

Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)

---

Факультет «Информатика и системы управления»  
Кафедра «Компьютерные системы и сети»

**В.Ю. Мельников**

# **Исследование способов удалённого управления Windows и Linux.**

**Электронное учебное издание**

Методические указания по выполнению лабораторных работ  
по дисциплине "Операционные системы"

2019

## Введение

Цель работы: получение теоретических и практических сведений об удалённом управлении linux и Windows серверами, а так же копирования файлов между Windows и Linux.

Время выполнения - 4 часа.

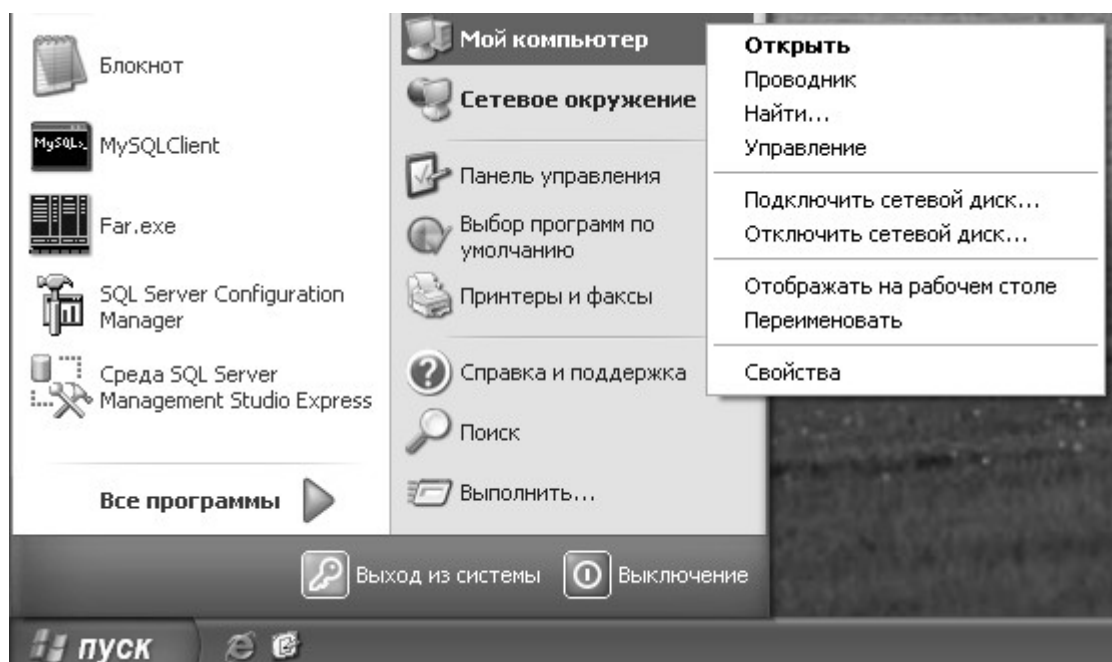
Серверы располагаются в серверной. Там созданы все условия для серверов. К сожалению, для людей эти условия мало пригодны. В серверной очень шумно, очень холодно, маленький монитор, неудобная клавиатура и работать приходится стоя. К счастью, сразу после инсталляции ОС, и настройки сетевых интерфейсов можно покинуть это негостеприимное место и продолжить настройку сервера по сети.

Так же легко, находясь в Москве установить и настроить разработанную Вами программу в Нижнем Новгороде, Владивостоке, Америке — в любом месте где доступен Интернет.

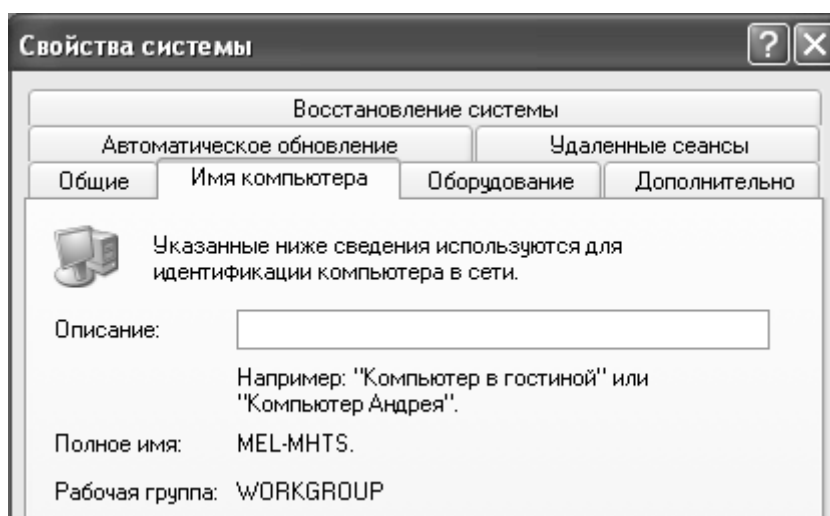
## Удалённое управление Windows

### Имя компьютера

Для начала, определим имя нашего компьютера. Нажмите кнопку «Пуск» и вызовите правой кнопкой мыши контекстное меню на элементе «Мой компьютер», выберите пункт «Свойства»



Перейдите на вкладку «Имя компьютера». Занесите в отчёт имя вашего компьютера и домен или рабочую группу. Позже мы будем обращаться к этому компьютеру по сети.



### **Удалённый рабочий стол**

Полные возможности по управлению удалённым компьютером предоставляет программа «Подключение к удаленному рабочему столу».

Заранее, надо настроить удалённый компьютер. Пуск → Панель управления → Система.

Для Windows XP Professional — на вкладке «Удалённые сеансы»

Для Windows 7 — «Дополнительные параметры системы» → Удалённый доступ.

Для Windows 10 professional- «Настройка удалённого доступа» → Удалённый доступ

Следует отметить «Разрешить удалённый доступ ...» и нажать кнопку «Применить».

В настройках брандмауэра будет открыт порт 3389 и запущена служба дистанционного управления рабочим столом.

Администраторы компьютера и домена получают доступ сразу, других пользователей надо добавить, нажав кнопку «Выбрать пользователей». Всем пользователям, которые будут подключаться к удалённому рабочему надо задать пароль.

Обязательно настройте удалённое подключение, оно нам позже понадобится.

Для Windows 10 Home Edition служба дистанционного управления рабочим столом отсутствует. Если на вашем компьютере стоит эта версия, укажите в отчёте это обстоятельство и не выполняйте задания на подключение к удалённому рабочему столу.

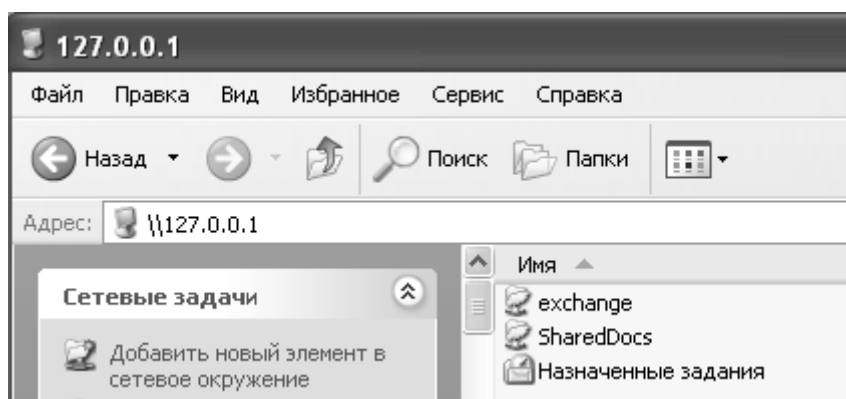
На своём локальном компьютере следует запустить Пуск → Все программы → Стандартные → Подключение к удаленному рабочему столу

В поле «Компьютер» следует ввести имя или IP адрес сервера и нажать кнопку «Подключить». Затем ввести имя пользователя и пароль.

## Общий доступ к папке

Если требуется скопировать данные на удалённый компьютер, можно открыть удалённый доступ к папке. Для этого в проводнике вызовите контекстное меню щелчком правой кнопки мыши на нужной папке, выберите команду «Свойства», выбрать вкладку «Доступ» и нажать кнопку (или флажок) «Открыть общий доступ». Можно ввести имя, под которым папка будет доступна по сети. Если требуется разрешить запись надо поставить отметку «Разрешить изменение файлов по сети». В просторечии эта операция называется «Расшарить папку». По запросу Windows разрешить «Обнаружение сети и совместный доступ к файлам». Теперь, на локальном компьютере в строке адреса проводника следует ввести «\\IP\_адрес» или «\\ИМЯ» удалённого компьютера и через несколько секунд откроется список доступных сетевых папок этого компьютера. Если вы знаете имя сетевой папки, лучше сразу указать «\\КОМПЬЮТЕР\ПАПКА». Так папка быстрее откроется.

Создайте папку «exchange» и откройте к ней доступ на запись. Откройте проводник, введите в строку адреса «\\127.0.0.1» и нажмите «Enter». В списке должна появиться папка «exchange».



Подключение к папкам на удалённом компьютере осуществляется по протоколу [SMB](#). В ранних версиях Windows использовался SMB первой версии. После появления печально знаменитого сетевого червя «[WannaCrypt](#)», который поразил 520 тысяч компьютеров, в том числе, энергетические компании Испании, больницы Великобритании, МВД России, Microsoft срочно выпустила обновления безопасности, в том числе для не поддерживаемой более Windows XP.

Если вы не устанавливали на ваш Windows обновления с мая 2017 года, обязательно установите обновление для защиты от этого вируса ([Windows Server 2003 SP2 x64](#), [Windows Server 2003 SP2 x86](#), [Windows XP SP2 x64](#), [Windows XP SP3 x86](#), [Windows XP Embedded SP3 x86](#), [Windows 8 x86](#), [Windows 8 x64](#))

Некоторые администраторы вообще запретили использование протокола SMB. Если использование сетевой папки невозможно, можно воспользоваться программой «Подключение к удаленному рабочему столу». В этом случае надо нажать кнопку

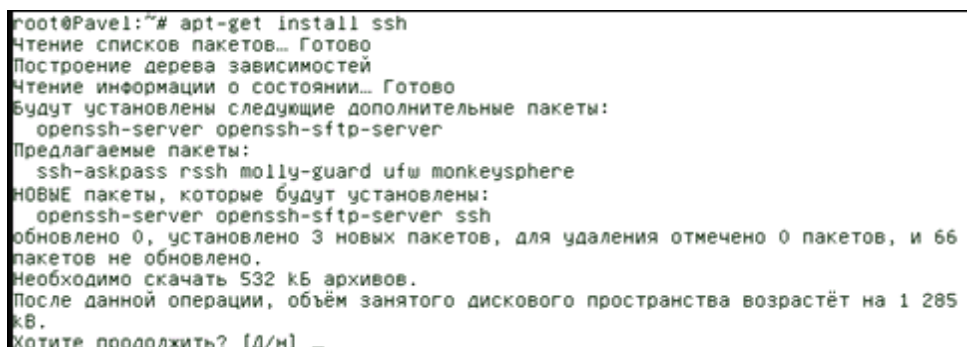
«Параметры>>>» и на вкладке «Локальные ресурсы» нажать кнопку «Подробнее», раскрыть элемент «Устройства» и отметить открываемый локальный диск вашего компьютера. Для безопасности рекомендуется использовать флешку. В этом случае, файлы надо сначала записать на локальном компьютере на флешку, а потом забрать с удалённого компьютера. Выбранный диск будет доступен на удалённом компьютере через проводник.

## Подключение к Linux из Linux

### Установка сервиса sshd

Чтобы подключиться к компьютеру с ОС Linux на удалённом компьютере надо установить сервис «sshd». Этот сервис обеспечивает доступ к компьютеру по протоколу «SSH» (Secure Shell).

Дайте команду «`apt-get install ssh`»



```
root@Pavel:~# apt-get install ssh
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  openssh-server openssh-sftp-server
Предлагаемые пакеты:
  ssh-askpass rssh molly-guard ufw monkeysphere
НОВЫЕ пакеты, которые будут установлены:
  openssh-server openssh-sftp-server ssh
обновлено 0, установлено 3 новых пакетов, для удаления отмечено 0 пакетов, и 66
пакетов не обновлено.
Необходимо скачать 532 кБ архивов.
После данной операции, объём занятого дискового пространства возрастёт на 1 285
кБ.
Хотите продолжить? [Y/n]
```

Будут установлены пакеты сервера ssh и запущен сервис «[sshd](#)»

Несмотря на то, что весь трафик шифруется, из соображений безопасности, в настройках запрещено подключаться по сети пользователем root (с аутентификацией по паролю). Но можно подключиться любым другим пользователем, и использовать команды `sudo`, или `su` для получения привилегий суперпользователя (которое в свою очередь надо разрешить в файле `/etc/sudoers`).

Во внутренней сети, если у вас сложный пароль и вы его надёжно храните, можно рискнуть разрешить прямое подключение суперпользователем по сети. Для этого, надо в файле «`etc/ssh/sshd_config`» найти параметр «`PermitRootLogin`» и задать «`PermitRootLogin = yes`». Затем надо рестартовать сервис командой «`service sshd restart`».

