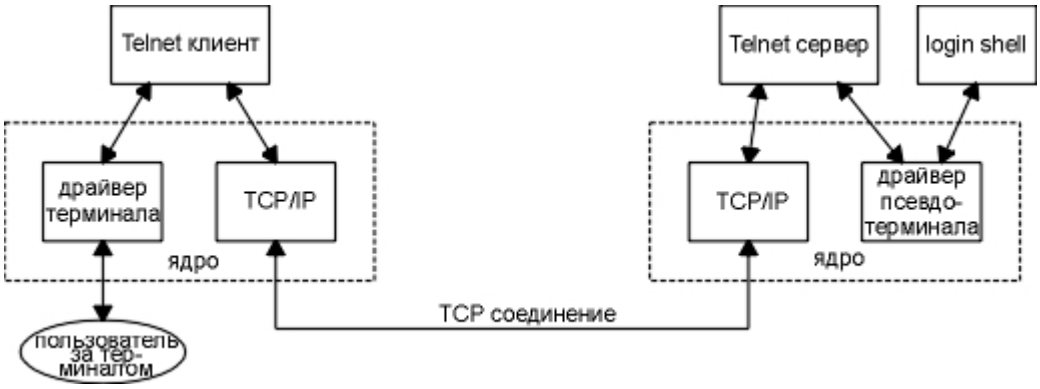
Протокол Telnet

Telnet был разработан для работаты между хостами под управлениием любых операционных систем, а также с любыми терминалами. Его спецификация определяет терминал, который может являться наиболее общим, и который называется виртуальным сетевым терминалом (NVT - network virtual terminal). NVT это воображаемое устройство, находящееся на обоих концах соединения, у клиента и сервера, с помощью которого устанавливается соответствие между их реальными терминалами. Таким образом, операционная система клиента должна определять соответствие между тем типом терминала, за которым работает пользователь, с NVT. В свою очередь, сервер должен устанавливать соответствие между NVT и теми типами терминалов, которые он (сервер) поддерживает.



NVT это символьное устройство с клавиатурой и принтером. Данные, введенные пользователем с клавиатуры, отправляются серверу, а данные, полученные от сервера, поступают на принтер. По умолчанию клиент отражает эхом на принтер все, что ввел пользователь, однако, ниже мы увидим что, существуют опции, которые позволяют изменить подобное поведение.

NVT ASCII

Термин NVT ASCII означает 7-битный вариант U.S. ASCII набора символов, который используется в семействе протоколов Internet. Каждый 7-битный символ отправляется как 8-битный байт со старшим битом установленным в 0.

Конец строки передается как двухсимвольная последовательность - CR (возврат каретки - carriage return), затем следует LF (пропуск строки - linefeed). В протоколах FTP, SMTP, Finger и Whois используют NVT ASCII для ввода команд клиента и откликов сервера.

Команды Telnet

Telnet использует команды в полосе (in-band signaling) в обоих направлениях. Байт 0xff (255 десятичный) называется IAC, "интерпретировать как команду". Следующий байт является командным байтом. Для того чтобы послать байт данных равный 255, отправляются два последовательных байта равных 255. (Выше было указано, что поток данных имеет формат NVT ASCII, то есть используются 7-битные значения, а это означает, что байт данных равный 255 не может быть отправлен посредством Telnet. Однако существует опция Telnet, описанная в RFC 856 [Postel and Reynolds 1983b], которая, позволяет передавать 8-битные данные.)

Имя	Код	Описание

	(десятичный)	
EOF	236	конец файла
SUSP	237	подавить текущий процесс (управление задачами)
ABORT	238	прекратить процесс
EOR	239	конец записи
SE	240	конец подопции
NOP	241	пустая операция
DM	242	маркер данных
BRK	243	прерывание
IP	244	прервать процесс
AO	245	прекратить вывод
AYT	246	вы здесь?
EC	247	escape символ
EL	248	стереть строку
GA	249	идем дальше
SB	250	начало подопции
WILL	251	обсуждение опции (рисунок 26.9)
WONT	252	обсуждение опции
DO	253	обсуждение опции
DONT	254	обсуждение опции
IAC	255	байт данных 255

Команды Telnet, предваряемые IAC (255).

