

Макросредства языка Ассемблер

Макросредства - инструменты (средства) модификации текста программы во время ее трансляции. Макросредства предназначены для облегчения написания программ на языке Ассемблер и для улучшения понимания исходного текста программы.

Основная идея. Повторяющийся фрагмент программы специальным образом описывается (макрос), именуется, а затем в нужных местах программы указывается ссылка на него. При создании объектного кода вместо ссылки подставляется сам фрагмент, т.е. происходит подстановка фрагмента вместо ссылки.

Транслятор ассемблера состоит из двух частей: макрогенератора и непосредственно транслятора. Транслятор при этом называют макроассемблером. Именно таким макроассемблером мы пользуемся в системе MASM (macroassembler language).

Обработка программы с использованием макросредств осуществляется транслятором в два этапа. На первом этапе работает макрогенератор, который производит замены для всех макросов, а на втором этапе уже преобразованный текст программы транслируется в объектный код.

Макроопределение

Это описание макроса. Синтаксис макроопределения:

```
<имя макрокоманды> MACRO [формальные параметры]
                                тело макроопределения
                                ENDM
```

Директива MACRO - это заголовок макроопределения. В ней указывается имя и через запятую перечисляются формальные параметры, если необходимо.

Формальные параметры позволяют копировать макрос не в неизменном виде, а с изменениями. Те величины, которые необходимо будет изменить описываются формальными параметрами.

Замечание. Имена формальных параметров локальны по отношению к макросу, т.е. они могут совпадать с именами в основной программе, макрогенератор понимает их как параметры.

Завершает макроопределение директива ENDM. !!! Не надо повторять имя макроса.

Пример 1 ; настройка сегмента данных

```
initds macro
    mov ax, @data
    mov ds, ax
endm
```

Пример 2 ; вывод строки на экран

```
outstr macro str ; форм. параметр - имя строки
    push ax
    mov ah, 09h
    lea dx, str
    int 21h
    pop ax
endm
```

Размещаться макроопределения могут :

1. В любом месте программы.

!!! Обязательно до первой ссылки на него.

2. В отдельном файле.

Чтобы сделать доступными макроопределения в программе, необходимо в начале программы использовать директиву INCLUDE <имя файла>. При этом на этапе работы макрогенератора текст указанного файла будет вставлен полностью на место директивы.

Пример. Masm
 model small
 include Mymacro.inc
 ...

Можно универсальные макрокоманды записать в один файл, в так называемую макробиблиотеку. Подключать ее с помощью директивы include.

Чтобы в текст программы не включать лишние макроопределения, можно воспользоваться директивой

PURGE <список через запятую имен макроопределений>

Директива указывает, какие макроопределения не должны включаться в текст программы.

Пример. ...
 include mymacro.inc
 purge outstr, initds
 ...

Макрокоманды

Макрокоманда - обращение к макроопределению. Или указание макрогенератору на то, что на указанном месте необходимо подставить тело макроопределения. Итак, одна макрокоманда заменяется на группу команд, поэтому она и наз. макро (большая).

Синтаксис макрокоманды:

<имя макроса> [<фактические параметры>]

Замечание. Факт. пар-ры можно разделять запятыми или пробелами.

Формальные параметры макроопределения заменяются соответствующими фактическими параметрами макрокоманды.

!!! i-тый фактич. пар-р соответствует i-тому формальному параметру

!!! Число фактических параметров должно быть равно числу формальных параметров, если фактических параметров больше, то лишние игнорируются

если формальных больше, считается что в качестве недостающих фактических указаны пустые тексты

Процесс замещения формальных параметров фактическими называется **макроподстановкой**.

Результат макроподстановки (т.е. полученный в результате текст называется **макрорасширением**.

Действия макрогенератора:

- 1) макрогенератор находит макроопределение с указанным именем
- 2) в его теле заменяет все формальные параметры фактическими
- 3) полученное макрорасширение подставляет в программу вместо макрокоманды

Пример. См. выше описание макроопределения OUTSTR

```
.data
message db 'Введите два числа',0dh,0ah,'$'
answer  db 0dh,0ah,'Полученный результат: $'
```

```
repeat      db  0dh,0ah, 'Продолжить ? (y/n) $'
```

```
...
```

```
.code  ...
```

```
outstr message ...
```

```
outstr answer ...
```

```
outstr repeat ...
```