

Федеральное государственное автономное образовательное учреждение высшего
образования

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет: ПИиКТ

Направление 09.03.04 «Системное и прикладное программное обеспечение»

Мегафакультет: КТиУ

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине:

«ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ»

"Асинхронный обмен данными с ВУ"



Выполнил:

Студент 1 курса

группа Р3115

Вариант 1532

Девяткин А. Ю.

Преподаватель:

Перцев Т.С.

Санкт-Петербург

2021

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.

1. Программа осуществляет асинхронный вывод данных на ВУ-1
2. Программа начинается с адреса 427_{16} . Размещаемая строка находится по адресу 609_{16} .
3. Строка должна быть представлена в кодировке Windows-1251.
4. Формат представления строки в памяти: АДР1: СИМВ1 СИМВ2
АДР2: СИМВ3 СИМВ4 ... СТОП_СИМВ.

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.

Текст исходной программы

Адрес	Метка	Код команды	Мнемоника	Комментарий
427		0609	ADDR_FIRST	Адрес начала строки
428		0000	ADDR_CURRENT	Адрес текущего элемента
429		0002	READ_BYTE	Указывает, какой символ передавать из ячейки (старший или младший байт)
42A		000D	STOP_SYMBOL	Символ окончания строки
42B		00FF	MASK	Вспомогательная переменная
42C	START	AEFA	LD ADDR_FIRST	Инициализация указатель на текущий элемент
42D		EEFA	ST ADDR_CURRENT	
42E	SPIN	1203	IN 3	Проверяем, нажата ли кнопка "готов"
42F		2F40	AND #0x40	
430		F0FD	BEQ SPIN	Если нет, проверяем ещё раз;
431		AEF7	LD READ_BYTE	работаем со старшим или
432		7F02	CMP #0x2	с младшим байтом
433		F004	BEQ CASE_2	
434	CASE_1	AF02	LD #0x2	меняем режим, чтобы на следующем проходе вывести старший байт
435		EEF3	ST READ_BYTE	
436		AAF1	LD (ADDR_CURRENT)+	оба символа из ячейки прочитаны, так что перемещаем указатель на след. ячейку памяти
437		CE04	JUMP OUTPUT	
438	CASE_2	AF01	LD #0x1	меняем режим, чтобы в последующем вывести младший байт
439		EEEF	ST READ_BYTE	
43A		A8ED	LD (ADDR_CURRENT)	загружаем оба символа из ячейки памяти,
43B		0680	SWAB	помещаем необходимый символ в младший байт
43C	OUTPUT	1302	OUT 2	выводим символ
43D		2EED	AND MASK	проверяем, является ли он стоп-символом если да, завершаем программу если нет, ждем нажатия кнопки "готов" для вывода следующего символа
43E		7EEB	CMP STOP_SYMBOL	
43F		F001	BEQ END_	
440		CEED	JUMP SPIN	
441	END_	0100	HLT	

Программа на языке ассемблера БЭВМ:

```
ORG      0x427;
ADDR_FIRST:  WORD  0x609;  Адрес начала строки
ADDR_CURRENT: WORD  0;      Адрес текущего элемента
READ_BYTE:   WORD  0x2;     Указывает, какой символ передавать из ячейки
                        ; (старший или младший байт)
STOP_SYMBOL: WORD  0x0D;     Символ окончания строки
MASK:        WORD  0x00FF;

START:       LD  ADDR_FIRST;
              ST  ADDR_CURRENT;  Инициализируем указатель на текущий элемент

SPIN:        IN  3;           Проверяем, нажата ли кнопка "готов"
              AND #0x40;
              BEQ SPIN;       Если нет, проверяем ещё раз;

              LD  READ_BYTE;
              CMP #0x2;       работаем со старшим или с младшим байтом
              BEQ CASE_2;

CASE_1:      LD #0x2;         меняем режим, чтобы на следующем проходе вывести старший байт
              ST READ_BYTE;   2 -> READ_BYTE
              LD (ADDR_CURRENT)+; оба символа из ячейки прочитаны, так что
              JUMP OUTPUT;    перемещаем указатель на след. ячейку памяти

CASE_2:      LD #0x1;         меняем режим, чтобы в последующем вывести младший байт
              ST READ_BYTE;   1 -> READ_BYTE
              LD (ADDR_CURRENT); загружаем оба символа из ячейки памяти,
              SWAB;           помещаем необходимый символ в младший байт

OUTPUT:      OUT 2;           выводим символ
              AND MASK;
              CMP STOP_SYMBOL; проверим, является ли он стоп-символом,
              BEQ END_;       если да, завершаем программу, если нет,
              JUMP SPIN;      ждем нажатия кнопки "готов" для вывода следующего символа

END_:        HLT;
```

Информация о программе

Программа осуществляет асинхронный вывод данных на ВУ-1

Программа находится в ячейках 42С – 441

Переменные находятся в ячейках 427 – 42В

Выводимая строка начинается с ячейки 609

Вывод

Писать программу на ассемблере БЭВМ куда быстрее и проще, чем вводить коды команд. А ещё в БЭВМ есть несколько интересных устройств ввода вывода, с помощью которых можно осуществлять ввод и вывод данных.