# Cheat Sheets

Contents

[Cheat Sheets 1](#_Toc178939547)

[Linux Operating System 2](#_Toc178939548)

[Basic System Commands 2](#_Toc178939549)

[Differences between <sudo su>, <sudo -i>, <sudo /bin/bash> 2](#_Toc178939550)

[Work with File Systems 2](#_Toc178939551)

[Catalog /proc 2](#_Toc178939552)

[Users and Groups 3](#_Toc178939553)

[Packages 3](#_Toc178939554)

[Network 3](#_Toc178939555)

[Basic Network Commands 3](#_Toc178939556)

[??? 5](#_Toc178939557)

[Kernel 5](#_Toc178939558)

[Loaders (GRUB/GRUB2) 5](#_Toc178939559)

[Initialization and Processes 5](#_Toc178939560)

[MUST HAVE Commands 5](#_Toc178939561)

[Monitoring and Debugging Commands 5](#_Toc178939562)

[Open files in Linux (management) 6](#_Toc178939563)

[??? 8](#_Toc178939564)

[Configuration Files 8](#_Toc178939565)

[Journaling 8](#_Toc178939566)

[Server Security 8](#_Toc178939567)

[Optimization and Automation 8](#_Toc178939568)

[Virtualization 8](#_Toc178939569)

[Utilities 8](#_Toc178939570)

[Telnet 8](#_Toc178939571)

[Archiving 9](#_Toc178939572)

[IP (utility) 10](#_Toc178939573)

[???? 11](#_Toc178939574)

[Bash 11](#_Toc178939575)

[Virtualization 11](#_Toc178939576)

[KVM 11](#_Toc178939577)

[Java 12](#_Toc178939578)

[JavaScript 12](#_Toc178939579)

[Node.js 12](#_Toc178939580)

[Python 13](#_Toc178939581)

## Linux Operating System

### Basic System Commands

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### Differences between <sudo su>, <sudo -i>, <sudo /bin/bash>

В чем разница между sudo su, sudo -i, sudo /bin/bash

Команда su позволяет изменить текущего пользователя терминала на другого. Команда sudo выполняет команду от имени root пользователя. Теперь разберем сам вопрос.

sudo su - выполняет команду su от имени суперпользователя. Таким образом, сначала используется sudo чтобы получить права суперпользователя, а затем пользователь меняется на root с помощью su. При этом вы останетесь в той же директории потому, что выполняется ваш .bashrc. А также .profile пользователя root поэтому вы окажетесь в окружении root.

sudo -i - похожая команда, опция -i указывает утилите что нужно переключиться в консоль от имени суперпользователя. На этот раз вы уже окажетесь в домашней папке root, и будет выполнен его .bashrc и .profile.

sudo /bin/bash - запускает еще одну оболочку bash от имени суперпользователя. Файлы конфигурации не читаются, но выполняется только .bashrc вашего пользователя. Вы не окажетесь в окружении root, а просто останетесь в своем окружении с правами суперпользователя.

### Work with File Systems

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

#### Catalog /proc

|  |  |
| --- | --- |
| /proc/cmdline | OS boot parameters |
| /proc/consoles | Current console info |
| /proc/devices | Devices drivers, working at the moment |
| /proc/dma | Current DMA channels |
| /proc/fb | Framebuffer devices |
| /proc/filesystems | Supported file systems by the current kernel version |
| /proc/iomem | Current map of the system memory for devices |
| /proc/ioports | Registered ports ranges and devices, using them |
| /proc/loadavg | System load in the time aspect + number of running processes + PID of the last process |
| /proc/locks | Files locked by kernel |
| /proc/meminfo | System memory info |
| /proc/misc | Applied devices drivers |
| /proc/modules | Loaded kernel modules (in the current moment) |
| /proc/mounts | Mount points, used by system |
| /proc/partitions | Available partitions for the system |
| /proc/pci | PCI-devices info |
| /proc/stat | Records and stat info saved since the latest restart |
| /proc/swaps | SWAP info |
| /proc/uptime | System working time in seconds |
| /proc/version | Kernel version, GCC version + installed Linux info |
|  |  |

