

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных
систем

Лабораторная работа № 6

по дисциплине: ООП
тема: «**Потоки в C++**»

Выполнил: ст. группы ПВ202
Аладиб язан
Проверил:
Буханов Д.Г.

Белгород 2022

Цель работы: изучение основных возможностей потоков управления и потоков ввода-вывода. Получение навыков работы со стандартными средствами управления потоками в C++11. Знакомство с классом Thread и стандартными средствами синхронизации потоков.

Задание:

1. Изучить основные классы и их возможности работы с потоками в C++11.
2. Разработать программу в соответствии с вариантом задания. Программа должна содержать 2 потока Thread для реализации основного задания лабораторной работы. Вывод организовать в отдельном потоке.
3. Реализовать классы и выполнить перегрузку оператора функтора для реализации поставленной основной задачи.
4. Разработать программу в соответствии с вариантом задания (номер варианта + 3), используя API CreateThread.
5. Сделать выводы о проделанной работе.

Задание Варианта :

Один поток удаляет пробелы в строке и вставляет их в случайное место, а другой поток выполняет циклический сдвиг текста. Произвести синхронный вывод при каждой итерации. Показать выполнение работы программы в синхронном и асинхронном режимах.

Выполнение работы:

```
#include <windows.h>
#include <iostream>
#include <cstdlib>
#include <string>
#include <ctime>
#include <tchar.h>

#define MAXSIZE 255
using namespace std;

char text[MAXSIZE] = "hello world";

DWORD WINAPI thread_1(LPVOID lpParam) {
    int i = 0;
    for (int i = 0; i < 100; i++) {
        HANDLE mutex = OpenMutex(SYNCHRONIZE, FALSE, _T("mutex"));
        int pos = strlen(text);
        for (int j = pos - 1; j != -1; j--) {
            text[j + 1] = text[j];
        }
        text[0] = ' ';
        text[pos + 2] = '\\0';
        i++;
        cout << text << endl;
        ReleaseMutex(mutex);
    }
    return 0;
}

DWORD WINAPI thread_2(LPVOID lpParam) {
    int i = 0;
    for (int i = 0; i < 100; i++) {
        HANDLE mutex = OpenMutex(SYNCHRONIZE, FALSE, _T("mutex"));
        srand(unsigned(time(NULL)));
        int j = 0;
        if (strlen(text) != 0) {
            j = int(rand() % strlen(text));
            if (text[j] == ' ') {
                int i;
                for (i = j; text[i] != '\\0'; i++)
                    text[i] = text[i + 1];
                text[i] = '\\0';

                j = int(rand() % strlen(text));
                int pos = strlen(text);
                for (int k = pos - 1; k != j - 1; k--) {
```

```

        text[k + 1] = text[k];
    }
    text[j] = ' ';
    text[pos + 2] = '\\0';
}
}
ReleaseMutex(mutex);
}
return 0;
}

int main() {
    SetConsoleOutputCP(CP_UTF8); //Подключение русского языка
    HANDLE arr[2];
    int data_1 = 0, data_2 = 0;

    HANDLE handle_1 = CreateThread(NULL, 0, thread_1, &data_1, 0, NULL);
    HANDLE handle_2 = CreateThread(NULL, 0, thread_2, &data_2, 0, NULL);
    arr[0] = handle_1;
    arr[1] = handle_2;
    WaitForMultipleObjects(2, arr, TRUE, INFINITE);
    CloseHandle(handle_1);
    CloseHandle(handle_2);
    return 0;
}

```