

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Организация ЭВМ и систем»
Тема: Представление и обработка символьной информации с
использованием строковых команд.

Студент гр. 0382

Тихонов С.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Научится обработке символьной информации с использованием строковых команд. Также научится встраивать ассемблерный код в код языков высокого уровня по принципу *in-line*.

Задание.

Разработать программу обработки символьной информации, реализующую функции:

- инициализация (вывод титульной таблички с указанием вида преобразования и автора программы) - на ЯВУ;
- ввода строки символов, длиной не более N_{\max} (≤ 80), с клавиатуры в заданную область памяти - на ЯВУ; если длина строки превышает N_{\max} , остальные символы следует игнорировать;
- выполнение заданного в таблице 5 преобразования исходной строки с записью результата в выходную строку - на Ассемблере;
- вывода результирующей строки символов на экран и ее запись в файл - на ЯВУ.

Ассемблерную часть программы включить в программу на ЯВУ по принципу встраивания *in-line*.

Вариант 18

Заменить все русские буквы на их порядковый номер в алфавите

Замечания:

- 1) При выполнении преобразования обязательно использовать команды работы со строками;
- 2) При выполнении преобразования нельзя портить входную строку. Результат преобразования должен записываться в выходную строку.

Выполнение работы.

Выделяем память под вводимую и выводимую строку.

```
char input[81];  
char output[81];
```

Функция `main()` выводит приветственное сообщение с автором программы и описанием преобразования которое она делает. Считывает вводимую строку, открывает файл для чтения, и подключает русский язык.

```
system("chcp 1251 > nul");  
setlocale(LC_CTYPE, "rus");  
cout << "Тихонов Сергей группа 0382. Замена русских букв на их номер" << endl;  
ofstream file;  
file.open(R"(C:\Coding\Assembly\lab4\out.txt)");  
cin.getline(input, 80);
```

Затем идет код на ассемблере встроенный в программу по принципу *in-line*, которые преобразует вводную строку в соответствии с заданием.

Ассемблерная часть программы:

1. Сохраняем в индексные регистры смещение в базе данных для входной и выходной строки.

```
mov esi, offset input  
mov edi, offset output
```

2. Помечаем начало блока обработки текущего символа *loop_start* и используем команду *lodsb*, которая загружает в регистр AL байт из ячейки памяти, указываемой при помощи индексного регистра-источника.

```
loop_start :  
lodsb
```

3. Далее с помощью команды сравнения *cmp* и команд условного перехода определяется в каком диапазоне находится текущий символ в кодировке WIN-1251 (текущий символ хранится в AL). Если этот символ находится в диапазоне русских Заглавных или строчных букв, то происходит замена этого символа на его номер по алфавиту. Так как ё/Ё находится отдельно для него существует отдельная проверка вначале.

```
cmp al, '\0'
je loop_final
cmp al, 'ё'
je io
cmp al, 'а'
jl otherAB
cmp al, 'я'
jg otherAB
cmp al, 'ж'
jl a_e
cmp al, 'и'
jl j_i
cmp al, 'т'
jl i_t
cmp al, 'ь'
jl t_b
cmp al, 'я'
jle b_ia
```

4. Блок ответственный за сохранение в выходную строку нужных символов. С помощью команды *stosb* , которая сохраняет регистр AL в ячейке памяти по адресу DS:EDI. И перенаправляет выполнения программы на обработку следующего символа.

```
other :
    stosb
    jmp loop_start
```

5. Блок кода выхода из цикла обработки строки , выполняется когда встречен символ "\0". Записывает этот символ в выходную строку с помощью команды *stosb*.

```
loop_final :
    stosb
```

После ассемблерной части обработки входной строки , программа выводит выходную строку в консоль и файл и закрывает файл.

```
std::cout << output << std::endl;
file << output;
file.close();
```

Тестирование.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	А а Б б Ё	01 01 02 02 07	Программа работает корректно
2.	Russian ё	Russian 07	Программа работает корректно
3.	Ты gde	2029 gde	Программа работает корректно

Выводы.

Была изучена обработка символьной информации на языке ассемблер с использованием строковых команд.

В ходе данной лабораторной работы была разработана программа , которая формирует выходную строку только из цифр и русских букв входной строки. Программа написана на языке высокого уровня (C++), с использованием встроенного по принципу *in-line* ассемблерного кода .

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb4.cpp

```
#include <iostream>
```

```
#include <fstream>
```

```
using namespace std;
```

```

char input[81];
char output[81];
int main()
{
    system("chcp 1251 > nul");
    setlocale(LC_CTYPE, "rus");
    cout << "Тихонов Сергей группа 0382. Замена русских букв на их номер" << endl;
    ofstream file;
    file.open(R"(C:\Coding\Assembly\lab4\out.txt)");
    cin.getline(input, 80);
    __asm {
        mov ax, ds
        mov es, ax
        mov esi, offset input
        mov edi, offset output
    loop_start :
        lodsb
        cmp al, '\0'
        je loop_final
        cmp al, 'ё'
        je io
        cmp al, 'Ё'
        je io

        cmp al, 'а'
        jl otherAB
        cmp al, 'я'
        jg otherAB

        cmp al, 'ж'
        jl a_e
        cmp al, 'и'
        jl j_i
        cmp al, 'т'
        jl i_t
        cmp al, 'ь'
        jl t_b
        cmp al, 'я'
        jle b_ia

        io:

```

```

        mov ah, '7'
        mov al, '0'
        stosw
        jmp loop_start
a_e :
        sub al, 175
        mov ah, al
        mov al, '0'
        stosw
        jmp loop_start
j_i :
        sub al, 175
        mov ah, al
        add ah, 1
        mov al, '0'
        stosw
        jmp loop_start
i_t :
        sub al, 175
        sub al, 10
        mov ah, al
        add ah, 1
        mov al, '1'
        stosw
        jmp loop_start
t_b :
        sub al, 175
        sub al, 20
        mov ah, al
        add ah, 1
        mov al, '2'
        stosw
        jmp loop_start
b_ia :
        sub al, 175
        sub al, 30
        mov ah, al
        add ah, 1
        mov al, '3'
        stosw
        jmp loop_start

```

```

        otherAB :
        cmp al, 'A'
        jl other
        cmp al, 'Я'
        jg other
        add al, 32
        cmp al, 'ж'
        jl a_e
        cmp al, 'и'
        jl j_i
        cmp al, 'т'
        jl i_t
        cmp al, 'ь'
        jl t_b
        cmp al, 'я'
        jle b_ia
    other :
        stosb
        jmp loop_start
loop_final :
    stosb
};
std::cout << output << std::endl;
file << output;
file.close();
system("pause");
return 0;
}

```