

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Невинномысский технологический институт (филиал)

ПРАКТИКУМ
по дисциплине
«Распределенные компьютерные информационно-управляющие системы»
Методические указания к выполнению практических
работ для студентов направления

15.03.04 – «Автоматизация технологических
процессов и производств»
Направленность (профиль)
Информационно-управляющие системы

Невинномысск 2022

УДК 004.4

Автор-составитель:
канд. техн. наук, доцент Э.Е. Тихонов

«Распределенные компьютерные информационно-управляющие системы»: методические указания по выполнению практических работ для студентов направления 15.04.04 Автоматизация технологических процессов и производств, направленность (профиль) Информационно-управляющие системы / автор – составитель: Э.Е. Тихонов Невинномысск: НТИ (филиал) СКФУ, 2022. – 48 с.

Методические указания к выполнению практических работ и по освоению дисциплины «Распределенные компьютерные информационно-управляющие системы» предназначены для бакалавров заочной обучения по направления 15.03.04 Автоматизация технологических процессов и производств.

Методические указания составлены на основании федерального государственного образовательного стандарта высшего образования по направлению подготовки 15.03.04 Автоматизация технологических процессов и производств.

© НТИ, 2022
© Тихонов Э.Е., 2022

Содержание

| | |
|------------------------------------------------------------------------------------------------------------------------|----|
| ВВЕДЕНИЕ | 4 |
| ПРАКТИЧЕСКАЯ РАБОТА №1 «ИССЛЕДОВАНИЕ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ С ЛИНЕЙНОЙ АРХИТЕКТУРОЙ» | 6 |
| ПРАКТИЧЕСКАЯ РАБОТА №2 «ИССЛЕДОВАНИЕ РАСПРЕДЕЛЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ С КАНАЛАМИ МЕЖПРОЦЕССОРНОГО ОБМЕНА ДАННЫМИ» | 10 |
| ПРАКТИЧЕСКАЯ РАБОТА №3 «ИССЛЕДОВАНИЕ ТОПОЛОГИИ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ С ЗВЕЗДООБРАЗНОЙ АРХИТЕКТУРОЙ» | 15 |
| ПРАКТИЧЕСКАЯ РАБОТА №4 «УСТАНОВКА ГРИД-СИСТЕМ НА БАЗЕ GLOBUS TOOLKIT 5.0.3» | 21 |
| ПРАКТИЧЕСКАЯ РАБОТА №5 «ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ В ГЛОБАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ» | 31 |
| ПРАКТИЧЕСКАЯ РАБОТА №6 «ИССЛЕДОВАНИЕ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ С МАРШРУТИЗАТОРАМИ» | 33 |
| ПРАКТИЧЕСКАЯ РАБОТА №7 «ИССЛЕДОВАНИЕ ТОПОЛОГИИ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ С ПЕТЛЕВОЙ АРХИТЕКТУРОЙ» | 37 |
| ПРАКТИЧЕСКАЯ РАБОТА №8 «КОНФЛИКТЫ ПРИ КОНВЕЙЕРНОЙ ОБРАБОТКЕ ДАННЫХ В ПРОЦЕССОРЕ» | 41 |
| ПРАКТИЧЕСКАЯ РАБОТА №9 «ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА ТОМАСУЛО» | 45 |

Введение

Область распределенных вычислительных систем в настоящее время характеризуется быстрыми темпами изменения идеологий и подходов. За короткую историю существования систем такого типа появилось множество различных парадигм реализации распределенных вычислений, набравших большой вес и общее признание, но практически исчезнувших впоследствии под давлением более новых и модных подходов.

Однако когда технология исчезает из виду, очень часто она появляется вновь под новым именем. В результате происходит непрерывное перемешивание базовых концепций с новейшими подходами к разработке. В середине 1990-х существовало два основных подхода к разработке распределенных вычислительных систем. С одной стороны, концепция Веб представляла собой ориентированное на человека распределенное информационное пространство. С другой стороны, технологии распределенных объектов, такие как CORBA и DCOM были в первую очередь ориентированы на создание распределенных сред, которые эмулировали процесс разработки и исполнения локальных приложений, обеспечивая преимущества доступа к сетевым ресурсам. Но, несмотря на первоначальную идею Веб как пространства, которое позволяло многим людям обмениваться информацией, большинство пользователей просто потребляли информацию, не публикуя ничего взамен. Между тем системы распределенных объектов росли с точки зрения предоставляемых возможностей, но становились все более тяжелыми в плане разработки и использования. Сразу после начала нового тысячелетия произошел взрыв развития новых методов и промежуточного программного обеспечения для распределенных вычислительных систем, включая технологии одноранговых сетей (peer-to-peer или P2P) и грид-технологии.

Применение P2P позволило множеству пользователей, которые раньше были простыми потребителями информации, поучаствовать в предоставлении контента. С другой стороны, применение технологии грид позволило интегрировать крупные комплексы обработки и хранения данных, обеспечивая их доступность для различных правительственных и научных пользователей. Концепция грид-вычислений была ориентирована на построение инфраструктуры, обеспечивающей «вычисления по требованию», аналогично тому, как мы сейчас получаем доступ к коммунальным услугам, например, к электричеству.

Практическая работа №1

«Исследование распределенной системы с линейной архитектурой»

Цель работы. Изучение принципов работы распределенных вычислительных сетей на примере простейшей имитационной модели, а также оценка времени обслуживания и конфликтов в этих сетях.

В настоящее время широкое применение получили локальные вычислительные сети (ЛВС), которые строятся на ПЭВМ, соединенных линией связи и аппаратурой, поддерживающей соответствующий интерфейс. Информация в таких сетях передается в виде пакетов стандартной длины и формата. Структура пакета и методы его обработки регламентируются специальными протоколами. Если передаваемое сообщение имеет большую длину, то оно представляется как последовательность пакетов. Такая операция выполняется с использованием сетевого оборудования и его программного обеспечения.

В качестве канала чаще всего применяют высокочастотный телефонный кабель с пропускной способностью 40 1000 Кбод. Для подключения к нему используют сетевые адаптеры (контроллеры, сетевые карты), в состав которых могут быть включены модемы.

Наиболее распространенными структурами сети являются:

- *моноканал или шина (рис. 1.1), типичным примером которого служит Ethernet;
- *звезда (рис. 1.2), например Fast Ethernet;
- *петля (кольцо) рис. 1.3, классическим примером которой является сеть Token Ring.

В линейных сетях используется случайный метод доступа (захват канала в любой момент CSMA/CD). При этом возможно наложение пакетов, передаваемых с разных станций, т.е. искажение информации. Оно обнаруживается оборудованием сетевых адаптеров (СА). Случайный доступ реализуется двумя основными способами: с проверкой столкновений и несущей. Первый способ (с проверкой столкновений) только обнаруживает наложение передаваемого пакета на тот, который движется по каналу. При обнаружении столкновения передача прекращается, и следующая попытка предпринимается через случайный интервал времени. Этот способ применяется в слабо загруженных сетях. При проверке несущей адаптер прослушивает канал и начинает передачу только, если в ней отсутствует пакет (нет несущей частоты). Такой способ обеспечивает меньшую вероятность столкновения пакетов и лучшую загрузку линии связи.

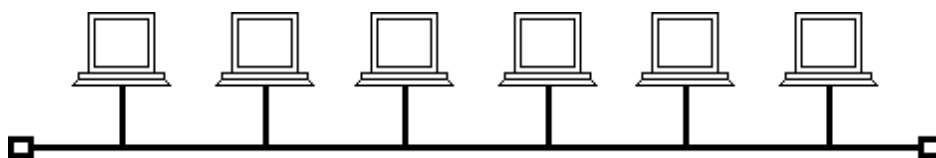


Рис. 1.1. Типовая структура линейной ЛВС

В настоящее время всё большее распространение получает звездообразная структура сети (рис. 1.2). Она наилучшим образом обеспечивает реализацию режима обработки задач «Клиент – сервер». В такой сети имеется мощная центральная ЭВМ, выполняющая функции файлового сервера и управляющая работой всей сети. Эта структура носит название активная звезда. Более простые сети вместо центральной ЭВМ используют концентраторы или коммутаторы. Их архитектура называется пассивной звездой.

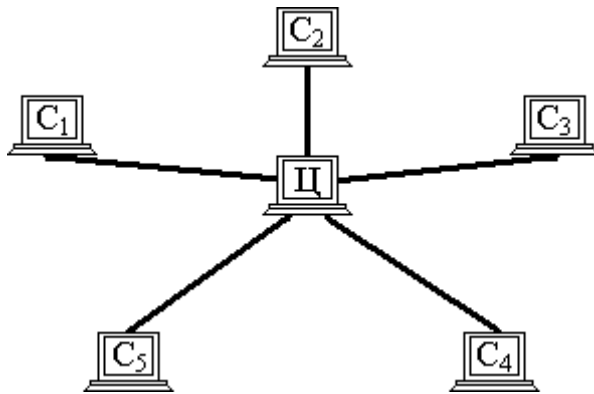


Рис. 1.2. Типовая структура звездообразной ЛВС : Ц – Центральная ЭВМ, С1-С5 – локальные станции

В кольцевых сетях используется детерминированный (маркерный, эстафетный) метод доступа. При этом сетевой адаптер получает разрешение на передачу пакета в момент поступления специального маркера. Обычно его роль играет пустой пакет. После получения заполненного пакета адаптер адресата освобождает его от информации и гонит по каналу.

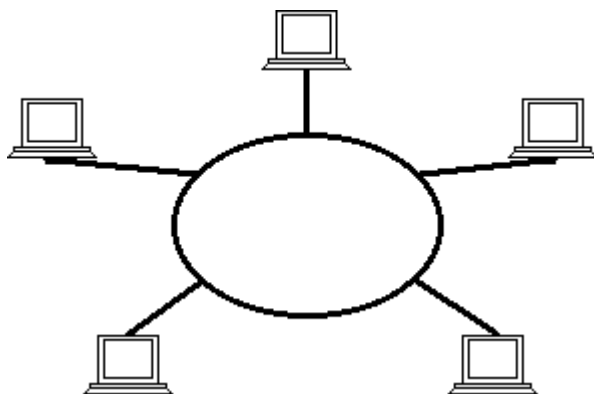


Рис. 1.3. Типовая структура кольцевой ЛВС

Описание практической установки

Лабораторная работа выполняется на ПЭВМ типа IBM, работающей в среде Windows. В ней используется программа имитационного моделирования наиболее распространенной сети, имеющей архитектуру типа моноканала (аналога Ethernet). Программа составлена на языке Delphi. Она позволяет выполнить следующие операции:

1. Варьировать количество и типы программ, обрабатываемых каждой станцией (от 1 до 20).
2. Изменять для всех программ одновременно или для каждой в отдельности
 - *количество и длину передаваемых пакетов, а также
 - *максимальное значение интервала времени между отправлениями двух соседних пакетов.
3. Варьировать пропускную способность линии связи от 40 до 1024 Кбайт.
4. Задать один из двух методов доступа к линии: с проверкой столкновений или с проверкой несущей.
5. Установить один из двух режимов моделирования: непрерывно или по тактам. По умолчанию моделирование осуществляется в непрерывном режиме.

Моделируемая сеть может содержать до 10 станций, соединенных между собой каналом связи. Работа каждой станции представляется закраской ее на схеме соответствующим цветом. Станция может находиться в состоянии обслуживания программы или передачи пакета. Пакеты передаются в канал в случайные моменты времени, которые определяются с помощью генератора случайных чисел. Максимальное значение интервала времени между отправлениями двух соседних пакетов, как отмечалось выше, является характеристикой соответствующей программы. Передача пакета в модели представляется закраской канала цветом ее станции.

Как отмечалось выше, важной особенностью и недостатком случайного метода доступа к каналу являются наложения (столкновения) пакетов. В модели предусмотрено обнаружение этой ситуации, прекращение передачи и возобновление ее через случайный промежуток времени. Количество столкновений подсчитывается программой и является одной из характеристик сети.

Исполняемый модуль программы моделирования имеет имя **Lan.exe** и находится в папке **Lab 9**. Запуск программы осуществляется традиционным способом. После запуска на экране появляется первое окно, позволяющее задать параметры модели и режим моделирования.

В левой части окна находится панель управления приложением. На ней размещено несколько кнопок, каждая из которых снабжена всплывающей подсказкой. Кнопка с изображением раскрытой книги переводит приложение в режим ввода характеристик или моделирования. После запуска установлен первый режим.

После того как ввод всех параметров закончен нужно переключиться на режим моделирования. Запуск модели осуществляется нажатием на кнопку с изображением зеленой лампочки. При этом будет поддерживаться непрерывный режим, а кнопка утоплена. В любой момент можно остановить моделирование повторным нажатием на ту же кнопку, а затем – снова продолжить. Имеется возможность исследовать работу сети в каждом такте, определяемом ее новым состоянием (началом или окончанием передачи очередного пакета). Выполнение такта начинается после нажатия мышью кнопки с изображением красной лампочки.

Во время моделирования появляется окно приложения, в верхней части которого изображена структура сети (станции и линия связи), а в нижней представлена трасса загрузки канала и станций. Каждой станции, как отмечалось выше, соответствует свой цвет. В нижней части экрана выводится столько трасс (временных диаграмм), сколько рабочих станций считалось активными в моделируемой сети. При возникновении конфликта через все диаграммы проводится вертикальная линия белого цвета.

В средней части экрана в процессе моделирования в полях «Прошло времени», «Количество конфликтов», «Среднее время выполнения» выводятся основные характеристики сети. После каждого запуска модели программа запоминает значение среднего времени выполнения задач и строит график этого времени, который располагается в рамке, находящейся в правой половине средней части экрана. Для его получения нужно запустить программу, по меньшей мере, дважды.

Порядок выполнения практической работы

Подготовка к работе

1. Знакомство со всеми разделами руководства.
2. Получение у преподавателя задания на исследование сети с различными параметрами устройств, потока задач и режимов их обслуживания.
3. Исследование заданной сети.
4. Оформление отчета.

Последовательность выполнения практической работы

В практической работе необходимо провести следующие исследования.

1. Меняя **количество программ**, обрабатываемых сетью и станциями, **при одинаковых параметрах** самих программ определить времена их пребывания в сети и среднее время выполнения одной программы, а также количество столкновений для каждого из двух способов доступа.
2. Задавая **разные параметры программ**, обрабатываемых сетью и станциями, **при постоянном общем количестве программ**, определить времена их пребывания в сети и среднее время выполнения одной программы, а также количество столкновений для каждого из двух способов доступа.
3. Варьируя **параметры станций и линии связи** при **постоянных параметрах программ** и их **количестве**, определить времена их пребывания в сети и среднее время выполнения одной программы, а также количество столкновений для каждого из двух способов доступа.
4. Построить графики зависимости времени и количества столкновений от изменяемых параметров.

Содержание отчета о выполненной работе

Отчет должен содержать следующее:

1. Название и цель работы.

2. Исходные данные.

3. Графики следующих зависимостей для каждого из двух способов доступа к линии связи:

- 1) общего и среднего времени выполнения программ от количества программ;
- 2) количества столкновений от числа программ;
- 3) общего и среднего времени выполнения программ от длины пакета и максимального интервала времени между последовательными моментами отправления двух пакетов;
- 4) количества столкновений от длины пакета и максимального интервала времени между последовательными моментами отправления двух пакетов;
- 5) общего и среднего времени выполнения программ от заданного преподавателем параметра одной или нескольких станций или линии связи;
- 6) количества столкновений от заданного преподавателем параметра одной или нескольких станций или линии связи.

