

Organización del Procesador - 2015

Versión: Lenguaje Assembler ATM 328

Título: “Encontrar codificación”

El proyecto “Encontrar codificación” consiste en implementar una versión de un algoritmo básico para encontrar una clave de codificación de mensajes en lenguaje assembler para microcontroladores tipo ATM 328 (Arduino). El mismo consiste en considerar como entrada una cadena de byte (caracteres) de datos (en adelante cadena, que es un arreglo de byte de X posiciones) y una clave de 4 bits. Esta clave de 4 bits deber ser tomada de los pines 8 a 11 de arduino (pines 0 a 3 puerto B).

La cadena de entrada se encuentra codificada de la siguiente manera:

Cada carácter de la cadena original esta rotado N veces hacia la izquierda o la derecha según bit más significativo de la clave, esto es 0: izquierda, 1: derecha. La cantidad N de rotaciones está dado por los restantes 3 bit de la clave.

El algoritmo a implementar debe encontrar en la cadena codificada, comenzando en cualquier posición, la subcadena “hola”, para lograr esto se debe ir “probando” por fuerza bruta con diferentes combinaciones de la clave (de 4 bits) de entrada hasta encontrar en alguno de los intentos la subcadena “hola” en la cadena codificada. Cuando el algoritmo encuentre la clave correcta (encontró la subcadena hola) debe prender un led por medio del pin 2 de arduino (pin 2 puerto C)

Ejemplo

Cadena Original: sin codificar “dbholaj0” donde 0(cero) indica el fin de la cadena.

Donde el código ascii de ‘a’ es 97, de b es 98, etc.

Clave de encriptación: 9 (1001b)

Consideraciones:

El arreglo de bytes(char) en memoria que representa la cadena Original codificada está apuntado por *cadeChar* y tendrá almacenado lo siguiente:

Representación de la cadena en memoria:

Con representación codificada
Con la clave dada

MEMORIA	
Dirección	Contenido
<i>cadeChar</i>	00110010
<i>cadeChar+1</i>	00110001
<i>cadeChar+2</i>	00110100
<i>cadeChar+3</i>	10110111
<i>cadeChar+4</i>	00110110...
<i>cadeChar+5</i>	10110000...
<i>cadeChar+6</i>	00110101...
<i>cadeChar+7</i>	00011110...

Representación Real interna
Decodificada con la clave dada

MEMORIA	
Dirección	Contenido
<i>cadeChar</i>	01100100
<i>cadeChar+1</i>	01100010
<i>cadeChar+2</i>	01101000
<i>cadeChar+3</i>	01101111
<i>cadeChar+4</i>	01101100.
<i>cadeChar+5</i>	01100001
<i>cadeChar+6</i>	01101010...
<i>cadeChar+7</i>	00011110...

Representación en ASCII de la cadena decodificada

MEMORIA	
Dirección	Contenido
<i>cadeChar</i>	d
<i>cadeChar+1</i>	b
<i>cadeChar+2</i>	h
<i>cadeChar+3</i>	o
<i>cadeChar+4</i>	l
<i>cadeChar+5</i>	a
<i>cadeChar+6</i>	j
<i>cadeChar+7</i>	0

Requisitos

- El proyecto deberá ser realizado en grupos de 3 alumnos.
- El código deberá estar debidamente legible y comentado.
- El programa deberá ser implementado en el lenguaje assembler para procesador ATM328p (Arduino), avr-gcc.
- La decodificación, debe ser subrutinas assembler que va a ser invocadas desde un programa C provisto por la cátedra y pueden descargarse desde el Aula Virtual
- Deberá confeccionarse un informe en donde consten: datos personales de los autores, y además se expliquen detalladamente el procesamiento

realizado, así como también, variables importantes, subrutinas implementadas, etc; todo aquello que considere que facilitará su lectura y corrección.

- La utilización de subrutinas es obligatoria y el pasaje de parámetros se debe realizar a través de la pila.
- Los proyectos serán evaluados por el cuerpo docente, calificando creatividad, eficiencia, prolijidad y resultados.
- En el caso de detectarse copias entre proyectos, los integrantes de cada grupo serán sancionados.
- Fechas:

Primera entrega (Decodificador) 16 /11/15.