

Declaración de Macros

Prof. José Rafael Rojano Cáceres

<http://www.uv.mx/rrojo>

Arquitectura de Computadoras II

¿Qué es una macro?

- Es la representación de código bajo un nombre.
- Una macro **no es un** procedimiento
- Una macro puede recibir parámetros, pero **no es una** función.
- Una macro se define con una etiqueta, pero **no provoca un salto** del registro IP.
- Una macro representa porciones de instrucciones que definen bajo una etiqueta. Dicha etiqueta se coloca en un programa principal (como si fuera un procedimiento) y posteriormente el compilador sustituye todas las apariciones de la etiqueta por el conjunto completo de instrucciones que hayamos definido bajo ese nombre.

Macros

- En el nasm existen varias macros predefinidas que podemos emplear.
 - Leer el capítulo “The NASM Preprocessor” de la documentación del nasm ubicado dentro de la carpeta doc.
- Las macros en nasm se definen mediante el símbolo %, que es una etiqueta de preprocesamiento

Definiendo macros simples

- Existen los llamados macros de una sola línea
 - Éstos se definen mediante la palabra reservada `define`
 - Su sintaxis es:

- `%define` nombre(argumentos) código

- Ejemplo:

- `%define` incrementa(x,y) add x,y

Escribiendo un programa

- El ejemplo anterior se escribiría en un programa como:

```
%define incrementa(x,y) add x,y  
segment .text  
    ..start:  
        mov ax,4  
        incrementa(ax,3)  
        mov ah,4ch  
        int 21h
```

- ¿Cómo queda este código traducido?

Definiendo macros multilínea

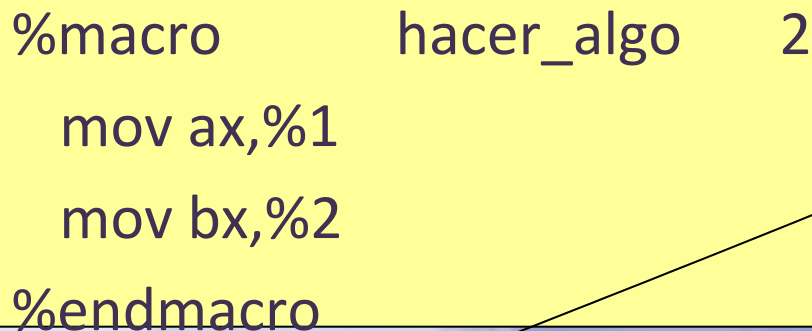
- Por otro lado podemos definir macros que contengan más de una línea
- Para ello se usan las instrucciones:
 - **%macro** y **%endmacro** para englobar el conjunto de instrucciones deseadas
- Sintaxis macro multilínea:

```
%macro nombre num_parametros  
    conjunto de instrucciones  
%endmacro
```

Definiendo macros multilínea

- Ejemplo:

```
%macro      hacer_algo      2  
    mov ax,%1  
    mov bx,%2  
%endmacro
```



- El numero **2** representa que la macro recibe dos parámetros
- El símbolo de **%1** y **%2** es la forma de recuperar los parámetros usados, aquí no se definen variables como nombres parámetros.

Observaciones

- Las macros son sensible al caso, por ello si defines una macro cuyo nombre sea con minúsculas, para posteriormente invocarla con mayúsculas esto provocará un error.

Definiendo librerías

- Un conjunto de macros representan una librería.
- Dichas macros no siempre se definen dentro del archivo principal, sino que se pueden colocar externamente en otro archivo.
- Para incluir las librerías en tu archivo deseado solo se usa la palabra reservada

```
%include "archivo_macros.inc"
```

Etiquetas dentro de las macros

- Al igual que en el resto del código en una macro se pueden definir etiquetas, el **único problema** es que si una macro es invocada más de una vez en el mismo archivo la etiqueta estaría duplicada.
- Por ello para definir etiquetas como locales se emplea el símbolo `%%`
- Veamos un ejemplo

Etiquetas dentro de las macros

```
%macro repite 0  
    mov cx,10  
    mov dx,1  
    %%ciclo:  
        inc dx  
    loop %%ciclo  
%endmacro
```

Ejemplo de uso de repite

```
%macro repite 0
    mov cx,10
    mov dx,1
    %%ciclo:
        inc dx
    loop %%ciclo
%endmacro
```

```
segment .text
..start:
    repite
    repite
```



Let's try some examples

Ver video en:

<http://148.226.81.254/~rocajo16/videos/nasm/>

Ejercicios

1. Crear un programa en nasm con macros que se vea como a continuación se muestran:

```
%include "libreria.inc"
```

```
Segment .data
```

```
msj1 db "mensaje 1$"
```

```
msj2 db "mensaje 2$"
```

```
buffer db 10,0
```

```
resb 10
```

```
Segment .text
```

```
Inicio
```

```
Imprime msj1
```

```
Leer cadena
```

```
Imprime msj2
```

```
Impr_buff buffer
```

```
fin
```

; esta debe ser una macro de una línea

; esta debe ser una macro en libreria.inc

; esta debe ser una macro en libreria.inc

; esta debe ser una macro en libreria.inc

Referencias

- Documentación del nasm, The nasm development team, versión 0.98.39, 2003