Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н. Э. Баумана
(национальный исследовательский университет)»
Факультет «Робототехника и комплексная автоматизация (РК)»

Кафедра «Системы автоматизированного проектирование (РК6)»

Отчёт по проектно-технологической практике

Выполнил:

Студент: Василян Артур

Группа: РК6-33Б

Проверил:
Берчун Ю. В.

Дата _____

Подпись _____

Москва, 2020 г.

1.Задание

Требуется разработать программу, реализующую дискретно-событийное моделирование системы, рассмотренной в задании 2 домашнего задания №4. Обратите внимание, что все интервалы времени подчиняются законам распределений, носящим непрерывный характер. Поэтому категорически неверными является выбор целочисленных типов данных для моментов и интервалов времени, и тем более инкремент модельного времени с единичным шагом. Нужно реализовать именно переход от события к событию, как это сделано в GPSS и других проблемно-ориентированных системах. Для упрощения можно ограничиться использованием единственного потока случайных чисел для генерации всех необходимых случайных величин. Результатом работы программы должен быть лог-файл, содержащий записи типа: «В момент времени 12.345 транзакт с идентификатором 1 вошёл в модель», «В момент времени 123.456 транзакт с идентификатором 123 встал в очередь 1», «В момент времени 234.567 транзакт с идентификатором 234 занял устройство 2», «В момент времени 345.678 транзакт с идентификатором 345 освободил устройство 1», «В момент времени 456.789 транзакт с идентификатором 456 вышел из модели».

2. Описание входных данных.

На вход поступают значения R1, G1 и N1=R1+G1+B1.

3. Описание работы алгоритма

Определён класс Cash. В поле private находятся переменные x,y для обозначение интервала из которого будут браться случайные числа для вычисления времени обслуживания. В поле public находятся конструктор с параметрами, метод service, метод vhod, метод vihod. Так же в поле public находятся 4 переменных типа вектор : line (номера людей, ожидающих обслуживания и окончания своего обслуживания), Smoment (моменты времени, в которые закончится обслуживание транзактов), gate(номера транзактов входящих в модель), Emoment (моменты времени входов транзактов в модель).

Метод service рассчитывает время, в которое заканчивается обслуживание транзакта, складывая время обслуживания с временем начала обработки.

Метод vhod удаляет из очереди на вход транзакт, обслуживание которых уже началось, сравнивая время начала их обслуживания с текущим временем. В начале и конце функции прописаны mtx.lock() и mtx.unlock чтобы одновременно только один поток мог работать с ней.

Метод vihod удаляет транзакты, обслуживание которых закончилось, из очереди на обслуживание, сравнивая время окончания их обслуживания с текущим временем (entry_time). В начале и конце функции прописаны mtx.lock() и mtx.unlock чтобы одновременно только один поток мог работать с ней.

Функция random - реализация функции UNIFORM из GPSS.

В начале функции main создан объект класса ofstream и связан с файлом result.txt, который будет использоваться для вывода. Создаются объекты класса Cash q1, q2 первая и вторая касса соответственно. Затем идёт обработка транзактов в цикле до тех пор, пока (entry_time + случайное вещ. число от 0 до N1) не будет больше 3600 (entry_time это время прибытия текущего транзакта). Дальше вызываются для каждой кассы методы vhod и vihod в отдельных потоках. И далее

идет работа с переменными типа vector(gate, Emoment, line, Smoment) кассы с меньшим числом транзактов в очереди (если очереди равны, то транзакт идёт в первую очередь). В переменную line добавляется в конец номер текущего транзакта. Затем, если в Emoment не пусто, то в конец добавляется время окончания обслуживания предыдущего транзакта, иначе добавляется время входа в модель текущего транзакта. А так же в конец переменной gate добавляется номер текущего транзакта. Дальше вызывается service для вычисления времени окончания обслуживания.
И в самом конце, после окончания работы цикла, закрывается файл и программа прекращает работу.

4. Описание выходных данных

Файл result.txt, в котором содержится описание моделирования.