### Удалённое выполнение команд

Linux, как и Windows, позволяет подключаться к удалённому компьютеру в графическом режиме. Но поскольку текстового режима для администрирования сервера достаточно, а нагрузка на сеть и на сервер на порядок меньше, чем в графическом режиме, то подавляющее большинство администраторов Linux использует именно текстовый режим.

Поскольку наша виртуальная машина не имеет пока доступа к другим компьютерам, для начала установим сетевое соединение с localhost – значит со своей - же виртуальной машиной. Но работа с удалённым компьютером будет происходить так же.

Дайте команду «[ssh](#) root@localhost»

```
root@debian:~# ssh root@localhost
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is d4:37:e3:d7:31:4e:fe:fe:cc:f5:2c:14:8e:10:c4:73.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
```

При первом обращении к новому серверу, команда ssh спрашивает, доверяете ли вы этому серверу. Если позже злоумышленник подключится к сети с именем вашего сервера, ssh не даст подключиться к этому серверу.

Защита это хорошо. С другой стороны, если вы замените сервер (например, на более мощный), и дадите ему тот же адрес, ssh увидит, что отпечаток изменился и откажется с ним работать. Для удаления отпечатка надо дать команду, которая приводится в сообщении об ошибке.

Затем запрашивается пароль пользователя, указанного в нашей команде

```
root@localhost's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Nov 17 22:34:28 2016
root@debian:~# _
```

Чтобы убедиться, что мы подключились действительно по ssh дайте команду «[pstree](#)»

```
root@debian:~# pstree
systemd--acpid
          |
          |--atd
          |--cron
          |--dbus-daemon
          |--2*[dhclient]
          |--exim4
          |--login--bash--ssh
          |--rpc.idmapd
          |--rpc.statd
          |--rpcbind
          |--rsyslogd--{in:imklog}
                      |--{in:imuxsock}
                      |--{rs:main Q:Reg}
          |--sshd--sshd--bash--pstree
          |--systemd-journal
          |--systemd-logind
          |--systemd-udev
root@debian:~# _
```

Мы в строке login—bash—ssh мы видим, что выполняется команда ssh в которой мы вводим команды. Каждый символ, который вы вводите в терминале на локальном компьютере поступает в стандартный поток ввода (stdin) команды ssh. Далее этот символ передаётся по сети сервису sshd на удалённом компьютере. Сервис sshd заносит его в стандартный поток ввода. Интерпретатор команд «bash», запущенный сервисом «sshd» в

режиме конвейера, читает символ из стандартного потока ввода, запоминает введенный символ и выводит его в стандартный поток вывода. Выведенный символ поступает по конвейеру сервису «sshd». Сервис по сети передает символ на локальный компьютер команде «ssh», которая выводит символ на экран локального компьютера.

При неустойчивой работе сети, можно заметить, что введенные символы набираемой команды появляются через пару секунд после их ввода. В этом случае, вслепую вводите команды. В очередном пакете будет передано по сети сразу несколько символов.

Наконец мы набрали команду и нажали «ENTER». Код клавиши «Enter» передается как обычный символ с кодом «0xD» по той же цепочке. На удаленном компьютере, команда «bash» получив код «0xD», так же как и при выполнении команды на локальном компьютере выполняет эту команду. В нашем случае это «pstree». Команда «pstree» выводит текст в стандартный поток вывода, который по конвейеру поступает сервису «sshd». Сервис «sshd» передает текст по сети команде «ssh» на локальном компьютере, которая выводит текст на экран.

Все команды работают со стандартными потоками ввода и вывода и ничего не знают, о том, что они вызваны дистанционно.

Точно так же работала бы команда, и если бы сервер был далеко от нас. Все передаваемые данные шифруются, так что удаленное администрирование безопасно.

По ssh прекрасно будут работать и интерактивные команды, работающие в текстовом режиме. Для примера установим текстовый файловый менеджер «Midnight Commander», командой «apt-get install mc». И запустим его командой «mc»

Левая панель				Правая панель			
Файл				Настройки			
Команда				Правая панель			
< ~ .[^]>				< ~ .[^]>			
.и Имя				.и Имя			
Размер				Размер			
Время правки				Время правки			
-ВВЕРХ-				-ВВЕРХ-			
/..				/..			
/.cache				/.cache			
/.config				/.config			
/.local				/.local			
/.ssh				/.ssh			
/.w3m				/.w3m			
.Xauthority				.Xauthority			
.bash_history				.bash_history			
.bashrc				.bashrc			
.profile				.profile			
.selected_editor				.selected_editor			
.xsession-errors				.xsession-errors			
-ВВЕРХ-				-ВВЕРХ-			
6512M/7555M (86%)				6512M/7555M (86%)			
Совет: На медленных терминалах может помочь флаг -s.							
root@debian:~#							
1Помощь 2Меню 3Про~тр 4Правка 5Копия 6Пер~ос 7Нвк~ог 8Уда~ть 9МенюМС10Выход							

Цвет, рамки! И как всё это проходит через stdout?

Дело в том, что терминал обрабатывает ESC последовательности. Это последовательности символов, начинающиеся с символа ESC (33<sub>16</sub>), управляющие терминалом. Например, команда «echo -e "\033[42;33m Hello"» выведет текст «Hello» красными буквами на зелёном фоне. Есть управляющие последовательности для перемещения курсора в нужную позицию терминала, очистки экрана и так далее.

Но вернёмся к tc.

Перейдём в каталог «/etc». Для этого, используя клавиши стрелок вверх и вниз выберите каталог «..» и нажмите «Enter». Мы перешли на уровень выше. В нашем случае это корневой каталог. используя клавиши стрелок вверх и вниз выберите каталог «..» и нажмите «Enter» текущим каталогом стал «/etc» и мы видим его содержимое.

«Midnight Commander» позволяет вводить выполнять команды, так, что для перехода можно было бы воспользоваться командой «cd /etc»

Ctrl+O — Скрывает панели. В этом режиме удобно выполнять команды как в терминале. Повторное нажатие Ctrl+O — показывает панели.

С помощью «tc» можно копировать и перемещать файлы и каталоги. Для этого перейдите клавишей «Tab» на вторую панель, выберите нужный файл в нужном каталоге и нажмите «F5» или «F6» (подсказка в нижней строке).

Клавиша «Ins» позволяет выделить несколько файлов и каталогов.

Клавиша «F8» их удалит (осторожнее)

Клавиша «F4» открывает довольно мощный редактор с привычным интерфейсом.

Клавиша «F9» выведет меню в котором много других команд.

Если привыкнуть, то работать так же удобно, как и в графическом файловом менеджере. При этом, нагрузка на сеть и процессор минимальная,

Для выхода нажмите клавишу F10

Для завершения работы с удалённым компьютером дайте команду  
exit

## **Подключение с аутентификацией по ключу**

Если вам приходится много раз в день подключаться к разным серверам (и по несколько раз), удобно можно настроить ssh на подключение без ввода пароля. Может показаться, что этот подход — дыра в безопасности. Но не забывайте, что, вводимый вами пароль будет передаваться по сети (пусть и зашифрованный), а если злоумышленник подберёт пароль к вашему компьютеру, он сможет перехватить вводимый вами пароль. Подключаться желательно не root, а обычным пользователем, при необходимости можно получить полномочия суперпользователя командой sudo.

Сначала сформируем на вашем рабочем компьютере пару ключей командой:



## ssh-keygen

Путь к файлу пароля оставим по умолчанию (нажмите Enter), пароль не задаём.

```
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
3c:f1:9d:2f:4e:27:73:52:b0:55:9d:42:f0:b3:8d:b5 root@debian
The key's randomart image is:
+---[RSA 2048]---+
|                 |
|      .O.  +     |
|      ..O.      |
|      .OO.      |
|    .O.  =*     |
|    S.  +O.E    |
|      .O        |
|      * +       |
|      O B       |
|      .         |
+---+-----+

```

Передадим открытый ключ на удалённый компьютер (снова для тренировки подойдёт наша же виртуальная машина localhost)

ssh-copy-id user1@localhost

Последний раз вводим пароль пользователя user1

```
root@debian:~# ssh-copy-id user1@localhost
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
user1@localhost's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'user1@localhost'"
and check to make sure that only the key(s) you wanted were added.
```

Теперь команда «ssh user1@localhost» установит подключение от имени пользователя user без ввода пароля.

## **Копирование файлов на удалённый компьютер**

Сервис ssh позволяет не только выполнять команды, но и безопасно копировать файлы на удалённый сервер. Для этого используется команда «scp» (secure copy).

Синтаксис этой команды «scp [[user1@]host1:]file1 [[user2@]host2:]file2»

Например, для тренировки, скопируем файл «xx.txt» на наш же компьютер (но по сети)

scp /root/xx.txt user1@localhost:~/yy.txt

В данной команде копирование осуществляется от имени пользователя user1 в его домашний каталог, в чём мы можем убедиться дав команду «ls -l /home/user1»

Если надо скопировать несколько файлов, можно использовать знак групповой

операции. В качестве file2 укажите каталог. Например,

```
scp /root/*.txt user1@localhost:~/
```

Если понадобится скопировать каталог с подкаталогами, можно воспользоваться конвейером с командой tar. Разберитесь, как работает следующая команда:

```
MBP-Ekaterina:~ ekaterina$ tar -c /Users/ekaterina/Desktop/cat.jpg /Users/ekaterina/Desktop/wall.jpg | gzip -c - | ssh kate@10.37.129.6 'gunzip -c | tar x'
tar: Removing leading '/' from member names
kate@10.37.129.6's password:
tar: Removing leading '/' from member names
```

## Резервное копирование больших каталогов

При необходимости копировать много файлов лучше воспользоваться мощной командой «rsync»

```
apt-get install rsync
```

Скопируем с рабочего компьютера каталог /root на удалённый компьютер (снова для тренировки подойдёт наша же виртуальная машина localhost)

```
rsync -vrtplz ssh --progress --stats --delete /root
user1@localhost:~/backup
```

Разберите по данному пособию или по справке «man rsync» опции этой команды

```
root/.config/mc/mcdir/
root/.local/
root/.local/share/
root/.local/share/mc/
root/.local/share/mc/filepos
      122 100%   7.94kB/s   0:00:00 (xfr#13, to-chk=5/32)
root/.local/share/mc/history
      455 100%  29.62kB/s   0:00:00 (xfr#14, to-chk=4/32)
root/.local/share/mc/mcdir/
root/.ssh/
root/.ssh/id_rsa
      1,679 100% 109.31kB/s   0:00:00 (xfr#15, to-chk=2/32)
root/.ssh/id_rsa.pub
      393 100%  25.59kB/s   0:00:00 (xfr#16, to-chk=1/32)
root/.ssh/known_hosts
      222 100%  14.45kB/s   0:00:00 (xfr#17, to-chk=0/32)
root/.w3m/

Number of files: 32 (reg: 17, dir: 15)
Number of created files: 32 (reg: 17, dir: 15)
Number of deleted files: 0
Number of regular files transferred: 17
Total file size: 9,718 bytes
Total transferred file size: 9,718 bytes
Literal data: 9,718 bytes
Matched data: 0 bytes
File list size: 0
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 6,858
Total bytes received: 414

sent 6,858 bytes received 414 bytes  14,544.00 bytes/sec
total size is 9,718 speedup is 1.34
```

Если мы повторим эту команду, будет скопированы только новые и изменившиеся файлы (в отличии от команды [scp](#)).

Обратите внимание, что в каталог «backup» записалась копия каталога «root» (вместе с содержимым). Если в пути с исходному каталогу поставить завершающую «/», то сам каталог не создаётся, а его содержимое записывается непосредственно в целевой каталог.

При копировании «rsync» запрашивала пароль. Если мы захотим настроить ежедневное копирование ценного каталога, проще всего настроить аутентификацию по ключу.

Хотелось бы обратить внимание ещё на другую форму указания удалённого каталога:

```
rsync -av --delete ~/dirSrc backup@xxxxxxx.ru::backup-dir/2020
```

«::**backup-dir**» это имя модуля. Администратор сервера xxxxxx.ru установил сервис «rsyncd». В файле «/etc/rsyncd.conf», в описании модуля указывается какому каталогу соответствует модуль, с каких адресов разрешено подключаться, имена пользователей, которым разрешён доступ к этому каталогу, файл паролей. Такой способ безопаснее.

Команду «[rsync](#)» удобно использовать и для копирования в пределах своего компьютера. Копируются только новые и изменившиеся файлы

```
root@debian95:~$ rsync -vrtplze --stats --delete /root /home/user1/backup/  
sending incremental file list
```

```
sent 1,595 bytes  received 43 bytes  3,276.00 bytes/sec  
total size is 10,000  speedup is 6.11
```

## Настройка соединения виртуальной машины с реальной.

При создании виртуальной машины в первой лабораторной работе мы оставили тип сетевого подключения по умолчанию — «NAT». В этом режиме наша виртуальная машина имеет подключение к Интернет (вашего реального компьютера) но доступа к реальному компьютеру и другим виртуальным машинам не имеет. Так же, нет возможности установить соединение в вашей виртуальной машиной извне. До сих пор нам этого было достаточно.

