

Строковые операции называют цепочечными или операциями над цепочками байтов.

Строка-это цепочка байтов, для которой известен алгоритм определения ее длины. В языках программирования прописывают процедуру определения длины строки, а также вводят ограничение на то, какие байты могут содержаться в строке, а также в какой кодировке должны интерпретироваться строки- сколько байтов кодирует символ, и какая таблица будет использоваться для сопоставления байтов и символов. В языке Си под строкой понимается последовательность байтов, заканчивающаяся байтом со значением 0.

Строковые операции работают с одним элементом строки: байтом, словом, двойным словом. Для того чтобы команда манипулировала последовательностью таких элементов, используются команды-префиксы:

REP-повторять, пока содержимое *ECX* не обратится в 0

REPE/REPZ-повторять пока равно/нуль. Данный префикс заставляет выполнять строковую команду до тех пор, пока содержимое *ECX* не обратится в 0

REPNE/REPZ-повторять пока не равно/не нуль. Данный префикс заставляет выполнять строковую команду до тех пор, пока содержимое *ECX* не обратится в 0

Строковые команды считают, что строка -источник находится по адресу DS:ESI(для нас это просто ESI),а строка-приемник по адресу ES:EDI(для нас просто EDI). Все строковые команды можно разделить на шесть групп:

Команды пересылки

MOVS - копирование строки байтов.

MOVSW - копирование строки слов.

MOVSD - копирование строки двойных слов

Возможен также формат *MOVS* приемник, источник-в этом случае ассемблер по типу операндов сам определяет, какую из трех форм команд следует выбрать. Команда копирует элемент строки из адреса, определяемого ESI, в адрес, определяемый EDI. После выполнения такой команды содержимое ESI и EDI увеличивается(флаг DF=0) или уменьшается(флаг DF=1) на размерах элемента строки

Команды сравнения

CMPS - сравнение строк байтов.

CMPSW - сравнение строк слов.

CMPSD - сравнение строк двойных слов

Команды поиска(сканирования)

SCAS - сканирование строки байтов.

SCASW - сканирование строки слов

SCASD - сканирование строки двойных слов

Команды чтения из строки

LODSB - чтение байта из строки.



LODSW - чтение слова из строки.

LODSD - чтение двойного слова из строки.

Возможен также формат LODS источник-в этом случае ассемблер по типу операндов сам определяет, какую из трех форм команд следует выбрать. Команда осуществляет копирование из памяти, которая адресуется ESI, элемента в AL,AX или EAX. После выполнения такой команды содержимое EDI увеличивается на размер элемента строки

Команда записи в строку

STOSB - запись байта в строку.

STOSW - запись слова в строку.

STOSD - запись двойного слова в строку.

Возможен также формат STOS приемник, источник-в этом случае ассемблер по типу операндов сам определяет, какую из трех форм команд следует выбрать. Команда осуществляет копирование в память, которая адресуется EDI, элемента из AL,AX или EAX. После выполнения такой команды содержимое EDI увеличивается на размер элемента строки

Команды чтения/записи строки из порта

При использовании префиксов (*REP*, *REPZ/REPE*, *REPNZ/REPNE*)действие команд распространяется на цепочки байтов.

Использование строковых команд

```
// string.cpp: определяет точку входа для консольного приложения.
```

```
//
```

```
#include "stdafx.h"
```

```
#include <windows.h>//необходим для работы DWORD
```

```
#include <stdio.h> //необходим для работы printf
```

```
#include <conio.h>//необходим для работы _getch()
```

```
/*Объявление функции*/
```

```
DWORD lens (char *);//функция определения длины строки
```

```
void cats(char*, char*,char*);//функция слияния двух строк в третью
```

```
/*Объявление переменных*/
```

```
char a[]="fdhfljliop";
```

```
char b[]="12345";
```

```
char c[]="4";
```

```
int _tmain(int argc, _TCHAR* argv[])
```

```
{
```

```
__asm{
```

```
/*мы собираемся вызывать процедуру cats,которая своими параметрами имеет 3 указателя
```

```
на строку типа char(в порядке a,b,c). Чтобы процедура cats могла получить параметры,
```



передаваемые по значению, мы должны поместить их в стек(в порядке с,b,a), а затем выровнять стек */

LEA EAX,c;//помещаем в регистр EAX адрес первого элемента строки c

PUSH EAX;//помещаем содержимое EAX в стек

LEA EAX,b;//помещаем в регистр EAX адрес первого элемента строки b

PUSH EAX;//помещаем содержимое EAX в стек

LEA EAX,a;//помещаем в регистр EAX адрес первого элемента строки a

PUSH EAX;//помещаем содержимое EAX в стек

CALL cats;//вызываем процедуру

ADD ESP,12;// выравниваем стек

};

printf("%s\n",c);//выводим суммарную строку

_getch();

return 0;

}

//функция определения длины строки

DWORD lens (char * s)

{

DWORD l=0;

__asm

{

CLD;//задаем направление сканирования (сбрасываем флаг DF-флаг направления).

