Kpi-best

Министерство образования и науки Украины

Национальный технический университет Украины “Киевский политехнический институт”

Факультет информатики и вычислительной техники

Кафедра вычислительной техники

Лабораторная работа № 1

с курса “Компьютерные системы”

Выполнил: студент ІV курса группы ІВ – 21

Таран В. И.

Киев – 2015

**ЛАБОРАТОРНАЯ РАБОТА № 1**

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 | V12 |
| V1 | -1 | 1 |  |  |  |  |  |  |  |  |  |  |
| V2 |  | -1 | 0,25 | 0,61 |  | 0,14 |  |  |  |  |  |  |
| V3 |  |  | -1 |  | 1 |  |  |  |  |  |  |  |
| V4 |  |  |  | -1 | 1 |  |  |  |  |  |  |  |
| V5 |  |  | 0,25 | 0,61 | -1 | 0,14 |  |  |  |  |  |  |
| V6 |  |  |  |  |  | -1 | 0,375 |  | 0,625 |  |  |  |
| V7 |  |  |  |  |  |  | -1 | 1 |  |  |  |  |
| V8 |  |  |  |  |  |  |  | -1 |  | 0,5 | 0,5 |  |
| V9 |  |  |  |  |  |  |  |  | -1 | 0,5 | 0,5 |  |
| V10 |  |  |  |  |  |  |  |  |  | -1 | 1 |  |
| V11 |  |  |  |  |  |  |  |  |  |  | -1 | 1 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | V11 |
| l | 100 |  |  |  | 200 | 150 |  | 125 |  | 50 | 250 |
| k |  | 50 | 75 | 90 |  |  | 65 |  | 100 |  |  |
| Nf | 1 |  |  |  | 1 | 2 |  | 2 |  | 3 | 3 |

Граф – схема алгоритма



Код программы

**public** **class** Program {

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

**private** **static** **double**[] *stat*=**new** **double** [11];

**private** **static** **double**[] *n*=**new** **double** [11];

**private** **static** **int** *cycleAmount*=10000;

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

{75, 2},

{90, 3},

{65, 6},

{100, 8}};

//

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

{200, 1, 4},

{150, 2, 5},

{125, 2, 7},

{50, 3, 9},

{250, 3, 10}};

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

//Cycle task "cycleAmount" time

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

*task*();

}

//Calculate "n"

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

*n*[i]=*stat*[i]/*cycleAmount*;

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

}

//Calculate "average processor operations"

**double** apo=0;

**double** N=0;

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

apo+=(*n*[*generalOperation*[i][1]]\**generalOperation*[i][0]);

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

}

System.***out***.println("Cреднее число процессорных операций: "+apo);

//

**double** fileN;

**double** fileInfoN;

**double** fileNumber;

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

fileN=0;

fileInfoN=0;

fileNumber=i+1;

//Calculate "average file-N operation"

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

**if**(fileNumber==*fileOperation*[j][1])

fileN+=*n*[*fileOperation*[j][2]];

}

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

//Calculate "average file-N information"

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

**if**(fileNumber==*fileOperation*[j][1])

fileInfoN+=(*n*[*fileOperation*[j][2]]\**fileOperation*[j][0]);

}

System.***out***.println("Cреднее количество информации, передаваемой при одном обращении к файлу "+(i+1)+" : "+(fileInfoN/fileN));

}

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

}

**private** **static** **void** task(){

**int** x, j, k;

//Start

//V\_beta\_1\_1 (1)

*stat*[0]++;

//V\_alpha\_1 (2)

*stat*[1]++;

//Cycle\_N=7

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

//If\_X<=0

x=*random*(-4, 2);

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

//V\_alpha\_3 (4)

*stat*[3]++;

}

**else**{

//V\_alpha\_2 (3)

*stat*[2]++;

}

//V\_beta\_2\_1 (5)

*stat*[4]++;

}

//V\_beta\_3\_2 (6)

*stat*[5]++;

//If\_j>0

j=*random*(-2, 5);

**if**(j>0){

//V\_alpha\_5 (9)

*stat*[8]++;

}

**else**{

//V\_alpha\_4 (7)

*stat*[6]++;

//V\_beta\_3\_2 (8)

*stat*[7]++;

}

//If\_k<1

k=*random*(-2, 3);

**if**(k>=1){

//V\_beta\_4\_3 (10)

*stat*[9]++;

}

//V\_beta\_5\_3 (11)

*stat*[10]++;

//End

}

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

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

}

}

Результат работы программы

n[1]=1.0

n[2]=1.0

n[3]=1.7105

n[4]=4.2895

n[5]=6.0

n[6]=1.0

n[7]=0.3738

n[8]=0.3738

n[9]=0.6262

n[10]=0.5088

n[11]=1.0

Cреднее число процессорных операций: 651.2595

Cреднее число обращений к файлу 1 : 7.0

Cреднее количество информации, передаваемой при одном обращении к файлу 1 : 185.71428571428572

Cреднее число обращений к файлу 2 : 1.3738000000000001

Cреднее количество информации, передаваемой при одном обращении к файлу 2 : 143.1976998107439

Cреднее число обращений к файлу 3 : 1.5088

Cреднее количество информации, передаваемой при одном обращении к файлу 3 : 182.5556733828208

Средняя трудоемкость этапа счета: 81.4074375