

Министерство образования Российской Федерации

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

**И.Г. Губин**

**РАЗРАБОТКА СИСТЕМ  
АВТОМАТИЗИРОВАННОГО  
ПРОЕКТИРОВАНИЯ (САПР)**

**Учебное методическое пособие**

**2001**

**Губин И.Г.**

Разработка систем автоматизированного проектирования (САПР): Учебное методическое пособие. – Томск: Томский межвузовский центр дистанционного образования, 2001. – 38 с.

Методическое пособие содержит практические указания по выполнению лабораторных и контрольных работ по курсу "Разработка САПР", часть 1 и часть 2.

Предназначено для студентов, обучающихся на всех формах обучения с использованием дистанционных образовательных технологий.

© Губин Игорь Георгиевич, 2001

© Томский межвузовский центр  
дистанционного образования, 2001

## СОДЕРЖАНИЕ

Введение.....	4
1. Программа лекционного курса.....	4
2. Рекомендуемая литература .....	6
3. Последовательность выполнения лабораторных и контрольных работ .....	7
4. Описание пакета программ имитационного моделирования EVA.....	7
4.1.Базовый набор типовых элементарных сетей.....	7
4.2. Состав и структура системы моделирования.....	10
4.3. Графический редактор EEE .....	12
4.4. Язык описания E-сети EDEF .....	16
4.5. Функции статистики.....	19
4.6. Пакетный файл UNION.BAT .....	22
4.7. Графический отладчик GRDEB.....	23
5. Контрольная работа №1 .....	25
5.1. Задание на контрольную работу №1 .....	25
5.2. Содержание отчета по контрольной работе .....	26
5.3. Порядок выполнения работы.....	26
6. Контрольная работа №2 .....	32
7. Лабораторная работа №1.....	32
7.1. Задание на лабораторную работу №1 .....	32
7.2. Содержание отчета по лабораторной работе .....	34
7.3. Метод построения модели .....	34
8. Лабораторная работа №2.....	35
8.1. Задание на лабораторную работу №2 .....	36
8.2. Содержание отчета по лабораторной работе .....	37
8.3. Метод построения модели .....	37
9. Контрольные работы №№ 3-4.....	38

## **ВВЕДЕНИЕ**

Целью курса "Разработка САПР" (части 1 и 2) является изучение принципов проектирования и эксплуатации современных систем автоматизированного проектирования.

Часть 1 курса изучается в 10 семестре, а часть 2 - в 11 семестре общего учебного плана. В 10 семестре, при изучении 1 части курса, предусмотрены две контрольные работы (первая письменная, вторая компьютерная) и две лабораторные работы. В 11 семестре, при изучении 2 части курса, предусмотрены две компьютерные контрольные работы, две лабораторные работы (выполняются в Томске) и курсовой проект. В конце каждой части сдаются компьютерные экзамены.

### **1. ПРОГРАММА ЛЕКЦИОННОГО КУРСА**

#### **Часть 1**

##### **1. Введение.**

2. Методологические основы системотехнической деятельности инженера системотехника. Современное состояние и перспективы развития САПР. Организация СТД. Обеспечение целостности описания системы. Виды представлений СТС. Представление САПР. Формы представлений. Структурные модели. Обобщенная функционально-организационная структура ИАСУ. Основные требования к САПР и средства их реализации. Уровни автоматизации проектных работ. Типовая структура программно-информационного и лингвистического обеспечения САПР.

3. САПР как объект проектирования. Принципы создания САПР. Состав и структура САПР. Комплекс средств автоматизированного проектирования. Мониторные системы. Информационно-поисковые системы в САПР. Система управления базами данных. ПМК машинной графики. Компоненты видов обеспечения САПР. Математическое обеспечение САПР (МО САПР). Программное обеспечение САПР (ПО САПР). Информационное обеспечение САПР (ИО САПР). Техническое обеспечение САПР (ТО САПР). Лингвистическое обеспечение САПР (ЛО САПР). Методическое обеспечение САПР (МеО САПР). Организационное обеспечение САПР (ОО САПР). Классификация САПР. Тип объекта проектирования. Разновидность объектов проектирования. Сложность объекта проектирования. Уровень автоматизации проектирования. Комплексность автоматизации проектирования. Характер выпускаемых проектных документов. Число выпускаемых проектных документов. Число уровней в структуре ТО. Взаимодействие САПР с другими автоматизированными системами.

4. Формализация процесса проектирования. Основные понятия. Аспекты проектирования. Составные части процесса проектирования. Стадии

проектирования. Этапы проектирования. Проектные процедуры и проектные операции. Способы организации процесса проектирования. Различия между обычным проектированием и автоматизированным. Процессы АП. Процессы проектирования и процессы АП. Характеристики процесса проектирования и их влияние на АП. Среда АП. Организация АП. Среда проектного подразделения. Ресурсы ЭВМ. Этапы взаимодействия в АП. Состояние процессов АП. Время жизни процессов. Представление состояния процесса, рабочие состояния. Проблема ресурсов. Наличие ресурсов и конфликты при их распределении. Эффективность использования ресурсов.

5. Имитационное моделирование. Теория сетей Петри и моделирование систем. Сведения из теории комплектов. Структура сети Петри. Графы сетей Петри. Маркировка сетей Петри. Правила выполнения СП. Пространство состояний сети Петри. События и условия. Одновременность и конфликт. Моделирование аппаратного обеспечения. Моделирование программного обеспечения. Конструкции о взаимном исключении. Задача о производителе/потребителе. Механизмы синхронизации в СП (P- и V-операции над семафорами). Расширение и ограничения модели в СП. Маркированные графы (E-сети). Базовый набор типовых элементов сетей. Сеть типа T. Сеть типа F. Сеть типа J. Сеть типа X. Сеть типа Y. Сеть типа I. Генератор фишек. Поглотитель фишек. Язык описания E – сети (EDEF). Системные переменные Eva.

6. Эффективность и качество САПР.

7. Системотехническая деятельность в процессе создания САПР. Место системотехнических задач в процессе проектирования САПР. Анализ проблемной ситуации. Формирование ОЗП как задачи принятия решений и подготовка процесса ее решения. Декомпозиция ОЗП. Согласование результатов решения ЧЗП. Структура и содержание процесса создания САПР. Исследование и обоснование создания САПР. Техническое задание. Эскизный проект. Технический проект. Рабочая документация Изготовление несерийных компонентов КСАП. Ввод в действие.

## Часть 2

8. Сети связи ЭВМ. Краткая история Internet. Общие сведения о сетях связи ЭВМ. Назначение сетей. Взаимное соединение сетей. Многоуровневые архитектуры связи. Физический уровень. Уровень канала. Сетевой уровень. Транспортный уровень. Уровень сеанса. Уровень представления. Прикладной уровень. Возможности ЛВС. Компоненты ЛВС. Рабочие станции. Файловые серверы. Сетевые кабели. Сетевые адаптеры (CA). Прерывания. Базовый порт ввода/вывода (base i/o port). Топология сетей. Топология типа «звезда». Кольцевая топология. Шинная топология. Древо-видная структура ЛВС. Объединение вычислительных сетей. Сети с коммутацией пакетов и коммутацией каналов. Фазы связи ЭВМ. Протокол

X.25. Фазы связи протокола X.25. Архитектура сети систем SNA. Уровень канала передачи данных: примеры и анализ характеристик. Процедуры обработки ответов. Протокол с остановками и ожиданием. Протокол с N - возвращениями или непрерывная передача. Процедура выборочного повторения (селективного отказа). Анализ пропускной способности протокола с остановкой и ожиданием. Анализ пропускной способности протокола с N-возвращениями. Эффективность производительности и оптимальная длина пакетов. Высокоуровневое управление каналом передачи данных. Сетевой уровень: управление потоком и контроль перегрузки. Протокол X.25. Механизм управления протоколом в X.25. Управление окном с подтверждением в конце.

9. Сетевые операционные системы. Централизованные и одноранговые ЛВС. Сети с централизованным управлением. COC NetWare фирмы Novell. COC LAN Manager и LAN Server. Основы LAN Manager и LAN Server. Высокопроизводительная файловая система. Архитектура клиент сервер. COC Windows NT Advanced Server. Одноранговые ЛВС. COC Windows for Workgroups. COC NetWare Life и Personal NetWare. COC LANTastic. COC POWERLan. Сравнение производительности (времени доступа) для различных COC. ЛВС на базе ОС UNIX. Сетевая файловая система NFS. COC PC Interface/AADU. Системы POWERfusion и PoWERserve.

## **2. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА**

1. Норенков И.П., Маничев В.П. Основы теории и проектирования САПР. - М.:Высш. шк., 1990.
2. Костин А.Е., Шаньгин Ф.В., Организация и обработка структур данных в вычислительных системах. - М.: Высшая школа, 1987.
3. Разработка САПР. В 10 кн. Практ. пособие. Под ред. А.В.Петрова - М.: Высш. шк., 1990.
4. Шварц М. Сети связи: протоколы, моделирование и анализ: В 2-х ч. Пер. с англ. - М.: Наука, 1992.
5. Артамонов Г.Т., Тюрин В.Д. Топология сетей ЭВМ и многопроцессорных систем. -М: Радио и связь, 1991.-248 с.
- 6.Фролов А.В., Фролов Г.В. Локальные сети персональных компьютеров. -М.: "Диалог-Мифи", 1993.
7. Губин И.Г. Система имитационного моделирования на базе Е-сетей (руководство к лабораторным работам), Томск.: Препринт, 41 с. ТИАСУР, заказ N 291,1993
8. Губин И.Г. Методические указания по оформлению курсовых проектов (работ) для студентов специальности 22.03 Томск.: ТАСУР, 8 с. 1994.
9. NUTT G.J. Evaluation Nets for Computer System Performance Analysis.- AFIPS FJCC, 1972.-Pt.1 vol. 41.- Pp. 279-286.

### **3. ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ И КОНТРОЛЬНЫХ РАБОТ**

Лабораторные и контрольные работы выполняются в последовательности приведенной в методических указаниях по курсу. Перед выполнением лабораторных и контрольных работ необходимо изучить теоретический материал учебного пособия по курсу. Перед непосредственным выполнением конкретной работы, необходимо детально изучить рекомендуемые разделы, указанные в методических указаниях.

