



UNIVERSIDAD AUTONOMA DE SAN LUIS POTOSÍ

FACULTAD DE INGENIERÍA

ÁREA DE CIENCIAS DE LA COMPUTACIÓN

“LABORATORIO PROGRAMACIÓN DE SISTEMAS”

PRACTICA 5:

“Implementación del paso 2, SIC Estándar”

PROFESOR:

ING. AGUSTÍN HERNÁNDEZ GARCÍA

ALUMNO: Braulio Alejandro García Rivera

CLAVE ÚNICA: 239196

FECHA: 19/10/2020

SEMESTRE: 2020-2021/I

1. INTRODUCCIÓN

En este documento se explica cómo se implementó el paso 2 de la SIC Estándar continuando las practicas anteriores, se explican los procesos más importantes, así como problemas encontrados y su respectiva solución.

2. INSTALACIÓN Y CONFIGURACIÓN

En Visual Studio 2017 se instalaron las siguientes extensiones:

- Antlr4Code
- ANTLR Language Support

También se instalaron las bibliotecas de Antlr4 como paquetes NuGet:

- Antlr4 v4.6.6
- Antlr4.CodeGenerator v4.6.6
- Antlr4.Runtime v4.6.6

La configuración se realizó en base a la explicación del profesor y los ejemplos de la plataforma didacTIC.

3. IMPLEMENTACIÓN DEL PASO 2

- 1.- Ensamble de las instrucciones, se traducen los códigos de operación y se obtienen las direcciones de la tabla de símbolos.
- 2.- Las directivas START, RESW, RESB Y END no generan código objeto.
- 3.- La directiva BYTE si es de caracteres su código objeto es el código ASCII de cada carácter, pero si es hexadecimal se tomas estos bytes y se transcriben en el código objeto.
- 4.- La directiva WORD su código objeto es el valor que la directiva tiene, si está en decimal se convierte a hexadecimal y se completa a 6 bytes.
- 5.- Si existen errores de sintaxis en el archivo intermedio no se genera código objeto.

4. ESTRUCTURA DE LA TABLA DE CÓDIGOS DE OPERACIÓN

Para armar los códigos de operación con su respectiva instrucción se utilizó un diccionario compuesto de dos string, y así tener enlazados estos dos para al momento de estar analizando el código objeto saber a qué tipo de instrucción se refiere y de esta manera establecer la tabla de códigos de operación.

5. PROCEDIMIENTO: ENSAMBLAR INSTRUCCIONES

Para ensamblar las instrucciones se procedió a primero tenerlas en un diccionario listas para el momento en que se inicie la traducción de código, después va identificando los valores de cada instrucción para iniciar con los dos primeros dígitos que componen el código de operación para después dependiendo de la tabla de símbolos ir concatenando estos datos y obtener el código objeto como resultado final, posteriormente se procede a generar los registros con los datos obtenidos.

6. PROCEDIMIENTO: ENSAMBLAR DIRECTIVAS BYTE Y WORD

La directiva BYTE si es de caracteres su código objeto es el código ASCII de cada carácter, pero si es hexadecimal se tomas estos bytes y se transcriben en el código objeto.

La directiva WORD su código objeto es el valor que la directiva tiene, si está en decimal se convierte a hexadecimal y se completa a 6 bytes.

7. PROCEDIMIENTO:

GENERAR REGISTROS -> ENCABEZADO, TEXTO Y FIN

Registro de encabezado:

Contiene el nombre, la dirección de inicio y la longitud del programa.

Número de columna	Contenido
1	H
2-7	Nombre del programa
8-13	Dirección de inicio del programa objeto (hexadecimal)
14-19	Longitud en bytes del programa objeto (hexadecimal)

Registro de texto:

Contiene las instrucciones traducidas (código de máquina), los datos del programa e indica la dirección donde serán cargadas en memoria.

Número de columna	Contenido
1	T
2-7	Dirección de inicio del código objeto en este registro (hexadecimal)
8-9	Longitud en bytes del código objeto en este registro (hexadecimal)
10-69	Código objeto, representado en hexadecimal (dos columnas por byte de código objeto)

Registro de fin:

Indica el fin del programa objeto y especifica la dirección de programa donde empezará la ejecución, esta dirección es indicada por la directiva END en el programa fuente, si no se especifica, se toma la dirección de la primera instrucción ejecutable.

Número de columna	Contenido
1	E
2-7	Dirección de la primera instrucción ejecutable del programa objeto (hexadecimal)

8. MANEJO DE LOS ARCHIVOS: ENTRADA Y SALIDA

1.- Ejecuta el programa y selecciona el botón ABRIR, busca el archivo suma.s y genera la lectura, después selecciona el botón ANALIZAR y genera el archivo intermedio, la tabla de símbolos y los errores encontrados, al final selecciona el botón PASO 2 y genera los registros del archivo obj.

