

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.

Студентка гр. 9383

Хотяков
Е.П.

Преподаватель

Ефремов
М.А.

Санкт-Петербург

2020

Цель работы.

Изучить представление и обработку символьной информации — строки символов на языке Ассемблер.

Написать программу, обрабатывающую строку по определенному принципу на языке высокого уровня (C++) с включением фрагмента на языке Ассемблер по принципу встраивания (in-line).

Текст задания.

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).

Исходные данные.

Вариант 20:

Заменить введенные во входной строке русские буквы на числа, соответствующие их номеру по алфавиту, представленному в шестнадцатиричной СС, остальные символы входной строки передать в выходную строку непосредственно.

Ход работы.

В ходе работы была разработана программа на языке C++ и Ассемблер, которая заменяет все русские буквы на их порядковый номер в алфавите, переведенный в шестнадцатеричную СС.

Русские символы(строчные и заглавные) были разбиты на несколько групп:

1. Символы, которые можно записать с помощью одного символа от 1 до 9
2. Символы, которые можно записать с помощью одного символа от А до F(т.е. от 10 до 15 в 10ой СС)
3. Символы, которые записываются как 1 + группа (1)
4. Символы, которые записываются как 1 + группа (2)
5. И отдельные группы для Я, я(20 в 16-ой СС), П, п.

Для каждой группы по таблице высчитывалось значение до символа, который соответствовал бы их порядковому номеру(для А — 1, для Б — 2 и тп). Когда число порядок доходило до F(15), то в итоговую строку добавлялся символ „1“ и дальше порядок вновь считался от 1 до F.

В файле *lb4.cpp* находится три функции:

void print_info() - она выводит на экран информацию об авторе и преобразованию.

char func(char* str_in)* — функция, которая принимает на вход исходную строку, преобразовывает как указано в задании и возвращает новую строку.

int main() - основная функция, в которой выделяется память под исходную строку, после чего происходит считывание. В ней же происходит вызов двух вышеперечисленных функций и вывод выходной строки в консоль и файл *out.txt*.

Исходный код и листинг программы представлены в приложении А.

Примеры работы программы.

Таблица 2 — Примеры работы программы.

Входная строка	Выходная строка
123 Ф И А JUI	123 15 9 1 JUI
А И Й О П Р Ш Щ Ю Я	1 9 А F 10 11 19 1A 1F 20
а Е о Л т И examp	1 6 F C 13 9 examp

Выводы.

Было изучено представление и обработка символьной информации — строки символов на языке Ассемблер.

Была написана программа, обрабатывающая строку по определенному принципу на языке высокого уровня (C++) с включением фрагмента на языке Ассемблер по принципу встраивания (in-line).

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab4.cpp

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

using namespace std;

#define N 80

void print_info()
{
    cout << "-----" << "\n";
    cout << "| Программу разработал: Хотяков Евгений, группа 9383 |" << "\n";
    cout << "| Задание: Заменить введенные во входной строке русские |" << "\n";
    cout << "| буквы на числа, соответствующие их номеру по алфавиту, |" << "\n";
    cout << "| представленному в шестнадцатиричной СС, остальные символы |" << "\n";
    cout << "| входной строки передать в выходную строку непосредственно |" << "\n";
    cout << "-----" << "\n";
    cout << "\n";
}

char* func(char * str_in){
    char *str_out = new char[N*2+1];
    asm(
        "mov r8, %0\n"
        "mov rdi, %1\n"
        "f:\n"
        "mov al, [rdi]\n" //берем текущий символ
        "inc rdi\n" //сдвигаемся к следующему символу
        "cmp al, 0\n" //проверяем является ли символ концом строки
        "je f_end\n"

        "cmp al, 0x80\n"
        "jb Write\n"

        "mov ah, al\n"
        "mov al, [rdi]\n"
        "inc rdi\n"
        "cmp ax, 0xD18F\n" // > я
        "jg Write2\n"
        "cmp ax, 0xD090\n" // < А
        "jl Write2\n"
        "cmp ax, 0xD0AF\n" // <= Я
        "jle Big_char\n" //заглавные буквы
        //строчные буквы
        "cmp ax, 0xD0B9\n" // от а до и
        "jle f1_1_9\n"
        "cmp ax, 0xD0BE\n" //от й до о
        "jle f1_A_F\n"
        "cmp ax, 0xD18F\n" //я
        "je Write_Ya\n"
        "mov bl, 0x31\n" //добавляем '1' в массив
        "mov [r8], bl\n"
        "inc r8\n"
        "cmp ax, 0xD0BF\n" //п
        "je f_pi\n"
```

```
"cmp ax, 0xD188\n" //от р до ш
"jle f2_1_9\n"
"cmp ax, 0xD18E\n" //от щ до ю
"jle f2_A_F\n"
```

```
"Big_char:"
"cmp ax, 0xD098\n" // от А до И
"jle f1_1_9\n"
"cmp ax, 0xD09E\n" //от Ё до О
"jle f1_A_F\n"
"cmp ax, 0xD0AF\n" //Я
"je Write_Ya\n"
"mov bl, 0x31\n"
"mov [r8], bl\n"
"inc r8\n"
"cmp ax, 0xD09F\n" //П
"je f_pi\n"
"cmp ax, 0xD0A8\n" //от Р до Ш
"jle f2_1_9\n"
"cmp ax, 0xD0AE\n" //от Ц до Ю
"jle f2_A_F\n"
```

```
"f1_1_9:\n"
"cmp al, 0x98\n"//и
"jle big_char_1_9\n"
"sub al, 0x7F\n"
"jmp Write\n"
"big_char_1_9:\n"
"sub al, 0x5F\n"//И
"jmp Write\n"
"f1_A_F:\n"
"cmp al, 0x9E\n" //о
"jle big_char_A_F\n"
"sub al, 0x78\n"
"jmp Write\n"
"big_char_A_F:\n"
"sub al, 0x58\n"//О
"jmp Write\n"
```

```
"f_pi:\n"
"mov al, 0x30\n" //0
"jmp Write\n"
```

```
"f2_1_9:\n"
"cmp al, 0x88\n" //ш
"jg big_char_1_9_2\n"
"sub al, 0x4F\n"
"jmp Write\n"
"big_char_1_9_2:\n"
"sub al, 0x6F\n" //Ш
"jmp Write\n"
```

```
"f2_A_F:\n"
"cmp al, 0x8E\n"//ю
"jg big_char_A_F_2\n"
"sub al, 0x48\n"
"jmp Write\n"
"big_char_A_F_2:\n"
"sub al, 0x68\n" //Ю
"jmp Write\n"
```

```

"Write_Ya:\n"
"mov al, 0x32\n" //2
"mov ah, 0x30\n"//0
"Write2:\n"
"mov [r8], al\n" //записываем младший байт
"xchg al, ah\n" //меняем местам младший и старший байт
"inc r8\n"
"Write:\n"
"mov [r8], al\n"
"inc r8\n"
"jmp f\n"
"f_end:\n"
:"=m"(str_out)
:"m"(str_in)
);
return str_out;
}
int main(){
setlocale(0, "");
char str_in[N+1];
print_info();
fgets(str_in, N, stdin);
char* str_out = func(str_in);
cout << str_out;
ofstream file;
file.open("out.txt");
file << str_out;
file.close();
delete str_out;
return 0;
}

```