В первой части курса выполняются контрольные работы №№1-2 и лабораторные работы №№1-2. Контрольная работа №1 письменная, ее необходимо выполнить перед лабораторными работами №№1-2. Контрольная работа №2- компьютерная. Перед ее выполнением необходимо изучить разделы 1-6 учебного пособия "Разработка систем автоматизированного проектирования", (Организация системотехнической деятельности при создании САПР), часть 1. Лабораторные работы №№1-2 письменные. Контрольная работа №1, лабораторные работы №№1-2 и курсовой проект выполняются с использованием пакета программ имитационного моделирования EVA. Перед их выполнением необходимо изучить раздел 4 учебного пособия и методические указания по курсу.

Во второй части курса выполняются лабораторные работы №№3-4 (в Томске), компьютерные контрольные работы №№3-4 и курсовой проект. Перед выполнением контрольных работ необходимо изучить разделы 7-8 учебного пособия.

### **4. ОПИСАНИЕ ПАКЕТА ПРОГРАММ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ EVA**

#### **4.1. Базовый набор типовых элементарных сетей**

Пакет программ имитационного моделирования EVA используется при выполнении контрольной работы №2, лабораторных работ №№1-2 и курсового проекта. Данное описание предназначено для приобретения практических навыков моделирования сложных дискретных систем с помощью аппарата Е-сетей.

В 1962 году Карл Адам Петри в своей докторской диссертации "Связь автоматов" предложил в качестве инструмента моделирования дискретных параллельных систем концепцию сетей Петри. С тех пор теория сетей Петри активно развивалась, и в 70-х годах возникли Е-сети, позволяющие более адекватно описывать и оценивать моделируемые системы. Однако, в нашей стране данный аппарат пока широкого распространения не получил.

Е-сеть, как и сеть Петри, имеет структуру двудольного ориентированного графа, у которого вершины одного типа образуются позициями, а другого - переходами. Дуги могут связывать только переход с позицией или позицию с переходом.

По сравнению с сетями Петри Е-сети имеют следующие отличия:

- 1) входить и выходить из позиции может не более одной дуги;
- 2) существует несколько типов позиций:

простые (могут содержать не более одной фишки и изображаются кружком);

очереди (могут содержать произвольное число фишек и изображаются овалами);

разрешающие (выполняют управляющую функцию при определении откуда и куда перемещать фишки, изображаются квадратом);

фишки могут обладать набором атрибутов (в данной системе это вектор целых чисел языка Си типа long);

с каждым переходом Е-сети (изображается отрезком прямой линии) можно связать временную задержку, процедуру преобразования атрибутов проходящих через него фишек, а если переход имеет входную разрешающую позицию, то и разрешающую процедуру.

Следует отметить, что для усиления выразительных средств представленных выше Е-сетей сняты ограничения на общий вид разрешающих процедур, процедур преобразования атрибутов и процедур вычисления временных задержек для переходов. В предлагаемой системе эти процедуры могут иметь абсолютно произвольный вид и реализовывать любые вспомогательные вычисления и действия, необходимые пользователю.

В качестве базового набора типов элементарных сетей, из которых строится произвольная Е-сеть, используется набор, приведенный на рисунке 4.1.

Процесс работы Е-сети заключается в перемещении фишек из входных позиций переходов в выходные, в результате срабатывания переходов. Для этого на каждом шаге модельного времени определяются переходы, готовые сработать, и при этом выполняются следующие действия для каждого из переходов:

- 1) проверка выполнения условия готовности срабатывания перехода по разметке его входных и выходных позиций, а также условие завершения его фазы активности;

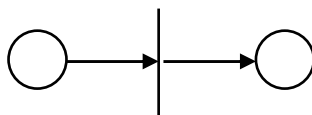
- 2) выполнение разрешающей процедуры, если таковая имеется у данного перехода;

- 3) определение длительности активной фазы перехода с помощью процедуры вычисления задержки;

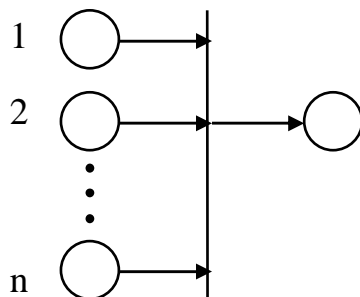
- 4) после завершения активной фазы фишки перемещаются из входных позиций в выходные и выполняется процедура преобразования атрибутов.



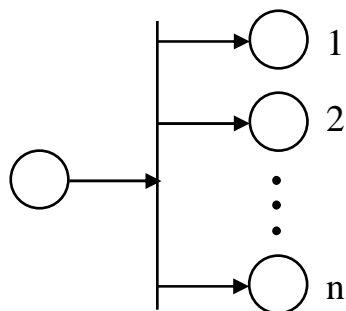
Простой переход



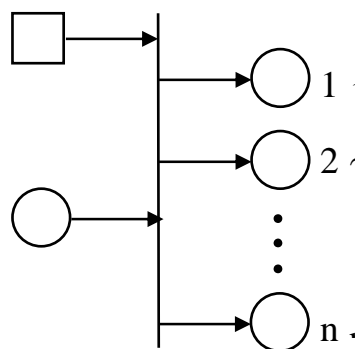
Объединение



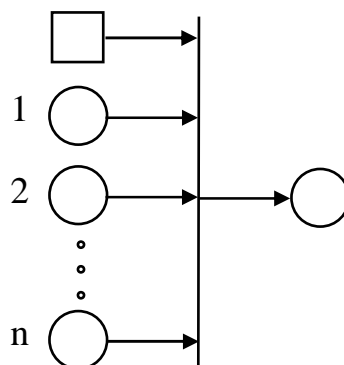
Размножение



Переключатель



Выборка



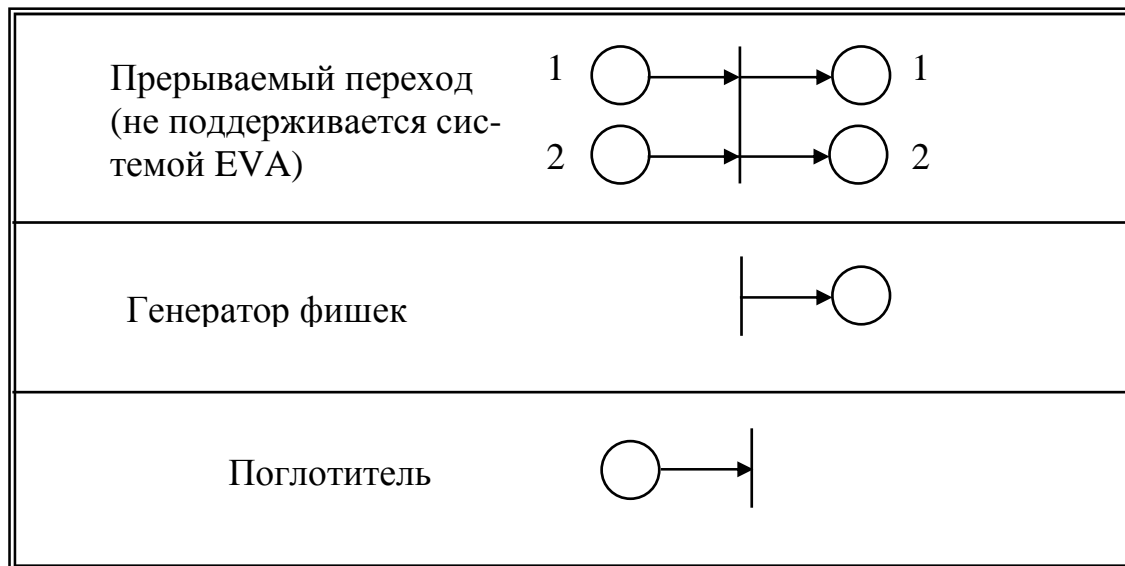


Рис. 4.1. Базовый набор типовых элементарных сетей

## 4.2. Состав и структура системы моделирования

### 4.2.1. Взаимосвязи компонентов системы EVA

Система имитационного моделирования EVA (E – net Valuation Adviser) включает в себя:

- 1) графический редактор (EEE) для задания структуры E - сети, описывающей моделируемую систему;
- 2) препроцессор (PREPARE) языка EDEF, который используя определения E-сети, созданные с помощью редакторов EEE и EDIT, генерирует описание моделируемой системы на языке Си;
- 3) модуль MODEL на языке Си, который в качестве своей составной части включает текст описания модели, подготовленный препроцессором PREPARE, и после компиляции представляет собой исполняемый файл программы интерпретации E-сети, описывающей работу моделируемой системы;
- 4) графический отладчик (GRDEB), который в случае необходимости отладки модели в графическом режиме отображения работы E-сети, используется вместо модуля MODEL;
- 5) компилятор (TCC) и редактор связей (TLINK) языка Turbo C;
- 6) другие вспомогательные компоненты, которые пользователем используются неявно, например пакетные файлы, с помощью которых вызываются на исполнение программы PREPARE, TCC и TLINK применительно к указанной модели и создается исполняемый файл модели.

Описательная часть E-сети (текстовый файл с расширением DEF), включающая определение длительностей активных фаз переходов, описа-

ние процедур преобразования атрибутов, выбор разрешающих процедур и т.д. создается с помощью текстового редактора. Описательная часть создается с помощью специального языка EDEF, разработанного на базе языка Си. В качестве текстового редактора необходимо использовать любой ASCII редактор, имеющийся на персональной ЭВМ пользователя, например встроенные редакторы NC или FAR.

