

# SEMINARIO DE SOLUCION DE PROBLEMAS DE TRADUCTORES DE LENGUAJES I - PRACTICA 3

## INTRODUCCION

Recordando.... El ensamblador de dos pasos puede ser muy eficiente para la administración de la memoria que ocupa durante su ejecución, pues el primero y segundo paso no tiene necesidad de estar cargados en memoria al mismo tiempo pues sus tareas respectivas en lo fundamental son distintas. Cuyas características son las siguientes:

- Dependientes de la máquina
- Independientes de la máquina

Son características dependientes del procesador, desde luego el conjunto de instrucciones, los modos de direccionamiento, la generación de código y sin faltar la interfaz con la plataforma en la cual corre el ensamblador. Un aspecto importante que no se nos debe pasar por alto, es que los ensambladores pueden tener una clasificación en virtud de su generación de código.

COP: código de operación, pues si bien el código 8605 significa para el HC12 cargar al Acumulador A un 5, para el HC08 de la misma familia de Motorola es otra cosa muy diferente (el código correspondiente para dicha acción: es 9605)

REGISTROS INTERNOS: como parte de la arquitectura de cada micro

MODO DE DIRECCIONAMIENTO: aunque existen "N" modos de direccionamientos, no todos son aplicables a los micros.

MNEMONICO: las instrucciones aunque en concepto podrían ser las mismas, se enuncian de diferente manera, es decir:

Las características independientes de la máquina son aquellas relacionadas más bien con el tipo de implementación del programa y sus estructuras de datos que con la máquina objeto.

DIRECTIVAS: También existen las llamadas DIRECTIVAS o PSEUDOINSTRUCCIONES, las cuales especifican acciones auxiliares que se llevan a cabo por el ensamblador.

PERIFERICOS: todo dispositivo que será controlado por el micro

ARCHIVOS AUXILIARES: \*.LST, \*.OBJ, \*.S19, \*.TABSIM, etc, dependiendo del algoritmo con que se trabaje

.... De igual manera podríamos seguir numerando características globales para los micros

DIRECTIVAS: Las directivas de ensamblador controlan acciones auxiliares que se realizan durante el proceso de ensamblado, tales como reservar posiciones de memoria, inicializar posiciones de memoria a un determinado valor, definir etiquetas, indicar el inicio y el final del programa, o definir macros. Las directivas no son traducibles a código máquina. Nota: las directivas NO tienen código máquina.

ORG	BSZ
END	FILL
START	DC.W
EQU	FCC
DC.B	FCB

### PRACTICA 3: IDENTIFICACION DE DIRECTIVAS-CONT-LOC-TABSIM

**MODIFICARA EL PROGRAMA DE LA PRACTICA ANTERIOR DE MANERA QUE VALIDE LAS SINTAXIS DE 10 DIRECTIVAS, GENERE EL CONLOC, LST, TABSIM. (20 1ROS MNEMONICOS DEL HC12)**

Implementar el algoritmo del 1er paso del ensamblador empleando los archivos auxiliares.

Implementar un programa para separar las partes de una línea en lenguaje ensamblador previo análisis de la misma

#### CARACTERISTICAS DE LA PRACTICA

LA PRESENTE PRACTICA SERA VALIDA SI Y SOLO SI:

1. Mismas condiciones que P2.

#### PASOS:

1. Capture en el block de notas el siguiente archivo fuente a analizar, cuyo nombre será P3.ASM:

		CONTLOC	COP		
E3	EQU 50	0000		E3	EQU 50
	ORG \$4000	0000			ORG \$4000
E2	EQU 15	4000		E2	EQU 15
	ADCA #15	4000	89 ii		ADCA #15
	ADCA 15	4002	99 dd		ADCA 15
	START				START
	ADCA 150				ADCA 150
	ADCA 1500				ADCA 1500
	ADDD #10				ADDD #10
	ADDD 10				ADDD 10
	DC.B				DC.B
	DC.B 10				DC.B 10
	DC.B 10,11				DC.B 10,11
	BSZ 10				BSZ 10
	FILL 2,3				FILL 2,3
	DC.W				DC.W
	DC.W 1,2				DC.W 1,2
	FCC /HOLA/				FCC /HOLA/

	FCB 10					FCB 10
E1	END				E1	END

TABSIM:

E3	
E2	
E1	

## EJEMPLOS PRACTICOS

El ensamblador de dos pasos puede ser muy eficiente para la administración de la memoria que ocupa durante su ejecución, pues el primero y segundo paso no tiene necesidad de estar cargados en memoria al mismo tiempo pues sus tareas respectivas en lo fundamental son distintas. Cuyas características son las siguientes:

