

Министерство цифрового развития, связи
и массовых коммуникаций Российской Федерации

Сибирский государственный университет
телекоммуникаций и информатики

Кафедра прикладной математики и кибернетики

ЛАБОРАТОРНАЯ РАБОТА №11

По дисциплине: «Операционные системы»

Выполнили:

Студенты 3 курса группы ИП-111
Корнилов А.А.,
Попов М.И.,
Толкач А.А.

Проверил:

Профессор кафедры ПМиК
Малков Е.А.

Новосибирск, 2023

Задание: реализуйте алгоритм “производитель-потребитель” для конечного буфера и циклического буфера.

Цель: получение навыков синхронизации с использованием мьютексов и семафоров.

Выполнение работы:

Была написана программа генерации и вывода чисел для конечного буфера (Листинг 1) и бесконечного буфера (Листинг 2)

```
#include <iostream>
#include <queue>
#include <thread>
#include <mutex>
#include <condition_variable>
using namespace std;
const int buffer_size = 5; // Размер буфера

queue<int> buffer; // Очередь для буфера
mutex mtx; // Мьютекс для синхронизации доступа к буферу
condition_variable buffer_empty; // Условная переменная для оповещения о пустом буфере
condition_variable buffer_full; // Условная переменная для оповещения о полном буфере

// Функция производителя
void producer(int id) {
    for (int i = 0; i < 10; ++i) {
        unique_lock<mutex> lock(mtx);

        // Проверка, если буфер полон, ждем, пока не появится место
        buffer_full.wait(lock, [] { return buffer.size() < buffer_size; });

        // Генерация элемента
        int item = rand() % 100;
        cout << "Изготовитель №" << id << " придумал число: " << item << endl;

        // Добавление элемента в буфер
        buffer.push(item);

        // Оповещаем потребителя о наличии нового элемента
        buffer_empty.notify_all();
    }
}

// Функция потребителя
void consumer(int id) {
    for (int i = 0; i < 10; ++i) {
        unique_lock<mutex> lock(mtx);

        // Проверка, если буфер пуст, ждем, пока не появится элемент
        buffer_empty.wait(lock, [] { return !buffer.empty(); });

        // Извлечение элемента из буфера
        int item = buffer.front();
        buffer.pop();
        cout << "Потребитель №" << id << " получил число: " << item << endl;
    }
}
```

```

        // Оповещаем производителя о наличии места в буфере
        buffer_full.notify_all();
    }
}

int main() {
    // Создание потоков производителей и потребителей
    thread producer1(producer, 1);
    thread producer2(producer, 2);
    thread consumer1(consumer, 1);
    thread consumer2(consumer, 2);

    // Ожидание завершения потоков
    producer1.join();
    producer2.join();
    consumer1.join();
    consumer2.join();

    return 0;
}

```

Листинг 1 – программа lab11_1.c

Команда компиляции и результат работы программы:

```

miron@DESKTOP-UMC1Q46:/mnt/u/Documents/B BY3/OS/11$ g++ -std=c++11 -
pthread lab11_1.cpp -o lab11_1
miron@DESKTOP-UMC1Q46:/mnt/u/Documents/B BY3/OS/11$ ./lab11_1
Изготовитель №1 придумал число: 83
Потребитель №2 получил число: 83
Изготовитель №1 придумал число: 86
Потребитель №1 получил число: 86
Изготовитель №1 придумал число: 77
Потребитель №2 получил число: 77
Изготовитель №1 придумал число: 15
Потребитель №2 получил число: 15
Изготовитель №2 придумал число: 93
Потребитель №1 получил число: 93
Изготовитель №2 придумал число: 35
Изготовитель №2 придумал число: 86
Изготовитель №2 придумал число: 92
Изготовитель №2 придумал число: 49
Изготовитель №2 придумал число: 21
Потребитель №1 получил число: 35
Изготовитель №2 придумал число: 62
Потребитель №2 получил число: 86