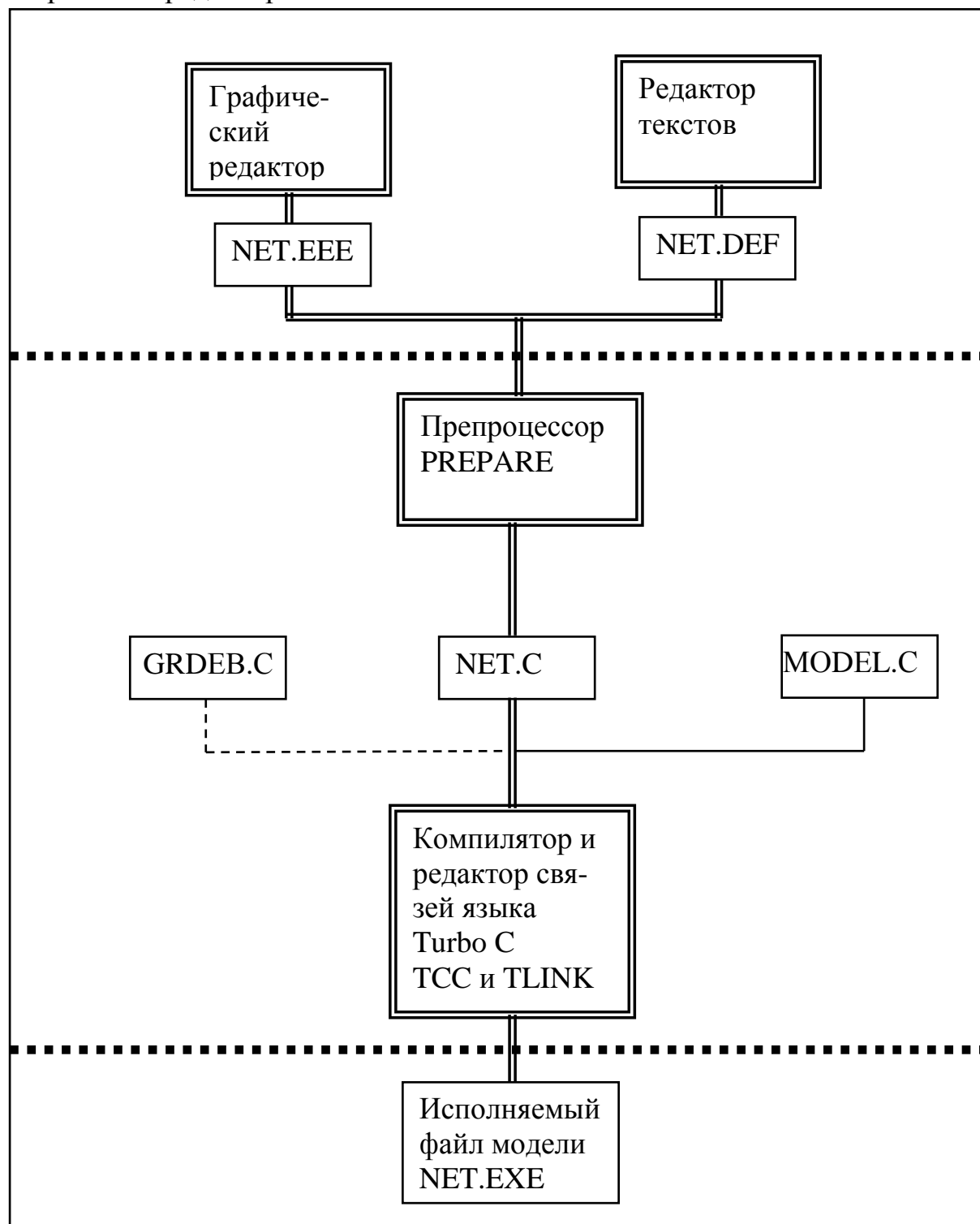


Рис.4.2.Схема процесса создания модели и взаимосвязи между файлами

Объем всех программных компонентов системы EVA занимает приблизительно 500 Кбайт дисковой памяти.

Рассмотрим подробнее процесс создания модели исследуемой системы с использованием перечисленных средств. Работа по созданию модели заключается в задании двух исходных файлов описания модели и их последовательного преобразования до получения исполняемого файла модели системы. На рисунке 4.2. приводится схема процесса создания модели и взаимосвязей между файлами (в предположении, что модель имеет имя NET). Участок выделенный квадратными точками означает, что работа с указанными программами и файлами автоматизирована с помощью пакетного файла UNION.BAT. Двойной рамкой представлены программные компоненты системы моделирования EVA, а одинарной - файлы, описывающие моделируемую систему.

Не будем останавливаться на этапах имитационного моделирования, которые не зависят от конкретного его математического аппарата: формулировка проблемы и определение целей имитации, выделение релевантных элементов и взаимосвязей моделируемой системы, формализация описания имитационной модели с использованием выбранного математического аппарата. Будем предполагать, что эти этапы пользователем уже пройдены, и он приступает к реализации имитационной модели (разработанной им Е-сети) на персональном компьютере. Рассмотрим более подробно процесс работы с каждой из перечисленных компонентов системы EVA.

### **4.3. Графический редактор ЕЕЕ**

#### **4.3.1. Запуск графического редактора**

Для построения имитационной модели сначала необходимо, с помощью графического редактора, задать (изобразить графически) имитируемую Е-сеть. Область, на которой возможно рисование, представляет собой прямоугольник размером 100x100 экранов персонального компьютера.

Для работы редактора требуется ПЭВМ с CGA или EGA-адаптером. Работа ведется в цветном режиме и с разрешением 320x200 точек. Редактор вызывается на исполнение файлом E.EXE. На экране появляется подсказка вида "Enter a command: ". Это означает, что редактор находится в режиме ожидания команды (командный режим). Существует также экраный режим, когда пользователь с помощью графического курсора может производить операции с графическим изображением Е - сети.

Курсор перемещается по экрану с помощью клавиш управления курсором, шаг его перемещения регулируется клавишами > (увеличение шага) и < (уменьшение шага). Для перемещения курсора в то место области рисования, которое не попадает на экран, необходимо воспользоваться клавишами управления курсором при нажатой клавише SHIFT.

Графический редактор ЕЕЕ позволяет выполнять следующие функции.

#### **4.3.2. Создание и удаление позиции, переходов и дуг**

Для этого предназначены команды:

S, Q, R - для создания простых позиций, очередей и разрешающих позиций соответственно. Эти команды запрашивают номер создаваемой позиции и переводят редактор в экранный режим, после чего с помощью графического курсора необходимо выбрать положение новой позиции и нажать клавишу ESC. На экране появляется соответствующая позиция и редактор переходит в командный режим.

T - для создания переходов. Команда запрашивает номер нового перехода и переводит редактор в экранный режим, после чего с помощью графического курсора выбирается положение одного из концов перехода и нажимается клавиша ESC. Затем перемещая курсор вверх или вниз определяется положение другого конца перехода и вновь нажимается ESC. На экране появляется новый переход и редактор переходит в командный режим.

A - для создания дуг. Команда переводит редактор в экранный режим, после чего надо с помощью графического курсора выбрать объект из которого будет исходить дуга (позицию или переход), разместив на нем курсор, и нажать клавишу ESC. Затем с помощью курсора рисуется дуга, второй конец которой необходимо расположить на том объекте, с которым соединяется первый объект, и нажимается вновь ESC. Если синтаксически такое соединение выбранных объектов недопустимо, то выдается сообщение об ошибке и дуга пропадает. Число точек перегиба дуги не может превышать шести.

D - для удаления позиций переходов и дуг. Команда переводит редактор в экранный режим, после чего необходимо с помощью графического курсора выбрать объект, который будет удаляться (позицию, переход или дугу), разместив на нем курсор, и нажать клавишу ESC. Выбранный объект удаляется и редактор переходит в командный режим.

#### **4.3.3. Маркировка позиции**

Для этого предназначена команда M. Она переводит редактор в экранный режим, затем с помощью курсора выбирается простая позиция или позиция-очередь, с которой будет проводится операция маркировки. Если выбрана простая позиция, содержащая уже фишку, то после нажатия кла-

виши ESC фишка будет удалена из нее. Если же фишки не было, то производится маркировка простой позиции. В случае позиции-очереди запрашивается число фишек, которые необходимо поместить в ней, и после нажатия клавиши ENTER, в овале позиции появляется число, равное выбранной длине очереди. После чего редактор переходит в командный режим.

#### **4.3.4. Просмотр сети, перемещение по ней и масштабирование**

Для этого предназначена команда V. Редактор переходит в экранный режим, и перемещая курсор, как описано выше, можно расположить на экране любую часть создаваемой сети. Для изменения масштаба изображения необходимо воспользоваться клавишами "+"(увеличение) и "-" (уменьшение). Для того, чтобы проще ориентироваться в области рисования создаваемая Е-сеть всегда заключена в прямоугольник, вне которого нет объектов сети. Для выхода в командный режим необходимо нажать клавишу ESC.

#### **4.3.5. Загрузка ранее созданной сети**

Для этого предназначена команда LOAD. У пользователя запрашивается имя файла, в котором хранится информация о загружаемой сети (файл создается командой SAVE). Если в момент загрузки область рисования пуста, то сеть загружается по центру этой области и редактор переходит в командный режим. Если же при загрузке в области рисования уже существуют объекты Е-сети, то пользователь может выбрать место, в которое будет производиться загрузка, при этом на экране изображается вся Е-сеть, очерченная прямоугольником. В этом случае с помощью графического курсора определяется положение одной из вершин прямоугольника, который будет очерчивать загружаемую сеть, и нажимается клавиша ESC.

Эта вершина определяется следующим образом:

- 1) если курсор установлен выше существующей Е-сети – задается левый нижний угол прямоугольника загружаемой сети;
- 2) если курсор установлен ниже существующей Е-сети - задается левый верхний угол прямоугольника загружаемой сети;
- 3) если курсор установлен слева и на уровне существующей Е - сети - задается правый нижний угол прямоугольника загружаемой сети;
- 4) если курсор установлен справа и на уровне существующей Е -сети - задается левый нижний угол загружаемой сети.

При этом курсор не может располагаться внутри прямоугольника существующей Е-сети. Графически это можно представить в следующем виде (рис.4.3.)



Рис. 4.3. Определение местоположения ранее созданной сети

#### 4.3.6. Сохранение сети в файле

Для этого предназначена команда SAVE. У пользователя запрашивается имя файла, в котором будет сохраняться информация о созданной Е-сети. Имя необходимо указывать без расширения, его автоматически добавляет редактор: <имя>.EEE. Этот файл затем можно использовать для загрузки сети с помощью команды LOAD.

#### 4.3.7. Синтаксический контроль сети

Для этого служит команда VERIFY. В случае выявления ошибок в соединениях объектов Е-сети, о них выдаются сообщения с указанием объекта.