Контрольные вопросы

1. Назовите основные типы структур локальных вычислительных сетей.
2. Какие методы доступа используются в сетях?
3. Как обнаруживается наложение пакетов при передаче и что делается в этой ситуации?
4. Как влияет количество задач, обрабатываемых одной станцией, на характеристики сети?
5. Как влияет общее количество задач, обрабатываемых сетью, на ее характеристики?
6. Как влияет на характеристики сети пропускная способность канала?
7. Как влияют характеристики задач, обрабатываемых одной станцией, на характеристики сети в целом?
8. Какое влияние на работу сети оказывают общие характеристики задач, обрабатываемых станциями?
9. Как влияет на характеристики сети длина пакета сообщения?
10. Как влияет на характеристики сети максимальный интервал времени между последовательными моментами отправления двух пакетов?
11. При каком методе доступа наложения пакетов наиболее вероятны?
12. Какой из исследованных вами режимов работы сети можно считать лучшим и почему?

Практическая работа №2

«Исследование распределенной вычислительной системы с каналами межпроцессорного обмена данными»

Цель работы. Изучение на простейшей имитационной модели особенностей работы одноранговых вычислительных сетей при наличии помех в канале.

В настоящее время, как отмечалось в руководстве к работе 1, широкое применение получили локальные вычислительные сети (ЛВС), которые строятся на ПЭВМ, соединенных линией связи и аппаратурой, поддерживающей соответствующий интерфейс. Информация в таких сетях передается в виде пакетов стандартной длины и формата. Структура пакета и методы его обработки регламентируются специальными протоколами. Такая операция выполняется с использованием сетевого оборудования и его программного обеспечения. В качестве канала чаще всего применяют высокочастотный телефонный кабель с пропускной способностью 40 1000 Кбод.

Одной из наиболее распространенных структур локальных сетей является моноканал или шина (рис. 1.1) [1]. В таких сетях используется случайный метод доступа. При этом возможно наложение пакетов, передаваемых с разных станций, т.е. искажение информации. Случайный доступ, как отмечалось в руководстве к практической работе 4, реализуется двумя основными способами: с проверкой столкновений и несущей. Первый из них самый простой и применяется в слабо загруженных сетях, а второй обеспечивает меньшую вероятность столкновения пакетов и лучшую загрузку линии связи.

Передача информации по каналу занимает определенную часть времени ее обработки. Она может быть прервана из-за появления помех в линии. Для сетей любого уровня и архитектуры характерны как единичные, так и продолжительные помехи. При их обнаружении выполняется повторная передача очередного пакета. Если помехи длительные, то таких попыток может быть несколько. Их количество обычно ограничивается разумными пределами (например, не более 15 попыток). При достижении такого предела задача снимается (работа станции блокируется).

Описание практической установки

Лабораторная работа выполняется на ПЭВМ типа IBM, работающей в среде Windows. В ней используется программа имитационного моделирования, аналогичная той, которая была в работе 7. Программа составлена на языке Delphi. Она позволяет выполнить те же операции, что и в предыдущей работе, а именно:

1. Варьировать количество и типы программ, обрабатываемых каждой станцией (от 1 до 20).
2. Изменять для всех программ одновременно или для каждой в отдельности:
 - количество и длину передаваемых пакетов, а также
 - максимальное значение интервала времени между отправлениями двух соседних пакетов.
3. Варьировать пропускную способность линии связи от 1024 до 10240 Кбайт, а также задавать меньшие значения, начиная от 1.
4. Задать один из двух методов доступа к линии: с проверкой столкновений или с проверкой несущей.
5. Установить один из двух режимов моделирования: непрерывно или по тактам. По умолчанию моделирование осуществляется в непрерывном режиме.

Кроме того, имеется дополнительная возможность задавать **параметры помех** в канале:

- максимальную длительность и
- максимальное количество попыток повторной передачи пакета при искажении.

В программе предусмотрена стандартная реакция на продолжительные помехи: если после заданного числа попыток пакет так и не передан, то соответствующая задача снимается.

Моделируемая сеть может содержать до 10 станций, соединенных между собой каналом связи. Так же, как и в предыдущей работе, функционирование каждой станции представляется закраской ее на схеме соответствующим цветом. Станция работает по тому же алгоритму. Передача пакета в модели представляется закраской канала цветом ее станции. Возникновение конфликта отображается на временной диаграмме работы станций вертикальной линией пурпурного цвета. Помехи также представляются на временной диаграмме группой серых линий, количество которых соответствует длительности помех.

Исполняемый модуль программы моделирования имеет имя **Lan.exe** и находится в папке **Lab 10**. Запуск программы осуществляется традиционным способом. После запуска на экране появляется первое окно, позволяющее задать параметры модели и режим моделирования.

Слева в окне расположена панель управления приложением. На ней находятся несколько кнопок, каждая из которых снабжена всплывающими подсказками. Кнопка с изображением раскрытой книги переводит приложение в режим ввода характеристик или моделирования. После запуска установлен первый.

Все поле ввода параметров разделено на пять основных блоков.

1. «Типы задач» обеспечивает задание до двадцати типов задач, каждый из которых характеризуется числом пакетов, их размером, максимальным интервалом времени между двумя соседними пакетами и максимальным числом попыток передачи пакета при помехах.

2. «Помехи» обеспечивает задание максимального интервала времени между помехами и их максимальной продолжительности. Блок «Линия связи» позволяет ввести пропускную способность линии. Она может изменяться в пределах от 1024 до 10240 Кбайт в секунду.

3. «Характеристики станции», в котором можно задать как для отдельной станции, так и для всех одновременно их характеристики. Характеристиками станций являются количество, типы задач и порядок их обслуживания. Для их ввода необходимо выполнить те же действия, что и в работе 7, т.е. выбрать тип задачи из соответствующего блока, а затем нажать кнопку пересылки. Указанная кнопка находится между блоками. На ней изображена стрелка, направленная вправо. Задачи, которые приписаны к станции, изображаются в ее списке в том порядке, в котором они будут впоследствии обрабатываться. Этот список можно очистить или удалить из него какую-либо запись с помощью соответствующих кнопок (название кнопки можно увидеть во всплывающей подсказке).

4. Станции, как отмечалось выше, могут работать в двух режимах: с проверкой столкновений и с проверкой несущей. Эти

характеристики можно задать в последнем блоке: «Дисциплина доступа станций к линии связи».

После того как ввод всех параметров закончен, нужно переключиться в режим моделирования. *Запуск модели* осуществляется последовательным **нажатием на кнопки с изображением книги** (настройки параметров), а затем — **зеленой лампочки**. При этом будет поддерживаться непрерывный режим и кнопка западет. В любой момент можно остановить модель повторным нажатием на зеленую кнопку, а затем снова продолжить. Генерацию каждого следующего такта можно выполнить вручную нажатием на кнопку с изображением красной лампочки.

При имитации работы станций и канала считается, что интервалы между пакетами в задачах, а также интервалы между помехами и длительность самих помех являются случайными величинами, диапазон изменения которых определяется исходными данными [2].

Во время моделирования появляется окно приложения, в верхней части которого изображена структура сети (станции и линия связи), а в нижней представлены временные диаграммы (трассы) загрузки канала и станций. При этом количество трасс соответствует числу активных рабочих станций. Каждой станции, как отмечалось выше, соответствует свой цвет.

В средней части окна в полях: «Прошло времени», «Общее число наложений помех на пакеты», «Количество снятых задач» представлены основные характеристики. По окончании работы программа запоминает значение общего времени работы сети. При нескольких последовательных запусках приложения в центральной части экрана строится график изменения этого времени.

Порядок выполнения практической работы

Подготовка к работе

1. Знакомство со всеми разделами руководства.
2. Получение у преподавателя задания на исследование сети с помехами в канале.
3. Исследование заданной сети.
4. Оформление отчета.

Последовательность выполнения практической работы

В практической работе необходимо провести следующие исследования.

1. Варьируя **длительность помех** при **постоянных параметрах станций** и программ и фиксированном количестве тех и других, определить характеристики сети для каждого из двух способов доступа.

2. Варьируя **интервалы между помехами** при **постоянных параметрах станций** и программ и фиксированном количестве тех и других, определить характеристики сети для каждого из двух способов доступа.

3. Задавая разное **количество программ**, обрабатываемых сетью и станциями, **при одинаковых параметрах** самих программ и варьируя параметры помех определить те же характеристики, что и в п.п.1 и 2.

4. Определить те же характеристики, что и в п.п.1 и 2, для разных параметров программ при постоянном общем их количестве и изменяющихся параметрах помех.

5. Построить графики зависимости времени, количества столкновений и числа снятых задач

от параметров помех для разных вариантов параметров задач, выбранных при выполнении п.п. 2 — 4.

Содержание отчета о выполненной работе

Отчет должен содержать следующее:

1. Название и цель работы.
2. Исходные данные.
3. Графики зависимостей для каждого из двух способов доступа к линии связи и вариантов исследованных характеристик задач:
 - a) общего и среднего времени выполнения программ при одинаковых характеристиках задач и станций от длительности помех;
 - b) общего и среднего времени выполнения программ при одинаковых характеристиках задач и станций от длительности интервалов между помехами;
 - c) общего и среднего времени выполнения программ при одинаковых характеристиках задач и станций и нескольких вариантах длин пакетов и максимальных интервалов между последовательными отправлениями пакетов от длительности интервалов между помехами;
 - d) общего и среднего времени выполнения программ при одинаковых характеристиках задач и станций, нескольких вариантах длин пакетов и интервалов между последовательными отправлениями пакетов от длительности помех;
 - e) общего и среднего времени выполнения программ при одинаковых характеристиках задач и станций, нескольких вариантах длин пакетов и максимальных интервалов между последовательными отправлениями пакетов от длительности интервалов между помехами;
 - f) количества снятых задач при одинаковых характеристиках их и станций от длительности помех;
 - g) количества снятых задач при одинаковых характеристиках их и станций от длительности интервалов между помехами;
 - h) количества снятых задач при нескольких вариантах их характеристик и станций от длительности помех;
 - i) количества снятых задач при нескольких вариантах их характеристик и станций от длительности интервалов между помехами.

Контрольные вопросы

1. Назовите основные типы структур локальных вычислительных сетей и области их применения.
2. Какие методы доступа используются в сетях?
3. Как обнаруживается наложение пакетов при передаче и что предпринимается в этой ситуации?
4. Какие виды помех возникают в канале сети?

5. Какова стандартная реакция на обнаружение помехи в канале сети?
6. Как влияет длительность помех в канале на характеристики сети?
7. Как влияет на характеристики сети длительность интервалов между помехами в канале?
8. Как влияет на характеристики сети допустимое количество повторных передач пакетов при помехах в канале?
9. При каком методе доступа к каналу влияние помех наиболее ощутимо?
10. Как влияет на характеристики сети пропускная способность канала при наличии помех в нем?
11. Как влияет на характеристики сети при помехах длина пакета сообщения?
12. Как влияет на характеристики сети при помехах максимальный интервал времени между последовательными моментами отправления двух пакетов?

Практическая работа №3

«Исследование топологии распределенной системы с звездообразной архитектурой»

Цель работы. Изучение на простейшей имитационной модели особенностей работы вычислительных сетей со звездообразной архитектурой.

«Звезда» — это топология с явно выделенным центром, к которому подключаются все остальные абоненты (см. рис. 3.1). Обмен информацией происходит исключительно через центральный компьютер, на который, таким образом, ложится очень большая нагрузка. Поэтому никакими другими операциями, кроме сетевых, он заниматься не может.

Конфликты в сети с топологией «звезда» в принципе невозможны, так как управление ею полностью централизовано [3].

Если говорить об устойчивости звезды к отказам, то выход из строя периферийного компьютера никак не отражается на функционировании оставшейся части сети. В то же время отказ центральной ЭВМ делает сеть полностью неработоспособной. Обрыв любого кабеля или короткое замыкание в нем нарушает обмен только с одним компьютером, а все остальные могут нормально продолжать работу.

В отличие от шины, в звезде на каждой линии связи находятся только два абонента: центральный и один из периферийных. Чаще всего для их соединения используется две линии связи, каждая из которых передает информацию только в одном направлении. Недостаток топологии «звезда» состоит в ограничении количества абонентов. Обычно центральная ЭВМ может обслуживать не более 8-16 периферийных абонентов.

Звезда, показанная на рис. 3.1, носит название активной, или истинной. Существует также топология, называемая пассивной звездой (рис. 3.2). В настоящее время она распространена гораздо больше, чем первая. Достаточно сказать, что она используется в самой популярной на сегодняшний день архитектуре сети Ethernet. В центре такой сети помещается не компьютер, а концентратор, или хаб (hub). Он восстанавливает входящие сигналы и пересылает их в другие линии связи. Хотя схема прокладки кабелей подобна истинной или активной звезде, фактически имеет место шинная топология, так как информация от каждого компьютера одновременно передается ко всем остальным, а центрального абонента не существует. Естественно, пассивная звезда получается дороже обычной

шины, так как она содержит еще и концентратор. Однако такая архитектура предоставляет целый ряд дополнительных возможностей, связанных с преимуществами звезды. Именно поэтому в последнее время пассивная звезда все больше вытесняет истинную шину, которая считается малоперспективной.

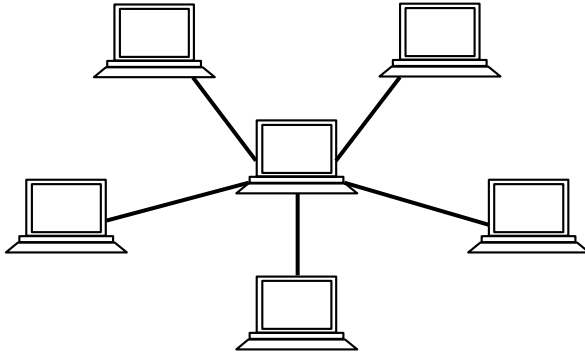


Рис. 3.1. Сетевая топология «звезда»

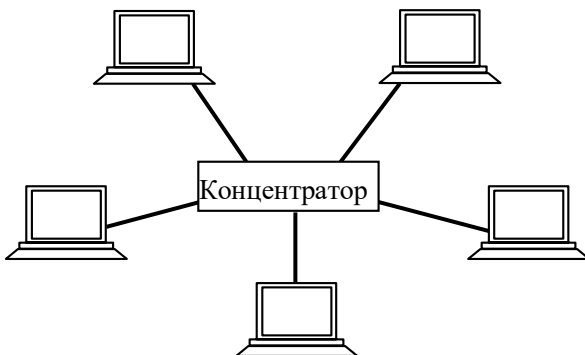


Рис. 3.2. Топология «пассивная звезда»

Существует также промежуточный тип топологии. В нем вместо концентратора используется переключатель (switch), который не только ретранслирует поступающие на него сигналы, но и управляет обменом, хотя сам в обмене не участвует.

Большое достоинство звезды (как активной, так и пассивной) состоит в том, что все точки подключения ее собраны в одном месте. Это позволяет легко контролировать работу сети, локализовать неисправности путем простого отключения от центра тех или иных абонентов (что невозможно, например, в случае шины), а также ограничивать доступ посторонних лиц к жизненно важным для сети точкам подключения.

Общим недостатком для всех топологий типа «звезда» является значительно больший, чем у других архитектур, расход кабеля. Это существенно влияет на стоимость всей сети в целом.

Классическим примером современных сетей, имеющих звездообразную архитектуру, является сеть 100VG AnyLAN (рис. 3.3). Она отличается от Ethernet следующими показателями:

- использует другой метод доступа Demand Priority (приоритетный), который обеспечивает более справедливое распределение пропускной способности сети по сравнению со случайным доступом;
- кадры передаются не всем станциям, а только станции назначения;
- в сети есть выделенный арбитр доступа – концентратор, и это заметно отличает рассматриваемую технологию от других, в которых применяется распределенный между станциями сети алгоритм доступа;
- в AnyLAN поддерживаются кадры двух технологий – Ethernet и Token Ring.

