

**Санкт-Петербургский национальный
исследовательский университет
информационных технологий, механики и
ОПТИКИ**

Кафедра информатики и прикладной математики

Операционные системы

Лабораторная работа 23

"Планирование потоков"

Вариант 9

Выполнил: Шкаруба Н.Е.

Группа: Р3318

2016 г

Задание

Поток main должен выполнить следующие действия:

- создать массив, размерность и элементы которого вводятся пользователем с консоли;
- вывести размерность и элементы исходного массива на консоль;
- ввести число k;
- запустить поток work;
- запустить поток MultElement;
- освобождение выходного потока stdout после вывода на консоль каждого нового элемента массива (выводить на экран поэлементно элементы массива (итогового) параллельно с работой потока work);
- известить поток MultElement о начале работы (момент запуска произойдёт после того, как будут выведены на консоль k элементов массива).

Поток work должен выполнить следующие действия:

- запросить у пользователя временной интервал, требуемый для отдыха после подготовки одного элемента в массиве;
- Поиск в массиве элементов $< A$ (разместить их в массиве слева, остальные элементы массива - справа). Элементы - вещественные числа. Число A ввести в потоке main.
- извещать поток main о новом элементе;
- после каждого готового элемента отдыхать в течение заданного интервала времени;

Поток MultElement должен выполнить следующие действия:

- ждёт от потока main сигнал о начале суммирования;
- выполнить произведение элементов итогового массива до заданной позиции k;
- вывести итоговое произведение.

Код:

```
DWORD WINAPI work(ArrStruct2 *arrStr) {
    int rest_time;
    cout << "Please enter rest time: ";
    cin >> rest_time;

    arrStr->rest_time = rest_time;

    float *left = new float[arrStr->size_array];
    float *right = new float[arrStr->size_array];

    int l = 0, r = 0;

    for (int i = 0; i < arrStr->size_array; i++) {
        if (arrStr->arr[i] < arrStr->a)
            left[l++] = arrStr->arr[i];
        else
            right[r++] = arrStr->arr[i];
    }

    for (int i = 0; i < arrStr->size_array; i++) {
        if (i < l)
            arrStr->arr[i] = left[i];
        else
            arrStr->arr[i] = right[i - l];
        ReleaseSemaphore(hWorkSemaphore, 1, NULL);
        Sleep(arrStr->rest_time * 1000);
    }
}
```

```

        return 0;
    }

    DWORD WINAPI mult(ArrStruct *arrStr) {
        WaitForSingleObject(hMultSemaphore, INFINITE);

        float mult = 1;
        for (int i = 0; i < arrStr->position_k; i++)
            mult *= arrStr->arr[i];
        cout << endl << "Multiplication = " << mult << endl;

        return 0;
    }

    int main() {
        HANDLE hWork, hMult;
        DWORD IDWork, IDMult;

        int position_k;
        volatile float *arr;

        int a;
        int size_array;

        cout << "Please enter position_k: ";
        cin >> position_k;

        cout << "Please enter size_array: ";
        cin >> size_array;

        arr = new float[size_array];

        cout << "Please enter " << size_array << " elements of array divided by space: ";
        for (int i = 0; i < size_array; i++) {
            float b;
            cin >> b;
            arr[i] = b;
        }

        cout << "Please enter a: ";
        cin >> a;

        cout << "Source array: " << endl;
        for (int i = 0; i < size_array; i++)
            cout << arr[i] << " ";
        cout << endl;

        hMultSemaphore = CreateSemaphore(NULL, 0, 1, NULL);
        hWorkSemaphore = CreateSemaphore(NULL, 0, size_array, NULL);

        ArrStruct arrStr; //MultElement
        arrStr.position_k = position_k;
        arrStr.arr = arr;

        ArrStruct2 arrStr2; // work
        arrStr2.size_array = size_array;
        arrStr2.arr = arr;
        arrStr2.a = a;

        hWork = CreateThread(NULL, 0, (LPTHREAD_START_ROUTINE)work, &arrStr2, 0, &IDWork);
        hMult = CreateThread(NULL, 0, (LPTHREAD_START_ROUTINE)mult, &arrStr, 0, &IDMult);

        for (int i = 0; i < size_array; i++) {
            WaitForSingleObject(hWorkSemaphore, INFINITE);
            if (i == 0)
                cout << "New array: " << endl;
            cout << arr[i] << " " << endl;
        }
    }

```

```

        if (i == position_k - 1)
            ReleaseSemaphore(hMultSemaphore, 1, NULL);
    }

    CloseHandle(hMultSemaphore);
    CloseHandle(hWorkSemaphore);
    CloseHandle(hMult);
    CloseHandle(hWork);
    delete[] arr;

    system("pause");
    return 0;
}

```

Результат выполнения:

```

Please enter position_k: 3
Please enter size_array: 5
Please enter 5 elements of array divided by space: 1.1 5.5 3.3 4.4 2.2
Please enter a: 3
Source array:
1.1 5.5 3.3 4.4 2.2
Please enter rest time: 2
New array:
1.1
2.2
5.5

Multiplication = 13.31
3.3
4.4
Для продолжения нажмите любую клавишу . . .

```