

Задания по курсовым проектам по дисциплине
«Программирование сетевых приложений»
 для направления специальности
 1-40 05 01-01 «Информационные системы и технологии
 (в проектировании и производстве)»

1. Задания к курсовому проекту

№ п/п	Тема проекта	Исходные данные к проекту	ФИО, дата, подпись
1	Клиент-серверная система обработки web-запросов на решение СЛАУ	<p>Разработать сетевое приложение, обрабатывающее запросы клиентов по протоколу HTTP на решение СЛАУ. Клиентская часть отправляет на сервер в формате JSON матрицу системы и столбец свободных членов, сервер принимает данные и осуществляет решение СЛАУ методом Зейделя, а затем найденное решение возвращает клиенту.</p> <p>Предусмотреть возможность решения систем больших размерностей (максимальное количество неизвестных – 50000). Сравнить найденное решение с решением по методу Гаусса. Разработать модульные и нагрузочные тесты.</p>	
2	Сетевое приложение на основе RESTful архитектуры для решения СЛАУ	<p>Разработать сетевое web-приложение, обрабатывающее запросы клиентов на решение СЛАУ. Клиентская часть отправляет на сервер в формате XML матрицу системы и столбец свободных членов, сервер принимает данные и осуществляет решение СЛАУ методом Холесского, а затем найденное решение возвращает клиенту.</p> <p>Предусмотреть возможность решения систем больших размерностей (максимальное количество неизвестных – 50000). Сравнить найденное решение с решением по методу Гаусса. Разработать модульные и нагрузочные тесты.</p>	
3	Клиент-серверное приложение обработки AJAX-запросов на решение СЛАУ	<p>Разработать сетевое web-приложение, обрабатывающее AJAX запросы клиентов на решение СЛАУ. Клиентская часть отправляет на сервер матрицу системы и столбец свободных членов, сервер принимает данные и осуществляет решение СЛАУ методом сопряжённых градиентов, а затем найденное решение возвращает клиенту.</p> <p>Предусмотреть возможность решения систем больших размерностей (максимальное количество неизвестных – 50000). Сравнить найденное решение с решением по методу Гаусса. Разработать модульные и нагрузочные тесты.</p>	

№ п/п	Тема проекта	Исходные данные к проекту	ФИО, дата, подпись
4	Сетевое приложение для распределённого решения СЛАУ с использованием протокола TCP	Разработать программу, осуществляющую распределённое решение СЛАУ в ЛВС средствами протокола TCP на основе метода Гаусса с циклическим размещением по столбцам. В качестве исходных данных использовать: матрицу системы, столбец свободных членов, файл с перечнем счётных узлов. Максимальное количество неизвестных – 50000. Сравнить реализованный вариант метода по скорости нахождения решения с линейным вариантом метода Гаусса. Разработать модульные и нагрузочные тесты.	
5	Сетевое приложение для распределённого решения СЛАУ с использованием протокола UDP	Разработать программу, осуществляющую распределённое решение СЛАУ в ЛВС средствами протокола UDP на основе метода Гаусса с клеточным размещением. В качестве исходных данных использовать: матрицу системы, столбец свободных членов, файл с перечнем счётных узлов. Максимальное количество неизвестных – 50000. Сравнить реализованный вариант метода по скорости нахождения решения с линейным вариантом метода Гаусса. Разработать модульные и нагрузочные тесты.	
6	Сетевое приложение для распределённого решения СЛАУ с использованием протокола HTTP	Разработать программу, осуществляющую распределённое решение СЛАУ средствами протокола HTTP на основе метода Гаусса с циклическим размещением по строкам. В качестве исходных данных использовать: матрицу системы, столбец свободных членов, файл с перечнем счётных узлов. Максимальное количество неизвестных – 50000. Сравнить реализованный вариант метода по скорости нахождения решения с линейным вариантом метода Гаусса. Разработать модульные и нагрузочные тесты.	
7	Система планирования заданий при сериализуемой обработке в ЛВС с использованием протокола TCP	Разработать программу, осуществляющую управление сериализуемой обработкой в ЛВС на примере вычисления градиентного изображения путем сравнения соответствующих выборок пикселей, взятых из двух половин окна с использованием t -теста Стьюдента. Максимальное количество обрабатываемых изображений 1000. Сравнить реализованный вариант по скорости нахождения решения с его линейным вариантом. Разработать модульные и нагрузочные тесты.	
8	Система планирования заданий при сериализуемой обработке в ЛВС с использованием протокола UDP	Разработать программу, осуществляющую управление сериализуемой обработкой в ЛВС на примере вычисления гауссовского размытия изображения. Максимальное количество обрабатываемых изображений 1000. Сравнить реализованный вариант по скорости нахождения решения с его линейным вариантом. Разработать модульные и нагрузочные тесты.	

№ п/п	Тема проекта	Исходные данные к проекту	ФИО, дата, подпись
9	Сетевое приложение с распределённой обработки использованием протокола FTP	<p>Разработать программу, осуществляющую управление распределённой обработкой по запросам клиентов. Запрос клиента представляет собой тестовый файл, отправляемый по протоколу FTP, ответ сервера – сколько раз встречается каждая буква в тексте.</p> <p>Сервер должен распределять вычисления на доступный ему вычислительный кластер посредством протокола HTTP.</p> <p>Сравнить реализованный вариант по скорости нахождения решения с его линейным вариантом.</p> <p>Разработать модульные и нагрузочные тесты.</p>	
10	Сетевое приложение с распределённой обработки использованием протокола TCP	<p>Разработать программу, осуществляющую управление распределённой обработкой по запросам клиентов. Запрос клиента представляет собой изображение, отправляемое по протоколу TCP, ответ сервера – размытое изображение с использованием t-теста Стьюдента.</p> <p>Используя метод скользящего окна, сервер должен распределять вычисления на доступный ему вычислительный кластер посредством протокола TCP.</p> <p>Сравнить реализованный вариант по скорости нахождения решения с его линейным вариантом.</p> <p>Разработать модульные и нагрузочные тесты.</p>	
11	Клиент серверное приложение обработки изображений	<p>Разработать программу, осуществляющую управление распределённой обработкой по запросам клиентов. Запрос клиента представляет собой изображение, отправляемое с использованием WEB-socket, ответ сервера – сглаженное средним изображение.</p> <p>Используя метод скользящего окна, сервер должен распределять вычисления на доступный ему вычислительный кластер посредством протокола TCP.</p> <p>Сравнить реализованный вариант по скорости нахождения решения с его линейным вариантом.</p> <p>Разработать модульные и нагрузочные тесты.</p>	
12	Организация параллельной обработки изображений в ЛВС с использованием протокола TCP	<p>Разработать программу, осуществляющую управление параллельной обработкой в ЛВС на примере вычисления сглаженного средним изображения.</p> <p>Сравнить реализованный вариант по скорости нахождения решения с его линейным вариантом.</p> <p>Разработать модульные и нагрузочные тесты.</p>	
13	Клиент-серверная система обработки запросов на решение СЛАУ посредством web-socket	<p>Разработать сетевое приложение, обрабатывающее запросы клиентов через web-socket на решение СЛАУ.</p> <p>Клиентская часть отправляет на сервер матрицу системы и столбец свободных членов, сервер принимает данные и осуществляет решение СЛАУ методом LL^T-разложения, а затем найденное решение возвращает клиенту.</p> <p>Предусмотреть возможность решения систем больших размерностей (максимальное количество неизвестных – 50000). Сравнить найденное решение с решением по методу Гаусса. Разработать модульные и нагрузочные тесты.</p>	

№ п/п	Тема проекта	Исходные данные к проекту	ФИО, дата, подпись
14	Многопользовательский чат на основе протокола TCP	<p>Разработать сетевое приложение, реализующее многопользовательский чат на основе протокола TCP. Реализовать систему аутентификации пользователей и ролей. Предусмотреть возможность хранения истории сообщений каждого пользователя. При этом администратор может просмотреть историю сообщений любого из пользователей, а пользователи могут видеть только свои сообщения. Обеспечить возможность передачи в чат не только тестовых сообщений, но и изображений.</p> <p>Разработать модульные и нагрузочные тесты.</p>	
15	Многопользовательский чат на основе протокола UDP	<p>Разработать сетевое приложение, реализующее многопользовательский чат на основе протокола UDP. Реализовать систему аутентификации пользователей и ролей. Предусмотреть возможность хранения истории сообщений каждого пользователя. При этом администратор может просмотреть историю сообщений любого из пользователей, а пользователи могут видеть только свои сообщения. Реализовать механизм как общих, так и личных сообщений. Разработать модульные и нагрузочные тесты.</p>	
16	Клиент-серверное приложение, реализующее многопользовательский чат на основе web-socket	<p>Разработать сетевое приложение, реализующее многопользовательский чат на основе web-socket. Реализовать систему аутентификации пользователей и ролей. Предусмотреть возможность хранения истории сообщений каждого пользователя. При этом администратор может просмотреть историю сообщений любого из пользователей, а пользователи могут видеть только свои сообщения. На сервере запросы от клиентов обрабатывать посредством пула потоков.</p> <p>Разработать модульные и нагрузочные тесты.</p>	
17	Клиент-серверное приложение, реализующее многопользовательский чат на основе протокола HTTP	<p>Разработать сетевое приложение, реализующее многопользовательский чат на основе протокола HTTP. Реализовать систему аутентификации пользователей и ролей. Предусмотреть возможность хранения истории сообщений каждого пользователя. При этом администратор может просмотреть историю сообщений любого из пользователей, а пользователи могут видеть только свои сообщения. Реализовать 2 типа клиентов: десктопное приложение и web-приложение.</p> <p>Разработать модульные и нагрузочные тесты.</p>	
18	Многопользовательское удалённое файловое хранилище на основе протокола TCP	<p>Разработать сетевое приложение, реализующее многопользовательское удалённое файловое хранилище на основе протокола TCP, реализующее операции загрузки, выгрузки, удаления файлов и переименования файлов. Предусмотреть возможность создания каталогов. Реализовать систему аутентификации пользователей и ролей. Предусмотреть возможность как персонализированного доступа к файлам, так и общего с разграничением прав доступа.</p> <p>Разработать модульные и нагрузочные тесты.</p>	

№ п/п	Тема проекта	Исходные данные к проекту	ФИО, дата, подпись
19	Многопользовательское удалённое файловое хранилище на основе протокола UDP	<p>Разработать сетевое приложение, реализующее многопользовательское удалённое файловое хранилище на основе протокола UDP, реализующее операции загрузки, выгрузки, удаления файлов и переименования файлов. Реализовать систему аутентификации пользователей и ролей. Предусмотреть возможность как персонализированного доступа к файлам, так и общего с разграничением прав доступа. Все пользовательские файлы на сервере должны храниться в одном физическом файле.</p> <p>Разработать модульные и нагрузочные тесты.</p>	
20	Многопользовательское удалённое файловое хранилище на основе протокола HTTP	<p>Разработать сетевое приложение, реализующее многопользовательское удалённое файловое хранилище на основе протокола HTTP, реализующее операции загрузки, выгрузки, удаления файлов и переименования файлов. Реализовать систему аутентификации пользователей и ролей. Предусмотреть возможность как персонализированного доступа к файлам, так и общего с разграничением прав доступа.</p> <p>Необходимо разработать как web-клиента, так и консольного клиента для доступа к хранилищу.</p> <p>Разработать модульные и нагрузочные тесты.</p>	
21	Многопользовательское удалённое файловое хранилище на основе web-socket	<p>Разработать сетевое приложение, реализующее многопользовательское удалённое файловое хранилище на основе web-socket, реализующее операции загрузки, выгрузки, удаления файлов и переименования файлов. Реализовать систему аутентификации пользователей и ролей. Предусмотреть возможность как персонализированного доступа к файлам, так и общего с разграничением прав доступа.</p> <p>Разработать модульные и нагрузочные тесты.</p>	

2. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат разработке):

- *титульный лист;*
- *лист для рецензии научного руководителя;*
- *задание по курсовому проектированию;*
- *содержание;*
- *введение*, в котором обосновывается актуальность темы, указываются цель и задачи работы, дается краткий перечень соответствующих информационных технологий и программных средств, позволяющих осуществлять решение подобных задач;
- *основная часть*, как правило, включает три раздела:
 - *Раздел 1* содержит обзор литературы по математическому, программному обеспечению, касающемуся непосредственно решаемой задачи. При этом, в тексте пояснительной записки делаются ссылки на литературу. Указываются существующие подходы и методы решения рассматриваемой задачи. Обосновывается выбор одного из средств решения задачи и делаются соответствующие выводы.
 - *Раздел 2* посвящен алгоритмическому анализу задачи. Он содержит полную постановку задачи, перечень исходных данных и предполагаемых результатов, обобщенную графическую схему алгоритма, функциональную схему, иерархическую схему взаимодействия объектов.
 - *Раздел 3* содержит описание программы, реализующего разработанный алгоритм, описывается проведенное решение и даются выводы по проведенным исследованиям, причём необходимо приводить ссылки на соответствующие документы с результатами, размещённые в «Приложениях».
- *Заключение* содержит общие выводы по работе в целом, отмечаются достоинства работы, указывается ее возможная практическая значимость.
- *Список литературы* оформляется согласно требованиям ГОСТ.
- *Приложения* содержат листинги программ, реализующих алгоритм решения задач, внешний вид окон интерфейса программы, результаты верификации и проводимых исследований.

3. Календарный график работы над проектом на весь период проектирования (с указанием сроков исполнения и трудоемкости основных этапов)

№ п/п	Сроки выполнения	Наименование этапа
1	16.09.2021	Создание приватного репозитория на Git для работы над проектом
2	30.09.2021	Подбор литературных источников и изучение сетевых технологий, необходимых для выполнения проекта
3	21.10.2021	Создание основных классов приложения
4	11.11.2021	Создание графического интерфейса
5	25.11.2021	Разработка модульных тестов
6	02.12.2021	Тестирование и опытная эксплуатация приложения
7	10.12.2021	Работа над пояснительной запиской

4. Литература

Основная литература

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 3-е изд. – СПб.: Питер, 2006.
2. Таненбаум Э. Компьютерные сети. – М. «Вильямс», 2003
3. Олифер В.Г., Олифер Н.А. Сетевые операционные системы. Учебник. – СПб.: Питер, 2001.
4. Вишнеvский В.М. Теоретические основы проектирования компьютерных сетей. – Москва: Техносфера, 2003.
5. Уэнделл О. Компьютерные сети. Первый шаг.: Пер. с англ. – Москва: Издательский дом «Вильямс», 2006.

Дополнительная литература

6. Крелл М., Манн С. Linux. Администрирование сетей TCP/IP. – М. «Вильямс», 2003
7. Андреев А.Г. и др. Microsoft Windows 2000: Server и Professional. Русские версии / Под общ. ред. А.Н. Чекмарева и Д.Б.Вишнякова. – СПб.: БХВ-Петербург, 2001
8. Стахнов А. Сетевое администрирование Linux. - СПб.: Питер-пресс, 2004
9. Администрирование сети на основе Windows 2000. Учебный курс MCSE. Сертификационный экзамен 70-216. - СПб.: БХВ-Петербург, 2004

Методическая литература

10. Курочка К.С. Практическое пособие по теме «Компьютерные сети» курсов «Сетевые технологии», «Компьютерные информационные технологии», «Информатика» для студентов дневного и заочного отделений. Гомель, 2005 (м/у 3105).