

Лабораторная работа №6.

Взаимодействие через сеть

Цель

Цель этой работы — ознакомиться с интерфейсом сокетов ОС для взаимодействия программ через сеть.

В результате ее выполнения должно возникнуть понимание принципа сетевого взаимодействия на прикладном уровне и навыки использования сокетов для создания клиентских сетевых приложений.

Задание

На языке C, не используя клиентские библиотеки, а работая напрямую с сокетными соединениями, связаться с одним из следующих серверов и выполнить заданные команды.

Для выполнения работы необходимо установить указанный сервер на локальный компьютер и соединение выполнять к хосту localhost. Проверить результаты нужно с помощью родных клиентов или другим способом, который указан в варианте задания.

Интерфейс сокетов

API сокетов — это стандартный способ взаимодействия программ через сеть. Оно реализует клиент-серверную модель взаимодействия. Подробнее об этом см. конспект лекции по теме «Сеть».

Базовые сетевые утилиты

Для проверки работоспособности сети и выполнения простых операций с ее использованием в ОС присутствует следующий полу-стандартный ряд инструментов.

Утилита `ping` позволяет проверить доступность определенного хоста, IP-адрес или DNS-имя которого известны. Пример работы:

```
$ ping 8.8.8.8 PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data. 64 bytes from 8.8.8.8: icmp_seq=1 ttl=46 time=50.3 ms 64 bytes from 8.8.8.8: icmp_seq=2
```

```
ttl=46 time=49.8 ms ^C --- 8.8.8.8 ping statistics --- 2 packets transmitted, 2
received, 0% packet loss, time 1001ms rtt min/avg/max/mdev =
49.885/50.101/50.318/0.311 ms
```

В данном случае можно увидеть, что хост 8.8.8.8 доступен и скорость соединения высокая (небольшое время ответа, до сотни миллисекунд: time=50.3 ms). В случае отсутствия возможности подключиться к хосту пинг не выдает результатов.

Утилита `nslookup` позволяет узнать IP-адрес хоста по его DNS-имени. Пример работы:

```
$ nslookup kpi.ua Server: 127.0.1.1 Address: 127.0.1.1#53
Non-authoritative answer: Name: kpi.ua Address: 77.47.133.222
```

В данном случае мы установили, что символическому имени хоста `kpi.ua` соответствует IP-адрес `77.47.133.222`.

Утилита `netstat` показывает текущее состояние сетевых соединений данного хоста. Пример работы:

```
$ netstat -nat Active Internet connections (servers and established) Proto
Recv-Q Send-Q Local Address Foreign Address State tcp 0 0 0.0.0.0:445
0.0.0.0:* LISTEN tcp 0 0 0.0.0.0:139 0.0.0.0:* LISTEN tcp 0 0
192.168.1.2:60014 199.16.156.8:443 ESTABLISHED tcp 1 0
192.168.1.2:50013 87.245.216.59:80 CLOSE_WAIT
```

В данном случае на хосте активны 4 TCP соединения, которые находятся в разных состояниях (LISTEN, ESTABLISHED, CLOSE_WAIT).

Утилита `telnet` позволяет установить TCP соединение с заданным хостом по указанному порту и передавать текстовые данные через это соединение. Пример работы:

```
$ telnet rainmaker.wunderground.com 23 Trying 38.102.137.140... Connected
to rainmaker.wunderground.com. Escape character is '^]'.
----- * Welcome to THE
WEATHER UNDERGROUND telnet service! *
----- ...
```

В этом примере мы подключились к хосту `rainmaker.wunderground.com` по порту 23 и получили в ответ текстовую информацию. Эта утилита — это базовый клиент для любого TCP-сервиса, с помощью которого можно выполнять отладку его работы.

- Утилита `netcat` — это более функциональный аналог `telnet`, который может выполнять как TCP, так и UDP соединения, а также работать не только в режиме клиента, но и сервиса.

- Утилита `traceroute` позволяет проверить наличие сетевого маршрута между нашим хостом и другим хостом. При этом она показывает, какие промежуточные узлы присутствуют на этом маршруте. Более современной ее альтернативой является утилита ```