

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. 1383

Сапожников А.Э.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Разработать программу обработки символьной информации 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} (≤ 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).

Преобразование введенных во входной строке латинских букв в русские в соответствие с правилами транслитерации, остальные символы входной строки передаются в выходную строку непосредственно.

Выполнение работы.

В качестве ЯВУ использовался ЯП С.

В начале работы программы выводится имя, номер группы и задание. Создается символьный массив `inp` и в него считывается 80 символов с консоли, в конец добавляется символ `'\0'` - символ конца строки. Вызывается функция `transliterate`. В ней создается символьный массив `res` -

результат работы программы, а также символьный массив translit - в нём какой символ кириллицы соответствует символу латиницы.

В ассемблерном блоке с помощью lodsb символ считывается в al и с помощью stosb записывается, если он равен '\0' то цикл прерывается. Другие символы сравниваются с 127, если al больше то он записывается с помощью stosb, иначе берётся два символа из массива translit и записываются в ax и если он не равен 0 то записывается символ из ax, при равенстве 0 записывается символ из изначальной строки.

По завершению работы функция возвращает res, а в функции main она выводится, затем память освобождается и программа завершается.

Таблица 1- Тестирование программы

№	Входная строка	Выходная строка	Комментарий
1	QWEqwe123	КВЕкве123	Вся латиница стала кириллицей
2	123QWEйцу	123КВЕйцу	Кириллица осталась кириллицей
3	123йцуЙЦУqwe	123йцуЙЦУкве	Всё правильно
4	abcdefghijklmnopqrstvwxyz	абсдефгхijklmn опкрстюvwхуз	Все буквы стали кириллицей
5	ABCDEFGHIJKLMNOPS TUVWXYZ	АБСДЕФГХИЖ КЛМНОПКРСТ ЮВВХУЗ	Всё правильно
	!@	!@	Символы остались теми же симловами

Программный код см. в приложении А.

Выводы.

Разработана программу обработки символьной информации с использованием строковых команд.

Исходный код программы

Название файла: lb4.c

[illegible]


```
0xD0, 0xA5, // H->X
0xD0, 0x98, // I->И
0xD0, 0x96, // J->Ж
0xD0, 0x9A, // K->К
0xD0, 0x9B, // L->Л
0xD0, 0x9C, // M->М
0xD0, 0x9D, // N->Н
0xD0, 0x9E, // O->О
0xD0, 0x9F, // P->П
0xD0, 0x9A, // Q->К
0xD0, 0xA0, // R->Р
0xD0, 0xA1, // S->С
0xD0, 0xA2, // T->Т
0xD0, 0xAE, // U->Ю
0xD0, 0x92, // V->В
0xD0, 0x92, // W->В
0xD0, 0xA5, // X->X
0xD0, 0xA3, // Y->У
0xD0, 0x97, // Z->З
0, 0,
0, 0,
0, 0,
0, 0,
0, 0,
0, 0,
0xD0, 0xB0, // a->a
0xD0, 0xB1, // b->б
0xD1, 0x81, // c->с
0xD0, 0xB4, // d->д
0xD0, 0xB5, // e->е
0xD1, 0x84, // f->ф
0xD0, 0xB3, // g->г
0xD1, 0x85, // h->х
0xD0, 0xB8, // i->и
0xD0, 0xB6, // j->ж
0xD0, 0xBA, // k->к
0xD0, 0xBB, // l->л
0xD0, 0xBC, // m->м
0xD0, 0xBD, // n->н
0xD0, 0xBE, // o->о
0xD0, 0xBF, // p->п
0xD0, 0xBA, // q->к
0xD1, 0x80, // r->р
0xD1, 0x81, // s->с
```

```

0xD1, 0x82, // t->T
0xD1, 0x8E, // u->Ю
0xD0, 0xB2, // v->В
0xD0, 0xB2, // w->В
0xD1, 0x85, // x->X
0xD1, 0x83, // y->Y
0xD0, 0xB7, // z->З
0, 0,
0, 0,
0, 0,
0, 0,
0, 0
};

```

```

asm volatile(
    "cld                                \n\t"
    "mov rsi, %[str]                   \n\t"
    "mov rdi, %[res]                   \n\t"
    "mov rcx, 80                       \n\t"
    "_loop:                            \n\t"
    "    mov rax, 0                    \n\t"
    "    lodsb                         \n\t"
    "    cmp al, 0                     \n\t"
    "    jne _check                   \n\t"
    "    mov rcx, 1                    \n\t"
    "    jmp _write                    \n\t"
    "_check:                           \n\t"
    "    cmp al, 127                  \n\t"
    "    ja _write                     \n\t"
    "    mov rdx, rax                  \n\t"
    "    mov rbx, rsi                  \n\t"
    "    mov rsi, rax                  \n\t"
    "    mov rax, [%[tr] + rsi * 2]    \n\t"
    "    mov rsi, rbx                  \n\t"
    "    cmp ax, 0                     \n\t"
    "    jne _write2                   \n\t"
    "    mov rax, rdx                  \n\t"
    "    jmp _write                    \n\t"
    "_write:                           \n\t"
    "    stosb                         \n\t"
    "    jmp _next                     \n\t"
    "_write2:                          \n\t"
    "    stosw                         \n\t"
    "_next:                           \n\t"

```



```

        "    loop _loop                                \n\t"
        :[res] "=m"(res)
        :[str] "r"(str),
        [tr] "r"(translit)
        : "rax", "rbx", "rcx", "rdx", "rdi", "rsi",
"memory", "cc"
    );
    res[80] = '\0';
    return res;
}

```

```

int main() {
    printf("A.Sapozhnikov - 1383");
    printf("Conversion of Latin letters entered
in the input string into Russian ones in accordance
with the transliteration rules, the remaining
characters of the input string are transmitted to
the output string directly\n");
    char *inp = (char *) calloc(81,
sizeof(char));
    fgets(inp, 80, stdin);
    inp[80] = '\0';

    char *res = transliterate(inp);
    printf("%s\n", res);

    free(inp);
    free(res);
    return 0;
}

```