**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

Дисциплина: «Архитектура вычислительных систем»

Первая задача о Винни-Пухе, или неправильные пчелы.

Пояснительная записка

**Выполнил:**  
Темирханов Михаил,   
*студент гр. БПИ199.*

*17.11.2020г.*

**Москва**2020

Содержание

[1. Текст задания 2](#_Toc55160397)

[2. Описание решения 3](#_Toc55160398)

[3. Тестирование программы 4](#_Toc55160399)

[ПРИЛОЖЕНИЕ 1 5](#_Toc55160400)

[Список литературы 5](#_Toc55160401)

[ПРИЛОЖЕНИЕ 2 6](#_Toc55160402)

[Код программы 6](#_Toc55160403)

1. Текст задания

Неправильные пчелы, подсчитав в конце месяца убытки от наличия в лесу Винни-Пуха, решили разыскать его и наказать в назидание всем другим любителям сладкого. Для поисков медведя они поделили лес на участки, каждый из которых прочесывает одна стая неправильных пчел. В случае нахождения медведя на своем участке стая проводит показательное наказание и возвращается в улей. Если участок прочесан, а Винни-Пух на нем не обнаружен, стая также возвращается в улей. Требуется создать многопоточное приложение, моделирующее действия пчел. При решении использовать парадигму портфеля задач.

1. Описание решения

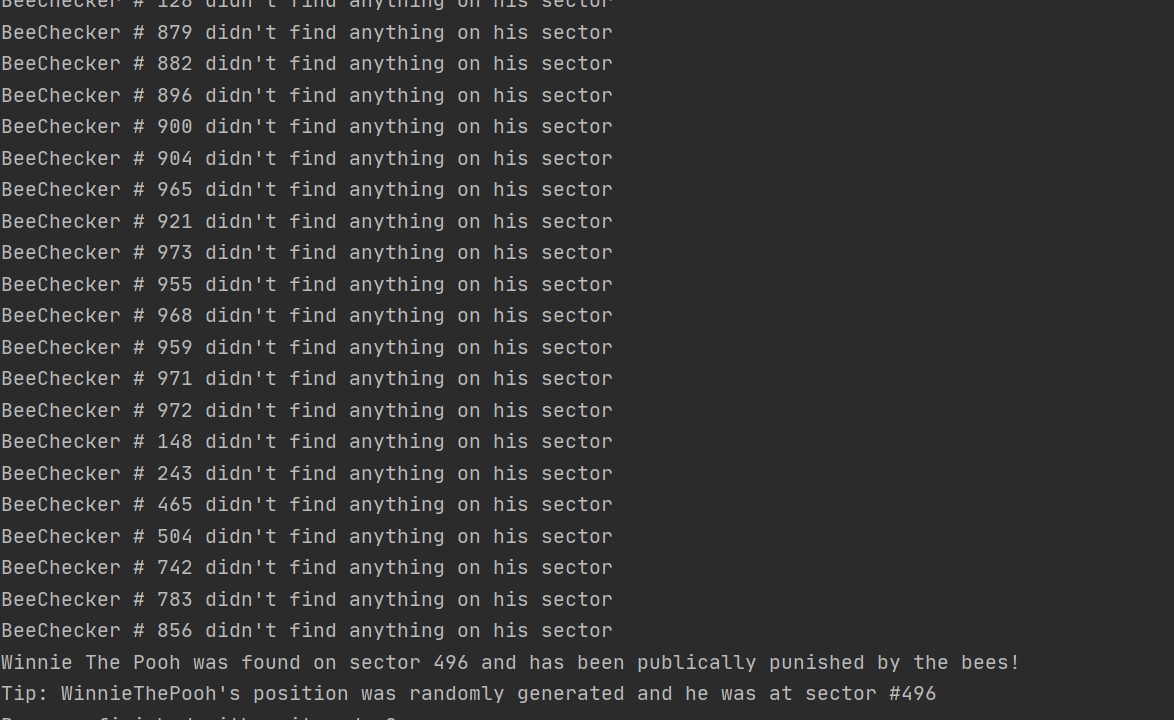
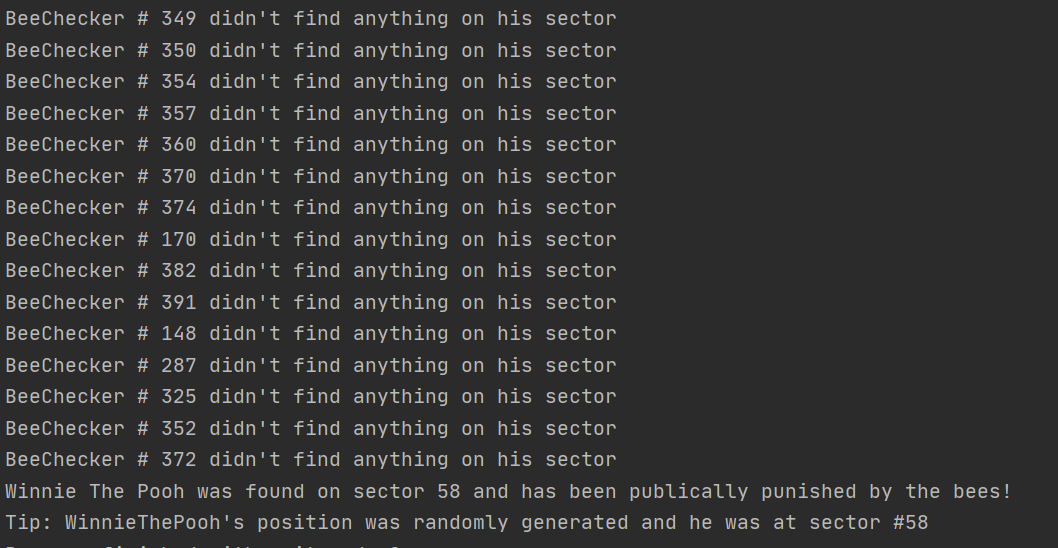
Положение Винни-Пуха и количество секторов, на которые пчёлы делят лес задаются случайным образом в разумных рамках (не может быть <1 и т.д.).