Codigo Fuente

Linea	Etiqueta	Instruccion	Operando
1	COPY	START	1000h
2	FIRST	STL	RETADR
3	CLOOP	JSUB	RDREC
4		LDA	LENGTH
5		COMP	ZERO
6		JEQ	ENDFIL
7		JSUB	WRREC
8		J	CLOOP
9	ENDFIL	LDA	EOF
10		STA	BUFFER
11		BYTE	X'123
12		LDA	THREE
13		STA	LENGTH
14		JSUB	WRREC
15		LDL	RETADR
16		RSUB	
17	BYTE	BYTE	C'11'
18	EOF	BYTE	C'EOF'
19	THREE	WORD	3
20	ZERO	WORD	0
21	RETADR	RESW	1
22	LENGTH	RESW	1
23	HOLA	BYTE	X'123'
24	OTRO	BYTE	'ASD'
25	BUFFER	RESB	4096

Archivo Intermedio

Linea	CP	Etiqueta	Instruccion	Direccion	Error	Codigo Obj
1	1000	COPY	START	1000H	No	
2	1000	FIRST	STL	RETADR	No	
3	1003	CLOOP	JSUB	RDREC	No	
4	1006		LDA	LENGTH	No	
5	1009		COMP	ZERO	No	
6	100C		JEQ	ENDFIL	No	
7	100F		JSUB	WRREC	No	
8	1012		J	CLOOP	No	
9	1015	ENDFIL	LDA	EOF	No	
10	1018		STA	BUFFER	No	
11	101B		BYTE	X'123	Syntax	
12	101B		LDA	THREE	No	
13	101E		STA	LENGTH	No	

Simbolo

Simbolo	Direccion
FIRST	1000
CLOOP	1003
ENDFIL	1015
EOF	102A
THREE	102D
ZERO	1030
RETADR	1033
LENGTH	1036
BUFFER	1039
RDREC	2039
RLOOP	203F

Resultados

Archivo Obj

Tamaño del archivo: 107A

Errores

Linea11: Error de sintaxis el operando: "X'123" Esta mal escrito
Linea17: Error de sintaxis la etiqueta no puede ser la palabra reservada "BYTE"
Linea23: Error de sintaxis el operando: "X'123" Esta mal escrito
Linea24: Error de sintaxis el operando: "ASD" Esta mal escrito
Linea40: Error de sintaxis no debe haber lineas vacias

2.- Ejecuta el programa y selecciona el botón ABRIR, busca el archivo copy.s y genera la lectura, después selecciona el botón ANALIZAR y genera el archivo intermedio, la tabla de símbolos y los errores encontrados, al final selecciona el botón PASO 2 y genera los registros del archivo obj.

Form1

Codigo Fuente

Linea	Etiqueta	Instruccion	Operando
1	COPY	START	1000h
2	FIRST	STL	RETADR
3	CLOOP	JSUB	RDREC
4		LDA	LENGTH
5		COMP	ZERO
6		JEQ	ENDFIL
7		JSUB	WRREC
8		J	CLOOP
9	ENDFIL	LDA	EOF
10		STA	BUFFER
11		BYTE	X'123
12		LDA	THREE
13		STA	LENGTH
14		JSUB	WRREC
15		LDL	RETADR
16		RSUB	
17	BYTE	BYTE	C'11'
18	EOF	BYTE	C'EOF'
19	THREE	WORD	3
20	ZERO	WORD	0
21	RETADR	RESW	1
22	LENGTH	RESW	1
23	HOLA	BYTE	X'123'
24	OTRO	BYTE	'ASD'
25	BUFFER	RESR	4096

ArchivoIntermedio

Linea	CP	Etiqueta	Instruccion	Direccion	Error	Codigo Obj
1			COPY	START	1000h	No
2	1000	FIRST	STL	RETADR	No	141033
3	1003	CLOOP	JSUB	RDREC	No	482039
4	1006		LDA	LENGTH	No	001036
5	1009		COMP	ZERO	No	281030
6	100C		JEQ	ENDFIL	No	301015
7	100F		JSUB	WRREC	No	482061
8	1012		J	CLOOP	No	3C1003
9	1015	ENDFIL	LDA	EOF	No	00102A
10	1018		STA	BUFFER	No	0C1039
11	101B		BYTE	X'123	Syntax	Error: de Sintaxis
12	101B		LDA	THREE	No	00102D
13	101E		STA	LENGTH	No	0C1036

Simbolo

Simbolo	Direccion
FIRST	1000
CLOOP	1003
ENDFIL	1015
EOF	102A
THREE	102D
ZERO	1030
RETADR	1033
LENGTH	1036
BUFFER	1039
RDREC	2039
RLOOP	203F

Resultados

Archivo Obj

Tamaño del archivo: 107A

```

HCOPY 00100000107A
T0010001E1410334820390010362810303010154820613C100300102AD
C103900102D
T00101E150C10364820610810334C0000454F46000003000000
T00103900
T0020391E041030001030E0205D30203FD8205D281030302057549039
2C205E38203F
T0020571C1010364C0000F1001000041030E02079302064509039DC20
792C1036
T002073073820644C000005
E001000

```

Errores

```

Linea11: Error de sintaxis el operando: "X'123'" Esta mal escrito
Linea17: Error de sintaxis la etiqueta no puede ser la palabra reservada
"BYTE"
Linea23: Error de sintaxis el operando: "X'123'" Esta mal escrito
Linea24: Error de sintaxis el operando: "'ASD'" Esta mal escrito
Linea40: Error de sintaxis no debe haber lineas vacias

```

9. PROBLEMAS

Un problema fue para detectar cuando el código tenía directivas BYTE ya que se tiene que hacer una conversión entre código ASCII y hexadecimal para ir rellenando con los valores el código objeto, también al momento de la instrucción RSUB está más complejo ya que hay que detectar también si es indirecto o directo.

10. POSIBLES MEJORAS

Mejorar el procesamiento del código obj para poder hacerlo de manera más eficiente y menos revuelta ya que el proceso requiere de distintas validaciones para poder ser generado de manera satisfactoria, más sin embargo funciona correctamente.

11. CONCLUSIÓN

En esta práctica se implementó el paso 2 donde nos encontramos con un reto diferente al paso 1 ya que aquí después del procedimiento anterior ahora había que procesar y generar código obj y su despliegue para vista del usuario de manera correcta continuando así con el ensamblado de la SIC Estándar.