

Санкт-Петербургский Национальный Исследовательский Университет  
Информационных Технологий, Механики и Оптики

Факультет программной инженерии и компьютерной техники

Лабораторная работа №5  
по дисциплине  
«Основы профессиональной деятельности»

Выполнил: Студент группы Р3233  
Сабитов Данил Тимурович

Преподаватель:  
Блохина Елена Николаевна

Санкт-Петербург  
2021 г.

### Задание: (4135 вариант):

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.

### Функционал программы:

1. Программа осуществляет асинхронный ввод данных с ВУ-9 и вывод введенных ранее данных на ВУ-7.
2. Программа начинается с адреса 279<sub>16</sub>. Размещаемая строка находится по адресу 62E<sub>16</sub>.
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 (+) код В. Стоп символ является обычным символом строки и подчиняется тем же правилам расположения в памяти что и другие символы строки.

### Сообщение: «Декабрь:»

В кодировке Windows-1251: C4E5 EAE0 E1F0 FC3A

В кодировке UTF-8: D094 D0B5 D0BA D0B0 D0B1 D180 D18C 003A

В кодировке UTF-16: 0414 0435 043A 0430 0431 0440 044C 003A

### Текст программы на Ассемблере

СТОЛБЕЦ 1	СТОЛБЕЦ 2	СТОЛБЕЦ 3	СТОЛБЕЦ 4
ORG 0x279	BMI WAIT	SUB #0x10	HLT
START: CLA	ST MEM	BMI WAIT	EXIT2:
S1: IN 0x1D	CLA	ST MEM	POP
AND #0x40	S2 :IN 0x1D	CLA	IN 0x1C
BEQ S1	AND #0x40	CNT: JUMP S1	OUT 0x14
IN 0x1C	BEQ S1	WAIT: IN 0x1D	ST (ADDR)+
CMP #0x0B	IN 0x1C	AND #0x40	HLT
BEQ EXIT	CMP #0x0B	BEQ WAIT	EXIT3: HLT
ADD MEM	BEQ EXIT2	IN 0x1C	MEM: WORD 0x70
OUT 0x14	POP	CMP #0x0B	ADDR: WORD \$addr
ST (ADDR)	IN 0x1C	BEQ EXIT3	ORG 0x62E
SWAB	ADD MEM	JUMP WAIT	addr: NOP
PUSH	OUT 0x14	EXIT:	
LD MEM	ST (ADDR)+	OUT 0x14	
SUB #0x10	LD MEM	ST (ADDR)+	

### Текст исходной программы:

Адрес	Код команды	Мнемоника	Комментарий
279	+0200	CLA	Очистка аккумулятора
27A	121D	IN 0x1D	Проверка регистра состояния ВУ-9

