

## АННОТАЦИЯ К РАБОЧЕЙ ПРОГРАММЕ ДИСЦИПЛИНЫ «Программирование параллельных процессов»

по основной профессиональной образовательной программе по направлению подготовки 09.04.04 «Программная инженерия» (уровень магистратуры)

**Направленность (профиль):** Разработка программно-информационных систем

**Трудоемкость дисциплины** – 4 з.е. (144 часов)

**Форма промежуточной аттестации** – Экзамен.

**В результате освоения дисциплины обучающийся должен обладать следующими компетенциями:**

- ПК-10 Владение навыками программной реализации систем с параллельной обработкой данных и высокопроизводительных систем;
- ПК-11 Владение навыками организации промышленного тестирования создаваемого программного обеспечения.

**Содержание дисциплины:**

Дисциплина «Программирование параллельных процессов» включает в себя следующие разделы:

**Форма обучения очная. Семестр 1.**

**1. Введение в параллельные вычисления.** Методы и средства параллельной обработки информации: параллельные вычислительные методы, параллельные вычислительные системы, параллельное программирование. Распараллеливание на уровне задач, на уровне передачи сообщений, на уровне разделяемой памяти, функциональная декомпозиция, декомпозиция по данным, геометрический параллелизм. Эффективность параллельных вычислений, закон Амдала. Блокировка, гонки данных, синхронизация. Основные понятия унифицированного языка моделирования (UML) для представления взаимодействующих процессов. Конвейерные алгоритмы. Блочные алгоритмы умножения матриц. Метод сдвигания. Чет-нечетная сортировка. Каскадные алгоритмы.

**2. Технология разработки параллельной программы на основе стандарта OpenMP.** Параллельное программирование в системах с общей памятью на основе OpenMP. Основные директивы Open MP. Балансировка нагрузки. Циклы и секции. Планирование. Примеры правильного и ошибочного кода. Отладка программ.

**3. Объекты ядра Windows, обеспечивающие параллелизм.** Создание и завершение процессов и потоков. Планирование потоков, приоритет, привязка к процессорам. Барьеры. Реализация базовых механизмов в ядре ОС. Однопроцессорное ядро. Многопроцессорное ядро. Реализация семафоров в ядре. Реализация мониторов в ядре. Реализация мониторов с помощью семафоров. Файлы, отображаемые на память, мьютексы, мониторы, условные переменные. Синхронизация потоков с использованием объектов ядра, критические секции, взаимное исключение. Барьеры и сигнализирующие события. Производители и потребители. Кольцевые буферы. Задача об обедающих философах. Задача о читателях и писателях. Распределение ресурсов и планирование.

**4. Проектирование распределенных приложений, основанных на передаче сообщений.** Способы построения и реализации компонентов распределенной системы. Обсуждаются синхронный и асинхронный подход к взаимодействию подсистем. Асинхронная передача сообщений. Клиенты и серверы. Обмен данными. Синхронная передача сообщений. Примеры взаимодействий типа "клиент-сервер". Сервер приложений. Блокирующий и неблокирующий сервер. Распределенный портфель задач. Алгоритмы типа "зонд-эхо". Алгоритмы рассылки. Логические часы и упорядочение событий. Алгоритмы передачи маркера. Распределенный таймер. Распределенные семафор и распределенная критическая секция.

**5. Технология разработки параллельной программы на основе стандарта MPI.** Параллельное программирование в системе MPI (Message Passing Interface). Основные

термины MPI. Коммуникатор как контекст взаимодействия процессов с помощью функций MPI. Структура MPI-программы. Типы данных и производные типы. Типы функций: функции инициализации и закрытия MPI процессов; функции, реализующие коммуникационные операции типа точка-точка (point-to-point); функции, реализующие коллективные операции (collective); функции для работы с группами процессов и коммуникаторами; функции для работы со структурами данных; функции формирования топологии процессов. Виртуальные топологии.

**6. Паттерны проектирования конкурирующих процессов.** Однопоточное выполнение (Single Threaded execution), Активный объект (Active Object), Объект блокировки (Lock Object), Охраняемая приостановка (Guarded Suspension), Отмена (Balking), Планировщик (Scheduler), Блокировка чтения - записи (Read-Write Lock), Асинхронная обработка (Asynchronous Processing), Производитель - потребитель (Producer - Consumer), Двойная буферизация (Double Buffering), (Half Sync-Async), Пул потоков (Thread Pool).

Разработал:  
профессор  
кафедры ПМ  
Проверил:  
Декан ФИТ



Е.Н. Крючкова

А.С. Авдеев