

# Учебный курс. Часть 17. Команды LOOPZ и LOOPNZ

Автор: xrnd | Рубрика: [Учебный курс](#) | 05-05-2010 |  [Распечатать запись](#)

Кроме команды [LOOP](#) и команд условных переходов существуют ещё две команды, позволяющие организовывать циклы. Это команды [LOOPZ](#) (или её синоним [LOOPE](#)) и [LOOPNZ](#) (синоним — [LOOPNE](#)). Действие этих команд очень напоминает [LOOP](#), за исключением того, что дополнительно анализируется флаг нуля ZF.

Переход к метке цикла осуществляется в том случае, если после декремента содержимое CX не равно 0 и выполняется условие: ZF=1 (для команды [LOOPZ/LOOPE](#)) или ZF=0 ([LOOPNZ/LOOPNE](#)).

Эти команды удобно использовать в алгоритмах, где цикл должен завершаться в двух случаях:

- выполнено требуемое количество итераций;
- выполнено некоторое условие досрочного завершения цикла.

Простейший пример такого алгоритма — поиск числа или символа в массиве. Поиск завершается, если один из элементов массива совпал с искомым или если достигнут конец массива. В качестве примера рассмотрим программу для поиска символов в строке. Пользователь вводит символ, а программа определяет, содержится такой символ в строке или нет. Для выхода из программы нужно нажать ENTER.

```
1 use16 ;Генерировать 16-битный код
2 org 100h ;Программа начинается с адреса 100h
3 jmp start ;Безусловный переход на метку start
4 ;-- Данные -----
5 string db 'Hello!',13,10,'$'
6 length db 6
7 s_entchar db 'Enter char (Press ENTER to exit):$'
8 s_found db ' - found!',13,10,'$'
9 s_nfound db ' - not found!',13,10,'$'
10 ;-----
11 start:
12 mov ah,09h ;\
13 mov dx,string ; > Вывод строки
14 int 21h ;/
15 main:
16 mov dx,s_entchar ;\
17 int 21h ;/ Вывод приглашения для ввода символа
18
19 mov ah,01h ;\
20 int 21h ;/ Ввод символа
21 cmp al,0Dh ;Нажата клавиша ENTER?
22 je exit ;Если да, то переход на метку exit
23
24 mov bx,string-1 ;BX = (адрес строки - 1)
25 movzx cx,[length] ;CX = длина строки
26 search:
27 inc bx ;Инкремент BX
28 cmp al,[bx] ;Сравнение введённого символа с символом строки
29 loopne search ;Цикл, если не равно.
30 je found ;Если равно, то символ найден.
31
32 mov dx,s_nfound ;DX = адрес строки ' - not found!'
```

```

32     mov ax,s_notfound    ;DX = адрес строки ' - not found!
33 print_result:
34     mov ah,09h           ;\
35     int 21h              ;/ Вывод результата поиска
36     jmp main             ;Безусловный переход на метку main
37 found:
38     mov dx,s_found       ;DX = адрес строки ' - found!'
39     jmp print_result     ;Безусловный переход на метку print_result
40 exit:
41     mov ax,4C00h         ;\
42     int 21h              ;/ Завершение программы

```

Обратите внимание, изначально в ВХ загружается значение (адрес строки — 1), так как цикл начинается с команды инкремента. Результат работы программы:

```

C:\search.com
Hello!
Enter char (Press ENTER to exit):h - not found!
Enter char (Press ENTER to exit):l - found!
Enter char (Press ENTER to exit):o - found!
Enter char (Press ENTER to exit):H - found!
Enter char (Press ENTER to exit):a - not found!
Enter char (Press ENTER to exit):z - not found!
Enter char (Press ENTER to exit):! - found!
Enter char (Press ENTER to exit):

```

## Упражнение

Объявите в программе два массива слов. Размер массивов должен быть одинаков и храниться в 8-битной переменной без знака. Напишите программу сравнения двух массивов, используя команду [LOOPZ](#). (Массивы равны, если все их элементы соответственно равны. Цикл можно завершить, если найдена хотя бы одна пара не совпадающих элементов). Выведите на экран строку, сообщающую о результате сравнения. Сами массивы печатать не нужно. Свои программы можете выкладывать в комментариях.

