**Санкт-Петербургский государственный университет**

**Р А Б О Ч А Я П Р О Г Р А М М А**

**УЧЕБНОЙ ДИСЦИПЛИНЫ**

Параллельное программирование

Parallel Programming

**Язык(и) обучения**

русский

Трудоемкость в зачетных единицах: 3

Регистрационный номер рабочей программы: 027270

Санкт-Петербург

2020

**Раздел 1. Характеристики учебных занятий**

**1.1. Цели и задачи учебных занятий**

Дать обучающемуся общее представление о параллельных вычислениях: о суперсложных задачах, для которых возникла проблема ускорения вычислений, об устройстве компьютера и источниках его быстродействия, о причинах появления идеи распараллеливания вычислительного процесса и о параллельных вычислительных системах, на которых возможна реализация идеи распараллеливания, о способах построения параллельных вычислительных методов, представления их в виде алгоритмов и о реализации полученных алгоритмов в виде программ для запуска их на параллельных вычислительных системах.

Формирование понимания идеи параллельных вычислений, эффективности распараллеливания для определенных классов задач вычислительной математики, целесообразности распараллеливания в тех или иных задачах (с учетом сложности реализации алгоритмов распараллеливания, устойчивости вычислений, объема используемых ресурсов вычислительной системы, а также коммерческой стоимости их реализации), развитие навыков самостоятельного использования обучающимуия методов отбора классов задач, допускающих распараллеливание, умения самостоятельно оценить целесообразность их распараллеливание на априорно указанной реальной вычислительной системе и реализовать упомянутое распараллеливание на указанной параллельной вычислительной системе.

**1.2. Требования подготовленности обучающегося к освоению содержания учебных занятий (пререквизиты)**

Знакомство с дискретной математикой, математической логикой, теорией алгоритмов и интерфейсами распараллеливания.

**1.3. Перечень результатов обучения (learning outcomes)**

Знания:

* знание содержания дисциплины «Параллельное программирование» и обладание достаточно полным представлением о возможностях применения её разделов при разработке и программировании конкретных алгоритмов.

Умения:

* умение разрабатывать параллельные алгоритмы;
* умение находить эффективные алгоритмы при решении конкретных задач.

Навыки:

* практические навыки анализа эффективности параллельных алгоритмов при подготовке реализации на параллельных вычислительных системах;
* навык выбора того или иного алгоритма решения в зависимости от специфики задачи;
* навыками в проведении комбинаторных исследований.

Знать содержание дисциплины «параллельного программирования». Уметь формализовывать поставленные задачи и выбирать алгоритмы решения поставленных задач, обеспечивающих эффективную реализацию, учитывающую специфику задачи.

Способен преподавать математику и информатику в средней школе, специальных учебных заведениях на основе полученного фундаментального образования и научного мировоззрения

Способен использовать знания направлений развития компьютеров с традиционной (нетрадиционной) архитектурой; современных системных программных средств: операционных систем, операционных и сетевых оболочек, сервисных программ; тенденции развития функций и архитектур проблемно-ориентированных программных систем и комплексов в профессиональной деятельности

**1.4. Перечень и объём активных и интерактивных форм учебных занятий**

В качестве основных интерактивных форм (25 академических часов) предполагается проведение семинарских занятий и лекционных занятий, предполагающих активную дискуссию с преподавателем.

**Раздел 2. Организация, структура и содержание учебных занятий**

**2.1. Организация учебных занятий**

**2.1.1 Основной курс**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Трудоёмкость, объёмы учебной работы и наполняемость групп обучающихся | | | | | | | | | | | | | | | | | | |
| Код модуля в составе дисциплины,  практики и т.п. | Контактная работа обучающихся с преподавателем | | | | | | | | | | | | Самостоятельная работа | | | | Объём активных и интерактивных  форм учебных занятий | Трудоёмкость |
| лекции | семинары | консультации | практические  занятия | лабораторные работы | контрольные работы | коллоквиумы | текущий контроль | промежуточная  аттестация | итоговая аттестация | под руководством преподавателя | в присутствии  преподавателя | сам. раб. с использованием  методических материалов | текущий контроль (сам.раб.) | промежуточная аттестация (сам.раб.) | итоговая аттестация  (сам.раб.) |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | | | | | | | | | | | | | |
| Форма обучения: очная | | | | | | | | | | | | | | | | | | |
| Семестр 2 | 15 | 15 | 2 |  |  |  |  |  | 2 |  |  |  | 44 |  | 30 |  | 25 | 3 |
|  | 2-100 | 2-30 | 2-100 |  |  |  |  |  | 2-100 |  |  |  | 1-1 |  | 1-1 |  |  |  |
| ИТОГО | 15 | 15 | 2 |  |  |  |  |  | 2 |  |  |  | 44 |  | 30 |  |  | 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Виды, формы и сроки текущего контроля успеваемости и промежуточной аттестации | | | | | | |
| Код модуля в составе дисциплины, практики и т.п. | Формы текущего контроля успеваемости | | Виды промежуточной аттестации | | Виды итоговой аттестации  (только для программ итоговой аттестации и дополнительных образовательных программ) | |
| Формы | Сроки | Виды | Сроки | Виды | Сроки |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | |
| Форма обучения: очная | | | | | | |
| Семестр 2 |  |  | экзамен, устно, традиционная форма | по графику промежуточной аттестации |  |  |

