

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И
КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине

«Основы профессиональной деятельности»

Асинхронный обмен данными с ВУ

Вариант № 14228

Выполнил:

Студент группы Р3107

Чусовлянов Максим Сергеевич

Принял:

Вербовой Александр Александрович

Содержание

Задание	3
Описание программы	4
Текст исходной программы	5
Описание программы	6
Таблица трассировки	7
Заключение	9

Задание

Лабораторная работа №5

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.

Введите номер варианта

1. Программа осуществляет асинхронный ввод данных с ВУ-3
2. Программа начинается с адреса 068_{16} . Размещаемая строка находится по адресу $5B9_{16}$.
3. Строка должна быть представлена в кодировке ISO-8859-5.
4. Формат представления строки в памяти: АДР1: СИМВ2 СИМВ1 АДР2: СИМВ4 СИМВ3 ... СТОП_СИМВ.
5. Ввод или вывод строки должен быть завершен по символу с кодом 0D (CR). Стоп символ является обычным символом строки и подчиняется тем же правилам расположения в памяти что и другие символы строки.

Описание программы

```
1  ORG 0x010
2  RES_ADDRESS: WORD $RES          ; current string address
3  EOF: WORD 0x0d                  ; EOF char
4  TMP: WORD ?                     ; tmp
5
6  ORG 0x068
7  START:      CLA                  ; clear
8
9  ▾ FIRST_SYMBOL: IN 7              ; \
10             AND #0x40            ; -> spin-loop while not ready
11             BEQ FIRST_SYMBOL     ; /
12
13             IN 6                  ; \
14             SWAB                  ; -> read char, save to first byte of the cell
15             ST (RES_ADDRESS)     ; /
16
17             SWAB                  ; \
18             CMP EOF              ; -> if eof then stop
19             BEQ STOP             ; /
20
21  ▾ SECOND_SYMBOL: IN 7            ; \
22             AND #0x40            ; -> spin-loop while not ready
23             BEQ SECOND_SYMBOL    ; /
24
25             IN 6                  ; \
26             ST TMP               ; | -> read char, save to second byte of the cell
27             ADD (RES_ADDRESS)    ; |
28             ST (RES_ADDRESS)+    ; /
29
30             LD TMP               ; \
31             CMP EOF              ; -> if eof then stop
32             BEQ STOP             ; /
33
34             JUMP FIRST_SYMBOL    ; while (true)
35
36  STOP:      HLT                  ; exit programm
37
38  ORG 0x5B9
39  RES: WORD ?                     ; string beginning
```

Код доступен на <https://github.com/Vaneshik/VT-Labs/tree/main/opd/lab5>

Текст исходной программы

Адрес	Код команды	Мнемоника	Комментарии
010	-	RES_ADDRESS	Адрес текущей ячейки
011	-	EOF	Символ конца ввода
012	-	TMP	переменная для хранения временных данных
068z	0200	CLA	Очистить аккумулятор
069	1207	IN #7	Ожидание ввода
06A	2F40	AND #0x40	Проверка статуса
06B	F0FD	BEQ IP-3	Спин-луп в ожидании готовности
06C	1206	IN #6	Считывание символа
06D	0680	SWAB	Меняем старший и младший байты
06E	E8A1	ST (RES_ADDRESS)	Сохраняем в текущей ячейке
06F	0680	SWAB	Меняем обратно для проверки EOF
070	7EA0	CMP EOF	Проверяем на наличие EOF
071	F00B	BEQ IP+11	Если EOF, то завершаем
072	1207	IN #7	Ожидание ввода
073	2F40	AND #0x40	Проверка статуса
074	F0FD	BEQ IP-3	Спин-луп в ожидании готовности
075	1206	IN #6	Считывание символа
076	EE9B	ST TMP	Сохраним во временной переменной
077	4898	ADD (RES_ADDRESS)	Добавим значение предыдущего символа
078	EA97	ST (RES_ADDRESS)+	Сохраним 2 символа в текущей ячейке
079	AE98	LD TMP	Загружаем в аккумулятор временную переменную
07A	7E96	CMP EOF	Проверяем на наличие EOF
07B	F001	BEQ IP+1	Если EOF, то завершаем
07C	CEEC	JUMP IP-20	Прыжок на ячейку 069
07D	0100	HLT	Завершение программы
5B9	-	RES	Строка результата

Описание программы

Программа осуществляет посимвольный асинхронный ввод данных с ВУ-3. Программа будет получать символы до тех пор, пока на ВУ-3 не будет введен стоп-символ (EOF) с кодировкой 0x0D, который она запишет в память и прекратит свое выполнение.

Область представления:

RES -? - 16-разрядные ячейки, хранящие в себе по два символа в кодировке Windows-1251

RES_ADDRESS - 11 разрядная ячейка, хранящая адрес текущей ячейки

EOF - 16-разрядная константа.

TMP - 16-разрядная ячейка для временного хранения введенных символов.

