

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

**Основы Профессиональной Деятельности
Работа с БЭВМ
Лабораторная работа №5
Вариант 414085**

Выполнил:
Студент группы Р3116
Брагин Роман Андреевич

Проверила:
Блохина Елена Николаевна

г. Санкт-Петербург

2024 г

Задание:

Введите номер варианта

1. Программа осуществляет асинхронный ввод данных с ВУ-3
2. Программа начинается с адреса 29D₁₆. Размещаемая строка находится по адресу 5C0₁₆.
3. Строка должна быть представлена в кодировке ISO-8859-5.
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. Ввод строки начинается со ввода количества символов (1 байт), и должен быть завершен по вводу их необходимого количества.

Доп. Задание:

Через ВУ-3 последовательно вводим кол-во символов, потом стоп-символ (тоже через клавиатуру), далее ввод осуществляем через клавиатуру, а вывод на текстовый принтер в **обратном** порядке только четные по номеру символы.

Размещение в памяти аналогично заданию, но длина должна соответствовать кол-ву реально введенных, введенный стоп-символ сохраняется по правилу хранения остальных символов.

Текст программы на ассемблере:

```
ORG             0x29D; адрес начала программы
res:            WORD 0x5C1; начало слова
length:         word 0x5C0; длина слова
leny:           word 0x5BF ; временная переменная для длины
START: CLA      ; очистка аккумулятора
S0: IN 7         ; ожидание ввода длины слова
    AND #0x40    ; проверка на наличие введенного символа
    BEQ S0       ; нет - "Спин-луп"
    IN 6         ; ввод байта в AC
    ST (length)  ; сохраняем длину в length
    INC         ; делаем длину +1 так как в loop будет -1
    ST (leny)    ; сохраняем длину в leny
    LOOP (leny)  ; делаем длину - 1, и если length ≤ 0, ip → ip+1
    JUMP s1      ; идем к записи 1 буквы
JUMP exit       ; выходим из программы
s1: CLA         ; очистка аккумулятора
    IN 7         ; ожидание ввода 1 буквы
    AND #0x40    ; проверка на наличие введенного символа
    BEQ s1       ; нет - "Спин-луп"
    IN 6         ; ввод байта в AC
    SWAB        ; перемещение 1 буквы в старшую часть AC
    ST (res)     ; сохраняем 1 букву по нужному нам адресу
    LOOP (leny)  ; делаем длину - 1, и если length ≤ 0, ip → ip+1
    JUMP s2      ; идем к записи 2 буквы
JUMP exit       ; выходим из программы
s2: CLA         ; очистка аккумулятора
    IN 7         ; ожидание ввода 2 буквы
    AND #0x40    ; проверка на наличие введенного символа
    BEQ s2       ; нет - "Спин-луп"
    IN 6         ; ввод байта в AC
    OR (res)     ; совмещаем 2 букву с 1-ой буквой
    ST (res)+    ; сохраняем в память 2 буквы
    LOOP (leny)  ; делаем длину - 1, и если leny ≤ 0, ip → ip+1
    JUMP s1      ; идем к записи новых двух букв
exit: HLT       ; остановка программы
```

Текст исходной программы:

Адрес	Код команд ы	Мнемоника	Комментарии
29D	05C1	res 1472	Ссылка на результат
29E	05C0	length	длина
29F	05BF	leny	Переменная для длины
2A0	0200	CLA	Очистка аккумулятора
2A1	1207	IN 7	Чтение регистра состояния ВУ-3
2A2	2F40	AND #0x40	Проверка на наличие введенного символа
2A3	F0FD	BEQ IP-3	Нет - "Спин-луп"
2A4	1206	IN 6	Чтение регистра данных ВУ-3
2A5	EEF8	ST IP-8	Сохраняем длину
2A6	0700	INC	ДЛИНА +1
2A7	EEF8	ST IP-9	Сохраняем длину в переменную
2A8	8EF7	LOOP IP-9	делаем длину - 1, и если length <= 0, ip -> ip+1
2A9	CE01	JUMP IP+01	идем к записи слова
2AA	CE0C	JUMP IP +19	выходим из программы
2AB	0200	CLA	Очистка аккумулятора
2AC	1207	IN 7	Чтение регистра состояния ВУ-3
2AD	2F40	AND #0x40	Проверка на наличие введенного символа
2AE	F0FD	BEQ IP-3	Нет - "Спин-луп"
2B0	1206	IN 6	Чтение регистра состояния ВУ-3
2B1	EED6	ST IP-20	Сохраняем 1 букву
2B2	8ED5	LOOP IP-21	делаем длину - 1, и если length <= 0, ip -> ip+1
2B3	CE01	JUMP IP+01	идем к записи слова
2B4	CE09	JUMP IP +9	выходим из программы
2B5	0200	CLA	Очистка аккумулятора
2B6	1207	IN 7	Чтение регистра состояния ВУ-3
2B7	2F40	AND #0x40	Проверка на наличие введенного символа
2B8	F0FD	BEQ IP-3	Нет - "Спин-луп"
2B9	1206	IN 6	Чтение регистра данных ВУ-3

