

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

Бочаров Г.С.

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).

2 Вариант

2. Формирование выходной строки только из цифр и латинских букв входной 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.

Для выполнения работы создается 2 строки с фиксированным количеством выделенной памяти. В одну из них загружается ввод, после чего ассемблерная вставка по символно обрабатывает одну из строк и результат записывает во вторую. Вторая строка выводится.

Тестирование.

Вводные данные	Результат
фывафasajsdajsj2kl3142r2[4] [1235]r2erlqw	asajsdajsj2kl3142r241235r2erlqw
1289!@*(ABYZ{ }[]\vabyz	1289ABYZabyz
ф і б сидели [na] {trube}	inatrube

Выводы.

В результате выполнения лабораторной работы был разработан код для определенной обработки строк. Освоен один из методов работы ассемблерными вставками.

Приложение.

Текст файла main.cpp

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

#include <Windows.h>

int main() {
    system("chcp 65001");

    std::cout << "Вид преобразования 2:\n"
                "Формирование выходной строки только из цифр и
латинских букв входной строки" << std::endl;
    std::cout << "Hello, World!" << std::endl;

    //SetConsoleCP(866);
    //SetConsoleOutputCP(866);

    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    const int size = 80;

    char s1[size + 1] = {0}; // input string
    char s2[size + 1] = {0}; // output string

    std::cin.getline(s1, size);

    //std::cout<<"size::"<<sizeof (s1)<<std::endl;

    asm("mov  %0, %%esi\n\t"      // SI = source_str
        "mov  %1, %%edi\n\t"      // DI = dest_str
        "mov  $80, %%ecx\n\t"     // ECX = Nmax

        "get_symbol:\n\t"
        "lods b(%%esi)\n\t"        // Çàãðóæààì ñèìàîë â AL
        "cmpl $'0', %%al\n\t"      // Ñðààíèèàààì ñèìàîë ñ èíàìì òèððð 0
        "jle is_character\n\t"     // Àñèè ìàíüðà, òí ìà òèððà, èààì
    ààèüðà è ìðíàððèà ìà áóèàü
        "cmpl $'9', %%al\n\t"      // Ñðààíèèàààì ñèìàîë ñ èíàìì òèððð 7
        "jge is_character\n\t"     // Àñèè áíèüðà, òí ìà àíñüìàððè-ìàÿ
    òèððà, èààì è ìðíàððèà ìà áóèàü

        "stosb (%%edi)\n\t"        // Ìàðððíàè è àóàíàó â àóðíàíóð
    ñððíèè
        "jmp final\n\t");
}
```

```

        "is_character:\n"

        "is_H:\n\t"
        "cmpb '$A', %%al\n\t "           // Cðàâíèääâì ñ ñèìâíëìì 'A'
        "jl is_L\n\t"                   // Tñèè ìâíüøâ, ìâðâðîâè è
        âûâîäó â âûðîäîð ñððîéó
        "cmpb '$Z', %%al\n\t"           // Cðàâíèääâì ñ ñèìâíëìì
        'ß'

        "jg is_L\n\t"
        "stosb (%%edi)\n\t"
        "jmp final\n"
        "is_L:\n\t"
        "cmpb '$a', %%al\n\t "           // Cðàâíèääâì ñ ñèìâíëìì 'A'
        "jl final\n\t"                   // Tñèè ìâíüøâ, ìâðâðîâè
        ê âûâîäó â âûðîäîð ñððîéó
        "cmpb '$z', %%al\n\t"           // Cðàâíèääâì ñ ñèìâíëìì
        'ß'

        "jg final\n\t"
        "stosb (%%edi)\n\t"

        "final:\n\t"
        "loop get_symbol\n\t"
        ::"r"(s1), "r"(s2)
    );

    std::cout << "vot_tak_vot ->> " << s2 << std::endl;
    return 0;
}

```