Linux cheatsheet

Content

1. System information
2. Hardware information
3. Performance monitoring and statistics
4. User information and management
5. File and directory command
    - Config files
6. Process management
7. Networking
8. SSH management
    - Client
    - Server
    - Tunnel
    - Proxy
9. Archives (tar files)
10. Installing packages
11. Search
12. Remote work
13. FFMPEG
    - Summary
    - Codecs and containers
    - Examples

To be added

1. timeout
2. SUID
3. rdiff-backup
4. tmux
5. screen to UART

SYSTEM INFORMATION

System monitor

```
htop
```

System monitor older (standard on all Linux distros)

```
top
```

Display Linux system information

```
uname -a
```

Display kernel release information

```
uname -r
```

Show which version of Red Hat installed

```
cat /etc/redhat-release
```

Show how long the system has been running + load

```
uptime
```

Show system host name

```
hostname
```

Display all local IP addresses of the host.

```
hostname -I
```

Show system reboot history

```
last reboot
```

Show the current date and time

```
date
```

Show this month's calendar

```
cal
```

Display who is online

```
w
```

Who you are logged in as

```
whoami
```

History of commands

```
history
```

System daemons

```
systemctl (stop/start/restart/status) daemonName
```

HARDWARE INFORMATION

Device manager equivalent + benchmarking software

```
hardinfo
```

Display messages in kernel ring buffer

```
dmesg
```

Display CPU information

```
cat /proc/cpuinfo
```

Display memory information

```
cat /proc/meminfo
```

Display free and used memory (-h for human readable, -m for MB, -g for GB.)

```
free -h
```

Display PCI devices

```
lspci -tv
```

Display USB devices

```
lsusb -tv
```

Display DMI/SMBIOS (hardware info) from the BIOS

```
dmidecode
```

Show info about disk sda

```
hdparm -i /dev/sda
```

Perform a read speed test on disk sda

```
hdparm -tT /dev/sda
```

Test for unreadable blocks on disk sda

```
badblocks -s /dev/sda
```

PERFORMANCE MONITORING AND STATISTICS

Display and manage the top processes

```
top
```

Interactive process viewer (top alternative)

```
htop
```

Display processor related statistics

```
mpstat 1
```

Display virtual memory statistics

```
vmstat 1
```

Display I/O statistics

```
iostat 1
```

Display the last 100 syslog messages (Use /var/log/syslog for Debian based systems.)

```
tail -100 /var/log/messages
```

Capture and display all packets on interface eth0

```
tcpdump -i eth0
```

Monitor all traffic on port 80 (HTTP)

```
tcpdump -i eth0 'port 80'
```

List all open files on the system

```
lsof
```

List files opened by user

```
lsof -u user
```

Display free and used memory (-h for human readable, -m for MB, -g for GB.)

```
free -h
```

Execute "df -h", showing periodic updates

```
watch df -h
```

USER INFORMATION AND MANAGEMENT

Display the user and group ids of your current user.

```
id
```

Display the last users who have logged onto the system.

```
last
```

Show who is logged into the system.

```
who
```

Show who is logged in and what they are doing.

```
w
```

Change password

```
passwd
```

Change password of user (sudo)

```
sudo passwd user
```

Create a group named "test".

```
groupadd test
```

Create an account named john with creating process

```
adduser john
```

Create an account named john, with a comment of "John Smith" and create the user's home directory.

```
useradd -c "John Smith" -m john
```

Add the john account to the sales group

```
usermod -aG sales john
```

Delete the john account. (leaves home dir and mail spool untouched)

```
userdel john
```

Delete the john including his home dir and mail spool.

```
userdel -r john
```

Config files

Files that are executed on user login

- **~/.profile** - for things that are not specifically related to Bash, like environment variables PATH and friends, and should be available anytime
- **~/.bashrc** - for the configuring the interactive Bash usage, like Bash aliases, setting your favorite editor, setting the Bash prompt
- **~/.bash_profile** - for making sure that both the things in .profile and .bashrc are loaded for login shells.
- **/etc/profile** - for all users logging in to the bash, ksh, or sh shells. This is usually where the PATH variable, user limits, and other settings are defined for users. This file is only run for login shell and therefore does not run when a script is executed.
- **/etc/sudoers** - to config this file always use command *visudo*. This file contains configuration for *sudo* users including their own *PATH* variable which is used during usage of *sudo* command instead of their own.