Сеть состоит из центрального концентратора, называемого также корневым, и соединенных с ним конечных узлов и других концентраторов. Концентратор циклически выполняет опрос портов. Станция, желающая передать пакет, посылает ему специальный низкочастотный сигнал, запрашивая разрешение на передачу кадра и указывая его приоритет. В сети 100VG AnyLAN используются два уровня приоритетов – низкий и высокий. Низкий уровень соответствует обычным данным (файловая служба, служба печати и т.п.), а высокий приоритет присваивается данным, чувствительным к временным задержкам (например, мультимедиа). Приоритеты запросов имеют статическую и динамическую составляющую. Это позволяет станциям с низким уровнем приоритета, которые долго не имеют доступ к сети, получить высокий приоритет и быть обслуженной за конечное время [5].

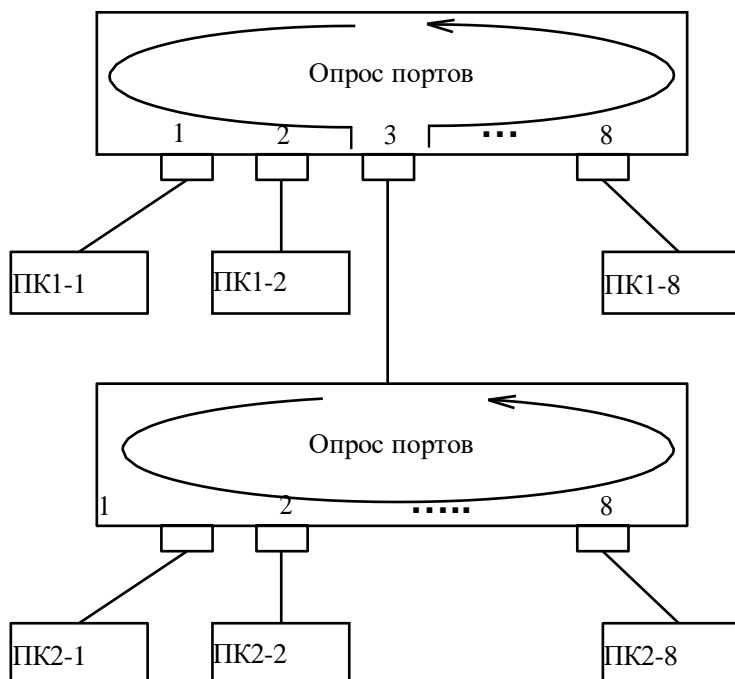


Рис. 3.3 Архитектура сети 100VG AnyLAN

Если концентратор нижнего уровня свободен, то он разрешает передачу пакета. В противном

случае запрос на передачу ставится в очередь, которая обрабатывается в соответствии с порядком его поступления и приоритетом. Концентратор верхнего уровня (корневой) работает аналогично. При передаче через него пакетов в нижний сегмент опрос приостанавливается до завершения работы концентратора нижнего уровня. Станции, подключенные к концентраторам различного уровня, не имеют преимуществ по доступу к разделяемой среде, так как решение о предоставлении доступа принимается после проведения опроса всеми концентраторами своих портов.

Концентратор определяет, к какому порту подключена станция назначения следующим образом. Вначале в момент физического присоединения станции к сети он распознает ее MAC адрес и запоминает его в соответствующей таблице. Аналогично работает мост/коммутатор. Отличие концентратора 100VG AnyLAN от моста/коммутатора в том, что у него нет внутреннего буфера для хранения кадров. Поэтому он прини-

мает от станции только один кадр, отправляет его на порт назначения и, пока этот кадр не будет полностью принят станцией назначения, новые кадры не принимаются. Таким образом сохраняется эффект разделяемой среды и улучшается безопасность сети, т.к. кадры не попадают на чужие порты, и их труднее перехватить.

Описание практической установки

Лабораторная работа выполняется на ПЭВМ типа IBM, работающей в среде Windows. В ней используется программа имитационного моделирования, составленная на языке Delphi. Она позволяет выполнить следующие операции.

1. Варьировать количество активных станций (от 2 до 8).
2. Задавать для каждой станции список адресатов, которым она передает сообщения.
3. Варьировать количество и типы программ, обрабатываемых каждой станцией (от 1 до 20).
4. Изменять для всех программ одновременно или для каждой в отдельности:
 - количество и длину передаваемых пакетов, а также
 - максимальное значение интервала времени между отправлениями двух соседних пакетов.
5. Варьировать пропускную способность линии связи от 3000 до 1024000 Байт/с.
6. Установить один из двух режимов моделирования: непрерывно или по тактам. По умолчанию моделирование осуществляется в непрерывном режиме.

В практической работе исследуется простейшая архитектура типа

«пассивная звезда», приведенная на рис. 3.2. Центром сети является концентратор (hub), имеющий до 8 входов. Концентратор циклически выполняет опрос портов. Станции, желающие передать свой пакет, посылают ему свои запросы. Если хаб свободен, то он разрешает передачу

пакета, в противном случае полученный запрос ставится в очередь и обрабатывается в соответствии с порядком поступления. Таким образом, концентратор принимает от станции только один кадр, отправляет его в порт назначения и, пока он не будет полностью передан, новые пакеты не принимаются и не отсылаются.

Исполняемый модуль программы моделирования имеет имя **Star.exe** и находится в папке **Lab 11**. Запуск программы осуществляется традиционным способом. После запуска на экране появляется первое окно, позволяющее задать параметры модели и режим моделирования.

В левой части окна находится панель управления, снабженная функциональными кнопками, как в модели одноранговой локальной сети.

Все поле для ввода параметров разделено на три основных блока.

1. «Типы задач» обеспечивает задание до двадцати типов задач, каждый из которых характеризуется числом пакетов, их размером и максимальным интервалом времени между двумя соседними пакетами. При имитации работы сети считается, что интервалы между пакетами в задачах являются случайными величинами, диапазон изменения которых определяется исходными данными.

2. «Линия связи» позволяет ввести пропускную способность линии, которая может изменяться в пределах от 40 до 1024000 байт в секунду. В верхней части приложения вводится количество активных станций.

3. «Характеристики станции» обеспечивает ввод количества, типов задач, порядка их обслуживания и станций-адресатов в виде списка. Задачи, которые приписаны к станции, изображаются в том порядке, в котором они будут передаваться адресатам из списка. Списки можно очистить или удалить из них какую-либо запись с помощью соответствующих кнопок (название кнопки можно увидеть во всплывающей подсказке).

После того как ввод всех параметров закончен, нужно переключиться в режим моделирования. Запуск модели осуществляется нажатием на кнопку с изображением зеленой лампочки. Генерацию каждого следующего такта можно выполнить вручную нажатием на кнопку с изображением красной лампочки.

Во время моделирования появляется окно приложения, в верхней части которого изображены сами станции, Hub и линии связи, а в нижней — представлена временная диаграмма передачи пакетов. Каждой станции соответствует свой цвет. Вертикальные цветные линии отмечают поступление в концентратор запросов на передачу пакета в момент его занятости. В этом случае соответствующая станция ставится в очередь, и начинается отсчет времени ее ожидания. В верхнем правом углу выводятся сообщения об операциях, выполняемых в процессе моделирования.

В средней части представлены основные характеристики в полях:

«Прошло времени», «Количество задержанных передач», «Время ожидания Hub» и «Среднее время ожидания». В конце работы программа запоминает значение среднего времени ожидания, и при нескольких последовательных запусках в центральной части экрана строится его график.

Порядок выполнения практической работы

Подготовка к работе

1. Знакомство со всеми разделами руководства.
2. Получение у преподавателя задания на исследование сети с различными параметрами

устройств, потока задач и режимов их обслуживания.

3. Исследование заданной сети.
4. Оформление отчета.

Последовательность выполнения практической работы

В практической работе необходимо провести следующие исследования.

1. Задавая **разное количество программ**, обрабатываемых станциями, при **одинаковых параметрах** самих программ и фиксированном количестве адресатов каждой станции (например, 3) определить общее время их пребывания в сети, а также среднее время ожидания и количество задержанных пакетов.

2. Выполнить исследования, как предложено в п.1, для **количеств адресатов**, меняющихся в пределах от 4 до 7.

3. Задавая **разные параметры программ**, обрабатываемых станциями, при **постоянном общем количестве программ** и **фиксированном количестве адресатов** каждой станции (например, 3), определить общее время их пребывания в сети, а также среднее время ожидания и количество задержанных пакетов.

4. Выполнить исследования, как предложено в п.3, для **количеств адресатов**, меняющихся в пределах от 4 до 7.

5. Варьируя **пропускную способность линий связи** при **постоянных параметрах программ** и их количестве, а также **фиксированном количестве адресатов** каждой станции (например, 3), определить общее время их пребывания в сети, а также среднее время ожидания и количество задержанных пакетов.

6. Выполнить исследования, как предложено в п.5, для **количеств адресатов**, меняющихся в пределах от 4 до 7.

7. Построить графики зависимости времени пребывания и ожидания задач, а также количества задержанных пакетов от изменяемых в п.п. 1 — 6 параметров.

Содержание отчета о выполненной работе

Отчет должен содержать следующее:

1. Название и цель работы.
2. Исходные данные.
3. Графики зависимостей времени пребывания и ожидания задач, а также количества задержанных пакетов от изменяемых параметров для вариантов, перечисленных в разделе «Порядок выполнения работы».

Контрольные вопросы

1. Назовите основные типы структур локальных вычислительных сетей и области их применения.

2. Назовите основные виды структур сетей «звездообразной» архитектуры.
3. Какие методы доступа используются в сетях?
4. Какие достоинства и недостатки имеют сети с архитектурой типа «звезда»?
5. Как реализуется передача пакетов в сети с архитектурой типа активная или пассивная «звезда»?
6. Как влияет на характеристики сети количество задач, обслуживаемых станциями?
7. Как влияет на характеристики сети количество пакетов, передаваемых станциями?
8. Как влияет на характеристики сети количество адресатов, с которыми работает каждая станция?
9. Как влияет на характеристики сети длина пакета сообщения?
10. Как влияет на характеристики сети пропускная способность каналов?
11. Используется ли в моделируемой сети приоритетное обслуживание пакетов?

Практическая работа №4 «Установка грид-систем на базе Globus Toolkit 5.0.3»

Цель работы: Рассмотреть как устанавливаются грид-система на базе Globus Toolkit 5.0.3

1 Настройка первой машины

1.1. Перед установкой

Итак, вы решили установить Globus Toolkit 5.0, и не знаете с чего начать. А начать следует с **подготовки всего необходимого** для развертывания глобуса (я позволю себе здесь и далее так называть Globus Toolkit 5).

Для начала нам необходимы два реальных или (что для начала даже лучше) виртуальных компьютера. Для создания виртуальных машин прекрасно подойдет бесплатная программа [VirtualBox](#). В качестве операционной системы я выбрал [Debian 6 Squeeze](#). Данное руководство рассматривает установку на виртуальных машинах, но аналогично настраиваются и реальные компьютеры.

Важно

При установке операционной системы выберите в качестве domainname что-либо, отличное от localdomain, иначе будут проблемы при установке глобуса. В данной инструкции рассматривается пример с именем домена «WORKGROUP»

Совет

В Debian установку пакетов можно совершить, запустив в консоли из-под root'a приложение aptitude или synaptic. При необходимости добавьте репозиторий 'deb-src http://ftp.ru.debian.org/debian squeeze main'

Проверим уже установленные пакеты

```
root@GlobusNode:~$ openssl version
OpenSSL 0.9.8o 01 Jun 2010
```

```
root@GlobusNode:~$ dpkg --get-architecture | grep libssl
ii libssl-dev 0.9.8o-4squeeze1 SSL development libraries,
header files and documentation
ii libssl0.9.8 0.9.8o-4squeeze1 SSL shared libraries
```

```
root@GlobusNode:~$ dpkg --list | grep zlib
ii  zlib1g          1:1.2.3.4.dfsg-3      compression library - runtime
ii  zlib1g-dev     1:1.2.3.4.dfsg-3      compression library -
development
```

(аналогично как и с SSL libraries, zlib development может потребовать установки)

Важно чтобы для Zlib и Libssl присутствовали development библиотеки (а не только shared). Их, скорее всего, потребуется доустановить. Openssl подойдет, начиная с версии 0.9.7.

Также необходимо установить gcc и g++. Проверить корректность установки можно с помощью следующих команд:

```
root@GlobusNode:~$ which gcc
/usr/bin/gcc
root@GlobusNode:~$ which g++
/usr/bin/g++
```

Также проверим версии (и в случае необходимости, установим) приложений tar/make/sed:

```
root@GlobusNode:~$ tar --version
tar (GNU tar) 1.23
Copyright (C) 2010 Free Software Foundation, Inc.
Лицензия GPLv3+: GNU GPL версии 3 или новее <http://gnu.org/licenses/gpl.html>
Это свободное ПО: вы можете продавать и распространять его.
Нет НИКАКИХ ГАРАНТИЙ до степени, разрешённой законом.
```

```
root@GlobusNode:~$ sed --version
GNU sed версия 4.2.1
Copyright (C) 2009 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE,
to the extent permitted by law.
```

```
root@GlobusNode:~$ make --version
GNU Make 3.81
Copyright (C) 2006 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.
```

Также крайне желательно установить текстовый редактор **vim** ([руководство по vim для новичков](#)) и **xinetd**, с помощью которого в дальнейшем будут запускаться сервисы. Также потребуется установить **ssh** для передачи данных между машинами.

Итак, еще раз, полный список того, что должно быть установлено:

```
openssl
libssl (dev+shared libs)
zlib (dev+shared libs)
gcc
g++
tar
sed
make
vim
xinetd
ssh
```

[Системные требования](#) в [полном руководстве по установке](#) (англ).

1.2. Сборка Globus Toolkit

Итак, все необходимое установлено, приступаем к сборке самого глобуса. Для начала необходимо создать пользователя **globus**, под которым будет установлен Globus Toolkit. Теперь нужно [скачать инсталлятор](#) (архив .tar.bz2, 53 Мб) и поместить его куда-нибудь (например, в каталог пользователя, т.е. /home/globus/). Установка пакета под root'ом может повлечь за собой дальнейшие проблемы при эксплуатации GRAM5. После создания пользователя необходимо дать ему права на запись в папку /sandbox/

```
root@GlobusNode:/# mkdir /sandbox/
root@GlobusNode:/# chown globus /sandbox/
```

Создадим каталог, куда будет установлен Globus Toolkit.

```
globus@GlobusNode:~$ mkdir /sandbox/globus-5/
```

Распакуем архив:

```
globus@GlobusNode:~$ tar xjf gt5.0.3-all-source-installer.tar.gz
globus@GlobusNode:~$ cd gt5.0.3-all-source-installer
```

Сконфигурируем файл установки, указав путь, куда будет проинсталлирован глобус, а именно /sandbox/globus-5/

```
globus@GlobusNode:/gt5.0.3-all-source-installer$ ./configure --
prefix=/sandbox/globus-5/
```

```
checking build system type... i686-pc-linux-gnu
configure: creating ./config.status
config.status: creating Makefile
```

Теперь можно приступить к самой сборке:

```
globus@GlobusNode:~/gt5.0.3-all-source-installer$ make | tee installer.log
cd gpt-3.2autotools2004 && OBJECT_MODE=32 ./build_gpt
build_gpt ====> installing GPT into /sandbox/globus-5/
...
```

Можно идти пить чай с пряниками, сборка займет около часа (в зависимости от вашего компьютера):

```
...
Your build completed successfully. Please run make install.
```

```
globus@GlobusNode:~/gt5.0.3-all-source-installer$ make install
/sandbox/globus/globus-5.0.3//sbin/gpt-postinstall
...
..Done
```

Теперь для удобства дальнейшей работы добавим подключение команд Globus Toolkit в файл /etc/bash.bashrc. Таким образом мы избавим себя от обязанности подключать команды при открытии нового терминала и команды глобуса будут подключаться автоматически:

```
root@GlobusNode:/#vim /etc/bash.bashrc
```

Добавим в начало файла следующие две строки:

```
export GLOBUS_LOCATION=/sandbox/globus-5
source $GLOBUS_LOCATION/etc/globus-user-env.sh
```

Сохраняем, подключаем заново файл /etc/bash.bashrc и перезапускаем терминал.

```
root@GlobusNode:/#source /etc/bash.bashrc
```

Теперь наша машина готова для дальнейшей настройки. Настоятельно рекомендуется сделать копию виртуальной машины в данный момент времени, и из этой копии уже потом будет настроена вторая (и последующие машины).

