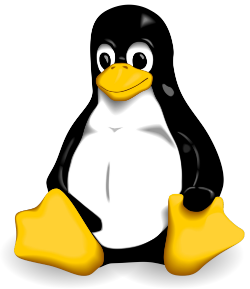


Linux
Quick Reference Guide

8th edition January 2020

Foreword

This guide stems from the notes I have been taking while studying and working as a Linux sysadmin and engineer.
It contains useful information about standards and tools for Linux system administration, as well as a good amount of topics from the certification exams LPIC-1 (Linux Professional Institute Certification level 1), LPIC-2, RHCSA (Red Hat Certified System Administrator), and RHCE (Red Hat Certified Engineer). Unless otherwise specified, the shell of reference is Bash.

This is an independent publication and is not affiliated with LPI or Red Hat. You can freely use and share the whole guide or the single pages, provided that you distribute them unmodified and not for profit.
This document has been composed with Apache OpenOffice.

Happy Linux hacking,

Daniele Raffo

Version history

1st edition	May 2013
2nd edition	September 2014
3rd edition	July 2015
4th edition	June 2016
5th edition	September 2017
6th edition	August 2018
7th edition	May 2019
8th edition	January 2020

Bibliography and suggested readings

- Evi Nemeth et al., *UNIX and Linux System Administration Handbook*, O'Reilly
- Rebecca Thomas et al., *Advanced Programmer's Guide to Unix System V*, McGraw-Hill
- Mendel Cooper, *Advanced Bash-Scripting Guide*, http://tldp.org/LDP/abs/html
- Adam Haeder et al., *LPI Linux Certification in a Nutshell*, O'Reilly
- Heinrich W. Klöpping et al., *The LPIC-2 Exam Prep*, http://lpic2.unix.nl
- Michael Jang, *RHCSA/RHCE Red Hat Linux Certification Study Guide*, McGraw-Hill
- Asghar Ghori, *RHCSA & RHCE RHEL 7: Training and Exam Preparation Guide*, Lightning Source Inc.
- Colin Barschel, *Unix Toolbox*, http://cb.vu/unixtoolbox.xhtml
- Ellen Siever et al., *Linux in a Nutshell*, O'Reilly, http://archive.oreilly.com/linux/cmd
- Christoph Braun, *Unix System Security Essentials*, Addison-Wesley
- Bruce Barnett, *The Grymoire*, http://www.grymoire.com/Unix
- Brendan Gregg, *Linux performance*, http://www.brendangregg.com/linuxperf.html
- Linus Torvalds' Linux documentation, https://github.com/torvalds/linux/tree/master/Documentation
- RHEL manuals, https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux
- A-Z index of Bash command line, http://ss64.com/bash
- GNU software manuals, http://www.gnu.org/manual
- Shell command line snippets, http://www.commandlinefu.com
- Bash command line snippets, http://www.bashoneliners.com
- RAM management in Linux, http://www.linuxatemyram.com
- Regular expressions tester, http://www.regextester.com
- Bash pitfalls, http://mywiki.wooledge.org/BashPitfalls
- Linux man pages, https://www.kernel.org/doc/man-pages
- CentOS 7 man pages, https://www.unix.com/man-page-centos-repository.php

Index

Logical Volume Management (LVM) introduces an abstraction between physical and logical storage, allowing a more versatile use of filesystems. LVM uses the Linux device mapper feature (`/dev/mapper`).

Disks, partitions, and RAID devices are made of **Physical Volumes**, which are grouped into a **Volume Group**.
A Volume Group is divided into small fixed-size chunks called Physical Extents, which are mapped 1-to-1 to Logical Extents.
Logical Extents are grouped into **Logical Volumes**, on which filesystems are created.

How to create a Logical Volume

1.	Add a new disk to the machine	
2.	`lsblk`	Verify that the new disk is recognized e.g. as `/dev/sda`
3.	`fdisk /dev/sda`	Create a new partition (of type 0x8E = Linux LVM) on the new disk. This is not necessary but recommended, because other OSes might not recognize the LVM header and see the whole unpartitioned disk as empty
4.	`pvcreate /dev/sda1`	Initialize the Physical Volume to be used with LVM
5.	`vgcreate -s 8M myvg0 /dev/sda1`	Create a Volume Group and define the size of Physical Extents to 8 Mb (default value is 4 Mb)
or	`vgextend myvg0 /dev/sda1`	or add the Physical Volume to an existing Volume Group
6.	`lvcreate -L 1024M -n mylv myvg0`	Create a Logical Volume
7.	`mkfs -t ext3 /dev/myvg0/mylv`	Create a filesystem on the Logical Volume
8.	`mount /dev/myvg0/mylv /mnt/mystuff`	Mount the Logical Volume

How to increase the size of a Logical Volume (operation possible only if the underlying filesystem allows it)

1.	Add a new disk to the machine, to provide the extra disk space	
2.	`pvcreate /dev/sdc`	Initialize the Physical Volume
3.	`vgextend myvg0 /dev/sdc`	Add the Physical Volume to an existing Volume Group
or		
1.	Increase the size of an existing disk (already initialized as PV)	
2.	`partprobe`	Notify the kernel of the new disk size
3.	`pvresize /dev/sdc`	Accommodate the Physical Volume to the new size
Then:		
4.	`lvextend -L 2048M /dev/myvg0/mylv`	Extend the Logical Volume by 2 Gb
or	`lvresize -L+2048M /dev/myvg0/mylv`	
or	`lvresize -l+100%FREE /dev/myvg/mylv`	or extend the Logical Volume taking all free space
5.	`resize2fs /dev/myvg0/mylv` (ext) `xfs_growfs /dev/myvg0/mylv` (XFS)	Extend the filesystem. Alternatively, use `lvresize -r` on the previous step

How to reduce the size of a Logical Volume (operation possible only if the underlying filesystem allows it)

1.	`resize2fs /dev/myvg0/mylv 900M`	Shrink the filesystem to 900 Mb
2.	`lvreduce -L 900M /dev/myvg0/mylv`	Shrink the Logical Volume to 900 Mb
or	`lvresize -L 900M /dev/myvg0/mylv`	

How to snapshot and backup a Logical Volume

1.	`lvcreate -s -L 1024M -n mysnap /dev/myvg0/mylv`	Create the snapshot like a Logical Volume
2.	`tar cvzf mysnap.tar.gz mysnap`	Backup the snapshot with any backup tool
3.	`lvremove /dev/mvvg0/mysnap`	Delete the snapshot

LVM commands

PV commands		VG commands		LV commands	
pvs	Report information about Physical Volumes	vgs	Report information about Volume Groups	lvs	Report information about Logical Volumes
pvscan	Scan all disks for Physical Volumes	vgscan	Scan all disks for Volume Groups	lvscan	Scan all disks for Logical Volumes
pvdisplay	Display Physical Volume attributes	vgdisplay	Display Volume Group attributes	lvdisplay	Display Logical Volume attributes
pvck	Check Physical Volume metadata	vgck	Check Volume Group metadata		
pvcreate	Initialize a disk or partition for use with LVM	vgcreate	Create a Volume Group using Physical Volumes	lvcreate	Create a Logical Volume in a Volume Group
pvchange	Change Physical Volume attributes	vgchange	Change Volume Group attributes	lvchange	Change Logical Volume attributes
pvremove	Remove a Physical Volume	vgremove	Remove a Volume Group	lvremove	Remove a Logical Volume
		vgextend	Add a Physical Volume to a Volume Group	lvextend	Increase the size of a Logical Volume
		vgreduce	Remove a Physical Volume from a Volume Group	lvreduce	Shrink the size a Logical Volume
pvresize	Modify the size of a Physical Volume			lvresize	Modify the size of a Logical Volume
		vgmerge	Merge two Volume Groups		
		vgsplit	Split two Volume Groups		
		vgimport	Import a Volume Group into a system		
		vgexport	Export a Volume Group from a system		
pvmove	Move the Logical Extents on a Physical Volume to wherever there are available Physical Extents (within the Volume Group) and then put the Physical Volume offline				

LVM global commands	
dmsetup *command*	Perform low-level LVM operations
lvm *command*	Perform LVM operations. May also be used as an interactive tool
lvmsar	LVM system activity reporter. Unsupported on LVM2
lvmdiskscan	Scan the system for disks and partitions usable by LVM
lvmconfig	Show the current LVM disk configuration

/dev/mapper/*vgname*-*lvname* /dev/*vgname*/*lvname*	Mapping of Logical Volumes in the filesystem
/etc/lvm/archive/	Directory containing Volume Groups metadata backups

Boot sequence	
POST (Power-On Self Test)	Low-level check of PC hardware.
BIOS (Basic I/O System)	Detection of disks and hardware.
Chain loader GRUB (GRand Unified Bootloader)	GRUB stage 1 is loaded from the MBR and executes GRUB stage 2 from filesystem. GRUB chooses which OS to boot on. The chain loader hands over to the boot sector of the partition on which resides the OS. The chain loader also mounts `initrd`, an initial ramdisk (typically a compressed ext2 filesystem) to be used as the initial root device during kernel boot; this make possible to load kernel modules that recognize hard drives hardware and that are hence needed to mount the real root filesystem. Afterwards, the system runs `/linuxrc` with PID 1. (From Linux 2.6.13 onwards, the system instead loads into memory `initramfs`, a cpio-compressed image, and unpacks it into an instance of tmpfs in RAM. The kernel then executes `/init` from within the image.)
Linux kernel	Kernel decompression into memory. Kernel execution. Detection of devices. The real root filesystem is mounted on `/` in place of the initial ramdisk.

init	Execution of `init`, the first process (PID 1).
	The system tries to execute in the following order:
	`/sbin/init`
	`/etc/init`
	`/bin/init`
	`/bin/sh`
	If none of these succeeds, the kernel panics.

Startup	The system loads startup scripts and runlevel scripts.
Login	If in text mode, `init` calls the `getty` process, which runs the `login` command that asks the user for login and password. If in graphical mode, the X Display Manager starts the X Server.

Newer systems use UEFI (Unified Extensible Firmware Interface) instead of BIOS. UEFI does not use the MBR boot code; it has knowledge of partition table and filesystems, and stores its application files required for launch in a EFI System Partition, mostly formatted as FAT32.
After the POST, the system loads the UEFI firmware which initializes the hardware required for booting, then reads its Boot Manager data to determine which UEFI application to launch. The launched UEFI application may then launch another application, e.g. the kernel and `initramfs` in case of a boot loader like GRUB.
Information about the boot process can be found in the manpages `man 7 boot` and `man 7 bootup`.

Startup sequence	Debian	Red Hat
At startup `/sbin/init` executes all instructions on `/etc/inittab`. This script at first switches to the default runlevel...	`id:2:initdefault:`	`id:5:initdefault:`
... then it runs the following script (same for all runlevels) which configures peripheral hardware, applies kernel parameters, sets hostname, and provides disks initialization...	`/etc/init.d/rcS`	`/etc/rc.d/rc.sysinit` or `/etc/rc.sysinit`
... and then, for runlevel *N*, it calls the script `/etc/init.d/rc` *`N`* (i.e. with the runlevel number as parameter) which launches all services and daemons specified in the following startup directories:	`/etc/rc`*`N`*`.d/`	`/etc/rc.d/rc`*`N`*`.d/`

The startup directories contain symlinks to the init scripts in `/etc/init.d/` which are executed in numerical order. Links starting with K are called with argument `stop`, links starting with S are called with argument `start`.

```
lrwxrwxrwx.  1 root root   14 Feb 11 22:32 K88sssd -> ../init.d/sssd
lrwxrwxrwx.  1 root root   15 Nov 28 14:50 K89rdisc -> ../init.d/rdisc
lrwxrwxrwx.  1 root root   17 Nov 28 15:01 S01sysstat -> ../init.d/sysstat
lrwxrwxrwx.  1 root root   18 Nov 28 14:54 S05cgconfig -> ../init.d/cgconfig
lrwxrwxrwx.  1 root root   16 Nov 28 14:52 S07iscsid -> ../init.d/iscsid
lrwxrwxrwx.  1 root root   18 Nov 28 14:42 S08iptables -> ../init.d/iptables
```

The last script to be run is `S99local -> ../init.d/rc.local`; therefore, an easy way to run a specific program upon boot is to call it from this script file.

`/etc/init.d/boot.local`		runs only at boot time, not when switching runlevel.
`/etc/init.d/before.local`	(SUSE)	runs only at boot time, before the scripts in the startup directories.
`/etc/init.d/after.local`	(SUSE)	runs only at boot time, after the scripts in the startup directories.

To add or remove services at boot sequence:	`update-rc.d` *`service`* `defaults`
	`update-rc.d -f` *`service`* `remove`
		`chkconfig --add` *`service`*
		`chkconfig --del` *`service`*

When adding or removing a service at boot, startup directories will be updated by creating or deleting symlinks for the default runlevels: K symlinks for runlevels 0 1 6, and S symlinks for runlevels 2 3 4 5.
Service will be run via the `xinetd` super server.

Supported service operations		
`start`	Start the service	Mandatory
`stop`	Stop the service	
`restart`	Restart the service (stop, then start)	
`status`	Display daemon PID and execution status	
`force-reload`	Reload configuration if service supports it, otherwise restart	
`condrestart` `try-restart`	Restart the service only if already running	Optional
`reload`	Reload the service configuration	

Linux Standard Base (LSB)

The Linux Standard Base defines a format to specify default values on an init script `/etc/init.d/foo`:

```
### BEGIN INIT INFO
# Provides: foo
# Required-Start: bar
# Defalt-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Description: Service Foo init script
### END INIT INFO
```

Default runlevels and S/K symlinks values can also be specified as such:

```
# chkconfig: 2345 85 15
# description: Foo service
```

`/etc/init/start-ttys.conf` (Red Hat)	Start the specified number of terminals at bootup via `getty`, which manages physical or virtual terminals (TTYs)
`/etc/sysconfig/init` (Red Hat)	Control appearance and functioning of the system during bootup
`/etc/machine-id` (Red Hat)	Randomly-generated machine ID. The machine ID can be safely regenerated by deleting this file and then running the command `systemd-machine-id-setup`
`/etc/securetty`	List of TTYs from which the root user is allowed to login
`/etc/issue`	Message printed before the login prompt. Can contain these escape codes:

`\b`	Baudrate of line	`\o`	Domain name
`\d`	Date	`\r`	OS release number
`\s`	System name and OS	`\t`	Time
`\l`	Terminal device line	`\u`	Number of users logged in
`\m`	Machine architecture identifier	`\U`	"*n* users" logged in
`\n`	Nodename aka hostname	`\v`	OS version and build date

`/etc/issue.net`	Message printed before the login prompt on a remote session
`/etc/motd`	Message Of The Day, printed after a successful login, but before execution of the login shell
`/etc/nologin`	If this file exists, `login` and `sshd` deny login to all unprivileged users. Useful when doing system maintenance

`/var/log/secure` (Red Hat) `/var/log/auth.log` (Debian)	Logfile containing user logins (both successful and failed) and authentication mechanisms
`/var/log/pwdfail`	Logfile containing failed authentication attempts

To prevent a specific user to log in, their shell can be set either as:

`/bin/false`	user is forced to exit immediately
`/sbin/nologin`	user is prompted a message and forced to exit; the message is "This account is currently not available" or the contents of file `/etc/nologin.txt` if it exists

`who`	Print the list of users logged into the system
`w`	Print the list of users logged into the system, and what they are doing
`last`	Print the list of users that logged in and out. Searches through the file `/var/log/wtmp`
`lastb`	Print the list of bad login attempts. Searches through the file `/var/log/btmp`
`fail2ban`	Temporarily ban IP addresses (via firewall rules) that have too many failed password logins. This information is taken from authentication logs
`pam_tally2`	Deny access to users that have too many failed logins

`acct on`
`acct off`
	Turn process accounting on or off

`ac`	Print statistics about connect time of users
`lastcomm`	Print information about previously executed commands
`sa`	Print summarized information about previously executed commands

	Runlevel (SysV)	Target (Systemd)	Debian	Red Hat
	0		Shutdown	
	1		Single user / maintenance mode	
default runlevels	**2**		Multi-user mode (default)	Multi-user mode without network
	3	**multi-user.target**	Multi-user mode	Multi-user mode with network
	4		Multi-user mode	Unused, for custom use
	5	**graphical.target**	Multi-user mode	Multi-user mode with network and X (default)
	6		Reboot	
	S		Single user / maintenance mode (usually accessed through runlevel 1)	

Systemd's target `runlevel`*`n`*`.target` emulates a SysV's runlevel *n*.

`runlevel`
`who -r`
	Display the previous and the current runlevel

`init` *`runlevel`*
`telinit` *`runlevel`*
	Change to *runlevel*

`systemctl get-default`	Get the default target
`systemctl set-default` *`target`*	Set *target* as the default target
`systemctl isolate` *`target`*	Change to *target*
`systemctl emergency`	Change to maintenance single-user mode with only `/root` filesystem mounted
`systemctl rescue`	Change to maintenance single-user mode with only local filesystems mounted
`systemctl -t target`	List targets

`init 0`
`telinit 0`
`shutdown -h now`
`halt`
`poweroff`
`systemctl isolate shutdown.target`
	Halt the system

`init 6`
`telinit 6`
`shutdown -r now`
`reboot`
`systemctl isolate reboot.target`
	Reboot the system

`shutdown`	Shut down the system in a secure way: all logged-in users are notified via a message to their terminal, and login is disabled. Can only be run by the root user
`shutdown -a`	Non-root users that are listed in `/etc/shutdown.allow` can use this command to shut down the system
`shutdown -h 16:00` *`message`*	Schedule a shutdown for 4 PM and send a warning message to all logged-in users
`shutdown -f`	Skip fsck on reboot
`shutdown -F`	Force fsck on reboot
`shutdown -c`	Cancel a shutdown that has been already initiated

SysV service management

/etc/init.d/*service operation* service *service operation* rc*service operation*	 (Red Hat) (SUSE)	Perform the specified operation (start, stop, status, etc.) on the specified service
update-rc.d *service* defaults chkconfig --add *service*	(Debian) (Red Hat)	Add a service at boot
update-rc.d -f *service* remove chkconfig --del *service*	(Debian) (Red Hat)	Remove a service at boot
update-rc.d -f *service* \\ start 30 2 3 4 5 . stop 70 0 1 6 .		Add a service on the default runlevels; creates S30 symlinks for starting the service and K70 symlinks for stopping it
chkconfig --levels 245 *service* on		Add the service on runlevels 2 4 5
chkconfig *service* on		Add the service on default runlevels
chkconfig *service* off		Remove the service on default runlevels
chkconfig *service*		Check if the service is enabled on the current runlevel
chkconfig *service* reset		Reset the on/off state of the service for all runlevels to whatever the LSB specifies in the init script
chkconfig *service* resetpriorities		Reset the start/stop priorities of the service for all runlevels to whatever the LSB specifies in the init script
chkconfig --list *service*		Display current configuration of service (its status and the runlevels in which it is active)
chkconfig chkconfig --list		List all active services and their current configuration
ls /etc/rc*n*.d	(Debian)	List services started on runlevel *n*

`systemctl` *operation service*	Perform the specified operation (`start`, `stop`, `status`, etc.) on the specified service (unit file)
`systemctl enable` *service*	Add the service on the current target
`systemctl disable` *service*	Remove the service on the current target
`systemctl is-enabled` *service*	Check if the service is enabled on the current target
`systemctl mask` *service*	Mask the service on the current target. This prevents the service to be enabled or started
`systemctl unmask` *service*	Unmask the service on the current target
`systemctl list-unit-files --type=service`	List all active services and their current configuration
`systemctl`	List loaded and active units
`systemctl --all`	List all units, including inactive ones

/etc/inittab

```
# The default runlevel.
id:2:initdefault:

# Boot-time system configuration/initialization script.
# This is run first except when booting in emergency (-b) mode.
si::sysinit:/etc/init.d/rcS

# What to do in single-user mode.
~~:S:wait:/sbin/sulogin

# /etc/init.d executes the S and K scripts upon change of runlevel.
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
# Normally not reached, but fall through in case of emergency.
z6:6:respawn:/sbin/sulogin

# /sbin/getty invocations for the runlevels.
# Id field must be the same as the last characters of the device (after "tty").
1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
```

`/etc/inittab` describes which processes are started at bootup and during normal operation; it is read and executed by `init` at bootup.
All its entries have the form ***`id:runlevels:action:process`***.

id	1-4 characters, uniquely identifies an entry. For gettys and other login processes it should be equal to the suffix of the corresponding tty	
runlevels	Runlevels for which the specified action must be performed. If empty, action is performed on all runlevels	
action	`respawn`	Process will be restarted when it terminates
	`wait`	Process is started at the specified runlevel and `init` will wait for its termination (i.e. execution of further lines of `/etc/inittab` stops until the process exits)
	`once`	Process is executed once at the specified runlevel
	`boot`	Process is executed at system boot. Runlevels field is ignored
	`bootwait`	Process is executed at system boot and `init` will wait for its termination. Runlevels field is ignored
	`off`	Does nothing
	`ondemand`	Process is executed when an on-demand runlevel (A, B, C) is called
	`initdefault`	Specifies the default runlevel to boot on. Process field is ignored
	`sysinit`	Process is executed at system boot, before any `boot` or `bootwait` entries. Runlevels field is ignored
	`powerfail`	Process is executed when power goes down and an UPS kicks in. `init` will not wait for its termination
	`powerwait`	Process is executed when power goes down and an UPS kicks in. `init` will wait for its termination
	`powerfailnow`	Process is executed when power is down and the UPS battery is almost empty
	`powerokwait`	Process is executed when power has been restored from UPS
	`ctrlaltdel`	Process is executed when `init` receives a SIGINT via CTRL ALT DEL
	`kbdrequest`	Process is executed when a special key combination is pressed on console
process	Process to execute. If prepended by a `+`, `utmp` and `wtmp` accounting will not be performed	

Filesystem Hierarchy Standard (FHS)	
/bin	Essential command binaries for all users
/boot	Bootloader files (OS loader, kernel image, initrd, etc.)
/dev	Virtual filesystem containing device nodes to devices and partitions
/etc	System configuration files and scripts
/home	Home directories for users
/lib	Libraries for the binaries in /bin and /sbin, kernel modules
/lost+found	Storage directory for recovered files in this partition
/media	Mount points for removable media
/mnt	Mount points for temporary filesystems
/net	Access to directory tree on different external NFS servers
/opt	Optional, large add-on application software packages
/proc	Virtual filesystem providing kernel and processes information
/root	Home directory for the root user
/run	Runtime variable data; replaces /var/run
/sbin	Essential system binaries, system administration commands
/srv	Data for services provided by the system
/sys	Virtual filesystem providing information about hotplug hardware devices
/tmp	Temporary files; deleted at reboot
/usr	User utilities and applications
/usr/bin	Non-essential command binaries for all users
/usr/include	C header files
/usr/lib	Libraries for the binaries in /usr/bin and /usr/sbin
/usr/local	Software installed locally
/usr/local/bin	Local software binaries
/usr/local/games	Local game binaries
/usr/local/include	Local C header files
/usr/local/lib	Local libraries for the binaries in /usr/local/bin and /usr/local/sbin
/usr/local/man	Local man pages
/usr/local/sbin	Local system binaries
/usr/local/share	Local architecture-independent hierarchy
/usr/local/src	Local source code
/usr/sbin	Non-essential system binaries (daemons and services)
/usr/share	Architecture-independent files (e.g. icons, fonts, documentation)
/usr/share/doc	Package-specific documentation not included in man pages
/usr/share/man	Man pages
/usr/share/info	Documentation in Info format
/usr/src	Source code for the current OS
/var	Variable files (e.g. logs, caches, mail spools)
/var/log	Logfiles
/var/opt	Variable files for the application software installed in /opt
/var/spool	Queued items to be processed (e.g. mail messages, cron jobs, print jobs)
/var/tmp	Temporary files that need to be stored for a longer time; preserved between reboots

The manpage `man hier` contains information about filesystem hierarchy.

The **superblock** contains information relative to the filesystem e.g. filesystem type, size, status, metadata structures.
The **Master Boot Record (MBR)** is a 512-byte program located in the first sector of the hard disk; it contains information about hard disk partitions and has the duty of loading the OS. On recent systems, the MBR has been replaced by the **GUID Partition Table (GPT)**.
Almost all modern filesystems use **journaling**; in a journaling filesystem, the journal logs changes before committing them to the filesystem, which ensures faster recovery and less risk of corruption in case of a crash.

Partitioning limits for Linux using MBR:
Max 4 primary partitions per hard disk, or 3 primary partitions + 1 extended partition. Partitions are numbered from 1 to 4.
Max 11 logical partitions (inside the extended partition) per hard disk. Partitions are numbered from 5 to 15.
Max disk size is 2 Tb.

GPT makes no difference between primary, extended, or logical partitions. Furthermore, it practically has no limits concerning number and size of partitions.

FUSE (Filesystem in Userspace) is an interface for userspace programs to export a filesystem to the Linux kernel, and is particularly useful for virtual file systems.

`fdisk /dev/sda`	Disk partitioning interactive tool
`fdisk -l /dev/sda`	List the partition table of `/dev/sda`
`parted`	Disk partitioning interactive tool
`sfdisk /dev/sda`	Disk partitioning non-interactive tool
`cfdisk`	Disk partitioning tool with text-based UI
`gparted` `gnome-disks`	Disk partitioning tool with GUI
`partprobe` *device* `hdparm -z` *device*	Notify the OS about partition table changes. Otherwise, the changes will take place only after reboot

`mkfs -t` *fstype device*	Create a filesystem of the specified type on a partition (i.e. format the partition). mkfs is a wrapper utility for the actual filesystem-specific maker commands:
	`mkfs.ext2` aka `mke2fs`
	`mkfs.ext3` aka `mke3fs`
	`mkfs.ext4`
	`mkfs.msdos` aka `mkdosfs`
	`mkfs.ntfs` aka `mkntfs`
	`mkfs.reiserfs` aka `mkreiserfs`
	`mkfs.jfs`
	`mkfs.xfs`

`mkfs -t ext2 /dev/sda`
`mkfs.ext2 /dev/sda`
`mke2fs /dev/sda`
	Create an ext2 filesystem on `/dev/sda`

`mke2fs -j /dev/sda`
`mkfs.ext3 /dev/sda`
`mke3fs /dev/sda`
	Create an ext3 filesystem (ext2 with journaling) on `/dev/sda`

`mkfs -t msdos /dev/sda`
`mkfs.msdos /dev/sda`
`mkdosfs /dev/sda`
	Create a MS-DOS filesystem on `/dev/sda`

`mount`
`cat /proc/mounts`
`cat /etc/mtab`
	Display the currently mounted filesystems.
	The commands `mount` and `umount` maintain in `/etc/mtab` a database of currently mounted filesystems, but `/proc/mounts` is authoritative

`mount -a`	Mount all devices listed in `/etc/fstab`, except those indicated as `noauto`
`mount -t ext3 /dev/sda /mnt`	Mount a Linux-formatted disk. The mount point (directory) must exist
`mount -t msdos /dev/fd0 /mnt`	Mount a MS-DOS filesystem floppy disk to mount point `/mnt`
`mount /dev/fd0`	Mount a floppy disk. `/etc/fstab` must contain an entry for `/dev/fd0`
`mount -o remount,rw /`	Remount the root directory as read-write, supposing it was mounted read-only. Useful to change flags (in this case, read-only to read-write) for a mounted filesystem that cannot be unmounted at the moment
`mount -o nolock 10.7.7.7:/export/ /mnt/nfs`	Mount a NFS share without running NFS daemons. Useful during system recovery
`mount -t iso9660 -o ro,loop=/dev/loop0 cd.img /mnt/cdrom`	Mount a CD-ROM ISO9660 image file like a CD-ROM (via the loop device)

`umount /dev/fd0`
`umount /mnt`
	Unmount a floppy disk that was mounted on `/mnt` (device must not be busy)

`umount -l /dev/fd0`	Unmount the floppy disk as soon as it is not in use anymore

`eject /dev/fd0`
`eject /mnt`
	Eject a removable media device

`mountpoint /mnt`	Tell if a directory is a mount point

`blockdev --getbsz /dev/sda1`	Get the block size of the specified partition

The **UUID (Universal Unique Identifier)** of a partition is a 128-bit hash number, which is associated to the partition when the partition is initialized.

`blkid /dev/sda1`	Print the UUID of the specified partition
`blkid -L /boot`	Print the UUID of the specified partition, given its label
`blkid -U 652b786e-b87f-49d2-af23-8087ced0c667`	Print the name of the specified partition, given its UUID

`findfs UUID=652b786e-b87f-49d2-af23-8087ced0c667`	Print the name of the specified partition, given its UUID
`findfs LABEL=/boot`	Print the name of the specified partition, given its label

`e2label /dev/sda1`	Print the label of the specified partition

Partition types					
0x00	Empty	0x4e	QNX4.x 2nd part	0xa8	Darwin UFS
0x01	FAT12	0x4f	QNX4.x 3rd part	0xa9	NetBSD
0x02	XENIX root	0x50	OnTrack DM	0xab	Darwin boot
0x03	XENIX usr	0x51	OnTrack DM6 Aux1	0xaf	HFS / HFS+
0x04	FAT16 <32M	0x52	CP/M	0xb7	BSDI fs
0x05	Extended	0x53	OnTrack DM6 Aux3	0xb8	BSDI swap
0x06	FAT16	0x54	OnTrackDM6	0xbb	Boot Wizard hidden
0x07	HPFS / NTFS / exFAT	0x55	EZ-Drive	0xbe	Solaris boot
0x08	AIX	0x56	Golden Bow	0xbf	Solaris
0x09	AIX bootable	0x5c	Priam Edisk	0xc1	DRDOS/sec (FAT-12)
0x0a	OS/2 Boot Manager	0x61	SpeedStor	0xc4	DRDOS/sec (FAT-16 < 32Mb)
0x0b	W95 FAT32	0x63	GNU HURD or SysV	0xc6	DRDOS/sec (FAT-16)
0x0c	W95 FAT32 (LBA)	0x64	Novell Netware 286	0xc7	Syrinx
0x0e	W95 FAT16 (LBA)	0x65	Novell Netware 386	0xda	Non-FS data
0x0f	W95 extended (LBA)	0x70	DiskSecure Multi-Boot	0xdb	CP/M / CTOS / ...
0x10	OPUS	0x75	PC/IX	0xde	Dell Utility
0x11	Hidden FAT12	0x80	Old Minix	0xdf	BootIt
0x12	Compaq diagnostics	0x81	Minix / old Linux	0xe1	DOS access
0x14	Hidden FAT16 <32Mb	0x82	Linux swap / Solaris	0xe3	DOS R/O
0x16	Hidden FAT16	0x83	Linux	0xe4	SpeedStor
0x17	Hidden HPFS/NTFS	0x84	OS/2 hidden C: drive	0xeb	BeOS fs
0x18	AST SmartSleep	0x85	Linux extended	0xee	GPT
0x1b	Hidden W95 FAT32	0x86	NTFS volume set	0xef	EFI (FAT-12/16/32)
0x1c	Hidden W95 FAT32 (LBA)	0x87	NTFS volume set	0xf0	Linux/PA-RISC boot
0x1e	Hidden W95 FAT16 (LBA)	0x88	Linux plaintext	0xf1	SpeedStor
0x24	NEC DOS	0x8e	Linux LVM	0xf4	SpeedStor
0x27	Hidden NTFS WinRE	0x93	Amoeba	0xf2	DOS secondary
0x39	Plan 9	0x94	Amoeba BBT	0xfb	VMware VMFS
0x3c	PartitionMagic recovery	0x9f	BSD/OS	0xfc	VMware VMKCORE
0x40	Venix 80286	0xa0	IBM Thinkpad hibernation	0xfd	Linux raid autodetect
0x41	PPC PReP Boot	0xa5	FreeBSD	0xfe	LANstep
0x42	SFS	0xa6	OpenBSD	0xff	BBT
0x4d	QNX4.x	0xa7	NeXTSTEP		

The command `sfdisk -T` prints the above list of partition IDs and names.

Most used Linux-supported filesystems	
ext2	The oldest Linux ext filesystem, without journaling
ext3	ext2 with journaling
ext4	Linux journaling filesystem, an upgrade from ext3
Reiserfs	Journaling filesystem
XFS	Journaling filesystem, developed by SGI
JFS	Journaling filesystem, developed by IBM
Btrfs	B-tree filesystem, developed by Oracle
msdos	DOS filesystem, supporting only 8-char filenames
umsdos	Extended DOS filesystem used by Linux, compatible with DOS
fat32	MS-Windows FAT filesystem
vfat	Extended DOS filesystem, with support for long filenames
ntfs	Replacement for fat32 and vfat filesystems
minix	Native filesystem of the MINIX OS
iso9660	CD-ROM filesystem
cramfs	Compressed RAM disk
nfs	Network filesystem, used to access files on remote machines
SMB	Server Message Block, used to mount Windows network shares
proc	Pseudo filesystem, used as an interface to kernel data structures
swap	Pseudo filesystem, Linux swap area

The **swap** space is an area on disk (a file or a partition) used as a RAM extension. When there is not enough free physical RAM for a process, inactive pages in memory are temporarily **swapped out** of memory to disk, to later be **swapped in** to memory when RAM resources are available again. If both RAM and swap space become nearly full, the system may get clogged by spending all the time paging blocks of memory back and forth between RAM and swap (**thrashing**).
The amount of RAM plus the swap is defined as the **virtual memory**.

In Linux, a swap partition is usually preferred over a swap file. While a swap file can be resized more easily, it cannot be used for hibernation; this because the system must first locate the swap file's header, but in order to do so the filesystem containing the swap file must be mounted, and journaled filesystems such as ext3 or ext4 cannot be mounted during resume from disk. Also, in older Linux versions a swap partition used to have faster disk access and less fragmentation than a swap file, but the difference is negligible nowadays.
Although listed as filesystem type 0x82, the swap partition is not a filesystem but a raw addressable memory space with no structure; therefore it does not appear in the output of `mount` or `df` commands.
A swap partition can be created via any partitioning tool e.g. `fdisk`.

dd if=/dev/zero of=/*swapfile* bs=1024 count=512000	Create a 512-Mb swap file
mkswap /*swapfile*	Initialize a (already created) swap file or partition
swapon /*swapfile*	Enable a swap file or partition, thus telling the kernel that it can use it now
swapoff /*swapfile*	Disable a swap file or partition
swapon -s cat /proc/swaps cat /proc/meminfo free top	Show the sizes of total and used swap areas

How to extend a LVM swap partition

1.	`lvs`	Determine the name of the swap Logical Volume
2.	`swapoff /dev/volgroup0/swap_lv`	Turn off the swap volume
3.	`lvresize -L+1G /dev/volgroup0/swap_lv`	Extend the swap volume with an additional 1 Gb of space
4.	`mkswap /dev/volgroup0/swap_lv`	Format the swap volume
5.	`swapon /dev/volgroup0/swap_lv`	Turn on the swap volume

/etc/fstab

```
# <filesystem>     <mount point>     <type>    <options>                        <dump> <pass>

/dev/sda2          /                 ext2      defaults                               0 1
/dev/sdb1          /home             ext2      defaults                               1 2
/dev/cdrom         /media/cdrom      auto      ro,noauto,user,exec                    0 0
/dev/fd0           /media/floppy     auto      rw,noauto,user,sync                    0 0
proc               /proc             proc      defaults                               0 0
/dev/hda1          swap              swap      pri=42                                 0 0
nfsserver:/dirs    /mnt              nfs       intr                                   0 0
//smbserver/jdoe   /shares/jdoe      cifs      auto,credentials=/etc/smbcreds         0 0
LABEL=/boot        /boot             ext2      defaults                               0 0
UUID=652b786e-b87f-49d2-af23-8087ced0c667  /test  ext4  errors=remount-ro,noatime     0 0
```

`/etc/fstab` contains information about filesystems, including all filesystems that must be automatically mounted at bootup.		
filesystem	Device or partition. The filesystem can be identified either by its name, label, or UUID	
mount point	Directory on which the partition will be mounted	
type	Filesystem type, or `auto` if detected automatically	
options	`defaults`	Use the default options. The default options depend on the filesystem type and can be found via the command: `tune2fs -l` *`device`* `| grep "Default mount options"` Most common default options: `rw, suid, dev, auto, nouser, exec, async`
	`ro`	Mount read-only
	`rw`	Mount read-write (default)
	`suid`	Permit SUID and SGID bit operations (default)
	`nosuid`	Do not permit SUID and SGID bit operations
	`dev`	Interpret block special devices on the filesystem (default)
	`nodev`	Do not interpret block special devices on the filesystem
	`auto`	Mount automatically at bootup, or when command `mount -a` is given (default)
	`noauto`	Mount only if explicitly demanded
	`user`	Partition can be mounted by any user
	`nouser`	Partition can be mounted only by the root user (default)
	`exec`	Binaries contained on the partition can be executed (default)
	`noexec`	Binaries contained on the partition cannot be executed
	`sync`	Write files immediately to the partition
	`async`	Buffer write operations and commit them at once later, or when device is unmounted (default)
	`noatime`	Do not update atime (access time) information for the filesystem. This results in a performance improvement because the system does not need anymore to do filesystem writes for files which are just being read
	`acl`	Support ACLs on files contained in the partition
	`context="`*`context`*`"`	Apply a specific SELinux context to the mount
	Other specific options apply to specific partition types (e.g. NFS or Samba)	
dump	Options for the `dump` backup utility. `0` = do not backup	
pass	Order in which the filesystem must be checked by `fsck`. `0` = do not check	

df	Report filesystem disk space usage
df -h	Report filesystem disk space usage in human-readable output
df *directory*	Shows on which device the specified *directory* is mounted
du *directory*	Report disk usage, as the size of each file contained in *directory*, in Kb
du -s *directory*	Show the total sum of the sizes of all files contained in *directory*
du -h *directory*	Report disk usage in human-readable output
du -hs * | sort -hr	Print out all files and directories in the current directory, ordered by size (largest first), in human-readable output
du -a */path* | sort -nr | head	Print out the 10 biggest files and directories under *path*
find */path* -type f -exec du -Sh {} + \\ | sort -hr | head	Print out the 10 biggest files under *path*
ncdu	Disk usage analyzer with Ncurses UI
resize2fs *options device size*	Resize an ext2/ext3/ext4 filesystem
lsblk	List information about all available block devices
lsscsi	List information about all SCSI devices
sync	Flush the buffer and commit all pending writes. To improve performance of Linux filesystems, many write operations are buffered in RAM and written at once; writes are done in any case before unmount, reboot, or shutdown
chroot */path/to/newrootdir command*	Run a command in a chroot jail (i.e. in a new root directory). The command process will be unable to access files outside the chroot jail
chroot /mnt/sysimage	Start a shell with /mnt/sysimage as filesystem root. Useful during system recovery when the machine has been booted from a removable media; this device is defined as the filesystem root and often needs to be changed to perform operations on the machine
mknod /dev/sda	Create a directory allocating the proper inode. Useful if experiencing filesystem problems during system recovery
multipath *options device*	Detect and aggregate multiple I/O paths (SAN connections) to a device
hdparm	Get/set drive parameters for SATA/IDE devices
hdparm -g /dev/hda	Display drive geometry (cylinders, heads, sectors) of /dev/hda
hdparm -i /dev/hda	Display identification information for /dev/hda
hdparm -tT /dev/hda	Perform disk read benchmarks on the /dev/hda drive
hdparm -p 12 /dev/hda	Reprogram IDE interface chipset of /dev/hda to mode 4. Warning: using an unsupported mode can cause filesystem corruption
sdparm	Access drive parameters for SCSI devices

`fsck` *`device`*	Check and repair a Linux filesystem (which must be unmounted). Corrupted files will be placed into the `/lost+found` directory of the partition. The exit code returned is the sum of the following conditions:

0	No errors	8	Operational error
1	File system errors corrected	16	Usage or syntax error
2	System should be rebooted	32	Fsck canceled by user
4	File system errors left uncorrected	128	Shared library error

Fsck is a wrapper utility for the actual filesystem-specific checker commands:
`fsck.ext2` aka `e2fsck`
`fsck.ext3` aka `e2fsck`
`fsck.ext4` aka `e2fsck`
`fsck.msdos`
`fsck.vfat`
`fsck.cramfs`

`fsck`
`fsck -As`
	Check and repair serially all filesystems listed in `/etc/fstab`

`fsck -f /dev/sda1`	Force a filesystem check on `/dev/sda1` even if it thinks is not necessary
`fsck -y /dev/sda1`	During filesystem repair, do not ask questions and assume that the answer is always yes

`fsck.ext2 -c /dev/sda1`
`e2fsck -c /dev/sda1`
	Check an ext2 filesystem, running the `badblocks` command to mark all bad blocks and add them to the bad block inode so they will not be allocated to files or directories

`touch /forcefsck` (Red Hat)	Force a filesystem check after next reboot

`tune2fs` *`options device`*	Adjust tunable filesystem parameters on ext2/ext3/ext4 filesystems
`tune2fs -l /dev/sda1`	List the contents of the filesystem superblock
`tune2fs -j /dev/sda1`	Add a journal to this ext2 filesystem, making it an ext3
`tune2fs -m 1 /dev/sda1`	Reserve 1% of the partition size to privileged processes. This space (5% by default, but can be reduced on modern filesystems) is reserved to avoid filesystem fragmentation and to allow privileged processes to continue to run correctly when the partition is full
`tune2fs -C 7 /dev/sda1`	Set the mount count of the filesystem to 7
`tune2fs -c 20 /dev/sda1`	Set the filesystem to be checked by fsck after 20 mounts
`tune2fs -i 15d /dev/sda1`	Set the filesystem to be checked by fsck each 15 days

Both mount-count-dependent and time-dependent checking are enabled by default for all hard drives on Linux, to avoid the risk of filesystem corruption going unnoticed.

`dumpe2fs` *`options device`*	Dump ext2/ext3/ext4 filesystem information
`dumpe2fs -h /dev/sda1`	Display filesystem's superblock information (number of mounts, last checks, UUID, etc.)
`dumpe2fs /dev/sda1 | grep -i superblock`	Display locations of superblock (primary and backup) of filesystem
`dumpe2fs -b /dev/sda1`	Display blocks that are marked as bad in the filesystem

`debugfs` *`device`*	Interactive ext2/ext3/ext4 filesystem debugger
`debugfs -w /dev/sda1`	Debug `/dev/sda1` in read-write mode (by default, debugfs accesses the device in read-only mode)

Many hard drives feature the **Self-Monitoring, Analysis and Reporting Technology (SMART)** whose purpose is to monitor the reliability of the drive, predict drive failures, and carry out different types of drive self-tests.
The `smartd` daemon attempts to poll this information from all drives every 30 minutes, logging all data to syslog.