Область определения:

- RES_ADDRESS (указатель на ячейки массива, хранящий результат ввода) \in [5B9;2047]
- EOF: [00; FF]

Таблица трассировки

Строка для трассировки: «тетрагидроканнабинол».

ISO-8859-5: E2 D5 E2 E0 D0 D3 D8 D4 E0 DE DA D0 DD DD D0 D1 D8 DD DE DB

UTF-8: D1 82 D0 B5 D1 82 D1 80 D0 B0 D0 B3 D0 B8 D0 B4 D1 80 D0 BE D0 BA D0 B0 D0 BD D0 BD D0 B0 D0 B1 D0 B8 D0 BD D0 BE D0 BB

UTF-16: FF FE 42 04 35 04 42 04 40 04 30 04 33 04 38 04 34 04 40 04 3E 04 3A 04 30 04 3D 04 3D 04 30 04 31 04 38 04 3D 04 3E 04 3B 04

Выполняемая команда		Содержимое регистров процессора после выполнения команды									Ячейка, содержимое которой изменилось после выполнения команды	
Адрес	Код команды	IP	CR	AR	DR	SP	BR	AC	PS	NZVC	Адрес	Новый код
068	0200	069	0200	068	0200	000	0068	0000	004	0100		
069	1207	06A	1207	069	1207	000	0069	0040	004	0100		
06A	2F40	06B	2F40	06A	0040	000	0040	0040	000	0000		
06B	F0FD	06C	F0FD	06B	F0FD	000	006B	0040	000	0000		
06C	1206	06D	1206	06C	1206	000	006C	00E2	000	0000		
06D	0680	06E	0680	06D	0680	000	006D	E200	008	1000		
06E	E8A1	06F	E8A1	5B9	E200	000	FFA1	E200	008	1000	5B9	E200
06F	0680	070	0680	06F	0680	000	006F	00E2	000	0000		
070	7EA0	071	7EA0	011	000D	000	FFA0	00E2	001	0001		
071	F00B	072	F00B	071	F00B	000	0071	00E2	001	0001		
072	1207	073	1207	072	1207	000	0072	0040	005	0101		
073	2F40	074	2F40	073	0040	000	0040	0040	001	0001		
074	F0FD	075	F0FD	074	F0FD	000	0074	0040	001	0001		
075	1206	076	1206	075	1206	000	0075	00D5	001	0001		
076	EE9B	077	EE9B	012	00D5	000	FF9B	00D5	001	0001	012	00D5
077	4898	078	4898	5B9	E200	000	FF98	E2D5	008	1000		

078	EA97	079	EA97	5B9	E2D 5	000	FF97	E2D5	008	1000	010, 5B9	05BA, E2D5
079	AE98	07A	AE98	012	00D5	000	FF98	00D5	000	0000		
07A	7E96	07B	7E96	011	000D	000	FF96	00D5	001	0001		
07B	F001	07C	F001	07B	F001	000	007B	00D5	001	0001		
07C	CEEC	069	CEEC	07C	0069	000	FFEC	00D5	001	0001		
069	1207	06A	1207	069	1207	000	0069	0000	001	0001		
06A	2F40	06B	2F40	06A	0040	000	0040	0000	005	0101		
06B	F0FD	069	F0FD	06B	F0FD	000	FFFD	0000	005	0101		
069	1207	06A	1207	069	1207	000	0069	0000	005	0101		
06A	2F40	06B	2F40	06A	0040	000	0040	0000	005	0101		
06B	F0FD	069	F0FD	06B	F0FD	000	FFFD	0000	005	0101		
069	1207	06A	1207	069	1207	000	0069	0000	005	0101		
06A	2F40	06B	2F40	06A	0040	000	0040	0000	005	0101		
06B	F0FD	069	F0FD	06B	F0FD	000	FFFD	0000	005	0101		
069	1207	06A	1207	069	1207	000	0069	0040	005	0101		
06A	2F40	06B	2F40	06A	0040	000	0040	0040	001	0001		
06B	F0FD	06C	F0FD	06B	F0FD	000	006B	0040	001	0001		
06C	1206	06D	1206	06C	1206	000	006C	000D	001	0001		
06D	0680	06E	0680	06D	0680	000	006D	0D00	001	0001		
06E	E8A1	06F	E8A1	5BA	0D00	000	FFA1	0D00	001	0001	5BA	0D00
06F	0680	070	0680	06F	0680	000	006F	000D	001	0001		
070	7EA0	071	7EA0	011	000D	000	FFA0	000D	005	0101		
071	F00B	07D	F00B	071	F00B	000	000B	000D	005	0101		
07D	0100	07E	0100	07D	0100	000	007D	000D	005	0101		

Заключение

В ходе проделанной лабораторной работы, я познакомился с асинхронным вводом-выводом данных в БЭВМ, узнал о внешних устройствах и их регистрах. Также познакомился с представлением данных в различных кодировках и попрактиковался в написания кода на Ассемблере БЭВМ.