FILE AND DIRECTORY COMMANDS

```
Linux chmod example
PERMISSION      EXAMPLE

 U   G   W
rwx rwx rwx     chmod 777 filename
rwx rwx r-x     chmod 775 filename
rwx r-x r-x     chmod 755 filename
rw- rw- r--     chmod 664 filename
rw- r-- r--     chmod 644 filename


LEGEND
U = User
G = Group
W = World

r = Read (for directory it enables list the files inside)
w = write
x = execute (for directory it enables access to files inside)
- = no access
d = directory
s = symlink


SUID - file will run with its owners privileges
```

List all files in a long listing (detailed) format

```
ls -al
```

Change group of file:

```
chgrp group_name file_name
```

Display the present working directory

```
pwd
```

Create a directory

```
mkdir directory
```

Remove (delete) file

```
rm file
```

Remove the directory and its contents recursively

```
rm -r directory
```

Force removal of file without prompting for confirmation

```
rm -f file
```

Forcefully remove directory recursively

```
rm -rf directory
```

Copy file1 to file2

```
cp file1 file2
```

Copy source_directory recursively to destination. If destination exists, copy source_directory into destination, otherwise create destination with the contents of source_directory.

```
cp -r source_directory destination
```

Rename or move file1 to file2. If file2 is an existing directory, move file1 into directory file2

```
mv file1 file2
```

Create symbolic link to linkname

```
ln -s /path/to/file linkname
```

Create an empty file or update the access and modification times of file.

```
touch file
```

View the contents of file

```
cat file
```

Browse through a text file

```
less file
```

Display the first 10 lines of file

```
head file
```

Display the last 10 lines of file

```
tail file
```

Display the last 10 lines of file and "follow" the file as it grows.

```
tail -f file
```

Securely remove data

```
shred fileToDel rm fileToDel
```

Find file in system

```
find /dirToSearch -name "regexOfName"
```

Find empty directories

```
find /dir -type f -empty
```

Find executable files

```
find /dir -perm /a=x
```

PROCESS MANAGEMENT

Display your currently running processes

```
ps
```

Display all the currently running processes on the system.

```
ps -ef
```

Display process information for processname

```
ps -ef | grep processname
```

Display max power processes

```
top
```

Interactive process viewer (top alternative)

```
htop
```

Kill process with process ID of pid

```
kill pid
```

Kill all processes named processname

```
killall processname
```

Kill process by name

```
pkill -f nameofprocess
```

Start program in the background

```
program &
```

Display stopped or background jobs

```
bg
```

Brings the most recent background job to foreground

```
fg
```

Brings job n to the foreground

```
fg n
```

NETWORKING

Display all network interfaces and IP address

```
ip a
```

Display eth0 address and details

```
ip addr show dev eth0
```

Query or control network driver and hardware settings

```
ethtool eth0
```

Send ICMP echo request to host

```
ping host
```

Network path to host

```
traceroute host
```

Display whois information for domain

```
whois domain
```

Display DNS information for domain

```
dig domain
```

Reverse lookup of IP_ADDRESS

```
dig -x IP_ADDRESS
```

Display DNS IP address for domain

```
host domain
```

Display the network address of the host name.

```
hostname -i
```

Display all local IP addresses of the host.

```
hostname -I
```

Download http://domain.com/file

```
wget http://domain.com/file
```

Display listening tcp and udp ports and corresponding programs

```
netstat -nutlp
```

```
sudo lsof -i -P -n | grep LISTEN
```

Config of DNS server

```
resolvectl status | less
```

Allow port 80

```
ufw allow 80
```

SSH MANAGEMENT

Client:

User specific ssh data should be stored in `~/.ssh directory`

Connect to host as with your username.

```
ssh host
```

Connect to host as user

```
ssh user@host
```

Connect to host using port

```
ssh -p port user@host
```

Open tunnel for application display

```
ssh -X user@host
```

Generate new RSA key pair

```
ssh-keygen
```

Copy local file to home folder on server via SSH using key:

```
scp -i ~/.ssh/id_rsa.pub file_to_copy USER@SERVER:~/remote_file_path
```

Copy remote file to local via SSH using key:

```
scp -i ~/.ssh/id_rsa.pub USER@SERVER:~/remote_file_to_copy local_file_path
```

SSH config file record structure

```
Host name1
    ...

Host my-ssh-host
    HostName 10.0.0.5
    Port 22
    User myuser
        IdentityFile ~/.ssh/id_ed25519_myuser #file with private key
        IdentitiesOnly yes   #ssh won't try all keys, just mentioned ones
    ProxyJump name1, name2 #chaining proxies
```

Save public key on SSH server

```
ssh-copy-id -i ~/.ssh/id_rsa.pub user@host
```

Server:

