

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

Санкт-Петербургский национальный исследовательский университет ИТМО

Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

**Дополнительная работа № 1. Ассемблерная вставка в С с сортировкой  
массива.**

По дисциплине «Аппаратное обеспечение вычислительных систем»

Выполнил студент группы №М3119

*Капленков Максим Олегович*

Проверил

*Прядкин Александр Олегович*



**УНИВЕРСИТЕТ ИТМО**

Санкт-Петербург

2024

## 1. Условие программы.

В моей архитектуре ARM я могу использовать регистры от w0 до w30 – регистры общего назначения.

В моей программе есть следующие методы:

- read\_n – читает из файла количество элементов в массиве
- read\_array – читает из файла все цифры и записывает их в массив
- save\_in\_file – сохраняет в файл отсортированный массив
- bubble\_sort – сама ассемблерная вставка, осуществляющая сортировку массива

используемые регистры:

w0 – n

w1 – flag for swap

w2 – индекс i

w3 – индекс i+1

w4 – array [i] или array[w10]

w5 – array [i+1] или array[w11]

нужно сделать 2 цикла, в которых будет производится сравнение элементов

mov - присвоение

add – сложение двух элементов

sub – вычитание

ldr – присвоение регистру значения

cmp – сравнение двух элементов

str – присвоение одному элементу массива значение другого элемента

## 2. Текст программы.

```
#include <stdio.h>
#include <stdlib.h>

int read_n() {
    int n;
    FILE *file = fopen("input.txt", "r");
    if(file == NULL) {
        perror("Error open file input.txt in read_n");
        exit(1);
    }
    fscanf(file, "%d", &n);
    fclose(file);
    return n;
}

int* read_array(int n) {
    int* array_input = (int*)malloc(n * 4);
    if(array_input == NULL) {
        perror("Error allocating memory");
        exit(1);
    }

    FILE *file = fopen("input.txt", "r");
    if(file == NULL) {
        perror("Error open file input.txt in read_array");
        exit(1);
    }

    char buffer[100];
```

```

fgets(buffer, 100, file);

for(int i = 0; i < n; i++) {
    fscanf(file, "%d", &array_input[i]);
}

fclose(file);
return array_input;
}

void save_in_file(int n, int *array) {
    FILE *file_output = fopen("output.txt", "w");
    if(file_output == NULL) {
        perror("Error open file output.txt");
        return;
    }
    bool flag_save_in_file = 0;

    for(int i = 0; i < n; i++) {
        fprintf(file_output, "%d ", array[i]);
        flag_save_in_file = 1;
    }
    fclose(file_output);

    if(flag_save_in_file == 1) {
        printf("__Numbers have been written to the file__\nIn order: ");
        for(int i = 0; i < n; i++)
            printf("%d ", array[i]);
        printf("\n");
    }
    else {
        printf("\nERROR! Numbers have been not written to the file\n");
        return;
    }
}

void bubble_sort(int n, int *array) {
    __asm__ (
        "mov w0, %w[n]\n" // w0 = n
        "sub w0, w0, #1\n" // w0 = n - 1 (цикл n-1 раз)
        "1:\n" // начало внешнего цикла
        "mov w1, #0\n" // w1 = 0 (flag для swap)
        "mov w2, #0\n" // w2 = 0 (index i)
        "2:\n" // начало внутреннего цикла
        "add w3, w2, #1\n" // w3 = i + 1
        "ldr w4, [%[arr], w2, uxtw #2]\n" // w4 = arr[i]
        "ldr w5, [%[arr], w3, uxtw #2] \n" // w5 = arr[i+1]
        "cmp w4, w5\n" // сравнение w4 и w5 (arr[i] и arr[i+1])
        "ble 3f\n" // if arr[i] <= arr[i + 1], тогда пропуск swap
        "str w4, [%[arr], w3, uxtw #2]\n" // arr[i] = arr[i+1]
        "str w5, [%[arr], w2, uxtw #2]\n" // arr[i + 1] = arr[i]
        "mov w1, #1\n" // swapped = 1
        "3:\n" // Метка завершения условия и обработки двух элементов
        "add w2, w2, #1\n" // i++
        "cmp w2, w0\n" // Сравнивает индекс i и общее число элементов n
        "blt 2b\n" // if i < n - 1, повтор внутреннего цикла (переход к
метке 2)
        "sub w0, w0, #1\n" // n--
        "cbnz w1, 1b\n" // if swapped != 0, повтор внешнего цикла
(переход к метке 1)
    );
}

```

```

        :
        : [arr] "r" (array), [n] "r" (n) // входные параметры
        : "w0", "w1", "w2", "w3", "w4", "w5" // используемые регистры
        );
    }

int main() {
    int n = read_n();
    int *array_input = read_array(n);

    bubble_sort(n, array_input);

    save_in_file(n, array_input);
    return 0;
}

```