**2.2. Структура и содержание учебных занятий**

**Модуль 1**

1. Общий взгляд на высокопроизводительные системы начиная от расширений SSE, через многоядерность к узлам кластеров. Обзор технологий многопоточного программирования: Posix threads, Java threads, Boost threads, OpenMP, Intel TBB, Java.util.concurrent, Fork/Join Framework.
2. Начало работы с потоками. Отличия процессов и потоков. Механизм запуска потока. Необходимость синхронизации. Реализация примитивов синхронизации: алгоритмы Петерсона и Лампорта. Примитивы синхронизации POSIX: мьютексы.
3. Организация завершения потоков. Условные переменные. Boost-потоки. Корректное завершение потоков: cancellation points, InterruptedException. Сравнение синтаксиса: POSIX, Boost, Java.
4. Синхронизация. Корректный захват и освобождение примитивов через замки (locks). Виды мьютексов. Организация пула потоков. Ограничения на реализацию потокобезопасных контейнеров.
5. Виды синхронизации. CAS-операции и атомики. Альтернативный подход: Software transactional memory. Виды алгоритмов реализации потокобезопасных контейнеров.
6. lock-free контейнеры. Виды алгоритмов реализации потокобезопасных контейнеров. Сложные ошибки синхронизации: потерянный сигнал, ABA.
7. Поиск ошибок и анализ производительности. Сложные ошибки синхронизации: fork в параллельных приложениях, инверсия приоритетов. Средства профилирования и поиска ошибок: valgrind.
8. Шаблоны параллельных программирования. Структурные шаблоны: Декомпозиция по задачам, Геометрическая декомпозиция, Recursive Data, Pipeline. Некоторые программные структуры: SPMD, Parallel loops, Boss/Worker.
9. OpenMP и Intel TBB. Стандарт OpenMP. Обзор Intel TBB: алгоритмы, аллокаторы, деревья задач, планирование. TLS (Thread Local Storage).
10. Консенсус. Атомарный регистр. Консенсусное число. Классификация примитивов синхронизации по консенсусному числу. lock-free алгоритм атомарного снимка регистров.
11. Кластерные вычисления. Классификация кластеров. Обзор стандарта MPI: средства запуска приложений, обмена сообщения, редукции по данным.
12. Wait-free алгоритмы. Отличия lock-и wait-free алгоритмов. wait-free алгоритм атомарного снимка регистров.
13. Моделирование параллельных систем. Сети Петри.

**Раздел 3. Обеспечение учебных занятий**

**3.1. Методическое обеспечение**

**3.1.1 Методические указания по освоению дисциплины**

Успешное освоение дисциплины возможно благодаря посещению лекций, участию в обсуждении рассматриваемых вопросов, самостоятельной работе, включающей в себя чтение специальной литературы по разделам темы.

**3.1.2 Методическое обеспечение самостоятельной работы**

Самостоятельная работа обучающихся в рамках данной дисциплины является важным компонентом обучения, предусмотренным компетентностно-ориентированным учебным планом и рабочей программой учебной дисциплины.

Настоящей программой предусмотрены формы самостоятельной работы с использованием методических материалов по тематике курса и источников, указанных в обязательной и дополнительной литературе, указанных в данной программе.

**3.1.3 Методика проведения текущего контроля успеваемости и промежуточной аттестации и критерии оценивания**

*Устный экзамен.*

Оценка ставится за ответы на вопросы билета и активное выступления с докладами.

В билете два вопроса.

Отлично A – сделано два доклада с презентацией, уровень докладов показывает отличное владение материалом и на оба вопроса билета даны правильные ответы.

Хорошо B – сделано два доклада, уровень докладов показывает очень хорошее владение материалом, на оба вопроса билета даны правильные ответы с незначительными погрешностями.

Хорошо C – сделано два доклада, уровень докладов показывает хорошее владение материалом, на оба вопроса билета даны правильные ответы со значительными погрешностями.

Удовлетворительно D – сделано два доклада, уровень докладов показывает удовлетворительное владение материалом, дан правильный ответ на один вопрос.

Удовлетворительно E – сделан один доклада, уровень доклада показывает посредственное владение материалом, дан правильный ответ на один вопрос.

F – не сделано ни одного доклада.

