Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет Программной инженерии и компьютерной техники

Дисциплина “Основы профессиональной деятельности”

Отчёт по лабораторной работе №4

Вариант №2479

Выполнил:

Ануфриев Андрей Сергеевич, Р3119

Проверила:

Остапенко Ольга Денисовна

г. Санкт-Петербург

2025 год

Оглавление

[Задание 3](#_Toc191798607)

[Ход выполнения 4](#_Toc191798608)

[1) Текст исходной программы 5](#_Toc191798609)

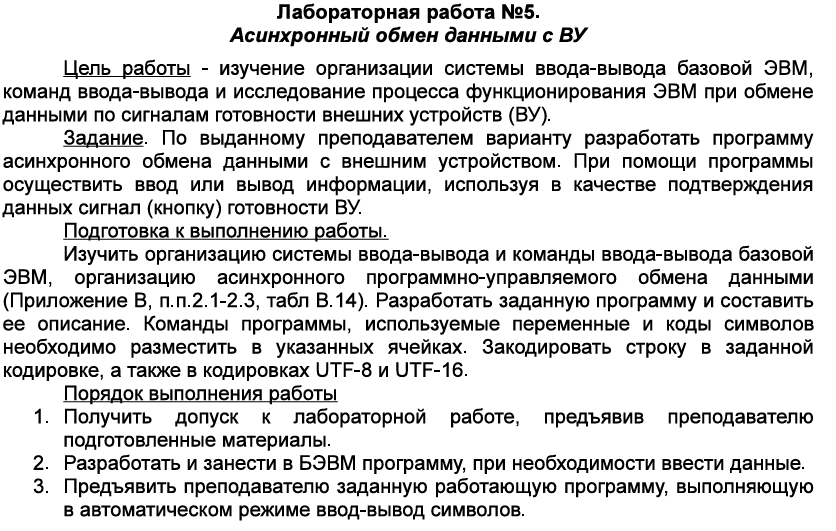
[2) Описание программы 7](#_Toc191798610)

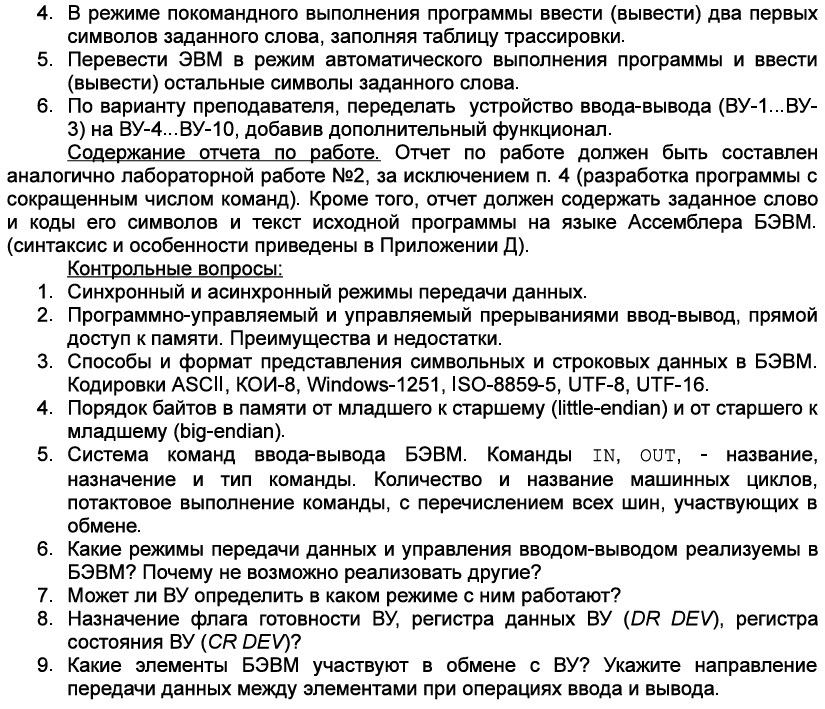
[3)Трассировка 9](#_Toc191798611)

[4)График функции и проверка результата 9](#_Toc191798612)

# Задание

1. Программа осуществляет асинхронный вывод данных на ВУ-1
2. Программа начинается с адреса 13816. Размещаемая строка находится по адресу 5B416.
3. Строка должна быть представлена в кодировке Windows-1251.
4. Формат представления строки в памяти: АДР1: СИМВ1 СИМВ2 АДР2: СИМВ3 СИМВ4 ... СТОП\_СИМВ.
5. Ввод или вывод строки должен быть завершен по символу c кодом 00 (NUL). Стоп символ является обычным символом строки и подчиняется тем же правилам расположения в памяти что и другие символы строки.