1.3. Настройка безопасности на первой машине

На данном этапе будет настроена аутентификация и авторизация пользователей с помощью сертификатов. Потребуется сертификаты как для пользователей, так и для сервисов. Для сервисов они будут совпадать с сертификатами хоста. Для создания сертификатов будет использоваться **Simple CA**, которая поставляется вместе с Globus Toolkit. Подробная инструкция может быть найдена тут [SimpleCA Admin](#).

Прежде чем приступить к настройке сертификатов необходимо проверить файл /etc/hosts.

```
root@GlobusNode:/# vim /etc/hosts

127.0.0.1      GlobusNode    localhost
127.0.1.1      GlobusNode.WORKGROUP  GlobusNode.localdomain
.....
```

Главное – чтобы после 127.0.0.1 стояло GlobusNode без указаний на рабочую группу, иначе в дальнейшем может возникнуть конфликт имен. Вторая строка менее важна, но лучше привести ее к подобному виду. Можно приступить к установке и настройке **Simple CA**.

```
root@GlobusNode:~# cd /gt5.0.3-all-source-installer/quickstart
root@GlobusNode:gt5.0.3-all-source-installer/quickstart# perl gt-server-ca.pl -y
Setting up /sandbox/globus-5/
Please enter a password of at least four characters for the CA: <CApassword>
Confirm password: <CApassword>
Creating a new simpleCA, logging to gt-server-ca.log...
Running setup-gsi...
Your CA hash is: d58f5223
It is located at /sandbox/globus-5//share/certificates/d58f5223.0
```

В этот момент выполнение кода приостановится. Нужно будет ввести 'y' и нажать Enter для продолжения. В этот момент возникать должен запрос на подтверждение создания SimpleCA, но из-за ошибки в скрипте вопрос не выводится.

```
y
Your host DN is /O=Grid/OU=GlobusTest/OU=simpleCA-globusnode/CN=host/globusnode
The hostcert is located at /sandbox/globus-5/etc/hostcert.pem
```



Примечание

Может не сработать, если /tmp смонтирована как poehes. В таком случае попробуйте задать переменную `GLOBUS_SH_TMP=`pwd`` и попробовать снова.

Проверяем:

```
root@GlobusNode:~# ls /root/.globus/
simpleCA
root@GlobusNode:~# ls /root/.globus/simpleCA/
cacert.pem certs crl globus_simple_ca_d58f5223_setup-0.20.tar.gz grid-ca-ssl.conf
index.txt index.txt.attr index.txt.old newcerts private serial serial.old
```

Описание, что это за файлы, можно найти тут [Security Admin Guide](#). SimpleCA установлена.

Последним шагом на данном этапе будет копирование подписанных и доверенных сертификатов в /etc:

```
root@GlobusNode:~# mkdir /etc/grid-security
root@GlobusNode:~# mv $GLOBUS_LOCATION/etc/host*.pem /etc/grid-security/
root@GlobusNode:~# cp -r $GLOBUS_LOCATION/share/certificates/ /etc/grid-security/
```

1.4. Создание MyProxy сервера

На данном этапе будет создан и сконфигурирован MyProxy сервер на GlobusNode. Он используется для хранения и выдачи сертификатов пользователей и сервисов. Полная инструкция может быть найдена тут: [настройка MyProxy](#).

```
root@GlobusNode:~# cp $GLOBUS_LOCATION/share/myproxy/myproxy-server.config /etc
root@GlobusNode:~# vim /etc/myproxy-server.config
```

Необходимо раскомментировать строки 18-24. Сравним полученный файл с исходным.

```
root@GlobusNode:~# diff /etc/myproxy-server.config
$GLOBUS_LOCATION/share/myproxy/myproxy-server.config
18,24c18,24
< accepted_credentials "*"
< authorized_retrievers "*"
< default_retrievers "*"
< authorized_renewers "*"
< default_renewers "none"
< authorized_key_retrievers "*"
< default_key_retrievers "none"
---
> #accepted_credentials "*"
> #authorized_retrievers "*"
> #default_retrievers "*"
> #authorized_renewers "*"
> #default_renewers "none"
> #authorized_key_retrievers "*"
> #default_key_retrievers "none"
root@GlobusNode:~# cat $GLOBUS_LOCATION/share/myproxy/etc.services.modifications >>
/etc/services
root@GlobusNode:~# tail /etc/services
binkp          24554/tcp      # binkp fidonet protocol
asp            27374/tcp      # Address Search Protocol
asp            27374/udp
dircproxy      57000/tcp      # Detachable IRC Proxy
tfido          60177/tcp      # fidonet EMSI over telnet
fido           60179/tcp      # fidonet EMSI over TCP
# Local services
myproxy-server 7512/tcp      # Myproxy server
```



```
root@GlobusNode:~# cp $GLOBUS_LOCATION/share/myproxy/etc.xinetd.myproxy
/etc/xinetd.d/myproxy
```

Исправляем файл с настройками сервиса.

```
root@GlobusNode:~# vim /etc/xinetd.d/myproxy
service myproxy-server
{
  socket_type = stream
  protocol   = tcp
  wait       = no
  user       = root
  server     = /sandbox/globus-5/sbin/myproxy-server
  env        = GLOBUS_LOCATION=/sandbox/globus-5
LD_LIBRARY_PATH=/sandbox/globus-5/lib
  disable    = no
}
```

Перезапускаем xinetd и проверяем работоспособность сервиса.

```
root@GlobusNode:~# /etc/init.d/xinetd reload
Reloading internet superserver configuration: xinetd.
root@GlobusNode:~# netstat -an | grep 7512
tcp        0      0 0.0.0.0:7512          0.0.0.0:*            LISTEN
```

MyProxy сервер запущен и работает. Теперь необходимо создать сертификат для тестового пользователя. Создаем в системе реального пользователя с полным именем *Quick Start User* и логином *quser*. При создании сертификата будет необходимо указать два пароля. Первый - это пароль, который будет в дальнейшем использоваться для авторизации пользователя *quser* (минимум 6 символов), а второй – пароль, который был указан при создании **Simple CA** для доступа к ней.

```
root@GlobusNode:~ # myproxy-admin-adduser -c "QuickStart User" -l quser
```

```
Enter PEM pass phrase: <придумываем пароль, например qpassword>
Verifying - Enter PEM pass phrase: <qpassword>
```

```
To sign the request
please enter the password for the CA key: <пароль от SimpleCA>
```

```
The new signed certificate is at: /root/.globus/simpleCA//newcerts/02.pem
```

```
using storage directory /var/myproxy
Credential stored successfully
Certificate subject is:
/O=Grid/OU=GlobusTest/OU=simpleCA-globusnode/CN=QuickStart User
```

И, последний шаг – создаем файл `grid-mapfile` под `root` который будет использоваться для авторизации. Копируем целиком строку `/O=Grid/OU=...` из предыдущего вывода:

```
root@GlobusNode:/etc/grid-security# vim /etc/grid-security/grid-mapfile
"/O=Grid/OU=GlobusTest/OU=simpleCA-globusnode/CN=QuickStart User" quser
```



Замечание

Для пользователя `globus` сертификат создавать не нужно. Это просто аккаунт для установки `Globus Toolkit`. Сертификаты нужны только реальным пользователям.

Теперь, когда уже настроена аутентификация, можно приступать к сервисам. Сначала установим GridFTP – сервис для передачи данных по протоколу gsiftp. Полная инструкция [GridFTP Admin Guide](#).

Создаем файл `/etc/xinetd.d/gridftp` и редактируем его.

```
root@GlobusNode:/etc/grid-security# vim /etc/xinetd.d/gridftp
service gsiftp
{
instances                = 100
socket_type              = stream
wait                     = no
user                     = root
env                      += GLOBUS_LOCATION=/sandbox/globus-5
env                      += LD_LIBRARY_PATH=/sandbox/globus-5/lib
server                   = /sandbox/globus-5/sbin/globus-gridftp-server
server_args              = -i
log_on_success           += DURATION
disable                  = no
}
```

Добавляем в конец файла `/etc/services` информацию о новом сервисе – gsiftp.

```
root@GlobusNode:/etc/grid-security# vim /etc/services

vboxd                    20012/udp
binkp                    24554/tcp                # binkp fidonet protocol
asp                      27374/tcp                # Address Search Protocol
asp                      27374/udp
dircproxy                57000/tcp                # Detachable IRC Proxy
tfido                    60177/tcp                # fidonet EMSI over telnet
fido                     60179/tcp                # fidonet EMSI over TCP

# Local services
myproxy-server          7512/tcp                # Myproxy server
gsiftp                  2811/tcp
```

Перезапускаем xinetd и проверяем работоспособность сервиса.

```
root@GlobusNode:/etc/grid-security# /etc/init.d/xinetd reload
Reloading internet superserver configuration: xinetd.
root@GlobusNode:/etc/grid-security# netstat -an | grep 2811
tcp        0      0 0.0.0.0:2811          0.0.0.0:*              LISTEN
```

1.6. Настройка GRAM5

Приступим к установке GRAM5 – менеджера вычислительных ресурсов. Можно настроить несколько различных Локальных Менеджеров Ресурсов (LRM) с которыми будет работать GRAM, но в этом руководстве будет рассмотрен лишь простой "fork" jobmanager, отвечающий за запуск задачи на локальной машине. Информация о других LRM и более подробные инструкции могут быть найдены тут [GRAM5 Admin Guide](#).

Создаем файл `/etc/xinetd.d/gsigatekeeper` и редактируем его.

```
root@GlobusNode:/etc/grid-security# vim /etc/xinetd.d/gsigatekeeper
service gsigatekeeper
{
    socket_type = stream
    protocol = tcp
    wait = no
```

```

user = root
env = LD_LIBRARY_PATH=/sandbox/globus-5/lib
server = /sandbox/globus-5/sbin/globus-gatekeeper
server_args = -conf /sandbox/globus-5/etc/globus-gatekeeper.conf
disable = no
}

```

Добавляем в конец файла `/etc/services` информацию о новом сервисе - `gsigatekeeper`.

```

root@GlobusNode:/etc/grid-security# vim /etc/services
vboxd          20012/udp
binkp          24554/tcp          # binkp fidonet protocol
asp            27374/tcp          # Address Search Protocol
asp            27374/udp
dircproxy      57000/tcp          # Detachable IRC Proxy
tfido          60177/tcp          # fidonet EMSI over telnet
fido           60179/tcp          # fidonet EMSI over TCP

# Local services
myproxy-server 7512/tcp          # Myproxy server
gsiftp         2811/tcp
gsigatekeeper  2119/tcp

```

Перезапускаем `xinetd` и проверяем работоспособность сервиса.

```

root@GlobusNode:/etc/grid-security# /etc/init.d/xinetd reload
Reloading internet superserver configuration: xinetd.
root@GlobusNode:/etc/grid-security# netstat -an | grep 2119
tcp        0      0 0.0.0.0:2119          0.0.0.0:*            LISTEN

```

`Gatekeeper` теперь автоматически запускается с `xinetd`, и сконфигурирован, чтобы принимать и перенаправлять задачи `fork jobmanager`'у. Протестируем работоспособность:

```

quser@GlobusNode:~$ myproxy-logon -s globusnode
Enter MyProxy pass phrase:
A credential has been received for user quser in /tmp/x509up_u1001.
quser@GlobusNode:~$ globusrun -o -r globusnode '&(executable=/bin/hostname) '
GlobusNode
quser@GlobusNode:~$ globusrun -o -r globusnode '&(executable=/usr/bin/whoami) '
quser

```

2. Настройка второй машины

2.1. Настройка системы

Сделаем копию ранее скопированной на шаге 1.2 виртуальной машины с уже установленным `Globus Toolkit`'ом, и настроим эту машину как клиента, с которого будем передавать задачи на сервер.

На этой машине также необходимо создать пользователя *QuickStart User* с логином `quser`. Затем, в настройках сети необходимо поменять сетевое имя машины на `GlobusNode2`. Соответственно должен быть переписан файл `/etc/hosts`.

Помимо этих изменений на обеих машинах следует внести в файл `/etc/hosts` IP-адреса другой машины, чтобы они могли найти друг друга. Узнать IP можно с помощью команды `ifconfig`

```

root@GlobusNode2:/# vim /etc/hosts

```

```
127.0.0.1      GlobusNode2 localhost
127.0.1.1      GlobusNode2.WORKGROUP  GlobusNode2.localdomain
.....
192.168.231.147 Globusnode
```

Проверим, что машины видят друг друга:

```
root@GlobusNode:/# ping globusnode2
root@GlobusNode2:/# ping globusnode
```

2.3. Настройка безопасности на второй машине

Настройка безопасности на второй машине сводится к копированию созданных SimpleCA файлов. Необходимо скопировать папку `$GLOBUS_LOCATION/share/certificates` с первой машины на вторую:

```
globus@GlobusNode2:~$ scp -r globusnode:/sandbox/globus-5/share/certificates
/sandbox/globus-5/share
The authenticity of host 'globusnode (192.168.231.147)' can't be established.
RSA key fingerprint is cc:ca:74:65:34:86:6a:43:f8:d7:f5:0b:b1:02:35:34.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'globusnode,192.168.231.147' (RSA) to the list of known
hosts.
globus@globusnode's password:
d58f5223.signing_policy          100% 1337 1.3KB/s 00:00
globus-host-ssl.conf.d58f5223  100% 2709 2.7KB/s 00:00
d58f5223.0                      100%  924 0.9KB/s 00:00
grid-security.conf.d58f5223     100% 1324 1.3KB/s 00:00
globus-user-ssl.conf.d58f5223  100% 2809 2.7KB/s 00:00
```

Теперь создадим на первой машине (GlobusNode) сертификат для второй машины:

```
root@GlobusNode:/home/globus# myproxy-admin-addservice -c "globusnode2" -l quser
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

```
To sign the request
please enter the password for the CA key:
```

```
The new signed certificate is at: /root/.globus/simpleCA//newcerts/03.pem
```

```
using storage directory /var/myproxy
Credential stored successfully
Certificate subject is:
/O=Grid/OU=GlobusTest/OU=simpleCA-globusnode/CN=globusnode2
```

```
root@GlobusNode2:/home/globus# myproxy-retrieve -s globusnode -k "globusnode2" -l
quser
```

```
Enter MyProxy pass phrase:
Credentials for quser have been stored in
/etc/grid-security/hostcert.pem and
/etc/grid-security/hostkey.pem.
```

```
root@GlobusNode2:/home/globus# ls -l /etc/grid-security/*.pem
```

```
-rw----- 1 root root 843 Май 31 14:30 /etc/grid-security/hostcert.pem
-rw----- 1 root root 887 Май 31 14:30 /etc/grid-security/hostkey.pem
```

```
root@GlobusNode2:/home/globus# myproxy-destroy -s globusnode -k "globusnode2" -l
quser
```

