ЛБ1

Адресация узлов в сети. Порядок разрешения адресов

> **В ip-сетях используется три типа сетевых адресов: mac-адрес, сетевой адрес и доменное имя. Они используются на разных уровнях сетевой модели для идентификации хостов.**

Цель работы: Рассмотреть схему адресации узлов в ip-сетях. Получить представление о порядке разрешения адресов, используемых на различных уровнях стека TCP/IP.

Задания к работе

1. Определить физический и сетевой адреса локального хоста и его доменное имя.
2. Просмотреть таблицу преобразования физических адресов. Сохранить полученную информацию в файле.
3. Командой ping проверить доступность следующих узлов:
    - 127.0.0.1;
    - locahost;
    - example.com
    - трех-четырех соседних компьютеров.
4. Просмотреть таблицу преобразования адресов и сравнить ее с результатами, полученными в задании 1.
5. Сделать перерыв в сетевой активности на несколько минут, после которого повторить предыдущий пункт. Пояснить причины изменений (или отсутствия таковых) в таблице arp за время перерыва.
6. Добавить в таблицу *статическую* запись (действительные аппаратный и сетевой адреса одной из соседних машин)
7. Выполнить ping добавленного в предыдущем пункте сетевого адреса
8. Добавить в таблицу преобразований следующие записи (пары "mac-адрес — ip-адрес"):
    - действительный mac-адрес — недействительный сетевой адрес;
    - недействительный mac-адрес — действительный сетевой адрес;
9. Проверить доступность добавленных узлов. Объяснить полученные результаты.
10. Просмотреть таблицу arp и сохранить ее в файле для дальнейшего использования.
11. Перезагрузить компьютер и снова просмотреть кэш arp. Сравнить с результатами задания 9. Что стало с записями, добавленными вами в заданиях 5 и 7?
12. Добавить в файл hosts (путь к файлу в ОС Windows: %systemroot%\System32\Drivers\etc\hosts, в UNIX: /etc/hosts, в обоих случаях нужны привилегии администратора) следующую запись:

    ```
    194.188.210.1    edu.asoiu
    ```

    Если такая запись уже имеется, то перейти к следующему заданию

13. Выполнить ping узла edu.asoiu
14. Определить по таблице arp mac-адрес узла edu.asoiu.
15. Определить все ip-адреса (публичные) одного из указанных сервисов: mail.ru, ya.ru, google.com или подобного.
16. Определите имя и ip-адрес первичного DNS-сервера зоны ru.
17. Ответить на контрольные вопросы

Указания к работе

Преобразование адресов

Для сопоставления сетевого адреса с аппаратным адресом интерфейса в стеке TCP/IP имеются специализированные протоколы типа *arp* (address resolution protocol, RFC-826). Это позволяет использовать сетевые протоколы стека поверх различных протоколов канального уровня. Все операции преобразования выполняются прозрачно для протоколов верхних уровней. Результаты преобразований кэшируются и сохраняются на некоторый интервал времени, что позволяет не выполнять преобразование при повторном обращении к ранее взаимодействовавшим узлам.

Кэш arp представлен в виде таблицы, заполненной записями примерно такого вида:

```
"сетевой адрес — MAC-адрес — интерфейс — способ назначения"
```

Эта таблица формируется *динамически*, при любом сетевом взаимодействии узла. Для просмотра кэша arp используется одноименная команда — arp. Эта же команда позволяет формировать таблицу MAC-адресов *статически*, передавая записи через список аргументов. Команда arp используется как в UNIX, так и в Windows-системах.

Основной способ заполнения таблицы преобразований — динамический, при котором записи добавляются по мере участия узла в сетевом обмене. Это означает, что в отсутствие сетевой активности кэш arp пуст (если не задано статических записей). Для выполнения заданий к этой работе вам необходимо организовать некоторое сетевое взаимодействие. Пожалуй, самым доступным способом для этого является использование команды *ping*.

Команда ping использует протокол ICMP (Internet Control Message Protocol; RFC-792, RFC-1256) для отправки запросов датаграммного типа (ECHO_REQUEST) и ожидает ответ (ECHO_RESPONSE) от запрашиваемого хоста или шлюза.

ECHO_REQUEST — это датаграмма, имеющая заголовок IP и ICMP. Поле данных заполнено некоторым количеством произвольной информации. Для анализа сети выполняется отправка определенного количество таких датаграмм. По результатам анализа можно судить о доступности запрашиваемого хоста и некоторых аспектах работы сети в целом.

Обязательным параметром команды ping является сетевой адрес узла, заданный в числовом виде:

```
ping 192.0.32.10
```

или в символьном представлении:

```
ping example.com
```

Если задан символьный адрес, то ping попытается выполнить преобразование символьного имени в сетевой адрес. Для этого сначала будет перечитываться содержимое файла *hosts*, который является своего рода сервером DNS в масштабе отдельно взятого сетевого узла. Содержательно файл hosts — обычный текстовый файл, где прописано соответствие ip-адресов доменным именам. Его основное назначение — ускорить преобразование имен компьютеров в сетевые адреса. Формат файла приведен ниже:

```
#ip-address      hostname         aliaces
x.x.x.x          hostname         [aliace1 [aliace2 [...[aliaceN]]]]
```

Обычно в этом файле содержится единственная запись:

```
127.0.0.1        localhost
```

Если требуемое имя узла найдено в файле hosts, то возвращается соответствующий ему сетевой адрес. Иначе — выполняется запрос к внешнему серверу DNS, указанному в настройках сетевого интерфейса.

Как узнать MAC-адрес и ip-адрес?

Чтобы узнать физический адрес локального хоста и его ip-адрес нужно выполнить команду ifconfig (в ОС Windows - ipconfig). Запущенная без параметров, команда ifconfig отображает информацию об имеющихся в системе сетевых интерфейсах и их физических и сетевых адресах:

```
aag@localhost:~> sudo /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:1D:92:A2:90:E7
          inet addr:192.168.1.250  Bcast:192.168.255.255  Mask:255.255.0.0
          inet6 addr: fe80::21d:92ff:fea2:90e7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:811957 errors:0 dropped:0 overruns:0 frame:0
          TX packets:446207 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:596559482 (568.9 Mb)  TX bytes:114698114 (109.3 Mb)
          Interrupt:28 Base address:0xe000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:24226 errors:0 dropped:0 overruns:0 frame:0
          TX packets:24226 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:50861906 (48.5 Mb)  TX bytes:50861906 (48.5 Mb)
```

Как узнать доменное имя?

Узнать доменное имя хоста можно командой hostname.

Как узнать адрес сервера DNS?

Узнать адрес сервера DNS можно разными способами, самый простой — посмотреть содержимое файла resolv.conf:

```
cat /etc/resolv.conf
```

Расширенную информацию о сервере DNS можно получить используя специальные команды, такие как dig (man 1 dig) или host (man 1 host). В ОС Windows можно использовать утилиту nslookup.

В Листинге 1 приведен простой пример использования утилиты dig. Шрифтом выделено имя отвечающего сервера имен (сравните с записью в resolv.conf).

Листинг 1. Пример использования команды dig.

```
aag@localhost:~> dig example.com

; <<>> DiG 9.7.3 <<>> example.com
;; global options: +cmd
;; Got answer:
```

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34206
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;example.com.                    IN      A

;; ANSWER SECTION:
example.com.             1095    IN      A       192.0.32.10

;; AUTHORITY SECTION:
example.com.             172792  IN      NS      a.iana-servers.net.
example.com.             172792  IN      NS      b.iana-servers.net.