[Следующая часть »](#)

## Комментарии:

RoverWWorm  
14-05-2010 16:24

use16  
org 100h

```
jmp start
;_____
array1 dw 2123,1456,2000,2222,1111
array2 dw 2123,1456,2000,2222,1111
```

```
length db 5
```

```
ravno db 'massivbl ravnbl$'
neravno db 'massivbl ne ravnbl$'
;_____
```

```
start:
movzx cx,[length]
sub si,si
```

```
loop1:
mov ax,word[array1+si]
mov bx,word[array2+si]
add si,2
cmp ax,bx
loopz loop1
```

```
jz ravno1
```

```
mov ah,09h
mov dx,neravno
int 21h
```

```
jmp exit
```

```
ravno1:
mov ah,09h
mov dx,ravno
int 21h
```

```
exit:
mov ah,08h
int 21h
```

```
mov ax,4C00h
int 21h
```

[\[ОТВЕТИТЬ\]](#)

[xrnd](#)

14-05-2010 20:00

Замечательно, всё правильно.

[\[ОТВЕТИТЬ\]](#)

argir

12-12-2010 12:36

Задание понял немного неправильно — слова...слова, но интересно было посмотреть размерность строкового массива. И как можно поставить в строку знак «≠»?

```
use16
```

```
org 100h
```

```
jmp start
```

```
array1 db 'топот$', 'шорох$', 'поп$', 'горох$'
```

```
array2 db 'топот$', 'поп$', 'шорох$', 'горох$'
```

```
rm db 22
```

```
vivod1 db 'array1 = array2$'
```

```
vivod2 db 'array1 array2$'
```

```
pak db 13,10,'Press any key...$'
```

```
start:
```

```
xor ax,ax
```

```
movzx cx,[rm]
```

```
mov si,-1
```

```
prov: inc si
```

```
mov al,[array1+si]
```

```
cmp al,[array2+si]
```

```
loopz prov
```

```
jz v1
```

```
mov dx, vivod2
```

```
jmp vivod
```

```
v1: mov dx, vivod1 ; Так чуть меньше кода, чем в предыдущем примере
```

```
vivod : mov ah,09h
```

```
int 21h
```

```
exit:
```

```
mov ah,09h
```

```
mov dx,pak
```

```
int 21h
```

```
mov ah,08h
```

```
int 21h
```

```
mov ax,4C00h
```

```
int 21h
```

[\[Ответить\]](#)

[xrnd](#)

12-12-2010 22:40

Массив слов порадовал 😊

Программа написана правильно.

Знака «≠» нет в консольном шрифте, его в строку не добавить.

[\[Ответить\]](#)

Гость

15-01-2011 20:51

```
use16 ;Ãíð¸ðíòü 16-¸òíú¸ êî
```

```
org 100h ;Ïðíðì ì÷¸íðñý ñ ðñ 100h
```

```
jmp start ;Ãçóñ¸ííú¸ ìððí ì ìð¸ó start
```

```

;— Äàíüâ —————
array1 db 'l$', 'o$', 'o$', 'p$'
array2 db 'l$', 'o$', 'o$', 'p$'
length db 5
press db 'ne pavn $'
press1 db ' pavn $'
;—————
start:
movzx cx,[length]
mov si,1
mov ah,09h

E:
mov dl,byte[array1+si+1]; +1 чтобы убрать '$'
mov dh,byte[array2+si+1]
cmp dl,dh ;если результат =0 , значит равны
Jnz E1 ; если результат не ноль переход по метке
inc si ; si +1 , z =0
mov di,cx
sub di,di ; z=1

loopz E

mov dx,press1
int 21h
jmp exit
E1:
mov dx,press
int 21h
exit:
mov ah,08h
int 21h
mov ax,4C00h
int 21h

```

[\[Ответить\]](#)

[xrnd](#)

18-01-2011 21:40

Слова без знака — это 16-битные целые, а вовсе не те слова, которыми люди разговаривают :))))))

А это какой-то хитрый хак:

```

mov di,cx
sub di,di ; z=1
loopz E

```

Я имел в виду совсем другое. Вместо этих трех команд можно просто использовать команду LOOP. Смысл этого упражнения в том, чтобы проверять флаг ZF командой LOOPZ. А переход сделать после команды цикла. Если ZF=1 массивы равны, иначе — массивы не равны.

[\[Ответить\]](#)

Гость

18-01-2011 22:32

На самом деле всё дела в

inc si , эта команда меняет флаг , поэтому я не как немог понять , как сделать так чтобы , +1 , но флаг не трогать , нечего лутьше придумать сам не смог )

На самом деле я понял принцы , но не понял как его реализовать , ну потом подсматрел как другие это сделали ,)

И понял cmp тоже меняет флаг , не тралезуя эффект inc si

В конечном итоге я хотел вот это ,)

E:

```
mov dl,byte[array1+si+1];
```

```
mov dh,byte[array2+si+1];
```

```
inc si
```

```
cmp dl,dh
```

```
loopz E
```

```
Jz E1 ; неравны
```

```
mov dx,press1 ; равны
```

[\[Ответить\]](#)

[xrnd](#)

19-01-2011 18:33

Ага. Только неравны и равны надо местами поменять.

