Linux Terminal

Tips and Tricks

Keyboard shortcuts

Tab	Auto-completion of file or command
Ctrl+U	Delete all contents to the left of the cursor
Ctrl+D	Logout of the current session. It is similar to exit
Ctrl+L	Clear the terminal
Ctrl+E	Cursor the end of line
Ctrl+K	Delete right of the cursor
Ctrl+W	Delete word on the left
Ctrl+R	Reverse search history. Type to bring up recent command
Ctrl+C	Halt the current command
Ctrl+Z	Stop current cmd and run in the bkgd (with **bg** or **fg**)
Ctrl+Shift+C/V (also Insert)	Copy and paste
!!	Repeat last command

General commands

BASIC	
uname -a/-r	Show system/kernel information
head -n1 /etc/issue	Show distribution
mount	Show mounted filesystems
hostname	Show system host name (**-I** = all local IP addresses)
last reboot	Show system reboot history
date/uptime/cal	Show system date, uptime, calendar
who	Show who is logged into the system
whoami	Print the current user id and name ('id -un')
w/whois	Display who is online
man <cmd>	Show manual for command
HARDWARE INFORMATION	
dmesg	Display messages in kernel ring buffer
fdisk -l	Display disks partitions sizes and types
df -h/-H	Give free hard disk space (**-h** = all mounted devices)
du -h	Show disk usage of folders in humanly readable format (**-h** = humanly readable; **-a** = all files; **-s** = total disk usage off)
free -h	Show free and used memory (**-h** = human readable, **-m** = MB, **-g** = GB)
cat /proc/cpuinfo(meminfo)	Display CPU/memory information
lspci -tv	Display PCI devices
lsusb -tv	Display USB devices
dmidecode	Display DMI/SMBIOS from the BIOS
hdparm -i /dev/sda	Show info about disk sda
hdparm -tT /dev/sda	Perform a read speed test on disk sda
badblocks -s /dev/sda	Test for unreadable blocks on disk sda
PERFORMANCE MONITORING AND STATISTICS	
top	Display and manage the top processes
htop	Interactive process viewer (top alternative)
ps **ps -aux | more/less**	Show snapshot of processes (**-aux** = show all users, and not connect processes. With PID, give **cd /proc/<pid>**)
pidof	Give the PID of a process
nice	Start a process with a given priority (**renice** = change pri.)
kill <PID>	Kill process with id PID
pkill <name>	Kill process with name
killall <name>	Kill all processes with names beginning <name>
mpstat 1	Display processor related statistics
vmstat 1	Display virtual memory statistics
iostat 1	Display I/O statistics
tail -100 /var/log/messages	Display the last 100 syslog messages (Use /var/log/syslog for Debian based systems.)
tcpdump -i eth0	Capture and display all packets on interface eth0
tcpdump -i eth0 'port 80'	Monitor all traffic on port 80 (HTTP)
lsof	List all open files on the system (**-u <user>**= by user)
watch df -h	Execute "df -h", showing periodic updates
watch -n 5 'ntpq -p'	Issue the 'ntpq -p' command every 5s and display output
USER INFORMATION AND MANAGEMENT	
id	Display the user and group ids of your current user
last	Display the last users who have logged onto the system
finger	Show information of all the users logged in (<username> = of a particular user)
groupadd test	Create a group named "test"
useradd -c "John Smith" -m john	Create an account named john, with a comment of "John Smith" and create the user's home directory
userdel john	Delete the john account
usermod -aG sales john	Add the john account to the sales group
FILE AND DIRECTORY	
ls	List all files and directories
ls -R	List files in sub-directories as well
ls -la	List files and directories with detailed information
pwd	Print Working Directory
mkdir <dir>	Make Directory - Create new folder(s)
cd <dir> **cd $HOME** **cd ~** **cd**	Change Directory. The following commands are similar and return to the same place: /home/user (tilda expensions)
rm <file> **rm -f <file>**	Remove files (**-f** = force removal)
rm -r <dir>	Remove the directory and its contents recursively
rmdir <folder> **rm -r <dir>**	Remove folders (**-rf** = forcefully remove directory recursively)
cp <file1> <file2>	Copy file1 to file2
cp -r <source> <dest.>	Copy source directory recursively to destination
mv <file1> <file2> **mv <file1> <dir>**	Rename or move file1 to file2. If file2 is an existing directory, move file1 into directory file2
rename <file> <new name>	Rename files
ln -s /path/to/file <link>	Create symbolic link to link name
touch <file>	Create an empty file or update the access and modification times of file
source <file>	Read and execute the content of a file
cat <file> **cat > <file>** **cat <file1> <file2> > <file3>**	Concatenate (for small files). View the contents of file (**>** = create a new file)
more <file>	Show text files or other output in a scrollable manner
less <file>	Browse through a text file
head <file>	Display the first 10 lines of file
tail <file> **tail -f <file>**	Display the last 10 lines of file (-f = "follow" the file as it grows
chmod +x <file>	Change file mode bits. Make it executable
chown <user> <file>	Change file owner (**<user>:<group>** = change group)
nautilus <dir>/.	Open directory in GNOME file explorer
echo <msg> > <file>	Write msg to file (**>>** = if you don't want to overwrite)
awk	command-line text manipulation dynamo, as well as a powerful scripting language

Advanced commands

BASH VARIABLES	
env	Show environment variables
printenv | less/more **set | less**	Print environment variables
echo $NAME	Output value of $NAME variable
export NAME=value	Set $NAME to vale
I/O REDIRECTION	
<cmd> < <file>	Input of command from file
<cmd1> < <cmd2>	Output of cmd2 as file input to cmd1
<cmd> > <file>	Standard output (**stdout**) of cmd to file
<cmd> >> <file>	Append stdout to file
<cmd> > /dev/null	Discard stdout of cmd
<cmd> 2> <file>	Error output (**stderr**) of cmd to file
<cmd> 1>&2	stdout to same place as stderr
<cmd> 2>&1	stderr to same place as stdout
<cmd> &> <file>	Every output of cmd to file
PIPES	
<cmd1> | <cmd2>	Stdout of cmd1 to cmd2 standard input (**stdind**)
<cmd1> |& <cmd2>	Stderr of cmd1 to cmd2
COMMAND LISTS	
<cmd1> ; <cmd2>	Run cmd1 then cmd2
<cmd1> && <cmd2>	Run cmd2 if cmd1 is successful
<cmd1> || <cmd2>	Run cmd2 if cmd1 is not successful
<cmd1> & <cmd2>	Run cmd in a subshell
SEARCH FILES	
which <cmd>	Identify the location of executables
which <pattern> <files>	Search for patterns in files (**-i** = case insensitive; **-r** = recursive; **-v** = inverted; **-o** = show matched part of file)
ag <pattern> <files>	Searching tool similar to ack, with a focus on speed.
find <dir> -name <name>*	Find files starting with name in directory
find <dir> -user <name>	Find files owned by name in directory
find <dir> -mnin <num>	Find files modified less than num minutes ago in directory
whereis <cmd>	Find binary /source/ manual for command
locate <file>	Find file (quick search of system index)
NETWORK	
ifconfig	Display and manipulate route and network interfaces
ip **ip addr show dev <nic>**	It is a replacement of ifconfig command
ethtool <nic>	Query or control network driver and hardware settings
traceroute	Network troubleshooting utility
tracepath	Similar to traceroute but doesn't require root privileges
ping	To check connectivity between two nodes
tcpdump	Capture and display packets
netstat	Display connection information
ss	It is a replacement of netstat
dig	Query DNS related information
nslookup	Find DNS related query
route	Shows and manipulate IP routing table
host	Performs DNS lookups
arp	View or add contents of the kernel's ARP table
iwconfig	Used to configure wireless network interface
curl/wget	To download a file from internet
mtr	Combines ping and tracepath into a single command
ifplug	Tells whether a cable is plugged in or not
hostname and whois	It is also used for identify a network and website's whois

Packages

DPKG	
dpkg -i <deb.pkg>	Install package (or upgrade if it is installed)
dpkg -R <dir>	Install all package recursively from directory
dpkg -r <pkg>	Remove/Delete an installed package except config. files
dpkg -P <pkg>	Uninstall package, including conf files
dpkg -p <pkg>	Display details about package group, version, etc.
dpkg -l	List all installed packages (**| less**, **'*apache*'**, **| grep -i 'sudo'**, **<pkg>**)
dpkg -L <pkg>	Find out files are provided by the installed package
dpkg -c <deb.pkg>	List files provided (or owned) by the package
dpkg -s <pkg>	Check whether a package file has been installed before (**| grep Status =** is installed or not)
dpkg -S <file>	Find what package owns the file
APT	
apt-cache policy	List all repositories
apt-cache pkgnames	List all packages
apt-cache rdepends <pkg>	Show package reverse dependencies
apt search <pkg>	Search package (also **apt-cache**)
apt list --installed	List installed packages (also **--upgradeable**. Used with **| grep <pkg>**)
apt list	
apt install <pkg>	Install package (**--reinstall**)
apt remove <pkg>	Remove package
apt --purge remove <name>	Remove package and configuration (or just **apt purge**)
apt autoremove	Remove unused packages
apt --purge autoremove	Remove orphan packages
apt update	Check for updates
apt upgrade	Upgrade all packages (see **full-upgrade**)
apt update && apt-get -y upgrade -y	Run this to upgrade all packages
apt show <pkg>	Show package records (also **apt-cache**)
apt-mark unhold <pkg>	Prevent a package from being updated (Hold)
apt clean	Clean the local repository cache of retrieved packages
apt autoclean	Clean retrieved packages that have newer version now
ASDF	
asdf plugin add <n> <url>	Add a plugin from the plugin repo (**<name> <git-url>**)
asdf plugin list [--urls] [--refs]	List installed plugins (**all** = registered on asdf-plugins repository with URLs)
asdf plugin remove <name>	Remove plugin and package versions
asdf plugin update <name> <git-ref>	Update a plugin to latest commit on default branch or a particular git-ref (**--all** = update all plugins)
asdf install (<n> <v> latest:)	Install all the package versions listed in .tool-versions file
asdf uninstall <name> <v>	Remove a specific version of a package
asdf current <name>	Display current version set or being used for package
asdf where <name> <v>	Display install path for an installed or current version
asdf which <cmd>	Display the path to an executable
asdf local/global <name> <v>	Set the package local/global version (**latest:<v>**= to the latest provided version)
asdf shell <name> <v>	Set the package version to `ASDF_${LANG}_VERSION`
asdf latest <name> <v>	Show latest stable version of a package (**--all**)
asdf list <name> <v>	List installed versions of a package (**all**)
asdf help <name> <v>	Output documentation for plugin and tool
asdf exec <cmd> [args...]	Executes the command shim for current version
asdf env <cmd> [util]	Run util (default: `env`) inside the env used for cmd shim
asdf info	Print OS, Shell and ASDF debug information
asdf reshim <name> <v>	Recreate shims for version of a package
asdf shim-versions <cmd>	List the plugins and versions that provide a command
asdf update	Update asdf to the latest stable (**--head** = master branch)

SUDO commands

sudo su -	Become root with root env
sudo -n <user>	Ensure to be non-interactive
sudo -i <user>	Sudo and run login shells of the user
sudo -l -U <user>	List sudo permissions of given user
sudo -k	Force password on next sudo
sudo visudo	Safely change /etc/sudoers file (also **/usr/sbin/visudo**)
sudo visudo -f /etc/sudoers.d/<file>	Edit sudoers.d file
sudo pkexex visudo	Edit broken /etc/sudoers files (**-f /etc/sudoers.d/<file>**)
sudo adduser <username>	Add a new user
sudo passwd -l <username>	Change the password of a user
sudo userdel -r <username>	Remove a newly created user
sudo usermod -a -G <group> <username>	Add a user to a group
sudo deluser <username> <group>	Remove a user from group
sudo service <service> start/stop/status	/etc/init.d
sudo systemctl start/stop/status <service>	
/etc/sudoers SYNTAX	
User_name Machine_name=(Effective_user) command	
ALL=(ALL:ALL) ALL	Allow group 'sudo' with password
ALL=(ALL:ALL) NOPASSWD: ALL	Allow group 'sudo' without password
jonathan ALL=(ALL) ALL	Allow user 'jonathan' with password

SSH commands

CONNECTING	
ssh user@host	Login into a remote Linux machine using SSH
ssh -p <port> user@host	Connect to host using port
ssh -i /path/file.pem user@host	Connect via pem file (0400 permissions)
ssh host	Connect to host as your local username
EXECUTING	
ssh user@host '<cmd>'	Execute remote command
ssh user@host bash < script.sh	Invoke a local script
ssh user@host "tar cvzf - ~/source" > output.tgz	Compress and download from a server
CONFIG LOCATION	
/etc/ssh/ssh_config	System-wide config
~/.ssh/config	User-specific config
~/.ssh/id_{type}	Private key
~/.ssh/id_{type}.pub	Public key
~/.ssh/known_hosts	Logged in host
~/.ssh/authorized_keys	Authorized login key
CONFIG SAMPLE	
Launch by alias: **ssh server1**	Host **server1** HostName 192.168.1.5 User root Port 22 IdentityFile ~/.ssh/server1.key
OTHER	
ssh -J proxy_host1 remote_host2	Proxy Jump (**user@proxy_host1 user@remote_host2**)
user@proxy_host1:port1, user@proxy_host2:port2, user@remote_host3	Multiple jumps
ssh-copy-id user@host	Copy ssh keys to servers
put <file>	Upload 'file' from local to remote computer
get <file>	Download 'file' from remote to local computer
quit	Logout

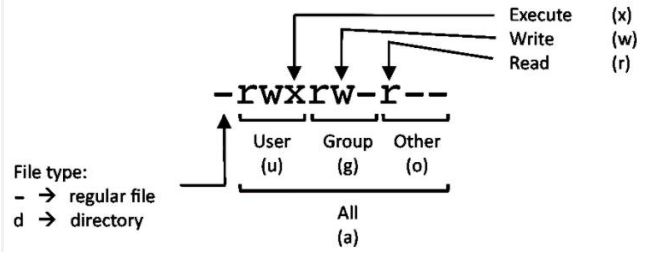
SSH KeyGen

ssh-keygen	Generate a key interactively
ssh-keygen -t rsa -b 4096 -C "<email>"	**-t** = Type of key, **-b** = The number of bits in the key, **-C** = Provides a new comment
ssh-keygen -f ~/.ssh/name	Specify filename
ssh-keygen -y -f private.key > public.pub	Generate public key from private key
ssh-keygen -c -f <ssh-key>	Change comment
ssh-keygen -p -f <ssh-key>	Change private key passphrase
ssh-keygen -F <ip/host_n>	Search from known_hosts (**-R** = remove host)

File transfer

scp file.txt server:/tmp	Secure copy file.txt to the /tmp folder on server
scp server:/var/www/*.html /tmp	Copy *.html files from server to the local /tmp folder
scp -r server:/var/www /tmp	Copy all files and directories recursively from server to the current system's /tmp folder.
rsync -a /home /backups/	Synchronize /home to /backups/home
rsync -avz /home server:/backups/	Synchronize files/directories between the local and remote system with compression enabled
SSH CONNECTION	
scp user@server:/dir/file.ext dest/	Copy from remote to local
scp user@server:/file user@server:/dir	Copy between two servers
scp dest/file.ext user@server:/dir	Copy from local to remote
scp -r user@server:/dir dest/	Copy a whole folder
scp user@server:/dir/* dest/	Copy all files from a folder
scp user@server:/dir/* .	Copy from a server folder to the current folder

File permissions



	U	G	W
chmod 777 <file>	rwx	rwx	rwx
chmod 775 <file>	rwx	rwx	r-x
chmod 755 <file>	rwx	r-x	r-x
chmod 664 <file>	rwx	rw-	r--
chmod 644 <file>	rw-	r--	r--