;; ADDITIONAL SECTION:
b.iana-servers.net.      28792   IN      A       193.0.0.236
b.iana-servers.net.      28792   IN      AAAA    2001:610:240:2::c100:ec

;; Query time: 1 msec
;; SERVER: 192.168.3.1#53(192.168.3.1)
;; WHEN: Thu Apr  7 10:34:46 2011
;; MSG SIZE  rcvd: 137
```

Контрольные вопросы

1. Что и почему изменилось в таблице arp после выполнения задания №2?
2. Что произойдет, если в таблицу arp добавить две или более записей, в которых одному mac-адресу сопоставлены разные сетевые адреса?
3. Что произойдет, если в таблицу arp добавить две или более записей, в которых одному сетевому адресу сопоставлены разные аппаратные адреса?
4. Как отличается "время жизни" динамических и статических записей в таблице arp?
5. Почему в ip-сетях не используется прямое сопоставление символьного адреса физическому адресу?
6. Что произойдет, если в файл hosts записать два (или более) узла с одинаковыми именами (например, myhost.mydomain), но разными сетевыми адресами, а затем обратиться к ним по имени (например так: ping myhost.mydomain)?

ЛБ2

Мониторинг сети. Сниффер Wireshark

Под мониторингом сети понимают процесс сбора и анализа сетевого трафика, по результатам которого можно судить о качественных и количественных характеристиках работоспособности сети или ее отдельных компонентов. Программы мониторинга сети позволяют выполнять захват пакетов и их реассемблирование для дальнейшего анализа.

Цель работы: Освоить базовые навыки мониторинга сети с использованием программ для анализа протоколов.

Задания к работе

1. Запустить ENA в режиме захвата трафика, проходящего через интерфейс, подключенный к локальной сети (обычно это eth0). Перейти к следующему заданию.
2. Эмулировать сетевую активность в течении 10-15 минут. Для этого можно выполнить, например, некоторые из указанных действий.
    - Открыть сайт http://asoiu.com;
    - Подключиться к серверу ftp://ftp.omgtu;
    - Выполнить пинг любых узлов;
    - Подключиться к одному из доступных сетевых дисков Windows (если такие ресурсы представлены в сети)
    - Выполнить прочие действия, требующие сетевого подключения.
3. Остановить захват.
4. Заполнить таблицу 2.1. Исходные данные для таблицы представлены в отчете Statistics/Summary. При заполнении таблицы обратите внимание на соблюдение размерности величин (кб, Мб, Мбит).

    Таблица 2.1.

Параметр	**Значение**
Время захвата, мин	
К-во захваченных пакетов	
Объем, Мб	
Средн.размер пакета, Кб	
Средняя скорость, пакетов/сек	
Средняя скорость, Мбит/сек	

5. Составить таблицу распределения трафика по протоколам (табл. 2.2). Исходные данные для таблицы можно получить из отчета Statistics/Protocol Hierarchy.

    Таблица 2.2.

Протокол	**Трафик, Мб**	**Трафик, %**
HTTP		
FTP		
...		
ИТОГО		**100**

6. Составить таблицу распределения Ethernet-трафика по узлам сети (табл. 2.3). Исходные данные для заполнения таблицы получить из отчета Statistics/Endpoint list/Ethernet.

   Таблица 2.3.

MAC-адрес	**IP-адрес**	**Трафик**					
		входящий		*исходящий*		*общий*	
		Мб	%	Мб	%	Мб	%
ИТОГО			**100**		**100**		**100**

7. По данным табл. 2.1 определить *относительную загрузку* сети (в %) за контрольный период времени по формуле:

$$Загрузка = \frac{(Трафик, Мбит/Время, сек)\cdot 100}{(Пропускная способность, Мбит/сек)}$$

8. По данным табл. 2.2 сделать выводы о качественном составе трафика, т.е. о соотношении *прикладных* и *служебных* протоколов.
9. По данным табл. 2.3 определить, какие из узлов являются наиболее загруженными с учетом направления трафика (исходящий, входящий, общий).

Указания к работе

Для мониторинга используют специальные программы - анализаторы сети. Таких программ много, например Windows Network Monitor, tcpdump, Ethereal Network Analyzer (ENA), Wireshark и т.п. Они схожи по функциям, а отличаются в основном пользовательским интерфейсом и возможностями генерации статистических отчетов. На рис. 2.1 приведены примеры таких программ.

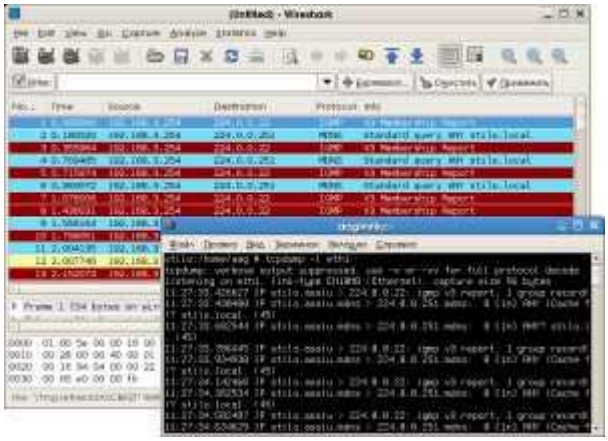


Рис.2.1. Программы анализа трафика. Главное окно программы Wireshark с результатами захвата и программа tcpdump (в консоли).

Онлайн-анализ трафика

В глобальной сети все большее распространение получают онлайн-сервисы, выполняющие мониторинг серверов. Основное назначение таких сервисов - контроль за работоспособностью узлов и оповещение администратора о нештатных ситуациях по эл.почте, через IM и по SMS. Основные проверки выполняются для сервисов прикладного уровня (HTTP, FTP, SMTP, POP3 и т.п.) с возможностью указания интервала проверок. Дополнительными возможностями являются, например, uptime-информеры, средства контроля за появлением вредоносного кода, подключение нескольких ресурсов на аккаунт и т.п. Детальное изучение онлайн-сервисов мониторинга выходит за рамки этой лабораторной работы.

Для выполнения этой работы рекомендуется использовать программы Ethereal Network Analyzer или Wireshark (версии для UNIX/Linux, Windows-версия работает не стабильно). Эти программы практически идентичны как по возможностям, так и по использованию.

Прежде чем приступить к выполнению заданий лабораторной работы, необходимо выполнить следующие действия:

- Установить программу WireShark (см. Управление пакетами в Linux).
- Ознакомиться с кратким руководством пользователя и документацией man (англ.).
- Запустить программу (требуются права суперпользователя) и ознакомиться с пользовательским интерфейсом и основными пунктами меню.

ЛБ3

Сетевые сервисы. Понятие сетевого порта. Контроль состояния портов

Компоненты сетевого приложения связываются через сетевые порты. Каждый сервисный порт имеет собственный номер, по которому клиенты могут подключаться к серверу. Активность клиента влияет на текущее состояние порта.

Цель работы: Ознакомиться с основыми сетевыми сервисами и связанными с ними портами. Научиться использовать команду netstat для контроля за состоянием локальных портов.

Задания к выполнению

1. Запустить netstat в режиме непрерывного вывода. Перенаправить вывод в файл out.txt
2. Инициировать сетевую активность(открыть несколько веб-сайтов, ftp-узлов, запустить торрент-клиент, клиент IM (например ICQ) и т.п.)
3. Закрыть соединения
4. Завершить работу netstat
5. По данным файла out.txt определить:
    - Какие ip-адреса у серверов, к которым были обращения
    - К каким сервисам были подключения (номера портов и названия сервисов)
    - Какие клиентские порты были задействованы
    - Какие приложения (и их компоненты) были задействованы
    - Общая статистика по транспортным протоколам

Методические указания

Сетевые сервисы и порты приложений

Сетевой порт - это предопределенная приложением или процессом точка подключения сетевых приложений, предоставленная операционной системой узла. Порт связан с сетевым адресом хоста (пример явного указания порта: 192.168.0.1**:3128**) и используемым протоколом взаимодействия.

В стеке TCP/IP понятие порта возникает на транспортном уровне, где порт представлен в виде 16-битного числа (номера порта). Протоколы TCP и UDP используют номера портов для идентификации компонентов сетевого приложения в рамках локального хоста. В клиент-серверной модели порты используются для мультиплексирования клиентских подключений.

IANA (Internet Assigned Numbers Authority) представляет список сетевых портов, который входит в поставку современных операционных систем в виде текстового файла services (/etc/services). В нем хранится информация о всех зарегистрированных в IANA службах internet, назначенных им номерах портов и типах сетевых протоколов. Фрагмент этого файла приведен в листинге 1. Подробности - в man 5 services.

Листинг 1. Фрагмент файла services

