

Linux Commands Cheat Sheet

1 – SYSTEM INFORMATION

Display Linux system information
uname -a

Display kernel release information
uname -r

Show which version of redhat installed
cat /etc/redhat-release

Show how long the system has been running + load
uptime

Show system host name
hostname

Display the 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

2 – HARDWARE INFORMATION

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
```

3 – 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
```

4 – 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
```

```
# Create a group named "test".
groupadd test
```

```
# Create an account named john, with a comment of "John Smith" and
create the user's home directory.
useradd -c "John Smith" -m john
```

```
# Delete the john account.
userdel john
```

```
# Add the john account to the sales group
usermod -aG sales john
```

5 – FILE AND DIRECTORY COMMANDS

```
# List all files in a long listing (detailed) format
ls -al
```

```
# 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
```

6 – 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 and manage the top processes
top
```

```

# Interactive process viewer (top alternative)
htop

# Kill process with process ID of pid
kill pid

# Kill all processes named processname
killall processname

# Start program in the background
program &

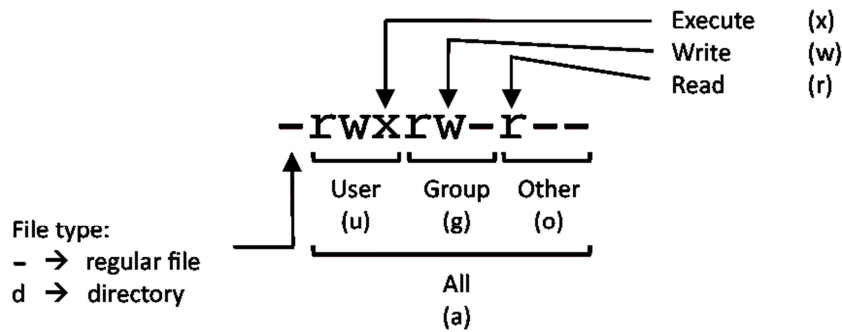
# Display stopped or background jobs
bg

# Brings the most recent background job to foreground
fg

# Brings job n to the foreground
fg n

```

7 – FILE PERMISSIONS



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

NOTE: Use 777 sparingly!

LEGEND
U = User
G = Group
W = World

r = Read
w = write
x = execute
- = no access

8 – NETWORKING

```
# Display all network interfaces and ip address
ifconfig -a

# Display eth0 address and details
ifconfig eth0

# Query or control network driver and hardware settings
ethtool eth0

# Send ICMP echo request to host
ping 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
hostname -I

# Download http://domain.com/file
wget http://domain.com/file

# Display listening tcp and udp ports and corresponding programs
netstat -nutlp
```

9 – 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
```

10 – INSTALLING PACKAGES

```
# Search for a package by keyword.
yum search keyword
```

```
# Install package.
yum install package
```

```
# Display description and summary information about package.
yum info package
```

```
# Install package from local file named package.rpm
rpm -i package.rpm
```

```
# Remove/uninstall package.
yum remove package
```

```
# Install software from source code.
tar zxvf sourcecode.tar.gz
cd sourcecode
./configure
make
make install
```

11 – 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
```

12 – SSH LOGINS

```
# Connect to host as your local username.
ssh host
```

```
# Connect to host as user
ssh user@host
```

```
# Connect to host using port
ssh -p port user@host
```

13 – FILE TRANSFERS

```
# Secure copy file.txt to the /tmp folder on server
scp file.txt server:/tmp
```

```
# Copy *.html files from server to the local /tmp folder.
scp server:/var/www/*.html /tmp
```

```
# Copy all files and directories recursively from server to the
current system's /tmp folder.
scp -r server:/var/www /tmp
```

```
# Synchronize /home to /backups/home
rsync -a /home /backups/
```

```
# Synchronize files/directories between the local and remote system
with compression enabled
rsync -avz /home server:/backups/
```

14 – DISK USAGE

```
# Show free and used space on mounted filesystems
df -h
```

```
# Show free and used inodes on mounted filesystems
df -i
```

```
# Display disks partitions sizes and types
fdisk -l
```

```
# Display disk usage for all files and directories in human readable
format
du -ah
```

```
# Display total disk usage off the current directory
du -sh
```

15 – DIRECTORY NAVIGATION

```
# To go up one level of the directory tree. (Change into the parent
directory.)
cd ..
```

```
# Go to the $HOME directory
cd
```

```
# Change to the /etc directory
cd /etc
```


Vim Cheat Sheet

Global

:help keyword - open help for keyword
:o file - open file
:saveas file - save file as
:close - close current window

Cursor Movements

h - move cursor left
j - move cursor down
k - move cursor up
l - move cursor right
H - move to top of screen
M - move to middle of screen
L - move to bottom of screen
w - jump forwards to the start of a word
W - jump forwards to the start of a word (words can contain punctuation)
e - jump forwards to the end of a word
E - jump forwards to the end of a word (words can contain punctuation)
b - jump backwards to the start of a word
B - jump backwards to the start of a word (words can contain punctuation)
0 - jump to the start of the line
^ - jump to the first non-blank character of the line
\$ - jump to the end of the line
g_ - jump to the last non-blank character of the line
gg - go to the first line of the document
G - go to the last line of the document
5G - go to line 5
fx - jump to next occurrence of character x
tx - jump to before next occurrence of character x
} - jump to next paragraph (or function/block, when editing code)
{ - jump to previous paragraph (or function/block, when editing code)
zz - center cursor on screen
Ctrl + b - move back one full screen
Ctrl + f - move forward one full screen
Ctrl + d - move forward 1/2 a screen
Ctrl + u - move back 1/2 a screen

Tip: Prefix a cursor movement command with a number to repeat it. For example, 4j moves down 4 lines.

