

Федеральное государственное автономное образовательное учреждение высшего
образования
«Национальный исследовательский университет ИТМО»
Факультет программной инженерии и компьютерной техники

Отчет

по лабораторной работе №5

«Асинхронный обмен данными с ВУ»

по дисциплине «Основы профессиональной деятельности»

вариант 1413

Выполнил: Векшин А.И.,

группа Р3116

Преподаватель: Афанасьев

Д.Б.

Санкт-Петербург

~ 2023 ~

Содержание

Отчет	1
Содержание	2
Текст задания	3
Текст исходной программы	4
Описание программы	5
Исходная строка	6
Трассировка	7
Вывод.....	8

Текст задания

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 $56C_{16}$. Размещаемая строка находится по адресу $5D7_{16}$.
3. Строка должна быть представлена в кодировке КОИ-8.
4. Формат представления строки в памяти: АДР1: СИМВ1 СИМВ2 АДР2: СИМВ3 СИМВ4 ... 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.

Текст исходной программы

```
                                ORG 0x056C                ; Адрес начала программы
STR_POINTER: WORD 0x05D7        ; Указатель на строку
END_CHAR:   WORD 0x000A        ; Стоп символ
MASK:       WORD 0x00FF        ; маска

START:      LD (STR_POINTER)    ; Загрузка слова
            SWAB                ; Обмен байт
            PUSH                ; Сохранение в стек
            CALL OUT_CHAR       ; Вызов подпрограммы записи
                                ; для старших 8 бит
            POP                 ;
            AND MASK            ; Проверка на стоп символ
            CMP END_CHAR        ;
            BEQ EXIT            ;
            LD (STR_POINTER)+   ; Загрузка слова
            PUSH                ; Сохранение в стек
            CALL OUT_CHAR       ; Вызов подпрограммы записи
                                ; для младших 8 бит
            POP                 ;
            AND MASK            ; Проверка на стоп символ
            CMP END_CHAR        ; Проверка на стоп символ
            BEQ EXIT            ;
            JUMP START          ; Безусловный переход в
                                ; начало программы

EXIT:       HLT

                                ; Подпрограмма вывода
                                ; символа на устройство
OUT_CHAR:   IN 3                ; Спин-луп по флагу ВУ-1
            AND #0x40           ;
            BEQ OUT_CHAR        ;
            LD &1               ; Загрузка из стека
            OUT 2               ; Вывод символа на
                                ; устройство
            RET                 ; Выход из подпрограммы
                                ; записи
```

Описание программы

Назначение

Программа осуществляет асинхронный вывод строки из памяти на ВУ-1 до тех пор, пока не будет выведен стоп-символ (0x0A).

Переменные

	Адрес	Описание
POINTER	56C	Указатель на первый символ
LEN		Длина строки

Область представления

POINTER – беззнаковое 11-ти разрядное число

ОДЗ

$$\left[\begin{array}{l} 0 \leq POINTER \leq 56B \\ 1 \leq LEN \leq AD6 - POINTER * 2 \\ 588 \leq POINTER \leq 7FE \\ 1 \leq LEN \leq 4EE - POINTER * 2 \end{array} \right.$$

Примечание: длина строки указана в **символах**

Допустимы 8-бит кодировки, корректно будут отображаться кодировки семейства KOI8(разные языки) и ASCII

Расположение в памяти

Программа: от 56C до 586

Строка: от POINTER до POINTER + LEN/2

Исходная строка

Троллить бодибилдеров просто, но...

KOI8-R:

Т	р	о	л	л	и	т	ь	
F4	D2	CF	CC	CC	C9	D4	D8	20

б	о	д	и	б	и	л	д	е	р	о	в	
C2	CF	C4	C9	C2	C9	CC	C4	C5	D2	CF	D7	20

п	р	о	с	т	о	,		н	о	.	.	.
D0	D2	CF	D3	D4	CF	2C	20	CE	CF	2E	2E	2E

UTF-8

Т	р	о	л	л	и	т	ь	
D0A2	D180	D0BE	D0BB	D0BB	D0B8	D182	D18C	20

б	о	д	и	б	и	л	д	е	р	о	в	
D0B1	D0BE	D0B4	D0B8	D0B1	D0B8	D0BB	D0B4	D0B5	D180	D0BE	D0B2	20

п	р	о	с	т	о	,		н	о	.	.	.
D0BF	D180	D0BE	D181	D182	D0BE	2C	20	D0BD	D0BE	2E	2E	2E

UTF-16:

Т	р	о	л	л	и	т	ь	
0422	0440	043E	043B	043B	0438	0442	044C	0020

б	о	д	и	б	и	л	д	е	р	о	в	
0431	043E	0434	0438	0431	0438	043B	0434	0435	0440	043E	0432	0020

п	р	о	с	т	о	,		н	о	.	.	.
043F	0440	043E	0441	0442	043E	002C	0020	043D	043E	002E	002E	002E

Трассировка

Строка: "тр"

Выполняемая команда		Содержимое регистров процессора после выполнения команды								Ячейка, содержимое которой изменилось	
Адрес	Знач	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Знач
56F	A8FC	56F	0000	000	0000	000	0000	0000	0100		
56F	A8FC	570	A8FC	5D7	F4D2	000	FFFC	F4D2	1000		
570	0680	571	0680	570	0680	000	0570	D2F4	1000		
571	0C00	572	0C00	7FF	D2F4	7FF	0571	D2F4	1000	7FF	D2F4
572	DE0D	580	DE0D	7FE	0573	7FE	0580	D2F4	1000	7FE	0573
580	1203	581	1203	580	1203	7FE	0580	D200	1000		
581	2F40	582	2F40	581	0040	7FE	0040	0000	0100		
582	F0FD	580	F0FD	582	F0FD	7FE	FFFD	0000	0100		
580	1203	581	1203	580	1203	7FE	0580	0040	0100		
581	2F40	582	2F40	581	0040	7FE	0040	0040	0000		
582	F0FD	583	F0FD	582	F0FD	7FE	0582	0040	0000		
583	AC01	584	AC01	7FF	D2F4	7FE	0001	D2F4	1000		
584	1302	585	1302	584	1302	7FE	0584	D2F4	1000		
585	0A00	573	0A00	7FE	0573	7FF	0585	D2F4	1000		
573	0800	574	0800	7FF	D2F4	000	0573	D2F4	1000		
574	2EF9	575	2EF9	56E	00FF	000	FFF9	00F4	0000		
575	7EF7	576	7EF7	56D	000A	000	FFF7	00F4	0001		
576	F008	577	F008	576	F008	000	0576	00F4	0001		
577	AAF4	578	AAF4	5D7	F4D2	000	FFF4	F4D2	1001	56C	05D8
578	0C00	579	0C00	7FF	F4D2	7FF	0578	F4D2	1001	7FF	F4D2
579	DE06	580	DE06	7FE	057A	7FE	0580	F4D2	1001	7FE	057A
580	1203	581	1203	580	1203	7FE	0580	F400	1001		
581	2F40	582	2F40	581	0040	7FE	0040	0000	0101		

Санкт-Петербург

~ 2023 ~

Вывод

.Λ_Λ

(·ω·.)⌋—☆·*。

⊂ √ ·° +.

⌋—J °. + *''')

·'·*''')

(,·' (,·'* ☆ Нужно читать методичку