

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. 9382

Демин В.В.

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.

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 5

Преобразование всех строчных латинских букв входной строки в заглавные, а десятичных цифр в инверсные, остальные символы входной строки передаются в выходную строку непосредственно.

Выполнение работы.

1)Используем функцию lea, чтобы получить адрес строки входа и строки выхода.

2) CLD – очисти флаг DF. Так как DF показывает как должны происходить команды работы со строкой. Увеличивать DI на 1 или уменьшать на 1.

3) LODS – копирует один байт памяти по адресу DS:SI в регистр AL. После чего регистр SI увеличивается или уменьшаются на 1, в зависимости от флага DF.

4)STOS – копирует в ES:DI регистр AL. После чего регистр DI увеличивается или уменьшаются на 1, в зависимости от флага DF.

5) LOOP – цикл которые будет завершен пока не встретится символ конца строки.

Выводы.

В процессе выполнения работы были изучены принципы работы со строками на языке ассемблер.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: MAIN.CPP

```
#include <iostream>
#include <stdio.h>
```

```
using namespace std;
#define Nmax 80
```

```
int main() {
    system("chcp 1251 > nul");
    char strLine[Nmax];
    cin.getline(strLine, Nmax);
    cout << "Лабораторная работа№4. Выполнил Демин Виктор 9382\n"
```

"Преобразование всех строчных латинских букв входной строки в заглавные, \n"

"а десятичных цифр в инверсные, остальные символы входной строки передаются в выходную\n"

```
    "строку непосредственно.\n";
    char strOut[Nmax];
```

```
    _asm {
        mov ecx, Nmax
        sub eax, eax
        lea edi, strOut
        lea esi, strLine
        cld
```

```
        start : ; весь цикл
                lodsb
```

```
                cmp al, 0
                jne ife
                jmp exitLoopIter
            ife:
```

```
        ; основной цикл
```

```
                cmp al, '0'; проверка будет это цифра или буква
                jae digitOrletter
                jmp exitLoopIter
            digitOrletter:
```

```
                cmp al, '9'; проверка если цифра
                jbe digit
                cmp al, 'A'
                jae maybeLetter
                jmp exitLoopIter
```

```
            maybeLetter:
                cmp al, 'Z'
                jbe BigLetter
                cmp al, 'z'
                jbe maybeSmallLetter
                jmp exitLoopIter
```

```

        maybySmallLatter:
        cmp al, 'a'
        jae SmallLatter

        jmp exitLoopIter

SmallLatter : ;получение заглавной
sub al,32

        jmp exitLoopIter
BigLetter : ;получение строчной
add al, 32
        jmp exitLoopIter
digit:
        mov ah, 57
        sub ah,al
        mov al,48
        add al, ah
        jmp exitLoopIter

        exitLoopIter:
        stosb
        loop    start
};

cout <<"Строка до преобразования:"<< strLine << "\n";
cout<< "Строка после преобразования:" << strOut;

return 0;
}

```