

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Организация ЭВМ и систем»
ТЕМА: ПРЕДСТАВЛЕНИЕ И ОБРАБОТКА СИМВОЛЬНОЙ ИНФОРМАЦИИ С
ИСПОЛЬЗОВАНИЕМ СТРОКОВЫХ КОМАНД.

Студент гр. 0382

Андрющенко К.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить работу со строками на языке программирования Ассемблер. Написать программу на языке высокого уровня с включениями Ассемблера по принципу встраивания.

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)

Индивидуальное задание.

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

Ход работы.

В ходе работы была разработана программа на языке C++ и Assembler, которая меняет заданную строчку в соответствие с индивидуальным заданием.

Инициализируем входную и выходную строку:

```
char str_input[81];  
char str_out[81];
```

В функции `main()` происходит считывание входных данных в массив символов `str_input` с помощью функции `cin.getline(str_input,81)`. Затем используется код, написанный на Ассемблере.

В нем команда `Lodsb` читает байт из входной строки (копирует один байт из памяти по адресу `DS:ESI` (32-разрядный режим адресации) в регистр `AL`, `ESI` увеличивается на 1).

Затем, с помощью инструкции `cmp` сравниваем ASCII код символа из регистра `al` с принадлежностью к нужным интервалам(HEX):

Цифры: `0x30...0x39`

Заглавные английские буквы: `0x41...0x5A`

Строчные английские буквы: `0x61...0x7A`

Если полученный символ принадлежит одному из интервалов, то записываем его в выходную строку с помощью команды `stosb` (Команда `STOSB` сохраняет регистр `AL` в ячейке памяти по адресу `ES:EDI`. После выполнения команды, регистр `EDI` увеличивается на 1).

Когда программа доходит до нулевого символа (конец строки), она прекращает цикл и возвращает полученную строку.

Затем в функции `main()` происходит вывод полученной строки и ее запись в файл.

Тестирование программы.

Проверим:

1. Удаление символов из начала и конца строки.
2. Строку, состоящую только из цифры и английских букв.
3. Пустую строку.
4. Строку, содержащую символы, ASCII код которых является крайним значением заданных интервалов.
5. Строку, содержащую различные печатные символы.

Результаты тестирования см. Таблица 1.

Таблица 1 — Тестирование программы.

№	Входная строка	Выходная строка
1	Ф + Q123 - Щ	Q123
2	XCV	XCV
3		(Корректный результат)
4	/09:@AZ['az{	09AZaz
5	//{\$фщй!QMN""@	QMN

Выводы.

В ходе данной лабораторной работы была изучена обработка символьной информации на языке Ассемблер. Была написана программа на языке высоко уровня с включениями Ассемблера по принципу встраивания.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb4.cpp

```
#include <iostream>
#include <fstream>
#include <stdio.h>
#include <windows.h>
#include <tchar.h>
using namespace std;
#define INPUT_FILENAME "C:\\Users\\Xenia\\Desktop\\prog\\file1.txt"
char str_input[81];
char str_out[81];
int main()
{
    setlocale(LC_ALL, "Russian");

    wchar_t initial_msg[] = L"Студент гр.0382 Андрющенко К.С \n2)
Формирование выходной строки только из цифр и латинских букв входной
строки.";

    wcout << initial_msg << endl;

    cin.getline(str_input, 81);

    __asm {
        push ds; Сохранение адреса начала PSP в стеке
        pop es
        mov esi, offset str_input
        mov edi, offset str_out
        mov ecx, '\\0'

        symbol_selection:
            lodsb; копирует один байт из памяти по адресу DS : ESI(32) в
AL
            cmp al, '\\0'
            je end_line
            cmp al, 0x7A; Расширенные символы ASCII Win - 1251 кириллица
+ спец.символы
            jbe number_or_latin_character; если символ не является
символом русского языка и спец.символами
            jmp symbol_selection; выбираем следующий символ

        number_or_latin_character :
            cmp al, 0x61
            jae write_symbol; строчные символы английского языка
            cmp al, 0x5A
```

```

        ja symbol_selection; выбираем следующий символ
        cmp al, 0x41
        jae write_symbol; строчные символы английского языка
        cmp al, 0x39
        ja symbol_selection; выбираем следующий символ
        cmp al, 0x30
        jae write_symbol; числа
        ; иначе
        jmp symbol_selection; выбираем следующий символ

write_symbol:
        stosb; сохраняет регистр AL в ячейке памяти по адресу ES :
EDI. После выполнения команды, регистр EDI увеличивается на 1, если флаг
DF = 0, или уменьшается на 1, если DF = 1.
        cmp ecx, [esi]
        je end_line
        jmp symbol_selection; выбираем следующий символ

end_line:
};
ofstream file(INPUT_FILENAME);
file << str_out;
cout << str_out;
return 0;
}

```