Теперь мы хотим научиться подключаться по сети с реальной машины к виртуальной и наоборот.

Завершите работу вашей виртуальной машины (если она запущена). Выделите вашу виртуальную машину и выберите из контекстного меню команду «настроить». Выделите элемент «сеть».

Кроме NAT, имеются следующие типы подключений:

В режиме «Сетевой мост» виртуальная машина подключается напрямую к сети, к которой подключён реальный компьютер, получает свой IP адрес и становится одним из устройств этой сети.

В режиме «Виртуальный адаптер хоста» создаётся новая сеть, к которой подключены виртуальные машины и реальный компьютер. VirtualBox сам назначает этим устройствам IP адреса.

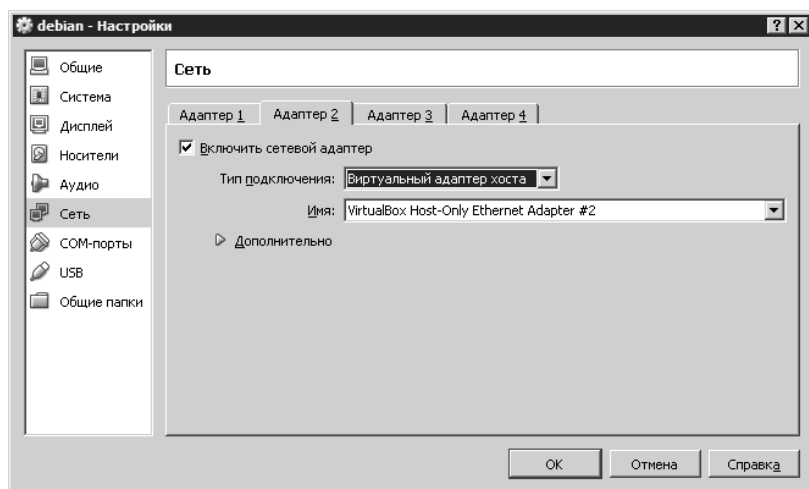
В режиме «Внутренняя сеть» создаётся новая сеть, к которой подключены только

выбранные виртуальные машины, но не связанная ни с реальным компьютером ни с реальной сетью. Этот режим хорош при написании вирусов и других опасных программ.

Тип подключения	Доступ ВМ к локальной сети и интернет	Доступ ВМ к реальному компьютеру	Доступ к ВМ из локальной сети	Доступ с реального компьютера
NAT	да	нет	Требуется «проброс портов»	Требуется «проброс портов»
Сетевой мост	да	да	да	да
Виртуальный адаптер хоста	нет	да	нет	да
Внутренняя сеть	нет	нет	нет	нет

Нам надо не лишиться доступа к интернету и при этом установить соединение с реальным компьютером. Согласно приведённой выше таблице можно попробовать режим «Сетевой мост». Но мы пойдём другим путём. Добавим второй сетевой адаптер на нашу виртуальную машину и установим режим «Виртуальный адаптер хоста». Для виртуальной машины это будет соответствовать установке второй сетевой карты. Через первый сетевой интерфейс мы будем иметь доступ в интернет, а через второй — к реальному компьютеру.

Выберите в настройках виртуальной машины вкладку «Адаптер 2», поставьте отметку «Включить» и выберите тип подключения «Виртуальный адаптер хоста».



Сохраняем настройки и запускаем виртуальную машину.

Войдите в систему как «root» и дайте команду «[ifconfig](#) -a»

```
eth0      Link encap:Ethernet  HWaddr 08:00:27:5f:c6:00
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe5f:c600/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:25 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2736 (2.6 KiB)  TX bytes:2430 (2.3 KiB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:c6:99:32
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Наш старый сетевой интерфейс имеет имя «eth0». Он получил IP адрес и работает. Наш новый сетевой интерфейс получил имя «eth1» но IP адрес не получил.

Начиная с debian 9 имена интерфейсов будут «enp0s3» и «enp0s8». Эти имена следует использовать во всех командах и настройках.

Настроим второй интерфейс. Для этого, отредактируем файл «[/etc/network/interfaces](#)»

По аналогии с настройками интерфейса «eth0» добавим строки

```
allow-hotplug enp0s8
iface enp0s8 inet dhcp
```

Согласно этой настройке интерфейс должен будет получить IP адрес автоматически. На нашей виртуальной машине, в сети типа «виртуальный адаптер хоста», адреса будет назначать VirtualBox.

Для сети в которой нет [DHCP сервера](#), (в частности, сеть типа «Внутренняя сеть» VirtualBox), придётся назначить вручную статический адрес. Например:

```
auto enp0s8
iface enp0s8 inet static
    address 192.168.56.102
    netmask 255.255.255.0
```

«Поднимем» интерфейс «eth1» командой «ifup enp0s8»

```
root@debian:~# ifup eth1
Internet Systems Consortium DHCP Client 4.3.1
Copyright 2004-2014 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth1/08:00:27:c6:99:32
Sending on   LPF/eth1/08:00:27:c6:99:32
Sending on   Socket/fallback
DHCPDISCOVER on eth1 to 255.255.255.255 port 67 interval 3
DHCPREQUEST on eth1 to 255.255.255.255 port 67
DHCPOFFER from 192.168.56.100
DHCPACK from 192.168.56.100
bound to 192.168.56.102 -- renewal in 1780 seconds.
```

Дайте команду «[ifconfig](#)»

```
eth1      Link encap:Ethernet  HWaddr 08:00:27:c6:99:32
          inet addr:192.168.56.102  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fec6:9932/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:37 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6981 (6.8 KiB)  TX bytes:1332 (1.3 KiB)
```

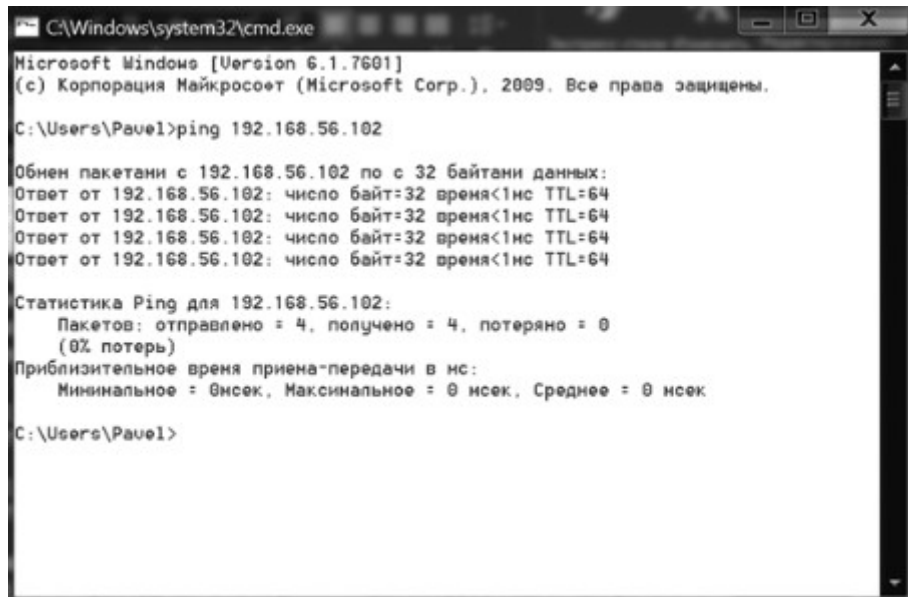
В блоке «eth1(или enp0s8)» мы видим, какой адрес получил наш сетевой интерфейс - «inet addr» = 192.168.56.102

По этому адресу мы получим доступ с реальной машины на виртуальную.

Если адрес другой, не забудьте менять его во всех командах этой лабораторной работы.

Переключитесь в вашу реальную машину, запустите терминал и дайте команду

ping 192.168.56.102

A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window shows the output of a ping command. The text is as follows:  
Microsoft Windows [Version 6.1.7601]  
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.  
C:\Users\Pavel>ping 192.168.56.102  
  
Обмен пакетами с 192.168.56.102 по 32 байтам данных:  
Ответ от 192.168.56.102: число байт=32 время<1мс TTL=64  
Ответ от 192.168.56.102: число байт=32 время<1мс TTL=64  
Ответ от 192.168.56.102: число байт=32 время<1мс TTL=64  
Ответ от 192.168.56.102: число байт=32 время<1мс TTL=64  
  
Статистика Ping для 192.168.56.102:  
Пакетов: отправлено = 4, получено = 4, потеряно = 0  
(0% потерь)  
Приблизительное время приема-передачи в мс:  
Минимальное = 0 мсек, Максимальное = 0 мсек, Среднее = 0 мсек  
C:\Users\Pavel>

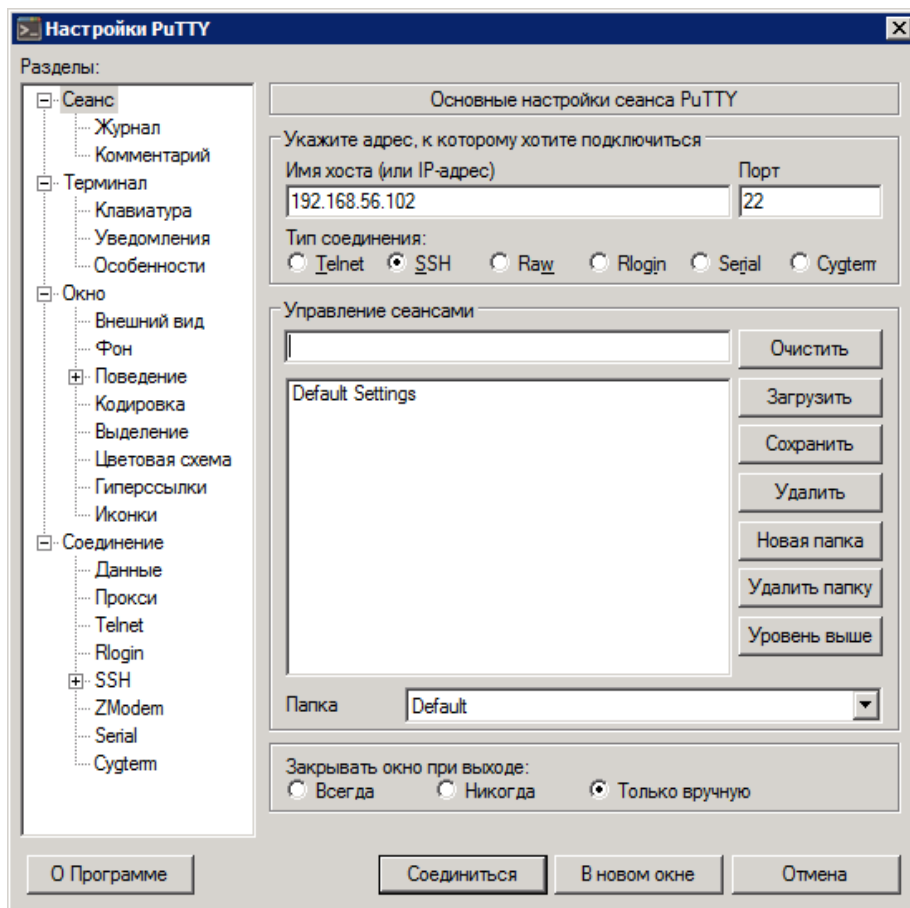
Ответ получен, значит соединение есть.

## Подключение к Linux из Windows

### Удалённое выполнение команд

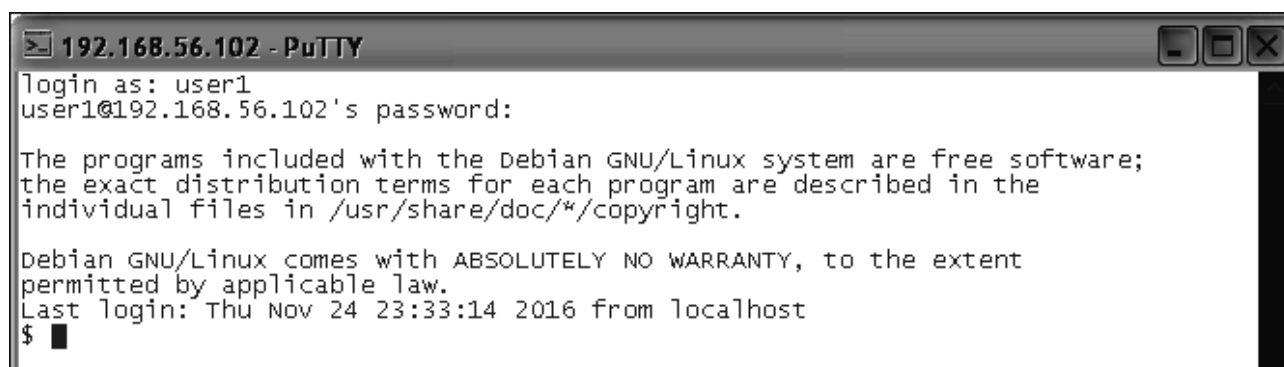
Для подключения к Linux по протоколу ssh из Windows имеется свободно распространяемая программа «putty». Скачайте её со страницы «<http://putty.org.ru/download.html>» (Жмите не большую зелёную кнопку, а незаметную ссылку «Скачать PuTTY») Загрузится архив PuTTY-0.66-RU-16.zip распакуйте его в одноимённую папку и найдите putty.exe (В принципе, можно было скачать только его и puttygen.exe)

Запустите «putty.exe» и введите IP адрес нашей виртуальной машины.



В будущем, если вы будете работать с несколькими серверами, чтобы не вводить адрес каждый раз введите имя сервера в поле «Управление сеансами» и нажмите кнопку «Сохранить».

Для подключения нажмите кнопку «Соединиться».



Точно так же, как в команде `ssh` мы выполняем команды на удалённом сервере, а результаты отображаются в окне программы PuTTY.

### **Аутентификация по ключу.**

В том же каталоге, где лежит «PuTTY.exe» лежит программа «puttygen.exe» запустите её и нажмите кнопку «Генерировать».

**Генератор ключей PuTTY**

Файл Ключ Конвертация Справка

Ключ

Открытый ключ для вставки в файл `authorized_keys` формата OpenSSH:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAqYKdjMsJ5Xr9YBLVP05Jzgf9QjsFacHo7M+dlarN8s14fpLNha4SP5n0c+7AsZ02Ej8nwhRLu/KCoThYqZBRx09z0wFQ0bum1D0mYTKd/2C80wSxp60qsBPJeFYklhGr73YysUQRwvNDe4B0u5BZHUajwvzKJoqBwjPwRU6h/w/6v4h3rzBBajZWch9xgiivDxL4fNSrl5YkKOLeLAs5xKQJYr4LjcxoAzol/ykBV1I
```