2BA	3ED7	OR IP-29	Добавление к 1 слову 2 слова
2BB	EED9	ST (IP-29) +	Сохраняем в память 2 буквы
2BC	8ECF	LOOP IP-30	делаем длину - 1, и если length <= 0, ip -> ip+1
2BD	CEE7	JUMP IP-18	Переходить в s1
2BE	0100	HLT	Остановка программы

Слово:

Плутон!

ISO-8859-5: BF DB E3 E2 DE DD 21

UTF-8: D0 9F D0 BB D1 83 D1 82 D0 BE D0 8D C0 A1

UTF-16: 04 F1 04 3B 04 43 04 42 04 3E

Область представления:

res – 11-разрядная ячейка со ссылкой на результат.

Расположение данных в памяти:

29D -2B9 – команды;

06F, 070, 071 – исходные данные;

[5C1; 5C5] – результат.

Область допустимых значений:

res – [5C1; 2047];

length – [0; 573]

введенный символ – [0;256(FF)]

4.Трассировка

Выполняемая команда		Содержимое регистров процессора после выполнения команды								Ячейка, содержимое которой изменилось	
Адрес	Код	IP	CR	AR	DR	SP	BR	AC	NZVC	Адрес	Новый код
2A0	0200	2A1	200	2A0	200	000	02A0	0000	100		
2A1	1207	2A2	1207	2A1	1207	000	02A1	0040	100		
2A2	2F40	2A3	2F40	2A2	0040	000	0040	0040	0000		
2A3	F0FD	2A4	F0FD	2A3	F0FD	000	02A3	0040	0000		
2A4	1206	2A5	1206	2A4	1206	000	02A4	0002	0000		
2A5	E8F8	2A6	E8F8	5C0	0002	000	FFF8	0002	0000	5C0	0002
2A6	0700	2A7	700	2A6	700	000	02A6	0003	0000		
2A7	E8F7	2A8	E8F7	5BF	0003	000	FFF7	0003	0000	5BF	0003
2A8	88F6	2A9	88F6	5BF	0002	000	0001	0003	0000	5BF	0002
2A9	CE01	2AB	CE01	2A9	02AB	000	0001	0003	0000		

2AB	0200	2AC	200	2AB	200	000	02AB	0000	0100		
2AC	1207	2AD	1207	2AC	1207	000	02AC	0040	0100		
2AD	2F40	2AE	2F40	2AD	0040	000	0040	0040	0000		
2AE	F0FC	2AF	F0FC	2AE	F0FC	000	02AE	0040	0000		
2AF	1206	2B0	1206	2AF	1206	000	02AF	00BF	0000		
2B0	0680	2B1	680	2B0	680	000	02B0	BF00	1000		
2B1	E8EB	2B2	E8EB	5C1	BF00	000	FFEB	BF00	1000	5C1	BF00
2B2	88EC	2B3	88EC	5BF	0001	000	0000	BF00	1000	5BF	0001
2B3	CE01	2B5	CE01	2B3	02B5	000	0001	BF00	1000		
2B5	0200	2B6	0200	2B5	0200	000	02B5	0000	0100		
2B6	1207	2B7	1207	2B6	1207	000	02B6	0040	0100		
2B7	2F40	2B8	2F40	2B7	0040	000	0040	0040	0000		
2B8	F0FC	2B9	F0FC	2B8	F0FC	000	02B8	0040	0000		
2B9	1206	2BA	1206	2B9	1206	000	02B9	00DB	0000		
2BA	38E2	2BB	38E2	5C1	BF00	000	4024	BFDB	1000		
2BB	EAE1	2BC	EAE1	5C1	BFDB	000	FFE1	BFDB	1000	29D	05C2
2BC	88E2	2BE	88E2	5BF	0000	000	FFFF	BFDB	1000	5BF	0000
2BE	0100	2BF	0100	2BE	0100	000	02BE	BFDB	1000		

