

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ»
ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)»
Кафедра МО ЭВМ**

**ОТЧЁТ
по лабораторной работе № 4
по дисциплине «Организация ЭВМ и систем»
Тема: Представление и обработка символьной информации с
использованием стоковых команд.**

Студент гр. 1303

Преподаватель

_____ Карагезов С.Ю.

_____ Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Разработать программу на языке Ассемблера, обрабатывающую символьную информацию 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 N_{max} 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 её запись в файл — на ЯВУ.

Ассемблерную часть программы включить в программу на ЯВУ по принципу встраивания (in-line).

Задание 11: Преобразование введенных во входной строке десятичных цифр в двоичную СС, остальные символы входной строки передаются в выходную строку непосредственно.

Выполнение работы.

Для выполнения лабораторной работы был использован ЯВУ Си с подключёнными библиотеками `locale.h` и `wchar.h` для возможности работы с кириллицей.

Программа создаёт массивы для хранения входной и выходной строки, считывает строку из потока ввода, а после передаёт их в ассемблерную вставку: входная строка передаётся в регистр `rsi`, выходная строка передаётся в регистр `rdi`, также мы запрещаем использовать

компилятору регистр `rax`, поскольку он будет использоваться для считывания букв строки, и регистр `rdx`, поскольку мы его будем использовать для алгоритма перевода цифры из десятичной системы в двоичную.

Алгоритм программы в ассемблерной вставке следующий:

С помощью инструкции `lodsd` из памяти, на который указывает регистр `rsi`, считывается 4 байта (размер символа `wchar_t`), которые записываются в регистр `eax`, после чего значение в регистре `rsi` увеличивается на 4. После происходит проверка, окончилась ли строка? (является ли считанный символом нулевым). Если да, то программа записывает нулевой символ в выходную строку (запись происходит с помощью инструкции `stosd`, которая записывает данные из регистра `eax` в память, на которую указывает регистр `rdi` с увеличением последнего на 4).

Если же считанный символ не является концом строки, то происходит проверка: является ли считанный символ меньше кода символа `'0'` (48), а также больше ли считанный символ кода символа `'9'` (57), то есть является ли считанный символ цифрой: если нет, то число записывается в выходную строку и считывается новый символ; если да, то символ передаётся в регистр `rdx` из которого после вычитается символ нуля для конвертирования символа в цифру. Поскольку для хранения цифры от 0 до 9 требуется не более 4-х байт, то сдвигаем влево регистр `dl`; далее, программа совершает следующий цикл из 4-х итераций: а в `eax` записывается символ нуля, регистр `dl` циклически сдвигается влево с использованием флага `cf` на 1, если `cf` стал равен единице, то к регистру `eax` прибавляется единица (символ становится `'1'`) и символ записывается в выходную строку. По итогу цикла в строку запишется бинарная запись цифры, программа перейдёт к обработке следующего символа исходной строки.

По выходу из ассемблерной вставки в выходной строке будет находиться обработанная строка, которая будет выведена на экран.

Тестирование.

```
Карагезов Савелий, гр. 1303.  
Вариант 4: Преобразование введенных во входной строке десятичных цифр в двоичную СС, остальные символы  
входной строки передаются в выходную строку непосредственно.  
ЁЦЙУКйцу 1 2 3 4 5 6 7 8 9 0  
ЁЦЙУКйцу 0001 0010 0011 0100 0101 0110 0111 1000 1001 0000
```

Рисунок 1 – Тестирование программы со входной строкой.

Выводы

В результате лабораторной работы была изучена обработка символьной информации с использованием языка Ассемблера; разработана программа на ЯВУ (Си) с использованием ассемблерной вставки.

ПРИЛОЖЕНИЕ А

КОД ПРОГРАММ

Название файла: main.c

```
#include <locale.h>
#include <stdio.h>
#include <wchar.h>

#define N 81

int main() {
    setlocale(LC_CTYPE, "");

    wprintf(L"Карагезов Савелий, гр. 1303.\nВариант 4: Преобразование введенных "
            L"во входной строке десятичных цифр в двоичную СС, остальные символы "
            L"входной строки передаются в выходную строку непосредственно.\n");

    wchar_t in[4 * N];
    wchar_t out[N];

    fgetws(in, N, stdin);

    asm("process_str:          \n"
        " xor rax, rax          \n"
        " lodsd                 \n"
        " cmp eax, 0            \n"
        " je end_process        \n"

        "process_ch:           \n"
        " cmp eax, 48            \n" // '0'
        " jl write_ch           \n"
        " cmp eax, 57            \n" // '9'
        " jg write_ch           \n"
        " mov rdx, rax           \n"
        " sub edx, 48            \n"
        " mov rcx, 4             \n"
        " sal dl, 4              \n"
        "print_bin:             \n"
        " mov eax, 48            \n"
        " rcl dl, 1              \n"
        " jnc zero_ch            \n"
        " add eax, 1             \n"
        "zero_ch:               \n"
        " stosd                  \n"
        " loop print_bin         \n"
        " jmp process_str        \n"

        "write_ch:              \n"
        " stosd                  \n");
```

```

" jmp process_str      \n"

"end_process:          \n"
" mov eax, 0            \n"
" stosd                 \n"
:
: [in] "S"(in), [out] "D"(out)
: "rax", "rcx", "rdx");

wprintf(L"%ls\n", out);

return 0;
}

```