Отпечаток ключа: `ssh-rsa 2048 79:01:52:0d:60:47:6b:f9:db:44:1a:7a:f2:02:7c:ae`

Комментарий: `rsa-key-20161125`

Парольная фраза:

Повторить пароль:

Действия

Генерировать ключевую пару Генерировать

Загрузить существующий файл личного ключа Загрузить

Сохранить сгенерированные ключи Открытый ключ Личный ключ

Параметры

Тип ключа для генерации:

☐ SSH-1 (RSA) ☒ SSH-2 RSA ☐ SSH-2 DSA

Длина генерируемого ключа (в битах): `2048`

Используя кнопки «Открытый ключ» и «Личный ключ» создайте файлы открытого и закрытого ключа. А ещё скопируйте текст открытого ключа из верхнего поля в буфер обмена.

Возвращаемся в окно «PuTTY». Открываем файл «» из скрытого каталога «.ssh» домашнего каталога пользователя командой:

```
nano ~/.ssh/authorized_keys
```

```
GNU nano 2.7.4      Файл: /home/user1/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQD0zM+u05j88Qr7MIaaSF
rsa-key AAAAB3NzaC1yc2EAAAABJQAAAQEAqYKdjMsJ5Xr9YBLVP05Jzg
█
```

Вставьте в конец файла открытый ключ из буфера обмена (Shift+Ins)

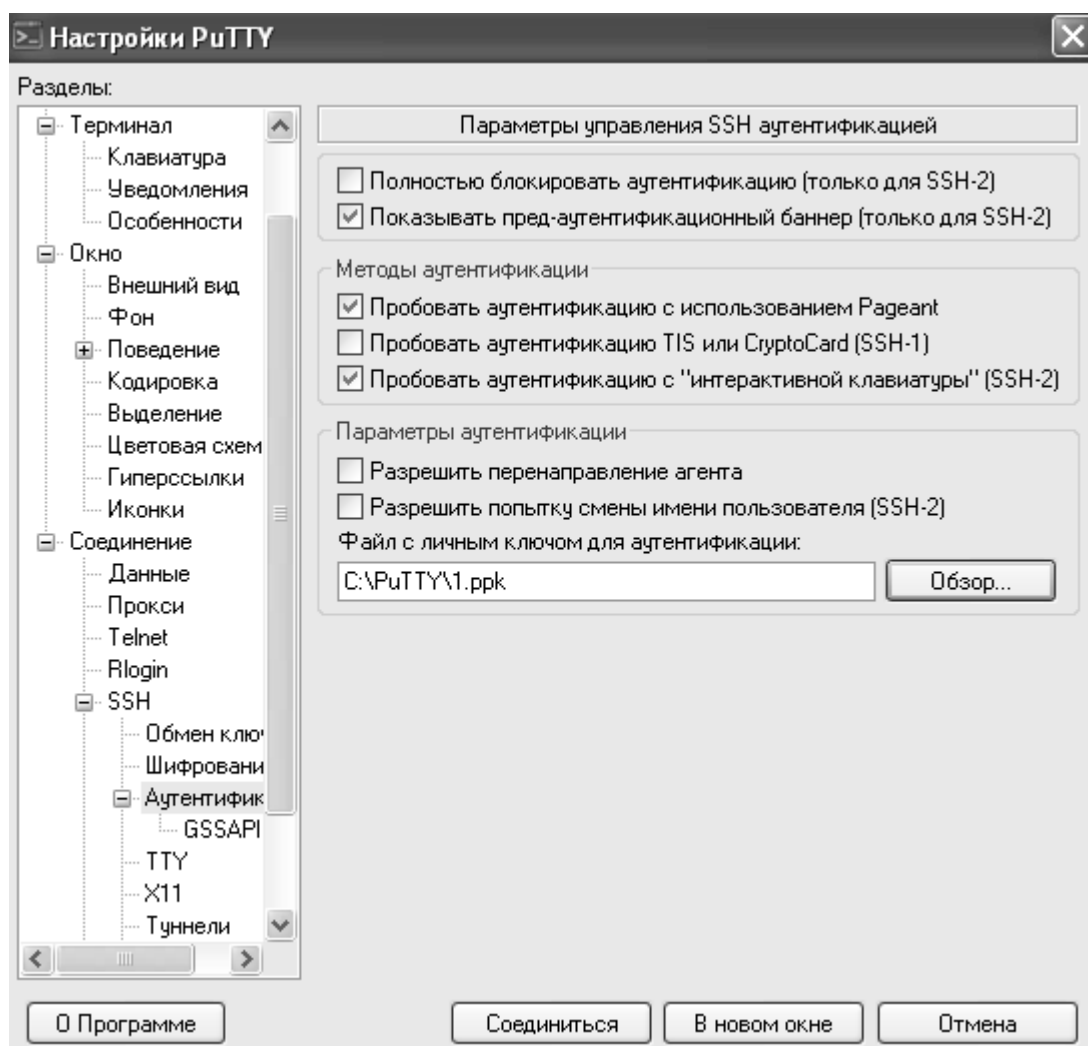
Сохраните файл (Ctrl+O)

Выйдите из редактора (Ctrl+X)

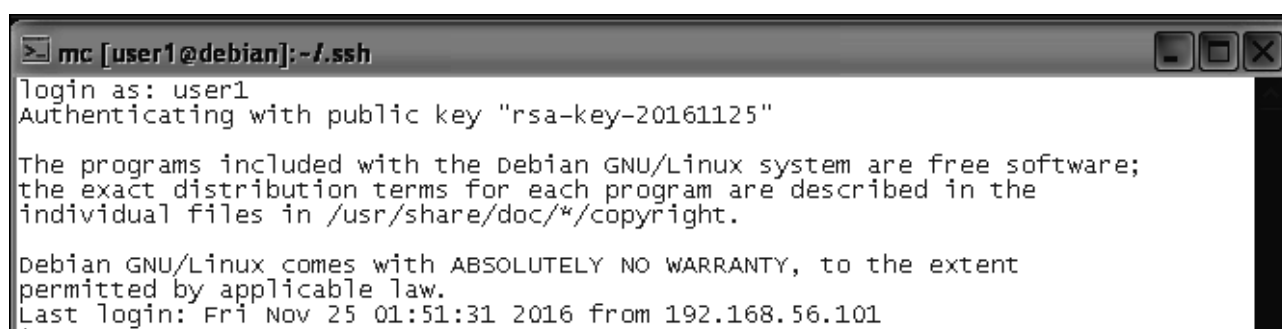
Закройте PuTTY и снова запустите его.



В левом окне раскройте элементы «Соединение», «SSH» и выберите элемент «Аутентификация» и задайте файл личного (закрытого ключа). Сохраните конфигурацию.

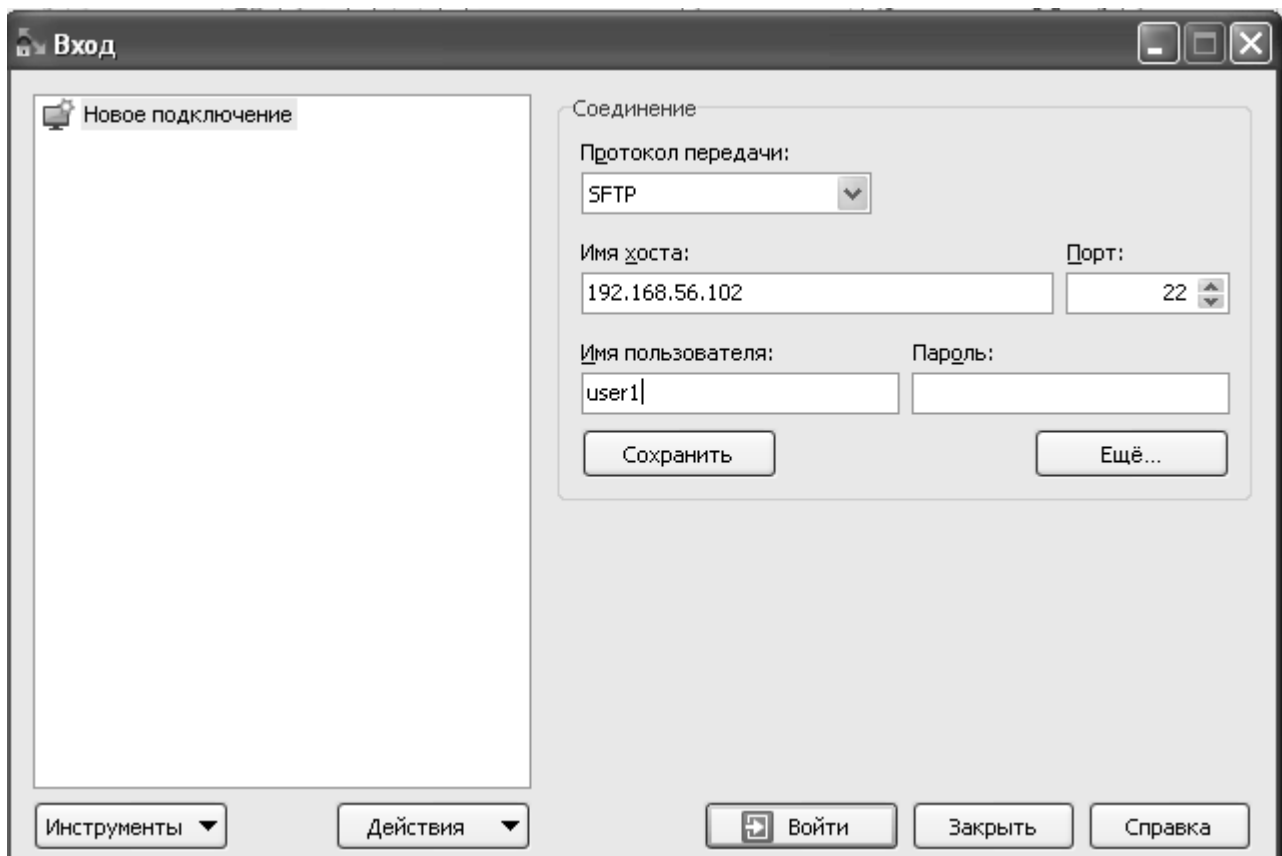


Нажмите кнопку «Соединиться», введите имя пользователя. Пароль вводить не придётся.