```

Изготовитель №2 придумал число: 27
Потребитель №1 получил число: 92
Потребитель №1 получил число: 49
Потребитель №1 получил число: 21
Потребитель №1 получил число: 62
Потребитель №1 получил число: 27
Изготовитель №2 придумал число: 90
Потребитель №1 получил число: 90
Изготовитель №1 придумал число: 59
Потребитель №1 получил число: 59
Изготовитель №1 придумал число: 63
Потребитель №2 получил число: 63
Изготовитель №1 придумал число: 26
Потребитель №2 получил число: 26
Изготовитель №1 придумал число: 40
Потребитель №2 получил число: 40
Изготовитель №1 придумал число: 26
Потребитель №2 получил число: 26
Изготовитель №1 придумал число: 72
Потребитель №2 получил число: 72
Изготовитель №2 придумал число: 36
Потребитель №2 получил число: 36

```
#include <iostream>
#include <queue>
#include <thread>
#include <mutex>
#include <condition_variable>

const int bufferSize = 5; // Размер буфера

std::queue<int> buffer; // Очередь буфера
std::mutex mtx; // Мьютекс для обеспечения безопасности доступа к буферу
std::condition_variable producerCV, consumerCV; // Условные переменные для синхронизации
производителя и потребителя

void producer() {
    for (int i = 0; ; ++i) {
        std::unique_lock<std::mutex> lock(mtx);
        producerCV.wait(lock, []() { return buffer.size() < bufferSize; }); // Ждем,
пока буфер не станет меньше максимального размера
        buffer.push(i);
        std::cout << "Производитель произвел: " << i << std::endl;
        lock.unlock();
        consumerCV.notify_one(); // Уведомляем потребителя
    }
}
```

```

void consumer() {
    for (;;) {
        std::unique_lock<std::mutex> lock(mtx);
        consumerCV.wait(lock, []() { return !buffer.empty(); }); // Ждем, пока буфер не
        станет непустым
        int data = buffer.front();
        buffer.pop();
        std::cout << "Потребитель потребил: " << data << std::endl;
        lock.unlock();
        producerCV.notify_one(); // Уведомляем производителя
    }
}

int main() {
    std::thread producerThread(producer);
    std::thread consumerThread(consumer);

    producerThread.join();
    consumerThread.join();

    return 0;
}

```

Листинг 2 – программа lab11_2.cpp

Команда компиляции и результат работы программы:

```

miron@DESKTOP-UMC1Q46:/mnt/u/Documents/B BY3/OS/11$ g++ -std=c++11 -
pthread lab11_2.cpp -o lab11_2
miron@DESKTOP-UMC1Q46:/mnt/u/Documents/B BY3/OS/11$ ./lab11_2
Производитель произвел: 0
Производитель произвел: 1
Производитель произвел: 2
Производитель произвел: 3
Производитель произвел: 4
Потребитель потребил: 0
Потребитель потребил: 1
Потребитель потребил: 2
Потребитель потребил: 3
Потребитель потребил: 4
Производитель произвел: 5
Производитель произвел: 6
Производитель произвел: 7
Производитель произвел: 8
Производитель произвел: 9
Потребитель потребил: 5
Потребитель потребил: 6

```

Потребитель потребил: 7
Потребитель потребил: 8
Потребитель потребил: 9
Производитель произвел: 10
Производитель произвел: 11
Производитель произвел: 12
Производитель произвел: 13
Производитель произвел: 14
Потребитель потребил: 10
Потребитель потребил: 11
Потребитель потребил: 12
Потребитель потребил: 13
Потребитель потребил: 14
Производитель произвел: 15
Производитель произвел: 16
Производитель произвел: 17
Производитель произвел: 18
Производитель произвел: 19
Потребитель потребил: 15
Потребитель потребил: 16
Потребитель потребил: 17
Потребитель потребил: 18
Потребитель потребил: 19
...