

Федеральное государственное автономное образовательное

учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки: 09.03.04 – Системное и прикладное программное обеспечение

Дисциплина «Основы профессиональной деятельности»

**Отчёт по лабораторной работе №5**

**Вариант - 1555**

Выполнил

Линейский Аким Евгеньевич

P3115

Проверил

Блохина Елена Николаевна

Санкт - Петербург 2025

# Содержание

1. [Содержание 2](#_Toc191396019)
2. [Задание 2](#_Toc191396020)
3. [Ход работы 3](#_Toc191396021)

[Текст исходной программы: 4](#_Toc191396022)

[Назначение программы и реализуемая ею функция: 4](#_Toc191396023)

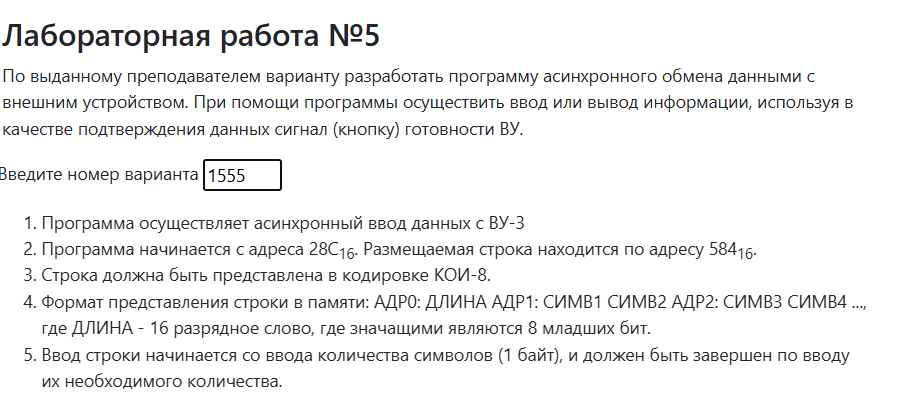
[Область представления: 4](#_Toc191396024)

[Область допустимых значений: 4](#_Toc191396025)

[Расположение в памяти ЭВМ программы, исходных данных и результатов: 5](#_Toc191396026)

1. [Трассировка программы 5](#_Toc191396027)
2. [Вывод 6](#_Toc191396028)

# Задание



# Ход работы

## Текст исходного программного комплекса:

ORG 0x28C

POINTER: WORD 0x584 ; 584 - len

; 585 - stop symbol

ITER: WORD 0

FLAG\_SYM: WORD 0x1

INPUT\_LEN:

CALL READ3

ST (POINTER)+

INC

ST ITER

CLA

CALL READ3

JUMP MAIN

FIRST\_S:

CALL READ8

CMP 0x585

BEQ SECOND\_P

ST (POINTER)

SECOND\_S:

LD ITER

DEC

ST ITER

BEQ FIRST\_P

CLA

CALL READ8

CMP 0x585

BEQ FIRST\_P

SWAB

ADD (POINTER)

MAIN:

ST (POINTER)+

LOOP ITER

JUMP FIRST\_S

JUMP PRINT\_LOOP

FIRST\_P:

LD (POINTER)

SWAB

JUMP SHOW

SECOND\_P:

LD -(POINTER)

JUMP SHOW

SHOW:

; ASR

; ASR

; ASR

; ASR

; CMP 0x1E

; BLT PRINT\_LOOP

; CMP 0x28

; BGE PRINT\_LOOP

PUSH

CALL PRINT

POP

PRINT\_LOOP:

LD POINTER

DEC

CMP 0x585

BEQ EXIT

LD FLAG\_SYM

NOT

ST FLAG\_SYM

BEQ SECOND\_P

JUMP FIRST\_P

EXIT: HLT

READ3:

IN 7

AND #0x40

BEQ READ3

IN 6

RET

READ8:

IN 0x19

AND #0x40

BEQ READ8

IN 0x18

RET

PRINT:

IN 0xD

AND #0x40

BEQ PRINT

LD $1

OUT 0xC

RET

## Описание исходного программного комплекса

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарии** |
| 28C | 584 | POINTER | Указатель адреса ячейки списка для записи |
| 28D | 0 | ITER | Счетчик цикла (по длины строки) |
| 28E | 0200 | CLA | Start, очистка AC |
| 28F | DE17 | CALL (IP+13) | Вызов подпрограммы чтения символа |
| 290 | 0700 | INC | Установка счетчика цикла |
| 291 | EEFB | ST ITER | Сохранить количество символов для ввода |
| 292 | 0740 | DEC | Восстановление длины слова |
| 293 | EAF8 | ST (POINTER)+ | Сохранение ввода в ячейку списка |
| 294 | 8EF8 | LOOP ITER | Цикл чтения символов, счетчик ITER |
| 295 | CE01 | JUMP (IP+1) | Переход к чтению ввода |
| 296 | 0100 | HLT | Завершение программы |
| 297 | AEF5 | LD ITER | Загрузка актуального ITER |
| 298 | 0740 | DEC | Вычитание 1 для сравнения |
| 299 | EEF3 | ST ITER | Сохранение значения счетчика |
| 29A | F005 | BEQ (IP+5) | Проверка осталось ил читать 1 символ |
| 29B | DE0B | CALL (IP+4) | Вызов подпрограммы чтения символа |
| 29C | 0680 | SWAB | Полученный символ -> старший байт |
| 29D | E8EE | ST (POINTER) | Сохранение 1 из 2 символов ячейки |
| 29E | DE08 | CALL (IP+1) | Вызов подпрограммы чтения символа |
| 29F | CEF3 | JUMP (IP-13) | Возвращение к цикле чтения |
| 2A0 | AEEC | LD ITER | Загрузка счетчика |
| 2A1 | 0700 | INC | Восстановление после сравнения |
| 2A2 | EEEA | ST ITER | Сохранение счетчика |
| 2A3 | A8E8 | LD (POINTER) | Считывание значения в ячейке списка |
| 2A4 | DE02 | CALL READ | Чтение символа |
| 2A5 | 0680 | SWAB | Перемещение символа в 2 байт |
| 2A6 | CEEC | JUMP MAIN | Возвращение к циклу |
| 2A7 | 1207 | IN 7 | Ожидание ввода |
| 2A8 | 2F40 | AND #40 | Проверка статуса |
| 2A9 | F0FD | BEQ (IP-3) | Спин-пул в ожидании готовности |
| 2AA | A8EA | LD (POINTER) | Чтение последней записи списка |
| 2AB | 1206 | IN 6 | Считывание символа |
| 2AC | 0A00 | RET | Возврат из подпрограммы |
|  |  |  |  |
| 584 | 0000 | LEN | Длина строки ввода |
| 585 | 0000 | ADR1 | Первый элемент списка слов |

## Назначение программы и реализуемая ею функция:

Назначение: программа асинхронного ввода данных с ВУ-3. Программа считывает с ввода количество символов строки, далее считывает символы в кодировке КОИ-8 ровно указанное количество раз (длинна строки). Программа записывает строку в память по 2 символа в ячейку (Big Endian).

## Область представления:

Переменная LEN (количество символов строки) – беззнаковое, 16-ти разрядное число в дополнительном коде [0; 216 – 1].

Символ ввода – беззнаковое, 8-ти разрядное число в дополнительном коде [00; FF].

Ячейки памяти с хранением 2х символов – беззнаковое, 16-ти разрядное число в дополнительном коде [0000; FFFF].

Счетчик цикла ITER – беззнаковое, 16-ти разрядное число в дополнительном коде [0; 216 – 1].

Указатель ячейки списка POINTER – беззнаковое, 16-ти разрядное число в дополнительном коде [0; 216 – 1].

## Область допустимых значений:

Переменная LEN: 0 ≤ LEN ≤ 28 – 1

Символ ввода: 00 ≤ S ≤ FF

Переменная счетчик ITER: 0 ≤ ITER ≤ 28 – 1

Указатель элемента списка POINTER: 0 ≤ POINTER ≤ 635

## Расположение в памяти ЭВМ программы, исходных данных и результатов:

Ячейка памяти 28E – подготовка к исполнению программы

Ячейка памяти 28F-293 – раздел программы ввода длины строки

Ячейки памяти 293-297 – цикл работы программы

Ячейка памяти 298-29F – раздел считывания 1 символа и переход для считывания 2 символа