```
MyProxy credential 'globusnode2' for user quser was successfully removed.
```

Теперь у второй машины есть все необходимые сертификаты. Последний шаг – создать **grid-mapfile** и вписать в него то же, что и на первой машины. Это необходимо для корректной авторизации машин:

```
root@GlobusNode2:~# vim /etc/grid-security/grid-mapfile
"/O=Grid/OU=GlobusTest/OU=simpleCA-globusnode/CN=QuickStart User" quser
```

2.3. Настройка второй машины: GridFTP

Установка GridFTP идентична установке на первой машине, описанной в пункте 1.5. Вот команды, которые необходимо выполнить:

```
root@GlobusNode2:~# vim /etc/xinetd.d/gridftp
root@GlobusNode2:~# vim /etc/services
root@GlobusNode2:~# /etc/init.d/xinetd reload
Reloading internet superserver configuration: xinetd.
```

2.4. Настройка второй машины: GRAM5

Настраивать GRAM5 не надо – он уже должен работать корректно. Протестируем:

```
quser@GlobusNode2:~$ myproxy-logon -s globusnode
Enter MyProxy pass phrase:
A credential has been received for user quser in /tmp/x509up_u1001.
quser@GlobusNode2:~$ globus-job-submit globusnode /bin/date
https://globusnode:38129/16145810981669093481/18380602028524064396/
quser@GlobusNode2:~$ globus-job-get-output
https://globusnode:38129/16145810981669093481/18380602028524064396/
Tue May 31 14:33:25 MSD 2011

quser@GlobusNode2:~$ globus-job-run globusnode /usr/bin/whoami
quser

quser@GlobusNode2:~$ globusrun -s -r globusnode '&(executable=/bin/hostname) '
GlobusNode

quser@GlobusNode2:~$ globusrun -s -r globusnode
'&(executable=gsiftp://globusnode2/bin/hostname) '
GlobusNode

quser@GlobusNode2:~$ vim a.rsl
&(executable=gsiftp://globusnode2/bin/hostname)