```
# http://www.iana.org/assignments/port-numbers
# See also: services(5), http://www.sethwklein.net/projects/iana-etc/
http             80/tcp    # World Wide Web HTTP
http             80/udp    # World Wide Web HTTP
www-http         80/tcp    # World Wide Web HTTP
www-http         80/udp    # World Wide Web HTTP
```

```
http             80/sctp   # HTTP
http-mgmt       280/tcp    # http-mgmt
http-mgmt       280/udp    # http-mgmt
https           443/tcp    # http protocol over TLS/SSL
https           443/udp    # http protocol over TLS/SSL
https           443/sctp   # HTTPS
```

Команда netstat

Команда netstat позволяет получить различную информацию о состоянии сетевой подсистемы хоста: статистику сетевых интерфейсов, данные о маршрутизации и сведения о сетевых соединениях. Команда netstat поддерживается всеми операционными системами, использующими стек TCP/IP, но в каждой конкретной реализации могут использоваться разные наборы опций. Список поддерживаемых опций можно получить командой netstat --help (см. листинг).

```
aag@stilo:~>  netstat --help
Использование: netstat [-veenNcCF] [<Af>] -r netstat {-V|--version|-h|--help}
       netstat [-vnNcaeol] [<Socket> ...]
       netstat { [-veenNac] -i | [-cnNe] -M | -s }

         -r, --route      отобразить таблицу маршрутизации
         -i, --interfaces         отобразить таблицу интерфейсов
         -g, --groups     отобразить членства в мултикаст группах
         -s, --statistics         отобразить сетевую статистику (как SNMP)
         -M, --masquerade         отобразить замаскированные соединения

         -v, --verbose    более детальный вывод
         -n, --numeric    не преобразовывать адреса в имена
                 --numeric-hosts  не преобразовывать адреса в имена компьютеров
                 --numeric-ports  не преобразовывать номера портов в имена
                 --numeric-users  не преобразовывать в имена пользователей
         -N, --symbolic   преобразовать имена устройств
         -e, --extend     отображать другую/больше информации
         -p, --programs   отображать номер процесса программы/имя программы для сокетов
         -c, --continuous         непрерывный вывод

         -l, --listening  отображать прослушиваемые сокеты сервера
         -a, --all, --listening   отобразить все сокеты (по умолчанию - в статусе
connected)
         -o, -timers      отобразить таймеры
         -F, -fib         отобразить информацию форвардинга базы (по умолчанию)
         -C, --cache      отобразить кэш маршрутизации вместо FIB

         <Socket>={-t|--tcp} {-u|--udp} {-w|--raw} {-x|--unix} --ax25 --ipx --netrom
         <AF>=Use '-6|-4' or '-A <af>' or '--<af>'; по умолчанию: inet
```

```
        Список возможный адресных семейств (которые поддерживают маршрутизацию):
                inet (DARPA Internet) inet6 (IPv6) ax25 (AMPR AX.25)
                netrom (AMPR NET/ROM) ipx (Novell IPX) ddp (Appletalk DDP)
                x25 (CCITT X.25)
```

Чаще всего команда netstat применяется для решения таких задач, как:

- проверка состояния сетевых соединений;
- анализ информации о конфигурации интерфейсов;
- изучение таблицы маршрутизации;
- получение статистики о различных сетевых протоколах.

Пример использования netstat для получения информации об открытых соединениях по протоколу tcp (см. ключи опции Socket):

```
aag@stilo:~> netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 4stud-new.asoi:www-http localhost:37092         TIME_WAIT
tcp        0      0 aag.asoiu:56574         192.168.3.1:ndl-aas     ESTABLISHED
tcp        0      0 aag.asoiu:56552         192.168.3.1:ndl-aas     ESTABLISHED
tcp        0      0 4stud.asoiu:www-http   localhost:37094         TIME_WAIT
tcp        0      0 4stud.asoiu:www-http   localhost:37095         TIME_WAIT
tcp        0      0 4stud.asoiu:www-http   localhost:37093         TIME_WAIT
tcp        0      0 4stud.asoiu:www-http   localhost:37090         TIME_WAIT
tcp        0      0 4stud.asoiu:www-http   localhost:37091         TIME_WAIT
tcp        0      0 aag.asoiu:56576         192.168.3.1:ndl-aas     ESTABLISHED
tcp        0      0 aag.asoiu:56578         192.168.3.1:ndl-aas     ESTABLISHED
```

ЛБ4

Использование ssh для удаленного управления сервером

> **Secure SHell является основным средством удаленного управления сетевыми компьютерами под управлением UNIX/Linux. Многие хостинг-провайдеры представляют владельцам сайтов доступ к серверам по протоколу ssh.**

Цель работы: Получить начальные навыки работы с удаленным хостом по протоколу ssh.

Задания к работе

1. С помощью команды ssh определите тип операционной системы, аппаратную платформу и версию ядра сервера (команда uname с соответствующими параметрами) без загрузки оболочки сервера.
2. С помощью ssh-клиента подключитесь к серверу edu.asoiu с учетной записью student.
3. Определите, какой каталог является текущим на удаленном сервере. Если он отличается от /home/student, то выполните переход в /home/student.
4. Определите, какие пользователи, в каких терминалах и с каких ip-адресов подключены к серверу (команда who)
5. В текущем каталоге создайте каталог *ваша_фамилия_транслитом* (например, ivanov).
6. Запустите файловый менеджер mc, просмотрите содержимое текущего каталога.
7. Завершите работу с файловым менеджером mc.
8. Завершите сеанс ssh.
9. Не входя в сеанс ssh, загрузите в ранее созданный вами на сервере каталог произвольный файл (команда scp).
10. Не входя в сеанс ssh, просмотрите содержимое серверного каталога /home/student .
11. Составить отчет о выполнении этой работы. Загрузить файл отчета с помощью ssh в созданный вами в ходе выполнения этой работы каталог на сервере.

Методические указания

Средства удаленного управления серверами широко используются как в локальных, так и в глобальных сетях. Основное назначение таких средств - организация канала передачи (в общем случае - через шлюз) команд управления сервером и возврат клиенту результата выполнения этих команд. Поэтому одними из основных требований к программам удаленного управления являются прозрачность для пользователя и небольшой трафик. Это позволяет централизованно управлять территориально распределенными узлами с одного рабочего места или представлять доступ удаленным терминальным клиентам по медленным линиям связи.

Сетевые шлюзы

Подключение локальных сетей к Интернет обычно реализуется через коммуникационный сервер - **шлюз** (gateway). Такой сервер обычно работает под управлением какой-нибудь версии UNIX (FreeBSD, Debian GNU/Linux, RHEL и т.п.) и имеет как минимум 2 сетевых интерфейса: один внутренний, к которому подключена ЛВС, второй - внешний, обращенный в Интернет. На шлюзе настроена маршрутизация, подняты сервисы NAT и прокси. Опционально могут быть подняты дополнительные сервисы (баннерорезка, учет трафика и т.п.). В руководстве С.Лазаренко «Установка и настройка интернет шлюза на Debian» детально описана процедура развертывания шлюза на основе Debian GNU/Linux (Lenny и Squeeze) с использованием iptables (файрвол+NAT) и SQUID (прокси).

Для управления используются как символьные протоколы (telnet, rlogin, ssh), так и бинарные, поддерживающие графические возможности (много разных). Для выделенных серверов графические средства как правило не используются, поскольку такие серверы не подразумевают использование их как

рабочих станций. Это означает, что нет необходимости выделять ресурсы (существенные!) для графического пользовательского интерфейса.

Текстовые протоколы tenet и rlogin просты и функциональны, но небезопасны. Исходя из указанных соображений в текущей работе предполагается освоение удаленного управления UNIX-сервером по протоколу ssh. Следует отметить, что протокол ssh поддерживает и работу с графическим режимом (тунеллирование X-сервера). Более того, протокол ssh позволяет туннеллировать любой сетевой трафик, использующий в качестве транспорта протокол TCP. (Подробности - в спецификациях протокола The Secure Shell (SSH) Protocol Assigned Numbers, RFC 4250, 2006; The Secure Shell (SSH) Protocol Architecture, RFC 4251, 2006; The Secure Shell (SSH) Authentication Protocol, RFC 4252, 2006 и др.)

Для управления сервером по протоколу ssh необходима его поддержка сервером и клиентское ssh-приложение. UNIX-серверы стандартно поддерживают протокол ssh. В качестве клиента как правило используется OpenSSH (вызывается командой ssh). Для Windows имеются клиенты разных производителей, наиболее популярные PuTTY и SecureCRT. В лабораторной работе предполагается использование клиента OpenSSH. Подробности использования этой программы доступны в руководстве man (man 1 ssh).

Некоторые команды файловой системы Linux рассмотрены в лабораторной работе №3 по дисциплине «Открытое ПО»

Подключение к серверу

Для подключения к серверу необходимо выполнить команду ssh, указав в качестве параметров имя пользователя и имя или сетевой адрес сервера:

```
aag@stilo:~> ssh -l student edu.asoiu // -l student - логин; edu.asoiu - ssh-сервер
```

Или так:

```
aag@stilo:~> ssh student@edu.asoiu
```

Если вы выполните ssh, не задав имя пользователя, то серверу будет отправлено имя текущего локального пользователя.

При первом ssh подключении к удалённой машине, вы увидите подобное сообщение:

```
The authenticity of host 'edu.asoiu' can't be established.
DSA key fingerprint is 94:68:3a:3a:bc:f3:9a:9b:01:5d:b3:07:38:e2:11:0c.
Are you sure you want to continue connecting (yes/no)?
```

Введите *yes* для продолжения. При этом сервер будет добавлен в ваш список известных серверов, о чём говорит следующее сообщение:

```
Warning: Permanently added 'edu.asoiu' (DSA) to the list of known hosts.
```

Затем будет выведено приглашение для ввода пароля удалённого компьютера. После авторизации на сервере пользователь попадает в оболочку UNIX (как правило это одна из версий shell, зависит от настроек сервера) и может приступать к вводу команд, так как если бы работал на локальной машине.

Выполнение команд

Для пользователя выполнение команд на удаленном сервере мало чем отличается от обычной работы с локальными командами и файлами. Приведем несколько примеров:

1. Определение текущего каталога:

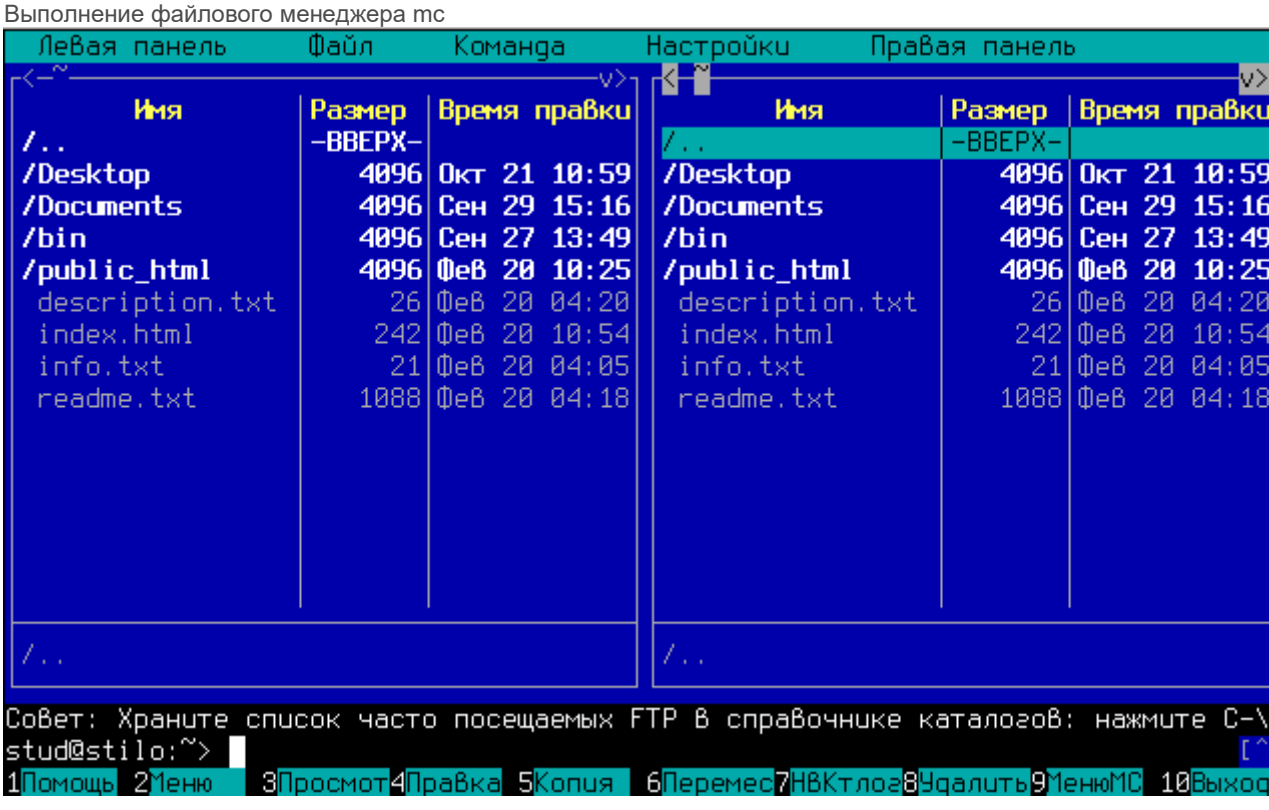
   ```
   aag@stilo:~> ssh student@edu.asoiu
   ```

```
Password: // пароль при вводе не отображается
Last login: Thu Feb 21 11:08:13 2008
Have a lot of fun... // авторизация прошла успешно
student@stilo:~> pwd // ввод команды
/home/student // вывод результатов выполнения
student@stilo:~> // приглашение оболочки
```

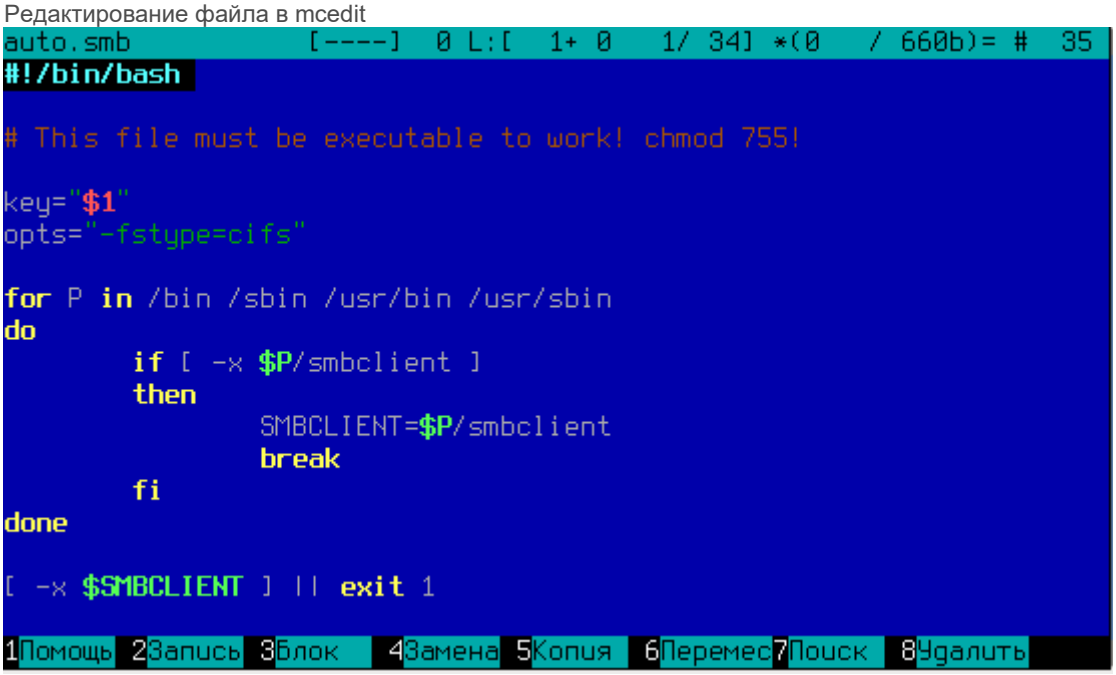
2. Просмотр списка файлов каталога:

```
student@stilo:~> ls -ABl
итого 212
-rw------- 1 student users   782 Фев 20 12:33 .bash_history
-rw-r--r-- 1 student users  1177 Сен 27 13:49 .bashrc
drwx------ 7 student nobody 4096 Окт 21 11:09 .beagle
drwxr-xr-x 2 student users  4096 Сен 27 13:49 bin
drwx------ 2 student nobody 4096 Окт 21 10:59 .config
-rw-r--r-- 1 student nobody   26 Фев 20 04:20 description.txt
drwxr-xr-x 2 student nobody 4096 Окт 21 10:59 Desktop
-rw------- 1 student nobody   24 Окт 21 10:59 .dmrc
drwx------ 2 student users  4096 Сен 29 15:16 Documents
-rw-r--r-- 1 student users   208 Сен 27 13:49 .dvipsrc
-rw-r--r-- 1 student users  1637 Сен 27 13:49 .emacs
...
```

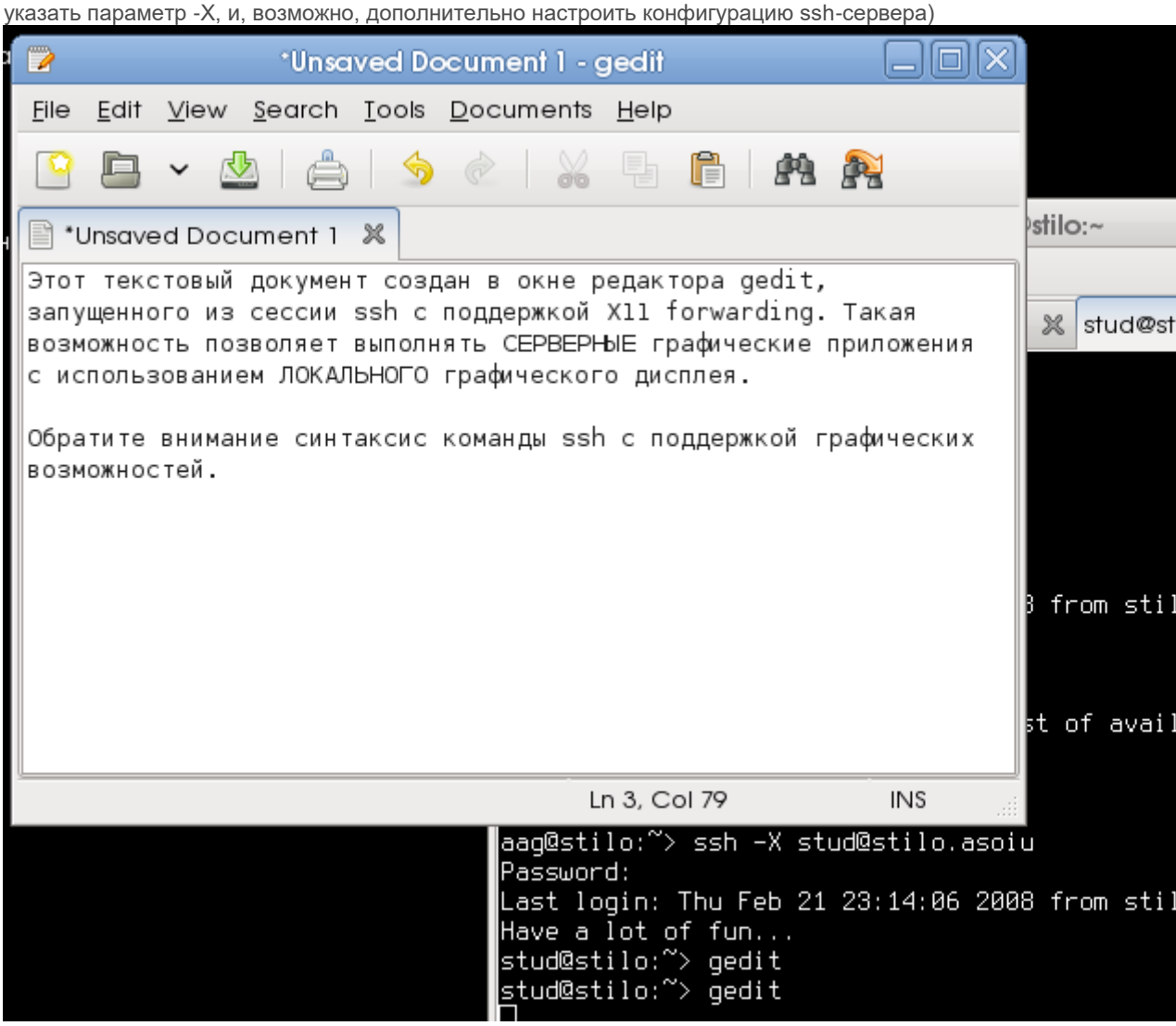
3. Выполнение файлового менеджера mc



4. Редактирование файла в mcedit



5. Редактор gedit, запущенный из сессии ssh (при установлении соединения с сервером требуется указать параметр -X, и, возможно, дополнительно настроить конфигурацию ssh-сервера)



Команда ssh может применяться для выполнения команд на удалённом компьютере и без загрузки удаленной оболочки. В этом случае требуемая команда передается в качестве параметра ssh:

```
ssh <hostname> <command>
```

Например, если вы хотите выполнить команду ls /usr/share/doc на удалённом компьютере edu.asoiu, введите в приглашении оболочки:

```
ssh edu.asoiu ls /usr/share/doc
```

Когда вы введёте правильный пароль, на экране появится содержимое /usr/share/doc, и вы вернётесь к своему приглашению оболочки. Следует отметить, что не все команды могут быть выполнены таким образом. Пример отказа при попытке запустить mc:

```
aag@stilo:~> ssh student@edu.asoiu mc
Password:
Cannot get terminal settings: Недопустимый аргумент (22) // Ошибка! Невозможно получить параметры терминала
TERM environment variable needs set.
aag@stilo:~>
```

Копирование файлов

Для передачи файлов между компьютерами через защищённое соединение используют команду scp (secure copy).

Для передачи локального файла на удалённый компьютер эта команда используется в виде:

```
scp localfile username@tohostname:/newfilename
```

localfile обозначает исходный файл, а комбинация *username@tohostname:/newfilename* определяет назначение.

Чтобы передать локальный файл somefile на компьютер edu.asoiu под своим именем пользователя, введите в приглашении оболочки (заменив *username* своим именем):

```
scp somefile username@edu.asoiu:/home/username
```

При этом локальный файл somefile скопируется в /home/*username*/somefile на сервере edu.asoiu.

Для передачи удалённого файла на локальный компьютер используется команда

```
scp username@tohostname:/remotefile /newlocalfile
```

remotefile обозначает исходный файл, а *newlocalfile* - назначение.

В качестве исходных файлов могут быть указаны несколько файлов. Например, чтобы передать содержимое каталога /downloads в существующий каталог с именем uploads на удалённый компьютер edu.asoiu, введите в приглашении оболочки следующую команду:

```
scp /downloads/* username@edu.asoiu:/uploads/
```

Завершение сеанса ssh

Для завершения работы по протоколу ssh необходимо выполнить команду exit, которая завершает пользовательский сеанс и прерывает соединение:

```
...
```

```
-rw-r--r-- 1 student nobody 1088 Фев 20 04:18 readme.txt
-rw-r--r-- 1 student nobody 2194 Окт 21 11:04 .recently-used.xbel
drwxr-xr-x 2 student nobody 4096 Окт 21 10:59 .skel
drwx------ 2 student nobody 4096 Сен 27 15:13 .ssh
student@stilo:/etc> exit
logout // сеанс пользователя завершен
Connection to edu.asoiu closed. // соединение закрыто
```

ЛБ5

Установка и настройка ftp-сервера

> **VSFTD - сервис, реализующий функции файлового сервера по протоколу ftp. Все настройки vsftpd хранятся в файле конфигурации /etc/vsftpd.conf.**

Цель работы: Изучение принципов сетевого взаимодействия с файловым сервером по протоколу ftp.

Задания к работе

1. Выполнить установку ftp-сервера vsftpd на локальный хост.
2. Сконфигурировать установленный ftp-сервер следующим образом:
    - разрешить анонимный доступ
    - разрешить анонимным пользователям создание каталогов
    - разрешить анонимным пользователям запись файлов
3. Запустить vsftpd.
4. Командой ftp подключиться к серверу ftp://localhost.
5. Определить настройки сервера (команда status)
6. Выяснить, какой каталог является текущим
7. Закрыть соединение
8. Локально создать от имени суперпользователя в каталоге /srv/ftp подкаталоги pub, temp и upload с правами доступа 755, 777, 733 соответственно (см. управление правами доступа).
9. Написать пакетный файл, выполняющий следующие действия:
    - Переход в каталог temp
    - Загрузка произвольного локального файла в каталог temp
    - Скачивание этого файла в домашний каталог
    - Переход в каталог pub
    - Загрузка произвольного локального файла в каталог pub
    - Скачивание этого файла в домашний каталог
    - Переход в каталог upload
    - Загрузка произвольного локального файла в каталог upload
    - Скачивание этого файла в домашний каталог
    - Отображение списка файлов в каталогах temp, pub и upload
    - Завершение работы с ftp-сервером
10. Анонимно подключиться к ftp-серверу localhost и выполнить пакетные команды из файла, созданного в предыдущем задании. В отчете объяснить причины отличий в результатах выполнения этих пакетных команд.
11. Изменить конфигурацию vsftpd таким образом, чтобы разрешить вход для локальных пользователей.
12. Подключиться к ftp-серверу localhost с учетной записью student и определить, какие каталоги будут доступны для этого пользователя (для этого можно выполнить, например, команды cd / и pwd)
13. Установить параметр chroot_local_user=yes в vsftpd.conf и повторить предыдущее задание. Как отличаются результаты выполнения этого и предыдущего заданий?
14. Завершить все ftp-сеансы.
15. Остановить сервер vsftpd
16. Удалить пакет vsftpd из системы

Указания к лабораторной работе

Сервер vsftpd

vsftpd (Very Secure File Transfer Protocol Daemon, очень безопасный ftp-сервер) — один из наиболее распространенных ftp-серверов для UNIX-систем. Сервер может быть запущен как служба через суперсерверы inetd или xinetd, также vsftpd можно использовать самостоятельно (standalone mode). vsftpd поддерживает все возможности, представляемые протоколом ftp, а также включает множество полезных функций, таких как:

- повышенная безопасность
- контроль над полосой пропускания канала
- расширяемость
- поддержка виртуальных пользователей
- поддержка IPv6
- высокая производительность
- возможность устанавливать виртуальные IP-адреса

Установка vsftpd осуществляется обычным образом и зависит только от формата пакета. Например, для установки в систему на базе rpm можно использовать одноименную программу или программы типа yum, zypper и подобные. Подразумевается, что репозитарии ПО настроены и доступны.

Примеры установки

1. Установка из rpm

```
edu.asoiu:~/soft # ls -l --color=no
-r--r--r-- 1 root root 122840 Июн  6  2008 vsftpd-2.0.6-25.1.i586.rpm
edu.asoiu:~/soft # rpm -ivh vsftpd*
Подготовка...     ########################################### [100%]
1:vsftpd          ########################################### [100%]
edu.asoiu:~/soft #
```

2. Установка из репозитария

```
edu.asoiu:~/ # zypper install vsftpd
Загружаются метаданные репозитария 'asoiu' [готово]
Собирается кэш репозитария 'asoiu' [готово]
Чтение установленных пакетов...
Будет установлен следующий НОВЫЙ пакет:
  vsftpd
Полный размер загрузки: 120,0 K. После этой операции будет использовано
дополнительно 274,0 K.
Продолжить? [да/нет]: yes
Загружается пакет vsftpd-2.0.6-25.1.i586 (1/1), 120,0 K (274,0 K
нераспакованный)
Загружается: vsftpd-2.0.6-25.1.i586.rpm [готово]
Устанавливается: vsftpd-2.0.6-25.1 [готово]
edu.asoiu:~/ #
```

3. Проверка установки

```
edu.asoiu:~/ # rpm -q vsftpd
vsftpd-2.0.6-25.1
```

Настройка vsftpd сводится к изменению записей в конфигурационнм файле /etc/vsftpd.conf. Структура этого файла проста: все опции задаются парами вида <параметр=значение>, где <параметр> — название опции, <значение> — непосредственное значение. Параметры могут быть следующих типов:

1. логический (yes/no)
2. строковый (строка символов)
3. числовой (целое число)

Строки комментариев начинаются с символа #("решетка") и игнорируются.

Опции настроек могут быть *независимыми*, *зависимыми* и *взаимоисключающими*. В первом случае изменение опции не влияет на прочие настройки, во втором — значение опции зависит от— или влияет на другие настройки, в третьем — значение опции противоречит или отменяет другие параметры (подробнее — man vsftpd.conf).

Перечислим некоторые, наиболее часто используемые, настройки vsftpd:

```
# Это  —  комментарий
# Разрешать ли анонимный доступ? (yes/no)
anonymous_enable=yes
# Разрешать ли загрузку файлов анонимному пользователю? (yes/no)
anon_upload_enable=yes
# Разрешать ли анонимному пользователю создавать свои директории? (yes/no)
anon_mkdir_write_enable=yes
# Разрешать ли пользователю производить операции с записью, такие как
# переименование или удаление? (yes/no)
anon_other_write_enable=yes
# Разрешить вход локальных пользователей? (yes/no)
local_enable=yes
# Должны ли пользователи находится только в своих директориях? (yes/no)
chroot_local_user=yes
# Максимальная скорость передачи данных для зарегистрированных пользователей.
# По умолчанию = 0 (неограниченная).
local_max_rate=7200
# Разрешать ли  запись в каталог? (yes/no)
write_enable=yes
# Включать сообщения при смене директории? (yes/no)
dirmessage_enable=yes
# Показ баннера при регистрации пользователя.
ftpd_banner="edu.asoiu: What?.. What do you want?.."
# Включить регистрацию событий (журналирование)? (yes/no)
xferlog_enable=yes
# Записывать в лог все активные FTP-соединения? (yes/no)
```

```
# Осторожно! Возможно на экране будет избыточная информация.
log_ftp_protocol=no
# Разрешать соединения только на порт 20 (ftp data)? (yes/no)
connect_from_port_20=yes
# Тайм-аут сессии (при бездействии), секунд
idle_session_timeout=600
# Тайм-аут передачи данных (при неудачном подключении), секунд
data_connection_timeout=120
```

Домашний каталог ftp

vsftpd, как и другие UNIX-серверы, выполняется от имени непривилегированного пользователя (обычно, ftp или nobody). Для этого пользователя создается домашний каталог, который и является основным для анонимных ftp-пользователей. Путь к этому каталогу указывается в файле /etc/passwd и узнать его можно, например, так:

```
root@edu.asoiu:~> cat /etc/passwd | grep ftp
ftp:x:40:49:FTP account:/srv/ftp:/bin/bash
```

Запуск vsftpd может выполняться в автоматическом или ручном режимах. Первый способ предусматривает использование *inetd* или *xinetd*, второй — непосредственно из командной строки. Здесь будет рассмотрен автоматический запуск через inetd с использованием скрипта service:

```
edu.asoiu:~ # service vsftpd start  # запуск ftp-сервера
Starting vsftpd                                            done
edu.asoiu:~ # service --status-all | grep vsftpd # проверка состояния
...
Checking for service vsftpd ..running
edu.asoiu:~ #
```

Для автоматической загрузки vsftpd при перезагрузке системы добавим его в автозапуск на 3 и 5 уровнях:

```
chkconfig -a vsftpd 35
vsftpd          0:off  1:off  2:off  3:on   4:off  5:on   6:off
```

После внесения изменений в конфигурацию, vsftpd должен быть перезагружен. Это можно сделать так:

```
edu.asoiu:~ # service vsftpd restart
Shutting down vsftpd                                              done
Starting vsftpd                                                   done
edu.asoiu:~ #
```

После успешной установки и настройки сервер vsftpd готов обслуживать подключения ftp-клиентов.

ftp-клиенты

Работа с файловым серверами по протоколу ftp осуществляется с помощью специализированных программ: ftp-клиентов. Как правило, в составе сетевых операционных систем имеются такие приложения (например, программа-клиент ftp, так и называется, причем как в Windows-, так и в UNIX-системах). Клиенты ftp могут иметь как консольный (ftp,mc, far), так и графический интерфейс пользователя (gftp, Total Commander). Эти программы представляют все средства работы с ftp-сервером, от просмотра списка файлов до управления

файловой системой (зависит от настроек сервера). Ограниченные возможности могут представляют и другие сетевые приложения, например, веб-браузеры, которые позволяют только скачивать файлы с ftp-сервера.

В качестве примера рассмотрим ftp-сеанс с использованием программы ftp.

Подключение к серверу

1. с явными указанием имени хоста в командной строке:

```
aag@stilo:~> ftp edu.asoiu
Connected to edu.asoiu.
220 "edu.asoiu: What?.. What do you want?.."
...
```

2. в интерактивном режиме ftp через команду open:

```
aag@stilo:~> ftp //запуск ftp-клиента в интерактивном режиме
ftp> open edu.asoiu // открытие соединения с сервером
Connected to edu.asoiu.
220 "edu.asoiu: What?.. What do you want?.."
...
```

Авторизация на сервере

Протокол ftp предусматривает два способа доступа пользователей к серверу:

1. анонимный (с использованием предопределенного имени пользователя anonymous, пароль — любой адрес электронной почты, некоторые сервера поддерживают вход без пароля);
2. авторизованный (с использованием учетной записи пользователя).

Имя пользователя можно указать явно, на этапе подключения к серверу:

```
aag@stilo:~> ftp student@edu.asoiu
Connected to edu.asoiu.
220 "edu.asoiu: What?.. What do you want?.."
331 Please specify the password.
Password:...
```

В интерактивном режиме авторизация выполняется по команде user, пароль задается командой pass (или в виде приглашения от сервера):

```
aag@stilo:~> ftp edu.asoiu
Connected to edu.asoiu.
220 "edu.asoiu: What?.. What do you want?.."
Name (edu.asoiu:aag): user // сервер запрашивает имя пользователя...
331 Please specify the password.
Password: // ... и его пароль
530 Login incorrect. // некорректный логин
ftp: Login failed.
ftp> user stud // явный вызов команды ввода логина
```

```
331 Please specify the password.
Password:                 // запрос пароля со стороны сервера (пароль не отображается)
230 Login successful. // вход выполнен
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> // установление сеанса и начало работы
...
```

Выполнение команд ftp

Протокол ftp определяет два типа команд: команды управления файлами и команды протокола. Первый тип команд предназначен для работы с файловой системой (табл. 1), эти команды доступны после установления соединения. Второй тип предназначен для управления ftp-сеансом как таковым и набор поддерживаемых команд может отличаться у различных ftp-серверов. Полный список команд обоих типов можно получить по командам *help* и *rhelp* соответственно.

Таблица 1. Некоторые команды ftp

Команды	Назначение
pwd	получение информации о текущем каталоге
dir [*каталог*], ls [*каталог*]	вывод списка файлов текущего [*заданного*] каталога
cd *путь* | .. | .	переход в **серверный** каталог, указанный параметром *путь* или в родительский каталог (..), или в корневой каталог (.)
lcd *путь*	переход в **локальный** каталог, указанный параметром *путь*
mkdir *имя_каталога*	создание каталога с указанным именем на сервере
rmdir *имя_каталога*	удаление указанного каталога на сервере
put *имя_локального_файла*[*имя_файла_на_сервере*]	передача указанного файла из текущего каталога на сервер [с новым именем]
get *имя_файла_на_сервере*[*имя_локального_файла*]	получение (скачивание) указанного файла с сервера в текущий локальный каталог
append *имя_локального_файлаимя_удаленного_файла*	добавление информации из локального файла в конец удаленного файла
help [*имя_команды*]	список команд управления файлами. Если задано *имя_команды*, то отображается справка о ней.
rhelp	список команд управления сеансом, поддерживаемых сервером.

В интерактивном режиме ввод команд выполняется в строке приглашения. Некоторые команды требуют указания одного или более параметров, которые разделяются пробелами. Далее приведено несколько примеров (обратите внимание на числа в начале строк — это коды отклика, возвращаемые сервером по результатам выполнения команд):

1. Просмотр списка файлов

```
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
229 Entering Extended Passive Mode (|||40049|)
150 Here comes the directory listing.
drwxr-xr-x    2 1001     65533        4096 Oct 21 03:59 Desktop
drwx------    2 1001     100          4096 Sep 29 08:16 Documents
drwxr-xr-x    2 1001     100          4096 Sep 27 06:49 bin
-rw-r--r--    1 1001     65533          26 Feb 19 22:20 description.txt
-rw-r--r--    1 1001     65533         242 Feb 20 04:54 index.html
-rw-r--r--    1 1001     65533          21 Feb 19 22:05 info.txt
drwxr-xr-x    2 1001     100          4096 Feb 20 04:25 public_html
-rw-r--r--    1 1001     65533        1088 Feb 19 22:18 readme.txt
226 Directory send OK.
ftp>
...
```

2. Перемещение в другой каталог

```
ftp> cd public_html
250 Directory successfully changed.
ftp>
...
```

3. Смена прав доступа к удаленному файлу

```
ftp> chmod 755 index.html
550 SITE CHMOD command failed. // ОШИБКА! Настройки сервера не позволяют смену прав доступа
ftp>
...
```

4. Скачивание удаленного файла с переименованием на клиентской стороне

```
ftp> get index.html ind.html // файл index.html будет сохранен как ind.html
local: ind.html remote: index.html
ftp: local: ind.html: Permission denied // ОШИБКА! Запись в текущий локальный каталог запрещена
ftp> lcd ~ // смена локального каталога (переход в домашний каталог пользователя)
Local directory now /home/aag
ftp> get index.html ind.html // вторая попытка скачать файл
```

local: ind.html remote: index.html

229 Entering Extended Passive Mode (|||40273|)

150 Opening BINARY mode data connection for index.html (550 bytes).

100% |**********************************| 550 8.64 KB/s 00:00 ETA

226 File send OK. // *успешно*

550 bytes received in 00:00 (8.55 KB/s) // *скорость и время операции*

ftp>

...

5. Дописывание информации из локального в удаленный файл
6. ftp> **append**
7. (local-file) *myfile.txt* // локальный файл
8. (remote-file) *index.html* // удаленный файл
9. local: myfile.txt remote: index.html
10. 229 Entering Extended Passive Mode (|||40437|)
11. 150 Ok to send data.
12. 100% |**********************************| 20 164.12 KB/s 00:00 ETA
13. 226 File receive OK.
14. 20 bytes sent in 00:00 (19.29 KB/s)
15. ftp>
16. Отображение состояния ftp-сервера (текущие настройки сервера и соединения)

...

ftp> **status**

Connected and logged into edu.asoiu.

No proxy connection. //*прямое соединение*

Gate ftp: off, server (none), port ftpgate. //*шлюз не используется*

Passive mode: on; fallback to active mode: on.//*пассивный режим включен*

Mode: stream; Type: binary; Form: non-print; Structure: file. //*потоковая передача бинарных файлов*

Verbose: on; Bell: off; Prompting: on; Globbing: on. //*опции пользов. интерфейса*

...

17. Получение списка команд, поддерживаемых сервером

ftp> **help**

Commands may be abbreviated. Commands are:

!	features	mls	proxy	size
$	fget	mlsd	put	sndbuf
account	form	mlst	pwd	status

```
append          ftp             mode            quit            struct
ascii           gate            modtime         quote           sunique
bell            get             more            rate            system
binary          glob            mput            rcvbuf          tenex
bye             hash            msend           recv
throttle
case            help            newer           reget           trace
cd              idle            nlist           remopts         type
cdup            image           nmap            rename          umask
chmod           lcd             ntrans          reset           unset
close           less            open            restart         usage
cr              lpage           page            rhelp           user
debug           lpwd            passive         rmdir           verbose
delete          ls              pdir            rstatus         xferbuf
dir             macdef          pls             runique         ?
disconnect      mdelete         pmlsd           send
edit            mdir            preserve        sendport
epsv4           mget            progress        set
exit            mkdir           prompt          site
ftp>_
```

Завершение сеанса ftp

```
stilo:~> ftp student@edu.asoiu
... // выполнение различных команд ftp
ftp> disconnect // Отключение от сервера. Можно выполнить новое подключение (open)
221 Goodbye.
ftp> quit // Завершение работы с ftp-клиентом
stilo:~>
```

Пакетный режим

Некоторые ftp-клиенты поддерживают пакетный режим работы. В этом режиме вся последовательность ftp-команд предварительно записывается в текстовый файл. Этот файл задается в виде параметра или по цепочке на вход ftp-клиента, который последовательно считывает эти команды и выполняет их.

Рассмотрим пример работы в пакетном режиме. Пусть требуется подключиться к серверу edu.asou, получить имя текущего каталога для указанного ftp-пользователя и список файлов в этом каталоге, а затем отключиться от сервера. Для этого создадим файл ftp.batch (имя может быть любое) следующего содержания:

```
pwd
dir
quit
```

Теперь, чтобы получить имя домашнего каталога на ftp-сервере и список файлов в нем для пользователя student, можно выполнить такую команду (обратите внимание на передачу логина и пароля):

```
stilo:~> ftp ftp://student:student@edu.asoiu < ftp.batch
Connected to edu.asoiu.
220 edu.asoiu: What?.. What do you want?..
331 Please specify the password.
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
200 Switching to Binary mode. // выполнение pwd
257 "/home/student"
229 Entering Extended Passive Mode (|||40185|)
150 Here comes the directory listing.// выполнение dir
drwxr-xr-x    2 1000     100          4096 Sep 02 08:04 Documents
-rw-r--r--    1 1000     100             0 Nov 03 06:10 ssh
226 Directory send OK.
221 Goodbye.// выполнение quit
stilo:~>
```

Контрольные вопросы

1. В каком из режимов (активном или пассивном) работает учебный ftp-сервер?
2. Какой режим передачи по умолчанию использует учебный ftp-сервер?
3. Какой код ответа будет выведен при удачном подключении к серверу?
4. Как определить, что на ftp-сервере имеются каталоги, доступные для записи?
5. Какие коды ответа будут выведены при удачной и неудачной смене каталога?

A