### Users and Groups

TBD

### Packages

TBD

### Network

#### Basic Network Commands

|  |  |
| --- | --- |
| Шпаргалка по работе с сетью в linux  Краткая шпаргалка по работе с сетью в Linux, которая охватывает основные команды для диагностики и управления сетевыми интерфейсами.  1. Информация о сетевых интерфейсах  - ip a или ip addr — показывает информацию о всех сетевых интерфейсах.  - ifconfig — старый способ просмотра интерфейсов. В современных системах заменён на ip a.  - ip link show — информация о сетевых интерфейсах (без IP адресов).  2. Настройка сетевого интерфейса  - ip link set dev eth0 up/down — включить или отключить сетевой интерфейс (заменить eth0 на нужный интерфейс).  - ip addr add 192.168.1.100/24 dev eth0 — назначить IP-адрес интерфейсу eth0.  - ip addr del 192.168.1.100/24 dev eth0 — удалить IP-адрес с интерфейса.  3. Маршрутизация  - ip route show — показать таблицу маршрутизации.  - ip route add default via 192.168.1.1 — установить шлюз по умолчанию (заменить на нужный IP).  - ip route del default — удалить маршрут по умолчанию.  4. DNS  - cat /etc/resolv.conf — показать используемые DNS-сервера.  - nslookup example.com — получить IP-адрес по доменному имени.  - dig example.com — расширенная диагностика DNS-запросов.  5. Проверка сетевых подключений  - ping 192.168.1.1 — отправить ICMP-запрос к узлу (заменить на нужный IP или домен).  - traceroute example.com — показать маршрут до узла.  - mtr example.com — комбинированная команда ping и traceroute с обновлением в реальном времени.  6. Прослушивание и управление портами  - ss -tuln — показать открытые порты и активные соединения.  - netstat -tuln — устаревшая, но часто используемая команда для тех же целей.  - lsof -i :80 — показать процессы, использующие порт 80.  7. Проверка доступных сетей и соединений  - iwconfig — информация о беспроводных интерфейсах (для Wi-Fi).  - nmcli — управление соединениями через NetworkManager.  - nmtui — текстовый интерфейс для управления сетью через NetworkManager.  8. Тестирование и диагностика  - tcpdump -i eth0 — захват пакетов с интерфейса eth0.  - curl ifconfig.me — получить внешний IP-адрес.  - wget http://example.com/file.zip — скачать файл по HTTP/HTTPS.  9. Firewall (iptables)  - iptables -L — показать правила файрвола.  - iptables -A INPUT -p tcp --dport 22 -j ACCEPT — разрешить доступ по SSH (порт 22).  - iptables -D INPUT -p tcp --dport 22 -j ACCEPT — удалить правило.  10. Проброс портов  - iptables -t nat -A PREROUTING -p tcp --dport 8080 -j REDIRECT --to-port 80 — перенаправить запросы на порт 8080 к порту 80. |  |
|  |  |
| Работа с сетью  Отладка Linux  # ethtool eth0 - Показать Ethernet статус  # ethtool -s eth0 speed 100 duplex full - Принудительная установка режима 100Mbit Full duplex  # ethtool -s eth0 autoneg off - Отключить автоопределение  # ethtool -p eth1 - Мигать индикатором сетевой карты - если поддерживается  # ip link show - Список сетевых интерфейсов в Linux (подобна ifconfig)  # ip link set eth0 up - Активировать сетевой интерфейс (или отключить). Аналог "ifconfig eth0 up"  # ip addr show - Список всех IP адресов в Linux ( аналог ifconfig)  # ip neigh show - Тоже что и arp -a  Печать таблицы маршрутизации  # route -n # Linux или используйте "ip route"  # netstat -rn # Linux, BSD и UNIX  # route print # Windows  Используйте "add -p" что-бы сделать маршрут постоянным.Настройка дополнительных IP адресов Linux  # ifconfig eth0 192.168.50.254 netmask 255.255.255.0 # Первый IP адрес  # ifconfig eth0:0 192.168.51.254 netmask 255.255.255.0 # Второй IP адрес  # ip addr add 192.168.50.254/24 dev eth0 # Эквивалентные команды для ip  # ip addr add 192.168.51.254/24 dev eth0 label eth0:1 |  |
|  |  |

