

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)



ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ

Лабораторная работа №6

по дисциплине: ООП

тема: «Потоки в C++»

Выполнил: ст. группы ПВ-223
Пахомов Владислав Андреевич

Проверили:
пр. Черников Сергей Викторович

Белгород 2024 г.

Лабораторная работа №6

«Потоки в C++»

Вариант 3

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

Разработать программу в соответствии с вариантом задания (номер варианта + 3), используя API CreateThread.

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.

Асинхронная программа:

```
#include <iostream>
#include <mutex>
#include <thread>
#include <vector>
#include <algorithm>
#include <windows.h>

void space_adder(std::vector<char>& val, std::mutex& m) {
    while (1) {
        m.lock();

        auto spaceFind = std::find(val.begin(), val.end(), ' ');
        while (spaceFind != val.end()) {
            val.erase(spaceFind);

            spaceFind = std::find(val.begin(), val.end(), ' ');
        }

        int newSpaces = val.size();
        for (int i = 0; i < newSpaces; i++) {
            int randIndex = val.size() * 1.0 * rand() / RAND_MAX;

            val.insert(val.begin() + randIndex, ' ');
        }

        m.unlock();

        std::this_thread::sleep_for(std::chrono::milliseconds(500));
    }
}

void move_val(std::vector<char>& val, std::mutex& m) {
    while (1) {
        m.lock();
```

```

        char b = val[0];
        val.erase(val.begin());
        val.push_back(b);

        m.unlock();

        std::this_thread::sleep_for(std::chrono::milliseconds(500));
    }
}

void outputter(std::vector<char>& val, std::mutex& m) {
    while (1) {
        m.lock();

        for (auto& ch : val) {
            std::cout << ch;
        }
        std::cout << "\n";

        m.unlock();

        std::this_thread::sleep_for(std::chrono::milliseconds(500));
    }
}

int main() {
    SetConsoleOutputCP(CP_UTF8);

    std::mutex m;
    std::vector<char> val = {'H', 'e', 'l', 'l', 'o'};
    std::thread th1(space_adder, std::ref(val), std::ref(m));
    std::thread th2(move_val, std::ref(val), std::ref(m));
    std::thread th3(outputter, std::ref(val), std::ref(m));

    th1.detach();
    th2.detach();
    th3.detach();

    int a;
    std::cin >> a;

    return 0;
}

```

Вывод в консоль:

```

He  l lo
He  ll  o
e  ll  oH
    lloHe
    lloHe
ll oH e

```

```
l   oHel
lo H   el
oHel   l
    H el lo
Hel l   o
    e   ll oH
    e l loH
e   l l o H
l lo H   e
    lo H e l
lo   H el
    o He ll
o H   el l
He   l lo
```

Синхронная программа:

```
#include <iostream>
#include <mutex>
#include <thread>
#include <vector>
#include <algorithm>
#include <windows.h>

int main() {
    SetConsoleOutputCP(CP_UTF8);

    std::vector<char> val = {'H', 'e', 'l', 'l', 'o'};

    while (1) {
        auto spaceFind = std::find(val.begin(), val.end(), ' ');
        while (spaceFind != val.end()) {
            val.erase(spaceFind);

            spaceFind = std::find(val.begin(), val.end(), ' ');
        }

        int newSpaces = val.size();
        for (int i = 0; i < newSpaces; i++) {
            int randIndex = val.size() * 1.0 * rand() / RAND_MAX;

            val.insert(val.begin() + randIndex, ' ');
        }

        char b = val[0];
        val.erase(val.begin());
        val.push_back(b);

        for (auto& ch : val) {
            std::cout << ch;
        }
        std::cout << "\n";
    }
}
```

```
std::this_thread::sleep_for(std::chrono::milliseconds(500));  
}  
  
return 0;  
}
```

Вывод в консоль:

```
He l lo  
e ll oH  
e ll o H  
elloH  
el lo H  
l loHe  
l l oHe  
l o Hel  
oHe ll  
o He ll  
He l lo  
H el lo  
H e llo  
e l l oH  
ll o He  
ll o H e  
l o Hel  
l oH el  
o He ll  
oH e ll
```

Ссылка на репозиторий

Вывод: в ходе лабораторной работы изучены основные возможности потоков управления и потоков ввода-вывода. Получены навыки работы со стандартными средствами управления потоками в C++11. Познакомились с классом Thread и стандартными средствами синхронизации потоков.