

Федеральное государственное автономное  
образовательное учреждение высшего образования  
Национальный Исследовательский Университет ИТМО

## Лабораторная работа №5.

Дисциплина: Основы профессиональной деятельности  
Вариант № 34152

Выполнил: Ватан Хатиб

Факультет: Программной инженерии и компьютерной техники

Группа: Р3113

Преподаватель: Блохина Елена Николаевна

Город Санкт-Петербург

2022 год

## Задание:

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 вывод данных на ВУ-3
2. Программа начинается с адреса  $0B7_{16}$ . Размещаемая строка находится по адресу  $62E_{16}$ .
3. Строка должна быть представлена в кодировке Windows-1251.
4. Формат представления строки в памяти: АДР0: ДЛИНА АДР1: СИМВ1 СИМВ2 АДР2: СИМВ3 СИМВ4 ..., где ДЛИНА - 16 разрядное слово, где значащими являются 8 младших бит.
5. Вывод строки начинается со вывода количества символов (1 байт), и должен быть завершен по выводу их необходимого количества.

## Текст исходной программы:

Адрес	Код команды	Мнемоника	Комментарии
075	+0200	CLA	0 -> AC
076	D001	CALL 1	Вызов подпрограмма 1
077	AAFC	LD (IP-4)+	Косвенная автоинкрементная (ip-4) -> AC
078	1302	OUT 2	Вывод количества символов (младшего байта аккумулятора)
079	0C00	PUSH	Положить на стек значение количества символов (длина)
07A	D001	CALL 1	Вызов подпрограмма 1
07B	AAF8	LD (IP-8)+	Косвенная автоинкрементная (ip-4) -> AC
07C	E073	ST 73	Сохранение значения аккумулятора в 73 (ячейку, хранящую адрес текущего элемента массива)
07D	1302	OUT 2	Вывод элемента массив (младшего байта аккумулятора)
07E	AC00	LD(SP)	Косвенная относительная, со смещением ,загрузка ячеек номер 7FF (длина)
07F	F006	BEQ IP+6	Переход к ячейке 086, если длина = 0
080	D001	CALL 1	Вызов подпрограмма 1
081	A073	LD 73	Прямая абсолютная загрузка ячеек 73 ( ячейку, хранящую адрес текущего элемента массива)
082	0680	SWAB	Менять младшего байта аккумулятора в старший байта аккумулятора
083	1302	OUT 2	Вывод элемента массив (младшего байта аккумулятора)
084	AC00	LD(SP)	Косвенная относительная, со смещением ,загрузка ячеек номер 7FF (длина)
085	F1F4	BNE IP-12	Переход к ячейке 07A(начала loop), если длина != 0
086	0800	POP	мы исправляем SP и ячейку который указывает на элементы в массиве, чтобы программа стала Реентерабельность
087	A072	LD 72	
088	E074	ST 74	
089	0100	HLT	Отключение ТГ, переход в пультовый режим

## ***Текст исходной подпрограммы:***

Адрес	Код команды	Мнемоника	Комментарии
001	1203	IN 3	Бесконечный цикл, цель которого получить флаг готовности КВУ-1 к выводу (При отсутствии флага происходит возврат к ячейке 001, иначе – продолжение программы)
002	2F40	AND #0x40	
003	F0FD	BEQ IP-3	
004	AC01	LD (SP+1)	ЗАГРУЗИТЕ ДЛИНУ, УМЕНЬШИТЕ ЕЕ НА ЕДИНИЦУ А ЗАТЕМ СОХРАНИТЕ ЕЕ
005	0740	DEC	
006	EC01	ST (SP+1)	
007	0A00	RET	(SP)+->IP, ВОЗВРАТ К ПРОГРАММЕ

## ***Назначение комплекса программ:***

Асинхронный вывод длины строки и самой строки в формате АДР0: ДЛИНА  
АДР1: СИМВ1 СИМВ2 АДР2: СИМВ3 СИМВ4 ...

## ***Назначение программы:***

Асинхронный вывод длины строки символов с передачей длину в подпрограмму.

