

Instituto Tecnológico de Costa Rica

Curso: IC-3101 Arquitectura de Computadores

I Semestre 2015

Tarea Programada #3:

Máquina Enigma

Profesor

Erick Hernández Bonilla

Estudiantes:

* Melissa Molina Corrales 2013006074
* Liza Chaves Carranza 2013016573
* Gabriel Pizarro Picado 201216833
* Fabián Monge García 2014088148

Índice

Propósito y Descripción del proyecto….………………………………………………………………….3

Funcionamiento del programa………………………………….……………………………………………5

Leer y abrir archivos…………..………………………………………………………………………………….6

Carga de rotores seleccionados…………..…………………………………………………………………8

Desplazamientos de los rotores…………………………………………………………………………….11

Bibliografía…………………………………………………………………………………………………………….16

Propósito y Descripción del proyecto

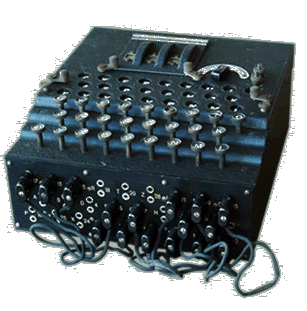


Figura 1. Máquina Enigma.

Enigma era el nombre de una máquina que disponía de un mecanismo de cifrado rotatorio, que permitía usarla tanto para [cifrar](http://es.wikipedia.org/wiki/Criptograf%C3%ADa) como para descifrar mensajes.

Su fama se debe a haber sido adoptada por las fuerzas militares de [Alemania](http://es.wikipedia.org/wiki/Alemania) desde [1930](http://es.wikipedia.org/wiki/1930). Su facilidad de manejo y supuesta inviolabilidad fueron las principales razones para su amplio uso. Su sistema de cifrado fue finalmente descubierto y la lectura de la información que contenían los mensajes supuestamente protegidos es considerada, a veces, como la causa de haber podido concluir la [Segunda Guerra Mundial](http://es.wikipedia.org/wiki/Segunda_Guerra_Mundial).

El objetivo de este proyecto es implementar una máquina enigma, esta tendrá el mismo funcionamiento que en la Segunda Guerra Mundial.

La tarea será programada en YASM (Ensamblador) en la sintaxis de Intel de x64.

El funcionamiento será el siguiente:

La máquina estará compuesta por un plugboard, un reflector y 5 rotores con configuraciones distintas de los cuales se escogerán 3 para llevar a cabo el funcionamiento.

El plugboard intercambiará las letras ingresadas, así por ejemplo cuando se ingresa una X esta será cambiada por una F según la configuración del plugboard. Su configuración será la siguiente:

**XF, PZ, SQ, GR, AJ, UO, CN, BV, TM, KI**

Después la letra pasa por tres rotores que intercambian las letras y luego el reflector que intercambia la letra una vez más y regresa a través de los tres rotores, por el plugboard y finalmente se muestra la letra encriptada por la máquina. Su configuración será de la siguiente manera:

**JPGVOUMFYQBENHZRDKASXLICTW**

Funcionamiento de los rotores

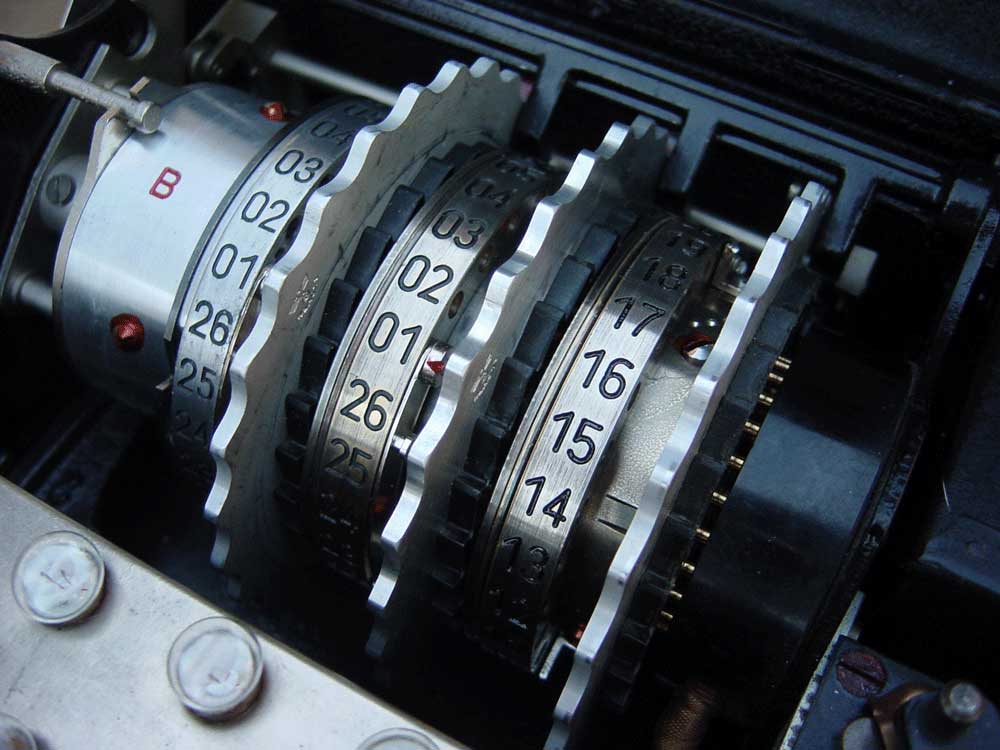


Figura 2. Rotores de una máquina enigma.

Cada vez que se desea encriptar una letra el rotor de la derecha avanza una posición, cuando este de una vuelta (lo cual corresponde a recorrer las 26 letras del alfabeto), avanza el rotor del medio y cuando este complete una vuelta, avanza el rotor de la izquierda de esta manera se van intercambiando las letras.

Estos rotores contarán con una posición inicial y su configuración será la siguiente:

|  |  |
| --- | --- |
| Rotor 1 | EKMFLGDQVZNTOWYHXUSPAIBRCJ |
| Rotor 2 | **AJDKSIRUXBLHWTMCQGZNPYFVOE** |
| Rotor 3 | **BDFHJLCPRTXVZNYEIWGAKMUSQO** |
| Rotor 4 | **ESOVPZJAYQUIRHXLNFTGKDCMWB** |
| Rotor 5 | **VZBRGITYUPSDNHLXAWMJQOFECK** |

Funcionamiento del programa

Para el funcionamiento de la máquina enigma se cargarán las configuraciones en 2 archivos, el primer archivo tendrá los rotores a utilizar, las posiciones iniciales de cada uno de ellos y la configuración del plugboard y en el segundo archivo se pasará el mensaje que se desea encriptar o desencriptar.

Ejemplo del formato del archivo con las configuraciones:

**IV, V, I = Rotores**

**23, 09, 20 = Posiciones iniciales de cada rotor**

**XF, PZ, SQ, GR, AJ, UO, CN, BV, TM, KI = Configuración del Plugboard**

Luego de cargar los archivos, el programa procederá a encriptar el mensaje que se haya ingresado en el segundo archivo y mostrará en pantalla una animación de todo el proceso de encriptación, y finalmente mostrará la palabra encriptada.

Leer y abrir archivos

Una parte importante del programa es abrir y leer los archivos con las configuraciones y el mensaje a encriptar para llevar a cabo está funcionalidad se implementó un programa que se encarga de leer los archivos que se le pasen y cuenta con varios procedimientos encargados de abrir y leer los archivos, quitar los espacios en blanco y las comas del archivo de configuraciones para efectos de facilitar la implementación de la máquina.

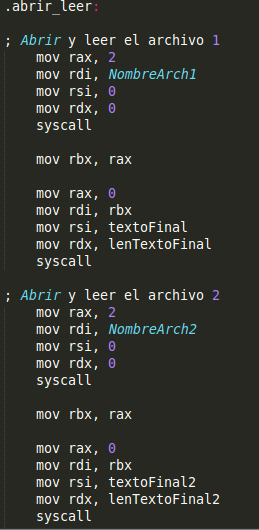


Figura 3. Procedimiento para abrir y leer un archivo

**Archivo de configuraciones**

**Archivo con el mensaje a encriptar.**

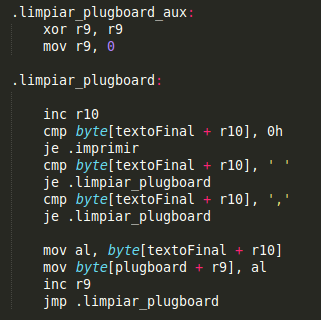
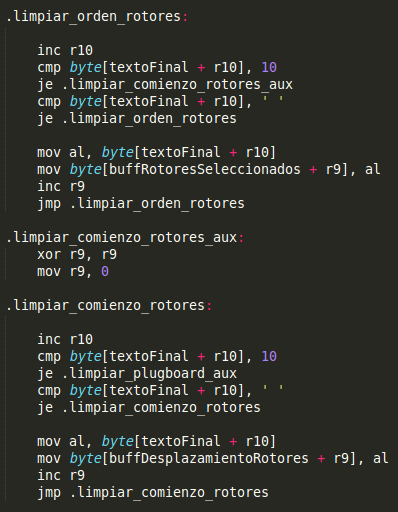


Figura 4. Procedimientos para quitar los espacios en blanco y las comas de los datos del archivo de configuraciones.