#### 4.3.8. Распечатка сети на принтере или графопостроителе

Для этого предназначены команды:

**PRINT** -последовательно выводит на экран части созданной Е-сети, при этом необходимые участки выводятся на печатающее устройство с помощью программы CGAPRINT, которая загружается перед входом в редактор. Печатающее устройство должно быть совместимым по кодам с Epson FX - 800/FX-1000. Активация программы копии с экрана производится нажатием клавиш Shift-PrtSc;

**PLOT** - предназначена для вывода изображения созданной Е-сети на графопостроитель BENSON 1565. У пользователя запрашивается размер выводимого изображения (до 80 см) и имя выходного файла изо-

бражения, если необходимо выводить его не на графопостроитель, а в промежуточный файл. Возможно использование трех различных цветов на изображении.

Можно также использовать и другие сервисные функции (команды HELP, STAT и др).

#### 4.4. Язык описания Е-сети EDEF

После задания структуры Е-сети и сохранения информации о ней в файле (если модели дано имя NET, то файл имеет имя NET.EEE) с помощью графического редактора ЕЕЕ, можно перейти к определению описательной части Е-сети. Для этого используется специальный язык EDEF. Ниже приводится не формальное, но довольно строгое описание этого языка:

```

<программа> ::= [<комментарии>]<разделы>
<комментарий> ::= <произвольный текст без символов двоеточий ":">
<разделы> ::= [<раздел А>][<раздел I>][<раздел R>][<раздел D>]
               [<раздел Т>][<раздел S>][<раздел Е>]
<раздел А> ::= А: <заголовки, объявления переменных и функций
               языка Си>
<раздел I> ::= I: <составной оператор языка Си>
<раздел R> ::= R: <номер перехода><составной оператор языка Си>
               [<номер перехода><составной оператор языка Си>]...
<раздел D> ::= D: <номер перехода><составной оператор языка Си>
               [<номер перехода><составной оператор языка Си>]...
<раздел Т> ::= Т: <номер перехода><составной оператор языка Си>
               [<номер перехода><составной оператор языка Си>]...
<номер перехода> ::= Т<число>:
<раздел S> ::= S: <функция статистики>[<функция статистики>]...
<функция статистики> ::= <функция stat_s> | <функция stat_a> |
                           <функция stat_q> | <функция stat_t>
<раздел Е> ::= Е: <составной оператор языка Си>

```

Программа на этом языке представляет собой набор разделов, каждый из которых является фрагментом программы на языке Си.

Каждый раздел в программе на языке EDEF может встречаться не более одного раза и, при отсутствии необходимости в нем, может быть опущен.

Переменные, используемые в них, можно разделить на два класса: пользовательские и системные. Пользовательские используются при необходимости, а системные определяют состояние и параметры имитационной модели. Основными системными переменными, которыми исследователь может оперировать в программе, описывающей Е-сеть, являются:



finish - модельное время, при котором прекращается имитация, тип-long;  
 clock - текущее модельное время, только для чтения, тип-long;  
 r - значение разрешающей позиции текущего перехода, тип-int;  
 d - значение временной задержки текущего перехода, тип-long;  
 s[i].atr[j] - j-й атрибут i-ой простой позиции, тип-long;  
 q[i].last->atr[j] - j-й атрибут последней фишки в i-ой позиции - очереди, тип-long;  
 q[i].first->atr[j] - j-й атрибут первой фишки в i-ой позиции-очереди, тип-long;  
 q[i].nmark - количество фишек в i-ой позиции - очереди, только для чтения, тип-int;  
 s[i].nmark - 1 - означает, что простая позиция с номером i содержит фишку, 0 - означает, что простая позиция с номером i свободна, только для чтения, тип-int;  
 q[i].strategy - порядок обслуживания фишек в очереди:  
 F - обслуживание по принципу FIFO (по умолчанию),  
 L - обслуживание по принципу LIFO, тип-char.

Кроме системных переменных пользователь может воспользоваться системными функциями-датчиками случайных чисел:

rnd(n) - функция генерации псевдослучайных чисел равномерно распределенных на отрезке [0,n], n – положительный аргумент типа int;  
 expon(n) - функция генерации псевдослучайных чисел, распределенных по экспоненциальному закону с математическим ожиданием n, n - положительный аргумент типа int.

При каждом прогоне модели датчики случайных чисел автоматически инициализируются, поэтому результаты прогонов одной модели, содержащей обращения к этим функциям различны, что дает возможность накапливать статистику поведения исследуемой системы.

Разделы имеют следующие функциональные назначения и имена:

A: - раздел описаний переменных и функций пользователя. Эти переменные и функции могут использоваться в любых раз делах программы, кроме раздела S. Их имена должны начинаться со знака подчеркивания "\_", а во всем остальном описания полностью эквивалентны описанию переменных и функций языка Си. Никаких ограничений на количество и содержание тел функций не накладывается. Пользователь может в этом разделе использовать директивы компилятора Си "#include", "#define" и др.;

I: - раздел определения начального состояния модели. Этот раздел можно рассматривать как тело функции, которая выполняется непосредственно перед началом процесса имитации, поэтому в него можно включить операторы по заданию начальных значений системных и пользовательских переменных модели: число шагов моделирования (finish), значения атрибутов фишек в помеченных ими позициях и т.д.

В разделе возможно использование любых операторов и функций языка Си;

R: - раздел описания процедур определения значений разрешающих позиций. В этом разделе для переходов типа "переключатель" и "выборка" определяется значение системной переменной *г*. Это значение должно быть типа *int* и лежать в интервале от 1 до *n*, где *n* - число выходных (для "переключателя") или входных (для "выборки") позиций перехода. Номер определяемого перехода "*n*" указывается с помощью метки вида: "*Tn*: ". Текст программы, следующий за такой меткой и до следующей метки или конца раздела, является телом процедуры определения значения разрешающей позиции данного перехода. Тут возможно использование любых операторов и функций языка Си, среди которых должен быть оператор присваивания значения системной переменной *г*.

Если переход указанного типа не описан в этом разделе, то для определения значения разрешающей позиции используется стандартная процедура - определяется наименьший номер среди свободных выходных позиций (для "переключателя") или занятых позиций (для "выборки");

D: - раздел описания процедур вычисления временных задержек срабатывания переходов. в этом разделе для переходов, имеющих ненулевую задержку, определяется значение системной переменной *d*. Это значение должно быть типа *long* и положительно. Номер определяемого перехода "*n*" указывается с помощью метки вида "*Tn*: ". Текст программы, следующий за такой меткой и до следующей метки или конца раздела, является телом процедуры определения временной задержки данного перехода. Тут возможно использование любых операторов и функций языка Си, среди которых должен быть оператор присваивания значения системной переменной *d*.

Если переход не описан в этом разделе, то временная задержка для него устанавливается равной нулю;

T: - раздел описания процедур преобразования атрибутов фишек при срабатывании переходов. Номер определяемого перехода "n" указывается с помощью метки вида "Tn:". Текст программы, следующий за такой меткой и до следующей метки или конца раздела, является телом процедуры преобразования атрибутов фишек, проходящих через данный переход. Тут возможно использование любых операторов и функций языка Си. Доступ к атрибутам осуществляется с помощью системных переменных вида `s[i].atr[j]`.

Если переход не описан в этом разделе, то преобразования атрибутов не происходит;

S: - раздел определения элементов Е-сети, для которых проводится сбор статистики. В данном разделе разрешается использование только системных функций сбора статистики языка EDEF. Эти функции не возвращают никаких значений и позволяют накапливать статистику стандартного вида по различным объектам Е-сети. По окончании процесса моделирования вся накопленная статистика выводится на экран.

E: - раздел определения процедуры, которая выполняется по окончании процесса моделирования. Этот раздел можно рассматривать как тело функции, которая выполняется непосредственно после завершения процесса имитации, поэтому, например, в него можно включить операторы по обработке переменных пользователя, принимавших участие в работе модели, получению на их основе результатов моделирования и любые другие операторы и функции языка Си.

## 4.5. Функции статистики

4.5.1. `stat_s(Ns,Min,St,Nst)` - иницирует процесс накопления статистики для простой позиции с номером `Ns` в виде гистограммы времени ее занятости фишками и суммарного времени нахождения в ней фишек. Графически гистограмму можно представить в виде, изображенном на рисунке 4.4.

В гистограмме для каждого интервала времени занятости накапливается суммарное количество раз, которое фишка приходила в позицию и находилась там в течение времен, попадающих в данный интервал гистограммы. Те случаи, когда фишка находилась в позиции время меньшее, чем `Min`, или большее, чем `Min+St×Nst`, не регистрируются в гистограмме, но учитываются при подсчете суммарного времени занятости позиции.

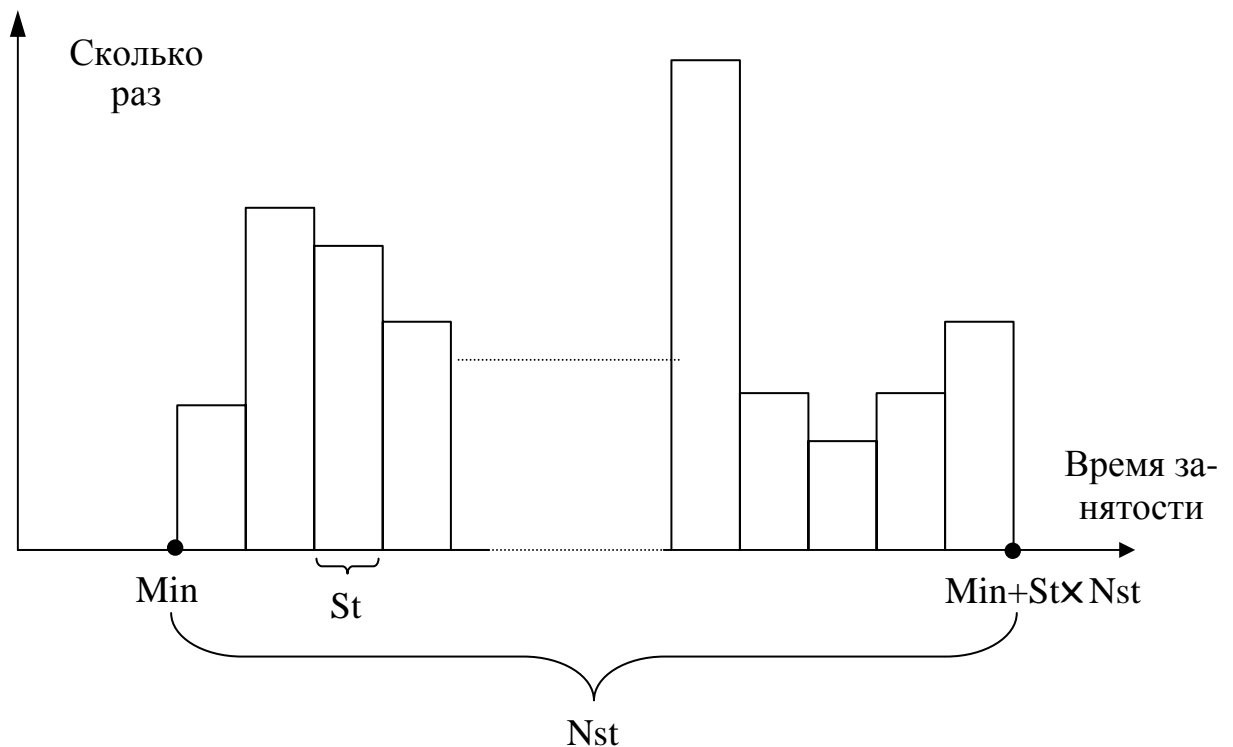


Рис. 4.4. Гистограмма времен занятости позиции

Параметры функции:

$N_s$  - номер простой позиции, для которой необходимо проводить сбор статистики, тип `int`;

$Min$  - левая граница первого интервала гистограммы занятости позиции, тип `long`;

$St$  - шаг гистограммы по времени занятости позиции, тип `long`;

$N_{st}$  - число шагов гистограммы, тип `int`.