`smartctl -a /dev/sda`	Print SMART information for drive `/dev/sda`
`smartctl -s off /dev/sda`	Disable SMART monitoring and log collection for drive `/dev/sda`
`smartctl -t long /dev/sda`	Begin an extended SMART self-test on drive `/dev/sda`

`xfs_growfs` *`options mountpoint`*	Expand an XFS filesystem. Note that a XFS filesystem cannot be shrunk

`xfs_info /dev/sda1`
`xfs_growfs -n /dev/sda1`
	Print XFS filesystem geometry

`xfs_check` *`options device`*	Check XFS filesystem consistency
`xfs_repair` *`options device`*	Repair a damaged or corrupt XFS filesystem
`xfsdump -v silent -f /dev/tape /`	Dump the root of a XFS filesystem to tape, with the lowest verbosity. Incremental and resumed dumps are stored in the inventory database `/var/lib/xfsdump/inventory`
`xfsrestore -f /dev/tape /`	Restore a XFS filesystem from tape
`xfsdump -J - / | xfsrestore -J - /new`	Copy the contents of a XFS filesystem to another directory, without updating the inventory database

`reiserfstune` *`options device`*	Adjust tunable filesystem parameters on ReiserFS filesystem
`debugreiserfs` *`device`*	Interactive ReiserFS filesystem debugger

`mkisofs -r -o cdrom.img data/`	Create a CD-ROM image from the contents of the target directory. Enables Rock Ridge extension and set all content on CD to be public readable, instead of inheriting the permissions from the original files

CD-ROM filesystems

Filesystem	Commands	
ISO9660	`mkisofs`	Create a ISO9660 filesystem
UDF (Universal Disk Format)	`mkudffs`	Create a UDF filesystem
	`udffsck`	Check a UDF filesystem
	`wrudf`	Maintain a UDF filesystem
	`cdrwtool`	Manage CD-RW drives (e.g. disk format, read/write speed)
HFS (Hierarchical File System)		

CD-ROM filesystem extensions

Rock Ridge	Contains the original file information (e.g. permissions, filename) for MS Windows 8.3 filenames
MS Joliet	Used to create more MS Windows friendly CD-ROMs
El Torito	Used to create bootable CD-ROMs

AutoFS is a client-side service that allows automounting of filesystems, even for nonprivileged users.
AutoFS is composed of the `autofs` kernel module that monitors specific directories for attempts to access them; in this case, the kernel module signals the `automount` userspace daemon, which mounts the directory when it needs to be accessed and unmounts it when is no longer accessed.
Mounts managed by AutoFS should not be mounted/unmounted manually or via `/etc/fstab`, to avoid inconsistencies.

AutoFS configuration files	
`/etc/sysconfig/autofs`	AutoFS configuration file.

`/etc/auto.master`	Master map file for AutoFS.
	Each line is an indirect map, and each map file stores the configuration for the automounting of the subdirectory.
	The `-hosts` map tells AutoFS to mount/unmount automatically any export from the NFS server *nfsserver* when the directory `/net/`*`nfsserver`*`/` is accessed.
	
	`# mount point   map               options`
	`/net            -hosts`
	`/-              /etc/auto.direct`
	`/misc           /etc/auto.misc`
	`/home           /etc/auto.home    --timeout=60`

AutoFS map files	

`/etc/auto.direct`	Direct map file for automounting of a NFS share.
	
	`# dir      filesystem`
	`/mydir     nfsserver1.foo.org:/myshare`

`/etc/auto.misc`	Indirect map file for automounting of directory `/misc`.
	
	`# subdir   options                           filesystem`
	`public     -ro,soft,intr                     ftp.example.org:/pub`
	`cd         -fstype=iso9660,ro,nosuid,nodev   :/dev/cdrom`

`/etc/auto.home`	Indirect map file for automounting of directory `/home` on a NFS share.
	The `*` wildcard matches any subdirectory the system attempts to access, and the `&` variable takes the value of the match.
	
	`# subdir   options                           filesystem`
	`*          -rw,soft,intr                     nfsserver2.bar.org:/home/&`

RAID levels		
Level	**Description**	**Storage capacity**
RAID 0	Striping (data is written across all member disks). High I/O but no redundancy	Sum of the capacity of member disks
RAID 1	Mirroring (data is mirrored on all disks). High redundancy but high cost	Capacity of the smaller member disk
RAID 4	Parity on a single disk. I/O bottleneck unless coupled to write-back caching	Sum of the capacity of member disks, minus one
RAID 5	Parity distributed across all disks. Can sustain one disk crash	Sum of the capacity of member disks, minus one
RAID 6	Double parity distributed across all disks. Can sustain two disk crashes	Sum of the capacity of member disks, minus two
RAID 10 (1+0)	Striping + mirroring. High redundancy but high cost	Capacity of the smaller member disk
Linear RAID	Data written sequentially across all disks. No redundancy	Sum of the capacity of member disks

```
mdadm -C /dev/md0 -l 5 \
-n 3 /dev/sdb1 /dev/sdc1 /dev/sdd1 \
-x 1 /dev/sde1
```

Create a RAID 5 array from three partitions and a spare.
Partitions type must be set to 0xFD.
Once the RAID device has been created, it must be formatted e.g. via `mke2fs -j /dev/md0`

```
mdadm --manage /dev/md0 -f /dev/sdd1
```

Mark a drive as faulty, before removing it

```
mdadm --manage /dev/md0 -r /dev/sdd1
```

Remove a drive from the RAID array.
The faulty drive can now be physically removed

```
mdadm --manage /dev/md0 -a /dev/sdd1
```

Add a drive to the RAID array.
To be run after the faulty drive has been physically replaced

```
mdadm --misc -Q /dev/sdd1
```

Display information about a device

```
mdadm --misc -D /dev/md0
```

Display detailed information about the RAID array

```
mdadm --misc -o /dev/md0
```

Mark the RAID array as readonly

```
mdadm --misc -w /dev/md0
```

Mark the RAID array as read & write

```
/etc/mdadm.conf
```

Configuration file for the `mdadm` command

```
DEVICE /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1
ARRAY /dev/md0 level=raid5 num-devices=3
  UUID=0098af43:812203fa:e665b421:002f5e42
  devices=/dev/sdb1,/dev/sdc1,/dev/sdd1,/dev/sde1
```

```
cat /proc/mdstat
```

Display information about RAID arrays and devices

<table>
<tr><th colspan="3">Non-GRUB bootloaders</th></tr>
<tr><td colspan="2">LILO
(Linux Loader)</td><td>Obsolete. Small bootloader that can be placed in the MBR or the boot sector of a partition.
The configuration file is <code>/etc/lilo.conf</code> (run <code>/sbin/lilo</code> afterwards to validate changes).</td></tr>
<tr><td rowspan="4">SYSLINUX</td><td>SYSLINUX</td><td>Able to boot from FAT and NTFS filesystems e.g. floppy disks and USB drives.
Used for boot floppy disks, rescue floppy disks, and Live USBs.</td></tr>
<tr><td>ISOLINUX</td><td>Able to boot from CD-ROM ISO 9660 filesystems.
Used for Live CDs and bootable install CDs.

The CD must contain the following files:

<code>isolinux/isolinux.bin</code> ISOLINUX image, from the SYSLINUX distro
<code>boot/isolinux/isolinux.cfg</code> ISOLINUX configuration
<code>images/</code> Floppy images to boot
<code>kernel/memdisk</code>

and can be burnt with the command:

<code>mkisofs -o output.iso -b isolinux/isolinux.bin -c isolinux/boot.cat \</code>
<code>-no-emul-boot -boot-load-size 4 -boot-info-table <i>cd_root_dir</i></code></td></tr>
<tr><td>PXELINUX</td><td>Able to boot from PXE (Pre-boot eXecution Environment). PXE uses DHCP or BOOTP to enable basic networking, then uses TFTP to download a bootstrap program that loads and configures the kernel.
Used for Linux installations from a central server or network boot of diskless workstations.

The boot TFTP server must contain the following files:

<code>/tftpboot/pxelinux.0</code> PXELINUX image, from the SYSLINUX distribution
<code>/tftpboot/pxelinux.cfg/</code> Directory containing a configuration file for each machine. A machine with Ethernet MAC address 88:99:AA:BB:CC:DD and IP address 192.0.2.91 (C000025B in hexadecimal) will search for its configuration filename in this order:
<code>01-88-99-aa-bb-cc-dd</code>
<code>C000025B</code>
<code>C000025</code>
<code>C00002</code>
<code>C0000</code>
<code>C000</code>
<code>C00</code>
<code>C0</code>
<code>C</code>
<code>default</code></td></tr>
<tr><td>EXTLINUX</td><td>General-purpose bootloader like LILO or GRUB. Now merged with SYSLINUX.</td></tr>
</table>

GRUB (Grand Unified Bootloader) is the standard boot manager on Linux distributions. The latest version is GRUB 2; the older version is GRUB Legacy.
GRUB Stage 1 (446 bytes), as well as the partition table (64 bytes) and the boot signature (2 bytes), is stored in the 512-byte MBR. It then accesses the GRUB configuration and commands available on the filesystem, usually on `/boot/grub`.

`/boot/grub/grub.cfg` or `/boot/grub2/grub.cfg` GRUB 2 configuration file

```
# Linux Red Hat
menuentry "Fedora 2.6.32" {    # Menu item to show on GRUB bootmenu
set root=(hd0,1)               # root filesystem is /dev/hda1
linux /vmlinuz-2.6.32 ro root=/dev/hda5 mem=2048M
initrd /initrd-2.6.32
}

# Linux Debian
menuentry "Debian 2.6.36-experimental" {
set root=(hd0,1)
linux (hd0,1)/bzImage-2.6.36-experimental ro root=/dev/hda6
}

# Windows
menuentry "Windows" {
set root=(hd0,2)
chainloader +1
}
```

The GRUB 2 configuration file must not be edited manually. Instead, one must edit the files in `/etc/grub.d/` (these are scripts that will be run in order) and the file `/etc/default/grub` (the configuration file for menu display settings), then run `update-grub` (Debian) or `grub2-mkconfig` (Red Hat) which will recreate this configuration file.

Common kernel parameters:	`root=`	Specify the location of the filesystem root. This is a required parameter
	`ro`	Mount read-only on boot
	`quiet`	Disable non-critical kernel messages during boot
	`debug`	Enable kernel debugging
	`splash`	Show splash image
	`single`	Boot in single-user mode (runlevel 1)
	`emergency`	Emergency mode: after the kernel is booted, run `sulogin` (single-user login) which asks for the root password for system maintenance, then run a Bash shell. Does not load `init` or any daemon or configuration setting
	`init=/bin/bash`	Run a Bash shell (may also be any other executable) instead of `init`

The GRUB menu, presented at startup, allows to choose the OS or kernel to boot:

ENTER	Boot the currently selected GRUB entry
C	Get a GRUB command line
E	Edit the selected GRUB entry (e.g. to edit kernel parameters in order to boot in single-user emergency mode, or to change IRQ or I/O port of a device driver compiled in the kernel)
B	Boot the currently selected GRUB entry. This is usually done after finishing modifying the entry
P	Bring up the GRUB password prompt. Necessary if a GRUB password has been set

`grub2-mkconfig -o /boot/grub2/grub.cfg`
`grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg`
	(BIOS)
	(EFI)
		Regenerate GRUB configuration file

`grub-install /dev/sda`	Install GRUB on first SATA drive
`grub`	Access the GRUB shell
`grub2-set-default 1`	Set GRUB to automatically boot the second entry in the GRUB menu
`grub2-editenv list`	Display the current GRUB menu entry that is automatically booted

`/boot/grub/device.map`	This file can be created to map Linux device filenames to BIOS drives

```
(fd0)   /dev/fd0
(hd0)   /dev/hda
```

GRUB Legacy shell commands

`blocklist` *file*	Print the block list notation of a file	`kernel` *file*	Load a kernel
`boot`	Boot the loaded OS	`lock`	Lock a GRUB menu entry
`cat` *file*	Show the contents of a file	`makeactive`	Set active partition on root disk to GRUB's root device
`chainloader` *file*	Chainload another bootloader	`map` *drive1 drive2*	Map a drive to another drive
`cmp` *file1 file2*	Compare two files	`md5crypt`	Encrypt a password in MD5 format
`configfile` *file*	Load a configuration file	`module` *file*	Load a kernel module
`debug`	Toggle debugging mode	`modulenounzip` *file*	Load a kernel module without decompressing it
`displayapm`	Display APM BIOS information	`pause` *message*	Print a message and wait for a key press
`displaymem`	Display memory configuration	`quit`	Quit the GRUB shell
`embed` *stage device*	Embed Stage 1.5 in the device	`reboot`	Reboot the system
`find` *file*	Find a file	`read` *address*	Read a 32-bit value from memory and print it
`fstest`	Toggle filesystem test mode	`root` *device*	Set the current root device
`geometry` *drive*	Print information on a drive geometry	`rootnoverify` *device*	Set the current root device without mounting it
`halt`	Shut down the system	`savedefault`	Save current menu entry as the default entry
`help` *command*	Show help for a command, or the available commands	`setup` *device*	Install GRUB automatically on the device
`impsprobe`	Probe the Intel Multiprocessor Specification	`testload` *file*	Test the filesystem code on a file
`initrd` *file*	Load an initial ramdisk image file	`testvbe` *mode*	Test a VESA BIOS EXTENSION mode
`install` *options*	Install GRUB (deprecated, use `setup` instead)	`uppermem` *kbytes*	Set the upper memory size (only for old machines)
`ioprobe` *drive*	Probe I/O ports used for a drive	`vbeprobe` *mode*	Probe a VESA BIOS EXTENSION mode

`/boot/grub/menu.lst` or `/boot/grub/grub.conf` GRUB Legacy configuration file

```
timeout 10   # Boot the default kernel after 10 seconds
default 0    # Default kernel is 0

# Section 0: Linux boot
title   Debian     # Menu item to show on GRUB bootmenu
root    (hd0,0)    # root filesystem is /dev/hda1
kernel  /boot/vmlinuz-2.6.24-19-generic root=/dev/hda1 ro quiet splash
initrd  /boot/initrd.img-2.6.24-19-generic

# Section 1: Windows boot
title   Microsoft Windows XP
root    (hd0,1)    # root filesystem is /dev/hda2
savedefault
makeactive         # set the active flag on this partition
chainloader +1     # read 1 sector from start of partition and run

# Section 2: Firmware/BIOS update from floppy disk
title   Firmware update
kernel /memdisk    # boot a floppy disk image
initrd /floppy-img-7.7.7
```

dpkg is the low-level package manager for Debian. It uses the DEB package format, which is compressed with ar.

dpkg -i *package*.deb	Install a package file
dpkg -r *package*	Remove a package
dpkg -l	List installed packages and their state
dpkg -L *package*	List the content of an installed package
dpkg -c *package*.deb	List the content of a package file
dpkg -S *file*	Show the package containing a specific file
dpkg-reconfigure *package*	Reconfigure a package

apt is the high-level package manager for Debian.
High-level package managers are able to install remote packages and automatically solve dependencies.

apt-get install *package*	Install a package
apt-get remove *package*	Remove a package
apt-get upgrade	Upgrade all installed packages
apt-get dist-upgrade	Upgrade all installed packages and handle dependencies with new versions
apt-get source *package*	Get the source code for a package
apt-get check	Check for broken dependencies and update package cache
apt-get install -f	Fix broken dependencies
apt-get update	Update information on available packages
apt-cache search *package*	Search for a package
apt-cache depends *package*	Show package dependencies
apt-cache show *package*	Show package records
apt-cache showpkg *package*	Show information about a package
apt-file update	Update information about package contents
apt-file list *package*	List the content of an uninstalled package
apt-file search *file*	Show which package provides a specific file
apt-key add *keyfile*	Add a key to the list of keys used to authenticate packages
apt-cdrom add	Add a CD-ROM to the sources list
cat /etc/apt/sources.list	Print list of available repositories

alien -i *package*.rpm	Convert a RPM package to DEB and install it. Warning: might break the package database system

dselect	Package manager with text interface, front-end to dpkg. Obsolete
aptitude	Package manager with Ncurses UI, front-end to apt
synaptic	Package manager with Gtk+ UI, front-end to apt

rpm is the low-level package manager for Red Hat. It uses the RPM package format, which is cpio-compressed.

rpm -i *package*.rpm rpm -i ftp://*host*/*package*.rpm rpm -i http://*host*/*package*.rpm	Install a package file
rpm -e *package*	Remove a package
rpm -U *package*.rpm	Upgrade a package (and remove old versions)
rpm -F *package*.rpm	Upgrade a package (only if an old version is already installed)
rpm -qa	List installed packages and their state
rpm -qa --last	List installed packages and their installation date, from newest to oldest
rpm -ql *package*	List the content of an installed package
rpm -qpl *package*.rpm	List the content of a package file
rpm -qf *file*	Show the package containing a specific file
rpm -V *package*	Verify an installed package
rpm -i *package*.src.rpm	Install a package source file
rpm -ba *package*.spec	Compile a package source file

rpm2cpio *package*.rpm	Convert a RPM package to a cpio archive
createrepo *directory*	Create an XML file of repository metadata from the set of RPMs contained in *directory*

pirut	Package manager with GUI. Obsolete

yum is the high-level package manager for Red Hat up to RHEL 7. In RHEL 8, it is a front-end to dnf.

yum install *package*	Install a package
yum install *package*.rpm yum localinstall *package*.rpm	Install a package file
yum remove *package*	Remove a package
yum update *package*	Upgrade an installed package
yum update	Upgrade all installed packages
yum swap *packageout packagein*	Replace a package with another
yum list	List all installed and available packages
yum list *searchterm*	List installed and available packages matching the search term
yum list installed	List installed packages
yum list available	List packages available for install
yum search *searchterm*	Search for packages that match the search term in the package name or summary
yum search all *searchterm*	Search for packages that match the search term in the package name, summary, or description
yum deplist *package*	Show package dependencies (recursively)
yum list *package*	Show package records
yum info *package*	Show information about a package
yum history yum history list	Show the installation history (installs, updates, etc.)
yum history list *n*	Show item *n* of the installation history
yum history info *n*	Show detailed information on item *n* of the installation history (begin and end times, packages altered, etc.)
yum history package *package* yum history list package *package*	Show the installation history about a package
yum whatprovides *file*	Show which package provides a specific file
yum *cmd* --disablerepo="*" --enablerepo="*repo*"	Execute the yum command but only considering a specific repository *repo*
yum repolist cat /etc/yum.repos.d/*.repo	Print list of available repositories
yum clean all rm -rf /var/cache/yum	Delete temporary files for repositories

yumdownloader --resolve *package*	Download package and all its dependencies
yumdownloader --urls *package*	Show URLs that would be downloaded
yum-complete-transaction	Try to complete unfinished or aborted package installations
repoquery --tree-requires *package*	Show a tree with all dependencies of *package*

Configuration of a Fedora repository (Red Hat)	
`[fedora]`	Repository ID
`name=Fedora $releasever - $basearch`	Repository name

`baseurl=http://download.fedoraproject.org/pub/fedora/\`
`        linux/releases/$releasever/Everything/$basearch/os/`
`        http://foo.org/linux/$releasever/$basearch/os/`
`        http://bar.org/linux/$releasever/$basearch/os/`
	List of URLs to the repository's repodata directory. Can be any of these types:
	`file:///` local file
	`file://` NFS
	`http://` HTTP
	`https://` HTTPS
	`ftp://` FTP

`enabled=1`	Whether this repository is enabled
`gpgcheck=1`	Whether to perform a GPG signature check on the packages downloaded from this repository
`failovermethod=priority`	Makes yum try the baseurls in the order they are listed. By default, if more than one baseurl is specified, yum chooses one randomly

`metalink=https://mirrors.fedoraproject.org/metalink?\`
`repo=fedora-$releasever&arch=$basearch`
	URL to a metalink file that specifies the list of mirrors to use. Can be used with or in alternative to a baseurl

`gpgkey=file:///etc/pki/rpm-gpg/\`
`RPM-GPG-KEY-fedora-$releasever-$basearch`
	ASCII-armored GPG public key file of the repository

This repository configuration must be located in a repo file e.g. `/etc/yum.repos.d/fedora.repo`. The same repo file can contain multiple repository definitions.
The manpage `man yum.conf` lists all repository configuration options.

How to install a package on an offline machine

The problem of installing a package on an offline machine is that the machine is unable to download the package dependencies. To solve this problem, first create an online machine identical to the offline machine, and with the smallest possible set of packages installed. Then proceed as follows.
On the online machine:

1. Install the package and all its dependencies in a local directory

```
mkdir /tmp/repo
yum --downloadonly --downloaddir=/tmp/repo install package
```

2. Create a local yum repository

```
createrepo /tmp/repo
chown -R root:root /tmp/repo
chmod -R 755 /tmp/repo
```

3. Transfer the directory `/tmp/repo` from the online machine to the offline machine

On the offline machine:

4. Create a yum repo file `/etc/yum.repos.d/local.repo` for the new repository

```
[local]
name=Local
baseurl=file:///tmp/repo
enabled=1
gpgcheck=0
protect=1
```

5. Install the package from the local repository

```
yum install package
```

`dd`	Tool to copy data, byte by byte, from a file or block device. Should not be used on a mounted block device, because of write cache issues.

`dd if=/dev/sda of=/dev/sdb`
`cat /dev/sda > /dev/sdb`
	Copy the content of one hard disk over another

`dd if=/dev/sda1 of=sda1.img`	Generate the image file of a partition
`dd if=/dev/cdrom of=cdrom.iso bs=2048`	Create an ISO file from a CD-ROM, using a block size transfer of 2 Kb
`dd if=install.iso of=/dev/sdc bs=512k`	Write an installation ISO file to a device (e.g. a USB thumb drive)
`ddrescue`	Tool for data recovery. Like `dd`, but with high tolerance for read errors
`rsync`	Tool for local and remote file synchronization. For all copies subsequent to the first, copies only the blocks that have changed, making it a very efficient backup solution in terms of speed and bandwidth

`rsync -rzv /home  /tmp/bak`
`rsync -rzv /home/ /tmp/bak/home`
	Synchronize the content of the home directory with the temporary backup directory. Use recursion, compression, and verbosity

`rsync -avz /home root@10.0.0.7:/backup/`	Synchronize the content of the home directory with the backup directory on the remote server, using SSH. Use archive mode (i.e. operates recursively and preserves owner, group, permissions, timestamps, and symlinks)
`burp`	Backup and restore program

Tape libraries		
Devices	`/dev/st0`	First SCSI tape device
	`/dev/nst0`	First SCSI tape device (no-rewind device file)
Utility for magnetic tapes	`mt -f /dev/nst0 asf 3`	Position the tape at the start of 3rd file
Utility for tape libraries	`mtx -f /dev/sg1 status`	Display status of tape library
	`mtx -f /dev/sg1 load 3`	Load tape from slot 3 to drive 0
	`mtx -f /dev/sg1 unload`	Unload tape from drive 0 to original slot
	`mtx -f /dev/sg1 transfer 3 4`	Transfer tape from slot 3 to slot 4
	`mtx -f /dev/sg1 inventory`	Force robot to rescan all slots and drives
	`mtx -f /dev/sg1 inquiry`	Inquiry about SCSI media device (Medium Changer = tape library)

cpio	`ls | cpio -o > archive.cpio`
	`ls | cpio -oF archive.cpio`
		Create a cpio archive of all files in the current directory

	`find /home/ | cpio -o > archive.cpio`	Create a cpio archive of all users' home directories
	`cpio -id < archive.cpio`	Extract all files, recreating the directory structure
	`cpio -i -t < archive.cpio`	List the contents of a cpio archive file
gzip	`gzip file`	Compress a file with gzip
	`gzip < file > file.gz`	Compress a file with gzip, leaving the original file into place
	`gunzip file.gz`	Decompress a gzip-compressed file
	`gunzip -tv file.gz`	Test the integrity of a gzip-compressed file
	`zcat file.gz`	Read a gzip-compressed text file
	`zgrep pattern file.gz`	`grep` for a gzip-compressed text file
	`zless file.gz`	`less` for a gzip-compressed text file
	`zmore file.gz`	`more` for a gzip-compressed text file
	`pigz file`	Parallel, multicore-optimized gzip
bzip2	`bzip2 file`	Compress a file with bzip2
	`bunzip2 file.bz2`	Decompress a bzip2-compressed file
	`bzcat file.bz2`	Read a bzip2-compressed text file
7-Zip	`7z a -t7z archive.7z dir/`	Create a 7-Zip archive (has the highest compression ratio)
xz	`xz file`	Compress a file with xz

	`unxz file.xz`
	`xz -d file.xz`
		Decompress a xz-compressed file

	`xzcat file.xz`	Read a xz-compressed file

LZMA	`lzma file`
	`xz --format=lzma file`
		Compress a file with LZMA

	`unlzma file.lzma`
	`xz --format=lzma -d file.lzma`
		Decompress a LZMA-compressed file

	`lzcat file.lzma`
	`xz --format=lzma --d --stdout file.lzma`
		Read a LZMA-compressed file

rar	`rar a archive.rar dir/`	Create a RAR archive
	`unrar x archive.rar`	Extract a RAR archive
tar	`tar cf archive.tar dir/`	Create a tarred archive (bundles multiple files in a single one)
	`tar czf archive.tar.gz dir/`	Create a tarred gzip-compressed archive
	`tar xzf archive.tar.gz`	Extract a tarred gzip-compressed archive
	`tar cjf archive.tar.bz2 dir/`	Create a tarred bzip2-compressed archive
	`tar xjf archive.tar.bz2`	Extract a tarred bzip2-compressed archive
	`tar cJf archive.tar.xz dir/`	Create a tarred xz-compressed archive
	`tar xJf archive.tar.xz`	Extract a tarred xz-compressed archive
	`tar tf archive.tar`	List the contents of a tarred archive
star	`star -c -f=archive.star dir/`	Create a star archive
	`star -x -f=archive.star`	Extract a star archive

`man` *command*	Show the manpage for *command*
`man` *n* *command*	Show section *n* of the *command* manpage
`man man`	Show information about manpages' sections: 1 - Executable programs or shell commands 2 - System calls (functions provided by the kernel) 3 - Library calls (functions within program libraries) 4 - Special files 5 - File formats and conventions 6 - Games 7 - Miscellaneous 8 - System administration commands (only for root) 9 - Kernel routines
`man` *n* `intro`	Show an introduction to the contents of section *n*
`mandb`	Generate or refresh the search database for manpage entries. This must be done after installing new packages, in order to obtain results from `apropos` or `man -k`
`yum whatprovides /usr/share/man/man`*n*`/`*command*`.`*n*`.gz`	Find which package provides section *n* of the *command* manpage
`yum install man-pages` (Red Hat)	Install a large number of manpages from the Linux Documentation Project
`yum install man-db` (Red Hat)	Install various manpage commands and utilities
`apropos` *keyword* `man -k` *keyword*	Show the commands whose manpage's short description matches the keyword. Inverse of the `whatis` command
`apropos -r` *regex* `man -k` *regex*	Show the commands whose manpage's short description matches the regex
`man -K` *regex*	Show the commands whose manpage's full text matches the regex
`whatis` *command*	Show the manpage's short description for a command
`info` *command*	Show the Info documentation for a command
`help`	Show the list of available shell commands and functions
`help` *command*	Show help about a shell command or function

`history`	Show the history of command lines executed up to this moment. Commands prepended by a space will be executed but will not show up in the history. After the user logs out from Bash, history is saved into `~/.bash_history`
`!n`	Execute command number *n* in the command line history
`history -c`	Clear the command line history
`history -d n`	Delete command number *n* from the command line history

`alias ls='ls -lap'`	Set up an alias for the `ls` command
`alias`	Show defined aliases
`unalias ls`	Remove the alias for the `ls` command
`\ls` `/bin/ls`	Run the non-aliased version of the `ls` command

Almost all Linux commands accept the option `-v` (verbose), and some commands also accept the options `-vv` or `-vvv` (increasing levels of verbosity).

All Bash built-in commands, and many other commands, accept the flag `--` which denotes the end of options and the start of positional parameters:

`grep -- -i file`	Search for the string "-i" in *file*
`rm -- -rf`	Delete a file called "-rf"

`cat /etc/debian_version`	(Debian)	Display Linux distribution name and version
`cat /etc/fedora-release`	(Fedora)	
`cat /etc/redhat-release`	(Red Hat)	
`cat /etc/lsb-release`		
`lsb_release -a`		
`cat /etc/os-release`		

`cat` *file*	Print a text file
`cat` *file1 file2* `>` *file3*	Concatenate text files
`cat` *file1* `>` *file2* `>` *file2* `<` *file1* `cat`	Copy *file1* to *file2*. The `cat` command is able to operate on binary streams as well and therefore it works also with binary files (e.g. JPG images)
`cat >` *file* `<<`*EOF* *line 1* *line 2* *line 3* *EOF*	Create a **Here Document**, storing the lines entered in input to *file*. *EOF* can be any text
command `<<< '`*string*`'`	Create a **Here String**, passing *string* as input to *command*
`cat -etv <<< '`*string*`'`	Print *string*, showing all invisible characters
`tac` *file*	Print or concatenate text files in opposite order line-wise, from last line to first line
`rev` *file*	Print a text file with every line reversed character-wise, from last char to first char
`head` *file* `head -n 10` *file*	Print the first 10 lines of a text file
`tail` *file* `tail -n 10` *file*	Print the last 10 lines of a text file
`tail -f` *file*	Output appended data as the text file grows. Useful to read a logfile in real-time
`tail -n +1` *file1 file2 file3*	Print each file with a filename header
`multitail -i` *file1* `-i` *file2*	`tail` for multiple files at the same time (Ncurses UI)
`column` *file*	Format a text file into columns
`pr` *file*	Format a text file for a printer
`fmt -w 75` *file*	Format a text file so that each line has a max width of 75 characters
`fold -w40` *file*	Wrap each line of a text file to 40 characters
`nl` *file*	Prepend line numbers to a text file
`wc` *file*	Print the number of lines, words, and bytes of a text file
`join` *file1 file2*	Join lines of two text files on a common field
`paste` *file1 file2*	Merge lines of text files
`split -l 1` *file*	Split a text file into 1-line files; these will be named `xaa`, `xab`, `xac`, etc.
`uniq` *file*	Print the unique lines of a text file, omitting consecutive identical lines
`sort` *file*	Sort alphabetically the lines of a text file
`shuf` *file*	Shuffle randomly the lines of a text file
`expand` *file*	Convert tabs into spaces
`unexpand` *file*	Convert spaces into tabs
`diff` *file1 file2*	Compare two text files line by line and print the differences
`cmp` *file1 file2*	Compare two files and print the differences

`cut -d: -f3` *file*	Cut the lines of a file, considering `:` as the delimiter and printing only the 3rd field
`cut -d: -f1 /etc/passwd`	Print the list of local user accounts in the system
`cut -c3-50` *file*	Print character 3 to 50 of each line of a file
`sed 's/foo/bar/'` *file*	Stream Editor: Replace the first occurrence on a line of "foo" with "bar" in *file*, and print on stdout the result
`sed -i 's/foo/bar/'` *file*	Replace "foo" with "bar", overwriting the results in *file*
`sed 's/foo/bar/g'` *file*	Replace all occurrences of "foo" with "bar"
`sed '0,/foo/s//bar/'` *file*	Replace only the first line match
`sed -n '7,13p'` *file*	Print line 7 to 13 of a text file
`sed "s/foo/$var/"` *file*	Replace "foo" with the value of variable $var. The double quotes are necessary for variable expansion
`tr a-z A-Z <`*file* `tr [:lower:] [:upper:] <`*file*	Translate characters: Convert all lowercase into uppercase in a text file
`tr -d 0-9 <`*file* `tr -d [:digit:] <`*file*	Delete all digits from a text file
`awk`	Interpreter for the AWK programming language, designed for text processing and data extraction
`grep foo` *file*	Print the lines of a file containing "foo"
`grep -v foo` *file*	Print the lines of a file not containing "foo"
`grep -e foo -e bar` *file* `grep -E 'foo|bar'` *file*	Print the lines of a file containing "foo" or "bar"
`grep -v -e foo -e bar` *file*	Print the lines of a file containing neither "foo" nor "bar"
`grep -E` *regex file* `egrep` *regex file*	Print the lines of a file matching the given Extended Regex

`tail -f` *file* `| grep --line-buffered foo`
`tail -f` *file* `| stdbuf -o0 grep foo`
	Output appended data as the text file grows, printing only the lines containing "foo"

`stdbuf` *option command*	Run *command* with modified stdin, stdout, or stderr buffering
`rpl` *oldstring newstring file*	Replace strings in a file
`tidy`	Correct and tidy up the markup of HTML, XHTML, and XML files
`tidy -asxml -xml -indent -wrap 2000 \` `-quiet --hide-comments yes` *file.xml*	Strip out comments from an XML file
`json_verify <` *file.json*	Validate the syntax of a JSON file
`json_reformat <` *file.json*	Pretty format a JSON file
`strings` *file*	Show all printable character sequences at least 4-characters long that are contained in *file*
`antiword` *file.doc*	Show text and images from a MS Word document
`catdoc` *file.doc*	Output plaintext from a MS Word document

`^`	Beginning of a line
`$`	End of a line
`\< \>`	Word boundaries (beginning of line, end of line, space, or punctuation mark)
`.`	Any character except newline
`[abc]`	Any of the characters specified
`[a-z]`	Any of the characters in the specified range
`[^abc]`	Any character except those specified
`*`	Zero or more times the preceding regex
`+`	One or more times the preceding regex
`?`	Zero or one time the preceding regex
`{5}`	Exactly 5 times the preceding regex
`{5,}`	5 times or more the preceding regex
`{,10}`	At most 10 times the preceding regex
`{5,10}`	Between 5 and 10 times the preceding regex
`|`	The regex either before or after the vertical bar
`( )`	Grouping, to be used for back-references. `\1` expands to the 1st match, `\2` to the 2nd, etc. until `\9`

The symbols above are used in POSIX EREs (Extended Regular Expressions).
In POSIX BREs (Basic Regular Expressions), the symbols `? + { | ( )` need to be escaped (by adding a backslash character `\` in front of them).

cp *file file2*	Copy a file
cp *file dir/*	Copy a file to a directory
cp -ar */dir1/*. */dir2/*	Copy a directory recursively
mv *file file2*	Rename a file
mv *file dir/*	Move a file to a directory
rm *file*	Delete a file

Common options:
-i Prompt before overwriting/deleting files (interactive)
-f Don't ask before overwriting/deleting files (force)

pv *file* > *file2*	Copy a file, monitoring the progress of data through a pipe
rename *str1 str2 file*	Rename a file, replacing the first occurrence of string *str1* with *str2*
unlink *file*	Remove the hard link to a file (equivalent to rm)
touch *file*	Change access timestamp and modify timestamp of a file as now. If the file does not exist, it is created
truncate -s *size file*	Shrink or extend a file to the specified size. If the file is larger than the specified size, it is truncated; if the file is shorter, the extra space is filled with zeros
mktemp	Create a temporary file or directory, using tmp.XXXXXXXXXX as filename template
fdupes *dir*	Examines a directory for duplicate files in it. To consider files a duplicate, first compares file sizes and MD5 signatures, then compares the file contents byte-by-byte
tmpwatch	Remove files which have not been accessed for some time
od *file*	Dump a file into octal (or other formats)
hexdump *options file*	Dump a file into hexadecimal (or other formats e.g. octal, decimal, ASCII)
xxd *options file*	Convert a file from binary to hexadecimal, or vice versa

File-naming wildcards (globbing)	
*	Matches zero or more characters
?	Matches one character
[abc]	Matches a, b, or c
[!abc]	Matches any character except a, b, or c
[a-z]	Matches any character between a and z

Brace expansion	
cp foo.{txt,bak}	Copy file "foo.txt" to "foo.bak"

touch foo_{a,b,c}
touch foo_{a..c}
	Create files "foo_a", "foo_b", "foo_c"

cd *directory*	Change to the specified directory
cd -	Change to the previously used directory
pwd	Print the current working directory
ls dir vdir	List the contents of the current directory
ls -d */	List only directories contained on the current directory
ls -lap --sort=v	List files, sorted by version number
mkdir *dir*	Create a directory
mkdir -m 755 *dir*	Create a directory with mode 755
mkdir -p */dir1/dir2/dir3*	Create a directory, creating also the parent directories if they don't exist
rmdir *dir*	Delete a directory (which must be empty)
tree	List directories and their contents in hierarchical format
dirs	Display the directory stack (i.e. the list of remembered directories)
pushd *dir*	Add *dir* to the top of the directory stack and make it the current working directory
popd	Remove the top directory from the directory stack and change to the new top directory
dirname *file*	Output the directory path in which *file* is located, stripping any non-directory suffix from the filename
realpath *file*	Output the resolved absolute path of *file*

Bash directory shortcuts	
.	Current directory
..	Parent directory
~	Home directory of current user
~*user*	Home directory of *user*
~-	Previously used directory

`stat` *file*	Display file or filesystem status
`stat -c %A` *file*	Display file permissions
`stat -c %s` *file*	Display file size, in bytes

`shred /dev/hda`	Securely wipe the contents of a device
`shred -u` *file*	Securely delete a file

`lsof`	List all open files
`lsof -u` *user*	List all files currently open by *user*
`lsof -i`	List open files and their sockets (equivalent to `netstat -ap`)
`lsof -i :80`	List connections of local processes on port 80
`lsof -i@10.0.0.3`	List connections of local processes to remote host 10.0.0.3
`lsof -i@10.0.0.3:80`	List connections of local processes to remote host 10.0.0.3 on port 80
`lsof -c mysqld`	List all files opened by `mysqld`, the MySQL daemon
`lsof` *file*	List all processes using a specific *file*
`lsof +L1`	List open files with a link count of 0 i.e. that have been unlinked. These files are not accessible but take up disk space. A process holding such a file prevents the system from deleting it (thus freeing disk space), until the process is killed or restarted

`fuser`	Show the name of processes using a specific file, directory, or socket
`fuser -v` *file*	Show the name of the process using *file*
`fuser -v -n tcp 443`	Show the name of the process running on port 443

`lslocks`	List information about all currently held file locks

`aide`	Advanced Intrusion Detection Environment. HIDS tool that makes a snapshot of the filesystem state and records it in a database, to check integrity of files at a later time

In Linux, everything is (displayed as) a file. File descriptors are automatically associated to any process launched.

File descriptors

#	Name	Type	Default device	Device file
0	Standard input (stdin)	Input text stream	Keyboard	`/dev/stdin`
1	Standard output (stdout)	Output text stream	Terminal	`/dev/stdout`
2	Standard error (stderr)	Output text stream	Terminal	`/dev/stderr`

```
mail user@email < file
```

Redirect *file* to the stdin of command `mail` (in this case, send via e-mail the contents of *file* to the email address *user@email*).
Redirection is handled by the shell, not by the command invoked. The space after the redirect operator is optional

```
ls > file
ls 1> file
```

Redirect the stdout of command `ls` to *file* (in this case, write on *file* the contents of the current directory). This overwrites *file* if it already exists, unless the Bash noclobber option is set (via `set -o noclobber`)

```
ls >| file
```

Redirect the stdout of command `ls` to *file*, even if noclobber is set

```
ls >> file
ls 1>> file
```

Append the stdout of command `ls` to *file*

```
ls 2> file
```

Redirect the stderr of command `ls` to *file* (in this case, write any error encountered by the command `ls` to *file*)

```
ls 2>> file
```

Append the stderr of command `ls` to *file*

```
ls 2> /dev/null
```

Silence any error coming from the command `ls`

```
cat <file1 >file2
<file1 cat >file2
<file1 >file2 cat
```

Redirect *file1* to the stdin and *file2* to the stdout of the command `cat` (in this case, copy *file1* to *file2*).
`cat >file2 <file1` also works but is not recommended, because it truncates *file2* if *file1* cannot be opened

```
cat /etc/passwd | wc -l
```

Pipe the stdout of command `cat` to the stdin of command `wc` (in this case, print the number of accounts in the system).
Piped commands run concurrently

```
echo "$(sort file)" > file
echo "`sort file`" > file
sort file | sponge file
```

Sort the contents of *file* and write the output to the file itself.
`sort file > file` would not produce the desired result, because the stdout destination is created (and therefore the content of the preexisting *file* is deleted) before the `sort` command is run

```
ls 2>&1
```

Redirect stderr of command `ls` to stdout

```
ls > file 2>&1
```

Redirect both stdout and stderr of command `ls` to *file*.
`ls &> file` and `ls >& file` also work on some systems but are not recommended, because they are not POSIX standard

```
> file
```

Create an empty file. If the file exists, its content will be deleted

```
ls | tee file
```

`tee` reads from stdin and writes both to stdout and *file* (in this case, writes the contents of the current directory to screen and to *file* at the same time)

```
ls | tee -a file
```

`tee` reads from stdin and appends both to stdout and *file*

```
read MYVAR
```

Read a variable from standard input

```
read -n 8 MYVAR
```

Read only max 8 chars from standard input

```
read -t 60 MYVAR
```

Read a variable from standard input, timing out after one minute

```
read -s MYVAR
```

Read a variable from standard input without echoing to terminal (silent mode)

```
while read -r line
do
   echo "Hello $line"
done < file
```

Process a text file line by line, reading from *file*, and output the lines.
If *file* is `/dev/stdin`, reads from standard input instead

```
while read line
do
   for word in $line
   do
      echo "Hello $word"
   done
done < file
```

Process a text file containing multiple words in each line, and output the words

```
while IFS=$'\t' read -r -a array
do
   echo "${array[0]}"
   echo "${array[1]}"
   echo "${array[2]}"
done < file
```

Process a text file containing three words per line separated by a tab, and output the words. Example of input file:

```
aaaa    bbb     ccc
dd      eeeee   ff
ggg     hhh     iiii
```

```
echo $MYVAR
```

Print a variable on screen

```
echo -n "message"
printf "message"
```

Print *message* onscreen without a trailing line feed

```
echo -e '\a'
```

Produce an alert sound (BEL sequence)

```
pv -qL10 <<< "message"
```

Print *message* onscreen, one character at a time

Any application, program, script, or service that runs on the system is a **process**. Processes whose parent is a shell are called **jobs**.
Signals are used for inter-process communication. Each process has a unique PID (Process ID) and a PPID (Parent Process ID); when a process spawns a child, the process PID is assigned to the child's PPID.
The `/sbin/init` process, run at bootup, has PID 1. It is the ancestor of all processes and becomes the parent of any orphaned process. It is also unkillable; should it die, the kernel will panic.
When a child process dies, its status becomes EXIT_ZOMBIE and a SIGCHLD is sent to the parent. The parent should then call the `wait()` system call to read the dead process' exit status and other information; until that moment, the child process remains a zombie.

`ps -ef` (UNIX options) `ps aux` (BSD options)	List all processes
`pstree` *`PID`*	Display all processes in hierarchical format. The process tree is rooted at *PID*, or at `init` if *PID* is omitted
`pidof` *`processname`*	Show PIDs of processes with name *processname*
`pidof -s` *`processname`*	Show PID of process with name *processname*, returning a single result

`pgrep sshd`
`ps -ef | grep "[s]shd"`
	Show processes whose name is "sshd"

`pgrep -u root sshd`	Show processes whose name is "sshd" and are owned by root
`pmap` *`PID`*	Display the memory map of process *PID*

`kill -9 1138`	Send a signal 9 (SIGKILL) to process 1138, hence killing it
`killall -9 sshd`	Kill processes whose name is "sshd"
`pkill -9 -u root sshd`	Kill processes whose name is "sshd" and are owned by root
`pkill -9 -u` *`user`*	Kill all processes owned by *user*, forcing him to log out
`skill`	Send a signal to a process or show process status. Obsolete
`xkill`	Kill a process by its X GUI resource. Pops up a cursor to select a window

`jobs`	List all jobs
CTRL Z	Suspend a job, putting it in the stopped state (send a SIGTSTP)
`bg %`*`n`*	Put job # *n* in the background (send a SIGCONT)
`fg %`*`n`*	Resume job # *n* in the foreground and make it the current job (send a SIGCONT)
`kill %`*`n`*	Kill job # *n*

`disown %`*`n`*	Remove job #*n* from the table of active jobs
`disown -h %`*`n`*	Prevent job #*n* from receiving a SIGHUP if the shell receives that signal

To each process is associated a niceness value: the higher the niceness, the lower the priority.
The niceness value ranges from -20 to 19, and a newly created process has a default niceness of 0.
Unprivileged users can modify a process' niceness only within the range from 1 to 19.