# Ход выполнения

## Текст исходной программы

|  |
| --- |
| aсинхронный вывод данных (в кодировке Windows-1251) на ВУ-1 (до 00(NUL) символа) |
| ORG 0x138 ; *Программа начинается с адреса* 138  STR: WORD 0x05B4; *Размещаемая строка находится по адресу* 5B4  POINTER: WORD ? ; *?? Указатель на текущие 2 символа* |
| START: CLA  LD STR  ST POINTER ; *Инициализируем указатель* |
| S1: IN 0x07 ; *Ожидание вывода первого символа*  AND #0x40 ; *Бит 6 SR == 0 («Готов» нажата?)*  BEQ S1 ; *Нет – «Спин-луп»*  LD (POINTER) ; *Иначе загрузим текущие 2 символа строки…*  OUT 0x00 ; *…И выведем первый символ на ВУ-1* |
| SXTB ; *Расширим знак, чтобы второй символ нам не мешал*  CMP #0x00 ; *Перед нами стоп-символ?*  BEQ PHLT ; *Да – на выход!* |
| S2: IN 0x07 ; *Ожидание вывода второго символа*  AND #0x40 ; *Бит 6 SR == 0 (“Готов” нажата?)*  BEQ S2 ; *Нет – «Спин-луп»*  LD (POINTER)+ ; *Иначе загрузим те же 2 символа строки, переместим указатель*  SWAB ; *Выберем второй символ в младшем байте…*  OUT 0x00 ; *…И выведем его на ВУ-1* |
| SXTB ; *Расширим знак, чтобы первый символ нам не мешал*  CMP #0x0000 ; *Перед нами стоп-символ?*  BEQ PHLT ; *Да – на выход!*  JUMP S1 ; *И так далее...* |
| PHLT: HLT ; *Тот самый выход!* |
| ORG 0x05B4 ; *Строка начинается здесь*  WORD 0XCFD1 ; ПС  WORD 0XC62C ; Ж,  WORD 0X20CF ; П  WORD 0XD1C6 ; СЖ  WORD 0X2E20 ; .  WORD 0XCDE0 ; На  WORD 0XEFE8 ; пи  WORD 0XE869 ; ши  WORD 0X20CF ; П  WORD 0XD1C6 ; СЖ  WORD 0X2C20 ; ,  WORD 0X20E3 ; у  WORD 0XF2E5 ; те  WORD 0XE1FF ; бя  WORD 0X20ED ; н  WORD 0XE5F2 ; ет  WORD 0X20E2 ; в  WORD 0XFBEB ; ыб  WORD 0XEEF0 ; ор  WORD 0XE021 ; а!  WORD 0X0000 ; <NUL> |

## Описание программы

**Назначение программы:** посимвольный асинхронный вывод строки, закодированной в кодировке Windows-1251 на ВУ-1.

Описание исходных данный:

* STR – Адрес 1-го элемента (начала) строки.
* STR…0A(NL-символ) – Строка в кодировке Windows-1251.

ОПИ:

* Элемент строки – до 2-х восьмиразрядных символов в кодировке Windows-1251. Младший байт – 1-й символ элемента, старший байт – 2-й.

ОДЗ:

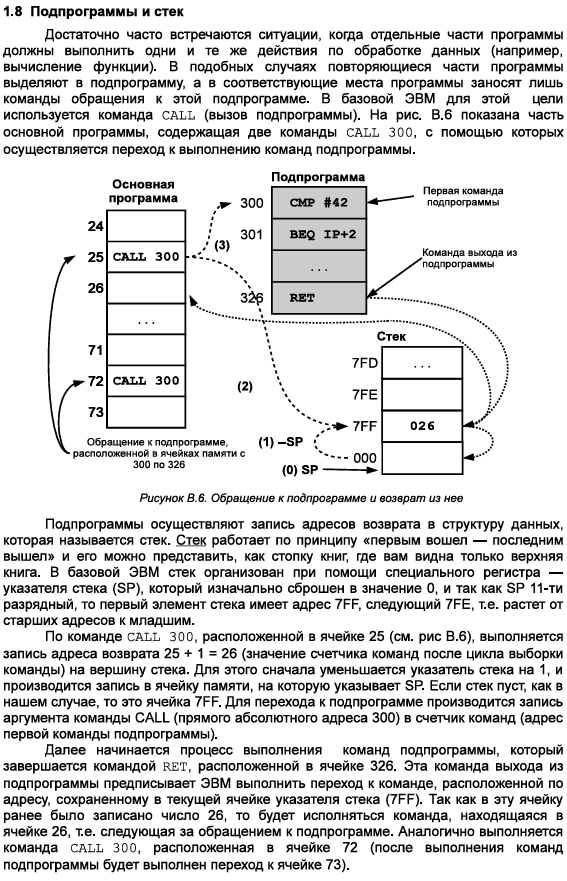
* Расположение в памяти ЭВМ программы и исходных данных:
  + Программа располагается в памяти в ячейках между адресами 138 и 15B включительно (без учета строки).
  + Исходные данные должны располагаться в ячейках памяти:
    - STR – 138.
    - Строка: начиная с адреса STR, заканчивая символом 0x00.
  + В программе также используется вспомогательная ячейка, находящаяся по адресу 0x139.
* Первая команда располагается в ячейке по адресу 13A. Последняя – 15B.

