Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное   
образовательное учреждение высшего образования   
«Самарский национальный исследовательский университет   
имени академика С.П. Королева»

Институт информатики, математики и электроники

Факультет информатики

Кафедра технической кибернетики

Отчет по лабораторной работе № 5

Дисциплина: «Операционные системы»

Тема «Использование семафоров и мьютексов для

клиент-серверного взаимодействия»

Выполнил: Абрамов Данила Олегович

Группа: 6208-010302D

Самара, 2021

**Задание на лабораторную работу**

**Вариант 12** Задача о Винни-Пухе и Пятачке. В одном лесу живут n пчел, Винни-Пух и его друг Пятачок, которые используют один горшок меда, вместимостью Н глотков. Сначала горшок пустой. Пока горшок не наполнится, медведь спит. Как только горшок заполняется, медведь просыпается и съедает весь мед, после чего снова засыпает. Каждая пчела многократно собирает по одному глотку меда и кладет его в горшок. Пчела, которая приносит последнюю порцию меда, будит Винни-Пуха или Пятачка. Смоделировать поведение пчел, медведя и его друга. Допустимо использовать в рамках 1 процесса пчелы ряд потоков, моделирующих отдельную пчелу. Винни-Пух спит в 2 раза больше и ест в 4 раза больше, чем Пятачок. Если медведь не выспался, пчела его разбудить не может, то есть попеременный доступ к горшку с медом.

**Текст программы**

Server

#include <iostream>

#pragma comment(lib, "ws2\_32.lib")

#include <winsock2.h>

#include <iostream>

#pragma warning(disable: 4996)

#include <mutex>

#include <chrono>

#include <thread>

using namespace std;

int barrel\_fullness = 0;

bool WinnieThePooh\_sleep = true;

bool Pyatochek\_sleep = true;

bool WinnieThePooh\_sleep\_got\_enough\_sleep = false;

bool Pyatochek\_sleep\_got\_enough\_sleep = false;

mutex mtx;

void Extraction(int i, int H);

void Eating();

void EatingPig();

int main()

{

srand(time(0));

setlocale(0, "");

WSAData wsaData; // ссылка на структуру wsaData

WORD word = 258; // запрашиваемая версия библиотеки winsock

if (WSAStartup(word, &wsaData)) { // если метод не вернёт 0 то библиотека загрузилась неудачно

cout << "Error: не удалось подключить библиотеку\n";

return 0;

}

SOCKADDR\_IN sockAddr;

int sizeofaddr = sizeof(sockAddr);

// заполняем информацию об адресе сокета

sockAddr.sin\_addr.S\_un.S\_addr = inet\_addr("127.1.1.1"); // IP

sockAddr.sin\_port = htons(1111); // порт

sockAddr.sin\_family = 2; // семейство интернет протоколов

SOCKET socketListen = socket(2, 1, 0); // создаем сокет sListen которому присваиваем результат выполнения функции socket

bind(socketListen, (SOCKADDR\*)&sockAddr, sizeof(sockAddr)); // привязываем адрес сокета; (SOCKADDR\*)&addr - указатель на структуру sockaddr

listen(socketListen, 0x7ffffff); // прослушивание порта в ожидании соединения со стороны клиента;

SOCKET newConnection; // объявляем новый сокет для удерживания соединения с клиентом

newConnection = accept(socketListen, (SOCKADDR\*)&sockAddr, &sizeofaddr); // функция accept возращает указатель на новый сокет который можно использовать для общения с клиентом

if (!newConnection) cout << "Error: не удалось подключиться\n";

else {

cout << "Пчёлы подключились!\n";

char N[256];

recv(newConnection, N, sizeof(N), 0);

int n = atoi(N);

recv(newConnection, N, sizeof(N), 0);

int H = atoi(N);

thread\* bees = new thread[n];

thread WinnieThePooh = thread(Eating);

thread Pyatochek = thread(EatingPig);

for (int i = 0; i - n; ++i) bees[i] = thread(Extraction, i, H);

for (int i = 0; i - n; ++i) bees[i].join();

WinnieThePooh.join();

Pyatochek.join();

}

return 0;

}

void Extraction(int i, int H) {

while (1) {

this\_thread::sleep\_for(chrono::seconds(1));

mtx.lock();

if (barrel\_fullness < H) {

++barrel\_fullness;

if (barrel\_fullness == H) {

cout << "Пчела " << i << " принесла каплю\n";

while (!WinnieThePooh\_sleep\_got\_enough\_sleep && !Pyatochek\_sleep\_got\_enough\_sleep) {

this\_thread::sleep\_for(chrono::seconds(1));

if (WinnieThePooh\_sleep\_got\_enough\_sleep) {

WinnieThePooh\_sleep = false;

