

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №4**  
**по дисциплине «Качество и метрология программного обеспечения»**  
**Тема: Построение операционной графовой модели программы (ОГМП)**  
**и расчет характеристик эффективности ее выполнения методом**  
**эквивалентных преобразований**

Студент гр. 7304

\_\_\_\_\_

Пэтайчук Н.Г.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2021

### **Цель работы.**

Изучение возможности построения операционной графовой модели программы (ОГМП) и расчета характеристик эффективности ее выполнения методом эквивалентных преобразований.

### **Постановка задачи.**

#### **1. Построение ОГМП.**

Для рассмотренного в лабораторных работах 1-3 индивидуального задания разработать операционную модель управляющего графа программы на основе схемы алгоритма. При выполнении работы рекомендуется для упрощения обработки графа исключить диалог при выполнении операций ввода-вывода данных, а также привести программу к структурированному виду.

Выбрать вариант графа с нагруженными дугами, каждая из которых должна представлять фрагмент программы, соответствующий линейному участку или ветвлению. При расчете вероятностей ветвлений, зависящих от распределения данных, принять равномерное распределение обрабатываемых данных в ограниченном диапазоне (например,  $[0, 100.00]$  - для положительных чисел или  $[-100.00, 100.00]$  - для произвольных чисел). В случае ветвлений, вызванных проверкой выхода из цикла, вероятности рассчитываются исходя из априорных сведений о числе повторений цикла. Сложные случаи оценки вероятностей ветвлений согласовать с преподавателем.

В качестве параметров, характеризующих потребление ресурсов, использовать времена выполнения команд соответствующих участков программы, полученные с помощью монитора Sampler в процессе выполнения работы №3. Если требуется, оценить с помощью монитора Sampler времена выполнения неучтенных ранее участков программы.

#### **2. Расчет характеристик эффективности выполнения программы методом эквивалентных преобразований.**

Полученную в части 1 данной работы ОГМП, представить в виде графа с нагруженными дугами, у которого в качестве параметров, характеризующих потребление ресурсов на дуге  $ij$ , использовать тройку  $\{P_{ij}, M_{ij}, D_{ij}\}$ , где:

$P_{ij}$  - вероятность выполнения процесса для дуги  $ij$ ,

$M_{ij}$  - мат. ожидание потребления ресурса процессом для дуги  $ij$ ,

$D_{ij}$  - дисперсия потребления ресурса процессом для дуги  $ij$ .

В качестве потребляемого ресурса в данной работе рассматривается время процессора, а оценками мат. ожиданий времен для дуг исходного графа следует принять времена выполнения операторов (команд), соответствующих этим дугам участков программы. Дисперсиям исходных дуг следует присвоить нулевые значения.

Выполнить описание построенной ОГМП на входном языке пакета CSA III в виде поглощающей марковской цепи (ПМЦ) – (англ.) AMC (absorbing Markov chain) или эргодической марковской цепи (ЭМЦ) - EMC (ergodic Markov chain).

#### **Ход выполнения.**

1. Для выполнения данной лабораторной работы и построения операционной графовой модели программы использовалась программа из третьей лабораторной работы, представленное в Приложении А. Граф управления для основной части программы, а именно функции линеаризации `linfit1`, представлен на Рисунке 1:

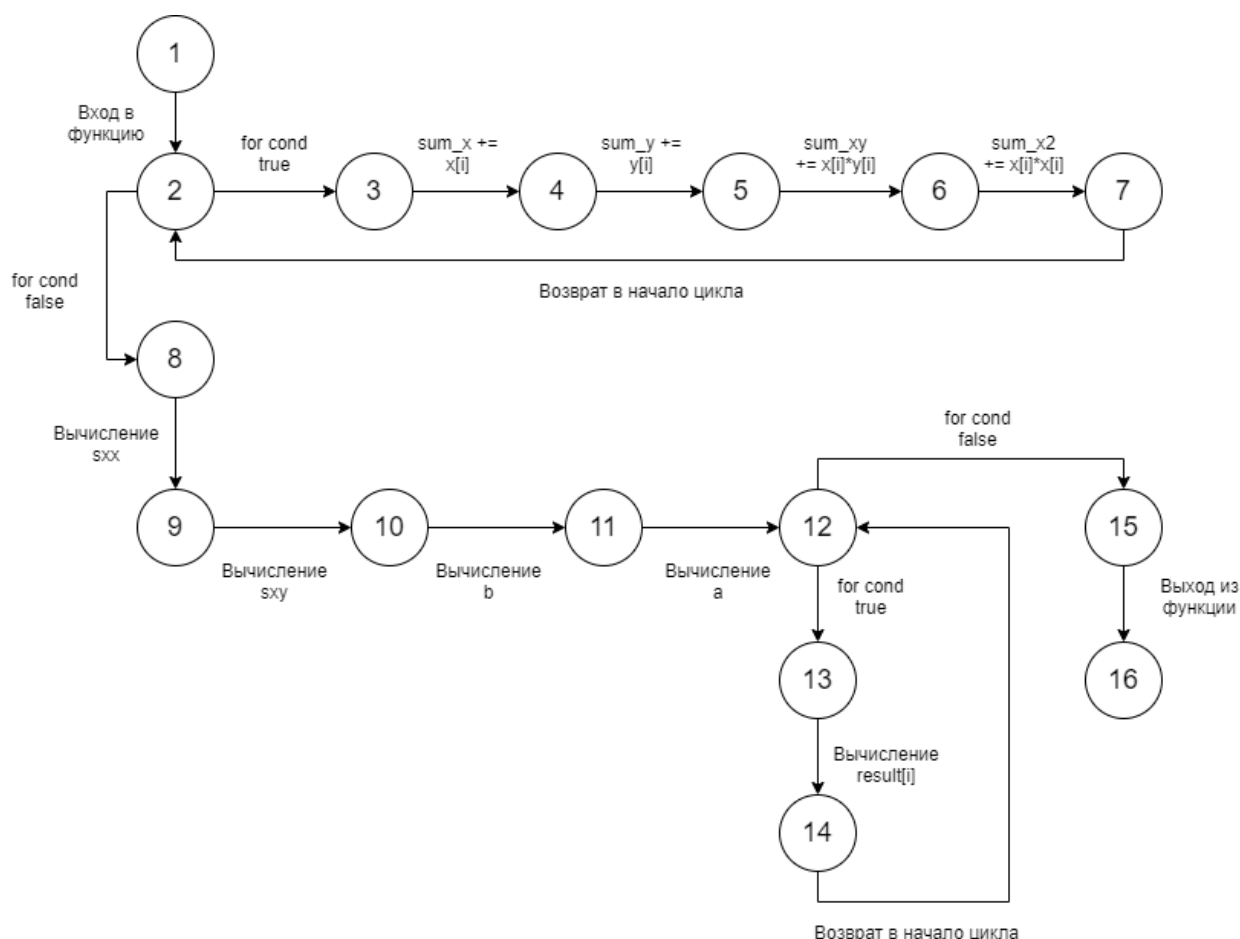


Рисунок 1: Управляющий граф функции linfit1

- На основе текста программы из третьей лабораторной работы был составлен текст программы для профилирования с использованием профилировщика SAMPLER. Текст программы для профилирования представлен в Приложении Б, результаты профилирования функции linfit1 представлены в Таблице 1:

Исход. Поз.	Прием. Поз.	Общее время (мкс)	Кол-во проходов	Среднее время (мкс)
1 : 13	1 : 16	2.51	1	2.51
1 : 16	1 : 18	427.43	80	5.34
1 : 18	1 : 20	472.69	80	5.91
1 : 20	1 : 22	626.90	80	7.84
1 : 22	1 : 24	575.77	80	7.20

1 : 24	1 : 16	168.46	79	2.13
1 : 24	1 : 26	1.68	1	1.68
1 : 26	1 : 28	122.36	1	122.36
1 : 28	1 : 30	122.36	1	122.36
1 : 30	1 : 32	70.40	1	70.40
1 : 32	1 : 34	174.32	1	174.32
1 : 34	1 : 37	1.68	1	1.68
1 : 37	1 : 39	669.64	80	8.37
1 : 39	1 : 37	144.15	79	1.82
1 : 39	1 : 41	0.84	1	0.84
1 : 41	1 : 59	2.51	1	2.51
1 : 57	1 : 13	162.59	1	162.59

Таблица 1: Результаты профилирования функции linfit1

Суммарное время работы  $T = 3746.29$  мкс.

