

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Bachillerato en Ingeniería en Computación



IC-3101 Arquitectura de Computadoras

Profesor: Ing. Eduardo A. Canessa Montero, M.Sc.

Tarea Programada #1

Lectura, modificación y conteo de palabras de archivo en texto plano.

Integrantes:

Sara Castro Sáenz 2014085332

Ericka Céspedes Moya 2017239557

José David Martínez Garay 2018160104

Fecha de Entrega: 2 de noviembre.

II Semestre, 2018

I. Descripción de diseño y arquitectura.

El algoritmo está implementado en una arquitectura de x64, este se compone de 3 módulos principales; la lectura de archivo, conversor de mayúsculas a minúsculas (y viceversa) y la recopilación de información acerca del proceso de conversión (palabras transformadas, letras cambiadas, porcentaje que no requirió conversión).

Se ha elegido la utilización de la arquitectura x64 por su facilidad para realizar llamadas del sistema, métodos de retorno y la utilización de 64 bytes para cada registro de propósito general y específico.

II. Descripción técnica.

El algoritmo consta de tres módulos principales:

1. *Lectura del archivo*: Solicita al usuario en nombre del archivo y lo abre. El contenido de este, es trasladado a una reservación de memoria, donde se va a realizar el recorrido de este. Una vez almacenado el contenido del archivo de texto, es mostrado al usuario
2. *Conversor de minúsculas a mayúsculas (y viceversa)*: Una vez mostrado el texto en pantalla, se le solicita al usuario realizar alguna de las tres acciones propuestas; *Lower to upper*, *Upper to lower*, *Exit*. Una vez que el usuario seleccionó una opción a convenir, se muestra el texto con la modificación elegida.
3. *Resultados*: Por último, el programa imprime en pantalla; las palabras del documento, cantidad total de letras del documento, cantidad de letras que fueron convertidas de mayúscula a minúscula (o viceversa) y el porcentaje del texto que no requirió modificación.

Se requirió la implementación de macros para imprimir en consola diversos datos (menús, contenido del documento de texto, etc), y también para realizar seteos de registros. La utilización de la recursividad se dio a la hora de realizar la conversión (mayúscula-minúscula y viceversa) ya que se necesita tomar cada byte del documento para realizar comparaciones y de esta forma proceder a la conversión. A lo largo del algoritmo, existen

diversos procedimientos para la realización de tareas específicas, por ejemplo el conversor de número a lista de caracteres, impresión del menú en consola.

IV. Tópicos de investigación

Se realizó la investigación para la apertura de archivos de texto en ensamblador *NASM*, las diversas llamadas del sistema y el desarrollo de macros, tanto en arquitectura x86 como en x64. Por motivos de la utilización de arquitectura x64 para el desarrollo del código, la investigación se basó en las llamadas y apertura de datos para dicho tipo de arquitectura.

De estos tres puntos, se pueden recalcar diversos aspectos como:

1. Realizar llamadas al sistema cuando sea necesario, no siempre.
2. Las llamadas del sistema son necesarias para realizar cambios en el programa como lo es la escritura en consola o la lectura de ella. Estos cambios pueden ser suministrados por recursos del sistema operativo. Estos se efectúan en el *Kernel Services Call Gate*.
3. Los macros facilitan la modularidad del código y la reutilización de este, ya que solo se necesita enviar por parámetro los valores a utilizar, estos van a ser reemplazados por las constantes determinadas por el usuario.
4. La apertura de archivos en *NASM* es relativamente sencilla, pero resulta compleja la impresión de este en pantalla, es por ello que la mayoría de programadores optan por la utilización de librerías externas para realizarlo.