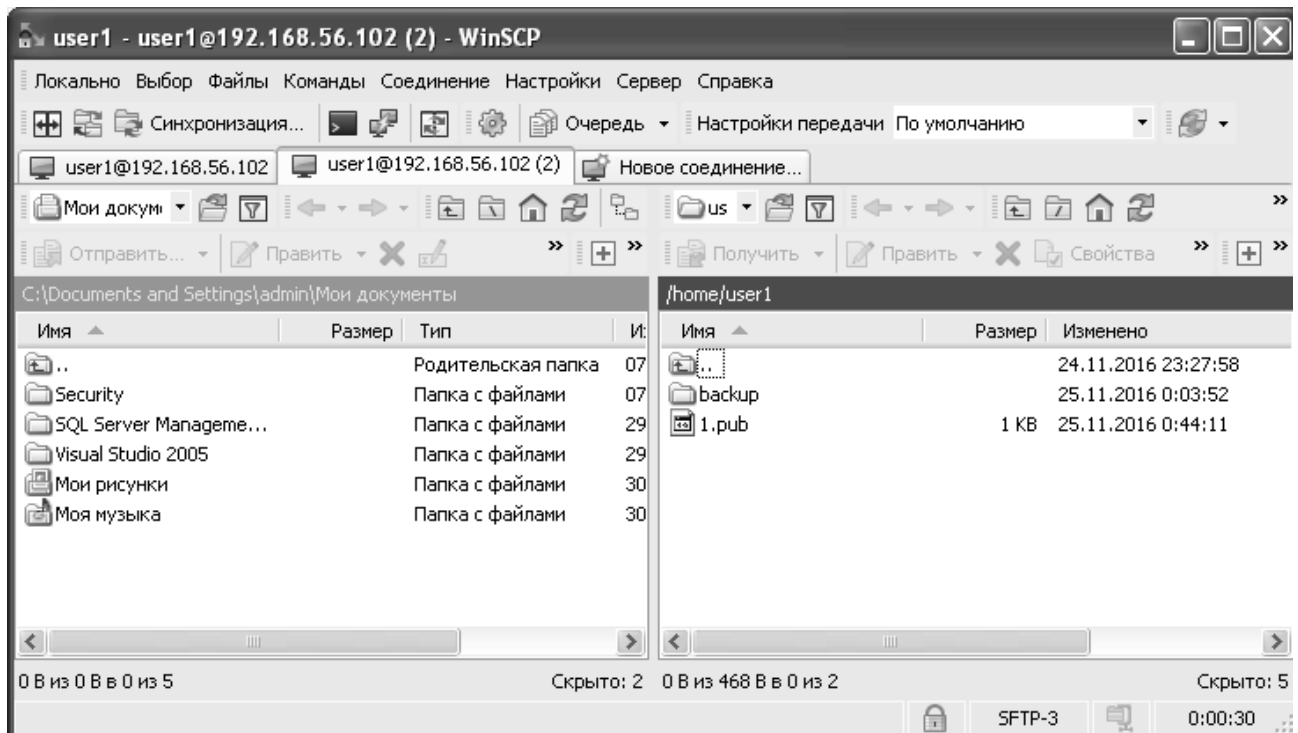


## Копирование файлов на удалённый компьютер

Загрузите с сайта <https://winscp.net/eng/download.php> программу WinSCP (ссылка Installation package) и установите её. Введите адрес удалённого компьютера, имя пользователя и сохраните конфигурацию.



Нажмите кнопку «Войти» и введите пароль. Открывается окно в котором вы можете копировать файлы с локального компьютера (левая половина) на удалённый компьютер (правая половина) и обратно.



## Копирование файлов по протоколу SMB (Server MessageBlock)

В случае, когда на большинстве компьютеров организации стоит Windows, а на Linux сервере нужны всем файлы, устанавливать на все компьютеры «WinSCP» и обучать пользователей нереально. В этом случае, надо установить на Linux сервере сервис «smbd» и настроить доступ к нужной папке.

```
apt-get install samba
```

Пусть, из общедоступного каталога «public» любой пользователь может забрать документ, и пользователи, зарегистрированные на сервере должны получить доступ в свои домашние каталоги.

### Настройка общедоступного каталога

Настроим общедоступный каталог «public», из которого любой пользователь может забрать документ.

Создаём каталог «public» и записываем в него тестовый файл:

```
mkdir -p /samba/public  
echo "test">/samba/public/test.txt
```

Сделаем копию файла конфигурации (чтобы восстановить, если что):

```
cp /etc/samba/smb.conf /etc/samba/smb.conf.default
```

Редактируем конфигурационный файл «nano [/etc/samba/smb.conf](#)»:

В разделе [global] задаются общие настройки для всех каталогов. Рассмотрим некоторые настройки, сформированные при установке, и внесём изменения (помечены жирным шрифтом):

```
# Имя этого компьютера в windows (допишите эту строку в начало секции) [global]  
netbios name = debian  
# Рабочая группа или домен, которому принадлежит сервер  
workgroup = WORKGROUP  
# файл протокола  
log file = /var/log/samba/log.%m  
# Уровень протоколирования (чем больше, тем подробней). В новых версиях заменён на «log level» (периодически выдаётся предупреждение)  
# syslog = 1  
log level = 1  
#запрещаем небезопасный протокол SMB1  
min protocol = SMB2  
#Но если вам надо обмениваться в Windows XP придётся его разрешить  
server min protocol = NT1
```

Добавим в конец файла настройку доступа к каталогу «public»

```
# В квадратных скобках указываем имя, под которым каталог будет виден в проводнике  
[public]  
# комментарий  
comment = Общедоступный каталог  
# путь к каталогу (на сервере)  
path = /samba/public
```

# разрешаем подключаться без аутентификации

**guest ok = yes**

# запрещаем запись в каталог

**read only = yes**

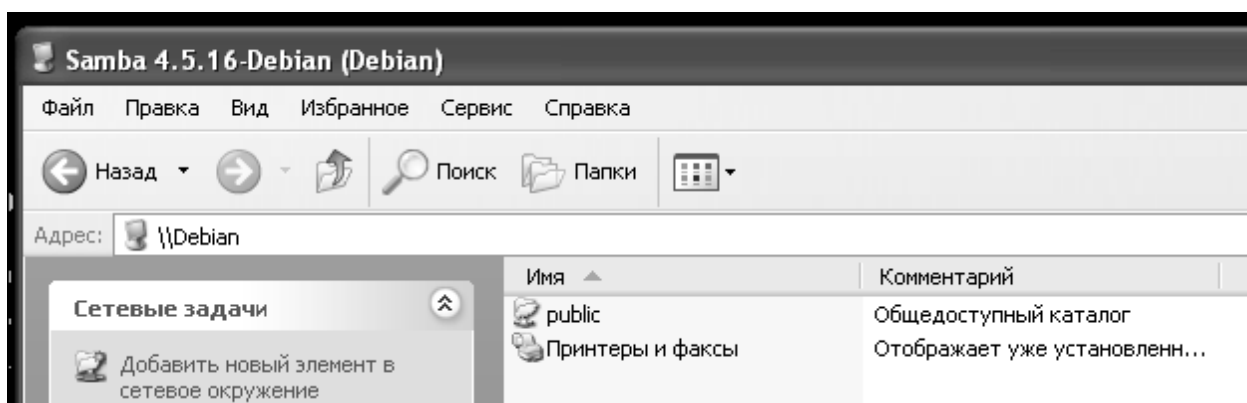
Сохраняем файл (Ctrl+O)

Проверяем правильность командой «testparm»

Перезапускаем сервисы «service nmb restart; service smb restart»

- «nmb» - служба имён (NetBios)
- «smb» - предоставляет доступ к этому компьютеру по протоколу SAMBA

В Windows открываем проводник вводим в строку адреса «\\debian» и нажимаем Enter



Отображается список сетевых ресурсов, в том числе наша папка «public»

Откройте эту папку и убедитесь, что файл читается, но изменение его запрещено.

Если Windows запрашивает пароль для доступа к нашей папке «public», значит на вашем компьютере ради безопасности запрещён анонимный доступ к общим папкам. Продолжайте выполнение лабораторной работы. После выполнения действий из следующего раздела вы сможете войти в этот каталог.

В случае, если Вы настраиваете обмен с WindowsXP необходимо учитывать,

## Настройка каталогов, ограниченного доступа

Дадим пользователю «user1» доступ к своему домашнему каталогу

Снова открываем конфигурационный файл «nano </etc/samba/smb.conf>»

Рассмотрим некоторые настройки, сформированные при установке, и внесём изменения (помечены жирным шрифтом):

# Роль сервера: Контроллер домена, член домена, отдельный сервер (standalone)

**server role = standalone server**

# Пользователи SAMBA хранятся отдельно. При изменении пароля пользователя SAMBA, будет автоматически изменён пароль пользователя linux

unix password sync = yes

# Включаем проверку пароля по протоколу «ntlm» (добавьте 2 строки)

**ntlm auth = Yes**

**client ntlmv2 auth = Yes**

```
# Если пользователь не прошел аутентификацию,  
# он имеет только гостевой доступ к общедоступным каталогам  
map to guest = bad user
```

Полный список настроек можно посмотреть командой «`man smb.conf`»

Ищем раздел `[homes]`. Этот раздел настраивает пользователям, зарегистрированным на данном сервере, доступ к своим домашним каталогам.

```
#Элемент «homes» не отображается в проводнике  
browseable = no  
# Даём пользователю право изменять каталог (измените этот параметр)  
read only = no  
# username из строки адреса «\\server\username» единственное допустимое имя пользователя  
valid users = %S  
# На новые файлы и каталоги имеет права только владелец  
create mask = 0700  
directory mask = 0700
```

Можно добавить каталог с доступом к нему группы пользователей «`exchange_group`»:

```
[exchange]  
path = /var/exchange  
valid users = @exchange_group  
force group = exchange_group  
create mask = 0770  
directory mask = 0770  
writable = yes
```

Сохраняем файл (Ctrl+O)

Проверяем правильность командой «`testparm`»

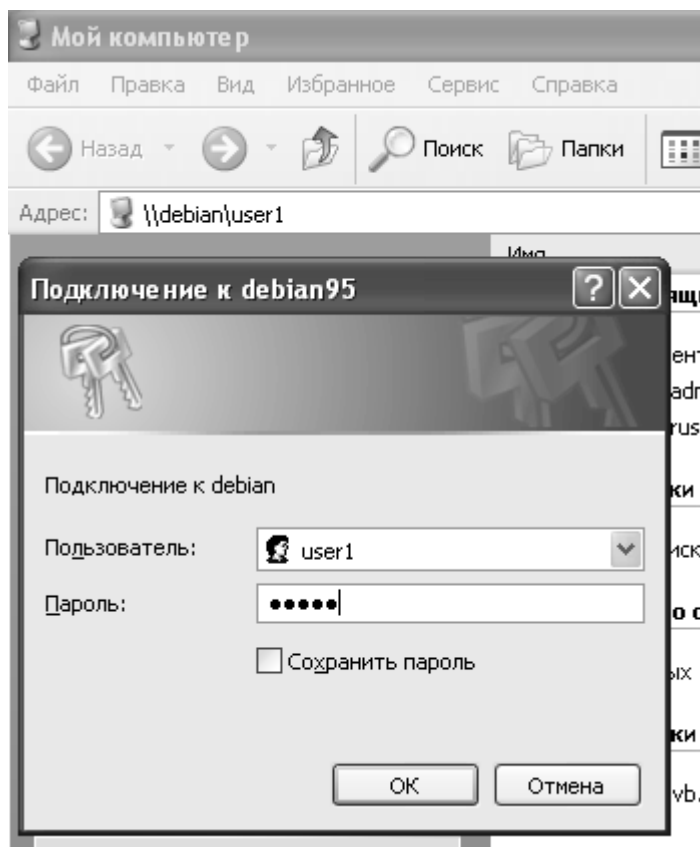
Перезапускаем сервисы «`service nmb restart; service smb restart`»

У SAMBA свои пользователи. Обязательно надо создать пользователя командой

```
smbpasswd -a user1
```

Список существующих пользователей SAMBA: «`pdbedit -L`»

Посмотрим, что получилось. В Windows открываем проводник вводим в строку адреса «\\debian\\user1». и нажимаем <Enter>, вводим имя пользователя (зарегистрированного в SAMBA) и пароль



# Подключение к Windows из Linux

## Подключение к удалённому рабочему столу Windows

Если у Вас Windows 10 Home ознакомьтесь с этим разделом теоретически.

Посмотрим, какие клиенты RDP имеются в стандартном репозитории:

```
apt-cache search "rdp client"
```

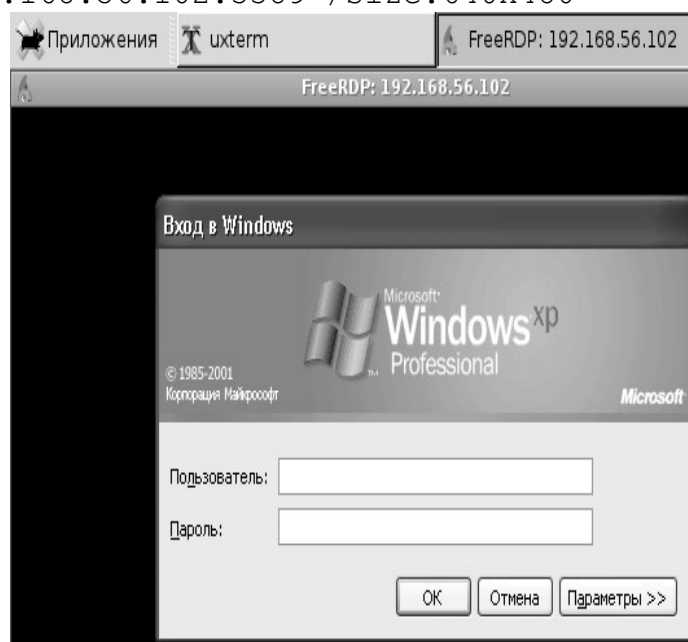
```
root@debian95:~$ apt-cache search "rdp client"
freerdp-x11 - RDP client for Windows Terminal Services (X11 client)
freerdp-x11-dbg - RDP client for Windows Terminal Services (X11 client, debug symbols)
libfreerdp-plugins-standard - RDP client for Windows Terminal Services (plugins)
libfreerdp-plugins-standard-dbg - RDP client for Windows Terminal Services (plugins debug)
libxfreerdp-client-dbg - RDP client for Windows Terminal Services (xfreerdp-client debug symbols)
```

Устанавливаем «freerdp-x11» («freerdp2-x11» для Debian 11 и выше)

```
apt-get install freerdp2-x11
```

Пусть, что на виртуальном или реальном компьютере с ОС Windows запущена служба RDP. Адрес этого компьютера «192.168.56.102». Запускаем клиент RDP:

```
xfreerdp /v:192.168.56.102:3389 /size:640x480
```



Посмотрите справку по «xfreerdp» и подключитесь к RDP в полноэкранном режиме.