Если длина строки символов не 0, то пользователь получит на КВУ-1 соответствующе элемент массив

## ***Назначение подпрограммы:***

дождитесь сигнала готовности, затем уменьшите длину на единицу

## ***Область представления:***

Пары символов – ячейки с 637 по 632 – 16-разрядные беззнаковые числа;

Длина – ячейка 636 – длина строки – 8-разрядное знаковое число;

ARRAYPOINTER – ячейка 074 – указывает на адрес текущего элемента массива – 11-разрядное беззнаковое число;

FIXEDPOINTER - ячейка 072 - мы используем его для фиксации флага, когда программа завершила свою задачу, чтобы программа могла быть

Реентерабельность– 11-разрядное беззнаковое число;

CURRENT– ячейка 073 – Содержать текущий элемент массива – 16-разрядное беззнаковое число;

## ***Область допустимых значений:***

Пары символов –  $[0 ; 2^{16}-1]$ ;

ДЛИНА –  $[0 ; 2^7-1]$ , поскольку по условию ДЛИНА – 16-разрядное знаковое слово со значащими младшими 8 битами;

POINTER –  $[0 ; 2^{11}-1]$ ;

CURRENT –  $[0 ; 2^{16}-1]$ ;

FIXED POINTER –  $[0 ; 2^{11}-1]$ ;

## ***Строка, выданная преподавателем:***

Снег&

В кодировке Windows-1251: D1 ED E5 E3 26

В кодировке UTF-8: D0A1 D0BD D0B5 D0B3 26

В кодировке UTF-16: 0421 043D 0435 0433 0026

КОИ-8 : F3 CE C5 C7 26

## ***Расположение в памяти БЭВМ программы и исходных данных:***

Программа: ячейки 075-089;

Подпрограмма: ячейки 001-007;

Исходные данные для программы: ячейки 072-074;

Размещаемая строка: ячейки 636-639.

## ***Ячейки с исходными данными:***

*Для программы:*

Адрес	Код команды	Мнемоника	Комментарии
072	636	FIXED POINTER	Переменная, указывает на адрес текущего элемента массива
073	XXXX	CURRENT	Переменная, хранящая в себе текущего элемента массива

074	636	POINTER	Переменная, указывает на адрес текущего элемента массива
-----	-----	---------	--

Для размещаемой строки:

Адрес	Код команды	Мнемоника	Комментарии
636	0005	ДЛИНА	Переменная, хранит в себе длину строки (количество символов). Её адрес – начало размещаемой строки.
637	CEF3	--	Переменная, хранит пару символов в определенной кодировке ( КОИ-8 ). Символы («С», «н»)
638	C7C5	--	Переменная, хранит пару символов в определенной кодировке ( КОИ-8 ). Символы («е», «г»)
639	0026	--	Переменная, хранит символ в определенной кодировке( КОИ-8 ). Символы («&»)

Трассировка:

Таблица трассировки.

Адрес	Знач	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Нов знач
075	+0200	076	0200	075	0200	000	0075	0000	0100		
076	D001	001	D001	7FF	0077	7FF	D001	0000	0100	7FF	0077
001	1203	002	1203	001	1203	7FF	0001	0040	0000		
002	2F40	003	2F40	002	0040	7FF	0040	0040	0000		
003	F0FD	004	F0FD	003	F0FD	7FF	0003	0040	0000		
004	AC01	005	AC01	000	0000	7FF	0001	0000	0100		
005	0740	006	0740	005	0740	7FF	0005	FFFF	1000		
006	EC01	007	EC01	000	FFFF	7FF	0001	FFFF	1000	000	FFFF
007	0A00	077	0A00	7FF	0077	000	0007	FFFF	1000		
077	AAFC	078	AAFC	636	0005	000	FFFC	0005	0000	074	0637
078	1302	079	1302	078	0005	000	0078	0005	0000		
079	0C00	07A	0C00	7FF	0005	7FF	0079	0005	0000	7FF	005
07A	D001	001	D001	7FE	007B	7FE	D001	0005	0000	7FE	007B
001	1203	002	1203	001	1203	7FE	0001	0040	0000		
002	2F40	003	2F40	003	0040	7FE	0040	0040	0000		
003	F0FD	004	F0FD	003	F0FD	7FE	0003	0040	0000		
004	AC01	005	AC01	7FF	0005	7FE	0001	0005	0000		
005	0740	006	0740	005	0740	7FE	0005	0004	0001		
006	EC01	007	EC01	7FF	0004	7FE	0001	0004	0001	7FF	004

