



# Lazy Linux: 10 важных практических приёмов для администраторов

Как стать более эффективным системным администратором Linux

Уровень сложности: простой

Валлард Бенинкоза, сертифицированный специалист по техническим продажам IBM, IBM

03.03.2009

Освойте эти 10 практических приемов и станьте самым лучшим администратором Linux®-систем во Вселенной - ну пусть и не во Вселенной, но для игр в высшей лиге эти знания точно понадобятся. Узнайте о туннелях SSH, VNC, восстановлении паролей, консольном шпионаже и прочем. Каждый прием сопровождается примерами, что дает вам возможность воспроизвести его на своих машинах.

Лучшие системные администраторы выделяются своей эффективностью. И если эффективный системный администратор может выполнить за 10 минут задачу, решение которой занимает 2 часа у простого смертного, становится очевидно, что труд такого эффективного системного администратора должен быть вознагражден (оплачен лучше), потому что фирма экономит время, а время — это деньги, не так ли?

Фокус в том, чтобы доказать вашу эффективность руководству. Не раскрывая именно *этого* фокуса в данной статье, я расскажу о десяти жемчужинах из сокровищницы практических фокусов администраторов Linux. Эти приемы сэкономят вам время,—и даже если вам не прибавят зарплату за повышенную производительность, то как минимум у вас появится побольше времени для игр.

## Прием 1: размонтирование не отвечающего DVD-накопителя

Новичок утверждает, что при нажатии на кнопку Eject DVD-накопителя на сервере с операционной системой, сделанной некой фирмой из Редмонда, выброс диска происходит незамедлительно. Далее он жалуется, что на большинстве промышленных серверов Linux, если в соответствующей директории запущен какой-то процесс, выброс диска не происходит. Много лет, администрируя Linux, когда я не знал, какой именно процесс работает и почему он "не отпускает" DVD-накопитель, я в таких ситуациях перегружал машину и извлекал диск. Но это неэффективно.

Вот как можно найти процесс, удерживающий DVD-накопитель, и извлечь диск к нашему удовольствию. Сначала смоделируем ситуацию. Вставьте диск в накопитель, запустите терминал и смонтируйте диск:

```
# mount /media/cdrom
# cd /media/cdrom
# while [ 1 ]; do echo "All your drives are belong to us!"; sleep 30; done
```

Теперь откройте вторую терминальную сессию и попытайтесь извлечь диск командой:

```
# eject
```

В ответ получаем:

```
umount: /media/cdrom: device is busy
```

Перед тем, как освободить диск, давайте узнаем, какой процесс его использует:

```
# fuser /media/cdrom
```

Вы увидите работающий процесс—и, увы, это наша вина, что диск заблокирован!

Теперь, имея права root, мы можем воспользоваться божественными привилегиями и убить процесс:

```
# fuser -k /media/cdrom
```

Бабах! Вот она, свобода! Теперь спокойно размонтируем накопитель:

```
# eject
```

Да здравствует команда fuser!

---

## Прием 2: очистка замусоренного экрана

Попробуйте следующее:

```
# cat /bin/cat
```

Смотрите! Экран терминала забит мусором. При попытке что-либо напечатать экран выглядит как в «Матрице». Что будем делать?

Будем набирать **reset**. Но погодите, говорите вы, команда **reset**—это почти **reboot** или **shutdown**. Как-то страшновато—особенно если все происходит на рабочем сервере.

Спокойно: можно сделать это, будучи уверенным, что машина не перезагрузится. Давайте сделаем так:

```
# reset
```

Ваш экран вернулся к нормальному состоянию. Это намного лучше, чем закрытие окна сессии и повторный вход в систему, особенно если мы заходим на сервер по SSH через пять промежуточных хостов.

---

## Прием 3: совместная работа с помощью screen

Звонит Дэвид, высокопоставленный пользователь из отдела разработки: «Мне нужна ваша помощь, я не могу понять, почему не получается откомпилировать supercode.c на новых машинах, установленных вами». «Отлично», —говорите вы, «На какой машине?»