`nice -n -5` *`command`*	Start *command* with a niceness of -5. If niceness is omitted, a default value of 10 is used
`renice -5` *`command`*	Change the niceness of a running *command* to -5
`snice`	Change the niceness of a process. Obsolete

Most frequently used signals		
Signal number	**Signal name**	**Effect**
1	SIGHUP	Used by many daemons to reload their configuration
2	SIGINT	Interrupt, stop
9	SIGKILL	Kill unconditionally (this signal cannot be ignored)
15	SIGTERM	Terminate gracefully
18	SIGCONT	Continue execution
20	SIGTSTP	Stop execution

The manpage `man 7 signal` lists all signal numbers and names.

`kill -l`	List all available signal names
`kill -l n`	Print the name of signal number *n*
`trap action condition`	Trap a signal
`strace command`	Trace the execution of *command*, intercepting and printing system calls called by a process and signals received by a process
`ipcs`	Show IPC facilities information (shared memory, message queues, and semaphores)
`:(){ :|:& };:`	Fork bomb: starts a process that continually replicates itself, slowing down or crashing the system because of resource starvation. Dangerous!
`( command )& pid=$!; sleep n; kill -9 $pid`	Run *command* and kill it after *n* seconds

`top`	Monitor processes in real-time
`htop`	Monitor processes in real-time (Ncurses UI)
`iotop`	Display I/O usage by processes in the system
`atop`	Advanced system monitor that displays the load on CPU, RAM, disk, and network
`powertop`	Power consumption and power management diagnosis tool
`uptime`	Show how long the system has been up, how many users are connected, and the system load averages for the past 1, 5, and 15 minutes
`time` *command*	Execute *command* and, at its completion, write to stderr timing statistics about the run: elapsed real time between invocation and termination, user CPU time, system CPU time
`sar`	Show reports about system activity (including reboots). Reports are generated from data collected via the cron job `sysstat` and stored in `/var/log/sa/s`*n*, where *n* is the day of the month

`sar -f /var/log/sa/sa13 \`
`-s 06:00:00 -e 09:00:00`
	Show reports for system activity from 6 to 9 AM on the 13th of the month

`sar -u` *n m*	Show real-time CPU activity, every *n* seconds for *m* times
`sar -n DEV`	Show real-time network activity (received and transmitted packets per second)
`sysbench`	Multi-threaded benchmark tool able to monitor different OS parameters: file I/O, scheduler, memory allocation, thread implementation, databases
`inxi`	Debugging tool to rapidly and easily gather system information and configuration
`stress-ng`	Tool for CPU and RAM stress tests

Linux monitoring tools	
collectd	System statistics collector
Nagios	System monitor and alert
MRTG	Network load monitor
Cacti	Network monitor
Munin	System and network monitor and alert
Zabbix	System and network monitor and alert
Centreon	System and network monitor and alert
netdata	Real-time performance and health monitor

`vmstat` Print a report about virtual memory statistics: processes, memory, paging, block I/O, traps, disks, and CPU activity

`iostat` Print a report about CPU utilization, device utilization, and network filesystem.
The first report shows statistics since the system boot; subsequent reports will show statistics since the previous report

`mpstat` Print a report about processor activities

`vmstat` *`n m`*
`iostat` *`n m`*
`mpstat` *`n m`*
Print the relevant report every *n* seconds for *m* times

Output of command `vmstat`

```
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0 296724 267120 3393400    0    0    17    56    0    3  2  2 95  1  0
```

procs	**r**	Number of runnable processes (running or waiting for run time)	
	b	Number of processes in uninterruptible sleep	
memory	**swpd**	Virtual memory used (swap)	in Kb
	free	Free memory (idle)	
	buff	Memory used as buffers	
	cache	Memory used as cache	
swap	**si**	Memory swapped in from disk	in Kb/second
	so	Memory swapped out to disk	
io	**bi**	Blocks received in from a block device	in blocks/second
	bo	Blocks sent out to a block device	
system	**in**	Number of interrupts	per second
	cs	Number of context switches	
cpu	**us**	Time spent running user code (non-kernel)	in percentage of total CPU time
	sy	Time spent running system code (kernel)	
	id	Time spent idle	
	wa	Time spent waiting for I/O	
	st	Time stolen from a virtual machine	

`free`	Show the amount of free and used memory in the system

Output of command `free`

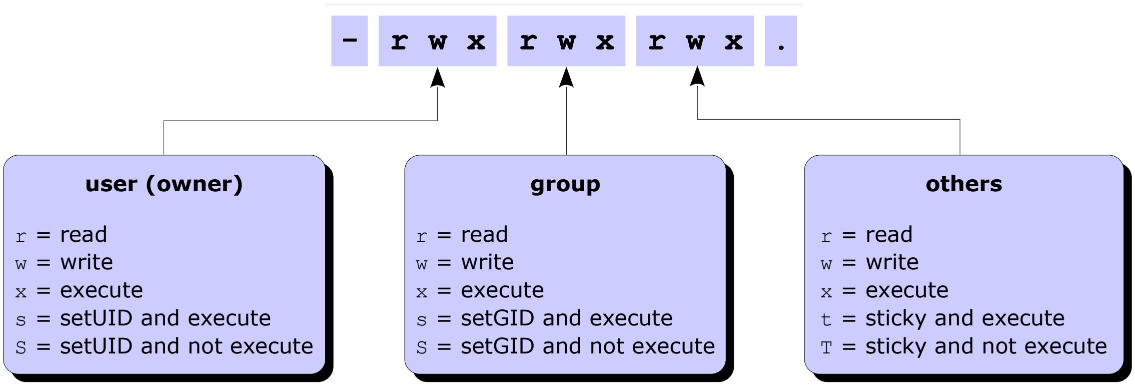
```
              total        used        free      shared  buff/cache   available
Mem:       16344088     2273312    11531400      776228     2539376    12935112
Swap:       1048572           0     1048572
```

```
             total       used       free     shared    buffers     cached
Mem:       1504544    1491098      13021          0      91112     764542
-/+ buffers/cache:     635212     869498
Swap:      2047686       7667    2040019
```

Mem	**total**	Total configured amount of memory
	used	Used memory
	free	Unused memory
	shared	Memory used by tmpfs, 0 if not available
	buff/cache	Memory used by kernel buffers, page cache, and slabs
	available	Memory available for new applications (without using swap) *
-/+ buffers/cache	**used**	Memory used by kernel buffers
	free	Memory available for new applications (without using swap) *
Swap	**total**	Total configured amount of swap space
	used	Used swap space
	free	Free swap space *

* These are the true values indicating the free system resources available.
All values are in Kb, unless options are used.

File permissions



- r w x r w x r w x .

user (owner)

r = read
w = write
x = execute
s = setUID and execute
S = setUID and not execute

group

r = read
w = write
x = execute
s = setGID and execute
S = setGID and not execute

others

r = read
w = write
x = execute
t = sticky and execute
T = sticky and not execute

Permission	Octal value	Command	Effect on file	Effect on directory
Read	user: 400	`chmod u+r`	Can open and read the file	Can list directory content
	group: 40	`chmod g+r`		
	others: 4	`chmod o+r`		
Write	user: 200	`chmod u+w`	Can modify the file	Can create, delete, and rename files in the directory
	group: 20	`chmod g+w`		
	others: 2	`chmod o+w`		
Execute	user: 100	`chmod u+x`	Can execute the file (binary or script)	Can enter the directory, and search files within (by accessing a file's inode)
	group: 10	`chmod g+x`		
	others: 1	`chmod o+x`		
SetUID (SUID)	4000	`chmod u+s`	Executable is run with the privileges of the file's owner	No effect
SetGID (SGID)	2000	`chmod g+s`	Executable is run with the privileges of the file's group	All new files and subdirectories inherit the directory's group ID
Sticky	1000	`chmod +t`	No effect	Files inside the directory can be deleted or moved only by the file's owner

`chmod 711 file`
`chmod u=rwx,go=x file`
	Set read, write, and execute permission to user; set execute permission to group and others

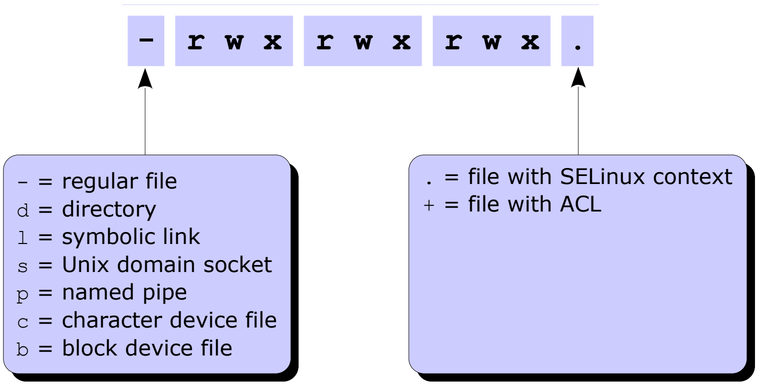
`chmod u+wx file`	Add write and execute permission to user
`chmod -x file`	Remove execute permission from everybody (user, group, and others)
`chmod -R g+x /path`	Set the group execute bit recursively on *path* and every dir and file underneath

`find /path -type d \`
`-exec chmod g+x {} \;`
	Set the group execute bit recursively on *path* and every dir, but not file, underneath

`chown user file`	Change the owner of the file to *user*
`chown user:group file`	Change the owner of the file to *user*, and group ownership of the file to *group*

`chown :group file`
`chgrp group file`
	Change group ownership of the file to *group*

`umask 022`	Set the permission mask to 022, hence masking write permission for group and others. Linux default permissions are 0666 for files and 0777 for directories. These base permissions are ANDed with the inverted umask value to calculate the final permissions of a new file or directory



- = regular file
d = directory
l = symbolic link
s = Unix domain socket
p = named pipe
c = character device file
b = block device file

. = file with SELinux context
+ = file with ACL

Attribute	Effect
`a`	File can only be opened in append mode for writing
`A`	When file is accessed, its atime record is not modified
`c`	File is automatically compressed on-the-fly on disk by the kernel
`C`	File is not subject to copy-on-write updates. This applies only to filesystems which perform copy-on-write
`d`	File will not be backed up by the `dump` program
`D`	When directory is modified, changes are written synchronously on disk. Equivalent to `dirsync` mount option
`e`	File is using extents for mapping the blocks on disk
`E`	Compression error on file. This attribute is used by experimental compression patches
`h`	File stores its blocks in units of filesystem blocksize instead of in units of sectors, and is larger than 2 Tb
`i`	File is immutable i.e. cannot be modified, linked, or changed permissions
`I`	Directory is being indexed using hashed trees
`j`	All file data is written to the ext3 or ext4 journal before being written to the file itself
`N`	File has data stored inline within the inode itself
`s`	File will be securely wiped by zeroing when deleted
`S`	When file is modified, changes are written synchronously on disk. Equivalent to the `sync` mount option
`t`	File will not have EOF partial block fragment merged with other files. This applies only to filesystems with support for tail-merging
`T`	Directory is the top of directory hierarchies for the purpose of the Orlov block allocator
`u`	After file is deleted, it can be undeleted
`X`	Raw contents of compressed file can be accessed directly. This attribute is used by experimental compression patches
`Z`	Compressed file is dirty. This attribute is used by experimental compression patches

`chattr +`*`attribute file`*	Add a file or directory attribute
`chattr -`*`attribute file`*	Remove a file or directory attribute
`chattr =`*`attribute file`*	Set a file or directory attribute, removing all other attributes
`lsattr `*`file`*	List file or directory attributes

Timestamp	Value tracked	Displayed via
mtime	Time of last **modification** to file contents (data itself)	`ls -l`
ctime	Time of last **change** to file contents or file metadata (owner, group, or permissions)	`ls -lc`
atime	Time of last **access** to file for reading contents	`ls -lu`

The POSIX standard does not define a timestamp for file **creation**. Some filesystems (e.g. ext4, JFS, Btrfs) store this value, but currently there is no Linux kernel API to access it.

Access Control Lists (ACLs) provide a fine-grained set of permissions that can be applied to files and directories.
An **access ACL** is set on an individual file or directory; a **default ACL** is set on a directory, and applies to all files and subdirs created inside it that don't have an access ACL.
The final permissions are the intersection of the ACL with the chmod/umask value.
A partition must have been mounted with the `acl` option in order to support ACLs on files.

`setfacl -m u:`*`user`*`:`*`permissions file`*	Set an access ACL on a file for an user
`setfacl -m g:`*`group`*`:`*`permissions file`*	Set an access ACL on a file for a group
`setfacl -m m:`*`permissions file`*	Set the effective rights mask on a file
`setfacl -m o:`*`permissions file`*	Set the permissions on a file for other users
`setfacl -x u:`*`user file`*	Remove an access ACL from a file for an user
`setfacl -x g:`*`group file`*	Remove an access ACL from a file for a group

The *permissions* are standard Unix permissions specified as any combination of `r w x`.

`setfacl -m d:u:`*`user`*`:`*`permissions dir`*
`setfacl -d -m u:`*`user`*`:`*`permissions dir`*
	Same as above, but set a default ACL instead of an access ACL.
	This applies to all commands above

`getfacl `*`file`*	Display the access (and default, if any) ACL for a file
`getfacl `*`file1`*` | setfacl --set-file=- `*`file2`*	Copy the ACL of *file1* and apply it to *file2*
`getfacl --access `*`dir`*` | setfacl -d -M- `*`dir`*	Copy the access ACL of a directory and set it as default ACL
`chacl `*`options`*	Change an ACL. This command exists to provide compatibility with IRIX
`man acl`	Show the manpage about ACLs

Links

An **inode** is a structure containing all file metadata: file type, permissions, owner, group, size, access/change/modification/deletion times, number of links, attributes, ACLs, and address where the actual file content (data) is stored. However, an inode does not contain the name of the file; this information is stored in the directory where the file is located (i.e. referenced).
A directory contains a list of mappings between filenames and inodes.

In Linux, there are two kinds of links: **hard links** and **symbolic links** (aka **soft links**).

The **link count** of a file is the total number of hard links to that file (i.e. to that file's inode). By default, files have a link count of 1, and directories have a link count of 2 (the directory itself, and the `.` link inside the directory). The link count of a directory is increased by one for each subdirectory (because of the `..` parent link inside the subdirectory). Once a file has no hard links pointing to it, the file is deleted, provided that no process holds the file open for reading.

	Hard link	**Symbolic link**
Definition	A link to an already existing inode	A path to a filename; a shortcut
Command to create it	`ln` *`file hardlink`*	`ln -s` *`file symlink`*
Link is still valid if the original file is moved or deleted	Yes (because the link still references the inode to which the original file pointed)	No (because the path now references a non-existent file)
Can link to a file in another filesystem	No (because inode numbers make sense only within a determinate filesystem)	Yes
Can link to a directory	No	Yes
Link permissions	Reflect the original file's permissions, even when these are changed	`rwxrwxrwx`
Link attributes	`-` (regular file)	`l` (symbolic link)
Inode number	The same as the original file	A different inode number (since it's a different file)

`ls -i`	Show a listing of the directory with the inode number for each file
`ls -l`	Show a listing of the directory with the link count for each file
`df -i`	Report filesystem inode usage
`find / -inum` *`n`*	Find all files linked to the same inode *n*
`find / -samefile` *`file`*	Find all files linked to the same inode as *file*

`find /path -name "foo*"`
`find /path -name "foo*" -print`
	Find all files and dirs, in the directory tree rooted at */path*, whose name starts with "foo"

`find / -name "foo*" -exec chmod 700 {} \;`	Find all files and dirs whose name start with "foo" and apply permission 700 to all of them
`find / -name "foo*" -ok chmod 700 {} \;`	Find all files and dirs whose name start with "foo" and apply permission 700 to all of them, asking for confirmation
`find / -size +128M`	Find all files larger than 128 Mb
`find / -type f -ctime +10`	Find all files last changed more than 10 days ago
`find / -type f -perm -4000`	Find all files with SUID set (a possible security risk, because a shell with SUID root is a backdoor)
`find / -type f -newermt "May 4 2:55" -delete`	Find and delete all files newer than the specified timestamp. Using `-delete` is preferable to using `-exec rm {} \;`
`find . -type f -print -exec cat {} \;`	Print all files, in the current directory and under, prepending them with a filename header
`find . \! -name "*.gz" -type f -exec gzip {} \;`	Find all files, in the current directory and under, which do not have the `gz` extension, and compress them

`find / -xdev -type f -size +100M \`
`-exec ls -lah {} \;`
	Find all files larger than 100 Mb in the current filesystem only and display detailed information about them

`locate file`
`slocate file`
	Locate *file* by searching the file index `/etc/updatedb.conf`, not by actually walking the filesystem. The search is fast but will only held results relative to the last rebuild of the file index

`updatedb`	Rebuild the file index
`which command`	Locate a binary executable command within the PATH
`which -a command`	Locate all matches of a command, not only the first one
`whereis command`	Locate the binary, source, and manpage files for a command
`whereis -b command`	Locate the binary files for a command
`whereis -s command`	Locate the source files for a command
`whereis -m command`	Locate the manpage files for a command
`type command`	Determine if a command is a program or a built-in (i.e. an internal feature of the shell)
`file file`	Analyze the content of a file or directory, and display the kind of file (e.g. executable, text file, program text, swap file)

The scope of **variables** is the current shell only, while **environment variables** are visible within the current shell as well as within all subshells and Bash child processes spawned by the shell.
Environment variables are set in `/etc/environment` in the form *`variable=value`*.
Conventionally, variable names are lowercase while environment variable names are uppercase.

`set`	Display all variables
`env`	Display all environment variables
`readonly -p`	Display all variables that are read-only

`VAR=value`
`((VAR=value))`
`let "VAR=value"`
	Set the value of a variable.
	There must be no spaces around the `=` sign. It is possible to add space around `((` and `))`

`readonly VAR=value`	Set a variable making its value unchangeable

`set ${VAR:=value}`
`VAR=${VAR:-value}`
	Set a variable only if it is not already set (i.e. does not exist) or is null

`unset VAR`	Unset (i.e. delete) a variable
`export VAR`	Export a variable, making it an environment variable

`command $VAR`
`command ${VAR}HELLO`
`command "${VAR}"`
	Pass a variable as argument to *command*.
	If other characters follow the variable name, it is necessary to specify the boundaries of the variable name via `{}` to make it unambiguous.
	It is recommended to double quote the variable when referencing it, to prevent interpretation of special characters (except `` \ $ ` ``) and word splitting (in case the variable value contains whitespaces), both of which will have unintended results

`VAR=$((5 + 37))`
`VAR=$[5 + 37]`
`VAR=$((VAR2 + 42))`
`` VAR=`expr $VAR2 + 42` ``
	Evaluate a numeric expression and assign the result to another variable

`((VAR++))`
`((++VAR))`
`((VAR+=1))`
`((VAR=VAR+1))`
	Increase a variable by 1

`` VAR=`command` ``
`VAR=$(command)`
	Command substitution. Assign to a variable the standard output resulting from *command* (which is executed in a subshell)

`for i in /path/*`
`do`
`   echo "Filename: $i"`
`done`
	Loop and operate through all the output tokens (in this case, files in the *path*).
	The equivalent construct `for i in $(ls /path/)` is unnecessary and harmful, because filenames containing whitespaces or glob characters will cause unintended results

`echo ${VAR:-message}`	If variable exists and is not null, print its value, otherwise print *message*
`echo ${VAR:+message}`	If variable exists and is not null, print *message*, otherwise print nothing
`echo ${VAR,,}`	Print a string variable in lowercase
`TOKENS=($STRING)`	String tokenizer. Splits a string stored in the variable *STRING* into tokens, according to the content of the shell variable `$IFS`, and stores them in the array *TOKENS*
`echo ${TOKENS[n]}`	Print the token number *n*
`echo ${TOKENS[*]}`	Print all tokens

Bash built-in variables	
`$0`	Script name
`$`*n*	*n*th argument passed to the script or function
`$@`	All arguments passed to the script or function; each argument is a separate word
`$*`	All arguments passed to the script or function, as a single word
`$#`	Number of arguments passed to the script or function
`$?`	Exit status of the last recently executed command
`${PIPESTATUS[`*n*`]}`	Exit status of the *n*th command in the executed pipeline
`$$`	PID of the script in which this variable is called
`$!`	PID of the last recently executed background command
`$SHLVL`	Deepness level of current shell, starting with 1
`$IFS`	Internal Field Separator; defines what are the token separators for strings (e.g. for word splitting after expansion). By default it has the value "space, tab, newline"
`$RANDOM`	Pseudorandom integer value between 0 and 32767

Bash shell event	Files run	

When a login shell is launched	`/etc/profile`
	`/etc/profile.d/*.sh`
	`~/.bash_profile`
	`~/.bash_login`
	`~/.profile`
		The shell executes the system-wide profile files, then the first of the 3 user files that exists and is readable

When a login shell exits	`~/.bash_logout`	

When a non-login shell is launched	`/etc/bash.bashrc`
	`/etc/bashrc`
	`~/.bashrc`	

set -*option* set -o *longoption*	Enable a Bash option
set +*option* set +o *longoption*	Disable a Bash option
set -o	Show the status of all Bash options
set -v set -o verbose	Print shell input lines as they are read
set -x set -o xtrace	Print command traces before execution of each command (debug mode)
set -e set -o errexit	Exit the script immediately if a command fails. Recommended option
set -u set -o nounset	Treat expansion of unset variables as an error. This avoids unintended results

There are three ways to run a script with a specific Bash option enabled:
- Run the script with `bash -`*option script.sh*
- Specify the shebang line in the script as `#!/bin/bash -`*option*
- Add the command `set -`*option* at the beginning of the script

shopt	Display the list of all shell options with their current value (on or off)
shopt -s *shelloption*	Set (enable) a specific shell option
shopt -u *shelloption*	Unset (disable) a specific shell option

Bash shell scripts must start with the shebang line `#!/bin/bash` indicating the location of the script interpreter.

Script execution	
source *script.sh* . *script.sh*	Script execution takes place in the same shell. Variables defined and exported in the script are seen by the shell when the script exits
bash *script.sh* ./*script.sh* (file must be executable)	Script execution spawns a new shell

command &	Execute *command* in the background
command1; *command2*	Execute *command 1* and then *command 2*
command1 && *command2*	Execute *command 2* only if *command 1* executed successfully (exit status = 0)
command1 || *command2*	Execute *command 2* only if *command 1* did not execute successfully (exit status > 0)
(*command1* && *command2*)	Group commands together for evaluation priority
(*command*)	Run *command* in a subshell. This is used to isolate *command*'s effects, as variable assignments and other changes to the shell environment operated by *command* will not remain after *command* completes
exit	Terminate a script
exit *n*	Terminate a script with the specified exit status number *n*. By convention, a 0 exit status is used if the script executed successfully, a non-zero value otherwise
command || exit 1	(To be used inside a script.) Exit the script if *command* fails
/bin/true	Do nothing and return immediately a status code of 0 (indicating success)
/bin/false	Do nothing and return immediately a status code of 1 (indicating failure)

if *command*
then echo "Success"
else echo "Failure"
fi
	Run a command, then evaluate whether it exited successfully or failed

function myfunc { *commands* }
myfunc() { *commands* }
	Define a function. A function must be defined before it can be used in a Bash script.
	Argument number *n* is accessed in the body of the function via $*n*.
	An advantage of functions over aliases is that functions can be passed arguments

myfunc *arg1 arg2* ...	Call a function
readonly -f myfunc	Mark an already defined function as read-only, preventing it to be redefined
typeset -f	Show functions defined in the current Bash session
readonly -p -f	Show functions which are read-only
expect	Dialogue with interactive programs according to a script, analyzing what can be expected from the interactive program and replying accordingly
zenity	Display GTK+ graphical dialogs for user messages and input

`getopts` Parse positional parameters in a shell script

`getopts` **syntax**	

`while getopts abc:d: `*`OPT`*
`do`
`   case $`*`OPT`*` in`
	Definition of accepted options

`      a)`
`         `*`command_a`*
`         exit 0`
`         ;;`
	Matches option `-a`.
	Executes a command

`      b)`
`         `*`command_b`*
`         exit 0`
`         ;;`	

`      c)`
`         `*`command_c`*` $OPTARG`
`         exit 0`
`         ;;`
	Matches option `-c` *`argument`*.
	Executes a command with argument

`      d)`
`         `*`command_d`*` $OPTARG`
`         exit 0`
`         ;;`	

`      *)`
`         `*`default_command`*
`         exit 1`
`         ;;`
	Command to execute if none of above options applies

`   esac`
`done`	

`watch` *command*	Execute *command* every 2 seconds
`watch -d -n 1` *command*	Execute *command* every second, highlighting the differences in the output
`timeout 30s` *command*	Execute *command* and kill it after 30 seconds
command `| ts`	Prepend a timestamp to each line of the output of *command*
`sleep 5`	Pause for 5 seconds
`sleep $[($RANDOM % 60) + 1]s`	Sleep for a random time between 1 and 60 seconds
`sleep infinity`	Pause forever
`usleep 5000`	Pause for 5000 microseconds
`yes`	Output endlessly the string "y"
`yes` *string*	Output endlessly *string*
`yes | fsck /dev/sda`	Automatically answer yes every time `fsck` asks for confirmation before fixing errors
`script` *file*	Generate a typescript of a terminal session. Forks a subshell and starts recording on *file* everything that is printed on terminal; the typescript ends when the user exits the subshell
`xargs` *command*	Call *command* multiple times, one for each argument found on stdin
`ls foo* | xargs cat`	Print via `cat` the content of every file whose name starts by "foo"
`parallel` *command*	Run *command* in parallel. This is used to operate on multiple inputs, similarly to `xargs`

`test "$MYVAR"` *operator* `"`*value*`" &&` *command* `[ "$MYVAR"` *operator* `"`*value*`" ] &&` *command* `if [ "$MYVAR"` *operator* `"`*value*`" ]; then` *command*`; fi`	Perform a test; if it results true, *command* is executed

Test operators

Integer operators	
`-eq` *value*	Equal to
`-ne` *value*	Not equal to
`-lt` *value*	Less than
`-le` *value*	Less than or equal to
`-gt` *value*	Greater than
`-ge` *value*	Greater than or equal to

Numeric operators	
`=` *value*	Equal to
`!=` *value*	Not equal to
`<` *value*	Less than
`<=` *value*	Less than or equal to
`>` *value*	Greater than
`>=` *value*	Greater than or equal to

Expression operators	
expr1 `-a` *expr2*	Logical AND
expr1 `-o` *expr2*	Logical OR
`!` *expr*	Logical NOT
`\(` *expr* `\)`	Priority

String operators	
`-z`	Is zero length
`-n` or nothing	Is non-zero length
`=` or `==` *string*	Is equal to
`!=` *string*	Is not equal to
`<` *string*	Is alphabetically before
`>` *string*	Is alphabetically after
`substr` *string pos len*	Substring
`index` *string chars*	Index of any chars in string
`length` *string*	String length
string `:` *regex* or `match` *string regex*	String matches regex

File operators	
`-e` or `-a` *file*	Exists
`-f` *file*	Is a regular file
`-d` *file*	Is a directory
`-b` *file*	Is a block special file
`-c` *file*	Is a character special file
`-r` *file*	Is readable
`-w` *file*	Is writable
`-x` *file*	Is executable
`-k` *file*	Is sticky
`-u` *file*	Is SUID
`-g` *file*	Is SGID
`-O` *file*	Is owned by the Effective UID
`-G` *file*	Is owned by the Effective GID
`-p` *file*	Is a named pipe (aka FIFO)
`-S` *file*	Is a socket
`-h` or `-L` *file*	Is a symbolic link
`-s` *file*	Is non-zero length
`-N` *file*	Was modified since last read
file1 `-nt` *file2*	Is newer than
file1 `-ot` *file2*	Is older than
file1 `-ef` *file2*	Refer to same device and inode as

`expr "$MYVAR" = "39 + 3"`	Evaluate an expression (in this case, assigns the value 42 to the variable)
`expr` *string* `:` *regex*	Return the length of the substring matching the regex
`expr` *string* `: \(`*regex*`\)`	Return the substring matching the regex

<table>
<tr><th colspan="4">Operators</th></tr>
<tr><th colspan="2">Mathematical operators</th><th colspan="2">Logical operators</th></tr>
<tr><td>+</td><td>Addition</td><td>!</td><td>Logical negation</td></tr>
<tr><td>-</td><td>Subtraction</td><td>&&</td><td>Logical AND</td></tr>
<tr><td>*</td><td>Multiplication</td><td>||</td><td>Logical OR</td></tr>
<tr><td>/</td><td>Division</td><th colspan="2">Bitwise operators</th></tr>
<tr><td>%</td><td>Remainder</td><td>~</td><td>Bitwise negation</td></tr>
<tr><td>**</td><td>Exponentiation</td><td>&</td><td>Bitwise AND</td></tr>
<tr><td>++</td><td>Pre/post increment</td><td>|</td><td>Bitwise OR</td></tr>
<tr><td>--</td><td>Pre/post decrement</td><td>^</td><td>Bitwise XOR</td></tr>
<tr><th colspan="2">Assignment operators</th><td><<</td><td>Left bitwise shift</td></tr>
<tr><td>=</td><td>Assignment</td><td>>></td><td>Right bitwise shift</td></tr>
<tr><td>*op*=</td><td>Operation and assignment</td><td></td><td></td></tr>
</table>

Tests

```
if [test 1]
then
   [command block 1]
elif [test 2]
then
   [command block 2]
else
   [command block 3]
fi
```

```
case $STRING in
   pattern1)
      [command block 1]
      ;;
   pattern2)
      [command block 2]
      ;;
   *)
      [command block default]
      ;;
esac
```

Loops

```
while [test]
do
   [command block]
done
```

The *command block* executes as long as *test* is true

```
until [test]
do
   [command block]
done
```

The *command block* executes as long as *test* is false

```
for item in [list]
do
   [command block]
done
```

The *command block* executes for each *item* in *list*

```
i=0
while [ $i -le 7 ]
do
   echo $i
   let i++
done
```

```
i=0
until [ $i -gt 7 ]
do
   echo $i
   let i++
done
```

```
for i in 0 1 2 3 4 5 6 7
do
   echo $i
done
```

```
for i in {0..7}
do
   echo $i
done
```

```
start=0
end=7
for i in $(seq $start $end)
do
   echo $i
done
```

```
start=0
end=7
for ((i = start; i <= end; i++))
do
   echo $i
done
```

`break`	Exit a loop
`continue`	Jump to the next iteration

`vi`	Vi, text editor
`vim`	Vi Improved, an advanced text editor
`gvim`	Vim with GUI
`vimdiff` *`file1 file2`*	Compare two text files in Vim

`pico`	Pico, simple text editor
`nano`	Nano, simple text editor (a GNU clone of Pico)
`rnano`	Restricted version of Nano: does not allow the user access the filesystem (except for files specified as argument) or a command shell

`emacs`	GNU Emacs, a GUI text editor
`gedit`	GUI text editor

`ed`	Line-oriented text editor

`hexedit`	Hexadecimal and ASCII editor

`more`	Text pager (obsolete)
`less`	Text pager
`most`	Text pager with advanced features (screen split, binary viewer, etc.)

`g`	Go to the first line in the file
`ng`	Go to line number *n*
`G`	Go to the last line in the file
`F`	Go to the end of the file, and move forward automatically as the file grows
CTRL C	Stop moving forward
`-N`	Show line numbers
`-n`	Don't show line numbers
`=`	Show information about the file
CTRL G	Show current and total line number, byte, and percentage of the file read
`/`*`pattern`*	Search *pattern* forward
`?`*`pattern`*	Search *pattern* backwards
`&`*`pattern`*	Display only lines matching *pattern*
`n`	Search next occurrences forward
`N`	Search next occurrences backwards
`:n`	When reading multiple files, go to the next file
`:p`	When reading multiple files, go to the previous file
`R`	Repaint the screen
`V`	Show version number
`h`	Help
`q`	Quit

`less +`*`command file`*	Open *file* for reading, applying *command* (see list above)
`less +F --follow-name` *`file`*	Move forward, attempting periodically to reopen *file* by name; useful to keep reading a logfile that is being rotated. Note that, by default, `less` continues to read the original input file even if it has been renamed

ESC	Go to Command mode		
i	Insert text before cursor	and go to Insert mode	
I	Insert text after line		
a	Append text after cursor		
A	Append text after line		
v	Go to Visual mode, character-wise	then use the arrow keys to select a block of text	
V	Go to Visual mode, line-wise		
d	Delete selected block	gu	Switch block to lowercase
y	Copy (yank) selected block into buffer	gU	Switch block to uppercase
w	Move to next word	$	Move to end of line
b	Move to beginning of word	1G	Move to line 1 i.e. beginning of file
e	Move to end of word	G	Move to end of file
0	Move to beginning of line	z RETURN	Make current line the top line of the screen
CTRL G	Show current line and column number		
ma	Mark position "a". Marks a-z are local to current file, while marks A-Z are global to a specific file		
'a	Go to mark "a". If using a global mark, it also opens the specific file		
y'a	Copy (yank) from mark "a" to current line, into the buffer		
d'a	Delete from mark "a" to current line		
p	Paste buffer after current line	yy	Copy current line
P	Paste buffer before current line	yyp	Duplicate current line
x	Delete current character	D	Delete from current character to end of line
X	Delete before current character	dd	Delete current line
7dd	Delete 7 lines. Almost any command can be prepended by a number to repeat it that number of times		
u	Undo last command. Vi can undo the last command only, Vim is able to undo several commands		
.	Repeat last text-changing command		
/*string*	Search for *string* forward	n	Search for next match of *string*
?*string*	Search for *string* backwards	N	Search for previous match of *string*
:s/*s1*/*s2*/	Replace the first occurrence of *s1* with *s2* in the current line		
:s/*s1*/*s2*/g	Replace globally every occurrence of *s1* with *s2* in the current line		
:%s/*s1*/*s2*/g	Replace globally every occurrence of *s1* with *s2* in the whole file		
:%s/*s1*/*s2*/gc	Replace globally every occurrence of *s1* with *s2* in the whole file, asking for confirmation		
:5,40s/^/#/	Add a hash character at the beginning of each line, from line 5 to 40		
!!*program*	Replace line with output from *program*		
:r *file*	Read *file* and insert it after current line		
:X	Encrypt current document. Vi will automatically prompt for the password to encrypt and decrypt		
:w *file*	Write to *file*		
:wq :x ZZ	Save changes and quit		
:q	Quit (fails if there are unsaved changes)	:q!	Abandon all changes and quit

vi -R *file*	Open *file* in read-only mode
cat *file* | vi -	Open *file* in read-only mode (this is done by having Vi read from stdin)

Option	Effect
ai	Turn on auto indentation
all	Display all options
ap	Print a line after the commands d c J m :s t u
aw	Automatic write on commands :n ! e# ^^ :rew ^} :tag
bf	Discard control characters from input
dir=*tmpdir*	Set *tmpdir* as directory for temporary files
eb	Precede error messages with a bell
ht=8	Set terminal tab as 8 spaces
ic	Ignore case when searching
lisp	Modify brackets for Lisp compatibility
list	Show tabs and EOL characters
set listchars=tab:>-	Show tab as > for the first char and as - for the following chars
magic	Allow pattern matching with special characters
mesg	Enable UNIX terminal messaging
nu	Show line numbers
opt	Speed up output by eliminating automatic Return
para=LIlPLPPPQPbpP	Set macro to start paragraphs for { } operators
prompt	Prompt : for command input
re	Simulate smart terminal on dumb terminal
remap	Accept macros within macros
report	Show the largest size of changes on status line
ro	Make file readonly
scroll=12	Set screen size as 12 lines
shell=/bin/bash	Set shell escape to /bin/bash
showmode	Show current mode on status line
slow	Postpone display updates during inserts
sm	Show matching parentheses when typing
sw=8	Set shift width to 8 characters
tags=/usr/lib/tags	Set path for files checked for tags
term	Print terminal type
terse	Print terse messages
timeout	Eliminate 1-second time limit for macros
tl=3	Set significance of tags beyond 3 characters (0 = all)
ts=8	Set tab stops to 8 for text input
wa	Inhibit normal checks before write commands
warn	Display the warning message "No write since last change"
window=24	Set text window as 24 lines
wm=0	Set automatic wraparound 0 spaces from right margin

:set *option*	turn on an *option*
:set no*option*	turn off an *option*
:set *option* ?	show the current value of *option*

Options can also be permanently set by including them in ~/.exrc (Vi) or ~/.vimrc (Vim)

`SHOW DATABASES;`	Show all existing databases
`USE CompanyDatabase;`	Select a database to use
`SELECT DATABASE();`	Show which database is currently selected
`DROP DATABASE CompanyDatabase;`	Delete a database
`SHOW TABLES;`	Show all tables from the selected database

`CREATE TABLE customers (`
`cusid INT NOT NULL AUTO_INCREMENT PRIMARY KEY,`
`firstname VARCHAR(32), lastname VARCHAR(32), dob DATE,`
`city VARCHAR(24), zipcode VARCHAR(5));`
	Create tables

`CREATE TABLE payments (`
`payid INT NOT NULL AUTO_INCREMENT PRIMARY KEY,`
`date DATE, fee INT, bill VARCHAR(128), cusid INT,`
`CONSTRAINT FK1 FOREIGN KEY (cusid) REFERENCES customers(cusid));`	

`INSERT INTO customers (firstname,lastname,dob)`
`VALUES ('Arthur','Dent',1959-08-01), ('Trillian','',1971-03-19);`
	Insert new records in a table

`DELETE FROM customers WHERE firstname LIKE 'Zaphod';`	Delete some records in a table
`UPDATE customers SET city = 'London' WHERE zipcode = 'L1 42HG';`	Modify records in a table

`CREATE INDEX lastname_index ON customers(lastname);`
`ALTER TABLE customers ADD INDEX lastname_index (lastname);`
	Create an index for faster searches

`DESCRIBE customers;`	Describe the columns of a table
`SHOW CREATE TABLE customers;`	Show the code used to create a table
`SHOW INDEXES FROM customers;`	Show primary key and indexes of a table
`DROP TABLE customers;`	Delete a table
`ALTER TABLE customers MODIFY city VARCHAR(32);`	Modify the type of a column

`CREATE VIEW cust_view AS`
`SELECT * FROM customers WHERE city != 'London';`
	Create a view. Views are used similarly to tables

`COMMIT;`	Commit changes to the database
`ROLLBACK;`	Rollback the current transaction, canceling any changes done during it

`START TRANSACTION;`
`BEGIN;`
	Disable autocommit for this transaction, until a COMMIT or ROLLBACK is issued

If no database has been selected for use, tables must be referenced by *`databasename.tablename`*.