Адрес	Знач	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Нов знач
<b>007</b>	<b>0A00</b>	07B	0A00	7FE	007B	7FF	0007	0004	0001		
07B	AAF8	07C	AAF8	637	CEF3	7FF	FFF8	CEF3	1001	074	0638
07C	E073	07D	E073	073	CEF3	7FF	007C	CEF3	1001	073	CEF3
07D	1302	07E	1302	07D	1302	7FF	007D	CEF3	1001		
07E	AC00	07F	AC00	7FF	0004	7FF	0000	0004	0001		
07F	F006	080	F006	07F	F006	7FF	007F	0004	0001		
080	D001	001	D001	7FE	0081	7FE	D001	0004	0001	7FE	0081
<b>001</b>	<b>1203</b>	002	1203	001	1203	7FE	0001	0040	0001		
<b>002</b>	<b>2F40</b>	003	2F40	002	0040	7FE	0400	0040	0001		
<b>003</b>	<b>F0FD</b>	004	F0FD	003	F0FD	7FE	0003	0040	0001		
<b>004</b>	<b>AC01</b>	005	AC01	7FF	0004	7FE	0001	0004	0001		
<b>005</b>	<b>0740</b>	006	0740	005	0740	7FE	0005	0003	0001		
<b>006</b>	<b>EC01</b>	007	EC01	7FF	0003	7FE	0001	0003	0001	7FF	0003
<b>007</b>	<b>0A00</b>	081	0A00	7FE	0081	7FF	0007	0003	0001		
081	A073	082	A073	073	CEF3	7FF	0081	CEF3	1001		
082	0680	083	0680	082	0680	7FF	0082	F3CE	1001		
083	1302	084	1302	083	1302	7FF	0083	F3CE	1001		
084	AC00	085	AC00	7FF	0003	7FF	0000	0003	0001		
085	F1F4	07A	F1F4	085	F1F4	7FF	FFF4	0003	0001		
07A	D001	001	D001	7FE	007B	7FE	D001	0003	0001	7FE	007B
<b>001</b>	<b>1203</b>	002	1203	001	1203	7FE	0001	0040	0001		
<b>002</b>	<b>2F40</b>	003	2F40	0040	0040	7FE	0040	0040	0001		
<b>003</b>	<b>F0FD</b>	004	F0FD	0003	F0FD	7FE	0003	0040	0001		
<b>004</b>	<b>AC01</b>	005	AC01	7FF	0003	7FE	0001	0003	0001		
<b>005</b>	<b>0740</b>	006	0740	005	0740	7FE	0005	0002	0001		
<b>006</b>	<b>EC01</b>	007	EC01	7FF	0002	7FE	0001	0002	0001	7FF	0002
<b>007</b>	<b>0A00</b>	07B	0A00	7FE	007B	7FF	0007	0002	0001		
07B	AAF8	07C	AAF8	638	C7C5	7FF	FFF8	C7C5	1001	074	0639
07C	E073	07D	E073	073	C7C5	7FF	007C	C7C5	1001	073	C7C5
07D	1302	07E	1302	07D	1302	7FF	007D	C7C5	1001		
07E	AC00	07F	AC00	7FF	0002	7FF	0000	0002	0001		
07F	F006	080	F006	07F	F006	7FF	007F	0002	0001		
080	D001	001	D001	7FE	0081	7FE	D001	0002	0001	7FE	0081
<b>001</b>	<b>1203</b>	002	1203	001	1203	7FE	0001	0040	0001		
<b>002</b>	<b>2F40</b>	003	2F40	002	0040	7FE	0040	0040	0001		
<b>003</b>	<b>F0FD</b>	004	F0FD	003	F0FD	7FE	0003	0040	0001		
<b>004</b>	<b>AC01</b>	005	AC01	7FF	0002	7FE	0001	0002	0001		
<b>005</b>	<b>0740</b>	006	0740	005	0740	7FE	0005	0001	0001		
<b>006</b>	<b>EC01</b>	007	EC01	7FF	0001	7FE	0001	0001	0001	7FF	0001
<b>007</b>	<b>0A00</b>	081	0A00	7FE	0081	7FF	0007	0001	0001		
081	A073	082	A073	073	C7C5	7FF	0081	C7C5	1001		