5. Код программы

```
#include <iostream>
#include <time.h>
#include <vector>
#include <fstream>
#include <unistd.h>
#include <thread>
#include <mutex>

using namespace std;

int R1=6, G1=9, N1=26; //начальные данные

double random(double min, double max)
{
   return ((double)rand()/RAND_MAX)*(max - min) + min;  //реализация функции UNIFORM из GPSS
}

mutex mtx;

class Cash
{
  private:
    double x;
    double y;
  public:
    Cash(double a, double b) {x = a; y = b;};
    void service();
    void vhod(double);
    void vihod(double, ofstream&);

    vector<int> line; //номера транзактов, ожидающих  окончания своего обслуживания
    vector<double> Smoment; //моменты времени, в которые закончится обслуживание транзактов
    vector<int> gate; //номера транзактов  входящих в модель
    vector<double> Emoment; // моменты времени входов транзактов в модель
```

```
};


void Cash::service() //Вычисление времени, в которое транзакт закончит обслуживаться
{
  double time = random (x, y);

  Smoment.push_back(Emoment.back() + time);
}


 void Cash::vihod(double entry_time, ofstream& f) //удаление транзактов, обслуживание которых закончилось, из очереди на обслуживание
{
    mtx.lock();
  if (Smoment.empty()==false)
  {
    for (int i = 0; i < Smoment.size();)
    {
      if (entry_time > Smoment[i] )
      {
        f << "В момент времени " << Smoment.front() << " транзакт с идентификатором " << line.front() << " вышел из модели.\n";
        Smoment.erase(Smoment.begin()+i);
        line.erase(line.begin()+i);
      }
      else
      {
        ++i;
      }
    }
  }
      mtx.unlock();
}


 void Cash::vhod(double entry_time) // удаление транзактов , обслуживание которых началось, из очереди на вход
{
   mtx.lock();
  if (Emoment.empty()==false)
  {
    for (int i = 0; i < Emoment.size();)
    {
      if (entry_time > Emoment[i] )
      {
        Emoment.erase(Emoment.begin()+i);
        gate.erase(gate.begin()+i);
```

```
      }
      else
      {
        ++i;
      }
    }
  }
    mtx.unlock();
}


int main ()
{
  srand((unsigned int)time(0));


  ofstream fout;
  fout.open("result.txt", ios::trunc);


  int number = 0;
  double entry_time = 0;

  Cash q1 (R1, N1); // первая касса
  Cash q2 (G1, N1); // вторая касса

  while ((entry_time += random(0, N1))<3600)
  {

    number++;

    thread th1 ([&]()
            {
              q1.vhod(entry_time);
            });

    thread th2 ([&]()
            {
              q2.vhod(entry_time);
            });

    thread th3 ([&]()
            {
              q1.vihod(entry_time, fout);
            });

        thread th4 ([&]()
            {
```

```
            q2.vihod(entry_time, fout);
           });

   th1.join();
   th2.join();
   th3.join();
   th4.join();

   if (q2.line.size() < q1.line.size())
   {
    fout << "В момент времени " << entry_time << " транзакт с идентификатором " << number << "
вошёл в модель.\n";
    fout << "В момент времени " << entry_time << " транзакт с идентификатором " << number << "
встал в очередь 2.\n";

    q2.line.push_back(number);//запись номера текущего транзакта

    if (q2.Emoment.empty()==false)
    {
     q2.Emoment.push_back(q2.Smoment.back());
     q2.gate.push_back(number);
    }
    else
    {
     q2.Emoment.push_back(entry_time);
     q2.gate.push_back(number);
    }
    q2.service();
   }
   else
    {
     fout << "В момент времени " << entry_time << " транзакт с идентификатором " << number << "
вошёл в модель.\n";
     fout << "В момент времени " << entry_time << " транзакт с идентификатором " << number << "
встал в очередь 1.\n";

     q1.line.push_back(number);//запись номера текущего транзакта

     if (q1.Emoment.empty()==false)
     {
      q1.Emoment.push_back(q1.Smoment.back());
      q1.gate.push_back(number);
     }
     else
     {
      q1.Emoment.push_back(entry_time);
      q1.gate.push_back(number);
     }
```

```
      q1.service();
    }
  }
  fout.close();
  return 0;
}
```