SELECT * FROM customers;	Select all columns from the customers table
SELECT firstname, lastname FROM customers LIMIT 5;	Select first and last name of customers, showing 5 records only

SELECT firstname, lastname FROM customers LIMIT 1000,5;
SELECT firstname, lastname FROM customers OFFSET 1000 LIMIT 5;
	Select first and last name of customers, skipping the first 1000 records and showing 5 records only

SELECT firstname, lastname FROM customers WHERE zipcode = 'L1 42HG';	Select first and last name of customers whose zip code is "L1 42HG"
SELECT firstname, lastname FROM customers WHERE zipcode IS NOT NULL;	Select first and last name of customers with an existing zip code
SELECT * FROM customers ORDER BY lastname, firstname;	Select customers in alphabetical order by last name, then first name
SELECT * FROM customers ORDER by zipcode DESC;	Select customers, sorting them by zip code in reverse order

SELECT firstname, lastname,
TIMESTAMPDIFF(YEAR,dob,CURRENT_DATE) AS age FROM customers;
	Select first name, last name, and calculated age of customers

SELECT DISTINCT city FROM customers;	Show all cities, retrieving each unique output record only once
SELECT city, COUNT(*) FROM customers GROUP BY city;	Show all cities and the number of customers in each city. NULL values are not counted
SELECT cusid, SUM(fee) FROM payments GROUP BY cusid;	Show all fee payments grouped by customer ID, summed up

SELECT cusid, AVG(fee) FROM payments GROUP BY cusid
HAVING AVG(fee)<50;
	Show the average of fee payments grouped by customer ID, where this average is less than 50

SELECT MAX(fee) FROM payments;	Show the highest fee in the table
SELECT COUNT(*) FROM customers;	Show how many rows are in the table

SELECT cusid FROM payments t1 WHERE fee =
(SELECT MAX(t2.fee) FROM payments t2 WHERE t1.cusid=t2.cusid);
	Show the customer ID that pays the highest fee (via a subquery)

SELECT @maxfee:=MAX(fee) FROM payments;
SELECT cusid FROM payments t1 WHERE fee = @maxfee;
	Show the customer ID that pays the highest fee (via a user set variable)

SELECT * FROM customers WHERE lastname IN (SELECT lastname
FROM customers GROUP BY lastname HAVING COUNT(lastname) > 1);
	Show the customers which have same last name as other customers

SELECT cusid FROM payments WHERE fee >
ALL (SELECT fee FROM payments WHERE cusid = 4242001;
	Show the customer IDs that pay fees higher than the highest fee paid by customer ID 4242001

SELECT * FROM customers WHERE firstname LIKE 'Trill%';	Select customers whose first name matches the expression: % any number of chars, even zero _ a single char
SELECT * FROM customers WHERE firstname REGEXP '^Art.*r$';	Select customers whose first name matches the regex

SELECT firstname, lastname FROM customers WHERE zipcode = 'L1 42HG'
UNION
SELECT firstname, lastname FROM customers WHERE cusid > 4242001;
	Select customers that satisfy any of the two requirements

SELECT firstname, lastname FROM customers WHERE zipcode = 'L1 42HG'
INTERSECT
SELECT firstname, lastname FROM customers WHERE cusid > 4242001;
	Select customers that satisfy both of the two requirements

SELECT firstname, lastname FROM customers WHERE zipcode = 'L1 42HG'
EXCEPT
SELECT firstname, lastname FROM customers WHERE cusid > 4242001;
	Select customers that satisfy the first requirement but not the second

SQL	MySQL	Operation

SELECT customers.name, payments.bill
FROM customers, payments
WHERE customers.cusid = payments.cusid;
 	
SELECT customers.name, payments.bill
FROM customers NATURAL JOIN payments;
 	
SELECT customers.name, payments.bill
FROM customers JOIN payments
USING (cusid);
 	
SELECT customers.name, payments.bill
FROM customers JOIN payments
ON customers.cusid = payments.cusid;
	SELECT customers.name, payments.bill
	FROM customers
	[JOIN | INNER JOIN | CROSS JOIN]
	payments
	ON customers.cusid = payments.cusid;
	
	SELECT customers.name, payments.bill
	FROM customers
	[JOIN | INNER JOIN | CROSS JOIN]
	payments
	USING (cusid);
		Perform a **join** (aka **inner join**) of two tables to select data that are in a relationship

SELECT customers.name, payments.bill
FROM customers CROSS JOIN payments;
	SELECT customers.name, payments.bill
	FROM customers JOIN payments;
		Perform a **cross join** (aka **Cartesian product**) of two tables

SELECT customers.name, payments.bill
FROM customers LEFT JOIN payments
ON customers.cusid = payments.cusid;
		Perform a **left join** (aka **left outer join**) of two tables, returning records matching the join condition and also records in the left table with unmatched values in the right table

SELECT customers.name, payments.bill
FROM customers RIGHT JOIN payments
ON customers.cusid = payments.cusid;
		Perform a **right join** (aka **right outer join**) of two tables, returning records matching the join condition and also records in the right table with unmatched values in the left table

MySQL is the most used open source RDBMS (Relational Database Management System). It runs on TCP port 3306.
On RHEL 7 and later it is replaced by its fork MariaDB, but the names of the client and of most tools remain unchanged.

mysqld_safe	Start the MySQL server (mysqld) with safety features such as restarting the server if errors occur and logging runtime information to the error logfile. This is the recommended command
mysql_install_db (deprecated) mysqld --initialize	Initialize the MySQL data directory, create system tables, and set up an administrative account. To be run just after installing the MySQL server
mysql_secure_installation	Set password for root, remove anonymous users, disable remote root login, and remove test database. To be run just after installing the MySQL server
mysql -u root -p	Login to MySQL as root and prompt for the password
mysql -u root -p*password*	Login to MySQL as root with the specified password
mysql -u root -p -h *host* -P *port*	Login to the specified remote MySQL host and port
mysql -u root -p -eNB'SHOW DATABASES'	Run a SQL command via MySQL. Flags are: e Run in batch mode N Do not print table header B Do not print table decoration characters +-|
mysqldump -u root -p --all-databases > *dump.sql*	Backup all databases to a dump file
mysqldump -u root -p *db* > *dump.sql*	Backup a database to a dump file
mysqldump -u root -p --databases *db1 db2* > *dump.sql*	Backup multiple databases to a dump file
mysqldump -u root -p *db table1 table2* > *dump.sql*	Backup some tables of a database to a dump file
mysql -u root -p < *dump.sql*	Restore all databases from a dump file (which contains a complete dump of a MySQL server)
mysql -u root -p *db* < *dump.sql*	Restore a specific database from a dump file (which contains one database)
mysql_upgrade -u root -p	Check all tables in all databases for incompatibilities with the current version of MySQL
mysqlcheck	Perform table maintenance. Each table is locked while is being processed. Options are: --check Check table for errors (default) --analyze Analyze table --optimize Optimize table --repair Repair table; can fix almost all problems except unique keys that are not unique
mysqlcheck --check *db table*	Check the specified table of the specified database
mysqlcheck --check --databases *db1 db2*	Check the specified databases
mysqlcheck --check --all-databases	Check all databases

`mysqlslap`	Tool for MySQL stress tests
`mysqltuner.pl`	Review the current MySQL installation configuration for performances and stability
`mysqlreport` (obsolete)	Generate a user-friendly report of MySQL status values
`mytop`	Monitor MySQL processes and queries
`innotop`	Monitor MySQL InnoDB transactions

```
dbs="$(mysql -uroot -ppassword -Bse'SHOW DATABASES;')"
for db in $dbs
do
   [operation on $db]
done
```

Perform an operation on each database name

`SELECT Host, User FROM mysql.user;`	List all MySQL users
`CREATE USER 'user'@'localhost' IDENTIFIED BY 'p4ssw0rd';`	Create a MySQL local user and set his password
`DROP USER 'user'@'localhost';`	Delete a MySQL user

`SET PASSWORD FOR 'user'@'localhost' = PASSWORD('p4ssw0rd');`
`SET PASSWORD FOR 'user'@'localhost' = '*7E684A3DF6273CD1B6DE53';`
	Set a password for a MySQL user.
	The password can be specified either in plaintext or by its hash value

`SHOW GRANTS FOR 'user'@'localhost';`	Show permissions for a user
`GRANT ALL PRIVILEGES ON database.* TO 'user'@'localhost';`	Grant permissions to a user
`REVOKE ALL PRIVILEGES ON database.* FROM 'user'@'localhost';`	Revoke permissions from a user; must match the already granted permission on the same database or table

`GRANT SELECT ON *.* TO 'john'@'localhost' IDENTIFIED BY 'p4ssw0rd';`
`GRANT SELECT ON *.* TO 'john'@'localhost' IDENTIFIED BY PASSWORD '*7E684A3DF6273CD1B6DE53';`
	Create a MySQL user and set his grants at the same time

`FLUSH PRIVILEGES;`	Reload and commit the grant tables; must be run after any GRANT command

`SELECT * INTO OUTFILE 'file.csv'`
`FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'`
`LINES TERMINATED BY '\n' FROM database.table;`
	Export a table to a CSV file

`USE database; SOURCE dump.sql;`	Restore a database from a dump file
`USE database; LOAD DATA LOCAL INFILE 'file' INTO TABLE table;`	Populate a table with data from a file (one record per line, values separated by tabs)

`DO SLEEP(n);`
`SELECT SLEEP(n);`
	Sleep for *n* seconds

`SET PROFILING=1;`	Enable profiling
`SHOW PROFILE;`	Show the profile of the last executed query, with detailed steps and their timing

`statement;`
`statement\g`
	Send an SQL statement to the server

`statement\G`	Display result in vertical format, showing each record in multiple rows
`SELECT /*!99999 comment*/ * FROM database.table;`	Insert a comment
`SELECT /*!v statement*/ * FROM database.table;`	The commented *statement* is executed only if MySQL is version *v* or higher
`\c`	Cancel current input
`\! command`	Run a shell command
`TEE logfile`	Log all I/O of the current MySQL session to the specified logfile

`SHOW VARIABLES;`
`SHOW SESSION VARIABLES;`
`SHOW LOCAL VARIABLES;`
	Print session variables (affecting current connection only)

`SHOW GLOBAL VARIABLES;`	Print global variables (affecting global operations on the server)
`SHOW VARIABLES LIKE '%query%';`	Print session variables that match the given pattern

`SHOW VARIABLES LIKE 'hostname';`
`SELECT @@hostname;`
	Print a session variable with the given name

`SET sort_buffer_size=10000;`
`SET SESSION sort_buffer_size=10000;`
`SET LOCAL sort_buffer_size=10000;`
`SET @@sort_buffer_size=10000;`
`SET @@session.sort_buffer_size=10000;`
`SET @@local.sort_buffer_size=10000;`
	Set a session variable

`SET GLOBAL sort_buffer_size=10000;`
`SET @@global.sort_buffer_size=10000;`
	Set a global variable

`SHOW STATUS;`
`SHOW SESSION STATUS;`
`SHOW LOCAL STATUS;`
	Print session status (concerning current connection only)

`SHOW GLOBAL STATUS;`	Print global status (concerning global operations on the server)
`SHOW STATUS LIKE '%wsrep%';`	Print session status values that match the given pattern
`SHOW WARNINGS;`	Print warnings, errors and notes resulting from the most recent statement in the current session that generated messages
`SHOW ERRORS;`	Print errors resulting from the most recent statement in the current session that generated messages
`SHOW TABLE STATUS;`	Print information about all tables of the current database e.g. engine (InnoDB or MyISAM), rows, indexes, data length
`SHOW ENGINE INNODB STATUS;`	Print statistics concerning the InnoDB engine

`SELECT * FROM information_schema.processlist;`
`SHOW FULL PROCESSLIST;`
	Print the list of threads running in your local session; if run as root, print the list of threads running on the system

`SELECT * FROM information_schema.processlist`
`WHERE user='you';`
	Print the list of threads running in your local session and all your other logged-in sessions

`SHOW CREATE TABLE table;`
`SHOW CREATE VIEW view;`
	Print the CREATE statement that created *table* or *view*

`SELECT VERSION();`	Print the version of the MySQL server

`SELECT CURDATE();`
`SELECT CURRENT_DATE;`
	Print the current date

`SELECT CURTIME();`
`SELECT CURRENT_TIME;`
	Print the current time

`SELECT NOW();`	Print the current date and time
`SELECT USER();`	Print the current user@hostname that is logged in
`\s`	Print status information about server and current connection

```
SELECT table_schema AS "Name",
SUM(data_length+index_length)/1024/1024 AS "Size in Mb"
FROM information_schema.tables GROUP BY table_schema;
```

Display the sizes of all databases in the system (counting data + indexes)

```
SELECT table_schema AS "Name",
SUM(data_length+index_length)/1024/1024 AS "Size in Mb"
FROM information_schema.tables WHERE table_schema='database';
```

Display the size of *database*

```
SELECT table_name AS "Name",
ROUND(((data_length)/1024/1024),2) AS "Data size in Mb",
ROUND(((index_length)/1024/1024),2) AS "Index size in Mb"
FROM information_schema.TABLES WHERE table_schema='database'
ORDER BY table_name;
```

Display data and index size of all tables of *database*

```
SELECT table_name, table_rows
FROM information_schema.tables WHERE table_schema='database';
```

Print an estimate of the number of rows of each table of *database*

```
SELECT SUM(data_length+index_length)/1024/1024 AS "InnoDB Mb"
FROM information_schema.tables WHERE engine='InnoDB';
```

Display the amount of InnoDB data in all databases

```
SELECT table_name, engine
FROM information_schema.tables WHERE table_schema = 'database';
```

Print name and engine of all tables in *database*

```
SELECT CONCAT('KILL ',id,';')
FROM information_schema.processlist WHERE user='user'
INTO OUTFILE '/tmp/killuser'; SOURCE /tmp/killuser;
```

Kill all connections belonging to *user*

```
SELECT COUNT(1) SlaveThreadCount
FROM information_schema.processlist WHERE user='system user';
```

Distinguish between master and slave server; returns 0 on a master, >0 on a slave

```
SELECT ROUND(SUM(CHAR_LENGTH(field)<40)*100/COUNT(*),2)
FROM table;
```

Display the percentage of rows on which the string *field* is shorter than 40 chars

```
SELECT CHAR_LENGTH(field) AS Length, COUNT(*) AS Occurrences
FROM table GROUP BY CHAR_LENGTH(field);
```

Display all different lengths of string *field* and the number of times they occur

```
SELECT MAX(CHAR_LENGTH(field)) FROM table;
```

Display the longest string stored in *field*

```
SHOW FULL TABLES IN database WHERE table_type LIKE 'VIEW';
```

Display the list of views in *database*

```
SELECT "Table 1" AS `set`, t1.* FROM table1 t1 WHERE
ROW(t1.col1, t1.col2, t1.col3) NOT IN (SELECT * FROM table2)
UNION ALL
SELECT "Table 2" AS `set`, t2.* FROM table2 t2 WHERE
ROW(t2.col1, t2.col2, t2.col3) NOT IN (SELECT * FROM table1)
```

Display the differences between the contents of two tables *table1* and *table2* (assuming the tables are composed of 3 columns each)

How to resync a master-slave replication

1. On the master, on terminal 1:

   ```
   mysql -uroot -p
   RESET MASTER;
   FLUSH TABLES WITH READ LOCK;
   SHOW MASTER STATUS;
   ```

   Note the values of MASTER_LOG_FILE and MASTER_LOG_POS; these values will need to be copied on the slave

2. On the master, on terminal 2:

   ```
   mysqldump -uroot -p --all-databases > /path/to/dump.sql
   ```

   It is not necessary to wait until the dump completes

3. On the master, on terminal 1:

   ```
   UNLOCK TABLES;
   ```

4. Transfer the dump file from the master to the slave

5. On the slave:

   ```
   mysql -uroot -p
   STOP SLAVE;
   SOURCE /path/to/dump.sql;
   RESET SLAVE;
   CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin.nnnnnn', MASTER_LOG_POS=mm;
   START SLAVE;
   SHOW SLAVE STATUS;
   ```

How to recover the MySQL root password

1. Stop the MySQL server

2. Restart the MySQL server skipping the grant tables

   ```
   mysqld_safe --skip-grant-tables --skip-networking &
   ```

3. Connect to the MySQL server passwordlessly

   ```
   mysql -uroot
   ```

4. Reload the grant tables

   ```
   FLUSH PRIVILEGES;
   ```

5. Change the root password

   ```
   SET PASSWORD FOR 'root'@'localhost' = PASSWORD('newpassword');
   ```

6. Stop the MySQL server and restart it normally

PostgreSQL (aka Postgres) is an open source object-relational database. By default it listens for connections on TCP port 5432.

\list \l	List all databases
\list+ \l+	List all databases, displaying database size and description
\connect *database* \c *database*	Connect to *database*
\q	Quit

How to set up PostgreSQL with a *database* owned by *user*

1.	Set up PostgreSQL	postgresql-setup initdb
2.	Change the password of the postgres shell user	passwd postgres
3	Create the *user* shell user	useradd *user*
4.	Switch to the postgres shell user and connect to PostgreSQL	su - postgres psql -U postgres
5.	In PostgreSQL, create the *user*	CREATE ROLE *user* WITH LOGIN; \password *user* \q
6.	Create a *database* owned by *user*	createdb -E utf8 -l C -T template0 *database* -O *user*
7.	Switch to the postgres shell user and connect to PostgreSQL	su - postgres psql -U postgres
8.	In PostgreSQL, grant the necessary privileges on *database* to *user*	GRANT ALL PRIVILEGES ON DATABASE *database* TO *user*; \q
9.	Verify that *user* can login to PostgreSQL	su - *user* psql -U *user* -W

The **X Window System** (aka **X11** or **X**) is a windowing system for Linux and UNIX-like OSes, providing a basic framework for GUI applications via a client-server model. A **display manager** provides a login screen to enter an X session and introduces the user to the **desktop environment** (e.g. GNOME, KDE, CDE, Enlightenment).

Display Manager		Configuration files		Display Manager greeting screen

xdm	X Display Manager	`/etc/x11/xdm/Xaccess`	Control inbound requests from remote hosts	Defined in `/etc/x11/xdm/Xresources` by the line:
				`xlogin*greeting: \`
				`Debian GNU/Linux (CLIENTHOST)`

		`/etc/x11/xdm/Xresources`	Configuration settings for X applications and the login screen	
		`/etc/x11/xdm/Xservers`	Association of X displays with local X server software, or with X terminals via XDMCP	
		`/etc/x11/xdm/Xsession`	Script launched by xdm after login	
		`/etc/x11/xdm/Xsetup_0`	Script launched before the graphical login screen	
		`/etc/x11/xdm/xdm-config`	Association of all xdm configuration files	
gdm	GNOME Display Manager	`/etc/gdm/gdm.conf` or `/etc/gdm/custom.conf`		Configured via `gdmsetup`
kdm	KDE Display Manager	`/etc/kde/kdm/kdmrc`		Configured via `kdm_config`

`/etc/init.d/xdm start`
`/etc/init.d/gdm start`
`/etc/init.d/kdm start`
	Start the appropriate Display Manager

`xorgconfig` (Debian) `Xorg -configure` (Red Hat)	Configure X (text mode)
`xorgcfg` (Debian) `system-config-display` (Red Hat)	Configure X (graphical mode)
`X -version`	Show which version of X is running
`xdpyinfo`	Display information about the X server
`xwininfo`	Display information about windows

`xhost + 10.3.3.3`
`xhost - 10.3.3.3`
	Add or remove 10.3.3.3 to the list of hosts allowed to make X connections to the local machine

`switchdesk gde`	Switch to the GDE Display Manager at runtime
`gnome-shell --version`	Show which version of GNOME is running

`/etc/X11/xorg.conf`	Configuration file for X
`~/.Xresources`	Configuration settings for X applications, in the form *`program`*`*`*`resource`*`:` *`value`*
`$DISPLAY`	Environment variable defining the display name of the X server, in the form *`hostname`*`:`*`displaynumber`*`.`*`screennumber`*

The following line in `/etc/inittab` instructs `init` to launch XDM at runlevel 5:

```
x:5:respawn:/usr/X11R6/bin/xdm -nodaemon
```

The following lines in `/etc/sysconfig/desktop` define GNOME as the default Display Environment and Display Manager:

```
desktop="gde"
displaymanager="gdm"
```

`xdotool`	X automation tool
`xdotool getwindowfocus`	Get the ID of the currently focused window (if run in command line, it is the terminal where this command is typed)
`xdotool selectwindow`	Pop up an X cursor and get the ID of the window selected by it
`xdotool key --window 12345678 Return`	Simulate a RETURN keystroke inside window ID 12345678

`xprop`	X property displayer. Pops up a cursor to select a window
`xprop | grep WM_CLASS`	Get process name and GUI application name of the selected window

`xrandr`
`xrandr -q`
	Show screen(s) size and resolution

`xrandr --output eDP1 --right-of VGA1`	Extend the screen on an additional VGA physical screen situated to the left

`xsel`	Manipulate the X selection (primary, secondary, and clipboard)
`xsel -b <` *`file`*	Copy the contents of a file to the X clipboard
`xsel -b -a <` *`file`*	Append the contents of a file to the X clipboard
`xsel -b -o`	Output onscreen the contents of the X clipboard

`cat` *`file`* `| xclip -i`	Copy the contents of a file to the X clipboard

`mkfontdir`	Catalog the newly installed fonts in the new directory
`xset fp+ /usr/local/fonts`	Dynamically add new installed fonts in `/usr/local/fonts` to the X server
`xfs`	Start the X font server
`fc-cache`	Install fonts and build font information cache

Main	
BackSpace	ff08
Tab	ff09
Linefeed	ff0a
Clear	ff0b
Return	ff0d
Pause	ff13
Scroll_Lock	ff14
Sys_Req	ff15
Escape	ff1b
Delete	ffff

Cursor control	
Home	ff50
Left	ff51
Up	ff52
Right	ff53
Down	ff54
Prior	ff55
Page_Up	ff55
Next	ff56
Page_Down	ff56
End	ff57
Begin	ff58

Misc functions	
Select	ff60
Print	ff61
Execute	ff62
Insert	ff63
Undo	ff65
Redo	ff66
Menu	ff67
Find	ff68
Cancel	ff69
Help	ff6a
Break	ff6b
Mode_switch	ff7e
script_switch	ff7e
Num_Lock	ff7f

Modifiers	
Shift_L	ffe1
Shift_R	ffe2
Control_L	ffe3
Control_R	ffe4
Caps_Lock	ffe5
Shift_Lock	ffe6
Meta_L	ffe7
Meta_R	ffe8
Alt_L	ffe9
Alt_R	ffea
Super_L	ffeb
Super_R	ffec
Hyper_L	ffed
Hyper_R	ffee

Latin 1	
space	0020
exclam	0021
quotedbl	0022
numbersign	0023
dollar	0024
percent	0025
ampersand	0026
apostrophe	0027
quoteright	0027
parenleft	0028
parenright	0029
asterisk	002a
plus	002b
comma	002c
minus	002d
period	002e
slash	002f
0 - 9	0030 - 0039
colon	003a
semicolon	003b
less	003c
equal	003d
greater	003e
question	003f
at	0040
A - Z	0041 - 005a
bracketleft	005b
backslash	005c
bracketright	005d
asciicircum	005e
underscore	005f
grave	0060
quoteleft	0060
a - z	0061 - 007a
braceleft	007b
bar	007c
braceright	007d
asciitilde	007e
nobreakspace	00a0
exclamdown	00a1
cent	00a2
sterling	00a3
currency	00a4
yen	00a5
brokenbar	00a6
section	00a7
diaeresis	00a8
copyright	00a9
ordfeminine	00aa
guillemotleft	00ab
notsign	00ac
hyphen	00ad
registered	00ae
macron	00af
degree	00b0
plusminus	00b1
twosuperior	00b2
threesuperior	00b3
acute	00b4
mu	00b5
paragraph	00b6
periodcentered	00b7
cedilla	00b8
onesuperior	00b9
masculine	00ba
guillemotright	00bb
onequarter	00bc
onehalf	00bd
threequarters	00be
questiondown	00bf
Agrave	00c0
Aacute	00c1
Acircumflex	00c2
Atilde	00c3
Adiaeresis	00c4
Aring	00c5
AE	00c6
Ccedilla	00c7
Egrave	00c8
Eacute	00c9
Ecircumflex	00ca
Ediaeresis	00cb
Igrave	00cc
Iacute	00cd
Icircumflex	00ce
Idiaeresis	00cf
ETH	00d0
Eth	00d0
Ntilde	00d1
Ograve	00d2
Oacute	00d3
Ocircumflex	00d4
Otilde	00d5
Odiaeresis	00d6
multiply	00d7
Oslash	00d8
Ooblique	00d8
Ugrave	00d9
Uacute	00da
Ucircumflex	00db
Udiaeresis	00dc
Yacute	00dd
THORN	00de
Thorn	00de
ssharp	00df
agrave	00e0
aacute	00e1
acircumflex	00e2
atilde	00e3
adiaeresis	00e4
aring	00e5
ae	00e6
ccedilla	00e7
egrave	00e8
eacute	00e9
ecircumflex	00ea
ediaeresis	00eb
igrave	00ec
iacute	00ed
icircumflex	00ee
idiaeresis	00ef
eth	00f0
ntilde	00f1
ograve	00f2
oacute	00f3
ocircumflex	00f4
otilde	00f5
odiaeresis	00f6
division	00f7
oslash	00f8
ooblique	00f8
ugrave	00f9
uacute	00fa
ucircumflex	00fb
udiaeresis	00fc
yacute	00fd
thorn	00fe
ydiaeresis	00ff

Latin 2	
Aogonek	01a1
breve	01a2
Lstroke	01a3
Lcaron	01a5
Sacute	01a6
Scaron	01a9
Scedilla	01aa
Tcaron	01ab
Zacute	01ac
Zcaron	01ae
Zabovedot	01af
aogonek	01b1
ogonek	01b2
lstroke	01b3
lcaron	01b5
sacute	01b6
caron	01b7
scaron	01b9
scedilla	01ba
tcaron	01bb
zacute	01bc
doubleacute	01bd
zcaron	01be
zabovedot	01bf
Racute	01c0
Abreve	01c3
Lacute	01c5
Cacute	01c6
Ccaron	01c8
Eogonek	01ca
Ecaron	01cc
Dcaron	01cf
Dstroke	01d0
Nacute	01d1
Ncaron	01d2
Odoubleacute	01d5
Rcaron	01d8
Uring	01d9
Udoubleacute	01db
Tcedilla	01de
racute	01e0
abreve	01e3
lacute	01e5
cacute	01e6
ccaron	01e8
eogonek	01ea
ecaron	01ec
dcaron	01ef
dstroke	01f0
nacute	01f1
ncaron	01f2
odoubleacute	01f5
rcaron	01f8
uring	01f9
udoubleacute	01fb
tcedilla	01fe
abovedot	01ff

This table is derived from `keysymdef.h` which defines keysym codes (i.e. characters or functions associated with each key in the X Window System) as `XK_`*`key`* and its hex value. The *key* can be passed as argument to the `xdotool key` command.

/etc/passwd User accounts

```
root:x:0:0:/root:/bin/bash
bin:x:1:1:/bin:/bin/bash
jdoe:x:500:100:John Doe,,555-1234,,:/home/jdoe:/bin/bash
1    2 3   4   5                    6          7
```

1 Login name

2 Hashed password (obsolete), or `x` if password is in `/etc/shadow`

3 UID – User ID

4 GID – Default Group ID

5 GECOS field – Information about the user: Full name, Room number, Work phone, Home phone, Other

6 Home directory of the user

7 Login shell (if set to `/sbin/nologin` or `/bin/false`, user will be unable to log in)

/etc/shadow User passwords

```
root:$6$qk8JmJHf$X9GfOZ/i9LZP4Kldu6.D3cx2pXA:15537:0:99999:7:::
bin:*:15637:0:99999:7:::
jdoe:!$6$YOiH1otQ$KxeeUKHExK8e3jCUdw9Rxy3Wu53:15580:0:99999:7::15766:
1    2 a b        c                           3     4 5     6 78     9
```

1 Login name

2 Hashed password (`*` if account is disabled, `!` or `!!` if no password is set, prefixed by `!` if the account is locked). Composed of the following subfields separated by `$`:
a Hashing algorithm: `1` = MD5, `2a` = Blowfish, `5` = SHA256, `6` = SHA512 (recommended)
b Random salt, up to 16 chars long. This is to thwart password cracking attempts based on rainbow tables
c String obtained by hashing the user's plaintext password concatenated to the stored salt

3 Date of last password change (in number of days since 1 January 1970)

4 Days before password may be changed; if 0, user can change the password at any time

5 Days after which password must be changed

6 Days before password expiration that user is warned

7 Days after password expiration that account is disabled

8 Date of account disabling (in number of days since 1 January 1970)

9 Reserved field

/etc/group Group accounts

```
root:x:0:root
jdoe:x:501
staff:x:530:jdoe,asmith
1     2 3   4
```

1 Group name

2 Encrypted password, or `x` if password is in `/etc/gshadow`

3 GID – Group ID

4 Group members (if this is not their Default Group)

/etc/gshadow Group passwords

```
root::root:root
jdoe:!::
staff:0cfz7IpLhW19i::root,jdoe
1     2             34
```

1 Group name

2 Encrypted password, or `!` if no password is set (default)

3 Group administrators

4 Group members

`/etc/shadow` and `/etc/gshadow` are mode 000 and therefore readable only by the root user.

useradd -m *user*	Create a user account, creating and populating his homedir from /etc/skel
useradd -mc "*Name Surname*" *user*	Create a user account, specifying his full name
useradd -ms /bin/ksh *user*	Create a user account, specifying his login shell
useradd -D	Show default values for user account creation, as specified in /etc/login.defs and /etc/default/useradd
usermod -c "*Name Surname*" *user*	Modify the GECOS field of a user account
usermod -L *user*	Lock a user account
usermod -U *user*	Unlock a user account

Most options for usermod and useradd are the same.

userdel -r *user*	Delete a user and his homedir
chfn *user*	Change the GECOS field of a user
chsh *user*	Change the login shell of a user
passwd *user*	Change the password of a user
passwd -l *user*	Lock a user account
passwd -S *user*	Show information about a user account: username, account status (L=locked, P=password, NP=no password), date of last password change, min age, max age, warning period, inactivity period in days
chage -E 2022-02-14 *user*	Change the password expiration date; account will be locked at that date
chage -d 13111 *user*	Change the date (in number of days since 1 January 1970) of last password change
chage -d 0 *user*	Force the user to change password at his next login
chage -M 30 *user*	Change the max number of days during which a password is valid
chage -m 7 *user*	Change the min number of days between password changes
chage -W 15 *user*	Change the number of days before password expiration that the user will be warned
chage -I 3 *user*	Change the number of days after password expiration before the account is locked
chage -l *user*	List password aging information for a user
chpasswd	Tool for batch update of passwords. Reads from stdin a list of *username*:*password*
vipw vigr	Edit manually /etc/passwd, /etc/shadow, /etc/group, or /etc/gshadow
adduser deluser	User-friendly front-end commands for user management
system-config-users (Red Hat)	GUI for user and group management

groupadd *group*	Create a group
groupmod -n *newname oldname*	Change a group name
groupdel *group*	Delete a group
gpasswd *group*	Set or change the password of a group
gpasswd -a *user group*	Add a user to a group
gpasswd -d *user group*	Delete a user from a group
gpasswd -A *user group*	Add a user to the list of administrators of the group
addgroup delgroup	User-friendly front-end commands for group management

On a system, every user is identified by a numeric **UID (User ID)**, and every group by a numeric **GID (Group ID)**.
UID 0 is assigned to the superuser.
UIDs from 0 to 99 should* be reserved for static allocation by the system and not be created by applications.
UIDs from 100 to 499 should* be reserved for dynamic allocation by the superuser and post-install scripts.
UIDs for user accounts start from 500 (Red Hat) or 1000 (SUSE, Debian).

* as recommended by the Linux Standard Base core specifications

A process has an effective, saved, and real UID and GID.

Effective UID	Used for most access checks, and as the owner for files created by the process. An unprivileged process can change its effective UID only to either its saved UID or its real UID.
Saved UID	Used when a process running with elevated privileges needs to temporarily lower its privileges. The process changes its effective UID (usually root) to an unprivileged one, and its privileged effective UID is copied to the saved UID. Later, the process can resume its elevated privileges by resetting its effective UID back to the saved UID.
Real UID	Used to identify the real owner of the process and affect the permissions for sending signals. An unprivileged process can signal another process only if the sender's real or effective UID matches the receiver's real or saved UID. Child processes inherit the credentials from the parent, so they can signal each other.

`/etc/login.defs`	Definition of default values (UID and GID ranges, mail directory, account validity, password encryption method, etc.) for user account creation

`whoami`	Print your username (as effective UID)

`id`	Print your real and effective UID and GID, and the groups you are a member of
`id -u`	Print your effective UID
`id` *`user`*	Print UID, GID, and groups information about *user*

Sudo is a mechanism that allows running a command as another user. Sudo access rights are defined in the sudoers files `/etc/sudoers` and `/etc/sudoers.d/*`; these files must be edited only via `visudo`.
Commands run by sudo users are logged via syslog on `/var/log/auth.log` (Debian) or `/var/log/secure` (Red Hat).

sudo -u *user command*	Run *command* as *user*
sudo *command* sudo -u root *command*	Run *command* as root
sudo su - sudo -i	Login on an interactive shell as root
sudo -u *user* -s	Login as *user* with a shell, even if the user's shell is `/sbin/nologin` or similar
sudo -l	List the allowed commands for the current user
sudo !!	Run again the last command, but this time as root
sudoedit /etc/passwd sudo -e /etc/passwd	Edit safely a file (in this case, `/etc/passwd`) according to security policies. It is recommended to allow users use this command instead of sudoing text editors as root on protected files, because the editor might be able to spawn a shell, causing security issues
visudo	Edit safely the sudoers file
visudo -c	Check the sudoers file for syntax errors, unused aliases, etc.
su *user*	Run a shell as *user*
su su root	Run a shell as root
su -c "fdisk -l"	Pass a single command to the shell
su - su -l	Ensure that the spawned shell is a login shell, hence running login scripts and setting the correct environment variables. Recommended option
gksudo -u root *command* gksu -u root -l	GUI front-ends to `su` and `sudo` used to run an X Window command or application as root. Pops up a requester prompting the user for root's password
runuser -u *user command*	Run *command* as *user*. Can be launched only by root

`chvt` *n* CTRL ALT F*n*	Make `/dev/tty`*n* the foreground terminal
`vlock` `away`	Lock the virtual console (terminal)
`tty`	Print your terminal device (e.g. `/dev/tty1`, `/dev/pts/1`)
`stty`	Change or display terminal line settings
`stty -ixon`	Disable XON/XOFF flow control
`clear` CTRL L	Clear the terminal screen
`tmux`	Terminal multiplexer
`nohup` *script.sh*	Prevent a process from terminating (receiving a SIGHUP) when its parent Bash dies. When a Bash shell is terminated cleanly via `exit`, its jobs will become child of the Bash's parent and will continue running. When a Bash shell is killed instead, it issues a SIGHUP to its children which will terminate
`screen`	Screen manager that multiplexes a single virtual VT100/ANSI terminal between multiple processes or shells. When the connection to a terminal is lost (e.g. because the terminal is closed manually, the user logs out, or the remote SSH session goes into timeout), a SIGHUP is sent to the shell and from there to all running child processes which are therefore terminated. The `screen` command starts an interactive shell screen session, to which the user will be able to reattach later
`screen -S` *sessionname*	Start a screen session with the specified session name
`screen` *command*	Start the specified command in a screen session; session will end when the command exits
`screen -list`	Show the list of detached screen sessions
`screen -r` *pid.tty.host* `screen -r` *owner/pid.tty.host*	Resume a detached screen session
`screen -R`	Resume the last detached screen session
`screen -d -R` *sessionname*	Detach a remote screen session and reattach your current terminal to it
CTRL A	Send a command to the window manager: `0` ... `9` Switch between screen sessions `c` Create a new screen session `?` Show help

How to detach an already running job that was not started in a `screen` session
(these commands detach the job from its parent shell, so that the job will not be killed when the terminal is closed)

1.	CTRL Z	Suspend the job
2.	`bg`	Send the job to background
3.	`jobs`	Show the number (let's assume is *n*) of the backgrounded job
4.	`disown -h %`*n*	Mark job *n* so it will not receive a SIGHUP from its parent shell

or

1.	`screen`	Start a screen session
2.	`reptyr` *pid*	Attach the job with process ID *pid* to the new terminal (screen session)

`write` *user*	Write interactively a message to the terminal of *user* (which must be logged in)
`echo "`*Message*`" | write` *user*	Write a message to the terminal of *user* (which must be logged in)
`wall`	Write interactively a message to the terminal of all logged in users
`echo "`*Message*`" | wall`	Write a message to the terminal of all logged in users
`talk` *user*	Open an interactive chat session with *user* (which must be logged in)

`mesg y`
`chmod g+w $(tty)`
	Allow the other users to message you via `write`, `wall`, and `talk`

`mesg n`
`chmod g-w $(tty)`
	Disallow the other users to message you via `write`, `wall`, and `talk`

`mesg`	Display your current message permission status

`mesg` works by enabling/disabling the group write permission of your terminal device, which is owned by system group `tty`. The root user is always able to message users.

cron

`cron` is a job scheduler, allowing the repeated execution of commands specified in crontab files.
The `crond` daemon checks the crontab files every minute and runs the command as the specified user at the specified times.
It is not necessary to restart `crond` after the modification of a crontab file, as the changes will be reloaded automatically.

If `/etc/cron.allow` exists, only users listed therein can access the service.
If `/etc/cron.deny` exists, all users except those listed therein can access the service.
If none of these files exist, all users can access the service.

`/etc/crontab` `/etc/cron.d/*`	System-wide crontab files

`/etc/cron.hourly/`
`/etc/cron.daily/`
`/etc/cron.weekly/`
`/etc/cron.monthly/`
	Scripts placed in these directories will be automatically executed on the specified periods

`/var/spool/cron/`*`user`*	Crontab of *user*. This file has the same format as the system-wide crontab files, except that the "user" field is not present

`crontab -e`	Edit your user crontab file
`crontab -l`	List the contents of your crontab file
`crontab -e -u `*`user`*	Edit the crontab file of another *user* (command available only to the superuser)

/etc/crontab

# m	h	dom	mon	dow	user	command	
25	6	*	*	1	root	/opt/script1.sh	every Monday at 6:25 AM
*/5	16	*	*	*	root	/opt/script2.sh	from 4:00 to 4:55 PM every 5 minutes every day
0,30	7	25	12	*	jdoe	/home/jdoe/foo.sh	at 7:00 and 7:30 AM on 25th December
3	17	*	*	1-5	root	/root/bar.sh	at 5:03 PM every day, from Monday to Friday

m	minutes
h	hours
dom	day of month (1-31)
mon	month (1-12 or jan-dec)
dow	day of week (0-7 or sun-sat; 0=7=Sunday)
user	User as whom the command will be executed
command	Command that will be executed at the specified times

The `crond` daemon also runs anacron jobs, which allow the execution of periodic jobs on a machine that is not always powered on, such as a laptop. Only the superuser can schedule anacron jobs, which have a granularity of one day (vs one minute for cron jobs).

`/var/spool/anacron/`*`jobid`*	Date of the last execution of the anacron job identified by *jobid*

/etc/anacrontab

# period	delay	job-identifier	command	
7	10	cron.weekly	/opt/script3.sh	If the job has not been run in the last 7 days, wait 10 minutes and then execute the command

period	period, in days, during which the command was not executed
delay	delay to wait, in minutes, before execution of the command
job-identifier	job identifier in anacron messages; should be unique for each anacron job
command	command that will be executed

at is used for scheduled execution of commands that must run only once. Execution of these commands is the duty of the atd daemon.

If /etc/at.allow exists, only users listed therein can access the service.
If /etc/at.deny exists, all users except those listed therein can access the service.
If none of these files exist, no user except the superuser can access the service.

`at 5:00pm tomorrow script.sh`
`at -f listofcommands.txt 5:00pm tomorrow`
`echo "rm file" | at now+2 minutes`
	Execute a command once at the specified time (absolute or relative)

`at -l`
`atq`
	List the scheduled jobs

`at -d 3`
`atrm 3`
	Remove job number 3 from the list

`batch`	Schedule execution of a command for when the system is not too charged. Reads a command from stdin and runs it when the system's load average falls below 0.8

`bc`	Calculator
`dc`	Calculator featuring unlimited precision arithmetic
`factor`	Finds the prime factors of a number
`units`	Converter of quantities between different units
`cal`	Calendar
`banner`	Print a text in large letters made of the character #
`figlet`	Print a text in large letters, in a specific font
`toilet`	Print a text in large colorful letters, in a specific font
`lolcat`	Print a text in rainbow coloring
`fortune`	Print a random aphorism, like those found in fortune cookies
`sensors`	Print sensor chips information (e.g. temperature)
`beep`	Produce a beep from the machine's speakers
`speaker-test`	Speaker test tone generator for the ALSA (Advanced Linux Sound Architecture) framework
`on_ac_power`	Return 0 (true) if machine is connected to AC power, 1 (false) if on battery. Useful for laptops
`ipcalc`	IP addresses calculator
`pwgen`	Random password generator
`pwqgen`	Random password generator with controllable quality
`uuidgen`	Generator of UUIDs (random or time-based)
`haveged`	Generator of random numbers via the HAVEGE (Hardware Volatile Entropy Gathering and Expansion) algorithm. Can be run as a daemon to automatically replenish `/dev/random` whenever the supply of random bits in the random device gets too low
`aspell`	Spell checker
`cloc`	Count lines of source code
`gnome-terminal`	GNOME shell terminal
`conky`	Highly configurable system monitor widget with integration for audio player, email, and news
`gkrellm`	System monitor widget

Locale environment variables	
`LANG` `LANGUAGE`	Language, stored in `/etc/default/locale`. When scripting, it is recommended to set `LANG=C` because this specifies the minimal locale environment for C translation, and guarantees a standard collation and formats for the execution of scripts
`LC_CTYPE`	Character classification and case conversion
`LC_NUMERIC`	Non-monetary numeric formats
`LC_TIME`	Date and time formats
`LC_COLLATE`	Alphabetical order
`LC_MONETARY`	Monetary formats
`LC_MESSAGES`	Language and encoding of system messages and user input
`LC_PAPER`	Paper size
`LC_NAME`	Personal name formats
`LC_ADDRESS`	Geographic address formats
`LC_TELEPHONE`	Telephone number formats
`LC_MEASUREMENT`	Measurement units (metric or others)
`LC_IDENTIFICATION`	Metadata about locale
`LC_ALL`	Special variable overriding all others
The values of these locale environment variables are in the format `*language_territory.encoding*` e.g. `en_US.UTF-8`. The list of supported locales is stored in `/usr/share/i18n/SUPPORTED`.	

`locale`	Show locale environment variables
`locale-gen it_IT.UTF-8`	Generate a locale (in this case IT) by compiling a list of locale definition files
`apt-get install manpages-it language-pack-it` (Debian)	Install a different locale (in this case IT); this affects system messages and manpages
`iconv -f IS6937 -t IS8859 *filein* > *fileout*`	Convert a text file from a codeset to another

ISO/IEC-8859 is a standard for 8-bit encoding of printable characters.
The first 256 characters in ISO/IEC-8859-1 (Latin-1) are identical to those in Unicode.
UTF-8 encoding can represent every character in the Unicode set, and was designed for backward compatibility with ASCII.

`date`	Show current date and time

`date -d "9999 days ago"`
`date -d "1970/01/01 + 4242"`
	Calculate a date and show it

`date +"%F %H:%M:%S"`	Show current date in the format specified
`date +"%s"`	Show current date in Unix time format (seconds elapsed since 00:00:00 1/1/1970)
`date -s "20130305 23:30:00"`	Set the date
`date 030523302013`	Set the date, in the format *MMDDhhmmYYYY*

`timedatectl`	Show current date and time

`timedatectl set-time 2013-03-05`
`timedatectl set-time 23:30`
	Set the date

`timedatectl list-timezones`	List all possible timezones

`zdump GMT`	Show current date and time in the GMT timezone

`tzselect`
`tzconfig`
`dpkg-reconfigure tzdata`
`timedatectl set-timezone` *`timezone`*
	
	(Debian)
	(Red Hat)
		Set the timezone

`/etc/timezone`	(Debian)	Timezone
`/etc/localtime`	(Red Hat)	Timezone, a symlink to the appropriate timezone file in `/usr/share/zoneinfo/`

`ntpd`	NTP daemon, keeps the clock in sync with Internet time servers
`ntpd -q`	Synchronize the time once and quit
`ntpd -g`	Force NTP to start even if clock is off by more than the panic threshold (1000 secs)
`ntpd -nqg`	Start NTP as a non-daemon, force synchronization of the clock, and quit. The NTP daemon must not be running when this command is launched

`ntpq -p` *`timeserver`*	Print the list of peers for the time server

`ntpdate` *`timeserver`*	Synchronizes the clock with the specified time server
`ntpdate -b` *`timeserver`*	Brutally set the clock, without waiting for it to adjust slowly
`ntpdate -q` *`timeserver`*	Query the time server without setting the clock

The `ntpdate` command is deprecated; to synchronize the clock, use `ntpd` instead.