27B	2F40	AND #0x40	Логическое умножение AC на 0x40: M & AC -> AC
27C	F0FD	BEQ (IP – 3)	Переход (при равенстве Z==1) в ячейку IP – 3 = 27A
27D	121C	IN 0x1C	Проверка регистра данных ВУ-9
27E	7F0B	CMP #0B	Сравнение со значением #0B (кода символа на завершение строки)
27F	F022	BEQ (IP + 22)	Переход (при равенстве Z==1) в ячейку IP + 22 = 2A2
280	4E2A	ADD 2AB	Прямая относительная: ST IP + 2A Сложение M + AC -> AC
281	1314	OUT 0x14	Значение аккумулятора записывается в регистр данных ВУ-7
282	E829	ST (IP + 29)	Косвенная адресация: (IP+29) Загрузка M->AC
283	0680	SWAB	Обмен старшего и младшего байтов
284	0C00	PUSH	Запись значения стека в AC
285	AE25	LD 2AB	Прямая относительная: ST IP + 25 Загрузка M->AC
286	6F10	SUB #10	Вычитание AC – M -> AC
287	F213	BMI (IP + 13)	Переход (при равенстве N==1) в ячейку IP + 13 = 29B
288	EE22	ST 2AB	Прямая относительная: (IP + 22) Сохранение AC->M
289	0200	CLA	Очистка аккумулятора
28A	121D	IN 0x1D	Проверка регистра состояния ВУ-9
28B	2F40	AND #0x40	Логическое умножение AC на 0x40: M & AC -> AC
28C	F0E2	BEQ (IP – 3)	Переход (при равенстве Z==1) в ячейку IP – 3 = 28A
28D	121C	IN 0x1C	Проверка регистра данных ВУ-9
28E	7F0B	CMP #0B	Сравнение со значением #0B (кода символа на завершение строки)
28F	F015	BEQ (IP + 15)	Переход (при равенстве Z==1) в ячейку IP + 15 = 2A5
290	0800	POP	Чтение значение стека в AC
291	121C	IN 0x1C	Проверка регистра данных ВУ-9
292	4E18	ADD 2AB	Прямая относительная: ST IP + 18 Сложение M + AC -> AC
293	1314	OUT 0x14	Значение аккумулятора записывается в регистр данных ВУ-7
294	EA17	ST (IP + 17)+	Косвенная автоинкрементная: (IP + 17)+ Сохранение AC->M
295	AE15	LD 2AB	Прямая относительная: ST IP + 15 Загрузка M->AC
296	6F10	SUB #10	Вычитание AC – M -> AC
297	F203	BMI (IP + 3)	Переход (при равенстве N==1) в ячейку IP + 13 = 29B
298	EE12	ST 2AB	Прямая относительная: (IP + 12) Сохранение AC->M
299	0200	CLA	Очистка аккумулятора
29A	CEDF	JUMP 27A	Прямая относительная: IP - 21 Переход к адресу 27A
29B	121D	IN 0x1D	Проверка регистра состояния ВУ-9
29C	2F40	AND #0x40	Логическое умножение AC на 0x40: M & AC -> AC
29D	F0FD	BEQ (IP – 3)	Переход (при равенстве Z==1) в ячейку IP – 3 = 29B

29E	121C	IN 0x1C	Проверка регистра данных ВУ-9
29F	7F0B	CMP #0B	Сравнение со значением #0B (кода символа на завершение строки)
2A0	F009	BEQ (IP + 9)	Переход (при равенстве Z=1) в ячейку IP + 9 = 2AA
2A1	CEF9	JUMP 29B	Прямая относительная: IP - 7 Переход к адресу 27B
2A2	1314	OUT 0x14	Значение аккумулятора записывается в регистр данных ВУ-7
2A3	EA08	ST (IP + 8)+	Косвенная автоинкрементная: (IP + 8)+ Сохранение AC->M
2A4	0100	HLT	Остановка программы
2A5	0800	POP	Чтение значение стека в AC
2A6	121C	IN 0x1C	Проверка регистра данных ВУ-9
2A7	1314	OUT 0x14	Значение аккумулятора записывается в регистр данных ВУ-7
2A8	EA03	ST (IP + 3)+	Косвенная автоинкрементная: (IP + 3)+ Сохранение AC->M
2A9	0100	HLT	Остановка программы
2AA	0100	HLT	Остановка программы

### Описание программы:

Программа осуществляет асинхронный вывод сообщения «Декабрь:» на ВУ-1 в кодировке Windows-1251. Вывод заканчивается на стоп-символе (0x00).

### Область представления:

279-2AA – 16-ти разрядные числа, команды.

2AC – 16-ти разрядное число, хранящее в себе адрес (addr).

2AB – 16-ти разрядное число, хранящее в себе счетчик.

62E-? – 16-ти разрядные числа, хранящие в себе по два символа.

### Область допустимых значений:

ADDR – указатель на ячейки массива. ADDR  $\in$  [62E<sub>16</sub>; 7FF<sub>16</sub>]

MEM – счетчик. MEM  $\in$  [0; 7]

Кол-во введенных символов  $\in$  [0 ; 930] , так как 7FF<sub>16</sub> – 62E<sub>16</sub> = 1D1<sub>16</sub> = 465<sub>10</sub> – кол-во ячеек, которые могут использоваться для записи. 465 \* 2 = 930 – максимальное кол-во символов, которое можно записать при использовании кодировки, где символ занимает 1 байт.