## Копирование файлов в папки Windows по протоколу Samba

Бывает, надо забрать или положить файл в папку на компьютере с ОС windows. В разделе «[Удалённое управление Windows](#)» мы создали папку «exchange». Получим доступ к ней из linux.

Устанавливаем пакет

```
apt-get install smbclient
```

Посмотрим список ресурсов, доступных пользователю «admin» (подставляйте в команды свои имена компьютера и пользователя):

```
smbclient -U admin -L //MEL-MHTS
```

```
root@debian95:/var/log/samba$ smbclient -L //MEL-MHTS -U admin
WARNING: The "syslog" option is deprecated
Enter admin's password:
OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]

      Sharename      Type      Comment
      -----      -
IPC$              IPC        Удаленный IPC
SharedDocs        Disk
exchange          Disk
ADMIN$            Disk        Удаленный Admin
C$                Disk        Стандартный общий ресурс
OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]

      Server          Comment
      -----
Workgroup           Master
-----

root@debian95:/var/log/samba$ █
```

Если у Вашего пользователя не назначен пароль, то политика безопасности не даст подключиться (NT\_STATUS\_ACCOUNT\_RESTRICTION). Создаёте нового пользователя и задаёте ему пароль. Если у Windows 10 Home Edition, через графический интерфейс пользователя не создать. Можно попробовать из командной строки Windows создать пользователя с помощью команды «net user ИМЯ\_ПОЛЬЗОВАТЕЛЯ ПАРОЛЬ /add». Но в рамках лабораторной работы, особенно на реальном компьютере, можете этого не делать.

Подключимся к общедоступному каталогу «exchange»

```
smbclient -U admin //MEL-MHTS/exchange
```

```
root@debian95:/var/log/samba$ smbclient //MEL-MHTS/exchange -U admin
WARNING: The "syslog" option is deprecated
Enter admin's password:
OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
smb: \>
```

Обратите внимание, что подсказка команды изменилась на «smb:\>» теперь команды выполняет «smbclient» и команды у него свои. Попробуем некоторые из них:

Создадим свой каталог «mkdir xxx»

И посмотрим что получилось командой «dir» - вывести содержимое каталога

Для выхода из «smbclient» дадим команду «exit»



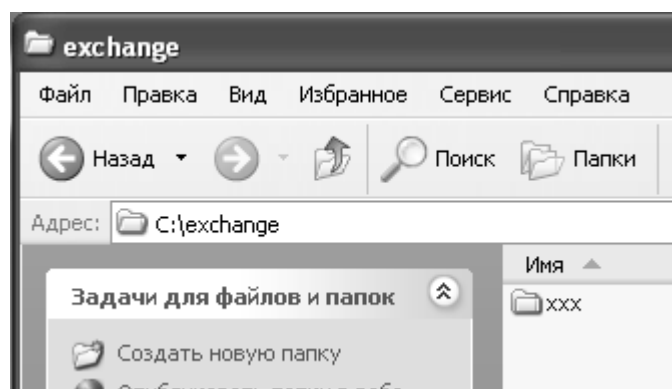
```

root@debian95:/var/log/samba$ smbclient //MEL-MHTS/exchange -U admin
WARNING: The "syslog" option is deprecated
Enter admin's password:
OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
smb: \> mkdir xxx
smb: \> dir
.                  D          0   Sat Apr 20 21:41:59 2019
..                 D          0   Sat Apr 20 21:41:59 2019
xxx                D          0   Sat Apr 20 21:41:59 2019

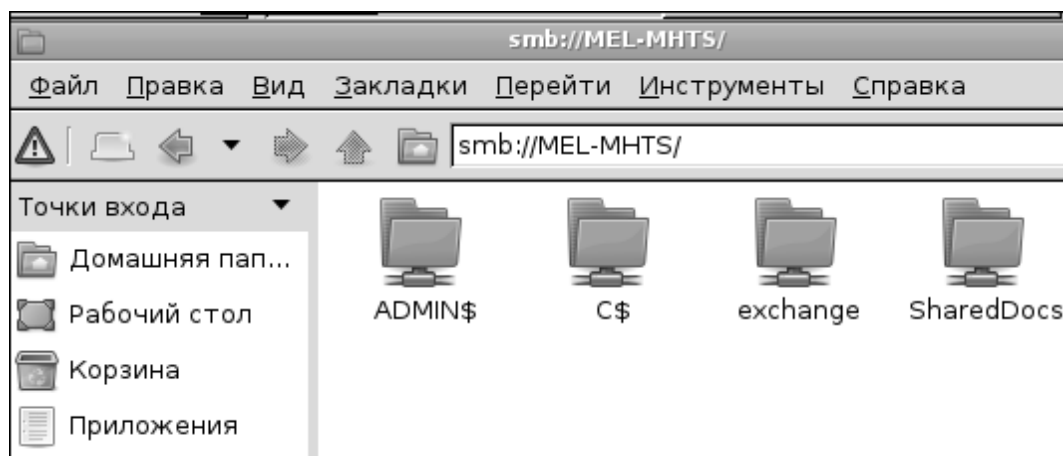
                2618587 blocks of size 4096. 324772 blocks available
smb: \> exit
root@debian95:/var/log/samba$ █

```

Посмотрим в Windows содержимое папки «exchange»:



Если установлена графическая среда, можно воспользоваться любым файловым менеджером. Перед именем компьютера надо указать «smb://»



Но пакет «smbclient» всё равно нужно установить.

## Монтирование удалённых каталогов

Примечание. Если не удалось подключиться через smbclient, приступать к этому разделу бесполезно.

Описанные выше способы удалённого взаимодействия не пригодны, для доступа к удалённым файлам программы, в которой не предусмотрено использование соответствующих протоколов. К счастью, в Linux имеется возможность смонтировать

удалённый каталог, как обычный каталог Linux

Устанавливаем пакет «cifs-utils» (в ранних версиях пакет «smbfs»)

```
apt-get install cifs-utils
```

Создаём пустой каталог в который будет отображаться удалённый каталог. Обычно, каталоги монтируют в каталог «/mnt».

```
mkdir -p /mnt/exchange
```

Наконец осуществляем монтирование. (Подставьте свои параметры)

```
mount -t cifs
```

```
-o user=admin,password=admin,workgroup=WORKGROUP,ioccharset=utf8 //192.168.56.103/exchange /mnt/exchange
```

Ознакомьтесь с разделами справки `mount` и `mount.cifs` и включите расшифровку параметров команды в отчёт.

Посмотрим, что получилось

```
ls /mnt/exchange
```

```
root@debian95:/$ mount -t cifs -o user=admin,password=admin,workgroup=WORKGROUP //192.168.56.103/exchange /mnt/exchange
root@debian95:/$ ls /mnt/exchange/
test_linux.txt  test_win.txt  xxx
```

В каталоге отображаются файлы и каталоги компьютера с ОС Windows.

Чтобы демонтировать наш сетевой каталог дайте команду:

```
umount /mnt/exchange
```

Чтобы каталог автоматически монтировался при загрузке компьютера, обычно рекомендуется добавить в конец файла «/etc/fstab» строку типа:

```
//192.168.56.103/exchange /mnt/exchange cifs
username=user,password=passwd,ioccharset=utf8 0 0
```

Если при перезагрузке нашего компьютера, удалённый компьютер был недоступен, монтирования не произойдёт. В этом случае надо повторить монтирование командой

```
mount /mnt/exchange
```

Обратите внимание, что указывается только имя каталога. Остальные параметры берутся из файла «/etc/fstab»

## Приложения

Текст этого раздела представляет собой перевод фрагментов справки из команды «man», относящихся к материалу данного учебного пособия. Данный материал даёт расширенную информацию по рассмотренным командам, а так же может использоваться для выполнения заданий, приведённых в тексте.

### **sshd**

Сервис «sshd» обеспечивает:

- защищённое шифрованное соединение
- Удалённое выполнение команд
- Проброс стандартных потоков ввода/вывода
- Копирование файлов между локальным и удалённым компьютером
- Монтирование удалённой файловой системы
- Проброс портов (прямой и обратный). Это даёт возможность безопасно использовать программы, использующие незащищённый протокол передачи данных.
- Проброс X-сервера. Обеспечивает удалённое подключение к графическому интерфейсу Linux
- Управление ключами шифрования.

По умолчанию сервис «sshd» использует TCP порт 22.

### **ssh**

```
ssh [options] [user@]host [command]
```

Команда «ssh» используется для удалённого подключения к удалённому компьютеру и выполнения команд на удалённой машине. Она обеспечивает защищённое шифрованное соединение.

Команда «ssh», взаимодействует с сервисом [«sshd»](#), запущенном на удалённом компьютере.

Параметры:

«**host**» - имя или адрес удалённого компьютера.

«**user**» - имя пользователя на удалённом компьютере

«**command**» - команда, которая будет выполнена на удалённом компьютере. Если команда не задана, запускает на удалённом компьютере интерпретатор команд в котором можно работать как на локальном компьютере.

Основные опции:

-C - включить сжатие всех передаваемых данных.

**-E log\_file** — Добавлять отладочную информацию в «log\_file» вместо stderr

**-e escape\_char** — Устанавливает на время сессии «escape\_char» в качестве управляющего ESC символа. По умолчанию «~»

**-F configfile** — Задаёт альтернативный файл конфигурации. По умолчанию используется файл конфигурации пользователя «~/.ssh/config» или общий «/etc/ssh/ssh\_config»

**-L [bind\_address:]port:host:hostport** — пересылать (пробрасывать) данные, направляемые на порт «port» локального компьютера на порт «hostport» удаленного компьютера «host»

**-R [bind\_address:]port:host:hostport** - пересылать (пробрасывать) данные, направляемые на порт «hostport» удаленного компьютера «host» на порт «port» локального компьютера

**-g** - Позволяет удаленным компьютерам подключаться к локальным проброшенным портам.

**-i identity\_file** - Задаёт файл закрытого ключа. По умолчанию используется ~/.ssh/id\_rsa

**-v -vv -vvv** — Вывод отладочной информации. Чем больше «v» тем подробнее информация. Помогает найти причины проблем при работе с этой командой.

**-y** — Записывать информационные сообщения данной команды в syslog вместо «stderr»

## **ssh-keygen**

Команда «ssh-keygen» используется для управления аутентификационными ключами.

Основные формы этой команды:

```
ssh-keygen [-q] [-b bits] [-t dsa | ecdsa | ed25519 | rsa | rsa1] [-N new_passphrase] [-C comment] [-f output_keyfile]
```

Создать новый ключ шифрования и записать его в файл «output\_keyfile».

new\_passphrase — пароль для доступа к ключу.

```
ssh-keygen -p [-P old_passphrase] [-N new_passphrase] [-f keyfile]
```

Сменить пароль на доступ к ключу

```
ssh-keygen -i [-m key_format] [-f input_keyfile]
```

Считать не зашифрованный файл закрытого (или открытого) ключа в формате, заданном параметром -m, и вывести совместимый с OpenSSH закрытый (или открытый) ключ в стандартный поток вывода. Эта опция позволяет импортировать ключи из другого программного обеспечения, Формат импорта по умолчанию - «RFC4716».

```
ssh-keygen -e [-m key_format] [-f input_keyfile]
```

Считать закрытый или открытый ключ OpenSSH и вывести ключ в формате, заданном параметром -m. По умолчанию используется формат "RFC4716". Эта опция позволяет экспортировать ключи OpenSSH для использования другими программами.

## **ssh-copy-id**

```
ssh-copy-id [-f] [-n] [-i [identity_file]] [-p port] [-o  
ssh_option] [user@]hostname
```

Утилита для копирования открытого (публичного) ключа «identity\_file» на удалённый компьютер. По умолчанию копируется «~/ssh/id\*.pub». Открытый ключ записывается в файл «~/ssh/authorized\_keys» в домашний каталог пользователя «user» удалённого компьютера «hostname»

## **scp**

```
scp [-12346BCpqrTv] [-c cipher] [-F ssh_config] [-i identity_file]  
[-l limit] [-o ssh_option] [-P port] [-S program]  
[[user@]host1:]file1 ... [[user@]host2:]file2
```

Копирует файлы между компьютерами в сети.

### **Параметры:**

«host1», «host2» - имя или адрес удалённого компьютера. Если не указан, подразумевается локальный компьютер.

«user1», «user2» - имя пользователя. Если не указан, подразумевается текущий пользователь.