Carga de rotores seleccionados

Para el funcionamiento de la máquina se necesita escoger 3 de los 5 rotores, estos 3 rotores se le pasan en el archivo de configuraciones, una vez se conozcan los rotores a usar, se necesita saber sus configuraciones y sus posiciones iniciales para esto se crean procedimientos que cargan los rotores seleccionados en diferentes buffers.

Se utilizarán buffers para guardar los 3 rotores seleccionados y variables para guardar las configuraciones de los rotores y la configuración del reflector.

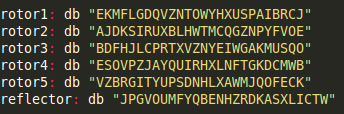


Figura 6. Variables que guardan las configuraciones de cada uno de los rotores y la configuración del reflector.

Figura 5. Buffers en donde se guardaran los rotores seleccionados.



Como se mencionó anteriormente se crea un procedimiento que carga los rotores en diferentes buffers para ser usados más adelante para poder encriptar el mensaje.

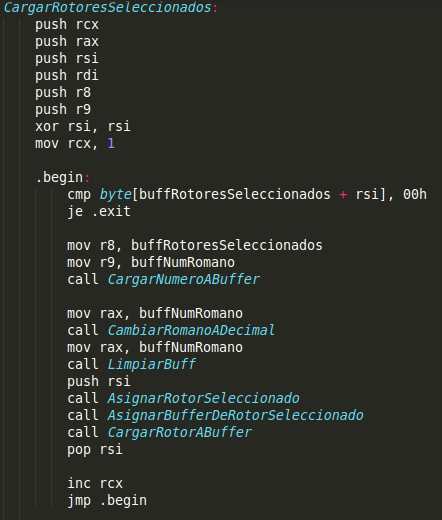


Figura 7. Procedimiento principal para cargar los rotores.

Como complemento a este procedimiento se utiliza un procedimiento que cambia los números romanos con los que vienen identificados los rotores en el archivo de configuraciones a números decimales esto con el fin de facilitar la carga de estos a los diferentes buffers.

Una vez se haya pasado los números romanos de los rotores a números en decimal se utilizan procedimientos para asignarle las configuraciones de los rotores seleccionados a cada uno de los buffers mostrados anteriormente.

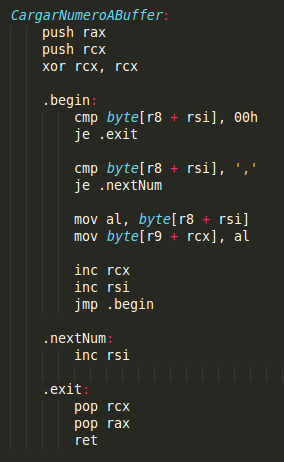


Figura 8. Procedimiento que carga los números romanos de los rotores al buffer de rotores seleccionados.

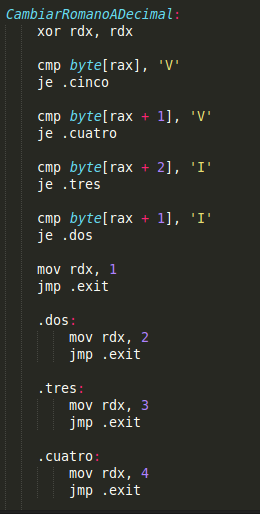


Figura 9. Procedimiento que cambia los números romanos de los rotores a números decimales.

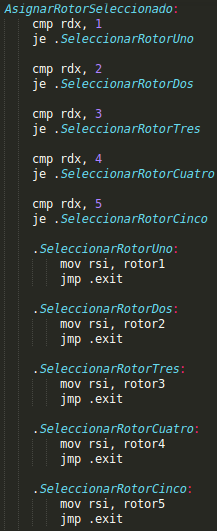


Figura 10. Procedimiento que asigna al rsi el rotor que se debe pasar.

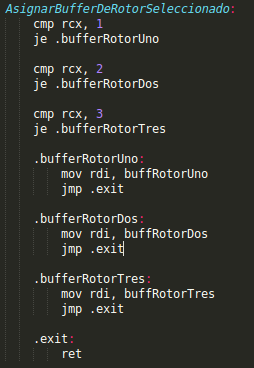


Figura 11. Procedimiento que asigna al rdi el buffer para guardar el rotor asignado.

