|  |  |
| --- | --- |
| 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 (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

**О Т Ч Е Т**

по домашнему заданию №\_1\_

**Название**:\_ Обработка символьной информации \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-42б |  | 17.04.2021 | И.С. Марчук |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Г.С. Иванова |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |

*2021 г.*

**Обработка символьной информации**

**Задание:**

Разработать программу на языке ассемблера, которая решает поставленную задачу.

**Вариант 2.27.**

Дан текст 8 слов, разделенных пробелом. Определить количество повторений буквы Е в каждом слове.

**Цель работы:**

Изучение команд обработки цепочек и приемов обработки символьной информации

**Схема алгоритма для решения поставленной задачи представлена на рисунке 1:**



Рисунок 1 – Схема алгоритма

**Текст программы с комментариями:**

**.DATA**

**.DATA?**

**inbuf DB 100 DUP (?)**

**outbuf DB 100 DUP (?)**

**allLength DD ?**

**ans DD ?**

**.CODE**

**Start:**

**; вводим строку**

**Invoke StdOut, ADDR msgEnter**

**Invoke StdIn, ADDR inbuf, LengthOf inbuf**

**; длинна всей строки**

**lea EDI, inbuf**

**mov ECX, 100**

**mov AL, 0**

**repne scasb**

**mov allLength, 100**

**sub allLength, ECX**

**dec allLength**

**; проходимся по словам**

**lea EDI, inbuf**

**mov ECX, allLength**

**; while**

**m\_while\_st:**

**cmp ECX, 0**

**jz m\_while\_end**

**; длинна текущего слова**

**mov AL, ' '**

**repne scasb**

**mov EAX, allLength**

**sub EAX, ECX**

**; уменьшаем длинну всей строки**

**sub allLength, EAX**

**push ECX**

**; находим количество E в слове**

**xor EBX, EBX; количество**

**sub EDI, EAX; адрес**

**mov ECX, EAX; счетчик**

**mov AL, 'E'**

**m\_do\_while:; do**

**repne scasb**

**; дошли до конца слова?**

**cmp ECX, 0**

**JZ m\_end**

**inc EBX ; если не дошли**

**m\_end:**

**; while**

**JNZ m\_do\_while**

**; выводим количество искомых символов**

**MOV ans, EBX**

**Invoke dwtoa, ans, ADDR outbuf; число в строку**

**Invoke StdOut, ADDR outbuf ; вывод**

**Invoke StdOut, ADDR nextS ; вывод разделителя**

**pop ECX**

**jmp m\_while\_st**

**m\_while\_end:**

**; Add you statements**

**XOR EAX, EAX**

**Invoke StdOut, ADDR MsgExit**

**Invoke StdIn, ADDR inbuf,LengthOf inbuf**

**Invoke ExitProcess, 0**

**End Start**

**Работа программы с тестовыми данными проиллюстрирована на рисунке 2, а результаты представлена в таблице 1:**

Таблица 1 – Отладка программы

|  |  |  |
| --- | --- | --- |
| Исходные данные | Ожидаемый результат | Полученный результат |
| **EEEE**  **EEE**  test**E**r  CHETSR**E**  PYAT  Shest  S**E**m  VOSeM | 4 3 1 2 0 0 1 0 | 4 3 1 2 0 0 1 0 |
| IT  IS  MY  T**E**ST  T**E**XT  WITH  **EEEEEEEE**  YAY | 0 0 0 1 1 0 8 0 | 0 0 0 1 1 0 8 0 |
| ASS**E**MBL**E**R  IS  TH**E**  B**E**ST  LANG  I  LOV**E**  IU6 | 2 0 1 1 0 0 1 0 | 2 0 1 1 0 0 1 0 |

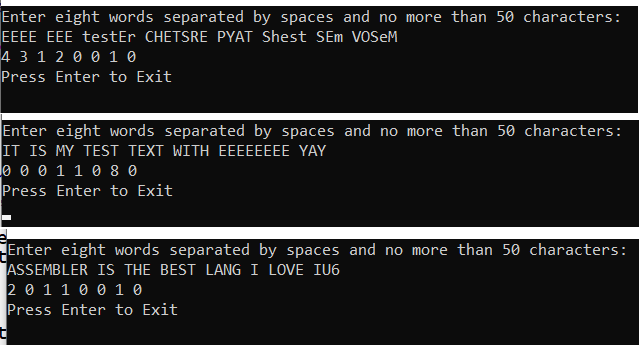


Рисунок 2 – Вывод программы при различных входных данных

**Контрольные вопросы:**

1. Дайте определение символьной строки.

Под строкой символов понимается последовательность байт идущих в памяти друг за другом.

2) Назовите основные команды обработки цепочек?

- пересылка цепочки:

movs адрес\_приемника,адрес\_источника

movsb

movsw

movsd

- сравнение цепочек:

cmps адрес\_приемника,адрес\_источника

cmpsb

cmpsw

cmpsd

- сканирование цепочки:

scas адрес\_приемника

scasb

scasw

scasd

- загрузка элемента из цепочки:

lods адрес\_источника

lodsb

lodsw

lodsd

- сохранение элемента в цепочке:

stos адрес\_приемника

stosb

stows

stosd

Какие операции выполняют строковые команды MOVS? Какие особенности характерны для этих команд?

Эта команда выполняет примитивный перенос элемента из цепочки источника в цепочку приемник и уменьшает значение ECX на 1 и увеличивает/уменьшает значение адреса источника и приемника на 1

3) Какие операции выполняют строковые команды CMPS, SCAS? Какие особенности характерны для этих команд?

CMPS – сравнение двух цепочек, поиск несовпадающих/совпадающих элементов

SCAS – сканирование цепочки, поиск нужного элемента

4) Как обеспечить циклическую обработку строк?

Поставить REP/REPZ/REPNZ перед оператором обработки строки

5) Какова роль флага DF во флаговом регистре при выполнении команд обработки строк?

Направление обработки, 0 – вперед, 1 – назад.

6) Какие макрокоманды используются в среде RADASM для ввода и вывода строк?

- movs адрес\_приемника, адрес\_источника

- cmps адрес\_приемника, адрес\_источника

- scas адрес\_приемника

- lods адрес\_источника

- stos адрес\_приемника

В зависимости от типа адреса заменяются на одну из команд указанных выше

1. Как правильно выбрать тестовые данные для проверки алгоритма обработки строки?

Тестовые данные для тестирования и отладки программы необходимо выбрать в соответствии с ее особенностями и ключевыми моментами. Также необходимо учесть все важные аспекты работы программы и граничные значения.

**Вывод:**

Я изучил команды обработки цепочек и приемы обработки символьной информации.