

ФГБОУ ВПО «Алтайский государственный технический университет  
им. И.И. Ползунова»

**Методические указания к выполнению лабораторных работ и курсового  
проектирования для  
УЧЕБНОЙ ДИСЦИПЛИНЫ**

**«ПРОЕКТИРОВАНИЕ СЕТЕВЫХ И МНОГОПОТОЧНЫХ ПРИЛОЖЕНИЙ»**

*Для студентов направления*  
**231000.68 «Программная инженерия**

Барнаул  
2014

## Ведение

Полноценное изучение дисциплины «Проектирование сетевых и многопоточных приложений» невозможно без практической реализации алгоритмов на лабораторных работах и курсовом проектировании, самостоятельной проработки материала по темам лабораторных работ. Поэтому практикум на ЭВМ является неотъемлемой частью этой дисциплины. Цель работ состоит в закреплении теоретического материала, приобретении практического опыта проектирования и реализации распределенных и многопоточных алгоритмов, получении навыков тестирования и анализа производительности реализованных программ.

Защита лабораторной работы осуществляется при наличии печатного или рукописного отчета, если он предусмотрен для данной работы. На защиту предъявляются выполненные задания в соответствии с перечнем для данной лабораторной работы (см. в нормативных материалах на групповом сервере требования к оформлению отчетов). Защита проводится в форме ответов на вопросы преподавателя о проделанной работе, проверки работоспособности программы на предъявленных тестах. При защите необходимо знать и уметь то, что перечислено в задании данной лабораторной работы.

*Контроль.* Текущий контроль успеваемости включает учет сроков и результатов защиты лабораторных работ, каждая из которых имеет вес, определяемый ее информативностью, значимостью и прикладной направленностью. В результате формируется текущий рейтинг по лабораторной части модулей. Оценка за выполненную работу включает в себя:

- демонстрацию работоспособности программы 35 баллов;
- объяснение теоретических вопросов 60 баллов;
- отчет 5 баллов.

## Указания к лабораторным работам

### **Лабораторная работа 1. Диаграммы взаимодействий параллельных процессов UML. (2 часа)**

#### **Цель работы**

Освоить разработку моделей многопроцессорных программ с использованием языка моделирования UML. Познакомиться с задачей описания последовательности работы взаимодействующих потоков.

#### **Формулировка задачи**

Есть список отдельных процессов, каждый из которых в основном цикле выполняется независимо от других. В некоторые моменты времени процессы вступают во взаимодействие. Необходимо спроектировать систему, свободную от проблем блокировки.

### **Лабораторная работа 2. Параллельные вычислительные алгоритмы. Технология Open MP (2 часа)**

#### **Цель работы**

Освоить разработку многопоточных программ с использованием Open MP. Познакомиться с особенностями распараллеливания вычислительных процессов.

#### **Формулировка задачи**

Есть последовательный вычислительный алгоритм. Необходимо предложить алгоритм параллельного решения поставленной задачи. Реализовать программу. Для оценки эффективности предложенного метода выполнять хронометраж последовательного и параллельного алгоритма в зависимости от размерности задачи. Построить график ускорения в зависимости от размерности.

### **Лабораторная работа 3. Параллельные матричные блочные и каскадные алгоритмы (4 часа)**

#### **Цель работы**

Освоить алгоритмы матричных блочных и каскадных вычислений на основе разработку многопоточных программ с использованием Open MP..

#### **Формулировка задачи**

Есть последовательный вычислительный алгоритм обработки данных большой размерности. Необходимо предложить алгоритм параллельного решения поставленной задачи на основе блочных или каскадных алгоритмов. Реализовать программу. Для оценки эффективности предложенного метода выполнять хронометраж последовательного и параллельного алгоритма в зависимости от размерности задачи. Построить график ускорения в зависимости от размерности.