Большинство из этих команд используется достаточно редко.

Обсуждение опций

Несмотря на то, что при начале работы Telnet подразумевается, что на каждом конце находится NVT, первый обмен данными, который происходит по Telnet соединению, является обсуждением опций. Обсуждение опций это симметричный процесс - каждая сторона может послать запрос другой.

Каждая сторона может послать один из четырех различных запросов для любой заданной опции.

1. WILL. Отправитель хочет включить эту опцию для себя.
2. DO. Отправитель хочет, чтобы получатель включил эту опцию.
3. WONT. Отправитель хочет выключить эту опцию для себя.
4. DONT. Отправитель хочет, чтобы получатель выключил опцию.

Так как правила Telnet позволяют стороне принять или отклонить запрос на включение опции (случаи 1 и 2), однако требуют, чтобы она всегда удовлетворяла запрос на выключение опции (случаи 3 и 4), из этих четырех возможных случаев может получиться шесть комбинаций:

	Отправитель		Получатель	Описание
1.	WILL	-> <-	DO	отправитель хочет включить опцию получатель говорит ДА
2.	WILL	-> <-	DONT	отправитель хочет включить опцию получатель говорит НЕТ
3.	DO	-> <-	WILL	отправитель хочет, чтобы получатель включил опцию получатель говорит ДА
4.	DO	-> <-	WONT	отправитель хочет, чтобы получатель включил опцию получатель говорит НЕТ
5.	WONT	-> <-	DONT	отправитель хочет выключить опцию получатель должен сказать ДА
6.	DONT	-> <-	WONT	отправитель хочет, чтобы получатель выключил опцию получатель должен сказать ДА

Шесть сценариев обсуждения опции Telnet.

Обсуждение опции занимает 3 байта: IAC байт, за которым следует байт WILL, DO, WONT или DONT, затем ID байт, указывающий на ту опцию, которую необходимо включить или выключить. Таким образом, может быть обсуждено 40 опций. Описания опций и их кодирование специфицировано в нескольких документах RFC (Request for Comments). Примерами могут служить:

ID опции (десятичный)	Имя	RFC
1	эхо	857
3	запрещение команды go ahead	858
5	статус	859
6	маркер времени	860
24	тип терминала	1091
31	размер окна	1073
32	скорость терминала	1079
33	удаленный контроль потоком данных	1372
34	линейный режим (linemode)	1184
36	переменные окружения	1408

Коды опций Telnet.

Обсуждение опции Telnet, как и многое другое в протоколе Telnet, процесс симметричный. Каждая сторона может начать процесс обсуждения опции. Однако заход удаленным терминалом не является симметричным процессом. Клиент решает свои задачи, а сервер свои. Некоторые опции Telnet применимы только к клиенту (например, требование включить линейный режим (linemode)), а некоторые предназначены только для сервера.

Обсуждение подопций

Некоторые опции требуют большего количества информации, нежели просто "включить" (enable) или "выключить" (disable). Например, установка типа терминала: для того чтобы клиент мог идентифицировать тип терминала, он должен отправить ASCII строку. Чтобы обработать эти опции, применяется обсуждение подопций.

Если клиент просит включить опцию, отправляя 3-байтовую последовательность

<IAC, WILL, 24>

где 24 (десятичное) это идентификатор опции типа терминала. Если получатель (сервер) говорит ДА, его ответ будет выглядеть как

<IAC, DO, 24>

Затем сервер посылает

<IAC, SB, 24, 1, IAC, SE>

спрашивая о типе терминала клиента. SB это команда, которая сообщает о начале подопций (suboption-begin). Следующий байт равный 24 указывает на то, что это

подопция типа терминала. (SB всегда следует за номером опции, к которой относятся подопции.) Следующий байт равный 1 означает "отправьте ваш тип терминала". Перед командой конец подопций (suboption-end) должен опять стоять IAC, так же как и перед командой SB. Клиент отвечает командой