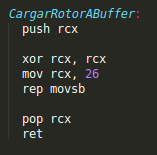


Figura 12. Procedimiento que carga un rotor a un buffer.

**Finalmente se utiliza un procedimiento que se encarga de cargar los rotores a los buffers, copia lo que tenga el rsi (Las configuraciones de los rotores seleccionados) en el rdi (buffers para los rotores) 26 veces que es el tamaño de cada uno de los rotores.**

Desplazamientos de los rotores

Para poder encriptar el mensaje que se le pase a la máquina se necesita que los rotores hagan rotaciones para intercambiar las letras cada vez que se quiera encriptar una letra, para esto se implementan algunos procedimientos que permitan mover de alguna manera la posición de los rotores para que estos vayan cambiando las letras.

Este procedimiento va a recorrer el buffer en donde están las posiciones iniciales de los rotores y dependiendo del número que haya en el registro rbx ese será el rotor que hay que rotar y seleccionará el rotor correspondiente , guardado en cada uno de los diferentes buffers y lo rotará la cantidad de veces que haya en el registro rcx.

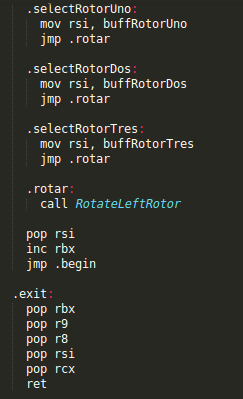
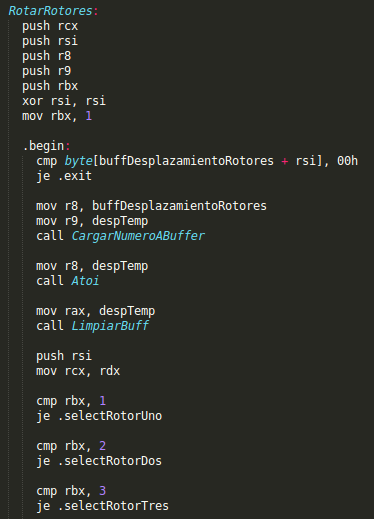


Figura 13. Procedimiento que desplaza los rotores n veces.

Como parte de este procedimiento para poder realizar las rotaciones a los rotores, se implementa un procedimiento para hacer las rotaciones correspondientes, importante mencionar que a los rotores se les aplicará una rotación a la izquierda.

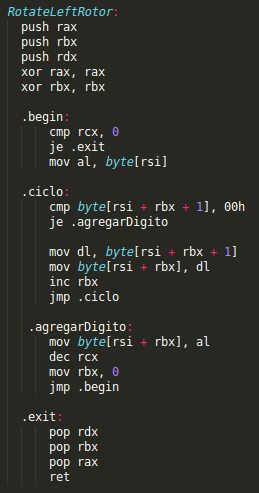


Figura 14. Procedimiento que aplica una rotación a la izquierda a un buffer.

Para convertir el desplazamiento de string a integer se usará el procedimiento Atoi.

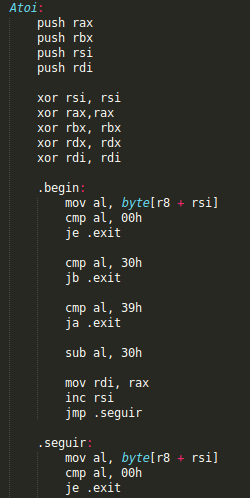
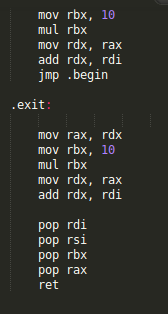


Figura 15. Procedimiento ATOI convierte de string a integer.

# Bibliografía

Dade, L. (2006-2015). *Enigma Emulator*. Obtenido de http://enigma.louisedade.co.uk/index.html

*Enigma rotor details*. (10 de Mayo de 2015). Obtenido de Wikipedia: http://en.wikipedia.org/wiki/Enigma\_rotor\_details

Rijmenants, D. (14 de Febrero de 2015). *Cipher Machines and Cryptology*. Obtenido de http://users.telenet.be/d.rijmenants/index.htm

Simon, P. R. (31 de mayo de 2014). *Crypto Museum*. Obtenido de http://www.cryptomuseum.com/crypto/enigma/working.htm

UCDavis. (1 de Febrero de 2010). *The Enigma Code*. Obtenido de https://www.youtube.com/watch?v=ncL2Fl6prH8