

Каждая лабораторная работа рассчитана на 2 академических часа.

# Семестр 1.

## Л.р. 1 - введение в сетевое программное обеспечение

Системные утилиты для тестирования и анализа сетевых протоколов: **telnet**, **netcat**, **netstat**, **tcpdump**, **nmap**(nmapsi4), **dsniff** (dsniff-X11), **wireshark**, **ip/ifconfig/ipconfig**, **route**

**ifconfig/ipconfig/ip**

1. найти имена интерфейсов и адреса

**netstat**

1. список открытых сокетов
2. серверных
3. узнать имена/pid серверов
4. узнать номера портов

**telnet/netcat**

- 1 чат
- 2 передача файла
- 3 tcp/udp

**nmap**

1. сканирование соседа
2. сканирование выделенных портов у соседа (поиск сервера чата)
3. узнать список открытых портов на всех машинах в 505
4. узнать список работающих машин

**wireshark/tcpdump**

0. pcap файлы/libpcap
1. запуск монитора
2. запуск чата
3. монитор-фильтр-анализ

## Л.р. 2 - Знакомство с программированием сокетов.

Простейшая программа-сервер с использованием протокола TCP (например возвращающая данные клиента). Использование системных утилит (telnet, netcat) в качестве клиента.

### **Л.р. 3 - Клиент-серверная программа для передачи файла по сети с использованием протокола TCP.**

Создать клиент и последовательный сервер, позволяющий передать файл с помощью протокола TCP. Уделить внимание обработке исключительных ситуаций, например физического или программного обрыва соединения. Проверять можно с помощью включения файерволла с отбрасыванием пакетов без уведомления (правило DROP) и с отбрасыванием пакетов с уведомлением (правило REJECT).

### **Л.р. 4 - Изучение внеполосного режима передачи данных.**

Модификация программы из Л.р. 3: во время передачи данных с использованием протокола TCP, передающая сторона должна генерировать внеполосные данные и выводить на экран общее количество переданных байт данных (не включая срочные), принимающая сторона должна выводить на экран общее количество принятых байт (не включая срочные) при получении срочных данных.

### **Л.р. 5 - Клиент-серверная программа для передачи файла по сети с использованием протокола UDP.**

Модификация программы из Л.р. 4. Добавить к клиенту и серверу возможность передачи файла с помощью протокола UDP. Уделить внимание обработке исключительных ситуаций, например физического или программного обрыва соединения. Проверять можно с помощью включения файерволла с отбрасыванием пакетов без уведомления (правило DROP) и с отбрасыванием пакетов с уведомлением (правило REJECT).

### **Л.р. 6 - организация параллельной обработки запросов на сервере с помощью мультиплексирования**

Модификация программы из Л.р. 5. Модифицировать сервер для организации параллельного обслуживания нескольких клиентов с помощью мультиплексирования (select, pselect, poll).

### **Л.р. 7 - организация параллельной обработки запросов на сервере с помощью потоков**

Модификация программы из Л.р. 5. Модифицировать сервер для организации параллельного обслуживания нескольких клиентов с помощью выделения отдельного программного потока

(thread).

## **Л.р. 8 - организация параллельной обработки запросов на сервере с помощью процессов**

Модификация программы из Л.р. 5. Модифицировать сервер для организации параллельного обслуживания нескольких клиентов с помощью выделения отдельного процесса.

# **Семестр 2**

## **Л.р. 1 - Изучение протокола ICMP и особенностей его программирования**

Реализация программы ping.

## **Л.р. 2 - Изучение протокола IP**

Модификация программы из Л.Р. 2: добавить возможность изменения адресов источника, назначения, а также поля TTL в заголовке IP-пакета.

## **Л.р. 3 - Изучение широковещательного режима передачи данных**

Разработать одноранговую программу - чат с использованием протокола UDP или IP, передающей и принимающей сообщения с помощью широковещательного режима передачи данных.

Предусмотреть нахождение и вывод списка IP-адресов запущенных приложений.

Программа должна автоматически определить сетевые параметры интерфейса: ip-адрес, сетевую маску и адрес широковещательной передачи данных.

## **Л.р. 4 - изучение режима многоадресной передачи данных**

Разработать одноранговую программу - чат с использованием протокола UDP или IP, передающей и принимающей сообщения с помощью многоадресного режима передачи данных. Предусмотреть нахождение и вывод списка IP-адресов запущенных приложений.

## **Л.р. 5 - MPI: Изучение парных коммуникаций.**

Программа для умножения матриц с использованием парных операций. Реализовать 2 варианта - с использованием блокирующего и неблокирующего режимов передачи данных.

## **Л.р. 6 - MPI: изучение коллективных операций, групп и коммутаторов**

Модифицировать программу из Л.р. 5 с использованием коллективных операций.

Программа должна создавать произвольное количество групп (задается из командной строки) и включить в них случайное количество процессов. Каждая группа должна умножить матрицы.

Замерить время вычисления в каждой группе и вывести на экран.

## **Л.р. 7 - MPI: файловые операции**

Модификация программы из л.р. 6: исходные данные для умножения задаются из 2-х файлов.

Каждый процесс выводит результаты умножения матриц в файл, принадлежащий соответствующей группе.

## **Л.р. 8 - Запуск приложения на кластере с помощью PBS и GRID**

Запустить приложение из Л.р. 7 в кластерной среде с помощью механизмов PBS.