Многопоточность, синхронизация, атомарные операции

1. Разработать программу выполняющую параллельное применение цветокоррекции к изображению. Сравнить производительность последовательной и параллельной версии.
2. Реализовать структуру данных дерево с параллельным поиском значения. Разработать версию программы с использованием ForkJoinPool
3. Разработать структуру данных неблокирующий стек. Обеспечить потокобезопасное выполнение операций.
4. Разработать структуру данных неблокирующая очередь. Обеспечить потокобезопасное выполнение операций.
5. Реализовать алгоритм балансировки поступающих задач между исполнителями. Из общей очереди вытаскивается задача и отправляется в очередь того исполнителя, которая окажется наименее заполненной. Если все очереди заполнены выше определенного предела то ждать, пока они не освободятся. Минимизировать количество блокировок и примитивов синхронизации.
6. Разработать свою реализацию концепции Future - Promise, с возможностью подписки слушателя на Future и неблокирующей композиции Future.
7. Смоделировать поиск и поглощение ресурсов в двумерном мире несколькими потоками. Проанализировать распределение поглощенных ресурсов в зависимости от количества потоков, количества ядер процессора, приоритетов потоков.
8. Разработать программу реализующую параллельное перемножение матриц. Сравнить быстродействие с однопоточным исполнением
9. Разработайте собственную реализацию CountdownLatch на основе wait-notify
10. Разработать критическую секцию, обеспечивающую доступ к ресурсу таким образом, чтобы ожидающие потоки помещались в стек, то есть после получения освобождения ресурса его получал поток, который затребовал ресурс последним. При превышении определенного размера стека, наиболее старые потоки должны прерываться с ошибкой. Допустимо вместо остановки потока продолжать исполнение кода в виде обратного вызова из другого потока.
11. Разработать критическую секцию, которая бы создавала копию разделяемого ресурса при превышении определенного количества ожидающих потоков и распределяла очереди между копиями
12. Разработать алгоритм параллельной сортировки массива.
13. Реализовать карточную игру "Пьяница", в которую играло бы несколько компьютерных игроков. Каждый игрок - отдельный поток.
14. Разработать двоичное дерево поиска, в котором каждая операция выполнялась бы в собственном потоке.
15. Реализовать мьютекс на основе спин-лока, сравнить производительность с обычным мьютексом.
16. Разработать шину событий, выполняющую обработку событий многопоточно.

Потоки ввода-вывода, сериализация, NetworkIO

1. Разработать приложение, передающие текстовые данные между двумя приложениями в сжатом виде. Предусмотреть возможность использования стандартных потоков ввода и pipe-ов.
2. Разработать приложение воспроизводящее звуковой файл с сервера, использовать протокол TCP/IP и сжатие данных (Клуб)
3. Разработать приложение воспроизводящее звуковой файл с сервера, использовать протокол UDP
4. Разработать децентрализованный чат. То есть в приложении не должно быть единого сервера. IP-адреса всех участников (или подсеть с участниками) известны заранее.
5. Разработать сетевую игру, в которой несколько участников бегают по лабиринту в виде прямоугольников различного цвета.
6. Разработать приложение аналог netcat с возможностью работы как в текстовом так и в hex-режиме (передача бинарных данных). Предусмотреть поддержку шифрования данных.
7. Разработать приложение для сканирования подсети
8. Разработать приложение-сниффер, логирующее все сообщения по определенному порту.
9. Реализовать многопользовательский чат с поддержкой шифрования по произвольному протоколу. (Кузнецов)
10. Реализовать сервер-экран, защищающий от DOS-атак. Сервер должен одновременно обрабатывать не более N запросов, остальные ставить в очередь, при превышении размеров очереди - сбрасывать соединение (Шаргородская).
11. Разработать приложение, позволяющее запускать некоторый процесс одновременно на нескольких машинах одной командой. И выводящее результат работы всех этих процессов. (Черепанов)
12. Разработать приложение распределяющее задачу поиска объекта на изображении на несколько машин.
13. Разработать приложение распределяющее задачу подбора пароля к архиву на несколько машин
14. Разработать сетевую игру для угадывания чисел на скорость
15. Разработать приложение для передачи файлов по сети, сравнить работу по протоколу TCP и по протоколу UDP

