Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №3 по курсу**

**«Операционные системы»**

**Тема работы**

Студент: Шапошник Никита Сергеевич

Группа: М8О-208Б-21

Вариант: 7

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2023

**Содержание**

1. Репозиторий
2. Постановка задачи
3. Общие сведения о программе
4. Общий метод и алгоритм решения
5. Исходный код
6. Демонстрация работы программы
7. Выводы

**Репозиторий**

[**https://github.com/Nikitashap/OS/tree/main/lab3**](https://github.com/Nikitashap/OS/tree/main/lab3)

**Постановка задачи**

Составить программу на языке Си, обрабатывающую данные в многопоточном режиме. При обработки использовать стандартные средства создания потоков операционной системы (Windows/Unix). Ограничение потоков должно быть задано ключом запуска вашей программы. Так же необходимо уметь продемонстрировать количество потоков, используемое вашей программой с помощью стандартных средств операционной системы.

В отчете привести исследование зависимости ускорения и эффективности алгоритма от входящих данных и количества потоков. Получившиеся результаты необходимо объяснить.

**Вариант 7**: Два человека играют в кости. Правила игры следующие: каждый игрок делает бросок 2-ух костей K раз; побеждает тот, кто выбросил суммарно большее количество очков. Задача программы экспериментально определить шансы на победу каждого из игроков. На вход программе подается K, какой сейчас тур, сколько очков суммарно у каждого из игроков и количество экспериментов, которые должна произвести программа

**Общие сведения о программе**

Программа представляет из себя один файл lab3.cpp.

**Общий метод и алгоритм решения**

Создаем массив потоков. Далее каждому потоку даем в задачу проведение одного эксперимента, реализованного в функции experiment\_iteration.

**Исходный код**

#include <iostream>

#include <string>

#include <thread>

#include <vector>

#include <random>

#include <ctime>

using namespace std;

void experement\_iteration (int sum1, int sum2, int k, int round, int &win1, int &win2)

{

int sum1\_exp=sum1;

int sum2\_exp=sum2;

random\_device rd;

mt19937 gen(rd());

uniform\_int\_distribution<> dist(1,6);

int j=0;

while(j<k-round){

sum1\_exp+=dist(gen);

sum1\_exp+=dist(gen);

sum2\_exp+=dist(gen);

sum2\_exp+=dist(gen);

j++;

}

if(sum1\_exp>sum2\_exp){

win1++;

}

if(sum2\_exp>sum1\_exp){

win2++;

}

}

int min(int a, int b){

if(a>=b){

return b;

} else{

return a;

}

}

int main(int argc, char \*argv[]){

int thread\_number;

if (argc < 2)

{

cout << "Enter thread amount:\n";

cin >> thread\_number;

}

else

{

thread\_number = stoi(argv[1]);

}

int k,round,sum1,sum2,exp\_num,win1=0,win2=0;

cin >> k >> round >> sum1 >> sum2 >> exp\_num;

vector<thread> th(thread\_number);

unsigned int clock\_start=clock();

for(int i=0;i<exp\_num;i+=thread\_number){

for(int j=0;j<min(exp\_num-i,thread\_number);j++){

th[j] = thread(experement\_iteration,ref(sum1),ref(sum2),ref(k),ref(round),ref(win1),ref(win2));

}

for(int j=0;j<min(exp\_num-i,thread\_number);j++){

th[j].join();

}

}

unsigned int clock\_end=clock();

cout << clock\_end-clock\_start << '\n';

cout << "1 player chance to win is " << float(win1) / float(exp\_num) << '\n';

cout << "2 player chance to win is " << float(win2) / float(exp\_num) << '\n';

}

**Демонстрация работы программы**

Ввод в консоль:

iMac-Apple:lab3 appleimac$ ./a.out

Enter thread amount:

1

Enter number of rounds: 10

Enter number of played rounds: 5

1st player sum: 10

2nd player sum: 12

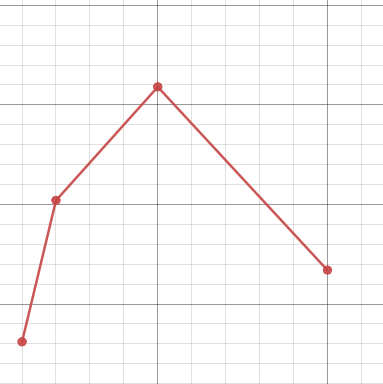
Number of experements: 10

831

1 player chance to win is 0.3

2 player chance to win is 0.6

График зависимости времени работы программы от числа потоков



Время работы программы становится меньше, если использовать более 5 потоков(самая высокая точка на графике).

**Выводы**

Проделав лабораторную работу, я приобрёл практические навыки в управлении потоками в ОС и обеспечил синхронизацию между ними.