### **Лабораторная работа.4 Распараллеливание на уровне задач, потоки (4 часа)**

#### **Цель работы**

Освоить разработку многопоточных программ с использованием Threads API.

Познакомиться с задачей синхронизации взаимодействующих потоков на основе барьеров.

#### **Формулировка задачи**

Есть список взаимодействующих процессов, каждое из которых в основном цикле выполняется независимо от других. В некоторые моменты времени, когда все задания завершили свою работу, требуется запустить новый цикл обработки. Задания могут иметь различные скорости работы, т.е. требовать при одних и тех же вычислительных ресурсах различного времени для выполнения. Так возникает задача барьерной синхронизации потоков.

### **Лабораторная работа 5. Синхронизация процессов на основе объектов ядра (2 часа)**

#### **Цель работы**

Познакомиться с задачей синхронизации взаимодействующих потоков на основе объектов ядра.

#### **Формулировка задачи**

Есть список взаимодействующих процессов, каждое из которых в основном цикле выполняется независимо от других. В некоторые моменты времени задания обмениваются данными с другими заданиями, а также используют общую память. Задания могут иметь различные скорости работы, т.е. требовать при одних и тех же вычислительных ресурсах различного времени для выполнения. Так возникает задача синхронизации потоков, которая требует анализа возможной взаимной блокировки и гонки данных. Для ее решения необходимо использовать объекты ядра Windows. У каждого потока для обеспечения правильной работы потока необходимо использовать средства синхронизации, в том числе каналы, семафоры, мьютексы, отображаемые на память файлы и критические секции.

### **Лабораторная работа 6 Паттерны программирования: Scheduler и Thread Pool (2 часа)**

### **Цель работы**

Освоить разработку эффективных многопоточных программ с использованием паттернов Scheduler и Thread Pool, познакомиться с задачей динамического распределения работы между процессорами.

### **Формулировка задачи**

Есть список неделимых заданий, каждое из которых может быть выполнено независимо от другого. Задания могут иметь различный вычислительный вес, т.е. требовать при одних и тех же вычислительных ресурсах различного времени для выполнения. Считается, что этот вес нельзя узнать, пока задание не выполнено. После того, как *все* задания из списка выполнены, появляется новый список заданий. Необходимо организовать параллельную обработку заданий на нескольких компьютерах или на нескольких процессорах одного компьютера. Количество заданий существенно превосходит количество процессоров.

Программа не должна зависеть от числа компьютеров и процессоров.

Понятно, что для распараллеливания задачи задания из списка нужно распределять между процессорами. Так как задания имеют различный вычислительный вес, а список обрабатывается итеративно, и требуется синхронизация перед каждой итерацией, то могут возникать ситуации, когда некоторые процессоры выполнили свою работу, а другие еще нет. Если ничего не предпринять, первые процессы будут простаивать в ожидании последних. Так возникает задача динамического распределения работы. Для ее решения на каждом процессоре заведем несколько потоков. Как минимум, потоков должно быть два:

- поток, который обрабатывает задания и, когда задания закончились, обращается к другим компьютерам за добавкой к работе,
- поток, ожидающий запросов о работе от других компьютеров

## **Лабораторная работа 7. Паттерны программирования: Producer - Consumer (2 часа)**

### **Цель работы**

Освоить алгоритмы параллельной работы потребителей и производителей информации.

### **Формулировка задачи**

Есть список производителей и потребителей информации. В качестве примеров решаемых задач рассмотреть задачи массовой обработки данных – системы видеонаблюдения, информационные киоски, системы мониторинга и т.п. Реализовать программу на основе паттерна Producer - Consumer. Оценить среднее время ожидания для каждого типа процессов. Для оценки эффективности предложенного метода выполнять хронометраж.

## **Лабораторная работа 8. Паттерны программирования: Read-Write Lock (2 часа)**

### **Цель работы**

Освоить алгоритмы параллельного доступа к общей памяти для записи и чтения данных.

