|  |  |
| --- | --- |
| Gerb-BMSTU_01 | ***«*Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_\_\_КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОТЧЕТ**

**по домашнему заданию № 1**

**Дисциплина: Машинно-зависимые языки и основы компиляции**

**Название лабораторной работы: Обработка символьной информации**

Студент гр. ИУ6-41Б  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С. А. Самодурова**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С. С. Данилюк**

(Подпись, дата) (И.О. Фамилия)

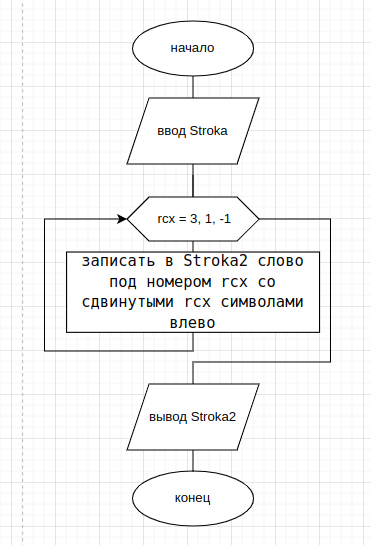
Москва, 2023

**Цель:** изучение команд обработки цепочек и приемов обработки символьной информации.

**Текст задания:**

Дан текст 18 символов: 3 слова по 6 символов. Осуществить кольцевой сдвиг каждого слова: 1-го на 1 символ влево, 2-го на 2 символа, 3-го на 3 символа.

Схема алгоритма программы представлена на рисунке 1.

Рисунок 1 — схема алгоритма

**Код программы:**

%include "io64.asm"

section .data

InpMsg db "Input 3 words: ",10

lenInp equ $-InpMsg

Stroka db "abcdef qwerty abcdef",10

lenStroka equ $-Stroka

Stroka2 db "rrrrrr rrrrrr rrrrrr",10

lenStroka2 equ $-Stroka2

Dlina dw 3

section .bss

InBuf resb 10

lenIn equ $-InBuf

StrokaInp resw 20

lenStr equ $-StrokaInp

section .text

global \_start

\_start:

mov rax, 1 ; системная функция 1 (write)

mov rdi, 1 ; дескриптор файла stdout=1

mov rsi, InpMsg ; адрес выводимой строки

mov rdx, lenInp ; длина строки

syscall ; вызов системной функции

; read

mov rax, 0 ; системная функция 0 (read)

mov rdi, 0 ; дескриптор файла stdin=0

mov rsi, StrokaInp ; адрес вводимой строки

mov rdx, lenStr ; длина строки

syscall ; вызов системной функции

lea rdi,[StrokaInp] ; загружаем адрес строки в edi

mov dx,0 ; обнуляем счетчик длины слова

;алгоритм замены во втором слове ecx=2

mov ecx,3

; mov edx,14

cycl: push rcx

mov eax,7

imul ecx

mov edx,eax

sub edx,7

mov ebx,6

sub ebx,ecx ;4 = ebx

lea esi, [StrokaInp+edx]

lea edi, [Stroka2+edx+ebx]

repe movsb

pop rcx

push rcx

lea esi, [StrokaInp+edx+6]

sub esi,ebx

lea edi, [Stroka2+edx]

mov ecx, ebx; ecx=4

repe movsb

pop rcx

loop cycl

mov rax, 1 ; системная функция 1 (write)

mov rdi, 1 ; дескриптор файла stdout=1

mov rsi, Stroka2 ; адрес выводимой строки

mov rdx, lenStroka2 ; длина строки

syscall ; вызов системной функции

exit:

xor rdi, rdi

mov rax, 60

syscall

**Результат работы программы:**

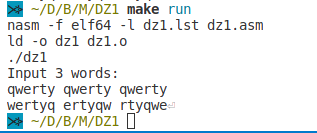
****

Рисунок 2 — результат выполнения

Таблица 1 — проведенные тесты

|  |  |  |  |
| --- | --- | --- | --- |
| № | Входные данные | Ожидаемый результат | Полученный результат |
| 1 | qwerty qwerty qwerty | wertyq ertyqw rtyqwe | wertyq ertyqw rtyqwe |
| 2 | asdfgh zxcvbn poiuyt | sdfgha cvbnzx uytpoi | sdfgha cvbnzx uytpoi |
| 3 | qazwsx edcrfv tgbyhn | azwsxq crfved yhntgb | azwsxq crfved yhntgb |

**Контрольные вопросы**

1. Почему в ассемблере не определены понятия «массив», «матрица»?

Массив и матрица во внутреннем представлении – это последовательность элементов в памяти. В ассемблере понятия «массив» и «матрица» не определены, так как они имеют одинаковое внутреннее представление, не принадлежащие к структуре, полностью контролируются программистом.

2. Как в ассемблере моделируются массивы?

Массив во внутреннем представлении – это последовательность элементов в памяти. В ассемблере такую последовательность можно определить, например, так:

A dw 10,13,28,67,0,-1 ; массив из 6 чисел размером «слово».

Программирование обработки выполняется с использованием адресного регистра, в котором хранится либо смещение текущего элемента относительно начала сегмента данных, либо его смещение относительно начала массива. При переходе к следующему элементу и то, и то смещение увеличивают на длину элемента.

3. Поясните фрагмент последовательной адресации элементов массива?

Почему при этом для хранения частей адреса используют регистры?

Элементы массива размещены в памяти друг за другом. Начало массива выражается смещением. Имея адрес одного, можно получить адрес следующего, добавив размер элемента в байтах. Так, если элементы нумерованы с единицы, то их адрес определяется как:

Aисп = Aначала+(<Номер>-1)\*<Длина элемента>.

Исполнительный адрес считается по формуле:

EA = (База) + (Индекс)\*<Масштаб>+<Смещение>

Полученный по формуле адрес записывается в регистр.

4. Как в памяти компьютера размещаются элементы матриц?

Элементы матриц в памяти компьютера размещаются по строкам или по столбцам. Матрицу можно развернуть в одномерный массив.

5. Чем моделирование матриц отличается от моделирования массивов? В

каких случаях при выполнении операций для адресации матриц используется

один регистр, а в каких – два?

Для инициализации матрицы необходим двойной цикл, в котором будет последовательное увеличение числа строк и изменения индекса в строке, в то время как для инициализации массива достаточно одного цикла и нет зависимости от количества строк.

Для адресации используется один регистр, когда матрица расположена в памяти по строке и просмотр идет по строке. Обработка будет как у массива, один регистр хранит смещение от первого элемента. В другом случае нужно использовать два регистра, в которых будут храниться адрес текущей строки и смещение от начала.

**Вывод:** в ходе выполнения данной лабораторной работы я изучила команды обработки цепочек и приемы обработки символьной информации, программа работает корректно на заданных при тестировании исходных данных.