# 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. 1304 Басыров В.А.

Преподаватель Кирьянчиков В. А.

Санкт-Петербург,2022

**Цель работы****:** Изучить представление символьной информации с использованием  
строковых команд. Разработать программу обработки символьной информации  
на языке Ассемблер и включить в программу на языке высокого уровня – С по принципу встраивания in-line.

**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. По заданию из таблицы 4 варианта:

Инвертирование введенных во входной строке цифр в восьмеричной СС и преобразование заглавных русских букв в строчные, остальные символы входной строки передаются в выходную строку непосредственно.  
- 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).

Выполнение работы.

Для реализации задачи, поставленной в лабораторной работе, был написан

программный код на языке С с использованием принципа встраивания

ассемблерной части. Программа была разработана в Visual Studio 2019.

После объявления всех необходимых библиотек си ,мы обьявляем 2 глобальных массива типа char.

char input\_string[81];

char output\_string[81];

Так как в задание сказано, что длина не превышает 80 символов, а последующая функция fgets считывает еще и символ переноса строки, то мы объявляем массив из 81 элемента.

Дальше в int main() мы проводим пункт 1 — инициализацию.

setlocale(LC\_ALL, "Russian.1251")

FILE\* f;

char name[] = "res.txt";

f=fopen(name, "w");

SetConsoleCP(1251);

printf("Студент-Басыров Владимир\nГруппа 1304\nЗадание-Инвертировать все цифры восьмизначной системы счисления, а также преобразовать все заглавные русские буквы к строчным\n");

В программе и в консоли устанавливается кодировка 1251. Это свяазно с тем, что в этой кодировки удобно представлять символы киррилицы — все символы в этой таблице расположены подряд,как строчные, так и заглавные. Также здесь создается и открывается файл для записи и выводиться текст задания.

Дальше согласно пункту 2 считывается с помощью функции fgets входная строка и обрубается символ перевода строки.

fgets(input\_string, 81, stdin);

input\_string[strlen(input\_string) - 1] = '\0';

Затем согласно пункту 3 идет ассемблерная вставка. После ключевого слова \_\_asm находится блок ассемблерного кода. Регистру ES присваиваем значение регистра DS, так как при работе со строками чтение происходит из памяти по адресу ES:ESI, а запись в ячейку с адресом ES:EDI. Далее присваем смещение input\_string ESI , а output\_string EDI.

push ds

pop es

mov esi, offset input\_string

mov edi, offset output\_string

Далее идет наш основный цикл line. Сначала с помощью команды lodsb в регистр al записывается исходный символ. Идет 2 последовательных сравнения, которые проверяют, находится ли символ в промежутке от 0 до 7. Стоит отметить, что так как коды символов априори неотрицательны, то можно использовать операторы беззнакового сравнения(ja,jb).Если символ не принадлежит промежутку от 0 до 7, то если код символа оказался меньше кода 0 идет переход к финальной метке(так как любой символ, код которого меньше 48 записывается в исходную строку), если же код символа больше либо равен кода 8 , то идет переход к метке F3( в этой метке будет проверяться, является ли символ — заглавной буквой русского алфавита).Если же символ является цифрой в восьмеричной системы счисления, то выполняется инвертирование, по формуле:

код\_итогового\_символа = код\_0+(код\_7-код\_исходного\_символа)

Зная, что код\_0=48 и код\_7=55, получим,что нужно из 103(код\_0+код\_7) вычесть код\_исходного\_символа, чтобы получить инвертированную цифру в восьмеричной системе счисления.

line

lodsb

cmp al, 48; сравнение с кодом символа '0'

jae F1;если код символа >=кода символ '0'

jmp Final;иначе переход к финальной метке

F1 :

cmp al, 56; сравнение с кодом символ '8'

jb F2;если код символа <кода символа '8'

jmp F3;иначе переход к следующей проверке

F2 :

neg al; -код символа

add al, 103;103(код символа '0'+код символа '7') - код исходного символа

jmp Final;переход к финальной метке

Далее идет метка F3 , в которой будет проверять , является ли символ большой буквой киррилицы. Для начала сравним, является ли символ буквой Ё, зная, что код\_символа Ё=168,а код\_символа\_ё=184, то можно к исходному символу прибавить 16.Если это символ не Ё , будет осуществлен переход к метке F4.После сравнения с кодом сивола А и Я , зная код символа А=192 и а=224, посчитав их разность получим 32, что и прибавим к коду исходного символа, таким образом и получив преобразование заглавной буквы в строчную.Затем произойдет переход к финальной метке.