6. Результаты тестирования

В момент времени 0.284461 транзакт с идентификатором 1 вошёл в модель.
В момент времени 0.284461 транзакт с идентификатором 1 встал в очередь 1.
В момент времени 9.31552 транзакт с идентификатором 1 вышел из модели.
В момент времени 10.956 транзакт с идентификатором 2 вошёл в модель.
В момент времени 10.956 транзакт с идентификатором 2 встал в очередь 1.
В момент времени 18.9714 транзакт с идентификатором 2 вышел из модели.
В момент времени 20.8982 транзакт с идентификатором 3 вошёл в модель.
В момент времени 20.8982 транзакт с идентификатором 3 встал в очередь 1.
В момент времени 22.471 транзакт с идентификатором 4 вошёл в модель.
В момент времени 22.471 транзакт с идентификатором 4 встал в очередь 2.
В момент времени 30.4097 транзакт с идентификатором 5 вошёл в модель.
В момент времени 30.4097 транзакт с идентификатором 5 встал в очередь 1.
В момент времени 38.3704 транзакт с идентификатором 4 вышел из модели.
В момент времени 31.422 транзакт с идентификатором 3 вышел из модели.
В момент времени 41.7578 транзакт с идентификатором 6 вошёл в модель.
В момент времени 41.7578 транзакт с идентификатором 6 встал в очередь 2.
В момент времени 47.6141 транзакт с идентификатором 5 вышел из модели.
В момент времени 52.6328 транзакт с идентификатором 6 вышел из модели.
В момент времени 62.2224 транзакт с идентификатором 7 вошёл в модель.
В момент времени 62.2224 транзакт с идентификатором 7 встал в очередь 1.
В момент времени 77.2566 транзакт с идентификатором 8 вошёл в модель.
В момент времени 77.2566 транзакт с идентификатором 8 встал в очередь 2.
В момент времени 82.1152 транзакт с идентификатором 7 вышел из модели.
В момент времени 84.6434 транзакт с идентификатором 9 вошёл в модель.
В момент времени 84.6434 транзакт с идентификатором 9 встал в очередь 1.
В момент времени 89.3215 транзакт с идентификатором 8 вышел из модели.

● ● ●

В момент времени 3524.19 транзакт с идентификатором 273 встал в очередь 1.
В момент времени 3540.78 транзакт с идентификатором 273 вышел из модели.
В момент времени 3548 транзакт с идентификатором 274 вошёл в модель.
В момент времени 3548 транзакт с идентификатором 274 встал в очередь 1.
В момент времени 3569.32 транзакт с идентификатором 274 вышел из модели.
В момент времени 3570.51 транзакт с идентификатором 275 вошёл в модель.
В момент времени 3570.51 транзакт с идентификатором 275 встал в очередь 1.
В момент времени 3571.78 транзакт с идентификатором 276 вошёл в модель.
В момент времени 3571.78 транзакт с идентификатором 276 встал в очередь 2.

В момент времени 3573 транзакт с идентификатором 277 вошёл в модель.
В момент времени 3573 транзакт с идентификатором 277 встал в очередь 1.
В момент времени 3576.72 транзакт с идентификатором 275 вышел из модели.
В момент времени 3582.16 транзакт с идентификатором 278 вошёл в модель.
В момент времени 3582.16 транзакт с идентификатором 278 встал в очередь 1.
В момент времени 3592.3 транзакт с идентификатором 277 вышел из модели.
В момент времени 3588.19 транзакт с идентификатором 278 вышел из модели.
В момент времени 3596.28 транзакт с идентификатором 276 вышел из модели.
В момент времени 3597.98 транзакт с идентификатором 279 вошёл в модель.
В момент времени 3597.98 транзакт с идентификатором 279 встал в очередь 1.

7. Список литературы

1) Волосатова Т. М., Родионов С. В. Объектно-ориентированное программирование на C++;URL: http://bigor.bmstu.ru/?cnt/?doc=VU/base.cou (дата обращения: 11.10.2020).
2) Лекции и семинары по курсу объектно-ориентированное программирование.
3) Прикладное программирование на языке C++ : учебное пособие / Т.М. Волосатова, С.В. Родионов, Д.Т. Шварц. – Москва : Издательство МГТУ им. Н. Э. Баумана, 2015. – 146, [2] с. : ил.