//данный флаг учитывается в строковых операциях.Если флаг равен 1, то в строковых операциях

//адрес автоматически уменьшается.Мы сбрасываем этот флаг,чтобы адрес автоматически увеличивался

MOV EDI,s;// помещаем адрес начала строки в регистр EDI

MOV ESI,EDI;//сохраняем адрес строки в регистре ESI

MOV ECX, 0fffffffh;//помещаем в регистр-счетчик цикла максимальное 32-битное целое число

XOR AL,AL;//запускаем бесконечный цикл-операция XOR от двух одинаковых элементов возвращает 0

//таким образом,эта команда представляет с собой "пока 0 ..."

REPNE SCASB;//сканируем строку байтов, пока не встретится 0

SUB EDI,ESI;//вычитаем из текущего адреса, который хранится в EDI, адрес начала строки(хранится в ES ^

//таким образом находим длину строки вместе с терминальным символом

DEC EDI;//вычитаем из полученной длины строки 1- исключаем терминальный символ

```
MOV I,EDI;//помещаем в переменную l на длину строки
```

```
}
```

```
return l;//возвращаем длину строки
```

```
}
```

```
//Функция слияния строк s1+s2->s3
```

```
void cats(char* s1, char* s2,char* s3)
```

```
{
```

```
__asm{
```

```
    /*определяем длину строки s1*/
```

```
    CLD; //задаем направление копирования
```

```
    MOV ESI,s1;// помещаем в регистр ESI указатель на начало строки s1
```

```
    PUSH ESI;//помещаем указатель на s1 в стек
```

```
    CALL lens;//вызываем функцию определения длины строки
```

```
    ADD ESP,4;//выравниваем стек
```

```
    /*копируем содержимое строки s1 в строку s3*/
```

```
    /*функция lens возвращает результат в регистр EAX*/
```

```
    MOV ECX, EAX;//помещаем длину строки s1 в регистр ECX
```

```
    MOV ESI,s1;//помещаем в регистр ESI указатель на строку s1
```

```
    MOV EDI,s3;//помещаем в регистр EDI указатель на строку s3
```

```
    REP MOVSB;//повторяем, пока содержимое ECX не обратится в 0, побайтовое копирование строки ,
```

```
    //адрес начала которой хранится в ESI, в строку адрес начала которой хранится в EDI
```

```
    //(копируем содержимое строки s1 в строку s3)
```

```
    /*копируем содержимое строки s2 в строку s3 начиная с позиции после s1*/
```

```
    MOV ESI,s2;//помещаем в ESI указатель на первый элемент строки s2
```

```
    PUSH ESI;//помещаем указатель на s2 в стек
```

```
    CALL lens;//вызываем функцию определения длины строки
```

```
    ADD ESP,4;//выравниваем стек
```

```
    /*функция lens возвращает результат в регистр EAX*/
```

```
    MOV ECX,EAX;//помещаем длину строки s2 в регистр ECX
```

```
    REP MOVSB;//повторяем, пока содержимое ECX не обратится в 0, побайтовое копирование строки ,
```

```
    //адрес начала которой хранится в ESI(адрес начала строки s2), в строку адрес начала которой хранится в
```

```
EDI
```

```
//(адрес конца строки s1 +1)

//(копируем содержимое строки s2 в строку s3)

MOV BYTE PTR [EDI],0;//фиксируем конец строки

}

}
```

[Добавить ответ на задание](#)

Состояние ответа

Состояние ответа на задание	Ответы на задание еще не представлены
Состояние оценивания	Не оценено

Информация

Официальный сайт ФГБОУ ВО
Белгородский ГАУ

Личный кабинет преподавателя
и студента

Расписание

Отдел электронных
образовательных ресурсов и
сетевого обучения

Структура университета

Контакты

308503, Белгородская обл.,
Белгородский р-н, п. Майский, ул.
Вавилова, 1, отдел электронных
образовательных ресурсов и
сетевого обучения, №321 (с 8.00 до
17.00, перерыв 12.00-13.00)

☎ Телефон : +7 (4722) 39-22-51 (по
вопросам ЭИОС). По вопросам
справок: +7 (4722) 38-05-17 (МФЦ
БелГАУ)

✉ Эл.почта : help@belgau.ru