F3 :

cmp al, 168; Сравнение с кодом символа 'Ё'

jne F4; если символ не Ё переход к метке

add al, 16;прибавить разность между 'ё' и 'Ё'

jmp Final;переход к финальной метке

F4 :

cmp al, 192; Сравнение с кодом символа 'А'

jae T1;Если код символа >='А'

jmp Final;иначе переход к финальной метке

T1 :

cmp al, 223; Сравнение с кодом символа 'Я'

jbe T2;Если код символа <='Я'

jmp Final;иначе переход к финальной метке

T2 :

add al, 32;Прибавить разность между 'а' и 'А'

В метке final из регистра al записывается получившееся буква с помощью команды stosb. Далее идет проверка, является ли символ нулевым(\0). Если да, происходит переход к окончанию нашей ассемблерной вставки (lineEnd). Иначе происходит переход к начальной метке line.

Final :

stosb

mov dl, 0;символ '\0'

cmp dl, [esi];проверка на конец строки

je lineEnd

jmp line

lineEnd :

};

Затем в соответствии с пунктом 4, в консоль выводится итоговая строка , а также она записывается в файл, после чего файл закрывается.

printf("%s", output\_string);

fprintf(f,"%s", output\_string);

fclose(f);

return 0;}

**Тестирование.**

|  |  |  |
| --- | --- | --- |
| Входная строка | Выходная строка | Комментарии |
| 1)1234567890 | 6543210897 | Проверка правильности инвертирования цифр восьмизначной системы счисления, а также проверка, что цифра 8,9 и 0 инвертирована не будет. |
| 2)АБВГДЕЁЖЗИКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯабвгдеёжзиклпмнопрстуфхцчшщъыьэюя | абвгдеёжзиклпмнопрстуфхцчшщъыьэюяабвгдеёжзиклпмнопрстуфхцчшщъыьэюя | Проверка, что все заглавные буквы киррилицы будут преобразованы к строчным буквам,а строчные останутся прежними. |
| 3)АБВ23578ыфпМВЦШ23н85 | абв54208ыфпмвцш54н82 | Проверка, что комбинация заглавных букв и цифр будут выведены корректно. |
| 4)АЮЯabxgAGD -=+ ~12KF | аюяabxgAGD -=+ ~65KF | Проверка, что посторонние символы(символы латиницы и другие) останутся исходными, а также проверка , что цифры и буквы киррилицы будут преобразованы корректно. |

**Вывод:**Были изучены основы ассемблерной вставки в язык программирования си.

Исходный код программы.

main.c

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <string.h>

#include <locale.h>

#include <stdlib.h>

#include "windows.h"

char input\_string[81];

char output\_string[81];

int main() {

setlocale(LC\_ALL, "Russian.1251");

FILE\* f;

f = fopen("res.txt", "w");

SetConsoleCP(1251);

printf("Студент-Басыров Владимир\nГруппа 1304\nЗадание-Инвертировать все цифры восьмизначной системы счисления, а также преобразовать все заглавные русские буквы к строчным\n");

fgets(input\_string, 81, stdin);

input\_string[strlen(input\_string) - 1] = '\0';

\_\_asm {

push ds

pop es

mov esi, offset input\_string

mov edi, offset output\_string

line :

lodsb

cmp al, 48; сравнение с кодом символа '0'

jae F1;если код символа >=кода символ '0' 1) заменить на jae F2

jmp Final; иначе переход к финальной метке

F1 :

cmp al, 56; сравнение с кодом символ '8'

jb F2;если код символа <кода символа '8'

jmp F3;иначе переход к следующей проверке

F2 :

neg al; -код символа

add al, 103; 103(код символа '0'+код символа '7') - код исходного символа

jmp Final; переход к финальной метке

F3 :

cmp al, 168; Сравнение с кодом символа 'Ё'

jne F4; если символ не Ё переход к следующей метке

add al, 16;прибавить разность между 'ё' и 'Ё'

jmp Final; переход к финальной метке

F4 :

cmp al, 192; Сравнение с кодом символа 'А'

jae T1;Если код символа >='А'

jmp Final; иначе переход к финальной метке 2) заменить на jmp F3

T1 :

cmp al, 223; Сравнение с кодом символа 'Я'

jbe T2;Если код символа <='Я'

jmp Final; иначе переход к финальной метке 3) удалить команду

T2 :

add al, 32;Прибавить разность между 'а' и 'А'