#### ???

TBD

### Kernel

TBD

### Loaders (GRUB/GRUB2)

TBD

### Initialization and Processes

TBD

### MUST HAVE Commands

#### Monitoring and Debugging Commands

|  |  |
| --- | --- |
| top | Show running processes and used resources with autoupdate |
| ps -eafw | Show running processes and used resources + other info – show once |
| ps -e -o pid,args --forest | Show PIDs and processes as a tree |
| pstree | Show processes tree |
| kill -9 98989  kill -KILL 98989 | Kill the process with PID = 98989 (no data consistency) |
| kill -TERM 98989 | Kill the process with PID = 98989 (gracefully) |
| kill -1 98989  kill -HUP 98989 | Force the process with PID = 98989 to reload it’s config |
| kill -s SIGSTOP 98989 | Stop running process with PID = 98989 |
| kill -s SIGCONT 98989 | Continue stopped process with PID = 98989 |
| lsof -p 98989 | Show files opened by the process with PID = 98989 |
| lsof /home/user1 | Show files opened from the directory /home/user1 |
| strace -c ls > /dev/null | Show list of the system calls created and received by the process ls |
| strace -f -e open ls > /dev/null | Show calls to the libraries |
| watch -n1 ‘cat /proc/interrupts’ | Show interrupts in the real time |
| last reboot | Show the system’s reboot history |
| last user1 | Show history of registration user1 in the system and time spent in the system |
| lsmod | Show loaded kernel modules |
| free -m | Show the RAM state in megabytes |
| smartctl -A /dev/had | Control the hard drive /dev/had state via SMART |
| smartctl -i /dev/had | Check whether SMART is available on the hard drive /dev/had |
| tail /var/log/dmesg | Show the last 10 records from the kernel load journal |
| tail /var/log/messages | Show the last 10 records from the system journal |
|  |  |

#### Open files in Linux (management)

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
| Как найти количество открытых файлов в Linux  Значение сохраняется в  # cat /proc/sys/fs/file-max  Проверка Hard лимит в Linux  # ulimit -Hn  Проверка Soft лимит в Linux  # ulimit -Sn  Чтобы просмотреть значения Hard и Soft для разных пользователей, переключите пользователя с «su» на пользователя, ограничения которого вы хотите проверить.  Например:  # su root  $ ulimit -Sn  $ ulimit -Hn  Как проверить количество системных дескрипторов файлов в Linux  На серверах, некоторые из ваших приложений могут потребовать более высокий лимит для открытых дескрипторов файлов. Хорошим примером этого являются службы MySQL / MariaDB или веб-сервер Apache.  Вы можете увеличить лимит открытых файлов в Linux, отредактировав опцию ядра fs.file-max. Для этого используйте утилиту sysctl.  Sysctl используется для настройки параметров ядра.  Например, чтобы увеличить ограничение на количество открых файла до 500000, вы можете использовать следующую команду:  # sysctl -w fs.file-max=500000  Вы можете проверить текущее значение для количества открытых файлов с помощью следующей команды:  $ cat /proc/sys/fs/file-max  С помощью указанной команды изменения, которые вы внесли, будут оставаться активными только до следующей перезагрузки. Если вы хотите применить их на постоянной основе, вам придется отредактировать следующий файл:  # vi /etc/sysctl.conf  Добавьте следующую команду:  fs.file-max=500000  Конечно, вы можете изменить число в соответствии с вашими потребностями. Чтобы снова проверить изменения, используйте:  # cat /proc/sys/fs/file-max  Чтобы изменения вступили в силу, пользователям необходимо будет выйти из системы и снова войти в систему. Если вы хотите немедленно применить ограничения, вы можете использовать следующую команду:  # sysctl -p  Установка количества открытых файлов для пользователя в Linux  Вышеприведенные примеры показали, как установить ограничения, но вы можете установить лимит для каждого пользователя. Для этой цели, с правами пользователя root, вам необходимо отредактировать следующий файл:  # vi /etc/security/limits.conf  Если вы администратор Linux, предлагаем Вам ознакомиться с этим файлом. Прочитайте все комментарии в нем, поскольку он обеспечивает гибкость в плане управления системными ресурсами путем установки лимита пользователям на разных уровнях.  Команды, которые вы должны добавить, имеют следующие параметры:  <domain>        <type>  <item>  <value>  Вот пример установки Soft и Hard лимитов для пользователя dimon:  ## Example hard limit for max opened files  dimon hard nofile 4096  ## Example soft limit for max opened files  dimon soft nofile 1024 |  |
|  |  |
|  |  |

