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. 9383 _____ Лапина А.А.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2020

Цель работы.

Научиться обрабатывать символьную информацию, изучить принцип 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 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 на ЯВУ по принципу встраивания (in-line).

Преобразование №8:

Преобразование введенных во входной строке шестнадцатиричных цифр в десятичную СС, остальные символы входной строки передаются в выходную непосредственно

Ход работы.

В функции main() выводится табличка с описанием вида преобразования и именем автора. Затем в функции main() создаются 2 строки str — размером 80 элементов и str2 — размером 160 элементов (так как строка может состоять

полностью из шестнадцатиричных цифр и для их преобразования потребуется в 2 раза больше памяти. Затем запрашивается ввод входной строки str.

Далее вызывается функция редактирующая строку strFunc(), возвращаемое значение кладется в str2. В функции strFunc() используется ассемблерная вставка, в которой происходит преобразование строки: проходим по каждому символу входной строки, проверяя является ли он шестнадцатиричной цифрой, если да, то в зависимости от символа переходим в нужную функцию — mA, mB, mC, mD, mE или mF — для букв A, B, C, D, E, F соответственно и записываем преобразованную цифру в строку, если символ не нужно преобразовывать, то просто записываем его в выходную строку. Далее в выводим результирующую строку на экран и в файл — out.txt, отчищаем динамически выделенную память.

Тестирование.

1. Входная строка: A99*F
   Выходная строка: 1099*15
2. Входная строка: ABCDEFG
   Выходная строка: 101112131415G
3. Входная строка: 0716DVxgewochew DDD
   Выходная строка: 071613Vxgewochew 131313
4. Входная строка:0Anndcewcweasd
   Выходная строка: 010nndcewcweasd
5. Входная строка:HkoDE98173
   Выходная строка: Hko131498173

Выводы.

Получены навыки обработки символьной информации, изучен принцип встраивания in-line

Содержимое файла lb4.cpp представлено в приложении А.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab4.cpp

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

#define SIZE 160

using namespace std;

char* strFunc(char* inp)
{
char* out = new char[SIZE];
asm(
"mov r8, %0\n" //записываем в регистр адрес начала выходной строки
"mov rdi, %1\n" //записываем в регистр адрес начала входной строки

"for_char:\n"
"mov al, [rdi]\n" //берем текущий символ
"inc rdi\n" //сдвигаемся к следующему символу
"cmp al, 0\n" //узнаем конец строки ли это
"je break\n" //если да, то заканчиваем

" cmp al, 0x41\n " //сравниваем с символом "A"
//je - если равно
" je mA\n" //если равно
" cmp al, 0x42\n " //сравниваем с символом "B"
" je mB\n"
" cmp al, 0x43\n " //сравниваем с символом "C"
" je mC\n"
" cmp al, 0x44\n " //сравниваем с символом "D"
" je mD\n"
" cmp al, 0x45\n " //сравниваем с символом "E"
" je mE\n"
" cmp al, 0x46\n " //сравниваем с символом "F"
" je mF\n"

"writeChar:\n"
"mov [r8], al\n" //записываем в выходную строку символ
"inc r8\n"
"jmp for_char\n"

"mA:\n"
"mov al, 0x31\n"
"mov [r8], al\n" //записываем в выходную строку символ "1"
"inc r8\n"
"mov al, 0x30\n"
"mov [r8], al\n" //записываем в выходную строку символ "0"
"inc r8\n"
"jmp for_char\n"
```

```
"mB:\n"
"mov al, 0x31\n"
"mov [r8], al\n" //записываем в выходную строку символ "1"
"inc r8\n"
"mov al, 0x31\n"
"mov [r8], al\n" //записываем в выходную строку символ "1"
"inc r8\n"
"jmp for_char\n"
"mC:\n"
"mov al, 0x31\n"
"mov [r8], al\n" //записываем в выходную строку символ "1"
"inc r8\n"
"mov al, 0x32\n"
"mov [r8], al\n" //записываем в выходную строку символ "2"
"inc r8\n"
"jmp for_char\n"

"mD:\n"
"mov al, 0x31\n"
"mov [r8], al\n" //записываем в выходную строку символ "1"
"inc r8\n"
"mov al, 0x33\n"
"mov [r8], al\n" //записываем в выходную строку символ "3"
"inc r8\n"
"jmp for_char\n"

"mE:\n"
"mov al, 0x31\n"
"mov [r8], al\n" //записываем в выходную строку символ "1"
"inc r8\n"
"mov al, 0x34\n"
"mov [r8], al\n" //записываем в выходную строку символ "4"
"inc r8\n"
"jmp for_char\n"

"mF:\n"
"mov al, 0x31\n"
"mov [r8], al\n" //записываем в выходную строку символ "1"
"inc r8\n"
"mov al, 0x35\n"
"mov [r8], al\n" //записываем в выходную строку символ "5"
"inc r8\n"
"jmp for_char\n"

"break:\n"
:"=m"(out)
:"m"(inp)
);

return out;
}

int main(){
```

```
cout <<
"_____\
n";
cout << "| |\n";
cout << "|Автор: Лапина Анастасия, вариант 8 |\n";
cout << "|Преобразование введенных во входной строке шестнадцатиричных
цифр в десятичную СС, |\n";
cout << "|остальные символы входной строки передаются в выходную
непосредственно |\n";
cout << "|
_____|\n";

int n = SIZE/2;
char str[n];
fgets(str, n, stdin);

char* str2 = strFunc(str);


ofstream file("out.txt");
file << str2 << '\n';
cout << str2 << '\n';
delete str2;
return 0;
}
```