

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.

Студентка гр. 9383

Пономаренко С. А.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2020

Цель работы.

Написать программу на языке Ассемблер, выполняющую функции с обработкой символов 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 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).

Вариант 14.

Исключение латинских букв и цифр, введенных во входной строке при формировании выходной строки.

Ход работы.

Считываются введенные пользователем символы в строку `str`, создается пустая строка `str_2`. В регистры `rsi`, `rdi` помещаются выходные и входные данные соответственно.

Функция `start` помещает текущий символ в регистр `ah`. Производится сравнение с 0, если тек. с-л является концом строки, переходим к концу ассемблерной вставки. Тек. с-л сравнивается с «0», если он ≥ 0 , переходим к проверке на число `is_digit`, иначе записываем символ в выходную строку.

Функция `is_digit` сравнивает тек. с-л с «9», если \leq , пропускаем символ, иначе сравниваем с «A» (лат.), если \geq , переходим к проверке на заглавную латинскую букву `is_latin_big`, иначе записываем символ в выходную строку.

Функция `is_latin_big` сравнивает тек. с-л с «Z», если \leq , пропускаем символ, иначе сравниваем с «a» (лат.), если \geq , переходим к проверке на строчную латинскую букву `is_latin_small`, иначе записываем символ в выходную строку.

Функция `is_latin_small` сравниваем тек. с-л с «z», если \leq , пропускаем символ, иначе записываем символ в выходную строку.

Функция `missing` — пропуск символа. Осуществляется с помощью команды `inc` (увелечения на 1 регистра `rdi`) и перехода к начальной функции `start`.

Функция `write` — запись символа в выходную строку. Осуществляется с помощью команд `mov` (перемещения тек. с-ла в выходную строку), `inc` (регистров `rsi` и `rdi`) и перехода к начальной функции `start`.

Далее на экран выводится получившаяся строка, она же записывается в файл `file.txt`, выводятся данные о работе.

Тестирование.

1. `sdfghjkenг4567[]=-@` → `кенг[]=-@`.
2. `07Aa` → [пусто].

Вывод.

Написали программу на языке Ассемблер, выполняющую функции по обработке символов.

Приложение А.

Код написанной программы.

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

int main()
{
    std::ofstream out;
    out.open("file.txt");
    char* str = new char[80];
    std::cout << "Входная строка:\n";
    std::cin.getline(str, 80);
    char* str_2 = new char[80];
    asm(
"mov rsi, %0\n" // rsi = *str_2
"mov rdi, %1\n" // rdi = *str

"start:\n"
"mov ah, [rdi]\n"
"cmp ah, 0\n"
"je break\n"
"cmp ah, 0x30\n"
"jge is_digit\n"
"jmp write\n"

"is_digit:\n"
"cmp ah, 0x39\n"
"jle missing\n"
"cmp ah, 0x41\n"
```

"jge is_latin_big\n"

"jmp write\n"

"is_latin_big:\n"

"cmp ah, 0x5A\n"

"jle missing\n"

"cmp ah, 0x61\n"

"jge is_latin_small\n"

"jmp write\n"

"is_latin_small:\n"

"cmp ah, 0x7A\n"

"jle missing\n"

"jmp write\n"

"missing:\n"

"inc rdi\n"

"jmp start\n"

"write:\n"

"mov [rsi], ah\n"

"inc rsi\n"

"inc rdi\n"

"jmp start\n"

"break:\n"

:"=m"(str_2)

:"m"(str)

);

```
std::cout << "Выходная строка:\n" << str_2 << '\n';  
out << "Выходная строка:\n" << str_2 << '\n';  
out.close();  
std::cout << "Работа студентки группы 9383 Пономаренко Софьи.\n";  
std::cout << "14. Исключение латинских букв и цифр, введенных во  
входной строке при формировании";  
std::cout << " выходной строки.\n";  
delete[] str;  
delete[] str_2;  
return 0;  
}
```