quser@GlobusNode2:~$ globusrun -f a.rsl -s -r globusnode
GlobusNode
```

Обратите внимание, в чем разница в двух последних примерах. В первом запускается на выполнение файл, находящийся на сервере, а во втором – файл находящийся на машине-клиенте.

Синтаксис и примеры работы с GRAM можно найти тут [GRAM usage scenarios](#), либо в помощи по каждой команде соответственно (например, `globusrun -help`)

Еще примеры команд для добавления задач можно посмотреть тут: [GRAM5 User Guide](#)

Практическая работа №5 **«Облачные вычисления в глобальной вычислительной сети»**

Цель работы: изучение технологии предоставления информационных услуг в виде высоко масштабируемых сервисов (Облачные вычисления –Cloud Computing).

Облачные вычисления относятся к технологиям метакomпьютинговой среды [1].

Ресурсы облака:

Аппаратная часть в дата-центрах:

- Серверы
- Системы хранения данных
- Сетевое оборудование

Программное обеспечение:

- Нативные ОС и гостевые (эмулируемые) ОС
- Программные средства управления серверами и сетевым оборудованием
- Компиляторы, трансляторы, интерпретаторы
- Среды кодирования, отладки и размещения приложений в сети
- Прикладные программы общего назначения (офисное ПО, СУБД и др.)

Возможности облаков:

- Все хранится, обновляется и контролируется в облаке
- Автоматическая антивирусная защита облачных ресурсов
- Практически безразмерное хранилище данных для приложений
- Автоматическое перераспределение вычислительных ресурсов
- Взаимная совместимость различных операционных систем
- Коллективный доступ пользователей к документам и другим данным
- Повышенная надежность хранения данных и программ
- Виртуализация аппаратных и программных средств
- Доступность ресурсов облака с любого устройства, подключенного к Internet.

Виды облаков.

1. Частное облако (Private cloud).

Облако обслуживает одну организацию. Его инфраструктура может управляться самой организацией или третьей стороной и может существовать как на стороне потребителя, так и у внешнего провайдера.

2. Публичное облако (Public cloud).

Облако доступно для всех или для большой группы потребителей, принадлежащих к одной области деятельности. Такое облако принадлежит организации, реализующей соответствующие облачные услуги и предоставляющей облачные сервисы.

3. Гибридное облако (Hybrid cloud).

Облако является композицией частных и общих облаков, остающихся уникальными

сущностями, но объединенными вместе возможностью портирования данных и приложений между собой.

Виды облачных сервисов.

1. Инфраструктура как сервис – Infrastructure as a Service (IaaS).

Предоставление аппаратных средств, ОС и системного ПО на время их фактического использования, за которое производится оплата. IaaS основана на технологии виртуализации серверов и ПО. а также автоматическом масштабировании и автобалансировке (Amazon Web Service и др.).

2. Платформа как сервис – Platform as a Service (PaaS).

Предоставление для разработчиков приложений программной платформы, средств создания, отладки, тестирования и публикации web-приложений в Internet (Google App Engine, Widows Azure). Используя такую платформу, программисты потребителя сами разрабатывают в этой среде собственные программы и размещают их в облаке. Такой сервис нужен для разработки и развертывания собственного облачного программного обеспечения. Сервис Google App Engine (GAE) позволяет в его среде создавать и размещать в нем собственные приложения пользователя на языках Java, Python и Go.

3. Программное обеспечение как сервис – Software as a Service (SaaS).

Предоставление конечному пользователю готовых прикладных программ с оплатой по факту их использования (Google Apps, Microsoft Office 365 и др.).

Потребителю предоставляются программные средства приложения провайдера, выполняемые на облачной инфраструктуре. Приложения доступны с различных клиентских устройств через интерфейс тонкого клиента, такой как браузер (например, электронная почта с web-интерфейсом). Облачными ресурсами могут пользоваться не только настольные компьютеры или ноутбуки, но и любые мобильные устройства, подключаемые к интернету. Это могут быть планшетики, таблетки, айфоны, айпады и другие устройства.

Потребитель не управляет и не контролирует саму облачную инфраструктуру, на которой выполняется приложение, будь то сети, серверы, операционные системы, системы хранения или даже некоторые специфичные для приложений возможности. В ряде случаев, потребителю может быть предоставлена возможность доступа к некоторым пользовательским конфигурационным настройкам.

Наибольшие возможности и удобства предоставляет сервис Google Apps, в который входит множество приложений, часть из которых бесплатна, например:

- Gmail (mail.google.com) электронная почта,
- Документы (docs.google.com) набор приложений, где можно создавать и работать с текстовыми, табличными и графическими документами, создавать свои HTML-формы, создавать каталожную систему в виде коллекций и размещать созданные документы;
- Сайты (sites.google.com) где можно создавать и размещать в облаке свои собственные сайты, доступные в Интернет.

Порядок выполнения работы

1. Получение у Google своего аккаунта, идентификатора приложения и его регистрация на GAE. Для получения общего аккаунта Google нужно использовать

свой адрес email и придумать какой-нибудь пароль. Получение или изменение аккаунта Google можно сделать на сайте <https://www.google.com/accounts/manageaccount> [6].

2. Создание простейшего приложения “Hello World”.
3. Создание класса Java-сервлета и файлов конфигурации приложения.
4. Запуск и тестирование приложения вне облака Google.
5. Использование Java-серверных страниц JSP для создания клиентских Web-страниц.
6. Использование хранилища данных с JDO (Java Data Object)
7. Использование объектов, запросов, транзакций и ключей при работе с хранилищем данных DataStore.
8. Использование технологии AJAX и вызовов удаленных процедур (RPC) для обновления Web-страниц.

Содержание отчета о выполненной работе

Отчет должен содержать следующее:

- 1) описание действий по работе с облаком при выполнении заданной преподавателем программы;
- 2) описание полученных результатов.
- 3) таблицу временных ресурсов, затраченных на облачные вычисления по заданной программе.

Контрольные вопросы

1. Какие типы облаков вы знаете?
2. Перечислите возможности облаков при обеспечении облачных вычислений.
3. Где выполняются приложения пользователей в частном облаке?
4. Где выполняются приложения пользователей в публичном облаке?
5. Как обеспечивается надежность хранения данных?
6. Что такое облачный сервис?
7. Какие функции выполняет Google App Engine?
8. Можно ли выполнять в облаке программы, ориентированные на различные ОС?
9. Как осуществляется отладка и тестирование программ пользователей?
10. Что такое дата-центр?
11. Перечислите преимущества выполнения приложений в облаке по сравнению с использованием персонального компьютера.

Практическая работа №6 «Исследование распределенной системы с маршрутизаторами»

Цель работы. Изучение на простейшей имитационной модели особенностей работы

вычислительных сетей с произвольной архитектурой.

Маршрутизатор – это универсальный компьютер или специальное устройство, работающие в сети и выполняющие специфическую функцию соединения двух или более её частей. Маршрутизаторы ретранслируют пакеты из одной части сети в другую (из одного ее сегмента в другой). Они поддерживают сети с множеством возможных путей передачи информации, так называемые ячеистые сети. Размер таких сетей практически ничем не ограничен. При этом для обеспечения надёжности связи используются альтернативные дублирующие пути распространения информации. Именно маршрутизаторы чаще всего применяются для связи локальных сетей с глобальными, в частности, с Internet, которая может рассматриваться как полностью маршрутизируемая сеть

[1,4,5]. При объединении маршрутизаторов между собой образуется так называемое маршрутизируемое облако или коммуникационная сеть (рис. 5.1). Модель такой сети и исследуется в работе. При этом оценивается влияние изменения маршрутов (разрывов или включения части связей) на время доставки пакетов.

Описание практической установки

Лабораторная работа выполняется на ПЭВМ типа IBM, работающей в среде Windows. В ней используется программа имитационного моделирования, составленная на языке Delphi. Она позволяет выполнить следующие операции.

1. Задавать различные пути маршрутизации, т. е. связи между станциями и маршрутизаторами, которые устанавливаются визуально.
2. Варьировать для каждой станции количество пакетов и номера адресатов.
3. Установить один из двух режимов моделирования: непрерывно или по тактам. По умолчанию моделирование осуществляется в непрерывном режиме.

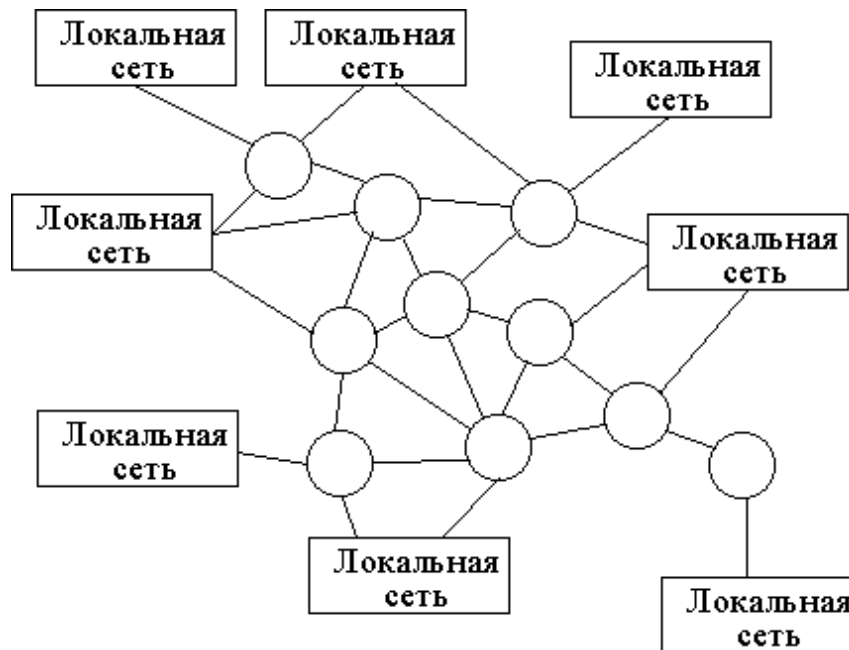


Рис. 5.1. Маршрутизируемое облако

Исполняемый модуль программы моделирования имеет имя **Marshrut.exe** и находится в папке **Lab 13**. Запуск программы осуществляется традиционным способом. После запуска на экране появляется окно, позволяющее задать параметры модели и режим моделирования.

В левом верхнем углу окна устанавливается флаг создания всех связей. Он обеспечивает получение полносвязной решетчатой структуры, у которой маршрутизируемое облако имеет вид прямоугольной решетки. В дальнейшем эту структуру можно изменять, удаляя связи с помощью мыши. Таким образом, можно получить самые разнообразные структуры коммуникационных сетей.

Например, если необходимо удалить связь между двумя маршрутизаторами, то сначала курсор помещается между ними, а затем одинарным нажатием кнопки мыши удаляется соединение. Добавление связи осуществляется аналогично.

После задания структуры сети выполняется ввод исходных данных для станций. При этом в поле «номер станции» устанавливается номер подсети, свойства которой задаются в рамке характеристики. Затем вводятся значения в два следующих поля:

- Количество пакетов (по умолчанию 1). Максимальное значение ничем не ограничено.
- Номер адресата (по умолчанию 0). Максимальное значение 16. Если номер совпадает с номером станции, то пакет никуда не отправляется.

Запуск модели происходит после нажатия кнопки «Пуск». Все результаты выводятся в многострочное поле. Основными характеристиками являются общее время работы, а также загрузка сети, которые отображаются на форме. Маршрутизаторы разделены на 4 части, каждая из которых может быть закрашена одним из 16 цветов, соответствующих станциям. Закраска показывает, пакетом какой станции занят буфер узла в данный момент.

После завершения процесса моделирования на экран выводится соответствующее сообщение.

Порядок выполнения практической работы

Подготовка к работе

1. Знакомство со всеми разделами руководства.
2. Получение у преподавателя задания на исследование сети с различными параметрами устройств и потока задач.
3. Исследование заданной сети.
4. Оформление отчета.

Последовательность выполнения практической работы

В практической работе необходимо провести следующие исследования.

1. Для **полносвязной структуры** сети и нескольких пар «Исходная станция — адресат» получить характеристики при количестве передаваемых пакетов, изменяющемся от 3 до 10.
2. Выполнить исследования, аналогичные п. 1, для **линейной архитектуры** (убрать лишние связи так, чтобы между исследуемыми парами узлов был один путь).
3. Выполнить исследования, аналогичные п. 1, для **петлевой архитектуры** (задать связи так, чтобы между исследуемыми парами узлов было замкнутое кольцо).
4. Выполнить исследования, аналогичные п. 1, для нескольких **произвольных архитектур** (по заданию преподавателя).
5. Построить графики зависимости полученных характеристик сети от исследованных параметров.

Содержание отчета о выполненной работе

Отчет должен содержать следующее:

4. Название и цель работы.
5. Исходные данные.
6. Графики зависимостей временных характеристик сети для вариантов, перечисленных в разделе «Порядок выполнения работы».

Контрольные вопросы

2. Назовите основные типы вычислительных сетей и области их применения.
3. Что такое маршрутизатор?
4. Что такое маршрутизируемое облако?
5. Как влияет на характеристики сети структура маршрутизируемого облака?
6. Какая из исследованных Вами структур обеспечивает минимальное, а какая максимальное время доставки пакетов?
7. Как влияет на характеристики сети количество пакетов, передаваемых станциями?
8. Имеются ли в модели ограничения на количество пакетов, принимаемых узлами коммуникационной сети?

9. В каких сетях применяются архитектуры, исследованные Вами в практической работе?
10. Что представляют собой пронумерованные узлы, расположенные по периферии коммуникационной сети?

Практическая работа №7

«Исследование топологии распределенной системы с петлевой архитектурой»

Цель работы. Изучение на простейшей имитационной модели особенностей работы вычислительных сетей с петлевой архитектурой.

Петля (кольцо) – это топология, в которой каждый компьютер соединён линиями связи только с двумя другими: от одного он получает информацию, а другому передаёт. На каждой линии связи работает один передатчик и один приёмник. При этом каждый компьютер ретранслирует (восстанавливает) проходящий к нему сигнал, то есть выступает в роли репитера. Чётко выделенного центра в данном случае нет, все компьютеры могут быть одинаковыми. Однако довольно часто в кольце выделяется специальный абонент, который управляет обменом или контролирует его [5].

Строго говоря, компьютеры в сети не являются полностью равноправными (в отличие, например, от шинной топологии). Одни из них получают информацию от источника раньше, а другие позже. Именно на этой особенности топологии и строятся методы управления обменом по сети типа «кольцо»: право на следующую передачу переходит последовательно к соседнему по кругу сети компьютеру.

Подключение новых абонентов в «кольцо» обычно осуществляется совершенно безболезненно, хотя и требует обязательной остановки всей сети на это время. Максимальное количество абонентов может быть довольно велико (до тысячи и больше). Кольцевая топология обычно является самой устойчивой к перегрузкам. Она обеспечивает уверенную работу с большими потоками информации, так как в ней, как правило, нет конфликтов. Сигнал в кольце проходит через все компьютеры сети, поэтому выход из строя хотя бы одного из них нарушает работу сети в целом. К такому же результату приводят обрыв или короткое замыкание в любом из кабелей. Кольцо наиболее уязвимо к повреждениям кабеля, поэтому в нём предусматривают прокладку двух или более параллельных линий связи, одна из которых является резервной. Указанное обстоятельство является одним из существенных недостатков рассматриваемой топологии.

В кольцевых сетях наиболее популярны маркерные методы управления, которые используют небольшой управляющий пакет специаль-

ного вида (маркер) [2]. Именно эстафетная передача маркера по кольцу позволяет передавать право на захват сети от одного абонента к другому. Маркерные методы относятся к децентрализованным и детерминированным методам управления обменом в сети. В них нет явно выраженного центра, но существует чёткая система приоритетов и потому не бывает конфликтов. Они гораздо эффективнее случайных при большой интенсивности обмена в сети и позволяют работать с большой нагрузкой, которая теоретически может приближаться к 100 %.

Описание практической установки

Лабораторная работа выполняется на ПЭВМ типа IBM, работающей в среде Windows. В ней используется программа имитационного моделирования, составленная на языке Delphi. Она позволяет выполнить следующие операции.

1. Варьировать количество активных станций.
2. Задавать для каждой станции список адресатов, которым она передает сообщения.
3. Варьировать количество пакетов, передаваемых каждой станцией.
4. Изменять для всех станций максимальное значение интервала времени между отправлениями двух соседних пакетов.
5. Варьировать пропускную способность линии связи от 1024 до 1024000 Бод (эта величина для наглядности изображения передачи пакетов всегда кратна длине пакета 1024 бит).
6. Установить один из двух режимов моделирования: непрерывно или по тактам. По умолчанию моделирование осуществляется в непрерывном режиме.

В практической работе исследуется простейшая архитектура типа

«кольцо», приведенная на рис. 4.1. Исполняемый модуль программы моделирования имеет имя **Ring.exe** и находится в папке **Lab 12**. Запуск программы осуществляется традиционным способом. После запуска на

экране появляется окно, позволяющее задать параметры модели и режим моделирования. Оно содержит пять полей: два для ввода параметров, два — для вывода характеристик и последнее — для изображения структуры сети и передачи пакетов по ней.

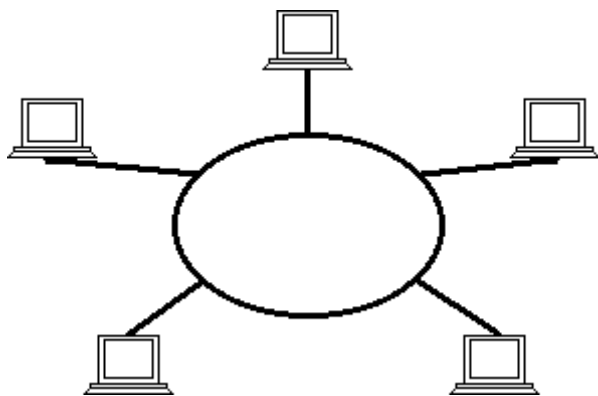


Рис. 4.1. Сеть с архитектурой типа «кольцо»

В первом блоке, «Характеристики задачи», расположены элементы, позволяющие задать:

1. Количество адресатов, которым будут посланы пакеты. Значение этого параметра варьируется от 1 до количества станций-1. По умолчанию оно равно 1.

2. Количество пакетов, которые будут отправлены каждому из адресатов. Изменяется от 1 до 10. По умолчанию оно равно 1.
3. Длину отправленного пакета. Значение изменяется от 1024 до 10240 с шагом, кратным пропускной способности сети. Единица измерения – бит.
4. Интервал между посылкой двух соседних пакетов. Измеряется в миллисекундах. По умолчанию он равен 5мс. Сам интервал является случайной величиной, значения которой находятся в диапазоне от 0 до заданной величины.