Либо делать переход командой

```
jnz E1 ; неравны
```

[\[Ответить\]](#)

Knight212

15-02-2011 21:57

```
use16
```

```
org 100h
```

```
jmp start
```

```
a1 db 0, 2, -5
```

```
a2 db 0, 2, -5
```

```
number db 3
```

```
result_yes db 'Arrays are equal', 0Dh, 0Ah, '$'
```

```
result_no db 'Arrays are not equal', 0Dh, 0Ah, '$'
```

```
start:
```

```
mov bx, a1-1
```

```
mov si, a2-1
```

```
movzx cx, [number]
```

```
jmp compare
```

```
compare:
```

```
inc bx
```

```
inc si
mov al, [bx]
cmp al, [si]
loope compare
jne no
mov dx, result_yes
jmp print_result
```

```
no:
mov dx, result_no
```

```
print_result:
mov ah, 09h
int 21h
```

```
mov ah, 08h
int 21h
mov ax, 4C00h
int 21h
```

[\[Ответить\]](#)

[xrnd](#)

18-02-2011 13:11

В общем всё правильно, но массивы состоят из байтов, а не из слов.  
И есть одна лишняя команда:

```
jmp compare
compare:
```

[\[Ответить\]](#)

Shov  
31-03-2011 01:09

```
use16
org 100h
jmp start
;_____
arr1 dw 1234, 4033, 5645
arr2 dw 1234, 4033, 5645
count db 3
s_equal db 10,13,'equal! ',10,13,'$'
s_nequal db 10,13,'not equal! ',10,13,'$'
;_____
start:
movzx cx,[count]
xor si,si
compar:
mov ax,word[arr1+si]
mov bx,word[arr2+si]
add si,2
```

```
cmp ax,bx
loopz compar
```

```
jz equal
mov dx,s_nequal
jmp output
equal:
mov dx,s_equal
output:
mov ah,09h
int 21h
mov ah,08h
int 21h
mov ax,4C00h
int 21h
```

[\[Ответить\]](#)

[xrnd](#)

01-04-2011 15:51

Здесь тоже всё правильно 😊

[\[Ответить\]](#)

алекс  
13-03-2012 18:38

```
use16
org 100h
```

```
jmp start
x db 5
y db 7
array1 dw 10,20,35,40,50
array2 dw 10,20,30,40,50
length db 5
equal db 13,10, 'Equal$'
nequal db 13,10, 'Not equal$'
press db 13,10, 'Press any key...$'
```

```
start:
movzx cx,[length]
xor si,si
mov bx,array1-2
che:
```

```
mov ax,[array2+si]
add bx,2
add si,2
cmp ax,[bx]
loopz che
jne nee
```



```
mov ah,09h
mov dx,equal
int 21h
jmp exit
```

```
nee:
mov ah,09h
mov dx,nequal
int 21h
```

```
exit:
mov ah,09h
mov dx,press
int 21h
mov ah,08h
int 21h
mov ax,4c00h
int 21h
```

[\[Ответить\]](#)

алекс  
13-03-2012 18:40

Я вот попробовал конструкцию

```
cmp [ax],[bx] и
cmp [array1],[array2]
```

так компилятор ругается....

Потом вычитал, что оба операнда не могут быть ячейками памяти, только один.....  
может я ошибаюсь.....

[\[Ответить\]](#)

T86  
14-03-2012 07:47

Оба операнда не могут быть ячейками памяти

[\[Ответить\]](#)

[disciple](#)  
10-08-2012 21:45

извините пока нашел сайт, пока дошел до 17-го упражнения уже и опоздал, но надеюсь вы подскажете мне где-же тут ошибка.

```
USE16
ORG 100H
JMP START
N DB 5
ARRAY1 DB 1,1,1,1,1
ARRAY2 DB 1,1,1,1,1
STRING0 DB 10,13,'=$'
STRING1 DB 10,13,'!=$'
```

```

STRING2 DB 'Press Enter to run programm$',10,13
STRING3 DB 10,13,'Press Esc to exit$'
START:
MOV AH,09H
MOV DX,STRING2
INT 21H
MOV AH,09H
MOV DX,STRING3
INT 21H
MOV AH,01H
INT 21H
CMP AL,01B
JE EXIT
CMP AL,0DH
JE LOOP2

EXIT:
MOV AH,08H
INT 21H
MOV AX,4C00H
INT 21H

EXIT1:
MOV AH,09H
MOV DX,STRING1
INT 21H
JMP EXIT

LOOP2:
MOV BX,ARRAY1
MOV DX,ARRAY2
MOVZX CX,[N]
COMPARE:
INC BX
INC DX
CMP BX,DX
LOOPNZ COMPARE
JNE EXIT1
MOV AH,09H
MOV DX,STRING0
INT 21H

