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

Студентка гр. 9382 _____ Балаева М.О.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Изучить команды для работы со строками ассемблера, написать программу, обрабатывающую вводимую строку определенным способом и познакомиться с принципом встраивания in-line на примере ЯВУ C++.

Задание:

5 Вариант

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 Nmax (<=80), с клавиатуры в заданную область памяти - на ЯВУ; если длина строки превышает Nmax, 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).

5.Исключение русских букв и цифр, введенных во входной строке, при формировании выходной строки.

Ход работы:

При разработке программы были использованы следующие команды:

LODSB - копирует один байт из памяти по адресу DS:SI в регистр AL. После выполнения команды, регистр SI увеличивается на 1, если флаг DF = 0, или уменьшается на 1, если DF = 1.

STOSB - сохраняет регистр AL в ячейке памяти по адресу ES:DI. После выполнения команды, регистр DI увеличивается на 1, если флаг DF = 0, или уменьшается на 1, если DF = 1.

CLD - очищает флаг направления (DF). Такая необходимость может возникнуть при работе с цепочечными командами.

Создаются два указателя – входная строка и выходная строка. Адреса данных строк записываются в регистры rsi и rdi. Затем проходимся по всей строке

Тестирование.

Вводные данные	Результат
ASSembler123	ASSEMBLER876
asd a 1 0 asd 9 hochy sdat laby	ASD A 8 9 ASD 0 HOCHY SDAT LABY
Пожалуйста	Пожалуйста

Выводы.

В результате выполнения лабораторной работы был разработан код для определенной обработки строк. Были улучшены навыки письма в ассемблере.

Приложение.

```
#include <iostream>
#include <fstream>

#define n 80
int main() {
   system("chcp 1251 > nul");
   setlocale(LC_CTYPE, "rus");
     std::cout << "Работу выполнила: студентка группы 9382 Балаева Милана" << std::endl;
      std::cout << "Вид преобразования: 5. Преобразование всех строчных латинских букв входной строки в" << std::endl;
       std::cout << "заглавные, а десятичных цифр в инверсные, остальные символы входной строки" << std::endl;
      std::cout << "передаются в выходную строку непосредственно." << std::endl;
      char str[n + 1];
      char answer[n + 1];
      std::cout << "Введите строку для обработки:\n";
      std::cin.getline(str, n + 1);
      std::cout << "Строка до обработки:\n" << str << "\n";
      _asm{
            mov ecx, n;длина строки в ecx
            mov al, 0
            lea    si, str; кладем в ds:si адрес str
            lea di, answer; кладем в es:di адрес answer
            cld; обнуление флага направления

            digit:
            lodsb; копирует один байт из памяти по адресу ds:si в регистр al
            cmp al, '0'
            jl character
            cmp al, '9'
            jg character
            sub al, '9'
            neg al
            add al, '0'
            jmp print

            character:
            cmp al, 'a'
```

```
            jl print
            cmp al, 'z'
            jg print
            sub al, 20h

            print:
            stosb; сохраняет регистр al в ячейке памяти по адресу es:di
            loop digit

            finish_processing:
            mov al, 0
            stosb
        }
        std::cout << "Вывод обработанной строки:\n" << answer;
        std::fstream fout("output.txt");
        fout << "Строка до обработки:\n" << str << "\nВывод обработанной строки:\n" <<
answer;
        return 0;
    }
```