Код для Доп. Задания на ассемблере:

```

ORG          0x29D    ; адрес начала программы
res:   word 0x5C1      ; начало слова
length: word 0x5C0     ; длина слова ( указывает кол-во введенных символов)
leny:   word 0x5BF     ; временная переменная для длины
stop:   word 0x5BE     ; stop-символ
temp:   word 0x5BD     ; для проверки символа на stop-слово

```

START:

```

CLA          ; очистка аккумулятора
ST (length) ; обнуляем длину

```

inputLength:

```

IN 7          ; ожидание ввода длины слова
AND #0x40     ; проверка на наличие введенного символа
BEQ inputLength ; нет - "Спин-луп"
IN 6          ; ввод байта в АС
INC           ; делаем длину +1 так как в loop будет -1
ST (leny)     ; сохраняем длину в leny
LOOP (leny)   ; делаем длину - 1, и если length <= 0, ip -> ip+1
JUMP inputStop ; идем к записи 1 буквы
JUMP exit     ; выходим из программы

```

inputStop:

```

CLA          ; очистка аккумулятора
IN 0x19      ; ожидание ввода 1 буквы
AND #0x40     ; проверка на наличие введенного символа
BEQ inputStop ; нет - "Спин-луп"
IN 0x18      ; ввод байта в АС
ST (stop)     ; сохраняем 1 букву по нужному нам адресу

```

inputS1:

```
IN 0x19          ; ожидание ввода 1 буквы
AND #0x40        ; проверка на наличие введенного символа
BEQ inputS1      ; нет - "Спин-луп"
IN 0x18          ; ввод байта в AC
ST (temp)        ; сохраняем 1 символ в temp
SWAB            ; перемещение 1 буквы в старшую часть AC
ST (res)         ; сохраняем 1 букву по нужному нам адресу
LD (length)      ; загружаем длину
INC             ; длина +1
ST (length)      ; возвращаем ее обратно
ST (temp)        ; возвращаем символ на проверку
CMP (stop)       ; проверка на стоп слово (AC - (stop)) - устанавливается флаги(Z)
BEQ check        ; проверка на Z == 1, то мы идем к выводу символов
LOOP (leny)      ; делаем leny - 1, и если length <= 0, ip -> ip+1
JUMP inputS2     ; идем к записи 2 буквы
JUMP check       ; идем к выводу символов
```

inputS2:

```
IN 0x19          ; ожидание ввода 2 буквы
AND #0x40        ; проверка на наличие введенного символа
BEQ inputS2      ; нет - "Спин-луп"
IN 0x18          ; ввод байта в AC
ST (temp)        ; сохраняем 2 символ в temp
OR (res)         ; совмещаем 2 буквы с 1-ой буквой
ST (res)+        ; сохраняем в память 2 буквы
LD (length)      ; загружаем длину
INC             ; длина +1
ST (length)      ; возвращаем ее обратно
LD (temp)        ; возвращаем символ на проверку
CMP (stop)       ; проверка на стоп слово (AC - (stop)) - устанавливается флаги(Z)
BEQ check        ; проверка на Z == 1, то мы идем к выводу символов
LOOP (leny)      ; делаем leny - 1, и если leny <= 0, ip -> ip+1
JUMP inputS1     ; идем к записи новых двух букв
```

check:

```
LD (length)      ; загружаем длину на проверку четности
ST (leny)        ; загружаем ее во временную переменную
ROR             ;
BHS outputnh     ; если нечетная длина то идем в вывод для нечетных, иначе остаемся
```

output:

```
LD -(res)        ; загружаем нашу ячейку
AND #0x00FF      ; убираем старшую часть
OUT 0x0C;        ; выводим на принтер
LD (leny)        ; загружаем длину
DEC             ; длина -1
ST (leny)        ; возвращаем ее обратно
LOOP (leny)      ; делаем leny - 1, и если leny <= 0, ip -> ip+1
JUMP output      ; идем к выводу заново
JUMP exit        ; выходим из программы
```

outputnh:

```
LD (leny)        ; загружаем длину
DEC             ; длина -1
ST (leny)        ; возвращаем ее обратно
LOOP (leny)      ; делаем leny - 1, и если leny <= 0, ip -> ip+1
JUMP output      ; идем к выводу заново
JUMP exit        ; выходим из программы
```

exit:

HLT

; остановка программы

Вывод: после выполнения данной лабораторной, я изучил основы работы ассемблера, научился пользоваться новыми командами ввода и вывода, а также познакомился ВУ-3.