В окне «Характеристики сети» вводятся:

1. Пропускная способность сети. Она измеряется в бодах и по умолчанию равна 1024 бод.
2. Количество станций. Значение меняется от 3 до 10. По умолчанию оно равно 3. При изменении значения перерисовывается сеть в области графического отображения процесса моделирования, а также динамически меняется размерность таблицы выводов результатов моделирования.

Изменения значений всех полей ввода на форме осуществляется путём нажатия на стрелки вверх и вниз, находящиеся справа от полей ввода, а также при наличии колеса на мыши путём его прокручивания.

Программа, как отмечалось выше, работает в двух режимах: непрерывно и по тактам. Режим определяется нажатием одной двух кнопок запуска:

- «Старт» (непрерывный режим работы).
- «Шаг» (потактовое выполнение).

В непрерывном режиме программа имитирует очередной такт автоматически, через определенный промежуток времени, который задаётся положением бегунка, расположенного рядом с кнопкой «Старт». Причём первая позиция бегунка соответствует решению задачи без отображения на экран процесса передачи пакетов по сети. Вторая и последующие позиции кратны 1/10 части секунды. Положение бегунка можно изменять и в процессе работы. При этом первая позиция соответствует паузе в моделировании.

Во время работы программы динамически изменяются значения временных характеристик. По окончании моделирования на экран выводятся:

1. Загрузка сети.
2. Среднее время доставки пакетов.
3. Для каждой станции отображается
 - Количество отправленных и полученных пакетов.
 - Пропуск пакетов в результате ожидания свободного маркера, а также само время ожидания.

Программа позволяет сохранить параметры модели и результаты моделирования в файле, имя которого задается пользователем. Кроме того, она обеспечивает возможность загрузки параметров из файла. Для этого окно снабжено меню «Файл», в котором находятся команды «Открыть» и «Сохранить как».

Команда «Сохранить как» позволяет сохранить два типа данных:

1. Структуру – количество станций, количество адресатов и так далее.
2. Отчёт, содержащий исходные данные и результаты моделирования.

Команда «Открыть» позволяет открыть сохранённую ранее структуру данных.

Порядок выполнения практической работы

Подготовка к работе

1. Знакомство со всеми разделами руководства.
2. Получение у преподавателя задания на исследование сети с различными параметрами устройств и потока задач.
3. Исследование заданной сети.
4. Оформление отчета.

Последовательность выполнения практической работы

В практической работе необходимо провести следующие исследования.

1. Задавая **разное количество станций N** , при **одинаковых параметрах пакетов** и канала, а также **фиксированном количестве адресатов** каждой станции (например, 1) определить характеристики сети в целом, а также отдельных станций.
2. Выполнить исследования, как предложено в п.1, для **количеств адресатов**, меняющихся в пределах от 2 до $N-1$.
3. Задавая **разные количества пакетов**, обрабатываемых станциями, при **постоянном интервале** между ними и общем количестве станций, а также **фиксированном количестве адресатов** каждой станции (например, 1), определить характеристики сети в целом, а также отдельных станций. Количество станций задается преподавателем.
4. Выполнить исследования, как предложено в п.3, для **количеств адресатов**, меняющихся в пределах от 2 до $N-1$.
5. Варьируя **пропускную способность линий** связи при **одинаковых параметрах пакетов и канала**, а также **фиксированном количестве адресатов** каждой станции (например, 1) определить характеристики сети в целом, а также отдельных станций. Количество станций задается преподавателем.
6. Выполнить исследования, как предложено в п.5, для **количеств адресатов**, меняющихся в пределах от 2 до $N-1$.
7. Построить графики зависимости полученных характеристик сети в целом и одной из станций (по заданию преподавателя) от изменяемых в п.п. 1 — 6 параметров.

Содержание отчета о выполненной работе

Отчет должен содержать следующее:

1. Название и цель работы.
2. Исходные данные.
3. Графики зависимостей временных характеристик сети в целом и одной из станций для вариантов, перечисленных в разделе «Порядок выполнения работы».

Контрольные вопросы

1. Назовите основные типы структур локальных сетей и области их применения.
 2. Каковы основные отличия кольцевых сетей от других архитектур?
 3. Как влияет на характеристики сети количество активных станций?
 4. Как влияет на характеристики сети количество пакетов, передаваемых задачами?
 5. Как влияет на характеристики сети длина пакетов, передаваемых задачами?
 6. Как влияет на характеристики сети величина интервала между пакетами?
 7. Как влияет на характеристики сети количество адресатов пакетов?
 8. Как влияет на характеристики сети пропускная способность линий связи?
 9. Как влияют на характеристики отдельных станций параметры передаваемых ими пакетов?
 10. Как влияет на характеристики отдельных станций количество адресатов пакетов, отправляемых ими?
- .Как влияет на характеристики отдельных станций пропускная способность линий?

Практическая работа №8

«Конфликты при конвейерной обработке данных в процессоре»

Цель работы: Ознакомление с принципами конвейерной обработки данных в процессоре. Изучение конфликтов при конвейерной обработке методов и способов их предотвращения и разрешения.

Теоретические сведения

Конфликты при работе конвейеров

1. Структурные конфликты (риски), возникают из-за конфликтов по ресурсам при попытке нескольких команд одновременно обратиться к одному и тому же ресурсу вычислительной машины (например к памяти).

2. Конфликты по данным, возникают в случае, когда выполнение одной команды зависит от результата выполнения предыдущей команды.

3. Конфликты по управлению, возникают при конвейеризации команд переходов и других команд передачи управления (вызов процедуры, прерывание и т. п.).

Разрешение конфликтов

1. Структурные конфликты возникают не часто в связи с тем, что память кэшируется. В некоторых случаях возникает необходимость дублирования некоторых ресурсов.

2. Конфликты по данным возникают при логической зависимости между командами. Для решения зависимости необходимо выбирать команды специальным образом.

Виды конфликтов по данным

1. Чтение после записи (ЧПЗ – RAW). Существует две команды и, если одна команда пытается прочитать операнд используемых данных прежде, чем другая команда туда запишет, возникает конфликт.

2. Запись после чтения (ЗПЗ – WAR). Одна команда пытается записать результат в приемник раньше, чем другая команда прочитывает эти данные.

3. Запись после записи (ЗПЗ – WAW). Одна команда пытается записать операнд прежде, чем будет записан результат другой команды, т. е. записи заканчиваются в неверном порядке.

Признак конфликта

Нарушение хотя бы одного условия Бернштейна, служащее критерием возможности возникновения конфликта по данным.

1. ЧПЗ

$$O(i) \cap I(j) = 0,$$

где $O(i)$ – множество ячеек, при команде i ; $I(j)$ – множество ячеек, при команде j , следующие за командой i .

2. ЗПЧ

$$I(i) \cap O(j) = 0,$$

где I -input; O -output.

3. ЗПЗ

$$O(i) \cap O(j) = 0.$$

Методы устранения конфликтов по данным

1. Программные:

а) на этапе трансляции создается объектный код, в котором между командами, склонными к конфликту по данным, вставляются пустые операции;

б) на этапе трансляции создается объектный код, в котором команды i и j , склонные к конфликту по данным, переставляются так, чтобы между ними были команды, не связанные по данным с этими командами; при этом команда i может быть перенесена вперед – что более затруднительно, или команда j переставлена назад, что легче выполнимо.

2. Аппаратные средства:

а) разрешают конфликт во время выполнения, т. е. команда j задерживается на несколько тактов, чтобы i успела завершиться и миновать ступени конвейера, вызвавшие конфликт;

б) ускоренное продвижение требуемой информации *forwarding data* (табл. 2).

Табл. 2. Пример ускоренного продвижения информации

| | | | | | | | | | |
|------------------|----|----|----|-----|-----|-----|-----|-----|----|
| Add R1, R2, R3 | IF | ID | EX | MEM | WB | | | | |
| SUB R4, R1 R5 | | IF | ID | EX | MEM | WB | | | |
| AND R6, R1, R7 | | | IF | ID | EX | MEM | WB | | |
| OR R8, R1, R9 | | | | IF | ID | EX | MEM | WB | |
| XOR R10, R1, R11 | | | | | IF | ID | EX | MEM | WB |

В регистр информация записывается упреждающим способом. Для этого существует специальные схемы АЛУ с цепями обхода и ускоренной пересылкой.

3. Оптимизация с помощью компилятора: $a = v + c; d = e + f$

(табл. 3).

Табл. 3. Пример оптимизации

| Неоптимальное | Оптимальное |
|-----------------------|-----------------------|
| <i>LW Rb, b</i> | <i>LW Rb, b</i> |
| <i>LW Rc, C</i> | <i>LW Rc, C</i> |
| <i>ADD Ra, Rb, Rc</i> | <i>LW Re, e</i> |
| <i>SW a, Ra</i> | <i>ADD Ra, Rb, Rc</i> |
| <i>LW Re, e</i> | <i>LW Rf, f</i> |
| <i>LW Rf, f</i> | <i>SW a, Ra</i> |
| <i>SUB Rd, Re, Rf</i> | <i>SUB Rd, Re, Rf</i> |
| <i>SW d, Rd</i> | <i>SW d, Rd</i> |

В простейшем алгоритме компиляции компилятор планирует распределение команд в одном и том же базовом блоке (это линейный участок кода, в котором нет переходов).

4. Использование временных регистровых результатов. Существуют логические регистры и физические регистры. Временные значения записываются в файловые регистры вместе с постоянными. Временные значения становятся новым постоянным только после фиксации результата (пятая ступень конвейера). Завершение выполнения команды происходит тогда, когда все предыдущие команды завершились успешно в заданном программой порядке. Такой подход называется методом переименования регистров.

Конфликты по управлению

Это неоднозначность при выборке следующей команды в случае перехода.

Методы борьбы:

- 1) буферы предвыборки;
- 2) множественные потоки;
- 3) задержанный переход;
- 4) предсказание перехода.

Наиболее часто используется предсказание перехода, для которого характерны следующие стратегии.

1. Статическое определение направления перехода:

- переход происходит всегда (ПВ), т. е. переход выполняемый, применяется при организации цикла;
- переход не происходит (ПНВ), т. е. переход невыполняемый;
- предсказание определяется по результатам профилирования (это тестовые прогоны программы);
- предсказание определяется кодом операции команды перехода; предсказание зависит от направления перехода (вперед или назад);
- при первом выполнении переход выполняется всегда.

2. Динамическое определение направления перехода выполняется на основе прогноза.

Прогнозы различаются по глубине:

- однобитовый – т. е. такой, какой был на предыдущем шаге.

Осуществляется на основе автомата.

- двухбитовый – т. е. на основе двух предыдущих выполненных переходов.

Выполнение работы

1. Изучить виды конфликтов при конвейерной обработке данных в процессоре и методы и способы их предотвращения и разрешения, используя литературу и материалы лекций.
2. Разработать алгоритмы и программы, позволяющие обнаружить, и либо

предотвратить конфликт, либо выйти из него.

3. Выполнить программу с несколькими (2–4) наборами исходных данных.

Примечания:

1. При программировании можно использовать любой язык.
2. Фрагмент программы, в котором возможен конфликт, должен быть написан на языке Ассемблера для RISC-процессора, т. е. с использованием только регистровых операндов и специальных операций работы с памятью.

1. Цель работы.

Содержание отчета

2. Теоретические сведения о возникновении и разрешении конфликта.
3. Алгоритм обнаружения и предотвращения конфликта.
4. Программа обнаружения и предотвращения конфликта.
5. Условия экспериментов (детальное описание параметров и конфигурации ВС).
6. Фрагменты исходных программ и программ, полученных в результате преобразования, в которых конфликты устранены.

7. Выводы.

Темы индивидуальных заданий

Вид конфликта, который надо разрешить, и способ разрешения:

1. Конфликт по данным типа ЧПЗ.
2. Конфликт по данным типа ЗПЧ.
3. Конфликт по данным типа ЗПЗ.
4. Конфликт по управлению – статическое предсказание выполняемого перехода.
5. Конфликт по управлению – статическое предсказание невыполняемого перехода.
6. Конфликт по управлению – статическое предсказание выполняемого перехода.
7. Конфликт по управлению – динамическое предсказание на основе однобитового прогноза перехода.
8. Конфликт по управлению – динамическое предсказание на основе двухбитового прогноза перехода.

Вопросы для контроля

1. Конвейерная организация. Типы конфликтов в конвейере.
2. Конвейерная организация. Структурные конфликты и способы минимизации потерь при них.
3. Конвейерная организация. Конфликты по данным и способы их разрешения.
4. Конвейерная организация. Конфликты по данным – классификация.
5. Конвейерная организация. Конфликты по данным – методика планирования компилятора для их устранения. Средства динамической оптимизации.
6. Конвейерная организация. Методы снижения потерь на выполнение команд перехода.
7. Конвейерная и суперскалярная обработка. Аппаратное прогнозирование направлений переходов и снижение потерь на организацию переходов.
8. Виды конфликтов по данным при работе конвейера и способы их устранения.
9. Виды конфликтов по управлению при работе конвейера и способы их устранения.
10. Автомат двухбитового предсказания.

Практическая работа №9 «ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА ТОМАСУЛО»

Цель работы: Исследовать алгоритм Томасуло и реализовать его программно.

Теоретические сведения
Обобщенное словесное описание алгоритма
Программа представляет собой программную модель архитектуры арифметико-логического устройства с плавающей точкой (ПТ), разработанную Томасуло.

Устройство состоит:

- из очереди операций ПТ;
- буфера загрузки из памяти;
- буфера записи в память;
- регистров ПТ;
- станций резервирования сложения;
- станций резервирования умножения;
- устройства сложения;
- устройства умножения;
- ОШД (общей шины данных).

Шаг прохода алгоритма Томасуло

На каждом шаге происходит просмотр состояния каждого из устройств.

1 . Просмотр очереди команд.

Из очереди забирается наиболее возможное число команд при условии наличия свободных буферов загрузки, записи, станций резервирования сложения, умножения.

2 . Просмотр буфера загрузки.

Если есть активный буфер загрузки, то он продолжает читать память еще один такт, если он закончил чтение, то результат выдается на ОШД, и буфер освобождается. Активного буфера загрузки теперь нет.

Если нет активного буфера загрузки, то он ищется. Если находится буфер, готовый к выполнению (с приоритетом 0), то он начинает выполняться и становится активным.

3 . Просмотр устройства сложения.

Так как у нас буферное устройство сложения, то оно параллельно может выполнять несколько команд сложения/вычитания. Поэтому каждая операция, выполняемая в нем, продолжает выполняться еще один такт. Если какая-то операция выполнена, то результат выдается на ОШД, если она свободна, а если занята – то результат сохраняется в устройстве сложения. Освобождаются станции резервирования сложения, где хранились операции, завершившиеся в этом такте.

4 . Просмотр устройства умножения.

Так как у нас буферное устройство умножения, то оно параллельно может выполнять несколько команд умножения/деления. Поэтому каждая операция, выполняемая в нем, продолжает выполняться еще один такт. Если какая-то операция выполнена, то результат выдается на ОШД,

если она свободна, а если занята – то результат сохраняется в устройстве умножения. Освобождаются станции резервирования умножения, где хранились операции, завершившиеся в этом такте.

5 . Просмотр ОШД.

Если на ОШД есть результат выполнения какой-либо операции, то все устройства, ждущие операнда, забирают значение с ОШД, если это ожидаемый операнд.

6 . Просмотр станции резервирования сложения.

Если после просмотра ОШД какие-либо операции получили ожидаемый операнд и готовы к выполнению, то они отправляются на выполнение в устройство сложения.

7 . Просмотр станции резервирования умножения.

Если после просмотра ОШД какие-либо операции получили ожидаемый операнд и готовы к выполнению, то они отправляются на выполнение в устройство умножения.

8 . Просмотр буферов записи.

Если есть активный буфер записи, то он продолжает записывать память еще один такт, если он закончил запись, то буфер освобождается. Активного буфера записи теперь нет.

Если нет активного буфера записи, то он ищется. Если находится буфер, готовый к выполнению (имеющий операнд, с приоритетом 0), то он начинает выполняться и становится активным.

Примечание: Блок-схемы работы программы и текст программы приведены в электронном варианте методических указаний.

Интерфейс программы (рис. 3 – б)

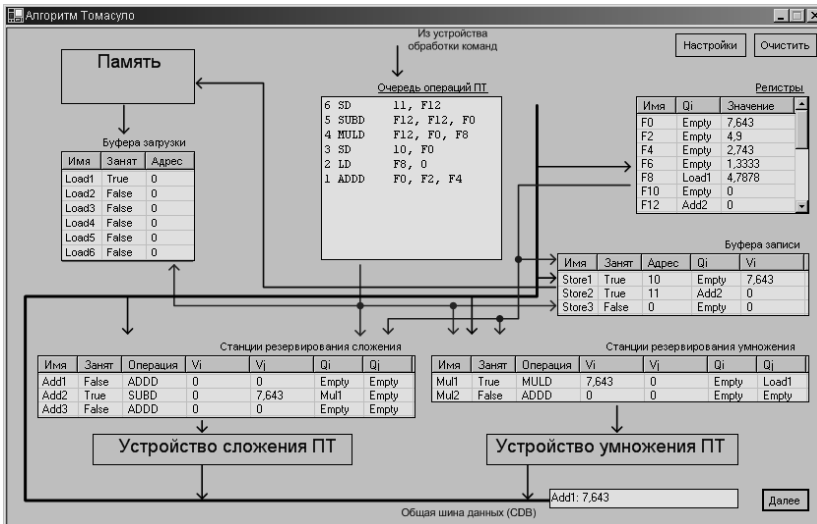


Рис. 3. Окно «Алгоритм Томасуло»

The 'Память' (Memory) window shows the following data:

| Address | Value |
|---------|-------|
| 0 | 121 |
| 1 | 33 |
| 2 | 45 |
| 3 | 656 |
| 4 | 33,3 |
| 5 | 4454 |
| 6 | 234 |
| 7 | 445 |
| 8 | 1,32 |
| 9 | 3,34 |
| 10 | 0 |
| 11 | 0 |

| Name | Owner | Entry |
|------|-------|--------|
| F0 | Empty | 7,643 |
| F2 | Empty | 4,9 |
| F4 | Empty | 2,743 |
| F6 | Empty | 1,3333 |
| F8 | Load1 | 4,7878 |
| F10 | Empty | 0 |
| F12 | Add2 | 0 |
| F14 | Empty | 0 |
| F16 | Empty | 0 |
| F18 | Empty | 0 |
| F20 | Empty | 0 |
| F22 | Empty | 0 |

Рис. 4. Окно «Память»

Рис. 5. Окно «Регистры»