#### ???

TBD

### Configuration Files

TBD

### Journaling

TBD

### Server Security

TBD

### Optimization and Automation

TBD

### Virtualization

TBD

## Utilities

### Telnet

|  |  |
| --- | --- |
| telnet <hostname> <port> | Basic usage |
| telnet example.com 80 | Basic usage - example |
| telnet <dns-server> 53 | Check DNS server |
| Ctrl + ]  *quit*  *help*  *open <hostname> <port>*  *close*  *display*  *mode*  *send*  *status*  *set / unset* | Close session (move to the command mode), below see some of the commands:  - type <quit> after move to the command mode – close connection  - show possible commands  - open new connection to the host specified  - close connection  - current connection parameters  - switch current mode, for ex. *line* or *`character`*  - send control symbols, for ex. *send escape*  - current connection status  - sets / unsets the current connection parameters, for ex. *set echo* / *unset echo* |
| telnet <HTTP server> <port>  *GET / HTTP/1.1*  *Host: example.com* | Sending text request after connection to the HTTP server |
|  |  |

### 

### Archiving (zip/gzip/unzip/rar)

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Полезные команды Linux. Архивация и сжатие файлов  # gzip -9 file1  Поместить файл file1 в архив gzip с максимальной степенью сжатия  # rar a file1.rar file1 file2 dir1  Создать rar архив file1.rar, заархивировав файлы: file1, file2 и директорию: dir1  # rar a file1.rar test\_file  Упаковать в rar архив file1.rar файл test\_file  # rar x file.rar  Разархивировать rar архив file.rar  # bzip2 file1  Сжимает файл file1  # bunzip2 file1.bz2  Разжимает файл file1.bz2  # gzip file1  Сжимает файл file1  # gunzip file1.gz  Разжимает файл file1.gz  # tar -cvf archive.tar file1 file2 dir1  Создать tar архив archive.tar, упаковав в него файлы file1, file2 и директорию dir1  # tar -cvf archive.tar file  Упаковать в tar-архив archive.tar, файл file  # tar -tf archive.tar  Вывести содержимое tar архива  # tar -xvf archive.tar  Распаковать tar архив  # tar -xvf archive.tar -C /tmp  Распаковать архив в /tmp  # tar -cvfz archive.tar.gz dir1  Создать tar архив и сжать его с помощью программы gzip  # tar -xvfz archive.tar.gz  Разжать tar архив и распаковать его  # tar -cvfj archive.tar.bz2 dir1  Создать архив и сжать его с помощью bzip2 ( ключ -j работает не во всех \*nix системах )  # tar -xvfj archive.tar.bz2  Разжать архив и распаковать его ( ключ -j работает не во всех \*nix системах )  # zip file1.zip file1  Создать сжатый zip-архив  # zip -r file1.zip file1 file2 dir1  Запаковать в архив несколько файлов и/или директорий  # unzip file1.zip  Разжать и распаковать zip-архив  # unrar x file1.rar  Распаковать rar-архив |  |
|  |  |
|  |  |