Дэвид отвечает: «Posh». (Да, эта вымышленная компания дала пяти своим промышленным серверам имена девушек из группы Spice Girls). «ОК»,—говорите вы. Вы реализуете свои суперправа администратора и становитесь Дэвидом на другой машине:

```
# su - david
```

Затем вы заходите на Posh:

```
# ssh posh
```

Оказавшись там, выполняете команду:

```
# screen -S foo
```

Затем говорите Дэвиду: «Запусти-ка на своем терминале следующую команду»: **# screen -x foo.**"

Это приведет к объединению ваших с Дэвидом сессий в священной командной оболочке Linux. И вы, и Дэвид можете вводить команды, и вы оба будете видеть, что происходит. Это экономит время, позволяет не бегать с этажа на этаж и дает вам обоим возможность одинаково контролировать сеанс. Польза тут в

том, что Дэвид увидит ваши познания в области устранения проблем, и увидит, как вы их устраняете.

Наконец, вам обоим становится ясно, в чем проблема: сборочный скрипт Дэвида жёстко привязан к старой директории, которой больше не существует на новом сервере. Вы монтируете директорию, перекомпилируете, решаете проблему, и Дэвид возвращается к работе. А вы—к своему ленивому времяпровождению.

Единственное замечание для этого фокуса: вы оба должны зарегистрироваться в системе под одним и тем же пользователем. В `screen` можно проделывать еще много замечательных вещей: создавать несколько окон сессий, разделять экран. За подробной информацией обратитесь к руководству `man`.

Пока вы находитесь в сессии `screen`, я дам ещё один совет. Чтобы выйти из неё, оставив её открытой, введите `Ctrl-A D` (то есть нажмите клавишу `Ctrl` и нажмите на клавишу `A`. Затем нажмите на клавишу `D`).

Вы можете повторно зайти в сессию, набрав команду `screen -x foo`.

---

## Прием 4: Восстанавливаем пароль root

Вы забыли пароль для `root`. Замечательно! Все, что остается—переустановить систему. К сожалению, я видел немало людей, которые так и поступали. На самом деле довольно легко зайти в систему и изменить пароль. Это годится не для всех ситуаций (например, установив пароль для `GRUB`, вы вдруг тоже позабыли его), но здесь приведен набор действий, которые следует выполнить в обычном случае. В качестве примера я беру `Cent OS Linux`.

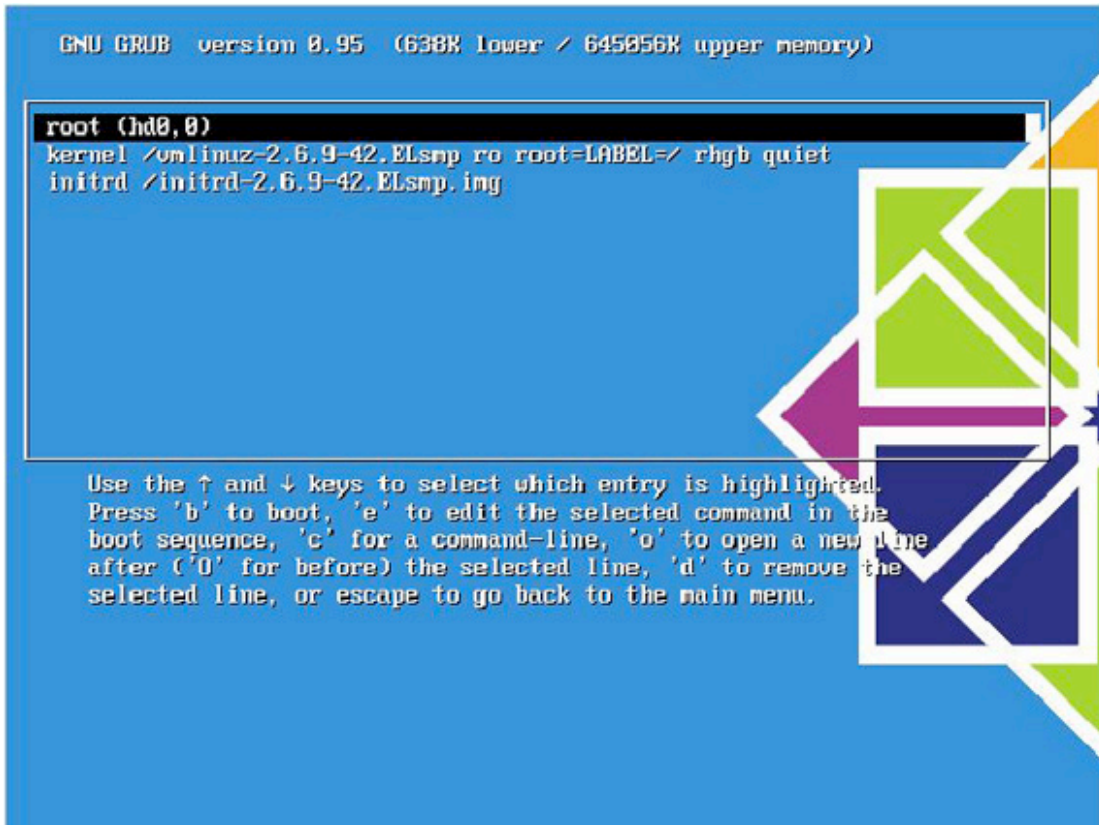
Для начала перегрузите систему. В ходе перезагрузки мы попадём в экран `GRUB`, как показано на рисунке 1. Переместите курсор, чтобы остаться в этом экране вместо загрузки по обычному сценарию.