SSH config file: `/etc/ssh/sshd_config`

Installation of openssh-server (might fail - in that case try 'apt remove openssh-client')

```
apt install openssh-server
```

Disable password login

```
in config file add/modify line PasswordAuthentication no and then restart
```

Disable root login

```
in config file add/modify line PermitRootLogin no then restart
```

Restart ssh server

```
systemctl restart sshd
```

SSH public keys usable for login to specific user are in file: `~/.ssh/authorized_keys`

SSH tunneling

This will open tunnel from local port 9000 to remote port 25 of mail.server.com

```
ssh -L 9000:remoteserver.com:25
```

Tunnel from localhost to host1 (connection from host1 to host2 is not secured)

```
ssh -L 9999:host2:1234 -N host1
```

This will open a tunnel from localhost to host1 and another tunnel from host1 to host2. However the port 9999 to host2:1234 can be used by anyone on host1. This may or may not be a problem.

```
ssh -L 9999:localhost:9999 host1
```

```
ssh -L 9999:localhost:1234 -N host2
```

This will open a tunnel from localhost to host1 through which the SSH service on host2 can be used. Then a second tunnel is opened from localhost to host2 through the first tunnel.

```
ssh -L 9998:host2:22 -N host1
```

```
ssh -L 9999:localhost:1234 -N -p 9998 localhost
```

SSH proxy

Dynamic or multi-port forwarding

List used ports:

```
sudo lsof -i -P -n | grep LISTEN
```

Create socket on localhost:1080

```
ssh -D 1080 remoteServer
```

Wherever any request comes to localhost:1080, it is forwarded to remoteServer which asks on behalf of original user

For example create dynamic tunel with previous command. In brower go to network configuration a set SOCKS proxy server to LOCALHOST:1080 Now every request will got through remoteServer first.

ARCHIVES (TAR FILES)

Create tar named archive.tar containing directory.

```
tar cf archive.tar directory
```

Extract the contents from archive.tar.

```
tar xf archive.tar
```

Create a gzip compressed tar file name archive.tar.gz.

```
tar czf archive.tar.gz directory
```

Extract a gzip compressed tar file.

```
tar xzf archive.tar.gz
```

Create a tar file with bzip2 compression

```
tar cjf archive.tar.bz2 directory
```

Extract a bzip2 compressed tar file.

```
tar xjf archive.tar.bz2
```

INSTALLING PACKAGES

Debian package managers: apt or yum Search for a package by keyword.

```
apt search keyword
```

Install package.

```
apt install package
```

Display description and summary information about package.

```
apt info package
```

Install package from local file named package.rpm

```
rpm -i package.rpm
```

Remove/uninstall package.

```
apt remove package
```

Install software from source code.

```
tar zxvf sourcecode.tar.gz
cd sourcecode
./configure
make
make install
```

SEARCH

Search for pattern in file

```
grep pattern file
```

Search recursively for pattern in directory

```
grep -r pattern directory
```

Find files and directories by name

```
locate name
```

Find files in /home/john that start with "prefix".

```
find /home/john -name 'prefix*'
```

Find files larger than 100MB in /home

```
find /home -size +100M
```

Search for process by name (use `echo $?` to check result)

```
pgrep process_name
```

REMOTE WORK

Screen

Start a screen session

```
screen
```

Start named screen session

```
screen -S session_name
```

Shortcuts to manage single screen session

```
Ctrl+a c      Create a new window (with shell).
Ctrl+a "      List all windows.
Ctrl+a 0      Switch to window 0 (by number).
Ctrl+a A      Rename the current window.
Ctrl+a S      Split current region horizontally into two regions.
Ctrl+a |      Split current region vertically into two regions.
Ctrl+a tab    Switch the input focus to the next region.
Ctrl+a Ctrl+a Toggle between the current and previous windows
Ctrl+a Q      Close all regions but the current one.
Ctrl+a X      Close the current region.
```

Detach from screen session

```
Ctrl+a d
```

Detach and terminate session

```
Ctrl+d
```

List screens

```
screen -ls
```

Restore screen "10835.session_name"

```
screen -r 10835
```

Nano editor

Install nano

```
sudo apt-get install nano
```

Open file using nano

```
nano filename
```

Options:

```
-w       Opens the file in a standard format - does not wrap text to fit
screen
```