Сервлеты, Основы http, сессия

1. Разработать программу-файловый менеджер, позволяющую просматривать файлы и структуру каталогов на сервере
2. Разработать систему авторизации пользователей с проверкой паролей по хеш-функции и поддержкой функции “запомнить меня”
3. Разработать приложение-сервер, запускающее сторонние программы для генерации http-ответа наподобие CGI
4. Реализовать программу-файловое хранилище, которая бы не хранила одинаковые файлы. Предусмотреть возможность хранения публичных и личных файлов пользователя.
5. Реализовать веб-игру крестики-нолики для двух пользователей.
6. Написать приложение “гостевая книга”.
7. Разработать приложение для прохождения тестов по заранее подготовленным вопросам и вариантам ответов.
8. Разработать приложение для мониторинга состояния сервера. Показывать вывод команды `top`, `df`, `iostat`. Обеспечивать доступ только авторизованным пользователям.
9. Разработать приложение “калькулятор валют”. Информацию о курсах валюты брать из открытых источников в сети Интернет.
10. Разработать приложение “список задач”. Каждый пользователь имеет свой список, но также может разделять задачи с другими пользователями.
11. Разработать приложение-фотоальбом. Предоставить пользователям возможность просмотра фотоальбома в режиме миниатюр.
12. При помощи средств HTML5 разработать приложение для прослушивания аудио-коллекций. Предусмотреть возможность конвертации неподдерживаемых браузерами аудиоформатов.
13. Разработать сервис `pastebin`-подобный сервис с сохранением истории изменений. Допустимо использовать утилиты `diff` и `patch`.
14. Разработать приложение передающее файл другому пользователю без сохранения его на сервере. Сервис может работать аналогично сервису `jetbytes.com`
15. Разработать приложение, позволяющее редактировать `html`-файлы на сервере в заданной папке и просматривать их.

Базы данных

1. Создать мини-форум. Сохранять пользователей форума, темы и сообщения. В одной теме на странице выводить не более десяти сообщений. Предусмотреть возможность поиска всех сообщений пользователя.
2. Разработать приложение “учетную книгу”. Каждый пользователь имеет право забирать и возвращать со склада оборудование, при этом отмечаясь в учетной книге. Предусмотреть возможность отслеживания какое оборудование у кого находится. Историю использования оборудования пользователями. Обеспечить поддержку транзакций.
3. База данных для ведения личных дел сотрудников: список сотрудников с их характеристиками, журнал продвижения по службе (должность, срок вступления в должность, оклад, дата поступления на работу, дата увольнения и т.д.). Предусмотреть возможность выполнения типичных запросов: личный состав на текущий момент времени, количество сотрудников с заданным окладом или выше, средний оклад, средняя численность

4. Разработать приложение, позволяющее просматривать через web-интерфейс структуру и содержимое произвольной базы данных Derby.
5. Учет отпуска блюд и количества продуктов на складе (сложный товар, комплект). Существует меню готовых блюд, изготовление каждого из которых, естественно, приводит к уменьшению количества продуктов на складе. Предусмотреть возможность по готовому плану выпуска блюд и раскладке продуктов для каждого блюда, создать заказ на необходимые для этого продукты.
6. Разработать веб-приложение для хранения отзывов о прочитанных книгах. Отслеживать рейтинги книг и поиск отзывов определенного посетителя.
7. Разработать базу данных музыкальных композиций. Обеспечить поиск по названию композиции, альбому, группе, году записи, части названия.
8. Реализовать средства для выгрузки содержимого базы данных в различных форматах: SQL, JSON, XML через Web-интерфейс.
9. Разработать веб-интерфейс для каталога товаров. Каталог имеет древовидную структуру. Товары имеют название, производителя, стоимость и описание.
10. Разработать приложение для хранения адресов домов и улиц. Создать отдельные таблицы для домов, улиц, районов, организаций в отдельных домах.
11. Разработать веб интерфейс для базы знаний. В базе данных хранить отдельно сущности отдельно связи между ними. Обеспечить вывод всех утверждений относительно определенной сущности, всех утверждений определенного типа.
12. Разработать базу данных для отслеживания активности пользователя на сайте. Отслеживать когда пользователь заходил на сайт, какие действия совершал. Отдельно хранить базу данных пользователей.
13. Создать базу данных туристического агентства . Имеется определенное количество маршрутов, описание маршрута. Турагентство обеспечивает клиентов путевками, общая стоимость которых формируется из совокупности предоставляемых услуг: проезд (билеты , проживание в гостиницах, экскурсионное обслуживание и т.д.). Создать справочник предлагаемых услуг с расценками, а на основе выбранных услуг формировать путёвку. Вести учет реализованных путевок.
14. Создать базу данных для учета клиентов дома отдыха. Создать таблицы: справочник список клиентов, служебные таблицы, таблицу журнал - учитывающую приезды и убытие клиентов. Предусмотреть возможность выполнения запросов для получения информации о клиентах на данный момент времени и т.д.
15. Создать базу данных проведения конференций в экспоцентре. Отслеживать залы, в которых проводятся мероприятия, компании, предоставляющие услуги, посетителей.
16. Разработать базу данных телефонов клиентов call-центра, сохранять историю звонков, статус клиента.