|  |  |
| --- | --- |
| 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 домашнего задания: Обработка символьной информации**

Студент гр. ИУ6-41Б **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Тарасова М.А.**

(Подпись, дата) (И.О. Фамилия)

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

(Подпись, дата) (И.О. Фамилия)

Москва, 2023

**Цель домашнего задания:** Научиться обрабатывать символьную информацию.

**Задание:** Дан текст 32 символа, состоящий из слов, разделенных одним пробелом. Определить количество слов и количество букв в каждом слове.

**Код программы:**

%include "lib.asm"

; %define INPUT\_LEN 33

section .data

EnterMsg db "Enter the string 32 letters long or smaller: ", 10

lenEnter equ $-EnterMsg

WordsNumber db "Words number is: "

lenWordsNumber equ $-WordsNumber

INPUT\_LEN db 33

LettersNumber db "Letters numbers are: "

lenLettersNumber equ $-LettersNumber

ExitMsg db "",10

lenExit equ $-ExitMsg

; input db "qwerty d fgh vbnmf cser ah kljwert"

section .bss

LetterCount resb 32

OutBuf resb 32

input resb 33

section .text

global \_start

\_start:

; write EnterMsg

mov rax, 1

mov rdi, 1

mov rsi, EnterMsg

mov rdx, lenEnter

syscall

mov rax, 0

mov rdi, 0

mov rsi, input

mov rdx, INPUT\_LEN

syscall

mov rcx, rax

mov ebx, 0

xor rdx, rdx ; letter counter

xor rax, rax ; words counter

dec rsi

main\_cycle:

inc rsi

inc rdx

cmp byte[rsi], ' '

je countW

cmp byte[rsi], 10

je countW

jmp next

countW:

dec rdx

inc rax

mov [ebx + LetterCount], rdx

xor rdx, rdx

inc ebx

jmp next

next:

loop main\_cycle

push rax

; выводим сообщение о количестве слов

mov rax, 1

mov rdi, 1

mov rsi, WordsNumber

mov rdx, lenWordsNumber

syscall

pop rax

push rax

; конвертируем частное из целого в строку

mov esi, OutBuf

cwd

call IntToStr64

; write

mov edx, eax

mov eax, 1

mov edi, 1

syscall

; write ExitMsg

mov rax, 1

mov rdi, 1

mov rsi, ExitMsg

mov rdx, lenExit

syscall

pop rax

push rax

; выводим сообщение о количестве букв в словах

mov rax, 1

mov rdi, 1

mov rsi, LettersNumber

mov rdx, lenLettersNumber

syscall

pop rax

push rax

mov ebx, 0

mov ecx, eax

OutputCycle:

mov eax, [ebx + LetterCount]

