

Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Кафедра вычислительной техники

Дисциплина «Организация ЭВМ и систем»

Лабораторная работа №2: преобразование текстовых строк

Вариант 1

Выполнил:

Съестов Дмитрий Вячеславович
Группа Р3317

Преподаватель:

Скорубский Владимир Иванович

Санкт-Петербург
2018

Цель работы

Разработать алгоритмическую схему преобразования символических строк и микропрограммы в C++, C51 с прямым доступом и указателем. Измерить время исполнения и привести объем требуемой памяти Code. Упорядочить текст лексикографически, в порядке возрастания ASCII кода “This programmer” → “aaghimootTrrs”

Исходный код

```
#include <reg51.h>

#define INDIRECT
#define LENGTH 43 //not including null terminator

char code input[LENGTH+1] = "The quick brown fox jumps over the lazy dog";
char xdata output[LENGTH+1];

char i, temp, swapped;

int main()
{
    P2 = 0;
    for (i=0; i < LENGTH+1; i++)
    {
        output[i] = input[i];
    }

    P2 = 80;
    do
    {
        swapped = 0;
        for (i = 0; i < LENGTH-1; i++)
        {
            #ifdef INDIRECT
            if (*(output + (i+1)*sizeof(int)) < *(output + i*sizeof(int)))
            {
                temp = *(output + i*sizeof(int));
                *(output + i*sizeof(int)) = *(output + (i+1)*sizeof(int));
                *(output + (i+1)*sizeof(int)) = temp;

                swapped = 1;
            }
            #else
            if (output[i+1] < output[i])
            {
                temp = output[i];
                output[i] = output[i+1];
                output[i+1] = temp;

                swapped = 1;
            }
            #endif
        }
        P2 = 0;
    } while(swapped);
    return 0;
}
```

Результаты

Упорядоченная строка:

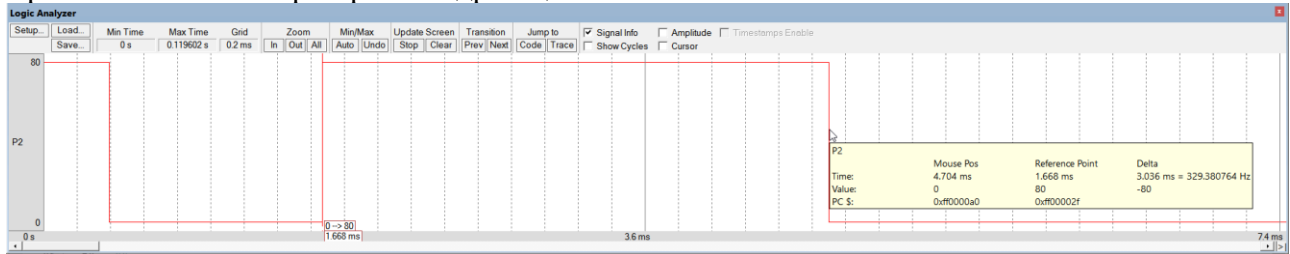
Watch 1		
Name	Value	Type
dy	C:\0x0177 dy[] "The quick brown fox jumps over t..."	array[44] of char
xB	X:\0x000000 xB[] " Tabcdeefghhijklmnooopq..."	array[44] of char
<Enter expres...		

Необходимая память:

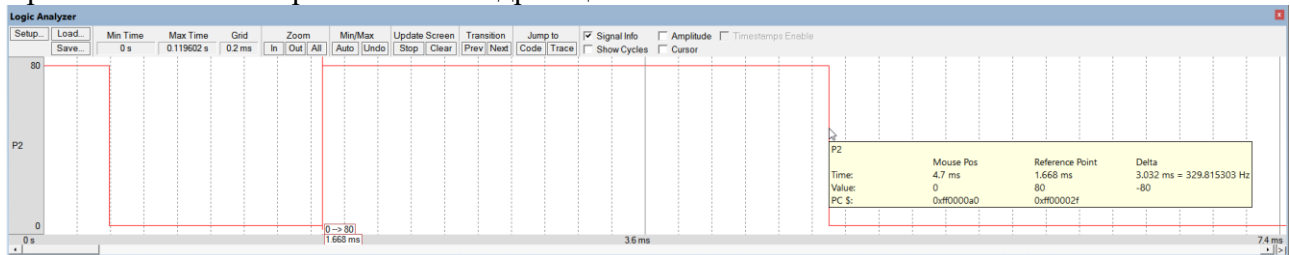
Build Output

```
Build target 'Target 1'
compiling lab2.c...
linking...
Program Size: data=12.0 xdata=44 code=207
```

Время выполнения при прямой адресации:



Время выполнения при косвенной адресации:



Вывод

Константные строки хранятся в сегменте кода, а изменяемые — в сегменте xdata.

Поэтому строку надо скопировать, прежде чем её можно будет обработать. Обработка строки с использованием прямой и косвенной адресации заняла одинаковое время, потому что прямая адресация является синтаксическим сахаром для косвенной.