Final :

stosb

mov dl, 0; символ '\0'

cmp dl, [esi];проверка на конец строки

je lineEnd 4) заменить на jne F4

jmp line

lineEnd :

};

printf("%s", output\_string);

fprintf(f, "%s", output\_string);

fclose(f);

return 0;

}

**Защита. Ответы на вопросы.**

1)jae F2. При замене на такую команду, будет пропущена проверка на то, что код символа меньше кода символа 8. В таком случае все символы, код которых больше либо равен 48(кода символа 8) будут записаны в выходную строку по формуле 103-код\_исходного\_символа. Тут возможны 3 ситуации. Если код символа больше 47 и меньше 103, то исходному символу сопоставится символ с кодом 103-код\_исходного\_символа. Для символа с кодом 103(символ g) сопоставится 0-ой символ, программа продолжит считывание, однако функция печати печатает строку до 0 символа, то есть все последующие элементы распечатаны не будут. Для кода символа больше 103 код по формуле получится отрицательным, но так как диапазон значений кодов таблицы от 0 до 255, то это отрицательное число будет представлено как положительное число. Все тесты из таблицы тестирования не будут работать корректно, но для демонстрации всех возможных 3 случаев был разработан следующий тест.

Входные данные:/09<kлg145

Ожидалось: /79<kлg632.

Получено :/7.+ь|

Для / и 7 был получен ожидаемый результат, так как их код меньше 48. Для 9 код символа 57, по формуле получится 103-57=46(код символа .). Для k код символа 107, по формуле 103-104=-4. -410=FC16. Для положительного числа это 252. Это и есть код ь. Для л с кодом 235 получим 103-235=-132 -13210=7С16. Для положительного числа это 124, а коду 124 соответствует символ |. Для символа g как было сказано выше будет получен 0-ой символ. 6,3,2 обработуются корректно, но не будут выведены, так как строка выводится до 0 символа.

2)jmp F3. При замене на такую команду программа зациклится для любого символа большего либо равного кода 8 и меньше кода символа А. Действительно, так как в метке F3 происходит лишь сравнение с символом Ё, (а если мы попали в ветку F4, то символ точно не Ё), символ никак не меняется, то будет происходить вечная проверка символа, сначала на Ё, потом на то, больше ли код символа больше либо равен кода символа А. Это демонстрирует тесты 1,3,4. Для примера возьмем тест 1.

Входные данные:1234567890

Ожидалось :1234567890

Получено :<Ничего>

Получив 8, проведя первую проверку на принадлежания символа восьмеричной цифре и перейдя к проверке на большой символ кириллицы, программа зациклилась.

3)Удаление команды jmp Final. При удалении этой команды фактически сравнение cmp al,223 потеряет свой смысл, так как независимо от результата этой команды программа перейдет к метке T2. В любом случае к символу, который больше либо равно кода символа А будет прибавлено 32. Будут обработаны корректно все символы, кроме строчных букв кириллицы(их код больше 223 и они не являются заглавными). Для всех этих букв будет получено переполнение, то есть каждой строчной букве кириллицы будет сопоставлен код от 0 до 32. Для буквы а будет сопоставлен 0 символ, поэтому все символы, следующие после не будут выведены на экран( см пункт 1 для символа g). Тесты 2,3,4 демонстрируют эту ситуацию, однако рассмотрим собственный тест.

Входные данные:12Абяа34А

Ожидалось : 65абяа43а

Получено : 65а☺▼

Получено корректный результат для цифр и заглавных букв кириллицы. Для б с кодом символа 225 получено, что код итогового символа 225+32=257, однако так как диапазон значений от 0 до 255, то 25710=10116=0116(1 байт под символ). Этот символ соответствует ☺. Аналогично для кода я =255, 255+32=287, а 28710=11F16=1F16(1 байт) . Этому символу соответствует символ ▼. Для символа а получается 25610=10016=016. Программа выведет итоговую строку до 0 символа.

4)jne F4. Так как будет изменено это условие, цикл завершится сразу после считание любого первого элемента исходной строки. Если на вход будет получена нулевая строка, то программа будет считывать кроме 0 символа мусор, и закончит тогда, когда будет получен символ отличный от 0-го символа. Но этот мусор все-равно не будет выведен, так как строка выводится до 0-го символа(который идет первым), но этот мусор не будет выведен. Любой из всех тестов выдаст неверный результат. Рассмотрим тест 4.

Входные данные: АБВ23578ыфпМВЦШ23н85

Ожидалось : абв54208ыфпмвцш54н82

Получено : а

Прошел 1 итерацию цикла и закончил выполнение.