Ячейка памяти 2A0-2A5 – раздел считывания 2 символа

Ячейка памяти 2A6-2AB – подпрограмма считывания символа

Ячейка памяти 584 – ячейка с данными о длине строки

Ячейка памяти 585 – первая ячейка списка с данными строки

## Адреса первой и последней выполняемой команд программы:

28E – адрес первой команды, начало выполнения программы

2А2 – адрес последней команды, конец выполнения программы

# Трассировка программы

## Декодировка строки

Данная строка: Ясень\*

Длинна строки ввода: 6

Код строки в кодировке КОИ-8: F1-D3-C5-CE-D8-2A

## Таблица трассировки

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Выполняемая команда | | Содержимое регистров процессора после выполнения команды | | | | | | | | | Запись в ячейку памяти | |
| Адрес | Код | IP | CR | AR | DR | SP | BR | AC | PS | NZVC | Адрес | Новый код |
| 28E | 0200 | 28F | 0200 | 28E | 0200 | 000 | 028E | 0000 | 004 | 0100 |  |  |
| 28F | DE0E | 29E | DE0E | 7FF | 0290 | 7FF | 029E | 0000 | 004 | 0100 | 7FF | 0290 |
| 29E | 1207 | 29F | 1207 | 29E | 1207 | 7FF | 029E | 0000 | 004 | 0100 |  |  |
| 29F | 2F40 | 2A0 | 2F40 | 29F | 0040 | 7FF | 0040 | 0000 | 004 | 0100 |  |  |
| 2A0 | F0FD | 2A1 | F0FD | 2A0 | F0FD | 7FF | FFFD | 0000 | 004 | 0100 |  |  |
| 2A1 | A8EA | 2A2 | A8EA | 584 | 0000 | 7FF | FFEA | 0000 | 004 | 0100 |  |  |
| 2A2 | 1206 | 2A3 | 1206 | 2A2 | 1206 | 7FF | 02A2 | 0006 | 004 | 0100 |  |  |
| 2A3 | 0A00 | 290 | 0A00 | 7FF | 0290 | 000 | 02A3 | 0006 | 004 | 0100 |  |  |
| 290 | EEFC | 291 | EEFC | 28D | 0006 | 000 | FFFC | 0006 | 004 | 0100 | 28D | 0006 |
| 291 | EAFA | 292 | EAFA | 584 | 0006 | 000 | FFFA | 0006 | 004 | 0100 | 28C | 0585 |
|  |  |  |  |  |  |  |  |  |  |  | 584 | 0006 |
| 292 | 8EFA | 293 | 8EFA | 28D | 0005 | 000 | 0004 | 0006 | 004 | 0100 | 28D | 0005 |
| 293 | CE01 | 295 | CE01 | 293 | 0295 | 000 | 0001 | 0006 | 004 | 0100 |  |  |
| 295 | AEF7 | 296 | AEF7 | 28D | 0005 | 000 | FFF7 | 0005 | 000 | 0000 |  |  |
| 296 | F005 | 297 | F005 | 296 | F005 | 000 | 0296 | 0005 | 000 | 0000 |  |  |
| 297 | 0740 | 298 | 0740 | 297 | 0740 | 000 | 0297 | 0004 | 001 | 0001 |  |  |
| 298 | EEF4 | 299 | EEF4 | 28D | 0004 | 000 | FFF4 | 0004 | 001 | 0001 | 28D | 0004 |
| 299 | DE04 | 29E | DE04 | 7FF | 029A | 7FF | 029E | 0004 | 001 | 0001 | 7FF | 029A |
| 29E | 1207 | 29F | 1207 | 29E | 1207 | 7FF | 029E | 0000 | 001 | 0001 |  |  |
| 29F | 2F40 | 2A0 | 2F40 | 29F | 0040 | 7FF | 0040 | 0000 | 005 | 0101 |  |  |
| 2A0 | F0FD | 29E | F0FD | 2A0 | F0FD | 7FF | FFFD | 0000 | 005 | 0101 |  |  |
| 2A2 | 1206 | 2A3 | 1206 | 2A2 | 1206 | 7FF | 02A2 | 00F1 | 005 | 0101 |  |  |
| 2A3 | 0A00 | 29A | 0A00 | 7FF | 029A | 000 | 02A3 | 00F1 | 005 | 0101 |  |  |
| 29A | 0680 | 29B | 0680 | 29A | 0680 | 000 | 029A | F100 | 009 | 1001 |  |  |
| 29B | E8F0 | 29C | E8F0 | 585 | F100 | 000 | FFF0 | F100 | 009 | 1001 | 585 | F100 |
| 29C | DE01 | 29E | DE01 | 7FF | 029D | 7FF | 029E | F100 | 009 | 1001 | 7FF | 029D |
| 29E | 1207 | 29F | 1207 | 29E | 1207 | 7FF | 029E | F100 | 009 | 1001 |  |  |
| 29F | 2F40 | 2A0 | 2F40 | 29F | 0040 | 7FF | 0040 | 0000 | 005 | 0101 |  |  |
| 2A0 | F0FD | 29E | F0FD | 2A0 | F0FD | 7FF | FFFD | 0000 | 005 | 0101 |  |  |
| 2A1 | A8EA | 2A2 | A8EA | 585 | F100 | 7FF | FFEA | F100 | 009 | 1001 |  |  |
| 2A2 | 1206 | 2A3 | 1206 | 2A2 | 1206 | 7FF | 02A2 | F1D3 | 009 | 1001 |  |  |
| 2A3 | 0A00 | 29D | 0A00 | 7FF | 029D | 000 | 02A3 | F1D3 | 009 | 1001 |  |  |
| 29D | CEF3 | 291 | CEF3 | 29D | 0291 | 000 | FFF3 | F1D3 | 009 | 1001 |  |  |
| 291 | EAFA | 292 | EAFA | 585 | F1D3 | 000 | FFFA | F1D3 | 009 | 1001 | 28C | 0586 |
|  |  |  |  |  |  |  |  |  |  |  | 585 | F1D3 |

