

## Семестр 2

### Лабораторная работа № 1

**Тема:** Сервер на основе UDP протокола. Поддержка подтверждения получения информации.

#### Задание:

1. Написать клиент-серверную программу на основе транспортного протокола UDP. Реализовать: подтверждение приема для каждой датаграммы, сохранение целостности всей информации. Клиент передает файл или сообщение (несколькими датаграммами), сервер принимает.
2. Продемонстрировать реализованные возможности программ согласно заданию, при одновременной передаче файлов от нескольких клиентов к серверу. Например, при запуске сервера указать - какие пакеты и сколько раз будут потеряны. Результат правильности приема выводить на экран.
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, 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. <http://rdsn.ru/article/unix/sockets.xml> .

### Лабораторная работа № 2

**Тема:** Параллельный (мультипроцессный) сервер.

1. Написать программу обеспечивающую параллельную работу сервера, принимающего файлы от клиента по сети. Условие: мультипроцессная организация на основе функции fork, транспортный протокол – TCP [1, стр. 340].  
Обеспечить в сервере завершение «зомби-процессов» !!!
2. Написать клиентскую программу, передающую файл по сети серверу.
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], стр. 338-342, 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.

## Лабораторная работа № 3

### Тема: Параллельный (многопоточный) сервер.

#### Задание:

1. Написать программу обеспечивающую параллельную работу сервера, принимающего файлы 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 файл по сети серверу.
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. , например, [1], стр. 338-342, 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.

## Лабораторная работа № 4

### Тема: Псевдопараллельный сервер

#### 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.

## Лабораторная работа № 5

### Тема: Мультипротокольный сервер.

#### Задание:

1. Написать программу, реализующую работу сервера по двум протоколам (TCP и UDP) параллельно с помощью системного вызова `select`.
2. Написать две клиентские программы, передающие на сервер файлы по протоколам TCP и UDP соответственно.
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, 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.

## Лабораторная работа № 6

### Тема: Создание клиент-серверного приложения на базе транспортного протокола SCTP.

#### Задание:

Требуется разработать клиент-серверные программы передачи данных на базе транспортного протокола SCTP [1, 2]. Обеспечить передачу информации по нескольким потокам в одном соединении или ассоциации (особенность протокола SCTP).

1. Клиентская программа посылает в одном соединении или ассоциации: текстовое сообщение в первом потоке; файл с картинкой во втором потоке.
2. Серверная программа принимает в одном соединении или ассоциации от каждого из клиентов: текстовое сообщение с первого потока и выводит на экран; со второго потока принимает данные и сохраняет в файл.
3. В серверной программе реализовать (на выбор) параллельную или псевдопараллельную обработку клиентов.
4. Реализация - на языке C/C++, консольные приложения.

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

#### Полезные ссылки:

1. Протокол SCTP <http://rfc.com.ru/rfc2960.htm>
2. Стивенс У.Р., Феннер Б., Рудофф Э. М. UNIX: разработка сетевых приложений. - 3-е изд. - СПб. : ПИТЕР, 2007. - 1038с.
3. Фейт С. TCP/IP: Архитектура, протоколы, реализация (включая IP версии 6 и IP Security). – М.: Лори, 2000. – 424 с.

## Лабораторная работа № 7

**Тема:** Разработка сетевой программы – «снифер».

### **Задание:**

Используя библиотеку PCAP написать сетевую программу «снифер». По заданному правилу (регулярное выражение в стиле bpf (Berkley Packet Filter) обеспечить фильтрацию пакетов в сети. В выбранном пакете найти определенную информацию.

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

### **Например:**

Фильтровать все пакеты по протоколу TCP на порт 21. Определить IP адрес отправителя и получателя пакета или найти пакет содержащий слово USER.

Фильтр задавать либо при запуске программы (со строки) либо в диалоговом режиме.

## Лабораторная работа № 8

**Тема:** Разработка клиент-серверной программы на основе сетевой библиотеки Boost.Asio.

### **Задание:**

Используя сетевую библиотеку Boost.Asio разработать программы асинхронного сервера, клиента на протоколе TCP [1]. Реализация - на языке C/C++, консольные приложения.

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

### **Вариант первый.**

Клиент отправляет текстовое сообщение. В ответе сервер возвращает то же самое преобразованное сообщение, в котором каждое слово «перевернуто» относительно исходного, но сохранен порядок их расположения.

### **Вариант второй.**

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 возвращает клиенту квадрат числа  $i$ .

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.

**Полезная ссылка:** 1. Boost.Asio <http://habrahabr.ru/post/192284/> .

## Лабораторная работа 9

**Тема:** Разработка параллельных программ.

**Задание:**

Используя библиотеку MPI написать параллельную программу перемножения матрицы на вектор.

Исследовать эффективность распараллеливания на кластерной вычислительной системе - рассчитать коэффициент ускорения вычислений параллельной программы в зависимости от числа элементарных машин (вычислительных ядер) системы.

**Полезные ссылки:**

MPI

[http://parallel.ru/tech/tech\\_dev/mpi.html](http://parallel.ru/tech/tech_dev/mpi.html)

Центр Параллельных Вычислительных Технологий (ЦПВТ) СибГУТИ

<http://cpct.sibsutis.ru/index.php/Main/HomePage>

Руководство пользователя вычислительного кластера Jet ЦПВТ

<http://cpct.sibsutis.ru/uploads/Main/jet-cluster-howto>

Вычислительные ресурсы ЦПВТ

<http://cpct.sibsutis.ru/index.php/Main/Resources>