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

- **Расположение программы:** 279 – 2AA
- **Расположение исходных данных:** 2AB – 2AC
- **Результат:** 62E – ?
- Расположение указателя на адрес: 2AC
- Расположение счетчика: 2AB

**Адреса первой и последней выполняемой команд программы и подпрограммы:**

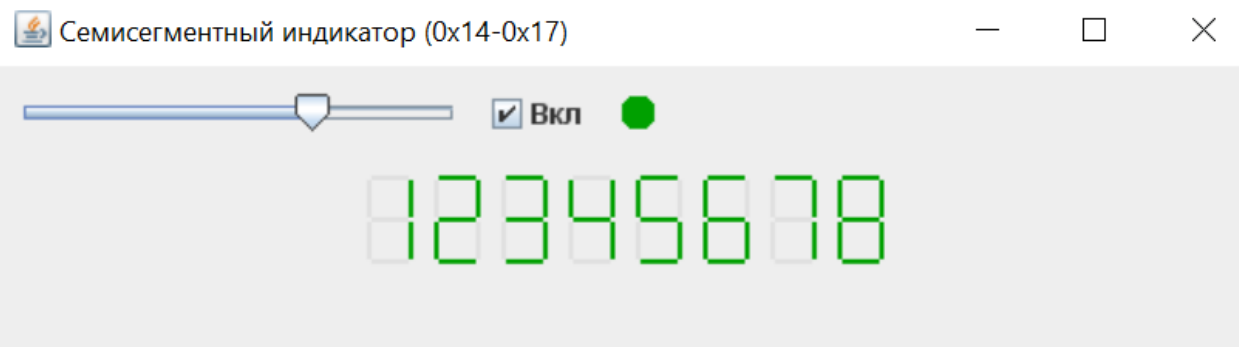
Первая выполняемая команда программы располагается в ячейке 279, последняя— в ячейке 2AA.

**Таблица трассировки (для первых двух символов):**

Выполняемая команда		Содержание регистров процессора после выполнения команды								Ячейка, содержимое которой изменилось после выполнения команды	
Адр.	Знач.	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Новый код
279	0200	279	0000	000	0000	000	0000	0000	0100		
279	0200	27A	0200	279	0200	000	0279	0000	0100		
27A	121D	27B	121D	27A	121D	000	027A	0040	0100		
27B	2F40	27C	2F40	27B	0040	000	0040	0040	0000		
27C	F0FD	27D	F0FD	27C	F0FD	000	027C	0040	0000		
27D	121C	27E	121C	27D	121C	000	027D	0002	0000		
27E	7F0B	27F	7F0B	27E	000B	000	000B	0002	1000		
27F	F022	280	F022	27F	F022	000	027F	0002	1000		
280	4E2A	281	4E2A	2AB	0070	000	002A	0072	0000		
281	1314	282	1314	281	1314	000	0281	0072	0000		
282	E829	283	E829	62E	0072	000	0029	0072	0000	62E	0072
283	0680	284	0680	283	0680	000	0283	7200	0000		
284	0C00	285	0C00	7FF	7200	7FF	0284	7200	0000	7FF	7200
285	AE25	286	AE25	2AB	0070	7FF	0025	0070	0000		
286	6F10	287	6F10	286	0010	7FF	0010	60	0001		
287	F213	288	F213	287	F213	7FF	0287	0060	0001		
288	EE22	289	EE22	2AB	0060	7FF	0022	0060	0001	2AB	0060
289	0200	28A	0200	289	0200	7FF	0289	0000	0101		
28A	121D	28B	121D	28A	121D	7FF	028A	0040	0101		
28B	2F40	28C	2F40	28B	0040	7FF	0040	0040	0001		
28C	F0ED	28D	F0ED	28C	F0ED	7FF	028C	0040	0001		
28D	121C	28E	121C	28D	121C	7FF	028D	0005	0001		
28E	7F0B	28F	7F0B	28E	000B	7FF	000B	0005	1000		

28F	F015	290	F015	28F	F015	7FF	028F	0005	1000		
290	0800	291	0800	7FF	7200	000	0290	7200	0000		
291	0800	291	0800	7FF	7200	000	0290	7200	0000		
292	121C	292	121C	291	121C	000	0291	7205	0000		
293	4E18	293	4E18	2AB	0060	000	0018	7265	0000		

### Скриншот ВУ-7:



### Вывод:

В результате проделанной лабораторной работы я узнал об асинхронном вводе-выводе данных в БЭВМ, познакомился с внешними устройствами ввода-вывода их регистрах. Также я узнал о представлении данных в различных кодировках.