

Команды LOOPZ и LOOPNZ

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 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.

```
use16                                ;Генерировать 16-битный код
org 100h                             ;Программа начинается с адреса 100h
    jmp start                        ;Безусловный переход на метку start

;-- Данные -----
string    db 'Hello!',13,10,'$'
length    db 6
s_entchar db 'Enter char (Press ENTER to exit):$'
s_found   db ' - found!',13,10,'$'
s_nfound  db ' - not found!',13,10,'$'
;-----

start:
    mov ah,09h                      ;\
    mov dx,string                   ; > Вывод строки
    int 21h                         ;/

main:
    mov dx,s_entchar                ;\
    int 21h                         ;/ Вывод приглашения для ввода символа

    mov ah,01h                      ;\
    int 21h                         ;/ Ввод символа
    cmp al,0Dh                      ;Нажата клавиша ENTER?
    je exit                         ;Если да, то переход на метку exit

    mov bx,string-1                 ;BX = (адрес строки - 1)
    movzx cx,[length]               ;CX = длина строки
search:
    inc bx                          ;Инкремент BX
    cmp al,[bx]                     ;Сравнение введённого символа с символом строки
    loopne search                   ;Цикл, если не равно.
```

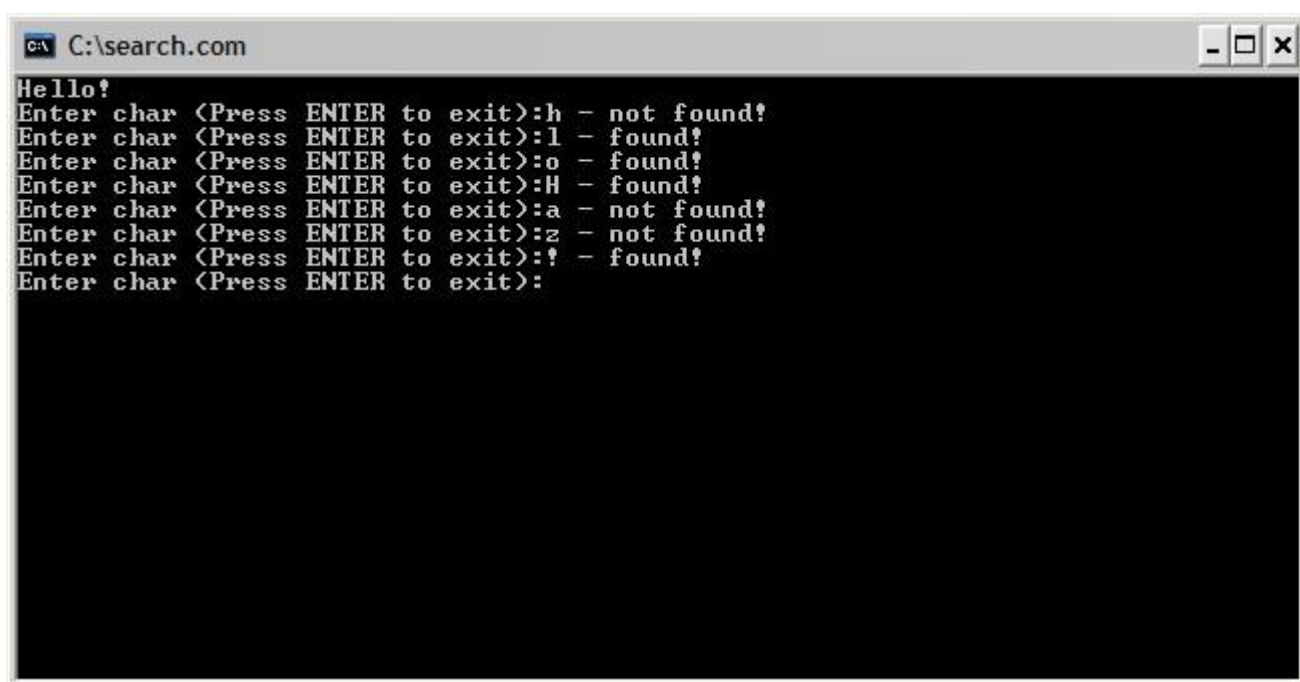
```

    je found                ;Если равно, то символ найден.

    mov dx,s_nfound        ;DX = адрес строки ' - not found!'
print_result:
    mov ah,09h             ;\
    int 21h               ;/ Вывод результата поиска
    jmp main               ;Безусловный переход на метку main
found:
    mov dx,s_found         ;DX = адрес строки ' - found!'
    jmp print_result       ;Безусловный переход на метку print_result
exit:
    mov ax,4C00h           ;\
    int 21h               ;/ Завершение программы

```

Обратите внимание, изначально в BX 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):

```

Задание.

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 можно не печатать.