```

Текст программы
add  f0, f2, f4
ld   f8, 0
sd   10, f0
muld f12, f0, f8
subd f12, f12, f0
sd   11, f12

Сохранить...  Из файла...  Компилировать

Ошибки компиляции
Ошибок нет

Закреть окно
  
```

Рис. 6. Окно «Код программы»

Код программы:

Тесты

; проверка конфликта типа RAW, WAW (регистры);

; задача: найти сумму первых 6-ти ячеек памяти;

; и записать результат в 7-ю;

; исходные данные: ненулевые значения

; 0-5 ячеек памяти,

; f2 = 0

ld f0, 0

add f2, f2, f0 ld f0, 1

| Address | Value |
|---------|-------|
| 0 | 2 |
| 1 | 3 |
| 2 | 4 |
| 3 | 5 |
| 4 | 6 |
| 5 | 7 |
| 6 | 27 |
| 7 | 0 |
| 8 | 0 |
| 9 | 0 |
| 10 | 0 |
| 11 | 0 |
| 12 | 0 |
| 13 | 0 |
| 14 | 0 |
| 15 | 0 |

```

add f2, f2, f0 ld f0, 2
add f2, f2, f0 ld f0, 3
add f2, f2, f0 ld f0, 4
add f2, f2, f0 ld f0, 5
add f2, f2, f0 sd 6, f2

```

После проведения теста окно памяти будет выглядеть как показано на рис. 7.

Рис. 7. Окно памяти после выполнения

Выполнение работы

1. Изучить алгоритм Томасуло и его программную реализацию.
2. Запустить тестовый пример и убедиться в правильной работе программы.
3. Исследовать алгоритм Томасуло и его программную реализацию при обработке программы, содержащей конфликт заданного типа в соответствии с вариантом. Для этого составить программу на языке Ассемблер, содержащую конфликт заданного типа, и проверить правильность его разрешения.

Варианты заданий – тип конфликта:

- 1) RAW;
- 2) WAW;
- 3) WAR.

Вопросы для контроля

1. Конвейерная организация. Конфликты по данным и способы их разрешения.
2. Конвейерная организация. АЛУ с цепями обхода и ускоренной пересылки.
3. Конвейерная организация. Конфликты по данным – классификация.
4. Конвейерная и суперскалярная обработка. Идеи динамической оптимизации и ее реализация с централизованной схемой обнаружения конфликтов.
5. Алгоритм Томасуло.

Литература

Перечень основной литературы

- 1 Ключев, А. О. Распределенные информационно-управляющие системы [Электронный ресурс] : учебное пособие / А. О. Ключев, П. В. Кустарев, А. Е. Платунов. — Электрон. текстовые данные. — СПб. : Университет ИТМО, 2015. — 58 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/68081.html>
- 2 Граничин О.Н. Информационные технологии в управлении. — Электрон. текст. дан. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — Режим доступа : <http://www.iprbookshop.ru/57379>.— ЭБС «IPRbooks», по паролю

Перечень дополнительной литературы

- 1 Болодурина, И. П. Проектирование компонентов распределенных информационных систем [Электронный ресурс] : учебное пособие / И. П. Болодурина, Т. В. Волкова. — Электрон. текстовые данные. — Оренбург : Оренбургский государственный университет, ЭБС АСВ, 2012. — 215 с. — 978-5-4417-0077-1. — Режим доступа: <http://www.iprbookshop.ru/30122.html>
- 2 Крищенко, В. А. Технологии создания кросс-платформенных распределенных приложений [Электронный ресурс] : учебное пособие / В. А. Крищенко. — Электрон. текстовые данные. — М. : Московский государственный технический университет имени Н.Э. Баумана, 2009. — 40 с. — 978-5-7038-3316-2. — Режим доступа: <http://www.iprbookshop.ru/31576.html>.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
НЕВИННОМЫССКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)

Методические указания к организации самостоятельной работы
студентов направления подготовки

15.03.04 «Автоматизация технологических процессов и производств»

Направленность (профиль) Информационно-управляющие системы
по дисциплине

**« РАСПРЕДЕЛЕННЫЕ КОМПЬЮТЕРНЫЕ ИНФОРМАЦИОННО-
УПРАВЛЯЮЩИЕ СИСТЕМЫ »**

Невинномысск 2022

Методические указания предназначены для студентов направления 15.03.04 «Автоматизация технологических процессов и производств» и других технических специальностей. Они содержат рекомендации по организации самостоятельных работ студента на направления 15.03.04 для дисциплины «Распределенные компьютерные информационно-управляющие системы».

Методические указания разработаны в соответствии с требованиями ФГОС ВПО в части содержания и уровня подготовки выпускников направления 15.03.04 «Автоматизация технологических процессов и производств».

Составитель

канд. техн. наук Э.Е. Тихонов

Ответственный редактор

канд. техн. наук Д.В. Болдырев

Содержание

| | |
|------------------------------------------------|---|
| 1 Подготовка к практическим занятиям..... | 4 |
| 2 Самостоятельное изучение темы. Конспект..... | 5 |
| 3 Подготовка к экзамену | 6 |
| 4 Требования к реферату | 7 |

1 Подготовка к практическим занятиям

Подготовку к каждому практическому занятию студент должен начать с ознакомления с методическими указаниями, которые включают содержание работы. Тщательное продумывание и изучение вопросов основывается на проработке текущего материала лекции, а затем изучения обязательной и дополнительной литературы, рекомендованную к данной теме. На основе индивидуальных предпочтений студенту необходимо самостоятельно выбрать тему доклада по проблеме и по возможности подготовить по нему презентацию.

Если программой дисциплины предусмотрено выполнение практического задания, то его необходимо выполнить с учетом предложенной инструкции (устно или письменно). Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса. Результат такой работы должен проявиться в способности студента свободно ответить на теоретические вопросы семинара, его выступлении и участии в коллективном обсуждении вопросов изучаемой темы, правильном выполнении практических заданий и контрольных работ.

В зависимости от содержания и количества отведенного времени на изучение каждой темы практическое занятие может состоять из четырех-пяти частей:

1. Обсуждение теоретических вопросов, определенных программой дисциплины.
2. Доклад и/ или выступление с презентациями по выбранной проблеме.
3. Обсуждение выступлений по теме – дискуссия.
4. Выполнение практического задания с последующим разбором полученных результатов или обсуждение практического задания.
5. Подведение итогов занятия.

Первая часть – обсуждение теоретических вопросов – проводится в виде фронтальной беседы со всей группой и включает выборочную проверку преподавателем теоретических знаний студентов. Примерная продолжительность — до 15 минут. Вторая часть — выступление студентов с докладами, которые должны сопровождаться презентациями с целью усиления наглядности восприятия, по одному из вопросов практического занятия. Обязательный элемент доклада – представление и анализ статистических данных, обоснование социальных последствий любого экономического факта, явления или процесса. Примерная продолжительность — 20-25 минут. После докладов следует их обсуждение – дискуссия. В ходе этого этапа практического занятия могут быть заданы уточняющие вопросы к докладчикам.

Примерная продолжительность – до 15-20 минут. Если программой предусмотрено выполнение практического задания в рамках конкретной темы, то преподавателями определяется его содержание и дается время на его выполнение, а затем идет обсуждение результатов. Подведением итогов заканчивается практическое занятие.

В процессе подготовки к практическим занятиям, студентам необходимо обратить особое внимание на самостоятельное изучение рекомендованной учебно-методической (а также научной и популярной) литературы. Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому

усвоению изучаемого материала, формирует у студентов свое отношение к конкретной проблеме. Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем по каждой теме семинарского или практического занятия, что позволяет студентам проявить свою индивидуальность в рамках выступления на данных занятиях, выявить широкий спектр мнений по изучаемой проблеме.

2 Самостоятельное изучение темы. Конспект

Конспект – наиболее совершенная и наиболее сложная форма записи. Слово «конспект» происходит от латинского «conspectus», что означает «обзор, изложение». В правильно составленном конспекте обычно выделено самое основное в изучаемом тексте, сосредоточено внимание на наиболее существенном, в кратких и четких формулировках обобщены важные теоретические положения.

Конспект представляет собой относительно подробное, последовательное изложение содержания прочитанного. На первых порах целесообразно в записях ближе держаться тексту, прибегая зачастую к прямому цитированию автора. В дальнейшем, по мере выработки навыков конспектирования, записи будут носить более свободный и сжатый характер.

Конспект книги обычно ведется в тетради. В самом начале конспекта указывается фамилия автора, полное название произведения, издательство, год и место издания. При цитировании обязательная ссылка на страницу книги. Если цитата взята из собрания сочинений, то необходимо указать соответствующий том. Следует помнить, что четкая ссылка на источник – неперемное правило конспектирования. Если конспектируется статья, то указывается, где и когда она была напечатана.

Конспект подразделяется на части в соответствии с заранее продуманным планом. Пункты плана записываются в тексте или на полях конспекта. Писать его рекомендуется четко и разборчиво, так как небрежная запись с течением времени становится малопонятной для ее автора. Существует правило: конспект, составленный для себя, должен быть по возможности написан так, чтобы его легко прочитал и кто-либо другой.

Формы конспекта могут быть разными и зависят от его целевого назначения (изучение материала в целом или под определенным углом зрения, подготовка к докладу, выступлению на занятии и т.д.), а также от характера произведения (монография, статья, документ и т.п.). Если речь идет просто об изложении содержания работы, текст конспекта может быть сплошным, с выделением особо важных положений подчеркиванием или различными значками.

В случае, когда не ограничиваются переложением содержания, а фиксируют в конспекте и свои собственные суждения по данному вопросу или дополняют конспект соответствующими материалами их других источников, следует отводить место для такого рода записей. Рекомендуется разделить страницы тетради пополам по вертикали и в левой части вести конспект произведения, а в правой свои дополнительные записи, совмещая их по содержанию.

Конспектирование в большей мере, чем другие виды записей, помогает вырабатывать навыки правильного изложения в письменной форме важные теоретических и практических вопросов, умение четко их формулировать и ясно излагать своими словами.

Таким образом, составление конспекта требует вдумчивой работы, затраты времени и труда. Зато во время конспектирования приобретаются знания, создается фонд записей.

Конспект может быть текстуальным или тематическим. В текстуальном конспекте сохраняется логика и структура изучаемого произведения, а запись ведется в соответствии с расположением материала в книге. За основу тематического конспекта берется не план произведения, а содержание какой-либо темы или проблемы.

Текстуальный конспект желательно начинать после того, как вся книга прочитана и продумана, но это, к сожалению, не всегда возможно. В первую очередь необходимо составить план произведения письменно или мысленно, поскольку в соответствии с этим планом строится дальнейшая работа. Конспект включает в себя тезисы, которые составляют его основу. Но, в отличие от тезисов, конспект содержит краткую запись не только выводов, но и доказательств, вплоть до фактического материала. Иначе говоря, конспект – это расширенные тезисы, дополненные рассуждениями и доказательствами, мыслями и соображениями составителя записи.

Как правило, конспект включает в себя и выписки, но в него могут войти отдельные места, цитируемые дословно, а также факты, примеры, цифры, таблицы и схемы, взятые из книги. Следует помнить, что работа над конспектом только тогда будет творческой, когда она не ограничена текстом изучаемого произведения. Нужно дополнять конспект данными из другими источниками.

В конспекте необходимо выделять отдельные места текста в зависимости от их значимости. Можно пользоваться различными способами: подчеркиваниями, вопросительными и восклицательными знаками, репликами, краткими оценками, писать на полях своих конспектов слова: «важно», «очень важно», «верно», «характерно».

В конспект могут помещаться диаграммы, схемы, таблицы, которые придадут ему наглядность.

Составлению тематического конспекта предшествует тщательное изучение всей литературы, подобранной для раскрытия данной темы. Бывает, что какая-либо тема рассматривается в нескольких главах или в разных местах книги. А в конспекте весь материал, относящийся к теме, будет сосредоточен в одном месте. В плане конспекта рекомендуется делать пометки, к каким источникам (вплоть до страницы) придется обратиться для раскрытия вопросов. Тематический конспект составляется обычно для того, чтобы глубже изучить определенный вопрос, подготовиться к докладу, лекции или выступлению на семинарском занятии. Такой конспект по содержанию приближается к реферату, докладу по избранной теме, особенно если включает и собственный вклад в изучение проблемы.

3 Подготовка к экзамену

Экзаменационная сессия – очень тяжелый период работы для студентов и ответственный труд для преподавателей. Главная задача экзаменов – проверка качества усвоения содержания дисциплины.

На основе такой проверки оценивается учебная работа не только студентов, но и преподавателей: по результатам экзаменов можно судить и о качестве всего учебного процесса. При подготовке к экзамену студенты повторяют материал курсов, которые они слушали и изучали в течение семестра, обобщают полученные знания,

выделяют главное в предмете, воспроизводят общую картину для того, чтобы яснее понять связь между отдельными элементами дисциплины.

При подготовке к экзаменам основное направление дают программы курса и конспект, которые указывают, что в курсе наиболее важно. Основной материал должен прорабатываться по учебнику, поскольку конспекта недостаточно для изучения дисциплины. Учебник должен быть проработан в течение семестра, а перед экзаменом важно сосредоточить внимание на основных, наиболее сложных разделах. Подготовку по каждому разделу следует заканчивать восстановлением в памяти его краткого содержания в логической последовательности.

До экзамена обычно проводится консультация, но она не может возместить отсутствия систематической работы в течение семестра и помочь за несколько часов освоить материал, требующийся к экзамену. На консультации студент получает лишь ответы на трудные или оставшиеся неясными вопросы. Польза от консультации будет только в том случае, если студент до нее проработает весь материал. Надо учиться задавать вопросы, вырабатывать привычку пользоваться справочниками, энциклопедиями, а не быть на иждивении у преподавателей, который не всегда может тут же, «с ходу» назвать какой-либо факт, имя, событие. На экзамене нужно показать не только знание предмета, но и умение логически связно построить устный ответ.

Получив билет, надо вдуматься в поставленные вопросы для того, чтобы правильно понять их. Нередко студент отвечает не на тот вопрос, который поставлен, или в простом вопросе ищет скрытого смысла. Не поняв вопроса и не обдумав план ответа, не следует начинать писать. Конспект своего ответа надо рассматривать как план краткого сообщения на данную тему и составлять ответ нужно кратко. При этом необходимо показать умение выражать мысль четко и доходчиво.

Отвечать нужно спокойно, четко, продуманно, без торопливости, придерживаясь записи своего ответа. На экзаменах студент показывает не только свои знания, но и учится владеть собой. После ответа на билет могут следовать вопросы, которые имеют целью выяснить понимание других разделов курса, не вошедших в билет. Как правило, на них можно ответить кратко, достаточно показать знание сути вопроса. Часто студенты при ответе на дополнительные вопросы проявляют поспешность: не поняв смысла того, что у них спрашивают, начинают отвечать и нередко говорят не по сути.

Следует помнить, что необходимым условием правильного режима работы в период экзаменационной сессии является нормальный сон, поэтому подготовка к экзаменам не должна быть в ущерб сну. Установлено, что сильное эмоциональное напряжение во время экзаменов неблагоприятно отражается на нервной системе и многие студенты из-за волнений не спят ночи перед экзаменами. Обычно в сессию студенту не до болезни, так как весь организм озабочен одним - сдать экзамены. Но это еще не значит, что последствия неправильно организованного труда и чрезмерной занятости не скажутся потом. Поэтому каждый студент помнить о важности рационального распорядка рабочего дня и о своевременности снятия или уменьшения умственного напряжения.

Реферат это одна из форм устной итоговой аттестации. Реферат – это самостоятельная исследовательская работа, в которой автор раскрывает суть исследуемой проблемы; приводит различные точки зрения, а также собственные взгляды на нее. Содержание реферата должно быть логичным; изложение материала носит проблемно-тематический характер.

Цель

Реферат как форма промежуточной (итоговой) аттестации стимулирует раскрытие исследовательского потенциала магистра, способность к творческому поиску, сотрудничеству, самораскрытию и проявлению возможностей.

Требования к реферату

Автор реферата должен продемонстрировать достижение им уровня мировоззренческой, общекультурной компетенции, т.е. продемонстрировать знания о реальном мире, о существующих в нем связях и зависимостях, проблемах, о ведущих мировоззренческих теориях, умении проявлять оценочные знания, изучать теоретические работы, использовать различные методы исследования, применять различные приемы творческой деятельности.

1. Необходимо правильно сформулировать тему, отобрать по ней необходимый материал.
2. Использовать только тот материал, который отражает сущность темы.
3. Во введении к реферату необходимо обосновать выбор темы.
4. После цитаты необходимо делать ссылку на автора, например [№произведения по списку, стр.].
5. Изложение должно быть последовательным. Недопустимы нечеткие формулировки, речевые и орфографические ошибки.
6. В подготовке реферата необходимо использовать материалы современных изданий не старше 5 лет.
7. Оформление реферата (в том числе титульный лист, литература) должно быть грамотным.
8. Список литературы оформляется с указанием автора, названия источника, места издания, года издания, названия издательства, использованных страниц.

Требования к оформлению реферата

Изложение текста и оформление реферата выполняются в соответствии с требованиями ГОСТ 7.32 – 2001, ГОСТ 2.105 – 95 и ГОСТ 6.38 – 90. Страницы текстовой части и включенные в нее иллюстрации и таблицы должны соответствовать формату А4 по ГОСТ 9327-60.

Реферат должен быть выполнен любым печатным способом на пишущей машинке или с использованием компьютера и принтера на одной стороне бумаги формата А4 через полтора интервала. Цвет шрифта должен быть черным, высота букв, цифр и других знаков не менее 1.8 (шрифт Times New Roman, 14 пт.).

Текст следует печатать, соблюдая следующие размеры полей: верхнее и нижнее — 20 мм, левое — 30 мм, правое — 10 мм. Абзацный отступ должен быть одинаковым по всему тексту и составлять 1,25 см.

Выравнивание текста по ширине.

Разрешается использовать компьютерные возможности акцентирования внимания на определенных терминах, формулах, применяя выделение жирным шрифтом, курсив, подчеркивание.

Перенос слов недопустим.

Точку в конце заголовка не ставят. Если заголовок состоит из двух предложений, их разделяют точкой.

Подчеркивать заголовки не допускается.

Расстояние между заголовками раздела, подраздела и последующим текстом так же, как и расстояние между заголовками и предыдущим текстом, должно быть равно 15мм (2 пробела).

Название каждой главы и параграфа в тексте работы можно писать более крупным шрифтом, жирным шрифтом, чем весь остальной текст. Каждая глава начинается с новой страницы, параграфы (подразделы) располагаются друг за другом.

В тексте реферат рекомендуется чаще применять красную строку, выделяя законченную мысль в самостоятельный абзац.

Перечисления, встречающиеся в тексте реферата, должны быть оформлены в виде маркированного или нумерованного списка.

Все страницы обязательно должны быть пронумерованы. Нумерация листов должна быть сквозной. Номер листа проставляется арабскими цифрами.

Нумерация листов начинается с третьего листа (после содержания) и заканчивается последним. На третьем листе ставится номер «3».

Номер страницы на титульном листе не проставляется.

Номера страниц проставляются в центре нижней части листа без точки. Список использованной литературы и приложения включаются в общую нумерацию листов.

Рисунки и таблицы, расположенные на отдельных листах, включают в общую нумерацию листов и помещают по возможности следом за листами, на которых приведены ссылки на эти таблицы или иллюстрации. Таблицы и иллюстрации нумеруются последовательно арабскими цифрами сквозной нумерацией. Допускается нумеровать рисунки и таблицы в пределах раздела. В этом случае номер таблицы (рисунка) состоит из номера раздела и порядкового номера таблицы, разделенных точкой.

Оформление литературы осуществляется с учетом требования ГОСТ.

Все источники, включенные в библиографию, должны быть последовательно пронумерованы и расположены в следующем порядке:

- законодательные акты;
- постановления Правительства;
- нормативные документы;
- статистические материалы;
- научные и литературные источники – в алфавитном порядке по первой букве фамилии автора.

В конце работы размещаются приложения. В тексте на все приложения должны быть даны ссылки. Каждое приложение следует начинать с новой страницы с указанием наверху посередине страницы слова «Приложение» и его номера. Приложение должно иметь заголовок, который записывают симметрично относительно текста с прописной буквы отдельной строкой.

Критерии оценок рефератов учащихся на итоговой аттестации

Критерии оценки:

- Актуальность темы
- Соответствие содержания теме
- Глубина проработки материала
- Правильность и полнота использования источников
- Соответствие оформления реферата стандартом.

На «отлично»:

1. присутствие всех вышеперечисленных требований;
 2. знание учащимся изложенного в реферате материала, умение грамотно и аргументировано изложить суть проблемы;
 3. присутствие личной заинтересованности в раскрываемой теме, собственную точку зрения, аргументы и комментарии, выводы;
 4. умение свободно беседовать по любому пункту плана, отвечать на вопросы, поставленные членами комиссии, по теме реферата;
 5. умение анализировать фактический материал и статистические данные, использованные при написании реферата;
 6. наличие качественно выполненного презентационного материала или (и) раздаточного, не дублирующего основной текст защитного слова, а являющегося его иллюстративным фоном.
- Т.е. при защите реферата показать не только «знание - воспроизведешь», но и «знание - понимание», «знание - умение».

На «хорошо»:

1. мелкие замечания по оформлению реферата;
2. незначительные трудности по одному из перечисленных выше требований.

На «удовлетворительно»:

1. тема реферата раскрыта недостаточно полно;
2. неполный список литературы и источников;
3. затруднения в изложении, аргументировании.

Порядок подготовки к защите реферата

Работа сдается руководителю не позже чем за 1 месяц до срока зачета с оценкой. Руководитель готовит рецензию, решает вопрос о допуске к защите реферата. Окончательное решение о допуске к защите реферата как формы аттестации принимает педагогический совет.

Литература

Перечень основной литературы:

- 1 Схиртладзе, А. Г. Автоматизация технологических процессов и производств [Электронный ресурс] : учебник / А. Г. Схиртладзе, А. В. Федотов, В. Г. Хомченко. — Электрон. текстовые данные. — Саратов : Вузовское образование, 2015. — 459 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/37830.html>
- 2 Павлов, Ю. А. Основы автоматизации производств [Электронный ресурс] : учебное пособие / Ю. А. Павлов. — Электрон. текстовые данные. — М. : Издательский Дом МИСиС, 2018. — 280 с. — 978-5-90846-78-5. — Режим доступа: <http://www.iprbookshop.ru/71666.html>

Перечень дополнительной литературы:

- 1 Герасимов, А. В. SCADA система Trace Mode 6 [Электронный ресурс] : учебное пособие / А. В. Герасимов, А. С. Титовцев. — Электрон. текстовые данные. — Казань : Казанский национальный исследовательский технологический университет, 2011. — 128 с. — 978-5-7882-1103-9. — Режим доступа: <http://www.iprbookshop.ru/62148.html>
- 2 Интегрированные системы проектирования и управления. SCADA-системы [Электронный ресурс] : учебное пособие / И. А. Елизаров, А. А. Третьяков, А. Н. Пчелинцев [и др.]. — Электрон. текстовые данные. — Тамбов : Тамбовский государственный технический университет, ЭБС АСВ, 2015. — 160 с. — 978-5-8265-1469-6. — Режим доступа: <http://www.iprbookshop.ru/63849.html>.