

# MACROPROCESADORES

Los ensambladores y compiladores cuentan con macroprocesadores. Estos permiten definir una abreviatura para representar una parte de un programa y utilizar esa abreviatura cuantas veces sea necesario, esto evita redundar código al programador, partes repetibles de programa.

Para utilizar una macro, primero hay que declararla. En la declaración se establece el nombre que se le dará a la macro y el conjunto de instrucciones que representará. Ejemplo:

```
MI MACRO
  JMP 0
  JMP 1
ENDM
```

La declaración se realiza una sola vez, pero la utilización o invocación a la macro (macrollamada) puede hacerse cuantas veces sea necesario. La utilización de macros posibilita la reducción del tamaño del código fuente, aunque el código objeto tiende a ser mayor que cuando se utilizan funciones.

Macroinstrucciones se les considera como una extensión de los lenguajes. De manera similar se considera al procesador de macroinstrucciones o macroprocesador como una extensión del ensamblador o compilador utilizado. El macroprocesador se encarga, en una primera pasada, de registrar todas las declaraciones de macros y de rastrear el programa fuente para detectar todas las macrollamadas. En cada lugar donde encuentre una macrollamada, el macroprocesador hará la sustitución por las instrucciones correspondientes. A este proceso de sustitución se le denomina expansión de la macro. El macroprocesador elabora dos tablas para el manejo de las macros:

Una tabla de macronombres que consiste de los nombres de las macros y un índice que le permite localizar la definición de la macro en otra tabla llamada tabla de macrodefiniciones.

Como su nombre lo indica, la tabla de macrodefiniciones contiene las definiciones de todas las macros a utilizar en el programa.

MACROINSTRUCCIÓN es una pseudo operación o un conjunto de instrucciones a la cual se le asigna un nombre para ser llamada y después utilizar su código. Estas permiten escribir una versión abreviada de un programa.

La tabla de nombres de macros es donde se almacenan las referencias de las macroinstrucciones, la de códigos es donde se almacenan los códigos de cada macro y la de argumentos los parámetros de cada macro.

Rutina vs macro una rutina es un procedimiento (en este caso un conjunto de código) que es usado cada vez que se le llame. Una rutina al ser llamada dentro

de un programa hace que el código principal y se dirija a ejecutar el código de la rutina, en cambio cuando se llama a una macro, el ensamblador llama al código de la macro y lo implanta donde fue llamado, aumentando así el código fuente y por consiguiente el objeto.

## **IMPORTANCIA TEORICA DE MACROEXPRESIONES Y USOS DE UN MACROPROCESADOR**

Una macroinstrucción (abreviada frecuentemente como macro), una macro representa un grupo de proposiciones utilizadas comúnmente en el lenguaje de programación fuente. El procesador de macros reemplaza cada macroinstrucción con el grupo correspondiente de proposiciones del lenguaje fuente, lo que se denomina expansión de macros. Por todo lo anterior, las macroinstrucciones permiten al programador escribir una versión abreviada de un programa, dejando que el procesador de macros maneje los detalles internos. Las funciones básicas de un procesador de macros son:

- Sustitución de líneas de código por una referencia simbólica.
- Calcular las direcciones efectivas de las referencias.
- Expansión de las referencias en un archivo intermedio.

El procesador de macros, como puede verse, no intenta ningún análisis ni traducción a código objeto del programa fuente, más bien parece que el procesador hace que aumente el tamaño de éste. Esto es un elemento clave para el programador, pues el uso indiscriminado de macros puede hacer que el tamaño del código objeto sea enorme y poco práctico, al contrario de las llamadas a subrutinas. Claro que también se paga un precio alto por el empleo de las llamadas, pues se pierde tiempo importante en la gestión de la pila, donde usualmente se pasan los parámetros. El macroprocesador requiere tres estructuras de datos para su exitosa operación.

- La tabla de nombres de macros (TABNOM).
- La tabla de código de macros (TABDEF).
- La tabla de argumentos (TABARG).