Объявляем vector<bool> — это наш лес, разделенный на сектора, и его длина, соответственно, равна количеству секторов, которое было определено выше.

Проходимся по массиву, всем секторам, в которых нет Винни-Пуха, придаём значение false, тому же сектору, где и находится Винни-Пух, придаем значение true. С этого момента и начнётся работа «пчёл» по нахождению Винни-Пуха. Создаём метод, который отправляет один рой пчел проверять соответствующую им клетку. Данный метод принимает на вход вектор секторов и номер клетки, которую нужно проверить, и выводит в консоль сообщения о том, найден ли в данной клетке Винни-Пух или нет. Описав метод, используем инструменты, которые мы получаем с помощью OpenMP, для того, чтобы реализовать несколько потоков, которые будут параллельно друг другу выполнять описанный выше метод, дабы ускорить процесс поиска Винни-Пуха. В конце концов, когда мы находим Винни-Пуха, выводим в консоль соответствующую информацию, а также выводим номер сектора, в котором он был найден.

1. Тестирование программы

Программа не принимает на вход никаких входных данных, все данные генерируются случайным образом. Программа работает корректно, что показывают следующие снимки экрана:

**

ПРИЛОЖЕНИЕ 1

Список литературы

1. [http://softcraft.ru/] (30.10.2020)
2. [https://fasmworld.ru/spravochnik-komand/] (31.10.2020)
3. [https://flatassembler.net/docs.php] (01.11.2020)
4. [https://en.wikipedia.org/wiki/Assembler] (01.11.2020)
5. [https://ru.wikipedia.org/wiki/Прямоугольная\_система\_координат] (01.11.2020)

ПРИЛОЖЕНИЕ 2

Код программы

#include <iostream>

#include <vector>

#include <ctime>

#include <string>

#include <thread>

#include <stdlib.h>

int winniePlace;

//Начальный вывод программы, показывающий значение случайно сгенерированных данных

void startProgramOutput(int length, int winnieThePooh) {

std::cout << "Bees divided forest into " + std::to\_string(length) + " sectors\n";

std::cout << "Winnie the Pooh is on " + std::to\_string(winnieThePooh) + " sector\n";

}

//Метод проверки одной стаей пчёл соответствующего сектора

void checkForestSector(std::vector<bool> forest, int beeChecker) {

if (forest[beeChecker]) {

std::cout << "BeeChecker #" + std::to\_string(beeChecker) +

" found Winnie The Pooh on his sector\n";

winniePlace = beeChecker;

} else {

std::cout << "BeeChecker # " + std::to\_string(beeChecker) + " didn't find anything on his sector\n";

};

};

//Установка положения Винни-Пуха

std::vector<bool> placeWinnieThePooh(int length, int winnieThePooh) {

std::vector<bool> forest(length);

for (int i = 0; i < length; ++i) {

if (i != winnieThePooh)

forest[i] = false;

else

forest[i] = true;

};

return forest;

}

int main() {

//Задаём случайные значения количеству секторов и положению Винни-Пуха

std::srand(std::time(NULL));

int length = std::rand() % 1000 + 1;

int winnieThePooh = std::rand() % length;

startProgramOutput(length, winnieThePooh);

std::vector<bool> forest = placeWinnieThePooh(length, winnieThePooh);

//Объявляем потоки и начинаем с ними работу

int numberOfThreads = length;

std::vector<std::thread> threads;

for (int i = 0; i < numberOfThreads; ++i) {

std::thread thr(checkForestSector, forest, i);

threads.emplace\_back(std::move(thr));

}

for (int i = 0; i < numberOfThreads; ++i) {

threads[i].join();

}

//Итоговый вывод программы

std::cout << "Winnie The Pooh was found on sector " + std::to\_string(winniePlace) +

" and has been publically punished by the bees!\n";

std::cout << "Tip: WinnieThePooh's position was randomly generated and he was at sector #" +

std::to\_string(winnieThePooh);

return 0;

};