**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. 1381 |  | Новак П.И. |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2022

**Цель работы.**

Научиться работать с представлением и обработкой символьной информации с использованием строковых команд.

**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 нa ЯВУ по принципу встраивания (in-line).

**Задание**

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

**Ход выполнения работы**

В программе создаются два символьных массива input, output соответствующей длины, в главной функции main() устанавливается кодировка вывода на консоль и ввода из консоли. Вы водится приветственная информация об исполнителе и само задание. Запрашивается ввод строки из консоли, результат записывается в input, также обозначается конец строки символом ‘\0’. Создаётся и открывается файл, далее следует часть на языке Ассемблер в блоке \_\_asm. В стек кладётся адрес сегмента данных, затем извлекается в регистр es. В регистр esi помещается смещение введённой строки input, в edi - выводимой строки output. В блоке check происходит копирование в регистр al 1 байта из памяти по адресу es:si (регистр si увеличивается на 1) и проверка символа. Если символ конечный, блок \_\_asm завершается и происходит вывод полученной строки в консоль и в файл. Если символ входит в промежуток символов до ‘0’ или после ‘z’, он сразу подаётся в блок write, иначе происходит переход в блок case1. В нём происходит проверка на то, входит ли символ в промежуток после ‘9’, если так, то осуществляется переход в блок case2, иначе символ не выводится и программа переходит в check. В case2 происходит проверка на то, входит ли символ в промежуток меньше ‘a’, если так, то переход в case3, иначе - в check. В case3 так же происходит сортировка, и символ либо переходит в блок write, либо программа переходит в блок check. В блоке write происходит сохранение регистра al в ячейке памяти по адресу es:di (регистр di увеличивается на 1).

**Тестирование**

|  |  |
| --- | --- |
| Входные данные | Результат |
| I love chocolate и молоко с печеньем! | и молоко с печеньем! |
| Я lюbлu Хaгi Вaгi!!! 345\\ [lp[dsfрАНнрЛО | Я юл Хг Вг!!! \\ [[рАНнрЛО |
| Абоба Abobus^\*(^ Абобик???\эж\ээ | Абоба ^\*(^ Абобик???\эж\ээ |

**Вывод**

В ходе выполнения лабораторной работы были приобретены навыки обработки символьной информации с использованием строковых команд.

ПРИЛОЖЕНИЕ А

код программы

Название файла: lab4.cpp

#include <iostream>

#include <fstream>

#include <windows.h>

char input[81];//вводимая и выводимая строки

char output[81];

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

std::cout << "Новак Полина, гр 1381.\n Задание: Исключение латинских букв и цифр, введенных во входной строке при формировании"

"выходной строки.\n";

std::cout << "Введите строку: ";

std::cin.getline(input, 81);//считывание строки

input[strlen(input)] = '\0';

std::ofstream file;

file.open("out.txt");//создание и открытие файла

\_\_asm {

push ds

pop es//адрес даты в es

mov esi, offset input//адрес начала

mov edi, offset output

check://проверка символов

lodsb

cmp al, '\0'//если конец строки, завершение программы

je end

cmp al, '0'//если код символа меньше 0 идем в write

jl write

cmp al, 'z'//если код символа больше z идем в write

jg write

case1:

cmp al, '9'//если код символа больше 9 идем в case2

jg case2

jmp check

case2:

cmp al, 'a'//если код символа меньше a идем в case3

jl case3

jmp check

case3:

cmp al,'A'//если код символа меньше A идем в write

jl write

cmp al,'Z'//если код символа больше Z идем в write

jg write

jmp check

write://вывод символа в edi

stosb

jmp check

end :

};

std::cout << "Итоговая строка: ";

std::cout << output<<'\n';

file << output;

file.close();

return 0;

}