP 0.  
  
Ojo que La fecha de entrega del TP0 es el martes 12 de septiembre.  
root@:~/chris/tp02q# ./tp0 -i prueba1.txt -o salida.txt  
Se recibió el archivo prueba1.txt  
Se recibió el archivo salida.txt  
Se procesó el archivo de entrada  
root@:~/chris/tp02q# cat salida.txt  
Somos  
0  
Ojo  
root@:~/chris/tp02q#
```

Figura 1: Prueba utilizando archivo de entrada y salida.

```
chris@chris-530U3C-530U4C: ~  
chris@chris-530U3C-530U4C: ~ 86x26  
root@:~/chris/tp02q# cat prueba1.txt  
Somos los primeros en completar el TP 0.  
  
Ojo que La fecha de entrega del TP0 es el martes 12 de septiembre.  
root@:~/chris/tp02q# ./tp0 -i prueba1.txt  
Se recibió el archivo prueba1.txt  
Se mostrará el resultado en pantalla.  
Se procesó el archivo de entrada  
Las palabras palíndromas detectadas son:  
Somos  
0  
Ojo  
root@:~/chris/tp02q#
```

Figura 2: Prueba utilizando solamente archivo de entrada.

```
chris@chris-530U3C-530U4C: ~  
chris@chris-530U3C-530U4C: ~ 80x24  
root@:~/chris/tp02q# cat prueba2.txt  
M  
root@:~/chris/tp02q# ./tp0 -i prueba2.txt -o salida.txt  
Se recibió el archivo prueba2.txt  
Se recibió el archivo salida.txt  
Se procesó el archivo de entrada  
root@:~/chris/tp02q# cat salida.txt  
M  
root@:~/chris/tp02q#
```

Figura 3: Otra prueba utilizando otro archivo de entrada y salida.

```

chris@chris-530U3C-530U4C: ~
chris@chris-530U3C-530U4C: ~ 86x26
root@:/chris/tp02q# cat prueba3.txt
Reconocer que 345 soldados6civiles murieron por una fruta como el anana en san luis m
e resulta extranio, ya que aca en neuquen sobran de a montones,..., pero bueno es solo
un comentario que m hizo ana el otro dia cuando me picaba el ojo, [{111341115;: ese n
umero no lo entiendo sera palindromo? no lo se.

salas es un jugador chileno? creo que sis :{11122.
root@:/chris/tp02q# ./tp0 -i prueba3.txt -o salida.txt
Se recibió el archivo prueba3.txt
Se recibió el archivo salida.txt
Se procesó el archivo de entrada
root@:/chris/tp02q# cat salida.txt
Reconocer
anana
aca
neuquen
a
m
ana
ojo
ese
salas
sis
11
22
root@:/chris/tp02q#

```

Figura 4: Prueba utilizando otro archivo de entrada y salida.

```

chris@chris-530U3C-530U4C: ~
chris@chris-530U3C-530U4C: ~ 86x26
root@:/chris/tp02q# cat prueba3.txt
Reconocer que 345 soldados6civiles murieron por una fruta como el anana en san luis m
e resulta extranio, ya que aca en neuquen sobran de a montones,..., pero bueno es solo
un comentario que m hizo ana el otro dia cuando me picaba el ojo, [{111341115;: ese n
umero no lo entiendo sera palindromo? no lo se.

salas es un jugador chileno? creo que sis :{11122.
root@:/chris/tp02q# ./tp0 -i prueba3.txt
Se recibió el archivo prueba3.txt
Se mostrará el resultado en pantalla.
Se procesó el archivo de entrada
Las palabras palindromas detectadas son:
Reconocer
anana
aca
neuquen
a
m
ana
ojo
ese
salas
sis
11
22
root@:/chris/tp02q#

```

Figura 5: Prueba utilizando otro archivo de entrada.

```

chris@chris-530U3C-530U4C: ~
chris@chris-530U3C-530U4C: ~ 86x26
root@:/chris/tp02q# ./tp0 -o salida.txt
Se recibió el archivo salida.txt
Ingrese el stream a procesar (máximo 300 caracteres):
Reconocer que 345 soldados6civiles murieron por una fruta como el anana en san luis m
e resulta extranio, ya que aca en neuquen sobran de a montones,..., pero bueno es solo
un comentario que m hizo ana el otro dia cuando me picaba el ojo, [{111341115;: ese n
umero no lo entiendo sera palindromo? no lo se.
Se procesó el archivo de entrada
root@:/chris/tp02q# cat salida.txt
Reconocer
anana
aca
neuquen
a
m
ana
ojo
ese
root@:/chris/tp02q#

```

Figura 6: Prueba utilizando solamente archivo de salida.

3.1. Textos utilizados

Prueba 1: Somos los primeros en completar el TP 0.

Ojo que La fecha de entrega del TP0 es el martes 12 de septiembre.

Prueba 2: M

Prueba 3: Reconocer que 345 soladadosciviles murieron por una fruta como el anana en san luis me resulta extranio, ya que aca en neuquen sobran de a montones,..., pero bueno es solo un comen-

tario que m hizo ana el otro dia cuando me picaba el ojo, [11134111;: *esenumeronoloentiendoserapalindromo?nolose.* salas es un jugador chileno? creo que sis :[11]22.

4. Código MIPS generado

4.1. Código fuente Assembly

5. Conclusiones

El trabajo práctico nos resultó interesante, no por el programa a desarrollar en sí, sino por lo que representó trabajar con el emulador GXEmul, emular la arquitectura MIPS, crear el túnel de comunicación entre el host OS (Linux, distribución Ubuntu) y el guest OS (NetBSD). Aprendimos como transferir archivos entre ambos sistemas y también ciertas cuestiones del lenguaje C con el cual no estábamos toalmente familiarizados.

Referencias

[1] GetOpt library, [https://www.gnu.org/software/libc/manual/html_node/Example – of – Getopt.html](https://www.gnu.org/software/libc/manual/html_node/Example%20of%20Getopt.html).

[2] StackOverflow, <https://www.stackoverflow.com>.