# Дополнительное задание

## Задание

Написать программу, которая реализует ввод через ВУ-3 сначала кол-во символов, потом СТОП-СИМВОЛ. Далее осуществляется ввод через клавиатуру до тех, пока не закончатся символы или не встретиться СТОП слово.

После завершения ввода необходимо вывести только введенные цифры в обратном порядке на текстовый принтер.

СТОП-СЛОВО хранится как и любой другой символ.

Все остальное согласно заданию.

## Код программного комплекса

ORG     0x28C

POINTER:        WORD    0x584   *; 584 - len*

*; 585 - stop symbol*

ITER:           WORD    0

FLAG\_SYM:       WORD    0xFFFF

START\_LIST:     WORD    0x585

START:  CLA

INPUT\_LEN:

        CALL    READ3

        ST      (POINTER)+

        INC

        ST      ITER

        CLA

        CALL    READ3

        JUMP    MAIN

FIRST\_S:

        CALL    READ8

        CMP     0x585

        BEQ     SECOND\_P

        ST      (POINTER)

SECOND\_S:

        LD      ITER

        DEC

        ST      ITER

        BEQ     FIRST\_P

        CLA

        CALL    READ8

        CMP     0x585

        BEQ     FIRST\_P

        SWAB

        ADD     (POINTER)

MAIN:

        ST      (POINTER)+

        LOOP    ITER

        JUMP    FIRST\_S

        JUMP    PRINT\_LOOP

SECOND\_P:

        LD      -(POINTER)

        JUMP    SHOW

FIRST\_P:

        LD      (POINTER)

        SWAB

        JUMP    SHOW

SHOW:

        ASR

        ASR

        ASR

        ASR

        ASR

        ASR

        ASR

        ASR

        CMP     #0x30

        BLT     PRINT\_LOOP

        CMP     #0x40

        BGE     PRINT\_LOOP

        PUSH

        CALL    PRINT

        POP

PRINT\_LOOP:

        LD      POINTER

        CMP     START\_LIST

        BEQ     EXIT

        LD      FLAG\_SYM

        NOT

        ST      FLAG\_SYM

        BEQ     SECOND\_P

        JUMP    FIRST\_P

EXIT:   HLT

READ3:

        IN      7

        AND     #0x40

        BEQ     READ3

        IN      6

        RET

READ8:

        IN      0x19

        AND     #0x40

        BEQ     READ8

        IN      0x18

        RET

PRINT:

        IN      0xD

        AND     #0x40

        BEQ     PRINT

        LD      &1

        OUT     0xC

        RET

# Вывод

Проделав данную лабораторной работу №5, я ознакомился с асинхронным обменом информацией, внешними устройствами БЭВМ, на попрактиковался в написании кода на Ассемблере БЭВМ.