3. На основании полученных с помощью профилировщика SAMPLER данных о работе функции linfit1 был проведён расчёт вероятностей и затрат ресурсов для дуг управляющего графа функции linfit1. Результаты расчётов приведены в Таблице 2:

Дуга	Номера строк	Количество проходов	Расчет вероятности	Затраты ресурсов (Среднее время), мкс
L1 – L2	57 : 13	1	1	162.59
L2 – L3	13 : 16	1	$80 / (80 + 1) = 0.99$	2.51
L3 – L4	16 : 18	80	1	5.34
L4 – L5	18 : 20	80	1	5.91
L5 – L6	20 : 22	80	1	7.84

L6 – L7	22 : 24	80	1	7.20
L7 – L2	24 : 16	79	1	2.13
L2 – L8	24 : 26	1	$1 - 0.99 = 0.01$	1.68
L8 – L9	26 : 28	1	1	122.36
L9 – L10	28 : 30	1	1	122.36
L10 – L11	30 : 32	1	1	70.40
L11 – L12	32 : 34	1	1	174.32
L12 – L13	34 : 37	1	$80 / (80 + 1) = 0.99$	1.68
L13 – L14	37 : 39	80	1	8.37
L14 – L12	39 : 37	79	1	1.82
L12 – L15	39 : 41	1	$1 - 0.99 = 0.01$	0.84
L15 – L16	41 : 59	1	1	2.51

Таблица 2: Расчёт вероятностей и затрат ресурсов

4. На основании полученных расчётов вероятностей и затрат ресурсов был построен операционная графовая модель программы, представленная на Рисунке 2:

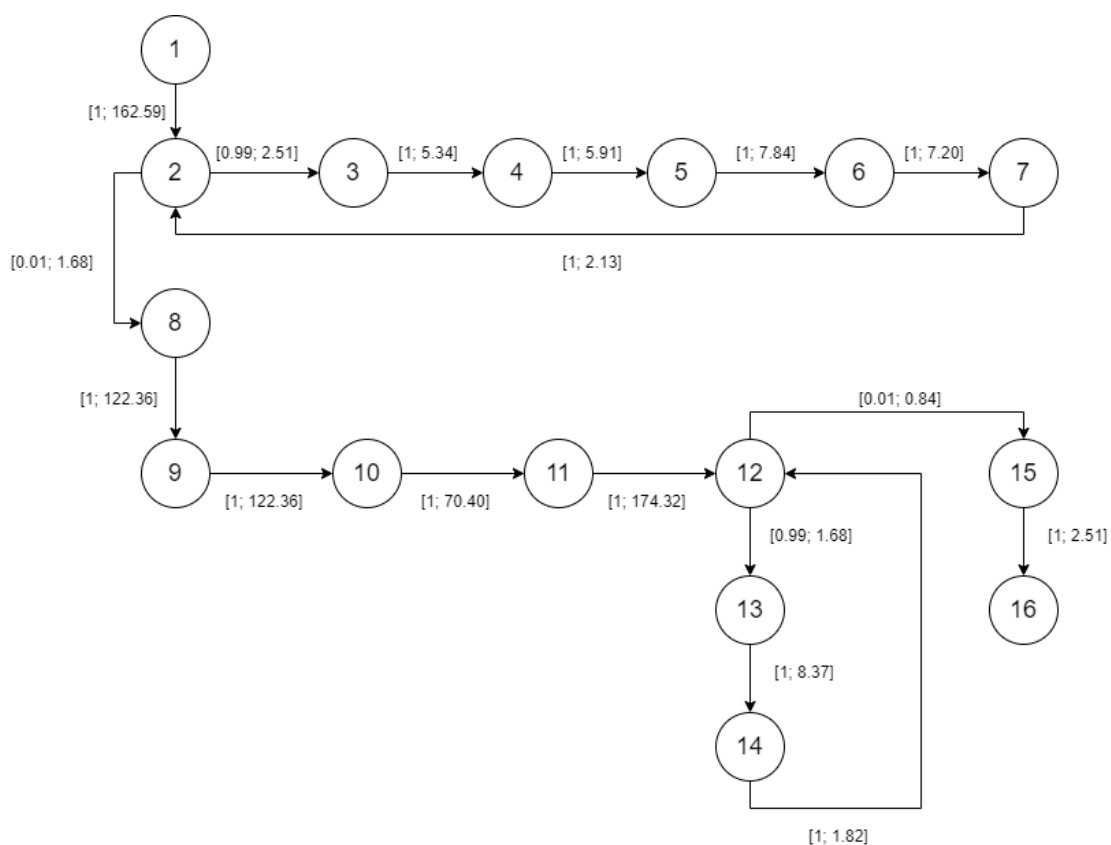


Рисунок 2: Операционная графовая модель программы

- По полученной ОГМП был создан XML-файл модели программы, приведённый в Приложении В, для расчёта характеристик эффективности выполнения программы методом эквивалентных преобразований с помощью пакета CSA III. Графическое отображение модели представлено на Рисунке 3:

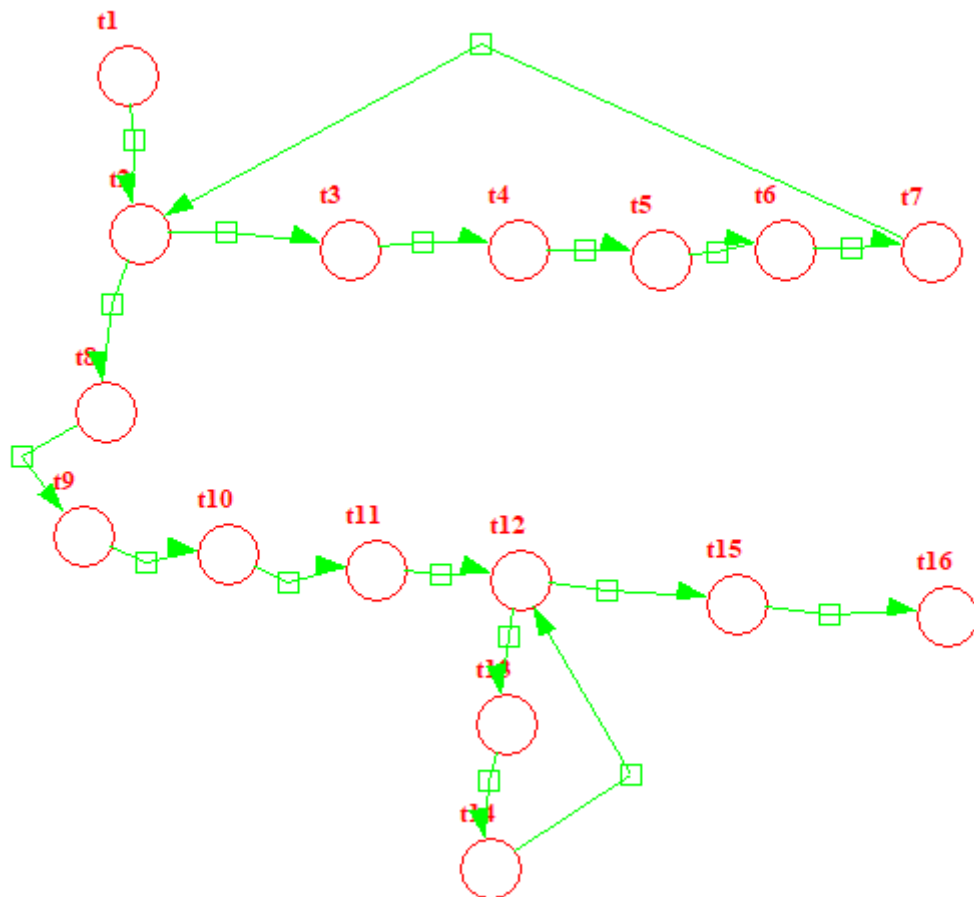


Рисунок 3: Модель программы в CSA III

1. С помощью пакета CSA III были вычислены математическое ожидание и дисперсия времени выполнения основной части программы, а именно функции linfit1. Результаты вычислений представлены на Рисунке 4:

t1-->t16: Objects::AMC::Link	
Name	Value
name	t1-->t16
probability	0.9999999999999998
intensity	4894.26
deviation	10865861.82

Рисунок 4: Результаты вычислений



## **Выводы.**

В ходе выполнения лабораторной работы была построена операционная графовая модель заданной программы, нагрузочные параметры которой были оценены с помощью профилировщика Sampler и методом эквивалентных преобразований с помощью пакета CSAIII были вычислены математическое ожидание и дисперсия времени выполнения основной части программы, а именно функции linfit1. Математическое ожидание равно 4894.26, дисперсия – 10865861.82.

Математическое ожидание сильно отличается от полученного суммарного времени работы, равного 3746.29, что может быть объяснено низкой точностью вычисления вероятностей (шло округление до 2 знаков после запятой).

## Приложение А: Исходный код программы.

```
#include <stdlib.h>
#include <time.h>
#include "Sampler.h"

void linfit1(double x[80], double y[80], double *result, double
*a, double *b, int n)
{
    double sum_x = 0.0;
    double sum_y = 0.0;
    double sum_xy = 0.0;
    double sum_x2 = 0.0;
    double sxy, sxx;
    int i;
    for (i = 0; i < n; i++)
    {
        sum_x += x[i];
        sum_y += y[i];
        sum_xy += x[i]*y[i];
        sum_x2 += x[i]*x[i];
    }
    sxx = sum_x2 - sum_x*sum_x/n;
    sxy = sum_xy - sum_x*sum_y/n;
    *b = sxy/sxx;
    *a = ((sum_x2*sum_y - sum_x*sum_xy)/n)/sxx;
    for (i = 0; i < n; i++)
    {
        *(result + i) = (*a) + (*b)*x[i];
    }
}

int main()
{
    double x_array[80];
    double y_array[80];
    double result_array[80];
    double a, b;
    int n = 80, i;
    srand(time(NULL));
    for (i = 0; i < n; i++)
    {
        x_array[i] = i + 1;
        y_array[i] = rand() % 100;
    }
    linfit1(x_array, y_array, result_array, &a, &b, n);
}
```

## Приложение Б: Код программы для профилирования.

```
#include <stdlib.h>
#include <time.h>
#include "Sampler.h"

void linfit1(double x[80], double y[80], double *result, double
*a, double *b, int n)
{
    double sum_x = 0.0;
    double sum_y = 0.0;
    double sum_xy = 0.0;
    double sum_x2 = 0.0;
    double sxy, sxx;
    int i;
    SAMPLE;
    for (i = 0; i < n; i++)
    {
        SAMPLE;
        sum_x += x[i];
        SAMPLE;
        sum_y += y[i];
        SAMPLE;
        sum_xy += x[i]*y[i];
        SAMPLE;
        sum_x2 += x[i]*x[i];
        SAMPLE;
    }
    SAMPLE;
    sxx = sum_x2 - sum_x*sum_x/n;
    SAMPLE;
    sxy = sum_xy - sum_x*sum_y/n;
    SAMPLE;
    *b = sxy/sxx;
    SAMPLE;
    *a = ((sum_x2*sum_y - sum_x*sum_xy)/n)/sxx;
    SAMPLE;
    for (i = 0; i < n; i++)
    {
        SAMPLE;
        *(result + i) = (*a) + (*b)*x[i];
        SAMPLE;
    }
    SAMPLE;
}

int main()
{
    double x_array[80];
    double y_array[80];
    double result_array[80];
```

```

double a, b;
int n = 80, i;
srand(time(NULL));
for (i = 0; i < n; i++)
{
    x_array[i] = i + 1;
    y_array[i] = rand() % 100;
}
SAMPLE;
linfit1(x_array, y_array, result_array, &a, &b, n);
SAMPLE;
}