V. Manual del Usuario

Primero aparece un mensaje de bienvenida que pide el nombre del archivo a abrir.

```
Bienvenidos  
Escriba el nombre del archivo que desea abrir:
```

Figura 1. Mensaje de bienvenida

Luego, si el archivo es leído con éxito, se imprime el texto original del archivo y aparece el menú de opciones, sino, se muestra un error en la consola y termina el programa. Se selecciona la opción (1) para convertir todas las letras del texto de minúsculas a mayúsculas, y la opción (2) para convertir todas las letras del texto de mayúsculas a minúsculas.

```
Bienvenidos  
Escriba el nombre del archivo que desea abrir:  
curie  
  
Un científico en su laboratorio no es sólo un técnico: es también un niño colocado ante fenómenos naturales que le impresionan como un cuento de hadas.  
  
- Madam Curie  
  
==== MENU ====  
1. Lower to upper  
2. Upper to lower  
3. Exit
```

Figura 2. Menú

Una vez dada la instrucción de opción 1 o 2 se imprime el texto y el análisis del contenido. En el caso de la opción 1 se desplegarán todos los caracteres del texto en mayúscula, para la opción 2 serán en minúscula. Cabe destacar que los caracteres especiales como acentos y ñs no cambian. Además, aparecen los resultados del conteo de palabras, de las letras cambiadas de minúscula a mayúscula, del total de letras del texto, y del porcentaje de texto que no requirió modificación.

```
==== MENU ====
1. Lower to upper
2. Upper to lower
3. Exit
1

UN CIENTÍFICO EN SU LABORATORIO NO ES SÓLO UN TÉCNICO: ES TAMBIÉN UN NIÑO COLOCA
DO ANTE FENÓMENOS NATURALES QUE LE IMPRESIONAN COMO UN CUENTO DE HADAS.

- MADAM CURIE

==== STATISTICS ====
Word count: 28
Letter case changed: 125
Letter count: 128
Percentage changed: 97
```

Figura 3. Opción 1 seleccionada

```
==== MENU ====
1. Lower to upper
2. Upper to lower
3. Exit
2

un científico en su laboratorio no es sólo un técnico: es también un niño coloca
do ante fenómenos naturales que le impresionan como un cuento de hadas.

- madam curie

==== STATISTICS ====
Word count: 28
Letter case changed: 3
Letter count: 128
Percentage changed: 2
```

Figura 4. Opción 2 seleccionada

La opción (3) sirve para salir del programa sin usar el menú de usuario.

```
Bienvenidos
Escriba el nombre del archivo que desea abrir:
curie

Un científico en su laboratorio no es sólo un técnico: es también un niño colocado ante fenómenos naturales que le impresionan como un cuento de hadas.

- Madam Curie

==== MENU ====
1. Lower to upper
2. Upper to lower
3. Exit
3
```

Figura 5. Salir del menú de usuario

Anexo pseudocódigo:

<pre> case_changed: total amount of letters converted letter_count: total amount of letters is_word: 0 if word hasn't started, 1 otherwise word_count: total amount of words // changes from lower to upper and vice versa macro change_case 1 // 1 = char xor %1, 0x20 letter_count++ case_changed++ endmacro // sets values for letter case change macro set_values 2 // 1 = min_char, 2 = max_char reg1 = %1 reg2 = %2 endmacro _start: read file print text call menu print new_text call print_statistics exit syscall menu: print options loop: if option == upper to lower // 1. Upper to lower set_values 'A', 'Z' call parse_text if option == lower to upper // 2. Lower to upper set values 'a', 'z' call parse_text if option != 1 or 2 jump loop </pre>	<pre> parse_text: letter_count = 0 counter = 0 is_word = 0 word_count = 0 get_char: char = [text + counter] if end of text jump done if char >= reg1 && char <= reg2 change_case char if is_word == 0 is_word = 1 else if is_word == 1 word_count++ is_word = 0 [new_text + counter] = char counter++ jump get_char done: print_statistics: percentage = (case_changed * 100) / letter_count print "Words: " + word_count print "Letter: " + letter_count print "Letters changed: " + case_changed print "{ercentage: " + percentage </pre>
--	--