<IAC, SB, 24, 0, 'I', 'B', 'M', 'P', 'C', IAC, SE>

в случае, если его тип терминала ibmpc. Четвертый байт равный 0 означает "у меня следующий тип терминала". ("Официальный" список приемлемых типов терминалов находится в Assigned Numbers RFC, однако для Unix систем приемлем любой тип терминала, поддерживаемый сервером. Обычно это терминалы, поддерживаемые базами termcap или terminfo.) Типы терминалов, указываемые в подопциях Telnet, пишутся большими буквами и обычно преобразуются в маленькие буквы уже сервером.

Полудуплексный, символ за один раз, строка за один раз или линейный режим (Linemode)?

Существуют четыре режима, в которых функционирует большинство Telnet клиентов и серверов:

1. Полудуплексный.

Этот режим используется редко. (NVT по умолчанию это полудуплексное устройство, которое требует исполнения команды GO AHEAD (GA) от сервера, перед тем как будет принят ввод от пользователя. Ввод пользователя отображается локальным эхом от NVT клавиатуры на NVT принтер, таким образом, от клиента к серверу посылаются только полные строки.)

2. Символ за один раз.

Именно таким образом работает Rlogin. Каждый вводимый символ отправляется серверу отдельно от других. Сервер отражает эхом большинство символов, если только у приложения на сервере не отключено отражение эхом.

Проблемы, связанные с этим режимом, в основном связаны с задержками, вызванными отражением эхом по медленным сетям, и с большим объемом сетевого трафика. Однакона это наиболее распространенный режим и являющийся к тому же режимом по умолчанию.

Для того чтобы сервер мог войти в этот режим, у него должна быть включена опция SUPPRESS GO AHEAD. Обсуждение этой опции осуществляется следующим образом: клиент посылает DO SUPPRESS GO AHEAD (требуя от сервера, чтобы тот включил опцию), или сервер посылает WILL SUPPRESS GO AHEAD клиенту (спрашивая о возможности включить эту опцию для самого себя). Затем сервер осуществляет WILL ECHO, спрашивая о возможности включить отражение эхом.

3. Строка за один раз.

Часто это называется "kludge line mode", потому что его реализация приходит от чтения между строк в RFC 858. Этот RFC декларирует, что должны присутствовать обе опции ECHO и SUPPRESS GO AHEAD, чтобы обеспечить ввод символа за один раз с удаленным эхом. Таким образом, если какая-либо из этих опций не включена, Telnet находится в режиме строка за один раз.

4. Линейный режим (linemode).

В данном случае этот термин означает реальную опцию linemode, определенную в RFC 1184 [Borman 1990]. Эта опция обсуждается клиентом и сервером и корректирует все недостатки в режиме строка за один раз. Новые реализации поддерживают эту опцию.

Сигнал синхронизации (Synch)

Telnet использует команду Data Mark в качестве сигнала синхронизации который передается в виде срочных данных TCP. Команда DM это метка синхронизации в потоке данных, которая сообщает принимающему о необходимости вернуться в обычный режим работы. Он может быть отправлен в любом направлении по Telnet соединению.

Когда один конец принимает уведомление о том, что другой конец вошел в режим срочности, он начинает читать из потока данных, отбрасывая все данные кроме Telnet команд. Последний байт срочных данных это DM байт. Причина, по которой используется режим срочности TCP, заключается в том, что он позволяет посылать Telnet команды по соединению, даже если поток TCP данных остановлен управлением потока данных TCP.

TELNET - клиент сервера Telnet для Windows.

Протокол прикладного уровня TELNET (от англ. **TE**rmina**L NET**work) — сетевой протокол для реализации текстового интерфейса по сети. Название **telnet** получили также клиентские программы реализации данного протокола, практически для всех существующих операционных систем. Протокол Telnet – один из старейших сетевых протоколов, разрабатывавшихся как средство связи между удаленными терминалами в тестовом режиме. Поэтому в нем не предусмотрено шифрование данных и использование современных средств проверки подлинности. Протокол уязвим для множества сетевых атак, и не может использоваться в качестве средства управления сетевыми операционными системами. В настоящее время, для удалённого доступа к системе применяется сетевой протокол SSH (Secure SHell), при создании которого упор делался именно на вопросы безопасности. Относительная безопасность сессий Telnet осуществляется только в полностью контролируемой сетевой среде или с применением защиты на сетевом уровне (различные реализации VPN - виртуальных частных сетей). Тем не менее, TELNET по-прежнему применяется для управления специализированными сетевыми устройствами (Коммутаторами, роутерами и т.п.), а также для сетевой диагностики, выполнения отладки и изучения других текст-ориентированных (telnet-like) протоколов на основе транспорта TCP. Современный стандарт протокола Telnet описан в RFC 854.

В современных ОС семейства Windows, утилита **telnet.exe** по умолчанию, не устанавливается. Для ее установки нужно перейти в **Панель управления - Программы и Компоненты – Включение или отключение компонентов Windows** и установить галочку для **Клиент Telnet**. Или в командной строке, запущенной от имени администратора, выполнить команду:

pkgmgr /iu:"TelnetClient"

Формат командной строки:

telnet [-a][-e Символ][-f Файл][-l Имя][-t Тип][Узел [Порт]]

Параметры командной строки:

-l Имя пользователя для входа в удаленную систему при условии, что поддерживается параметр TELNET ENVIRON.

-a Попытка автоматического входа в систему. Как и ключ -l, но использует текущее имя пользователя, под которым выполнен вход в систему.

-e Служебный символ переключения режима ввода в окне telnet-клиента.

-f Имя файла журнала на стороне клиента. В русскоязычной справке этот параметр неверно трактуется как Файл_входа - “Имя файла со стороны клиента для выполнения входа в систему”.

-t Тип telnet-терминала. Поддерживаются 4 типа терминалов: vt100, vt52, ansi и vtnt.

Узел Имя узла или IP-адрес удаленного компьютера, к которому выполняется подключение. **Порт** Номер порта или имя службы. Если номер не задан, то используется стандартный порт Telnet 23\TCP
При запуске без параметров, утилита переходит в режим ожидания ввода команд :
Добро пожаловать в программу-клиент Microsoft Telnet
Символ переключения режима: 'CTRL+|'
Microsoft Telnet>

При вводе символа **?** или **help** отображается справочная инвормация:
Команды могут быть сокращены. Поддерживаемыми командами являются:
c - close - закрыть текущее подключение
d - display - отобразить параметры операции
o - open имя_узла [Порт] - подключиться к сайту (по умолчанию, Порт = 23)
q - quit - выйти из telnet
set - set - установить параметры ("set ?" для вывода их списка)
sen - send - отправить строки на сервер
st - status - вывести сведения о текущем состоянии
u - unset - сбросить параметры ("unset ?" для вывода их списка)
? /h - help - вывести справку
Некоторые из команд позволяют получить подсказку по использовании, при вводе с символом вопроса:
Telnet> **set ?** - получить подсказку по использованию команды установки режимов .
Пример отображаемой информации:
bsasdel - символ BackSpace будет отправляться как символ Delete
crlf - режим возврата каретки; приводит к отправке символов CR & LF
delasbs - символ Delete будет отправляться как символ BackSpace
escape x - где x - символ переключения в режим telnet-терминала и обратно
localecho - включение локального эха.
logfile x - где x - файл журнала. В русском переводе неверно трактуется как "Файл входа текущего клиента в систему"
logging - запись текущей сессии в журнал. В русском переводе неверно трактуется как "выполнение входа в систему"
mode x - где x=console - консольный режим, используемый для работы с оконными приложениями (редактор vi) и x=stream - потоковый режим, используемый для работы в командной строке.
ntlm - включение проверки подлинности NTLM.
term x - тип эмулируемого терминала. Где x - ansi, vt100, vt52, или vtnt.

Для получения подсказки по отмене установленных параметров используется команда
Microsoft Telnet> **unset ?**
bsasdel - символ BackSpace будет отправляться как символ Delete
crlf - режим перевода строки; приводит к отправке символа CR
delasbs - символ Delete будет отправляться как символ Backspace
escape - символ переключения в режим telnet-терминала и обратно не задан
localecho - отключение локального эха
logging - отключение записи журнала. В русскоязычной весии неверно трактуется как "отключение выполнения входа в систему"
ntlm - отключение проверки подлинности NTLM.

Примеры команд в интерактивном режиме:
open 192.168.0.1 - подключиться к серверу Telnet с IP-адресом **192.168.0.1**

o zte-f660 - подключиться к Telnet-серверу с именем **zte-f660**. Используется сокращение команды **o**pen
set logfile C:\telnet.log - использовать в качестве файла журнала **C:\telnet.log**
set logging - выполнять запись текущей сессии в файл журнала.
display - отобразить параметры текущей сессии. Пример отображаемой информации:
Символ переключения режима: 'CTRL+]'
Проверка подлинности NTLM - включена
Вывод локального эха - отключен
Режим новой строки - Символ ВВОД будет отправляться как CR & LF
Текущий режим: Потоковый
РЕЖИМ ТЕРМИНАЛА
Предпочитаемый тип терминала ANSI

На практике, утилита **telnet.exe** используется как средство диагностики и отладки для подключения не только к серверу Telnet на TCP порт 23, но и на любой другой TCP-порт, тем самым, позволяя взаимодействовать с любым приложением, управляемым командной строкой. Так, например, с использованием утилиты **telnet** можно подключиться к серверам, поддерживающим текстовый (telnet-like) ввод команд и данных - SMTP, POP3, IMAP и т.п. Кроме этого, утилиту можно использовать в качестве средства грубой проверки возможности подключения на любой TCP-порт (проверки слушается ли определенный порт TCP).

telnet 192.168.1.1 8080 - подключиться к узлу 192.168.1.1 на порт 8080. В тех случаях, когда порт закрыт, утилита сообщит о невозможности подключения. Причем, для проверки доступности определенного порта даже необязательно, чтобы он слушался службой с поддержкой текстового ввода, как например, сервер VNC. Для отключения от удаленного сервера необходимо ввести символ переключения режима (по умолчанию - **CTRL+]**).

.

Задание1

С помощью утилиты Telnet выполнить запрос к веб-сайту.

Запросы и ответы в протоколе HTTP.

Протокол HTTP определяет форматы двух видов сообщений: запросов клиента и ответов сервера. Любое сообщение состоит из двух частей: заголовка и тела сообщения. Тело отделяется от заголовка пустой строкой. В запросе тело может отсутствовать, а в ответе оно обычно содержит текст возвращаемого HTML-документа. В заголовках выделяется первая строка (строка статуса для запросов и строка состояния для ответов), а все остальные строки состоят из имени поля и его значения, разделенных двоеточием. Поля обычно несут дополнительную информацию о запрошенной странице (напр. дата модификации, кодировка, язык и пр.).

Рассмотрим пример простого взаимодействия с HTTP-сервером с помощью клиента telnet.

"telnet www.rsu.ru 80" - подключение к удаленному серверу по протоколу HTTP (80 - это номер порта, на котором демон этого протокола ожидает подключения)
"GET / HTTP/1.1" - запрос главной страницы сервера по протоколу HTTP версии 1.1
"HOST: www.rsu.ru" - указываем имя сервера
"" - пустая строка, разделяющая заголовок и тело сообщения (здесь нужно просто нажимать клавишу Enter)

Ответ на этот запрос начнется со следующих строк:
HTTP/1.1 200 OK
Date: Mon, 14 Feb 2005 07:40:04 GMT

Server: Apache/1.3.27 (Unix) mod_auth_pgsql/0.9.12 mod_ssl/2.8.12 OpenSSL/0.9.6e rus/PL30.16
Transfer-Encoding: chunked
Content-Type: text/html; charset=koi8-r
Vary: accept-charset, user-agent
onnection: Close
<html>
<head> <title>Ростовский Государственный Университет</title>
<link type="text/css" rel="stylesheet" href="rsu↵↵↵css">

В первой строке ответа указывается код ответа (в данном случае "200 OK"). Далее следуют поля заголовка ответа (текущее время и дата на сервере, программное обеспечение сервера, тип содержимого - текст в формате HTML и пр.). После пустой строки начинается HTML-документ запрошенной страницы.

1. Подключитесь к веб-серверу xxx.yyy.ru по протоколу telnet ("telnet xxx.yyy.ru 80").
2. Введите текст запроса заголовка и чтения для главной страницы. Не забудьте дважды нажать клавишу Enter в конце запроса.
3. Исследуйте ответы сервера, попытайтесь определить их значение.
4. Попробуйте послать серверу запрос на несуществующую страницу и проанализируйте результат такого запроса.
5. Попробуйте послать серверу неправильный запрос (например, HELLO) и проанализируйте

Примеры:

C:\Users\Knn>telnet www.shellhacks.com 80
HEAD / HTTP/1.1
HOST: www.shellhacks.com

HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Sun, 22 Sep 2019 09:18:50 GMT
Content-Type: text/html; charset=iso-8859-1
Connection: keep-alive
x-ray: p699:0.020/wn1143:0.010/wa1143:D=8764
Location: https://www.shellhacks.com/
X-Page-Speed: on
Cache-Control: max-age=0, no-cache

Подключение к узлу утеряно.

:\Users\Knn>telnet alice.pnzgu.ru 8080
GET / HTTP/1.1
HOST: alice.pnzgu.ru

:\Users\Knn>telnet www.pguas.ru 80
HEAD / HTTP/1.1
HOST: www.pguas.ru

HTTP/1.1 200 OK
Server: nginx/0.8.54
Date: Sun, 22 Sep 2019 09:37:44 GMT
Content-Type: text/html; charset=utf-8
Connection: keep-alive
X-Powered-By: PHP/5.6.38
Set-Cookie: 8c67cb6b059d7002b748128a2e584c8c=30ekp70m7a89uvhkcdcivn94g7; path=/; HttpOnly
Expires: Wed, 17 Aug 2005 00:00:00 GMT
Last-Modified: Sun, 22 Sep 2019 09:37:44 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache

Подключение к узлу утеряно.

Задание 2

Удаленное управление компьютеров.

С помощью утилиты telnet выполнить поик и чтение текстового файла на удаленной ЭВМ. Действия промониторить с помощью программы Network Monitor 3.4.

1. Подключитесь к серверу Titan по протоколу telnet

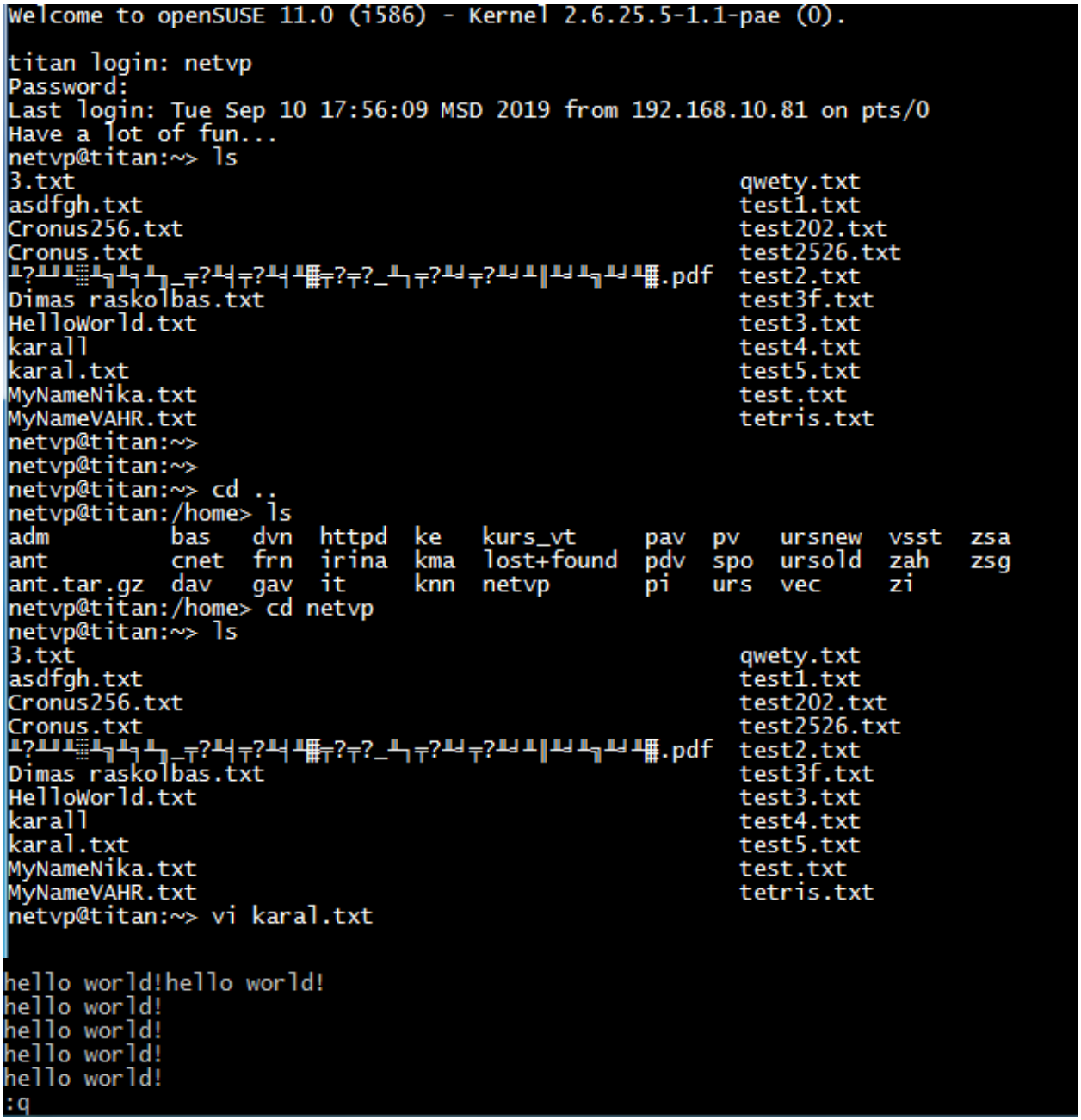
telnet
o 192.168.10.251

2. Войти на сервер Titan . Логин netvp, пароль netvp1.
3. Используя команды Linux для управления файлами (см. таблицу ниже) вывести на экран текстовый файл из директории netvp

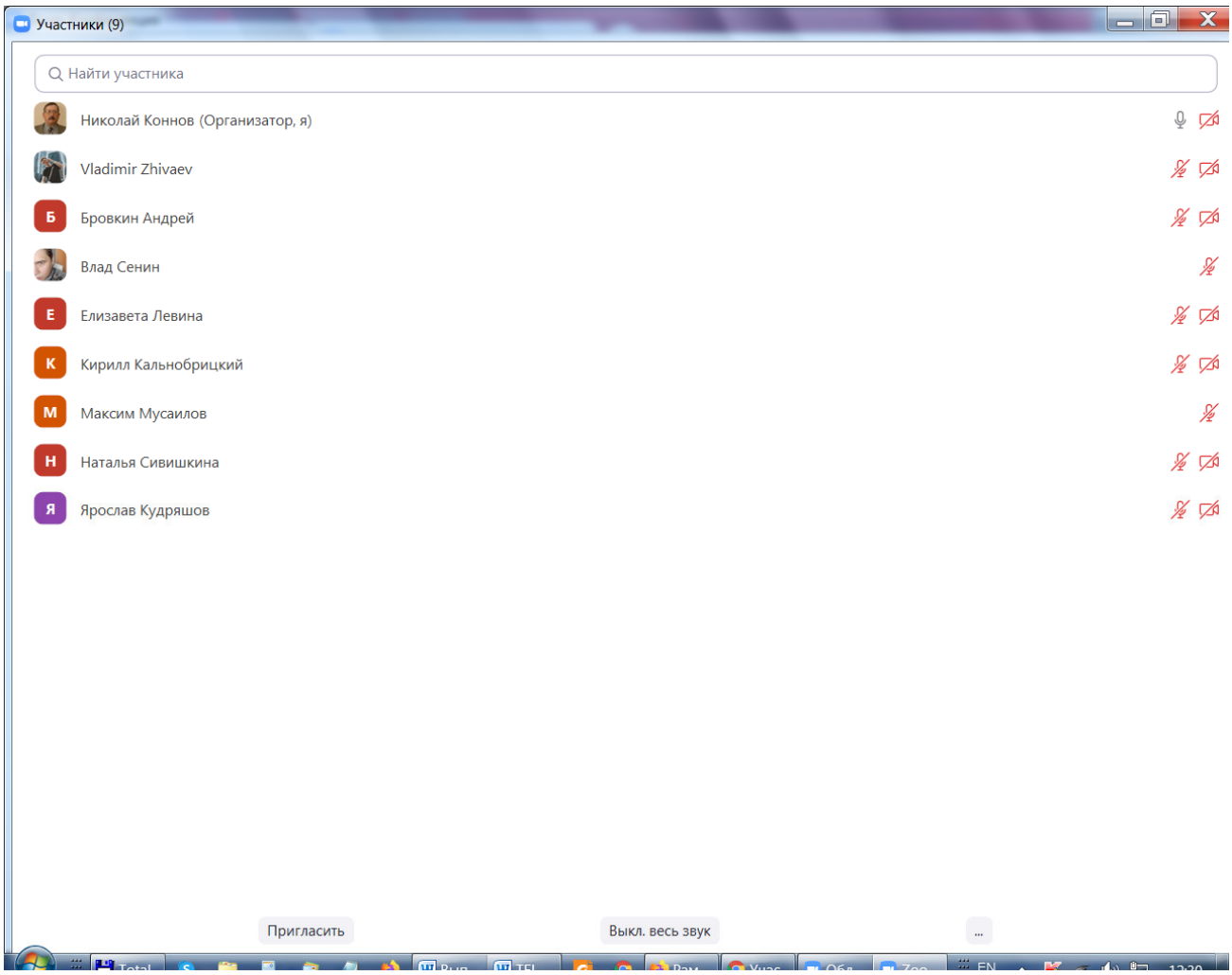
ls	Утилита для просмотра содержимого каталогов. По умолчанию показывает текущий каталог. Если в параметрах указать путь, то она перечислит содержимое конечного каталога. Полезные опции -l (**L**ist) и -a (**A**ll). Первая форматирует вывод в виде списка с более подробной информацией, а вторая включает показ скрытых файлов.
cat	Печатает содержимое файла, переданного в параметре, в стандартный вывод. Если передать несколько файлов, команда склеит их. Также можно перенаправить вывод в ещё один файл с помощью символа '>'. Если нужно вывести только определенное количество строк, используйте опцию -n (**N**umber).
cd	Позволяет перейти из текущего каталога в указанный. Если запустить без параметров - возвращает в домашний каталог. Вызов с двумя точками **(cd :)** возвращает на уровень вверх относительно текущего каталога. Вызов с тире **(cd -)** возвращает к предыдущему каталогу.
pwd	Печатает на экран текущий каталог. Это может быть полезно, если ваша командная строка Linux не выводит такую информацию. Эта команда будет востребована в Bash программировании, где для получения ссылки на каталог

	выполняется скрипт.
vi filename	Запуск редактора текста

Пример



5. Проанализируйте захваченную информацию.


Участники (9)
Найти участника
Николай Коннов (Организатор, я)
Vladimir Zhivaev
Бровкин Андрей
Влад Сенин
Елизавета Левина
Кирилл Кальнобрицкий
Максим Мусаилов
Наталья Сивишкина
Ярослав Кудряшов
Пригласить
Выкл. весь звук
...