Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики»

**факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА № 5**

по дисциплине

‘Основы профессиональной деятельности’

Вариант № 8611

*Выполнил:*

Студент группы P3115

Собитов Анвархон А.

*Преподаватель:*

*Абузов Ярослав А.*



Санкт-Петербург, 2023

Текст задания

По выданному преподавателем варианту разработать программу асинхронного обмена данными с внешним устройством. При помощи программы осуществить ввод или вывод информации, используя в качестве подтверждения данных сигнал (кнопку) готовности ВУ.

1. Программа осуществляет асинхронный ввод данных с ВУ-3
2. Программа начинается с адреса 46716. Размещаемая строка находится по адресу 59016.
3. Строка должна быть представлена в кодировке КОИ-8.
4. Формат представления строки в памяти: АДР1: СИМВ2 СИМВ1 АДР2: СИМВ4 СИМВ3 ... 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.

# Описание программы

Передаваемое сообщение: ИТМО

* КОИ-8: C9 D2 CC CE
* UTF-8: D0 98 D0 A2 D0 9C D0 9E
* UTF-16: 0418 0422 041C 041E
* WINDOWS-1251: C8 D2 CC CE
* ISO-8859-5: D8 D4 CC D5

## Текст программы на ассемблере

ORG 0x467 *; Адрес начала программы*

res: **WORD** 0x590 *; Ссылка на результат*

finish: **WORD** 0x00 *; Стоп-символ*

temp: **WORD** ? *; Ячейка для записи нечетных символов*

**START**: CLA *; Очистка аккумулятора*

s1: **IN** 7 *; Ожидание ввода нечетного символа*

**AND** #0x40 *; Проверка на наличие введенного символа*

BEQ s1 *; Нет - "Спин-луп"*

**IN** 6 *; Вывод байта в AC*

**ST** **(**res**)** *; Сохраняем символ в результат*

**ST** temp *; Сохраняем символ во временную переменную*

**CMP** finish *; Проверяем на стоп-символ*

BEQ exit *; Если стоп-символ - выход*

CLA *; Очистка аккумулятора*

s2: **IN** 7 *; Ожидание ввода четного символа*

**AND** #0x40 *; Проверка на наличие введенного символа*

BEQ s2 *; Нет - "Спин-луп"*

**IN** 6 *; Вывод байта в AC*

SWAB *; Перемещаем четный символ в старший байт*

**OR** temp *; Совмещаем с 1-м символом*

**ST** **(**res**)** *; Сохраняем в память по ссылки*

**SUB** temp *; Вычитаем 1-й символ*

SWAB *; Перемещаем четный символ в младший байт*

**CMP** finish *; Проверяем на стоп-символ*

BEQ exit *; Если стоп-символ - выход*

LD **(**res**)**+ *; Инкрементируем ссылку на результат*

CLA *; Очистка аккумулятора*

JUMP s1 *; Возвращаемся в начало цикла*

exit: LD **(**res**)**+ *; Инкрементируем ссылку на результат*

**HLT** *; Остановка программы*

**Текст исходной программы:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарии** |
| 467 | 0562 | res | Ссылка на результат |
| 468 | 000A | finish | Стоп-символ |
| 469 | 0000 | temp | Ячейка для записи нечетных символов |
| 46A | + 0200 | CLA | Очистка аккумулятора |
| 46B | 1207 | IN 7 | Чтение регистра состояния ВУ-3 |
| 46C | 2F40 | AND #0x40 | Проверка на наличие введенного символа |
| 46D | F0FD | BEQ IP-3 | Нет - "Спин-луп" |
| 46E | 1206 | IN 6 | Чтение регистра данных ВУ-3 |
| 46F | E8F7 | ST (IP-9) | Сохраняем символ в результат |
| 470 | EEF8 | ST IP-8 | Сохраняем символ во временную переменную |
| 471 | 7EF6 | CMP IP-10 | Проверяем на стоп-символ |
| 472 | F00F | BEQ IP+15 | Если стоп-символ - выход |
| 473 | 0200 | CLA | Очистка аккумулятора |
| 474 | 1207 | IN 7 | Чтение регистра состояния ВУ-3 |
| 475 | 2F40 | AND #0x40 | Проверка на наличие введенного символа |
| 476 | F0FD | BEQ IP-3 | Нет - "Спин-луп" |
| 477 | 1206 | IN 6 | Чтение регистра данных ВУ-3 |
| 478 | 0680 | SWAB | Обмен старшего и младшего байтов |
| 479 | 3EEF | OR IP-17 | Логическое или ^(^MEM & ^AC) → AC |
| 47A | E8EC | ST (IP-20) | Сохраняем в память по ссылки |
| 47B | 6EED | SUB IP-19 | Вычитание AC – MEM → AC |
| 47C | 0680 | SWAB | Обмен старшего и младшего байтов |
| 47D | 7EEA | CMP IP-22 | Проверяем на стоп-символ |
| 47E | F003 | BEQ IP+3 | Если стоп-символ - выход |
| 47F | AAE7 | LD (IP-25)+ | Инкрементируем ссылку на результат |
| 480 | 0200 | CLA | Очистка аккумулятора |
| 481 | CEE9 | JUMP IP-23 | Возвращаемся в начало цикла |
| 482 | AAE4 | LD (IP-30)+ | Инкрементируем ссылку на результат |
| 483 | 0100 | HLT | Остановка программы |
| … | | | |
| 590 | 0000 | 0000 | Строка результата |

**Область представления**

* res – 11-разрядная ячейка со ссылкой на результат.
* finish – 16-разрядная константа.
* temp – 16-разрядная ячейка для временного хранения введенных символов.
* 590- ? – 16-разрядные ячейки, хранящие в себе по два символа в кодировке Windows-1251.

**Расположение данных в памяти**

* 46A-483 – команды;
* 467, 468, 469– исходные данные;
* 590 - ? – результат.

**Адреса первой и последней выполняемой команды**

* Адрес первой команды: 46A
* Адрес последней команды: 483

**Область допустимых значений**

* res (указатель на ячейки массива, хранящий результат ввода) ∈ [562;2047]
* temp (ячейка для записи нечетных символов) ∈ [0;255], т.к. в нее записывается только 1 символ из 8 бит.
* Введенный символ: [00; FF]

Адрес первого элемента массива равен 590 по условию. Т.к. 2047 – 590 =1457 – кол-во ячеек, которые могут использоваться для записи результата => 1457\*2 = 2914 – максимально возможное кол-во введенных символов (т.к. в данной кодировке символ занимает 1 байт), включая обязательный стоп-символ => Кол-во введенных символов ∈ [1;2970].

# Вывод

При выполнении данной лабораторной работы я познакомился с асинхронным вводом-выводом данных в БЭВМ, узнал о внешних устройствах, их регистрах и принципах работы. Также, я познакомился с представлением данных в различных кодировках и попрактиковался с вводом данных на ВУ-3.