`chronyd`	Daemon for chrony, a versatile NTP client/server
`chronyc`	Command line interface for the chrony daemon

`hwclock --show`
`hwclock -r`
	Show the hardware clock

`hwclock --hctosys`
`hwclock -s`
	Set the system time from the hardware clock

`hwclock --systohc`
`hwclock -w`
	Set the hardware clock from system time

`hwclock --utc`	Indicate that the hardware clock is kept in Coordinated Universal Time
`hwclock --localtime`	Indicate that the hardware clock is kept in local time

syslogd rsyslogd (Ubuntu 14)	Daemon logging events from user processes
klogd	Daemon logging events from kernel processes

/etc/syslog.conf

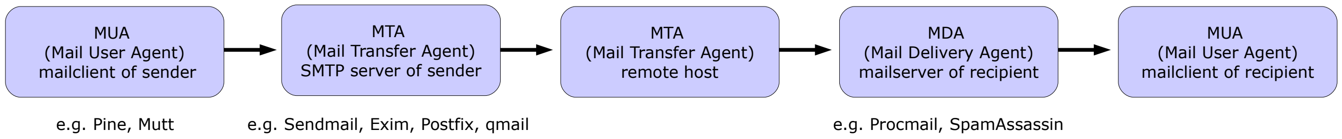
```
# facility.level                   action
*.info;mail.none;authpriv.none    /var/log/messages
authpriv.*                        /var/log/secure
mail.*                            /var/log/maillog
*.alert                           root
*.emerg                           *
local5.*                          @10.7.7.7
local7.*                          /var/log/boot.log
```

Facility Creator of the message	**Level** Severity of the message	**Action** Destination of the message	
auth or security† authpriv cron daemon kern lpr mail mark (for syslog internal use) news syslog user uucp local0 ... local7 (custom)	emerg or panic† (highest) alert crit err or error† warning or warn† notice info debug (lowest) none (facility disabled)	*file* @*host* *user1,user2,user3* *	message is written into a log *file* message is sent to a logger server *host* (via UDP port 514) message is sent to the specified users' consoles message is sent to all logged-in users' consoles
† = deprecated			

Facilities and levels are listed in the manpage `man 3 syslog`.

logger -p auth.info "*Message*"	Send a message to syslog with facility "auth" and priority "info"
logrotate	Rotate logs. It gzips, renames, and eventually deletes old logfiles according to the configuration files /etc/logrotate.conf and /etc/logrotate.d/*. It is usually scheduled as a daily cron job

/var/log/messages	Global system logfile
/var/log/dmesg	Kernel ring buffer information
/var/log/kern.log	Kernel log
/var/log/boot.log	Information logged during boot



MUA
(Mail User Agent)
mailclient of sender

e.g. Pine, Mutt

MTA
(Mail Transfer Agent)
SMTP server of sender

e.g. Sendmail, Exim, Postfix, qmail

MTA
(Mail Transfer Agent)
remote host

MDA
(Mail Delivery Agent)
mailserver of recipient

e.g. Procmail, SpamAssassin

MUA
(Mail User Agent)
mailclient of recipient

`~/.forward`	Mail address(es) to which forward the user's mail, or mail commands
`/etc/aliases` `/etc/mail/aliases`	Aliases database for users on the local machine. Each line has syntax *alias*`:` *user*
`/var/spool/mail/`*user*	Inbox for *user* on the local machine
`/var/log/mail.log` (Debian) `/var/log/maillog` (Red Hat)	Mail logs

`mail` `mailx`	Mailclient with advanced commands for non-interactive (batch) use
`pine`	Mailclient (obsolete)
`alpine`	Mailclient, a replacement for `pine`

`mailx -s "`*Subject*`" -S smtp="`*mailserver*`:25" \` *user@domain.com* `<` *messagefile*	Send a mail message to *user@domain.com* via an external SMTP server *mailserver*
`uuencode` *binaryfile* `|` `mail` *user@domain.com*	Send a binary file to *user@domain.com* (obsolete, not recommended because many mailclients will display the received attachment inline)
`mutt -a` *binaryfile* `--` *user@domain.com* `< /dev/null`	Send a binary file to *user@domain.com* using the Mutt MUA

Mailbox formats		
mbox	Each mail folder is a single file, storing multiple email messages. Advantages: universally supported; fast search inside a mail folder. Disadvantages: issues with file locking; possible mailbox corruption.	`$HOME/Mail/`*folder*
Maildir	Each mail folder is a directory, and contains the subdirectories `/cur`, `/new`, and `/tmp`. Each email message is stored in its own file with a unique filename ID. The process that delivers an email message writes it to a file in the `tmp/` directory, and then moves it to `new/`. The moving is commonly done by hard linking the file to `new/` and then unlinking the file from `tmp/`, which guarantees that a MUA will not see a partially written message as it never looks in `tmp/`. When the MUA finds mail messages in `new/` it moves them to `cur/`. Advantages: fast location/retrieval/deletion of a specific mail message; no file locking needed; can be used with NFS. Disadvantages: some filesystems may not efficiently handle a large number of small files; searching text inside all mail messages is slower.	`$HOME/Mail/`*folder*`/`

SMTP commands

```
220 smtp.example.com ESMTP Postfix      (server)
HELO xyz.linux.org                      (client)
250 Hello xyz.linux.org, glad to meet you
MAIL FROM: alice@linux.org
250 Ok
RCPT TO bob@foobar.com
250 Ok
RCPT TO carol@quux.net
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: Alice <alice@linux.org>
To: Bob <bob@foobar.com>
Cc: Carol <carol@quux.net>
Date: Wed, 13 August 2014 18:02:43 -0500
Subject: Test message

This is a test message.
.
250 OK id=1OjReS-0005kT-Jj
QUIT
221 Bye
```

`HELO xyz.linux.org`	Initiate the conversation and identify client host to server
`EHLO xyz.linux.org`	Like HELO, but tell server to use Extended SMTP
`MAIL FROM: alice@linux.org`	Specify mail sender
`RCPT TO: bob@foobar.com`	Specify mail recipient
`DATA`	Specify data to send. Ended with a dot on a single line
`QUIT` `RSET`	Disconnect
`HELP`	List all available commands
`NOOP`	Empty command
`VRFY alice@linux.org`	Verify the existence of an e-mail address (this command should not be implemented, for security reasons)
`EXPN mailinglist`	Check mailing list membership

SMTP response codes

first digit	**1**	Command accepted, but not processed until client sends confirmation
	2	Command successfully completed
	3	Command accepted, but not processed until client sends more information
	4	Command failed due to temporary errors
	5	Command failed due to permanent errors
second digit	**0**	Syntax error or command not implemented
	1	Informative response in reply to a request for information
	2	Connection response in reply to a data transmission
	5	Status response in reply to a mail transfer operation
third digit		Specifies further the response

211	System status or help reply
214	Help message
220	The server is ready
221	The server is ending the conversation
250	The requested action was completed
251	The specified user is not local, but the server will forward the mail message
354	Reply to the DATA command. After getting this, start sending the message body
421	The mail server will be shut down, try again later
450	The mailbox that you are trying to reach is busy, try again later
451	The requested action was not done. Some error occurred in the mail server
452	The requested action was not done. The mail server ran out of system storage
500	The last command contained a syntax error or the command line was too long
501	The parameters or arguments in the last command contained a syntax error
502	The last command is not implemented in the mail server
503	The last command was sent out of sequence
504	One of the parameters of the last command is not implemented by the server
550	The mailbox that you are trying to reach can't be found or you don't have access rights
551	The specified user is not local; part of message text will contain a forwarding address
552	The mailbox that you are trying to reach has run out of space, try again later
553	The mail address that you specified was not syntactically correct
554	The mail transaction has failed for unknown causes

Sendmail is an MTA distributed as a monolithic binary file.
Previous versions used to run SUID `root`, which caused many security problems; recent versions run SGID `smmsp`, the group that has write access on the mail queue.
Sendmail uses `smrsh`, a restricted shell, to run some external programs.

Configuration files (must not be edited by hand):

`/etc/mail/`	`submit.cf`	Sendmail local mail transfer configuration file
	`sendmail.cf`	Sendmail MTA configuration file

`m4 /etc/mail/submit.mc > /etc/mail/submit.cf`	Generate a `.cf` configuration file from an editable `.mc` text file

Database files (must not be edited by hand):

`/etc/mail/`	`access.db`	Access control file to allow or deny access to systems or users
	`local-host-names.db`	List of domains that must be considered as local accounts
	`virtusertable.db`	Map for local accounts, used to distribute incoming email
	`mailertable.db`	Routing table, used to dispatch emails from remote systems
	`domaintable.db`	Domain table, used for transitions from an old domain to a new one
	`genericstable.db`	Map for local accounts, used to specify a different sender for outgoing mail
	`genericsdomain.db`	Local FQDN

`makemap hash /etc/mail/access.db < /etc/mail/access`	Generate a `.db` database file from an editable text file

Temporary mailqueue files (where *nnn* is the Message ID):

`/var/spool/mqueue/`	`df`*nnn*	Mail body
	`qf`*nnn*	Message envelope with headers and routing information
	`Qf`*nnn*	Message envelope if abandoned
	`hf`*nnn*	Message envelope if held / quarantined by a milter (i.e. mail filter)
	`tf`*nnn*	Temporary file
	`lf`*nnn*	Lock file
	`nf`*nnn*	Backup file
	`xf`*nnn*	Transcript of delivery attempts

`newaliases`
`sendmail -bi`
	Update the aliases database. Must be run after any change to `/etc/aliases`

`mailq`
`sendmail -bp`
	Examine the mail queue

`sendmail -bt`	Run Sendmail in test mode
`sendmail -q`	Force a queue run

`hoststat`	Print statistics about remote hosts usage
`purgestat`	Clear statistics about remote host usage
`mailstats`	Print statistics about the mailserver
`praliases`	Display email aliases

Exim is a free MTA, distributed under open source GPL license.

/etc/exim.conf /usr/local/etc/exim/configure (FreeBSD)	Exim4 configuration file

exim4 -bp	Examine the mail queue
exim4 -M *messageID*	Attempt delivery of message
exim4 -Mrm *messageID*	Remove a message from the mail queue
exim4 -Mvh *messageID*	See the headers of a message in the mail queue
exim4 -Mvb *messageID*	See the body of a message in the mail queue
exim4 -Mvc *messageID*	See a message in the mail queue
exim4 -qf *domain*	Force a queue run of all queued messages for a *domain*
exim4 -Rff *domain*	Attempt delivery of all queued messages for a *domain*
exim4 -bV	Show version and other info
exinext	Give the times of the next queue run
exigrep	Search through Exim logfiles
exicyclog	Rotate Exim logfiles

Postfix is a fast, secure, easy to configure, open source MTA intended as a replacement for Sendmail. It is implemented as a set of small helper daemons, most of which run in a chroot jail with low privileges. The main ones are:

`master`	Postfix master daemon, always running; starts the other daemons when necessary
`nqmgr`	Queue manager for incoming and outgoing mail, always running
`smtpd`	SMTP daemon for incoming mail
`smtp`	SMTP daemon for outgoing mail
`bounce`	Manager of bounce messages
`cleanup`	Daemon that verifies the syntax of outgoing messages before they are handed to the queue manager
`local`	Daemon that handles local mail delivery
`virtual`	Daemon that handles mail delivery to virtual users

`/var/spool/postfix/`	`incoming`	Incoming queue. All new mail entering the Postfix queue is written here by the cleanup daemon. Under normal conditions this queue is nearly empty
	`active`	Active queue. Contains messages ready to be sent. The queue manager places messages here from the incoming queue as soon as they are available
	`deferred`	Deferred queue. A message is placed here when all its deliverable recipients are delivered, and delivery failed for some recipients for a transient reason. The queue manager scans this queue periodically and puts some messages back into the active queue to retry sending
	`bounce`	Message delivery status report about why mail is bounced (non-delivered mail)
	`defer`	Message delivery status report about why mail is delayed (non-delivered mail)
	`trace`	Message delivery status report (delivered mail)

`postfix reload`	Reload configuration
`postconf -e 'mydomain = example.org'`	Edit a setting in the Postfix configuration
`postconf -l`	List supported mailbox lock methods
`postconf -m`	List supported database types
`postconf -v`	Increase logfile verbosity
`postmap` *`dbtype:textfile`*	Manage Postfix lookup tables, creating a hashed map file of database type *dbtype* from *textfile*
`postmap hash:/etc/postfix/transport`	Regenerate the transport database
`postalias`	Convert `/etc/aliases` into the aliases database file `/etc/aliases.db`
`postsuper`	Operate on the mail queue
`postqueue`	Unprivileged mail queue manager

`/etc/postfix/main.cf`	**Postfix main configuration file**
`mydomain = example.org`	This system's domain
`myorigin = $mydomain`	Domain from which all sent mail will appear to originate
`myhostname = foobar.$mydomain`	This system's hostname
`inet_interfaces = all`	Network interface addresses that this system receives mail on. Value can also be `localhost`, `all`, or `loopback-only`
`proxy_interfaces = 1.2.3.4`	Network interface addresses that this system receives mail on by means of a proxy or NAT unit
`mynetworks = 10.3.3.0/24 !10.3.3.66`	Networks the SMTP clients are allowed to connect from

`mydestination = $myhostname, localhost,`
`   $mydomain, example.com,`
`   hash:/etc/postfix/otherdomains`
	Domains for which Postfix will accept received mail.
	Value can also be a lookup database file e.g. a hashed map

`relayhost = 10.6.6.6`	Relay host to which Postfix should send all mail for delivery, instead of consulting DNS MX records
`relay_domains = $mydestination`	Sources and destinations for which mail will be relayed. Can be empty if Postfix is not intended to be a mail relay

`virtual_alias_domains = virtualex.org`
`virtual_alias_maps = /etc/postfix/virtual`
or
`virtual_alias_domains = hash:/etc/postfix/virtual`
	Set up Postfix to handle mail for virtual domains too.
	The `/etc/postfix/virtual` file is a hashed map, each line of the file containing the virtual domain email address and the destination real domain email address:
	`jdoe@virtualex.org    john.doe@example.org`
	`ksmith@virtualex.org  kim.smith`
	`@virtualex.org        root`
	The `@virtualex.org` in the last line is a catch-all specifying that all other email messages to the virtual domain are delivered to the root user on the real domain

`mailbox_command = /usr/bin/procmail`	Use Procmail as MDA
A line beginning with whitespace or tab is a continuation of the previous line. A line beginning with a `#` is a comment. A `#` not placed at the beginning of a line is not a comment delimiter.	

`/etc/postfix/master.cf` Postfix master daemon configuration file

```
# service type private unpriv chroot wakeup maxproc command + args
smtp      inet n       -      -      -      -       smtpd
pickup    fifo n       -      -      60     1       pickup
cleanup   unix n       -      -      -      0       cleanup
qmgr      fifo n       -      -      300    1       qmgr
rewrite   unix -       -      -      -      -       trivial-rewrite
bounce    unix -       -      -      -      0       bounce
defer     unix -       -      -      -      0       bounce
flush     unix n       -      -      1000?  0       flush
smtp      unix -       -      -      -      -       smtp
showq     unix n       -      -      -      -       showq
error     unix -       -      -      -      -       error
local     unix -       n      n      -      -       local
virtual   unix -       n      n      -      -       virtual
lmtp      unix -       -      n      -      -       lmtp
```

service	Name of the service
type	Transport mechanism used by the service
private	Whether the service is accessible only by Postfix daemons and not by the whole system. Default is yes
unprivileged	Whether the service is unprivileged i.e. not running as root. Default is yes
chroot	Whether the service is chrooted. Default is yes
wakeup	How often the service needs to be woken up by the master daemon. Default is never
maxproc	Max number of simultaneous processes providing the service. Default is 50
command	Command used to start the service
The - indicates that an option is set to its default value.	

Procmail is a regex-based MDA whose main purpose is to preprocess and sort incoming email messages.
It is able to work both with the standard mbox format and the Maildir format.

To have all email processed by Procmail, the `~/.forward` file may be edited to contain:
`"|exec /usr/local/bin/procmail || exit 75"`

`/etc/procmailrc`	System-wide recipes
`~/.procmailrc`	User's recipes
`procmail -h`	List all Procmail flags for recipes
`formail`	Utility for email filtering and editing
`lockfile`	Utility for mailbox file locking
`mailstat`	Utility for generation of reports from Procmail logs

`/etc/procmailrc` and `~/.procmailrc`	Procmail recipes

`PATH=$HOME/bin:/usr/bin:/bin:/usr/sbin:/sbin`
`MAILDIR=$HOME/Mail`
`DEFAULT=$MAILDIR/Inbox`
`LOGFILE=$HOME/.procmaillog`
	Common parameters, nonspecific to Procmail

`:0h:` or `:0:`
`* ^From: .*(alice|bob)@foobar\.org`
`$DEFAULT`
	Flag: match headers (default) and use file locking (highly recommended when writing to a file or a mailbox in mbox format)
	Condition: match the header specifying the sender address
	Destination: default mailfolder

`:0:`
`* ^From: .*owner@listserv\.com`
`* ^Subject:.*Linux`
`$MAILDIR/Geekstuff1`
	Conditions: match sender address and subject headers
	Destination: specified mailfolder, in mbox format

`:0`
`* ^From: .*owner@listserv\.com`
`* ^Subject:.*Linux`
`$MAILDIR/Geekstuff2/`
	Flag: file locking not necessary because using Maildir format
	Conditions: match sender address and subject headers
	Destination: specified mailfolder, in Maildir format

`# Blacklisted by SpamAssassin`
`:0`
`* ^X-Spam-Status: Yes`
`/dev/null`
	Flag: file locking not necessary because blackholing to `/dev/null`
	Condition: match SpamAssassin's specific header
	Destination: delete the message

`:0B:`
`* hacking`
`$MAILDIR/Geekstuff`
	Flag: match body of message instead of headers

`:0HB:`
`* hacking`
`$MAILDIR/Geekstuff`
	Flag: match either headers or body of message

`:0:`
`* > 256000`
`| /root/myprogram`
	Condition: match messages larger than 256 Kb
	Destination: pipe message through the specified program

`:0fw`
`* ^From: .*@foobar\.org`
`| /root/myprogram`
	Flags: use the pipe as a filter (modifying the message), and have Procmail wait that the filter finished processing the message

`:0c`
`* ^Subject:.*administration`
`! secretary@domain.com`
 	
`:0:`
`$MAILDIR/Forwarded`
	Flag: copy the message and proceed with next recipe
	Destination: forward to specified email address, and (this is ordered by the next recipe) save in the specified mailfolder

The Courier MTA provides modules for ESMTP, IMAP, POP3, webmail, and mailing list services in a single framework. To use Courier, it is necessary first to launch the `courier-authlib` service, then launch the desired mail service e.g. `courier-imap` for the IMAP service.

`/usr/lib/courier-imap/etc/`
or
`/etc/courier/`
	`imapd`	Courier IMAP daemon configuration

	`imapd-ssl`	Courier IMAPS daemon configuration
	`pop3d`	Courier POP3 daemon configuration
	`pop3d-ssl`	Courier POP3S daemon configuration

`/usr/lib/courier-imap/share/`	Directory for public and private keys
`mkimapdcert`	Generate a certificate for the IMAPS service
`mkpop3dcert`	Generate a certificate for the POP3 service

`makealiases`	Create system aliases in `/usr/lib/courier/etc/aliases.dat`, which is made by processing a `/usr/lib/courier/etc/aliases/system` text file:
	`root          : postmaster`
	`mailer-daemon : postmaster`
	`MAILER-DAEMON : postmaster`
	`uucp          : postmaster`
	`postmaster    : admin`

`/usr/lib/courier-imap/etc/pop3d`	**Courier POP configuration file**
`ADDRESS=0`	Address on which to listen. 0 means all addresses
`PORT=127.0.0.1.900,192.168.0.1.900`	Port number on which connections are accepted. In this case, accept connections on port 900 on IP addresses 127.0.0.1 and 192.168.0.1
`POP3AUTH="LOGIN CRAM-MD5 CRAM-SHA1"`	POP authentication advertising SASL (Simple Authentication and Security Layer) capability, with CRAM-MD5 and CRAM-SHA1
`POP3AUTH_TLS="LOGIN PLAIN"`	Also advertise SASL PLAIN if SSL is enabled
`MAXDAEMONS=40`	Maximum number of POP3 servers started
`MAXPERIP=4`	Maximum number of connections to accept from the same IP address
`PIDFILE=/var/run/courier/pop3d.pid`	PID file
`TCPDOPTS="-nodnslookup -noidentlookup"`	Miscellaneous `couriertcpd` options. Should not be changed
`LOGGEROPTS="-name=pop3d"`	Options for `courierlogger`
`POP3_PROXY=0`	Enable or disable proxying
`PROXY_HOSTNAME=myproxy`	Override value from `gethostname()` when checking if a proxy connection is required
`DEFDOMAIN="@example.com"`	Optional default domain. If the username does not contain the first character of `DEFDOMAIN`, then it is appended to the username. If `DEFDOMAIN` and `DOMAINSEP` are both set, then `DEFDOMAIN` is appended only if the username does not contain any character from `DOMAINSEP`
`POP3DSTART=YES`	Flag intended to be read by the system startup script
`MAILDIRPATH=Maildir`	Maildir directory

/usr/lib/courier-imap/etc/imapd	**Courier IMAP configuration file**
ADDRESS=0	Address on which to listen. 0 means all addresses
PORT=127.0.0.1.900,192.168.0.1.900	Port number on which connections are accepted. In this case, accept connections on port 900 on IP addresses 127.0.0.1 and 192.168.0.1
AUTHSERVICE143=imap	Authenticate using a different service parameter depending on the connection's port. This only works with authentication modules that use the service parameter, such as PAM
MAXDAEMONS=40	Maximum number of IMAP servers started
MAXPERIP=20	Maximum number of connections to accept from the same IP address
PIDFILE=/var/run/courier/imapd.pid	PID file for couriertcpd
TCPDOPTS="-nodnslookup -noidentlookup"	Miscellaneous couriertcpd options. Should not be changed
LOGGEROPTS="-name=imapd"	Options for courierlogger
DEFDOMAIN="@example.com"	Optional default domain. If the username does not contain the first character of DEFDOMAIN, then it is appended to the username. If DEFDOMAIN and DOMAINSEP are both set, then DEFDOMAIN is appended only if the username does not contain any character from DOMAINSEP

IMAP_CAPABILITY="IMAP4rev1 UIDPLUS \\
CHILDREN NAMESPACE THREAD=ORDEREDSUBJECT \\
THREAD=REFERENCES SORT QUOTA IDLE"
	Specifies what most of the response should be to the CAPABILITY command

IMAP_KEYWORDS=1	Enable or disable custom IMAP keywords. Possible values are: 0 disable keywords 1 enable keywords 2 enable keywords with a slower algorithm
IMAP_ACL=1	Enable or disable IMAP ACL extension
SMAP_CAPABILITY=SMAP1	Enable the experimental Simple Mail Access Protocol extensions
IMAP_PROXY=0	Enable or disable proxying
IMAP_PROXY_FOREIGN=0	Proxying to non-Courier servers. Resends the CAPABILITY command after logging in to remote server. May not work with all IMAP clients
IMAP_IDLE_TIMEOUT=60	How often, in seconds, the server should poll for changes to the folder while in IDLE mode
IMAP_CHECK_ALL_FOLDERS=0	Enable or disable server check for mail in every folder
IMAP_UMASK=022	Set the umask of the server process. This value is passed to the umask command. Mostly useful for shared folders, where file permissions of the messages may be important
IMAP_ULIMITD=131072	Set the upper limit of the size of the data segment of the server process, in Kb. This value is passed to the ulimit -d command. Used as an additional safety check to stop potential DoS attacks that exploit memory leaks to exhaust all the available RAM on the server
IMAP_USELOCKS=1	Enable or disable dot-locking to support concurrent multiple access to the same folder. Strongly recommended when using shared folders
IMAP_SHAREDINDEXFILE=\\ /etc/courier/shared/index	Index of all accessible folders. This setting should normally not be changed
IMAP_TRASHFOLDERNAME=Trash	Trash folder
IMAP_EMPTYTRASH=Trash:7,Sent:30	Purge folders i.e. delete all messages from the specified folders after the specified number of days
IMAP_MOVE_EXPUNGE_TO_TRASH=0	Enable or disable moving expunged messages to the trash folder (instead of directly deleting them)
HEADERFROM=X-IMAP-Sender	Save the return address ($SENDER) in the X-IMAP-Sender mail header. This header is added to the sent message, but not in the copy of the message saved in the folder
MAILDIRPATH=Maildir	Mail directory

Dovecot is an open source, security-hardened, fast, and efficient IMAP and POP3 server.
It implements its own high-performance dbox mailbox format. By default, it uses PAM authentication.
The script `mkcert.sh` can be used to create self-signed SSL certificates.

/etc/dovecot.conf	Dovecot configuration file
base_dir = /var/run/dovecot/	Base directory where to store runtime data
protocols = imaps pop3s	Protocols to serve. If Dovecot should use dovecot-auth, this can be set to none
listen = *, [::]	Network interfaces on which to accept connections. In this case, listen to all IPv4 and IPv6 interfaces
disable_plaintext_auth = yes	If yes, disable LOGIN command and all other plaintext authentications unless SSL/TLS is used (LOGINDISABLED capability)
shutdown_clients = yes	If yes, kill all IMAP and POP3 processes when Dovecot master process shuts down; if no, Dovecot can be upgraded without forcing existing client connections to close
log_path = /dev/stderr	Log file to use for error messages, instead of sending them to syslog. In this case, log to stderr
info_log_path = /dev/stderr	Log file to use for informational and debug messages. Default value is the same as log_path
syslog_facility = mail	Syslog facility to use, if logging to syslog
login_dir = /var/run/dovecot/login	Directory where the authentication process places authentication UNIX sockets. The login process needs to be able to connect to these sockets
login_chroot = yes	Chroot login process to the login_dir
login_user = dovecot	User for the login process and for access control in the authentication process. This is not the user that will access mail messages
login_process_size = 64	Maximum login process size, in Mb
login_process_per_connection = yes	If yes, each login is processed in its own process (more secure); if no, each login process processes multiple connections (faster)
login_processes_count = 3	Number of login processes to keep for listening for new connections
login_max_processes_count = 128	Maximum number of login processes to create
login_max_connections = 256	Maximum number of connections allowed per each login process. This setting is used only if login_process_per_connection = no; once the limit is reached, the process notifies master so that it can create a new login process
login_greeting = Dovecot ready.	Greeting message for clients
login_trusted_networks = \ 10.7.7.0/24 10.8.8.0/24	Trusted network ranges (usually IMAP proxy servers). Connections from these IP addresses are allowed to override their IP addresses and ports, for logging and authentication checks. disable_plaintext_auth is also ignored for these networks

mbox_read_locks = fcntl
mbox_write_locks = dotlock fcntl
	Locking methods to use for locking mailboxes in mbox format.
	Possible values are:
	dotlock Create *mailbox*.lock file; oldest and NSF-safe method
	dotlock_try Same as dotlock, but skip if failing
	fcntl Recommended; works with NFS too if lockd is used
	flock May not exist in all systems; doesn't work with NFS
	lockf May not exist in all systems; doesn't work with NFS

maildir_stat_dirs = no	Option for mailboxes in Maildir format. If no (default), the LIST command returns all entries in the mail directory beginning with a dot; if yes, returns only entries which are directories

dbox_rotate_size = 2048
dbox_rotate_min_size = 16
	Maximum and minimum file size, in Kb, of a mailbox in dbox format until it is rotated

!include /etc/dovecot/conf.d/*.conf	Include configuration file
!include_try /etc/dovecot/extra.conf	Include optional configuration file, and do not report an error if file is not found

`/etc/dovecot.conf`	**Dovecot configuration file**

`mail_location = \`
`mbox:~/mail:INBOX=/var/spool/mail/%u`
or
`mail_location = maildir:~/Maildir`
	Mailbox location, in mbox or Maildir format. Variables:
	`%u` username
	`%n` user part in *user@domain*, same as `%u` if there is no domain
	`%d` domain part in *user@domain*, empty if there is no domain
	`%h` home directory

`namespace shared {`	Definition of a shared namespace, for accessing other users' mailboxes that have been shared. Private namespaces are for users' personal emails. Public namespaces are for shared mailboxes managed by root user
`   separator = /`	Hierarchy separator to use. It should be the same for all namespaces, and depends on the underlying mail storage format
`   prefix = shared/%%u/`	Prefix required to access this namespace; must be different for each. In this case, mailboxes are visible under `shared/`*`user@domain`*`/`; the variables `%%n`, `%%d`, and `%%u` are expanded to the destination user

`   location = maildir:%%h/Maildir:\`
`      INDEX=~/Maildir/shared/%%u`
	Mailbox location for other users' mailboxes; it is in the same format as `mail_location` which is also the default for it.
	`%`*`variable`* and `~/` expand to the logged in user's data;
	`%%`*`variable`* expands to the destination user's data

`   inbox = no`	Define whether this namespace contains the INBOX. Note that there can be only one INBOX across all namespaces
`   hidden = no`	Define whether the namespace is hidden i.e. not advertised to clients via NAMESPACE extension
`   subscriptions = no`	Namespace handles its own subscriptions; if set to no, the parent namespace handles them and Dovecot uses the default namespace for saving subscriptions. If `prefix` is empty, this should be set to yes
`   list = children`	Show the mailboxes under this namespace with LIST command, making the namespace visible for clients that do not support the NAMESPACE extension. In this case, lists child mailboxes but hide the namespace prefix; list the namespace only if there are visible shared mailboxes
`}`	

`mail_uid = 666`
`mail_gid = 666`
	UID and GID used to access mail messages

`mail_privileged_group = mail`	Group to enable temporarily for privileged operations. Currently this is used only with INBOX when its initial creation or a dotlocking fails
`mail_access_groups = tmpmail`	Supplementary groups to with grant access for mail processes. Used typically to set up access to shared mailboxes
`lock_method = fcntl`	Locking method for index files. Can be `fcntl`, `flock`, or `dotlock`

`first_valid_uid = 500`
`last_valid_uid = 0`
	Valid UID range for users; default is 500 and above. This makes sure that users cannot login as daemons or other system users.
	Denying root login is hardcoded to Dovecot and cannot be bypassed

`first_valid_gid = 1`
`last_valid_gid = 0`
	Valid GID range for users; default is non-root.
	Users with invalid primary GID are not allowed to login

`max_mail_processes = 512`	Maximum number of running mail processes. When this limit is reached, new users are not allowed to login
`mail_process_size = 256`	Maximum mail process size, in Mb
`valid_chroot_dirs =`	List of directories under which chrooting is allowed for mail processes
`mail_chroot =`	Default chroot directory for mail processes. Usually not needed as Dovecot does not allow users to access files outside their mail directory
`mailbox_idle_check_interval = 30`	Minimum time, in seconds, to wait between mailbox checks. When the IDLE command is running, mailbox is checked periodically for new mails or other changes

`/etc/dovecot.conf`	**Dovecot configuration file**
`protocol pop3 {`	Block with options for the POP3 protocol
`listen = *:110`	Network interfaces on which to accept POP3 connections
`login_executable = /usr/libexec/dovecot/pop3-login`	Location of the POP3 login executable
`mail_executable = /usr/libexec/dovecot/pop3`	Location of the POP3 mail executable
`pop3_no_flag_updates = no`	If set to no, do not try to set mail messages non-recent or seen with POP3 sessions, to reduce disk I/O. With Maildir format do not move files from `new/` to `cur/`; with mbox format do not write `Status-` headers
`pop3_lock_session = no`	Defines whether to keep the mailbox locked for the whole POP3 session
`pop3_uidl_format = %08Xu%08Xv`	POP3 UIDL (Unique Mail Identifier) format to use
`}`	
`protocol imap {`	Block with options for the IMAP protocol

`listen = *:143`
`ssl_listen = *:993`
	Network interfaces on which to accept IMAP and IMAPS connections

`login_executable = /usr/libexec/dovecot/imap-login`	Location of the IMAP login executable
`mail_executable = /usr/libexec/dovecot/imap`	Location of the IMAP mail executable
`mail_max_userip_connections = 10`	Maximum number of IMAP connections allowed for a user from each IP address
`imap_idle_notify_interval = 120`	Waiting time, in seconds, between "OK Still here" notifications when client is IDLE
`}`	
`ssl = yes`	SSL/TLS support. Possible values are `yes`, `no`, `required`
`ssl_cert_file = /etc/ssl/certs/dovecot-cert.pem`	Location of the SSL certificate
`ssl_key_file = /etc/ssl/private/dovecot-key.pem`	Location of private key
`ssl_key_password = p4ssw0rd`	Password of private key, if it is password-protected. Since `/etc/dovecot.conf` is usually world-readable, it is better to place this setting into a root-owned 0600 file instead and include it via the setting `!include_try /etc/dovecot/dovecot-passwd.conf`. Alternatively, Dovecot can be started with `dovecot -p p4ssw0rd`
`ssl_ca_file = /etc/dovecot/cafile.pem`	List of trusted SSL certificate authorities. This file contains CA certificates followed by CRLs
`ssl_verify_client_cert = yes`	Request client to send a certificate
`ssl_cipher_list = ALL:!LOW:!SSLv2`	List of SSL ciphers to use
`verbose_ssl = yes`	Show protocol level SSL errors

/etc/dovecot.conf	Dovecot configuration file
auth_executable = /usr/libexec/dovecot/dovecot-auth	Location of the authentication executable
auth_process_size = 256	Max authentication process size, in Mb
auth_username_chars = abcde ... VWXYZ01234567890.-_@	List of allowed characters in the username. If the username entered by the user contains a character not listed in here, the login automatically fails. This is to prevent a user exploiting any potential quote-escaping vulnerabilities with SQL/LDAP databases
auth_realms =	List of realms for SASL authentication mechanisms that need them. If empty, multiple realms are not supported
auth_default_realm = example.org	Default realm/domain to use if none was specified
auth_anonymous_username = anonymous	Username to assign to users logging in with ANONYMOUS SASL mechanism
auth_verbose = no	Defines whether to log unsuccessful authentication attempts and the reasons why they failed
auth_debug = no	Define whether to enable more verbose logging (e.g. SQL queries) for debugging purposes
auth_failure_delay = 2	Delay before replying to failed authentications, in seconds
auth default {	
mechanisms = plain login cram-md5	Accepted authentication mechanisms

passdb passwd-file {
args = /etc/dovecot.deny
deny = yes
}
	Deny login to the users listed in /etc/dovecot.deny (this file contains one user per line)

passdb pam {
args = cache_key=%u%r dovecot
}
	PAM authentication block.
	Enables authentication matching (username and remote IP address) for PAM

passdb passwd {
blocking = yes
args =
}
	System users e.g. NSS or /etc/passwd

passdb shadow {
blocking = yes
args =
}
	Shadow passwords for system users, e.g. NSS or /etc/passwd

passdb bsdauth {
cache_key = %u
args =
}
	PAM-like authentication for OpenBSD

passdb sql {
args = /etc/dovecot/dovecot-sql.conf
}
	SQL database

passdb ldap {
args = /etc/dovecot/dovecot-ldap.conf
}
	LDAP database

socket listen {
master {
path = /var/run/dovecot/auth-master
mode = 0600
user =
group =
}
client {
path = /var/run/dovecot/auth-client
mode = 0660
}
}
	Export the authentication interface to other programs.
	Master socket provides access to userdb information, and is typically used to give Dovecot's local delivery agent access to userdb so it can find mailbox locations. The default user/group is the one who started dovecot-auth (i.e. root).
	The client socket is generally safe to export to everyone. Typical use is to export it to the SMTP server so it can do SMTP AUTH lookups using it

}	

FTP (File Transfer Protocol) is a client-server unencrypted protocol for file transfer. Secure alternatives are FTPS (FTP secured with SSL/TLS) and SFTP (SSH File Transfer Protocol). It can operate either in active or in passive mode.

Active mode (default)
1. Client connects to FTP server on port 21 (control channel) and sends second unprivileged port number
2. Server acknowledges
3. Server connects from port 20 (data channel) to client's second unprivileged port number
4. Client acknowledges

Passive mode (more protocol-compliant, because it is the client that initiates the connection)
1. Client connects to FTP server on port 21 and requests passive mode via the PASV command
2. Server acknowledges and sends unprivileged port number via the PORT command
3. Client connects to server's unprivileged port number
4. Server acknowledges

FTP servers		
Very Secure FTP	Hardened and high-performance FTP implementation. The `vsftpd` daemon operates with multiple processes that run as a non-privileged user in a chrooted jail	
Pure-FTP	Free and easy-to-use FTP server	
	`pure-ftpd`	Pure-FTP daemon
	`pure-ftpwho`	Show clients connected to the Pure-FTP server
	`pure-mrtginfo`	Show connections to the Pure-FTP server as a MRTG graph
	`pure-statsdecode`	Show Pure-FTP log data
	`pure-pw`	Manage Pure-FTP virtual accounts
	`pure-pwconvert`	Convert the system user database to a Pure-FTP virtual accounts database
	`pure-quotacheck`	Manage Pure-FTP quota database
	`pure-uploadscript`	Run a command on the Pure-FTP server to process an uploaded file
FTP clients		
ftp	Standard FTP client	
	`ftp` *`ftpserver.domain.com`*	Connect to an FTP server
lftp	Sophisticated FTP client with support for HTTP and BitTorrent	
	`lftp` *`ftpserver.domain.com`*	Connect to an FTP server and try an anonymous login

`/etc/vsftpd/vsftpd.conf`	**Very Secure FTP server configuration file**
`listen=NO`	Run `vsftpd` in standalone mode (i.e. not via inetd)?
`local_enable=YES`	Allow local system users (i.e. in `/etc/passwd`) to log in?
`chroot_local_user=YES`	Chroot local users in their home directory?
`write_enable=YES`	Allow FTP commands that write on the filesystem (i.e. STOR, DELE, RNFR, RNTO, MKD, RMD, APPE, and SITE)?
`anonymous_enable=YES`	Allow anonymous logins? If yes, `anonymous` and `ftp` are accepted as logins
`anon_root=/var/ftp/pub`	Directory to go after anonymous login
`anon_upload_enable=YES`	Allow anonymous uploads?
`chown_uploads=YES`	Change ownership of anonymously uploaded files?
`chown_username=ftp`	User to whom set ownership of anonymously uploaded files
`anon_world_readable_only=NO`	Allow anonymous users to only download world-readable files?
`ssl_enable=YES`	Enable SSL?
`force_local_data_ssl=NO`	Encrypt local data?
`force_local_logins_ssl=YES`	Force encrypted authentication?
`allow_anon_ssl=YES`	Allow anonymous users to use SSL?

`ssl_tlsv1=YES`
`ssl_tlsv2=NO`
`ssl_tlsv3=NO`
	Allowed SSL/TLS versions

`rsa_cert_file=/etc/pki/tls/certs/vsftpd.pem`	Location of certificate file
`rsa_private_key_file=/etc/pki/tls/certs/vsftpd.pem`	Location of private key file

In Linux, printers are managed by `cupsd`, the CUPS (Common Unix Printing System) daemon.
Printers are administered via a web interface on the URL http://localhost:631.

`/etc/cups/cupsd.conf`	CUPS configuration file
`/etc/cups/printers.conf`	Database of available local CUPS printers
`/etc/printcap`	Database of printer capabilities, for old printing applications
`/var/spool/cups/`	Printer spooler for data awaiting to be printed
`/var/log/cups/error_log`	CUPS error log
`/var/log/cups/page_log`	Information about printed pages
`/etc/init.d/cupsys start`	Start the CUPS service
`gnome-cups-manager`	Run the CUPS Manager graphical application
`cupsenable` *printer0*	Enable a CUPS printer
`cupsdisable` *printer0*	Disable a CUPS printer
`cupsaccept` *printer0*	Accept a job sent on a printer queue
`cupsreject -r "`*Message*`"` *printer0*	Reject a job sent on a printer queue, with an informational message
`cupstestppd` *LEXC510.ppd*	Test the conformance of a PPD file to the format specification
`cupsaddsmb` *printer0*	Export a printer to Samba (for use with Windows clients)
`cups-config --cflags`	Show the necessary compiler options
`cups-config --datadir`	Show the default CUPS data directory
`cups-config --ldflags`	Show the necessary linker options
`cups-config --libs`	Show the necessary libraries to link to
`cups-config --serverbin`	Show the default CUPS binaries directory that stores filters and backends
`cups-config --serverroot`	Show the default CUPS configuration file directory
`lpstat`	Show CUPS status information
`lpadmin`	Administer CUPS printers
`lpadmin -p` *printer0* `-P` *LEXC750.ppd*	Specify a PPD (Adobe PostScript Printer Description) file to associate to a printer
`lp -d` *printer0 file*	Print a file on the specified printer
`lpq`	View the default print queue
`lpq -P` *printer0*	View a specific print queue
`lpq` *user*	View the print queue of a specific user
`lprm -P` *printer0 jobnumber*	Delete a specific job from a printer queue
`lprm -P` *printer0 user*	Delete all jobs from a specific user from a printer queue
`lprm -P` *printer0* `-`	Delete all jobs from a printer queue
`lpc`	Manage print queues
`a2ps` *file.txt*	Convert a text file to PostScript
`ps2pdf` *file.ps*	Convert a file from PostScript to PDF
`mpage` *file.ps*	Print a PostScript document on multiple pages per sheet on a PostScript printer
`gv` *file.ps*	View a PostScript document (the `gv` software is a derivation of GhostView)

IPv4 addressing

		Address range	Prefix	Number of addresses	Reference
Classful	Class A (Unicast)	0.0.0.0 – 127.255.255.255 first octet: 0XXX XXXX	/8	128 networks × 16,777,216 addresses	RFC 791
	Class B (Unicast)	128.0.0.0 – 191.255.255.255 first octet: 10XX XXXX	/16	16,384 networks × 65,536 addresses	RFC 791
	Class C (Unicast)	192.0.0.0 – 223.255.255.255 first octet: 110X XXXX	/24	2,097,152 networks × 256 addresses	RFC 791
	Class D (Multicast)	224.0.0.0 – 239.255.255.255 first octet: 1110 XXXX	/4	268,435,456	RFC 3171
	Class E (Experimental)	240.0.0.0 – 255.255.255.255 first octet: 1111 XXXX	/4	268,435,456	RFC 1166
Private	Private Class A	10.0.0.0 – 10.255.255.255	10.0.0.0/8	16,777,216	RFC 1918
	Private Class B	172.16.0.0 – 172.31.255.255	172.16.0.0/12	1,048,576	RFC 1918
	Private Class C	192.168.0.0 – 192.168.255.255	192.168.0.0/16	65,536	RFC 1918
Reserved	Source	0.0.0.0 – 0.255.255.255	0.0.0.0/8	16,777,216	RFC 1700
	Loopback	127.0.0.0 – 127.255.255.255	127.0.0.0/8	16,777,216	RFC 1700
	Autoconf	169.254.0.0 – 169.254.255.255	169.254.0.0/16	65,536	RFC 3330
	TEST-NET	192.0.2.0 – 192.0.2.255	192.0.2.0/24	256	RFC 3330
	6to4 relay anycast	192.88.99.0 – 192.88.99.255	192.88.99.0/24	256	RFC 3068
	Device benchmarks	198.18.0.0 – 198.19.255.255	198.18.0.0/15	131,072	RFC 2544

An IPv4 address is 32-bit long, and is represented divided in four octets (dotted-quad notation), e.g. 193.22.33.44.

There are approximately 4×10^{9} total possible IPv4 addresses.

IPv4 classful addressing is obsolete and has been replaced by CIDR (Classless Inter-Domain Routing).

IPv6 addressing

Unicast	64-bit network prefix (>= 48-bit routing prefix + <= 16-bit subnet id) + 64-bit interface identifier A 48-bit MAC address is transformed into a 64-bit EUI-64 by inserting ff:fe in the middle. A EUI-64 is then transformed into an IPv6 interface identifier by inverting the 7th most significant bit.
Link-local	fe80:0000:0000:0000 + 64-bit interface identifier
Multicast	ff + 4-bit flag + 4-bit scope field + 112-bit group ID

An IPv6 address is 128-bit long, and is represented divided in eight 16-bit groups (4 hex digits).
Leading zeros in each group can be deleted. A single chunk of one or more adjacent 0000 groups can be deleted.
e.g. 2130:0000:0000:0000:0007:0040:15bc:235f which can also be written as 2130::7:40:15bc:235f.

There are approximately 3×10^{38} total possible IPv6 addresses.

