



УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники
Системное программное обеспечение

Лабораторная работа №3
Вариант №6: Library catalog

Преподаватель: Кореньков Юрий Дмитриевич
Выполнил: Кульбако Артемий Юрьевич Р33112

Цель работы

Цель: изучение способов взаимодействия между сетевыми службами низкого уровня в асинхронном режиме.

Описание работы:

Разработать клиент-серверное приложение. Для организации взаимодействия по сети, поддержки множества соединений использовать программные интерфейсы (API) операционной системы.

Сервер и клиенты взаимодействуют по протоколу, реализованному на базе сокетов (использовать TCP, если в варианте задания не указано обратное). Сервер должен поддерживать условно неограниченное количество клиентов. На всех этапах взаимодействия клиента и сервера должна быть предусмотрена обработка данных независимо от их размера.

Порядок выполнения:

1. Выполнить анализ предметной области, которая задается вариантом к лабораторной работе. Результатом анализа должен быть набор сущностей, которые будут в качестве элементов данных (типов, структур, операций) и/или составных частей программной архитектуры (компонентов, модулей) при реализации программы.
2. Составить диаграмму, на которой схематически будут показаны результаты анализа: сущности, их атрибуты и взаимосвязи.
3. Составить план постепенного выполнения задания: какие части функциональности, в каком порядке предполагается реализовывать, в каком порядке и как проверять их работоспособность.
4. Загрузить все полученные артефакты в отдельную директорию «docs» репозитория, в корневую директорию положить readme.md с номером варианта и кратким описанием.
5. Продемонстрировать составленные диаграмму и план преподавателю. Для этого достаточно просто отправить преподавателю ссылку на репозиторий.

6. После проверки и получения рекомендаций приступить к реализации программы, создавая на каждый этап выполнения отдельную ветку в репозитории.
7. По завершении каждого этапа создать pull-request, включив преподавателя в число reviewer-ов.
8. Если pull-request отклоняется преподавателем, выполнить необходимые правки и обновить его, запросив повторное ревью.
9. Когда pull-request одобрен, «слить» его с основной веткой кода, после чего создать новую ветку для работы над следующим этапом.

Library catalog:

Программа может выполняться в двух режимах: сервер или клиент. Режим определяется аргументом командной строки. Завершение программы-сервера происходит по нажатию ключевой клавиши (например, Q).

При запуске в режиме клиента через аргументы командной строки задаётся имя пользователя и адрес сервера.

Каждая книга описывается карточкой, содержащей сведения о названии, авторе, дате издания, аннотации, и наборе тэгов (таких как жанры и т.п.), количество приписанных к каталогу и количество в наличии. Каждый подключающийся пользователь может помещать книги в каталог, брать на прочтение и возвращать. Предусмотреть поиск сведений о книгах по различным критериям. Если описание книги редактируется или изменяется статус наличия, пользователям должна своевременно показываться актуальная информация по мере её изменения. Визуально любые операции редактирования осуществляются «in-place», то есть по месту визуализации данных поля.

Исходный код



<https://gitlab.se.ifmo.ru/testpassword/system-software/-/tree/master/lab3-12.06.21>

Выводы

Сложность работы заключается в отсутствии готовых, высокоуровневых средств для передачи данных по сети (да, я понимаю, что это С, но всё же...), а также в создании пользовательского интерфейса, который хоть сколько бы то был удобен. Похожее по функциональности задание уже было на 1-ом курсе, но эти самостоятельные реализации того, что в высокоуровневых языках воспринимается как данность, в очередной раз служит напоминанием о том, как постарались предки, чтобы мы не испытывали всю эту боль, написав удобные библиотеки, языки, фреймворки и средства разработки.