

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

Кодуков А.В.

Преподаватель

Ефремов М.А .

Санкт-Петербург

2020

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 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).

10. Преобразование введенных во входной строке шестнадцатиричных цифр в двоичную СС, остальные символы входной строки передаются в выходную строку непосредственно.

Выполнение работы:

Использованные команды:

lea - загрузка адреса в регистр

rep, repe, repne –повторить следующую строковую операцию

loop - зациклить

Тестирование:

Ввод: А Б В Г Д Е Ё + - = , 1 2 3 4 5 6 7 8 9 А В С D E F G H I J K L M N

Вывод: А Б В Г Д Е Ё + - = , 0001 0010 0011 0100 0101 0110 0111 1000 1001
1010 1011 1100 1101 1110 1111 G H I J K L M N

Вывод:

В ходе выполнения работы были изучены команды работы со строками, реализована программа изменения введенной на ЯВУ строки.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

main.cpp

```
#include <stdio.h>

#include <iostream>

#define N 80
using namespace std;

int main() {
    system("chcp 1251 > nul");
    char _str[N + 1];
    cout << "ЛР4. Кодуков Александр 9382. \n 10. Преобразование введенных во
входной "
        "шестнадцатиричных цифр в двоичную СС, остальные символы входной
строки "
        "передаются в выходную строку непосредственно.\n";
    char str_out[N * 2 + 1];
    int i = 0;
    cin.getline(_str, N);
    _asm {
        sub eax, eax;
        mov al, 0;          in al code of str ending symbol
        mov ecx, N;         ecx = N
        lea edi, _str;      edi now points at start of _str
        repne scas;         ecx now contains N - str.length
        sub ecx, N;         ecx = -str.length
        neg ecx;            ecx = str.length
        mov edx, ecx;       edx = ecx
        sub edi, edi;       edi == 0
        sub esi, esi;       esi == 0

    traverse:
        mov edi, edx;      edi = edx
        sub edi, ecx;      edi - points at last element in str, when we
subtracting ecx we pointing to currentIdx, as ecx decreasing every iteration
        sub ax, ax
        mov al, _str[edi];  al contains currentElement

        cmp al, '0'; if symbol < 0 then its not hex number
        jl writeSymbol
        cmp al, 'F'; if symbol > F then its not hex number
        jg writeSymbol
        cmp al, '9'
        jle digit
        cmp al, 'A'
        jge letter

    digit:
        sub al, '0'
        jmp tobin

    letter:
        sub al, 'A'
        add al, 10
        jmp tobin

    tobin:
        push ecx; save counter for global loop
        mov ecx, 4
        mov edi, esi; edi <- current pointer in str_out
    tobin_0:
```

```

                                add edi, ecx
                                sub edi, 1; edi += ecx - 1 to inverse write
    shr al, 1
    jc tobin_1
    mov bl, '0'
    jmp tobin_2
    tobin_1:
        mov bl, '1'
            tobin_2:
    mov str_out[edi], bl
        add edi, 1
        sub edi, ecx; restore edi

    inc esi
    loop tobin_0
        pop ecx; restore counter
        loop traverse

        writeSymbol :
    mov str_out[esi], al
    inc esi
    loop traverse
        mov str_out[esi], 0
}
cout << str_out;
return 0;
}

```