4.5.2. `stat_a( $N_s, Na, Min, St, N_{st}$ )`-инициирует процесс накопления статистики для атрибута  $Na$  фишки из простой позиции с номером  $N_s$  в виде гистограммы значений этого атрибута. Гистограмма имеет вид, представленный выше для функции `stat_s()`. В ней для каждого интервала значений атрибута накапливается суммарное количество раз, которое фишка приходила в позицию и имела значения указанного атрибута, попадающие в данный интервал гистограммы. Те случаи, когда значение атрибута  $Na$  фишки, пришедшей в простую позицию  $N_s$ , меньше, чем  $Min$ , или больше, чем  $Min + St * N_{st}$ , не регистрируются в гистограмме.

Параметры функции:

$N_s$  - номер простой позиции, в которой необходимо проводить сбор статистики, тип `int`;

$Na$  - номер атрибута фишки для которого необходимо проводить накопление статистики, тип `int`.

Min - левая граница первого интервала гистограммы значений атрибута, тип long;

St - шаг гистограммы по значениям атрибута, тип long;

Nst - число шагов гистограммы, тип int.

4.5.3. stat\_q(Nq,Min,St,Nst) - иницирует процесс накопления статистики для позиции-очереди с номером Nq в виде гистограммы значений длины очереди, определения максимальной длины очереди и интегрального времени занятости позиции (сумма  $\{\phi(l \times t)\}$ , где l - длина очереди, t - интервал времени, в течение которого сохранялась длина l).

Гистограмма имеет вид, представленный выше для функции stat\_s(). В ней для каждого интервала длины очереди накапливается суммарное время, в течение которого сохранялись длины очереди, попадающие в данный интервал гистограммы. Те случаи, когда значение длины очереди меньше, чем Min, или больше, чем  $Min + St \times Nst$ , не регистрируются в гистограмме, но учитываются при определении максимальной длины очереди и интегрального времени занятости позиции.

Параметры функции:

Nq - номер позиции-очереди, для которой необходимо проводить сбор статистики, тип int;

Min - левая граница первого интервала гистограммы значений длины очереди, тип int;

St - шаг гистограммы по значениям длины очереди, тип int;

Nst - число шагов гистограммы, тип int.

4.5.4. stat\_t(Nt,Min,St,Nst) - иницирует процесс накопления статистики для перехода с номером Nt в виде гистограммы времен его активности (времен задержек), вычисления суммарного времени активности и числа фишек, прошедших через переход. Гистограмма имеет вид, представленный выше для функции stat\_s(). В ней для каждого интервала времени активности накапливается суммарное количество раз, которое переход был активным в течение времен, попадающих в данный интервал гистограммы. Те случаи, когда переход был активным время меньшее, чем Min, или большее, чем  $Min + St \times Nst$ , не регистрируются в гистограмме, но учитываются при подсчете суммарного времени активности и числа прошедших через него фишек.

Параметры функции:

Nt - номер перехода, для которой необходимо проводить сбор статистики, тип int;

Min - левая граница первого интервала гистограммы времен активности перехода, тип long;

St - шаг гистограммы по времени активности перехода, тип long;

Nst - число шагов гистограммы, тип int.

Описание Е-сети на языке EDEF создается с помощью любого ASCII-редактора текстов. Если модель имеет имя NET, то файл ее описания должен иметь имя NET.DEF.

В отличие от существующих систем имитационного моделирования на базе Е-сетей, язык EDEF позволяет во всех разделах использовать функции пользователя на языке Си произвольной сложности. Это дает возможность "обрамления" процесса имитации программами пользователя: создания собственного интерфейса между моделью и исследователем; создания точек прерывания моделирования по достижению определенных состояний; динамическое изменение параметров модели и другие полезные возможности.

#### 4.6. Пакетный файл UNION.BAT

После создания файлов структуры Е-сети и ее описания (NET.EEE и NET.DEF) пользователь с помощью пакетного файла UNION.BAT, вызывающего препроцессор PREPARE, объединяет их в один файл NET.C, содержимое которого является программой на языке Си. Препроцессор PREPARE также осуществляет синтаксический контроль структуры Е-сети. На этом этапе необходимо задать число атрибутов у фишек Е-сети.

Затем, также с помощью пакетного файла UNION.BAT, проводится компиляция файла NET.C совместно с файлом MODEL.C и создается исполняемый файл имитационной модели NET.EXE. Файл готов к исполнению. Если при этом не выявлены ошибки, то процесс создания модели завершен.

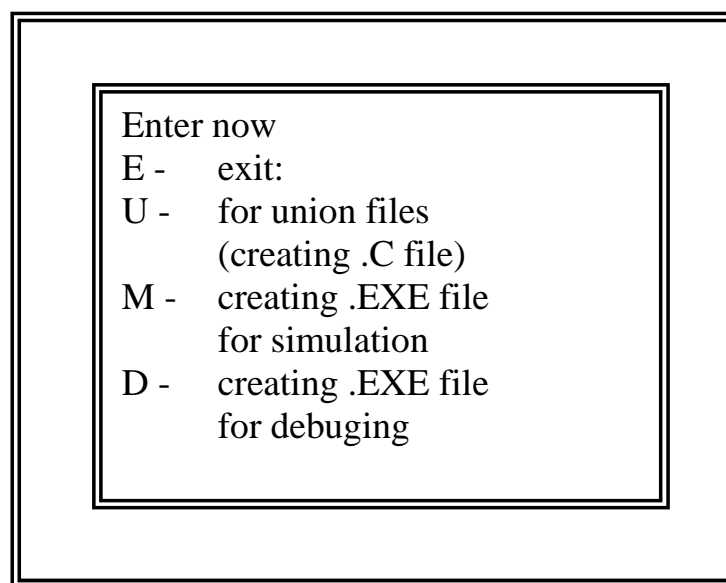
При желании можно использовать графический отладчик GRDEB для отладки работы Е-сети, для чего необходимо провести компиляцию файла NET.C совместно с файлом GRDEB.C и создать исполняемый файл отладочного варианта модели NET.EXE, что также автоматизируется пакетным файлом UNION.BAT.

Меню, которое предоставляет пользователю файл UNION.BAT имеет следующий вид, представленный на рисунке 4.5.

Опция "U" используется для вызова на исполнение препроцессора языка EDEF и получения файла описания модели на языке Си с именем NET.C на основе файлов NET.EEE и NET.DEF.

Опция "M" используется для получения исполняемого файла модели NET.EXE путем компиляции и линковки файла NET.C совместно с другими системными файлами.

Опция "D" используется для получения отладочного варианта исполняемого файла модели NET.EXE путем компиляции и линковки файла NET.C совместно с файлами графического отладчика GRDEB и другими системными файлами. Работа графического отладчика описана в подразделе 4.7.



Enter [ U | M | D | E ]:

Рис.4.5. Меню командного файла UNION.BAT

#### 4.7. Графический отладчик GRDEB

Файл NET.EXE отладочного варианта (полученный с помощью опции "D" файла UNION.BAT) позволяет исполнять модель под управлением графического отладчика GRDEB.

Процесс отладки модели в графическом режиме базируется на визуальном контроле на экране дисплея хода процесса имитации (перемещения фишек по позициям) в пошаговом или пошагово – непрерывном режимах. При этом можно просматривать отдельные участки модели или всю ее, в зависимости от выбранного масштаба графического изображения Е-сети и расположения ее на экране. Контролировать состояние модели, ее параметров, проводить контрольный просмотр накапливаемой статистики.

В самом начале работы отладчика на экране изображается Е-сеть, представляющая модель. Пользователь может с помощью клавиш управления курсором, и клавиш "+" , "-" выбрать желаемый масштаб и расположение сети на экране, как описано в подразделе 4.3.4. для команды "V" графического редактора ЕЕЕ. Для завершения этой операции необходимо нажать клавишу ESC. После этого начинается непосредственно процесс имитации под управлением отладчика.

В левом верхнем углу экрана появляется сообщение о текущем модельном времени: "Time: 0". Это всегда означает, что отладчик находится в режиме ожидания команды.