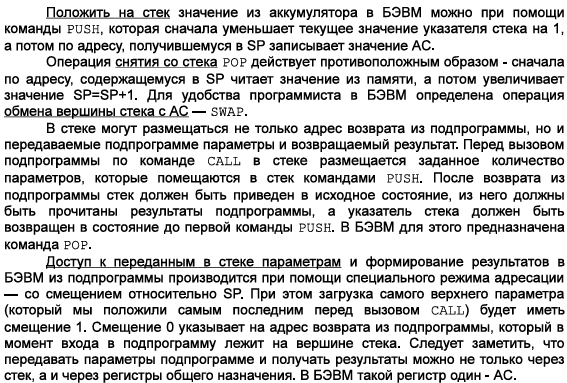
# 3)Трассировка

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Адр | Знчн | IP | CR | AR | DR | SP | BR | AC | PS | NZVC | Адр | Знчн |
| 13A | 0200 | 13A | 0000 | 000 | 0000 | 000 | 0000 | 0000 | 004 | 0100 |  |  |
| 13A | 0200 | 13B | 0200 | 13A | 0200 | 000 | 013A | 0000 | 004 | 0100 |  |  |
| 13B | AEFC | 13C | AEFC | 138 | 05B4 | 000 | FFFC | 05B4 | 000 | 0000 |  |  |
| 13C | EEFC | 13D | EEFC | 139 | 05B4 | 000 | FFFC | 05B4 | 000 | 0000 | 139 | 05B4 |
| 13D | 1203 | 13E | 1203 | 13D | 1203 | 000 | 013D | 0540 | 000 | 0000 |  |  |
| 13E | 2F40 | 13F | 2F40 | 13E | 0040 | 000 | 0040 | 0040 | 000 | 0000 |  |  |
| 13F | F0FD | 140 | F0FD | 13F | F0FD | 000 | 013F | 0040 | 000 | 0000 |  |  |
| 140 | A8F8 | 141 | A8F8 | 5B4 | CFD1 | 000 | FFF8 | CFD1 | 008 | 1000 |  |  |
| 141 | 0680 | 142 | 0680 | 141 | 0680 | 000 | 0141 | D1CF | 008 | 1000 |  |  |
| 142 | 1302 | 143 | 1302 | 142 | 1302 | 000 | 0142 | D1CF | 008 | 1000 |  |  |
| 143 | 7F00 | 144 | 7F00 | 143 | 0000 | 000 | 0000 | D1CF | 009 | 1001 |  |  |
| 144 | F008 | 145 | F008 | 144 | F008 | 000 | 0144 | D1CF | 009 | 1001 |  |  |
| 145 | 1203 | 146 | 1203 | 145 | 1203 | 000 | 0145 | D140 | 009 | 1001 |  |  |
| 146 | 2F40 | 147 | 2F40 | 146 | 0040 | 000 | 0040 | 0040 | 001 | 0001 |  |  |
| 147 | F0FD | 148 | F0FD | 147 | F0FD | 000 | 0147 | 0040 | 001 | 0001 |  |  |
| 148 | AAF0 | 149 | AAF0 | 5B4 | CFD1 | 000 | FFF0 | CFD1 | 009 | 1001 | 139 | 05B5 |
| 149 | 1302 | 14A | 1302 | 149 | 1302 | 000 | 0149 | CFD1 | 009 | 1001 |  |  |
| 14A | 7F00 | 14B | 7F00 | 14A | 0000 | 000 | 0000 | CFD1 | 009 | 1001 |  |  |
| 14B | F001 | 14C | F001 | 14B | F001 | 000 | 014B | CFD1 | 009 | 1001 |  |  |
| 14C | CEF0 | 13D | CEF0 | 14C | 013D | 000 | FFF0 | CFD1 | 009 | 1001 |  |  |
| 13D | 1203 | 13E | 1203 | 13D | 1203 | 000 | 013D | CF40 | 009 | 1001 |  |  |
| 13E | 2F40 | 13F | 2F40 | 13E | 0040 | 000 | 0040 | 0040 | 001 | 0001 |  |  |
| 13F | F0FD | 140 | F0FD | 13F | F0FD | 000 | 013F | 0040 | 001 | 0001 |  |  |
| 140 | A8F8 | 141 | A8F8 | 5B5 | C62C | 000 | FFF8 | C62C | 009 | 1001 |  |  |
| 141 | 0680 | 142 | 0680 | 141 | 0680 | 000 | 0141 | 2CC6 | 001 | 0001 |  |  |
| 142 | 1302 | 143 | 1302 | 142 | 1302 | 000 | 0142 | 2CC6 | 001 | 0001 |  |  |
| 143 | 7F00 | 144 | 7F00 | 143 | 0000 | 000 | 0000 | 2CC6 | 001 | 0001 |  |  |
| 144 | F008 | 145 | F008 | 144 | F008 | 000 | 0144 | 2CC6 | 001 | 0001 |  |  |
| 145 | 1203 | 146 | 1203 | 145 | 1203 | 000 | 0145 | 2C40 | 001 | 0001 |  |  |
| 146 | 2F40 | 147 | 2F40 | 146 | 0040 | 000 | 0040 | 0040 | 001 | 0001 |  |  |
| 147 | F0FD | 148 | F0FD | 147 | F0FD | 000 | 0147 | 0040 | 001 | 0001 |  |  |
| 148 | AAF0 | 149 | AAF0 | 5B5 | C62C | 000 | FFF0 | C62C | 009 | 1001 | 139 | 05B6 |
| 149 | 1302 | 14A | 1302 | 149 | 1302 | 000 | 0149 | C62C | 009 | 1001 |  |  |
| 14A | 7F00 | 14B | 7F00 | 14A | 0000 | 000 | 0000 | C62C | 009 | 1001 |  |  |
| 14B | F001 | 14C | F001 | 14B | F001 | 000 | 014B | C62C | 009 | 1001 |  |  |
| 14C | CEF0 | 13D | CEF0 | 14C | 013D | 000 | FFF0 | C62C | 009 | 1001 |  |  |
| 13D | 1203 | 13E | 1203 | 13D | 1203 | 000 | 013D | C640 | 009 | 1001 |  |  |
| 13E | 2F40 | 13F | 2F40 | 13E | 0040 | 000 | 0040 | 0040 | 001 | 0001 |  |  |
| 13F | F0FD | 140 | F0FD | 13F | F0FD | 000 | 013F | 0040 | 001 | 0001 |  |  |
| 140 | A8F8 | 141 | A8F8 | 5B6 | 0000 | 000 | FFF8 | 0000 | 005 | 0101 |  |  |
| 141 | 0680 | 142 | 0680 | 141 | 0680 | 000 | 0141 | 0000 | 005 | 0101 |  |  |
| 142 | 1302 | 143 | 1302 | 142 | 1302 | 000 | 0142 | 0000 | 005 | 0101 |  |  |
| 143 | 7F00 | 144 | 7F00 | 143 | 0000 | 000 | 0000 | 0000 | 005 | 0101 |  |  |
| 144 | F008 | 14D | F008 | 144 | F008 | 000 | 0008 | 0000 | 005 | 0101 |  |  |
| 14D | 0100 | 14E | 0100 | 14D | 0100 | 000 | 014D | 0000 | 005 | 0101 |  |  |

