

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ
ТЕХНИКИ

НАПРАВЛЕНИЕ ПРОГРАММНАЯ ИНЖЕНЕРИЯ
ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА СИСТЕМНОЕ И ПРИКЛАДНОЕ
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5
курса «Основы профессиональной деятельности»
по теме: «Асинхронный обмен данными с ВУ»
Вариант № 9501

Выполнил студент:

Шубин Егор Вячеславович

группа: Р3109

Преподаватель:

Лектор: Клименков С. В.,

Практик: Ткешелашвили Н. М.

Санкт-Петербург, 2025 г.

Содержание

Лабораторная работа № 5. Асинхронный обмен данными с ВУ	3
1. Задание варианта № 9501	3
2. Выполнение задания	4
1. Код программы на ассемблере:	4
2. Текст исходной программы:	5
3. Описание программы:	6
4. Размещение данных в памяти:	6
5. Допустимые символы в кодировке ISO-8859-5:	6
6. Получение Символов:	6
7. Таблица трассировки:	7
8. Дополнительное задание:	7
3. Вывод	8

Лабораторная работа № 5

Асинхронный обмен данными с ВУ

1. Задание варианта № 9501

- Программа осуществляет асинхронный вывод данных на ВУ-3
- Программа начинается с адреса 115_{16} . Размещаемая строка находится по адресу $55F_{16}$.
- Строка должна быть представлена в кодировке ISO-8859-5.
- Формат представления строки в памяти: АДР1: СИМВ1 СИМВ2 АДР2: СИМВ3 СИМВ4 ... СТОП_СИМВ.
- Ввод или вывод строки должен быть завершен по символу с кодом 00 (NUL). Стоп символ является обычным символом строки и подчиняется тем же правилам расположения в памяти что и другие символы строки.

2. Выполнение задания

2. 1. Код программы на ассемблере:

```
1  ORG 0x115
2
3  StartAddr : WORD 0x55F
4  CurAddr: WORD ?
5  Mask: WORD 0x00FF
6  Count: WORD 0x0002
7  START:
8      CLA
9      LD StartAddr
10     ST CurAddr
11
12  LOAD:
13     LD #0x02
14     ST Count
15     LD(CurAddr)+
16     CALLING:
17         SWAB
18         PUSH
19         CALL SYMBOL
20         POP
21         LOOP Count
22         JUMP CALLING
23     JUMP LOAD
24
25  SYMBOL:
26     IN 7
27     AND #0x40
28     BEQ SYMBOL
29     LD &1
30         OUT 6
31     AND Mask
32     BEQ STOP
33     RET
34
35  STOP:
36     POP
37     POP
38     HLT
39
40  ORG 0x55F
41  WORD 0xC2C0 ; T P
42  WORD 0xBEBD ; O H
43  WORD 0x0000
```

2. 2. Текст исходной программы:

Адрес	Код команды	Мнемоника	Комментарии
115	05FF	-	StartAddr
116	0000	-	CurAddr
117	0002	-	Count
118	0200	CLA	Очистить аккумулятор
119	AEFB	LD IP-4(115)	Прямая относительная загрузка StartAddr-> AC
11A	EEFB	ST IP-4(116)	Прямая относительная выгрузка AC -> CurAddr
11B	AF02	LD #0x02	Прямая загрузка кол-ва итераций 002 -> AC
11C	EEFA	ST (IP-5)	Прямая относительная выгрузка AC -> Count
11D	AAF8	LD (IP-7)+	Косвенная автоинкрементная загрузка MEM(MEM(116)) -> AC MEM(116)++
11E	0680	SWAB	Поменять байты $AC7 \dots AC0 \longleftrightarrow AC15 \dots AC8$
11F	0C00	PUSH	Положить символы в стек AC -> -(SP)
120	D125	CALL (125)	Вызов функции вывода символа
121	0800	POP	Очистить стек (SP)+ -> AC
122	8EF4	LOOP (IP-11)	Луп-спин для загрузки двух символов (Count)- < 0 -> Ip++
123	CEFA	JUMP(IP-5)	Относительный прыжок IP-5 -> IP
124	CEF6	JUMP(IP-9)	Относительный прыжок IP-9 -> IP
125	1207	IN 7	Чтение из регистра ВУ-3
126	2F40	AND #0x40	Проверка готовности регистра AC & 0x40 -> AC
127	F0FD	BEQ(IP-2)	Повторное чтение если 0
128	AC01	LD (SP+1)	Загрузка символа из стека (SP+1) -> AC
129	2E05	AND (IP+5+1)	Проверка на стоп символ AC & MASK -> AC
12A	1306	OUT 6	Запись в регистр ВУ-3
12B	0A00	RET	Возврат (SP)+ -> IP
12C	0800	POP	Очистить стек (SP)+ -> AC
12D	0800	POP	Очистить стек (SP)+ -> AC
12E	0100	HLT	Остановка
12F	00FF	Mask	Маска для проверки стоп-символа