### **Формулировка задачи**

Есть список алгоритмов –читателей и алгоритмов-писателей. В качестве примеров решаемых задач рассмотреть задачи независимой обработки поступивших заявок и генераторы таких заявок. Реализовать программу на основе паттерна Read-Write-Lock. Оценить среднее время ожидания для каждого типа процессов. Для оценки эффективности предложенного метода выполнять хронометраж. Построить графики простоя в зависимости от количества процессов каждого типа.

### **Лабораторная работа 9. Реализация распределенной системы на основе протокола TCP/IP (2 часа)**

#### **Цель работы**

Освоить технологию программирования распределенных приложений на основе интерфейса сокетов.

#### **Формулировка задачи**

В качестве примеров решаемых задач рассмотреть задачи, решаемые при выполнении лабораторной работы 5. Провести сравнительный анализ методов параллельной реализации параллельных алгоритмов в мультипрограммной и сетевой среде. Реализовать программу.

### **Лабораторная работа 10. Реализация неблокирующего сервера приложений. (2 часа)**

#### **Цель работы**

Освоить технологию программирования сервера на основе неблокирующих соединений.

#### **Формулировка задачи**

В качестве примеров решаемых задач рассмотреть задачи, решаемые при выполнении лабораторной работы 9. Провести сравнительный анализ методов реализации прикладной системы на основе парных взаимодействий между процессами и на основе сервера системы. Реализовать подсистему защиты от сбоев и восстановления системы.

### **Лабораторная работа 11. Реализация распределенных алгоритмов на основе технологии MPI (2 часа)**

#### **Цель работы**

Освоить технологию MPI. программирования кластеров.

#### **Формулировка задачи**

В качестве примеров решаемых задач рассмотреть задачи, решаемые при выполнении лабораторных работ 1 и 2. Провести сравнительный анализ методов параллельной реализации вычислительных алгоритмов с использованием open MP и MPI. Реализовать программу. Для оценки эффективности предложенного метода выполнять хронометраж последовательного и параллельного алгоритма в зависимости от размерности задачи. Построить график ускорения в зависимости от размерности.

### **Лабораторная работа 12. Реализация вычислительных алгоритмов на основе технологии CUDA и OpenCL (2 часа)**

#### **Цель работы**

Освоить технологию CUDA и OpenCL программирования суперкомпьютеров.

#### **Формулировка задачи**

В качестве примеров решаемых задач рассмотреть задачи, решаемые при выполнении лабораторных работ 1 и 2. Провести сравнительный анализ методов параллельной реализации вычислительных алгоритмов с использованием open MP и Open CL. Реализовать программу. Для оценки эффективности предложенного метода выполнять хронометраж последовательного и параллельного алгоритма в зависимости от размерности задачи. Построить график ускорения в зависимости от размерности.

## **Указания к выполнению курсового проекта**

Курсовые проекты рассматриваются как вид учебной работы по дисциплине и выполняются в пределах часов, отводимых на ее обучение. Студенты должны быть способны самостоятельно и творчески решить научно-производственные вопросы. Курсовое проектирование играет большую роль в развитии навыков самостоятельной творческой работы студентов, так как оно позволяет путем решения конкретных производственных задач приобщить студентов к практике, воспитывать их в духе ответственности за выполняемую инженерно-техническую работу, прививать им элементы научно-исследовательской работы. Работа выполняется самостоятельно и сдается в строго оговоренные сроки.

Курсовое проектирование должно способствовать закреплению, углублению и обобщению знаний, полученных студентами за время обучения, и применению этих знаний к комплексному решению конкретной инженерной задачи. Курсовой проект представляет собой результат выполнения следующих взаимосвязанных этапов:

1. Выбор темы;
2. Разработка рабочего плана, проектирование структуры программного изделия;
3. Сбор, анализ и обобщение материалов исследования, формулирование основных теоретических положений;
4. Разработка алгоритмов и программ на выбранном языке (тестирование, отладка).
5. Документирование, созданного программного продукта и оформление курсового проекта;
6. Защита.