inc ebx

push rcx

; конвертируем частное из целого в строку

mov esi, OutBuf

cbw

call IntToStr64

; write

mov edx, eax

mov eax, 1

mov edi, 1

syscall

pop rcx

loop OutputCycle

; write ExitMsg

mov rax, 1

mov rdi, 1

mov rsi, ExitMsg

mov rdx, lenExit

syscall

;exit

mov rax, 60

xor rdi, rdi

syscall

**Результат программы представлен на рисунке 1:**

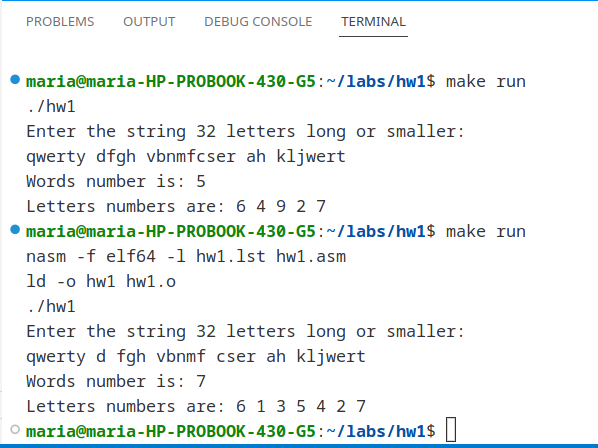
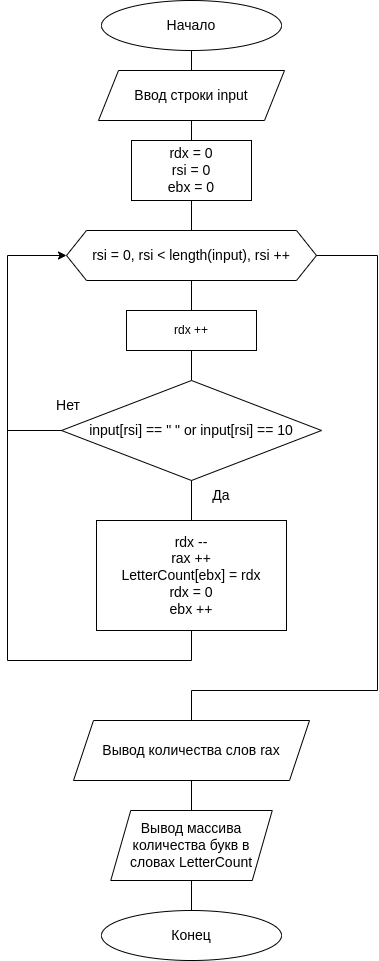


Рисунок 1 — Результат программы

**Схема алгоритма:**

****

**Вывод:** в данном домашнем задании строка разделялась на слова с параллельным подсчётом слов и букв в словах.

**Контрольные вопросы:**

1. **Дайте определение символьной строки.**

Это последовательность байтов в памяти.

1. **Назовите основные команды обработки цепочек?**

Команды обработки цепочек:

– movs – для пересылки из одной области памяти в другую;

– cmps – сравнение элементов цепочки-источника с элементами цепочки-приемника;

– scas – поиск некоторого значения в области памяти;

– lods – позволяет извлечь элемент из цепочки и загрузить регистр-аккумулятор eax/al/ax;

– stos – позволяет сохранить элемент из регистра-аккумулятора al/ax/eax в цепочке.

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

Транслятор преобразует команду movs в одну из трех машинных команд movsb, movsw и movsd. Та или иная команды выбирается от того, данные какого размера перемещала команды movs. Сама команда перемещает один элемент, но если нужно больше, то используются префиксы, такие как rep (работает до тех пор, пока ЕСХ не станет равно 0).

1. **Какие операции выполняют строковые команды CMPS, SCAS? Какие особенности характерны для этих команд?**

CMPS – команда сравнения элементов цепочек-источника с подстроками, то есть цепочки-приемника. После выхода из цикла, регистры содержат адреса элементов после тех, которые стали причиной выхода

SCAS – команда поиска некоторого значения в области памяти. Имеет один операнд, обозначающий местонахождение цепочки, который и нужно найти.

1. **Как обеспечить циклическую обработку строк?**

С помощью префиксов повторения:

– rep – работает до тех пор, пока ЕАХ не станет равной 0, используется с movs и stos;

– repe и repz – работает до тех пор, пока ЕАХ не станет равной 0 или zf не станет равным “1”, используется с cmps и scas;

– repne и repnz – работает до тех пор, пока ЕАХ не станет равной 0 или zf не станет равным “0”, используется с cmps и scas.

1. **Какова роль флага DF во флажковом регистре при выполнении команд обработки строк?**

Он будет указывать откуда начнется обработка строк. Если DF = 1, то с начала, если DF = 0, то с конца. Командами “cld” можно установить DF равный 0, а “std” DF равный 1

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

Надо выбрать те данные, которые проходили бы через весь код программы. Также можно ввести несколько частных случаев проверяющие конкретные мета программы на правильность её выполнения.