Таблица 1.1: Текст исходной программы

Адрес	Код команды	Мнемоника	Комментарии
55F	C2C0	-	1,2 символы
560	BEBD	-	3,4 символы
560	0000	-	стоп-символ

Таблица 1.2: Текст исходной программы

2. 3. Описание программы:

Программа осуществляет вывод символов с ВУ-3, начиная с ячейки $55F_{16}$, до ввода символа 00_{16} . Для хранения промежуточных данных используется стек. Также реализована отдельная функция для вывода символа, универсальная для обоих символов, хранящихся в 1 и 2 байте.

2. 4. Размещение данных в памяти:

115 - 117 - адрес первого, текущего символов и количество итераций для вывода символов

118 - 124 - основная программа

125 - 12B - подпрограмма для вывода символа

12C - 12E - подпрограмма для остановки

12F - Маска для проверки стоп-символа

55F - адрес начала строки

670_{10} - максимальная длина строки

2. 5. Допустимые символы в кодировке ISO-8859-5:

$$[21_{16}; 7E_{16}] \cup [A0_{16}; FF_{16}]$$

2. 6. Получение Символов:

Слово - ТРОН в кодировке ISO-8859-5.

Т - C2

Р - C0

О - BE

Н - BD

Слово - ТРОН в кодировке UTF-8.

Т - $0xD0A2$

Р - $0xD0A0$

О - $0xD09E$

Н - $0xD09D$

Слово - ТРОН в кодировке UTF-16.

T - 0x0422
P - 0x0420
O - 0x041E
H - 0x041D

2. 7. Таблица трассировки:

Адр	Знач	IP	CR	AR	DR	SP	BR	AC	NZVC	Адр	Знач
xxx	xxxx	xxx	xxxx	xxx	xxxx	xxx	xxxx	xxxx	xxxx	xxx	xxxx
119	0200	11A	0200	119	0200	000	0119	0000	0100		
11A	AEFA	11B	AEFA	115	055F	000	FFFA	055F	0000		
11B	EEFA	11C	EEFA	116	055F	000	FFFA	055F	0000	116	055F
11C	AF02	11D	AF02	11C	0002	000	0002	0002	0000		
11D	EEF9	11E	EEF9	117	0002	000	FFF9	0002	0000	117	0002
11E	AAF7	11F	AAF7	55F	C2C0	000	FFF7	C2C0	1000	116	0560
11F	0680	120	0680	11F	0680	000	011F	C0C2	1000		
120	0C00	121	0C00	7FF	C0C2	7FF	0120	C0C2	1000	7FF	C0C2
121	DE04	121	0000	000	0000	000	0000	0000	0100		
121	DE04	126	DE04	7FF	0122	7FF	0126	0000	0100	7FF	0122
126	1207	127	1207	126	1207	7FF	0126	0040	0100		
127	2F40	128	2F40	127	0040	7FF	0040	0040	0000		
128	F0FD	129	F0FD	128	F0FD	7FF	0128	0040	0000		
129	AC01	12A	AC01	000	0000	7FF	0001	0000	0100		
12A	1306	12B	1306	12A	1306	7FF	012A	0000	0100		
12B	2EEC	12C	2EEC	118	00FF	7FF	FFEC	0000	0100		
12C	F001	12E	F001	12C	F001	7FF	0001	0000	0100		
12E	0800	12E	0000	000	0000	000	0000	0000	0100		
12E	0800	12F	0800	000	0000	001	012E	0000	0100		
12F	0800	130	0800	001	0000	002	012F	0000	0100		
130	0100	131	0100	130	0100	002	0130	0000	0100		

Таблица 1.3: Трассировка программы

2. 8. Дополнительное задание:

С ВУ-8 (клавиатура) вводится дата в формате "дд.мм нажатие enter - завершение ввода, иные символы (кроме цифр и точки) игнорировать. После окончания ввода, на ВУ-6 (бегущая строка) вывести изображение знака зодиака. Сделать проверку формата введенных данных, в случае некорректной даты вывести ":(".

Код дополнительного задания

3. Вывод

Во время написания данной лабораторной работы я научился писать программы на ассемблере. Изучил как работает асинхронный ввод-вывод данных на различных ВУ. Использовал разные кодировки для вывода символов. А также изучил как работают внешние устройства.