

Тема 10

Строковые команды

© 2011-2019

Команды для работы со строками

Команды для работы со строками (строковые команды) не обладают, по сравнению с ранее рассмотренными командами, особыми функциональными возможностями.

Преимущества использования таких команд:

- более короткая и понятная запись исходного кода;
- более быстрая работа за счет того, что не теряется время на междуккомандные переходы;
- возможность работать в режиме «память – память».

Термины, используемые в строковых командах

- *Строка* или *цепочка* – последовательность байтов, слов или двойных слов (элементов строки), расположенная в смежных участках памяти.
- *Источник* и *приемник* – две строки, с которыми работают строковые команды. Источник не изменяется, а приемник может меняться. Источник находится в сегменте, связанном с регистром **DS**, а приемник – с регистром **ES**.
- *Текущий элемент* строки – элемент, адрес которого находится в регистре **SI/ESI** (для источника) или в регистре **DI/EDI** (для приемника).

Термины, используемые в строковых командах (продолжение)

- *Аккумулятор* – регистр **AL**, **AX** или **EAX** (в зависимости от длины элемента строки).
- *Направление* просмотра строки определяется флагом **DF** (если он установлен, строка просматривается от больших адресов к меньшим).

Фазы выполнения строковых команд

- обработка текущего элемента строки в соответствии с командой;
- переход к следующему элементу строки.

Мнемоника строковых команд

| <i>Код команды</i> | <i>Действие, выполняемое командой</i> |
|--------------------|---|
| MOVS [x] | пересылает текущий элемент строки-источника в строку-приемник |
| CMPS [x] | сравнивает текущие элементы строки-приемника и строки-источника, устанавливая соответствующие флаги |
| LODS [x] | пересылает текущий элемент строки-источника в регистр-аккумулятор |
| STOS [x] | пересылает содержимое регистра-аккумулятора в текущий элемент строки-приемника |
| SCAS [x] | сравнивает содержимое регистра-аккумулятора с текущим элементом строки-приемника |
| CLD | сбрасывает флаг DF |
| STD | устанавливает флаг DF |

x – **B**, **W** или **D** (в зависимости от длины элемента строки)

Две формы записи строковых команд (на примере MOVS)

- Первая форма

MOVS **приемник** , **источник**

- Вторая форма (без операндов)

MOVSB ; **цепочка байтов**

MOVSW ; **цепочка слов**

MOVSD ; **цепочка двойных слов**

Префиксы повторения в строковых командах

- **REP**
- **REPE / REPZ**
- **REPNE / REPNZ**

Префиксы повторения записываются непосредственно перед командой, в той же строке:

REPE CMQB

Префиксы работают с регистром **СХ/ЕСХ**, используя его как счетчик количества повторений

Шаги выполнения строковых команд с префиксом повторения

На каждом этапе цикла выполняются следующие действия:

- 1) проверка **ЕСХ**. Если он равен 0 – выход из цикла и переход к следующей команде;
- 2) обработка текущего элемента строки в соответствии с командой;
- 3) уменьшение **ЕСХ** на единицу без изменения значения флага (декремент);

Шаги выполнения строковых команд с префиксом повторения

- 4) проверка флага **ZF**, если префиксом является **REPE** или **REPNE**:

выход из цикла, если префиксом является **REPE** и **ZF=0** (последнее сравнение не совпало) или используется префикс **REPNE** и **ZF=1** (последнее сравнение совпало);

- 5) изменение значения индексных регистров в соответствии со значением флага направления и переход на начало цикла.

Пример 1 использования строковых команд

Задача: переслать содержимое одной строки в другую

```
int main() {  
    char s1[20], s2[20];  
    strcpy(s1, "Hello, world!");  
    _asm {  
        mov     ecx, 20  
        lea     edi, s2  
        lea     esi, s1  
        rep     movsb  
    }  
    cout << s1 << "\n";  
    cout << s2 << "\n";  
    return 0;  
}
```

Пример 2 использования строковых команд

Задача: Заполнить массив из 256 байтов последовательно символами с кодами от 255 до 0.

```
int main() {  
    char s1[256];  
    _asm {  
        cld  
        lea    edi, s1  
        mov    ecx, 256  
        mov    al, 0ffh  
m1:  
        stosb  
        dec    al  
        loop   m1  
    }  
    cout << s1 << "\n";  
    return 0;  
}
```

Пример 3 использования строковых команд

Задача: Выполнить циклический сдвиг элементов массива из 10 байт, т.е. переслать первый элемент на место второго, второй – на место третьего, ..., последний – на место первого.

```
void main() {  
    char s1[] = "0123456789";  
    _asm {  
        std      ; пересылку выполняем с конца массива  
        lea      edi, s1  
        add      edi, 9 ; edi указывает на последний элемент  
        mov      esi, edi  
        dec      esi    ; а esi – на предпоследний  
        mov      ah, es:[edi] ; последний элемент сначала  
                                ; сохраняем  
  
        mov      ecx, 9  
        rep movsb      ; пересылка  
        mov      es:[edi], ah ; на первое место записываем  
                                ; сохраненное  
    }  
}
```

Пример 4 использования строковых команд

Задача: Подсчитать количество пробелов в строке

```
int main() {  
    char s1[21];  
    char rez;  
    strcpy(s1, "0123b b vbbv b");  
    _asm {  
        cld  
        xor ah, ah ; здесь будет накапливаться  
результат  
        mov al, '  
        lea edi, s1  
        mov ecx, 20
```

Пример 4 использования строковых команд (продолжение)

```
m1:
    repne    scasb
        jne  m2
        inc  ah
        jmp  m1
m2:
    mov  rezt, ah
}
cout << s1 << "\n";
cout << (int)rezt << "\n";
return 0;

}
```