### IP (utility)

|  |  |
| --- | --- |
| Команда ifconfig считается устаревшей и постепенно вытесняется командой ip  ip addr show - отобразить список сетевых интерфейсов с адресами. Возможно сокращение параметров команды - ip a sh или ip a s  ip addr list permanent - отобразить только статические ip адреса  ip addr list dynamic - отобразить только динамические ip адреса  ip addr add 1.1.1.13/24 dev eth0 - установить ip адрес для интерфейса eth0  ip addr del 1.1.1.13/24 dev eth0 - удалить ip адрес интерфейса eth0  ip addr flush dev eth0 - удалить все ip адреса интерфейса eth0  ip addr help - отобразить справку по команде ip addr  ip route show - отобразить сведения о маршрутах. В сокращенном варианте - ip r s  ip route show table 255 - отобразить все маршруты из таблицы 255 ip route get 10.10.20.0/24 - отобразить маршрут к этой сети ip route get 10.10.20.0/24 from 192.168.1.100 - отобразить маршрут к этой сети от указного интерфейса с IP=192.168.1.100.  ip route add 10.10.20.0/24 via 192.168.5.1 - создать маршрут к заданной сети через интерфейс c IP=192.168.5.1  ip route delete 10.10.20.0/24 - удалить указанный маршрут.  ip route del 10.10.20.0/24 via 192.168.50.100 - удалить указанный маршрут.  ip route add default via 192.168.50.100 - создать маршрут по умолчанию.  ip route add 10.10.20.0/24 dev eth0 - создать маршрут к указной сети .  ip route add table nnn 10.10.20.0/24 dev eth0 - создать маршрут в специальной таблице маршрутов с номером nnn.  ip route add to blackhole 10.10.0.0/16- создать фиктивный маршрут.  ip route add unreachable 10.10.0.0/16 - создать недостижимый маршрут.  ip link show - отобразить список сетевых интерфейсов.  ip link show eth0 - отобразить состояние eth0  ip link list up - отобразить статус всех включенных интерфейсов  ip link set eth1 up - включить интерфейс eth1  ip link set eth1 down - выключить интерфейс eth1  nslookup - запуск nslookup в интерактивном режиме.  nslookup mail.ru - отобразить сведения для домена mail.ru с использованием сервера DNS из сетевых настроек.  nslookup mail.ru 8.8.8.8 - отобразить сведения для домена mail.ru с использованием сервера DNS с адресом 8.8.8.8 (публичный DNS-сервер Google).  nslookup –query=mx mail.ru - отобразить запись типа MX для домена mail.ru. |  |
|  |  |
|  |  |

## ????

TBD

## Bash

TBD

## Virtualization

### KVM (console commands)

|  |  |
| --- | --- |
| virsh list [--all] | Show list of running VMs, with the flag --all – show all VMs (even powered off) |
| virsh shutdown <vm name> | Shutdown the VM by name <vm name> |
| virsh start <vm name> | Start the VM by name <vm name> |
| virsh suspend <vm name> | Suspend the VM by name <vm name> |
| virsh resume <vm name> | Resume the VM by name <vm name> |
| virsh reboot <vm name> | Reboot the VM by name <vm name> |
| virsh destroy <vm name> | Destroy the VM by name <vm name> |
| virsh undefined <vm name> | Remove VM from the list and remove all files related, usually called after **virsh destroy <vm>** |
|  |  |
|  |  |
|  |  |
| # virsh vcpuinfo <vm name> — информация о процессоре на виртуальной машине  # virsh domid <vm name> — получить идентификатор виртуальной машины  # virsh domuuid <vm name> — получить UUID виртуальной машины  # virsh dominfo <vm name> — получить сведения о виртуальной машине  # virsh domstate <vm name> — просмотр состояния виртуальной машины  # virsh dumpxml <vm name> — вывести файл конфигурации указанной виртуальной машины в XML формате |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Containerization

TBD

### Docker / Docker Compose

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Java

TBD

## JavaScript

TBD

## Node.js

* <https://nodejs.org> – Node.js itself
* <https://www.npmjs.com> – Node Package Manager
* <https://github.com/nvm-sh/nvm> - Node Version Manager for Linux environment (POSIX-compliant bash script to manage multiple active node.js versions)
* <https://github.com/coreybutler/nvm-windows> - NVM for Windows environment

|  |  |
| --- | --- |
| npm init -y | Initialization Node.js packaje in the current directory (package.json will be created). |
| node <js script> | Execution of the Node.js application. |
| npm install [package(s) name(s)] [flag(s)] | Install the latest version of the node package. If specified package name as: <package>@<version> - the specified version will be installed. If no package name specified, packages specified in the **package.json** will be installed. Flags are relevant for all modes: **--global**|**-g** – globally, **--save**|**-S**|<no flags> - as runtime and development time dependency, **--save-dev**|**-D** – development time dependency. Can be specified multiple multiple packages, separated by space. |
| npm list | Show all installed packages. Flags: **-g** – show globally installed packages. |
| npm view <package> versions | Show all available package versions. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Python

TBD

### Virtual Environment

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |