

Лабораторная работа № 1

Тема: Сервер на основе UDP протокола.

На выполнение Лабораторной работы 1 дается 3 недели.

Срок – последняя неделя февраля.

Задание:

1. Написать клиент-серверную программу на основе транспортного протокола UDP [1-3].
Сервер отображает на экране монитора полученную информацию от клиента, а также его клиентский IP адрес и порт; и отправляет преобразованную информацию обратно клиенту. Клиентская программа передаёт заданное число i в цикле (определенное число раз с задержкой в i сек) на сервер, получает и выводит на экран преобразованную информацию от сервера.
2. Продемонстрировать реализованные возможности программ согласно заданию при одновременной передаче информации от нескольких клиентов к серверу.
3. Реализация - на языке C/C++, консольные приложения.

Серверная программа должна находить номер свободного порта и выводить его на экран (см., например, [1], стр. 338-342: обратите внимание на функции `bind`, `getsockname`). При запуске клиентской программы задавать со строки IP адрес сервера и порт.

Полезные ссылки:

1. Фейт С. TCP/IP: Архитектура, протоколы, реализация (включая IP версии 6 и IP Security). – М.: Лори, 2000. – 424 с.
2. Павский К. В., Ефимов А. В. Разработка сетевых приложений (протоколы TCP/IP, клиент-сервер, PCAP, Boost.ASIO): Учебное пособие / Сибирский государственный университет телекоммуникаций и информатики. – Новосибирск, 2018. – 80 с.
3. Протоколы TCP/IP и разработка сетевых приложений: учеб. пособие / К.В. Павский; Сиб. гос. ун-т телекоммуникаций и информатики. - Новосибирск: СибГУТИ, 2013. – 130с.

Лабораторная работа № 2

Тема: Параллельный (мультипроцессный) сервер.

На выполнение Лабораторной работы 2 дается 3 недели.

Срок – 3я неделя марта.

Задание:

1. Написать программу обеспечивающую параллельную работу сервера, принимающего информацию от клиента по сети. Условие: мультипроцессная организация на основе функции **fork**, транспортный протокол – TCP [1-3].
Обеспечить в сервере завершение «зомби-процессов» !!!
2. Написать клиентскую программу, передающую заданное число i в цикле (определенное число раз с задержкой в i сек) на сервер. Соответствующий процесс сервера выводит полученную информацию на экран.
3. Реализация - на языке C/C++, консольные приложения.

Серверная программа должна находить номер свободного порта и выводить его на экран (см. [1-3], стр. 338-342, функции `bind`, `getsockname`). При запуске клиентской программы задавать со строки IP адрес сервера и порт.

Лабораторная работа № 3

Тема: Параллельный (многопоточный) сервер.

На выполнение Лабораторной работы 4 дается 3 недели.

Срок – 2я неделя апреля.

Задание:

1. Написать программу обеспечивающую параллельную работу сервера, принимающего информацию от клиента по сети. Информацию получаемую от клиента сохранять в одном общем файле (обеспечить целостность данных). Условие: мультипоточная организация на основе функций библиотеки **pthread**, транспортный протокол – TCP.
2. Написать клиентскую программу, передающую заданное число *i* в цикле (определенное число раз с задержкой в *i* сек) на сервер. Соответствующий поток (thread функция) сервера выводит полученную информацию на экран.
3. Продемонстрировать реализованные возможности программ согласно заданию.
4. Реализация - на языке C/C++, консольные приложения.

Серверная программа должна находить номер свободного порта и выводить его на экран (см. [1-3], функции bind, getsockname). При запуске клиентской программы задавать со строки IP адрес сервера и порт.

Лабораторная работа № 4

Тема: Псевдопараллельный сервер

На выполнение Лабораторной работы 4 дается 3 недели.

Срок – 2я неделя мая.

Задание:

1. Разработать программу однопоточного сервера, использующую асинхронный ввод/вывод (организованный с помощью системного вызова **select**) обеспечивающую псевдопараллельную работу клиентов.
2. Написать клиентскую программу, передающую сообщения на сервер.
3. Продемонстрировать асинхронную работу сервера. Например, при запуске клиента пользователь задает число *i* от 1 до 10. Клиент передает серверу в цикле это число с задержкой в *i* секунд между передачей. Сервер отображает на экран полученную от клиентов информацию.

