

# ДЗ-3

## Вариант 26

Вторая задача об Острове Сокровищ. Шайка пиратов под предводительством Джона Сильвера высадилась на берег Острова Сокровищ. Не смотря на добытую карту старого Флинта, местоположение сокровищ по-прежнему остается загадкой, поэтому искать клад приходится практически на ощупь. Так как Сильвер ходит на деревянной ноге, то самому бродить по джунглям ему не с руки. Джон Сильвер поделил остров на участки, а пиратов на небольшие группы. Каждой группе поручается искать клад на нескольких участках, а сам Сильвер ждет на берегу. Группа пиратов, обшарив одну часть острова, переходит к другой, еще необследованной части. Закончив поиски, пираты возвращаются к Сильверу и докладывают о результатах. Требуется создать многопоточное приложение с управляющим потоком, моделирующее действия Сильвера и пиратов. При решении использовать парадигму портфеля задач.

## Использованные материалы

[https://l.wzm.me/\\_coder/custom/parallel.programming/003.htm](https://l.wzm.me/_coder/custom/parallel.programming/003.htm)

## Код программы

```
#include <iostream>
#include <algorithm>
#include <pthread.h>

const int height = 1024, width = 1024;

pthread_mutex_t mutex;

int island[height][width] = {0};

static int begin = 0;

int x, y;

bool found = false;

void CheckBlockOfLand(int row) {
```

```

    for (int i = 0; i < width; ++i) {
        if (found)
            break;
        if (island[row][i] == 1) {
            x = found ? x : row;
            y = found ? y : i;
            found = true;
        }
    }
}

void *func(void *param) {
    int tempBegin = 0;

    while (true) {

        pthread_mutex_lock(&mutex);
        tempBegin = begin;
        begin++;
        pthread_mutex_unlock(&mutex);

        CheckBlockOfLand(tempBegin);
        if (begin >= height) break;
    }
}

int main() {
    std::cout << "Enter the number of threads" << std::endl;
    std::string line;
    std::cin >> line;
    int numberOfThreads = std::stoi(line);

    srand(time(NULL));
    int rnd1 = rand() % height, rnd2 = rand() % width;
    island[rnd1][rnd2] = 1;

    pthread_mutex_init(&mutex, nullptr);

    pthread_t threads[numberOfThreads - 1];

    for (int i = 0; i < numberOfThreads - 1; ++i)
        pthread_create(&threads[i], NULL, func, 0);

    for (int i = 0; i < numberOfThreads - 1; ++i) {
        pthread_join(threads[i], NULL);
    }

    std::cout << "The treasure was found at x = " << x << " y = " << y << std::endl;

    return 0;
}

```

Для работы с потоками я использовал POSIX threads, а также используется mutex для безопасной работы с разделенными переменными.

Суть программы заключается в том, что есть двумерный массив целых чисел размером 1024x1024, заполненный нулями, при запуске программа запрашивает количество потоков, затем в случайную позицию помещается единица. Потом создается массив потоков размером  $n =$  введенное пользователем число - 1, тк у нас еще есть основной поток. Затем в цикле создаются потоки с методом func. Потоки выполняют работу по поиску сокровища в строке массива, по завершении переходит на другую строку.