The IANA (Internet Assigned Numbers Authority) manages the allocation of IPv4 and IPv6 addresses, assigning large blocks to RIRs (Regional Internet Registries) which in turn allocate addresses to ISPs (Internet Service Providers) and other local registries. These address blocks can be searched via a WHOIS query to the appropriate RIR, which is:

AFRINIC	for Africa
ARIN	for US, Canada, and Antarctica
APNIC	for Asia and Oceania
LACNIC	for Latin America
RIPE NCC	for Europe, Middle East, and Russia

VLSM chart - Last octet subnetting (CIDR notation)						
Prefix: /24 Netmask: .0 00000000 1 subnet 254 hosts each 254 total hosts	**Prefix: /25 Netmask: .128 10000000 2 subnets 126 hosts each 252 total hosts**	**Prefix: /26 Netmask: .192 11000000 4 subnets 62 hosts each 248 total hosts**	**Prefix: /27 Netmask: .224 11100000 8 subnets 30 hosts each 240 total hosts**	**Prefix: /28 Netmask: .240 11110000 16 subnets 14 hosts each 224 total hosts**	**Prefix: /29 Netmask: .248 11111000 32 subnets 6 hosts each 192 total hosts**	**Prefix: /30 Netmask: .252 11111100 64 subnets 2 hosts each 128 total hosts**
.0	.0	.0	.0	.0	.0	.0
						.4
					.8	.8
						.12
				.16	.16	.16
						.20
					.24	.24
						.28
			.32	.32	.32	.32
						.36
					.40	.40
						.44
				.48	.48	.48
						.52
					.56	.56
						.60
		.64	.64	.64	.64	.64
						.68
					.72	.72
						.76
				.80	.80	.80
						.84
					.88	.88
						.92
			.96	.96	.96	.96
						.100
					.104	.104
						.108
				.112	.112	.112
						.116
					.120	.120
						.124
	.128	.128	.128	.128	.128	.128
						.132
					.136	.136
						.140
				.144	.144	.144
						.148
					.152	.152
						.156
			.160	.160	.160	.160
						.164
					.168	.168
						.172
				.176	.176	.176
						.180
					.184	.184
						.188
		.192	.192	.192	.192	.192
						.196
					.200	.200
						.204
				.208	.208	.208
						.212
					.216	.216
						.220
			.224	.224	.224	.224
						.228
					.232	.232
						.236
				.240	.240	.240
						.244
					.248	.248
						.252

Each block of a column identifies a subnet, whose range of valid hosts addresses is [network address +1 — broadcast address -1] inclusive.
The network address of the subnet is the number shown inside a block.
The broadcast address of the subnet is the network address of the block underneath -1 or, for the bottom block, .255.

Most common well-known ports

Port number		Service
20	TCP	FTP (data)
21	TCP	FTP (control)
22	TCP	SSH
23	TCP	Telnet
25	TCP	SMTP
53	TCP/UDP	DNS
67	UDP	BOOTP/DHCP (server)
68	UDP	BOOTP/DHCP (client)
80	TCP	HTTP
110	TCP	POP3
119	TCP	NNTP
123	UDP	NTP
139	TCP/UDP	Microsoft NetBIOS
143	TCP	IMAP
161	UDP	SNMP
443	TCP	HTTPS (HTTP over SSL/TLS)
465	TCP	SMTP over SSL
993	TCP	IMAPS (IMAP over SSL)
995	TCP	POP3S (POP3 over SSL)

1-1023: privileged ports, used server-side
1024-65535: unprivileged ports, used client-side

`/etc/services` lists all well-known ports.
Many network services are run by the `xinetd` super server.

ISO/OSI and TCP/IP protocol stack models

Layer	ISO/OSI	TCP/IP	Standards	Data transmission unit
7	Application	Application	HTTP, SMTP, POP ...	Message
6	Presentation			
5	Session			
4	Transport	Transport	TCP, UDP	Segment (TCP), Datagram (UDP)
3	Network	Internet	IPv4, IPv6, ICMP ...	Packet
2	Data Link	Network Access	Ethernet, Wi-Fi, PPP ...	Frame
1	Physical			Bit

Network configuration commands

`ip a`
`ip addr`
`ip addr show`
`ifconfig -a`
	Display configuration of all network interfaces

`ip link show eth0`
`ifconfig eth0`
	Display configuration of `eth0`

`ip addr add dev eth0 10.1.1.3/24`
`ifconfig eth0 10.1.1.3 netmask 255.255.255.0 broadcast 10.1.1.255`
	Configure IP address of `eth0`

`ifconfig eth0 hw ether 45:67:89:ab:cd:ef`	Configure MAC address of `eth0`

`ip link set eth0 up`
`ifconfig eth0 up`
`ifup eth0`
	Activate `eth0`

`ip link set eth0 down`
`ifconfig eth0 down`
`ifdown eth0`
	Shut down `eth0`

`dhclient eth0`
`pump -i eth0`
`dhcpcd eth0` (SUSE)
	Request an IP address via DHCP

`ip neigh`
`arp -a`
	Show the ARP cache table (containing mappings of MAC to IP addresses)

`ip neigh show 10.1.1.4`
`arp 10.1.1.4`
	Show the ARP cache entry for a host

`ip neigh add 10.1.1.5 lladdr 01:23:45:67:89:ab dev eth0`
`arp -s 10.1.1.5 01:23:45:67:89:ab`
	Add a new ARP entry for a host

`ip neigh del 10.1.1.5 dev eth0`
`arp -d 10.1.1.5`
	Delete an ARP entry

`ip neigh flush all`	Delete the ARP table for all interfaces
`hostname`	Get the hostname
`hostname -f`	Get the FQDN (Fully Qualified Domain Name)
`hostname mybox` `hostnamectl set-hostname --static "mybox"` (Red Hat)	Set the hostname
`hostnamectl` (Red Hat)	Get the hostname, OS, and other information
`/etc/init.d/networking restart` (Debian) `/etc/init.d/network restart` (Red Hat)	Restart network services
`ethtool` *option device*	Query or control network driver and hardware settings
`ethtool eth0`	View hardware settings of `eth0`

Network configuration files

`/etc/hosts`

Mappings between IP addresses and hostnames, for name resolution

```
127.0.0.1  localhost.localdomain  localhost
10.2.3.4   myhost.domain.org      myhost
```

`/etc/nsswitch.conf`

Sources that must be used by various system library lookup functions

```
passwd:  files nisplus nis
shadow:  files nisplus nis
group:   files nisplus nis
hosts:   files dns nisplus nis
```

`/etc/host.conf`

Sources for name resolution, for systems before glibc2.
Obsolete, superseded by `/etc/nsswitch.conf`

```
order hosts,bind
multi on
```

`/etc/resolv.conf`

Domain names that must be appended to bare hostnames, and DNS servers that will be used for name resolution

```
search domain1.org domain2.org
nameserver  192.168.3.3
nameserver  192.168.4.4
```

`/etc/networks`

Mappings between network addresses and names

```
loopback  127.0.0.0
mylan     10.2.3.0
```

`/etc/services`

List of service TCP/UDP port numbers

`/etc/protocols`

List of available protocols

`/sys/class/net`

List of all network interfaces in the system

Red Hat network configuration	

/etc/sysconfig/network	Network configuration file
	
	ADDRESS=10.2.3.4
	NETMASK=255.255.255.0
	GATEWAY=10.2.3.254
	HOSTNAME=mylinuxbox.example.org
	NETWORKING=yes

/etc/sysconfig/network-scripts/ifcfg-eth0	Configuration file for eth0.
	This file is read by the ifup and ifdown scripts
	
	DEVICE=eth0
	TYPE=Ethernet
	HWADDR=AA:BB:CC:DD:EE:FF
	BOOTPROTO=none
	ONBOOT=yes
	NM_CONTROLLED=no
	IPADDR=10.2.3.4
	NETMASK=255.255.255.0
	GATEWAY=10.2.3.254
	DNS1=8.8.8.8
	DNS2=4.4.4.4
	USERCTL=no

/etc/sysconfig/network-scripts/ifcfg-eth0:0 /etc/sysconfig/network-scripts/ifcfg-eth0:1 /etc/sysconfig/network-scripts/ifcfg-eth0:2	Multiple configuration files for a single eth0 interface, which allows binding multiple IP addresses to a single NIC
/etc/sysconfig/network-scripts/route-eth0	Static route configuration for eth0 default 10.2.3.4 dev eth0 10.7.8.0/24 via 10.2.3.254 dev eth0 10.7.9.0/24 via 10.2.3.254 dev eth0
/etc/ethertypes	Ethernet frame types. Lists various Ethernet protocol types used on Ethernet networks
Debian network configuration	

/etc/network/interfaces	List and configuration of all network interfaces
	
	allow-hotplug eth0
	iface eth0 inet static
	   address 10.2.3.4
	   netmask 255.255.255.0
	   gateway 10.2.3.254
	   dns-domain example.com
	   dns-nameservers 8.8.8.8 4.4.4.4

/etc/hostname	Hostname of the local machine
/etc/ethers	ARP mappings

In RHEL7 and later the network configuration is managed by the NetworkManager daemon.
A **connection** is a network configuration that applies to a **device** (aka network interface). A device can be included in multiple connections, but only one of them may be active at a time.
The configuration for *connection* is stored in the file `/etc/sysconfig/network-scripts/ifcfg-`*`connection`*. Although it is possible to set up networking by editing these configuration files, it is much easier to use the command `nmcli`.

`nmcli device status`	Show all network devices
`nmcli device disconnect` *`iface`*	Disconnects the device *iface*. This command should be used instead of `nmcli connection down` *`connection`* because if *connection* is set to autoconnect, Network Manager will bring the connection (and the device) up again short time later
`nmcli connection show`	Show all connections. Connections with an empty device entry are inactive
`nmcli connection show --active`	Show active connections
`nmcli connection show` *`connection`*	Show the configuration of *connection*

`nmcli connection add con-name` *`connection`* `\`
`type ethernet ifname` *`iface`* `ipv4.method manual \`
`ipv4.addresses 10.0.0.13/24 ipv4.gateway 10.0.0.254`
	Configure a new *connection* that uses the Ethernet interface *iface* and assigns it an IPv4 address and gateway

`nmcli connection modify` *`connection options`*	Modify the configuration of *connection*
`nmcli connection up` *`connection`*	Brings up a *connection*
`nmcli connection reload`	Reload any manual change made to the files `/etc/sysconfig/network-scripts/ifcfg-*`

The manpage `man nmcli-examples` contains examples of network configuration.

Network teaming allows binding together two or more network interfaces to increase throughput or provide redundancy. RHEL7 and later implement network teaming via the `teamd` daemon.

How to set up a teaming connection

1.
   ```
   nmcli connection add type team con-name teamcon ifname teamif \
   config '{"runner":{"name":"loadbalance"}}'
   ```
   Set up a team connection *teamcon* and a team interface *teamif* with a runner (in JSON code) for automatic failover

2.
   ```
   nmcli connection modify teamcon ipv4.method manual \
   ipv4.addresses 10.0.0.14/24 ipv4.gateway 10.0.0.254
   ```
   Assign manually an IP address and gateway

3.
   ```
   nmcli connection add type team-slave ifname iface \
   master teamcon
   ```
   Add an existing device *iface* as a slave of team *teamcon*.
   The slave connection will be automatically named `team-slave-`*`iface`*
4. Repeat the previous step for each slave interface.

`teamdctl` *`teamif`* `state`	Show the state of the team interface *teamif*
`teamnl` *`teamif command`*	Debug a team interface *teamif*

A **network bridge** emulates a hardware bridge, i.e. a Layer 2 device able to forward traffic between networks based on MAC addresses.

How to set up a bridge connection

1.
   ```
   nmcli connection add type bridge con-name brcon ifname brif
   ```
   Set up a bridge connection *brcon* and a bridge interface *brif*

2.
   ```
   nmcli connection modify brcon ipv4.method manual \
   ipv4.addresses 10.0.0.15/24 ipv4.gateway 10.0.0.254
   ```
   Assign manually an IP address and gateway

3.
   ```
   nmcli connection add type bridge-slave ifname iface \
   master brcon
   ```
   Add an existing device *iface* as a slave of bridge *brcon*.
   The slave connection will be automatically named `bridge-slave-`*`iface`*
4. Repeat the previous step for each slave interface.

`brctl show` *`brif`*	Display information about the bridge interface *brif*

The manpage `man teamd.conf` contains examples of team configurations and runners.
The manpage `man nmcli-examples` contains examples of teaming and bridging configuration.

`iwlist wlan0 scan`	List all wireless devices in range, with their quality of signal and other information
`iwlist wlan0 freq`	Display transmission frequency settings
`iwlist wlan0 rate`	Display transmission speed settings
`iwlist wlan0 txpower`	Display transmission power settings
`iwlist wlan0 key`	Display encryption settings

`iwgetid wlan0` *option*	Print NWID, ESSID, AP/Cell address or other information about the wireless network that is currently in use

`iwconfig wlan0`	Display configuration of wireless interface `wlan0`
`iwconfig wlan0` *option*	Configure wireless interface `wlan0`

`iw dev wlan0 station dump`	On a wireless card configured in AP Mode, display information (e.g. MAC address, tx/rx, bitrate, signal strength) about the clients

`rfkill list`	List installed wireless devices
`rfkill unblock` *n*	Enable wireless device number *n*

`hcidump -i` *device*	Display raw HCI (Host Controller Interface) data exchanged with a Bluetooth *device*

dig *example.org*	Perform a DNS lookup for the specified domain or hostname. Returns information in BIND zone file syntax; uses an internal resolver and hence does not honor /etc/resolv.conf
host *example.org* nslookup *example.org* (deprecated)	Perform a DNS lookup for the specified domain or hostname. Does honor /etc/resolv.conf
dig @*nameserver* -t MX *example.org* host -t *example.org nameserver*	Perform a DNS lookup for the MX record of the specified domain, querying *nameserver*
dig *example.org* any host -a *example.org*	Get all DNS records for a domain
dig -x *a.b.c.d* host *a.b.c.d*	Perform a reverse DNS lookup for the IP address *a.b.c.d*
whois *example.org*	Query the WHOIS service for an Internet resource (usually a domain name)
ping *host*	Test if a remote host can be reached and measure the round-trip time to it. This is done by sending an ICMP Echo Request datagram and awaiting an ICMP Echo Response
fping -a *host1 host2 host3*	Ping multiple hosts in parallel and report which ones are alive
bing *host1 host2*	Calculate point-to-point throughput between two hosts
traceroute *host*	Print the route, hop by hop, packets trace to a remote host. This is done by sending a sequence of ICMP Echo Request datagrams with increasing TTL values, starting with TTL=1, and expecting ICMP Time Exceeded datagrams
tracepath *host*	Simpler traceroute
mtr *host*	traceroute and ping combined
redir --laddr=*ip1* --lport=*port1* \\ --caddr=*ip2* --cport=*port2*	Redirect all connections coming to local IP address *ip1* and port *port1*, to remote IP address *ip2* and port *port2*
telnet *host port*	Establish a telnet connection to the specified host and port number. If port is omitted, uses default port 23
wget --no-clobber --html-extension \\ --page-requisites --convert-links \\ --recursive --domains example.org \\ --no-parent *www.example.org/path*	Download a whole website *www.example.org/path*
curl *www.example.org/file.html* -o *myfile.html*	Download a file via HTTP and save it locally under another name
curl -u *user*:*password* 'ftp://*ftpserver/path/file*'	Download a file via FTP, after logging in to the server
curl -XPUT *webserver* -d'*data*'	Send an HTTP PUT command with *data* to *webserver*
hping3 *options host*	Send a custom TCP/IP packet to *host* and display the reply

`netstat`	Display network connections

`netstat --tcp`
`netstat -t`
	Display active TCP connections

`netstat -l`	Display only listening sockets
`netstat -a`	Display all listening and non-listening sockets
`netstat -n`	Display network connections, without resolving hostnames or portnames
`netstat -p`	Display network connections, with PID and name of program to which each socket belongs
`netstat -i`	Display network interfaces
`netstat -s`	Display protocol statistics
`netstat -r`	Display kernel routing tables (equivalent to `route -e`)
`netstat -c`	Display network connections continuously
`ss`	Display socket statistics (similarly to `netstat`)
`ss -t -a`	Display all TCP sockets
`nmap` *host* `nmap -sS` *host*	Scan for open TCP ports (TCP SYN scan) on remote host
`nmap -sP` *host*	Do a ping sweep (ICMP ECHO probes) on remote host
`nmap -sU` *host*	Scan for open UDP ports on remote host
`nmap -sV` *host*	Do a service and version scan on open ports
`nmap -p 1-65535` *host*	Scan all ports (1-65535), not only the common ports, on remote host
`nmap -O` *host*	Find which operating system is running on remote host (OS fingerprinting)
`arp-scan`	Scan all hosts on the current LAN. Uses ARP (Layer 2) packets and is therefore able to find even the hosts configured to drop all IP or ICMP traffic; for the same reason it cannot scan hosts outside the same LAN
`ngrep`	Filter data payload of network packets matching a specified regex
`dhcpdump -i eth0`	Sniff all DHCP packets on interface `eth0`
`nload`	Display a graph of the current network usage
`iptraf` `iptraf-ng`	IP LAN monitor (Ncurses UI)
`netserver`	Run a network performance benchmark server
`netperf`	Do network performance benchmarks by connecting to a netserver
`iperf -s`	Run a network throughput benchmark server
`iperf -c` *server*	Perform network throughput tests in client mode, by connecting to an iperf server

tcpdump

Tcpdump is a packet analyzer (aka packet sniffer). A GUI equivalent is Wireshark, previously called Ethereal.

`tcpdump -ni eth0`	Sniff all network traffic on interface `eth0`, suppressing DNS resolution
`tcpdump ip host 10.0.0.2 tcp port 25`	Sniff network packets on TCP port 25 from and to 10.0.0.2
`tcpdump ether host '45:67:89:ab:cd:ef'`	Sniff traffic from and to the network interface having MAC address 45:67:89:ab:cd:ef
`tcpdump 'src host 10.0.0.2 and (tcp port 80 or tcp port 443)'`	Sniff HTTP and HTTPS traffic having as source host 10.0.0.2
`tcpdump -ni eth0 not port 22`	Sniff all traffic on `eth0` except that belonging to the SSH connection
`tcpdump -vvnn -i eth0 arp`	Sniff ARP traffic on `eth0`, on maximum verbosity level, without converting host IP addresses and port numbers to names
`tcpdump ip host 10.0.0.2 and not 10.0.0.9`	Sniff IP traffic between 10.0.0.2 and any other host except 10.0.0.9

netcat

Netcat is "the Swiss Army knife of networking", a very flexible generic TCP/IP client/server.
Depending on the distribution, the binary is called `nc`, `ncat` (Red Hat), or `netcat` (SUSE).

```
nc -z 10.0.0.7 22
ncat 10.0.0.7 22
```

Scan for a listening SSH daemon on remote host 10.0.0.7

```
nc -l -p 25
```

Listen for connections on port 25 (i.e. mimic a SMTP server). Send any input received on stdin to the connected client and dump on stdout any data received from the client

```
nc 10.0.0.7 389 < file
```

Push the content of *file* to port 389 on remote host 10.0.0.7

```
echo "GET / HTTP/1.0\r\n\r\n" | nc 10.0.0.7 80
```

Connect to web server 10.0.0.7 and issue a HTTP GET

```
while true; \
do nc -l -p 80 -q 1 < page.html; done

while true; \
do echo "<html><body>Hello</body></html>" \
| ncat -l -p 80; done
```

Start a minimal web server, serving the specified HTML page to clients

```
nc -v -n -z -w1 -r 10.0.0.7 1-1023
```

Run a TCP port scan against remote host 10.0.0.7.
Probes randomly all privileged ports with a 1-second timeout, without resolving service names, and with verbose output

```
echo "" | nc -v -n -w1 10.0.0.7 1-1023
```

Retrieve the greeting banner of any network service that might be running on remote host 10.0.0.7

`/etc/hosts.allow`
`/etc/hosts.deny`

Host access control files used by the TCP Wrapper system.

Each file contains zero or more *`daemon`*`:`*`client`* lines. The first matching line is considered.

Access is granted when a *`daemon`*`:`*`client`* pair matches an entry in `/etc/hosts.allow`.
Otherwise, access is denied when a *`daemon`*`:`*`client`* pair matches an entry in `/etc/hosts.deny`.
Otherwise, access is granted.

`/etc/hosts.allow` and `/etc/hosts.deny` lines syntax	
`ALL: ALL`	All services to all hosts
`ALL: .example.edu`	All services to all hosts of the example.edu domain
`ALL: .example.edu EXCEPT host1.example.edu`	All services to all hosts of example.edu, except host1
`in.fingerd: .example.com`	Finger service to all hosts of example.com
`in.tftpd: LOCAL`	TFTP to hosts of the local domain only
`sshd: 10.0.0.3 10.0.0.4 10.1.1.0/24`	SSH to the hosts and network specified

`sshd: 10.0.1.0/24`
`sshd: 10.0.1.`
`sshd: 10.0.1.0/255.255.255.0`
	SSH to 10.0.1.0/24

`in.tftpd: ALL: spawn (/safe_dir/safe_finger \`
`-l @%h | /bin/mail -s %d-%h root) &`
	Send a finger probe to hosts attempting TFTP and notify root user via email

`portmap: ALL: (echo Illegal RPC request \`
`from %h | /bin/mail root) &`
	When a client attempts a RPC request via the portmapper (NFS access), echo a message to the terminal and notify the root user via email

Output of command `route -en`

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.3.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         192.168.3.1     0.0.0.0         UG    0      0        0 eth0
```

Destination	*network or host*	destination network or host
	0.0.0.0	default route
Gateway	*host*	gateway
	0.0.0.0 *	no gateway needed, network is directly connected
	-	rejected route
Genmask	*network mask*	network mask to apply for the destination network
	255.255.255.255	destination host
	0.0.0.0	default route
Flags	U	route is up
	G	use gateway
	H	target is host
	!	rejected route
	D	dynamically installed by daemon
	M	modified from routing daemon
	R	reinstate route for dynamic routing

`ip route`
`route -en`
`route -F`
`netstat -rn`
	Display IP routing table

`ip route show cache`
`route -C`
	Display kernel routing cache

`ip route add default via 10.1.1.254`
`route add default gw 10.1.1.254`
	Add a default gateway 10.1.1.254

`ip route add 10.2.0.1 dev eth0`
`ip route add 10.2.0.1 via 10.2.0.254`
`route add -host 10.2.0.1 gw 10.2.0.254`
	Add a route for a host 10.2.0.1

`ip route add 10.2.0.0/16 via 10.2.0.254`
`route add -net 10.2.0.0 netmask 255.255.0.0 gw 10.2.0.254`
	Add a route for a network 10.2.0.0/16

`ip route delete 10.2.0.1 dev eth0`
`route del -host 10.2.0.1 gw 10.2.0.254`
	Delete a route for a host 10.2.0.1

`ip route flush all`	Delete the routing table for all interfaces

The Netfilter framework provides firewalling capabilities in Linux. It is implemented by the user-space application programs `iptables` for IPv4 (which replaced `ipchains`, which itself replaced `ipfwadm`) and `ip6tables` for IPv6.
iptables is implemented in the kernel and therefore does not have a daemon process or a service.
The ability to track connection state is provided by the `ip_conntrack` kernel module.

In RHEL 7, the firewall is managed by the `firewalld` daemon which uses iptables as backend. It is possible, but discouraged, to use iptables directly by disabling firewalld and installing the package `iptables-services`, which provides systemd units for iptables.
In RHEL 8, iptables has been replaced by `nftables`, with firewalld as frontend.
In Ubuntu, the firewall is managed by the `ufw` (Uncomplicated Firewall) service, with iptables as backend.

`/etc/sysconfig/iptables`	Default file containing the firewall rules
`iptables-restore <` *`file`*	Load into iptables the firewall rules specified in the *file*
`iptables-save >` *`file`*	Save into iptables the firewall rules specified in the *file*

iptables rules file	

`*filter`
`:INPUT ACCEPT [0:0]`
`:FORWARD ACCEPT [0:0]`
`:OUTPUT ACCEPT [0:0]`
`COMMIT`
	Delete all rules and open the firewall to all connections

Iptables uses **tables** containing sets of **chains**, which contain sets of **rules**. Each rule has a **target** (e.g. ACCEPT).
The "filter" table contains chains INPUT, FORWARD, OUTPUT (built-in chains); this is the default table to which all iptables commands are applied, unless another table is specified via the `-t` option.
The "nat" table contains chains PREROUTING, OUTPUT, POSTROUTING.
The "mangle" table contains chains PREROUTING, OUTPUT.
When a packet enters the system, it is handed to the INPUT chain. If the destination is local, it is processed; if the destination is not local and IP forwarding is enabled, the packet is handed to the FORWARD chain, otherwise it is dropped.
An outgoing packet generated by the system will go through the OUTPUT chain.
If NAT is in use, an incoming packet will pass at first through the PREROUTING chain, and an outgoing packet will pass last through the POSTROUTING chain.

`iptables -A INPUT -s 10.0.0.6 -j ACCEPT`	Add a rule to accept all packets from 10.0.0.6
`iptables -A INPUT -s 10.0.0.7 -j REJECT`	Add a rule to reject all packets from 10.0.0.7 and send back a ICMP response to the sender
`iptables -A INPUT -s 10.0.0.8 -j DROP`	Add a rule to silently drop all packets from 10.0.0.8
`iptables -A INPUT -s 10.0.0.9 -j LOG`	Add a rule to log (via syslog) all packets from 10.0.0.9

`iptables -D INPUT -s 10.0.0.9 -j LOG`	Delete a specific rule
`iptables -D INPUT 42`	Delete rule 42 of the INPUT chain
`iptables -F INPUT`	Flush all rules of the INPUT chain
`iptables -F`	Flush all rules, hence disabling the firewall
`iptables -t mangle -F`	Flush all rules of the "mangle" table
`iptables -t mangle -X`	Delete all user-defined (not built-in) rules in the "mangle" table

`iptables -L INPUT`	List the rules of the INPUT chain
`iptables -L -n`	List all rules, without translating numeric values (IP addresses to FQDNs and port numbers to services)

`iptables -N mychain`	Define a new chain
`iptables -P INPUT DROP`	Define the chain policy target, which takes effect when no rule matches and the end of the rules list is reached

`iptables -A OUTPUT -d 10.7.7.0/24 -j DROP`	Add a rule to drop all packets with destination 10.7.7.0/24
`iptables -A FORWARD -i eth0 -o eth1 -j LOG`	Add a rule to log all packets entering the system via eth0 and exiting via eth1

`iptables -A INPUT -p 17 -j DROP`
`iptables -A INPUT -p udp -j DROP`
	Add a rule to drop all incoming UDP traffic (protocol numbers are defined in `/etc/protocols`)

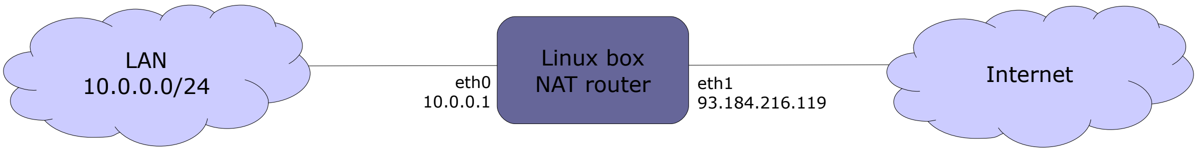
`iptables -A INPUT --sport 1024:65535 --dport 53 \`
`-j ACCEPT`
	Add a rule to accept all packets coming from any unprivileged port and with destination port 53

`iptables -A INPUT -p icmp --icmp-type echo-request \`
`-m limit --limit 1/s -i eth0 -j ACCEPT`
	Add a rule to accept incoming pings through eth0 at a maximum rate of 1 ping/second

`iptables -A INPUT -m state --state ESTABLISHED \`
`-j ACCEPT`
	Load the module for stateful packet filtering, and add a rule to accept all packets that are part of a communication already tracked by the state module

`iptables -A INPUT -m state --state NEW -j ACCEPT`	Add a rule to accept all packets that are not part of a communication already tracked by the state module
`iptables -A INPUT -m state --state RELATED -j ACCEPT`	Add a rule to accept all packets that are related (e.g. ICMP responses to TCP or UDP traffic) to a communication already tracked by the state module
`iptables -A INPUT -m state --state INVALID -j ACCEPT`	Add a rule to accept all packets that do not match any of the states above

iptables NAT routing



SNAT (Source Network Address Translation)

`iptables -t nat -A POSTROUTING -s 10.0.0.0/24 -o eth1 \`
`-j SNAT --to-source 93.184.216.119`
	Map all traffic leaving the LAN to the external IP address 93.184.216.119

`iptables -t nat -A POSTROUTING -s 10.0.0.0/24 -o eth1 \`
`-j SNAT --to-source 93.184.216.119:93.184.216.127`
	Map all traffic leaving the LAN to a pool of external IP addresses 93.184.216.119-127

`iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE`	Map all traffic leaving the LAN to the address dynamically assigned to eth1 via DHCP

DNAT (Destination Network Address Translation)

`iptables -t nat -A PREROUTING -i eth1 -d 93.184.216.119 \`
`-j DNAT --to-destination 10.0.0.13`
	Allow the internal host 10.0.0.13 to be publicly reachable via the external address 93.184.216.119

PAT (Port Address Translation)

`iptables -t nat -A PREROUTING -i eth1 -d 93.184.216.119 \`
`-p tcp --dport 80 -j DNAT --to-destination 10.0.0.13:8080`
	Make publicly accessible a webserver that is located in the LAN, by mapping port 8080 of the internal host 10.0.0.13 to port 80 of the external address 93.184.216.119

`iptables -t nat -A PREROUTING -i eth0 -d ! 10.0.0.0/24 \`
`-p tcp --dport 80 -j REDIRECT --to-ports 3128`
	Redirect all outbound HTTP traffic originating from the LAN to a proxy running on port 3128 on the Linux box

`sysctl -w net.ipv4.ip_forward=1`
`echo 1 > /proc/sys/net/ipv4/ip_forward`
	Enable IP forwarding; necessary to set up a Linux machine as a router. (This command causes other network options to be changed as well.)

In firewalld, a network interface (aka **interface**) or a subnet address (aka **source**) can be assigned to a specific **zone**.
To determine to which zone a packet belongs, first the zone of the source is analyzed, then the zone of the interface; if no source or interface matches, the packet is associated to the default zone (which is "public", unless set otherwise).
If the zone is not specified (via `--zone=`*zone*), the command is applied to the default zone.
By default, commands are temporary; adding the `--permanent` option to a command sets it as permanent, or shows permanent settings only.
Temporary commands are effective immediately but are canceled at reboot, firewall reload, or firewall restart.
Permanent commands are effective only after reboot, firewall reload, or firewall restart.

Firewalld zones (as obtained by `firewall-cmd --get-zones`)	
block	Rejects incoming connections with an ICMP HOST_PROHIBITED; allows only established connections
dmz	Used to expose services to the public; allows only specific incoming connections
drop	Drops all incoming packets; allows only outgoing connections
external	Used for routing and masquerading; allows only specific connections
home	Allows only specific incoming connections
internal	Used to define internal networks and allow only private network traffic
public	Allows only specific incoming connections. Default zone
trusted	Accepts all traffic
work	Used to define internal networks and allow only private network traffic

`systemctl status firewalld`
`firewall-cmd --state`
	Check the status of the firewall

`firewall-config`	Firewall management GUI
`firewall-cmd --reload`	Reload firewall configuration; this applies all permanent changes and cancels all temporary changes. Current connections are not terminated
`firewall-cmd --complete-reload`	Reload firewall configuration, stopping all current connections
`firewall-cmd --runtime-to-permanent`	Transform all temporary changes to permanent

`firewall-cmd --list-all-zones`	List all zones and their full settings
`firewall-cmd --get-default-zone`	Show the default zone
`firewall-cmd --set-default-zone=home`	Set "home" as the default zone
`firewall-cmd --get-active-zones`	Show the active zones i.e. zones bound to either an interface or a source
`firewall-cmd --get-zones`	Show all available zones
`firewall-cmd --get-zone-of-interface=eth0`	Show the zone assigned to `eth0`
`firewall-cmd --new-zone=test`	Create a new zone called "test"
`firewall-cmd --zone=home --change-interface=eth0`	Assign eth0 to the "home" zone
`firewall-cmd --zone=home --list-all`	List temporary settings of the "home" zone
`firewall-cmd --zone=home --list-all --permanent`	List permanent settings of the "home" zone
`firewall-cmd --zone=home --add-source=10.1.1.0/24`	Assign 10.1.1.0/24 to the "home" zone i.e. route all traffic from that subnet to that zone
`firewall-cmd --zone=home --list-sources`	List sources bound to the "home" zone

`firewall-cmd --zone=trusted --add-service=ssh`
`firewall-cmd --zone=trusted --add-port=22/tcp`
	Add the SSH service to the "trusted" zone

`firewall-cmd --zone=trusted --add-service={ssh,http,https}`	Add the SSH, HTTP, and HTTPS services to the "trusted" zone
`firewall-cmd --zone=trusted --list-services`	Show temporary and permanent services bound to the "trusted" zone
`firewall-cmd --zone=trusted --list-ports`	Show temporary and permanent ports open on the "trusted" zone

`firewall-cmd --get-services`	List all predefined services

Predefined services are configured in /usr/lib/firewalld/services/*service*.xml.
User-defined services are configured in /etc/firewalld/services/*service*.xml.

`firewall-cmd --get-icmptypes`	Show all known types of ICMP messages
`firewall-cmd --add-icmp-block=echo-reply`	Block a specific ICMP message type
`firewall-cmd --query-icmp-block=echo-reply`	Tell if a specific ICMP message type is blocked
`firewall-cmd --list-icmp-block`	Show the list of blocked ICMP message types

firewall-cmd --add-rich-rule='*richrule*'	Set up a **rich rule** (for more complex and detailed firewall configurations)

`firewall-cmd --add-rich-rule='rule \`
`family=ipv4 source address=10.2.2.0/24 service name=tftp`
`log prefix=tftp level=info limit value=3/m accept'`
	Set up a rich rule to allow tftp connections from subnet 10.2.2.0/24 and log them via syslog at a rate of 3 per minute

`firewall-cmd --list-rich-rules`	List all rich rules

The manpage `man firewalld.richlanguage` contains several examples of rich rules.

firewall-cmd --direct --add-rule *directrule*	Set up a **direct rule** (in iptables format)

`firewall-cmd --direct --add-rule \`
`ipv4 filter INPUT 0 -p tcp --dport 22 -j ACCEPT`
	Set up a direct rule to allow SSH connections

firewall-offline-cmd *directrule*	Set up a direct rule when firewalld is not running
`firewall-cmd --direct --get-all-rules`	Show all direct rules

The manpage `man firewalld.direct` documents the syntax of direct rules.
User-defined direct rules are stored in /etc/firewalld/direct.xml.

firewall-cmd --zone=*zone* --add-masquerade	Set up masquerading for hosts of *zone*; packets originating from *zone* will get the firewall's IP address on the "external" zone as source address
firewall-cmd --zone=*zone* --add-rich-rule='rule \\ family=ipv4 source address=10.2.2.0/24 masquerade'	Set up masquerading only for those hosts of *zone* located in subnet 10.2.2.0/24
firewall-cmd --zone=*zone* --add-forward-port=\\ port=22:proto=tcp:toport=2222:toaddr=10.7.7.7	Set up port forwarding for hosts of *zone*; incoming connections to port 22 for hosts of *zone* will be forwarded to port 2222 on host 10.7.7.7

Secure Shell (SSH) is a protocol (not a shell) for encrypted secure communications. It is mostly used as a replacement to Telnet to securely login to a remote server's terminal, but can be applied to any network protocol. Some of the most common applications of SSH are Secure Copy (SCP) and SSH File Transfer Protocol (SFTP).

ssh *user@host*	Connect to a remote *host* via SSH and login as *user*. Options: -v -vv -vvv Increasing levels of verbosity -p *n* Use port *n* instead of standard port 22
ssh *user@host command*	Execute a command on a remote host
autossh *user@host*	Connect to a remote host, monitoring the connection and restarting it automatically if it dies
sshpass -p *password* ssh *user@host*	Connect to a remote host using the specified password
pssh -i -H "*host1 host2 host3*" *command*	Execute a command in parallel on a group of remote hosts
ssh-keygen -t rsa -b 2048	Generate interactively a 2048-bit RSA key pair; will prompt for a passphrase
ssh-keygen -t dsa	Generate a DSA key pair
ssh-keygen -p -t rsa	Change passphrase of the private key
ssh-keygen -q -t rsa -f *keyfile* -N '' -C ''	Generate a RSA key with no passphrase (for non-interactive use) and no comment
ssh-keygen -lf *keyfile*	View key length and fingerprint of a public or private key
< *keyfile.pub* awk '{print $2}' \ | base64 -d | openssl *hashfunction*	View fingerprint of a key, calculated using *hashfunction*. RSA keys fingerprint use sha1 (deprecated) or md5
ssh-keyscan *host* >> ~/.ssh/known_hosts	Get the public key of *host* and add it to the user's known hosts file
ssh-agent	Echo to the terminal the environment variables that must be set in order to use the SSH Agent
eval `ssh-agent`	Start the SSH Agent daemon that caches decrypted private keys in memory; also shows the PID of ssh-agent and sets the appropriate environment variables. Once ssh-agent is started, the keys to cache must be added via the ssh-add command; cached keys will then be automatically used by any SSH tool e.g. ssh, sftp, scp
ssh-agent bash -c 'ssh-add *keyfile*'	Start ssh-agent and cache the specified key
ssh-add	Add the default private keys to the ssh-agent cache
ssh-add *keyfile*	Add a specific private key to the ssh-agent cache
ssh-copy-id *user@host*	Use locally available keys to authorize, via public key authentication, login of *user* on a remote *host*. This is done by copying the user's local public key ~/.ssh/id_rsa.pub to ~/.ssh/authorized_keys on the remote host

`scp /path1/file user@host:/path2/`
`scp user@host:/path1/file /path2/`
`scp user1@host1:/path1/file user2@host2:/path2/`
	Non-interactive secure file copy via SSH.
	Can transfer files from local to remote, from remote to local, or between two remote hosts

`sftp user@host`	SSH FTP-like tool for secure file transfer
`scponly`	SSH wrapper pseudo-shell providing access to remote users for secure file transfer, but without execution privileges

SSH port forwarding (aka SSH tunneling)

`ssh -L 2525:mail.foo.com:25 user@mail.foo.com`	Establish a SSH encrypted tunnel from localhost to remote host mail.foo.com, redirecting traffic from local port 2525 to port 25 of remote host mail.foo.com. Useful if the local firewall blocks outgoing port 25. In this case, port 2525 is used to go out; the application must be configured to connect to localhost on port 2525 (instead of mail.foo.com on port 25)
`ssh -L 2525:mail.foo.com:25 user@login.foo.com`	Establish a SSH encrypted tunnel from localhost to remote host login.foo.com. Remote host login.foo.com will then forward, unencrypted, all data received over the tunnel on port 2525 to remote host mail.foo.com on port 25

SSH reverse forwarding (aka SSH reverse tunneling)

`ssh -R 2222:localhost:22 user@login.foo.com`	Establish a SSH encrypted reverse tunnel from remote host login.foo.com back to localhost, redirecting traffic sent to port 2222 of remote host login.foo.com back towards local port 22. Useful if the local firewall blocks incoming connections so remote hosts cannot connect back to local machine. In this case, port 2222 of login.foo.com is opened for listening and connecting back to localhost on port 22; remote host login.foo.com is then able to connect to the local machine on port 2222 (redirected to local port 22)

SSH as a SOCKS proxy

`ssh -D 33333 user@login.foo.com`	The application supporting SOCKS must be configured to connect to localhost on port 33333. Data is tuneled from localhost to login.foo.com, then unencrypted to destination

X11 Forwarding

`ssh -X user@login.foo.com`	Enable the local display to execute locally a X application stored on a remote host login.foo.com

How to enable public key authentication

1. On remote host, set `PubkeyAuthentication yes` in `/etc/ssh/sshd_config`
2. On local machine, do `ssh-copy-id you@remotehost` (or copy your public key to the remote host by hand)

How to enable host-based authentication amongst a group of trusted hosts

1. On all hosts, set `HostbasedAuthentication yes` in `/etc/ssh/sshd_config`
2. On all hosts, create `/etc/ssh/shosts.equiv` and enter in this file all trusted hostnames
3. Connect via SSH manually from your machine on each host so that all hosts' public keys go into `~/.ssh/known_hosts`
4. Copy `~/.ssh/known_hosts` from your machine to `/etc/ssh/ssh_known_hosts` on all hosts

How to enable X11 Forwarding

1. On remote host 10.2.2.2, set `X11Forwarding yes` in `/etc/ssh/sshd_config`, and make sure that `xauth` is installed
2. On local host 10.1.1.1, type `ssh -X 10.2.2.2`, then run on remote host the graphical application e.g. `xclock &`

It is also possible to enable X11 Forwarding via telnet (but this is insecure and obsolete, and therefore not recommended):

1. On remote host 10.2.2.2, type `export DISPLAY=10.1.1.1:0.0`
2. On local host 10.1.1.1, type `xhost +`
3. On local host 10.1.1.1, type `telnet 10.2.2.2`, then run on remote host the graphical application e.g. `xclock &`

`/etc/ssh/sshd_config`	SSH server daemon configuration file
`/etc/ssh/ssh_config`	SSH client global configuration file
`/etc/ssh/ssh_host_key`	Host's private key (should be mode 0600)
`/etc/ssh/ssh_host_key.pub`	Host's public key
`/etc/ssh/shosts.equiv`	Names of trusted hosts for host-based authentication
`/etc/ssh/ssh_known_hosts`	Database of host public keys that were previously accepted as legitimate
`~/.ssh/`	User's SSH directory (must be mode 0700)
`~/.ssh/config`	SSH client user configuration file
`~/.ssh/id_rsa` `~/.ssh/id_dsa`	User's RSA or DSA private key, as generated by `ssh-keygen`
`~/.ssh/id_rsa.pub` `~/.ssh/id_dsa.pub`	User's RSA or DSA public key, as generated by `ssh-keygen`
`~/.ssh/known_hosts`	Host public keys that were previously accepted as legitimate by the user
`~/.ssh/authorized_keys` `~/.ssh/authorized_keys2` (obsolete)	Trusted public keys; the corresponding private keys allow the user to authenticate on this host

`/etc/ssh/sshd_config`	**SSH server configuration file**
`PermitRootLogin yes`	Control superuser login via SSH. Possible values are: `yes` Superuser can login `no` Superuser cannot login `without-password` Superuser cannot login with password `forced-commands-only` Superuser can only run commands in SSH command line

`AllowUsers jdoe ksmith`
`DenyUsers jhacker`
	List of users that can/cannot login via SSH, or `*` for everybody

`AllowGroups geeks`
`DenyGroups *`
	List of groups whose members can/cannot login via SSH, or `*` for all groups

`PasswordAuthentication yes`	Permit authentication via login and password
`PubKeyAuthentication yes`	Permit authentication via public key
`HostbasedAuthentication yes`	Permit authentication based on trusted hosts
`Protocol 1,2`	Specify protocols supported by SSH. Value can be 1 or 2 or both
`X11Forwarding yes`	Allow X11 Forwarding

`/etc/ssh/ssh_config` and `~/.ssh/config`	**SSH client configuration file**
`Host *`	List of hosts to which the following directives will apply, or `*` for all hosts
`StrictHostKeyChecking yes`	Ask before adding new host keys to the `~/.ssh/known_hosts` file, and refuse to connect if the key for a known host has changed. This prevents MITM attacks
`GSSAPIAuthentication yes`	Support authentication using GSSAPI
`ForwardX11Trusted yes`	Allow remote X11 clients to fully access the original X11 display
`IdentityFile ~/.ssh/id_rsa`	User identity file for authentication. Default values are: `~/.ssh/identity` for protocol version 1 `~/.ssh/id_rsa` and `~/.ssh/id_dsa` for protocol version 2

The X.509 standard defines the format of public key certificates and other related files. It includes cryptographic standards and protocols such as SSL/TLS, PKCS7, PKCS12, and OCSP.
The Public Key Infrastructure X.509 (PKIX) is described in RFC 5280.

X.509 file formats	
DER	Binary-encoded certificate

PEM	ASCII-armored Base64-encoded certificate, included between these two lines:
	`-----BEGIN` *`X.509_FILE_TYPE`*`-----`
	`-----END` *`X.509_FILE_TYPE`*`-----`

DER and PEM are also used as file extensions for different types of files; see below.	

X.509 file type extensions	
CRT **CER**	Certificate or certificate chain
CSR	Certificate Signing Request
KEY	Private key
CRL	Certificate Revocation List
DER	Certificate; DER-encoded
PEM	Certificate (including or not the private key), certificate chain, or Certificate Signing Request; PEM-encoded

Other file type extensions	
P12 **PFX**	Certificate (including or not the private key), certificate chain, or Certificate Signing Request; bundled in a PKCS#12 archive file format

openssl x509 -text -in *cert.crt* -noout	Read a certificate
openssl req -text -in *cert.csr* -noout	Read a Certificate Signing Request
openssl req -new -key *cert.key* -out *cert.csr*	Generate a Certificate Signing Request, given a private key
openssl req -new -keyout *cert.key* -out *cert.csr* \ -newkey rsa:2048 -nodes	Generate a Certificate Signing Request, creating also a 2048-bit RSA key pair (unencrypted, for non-interactive use)
openssl x509 -req -in *cert.csr* -CAcreateserial \ -CA *ca.crt* -CAkey *ca.key* -out *cert.crt* -days *validity*	Sign a certificate as a CA, given a Certificate Signing Request
openssl req -x509 -keyout *cert.key* -out *cert.crt* \ -newkey rsa:2048 -nodes -days *validity*	Generate a self-signed root certificate, and create a new CA private key
openssl ca -config *ca.conf* -in *cert.csr* \ -out *cert.crt* -days *validity* -verbose	Sign a certificate
openssl ca -config *ca.conf* -gencrl -revoke *cert.crt* \ -crl_reason *why*	Revoke a certificate
openssl ca -config *ca.conf* -gencrl -out *list.crl*	Generate a Certificate Revocation List containing all revoked certificates so far

openssl x509 -in *cert.pem* -outform DER -out *cert.der*	Convert a certificate from PEM to DER
openssl pkcs12 -export -in *cert.pem* \ -inkey *cert.key* -out *cert.pfx* -name *friendlyname*	Convert a certificate from PEM to PKCS#12 including the private key
openssl pkcs12 -in *cert.p12* -out *cert.crt* -clcerts \ -nokeys	Convert a certificate from PKCS#12 to PEM
openssl pkcs12 -in *cert.p12* -out *cert.key* -nocerts \ -nodes	Extract the private key from a PKCS#12 certificate
openssl pkcs12 -in *cert.p12* -out *ca.crt* -cacerts	Extract the CA certificate from a PKCS#12 certificate
cat *cert.crt* *cert.key* > *cert.pem*	Create a PEM certificate from CRT and private key

openssl dgst -*hashfunction* -out *file.hash* *file*	Generate the digest (hash) of a file
openssl dgst -*hashfunction* *file* | cmp -b *file.hash*	Check the hash of a file; no output means OK
openssl dgst -*hashfunction* -sign *private.key* \ -out *file.sig* *file*	Sign a file
openssl dgst -*hashfunction* -verify *public.key* \ -signature *file.sig* *file*	Verify the signature of a file
openssl enc -e -*cipher* -in *file* -out *file.enc* -salt	Encrypt a file
openssl enc -d -*cipher* -in *file.enc* -out *file*	Decrypt a file

openssl genpkey -algorithm RSA -cipher 3des \ -pkeyopt rsa_keygen_bits:2048 -out *keypair.pem*	Generate a 2048-bit RSA key pair protected by a TripleDES-encrypted passphrase
openssl pkey -text -in *private.key* -noout	Examine a private key
openssl pkey -in *old.key* -out *new.key* -*cipher*	Change the passphrase of a private key
openssl pkey -in *old.key* -out *new.key*	Remove the passphrase from a private key

1. openssl s_client -connect *www.site.com*:443 > *tmpfile* 2. CTRL C 3. openssl x509 -in *tmpfile* -text	Inspect an SSL certificate from a website

openssl list-message-digest-commands	List all available hash functions
openssl list-cipher-commands	List all available ciphers

`CA.pl -newca`	Create a Certification Authority hierarchy
`CA.pl -newreq`	Generate a Certificate Signing Request
`CA.pl -newreq-nodes`	Generate a Certificate Signing Request, creating also a key pair (unencrypted, for non-interactive use)
`CA.pl -signreq`	Sign a Certificate Signing Request
`CA.pl -pkcs12 "`*Certificate name*`"`	Generate a PKCS#12 certificate from a Certificate Signing Request
`CA.pl -newcert`	Generate a self-signed certificate
`CA.pl -verify`	Verify a certificate against the Certification Authority certificate for "demoCA"

GnuPG aka GPG (GNU Privacy Guard) is a well-known implementation of the OpenPGP standard described in RFC 4880. The OpenPGP standard derives from PGP (Pretty Good Privacy), the first tool for strong encryption available to the general public.

`gpg --gen-key`	Generate a key pair
`gpg --import` *alice.asc*	Import Alice's public key *alice.asc* into your keyring
`gpg --list-keys`	List the keys contained into your keyring
`gpg --list-secret-keys`	List your private keys contained into your keyring
`gpg --list-public-keys`	List the public keys contained into your keyring
`gpg --export -o` *keyring.gpg*	Export your whole keyring to a file *keyring.gpg*
`gpg --export-secret-key -a "You" -o` *private.key*	Export your private key to a file *private.key*
`gpg --export-public-key -a "Alice" -o` *alice.pub*	Export Alice's public key to a file *alice.pub*
`gpg --edit-key "Alice"`	Sign Alice's public key
`gpg -e -u "You" -r "Alice"` *file*	Sign *file* (with your private key) and encrypt it to Alice (with Alice's public key)
`gpg -d` *file.gpg* `-o` *file*	Decrypt *file.gpg* (with your own private key) and save the decrypted file to *file*

OpenVPN is an open source software that implements a Virtual Private Network (VPN) between two endpoints. The encrypted VPN tunnel uses UDP port 1194.

`openvpn --genkey --secret` *`keyfile`*

Generate a shared secret keyfile for OpenVPN authentication.
The keyfile must be copied on both server and client

`openvpn server.conf`

Start the VPN on the server side

`openvpn client.conf`

Start the VPN on the client side

`/etc/openvpn/server.conf`

Server-side configuration file:

```
dev tun
ifconfig server_IP client_IP
keepalive 10 60
ping-timer-rem
persist-tun
persist-key
secret keyfile
```

`/etc/openvpn/client.conf`

Client-side configuration file:

```
remote server_public_IP
dev tun
ifconfig client_IP server_IP
keepalive 10 60
ping-timer-rem
persist-tun
persist-key
secret keyfile
```

`md5sum`
`sha1sum`
`sha224sum`
`sha256sum`
`sha384sum`
`sha512sum`
`shasum`
	Print or check the digest of a file generated by a specific hashing algorithm

`stunnel`	TLS encryption wrapper. Can be used to secure any client-server protocol

Key bindings - terminal

Key	Alternate key	Function
CTRL F	→	Move cursor forward one character
CTRL B	←	Move cursor backward one character
CTRL A	HOME	Move cursor to beginning of line
CTRL E	END	Move cursor to end of line
CTRL H	BACKSPACE	Delete character to the left of cursor
CTRL W		Delete word to the left of cursor
CTRL U		Delete all characters to the left of cursor
CTRL K		Delete all characters to the right of cursor
CTRL T		Swap current character with previous one
ESC T		Swap current word with previous one
SHIFT PAGE UP		Scroll up the screen buffer
SHIFT PAGE DOWN		Scroll down the screen buffer
CTRL L		Clear screen (same as `clear`)
CTRL P	↑	Previous command in history
CTRL N	↓	Next command in history
CTRL R		Reverse history search
CTRL I	TAB	Autocomplete commands, filenames, and directory names
ALT /		Autocomplete filenames and directory names only
CTRL ALT E		Expand the Bash alias currently entered on the command line
CTRL J	RETURN	Line feed
CTRL M		Carriage return
CTRL S		Pause transfer to terminal Forward history search (if XON/XOFF flow control is disabled)
CTRL Q		Resume transfer to terminal
CTRL Z		Send a SIGTSTP to put the current job in background
CTRL C		Send a SIGINT to stop the current process
CTRL D		Send a EOF to current process (same as `logout` if process is a shell)
CTRL ALT DEL		Send a SIGINT to reboot the machine (same as `shutdown -r now`), as specified in `/etc/inittab` and `/etc/init/control-alt-delete`
CTRL ALT F1 ... F6		Switch between text consoles (same as `chvt` *n*)

Key	Alternate key	Function
CTRL ALT F7 ... F11		Switch between X Window consoles
CTRL ALT +		Increase X Window screen resolution
CTRL ALT -		Decrease X Window screen resolution
CTRL TAB		Switch between X Window tasks
CTRL ALT →	CTRL ALT ↓	Switch to next workspace
CTRL ALT ←	CTRL ALT ↑	Switch to previous workspace
CTRL ALT BACKSPACE		Reboot the X Window server
		GNOME
ALT TAB		Switch between windows in the current workspace
SUPER		Show activities overview
SUPER L		Lock screen
SUPER M		Show tray messages
SUPER ↑		Maximize current window
SUPER ↓		Restore normal size of current window
SUPER ←		Maximize current window to left half screen
SUPER →		Maximize current window to right half screen
ALT F2		Run command
CTRL +		Increase terminal font size
CTRL -		Decrease terminal font size

The Hardware Abstraction Layer (HAL) manages device files and provides plug-and-play facilities. The HAL daemon hald maintains a persistent database of devices.
udev is the device manager for the Linux kernel. It dynamically generates the device nodes in /dev/ for devices present on the system; it also provides persistent naming for storage devices in /dev/disk.
When a device is added, removed, or changes state, the kernel sends an uevent received by the udevd daemon which will pass the uevent through a set of rules stored in /etc/udev/rules.d/*.rules and /lib/udev/rules.d/*.rules.

udevadm monitor udevmonitor	Show all kernel uevents and udev messages
udevadm info --attribute-walk --name=/dev/sda	Print all attributes of device /dev/sda in udev rules key format
cat /sys/block/sda/size	Print the size attribute of disk sda in 512-byte blocks. This information is retrieved from sysfs
udevadm test /dev/sdb	Simulate an udev event run for the device and print debug output
gnome-device-manager	Browser for the HAL device manager

/etc/udev/rules.d/*.rules and /lib/udev/rules.d/*.rules udev rules	
KERNEL=="hda", NAME="mydisk"	Match a device which was named by the kernel as hda; name the device node as "mydisk". The device node will be therefore /dev/mydisk
KERNEL=="hdb", DRIVER=="ide-disk", SYMLINK+="mydisk myhd"	Match a device with kernel name and driver as specified; name the device node with the default name and create two symbolic links /dev/mydisk and /dev/myhd pointing to /dev/hdb
KERNEL=="fd[0-9]*", NAME="floppy/%n", SYMLINK+="%k"	Match all floppy disk drives (i.e. fd*n*); place device node in /dev/floppy/*n* and create a symlink /dev/fd*n* to it
SUBSYSTEM=="block", ATTR{size}=="41943040", SYMLINK+="mydisk"	Match a block device with a size attribute of 41943040; create a symlink /dev/mydisk
KERNEL=="fd[0-9]*", OWNER="jdoe"	Match all floppy disk drives; give ownership of the device file to user "jdoe"
KERNEL=="sda", PROGRAM="/bin/mydevicenamer %k", SYMLINK+="%c"	Match a device named by the kernel as sda; to name the device, use the defined program which takes on stdin the kernel name and output on stdout e.g. *name1 name2*. Create symlinks /dev/*name1* and /dev/*name2* pointing to /dev/sda
KERNEL=="sda", ACTION=="add", RUN+="/bin/myprogram"	Match a device named by the kernel as sda; run the defined program when the device is connected
KERNEL=="sda", ACTION=="remove", RUN+="/bin/myprogram"	Match a device named by the kernel as sda; run the defined program when the device is disconnected

%n = kernel number (e.g. = 3 for fd3)
%k = kernel name (e.g. = fd3 for fd3)
%c = device name as output from program

A kernel version number has the form *major.minor.patchlevel*.
Kernel images are usually gzip-compressed and can be of two types: zImage (max 520 Kb) and bzImage (no size limit).
Kernel modules can be loaded dynamically into the kernel to provide additional functionalities on demand, instead of being included when the kernel is compiled; this reduces memory footprint.
`kerneld` (daemon) and `kmod` (kernel thread) facilitate the dynamic loading of kernel modules.

`/lib/modules/X.Y.Z/*.ko`	Kernel modules for kernel version *X.Y.Z*
`/lib/modules/X.Y.Z/modules.dep`	Modules dependencies. This file needs to be recreated (via the command `depmod -a`) after a reboot or a change in module dependencies
`/etc/modules.conf` `/etc/conf.modules` (deprecated)	Modules configuration file
`/usr/src/linux/`	Directory containing the kernel source code to be compiled
`/usr/src/linux/.config`	Kernel configuration file

`freeramdisk`	Free the memory used for the `initrd` image. This command must be run directly after unmounting `/initrd`
`mkinitrd initrd_image kernel_version` (Red Hat)	Create an `initrd` image file
`mkinitramfs` (Debian)	Create an `initrd` image file according to the configuration file `/etc/initramfs-tools/initramfs.conf`
`dracut`	Create initial ramdisk images for preloading modules
`dbus-monitor`	Monitor messages going through a D-Bus message bus
`dbus-monitor --session`	Monitor session messages (default)
`dbus-monitor --system`	Monitor system messages

`kexec -l kernel_image --append=options \`
`--initrd=initrd_image && kexec -e`
	Load a kernel image file into memory and boot it. This allows running a different kernel without rebooting the machine

The runtime loader `ld.so` loads the required shared libraries of the program into RAM, searching in this order:

1.	`LD_LIBRARY_PATH`	Environment variable specifying the list of dirs where libraries should be searched for first
2.	`/etc/ld.so.cache`	Cache file
3.	`/lib` and `/usr/lib`	Default locations for shared libraries

Shared library locations (other than the default ones `/lib` and `/usr/lib`) can be specified in the file `/etc/ld.so.conf`.

`ldconfig`	Create a cache file `/etc/ld.so.cache` of all available dynamically linked libraries. This command should be run when the system complains about missing libraries
`ldd program_or_lib`	Print library dependencies

`lspci`	List PCI devices
`lspci -d 8086:`	List all Intel hardware present. PCI IDs are stored in: `/usr/share/misc/pci.ids` (Debian) `/usr/share/hwdata/pci.ids` (Red Hat)
`lsusb`	List USB devices
`lsusb -d 8086:`	List all Intel USB devices present. USB IDs are stored in: `/var/lib/usbutils/usb.ids` (Debian) `/usr/share/hwdata/usb.ids` (Red Hat)
`lsdev`	List information about the system hardware
`lshw`	List system hardware
`lscpu`	List information about the CPU architecture
`uname`	Print system information. Values that can be printed are: `-s` Kernel name `-n` Network node hostname `-r` Kernel release number *X.Y.Z* `-v` Kernel version number `-m` Machine hardware name `-p` Processor type `-i` Hardware platform `-o` Operating system `-a` All the above information, in that order
`evtest`	Monitor and query input device events in `/dev/input/event`*n*
`dmesg`	Print the messages of the kernel ring buffer. Options are: `-T` Print human-readable timestamps `-n 1` Set the logging level to 1 (= only panic messages)
`journalctl`	Display the Systemd journal, which contains the kernel logs
`journalctl -n` *n*	Display the most recent *n* log lines (default is 10)
`journalctl --since "1 hour ago"`	Display events happened in the last hour
`journalctl -x`	Display events, adding explanations from the message catalog
`journalctl -f`	Display the journal in real-time

`journalctl -u crond.service`
`journalctl _SYSTEMD_UNIT=crond.service`
	Display the log entries created by the cron service

`mkdir -p /var/log/journal/ && \`
`systemctl restart systemd-journald`
	Enable persistent storage of logs in `/var/log/journal/`
	(by default, journalctl stores the logfiles in RAM only)

Kernel compile and patching

Kernel compile		
Download	Download the kernel source code `linux-X.Y.Z.tar.bz2` from `http://www.kernel.org` to the base of the kernel source tree `/usr/src/linux`	
Clean	`make clean`	Delete most generated files
	`make mrproper`	Delete all generated files and kernel configuration
	`make distclean`	Delete temporary files, patch leftovers, and similar files
Configure	`make config`	Terminal-based (options must be set in sequence)
	`make menuconfig`	Ncurses UI

	`make xconfig`
	`make gconfig`
		GUI

	`make oldconfig`	Create a new configuration file, based on the options in the old configuration file and in the source code
	Components (e.g. device drivers) can be either: - not compiled - compiled into the kernel binary, for support of devices always used on the system or necessary for the system to boot - compiled as a kernel module, for optional devices The configuration command creates a configuration file `/usr/src/linux/.config` containing instructions for the kernel compilation	
Build	`make bzImage`	Compile the kernel
	`make modules`	Compile the kernel modules
	`make all`	Compile kernel and kernel modules
	`make -j2 all` will speed up compilation by allocating 2 simultaneous compile jobs	
Modules install	`make modules_install`	Install the previously built modules present in `/lib/modules/X.Y.Z`
Kernel install	`make install`	Install the kernel automatically
	To install the kernel by hand: 1. Copy the new compiled kernel and other files into the boot partition: `cp /usr/src/linux/arch/boot/bzImage /boot/vmlinuz-X.Y.Z` (kernel) `cp /usr/src/linux/arch/boot/System.map-X.Y.Z /boot` `cp /usr/src/linux/arch/boot/config-X.Y.Z /boot` (config options used for this compile) 2. Create an entry in GRUB to boot on the new kernel	
Package	Optionally, the kernel can be packaged for install on other machines	
	`make rpm-pkg`	Build source and binary RPM packages
	`make binrpm-pkg`	Build binary RPM package
	`make deb-pkg`	Builds binary DEB package

Kernel patching		
Download	Download and decompress the patch to `/usr/src`	
Patch	`patch -p1 < file.patch`	Apply the patch
	`patch -Rp1 < file.patch`	Remove (reverse) a patch. Alternatively, applying the patch again reverses it
Build	Build the patched kernel as explained above	
Install	Install the patched kernel as explained above	

Kernel modules

Kernel modules allow the kernel to access functions (symbols) for kernel services e.g. hardware drivers, network stack, or filesystem abstraction.

`lsmod`	List the modules that are currently loaded into the kernel
`insmod` *`module`*	Insert a module into the kernel. If the module requires another module or if it does not detect compatible hardware, insertion will fail
`rmmod` *`module`*	Remove a module from the kernel. If the module is in use by another module, it is necessary to remove the latter first
`modinfo` *`module`*	Display the list of parameters accepted by the module
`depmod -a`	Probe all modules in the kernel modules directory and generate the file that lists their dependencies

It is recommended to use `modprobe` instead of `insmod` and `rmmod`, because it automatically handles prerequisites when inserting modules, is more specific about errors, and accepts just the module name instead of requiring the full pathname.

`modprobe` *`module option=value`*	Insert a module into the running kernel, with the specified parameters. Prerequisite modules will be inserted automatically
`modprobe -a`	Insert all modules
`modprobe -t` *`directory`*	Attempt to load all modules contained in the directory until a module succeeds. This action probes the hardware by successive module-insertion attempts for a single type of hardware, e.g. a network adapter
`modprobe -r` *`module`*	Remove a module
`modprobe -c` *`module`*	Display module configuration
`modprobe -l`	List loaded modules

<table>
<tr><th colspan="3">Configuration of device drivers</th></tr>
<tr><td colspan="3">Device drivers support the kernel with instructions on how to use that device.</td></tr>
<tr><td><b>Device driver compiled into the kernel</b></td><td colspan="2">Configure the device driver by passing a kernel parameter in the GRUB menu:
<code>kernel /vmlinuz ro root=/dev/vg0/root vga=0x33c</code></td></tr>
<tr><td rowspan="3"><b>Device driver provided as a kernel module</b></td><td colspan="2">Edit module configuration in <code>/etc/modprobe.conf</code> or <code>/etc/modprobe.d/</code> (Red Hat):</td></tr>
<tr><td><code>alias eth0 3c59x</code></td><td>Specify that eth0 uses the <code>3c59x.ko</code> driver module</td></tr>
<tr><td><code>options 3c509 irq=10,11</code></td><td>Assign IRQ 10 and 11 to 3c509 devices</td></tr>
</table>

/proc is a pseudo filesystem that gives access to process data held in the kernel.

File	Information stored (can be viewed via cat)	Equivalent command
/proc/bus	Buses (e.g. PCI, USB, PC Card)	
/proc/cpuinfo	CPUs information	
/proc/devices	Drivers currently loaded	
/proc/dma	DMA channels in use	
/proc/filesystems	Filesystems supported by the system	
/proc/interrupts	Current IRQs (Interrupt Requests)	procinfo
/proc/ioports	I/O addresses in use	
/proc/kcore	Memory allocatable by the kernel	
/proc/loadavg	System load averages	uptime
/proc/mdstat	Information about RAID arrays and devices	
/proc/meminfo	Total and free memory	free
/proc/modules	Kernel modules currently loaded	lsmod
/proc/mounts	Mounted partitions	mount
/proc/net/dev	Network interface statistics	
/proc/partitions	Drive partition information	fdisk -l
/proc/swaps	Size of total and used swap areas	swapon -s
/proc/sys/ /proc/sys/kernel/ /proc/sys/net/	sysfs: exposes tunable kernel parameters Kernel information and parameters Network information and parameters	
/proc/uptime	Time elapsed since boot	uptime
/proc/version	Linux version	uname -a
/proc/*n*/ /proc/*n*/cmdline /proc/*n*/cwd /proc/*n*/environ /proc/*n*/exe /proc/*n*/fd /proc/*n*/root /proc/*n*/status	Information about process with PID *n* Command by which the process was launched Symlink to process' working directory Values of environment variables of process Symlink to process' executable Files currently opened by the process Symlink to process' filesystem root Status of process	ps *n* lsof -p *n*

/proc/sys is the only writable branch of /proc and can be used to tune kernel parameters on-the-fly.
All changes are lost after system shutdown, unless applied via sysctl -p.

sysctl fs.file-max cat /proc/sys/fs/file-max	Get the maximum allowed number of open files
sysctl -w "fs.file-max=100000" echo "100000" > /proc/sys/fs/file-max	Set the maximum allowed number of open files to 100000
sysctl -a	List all available kernel tuning options
sysctl -p	Apply all tuning settings listed in /etc/sysctl.conf. This command is usually run at boot by the system initialization script, to make permanent changes to kernel parameters

/dev contains the device files to access all devices in the system.

File	Device
/dev/sda	SCSI, PATA, or SATA hard drive
/dev/hda	IDE hard drive
/dev/pda	Parallel port IDE hard drive
/dev/vda	Virtual disk for KVM-based virtual machines
/dev/sda, /dev/sdb, /dev/sdc ...	First, second, third ... hard drive
/dev/sda1, /dev/sda2, /dev/sda3 ...	First, second, third ... partition of the first hard drive
/dev/md0	Metadisk group, for use with RAID
/dev/sr0	SCSI CD-ROM
/dev/pcd0	Parallel port CD-ROM
/dev/cdrom	CD-ROM. Usually symlinked to /dev/sr0
/dev/fd0	Floppy disk drive
/dev/ht0	IDE tape drive
/dev/pt0	Parallel port tape drive
/dev/sg0	Generic SCSI device
/dev/loop0	Loopback device
/dev/autofs	AutoFS device
/dev/fuse	FUSE device
/dev/dsp	Digital Signal Processor device. Interfaces with the soundcard
/dev/fb0	Framebuffer device. Interfaces with the graphics hardware
/dev/lp0	Parallel port printer device
/dev/parport0	Raw parallel port device
/dev/mem	Physical memory
/dev/kmem	Kernel virtual memory
/dev/core	Obsolete. Symlink to /proc/kcore
/dev/stdin	Standard Input
/dev/stdout	Standard Output
/dev/stderr	Standard Error
/dev/null	Null device, aka blackhole or bit bucket. Discards any received data
/dev/zero	Zero device. Outputs an infinite stream of zero bytes (NUL) on reads
/dev/full	"Always full" device. Similar to /dev/zero, and also returns an error "No space left on device" (ENOSPC) on writes
/dev/random	Non-deterministic random number generator. Gathers entropy from the system to generate randomness; once the entropy pool is depleted, the device blocks all reads until it can collect more entropy
/dev/urandom	Pseudo random number generator. Faster but unsafe for cryptographic purposes
/dev/console	System console
/dev/tty	Terminal for current process
/dev/tty0	Current virtual console
/dev/ttyS0	Serial port, usually used for modem connections
/dev/ptyp0	Pseudo-TTY master
/dev/ttyp0	Pseudo-TTY slave

If the kernel has been booted in emergency mode and `init` has not been run, some initial configuration is necessary e.g.

```
mount /proc
mount -o remount,rw /
mount -a
```

If mounting the filesystems fails:

```
mknod /dev/sda
mknod /dev/sda1
fdisk -l /dev/sda
fsck -y /dev/sda1
mount -t ext3 /dev/sda1 /mnt/sysimage
chroot /mnt/sysimage
```

To install a package using an alternative root directory (useful if the system has been booted from a removable media):

```
rpm -U --root /mnt/sysimage package.rpm
```

To install GRUB on the specified directory (which must contain `/boot/grub/`):

```
grub-install --root-directory=/mnt/sysimage /dev/sda
```

Alternative method:

```
chroot /mnt/sysimage
grub-install /dev/sda
```

Run `sync` and unmount all filesystems before exiting the shell, to ensure that all changes have been written on disk.

How to reset the root password (RHEL 7 and 8)

1. Power up the system and, once on the GRUB 2 boot screen, press E to edit the current entry
2. On the kernel line that mentions `linux16`, remove the `rhgb` and `quiet` parameters and add `rd.break` at the end
3. Press CTRL X ; the system will boot on the initramfs `switch_root` prompt
4. Remount the filesystem as writable `mount -o remount,rw /sysroot`
5. Change the filesystem root `chroot /sysroot`
6. Modify the root password `passwd root`
7. Force SELinux to relabel context on next boot `touch /.autorelabel`
8. Remount the filesystem as readonly (not strictly necessary) `mount -o remount,ro /sysroot`
9. Exit the chroot environment `exit`
10. Resume system boot `exit`

DNS implementations	
BIND	Berkeley Internet Name Domain system, is the standard DNS server for UNIX
Unbound	Standard DNS server in RHEL 7
dnsmasq	Lightweight DNS, DHCP and TFTP server for a small network
djbdns	Security-hardened DNS server that also includes DNS debugging tools
PowerDNS	Alternative open-source DNS server

`named`	BIND Name Daemon
`ndc`	Name Daemon Controller for BIND 8
`rndc`	Remote Name Daemon Controller for BIND 9, uses a shared key to communicate securely with `named`

`dnswalk` *`example.org.`*	DNS debugger

`rndc reconfig`	Reload BIND configuration and new zones
`rndc reload` *`example.org`*	Reload the zone *example.org*
`rndc freeze` *`example.org`*	Suspend updates for the zone *example.org*
`rndc thaw` *`example.org`*	Resume updates for the zone *example.org*
`rndc tsig-list`	List all currently active TSIG keys

DNSSEC was designed to secure the DNS tree and hence prevent cache poisoning.
The TSIG (Transaction SIGnature) standard, that authenticates communications between two trusted systems, is used to sign zone transfers and DDNS (Dynamic DNS) updates.

```
dnssec-keygen -a dsa -b 1024 \
-n HOST dns1.example.org
```

Generate a TSIG key with DNSSEC algorithm *nnn* and key fingerprint *fffff*.
This will create two key files
`Kdns1.example.org.+`*`nnn`*`+`*`fffff`*`.key`
`Kdns1.example.org.+`*`nnn`*`+`*`fffff`*`.private`
which contain a key number that must be inserted both in `/etc/named.conf` and `/etc/rndc.conf`

```
rndc-confgen -a
```

Generate a `/etc/rndc.key` key file:

```
key "rndc-key" {
   algorithm hmac-md5;
   secret "vyZqL3tPHsqnA57e4LT0Ek==";
};
options {
   default-key "rndc-key";
   default-server 127.0.0.1;
   default-port 953;
};
```

This file is automatically read both by `named` and `rndc`

`dnssec-signzone` *`example.org`*	Sign the zone *example.org*

`named -u named -g named`	Run BIND as user/group "named" (must be created if needed) instead of root
`named -t /var/cache/bind`	Run BIND in a chroot jail `/var/cache/bind` (actually it is the `chroot` command that starts the `named` server)

/etc/named.conf **DNS server configuration file**

```
controls {
   inet 127.0.0.1 allow {localhost;} keys {rndckey;};
};
key "rndc-key" {                                // TSIG key
   algorithm dsa;
   secret "HYZur46fftdUQ43BJKI093t4t78lkp";
};

acl "mynetwork" {10.7.0.0/24;};                // Alias definition
                                               // Built-in ACLs: any, none, localhost, localnets

options {
   directory "/var/named";                     // Working directory
   version "0.0";                              // Hide version number by replacing it with 0.0
   listen-on port 53 {10.7.0.1; 127.0.0.1;};   // Port and own IP addresses to listen on
   blackhole {172.17.17.0/24;};                // IPs whose packets are to be ignored
   allow-query {mynetwork;};                   // IPs allowed to do iterative queries
   allow-query-on {any;};                      // Local IPs that can accept iterative queries
   allow-query-cache {any;};                   // IPs that can get an answer from cache
   allow-recursion {mynetwork;};      // IPs to accept recursive queries from (typically
                                      // own network's IPs).  The DNS server does the full
                                      // resolution process on behalf of these client IPs,
                                      // and returns a referral for the other IPs
   allow-recursion-on {mynetwork;};   // Local IPs that can accept recursive queries
   allow-transfer {10.7.0.254;};      // Zone transfer is restricted to these IPs (slaves);
                                      // on slave servers, this option should be disabled
   allow-update {any;};               // IPs to accept DDNS updates from
   recursive-clients 1000;            // Max number of simultaneous recursive lookups
   dnssec-enable yes;                 // Enable DNSSEC
   dialup no;                         // Not a dialup connection: external zone maintenance
                                      // (e.g. sending heartbeat packets, external zone transfers)
                                      // is then permitted
   forward first;                              // Site-wide cache: bypass the normal resolution
   forwarders {10.7.0.252; 10.7.0.253;};       // method by querying first these central DNS
                                               // servers if they are available
};

// Define the root name servers
zone "." {
   type hint;
   file "root.cache";
}

// Configure system to act as a master server for the example.org domain
zone "example.org" IN {
   type master;
   file "master/example.org.zone";     // Zone file for the example.org domain
};
zone "240.123.224.in-addr.arpa" IN {   // Configure reverse lookup zone (for 224.123.240.0/24)
   type master;
   file "slave/example.org.revzone";
};

// Configure system to act as a slave server for the example2.org domain
zone "example2.org" IN {
   type slave;
   file "slave/example2.org.zone";     // Slave: do not edit this zone file!
   masters {10.7.0.254;};
};
zone "0.7.10.in-addr.arpa" IN {        // Configure reverse lookup zone (for 10.7.0.0/24)
   type slave;
   file "slave/10.7.0.revzone";
   masters {10.7.0.254;};
};
```

DNS zone file

/var/named/master/example.org.zone **DNS zone file for the example.org zone**

```
$TTL 86400      ; TTL (1 day)
$ORIGIN example.org.
example.org IN SOA dns1.example.org. help.example.org. (    ; Master DNS server is dns1.example.org
   2014052300   ; serial                                    ; If problems, contact help@example.org
   28800        ; refresh (8 hours)
   7200         ; retry (2 hours)
   604800       ; expire (1 week)
   600 )        ; negative TTL (10 mins)

        IN NS     dns1.example.org.
        IN NS     dns2.example.org.
        IN MX     10 mail1.example.org.
        IN MX     20 mail2.example.org.

dns1    IN A      224.123.240.3
dns2    IN A      224.123.240.4
mail1   IN A      224.123.240.73
mail2   IN A      224.123.240.77
foo     IN A      224.123.240.12
bar     IN A      224.123.240.13
www     IN A      224.123.240.19
baz     IN CNAME  bar

subdomain   IN NS   ns1.subdomain.example.org.   ; Glue records
            IN NS   ns2.subdomain.example.org.
ns1.subdomain.example.org.   IN A   224.123.240.201
ns2.subdomain.example.org.   IN A   224.123.240.202
```

/var/named/master/example.org.revzone **DNS reverse zone file for the example.org zone**

```
$TTL 86400      ; TTL (1 day)
example.org IN SOA dns1.example.org. help.example.org. (
   2014052300   ; serial
   28800        ; refresh (8 hours)
   7200         ; retry (2 hours)
   604800       ; expire (1 week)
   600 )        ; negative TTL (10 mins)