Insert Mode

i - insert before the cursor
I - insert at the beginning of the line
a - insert (append) after the cursor
A - insert (append) at the end of the line
o - append (open) a new line below the current line
O - append (open) a new line above the current line
ea - insert (append) at the end of the word
Esc - exit insert mode

Editing

r - replace a single character
J - join line below to the current line
cc - change (replace) entire line
cw - change (replace) to the end of the word
c\$ - change (replace) to the end of the line
s - delete character and substitute text
S - delete line and substitute text (same as cc)
xp - transpose two letters (delete and paste)
u - undo
Ctrl + r - redo
. - repeat last command

Marking Text (Visual Mode)

v - start visual mode, mark lines, then perform an operation (such as d-delete)
V - start linewise visual mode
Ctrl + v - start blockwise visual mode
o - move to the other end of marked area
O - move to other corner of block
aw - mark a word
ab - a block with ()
aB - a block with {}
ib - inner block with ()
iB - inner block with {}
Esc - exit visual mode

Visual Commands

> - shift text right
< - shift text left
y - yank (copy) marked text
d - delete marked text
~ - switch case

Registers

```
:reg - show registers content  
"xy - yank into register x  
"xp - paste contents of register x
```

Tip: Registers are being stored in `~/.viminfo`, and will be loaded again on next restart of vim.

Tip: Register 0 contains always the value of the last yank command.

Marks

```
:marks - list of marks  
ma - set current position for mark A  
`a - jump to position of mark A  
y`a - yank text to position of mark A
```

Macros

```
qa - record macro a  
q - stop recording macro  
@a - run macro a  
@@ - rerun last run macro
```

Cut and Paste

```
yy - yank (copy) a line  
2yy - yank (copy) 2 lines  
yw - yank (copy) the characters of the word from the cursor position  
to the start of the next word  
y$ - yank (copy) to end of line  
p - put (paste) the clipboard after cursor  
P - put (paste) before cursor  
dd - delete (cut) a line  
2dd - delete (cut) 2 lines  
dw - delete (cut) the characters of the word from the cursor position  
to the start of the next word  
D - delete (cut) to the end of the line  
d$ - delete (cut) to the end of the line  
x - delete (cut) character
```

Exiting

```
:w - write (save) the file, but don't exit  
:w !sudo tee % - write out the current file using sudo  
:wq or :x or ZZ - write (save) and quit  
:q - quit (fails if there are unsaved changes)  
:q! or ZQ - quit and throw away unsaved changes
```

Search and Replace

/pattern - search for pattern
?pattern - search backward for pattern
\vpattern - 'very magic' pattern: non-alphanumeric characters are interpreted as special regex symbols (no escaping needed)
n - repeat search in same direction
N - repeat search in opposite direction
:%s/old/new/g - replace all old with new throughout file
:%s/old/new/gc - replace all old with new throughout file with confirmations
:noh - remove highlighting of search matches

Search in Multiple Files

:vimgrep /pattern/ {file} - search for pattern in multiple files
e.g.
:vimgrep /foo/ **/*
:cn - jump to the next match
:cp - jump to the previous match
:copen - open a window containing the list of matches

Working With Multiple Files

:e file - edit a file in a new buffer
:bnext or :bn - go to the next buffer
:bprev or :bp - go to the previous buffer
:bd - delete a buffer (close a file)
:ls - list all open buffers
:sp file - open a file in a new buffer and split window
:vsp file - open a file in a new buffer and vertically split window
Ctrl + ws - split window
Ctrl + ww - switch windows
Ctrl + wq - quit a window
Ctrl + wv - split window vertically
Ctrl + wh - move cursor to the left window (vertical split)
Ctrl + wl - move cursor to the right window (vertical split)
Ctrl + wj - move cursor to the window below (horizontal split)
Ctrl + wk - move cursor to the window above (horizontal split)

Tabs

:tabnew or :tabnew file - open a file in a new tab
Ctrl + wT - move the current split window into its own tab
gt or :tabnext or :tabn - move to the next tab
gT or :tabprev or :tabp - move to the previous tab
#gt - move to tab number #
:tabmove # - move current tab to the #th position (indexed from 0)
:tabclose or :tabc - close the current tab and all its windows
:tabonly or :tabo - close all tabs except for the current one
:tabdo command - run the command on all tabs (e.g. :tabdo q - closes all opened tabs)