Адрес	Знач	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Нов знач
082	0680	083	0680	082	0680	7FF	0082	C5C7	1001		
083	1302	084	1302	083	1302	7FF	0083	C5C7	1001		
084	AC00	085	AC00	7FF	0001	7FF	0000	0001	0001		
085	F1F4	07A	F1F4	085	F1F4	7FF	FFF4	0001	0001		
07A	D001	001	D001	7FE	007B	7FE	D001	0001	0001	7FE	007B
001	1203	002	1203	001	1203	7FE	0001	0040	0001		
002	2F40	003	2F40	002	0040	7FE	0040	0040	0001		
003	F0FD	004	F0FD	003	F0FD	7FE	0003	0040	0001		
004	AC01	005	AC01	7FF	0001	7FE	0001	0001	0001		
005	0740	006	0740	005	0740	7FE	0005	0000	0101		
006	EC01	007	EC01	7FF	0000	7FE	0001	0000	0101	7FF	0000
007	0A00	07B	0A00	7FE	007B	7FF	0007	0000	0101		
07B	AAF8	07C	AAF8	639	0026	7FF	FFF8	0026	0001	074	063A
07C	E073	07D	E073	073	0026	7FF	007C	0026	0001	073	0026
07D	1302	07E	1302	07D	1302	7FF	007D	0026	0001		
07E	AC00	07F	AC00	7FF	0000	7FF	0000	0000	0101		
07F	F006	086	F006	07F	F006	7FF	0006	0000	0101		
086	0800	087	0800	7FF	0000	000	0086	0000	0101		
087	A072	088	A072	072	0636	000	0087	0636	0001		
088	E074	089	E074	074	0636	000	0088	0636	0001	074	0636
089	0100	08A	0100	089	0100	000	0089	0636	0001		

**Текст программы на языке Assembler:**

Текст программы и необходимых для её работы данных:

Метка	Мнемоника	Параметр	Описание
	ORG	0x0072	
FIXEDPOINTER	WORD	0x0636	Мы используем его для того, чтобы исправить программу после завершения
CURRENT	WORD	0x0000	текущий элемент массива
ARRAYPOINTER	WORD	0x0636	указывает на элементы массива
	ORG	0x0075	
START	CLA		очистите аккумулятор
	CALL	LISTEN	Вызов подпрограмма
	LD	(ARRAYPOINTER)+	получить текущий элемент (длина)
	OUT	2	выведите символ на ву-1 (длина)
	PUSH		Вставьте длину в стек
STARTLOOP	CALL	LISTEN	Вызов подпрограмма

	LD	(ARRAYPOINTER)+	получить текущий элемент
	ST	CURRENT	сохранить текущий элемент
	OUT	2	выведите символ на ву-1
	LD	(SP+0)	получить длину из стека
	BEQ	FINISH	если длина равна 0 перейдите к завершению
	CALL	LISTEN	Вызов подпрограмма
	LD	CURRENT	получить текущий элемент
	SWAB		поменяйте местами первые 8 битов на последние 8
	OUT	2	выведите символ на ву-1
	BNE	STARTLOOP	если длина не равна 0 вернитесь к началу цикла
FINISH	POP		исправьте так, чтобы программа снова работала без проблем
	LD	FIXEDPOINTER	
	ST	ARRAYPOINTER	
	HLT		остановка

*Текст подпрограммы и необходимых для её работы данных:*

Метка	Мнемоника	Параметр	Описание
	ORG	0x0001	
LISTEN	IN	3	дождитесь получения сигнала готовности от ву-1
	AND	#0x40	
	BEQ	LISTEN	
	LD	(SP+1)	получить длину из стека
	DEC		уменьшите длину на 1
	ST	(SP+1)	длина хранения в стека
	RET		вернуть

*Ячейки с данными (размещаемая строка):*

Метка	Мнемоника	Параметр	Описание
	ORG	0x0636	
	WORD	0x0005	Переменная, хранит в себе длину строки (количество символов). Её адрес – начало размещаемой строки.
	WORD	0xCEF3	Переменная, хранит пару символов в определенной кодировке ( КОИ-8 ). Символы («С», «н»)
	WORD	0xC7C5	Переменная, хранит пару символов в определенной кодировке ( КОИ-8 ). Символы («е», «г»)
	WORD	0x0026	Переменная, хранит символ в определенной кодировке( КОИ-8 ). Символы («&»)



Программа в вставляемом виде:

ORG 0x0001

LISTEN: IN 3

AND #0x40

BEQ LISTEN

LD (SP+1)

DEC

ST(SP+1)

RET

ORG 0x0636

WORD 0x0005

WORD 0xCEF3

WORD 0xC7C5

WORD 0x0026

ORG 0x0072

FIXEDPOINTER: WORD 0x0636

CURRENT:WORD 0x0000

ARRAYPOINTER:WORD 0x0636

ORG 0x0075

START: CLA

CALL LISTEN

LD (ARRAYPOINTER)+

OUT 2

PUSH

STARTLOOP: CALL LISTEN

LD (ARRAYPOINTER)+

ST CURRENT

OUT 2

LD (SP+0)

BEQ FINISH  
CALL LISTEN  
LD CURRENT  
SWAB  
OUT 2  
LD (SP+0)  
BNE STARTLOOP  
FINISH: POP  
LD FIXEDPOINTER  
ST ARRAYPOINTER  
HLT