### Команды графического отладчика:

- T** - команда трассировки перемещения фишек по Е-сети в течение одного очередного такта работы модели (единицы модельного времени). После каждого срабатывания перехода необходимо нажимать любую клавишу до тех пор, пока не завершится очередной такт;
- G** - команда прогона модели с текущего момента модельного времени до указанного времени. После нажатия клавиши **G** в левом верхнем углу экрана появляется запрос "Go to :", после чего необходимо ввести время до которого должна исследоваться работа модели. Если указанное время меньше текущего, то никаких действий не происходит, если же указанное время превышает время завершения моделирования, определенное в модели системной переменной *finish*, то моделирование продолжается до времени, совпадающим со значением указанной переменной;
- N** - команда прогона модели до ближайшего момента модельного времени, когда в системе сможет сработать какой - либо переход (переместиться хотя бы одна фишка). При этом автоматически для такого момента времени исполняется команда трассировки **T**. Команду **N** нельзя использовать в начальный момент моделирования, так как время равно нулю;
- R** - команда промежуточной выдачи на экран статистики, накапливаемой в процессе имитации в соответствии с разделом "S:" программы на языке EDEF. После выдачи статистики для продолжения работы необходимо нажать любую клавишу;
- <**F1**> - нажатие функциональной клавиши **F1** позволяет вывести на экран вспомогательную информацию по командам графического редактора. Для продолжения работы необходимо нажать любую клавишу;
- <**F2**> - нажатие функциональной клавиши **F2** позволяет в любой момент имитации с помощью клавиш управления курсором и клавиш "+", "-" выбрать масштаб и новое удобное расположение Е-сети на экране для команды " **V**" графического редактора **EEE**. Для завершения этой операции необходимо нажать клавишу **ESC**;



<F10> - нажатие функциональной клавиши F10 завершает работу графического отладчика. При этом на экран выводится накопленная за период моделирования статистика по объектам Е-сети.

Примечание. Первой исполняемой командой отладчика обязательно должна быть команда "Т".

Для работы редактора требуется ПЭВМ с CGA или EGA - адаптером. Работа ведется в цветном режиме и с разрешением 320x200 точек.

## 5. КОНТРОЛЬНАЯ РАБОТА №1

### Моделирования работы парикмахерской

**Цель работы:** Практическая работа с пакетом программ имитационного моделирования EVA на примере моделирования работы парикмахерской.

**Используемое программное обеспечение:** программа имитационного моделирования EVA.

#### 5.1. Задание на контрольную работу №1

В парикмахерской имеются мужской и женский залы, в каждом из которых работает по три парикмахера. Времена обслуживания клиентов для каждого из мастеров различны и равномерно распределены на некоторых интервалах. Причем среднее время обслуживания мужчин меньше, чем женщин. Для каждого зала имеется своя очередь. Посетители приходят с улицы через некоторый случайный распределенный экспоненциально интервал времени и становятся в очередь своего зала. Как только освобождается какой-либо из трех мастеров, очередной клиент поступает к нему на обслуживание. Пол приходящего в парикмахерскую клиента задается случайным образом и равновероятно.

Требуется определить максимальные длины для каждой из очередей и количество обслуженных мужчин и женщин за десятичасовой рабочий день.

Числовые данные для моделирования.

Время обслуживания мужчин для каждого из парикмахеров равномерно распределены на следующих временных интервалах:

первый парикмахер – от 10 до 25 мин.

второй парикмахер – от 10 до 30 мин.

третий парикмахер – от 10 до 35 мин.

Время обслуживания женщин для каждого из парикмахеров также равномерно распределены на следующих временных интервалах:

первый парикмахер – от 20 до 40 мин.

второй парикмахер – от 20 до 45 мин.

третий парикмахер – от 20 до 80 мин.

Интервал времени между приходом клиентов в парикмахерскую имеет случайный экспоненциально – распределенный закон с математическим ожиданием равным семи:  $\text{expon}(7)$ .

## 5.2. Содержание отчета по контрольной работе

Отчет по контрольной работе должен содержать результаты работы программы имитационного моделирования EVA, включающего в себя следующие сведения.

1. Файл графического изображения моделируемой системы (парикмахерской) в формате графического редактора EEE (имеющий расширение EEE).
2. Текстовый файл, описывающий функционирование этой E-сети, на языке EDEF (имеющий расширение DEF), с подробными комментариями.
3. Текстовый файл отчета (статистики) по очередям 1 и 2.
4. Исполняемый файл модели (имеющий расширение EXE).
5. Файл титульного листа к лабораторной работе.

## 5.3. Порядок выполнения работы

1. Перед выполнением лабораторной работы изучите раздел 4 учебного пособия "Разработка систем автоматизированного проектирования", (Организация системотехнической деятельности при создании САПР), часть 1.

2. Изучите раздел 4 данного методического указания.

3. С помощью графического редактора пакета имитационного моделирования EVA – EEE (исполняемый файл E.EXE) постройте структуру E-сети. Сохраните файл на диске, например с именем LAB1.EEE. Расширение при сохранении вводить не надо, оно присваивается редактором автоматически.

4. С помощью любого ASCII редактора текста создайте описательный файл функционирования этой E-сети на языке EDEF. Присвойте файлу такое же имя, как и графическому, но с расширением DEF (в нашем случае файл будет иметь имя LAB1.DEF).

5. Из командной строки (NC или FAR) запустите командный файл UNION.BAT с указанием имени созданных файлов (без расширения). Для файлов с именем LAB1 командная строка будет иметь вид:

C:\> UNION.BAT LAB1

6. Из меню командного файла выберите опцию "U".

7. В ответ на запрос программы "Type model name :" введите имя созданных файлов - LAB1 и нажмите клавишу <Enter>.

8. После появления на экране сообщения "All correct!" и "Press any key to continue ..." – нажмите любую клавишу для продолжения.

9. После появления на экране сообщения "Enter number of attributes", введите необходимое число используемых в модели атрибутов. В выполняемом примере используется только один атрибут, поэтому необходимо ввести цифру 1 и нажать клавишу <Enter>.

При появлении сообщений об ошибках в структуре Е-сети типа:

```
Syntax error in transition 10
```

```
-----
```

```
Sorry, 1 error(s) detected. Exit
```

```
Press any key to continue ...
```

Необходимо нажать любую клавишу для продолжения, исправить ошибки в структуре Е-сети и повторить запуск командного файла UNION.BAT. К пункту 10 можно перейти только после успешного выполнения этого файла.

10. После успешного выполнения командного файла UNION.BAT запустите его снова и выберите из меню опцию "D". После этого будет создан исполняемый файл LAB1.EXE.

11. Запустите этот файл, введите модельное время (в нашем случае 600) и исследуйте работу парикмахерской.

Для переназначения отчета в текстовый файл (например с именем "LAB1.TXT"), необходимо запустить программу в следующем виде:

```
C:\> LAB1.EXE > LAB1.TXT
```

В этом случае создастся текстовый файл, в котором будут содержаться статистические данные отчета (заказанные в разделе S: файла описания сети).

Е-сеть, моделирующая функционирование такой парикмахерской, может иметь вид, приведенный на рисунке 5.1.

Структура этой Е-сети (ее изображение) задается с помощью графического редактора ЕЕЕ (см. раздел 4.3. данных методических указаний).

Каждая фишка в сети имеет один атрибут, отвечающий за пол клиента (1 - мужчина, 2 - женщина). Переход 1 генерирует поступление фишек (клиентов) в парикмахерскую и при этом устанавливается значение их атрибута (1 или 2). Разрешающая позиция 1, перехода 2, определяет по этому атрибуту в какую из очередей поместить клиента (очередь 1 - в мужской зал, очередь 2 - в женский).

Через переходы 3 и 4 фишки из очередей поступают в первую из свободных выходных позиций этих переходов (на обслуживание к мас-

терам). Срабатывание переходов 5-10 означает, что соответствующий посетитель обслужен и покидает парикмахерскую.

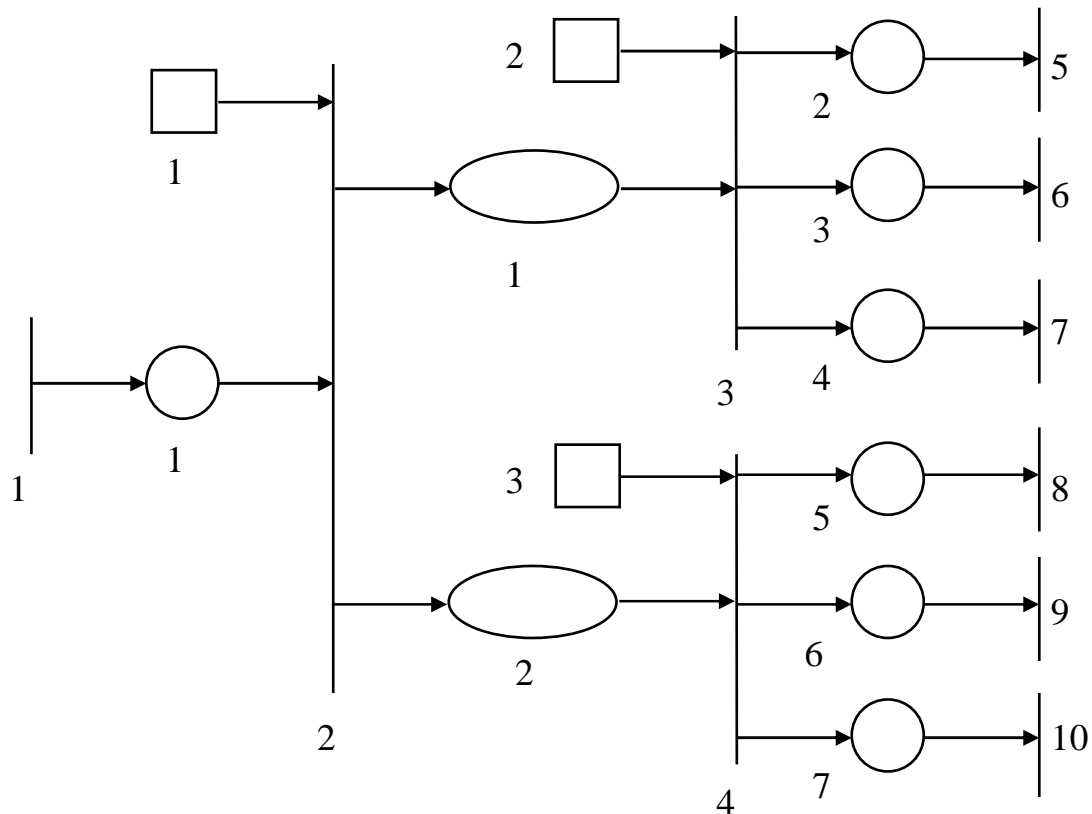


Рис. 5.1. Е-сеть моделирующая функционирование парикмахерской

Временные задержки срабатывания переходов 5-10 соответствуют длительности обслуживания клиентов соответствующими мастерами, перехода 1 - интервалу времени, через который поступают клиенты в парикмахерскую.