## **Тематика курсового проектирования**

Одним из основных требований к теме курсового проектирования является ее актуальность, т.е. решение ряда взаимно связанных между собой вопросов в свете современного развития науки и практики разработки параллельного и распределенного программного обеспечения. Тематика проектов посвящена разработке и применению системного программного обеспечения, языков программирования и программных библиотек в параллельных вычислениях с использованием многоядерных процессоров и распределенных вычислительных систем. Основные направления:

- Операционные системы и среды, технологии программирования, средства и инструменты для разработки, оптимизации и отладки параллельных приложений.
- Использование ускорителей и компьютеров с нетрадиционной архитектурой (GPU, FPGA и других) в высокопроизводительных вычислениях.
- Технологии распределенной обработки данных и распределенных вычислений с использованием многопроцессорных вычислительных систем (многоядерных рабочих станций, персональных суперкомпьютеров и кластерных систем).
- GRID-технологии и GRID-системы.
- Применение средств параллельного программирования для решения прикладных задач.

Примеры тем курсовых проектов:

- Распределение вычислений в среде с высокой коммуникацией с фиксацией хода выполнения программ и передачи данных между компьютерами,
- Мониторинг работы пользователей корпоративных сетей,
- Моделирование и анализ иерархических многопроцессорных систем баз данных,
- Разработка грид-сервиса управления заданиями,

- Автоматическое распараллеливание некоторого класса программ с отображением на кластер,
- Автоматизированная разработка параллельного программного обеспечения на основе параллельных алгоритмических структур,
- Диалоговая отладка параллельных программ на основе распределенной схемы взаимодействующих компонентов,
- Методы и программные средства визуализации научных данных на массивно-параллельных вычислительных системах,
- Методы обработки запросов в системах управления базами данных для многопроцессорных систем с иерархической архитектурой,
- Оценочное тестирование распределенных программных комплексов, Расширение языков параллельного программирования для многопроцессорных и распределенных систем,
- Программный комплекс для высокопроизводительных расчетов пространственных течений,
- Суперкомпьютерные технологии в томографических исследованиях,
- Современные методы статического и динамического анализа программ для решения приоритетных проблем программной инженерии,
- Моделирование быстропротекающих динамических процессов в механике деформируемого твердого тела с использованием суперкомпьютерных технологий,
- Суперкомпьютерное моделирование в задачах физики климатической системы,
- Применение параллельных алгоритмов синтеза и распознавания речи для моделирования интерфейсов в вычислительных системах,
- Интерактивная система анализа динамической активности человека,
- Модуль обработки изображений треков частиц в высокоскоростной интеллектуальной видеокамере,
- Экспертная система с элементами параллельных вычислений.

### **Содержание курсового проектирования**

Курсовой проект должен состоять из программной части и расчетно-пояснительной записки. Изложение пояснительной записки должно быть технически грамотным, четким и сжатым. Записка должна содержать обоснование выбранной технологии и предложенной , структуры программного комплекса.

Выполнение курсового проекта включает следующие этапы:

1. Постановка задачи.
2. Анализ существующих аналогов и применяемых технологий
3. Проектирование структуры и состава системы
4. Реализация
5. Тестирование и отладка.
6. Оценка полученной системы
7. Документирование и сопровождение
8. Оценка перспектив развития проекта.

Защита проекта производится в комиссии при непосредственном участии руководителя курсового проектирования и в присутствии студентов проектирующей группы. Защита состоит в коротком докладе (8-10 минут) студента по выполненному проекту и в ответах на вопросы. Проекты после защиты должны сдаваться на кафедру.

