

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.
ВАРИАНТ 22

Студент гр. 1383

Харитонов Н.М

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучить строковые команды на языке Ассемблер и применить на практике методы работы Ассемблера с ЯВУ.

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.

Вариант 22.

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

Выполнение работы.

Была выделена память для входного и выходного массивов, выведена информация об авторе программы и сути выполняемой работы. Затем происходит считывание исходной строки, которую вводит пользователь. Максимальная длина 80 символов. Далее происходит обработка строки на ассемблере.

При работе со строками необходимо прежде всего разместить регистр `esi` на начало строки-источника, а `edi` на начало строки-приемника. С метки `start` начинается основная программа. Команда `lods` загружает регистр `al` байтом памяти, на который указывает регистр `si`. Затем происходит сравнение значений

регистра al со значениями символов согласно таблице ASCII. Переходы к меткам second, next осуществляются при неправильном для данного условия диапазоне (после second переход идет к next). Если условия проходит, то выполняются действия и переход к метке next, которая записывает символ в выходную строку, соответствующий коду в регистре al. Также в метке next происходит сравнение с символом конца строки, и если нет, то происходит переход к метке start.

Таблица 1. Протокол main.cpp

№	Исходные данные	Вывод программы	Комментарий
1	AbCdEfGh1234	abcdefgh8765	Ответ корректен
2	AbCdEfGh1234/&*^%	abcdefgh8765/&*^%	Ответ корректен

Выводы.

В ходе выполнения работы были изучены строковые команды на языке Ассемблер.

ПРИЛОЖЕНИЕ А

КОД ПРОГРАММЫ

main.cpp:

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

char input[81];
char output[81];
int count = 0;

int main() {

    system("chcp 65001");

    std::string str1 = "Харитонов Никита\nГруппа
1383\nВариант 22\nПреобразование всех заглавных латинских
букв входной строки в строчные, а \n"
                        "десятичных цифр в инверсные,
остальные символы входной строки передаются в выходную
\n"
                        "строку непосредственно\n";

    std::ofstream file("output.txt");
    std::cout << str1;
    std::cout << "Введите строку: ";
    char ab[81];
    std::cin.getline(ab, 81);
    for (int i = 0; i < 81; ++i) {
        input[i] = ab[i];
        if (ab[i] == '\\0') break;
    }
```

```

}

std::cout << '\n';

int a = 0;

__asm{
    mov esi, offset input
    mov edi, offset output
    mov ecx, count
    start:
        add ecx, 1
        lods
        cmp al, 65
        jl second
        cmp al, 90
        jg second
        add al, 32
        jmp next
    second:
        cmp al, 48
        jl next
        cmp al, 57
        jg next
        mov bl, al
        sub bl, 48
        mov dl, 9
        sub dl, bl
        add dl, 48
        mov al, dl
    next:
        stosb
        cmp [esi], '\0'

```

```
        jne start
        mov count, ecx

    }

    std::cout << "Результат: ";
    std::cout << output;
    file<<output;
    file.close();
    return 0;
}
```