**3.1.4 Методические материалы для проведения текущего контроля успеваемости и промежуточной аттестации (контрольно-измерительные материалы, оценочные средства)**

*Пример списка вопросов к устному экзамену:*

1. Общий взгляд на высокопроизводительные системы начиная от расширений SSE, через многоядерность к узлам кластеров. Обзор технологий многопоточного программирования: Posix threads, Java threads, Boost threads, OpenMP, Intel TBB, Java.util.concurrent, Fork/Join Framework.
2. Начало работы с потоками. Отличия процессов и потоков. Механизм запуска потока. Необходимость синхронизации. Реализация примитивов синхронизации: алгоритмы Петерсона и Лампорта. Примитивы синхронизации POSIX: мьютексы.
3. Организация завершения потоков. Условные переменные. Boost-потоки. Корректное завершение потоков: cancellation points, InterruptedException. Сравнение синтаксиса: POSIX, Boost, Java.
4. Синхронизация. Корректный захват и освобождение примитивов через замки (locks). Виды мьютексов. Организация пула потоков. Ограничения на реализацию потокобезопасных контейнеров.
5. Виды синхронизации. CAS-операции и атомики. Альтернативный подход: Software transactional memory. Виды алгоритмов реализации потокобезопасных контейнеров.
6. lock-free контейнеры. Виды алгоритмов реализации потокобезопасных контейнеров. Сложные ошибки синхронизации: потерянный сигнал, ABA.
7. Поиск ошибок и анализ производительности. Сложные ошибки синхронизации: fork в параллельных приложениях, инверсия приоритетов. Средства профилирования и поиска ошибок: valgrind.
8. Шаблоны параллельных программирования. Структурные шаблоны: Декомпозиция по задачам, Геометрическая декомпозиция, Recursive Data, Pipeline. Некоторые программные структуры: SPMD, Parallel loops, Boss/Worker.
9. OpenMP и Intel TBB. Стандарт OpenMP. Обзор Intel TBB: алгоритмы, аллокаторы, деревья задач, планирование. TLS (Thread Local Storage).
10. Консенсус. Атомарный регистр. Консенсусное число. Классификация примитивов синхронизации по консенсусному числу. lock-free алгоритм атомарного снимка регистров.
11. Кластерные вычисления. Классификация кластеров. Обзор стандарта MPI: средства запуска приложений, обмена сообщения, редукции по данным.
12. Wait-free алгоритмы. Отличия lock-и wait-free алгоритмов. wait-free алгоритм атомарного снимка регистров.
13. Моделирование параллельных систем. Сети Петри.

**3.1.5 Методические материалы для оценки обучающимися содержания и качества учебного процесса**

Для оценки обучающимися содержания и качества учебного процесса применяется анкетирование в соответствии с методикой и графиком, утвержденными в установленном порядке.

**3.2. Кадровое обеспечение**

**3.2.1 Образование и (или) квалификация штатных преподавателей и иных лиц, допущенных к проведению учебных занятий**

К проведению занятий привлекаются преподаватели, имеющие базовое образование и/или ученую степень.

**3.2.2 Обеспечение учебно-вспомогательным и (или) иным персоналом**

Не требуется.

**3.3. Материально-техническое обеспечение**

**3.3.1 Характеристики аудиторий (помещений, мест) для проведения занятий**

В аудиториях, где проводятся занятия, необходимо наличие досок и средств письма на них.

**3.3.2 Характеристики аудиторного оборудования, в том числе неспециализированного компьютерного оборудования и программного обеспечения общего пользования**

Специальных требований нет.

**3.3.3 Характеристики специализированного оборудования**

Специальных требований нет.

**3.3.4 Характеристики специализированного программного обеспечения**

Специальных требований нет.

**3.3.5 Перечень и объёмы требуемых расходных материалов**

Фломастеры цветные для доски, губки.

**3.4. Информационное обеспечение**

**3.4.1 Список обязательной литературы**

1. Бурова И.Г., Демьянович Ю.К. Алгоритмы параллельных вычислений. СПб.: Издательство Санкт-Петербургского университета, 2010.
2. Амосов А.А., Дубинский Ю.А., Копченова Н.В. Вычислительные методы, 2014, Издательство «Лань».
3. Даугавет И.К. Введение в классическую теорию приближения функций, 2011, Издательство СПбГУ.
4. Саад, Юсеф. Итерационные методы для разряженных линейных систем (в двух томах). Изд-тво Московского университета, 2013, Т. 1.

**3.4.2 Список дополнительной литературы**

1. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб.: Изд-во БХВ-Петербург, 2002.

**3.4.3 Перечень иных информационных источников**

Не требуются.

**Раздел 4. Разработчики программы**

Демьянович Юрий Казимирович, профессор математико-механического факультета СПбГУ. Yuri.Demjanovich@gmail.com, тел. 428-73-19.

Бурова Ирина Герасимовна, профессор математико-механического факультета СПбГУ. burovaig@mail.ru, тел. 341-98-19