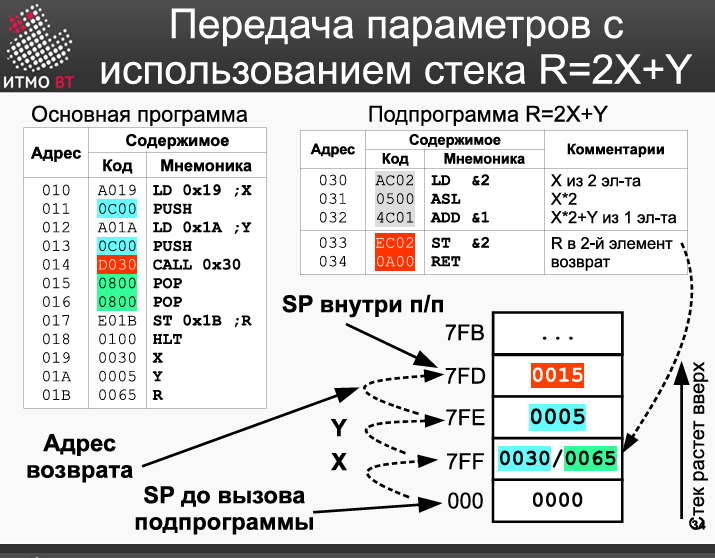
# Ответы на вопросы

1. Организация подпрограмм в БЭВМ. Команды вызова подпрограммы и возврата. Недостатки существующей реализации.





1. Аргументы и возвращаемые значения подпрограммы. Способы организации передачи аргументов и возвращаемых значений.



1. Рекурсивный вызов подпрограмм. Организация стека.
2. Описание команды CALL и RET: наименование, назначение, тип команды и вид адресации. Количество и название машинных циклов, потактовое выполнение команды, количество обращений к памяти.