Ahí se guarda toda la información pertinente a las macros, mientras el ensamblador analiza la expansión y hace la traducción a código objeto. A veces hay confusión entre macros y subrutinas. Subrutinas son rutinas comunes en el programa se accede a ellas haciendo llamados a estas, son normalmente usadas para rutinas mas complejas donde el retorno de llamada tolerada.

Macros es comúnmente usado para rutinas mas simples o donde la velocidad de en código de linea es requerida. Para poder utilizar una macro este debe ser definida esta debe ser dentro del programa fuente.

La macrodefinicion consta de 3 partes estas son:

- la macro cabecera que especifica el macro nombre y su lista de parámetro
- el macro cuerpo que es la parte que es insertada actualmente dentro el programa de fuente
- el macro terminador

la cabecera se tiene de la siguiente forma:

name MACRO <parameter list>

El nombre del campo contiene un simbolo unico que el usado para identificar la macro.

Tenemos 2 definiciones de terminación de macros: ENDM y EXITM, toda definición de macro debe tener un ENDM para notificar al ensamblador que la definición de macro ha terminado, EXITM es alternativa para usos de final de macros que son usadas con ensambles condicionales, cuando EXITM es encontrado en un programa las declaraciones de ENDM son ignoradas.

## MACROPROCESADORES CON ARGUMENTOS Y MACROPROCESADORES RECURSIVOS

En el caso de las macros de Intel, con el TASM 2.0 o el MASM de IBM, primero se empieza con el nombre de la macro y después con la palabra clave MACRO. Se cuenta con línea de argumentos.

versión SIC/XE

MACRO nombre\_macro &argumentos

cuerpo\_macro

MEND

versión INTEL

nombre\_macro MACRO &argumentos

cuerpo\_macro

ENDM

Es posible una expansión de macros recursiva, siempre y cuando aseguremos una posición definida de los símbolos que se van a emplear dentro de la macro. Por ejemplo para el SIC/XE, dicha tarea es un poco difícil, aunque si se programa el expansor de macros como una máquina de pila es posible la implementación. Por otra parte está la solución de Intel, la cual nos dice que los símbolos internos de las macros los deberemos marcar como "LOCAL", para que la reubicación de los mismos se factible en la segunda pasada del ensamblador.

- TIPOS RECURSIVOS EN GENERAL: Un tipo recursivo es aquel cuyos valores son compuestos de valores del mismo tipo. Es aquel que esta definido en términos de el mismo. En general, el conjunto de valores de un tipo recursivo T, puede ser definido por un conjunto de ecuaciones

recursivas de la forma:  $T = \dots T$ . El conjunto de ecuaciones recursivas puede tener muchas soluciones. **STRING** Una String es una secuencia de caracteres. Las strings son soportadas por todos los lenguajes de programación moderna. Pero no se tiene consenso en su clasificación: Una string puede ser primitiva o compuesta? Que tipo de operación debe tener? Una string como un tipo primitivo, con valores que son strings de cualquier longitud. Ejem. ML. Una string como una array de caracteres: Pascal y Ada. Una string como una lista de caracteres: Miranda y Prolog.

## **INCORPORACION DEL PROCESADOR DE MACROS AL ENSAMBLADOR**

### Ventajas:

- grupos pequeños Repetidos de instrucciones reemplazadas por 1 macro
- Errores en macros son fijos solo una vez, en la Duplicación
- Por definición el esfuerzo es reducido
- En efecto, instrucciones de nivel mas alto nuevo puede ser creado
- Programa es hecho mas fácil, error menos boca abajo
- Generalmente mas rápido en ejecución que subrutinas

### Desventajas

- En programas grandes, tamaño de código de producto mayor que procedimientos

### Quando usar macros:

- Para reemplazar grupos pequeños de instrucciones no merecedor de subrutinas
- Para crear un conjunto de instrucción mas alta para aplicaciones especificas
- Para crear compatibilidad con otros ordenadores
- Para reemplazar porciones de código que es repetido a menudo por todo

## **REFERENCIAS**

[www.wikipedia.org](http://www.wikipedia.org)  
[www.rincondelvago.com](http://www.rincondelvago.com)