- **Dependientes de la máquina** Son características dependientes del procesador, desde luego el conjunto de instrucciones, los modos de direccionamiento, la generación de código y sin faltar la interfaz con la plataforma en la cual corre el ensamblador. Un aspecto importante que no se nos debe pasar por alto, es que los ensambladores pueden tener una clasificación en virtud de su generación de código.
- **Independientes de la máquina** Las características independientes de la máquina son aquellas relacionadas más bien con el tipo de implementación del programa y sus estructuras de datos que con la máquina objeto.
  - ☉ **DIRECTIVAS:** También existen las llamadas DIRECTIVAS o PSEUDOINSTRUCCIONES, las cuales especifican acciones auxiliares que se llevan a cabo por el ensamblador. Las directivas de ensamblador controlan acciones auxiliares que se realizan durante el proceso de ensamblado, tales como reservar posiciones de memoria, inicializar posiciones de memoria a un determinado valor, definir etiquetas, indicar el inicio y el final del programa, o definir macros. Las directivas no son traducibles a código máquina. Nota: las directivas NO tienen código máquina.
  - ☉ **PERIFERICOS:** todo dispositivo que será controlado por el micro
  - ☉ **ARCHIVOS AUXILIARES:** \*.LST, \*.OBJ, \*.S19, \*.TABSIM, etc, dependiendo del algoritmo con que se trabaje

Ejemplos de uso de directivas:

<b>ORG</b>	Especificar el punto en el cual se comienza a ensamblar una sección de código. Esta directiva debe utilizarse siempre antes de comenzar a declarar código o zonas de memoria.
<b>END</b>	Especifica el fin de un archivo en código fuente ensamblador. Esta directiva deber ser incluida obligatoriamente al final de cada archivo. Cualquier expresión declarada después de esta directiva será ignorada.
<b>START</b>	Especificación del comienzo de una sección de código ensamblado a dirección \$0.
<b>EQU</b>	Sustituye un identificador por una cadena de texto previamente especificada en cualquier parte del programa por el valor declarado. Esto permite una gran versatilidad a la hora de declarar constantes y posiciones de memoria de especial interés para el programador.

contloc	LINEA DE PROGRAMA	COP
	E1 EQU \$800	
	ORG \$4000	
4000	JMP E1	06 0800
4003	START	
0000	END	

<b>DC.B</b>	Las directivas DB y DW permiten declarar o reservar zonas de memoria, tanto para valores numéricos como para cadenas de texto. Estas directivas poseen una amplia sintaxis que permite una gran flexibilidad a la hora de definir áreas de memoria. Los elementos definidos mediante las directivas DB y DW deben ir separados por comas. Para reservar espacio en la memoria sin especificar un valor concreto, se puede emplear el signo de interrogación.
<b>DC.W</b>	La especificación de cadenas de texto debe hacerse mediante la expresión literal de la cadena entre comillas.

Contloc	LINEA DE PROGRAMA	COP
	ORG \$4000	
4000	DC.B	00
4001	DC.W	00 00
4003	DC.B 15	0F
4004	DC.B 20,21,22	14 15 16
4007	DC.B 'H	48
4008	DC.W 'H	0048
400 <sup>a</sup>	DC.B 'H, 'O, 'L, 'A	48 4F 4C 41
400E	END	

<b>BSZ ZMB</b>	( BLOCK STORAGE OF ZEROS ) La directriz BSZ ( y ZMB ) ocasionan que el ensamblador asigne un bloque de bytes y a cada byte se le asigna el valor inicial de cero.
<b>FCB</b>	( FORM CONSTANT BYTE ) Esta directriz ocasiona que el valor del operando sea almacenado en un solo byte del programa objeto.
<b>FCC</b>	( FORM CONSTANT CHARACTER STRING ) Esta directriz se utiliza para almacenar una cadena de caracteres ASCII en bytes consecutivos de memoria.
<b>FILL</b>	( FILL MEMORY ) Ocasiona que el ensamblador inicialice un área de memoria con un valor constante. La

primera expresión indica el valor constante y la segunda es el número de bytes a ser inicializados.

Contloc	A.ASM	COP
LINEA DE PROGRAMA		
	ORG \$4000	
	BSZ 3	
	ZMB 5	
	FCB 50,30,20	
	FCB 50	
	FCC /HOLA/	
	FCB 10	
	FILL 5,10	
	END	

contloc	Línea de programa	cop
4000	ORG \$4000	
4000 00 00 00	BSZ 3	
4003 00 00 00 00 00	ZMB 5	
4008 32 1e 14	FCB 50,30,20	
400b 32	FCB 50	
400c 48 4f 4c 41	FCC /HOLA/	
4010 0a	FCB 10	
4011 05 05 05 05 05 05	FILL 5,10	
05 05 05 05		
	END	