```

## Приложение В: XML-файл модели программы.

```
<model type="Objects::AMC::Model" name="lab1">
  <node type="Objects::AMC::Top" name="t1"></node>
  <node type="Objects::AMC::Top" name="t2"></node>
  <node type="Objects::AMC::Top" name="t3"></node>
  <node type="Objects::AMC::Top" name="t4"></node>
  <node type="Objects::AMC::Top" name="t5"></node>
  <node type="Objects::AMC::Top" name="t6"></node>
  <node type="Objects::AMC::Top" name="t7"></node>
  <node type="Objects::AMC::Top" name="t8"></node>
  <node type="Objects::AMC::Top" name="t9"></node>
  <node type="Objects::AMC::Top" name="t10"></node>
  <node type="Objects::AMC::Top" name="t11"></node>
  <node type="Objects::AMC::Top" name="t12"></node>
  <node type="Objects::AMC::Top" name="t13"></node>
  <node type="Objects::AMC::Top" name="t14"></node>
  <node type="Objects::AMC::Top" name="t15"></node>
  <node type="Objects::AMC::Top" name="t16"></node>
  <link type="Objects::AMC::Link" name="t1-
2" probability="1" intensity="162.59" deviation="0.0" source="t1"
dest="t2"></link>
  <link type="Objects::AMC::Link" name="t2-
3" probability="0.99" intensity="2.51" deviation="0.0" source="t2"
dest="t3"></link>
  <link type="Objects::AMC::Link" name="t3-
4" probability="1" intensity="5.34" deviation="0.0" source="t3" de
st="t4"></link>
  <link type="Objects::AMC::Link" name="t4-
5" probability="1" intensity="5.91" deviation="0.0" source="t4" de
st="t5"></link>
  <link type="Objects::AMC::Link" name="t5-
6" probability="1" intensity="7.84" deviation="0.0" source="t5" de
st="t6"></link>
  <link type="Objects::AMC::Link" name="t6-
7" probability="1" intensity="7.20" deviation="0.0" source="t6" de
st="t7"></link>
  <link type="Objects::AMC::Link" name="t7-
2" probability="1" intensity="2.13" deviation="0.0" source="t7" de
st="t2"></link>
  <link type="Objects::AMC::Link" name="t2-
8" probability="0.01" intensity="1.68" deviation="0.0" source="t2"
dest="t8"></link>
  <link type="Objects::AMC::Link" name="t8-
9" probability="1" intensity="122.36" deviation="0.0" source="t8"
dest="t9"></link>
  <link type="Objects::AMC::Link" name="t9-
10" probability="1" intensity="122.36" deviation="0.0" source="t9"
dest="t10"></link>
  <link type="Objects::AMC::Link" name="t10-
11" probability="1" intensity="70.40" deviation="0.0" source="t10"
dest="t11"></link>
```

```

    <link type="Objects::AMC::Link" name="t11-
12" probability="1" intensity="174.32" deviation="0.0" source="t11
" dest="t12"></link>
    <link type="Objects::AMC::Link" name="t12-
13" probability="0.99" intensity="1.68" deviation="0.0" source="t1
2" dest="t13"></link>
    <link type="Objects::AMC::Link" name="t13-
14" probability="1" intensity="8.37" deviation="0.0" source="t13"
dest="t14"></link>
    <link type="Objects::AMC::Link" name="t14-
12" probability="1" intensity="1.82" deviation="0.0" source="t14"
dest="t12"></link>
    <link type="Objects::AMC::Link" name="t12-
15" probability="0.01" intensity="0.84" deviation="0.0" source="t1
2" dest="t15"></link>
    <link type="Objects::AMC::Link" name="t15-
16" probability="1" intensity="2.51" deviation="0.0" source="t15"
dest="t16"></link>
</model>

```