Программа, описывающая функционирование этой Е-сети на языке EDEF (см. раздел 4.4.) имеет следующий вид:

```
-----
data: 15.4.89      Rev 1.00
-----
```

```
A: int _Men = 0, _Women = 0; /* Суммарные количества обслуженных */
                             /* мужчин и женщин */
```

```
I: fprintf(stderr, "Enter Finish: ");
   scanf("%ld", &finish);
```

R:

```
T2: r = s[1].atr[1]; /* Если атрибут равен 1, то пришел */
      /* мужчина и он ставится в очередь 1 */
      /* Если атрибут равен 2, то пришла */
      /* женщина, и она ставится в очередь 2 */
```

T:

```
T1: if (rnd(100) > 50) s[1].atr[1] = 1L; /*Задание пола */
      else s[1].atr[1] = 2L;           /*клиента */

T3: _Men++; /* Подсчет общего числа обслуженных */
T4: _Women++; /* мужчин и женщин */
```

D:

```
T1: d = expon(7); /* Интервал времени между приходом */
      /* клиентов в парикмахерскую */

T5: d = 10 + rnd(15); /* Временные интервалы обслуживания */
T6: d = 10 + rnd(20); /* мужчин для каждого из парикмахеров */
T7: d = 10 + rnd(25);
T8: d = 20 + rnd(20); /* Временные интервалы обслуживания */
T9: d = 20 + rnd(45); /* женщин для каждого из парикмахеров */
T10: d = 20 + rnd(60);
```

S:

```
stat_q(1,1,1,10); /* Накопление статистики по очере- */
stat_q(2,1,1,15); /* дям 1 и 2 */
```

E: /\* Вывод на экран общего числа обслуженных мужчин и женщин \*/

```
printf("\n\n\n\n\n");
printf(" men = %d women = %d\n", _Men, _Women);
```

Создать файл описания модели можно с помощью любого текстового ASCII редактора.

В разделе A: описаны две переменных пользователя \_Men и \_Women, которые являются счетчиками обслуженных мужчин и женщин.

В разделе I: запрашивается ввод с клавиатуры значения системной переменной finish (число шагов работы модели). Можно условно считать один шаг модельного времени равным одной минуте реального времени. Поэтому, если задать значение этой переменной равным 600, то это означает, что будет осуществляться моделирование десяти часов работы парикмахерской.

В разделе R: определена разрешающая процедура перехода T2: - переместить фишку из позиции 1 по первой сверху выходной дуге перехода в очередь 1, если ее атрибут равен 1 (мужчина), или по второй дуге в очередь 2, если атрибут равен 2 (женщина). Остальные разрешающие процедуры для переходов T3 и T4 выбраны по умолчанию, следовательно, фишка перемещается в первую сверху свободную позицию.

В разделе T: определены процедуры преобразования атрибутов для переходов:

T1 - устанавливается значение атрибута фишки в позиции 1 на 1 или 2 (задание пола клиента);

T3 - фактически атрибут не изменяется, но при срабатывании перехода 3 счетчик обслуженных мужчин увеличивается на единицу;

T4 - фактически атрибут не изменяется, но при срабатывании перехода 4 счетчик обслуженных женщин увеличивается на единицу;

Процедуры преобразования атрибутов для остальных переходов не определены, а значит, атрибуты фишек при переходе через них не изменяются.

В разделе D: определены временные задержки срабатывания переходов T1 и с T5 по T10, выраженные в шагах модельного времени, а следовательно, в минутах работы парикмахерской. Временные задержки для остальных переходов не определены, что эквивалентно их равенству нулю.

В разделе S: используются системные функции накопления статистики для очередей q1 и q2. При этом в гистограммах для каждой из длин очередей подсчитывается общее время, в течение которого данная длина сохранялась. Для очереди q1 (мужчин) максимальная регистрируемая длина равна 10, а для очереди q2 (женщин) - 15.

В разделе E: определены действия системы по окончанию моделирования (перед выводом статистики на экран). Выводится на экран количество обслуженных за период моделирования мужчин и женщин.

Для подготовки исполняемого файла модели (вариант с графическим отладчиком GRDEB) необходимо воспользоваться пакетным файлом UNION.BAT (см. раздел 4.6.). При этом сначала выбирается опция "U", а затем - "D". Процесс прогона модели заключается в исполнении полученного в результате этого файла. Работа модели ведется под управлением графического отладчика GRDEB (см. раздел 4.7.).

Накопленная при моделировании статистика по очередям q1 и q2 выводится на экран в следующем виде:

```
***** _R_E_P_O_R_T___F_O_R___Q_U_E_U_E_S___*****
```

Queue 1 :

Time integral: 89

Max length : 2

Histogram for queue length:

Length	Time
~~~~~	~~~~~
[ 1 - 2)	39
[ 2 - 3)	25
[ 3 - 4)	0
[ 4 - 5)	0
[ 5 - 6)	0
[ 6 - 7)	0
[ 7 - 8)	0
[ 8 - 9)	0
[ 9 - 10)	0
[ 10 - 11)	0

-----  
Queue 2 :

Time integral: 8104

Max length : 14

Histogram for queue length:

Length	Time
~~~~~	~~~~~
[ 1 - 2)	4
[ 2 - 3)	10
[ 3 - 4)	17
[ 4 - 5)	26
[ 5 - 6)	22
[ 6 - 7)	27
[ 7 - 8)	47
[ 8 - 9)	57
[ 9 - 10)	17
[ 10 - 11)	23
[ 11 - 12)	13
[ 12 - 13)	5
[ 13 - 14)	15
[ 14 - 15)	32
[ 15 - 16)	0

-----  
Из этих результатов видно, что за время моделирования максимальная длина очереди для мужчин была равна 2, а для женщин - 14. Также можно определить суммарное время, в течение которого сохранялась та или иная длина очереди для мужчин и женщин.

Для получения оптимального варианта работы парикмахерской можно варьировать числом мастеров и длительностью обслуживания клиентов для выбранной скорости входного потока посетителей.

## **6. КОНТРОЛЬНАЯ РАБОТА №2**

Контрольная работа №2 - компьютерная. Перед ее выполнением необходимо изучить разделы 1-4 учебного пособия "Разработка систем автоматизированного проектирования", (Организация системотехнической деятельности при создании САПР), часть 1. Контрольная работа содержит вопросы по указанным разделам.

## **7. ЛАБОРАТОРНАЯ РАБОТА №1**

### **Моделирования процесса сборки и обжига детали**

**Цель работы:** Практическая работа с пакетом программ имитационного моделирования EVA на примере моделирования процесса сборки и обжига детали.

**Используемое программное обеспечение:** программа имитационного моделирования EVA.

### **7.1. Задание на лабораторную работу №1**

Производство деталей определенного вида включает длительный процесс сборки, заканчивающийся коротким периодом обжига в печи. Поскольку содержание печи обходится довольно дорого, несколько сборщиков используют одну печь, в которой одновременно можно обжигать только одну деталь. Сборщик не может начать новую сборку, пока не извлечет из печи предыдущую деталь.

Таким образом сборщик работает в следующем режиме.

1. Собирает деталь.
2. Ожидает возможности использования печи по принципу "первым пришел - первым обслужен".
3. Использует печь.
4. Возвращается к операции 1.

Время, необходимое на различные операции (сборка, обжиг), и стоимость операций и изделий представлены в таблице 7.1.



Таблица 7.1. Время, необходимое на различные операции и стоимость операций и изделий

Номер варианта	Операции		Стоимость операций и изделий			
	Сборка, мин	Обжиг, мин	Зарплата сборщика, долла- ров/час	Стоимость печи за 8-ми часовой рабочий день, долларов	Цена ма- териала на одну деталь, долларов	Стои- мость готовой детали, долларов
1	30+rnd(5)	8+rnd(1)	3,71	80	1	10
2	29+rnd(5)	8+rnd(2)	3,72	79	2	9
3	28+rnd(5)	8+rnd(3)	3,73	78	3	8
4	27+rnd(5)	8+rnd(4)	3,74	77	2	7
5	26+rnd(5)	8+rnd(5)	3,75	76	1	6
6	25+rnd(5)	9+rnd(1)	3,76	75	2	7
7	24+rnd(5)	9+rnd(2)	3,77	74	3	8
8	23+rnd(5)	9+rnd(3)	3,78	73	2	9
9	22+rnd(5)	9+rnd(4)	3,79	72	1	10
10	30+rnd(3)	9+rnd(5)	3,80	71	2	0
11	29+rnd(3)	7+rnd(1)	3,79	70	3	9
12	28+rnd(3)	7+rnd(2)	3,78	69	2	8
13	27+rnd(3)	7+rnd(3)	3,77	68	1	7
14	26+rnd(3)	7+rnd(4)	3,76	67	2	6
15	25+rnd(3)	7+rnd(5)	3,75	66	3	7
16	24+rnd(3)	8+rnd(5)	3,74	65	2	8
17	23+rnd(3)	8+rnd(5)	3,73	64	1	9
18	22+rnd(3)	8+rnd(4)	3,72	63	2	10
19	21+rnd(3)	8+rnd(4)	3,71	62	3	9
20	20+rnd(3)	8+rnd(3)	3,70	61	2	8

Необходимо построить модель описанного процесса с помощью аппарата Е-сетей и провести моделирование. Имея эту модель, следует определить оптимальное число сборщиков, использующих одну печь. Под оптимальным понимают такое число, которое дает максимальную прибыль. Моделирование необходимо выполнить в течении 40 часов модельного времени в предположении, что в течении рабочего дня нет перерывов, а рабочие дни идут без выходных дней.

## 7.2. Содержание отчета по лабораторной работе

Отчет по лабораторной работе должен содержать результаты работы программы имитационного моделирования EVA, включающего в себя следующие сведения.

1. Файл графического изображения моделируемой системы (участка по производству деталей) в формате графического редактора EEE (имеющий расширение EEE).
2. Текстовый файл, описывающий функционирование этой Е-сети, на языке EDEF (имеющий расширение DEF), с подробными комментариями.
3. Текстовый файл отчета для случая работы 4, 5 и 6 –ти сборщиков.
4. Исполняемый файл модели (имеющий расширение EXE).
5. Файл титульного листа к лабораторной работе.

## 7.3. Метод построения модели

В данном случае есть два вида ограничивающих условия, определяющих структуру Е-сети. Во-первых, имеется лишь одна печь. Во-вторых, существует некоторое фиксированное число сборщиков, работающих в системе. Естественно отождествить сборщиков с транзактами. Тогда можно считать, что сборщики циркулируют в системе. Аналогично тому, как они периодически осуществляют сборку и обжиг, транзакты циркулируют в Е-сети.