## Литература

### А) основная литература:

1. Ногл М. TCP/IP. Иллюстрированный учебник – М.: ДМК-Пресс, 480 стр. Режим доступа: <http://e.lanbook.com/view/book/1140/page1/>

### Б) дополнительная литература:

2. Боресков А.В., Харламов А.А. Основы работы с технологией CUDA - "ДМК Пресс", 2010, 232стр. [http://e.lanbook.com/books/element.php?pl1\\_id=1260](http://e.lanbook.com/books/element.php?pl1_id=1260)Ортега Дж. Введение в параллельные и векторные методы решения линейных систем: Пер. с англ. М.: Мир, 1991. 367с.
3. Эндрюс Г.Р. Основы многопоточного, параллельного программирования. Пер. с англ. М.: Издательский дом "Вильямс", 2003. 512с.
4. Антонов А.С. "Параллельное программирование с использованием технологии OpenMP: Учебное пособие".- М.: Изд-во МГУ, 2009
5. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.:БХВ-Петербург, 2002.
6. Немнюгин С., Стесик О. Параллельное программирование для многопроцессорных вычислительных систем. – СПб.: БХВ-Петербург, 2002, 400 с.
7. Баденко В. Л. Высокопроизводительные вычисления: учеб. пособие – СПб.: Изд-во Политехн. ун-та, 2010. – 180 с.
8. Мацяшек Л.А., Лионг Б. Практическая программная инженерия на основе учебного примера. –«Бином», Лаборатория знаний, 2012, 956 стр. Режим доступа: <http://e.lanbook.com/view/book/8766/page863/>
9. Сандерс Дж., Кэндрот Э. Технология CUDA в примерах: введение в программирование графических процессоров - "ДМК Пресс", 2011, 232 стр.. Режим доступа: [http://e.lanbook.com/books/element.php?pl1\\_id=3029](http://e.lanbook.com/books/element.php?pl1_id=3029)
10. Снейдер Й. Эффективное программирование TCP/IP - изд-во "ДМК Пресс", 2009, 320 стр. . Режим доступа: [http://e.lanbook.com/books/element.php?pl1\\_id=1272](http://e.lanbook.com/books/element.php?pl1_id=1272)
11. Гергель В.П. Теория и практика параллельных вычислений. – М.:Интернет-Университет, БИНОМ. Лаборатория знаний, 2007
12. Богачёв К.Ю. Основы параллельного программирования: учебное пособие - "Бином. Лаборатория знаний, 2013, 342 стр. Режим доступа: [http://e.lanbook.com/books/element.php?pl1\\_id=42626](http://e.lanbook.com/books/element.php?pl1_id=42626)
13. Дэвис А. Асинхронное программирование в C# 5.0 – Издательство "ДМК Пресс", 2013, 120 стр. . Режим доступа: [http://e.lanbook.com/books/element.php?pl1\\_id=9132](http://e.lanbook.com/books/element.php?pl1_id=9132)
14. Энтони Уильямс. Параллельное программирование на C++ в действии. Практика разработки многопоточных программ Пер. с англ. Слинкин А.А. - : изд-во "ДМК Пресс", 2012, 672 стр. . Режим доступа: [http://e.lanbook.com/books/element.php?pl1\\_id=4813](http://e.lanbook.com/books/element.php?pl1_id=4813)

### В) Программное обеспечение и интернет-ресурсы

1. Богданов А., Мареев В., Станнова Е., Корхов В. Архитектуры и топологии многопроцессорных вычислительных систем // электронный учебник . Режим доступа: <http://www.informika.ru/text/teach/topolog/index.htm>
2. Материалы Microsoft: <http://support.microsoft.com>.
3. Материалы информационно-аналитического центра НИВЦ МГУ – [www.parallel.ru](http://www.parallel.ru).
4. Букатов А.А., Дацюк В.Н., Жегуло А.И. Программирование многопроцессорных вычислительных систем. Ростов-на-Дону. Издательство



ООО «ЦВВР», 2003, 208 с. . Режим доступа:  
<http://rsusu1.rnd.runnet.ru/tutor/method/index.html>