

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ УКРАИНЫ

«КИЕВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ»

ФАКУЛЬТЕТ ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №1

ПО КУРСУ

«КОМПЬЮТЕРНЫЕ СИСТЕМЫ»

Выполнил:

студент IV курса

группы ІО-21

Журо Георгий Александрович

Киев – 2015

**Тема:** Моделирование временных характеристик вычислительных систем и сетей.

**Цель работы:** Разработать и реализовать в виде программы математическую модель расчета временных характеристик мультипроцессорной системы или сети.

**Задание:** По заданной граф-схеме алгоритма (по согласованию с преподавателем) построить матрицу вероятностей переходов. Определить основные параметры трудоемкости алгоритмов: а) среднее число операций при одном прогоне алгоритма; б) среднее число обращений к каждому из файлов; в) среднее число информации, передаваемое при одном обращении к файлу. Определить среднюю трудоемкость этапа счета.

**Исходные данные и матрица вероятностей переходов:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | V11 |
| V1 | -1 | 1 |  |  |  |  |  |  |  |  |  |
| V2 |  | -1 | 0,75 |  | 0,17 |  | 0,01 |  | 0,07 |  |  |
| V3 |  |  | -1 | 1 |  |  |  |  |  |  |  |
| V4 |  |  |  | -1 | 0,67 |  | 0,05 |  | 0,28 |  |  |
| V5 |  |  |  |  | -1 | 1 |  |  |  |  |  |
| V6 |  |  |  |  |  | -1 | 0,14 |  | 0,86 |  |  |
| V7 |  |  |  |  |  |  | -1 |  |  |  |  |
| V8 |  |  |  |  |  |  |  | -1 |  |  | 1 |
| V9 |  |  |  |  |  |  |  |  | -1 | 1 |  |
| V10 |  |  |  |  |  |  | 0,14 |  | 0,86 | -1 |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 |
| l | 100 |  |  | 150 |  | 175 |  | 125 |  | 250 |
| k |  | 50 | 75 |  | 65 |  | 90 |  | 100 |  |
| N | 1 |  |  | 1 |  | 3 |  | 3 |  | 2 |

**Схема алгоритма:**



**Код программы:**

**import** java.util.Random;

**public** **class** Lab1 {

**private** **static** Random *randomInteger* = **new** Random();

**private** **static** **int**[][] *regularOper* = {{50, 1},

{75, 2},

{65, 4},

{90, 6},

{100, 8}};

**private** **static** **int**[][] *fileOper* = {{100, 1, 0},

{150, 1, 3},

{175, 3, 5},

{125, 3, 7},

{250, 2, 9}};

**private** **static** **double**[] *n* = **new** **double** [10];

**private** **static** **double**[] *V* = **new** **double** [10];

**private** **static** **int** *cycleIterations* = 10000;

**private** **static** **void** algo(){

**int** i, j;

//V1

*V*[0]++;

//V2

*V*[1]++;

//Условие i <= 0

i = *randomInteger*(-5, 2);

**if**(i <= 0) {

//V3

*V*[2]++;

//V4

*V*[3]++;

}

**else** {

//Условие j > 0

j = *randomInteger*(-2, 6);

**if**(j > 0){

//V5

*V*[4]++;

//V6

*V*[5]++;

}

**else** {

//Цикл N = 7

**for**(**int** N = 1; N < 7; N++){

//V9

*V*[8]++;

//V10

*V*[9]++;

}

//V7

*V*[6]++;

//V8

*V*[7]++;

}

}

}

**public** **static** **void** main(String... args){

//Прогон алгоритма

**for**(**int** i = 0; i < *cycleIterations*; i++){

*algo*();

}

//Подсчёт n

System.***out***.println("Подсчёт n:");

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

*n*[i] = *V*[i] / *cycleIterations*;

System.***out***.println("n" + (i+1) + " = " + *n*[i]);

}

System.***out***.println("\n---------------------------------------------------------------\n");

**double** N = 0; // Cреднее число операций при одном прогоне алгоритма

**double** operNum = 0;

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

operNum += (*n*[*regularOper*[i][1]] \* *regularOper*[i][0]);

N += *n*[*regularOper*[i][1]];

}

System.***out***.println("Cреднее число операций при одном прогоне алгоритма: " + operNum);

**double** appealNum; // Среднее число обращений к каждому из файлов

**double** infoNum; // Среднее число информации, передаваемое при одном обращении к файлу

**double** fileNum; // Номер файла

**for**(**int** i = 0; i < 3; i++){

appealNum = 0;

infoNum = 0;

fileNum = i + 1;

//Cреднее число обращений к каждому из файлов

**for**(**int** j = 0; j < *fileOper*.length; j++){

**if**(fileNum == *fileOper*[j][1])

appealNum += *n*[*fileOper*[j][2]];

}

System.***out***.println("Cреднее число обращений к файлу " + (i + 1) + ": " + appealNum);

//Cреднее число информации, передаваемое при одном обращении к файлу

**for**(**int** j = 0; j < *fileOper*.length; j++){

**if**(fileNum == *fileOper*[j][1])

infoNum += (*n*[*fileOper*[j][2]] \* *fileOper*[j][0]);

}

System.***out***.println("Cреднее число информации, передаваемое при одном обращении к файлу " + (i + 1) + ": " + (infoNum / appealNum));

}

System.***out***.println("Средняя трудоемкость этапа счёта: " + (operNum / N));

}

**private** **static** **int** randomInteger(**int** min, **int** max){

**return** *randomInteger*.nextInt((max - min) + 1) + min;

}

}

**Результаты работы программы:**

Подсчёт n:

n1 = 1.0

n2 = 1.0

n3 = 0.751

n4 = 0.751

n5 = 0.1645

n6 = 0.1645

n7 = 0.0845

n8 = 0.0845

n9 = 0.507

n10 = 0.507

---------------------------------------------------------------

Cреднее число операций при одном прогоне алгоритма: 175.3225

Cреднее число обращений к файлу 1: 1.751

Cреднее число информации, передаваемое при одном обращении к файлу 1: 121.44488863506568

Cреднее число обращений к файлу 2: 0.507

Cреднее число информации, передаваемое при одном обращении к файлу 2: 250.0

Cреднее число обращений к файлу 3: 0.249

Cреднее число информации, передаваемое при одном обращении к файлу 3: 158.03212851405624

Средняя трудоемкость этапа счёта: 69.93318707618667