В реальной системе, после того как сборщик вынимает из печи обожженную деталь, он возвращается на свое рабочее место и начинает новый этап сборки.

Т. о. структура Е-сети должна содержать обратную связь по которой сборщик должен возвратиться к началу операции сборки, причем занять после этого именно ту позицию которую он покинул после сборки. Для реализации этого условия необходимо использовать переключатель. Причем общее число транзактов (сборщиков), циркулирующих в модели, будет соответствовать количеству выходных позиций переключателя.

Для того чтобы вычислить прибыль, соответствующую заданному числу сборщиков, необходимо знать, сколько готовых деталей они сделали в течение моделируемого периода. Число использования печи в точности совпадает с этим значением. Процесс вычисления прибыли необходимо автоматизировать путем помещения расчетной формулы в раздел Е: описательного файла функционирования Е-сети на языке EDEF. Такая формула может иметь следующий вид.

$$\text{Пр} = \text{Кд} \times (\text{Сд} - \text{См}) - \text{Кс} \times \text{Зс} \times \text{Тс} - \text{Сп} \times \text{Тп} ,$$

где:

Пр - прибыль;

Кд – количество собранных и обожженных в печи деталей;

Сд - стоимость готовой детали;  
 Кс – количество сборщиков;  
 Зс - зарплата сборщика за час;  
 Тс – время, в часах, работы сборщика (модельное время);  
 Сп- стоимость печи за 8-ми часовой рабочий день;  
 Тп - время, в днях, работы печи;  
 См- стоимость материала одной детали.

В синтаксисе языка EDEF раздел E: описательного файла для пяти сборщиков может иметь следующий вид:

```
E:  _summa=_isd*(7-2) -5*3.75*_hour-80*5;
    printf("прибыль = %d",_summa);
    printf("изготовлено деталей = %d",_isd);
```

где:

\_summa – системная переменная вычисляющая прибыль;

\_isd - количество собранных и обожженных в печи деталей (системная переменная);

выражение "(7-2)" - стоимость готовой детали за вычетом стоимости материала одной детали;

цифра "5" - количество сборщиков;

цифра "3,75" – зарплата сборщика за час;

\_hour - время, в часах, работы сборщика (системная переменная);

цифра "80" - стоимость печи за 8-ми часовой рабочий день;

цифра "5" - время, в днях, работы печи;

Для определения оптимального количества сборщиков, приносящих максимальную прибыль, необходимо выполнить моделирование при разном их числе. Например, при работе 4, 5 и 6 –ти сборщиков. Для каждого случая необходимо создать и проанализировать файл отчета. В файле отчета рекомендуется выводить не только значение прибыли, но значение изготовленных деталей. Вариант, при котором прибыль будет максимальной, и будет являться оптимальным.

## 8. ЛАБОРАТОРНАЯ РАБОТА №2

### Моделирование двухпроцессорной системы

**Цель работы:** Практическая работа с пакетом программ имитационного моделирования EVA на примере моделирования двухпроцессорной системы.

**Используемое программное обеспечение:** программа имитационного моделирования EVA.

### 8.1. Задание на лабораторную работу №2

Требуется промоделировать функционирование двухпроцессорной вычислительной системы с общей памятью, разделенной на  $n$  блоков. Каждой задаче отводится 1 блок. Интервалы времени между поступлениями задач распределены равномерно в диапазоне  $T1$  единиц времени. Время обработки порции информации подчинено случайному экспоненциальному закону с математическим ожиданием  $m1$  в процессоре №1 (CPU1) и с математическим ожиданием  $m2$  в процессоре №2 (CPU2). Между обработкой порций информации с вероятностью  $P$  возможно обращение к внешней памяти, в которой время обслуживания распределено равномерно в диапазоне  $T2$  единиц времени. С вероятностью  $1-P$  задачи оказываются решенными и покидают систему. Моделирование выполнить на отрезке времени, соответствующем решению не менее 100 задач.

Численные значения для различных вариантов приведены в таблице 8.1.

Таблица 8.1. Численные значения величин для различных вариантов задачи по моделированию двухпроцессорной системы

Номер варианта	$n$	$T1$	$T2$	$m1$	$m2$	$P$
1	1	2,10	2,8	2	5	0,6
2	2	2,11	2,7	1	6	0,5
3	3	2,12	2,6	3	7	0,4
4	4	2,13	2,5	4	8	0,3
5	5	2,14	2,4	2	9	0,4
6	6	2,15	2,3	1	8	0,5
7	7	2,16	2,2	3	7	0,8
8	8	2,17	2,1	4	6	0,7
9	9	2,18	2,0	5	5	0,6
10	10	2,19	2,1	3	4	0,5
11	11	2,20	2,2	2	5	0,6
12	12	2,21	2,3	1	6	0,7
13	13	2,22	2,4	4	7	0,8
14	14	2,23	2,5	3	8	0,7
15	15	2,24	2,6	5	9	0,6
16	16	2,25	2,7	6	8	0,4
17	17	2,26	2,8	4	7	0,5
18	18	2,27	2,9	3	6	0,6
19	19	2,28	2,10	2	5	0,7
20	20	2,29	2,11	3	4	0,8

Необходимо построить модель описанной двухпроцессорной системы с помощью аппарата Е-сетей, провести моделирование и собрать статистику по всем очередям.

## 8.2. Содержание отчета по лабораторной работе

Отчет по лабораторной работе должен содержать результаты работы программы имитационного моделирования EVA, включающего в себя следующие сведения.

1. Файл графического изображения моделируемой двухпроцессорной системы в формате графического редактора EEE (имеющий расширение EEE).
2. Текстовый файл, описывающий функционирование этой Е-сети, на языке EDEF (имеющий расширение DEF), с подробными комментариями.
3. Текстовый файл отчета со статистическими данными по всем очередям.
4. Исполняемый файл модели (имеющий расширение EXE).
5. Файл титульного листа к лабораторной работе.

## 8.3. Метод построения модели

Схема имитационной модели приведена на рисунке 8.1.

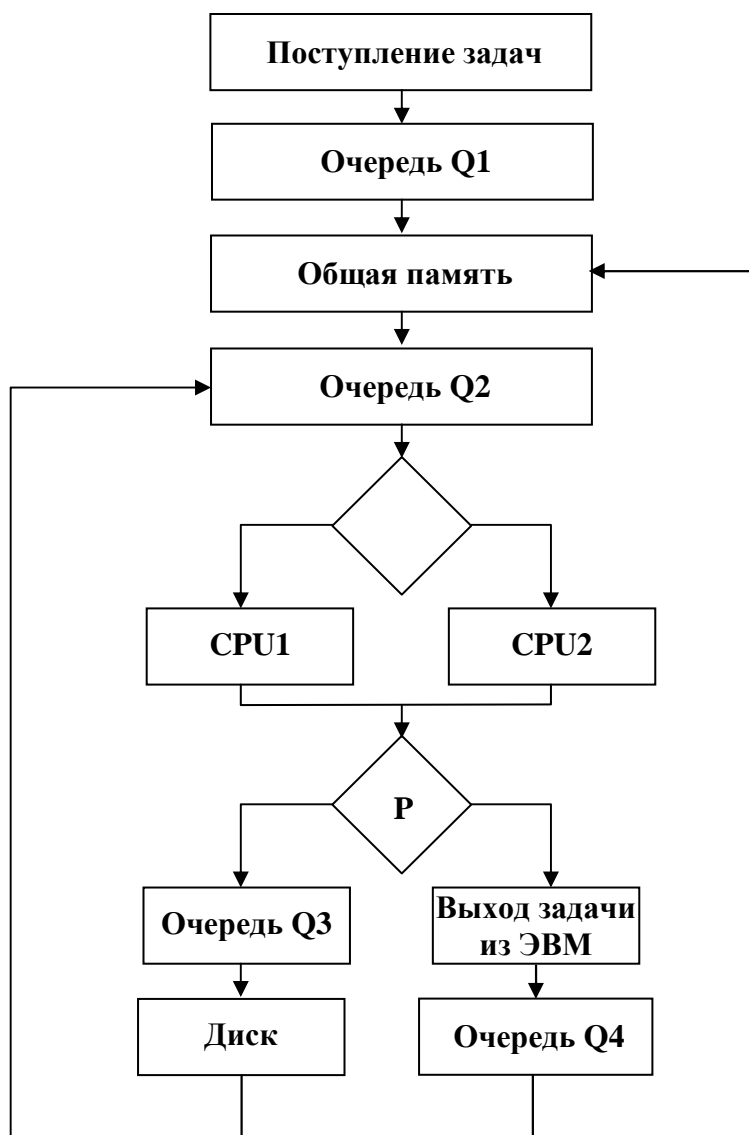


Рис.8.1.Схема имитационной модели  
двухпроцессорной системы

Особенностью моделируемой системы является то, что общая память разделена на  $n$  блоков. Это условие накладывает ограничение на количество одновременно находящихся в системе задач. Так, если количество блоков равно восьми, то это означает, что в системе могут одновременно обрабатываться не более восьми задач. Это условие можно реализовать с помощью перехода типа "объединение" с двумя входными позициями-очередями. В одну очередь должны приходить входные задачи, а во вторую – решенные задачи, которые покидают систему. Эта очередь (Q4) должна иметь предварительную маркировку в соответствии с числом блоков памяти  $n$  и быть включена в обратную связь системы. Начальная маркировка очереди  $n$  (в данном случае восемь) будет означать, что в системе может одновременно находиться не более восьми задач. По мере поступления в систему входных задач эта очередь будет уменьшаться. По мере поступления в нее решенных задач – увеличиваться. При наличии в системе восьми задач значение ее маркировки становится равным нулю, что приводит к нарушению условия срабатывания перехода.

Наличие на схеме второй обратной связи, заведенной от диска в очередь Q4 обусловлено требованием обращения к внешней памяти с вероятностью  $P$ .

## **9. КОНТРОЛЬНЫЕ РАБОТЫ №№ 3-4**

Контрольные работы №№ 3-4 – компьютерные. Перед их выполнением необходимо изучить разделы 7-8 учебного пособия "Разработка систем автоматизированного проектирования (САПР)", (Сети связи), часть 2. Контрольные работы содержат вопросы по указанным разделам.