«file1», «file2» - абсолютный или относительный путь к файлу. Можно использовать подстановочные знаки (\*,?,[])

-i identity\_file - Задаёт файл закрытого ключа. По умолчанию используется ~/ssh/id\_rsa

## **rsync**

```
rsync [OPTION...] [USER@]HOST:SRC... [DEST]  
rsync [OPTION...] [USER@]HOST::SRC... [DEST]  
rsync [OPTION...] SRC... [USER@]HOST:DEST  
rsync [OPTION...] SRC... [USER@]HOST::DEST
```

rsync - это быстрый и необычайно универсальный инструмент для копирования файлов. Он может копировать локально, или с удалённого компьютера на локальный компьютер или с локального компьютера на удалённый компьютер. Копирование осуществляется через удалённый сервис «sshd», или удалённый сервис «rsync», или любую другую удалённую оболочку. Команда имеет большое количество опций, которые позволяют

очень гибко задавать набор файлов для копирования. Используется дельта алгоритм передачи, который уменьшает объем передаваемых по сети данных. Отправляются только различия между исходными файлами и файлами, существующими в пункте назначения.

Rsync широко используется для резервного копирования и зеркалирования, а также как улучшенная команда копирования для повседневного использования.

Существует два разных способа для связи rsync с удаленной системой: Если между именем компьютера и путём указан один символ «:» - используется удалённая оболочка (например, ssh или rsh). Если указано «::» - используется обращение к удалённому сервису «rsync» напрямую через TCP. По умолчанию используется TCP порт 873

#### **Опции:**

-v, --verbose — выводится сокращённая информацию о процессе копирования

-vv, выводится подробная информацию о процессе копирования

-vvv, выводится отладочная информация

-a, --archive Режим архива. Эквивалентно набору опций -rlptgoD Это быстрый способ сказать, что вы хотите рекурсивно скопировать почти всё.

-q, --quiet подавить сообщения, не связанные с ошибками

-c, --checksum - определять изменился ли файл на основе контрольной суммы. Без этой опции анализируется время и размер.

-r, --recursive — Каталоги копировать рекурсивно

-R, - Отключает (!) использование относительных путей к файлам. Если задана эта опция, и в параметре SRC команды задан полный путь, передаются полные пути к файлам, и в целевом каталоге строится соответствующее дерево каталогов. Например, по команде «rsync -rR /home/user/work/src/ dest/» в каталоге «dest» будут созданы каталоги «home», «user» и т.д.

-b, --backup делать резервные копии файлов, имеющихся в целевом каталоге перед их удалением или изменением. Используется совместно с «--suffix» и «--backup-dir». --suffix=SUFFIX — задаёт префикс для резервных копий файлов. --backup-dir — задаёт каталог для резервных копий файлов.

-u, --update пропустить файлы, которые новее в целевом каталоге

--append добавить данные в более короткие файлы

--append-verify как --append, но со старыми данными в контрольной сумме файла

-d, --dirs переносят каталоги без рекурсии (без подкаталогов)

-l, --links копирует символические ссылки как символические ссылки

-L, --copy-links преобразует символическую ссылку в референтный файл или каталог

--copy-unsafe-links преобразуются только символические ссылки «unsafe»

--safe-links игнорируют символические ссылки, которые указывают за пределы исходного дерева

--munge-links munge обратимо преобразует символьные ссылки, чтобы сделать их более безопасными (но непригодными для использования). Это полезно, если вы не доверяете источнику данных.

-k, --copy-dirlinks преобразует символическую ссылку в каталог в целевом каталоге.

-K, --keep-dirlinks обрабатывают символьный каталог на приемнике как каталог

-H, --hard-links сохраняют жесткие ссылки

-p, --perms сохранять права доступа.

-E, --executability сохранить признак исполняемого файла.

--chmod = CHMOD задавать права доступа к файлу и / или каталогу

-A, --acls сохранять ACL права доступа (подразумевает наличие --perms)

-X, --xattrs сохранять расширенные атрибуты

-o, --owner сохранять владельца файлов и каталогов (только для суперпользователя)

-g, --group сохранять группу

--devices сохранить файлы устройств (только для суперпользователя)

--copy-devices копировать содержимое файлов устройств как обычных файлы

-D то же, что --devices —specials

-t, --times сохранить время модификации

-O, --omit-dir-times пропускать каталоги из —times

-J, --omit-link-times пропустить символические ссылки из —times

--super говорит приемнику что будут выполняться действия, характерные для суперпользователя (смена владельца и группы)

--fake-super хранить / восстанавливать привилегированные атрибуты, используя xattrs

-S, --sparse эффективно обрабатывать разреженные файлы

--preallocate выделить файлы dest перед записью их

-n, --dry-run выполнить пробный запуск без изменений

-W, --whole-файл копировать файлы целиком (без алгоритма delta-xfer)

-x, --one-file-system не пересекать границы файловой системы (при обработке ссылок)

-B, --block-size=SIZE принудительно установить фиксированный размер блока контрольной суммы

-e, --rsh=COMMAND указать удаленную оболочку для использования и задать параметры её запуска.

--rsync-path=PROGRAM указать выполняемый файл rsync для запуска на удаленной машине

-existing - запретить создание новых файлов на приемнике

--ignore-существующие пропустить обновления файлов, которые уже существуют на приемнике

--remove-source-files sender удаляет синхронизированные файлы (не-dirs)

--del - псевдоним для --delete-во время

--delete - удалить из целевого каталога те файлы, которых нет в исходном каталоге (лишние файлы)

--delete-before удалять лишние файлы перед передачей, а не во время

--delete-during — Удалить лишние файлы в процессе передачи

--delete-delay — Найти в процессе передачи лишние файлы, и после удалить их.

--delete-after — Удалить лишние файлы после передачи

--delete-exclude - также удаляет файлы, заданные опцией «--exclude»

--ignore-missing-args игнорировать отсутствующие исходные аргументы без ошибок

--delete-missing-args удалить отсутствующие исходные аргументы из пункта назначения

--ignore-errors - удалять, даже если есть ошибки ввода-вывода

--force - принудительно удалить каталоги, даже если они не пусты

--max-delete = NUM не удалять больше чем NUM файлов

--max-size=SIZE не передавать файлы размером больше SIZE

--min-size=SIZE не передавать файлы размером меньше SIZE

--partial сохранить частично переданные файлы

--partial-dir=DIR поместить частично переданный файл в DIR

--delay-updates поместит все обновленные файлы на место в конце передачи

-m, --prune-empty-dirs убрать пустые цепочки каталогов из списка файлов

--numeric-ids не отображает значения uid / gid по имени пользователя / группы

--usermap=STRING настраиваемое отображение имени пользователя

--groupmap=STRING пользовательское сопоставление имени группы

--chown=USER:GROUP сопоставление простого имени пользователя / группы

--timeout=SECONDS установить время ожидания ввода / вывода в секундах

--contimeout=SECONDS установить время ожидания подключения к удалённому сервису (в секундах)

-I, --ignore-times не пропускать файлы, которые соответствуют размеру и времени

--size-only пропустить файлы, которые соответствуют по размеру

--modify-window=NUM сравнить время модификации с пониженной точностью

-T, --temp-dir=DIR создавать временные файлы в каталоге DIR



-y, --fuzzy найти похожий файл для базы, если нет файла dest  
--compare-dest = DIR также сравнивает полученные файлы относительно DIR  
--copy-dest=DIR --/-- и включать копии неизмененных файлов  
--link-dest=DIR жесткая ссылка на файлы в DIR, если они не изменены  
-z, --compress сжать данные файла во время передачи  
--compress-level=NUM явно установить уровень сжатия  
--skip-compress=LIST пропустить сжатие файлов с суффиксами, перечисленными в

LIST

-C, --cvs-exclude автоматически игнорировать файлы так же, как CVS  
-f, --filter=RULE добавить правило фильтрации файлов  
-F то же, что --filter='dir-merge/.rsync-filter' и --filter='-.rsync-filter'  
--exclude=PATTERN исключить файлы, соответствующие PATTERN  
--exclude-from=FILE исключить файлы, шаблоны которых перечислены в файле FILE  
--include=PATTERN не исключать файлы, соответствующие PATTERN  
--include-from=FILE не исключать файлы, шаблоны которых перечислены в FILE  
--files-from=FILE читать список имен исходных файлов из FILE

## ***smb.conf***

Файл smb.conf является файлом конфигурации для пакета Samba. Полное описание, соответствующее вашей версии Linux можно получить командой «man smb.conf». В данном разделе приведено только общее описание структуры данного файла. Основные параметры так же приведены в разделе «[Копирование файлов по протоколу SMB](#)» данного руководства.

Файл состоит из разделов и параметров.

Раздел начинается с названия раздела в квадратных скобках и продолжается до начала следующего раздела.

Разделы содержат параметры в формате: «имя = значение»

Имена разделов и параметров не чувствительны к регистру.

Только первый знак равенства в параметре является значимым. Пробел до или после первого знака равенства отбрасываются. Начальные, конечные и внутренние пробелы в именах разделов и параметров не имеют значения. Ведущий и конечные пробелы в значении параметра отбрасываются. Внутренний пробел в значении параметра сохраняется.

Любая строка, начинающаяся с точки с запятой «;» или символа хеша «#», игнорируется, так же как и строки, содержащие только пробельные символы.

Любая строка, оканчивающаяся на «\», продолжается на следующей строке обычным способом UNIX.

Значения, следующие за знаком равенства в параметрах, представляют собой либо строку (без кавычек), либо логическое значение, которое может быть дано как yes / no, 1/0 или true / false. Регистр не имеет значения в логических значениях, важен в строковых значениях. Некоторые элементы, являются числовыми.

Каждый раздел в файле конфигурации (кроме раздела [global]) описывает общий ресурс. Имя раздела - это имя общего ресурса, а параметры в разделе определяют атрибуты ресурса.

Имеется три специальных раздела, [global], [homes] и [printers]. Прочие разделы называются общими.

Общий ресурс состоит из каталога, к которому предоставляется доступ, и описания прав доступа пользователю сервиса.

Разделы могут быть назначены «гостевыми», в этом случае для доступа к ним не требуется пароль.

Права доступа, предоставляемые сервисом, маскируются правами доступа, предоставленными указанному или гостевому пользователю Linux. Сервис не предоставляет больше доступа, чем позволено пользователю.

#### **Специальные разделы:**

Параметры в разделе [global] применяются к серверу в целом или являются значениями по умолчанию для разделов, которые не определяют конкретные элементы.

Если в файл конфигурации включен раздел под названием [homes], сервис может на лету создавать ресурсы, подключающие клиентов к их домашним каталогам. Параметры такого ресурса берутся из раздела [homes]. Имя ресурса соответствует имени пользователя. Задаётся путь к домашнему каталогу пользователя.

#### **Некоторые параметры:**

**netbios name** - устанавливает имя NetBIOS, под которым известен сервер Samba. По умолчанию он совпадает с первым компонентом DNS-имени хоста. Максимальная длина имени NetBIOS составляет 15 символов.

**workgroup** - определяет, в какой рабочей группе будет отображаться ваш сервер при запросе клиентами. Обратите внимание, что этот параметр также управляет именем домена, используемым с параметром security = domain. По умолчанию: WORKGROUP

**log file** - Эта опция позволяет вам переопределить имя файла журнала Samba. По умолчанию — не задано

**log level УРОВЕНЬ КЛАСС:УРОВЕНЬ КЛАСС:УРОВЕНЬ ...** Здесь УРОВЕНЬ - уровень отладки (по умолчанию 0 - не протоколировать). КЛАССЫ: all, tdb, printdrivers, lanman, smb, rpc\_parse, rpc\_srv, rpc\_cli, passdb, sam, auth, winbind, vfs, idmap, quota, acls,

locking, msdfs, dmap, registry

**path** - каталог, к которому пользователю службы должен быть предоставлен доступ. В случае сервисов для печати, это где данные печати будут буферизованы перед отправкой на хост для печати.

**server role** - основной режим работы сервера. **auto** (значение по умолчанию) - простой файловый сервер, который не подключен ни к какому домену. **STANDALONE** - В автономном режиме клиент должен сначала войти в систему с действительными именем пользователя и паролем (которые могут быть сопоставлены с помощью параметра «**username map**»), **MEMBER SERVER** - В этом режиме Samba попытается проверить имя пользователя / пароль, передав его в контроллер домена Windows или Samba, точно так же, как это сделает Windows Server. Обратите внимание, что должен существовать действительный пользователь UNIX, и учетная запись на контроллере домена. **CLASSIC PRIMARY DOMAIN CONTROLLER** — запускает классический первичный контроллер домена. **CLASSIC BACKUP DOMAIN CONTROLLER** — запускает классический резервный контроллер домена. **ACTIVE DIRECTORY DOMAIN CONTROLLER** - запускает Samba в качестве контроллера домена активного каталога, предоставляя услуги входа в систему для клиентов Windows и Samba домена. Эта роль требует особой настройки

**unix password sync** - Этот логический параметр определяет, пытается ли Samba синхронизировать пароль UNIX с паролем SMB при изменении зашифрованного пароля SMB в файле smbpasswd. По умолчанию - «no»

**ntlm auth** - Этот параметр определяет, будет ли smbd пытаться аутентифицировать пользователей, используя ответ с зашифрованным паролем NTLM. Если этот параметр отключен, клиент должен будет отправить хэш пароля lanman или ответ NTLMv2. По умолчанию «no»

**client ntlmv2 auth** - Если включено, будут отправлены только ответы NTLMv2 и LMv2 (оба гораздо более безопасные, чем в более ранних версиях). По умолчанию «yes»

**guest ok** - Если этот параметр имеет значение «yes», то для подключения к службе пароль не требуется. Будет использоваться полномочия учетной записи гостя.