### Рисунок 1. Экран `GRUB` после перезагрузки



Теперь посредством навигационных клавиш выберите нужное ядро для загрузки и введите **E** для редактирования строки ядра. Вы увидите что-то вроде:

**Рисунок 2: готовность к редактированию строки ядра**



Снова с помощью клавиш со стрелками выделим строку, начинающуюся со слова `kernel`, и нажимаем **E** для редактирования параметров загрузки ядра. Увидев нижеследующий экран (рисунок 3), просто добавьте `1` к аргументам, как показано ниже:

**Рисунок 3: добавьте цифру 1 к аргументу**

```
[ Minimal BASH-like line editing is supported. For the first word, TAB
 lists possible command completions. Anywhere else TAB lists the possible
 completions of a device/filename. ESC at any time cancels. ENTER
 at any time accepts your changes.]

grub edit> kernel /vmlinuz-2.6.9-42.ELsmp ro root=LABEL=/ rhgb quiet 1
```



Далее нажмите **Enter**, **B**, и ядро загрузится в однопользовательском режиме. Загрузившись, можно набрать команду `passwd` и поменять пароль пользователя `root`:

```
sh-3.00# passwd
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

Теперь можно перезагрузить систему, и она загрузится уже с новым паролем.

---

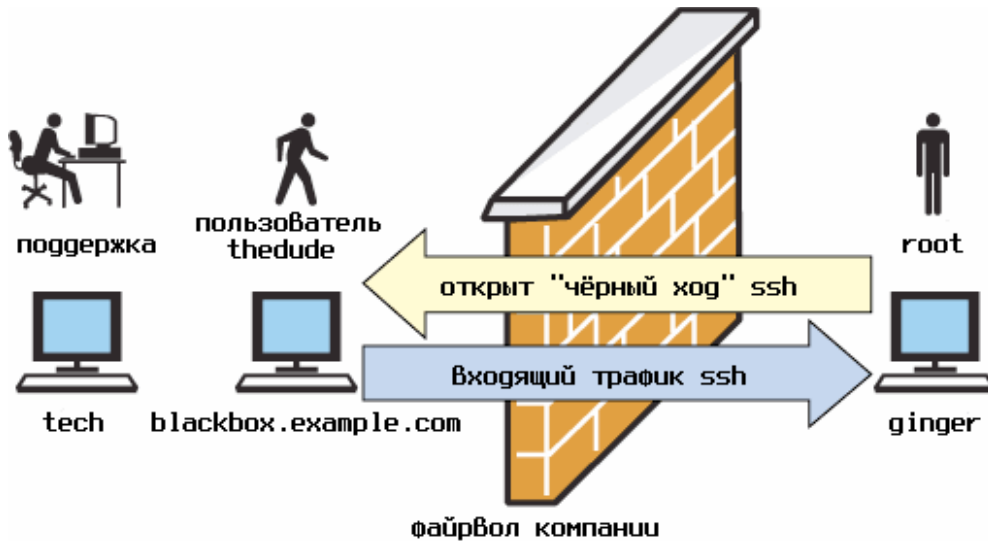
## Прием 5: Черный ход для SSH

Много раз я оказывался в ситуации, когда мне была необходима помощь от кого-то, заблокированного снаружи корпоративным брандмауэром. Немногие люди понимают, что если есть доступ через брандмауэр во внешний мир, то относительно просто можно проделать в нём ход, чтобы и внешний мир пришёл в гости к вам.

Грубо говоря, это называется «проткнуть дыру в брандмауэре». Я бы назвал это «*черным ходом для SSH*» в брандмауэре. Для его реализации нам понадобится хост в Интернете, который можно использовать как промежуточное звено.

В нашем примере мы назовем свою машину `blackbox.example.com`. Машина за корпоративным брандмауэром называется `ginger`. И, наконец, машину технической поддержки мы назовем `tech`. Следующий рисунок поясняет, как выглядит конфигурация.

### Рисунок 4: Прodelываем дырку в брандмауэре



Далее действуем так:

1. Получите разрешение на то, что вы хотите сделать, а также будьте уверены в том, что вы спрашиваете нужных людей. Большинство пользователей испугает то, что вы открыли брандмауэр, но они не понимают того, что он полностью зашифрован. Более того, тому, кто захочет проникнуть в сеть компании, понадобится сначала взломать вашу наружную машину. Опять же, возможно, вы принадлежите к школе «просить извинения вместо разрешения». В любом случае, решайте сами и не вините меня, если это противоречит вашим правилам.
2. Зайдите по SSH с хоста ginger на хост blackbox.example.com с опцией `-R`. Предполагается, что на ginger вы root, и техподдержке понадобятся его идентификационные данные, чтобы помочь вам с системой. С помощью флага `-R` мы перенаправляем инструкции с порта 2222 хоста blackbox на порт 22 ginger. Так мы организуем туннель SSH. Заметьте, что на ginger поступает только трафик SSH: мы не оставляем ginger в Интернете незащищенным.

Синтаксис нижеследующий:

```
~# ssh -R 2222:localhost:22 thedude@blackbox.example.com
```

Зайдя на хост blackbox, нужно просто оставаться зарегистрированным на нем. Я обычно использую команду типа следующей:

```
thedude@blackbox:~$ while [ 1 ]; do date; sleep 300; done
```

для того, чтобы оставить машину занятой. И сворачиваю окно.

3. 1. Теперь проинструктируйте своих друзей на хосте tech зайти по SSH от имени пользователя thedude без особых опций для SSH. Вам придется сообщить им свой пароль:
 

```
root@tech:~# ssh thedude@blackbox.example.com .
```
4. 1. Как только техподдержка вошла на blackbox, они могут зайти на ginger по SSH, используя следующую команду:
 

```
thedude@blackbox:~$ : ssh -p 2222 root@localhost
```
5. 1. Далее система запросит пароль у техподдержки. Им нужно будет ввести пароль root для хоста ginger.
6. Теперь вы и техподдержка можете работать совместно и решать проблему. Вы даже можете совместно использовать screen (см. [Приём 4.](#))

## Прием 6: удаленная сессия VNC через туннель SSH

VNC, вычисления в виртуальной сети, используются довольно давно. Как правило, я прибегаю к VNC, когда удаленный сервер выполняет графическое приложение, доступное только на этом сервере.

Например, представьте, что хост ginger из [приёма 5](#)—сервер хранения данных. Многие устройства хранения данных комплектуются программами с графическим интерфейсом для управления контроллерами внешней памяти. Часто этим графическим программам управления требуется прямое соединение с системой хранения данных по сети, которая часто бывает организована как частная подсеть. Следовательно, единственный способ получить доступ к этому графическому интерфейсу—реализовать его с хоста ginger.

Вы можете попробовать зайти на ginger по SSH с опцией `-X` и запустить графическую программу таким путём, но во многих случаях потребуется слишком большая пропускная способность канала, и вас замучит ожидание. VNC — намного более дружелюбный сетевой инструмент, готовый к использованию практически во всех операционных системах.

Давайте предположим, что конфигурация такая же, как и в нашем Приеме 5, но только нам нужно, чтобы у техподдержки был доступ по VNC вместо SSH. В этом случае мы делаем почти все то же самое, но перенаправляем порты VNC, а не SSH. Вот что нам нужно сделать:

1. Запустите сессию сервера VNC на ginger. Обычно это делается примерно следующим образом:

```
root@ginger:~# vncserver -geometry 1024x768 -depth 24 :99
```

Эти опции заставляют VNC стартовать с разрешением 1024x768 и глубиной цвета 24 бита на пиксел. Если подключение очень медленное, лучше выбрать 8 бит. Опция `:99` обозначает порт сервера VNC, с которого он будет доступен. Протокол VNC начинается с 5900, поэтому указание `:99` означает, что сервер доступен с порта 5999.

После начала сессии нас попросят ввести пароль. Идентификатор пользователя будет совпадать с тем, от имени которого был запущен сервер VNC (в нашем случае—root).

2. 1.Зайдите по SSH с ginger на `blackbox.example.com`, перенаправив порт 5999 с `blackbox` на ginger. Это делается путем исполнения команды с ginger:

```
root@ginger:~# ssh -R 5999:localhost:5999 thedude@blackbox.example.com
```

После выполнения этой команды нужно будет поддерживать сессию SSH открытой для обеспечения перенаправления порта на ginger. На этом этапе, будучи на `blackbox`, мы могли бы получить доступ к сессии VNC на ginger, просто набрав команду:

```
thedude@blackbox:~$ vncviewer localhost:99
```

Это перенаправит порт на ginger через SSH. Но нам нужно дать доступ к ginger по VNC для tech. Для реализации этого потребуется другой туннель.

3. 1.С хоста tech откройте туннель через SSH для перенаправления его порта 5999 на порт 5999 `blackbox`. Это можно сделать, введя команду:

```
root@tech:~# ssh -L 5999:localhost:5999 thedude@blackbox.example.com
```

На этот раз мы используем SSH с флагом `-L`, что позволяет вместо отправки принимать данные на 5999. Оказавшись на `blackbox`, оставьте сессию открытой. Теперь все готово для работы по VNC с хоста tech!

4. С хоста tech поднимите VNC-соединение с хостом ginger, набрав команду:

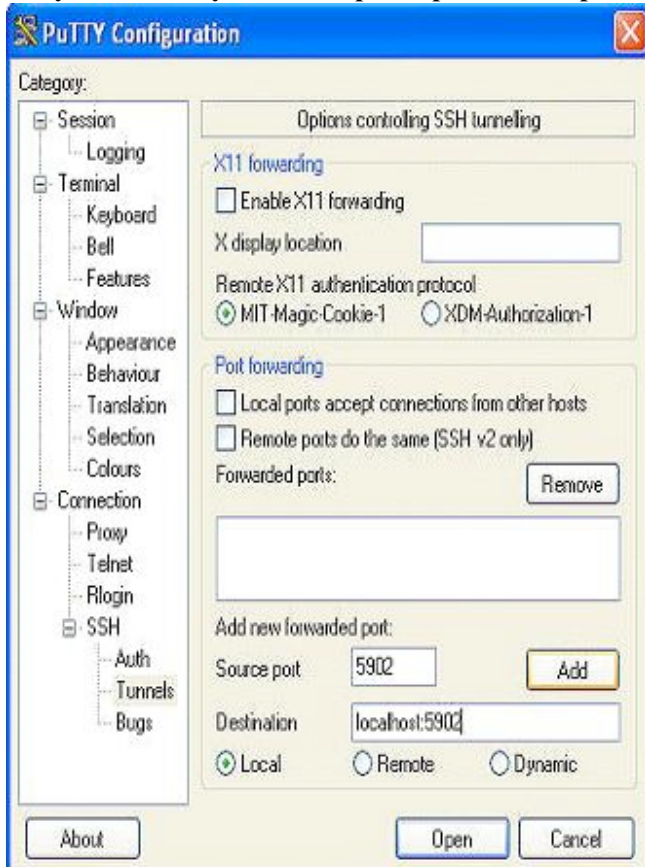
```
root@tech:~# vncviewer localhost:99 .
```

Теперь tech имеет прямое соединение с ginger по VNC.

Хотя процесс построения такой схемы может показаться сложным, он намного удобнее, чем перелет через всю страну для исправления ошибок в хранилищах данных. А если отработать его несколько раз, всё окажется не таким трудным.

Ещё один дополнительный фокус: если tech работает на платформе Windows® и не имеет консольного клиента SSH, то можно воспользоваться Putty. Putty можно настроить на перенаправление портов SSH посредством опций на панели управления. Для порта 5902 вместо использовавшегося в нашем примере 5999 нужно было бы применить настройки, показанные на рис. 5:

**Рисунок 5: Putty может перенаправлять порты SSH для туннелирования**



При таких настройках техподдержка могла бы зайти на localhost:2 по VNC точно так же, как если бы они работали под управлением Linux.

## Прием 7: проверка пропускной способности

Вообразите следующее: Компания А имеет сервер хранения данных (ginger), и он смонтирован по NFS клиентским узлом по имени beckham. Компания А решила, что им очень нужно увеличить пропускную способность доступа к ginger, потому что у них много узлов, которые нужно подключить по NFS к файловой системе ginger с общим доступом.

Наиболее распространенным и самым дешевым способом реализовать это является связка из двух гигабитных сетевых адаптеров. Это дешевле всего, поскольку, как правило, всегда найдётся дополнительный встроенный сетевой контроллер на плате и свободный порт в одном из коммутаторов.

Так все и делают. А теперь вопрос: какая реальная пропускная способность будет получена?

Теоретически гигабитный сетевой адаптер имеет предел в 128 мегабайт в секунду. Откуда взялось это число? Пожалуйста:

$1Gb = 102Mб; 1024Mб/8 = 128MB«б»$ —бит; «Б»—байт.

Но что мы видим на самом деле и как лучше провести измерения? Предлагаю инструмент под названием iperf. Скачать iperf можно, например, так:

```
# wget http://dast.nlanr.net/Projects/Iperf2.0/iperf-2.0.2.tar.gz
```

Вам понадобится установить его на общедоступной файловой системе, видимой и с ginger, и с backham. Или скомпилировать и установить на обоих узлах. Я скомпилирую пакет в домашней директории пользователя bob, которая видна с обоих узлов:

```
tar zxvf iperf*.gz
cd iperf-2.0.2
./configure --prefix=/home/bob/perf
make
make install
```

На ginger выполните команду:

```
# /home/bob/perf/bin/iperf -s -f M
```

На узле backham выполните:

```
# /home/bob/perf/bin/iperf -c ginger -P 4 -f M -w 256k -t 60
```

На обоих экранах можно будет увидеть информацию о том, какова скорость соединения. На обычном сервере с гигабитным Ethernet-адаптером скорость будет примерно 112 мегабайт/с. Это нормально, поскольку пропускная способность тратится на стек TCP и физический кабель. Соединяя напрямую два сервера, каждый с двумя связанными адаптерами Ethernet, я получаю порядка 220 мегабайт/с.

В реальности все, что вы получите с NFS при связанных сетях,—примерно 150-160 мегабит/сек. Тем не менее это хороший индикатор того, что имеющаяся пропускная способность близка к ожидаемому значению. Если результаты будут намного меньше, стоит поискать проблему.

Не так давно я столкнулся с ситуацией, когда объединили драйвером два сетевых адаптера, требовавших разных драйверов. Производительность была чрезвычайно низкая, около 20 Мбит/с, меньше, чем было бы, если бы адаптеры Ethernet не объединяли вообще!

## Прием 8: консольные скрипты и утилиты

Производительность труда системного администратора Linux можно существенно повысить путем правильного применения скриптов. Для этого нужно уметь составлять циклы и обрабатывать данные с помощью таких утилит, как **awk**, **grep** и **sed**. Существует множество ситуаций, когда их использование позволяет сократить количество команд и уменьшить вероятность ошибок пользователя.

Например, предположим, что нам нужно создать новый файл /etc/hosts для планируемого кластера Linux. Долгий способ: ручное добавление IP адресов в vi или другом любимом редакторе. Однако процесс можно существенно упростить, взяв уже существующий файл /etc/hosts и добавив к нему следующее, набрав в командной строке:

```
# P=1; for i in $(seq -w 200); do echo "192.168.99.$P n$i"; P=$((expr $P + 1));
done >>/etc/hosts
```

Это создаст две сотни имен хостов, с n001 по n200, с IP-адресами от 192.168.99.1 до 192.168.99.200. При заполнении файла такого типа вручную есть риск ввести дублирующиеся адреса или имена хостов, поэтому это очень хороший пример использования встроеной команды для предотвращения ошибок

пользователя. Обратите внимание, что это делается в оболочке `bash`, по умолчанию установленной в большинстве дистрибутивов Linux.

Далее, предположим, нам нужно убедиться в том, что объем памяти на всех узлах кластера Linux одинаков. В большинстве подобных случаев лучше всего использовать распределенную или параллельную оболочку, но для иллюстрации приведём пример с использованием SSH.

Предположим, что SSH настроена на аутентификацию без пароля. Тогда запустите:

```
# for num in $(seq -w 200); do ssh n$num free -tm | grep Mem | awk '{print $2}';  
done | sort | uniq
```

Командная строка выглядит весьма лаконично. (Может быть гораздо хуже, если добавить регулярные выражения). Давайте разберем её на составляющие и раскроем тайну.

Сначала выполняется цикл от 001 до 200. Такое заполнение нулями в начале делается с помощью опции `-w` команды `seq`. Далее мы замещаем переменную `num`, чтобы создать хост, на который нам нужно зайти по SSH. Как только он у нас есть, передаём ему команду. В нашем случае это:

```
free -m | grep Mem | awk '{print $2}'
```

Эта команда говорит:

- Вызвать команду `free` для определения памяти в мегабайтах
- Взять выходные данные этой команды и с помощью `grep` извлечь строку, содержащую текст `Mem`
- Взять эту строку и с помощью `awk` вывести второе поле, которое и показывает общий размер памяти узла

Эта операция выполняется на каждом узле.

После выполнения этой команды на каждом узле вывод для всех 200 узлов перенаправляется (`|d`) команде `sort`, чтобы отсортировать все полученные значения объемов памяти.

И, наконец, дубликаты удаляются командой `uniq`. Эта команда выдает один из следующих результатов:

- Если на всех узлах с `n001` по `n200` размер памяти одинаков, то будет выведено одно-единственное число. Это и есть размер оперативной памяти, видимый каждой операционной системой.
- Если размер памяти различается, мы увидим несколько разных чисел
- И, наконец, если соединение по SSH с каким-то узлом не состоится, будут выведены какие-то сообщения об ошибках.

Эта команда не идеальна. Если вы обнаружите, что размер памяти отличен от ожидаемого, вы не сможете определить, на каком именно узле или узлах. Для этого может понадобиться другая команда.

Тем не менее это быстрый способ что-то проверить и на скорую руку определить, все ли в порядке. В этом и суть: быстрая проверка, дешево и сердито!

---

## Прием 9: шпионаж за консолью

Некоторые программы выводят сообщения об ошибках на консоль, которая может быть не видна в сессии SSH. Но их можно просматривать с помощью устройств `vcs`. Из сессии SSH запустите на удаленном сервере следующую команду: `# cat /dev/vcs1`. Это покажет нам, что отображает первая консоль. Аналогично можно взглянуть и на другие виртуальные терминалы, подставляя вместо 1 значения 2, 3 и т.д. Если пользователь удаленной системы печатает что-либо, вы сможете видеть и то, что он набирает.

Для большинства серверных комплексов наилучшим способом для просмотра этой информации является

использование удаленного терминального сервера, коммутатора консоли или даже последовательного соединения по сети; это также дает дополнительное преимущество в виде возможности просмотра информации вне полосы пропускания основной сети. Использование устройства vcs обеспечивает быстрый внутрисетевой метод, который поможет сэкономить время, избавив от необходимости идти в серверную и смотреть на реальную консоль.

---

## Прием 10: получение различных сведений о системе

В [Приёме 8](#) мы видели пример использования командной строки для получения информации о размере оперативной памяти, установленной в системе. В этом приеме я предложу некоторые другие способы получения важных сведений о системе, которые могут понадобиться для проверки, поиска неисправностей или предоставления удаленной поддержки.

Сначала давайте получим информацию о процессоре. Это делается легко следующим образом:

```
# cat /proc/cpuinfo .
```

Эта команда выдаёт информацию о скорости, количестве и модели процессоров. Нужное значение во многих случаях можно извлечь с помощью `grep`.

Мне довольно часто приходится проверять количество процессоров, установленных в системе. Например, купив сервер с двумя четырехъядерными процессорами, я могу выполнить команду:

```
# cat /proc/cpuinfo | grep processor | wc -l .
```

Я ожидаю получить значение 8 на выходе значение 8. Если этого не случится, я позволю поставщику и попрошу прислать мне другой процессор.

Другим важным параметром является информация о диске. Ее можно получить командой `df`. Обычно я добавляю флаг `-h`, чтобы можно было увидеть вывод в гигабайтах или мегабайтах. Команда `# df -h` также показывает, какие разделы существуют на диске.

И наконец, вот способ посмотреть на микропрограммное обеспечение системы—узнать версию BIOS и микрокода сетевого адаптера.

Для проверки версии BIOS можно запустить команду `dmidecode`. К сожалению, извлечь нужную информацию командой `grep` в данном случае нелегко, так что более эффективно будет использовать `less`. На моем ноутбуке Lenovo T61 на выходе получается следующее:

```
#dmidecode | less
...
BIOS Information
Vendor: LENOVO
Version: 7LET52WW (1.22 )
Release Date: 08/27/2007
...
```

Это гораздо эффективнее, чем перезагружать машину и смотреть на вывод POST.

Для просмотра версий драйвера и прошивки Ethernet-адаптера используйте `ethtool`:

```
# ethtool -i eth0
driver: e1000
version: 7.3.20-k2-NAPI
firmware-version: 0.3-0
```

## Заключение

У опытных пользователей командной строки можно научиться тысячам полезных приемов. Лучшие способы для этого:

- Работа с другими людьми. Подключайтесь к сессиям screen и наблюдайте за тем, как работают другие—так можно узнать о различных способах решения задач. Возможно, вам придется проглотить собственную гордость и передать руководство другим, но в большинстве случаев можно научиться очень многому.
- Читайте руководства man. Seriously! Чтение страниц man, даже для команд, которые вы знаете как свои пять пальцев, может привести к удивительным озарениям. Вот, например, знаете ли вы, что с помощью awk можно реализовать сетевое программирование?
- Решайте проблемы. Будучи системным администратором, мы всегда решаем проблемы, свои или чужие—неважно. Это называется опытом, а опыт делает нас лучше и эффективнее.

Я надеюсь, что хотя бы один из этих полезных фокусов помог вам узнать что-то новое. Такие ключевые приемы делают нашу работу более квалифицированной и повышают наш опыт, но что самое главное—это экономит нам время, которое мы можем потратить на более интересные занятия, например на игры. Лучшие администраторы—лентяи, потому что они не любят работать. Они находят кратчайший путь выполнения задачи и быстрого ее решения—чтобы снова отдаться ленивому времяпровождению.

## Ресурсы

### Научиться

- Оригинал статьи [Lazy Linux: 10 essential tricks for admins](#) (EN).
- [Серия пособий по подготовке к экзамену Linux Professional Institute](#): прочная теоретическая база в дополнение к практическим приемам.
- "[Sharing computers on a Linux \(or heterogeneous\) network, Part 1](#)" (developerWorks, декабрь 2001 г.): более широкое обсуждение применения SSH и VNC. (EN)
- [Раздел Linux на developerWorks](#): дополнительные ресурсы для разработчиков Linux, а также [наиболее популярные статьи и пособия](#).
- [Изучите другие советы по Linux и учебные пособия по Linux](#) на developerWorks.
- Оставайтесь в курсе новостей, посещая раздел [технических мероприятий и Web-трансляций developerWorks](#). (EN)

### Получить продукты и технологии

- [Закажите SEK для Linux](#), комплект из двух DVD с новейшими ознакомительными версиями программного обеспечения IBM для Linux: DB2®, Lotus®, Rational®, Tivoli® и WebSphere®.(EN)
- Используйте в своем следующем проекте разработки для Linux [ознакомительное ПО IBM](#), которое можно к загрузить непосредственно с developerWorks.(EN)

### Обсудить

- [Примите участие в обсуждении материала на форуме.](#)
- [Участвуйте в жизни сообщества developerWorks](#)—посещайте блоги, форумы, подкасты и дискуссионные пространства.(EN)

## Об авторе

Валлард Бенинкоза (Vallard Benincosa)—ленивый сертифицированный ИТ-профессионал по Linux, работающий в группе IBM Linux Clusters. Он живет в Портленде, штат Орегон, с женой и двумя детьми.

## Поделиться этой статьей:

[забобрить](#)[сохранить в мемоги](#)

IBM, логотип IBM, ibm.com, DB2, developerWorks, Lotus, Rational, Tivoli, и WebSphere являются товарными знаками или зарегистрированными товарными знаками International Business Machines Corporation в США и/или других странах. Эти и другие термины товарных знаков IBM отмечены соответствующим символом (® или ™) при первом упоминании, обозначая, что зарегистрированные в США товарные знаки или знаки, охраняемые нормами общего права, принадлежали IBM на момент публикации информации. Эти товарные знаки могут также быть зарегистрированы или охраняться нормами общего права в других странах. Для просмотра текущего списка товарных знаков IBM, перейдите по [ссылке](#). Linux является зарегистрированным товарным знаком Линуса Торвальдса в США и/или других странах. Другая компания, продукт или название услуги могут быть торговыми марками или знаками обслуживания, принадлежащими иным физическим или юридическим лицам.

IBM обладает всеми авторскими правами касательно информации, расположенной на developerWorks. Использование информации приведенной на этом ресурсе без явного письменного разрешения от IBM или первоначального автора запрещены. Если Вы желаете использовать информацию с developerWorks, пожалуйста воспользуйтесь регистрационной формой для того, чтобы связаться с нами [запрос на использование материалов developerWorks Россия](#).