Shortcuts:

```
Command Explanation
CTRL + A    Lets you jump to the beginning of the line.
CTRL + E    Lets you to jump to the end of the line.
CTRL + Y    Scrolls page down.
CTRL + V    Scrolls page up.
CTRL + G    A Help window will pop out and show you all the available
commands.
CTRL + O    To save the file. Nano will ask you to edit or verify the
desired file name.
CTRL + W    Search for a specified phrase in your text. Press ALT + W to
search for the same phrase again.
CTRL + K    It cuts the entire selected line to the cut buffer (similar to
clipboard).
CTRL + U    To paste the text from the cut buffer into the selected line.
CTRL + J    Justifies the current paragraph.
CTRL + C    Shows the current cursor position in the text
(line/column/character).
CTRL + R    Opens a file and inserts it at the current cursor position.
CTRL + X    To exit Nano text editor. It prompts a save request if you made
any changes to the file.
CTRL + \    Replaces string or a regular expression.
CTRL + T    Invokes the spell checker, if available.
CTRL + _    Lets you go to the specified line and column number.
ALT + A To select text. You can combine this command with CTRL + K to cut a
specific part of the text to the cut buffer.
```

FFMPEG

Summary

```
file name as last param     Output file/url
-i file/url                 Input file/url
-c:v h264                   Set videocodec of output to H.264
-c:a copy                   Copy audio without any changes
-b:v 1M                     Set video bitrate to 1M/s
-b:a 1M                     Set audio bitrate to 1M/s
-r 30                       Limit framerate to 30FPS
-t 10 (before -i)           Listen on input for 10 seconds
-t 10 (after -i)            Listen on input until output file has 10
seconds
-ss 00:01:00                Start after 1 minute
-s  hd720                   Set video resolution of output
-vn                         Ignore video stream
-an                         Ignore audio stream
-sn                         Ignore subtitles stream
-y                          Overwrite output files without asking
-n                          Never overwrite existing output files
```

Codecs and containers

Codec	FFMPEG encoder	FFMPEG decoder	Audio/Video	Description
H.264 / AVC	libx264	h264	Video	(Advanced Video Coding) - Great compatibility, loseless compression
H.265 / HEVC	libx265	hevc	Video	(The High Efficiency Video Coding) lossy compression, less compatibility
VP8 VP9	libvpx, libvpx-vp9	libvpx, libvpx-vp9	Video	Webm format codecs. Usable for web
MP3	libmp3lame	mp3	Audio	MP3 uses lossy compression and offers high compression rates, resulting in small files practical for online streaming and internet download
FLAC	flac	flac	Audio	One of the best free lossless audio codecs. FLAC is an open-source codec

Container	Codecs	Description

Container	Codecs	Description
.mp4	H.264, H.265, MP3	(Advanced Video Coding) - Great compatibility, loseless compression
.webm	VP8, VP9	Great compatiblity with browsers
.mkv	any	Matroska, allows to use any codec

Examples

Full help

```
ffmpeg -h full
```

Available codecs

```
ffmpeg -codecs
```

Basic conversion with autodetect (requires well-defined formats)

```
ffmpeg -i input.mp3 output.ogg
ffmpeg -i input.mp4 output.webm
```

Create MKV (Matroska) container with a VP9 video stream and a Vorbis audio stream

```
ffmpeg -i input.mp4 -c:v vp9 -c:a libvorbis output.mkv
```

Copy the audio (-c:a copy) from input.webm and convert the video to a VP9 codec (-c:v vp9) with a bit rate of 1M/s (-b:v), all bundled up in a Matroska container (output.mkv).

```
ffmpeg -i input.webm -c:a copy -c:v vp9 -b:v 1M output.mkv
```

Creates a new Matroska with the audio stream copied over and the video stream's frame rate forced to 30 frames per second, instead of using the frame rate from the input (-r 30).

```
ffmpeg -i input.webm -c:a copy -c:v vp9 -r 30 output.mkv
```

Change the video resolution to 1280x720 in the output.

```
ffmpeg -i input.mkv -c:a copy -s hd720 output.mkv
ffmpeg -i input.mkv -c:a copy -s 1280x720 output.mkv
```

Copy the video and audio streams (-c:av copy) but trim the video. The -t option sets the cut duration to be 10 seconds and the -ss option sets the start point of the video for trimming, in this case at one minute (00:01:00).

```
ffmpeg -i input.mkv -c:av copy -ss 00:01:00 -t 10 output.mkv
```

Extract video (ignore audio and subtitles stream)

```
ffmpeg -i input.mkv -an -sn audio_only.ogg
```

Copy the audio (-c:a copy) from input.webm and convert the video to a VP9 codec (-c:v vp9) with a bit rate of 1M/s (-b:v), all bundled up in a Matroska container (output.mkv).

```
ffmpeg -i input.webm -c:a copy -c:v vp9 -b:v 1M output.mkv
```