**map to guest** — «Never» - запросы пользователя на вход с неверным паролем отклоняются. Используется по умолчанию. «Bad User» - вход пользователя с неверным паролем отклоняется, если только имя пользователя не существует, и в этом случае оно рассматривается как гостевой логин и отображается в гостевой учетной записи. «Bad Password» - означает, что логины пользователей с неверным паролем рассматриваются как гостевые и отображаются в гостевой учетной записи. «Bad Uid» - логины пользователей

успешно прошли аутентификацию, но не имеют действительной учетной записи пользователя Unix должен быть сопоставлен с определенной учетной записью гостя.

**valid users** - список пользователей, которым разрешен вход в эту службу. Имена, начинающиеся с @ означают группы пользователей.

**read only** - Если этот параметр имеет значение «yes», то пользователи службы не могут создавать или изменять файлы в каталоге службы. Обратите внимание, что служба печати ВСЕГДА разрешает запись в каталог. По умолчанию read only = yes

**writable** — параметр, обратный read only

**write list** - список пользователей, которым предоставлен доступ для чтения и записи. Имена, начинающиеся с @ означают группы пользователей.

**browseable** - Этот параметр определяет, будет ли этот общий ресурс отображаться в списке доступных общих ресурсов в сетевом представлении и в списке просмотра. По умолчанию: browseable = yes

**create mask** — При создании файла, разрешения Windows преобразуются в разрешения UNIX, а затем, выполняется побитовое AND с разрешениями пользователя UNIX. По умолчанию = 0744

**directory mask** - При создании каталога, разрешения Windows преобразуются в разрешения UNIX, а затем, выполняется побитовое AND с разрешениями пользователя UNIX. По умолчанию = 0755

**force group** - имя группы UNIX, которое будет назначено в качестве основной группы по умолчанию для всех пользователей, подключающихся к этой службе. Все новые файлы и каталоги будут принадлежать этой группе.

**printing** - Этот параметр определяет, как информация о состоянии принтера интерпретируется в вашей системе. В настоящее время поддерживается: BSD, AIX, LPRNG, PLP, SYSV, HPUX, QNX, SOFTQ, CUPS и IPRINT.

**cups options** - Вам следует установить этот параметр равным raw, если файл error\_log вашего сервера CUPS содержит сообщения, «Неподдерживаемый формат» application / octet-stream »» при печати из клиента Windows через Samba.

## ***xfreerdp***

**xfreerdp [file] [options] [/v:server[:port]]**

Подключается к удалённому рабочему столу сервера «server» по протоколу X11 (RDP)

По умолчанию использует порт 3389

Опции:

/v <server>[:port] — имя или ip адрес сервера, а так же порт

/w <width> — задаёт ширину рабочего стола в пикселях

/h <height> — задаёт высоту рабочего стола в пикселях

/size <width>x<height> - задаёт размеры рабочего стола в пикселях

/f полноэкранный режим (так же можно использовать для переключения в полноэкранный режим и обратно Alt+Ctrl+Enter)

/bpp <depth> - глубина цвета

/kbd 0x<layout id> or <layout name> - Параметры клавиатуры

/kbd-type <type id> - тип клавиатуры

/kbd-subtype <subtype id> - подтип клавиатуры

/kbd-fn-key <function key count> - число функциональных клавиш

/admin — консольная сессия (для администратора)

/multimon — Использовать несколько мониторов

/monitors <0,1,2...> - выбрать монитор для использования

/monitor-list - Список обнаруженных мониторов

/t <title> - Заголовок окна

/decorations — оформление окна

/smart-sizing - Масштабирование удаленного рабочего стола до размера окна

/u [<domain>\]<user> или <user>[@<domain>] - имя пользователя и домена

/p <password> - пароль

/d <domain> - домен

/app ||<alias> or <executable path> - запустить заданную прикладную программу

/app-name <app name>- Имя удаленного приложения для пользовательского интерфейса

/app-icon <icon path> - Значок удаленного приложения для пользовательского интерфейса

/app-cmd <parameters> - Параметры командной строки удаленного приложения

/app-file <file name> - открыть файл с помощью удаленного приложения

/compression — Использовать сжатие передаваемых данных

/shell Альтернативная оболочка командной строки

/shell-dir — Рабочий каталог для оболочки командной строки

/sound — передавать звук (Audio output)

/microphone - передавать звук с микрофона (Audio input)

/audio-mode <mode> - звуковой режим

/multimedia — передавать мультимедиа (видео)

/clipboard - передавать буфер обмена  
/serial:<device> - Перенаправить последовательное устройство  
/parallel:<device> - Перенаправить параллельное устройство  
/smartcard:<device> - Перенаправить устройство смарт-карты  
/printer:<device>,<driver> - Перенаправить устройство принтера  
/usb:<device> - Перенаправить USB устройство  
/multitouch - Перенаправление мультитач-ввода  
/window-drag — Полное перетаскивание окна  
/menu-anim — анимация меню  
/themes — темы рабочего стола  
/wallpaper — обои  
/gdi <sw|hw> - Рендеринг GDI  
/frame-ack <number> - частота кадров  
/jpeg — кодек JPEG  
/jpeg-quality <percentage> - Качество сжатия JPEG  
/sec <rdp|tls|nla|ext> - принудительный выбор протокола защиты  
/cert-name <name> - имя файла сертификата при аутентификации по ключу  
/mouse-motion — чувствительность мыши

## **smbclient**

smbclient - это клиент, который может «общаться» с сервером SMB / CIFS. Он предлагает интерфейс, аналогичный интерфейсу программы ftp. Операции включают в себя такие операции, как передача файлов с сервера на локальный компьютер, передача файлов с локального компьютера на сервер, получение информации о каталогах с сервера и т. д.

```
smbclient [-b <buffer size>] [-d debuglevel] [-e]
  [-L <netbios name>] [-U username] [-I destinationIP]
  [-M <netbios name>] [-m maxprotocol] [-A authfile] [-N] [-C]
  [-g] [-i scope] [-O <socket options>] [-p port]
  [-R <name resolve order>] [-s <smb config file>]
  [-t <per-operation timeout in seconds>] [-k] [-P]
  [-c <command>]
smbclient servicename [password] [-b <buffer size>]
  [-d debuglevel] [-e] [-D Directory] [-U username]
  [-W workgroup] [-M <netbios name>] [-m maxprotocol]
  [-A authfile] [-N] [-C] [-g] [-l log-basename]
  [-I destinationIP] [-E] [-c <command string>] [-i scope]
  [-O <socket options>] [-p port] [-R <name resolve order>]
  [-s <smb config file>] [-t <per-operation timeout in seconds>]
  [-T<c|x>IXFqgbNan] [-k]
```

### **Некоторые параметры:**

**servicename** - имя службы, которую вы хотите использовать на сервере в формате

«//сервер/ресурс», где сервер - это NetBIOS имя сервера SMB, а ресурс - это сетевое имя папки. Обратите внимание, что требуемое имя сервера НЕ обязательно является именем хоста IP (DNS) сервера!

**-U или --user=username[%password]** - Задаёт имя пользователя SMB и пароль. Если %пароль не указан, пользователю будет предложено ввести его. Если данный параметр не указан, сначала клиент проверяет переменную среды USER, затем переменную LOGNAME, и, если она существует, строка указывается в верхнем регистре. Если эти переменные среды не найдены, используется имя пользователя GUEST. Безопаснее использовать файл учетных данных, который содержит открытый текст имени пользователя и пароля. Убедитесь, что разрешения для файла ограничивают доступ от нежелательных пользователей.

**-A|--authentication-file=filename** - Эта опция задаёт файл для чтения имени пользователя и пароля в формате:

```
username = <value>
password = <value>
domain   = <value>
```

Убедитесь, что разрешения для файла ограничивают доступ от нежелательных пользователей.

**-L|--list** - Эта опция позволяет вам посмотреть, какие сервисы доступны на сервере.

**-t|--timeout <timeout-seconds>** - позволяет пользователю настраивать время ожидания, используемое для каждого запроса SMB. Значение по умолчанию составляет 20 секунд.

**-c|--command <Командная строка>** - задаёт список команд, которые должны быть выполнены (вместо запроса из стандартного потока ввода). Команды перечисляются через точку с запятой. Эту опцию удобно использовать в сценариях.

### **Внутренние команды smbclient:**

После запуска клиента пользователю предлагается приглашение:

```
smb: \>
```

Обратная косая черта («\») указывает текущий рабочий каталог на сервере и изменится, если текущий рабочий каталог будет изменен.

Подсказка указывает, что клиент готов и ожидает выполнения пользовательской команды. Каждая команда представляет собой имя команды, за которым могут следовать параметры, специфичные для этой команды. Вы можете указать имена файлов, в которых есть пробелы, заключив в кавычки имя, например «длинное имя файла»

Приведу основные команды:

**? [command]** - Выводит краткое информационное сообщение об указанной команде. Если команда не указана, отобразится список доступных команд.

**cd <directory name>** - Если указано «имя каталога», текущий рабочий каталог на сервере будет изменен на указанный каталог. Эта операция не будет выполнена, если по какой-либо причине указанный каталог недоступен. Если имя каталога не указано, будет выведен текущий рабочий каталог на сервере.

**dir <mask>** - Выводит список файлов текущего рабочего каталога, соответствующих маске.

**exit** — Разрывает соединение с сервером и выходит из программы.

**get <remote file name> [local file name]** - копирует файл с именем <remote file name> с сервера на компьютер, на котором работает клиент. Если указано, «local file name» файл записывается с этим именем. Обратите внимание, что все файлы в smbclient копируются как двоичные.

**mget <mask>** - Копирует все файлы текущего каталога, соответствующие маске, с сервера на компьютер, на котором работает клиент.

**mkdir <directory name>** - Создает новый каталог с указанным именем на сервере, если есть у пользователя есть права доступа.

**mput <маска>** - копирует файлы, соответствующие маске в текущем рабочем каталоге на локальном компьютере, в текущий рабочий каталог на сервере.

**put <local file name> [remote file name]** - копирует файл с именем <local file name> с компьютера, на котором работает клиент на сервер. Если указано, «remote file name» файл записывается с этим именем. Обратите внимание, что все файлы в smbclient копируются как двоичные.

## **mount**

Все файлы, доступные в системе Unix, расположены в одном большом дереве, иерархии файлов, с корнем в «/».

`mount [-fnrsvw] [-t fstype] [-o options] device dir`

Монтирует файловую систему, найденную на устройстве «**device**» (которая имеет тип **fstype**) в каталог **dir**. Предыдущее содержимое (если оно есть), владелец и права становятся невидимыми, и пока эта файловая система остается смонтированной, пока команда `umount` снова не отсоединит его.

Если указан только каталог или устройство, `mount` ищет точку монтирования в файле «/etc/fstab». Это облегчает повторное монтирование.

Список всех смонтированных файловых систем (типа `type`) можно получить командой:

`mount [-l] [-t type]`

Данная форма команды поддерживается только для обратной совместимости. Для



более надежного и настраиваемого вывода используйте «findmnt».

Обратите внимание, что управляющие символы в имени точки монтирования заменяются на «?».

Большинство устройств обозначается именем файла (блочного специального устройства), например /dev/sda1, но есть и другие возможности. Например, в случае монтирования NFS устройство может выглядеть как knuth.cwi.nl:/dir. Также можно указать блочное специальное устройство, используя его метку файловой системы или UUID (см. Параметры -L и -U ниже), или его метку раздела или UUID. Теги более читабельны, надежны и переносимы.

Обычно только суперпользователь может монтировать файловые системы. Однако, когда fstab содержит опцию «user» в соответствующей устройству строке, любой пользователь может смонтировать соответствующую файловую систему. Например:

```
/dev/cdrom /cd iso9660 ro,user,noauto,unhide
```

### Некоторые параметры:

**-L, --label label** - монтирует раздел с меткой label

**-U, --uuid uuid** – монтирует раздел с указанным uuid.

**-t, --types fstype** - используется для указания типа файловой системы. Подтип определяется суффиксом «.subtype». Например, 'fuse.sshfs'. Если опция -t не указана или задан автоматический тип, mount попытается угадать нужный тип с использованием библиотеки blkid. Затем mount попытается читать файл /etc/filesystems или, если он не существует, /proc/filesystems. Будут проверены все перечисленные типы файловых систем, кроме тех, которые помечены как «nodev»

**-o, --options opts** — задаёт параметры монтирования (через запятую). Параметры зависят от файловой системы.

## Литература

Вывод команды «map»

<https://ru.wikipedia.org/>

<https://wiki.debian.org/>

<https://habr.com/ru/>

<https://losst.ru/>

<https://www.opennet.ru/>