Например:

1-й клиент посылает число 1 в цикле с задержкой в 1 сек.

2-ой клиент посылает число 2 с задержкой в 2 сек.

3-й клиент посылает число 3 в цикле с задержкой в 3 сек.

Сервер отображает информацию полученную от клиентов. Если у Вас правильно организован асинхронный ввод/вывод, то на экран со стороны сервера будет выводиться с чередованием числа 1, 2 и 3. Причем частота появления определенного числа будет зависеть от задержки по времени его передачи.

4. Реализация - на языке C/C++, консольные приложения.

Серверная программа должна находить номер свободного порта и выводить его на экран. При запуске клиентской программы задавать со строки IP адрес сервера и порт.

Полезные ссылки:

1. Фейт С. TCP/IP: Архитектура, протоколы, реализация (включая IP версии 6 и IP Security). – М.: Лори, 2000. – 424 с.
2. Павский К. В., Ефимов А. В. Разработка сетевых приложений (протоколы TCP/IP, клиент-сервер, PCAP, Boost.ASIO): Учебное пособие / Сибирский государственный университет телекоммуникаций и информатики. – Новосибирск, 2018. – 80 с.
3. Протоколы TCP/IP и разработка сетевых приложений: учеб. пособие / К.В. Павский ; Сиб. гос. ун-т телекоммуникаций и информатики. - Новосибирск: СибГУТИ, 2013. – 130с.

РГР

РГР проект выполняется студентом самостоятельно в соответствии с вариантом задания, назначаемым преподавателем. Для реализации РГР проекта необходимо разработать консольное сетевое приложение на языке программирования C/C+ в OS Linux.

1. Разработка сетевого приложения. Почтовый клиент на базе протокола POP3.
2. Разработка сетевого приложения. Почтовый клиент на базе протокола SMTP.
3. Разработка сетевого приложения. Почтовый клиент на базе протокола IMAP4.
4. Разработка сетевого приложения. Клиент на базе протокола FTP в активном режиме.
5. Разработка сетевого приложения. Клиент на базе протокола FTP в пассивном режиме.
6. Разработка сетевого приложения. Клиент на базе протокола TFTP.
7. Разработка сетевого приложения HTTP-сервер.
8. Разработка сетевого приложения «Анализатор сетевого трафика».
9. Разработка сетевого приложения «Чат». Мультипоточная реализация сервера с установлением соединения с использованием библиотеки PTHREAD.
10. Разработка сетевого приложения «Чат». Мультипроцессная реализация сервера с установлением соединения с использованием функции fork.
11. Разработка сетевого приложения «Чат». Асинхронный сервер с установлением соединения с использованием функции select.
12. Разработка сетевого приложения «Чат». Сервер без установления соединения (протокол UDP).
13. Разработка сетевого приложения «Сетевая игра». Мультипоточная реализация сервера с установлением соединения с использованием библиотеки PTHREAD.
14. Разработка сетевого приложения «Сетевая игра». Мультипроцессная реализация сервера с установлением соединения с использованием функции fork.
15. Разработка сетевого приложения «Сетевая игра». Асинхронный сервер с установлением соединения с использованием функции select.
16. Разработка сетевого приложения «Сетевая игра». Сервер без установления соединения (протокол UDP).

Описание отчета к РГР

1. Титульный лист. Название – "Разработка сетевого приложения. *ТЕМА РАБОТЫ*" (Например, у вас тема работы – «*FTP клиент*», тогда название – «**Разработка сетевого приложения. *FTP клиент***»)
2. Содержание
3. Постановка задачи
4. Описание протокола
5. Описание вашей реализации
6. Скан экрана работы вашей программы
7. Текст программы
8. Список источников

На «удовлетворительно» - приложить к отчету текст сетевой программы одной из Лабораторных работ.

Темы рефератов (удовлетворительно)

1. POP3+SMTP
2. TFTP
3. Протокол IMAP4
4. Протокол SCTP
5. Протокол FTP
6. Протокол ICMP
7. TCP+UDP (заголовки; алгоритмы, повышающие производительность)
8. IPv4, IPv6 (заголовки, описание, характеристики)
9. Протокол ICQ
10. BGP, EGP
11. Token Ring
12. Протокол OSPF
13. Протокол RIP, RIP2
14. ICMPv6