```

[\[Ответить\]](#)

[webprog26](#)

06-02-2014 18:21

Как-то так. Работает.

```

USE16
ORG 100H
JMP START

```

```

;_____
N DB 5

```

```
ARRAY1 dw 1,1,1,1,1
ARRAY2 dw 1,1,1,1,1
STRING0 DB 10,13,'=$'
STRING1 DB 10,13,'!=$'
STRING2 DB 'Press Enter to run programm$',10,13
STRING3 DB 10,13,'Press Esc to exit$'
```

```
;_____
```

```
START:
```

```
sub si,si
```

```
MOVZX CX,[N]
```

```
MOV AH,09H
```

```
MOV DX,STRING2
```

```
INT 21H
```

```
MOV AH,09H
```

```
MOV DX,STRING3
```

```
INT 21H
```

```
MOV AH,01H
```

```
INT 21H
```

```
CMP AL,01B
```

```
JE EXIT
```

```
CMP AL,0DH
```

```
JE COMPARE
```

```
EXIT:
```

```
MOV AH,08H
```

```
INT 21H
```

```
MOV AX,4C00H
```

```
INT 21H
```

```
EXIT1:
```

```
MOV AH,09H
```

```
MOV DX,STRING0
```

```
INT 21H
```

```
JMP EXIT
```

```
COMPARE:
```

```
MOV AX,word[ARRAY1+si]
```

```
MOV BX,word[ARRAY2+si]
```

```
add si,2
```

```
CMP AX,BX
```

```
LOOPZ COMPARE
```

```
jz EXIT1
```

```
mov ah,09h
```

```
mov dx,STRING1
```

```
int 21h
```

```
jmp EXIT
```

[\[Ответить\]](#)

Александр  
05-11-2016 21:15

se16; вывод на консоль каждого сравнения. в столбик  
org 100h

```
mov ah,09h
movzx cx,[length]
mov bx,str1-2
mov si,str2-2
```

lp2:

```
add bx,2
add si,2
mov al,[bx]
cmp al,[si]
loope lp1
jg more
jl less
```

lp1:

```
mov dx,A
int 21h
cmp cx,0h
je exit
jmp lp2
```

more:

```
mov dx,B
int 21h
cmp cx,0h
je exit
jmp lp2
```

less:

```
mov dx,C
int 21h
cmp cx,0h
je exit
jmp lp2
```

exit:

```
mov ah,08h
int 21h
```

```
mov ax, 4C00h
int 21h
```

```
;
```

---

```
str1 dw 111,222,333,454,555,666,777,888,999,1001
str2 dw 101,222,333,444,555,666,777,888,999,1000
length db 10
A db 13,10,'a=b$'
B db 13,10,'a>b$'
C db 13,10,'a<b$'
```

[\[Ответить\]](#)

Александр  
06-11-2016 23:15

Можно немного оптимизировать  
«jcxz метка» (переход при CX=0)? вместо:  
«cmp cx,0h  
je метка»

[\[Ответить\]](#)

Asm  
18-03-2017 13:04

Почему не работает (При нажатие Enter, виснет dos-box):

```
use16
org 100h
```

```
jmp start
```

```
string db 'Hello!', 13, 10, '$'
length db 6
ask db 13, 10, 'Enter a char (ENTER to exit) : $'
f db 'The same character is exists!$'
nf db 'The same character is not exists!$'
```

```
start:
mov ah, 09h
mov dx, string
int 21h
```

```
main:
mov ah, 09h
mov dx, ask
int 21h
```

```
mov ah, 01h
int 21h
```

```
cmp al, 0Dh
je exit
jmp main
```

```
exit:
mov ax, 4c00h
int 21
```

[\[Ответить\]](#)

## Ваш комментарий

Имя \*

Почта (скрыта) \*

Сайт

Добавить

☐ Уведомить меня о новых комментариях по email.

☐ Уведомлять меня о новых записях почтой.