**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**Выполнила**: Асташкина Мария Николаевна   
**Группа**: БПИ 196  
**Вариант: 2**

**ЗАДАНИЕ**

Задача о Винни-Пухе или правильные пчелы. В одном лесу живут n пчел и один медведь, 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 медведя. Создать многопоточное приложение, моделирующее поведение пчел и медведя.

**ИСХОДНЫЙ КОД**

#include "pthread.h"

#include <iostream>

#include <unistd.h>

#include <ctime>

using namespace std;

int n, N, swallowCount, i;

int startTime;

pthread\_t \*bees;

pthread\_mutex\_t beeT, bearT;

void \*FillingThePot(void \*param) {

int beeNumber = \*((int \*) param);

while (time(nullptr) - startTime < 30)

{

pthread\_mutex\_lock(&beeT);

swallowCount++;

cout << "Пчела #" << beeNumber << " заполнила горшок " << swallowCount << " глотков." << endl;

if (swallowCount == N) {

cout << "Пчела #" << beeNumber << " будит медведя." << endl;

pthread\_mutex\_unlock(&bearT);

continue;

}

i++;

cout << "Другие пчелы теперь тоже имеют доступ к горшку!" << endl;

pthread\_mutex\_unlock(&beeT);

sleep(100);

}

pthread\_mutex\_unlock(&beeT);

pthread\_mutex\_unlock(&bearT);

}

void \*DrinkingThePot() {

while (time(nullptr) - startTime < 30){

pthread\_mutex\_lock(&bearT);

cout << "Медведь выпил " << swallowCount << " глотков меда." << endl;

swallowCount = 0;

sleep(100);

i++;

cout << "Другие пчелы теперь тоже имеют доступ к горшку!" << endl;

pthread\_mutex\_unlock(&beeT);

}

pthread\_mutex\_unlock(&beeT);

return nullptr;

}

int main() {

startTime = time(nullptr);

cout << "Введите количество пчел: ";

cin >> n;

while (n <= 0 || n >= 50){

cout << "Некорректный ввод! Попробуйте еще раз: ";

cin >> n;

}

cout << "Введите количество глотоков в горшке: ";

cin >> N;

while (N <= 0 || N >= 500){

cout << "Некорректный ввод! Попробуйте еще раз.\n";

cin >> N;

}

pthread\_mutex\_init(&beeT, nullptr);

pthread\_mutex\_init(&bearT, nullptr);

bees = new pthread\_t[n];

swallowCount = 0;

int \*beess = new int[n];

for (size\_t i = 0; i < n; i++) {

beess[i] = i + 1;

pthread\_create(&bees[i], nullptr, FillingThePot, (void \*) (beess + i));

}

pthread\_mutex\_lock(&bearT);

DrinkingThePot();

delete[] beess;

delete[] bees;

return 0;

}

**ОПИСАНИЕ**

Вначале медведь спит, а пчелки по очереди заполняют горшок. То есть невозможно в один момент времени двум пчелкам заполнить горшок, так как действия выполняются в критической секции, образованной mutex.lock() и unlock(). Когда мы определяем, что горшок полный, то будим медведя и он опустошает горшок. Затем повторяется то же самое, потому что медведь в конце своего метода говорит пчелкам, что можно продолжать работать. Для того, чтобы пчелки действительно работали по очереди, нужно делать sleep после того, как она принесла глоток меда, иначе одна пчелка будет крутиться в методе.

**ТЕСТЫ**

1. Изображение выглядит как текст

   Автоматически созданное описание
2. Изображение выглядит как текст

   Автоматически созданное описание