WinnieThePooh\_sleep\_got\_enough\_sleep = false;

cout << "Пчела " << i << " разбудила Винни-Пуxа\n";

break;

}

if (Pyatochek\_sleep\_got\_enough\_sleep) {

Pyatochek\_sleep = false;

Pyatochek\_sleep\_got\_enough\_sleep = false;

cout << "Пчела " << i << " разбудила Пятачка\n";

break;

}

}

if (WinnieThePooh\_sleep\_got\_enough\_sleep && Pyatochek\_sleep\_got\_enough\_sleep) {

int flag = rand();

if (flag & 1) {

Pyatochek\_sleep = false;

Pyatochek\_sleep\_got\_enough\_sleep = false;

cout << "Пчела " << i << " разбудила Пятачка\n";

}

else {

WinnieThePooh\_sleep = false;

WinnieThePooh\_sleep\_got\_enough\_sleep = false;

cout << "Пчела " << i << " разбудила Винни-Пуxа\n";

}

}

else if (WinnieThePooh\_sleep\_got\_enough\_sleep) {

WinnieThePooh\_sleep = false;

WinnieThePooh\_sleep\_got\_enough\_sleep = false;

cout << "Пчела " << i << " разбудила Винни-Пуxа\n";

}

else if (Pyatochek\_sleep\_got\_enough\_sleep) {

Pyatochek\_sleep = false;

Pyatochek\_sleep\_got\_enough\_sleep = false;

cout << "Пчела " << i << " разбудила Пятачка\n";

}

}

else cout << "Пчела " << i << " принесла каплю\n";

}

mtx.unlock();

}

}

void Eating() {

while (1) {

if (!WinnieThePooh\_sleep) {

this\_thread::sleep\_for(chrono::seconds(1));

cout << "Винни-Пуx всё съел\n";

barrel\_fullness = 0;

WinnieThePooh\_sleep = true;

WinnieThePooh\_sleep\_got\_enough\_sleep = false;

}

if (!WinnieThePooh\_sleep\_got\_enough\_sleep) {

mtx.lock();

cout << "Винни-Пуx лёг спать\n";

mtx.unlock();

this\_thread::sleep\_for(chrono::seconds(6));

cout << "Винни-Пуx выспался\n";

WinnieThePooh\_sleep\_got\_enough\_sleep = true;

}

this\_thread::sleep\_for(chrono::seconds(1));

}

}

void EatingPig() {

while (1) {

if (!Pyatochek\_sleep) {

this\_thread::sleep\_for(chrono::seconds(1));

cout << "Пятачёк съел 1/4 горшка\n";

barrel\_fullness = barrel\_fullness \* 0.75;

Pyatochek\_sleep = true;

Pyatochek\_sleep\_got\_enough\_sleep = false;

}

if (!Pyatochek\_sleep\_got\_enough\_sleep) {

mtx.lock();

cout << "Пятачёк лёг спать\n";

mtx.unlock();

this\_thread::sleep\_for(chrono::seconds(3));

cout << "Пятачёк выспался\n";

Pyatochek\_sleep\_got\_enough\_sleep = true;

}

this\_thread::sleep\_for(chrono::seconds(1));

}

}

Client

#include <iostream>

#pragma comment(lib, "ws2\_32.lib")

#include <winsock2.h>

#include <iostream>

#pragma warning(disable: 4996)

using namespace std;

int main()

{

setlocale(0, "");

WSAData wsaData;

WORD DLLVersion = 258;

if (WSAStartup(DLLVersion, &wsaData)) cout << "Error: не удалось подключить библиотеку\n";

else {

SOCKADDR\_IN addr;

int sizeofaddr = sizeof(addr);

addr.sin\_addr.S\_un.S\_addr = inet\_addr("127.1.1.1");

addr.sin\_port = htons(1111);

addr.sin\_family = 2;

SOCKET Connection = socket(2, 1, 0);

if (connect(Connection, (SOCKADDR\*)&addr, sizeof(addr))) cout << "Error: не удалось подключиться к серверу\n";

else {

cout << "Подключение произошло успешно!\n";

char N [256];

cout << "Введите число пчёл: ";

cin >> N;

while (atoi(N) < 1 || cin.peek() != '\n') {

cin.clear();

while (cin.get() != '\n');

cout << "Error: Введите натуральное положительное число: ";

cin >> N;

}

send(Connection, N, sizeof(N), 0);

cout << "Введите число глотков в горшке: ";

cin >> N;

while (atoi(N) < 1 || (cin.peek() != '\n')) {

cin.clear();

while (cin.get() != '\n');

cout << "Error: Введите натуральное положительное число: ";

cin >> N;

}

send(Connection, N, sizeof(N), 0);

}

}

}