12.240.123.224.in-addr.arpa   IN PTR   foo
13.240.123.224.in-addr.arpa   IN PTR   bar
19.240.123.224.in-addr.arpa   IN PTR   www
```

Resource Records

	$TTL	How long to cache a positive response
	$ORIGIN	Suffix appended to all names not ending with a dot. Useful when defining multiple subdomains inside the same zone
SOA	Start Of Authority for the example.org zone	
	serial	Serial number. Must be increased after each edit of the zone file
	refresh	How frequently a slave server refreshes its copy of zone data from the master
	retry	How frequently a slave server retries connecting to the master
	expire	How long a slave server relies on its copy of zone data. After this time period expires, the slave server is not authoritative anymore for the zone unless it can contact a master
	negative TTL	How long to cache a non-existent answer
A	Address: maps names to IP addresses. Used for DNS lookups.	
PTR	Pointer: maps IP addresses to names. Used for reverse DNS lookups. Each A record must have a matching PTR record	
CNAME	Canonical Name: specifies an alias for a host with an A record (even in a different zone). Discouraged as it causes multiple lookups; it is better to use multiple A records instead	
NS	Name Service: specifies the authoritative name servers for the zone	
MX	Mailserver: specifies address and priority of the servers able to handle mail for the zone	
Glue Records are not really part of the zone; they delegate authority for other zones, usually subdomains		

Most common HTTP response codes		
1XX **Informational**	**100 Continue**	The server received the request headers, so the client should continue by sending the remainder of the request
	101 Switching Protocols	The server agreed to switch protocol upon client's demand
2XX **Success**	**200 OK**	The request was successful
	201 Created	The request was successful, and resulted in a resource being created
	204 No Content	The request was successful, and the server does not need to return any content
	206 Partial Content	The request was successful, and the server is returning only partial content because the client sent a Range header field
3XX **Redirection**	**301 Moved Permanently**	The requested resource was permanently moved to a new URI
	302 Found	The requested resource was temporarily moved to a new URI
	303 See Other	The requested resource can be found on another URI, and should be retrieved from there via a GET
	304 Not Modified	The client sent a conditional GET request, and the resource has not been modified since last time it was requested
	307 Temporary Redirect	The requested resource was temporarily moved to a new URI, but future requests should use the original URI
4XX **Client Error**	**400 Bad Request**	The server was unable to understand the request due to bad syntax
	401 Unauthorized	The request requires user authentication
	403 Forbidden	The client did not have the necessary permissions to access the requested resource
	404 Not Found	The requested resource was not found on the server
	408 Request Timeout	The server timed out while waiting for the request
	409 Conflict	The request could not be processed because of a conflict in the resource state
	410 Gone	The requested resource is no longer available on the server and will not be available again
	451 Unavailable for Legal Reasons	The requested resource is not available due to government censorship
5XX **Server Error**	**500 Internal Server Error**	The server encountered a generic error while trying to fulfill the request
	501 Not Implemented	The server was unable to recognize the request method
	502 Bad Gateway	The server is acting as a gateway or proxy, and received an invalid response from the upstream server
	503 Service Unavailable	The server is temporarily unavailable due to overload or maintenance
	504 Gateway Timeout	The server is acting as a gateway or proxy, and a request to the upstream server timed out
	505 HTTP Version Not Supported	The server does not support the HTTP protocol version used in the request

Apache is an open source and widespread HTTP server, originally based on the NCSA HTTPd server.

`apachectl`	(Red Hat)	Manage the Apache webserver
`httpd`	(Red Hat)	
`apache2ctl`	(Debian)	

`apachectl start`	Start the Apache webserver daemon
`apachectl status`	Display a brief status report
`apachectl fullstatus`	Display a detailed status report
`apachectl graceful`	Gracefully restart Apache; currently open connections are not aborted
`apachectl graceful-stop`	Gracefully stop Apache; currently open connections are not aborted

`apachectl configtest`
`apachectl -t`
	Test the configuration file, reporting any syntax error

`apachectl -M`	List all loaded and shared modules

`/var/www/html`	Default document root directory
`$HOME/public_html`	Default document root directory for users' websites

Web content must be readable by the user/group the Apache process runs as. For security reasons, it should be owned and writable by the superuser or the webmaster user/group (usually `www-data`), not the Apache user/group.

`/etc/httpd/conf/httpd.conf` `/etc/httpd/conf.d/*.conf`	(Red Hat)	Apache configuration files
`/etc/apache2/httpd.conf`	(Debian and SUSE)	

The Apache webserver contains a number of MPMs (Multi-Processing Modules) which can operate following two methods:

prefork MPM — A number of child processes is spawned in advance, with each child serving one connection.
Highly reliable due to Linux memory protection that isolates each child process.

worker MPM — Multiple child processes spawn multiple threads, with each thread serving one connection.
More scalable but prone to deadlocks if third-party non-threadsafe modules are loaded.

HTTPS

HTTPS (i.e. HTTP over SSL/TLS) allows securing communications between the webserver and the client by encrypting all communications end-to-end between the two. A webserver using HTTPS hands over its public key to the client when the client connects to the server via port 443. The server's public key is signed by a CA (Certification Authority), whose validity is ensured by the root certificates stored into the client's browser.

The `openssl` command and its user-friendly `CA.pl` script are the tools of the OpenSSL crypto library that can be used to accomplish all public key crypto operations e.g. generate key pairs, Certificate Signing Requests, and self-signed certificates. Another user-friendly tool is `genkey`.

Virtual hosting with HTTPS requires assigning a unique IP address for each virtual host; this because the SSL handshake (during which the server sends its certificate to the client's browser) takes place before the client sends the `Host:` header (which tells to which virtual host the client wants to talk).
A workaround for this is SNI (Server Name Indication) which makes the browser send the hostname in the first message of the SSL handshake. Another workaround is to have all multiple name-based virtual hosts use the same SSL certificate with a wildcard domain e.g. *.example.org.

Apache configuration file	
Server configuration directives	
`ServerName www.mysite.org:80`	Name and port (if omitted, uses default HTTP port 80) of server
`ServerRoot /etc/httpd`	Root directory for configuration and log files
`ServerAdmin webmaster@mysite.org`	Contact address that the server includes in any HTTP error messages to the client. Can be an email address or a URL
`StartServers 5`	Number of servers to start initially

`MinSpareServers 5`
`MaxSpareServers 10`
	Minimum and maximum number of idle child server processes

`MaxClients 256` (before v2.3.13) `MaxRequestWorkers 256` (v2.3.13 and later)	Max number of simultaneous requests that will be served; clients above this limit will get a HTTP error 503 - Service Unavailable. Prefork MPM: max number of child processes launched to serve requests. Worker MPM: max total number of threads available to serve requests
`ServerLimit 256`	Prefork MPM: max configured value for `MaxRequestWorkers`. Worker MPM: in conjunction with `ThreadLimit`, max configured value for `MaxRequestWorkers`
`ThreadsPerChild 25`	Worker MPM: number of threads created by each child process
`ThreadLimit 64`	Worker MPM: max configured value for `ThreadsPerChild`
`MaxRequestsPerChild 16` (v2.2) `MaxConnectionsPerChild 16` (v2.4)	Max number of connections allowed per child
`LoadModule mime_module modules/mod_mime.so`	Load the module `mime_module` by linking in the object file or library `modules/mod_mime.so`

`Listen 10.17.1.1:80`
`Listen 10.17.1.5:8080`
	Make the server accept connections on the specified IP addresses (optional) and ports

`User nobody`
`Group nobody`
	User and group the Apache process runs as. For security reasons, this should not be root

Apache configuration file	
Main configuration directives	
`DocumentRoot /var/www/html`	Directory in filesystem that maps to the root of the website
`Alias /image /mydir/pub/image`	Map the URL `http://www.mysite.org/image/` to the directory `/mydir/pub/image` in the filesystem. This allows Apache to serve content placed outside of the document root
`TypesConfig conf/mime.types`	Media types file. The path is relative to `ServerRoot`
`AddType image/jpeg jpeg jpg jpe`	Map the specified filename extensions onto the specified content type. These entries add to or override the entries from the media types file `conf/mime.types`
`Redirect permanent /foo /bar`	Redirect to a URL on the same host. Status can be: `permanent` return an HTTP status 301 - Moved Permanently `temp` return an HTTP status 302 - Found (default) `seeother` return an HTTP status 303 - See Other `gone` return an HTTP status 410 - Gone
`Redirect /foo http://www.example.com/foo`	Redirect to a URL on a different host
`AccessFileName .htaccess`	Name of the distributed configuration file, which contains directives that apply to the document directory it is in and to all its subtrees

`<Directory "/var/www/html/foobar">`
`   AllowOverride AuthConfig Limit`
`</Directory>`
	Specify which global directives an `.htaccess` file can override:
	`AuthConfig` Authorization directives for directory protection
	`FileInfo` Document type and metadata
	`Indexes` Directory indexing
	`Limit` Host access control
	`Options` Specific directory features
	`All` All directives
	`None` No directive

Limited scope directives	

`<Directory "/var/www/html/foobar">`
`   `*`[list of directives]`*
`</Directory>`
	Limit the scope of the specified directives to the directory `/var/www/html/foobar` and its subdirectories

`<Location /foobar>`
`   `*`[list of directives]`*
`</Location>`
	Limit the scope of the specified directive to the URL `http://www.mysite.org/foobar/` and its subdirectories

Logging directives	
`LogFormat "%h %l %u %t \"%r\" %>s %b"`	Specify the format of a log
`LogFormat "%h %l %u %t \"%r\" %>s %b" common`	Specify a nickname for a log format. In this case, specifies "common" for the CLF (Common Log Format) which is defined as such: `%h` IP address of the client host `%l` Identity of client as determined by `identd` `%u` User ID of client making the request `%t` Timestamp the server completed the request `%r` Request as done by the user `%s` Status code sent by the server to the client `%b` Size of the object returned, in bytes
`CustomLog /var/log/httpd/access_log common`	Set up a log filename, with the format or (as in this case) the nickname specified
`TransferLog /var/log/httpd/access_log`	Set up a log filename, with format determined by the most recent `LogFormat` directive which did not define a nickname
`TransferLog "|rotatelogs access_log 86400"`	Set log rotation every 24 hours
`HostnameLookups Off`	Disable DNS hostname lookup to save network traffic. Hostnames can be resolved later by processing the log file: `logresolve <access_log >accessdns_log`

Apache configuration file	
Virtual hosts directives	
`NameVirtualHost *` (v2.2)	Specify which IP address will serve virtual hosting. The argument can be an IP address, an *address:port* pair, or `*` for all IP addresses of the server. The same argument need to be inserted in the relevant `<VirtualHost>` directive

`<VirtualHost *:80>`
`   ServerName www.mysite.org`
`   ServerAlias mysite.org *.mysite.org`
`   DocumentRoot /var/www/vhosts/mysite`
`</VirtualHost>`
	The first listed virtual host is also the default virtual host.
	It inherits those main settings that does not override.
	This virtual host answers to `http://www.mysite.org`, and also redirects there all HTTP requests on the domain mysite.org

`<VirtualHost *:80>`
`   ServerAdmin webmaster@www.mysite2.org`
`   ServerName www.mysite2.org`
`   DocumentRoot /var/www/vhosts/mysite2`
`   ErrorLog /var/www/logs/mysite2`
`</VirtualHost>`
	Name-based virtual host `http://www.mysite2.org`.
	Multiple name-based virtual hosts can share the same IP address; DNS must be configured accordingly to map each name to the correct IP address. Cannot be used with HTTPS

`<VirtualHost *:8080>`
`   ServerName www.mysite3.org`
`   DocumentRoot /var/www/vhosts/mysite3`
`</VirtualHost>`
	Port-based virtual host answering to connections on port 8080.
	A `Listen 8080` directive must also be present

`<VirtualHost 10.17.1.5:80>`
`   ServerName www.mysite4.org`
`   DocumentRoot /var/www/vhosts/mysite4`
`</VirtualHost>`
	IP-based virtual host answering to `http://10.17.1.5`

Apache configuration file	
Authorization directives	
`AuthName "Protected zone"`	Name of the realm. The client will be shown the realm name and prompted to enter a user and password
`AuthType Basic`	Type of user authentication: `Basic`, `Digest`, `Form`, or `None`
`AuthUserFile "/var/www/.htpasswd"`	User database file. Each line has the format *user*`:`*encryptedpassword* To add a user to the database file, use the command: `htpasswd /var/www/.htpasswd` *user* (will prompt for password)
`AuthGroupFile "/var/www/.htgroup"`	Group database file. Each line specifies a group followed by the usernames of all its members: *group*`:` *user1 user2 user3*
`Require valid-user`	Control who can access the protected resource. `valid-user` any user in the user database file `user` *user* only the specified user `group` *group* only the members of the specified group
`Satisfy Any`	Set the access policy concerning user and host control. `All` both `Require` and `Allow` criteria must be satisfied `Any` any of `Require` or `Allow` criteria must be satisfied
`Allow from 10.13.13.0/24` `Deny from 10.13.14.0/24` (v2.2)	Control which host can or cannot access the protected resource
`Order Allow,Deny` (v2.2)	Control the evaluation order of `Allow` and `Deny` directives. `Allow,Deny` First, all `Allow` directives are evaluated; at least one must match, or the request is rejected. Next, all `Deny` directives are evaluated; if any matches, the request is rejected. Last, any requests which do not match an `Allow` or a `Deny` directive are denied `Deny,Allow` First, all `Deny` directives are evaluated; if any match, the request is denied unless it also matches an `Allow` directive. Any requests which do not match any `Allow` or `Deny` directives are permitted

Apache configuration file	
SSL/TLS directives (`mod_ssl` module)	

`SSLCertificateFile \`
`/etc/httpd/conf/ssl.crt/server.crt`
	SSL server certificate

`SSLCertificateKeyFile \`
`/etc/httpd/conf/ssl.key/server.key`
	SSL server private key (for security reasons, this file must be mode 600 and owned by root)

`SSLCACertificatePath \`
`/usr/local/apache2/conf/ssl.crt/`
	Directory containing the certificates of CAs. Files in this directory are PEM-encoded and accessed via symlinks to hash filenames

`SSLCACertificateFile \`
`/usr/local/apache2/conf/ssl.crt/ca-bundle.crt`
	Certificates of CAs. Certificates are PEM-encoded and concatenated in a single bundle file in order of preference

`SSLCertificateChainFile \`
`/usr/local/apache2/conf/ssl.crt/ca.crt`
	Certificate chain of the CAs. Certificates are PEM-encoded and concatenated from the issuing CA certificate of the server certificate to the root CA certificate. Optional

`SSLEngine on`	Enable the SSL/TLS Protocol Engine

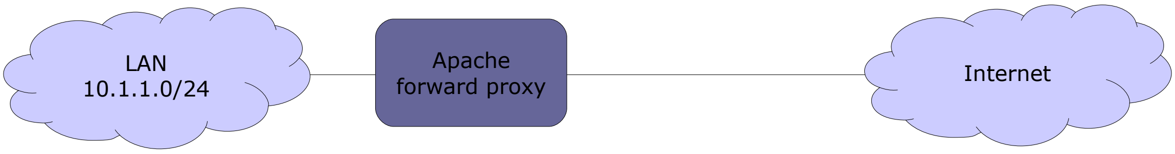
`SSLProtocol +SSLv3 +TLSv1.2`	SSL protocol flavors that the client can use to connect to server. Possible values are:
	`SSLv2` (deprecated)
	`SSLv3`
	`TLSv1`
	`TLSv1.1`
	`TLSv1.2`
	`All` (all the above protocols)

`SSLCipherSuite \`
`ALL:!aDH:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP`
	Cipher suite available for the SSL handshake (key exchange algorithms, authentication algorithms, cipher/encryption algorithms, MAC digest algorithms)

`ServerTokens Full`	Server response header field to send back to client. Possible values are:
	`Prod` sends `Server: Apache`
	`Major` sends `Server: Apache/2`
	`Minor` sends `Server: Apache/2.4`
	`Minimal` sends `Server: Apache/2.4.2`
	`OS` sends `Server: Apache/2.4.2 (Unix)`
	`Full` sends `Server: Apache/2.4.2 (Unix) \`
	`PHP/4.2.2 MyMod/1.2` (default)

`ServerSignature Off`	Trailing footer line on server-generated documents. Possible values are:
	`Off` no footer line (default)
	`On` server version number and `ServerName`
	`EMail` as above, plus a mailto link to `ServerAdmin`

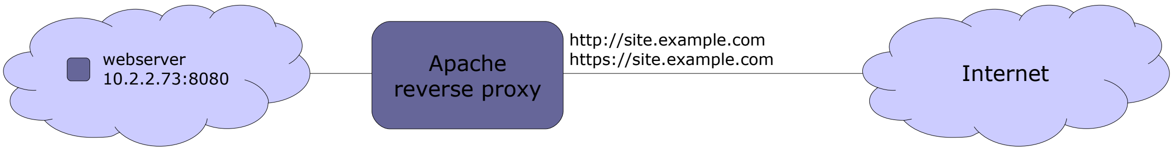
`SSLVerifyClient none`	Certificate verification level for client authentication. Possible values are: `none` no client certificate is required `require` the client needs to present a valid certificate `optional` the client may present a valid certificate (this option is unused as it doesn't work on all browsers) `optional_no_ca` the client may present a valid certificate but it doesn't need to be successfully verifiable (this option is practically used only for SSL testing)
`TraceEnable on`	Enable TRACE requests



A **forward proxy** provides proxy services, typically web content caching and/or filtering, for clients located in a LAN. All outgoing requests from the clients, and the responses from the Internet, pass through the proxy.
The clients must be manually configured (e.g. in the browser's connection settings) to use the proxy.

Apache configuration file	
Forward proxy	
`ProxyRequests On`	Enable forward proxy requests
`ProxyVia On`	Add a `Via:` HTTP header line to every request and reply

`<Proxy "*">`
`   Require ip 10.1.1`
`</Proxy>`
	Serve only proxy requests coming from 10.1.1.0/24



A **reverse proxy** aka **gateway** allows to expose a single entry point for one or more webservers in a LAN. This improves security and simplifies management, as features (e.g. load balancing, firewalling, automatic redirection from HTTP to HTTPS, redirection on default ports) can be configured centrally.
It is necessary to create a DNS A record that maps site.example.com to the public IP address of the proxy.

Apache configuration file	
Reverse proxy	
`<VirtualHost *:80>`	Virtual host for HTTP
`   ServerName site.example.com`	Define website name

`   RewriteEngine On`
`   RewriteCond %{HTTPS} off`
`   RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}`
Alternatively:
`   Redirect "/" "https://10.2.2.73:443/"`
	Redirect all HTTP requests to HTTPS

`</VirtualHost>`	
`<VirtualHost *:443>`	Virtual host for HTTPS
`   ServerName site.example.com`	Define website name
`   ServerSignature On`	Set a footer line under server-generated pages

`   <Proxy *>`
`      Require all granted`
`   </Proxy>`
	Serve all proxy requests

`   SSLEngine on`
`   SSLProtocol ALL -SSLv2 -SSLv3`
`   SSLHonorCipherOrder on`
`   SSLCipherSuite DEFAULT`
`   SSLCertificateFile    /etc/httpd/ssl/site.crt`
`   SSLCertificateKeyFile /etc/httpd/ssl/site.key`
`   SSLCACertificateFile  /etc/httpd/ssl/site.ca.crt`
	Enable and configure SSL

`   ProxyPass        "/" "http://10.2.2.73:8080/"`
`   ProxyPassReverse "/" "http://10.2.2.73:8080/"`
	Enable reverse proxying for server 10.2.2.73

`</VirtualHost>`	

Apache Tomcat is an open source Java Servlet Container implementing several Java EE specifications, originally part of the Jakarta Project. It is composed of:
- Catalina, the core component and servlet container implementation;
- Coyote, an HTTP connector component, providing a pure Java webserver environment to run Java code;
- Jasper, a JSP (Java Server Pages) engine, which parses JSP files and compiles them into Java servlets.

Tomcat has been removed from RHEL 8; instead, it is suggested to use the JBoss Enterprise Application Platform, which includes Apache and Tomcat.

`$JAVA_HOME`	Root of the Java installation e.g. `/usr/lib/jvm/java-1.8.0-openjdk.x86_64/`
`$CATALINA_HOME`	Root of the Tomcat installation e.g. `/usr/share/tomcat7/`
`$CATALINA_BASE`	Tomcat may also be configured for multiple instances by defining the variable `$CATALINA_BASE` for each instance. If a single instance of Tomcat is running, `$CATALINA_BASE` is the same as `$CATALINA_HOME`

Tomcat global files	
`$CATALINA_BASE/conf/server.xml`	Tomcat main configuration file
`$CATALINA_BASE/conf/web.xml`	Options and values applied to all web applications running on a specific Tomcat instance. These can be overridden by the application-specific servlet configuration defined in `$CATALINA_BASE/webapps/`*`appname`*`/WEB-INF/web.xml`
`$CATALINA_BASE/conf/context.xml`	Context applied to all web applications running on a specific Tomcat instance
`$CATALINA_BASE/conf/tomcat-users.xml`	Users, passwords, and roles applied to a specific Tomcat instance
`$CATALINA_BASE/conf/catalina.policy`	Tomcat's core security policy for the Catalina class
`$CATALINA_BASE/conf/catalina.properties`	Java properties file for the Catalina class
`$CATALINA_BASE/conf/logging.properties`	Java properties file for Catalina's built-in logging functions
`$CATALINA_BASE/lib/`	JAR files accessible by both web applications and internal Tomcat code
`$JAVA_HOME/jre/lib/security/`*`keystore`*`.jks`	Java keystore
Tomcat application-specific files	
`$CATALINA_BASE/webapps/`*`appname`*`/WEB-INF/`	HTML, JSP, and other files to serve to the client browser
`$CATALINA_BASE/webapps/`*`appname`*`/WEB-INF/web.xml`	Description of servlets and other components of the application, and initialization parameters
`$CATALINA_BASE/webapps/`*`appname`*`/WEB-INF/classes/`	Java class files that aren't in JAR format. The directory hierarchy from here reflects the class hierarchy
`$CATALINA_BASE/webapps/`*`appname`*`/WEB-INF/lib/`	Other JAR files (e.g. third-party libraries, JDBC drivers) required by the application
Tomcat log files	
`$CATALINA_BASE/logs/catalina.out`	Tomcat log
`$CATALINA_BASE/logs/localhost.log`	Host log
`$CATALINA_BASE/logs/localhost_access.log`	Host HTTP access log
`$CATALINA_BASE/logs/manager.log`	Application log
`$CATALINA_BASE/logs/host-manager.log`	Application log

`java -X`	Display all available `-X` options (nonstandard HotSpot JVM options)
`java -XshowSettings:properties -version`	Print Java runtime settings

Samba is a free-software, cross-platform implementation of SMB/CIFS. SMB (Server Message Block) is Microsoft's proprietary protocol for file and printer sharing, while CIFS (Common Internet File System) is the public version of SMB.

Commonly used ports in Samba		
TCP/UDP 137	netbios-ns	NetBIOS name service requests and responses
TCP/UDP 138	netbios-dgm	NetBIOS datagram services e.g. server announcements
TCP/UDP 139	netbios-ssn	NetBIOS session service e.g. file and printer sharing
TCP 445	microsoft-ds	Active Directory; registration and translation of NetBIOS names, network browsing
TCP 389		LDAP
TCP 901		SWAT service

The full list of used ports can be found via the command `grep -i netbios /etc/services`.

`smbd`	Server Message Block daemon. Provides SMB file and printer sharing, browser services, user authentication, and resource lock. An extra copy of this daemon runs for each client connected to the server
`nmbd`	NetBIOS Name Service daemon. Handles NetBIOS name lookups, WINS requests, list browsing and elections. An extra copy of this daemon runs if Samba functions as a WINS server; another extra copy of this daemon runs if DNS is used to translate NetBIOS names. WINS (Windows Internet Name Service) is a name service used to translate NetBIOS names to IP addresses.

`/etc/smb/` `/etc/samba/` (RHEL 7)	Samba directory
`/etc/samba/lmhosts`	Samba NetBIOS hosts file
`/etc/samba/netlogon`	User logon directory

`smbd -V`
`smbclient -V`
	Show the version of the Samba server

`testparm`	Check the Samba configuration file and report any error
smbpasswd *user*	Change the Samba password of *user*
smbpasswd -a *user*	Create a new Samba *user* and set his password
nmblookup *smbserver*	Look up the NetBIOS name of a server and map it to an IP address
nmblookup -U *winsserver* -R *WORKGROUP*#1B	Query recursively a WINS server for the Domain Master Browser for the specified workgroup
nmblookup -U *winsserver* -R *WORKGROUP*#1D	Query recursively a WINS server for the Domain Controller for the specified workgroup
`net`	Tool for administration of Samba and remote CIFS servers
net rpc shutdown -r -S *smbserver* -U root%*password*	Reboot a CIFS server
net rpc service list -S *smbserver*	List available services on a CIFS server
`net status sessions`	Show active Samba sessions
`net status shares`	Show Samba shares
`net rpc info`	Show information about the domain
`net groupmap list`	Show group mappings between Samba and Windows

mount.cifs smbmount	Mount a Samba share on a Linux filesystem, using the CIFS filesystem interface
mount //*smbserver*/share1 /mnt/share1 -t cifs \\ -o username=*user*	Mount a Samba share as *user*
smbstatus	Display current information about shares, clients connections, and locked files
smbclient //*smbserver*/share1	Access a Samba share on a server (with an FTP-like interface)
smbclient -L //*smbserver* -W *WORKGROUP* -U *user*	List the Samba resources available on a server, belonging to the specified workgroup and accessible to the specified user
cat msg.txt | smbclient -M *client* -U *user*	Show a message popup on the client machine, using the WinPopup protocol

Samba mount options	
username=*user*	Mount the share as *user*
password=*password*	Specify the mount user's *password*

credentials=*credfile*	Mount the share as the user defined in the credentials file *credfile* which must have this format:
	username=*user*
	password=*password*

multiuser	Mount the share in multiuser mode
sec=ntlmssp	Set the security level to NTLMSSP. This is required in RHEL 7 to enable multiuser mode

Samba global configuration

`/etc/samba/smb.conf`	**Samba configuration**
`[global]`	Global server settings: defines parameters applicable for the whole Samba server and sets the defaults that will be used for the parameters not mentioned in other sections
`workgroup = MYWORKGROUP`	Make Samba join the specified workgroup
`server string = Linux Samba Server %L`	Describe server to the clients
`hosts allow = 10.9.9.0/255.255.255.0`	Allow only the specified machines to connect to the server
`security = user`	Set up user-level authentication
`encrypt passwords = yes`	Use encrypted passwords

`smb passwd file = /etc/samba/smbpasswd`	Refer to the specified password file for user authentication.
	A new user's password will need to be set both in Linux and Samba by using these commands from shell prompt:
	`passwd` *`newuser`*
	`smbpasswd` *`newuser`*

`unix password sync = yes`	When the password of a client user (e.g. under Windows) is changed, change the Linux and Samba passwords accordingly

`username map = /etc/samba/smbusers`	Map each Samba server user name to client user name(s).
	The file `/etc/samba/smbusers` has the following format:
	`root = Administrator Admin`
	`jdoe = "John Doe"`
	`kgreen = "Kim Green"`

`netbios name = Mysambabox`
`netbios aliases = Mysambabox1`
	Set NetBIOS name and alias

`wins support = yes`	Make Samba play the role of a WINS server. Note: There should be only one WINS server on a network
`logon server = yes`	Enable logon support. Logon script parameters will be defined in a `[netlogon]` section
`log file = /var/log/samba/log.%m`	Use a separate logfile for each machine that connects
`max log size = 1000`	Maximum size of each logfile, in Kb
`syslog only = no`	Do not use only syslog to log
`syslog = 0`	Log everything to the logfiles `/var/log/smb/log.smbd` and `/var/log/smb/log.nmbd`, and log a minimum amount of information to syslog. This parameter can be set to a higher value to have syslog log more information

`panic action = \`
`/usr/share/samba/panic-action %d`
	Mail a backtrace to the sysadmin in case Samba crashes

`[netlogon]`
`comment = Netlogon for Windows clients`
	Section defining a logon script

`path = /home/netlogon`
`logon script = %U.bat`
	Specifies a per-user script e.g. `/home/netlogon/jdoe.bat` will be called when user jdoe logs in.
	It is also possible to specify a per-clientname script `%m.bat`, which will be called when a specific machine logs in.

`browseable = no`
`writeable = no`	

`guest ok = no`	Guest access to the service (i.e. access without entering a password) is disabled

`[Canon LaserJet 3]`
`printer name = lp`
`comment = Canon LaserJet 3 main printer`
`path = /var/spool/lpd/samba`
`printable = yes`
`writeable = no`
	Section defining a printer accessible via the network

`/etc/samba/smb.conf`	**Samba configuration**
`[public]`	Section defining a public share accessible on read/write by anyone
`comment = Public Storage on %L`	Describe the public share to users
`path = /home/samba`	Path of the public share on the server
`browsable = yes`	Show the public share when browsing
`writeable = yes`	Allow all users to write in this directory
`[homes]`	Section enabling users that have an account and a home directory on the Samba server to access it and modify its contents from a Samba client. The `path` variable is not set, by default is `path=/home/%S`
`comment = %U's home directory on %L from %m`	Describe the share to the user
`browseable = no`	Do not show the homes share when browsing
`writeable = yes`	Allow the user to write in his home directory
`[foobar]`	Section defining a specific share
`path = /foobar`	Path of the share on the server
`comment = Share Foobar on %L from %m`	Describe the share to users
`browsable = yes`	Show the share when browsing
`writeable = yes`	Allow the users to write in this share
`valid users = jdoe, kgreen, +geeks`	Allow access only to users "jdoe" and "kgreen", and to local group "geeks"
`invalid users = csmith`	Deny access to user "csmith"
`read list = bcameron`	Allow read-only access to user "bcameron"
`write list = fcastle`	Allow read-write access to user "fcastle"

`/etc/samba/smb.conf` **Samba configuration**	
User-level authentication	
`[global]`	
`   security = user`	Set up user-level authentication
`   guest account = nobody`	Map the guest account to the system user nobody (default)
`   map to guest = Never`	Specify how incoming requests are mapped to the guest account: `Bad User` redirect from an invalid user to guest account on server `Bad Password` redirect from an invalid password to guest account on server `Never` reject unauthenticated users
Server-level authentication	
`[global]`	
`   security = server`	Set up server-level authentication
`   password server = `*`srv1 srv2`*	Authenticate to server *srv1*, or to server *srv2* if the first one is unavailable
Domain-level authentication	
`[global]`	
`   security = ADS`	Set up domain-level authentication as an Active Directory member server
`   realm = KRB_REALM`	Join the specified realm. Kerberos must be installed and an administrator account must be created: `net ads join -U Administrator%`*`password`*
Share-level authentication	
`[global]`	
`   security = share`	Set up share-level authentication

`[foobar]`
`   path = /foobar`
`   username = `*`user`*
`   only user = yes`
	Define a "foobar" share accessible to any user which can supply *user*'s password.
	The *user* must be created on the system:
	`useradd -c "Foobar account" -d /tmp -m -s /sbin/nologin `*`user`*
	and added to the Samba password file:
	`smbpasswd -a `*`user`*

Samba macros

`%S`	Username
`%U`	Session username (the username that the client requested, not necessarily the same as the one he got)
`%G`	Primary group of session username
`%h`	Samba server hostname
`%M`	Client hostname
`%L`	NetBIOS name of the server
`%m`	NetBIOS name of the client
`%d`	Process ID of the current server process
`%a`	Architecture of remote machine
`%I`	IP address of client machine
`%i`	Local IP address to which a client connected
`%T`	Current date and time
`%D`	Domain or workgroup of the current user
`%w`	Winbind separator
`%$(`*`var`*`)`	Value of the environment variable *var*

Macros applied only to configuration options used once a connection has been established:

`%S`	Name of the current service, if any
`%P`	Root directory of the current service, if any
`%u`	Username of the current service, if any
`%g`	Primary group name of username
`%H`	Home directory of username
`%N`	Name of the NIS home directory server as obtained from the NIS `auto.map` entry. Same as `%L` if Samba was not compiled with the `--with-automount` option
`%p`	Path of service's home directory as obtained from the NIS `auto.map` entry. The NIS `auto.map` entry is split up as `%N:%p`

Samba setup

This procedure allows sharing on read-write the local directory `/smbshare` on server 10.1.1.1 to client 10.2.2.2.

Server setup:

1.	Create the group for write access to the share	`groupadd -r geeks`
2.	Create the user and assign it to the group	`useradd -G geeks jdoe`
3.	Add the user to Samba. You will be prompted to enter a password	`smbpasswd -a jdoe`
4.	Assign correct ownership to the share	`chgrp geeks /smbshare`
5.	Set the SGID bit to the share	`chmod 2775 /smbshare`

6.	Set the correct SELinux label to the share	`semanage fcontext -a -t samba_share_t '/smbshare'`
		`restorecon -FR /smbshare`

7.	Enable the SELinux boolean for write access to the share	`setsebool -P samba_export_all_rw=on`

8. Add a section for the share on `/etc/samba/smb.conf`:

```
[smbshare]
   path = /smbshare
   hosts allow = 10.2.2.2
   write list = @geeks
```

9. Ensure that the `smb` and `nmb` services are running

Client setup:

1. Add an entry to `/etc/fstab` to mount the Samba share device automatically:

```
//10.1.1.1/smbshare  /mountpoint  cifs  username=jdoe,password=s3cr3t  0 0
```

Client multiuser setup:

1. Add an entry to `/etc/fstab` to mount the Samba share device automatically in multiuser mode:

```
//10.1.1.1/smbshare  /mountpoint  cifs  username=jdoe,password=s3cr3t,multiuser,sec=ntlmssp  0 0
```

2.	Login as another user (there must be a matching Samba user on the Samba server 10.1.1.1)	`su - ksmith`
3.	Store the Samba username and password in the kernel keyring for the current session	`cifscreds add 10.1.1.1`

A Network File System (NFS) server makes filesystems available to remote clients for mounting.

NFS requires the portmapper to map incoming TCP/IP connections to the appropriate NFS RPC calls. Some Linux distributions use rpcbind instead of the portmapper.
For security reasons, the TCP Wrapper should be configured to limit access to the portmapper to NFS clients only:
file `/etc/hosts.deny` should contain `portmap: ALL`
file `/etc/hosts.allow` should contain `portmap:` *`IP_addresses_of_clients`*

NFS handles user permissions across systems by considering users with same UID and username as the same user.
Group permission is evaluated similarly, by GID and groupname.

`rpc.nfsd`
`rpc.mountd`
`rpc.lockd`
`rpc.statd`
	NFS daemons

`/etc/exports`	List of the filesystems to be exported (via the command `exportfs`)
`/var/lib/nfs/xtab`	List of exported filesystems, maintained by `exportfs`
`/proc/fs/nfs/exports`	Kernel export table (can be examined via the command `cat`)
`exportfs -ra`	Export or reexport all directories. When exporting, fills the kernel export table `/proc/fs/nfs/exports`. When reexporting, removes the entries in `/var/lib/nfs/xtab` that are deleted from `/etc/exports` (therefore synchronizing the two files), and removes the entries from `/proc/fs/nfs/exports` that are no longer valid
`exportfs -ua`	Unexport all directories. Removes from `/proc/fs/nfs/exports` the entries that are listed in `/var/lib/nfs/xtab`, and clears the latter file
`showmount`	Show the remote client hosts currently having active mounts
`showmount --directories`	Show the directories currently mounted by a remote client host
`showmount --exports`	Show the filesystems currently exported i.e. the active export list
`showmount --all`	Show both remote client hosts and directories
`showmount -e` *`nfsserver`*	Show the shares a NFS server has available for mounting
`rpcinfo -p` *`nfsserver`*	Probe the portmapper on a NFS server and display the list of all registered RPC services there
`rpcinfo -t` *`nfsserver`* `nfs`	Test a NFS connection by sending a null pseudo request (using TCP)
`rpcinfo -u` *`nfsserver`* `nfs`	Test a NFS connection by sending a null pseudo request (using UDP)
`nfsstat`	Display NFS/RPC client/server statistics.

Options:

	NFS	**RPC**	**both**
server	`-sn`	`-sr`	`-s`
client	`-cn`	`-cr`	`-c`
both	`-n`	`-r`	`-nr`

`mount -t nfs` *`nfsserver`*`:/share /usr`	Command to be run on a client to mount locally a remote NFS share. NFS shares accessed frequently should be added to `/etc/fstab` e.g. *`nfsserver`*`:/share   /usr   nfs   intr   0 0`

/etc/exports

```
/export/           10.3.3.3(rw)
/export2/          10.4.4.0/24
/export3/          *(ro,sync)
/home/ftp/pub      myhost(rw)   *.example.org(ro)
/home/crew         @FOOWORKGROUP(rw)   (ro)
```

filesystem	Filesystem on the NFS server to be exported to clients	
client identity	Client systems permitted to access the exported directory. Can be specified by hostname, IP address, wildcard, subnet, or @NIS workgroup. Multiple client systems can be listed, and each one can have different options	
client options	`ro`	Read-only access (default)
	`rw`	Read and write access. The client may choose to mount read-only anyway
	`sync`	Reply to requests only after the changes made by these requests have been committed to stable storage
	`async`	Reply to requests without waiting that changes are committed to stable storage. Improves performances but might cause loss or corruption of data if server crashes
	`root_squash`	Requests by user `root` on client will be done as user `nobody` on server (default)
	`no_root_squash`	Requests by user `root` on client will be done as same user `root` on server
	`all_squash`	Requests by a non-root user on client will be done as user `nobody` on server
	`no_all_squash`	Requests by a non-root user on client will be attempted as same user on server (default)

NFS mount options	
`rsize=`*`nnn`*	Size for read transfers (from server to client)
`wsize=`*`nnn`*	Size for write transfers (from client to server)
`nfsvers=`*`n`*	Use NFS version *n* for transport
`retry=`*`n`*	Keep retrying a mount attempt for *n* minutes before giving up
`timeo=`*`n`*	A mount attempt times out after *n* tenths of a second
`intr`	User can interrupt a mount attempt
`nointr`	User cannot interrupt a mount attempt (default)
`hard`	The system will try a mount indefinitely (default)
`soft`	The system will try a mount until an RPC timeout occurs
`bg`	Try a mount in the foreground; all retries occur in the background
`fg`	All mount attempts occur in the foreground (default)
`tcp`	Connect using TCP
`udp`	Connect using UDP
`sec=krb5p`	Use Kerberos to encrypt all requests between client and server
`v4.2`	Enable NFS v4.2, which allows the server to export the SELinux context

NFS setup

This procedure allows sharing on read-write the local directory `/nfsshare` on server 10.1.1.1 to client 10.2.2.2.

Server setup:

1. Ensure that the `nfs-server` service is running
2. Change ownership of the share `chown nfsnobody /nfsshare`
3. Add an entry for the share on `/etc/exports`:

```
/nfsshare  10.2.2.2(rw)
```

4. Reload the exports file `exportfs -r`

Client setup:

1. Add an entry to `/etc/fstab` to mount the NFS share device automatically:

```
10.1.1.1:/nfsshare  /mountpoint  nfs  defaults  0 0
```

Secure NFS setup

This procedure allows sharing on read-write the local directory `/nfsshare` on server 10.1.1.1 to client 10.2.2.2, securely with Kerberos enabled.

Server setup:

1. Install the appropriate server keytab on `/etc/krb5.keytab`
2. Ensure that the `nfs-secure-server` service is running
3. Change ownership of the share `chown nfsnobody /nfsshare`
4. Add an entry for the share on `/etc/exports`:

```
/nfsshare  10.2.2.2(sec=krb5p,rw)
```

5. Reload the exports file `exportfs -r`

Client setup:

1. Install the appropriate client keytab on `/etc/krb5.keytab`
2. Ensure that the `nfs-secure` service is running
3. Add an entry to `/etc/fstab` to mount the NFS share device automatically:

```
10.1.1.1:/nfsshare  /mountpoint  nfs  defaults,sec=krb5p  0 0
```

iSCSI (Internet Small Computer System Interface) is a network protocol that allows emulating an SCSI local storage device over a TCP/IP network. By default it uses TCP port 3260.
An iSCSI server can use a local block device (physical or virtual disk, disk partition, or Logical Volume), a file, a physical SCSI device, or a ramdisk as the underlying storage resource (**backstore**) and make it available by assigning it a **LUN** (Logical Unit Number). An iSCSI server provides one or more **targets**, each of which presents one or more LUNs and is able to accept connections from an iSCSI client (**initiator**).
Targets and initiators are called **nodes** and are identified by a unique **IQN** (iSCSI Qualified Name) e.g. `iqn.2017-11.org.example.subdomain:foo:bar`. The IP address and port of a node is called a **portal**.
A target accepts connections from an initiator via a **TPG** (Target Portal Group) i.e. its IP address and port. A TPG may have in place an **ACL** so to accept connections only from a specific initiator's IQN.

`targetcli`	Target configurator (server side). Can be used as a command line tool or as an interactive shell. Configuration is saved to `/etc/target/saveconfig.json`
`iscsiadm`	Administration tool for iSCSI devices (client side)

iSCSI setup

This procedure makes available the local disk `/dev/sbd` on server 10.1.1.1 to the client having IQN `iqn.2017-11.org.example:client`.

Server (target) setup:

1. Ensure that the `targetcli` service is running
2. Enter the targetcli shell

   ```
   targetcli
   ```

3. Create a backstore

   ```
   cd /backstores/block
   create mydisk /dev/sdb
   ```

4. Create a IQN for the target.
   This automatically creates a TPG for the IQN

   ```
   cd /iscsi
   create iqn.2017-11.org.example:target
   ```

5. On the TPG, create an ACL to allow connections from the initiator with a specific IQN

   ```
   cd /iscsi/iqn.2017-11.org.example:target/tpg1/acls
   create iqn.2017-11.org.example:client
   ```

6. On the TPG, create a LUN for the backstore

   ```
   cd /iscsi/iqn.2017-11.org.example:target/tpg1/luns
   create /backstores/block/mydisk
   ```

7. On the TPG, create a portal listening from the server's IP address

   ```
   cd /iscsi/iqn.2017-11.org.example:target/tpg1/portals
   delete 0.0.0.0 ip_port=3260
   create 10.1.1.1
   ```

8. Verify the configuration

   ```
   ls /
   ```

```
o- / ......................................................................................... [...]
  o- backstores .............................................................................. [...]
  | o- block .................................................................. [Storage Objects: 1]
  | | o- mydisk ......................................... [/dev/sdb (100.0MiB) write-thru activated]
  | |   o- alua ................................................................... [ALUA Groups: 1]
  | |     o- default_tg_pt_gp ....................................... [ALUA state: Active/optimized]
  | o- fileio ................................................................. [Storage Objects: 0]
  | o- pscsi .................................................................. [Storage Objects: 0]
  | o- ramdisk ................................................................ [Storage Objects: 0]
  o- iscsi ............................................................................ [Targets: 1]
  | o- iqn.2017-11.org.example:target .................................................... [TPGs: 1]
  |   o- tpg1 ............................................................... [no-gen-acls, no-auth]
  |     o- acls .......................................................................... [ACLs: 1]
  |     | o- iqn.2017-11.org.example:client ....................................... [Mapped LUNs: 1]
  |     |   o- mapped_lun0 ................................................ [lun0 block/mydisk (rw)]
  |     o- luns .......................................................................... [LUNs: 1]
  |     | o- lun0 .................................... [block/mydisk (/dev/sdb) (default_tg_pt_gp)]
  |     o- portals .................................................................... [Portals: 1]
  |       o- 10.1.1.1:3260 ....................................................................... [OK]
  o- loopback ......................................................................... [Targets: 0]
```

9. Exit the targetcli shell.
   Configuration is automatically saved

   ```
   exit
   ```

Client (initiator) setup:

1. Set the correct initiator IQN in the file `/etc/iscsi/initiatorname.iscsi`:

   ```
   InitiatorName=iqn.2017-11.org.example:client
   ```

2. Ensure that the `iscsi` service is running
3. Discover the iSCSI target(s) provided by the portal. This echoes the target(s) IQN found

   ```
   iscsiadm -m discovery -t sendtargets -p 10.1.1.1
   ```

4. Login to the target IQN found

   ```
   iscsiadm -m node -T iqn.2017-11.org.example:target -p 10.1.1.1 -l
   ```

   The iSCSI device is now locally available and can be formatted and mounted. Node records remain after logout or reboot; the system will login again to the target IQN automatically

5. Add an entry to `/etc/fstab` to mount the iSCSI device automatically:

   ```
   UUID=nnnnnnnn-nnnn-nnnn-nnnn-nnnnnnnnnnnn  /mountpoint  fstype  _netdev  0 0
   ```

DHCP (Dynamic Host Configuration Protocol) is a protocol for network management that automatically provides a requesting host with an IP address and other network configuration parameters. It is based on BOOTP (Bootstrap Protocol).
A DHCP server listens for requests on UDP port 67 and answers to UDP port 68. The assignment of an IP address to a host is done through a sequence of DHCP messages initiated by the client host: DHCP Discover, DHCP Offer, DHCP Request, and finally DHCP Acknowledgment.
Because DHCP Discover messages are broadcast and therefore not routed outside a LAN, a DHCP relay agent is necessary for those clients situated outside the DHCP server's LAN. The DHCP relay agent listens to DHCP Discover messages and relays them in unicast to the DHCP server.

`/etc/dhcpd.conf`	Configuration file for the DHCP server
`/etc/sysconfig/dhcrelay` (SUSE)	Configuration file for the DHCP relay agent
`/var/lib/dhcpd/dhcpd.leases`	DHCP current leases

`/etc/dhcpd.conf` **DHCP server configuration**	

`option domain-name-servers 10.2.2.2;`
`option smtp-servers 10.3.3.3;`
`option pop-servers 10.4.4.4;`
`option time-servers 10.5.5.5;`
`option nntp-servers 10.6.6.6;`
	Global parameters for DNS, mail, NTP, and news servers specification

`shared-network geek-net {`	Definition of a network
`   default-lease-time 86400;`	Time, in seconds, that will be assigned to a lease if a client does not ask for a specific expiration time
`   max-lease-time 172800;`	Maximum time, in seconds, that can be assigned to a lease if a client asks for a specific expiration time

`   option routers 10.0.3.252;`
`   option broadcast-address 10.0.3.255;`	

`   subnet 10.0.3.0 netmask 255.255.255.128 {`
`      range 10.0.3.1 10.0.3.101;`
`   }`
`   subnet 10.0.3.128 netmask 255.255.255.128 {`
`      range 10.0.3.129 10.0.3.229;`
`   }`
`}`
	Definition of different subnets in the network, with specification of different ranges of IP addresses that will be leased to clients depending on the client's subnet

`group {`	Definition of a group

`   option routers 10.0.17.252;`
`   option broadcast-address 10.0.17.255;`
`   netmask 255.255.255.0;`	

`   host linuxbox1 {`
`      hardware ethernet AA:BB:CC:DD:EE:FF;`
`      fixed-address 10.0.17.42;`
`      option host-name "linuxbox1";`
`   }`
`   host linuxbox2 {`
`      hardware ethernet 33:44:55:66:77:88;`
`      fixed-address 10.0.17.66;`
`      option host-name "linuxbox2";`
`   }`
`}`
	Definition of different hosts to whom static IP addresses will be assigned to, depending on their MAC address

PAM (Pluggable Authentication Modules) is an abstraction layer that allows applications to use authentication methods while being implementation-agnostic.

`/etc/pam.d/`*`service`*	PAM configuration for *service*
`/etc/pam.conf` (obsolete)	PAM configuration for all services
`ldd /usr/sbin/`*`service`*` | grep libpam`	Check if *service* is enabled to use PAM

/etc/pam.d/*service*

```
auth       requisite   pam_securetty.so
auth       required    pam_nologin.so
auth       required    pam_env.so
auth       required    pam_unix.so nullok
account    required    pam_unix.so
session    required    pam_unix.so
session    optional    pam_lastlog.so
password   required    pam_unix.so nullok obscure min=4 max=8
```

type	`auth`	Authentication module to verify user identity and group membership
	`account`	Authorization module to determine user's right to access a resource (other than his identity)
	`password`	Module to update a user's authentication credentials
	`session`	Module (run at end and beginning of a user session) to set up the user environment
control	`optional`	Module is not critical to the success or failure of *service*
	`sufficient`	If this module successes, and no previous module has failed, module stack processing ends successfully. If this module fails, it is non-fatal and processing of the stack continues
	`required`	If this module fails, processing of the stack continues until the end, and *service* fails
	`requisite`	If this module fails, *service* fails and control returns to the application that invoked *service*
	`include`	Include modules from another PAM service file
module	PAM module and its options, e.g.:	
	`pam_unix.so`	Standard UNIX authentication module via `/etc/passwd` and `/etc/shadow`
	`pam_nis.so`	Module for authentication via NIS
	`pam_ldap.so`	Module for authentication via LDAP
	`pam_fshadow.so`	Module for authentication against an alternative shadow passwords file
	`pam_cracklib.so`	Module for password strength policies (e.g. length, case, max number of retries)
	`pam_limits.so`	Module for system policies and system resource usage limits
	`pam_listfile.so`	Module to deny or allow the service based on an arbitrary text file

LDAP (Lightweight Directory Access Protocol) is a simplified version of the X.500 standard and uses TCP port 389.
LDAP allows to organize hierarchically a database of entries, each one of which is identified by a unique **DN (Distinguished Name)**. Each DN has a set of **attributes**, and each attribute has a **value**; an attribute may appear multiple times.
Special attributes called **objectClass** define which attributes are allowed and which are required, and determine the **schema** of the LDAP.

`dn: cn=John Doe,dc=example,dc=org`		Distinguished Name
Examples of LDAP attributes		
Attribute	**Attribute with value**	**Meaning**
`cn`	`cn: John Doe`	Common Name
`dc`	`dc=example,dc=org`	Domain Component
`givenName`	`givenName: John`	First name
`sn`	`sn: Doe`	Surname
`mail`	`mail: jdoe@example.org`	Email address
`telephoneNumber`	`telephoneNumber: +1 555 1234 567`	Telephone number
`uid`	`uid: jdoe`	User ID
`c`	`c: US`	Country code
`l`	`l: San Francisco`	Locality
`st`	`st: California`	State or province
`street`	`street: 42, Penguin Road`	Street
`o`	`o: The Example Foundation`	Organization
`ou`	`ou: IT Dept`	Organizational Unit
`manager`	`manager: cn=Kim Green,dc=example,dc=org`	Manager

LDIF (LDAP Data Interchange Format)	

`dn: cn=John Doe, dc=example, dc=org`
`changetype: modify`
`replace: mail`
`mail: johndoe@otherexample.com`
`-`
`add: jpegPhoto`
`jpegPhoto:< file://tmp/jdoe.jpg`
`-`
`delete: description`
`-`
	This LDIF file will change the email address of user "jdoe", add a picture, and delete the description attribute for the entry

All the LDAP commands below accept the following arguments, plus some extra arguments which are command-dependent.

	`-H ldap://`*`srv`*	Connect to the specified LDAP server
	`-H ldapi://`	Connect to the localhost LDAP server using IPC instead of a network socket
	`-D` *`binddn`*	Bind (authenticate) to the LDAP server as the specified DN
	`-w` *`password`*	Authenticate with the specified *password*
	`-W`	Prompt for authentication
	`-x`	Use simple authentication instead of SASL
	`-v`	Use verbose mode for output
`ldapsearch` *`args`*	Query a LDAP server and return the output in LDIF	
	`-b` *`base`*	Start searching from *base*
	`-z` *`n`*	Retrieve at maximum *n* entries as result
	`-LLL`	Terse output. Outputs the result in LDIFv1, does not print comments, and omits the LDIF version number
	`filter`	Search filter. If not specified, uses the default filter `(objectClass=*)`
	`attributes`	Attributes to return. If not specified, returns all attributes
`ldapmodify` *`args`*	Modify a LDAP entry	

`ldapadd` *`args`*
`ldapmodify -a` *`args`*
	Add a LDAP entry	

`ldapdelete` *`args`*	Delete a LDAP entry	
	`-f` *`file.ldif`*	Modify the entry according to the LDIF file
`ldappasswd` *`args`*	Change the password of a LDAP entry	
	`-s` *`password`*	Set the new password as *password*
	`-S`	Prompt for the new password

`ldapsearch -H ldap://ldap.example.org -s base \`
`-b "ou=people,dc=example,dc=com" "(sn=Doe)" \`
`cn sn telephoneNumber`
	Query a LDAP server for entries in the OU "people" whose surname is "Doe"; print common name, surname, and telephone number of the entries found

`ldapmodify -b -r -f` *`file.ldif`*	Modify an entry according to the LDIF file specified

`ldapadd -h ldap.example.org \`
`-D "cn=Admin,dc=example,dc=org" -W -f` *`file.ldif`*
	Authenticating as "Admin", add an entry by adding the content of the specified LDIF file to the directory

`ldapdelete -h ldap.example.org \`
`-D "cn=Admin,dc=example,dc=org" -W \`
`"uid=jdoe,dc=example,dc=org"`
	Authenticating as "Admin", delete the user "jdoe"

`ldappasswd -h ldap.example.org \`
`-D "cn=Admin,dc=example,dc=org" -W -x \`
`-S "uid=jdoe,ou=IT Dept,dc=example,dc=org"`
	Authenticating as "Admin" on example.org, change the password of user "jdoe" in the OU "IT Dept"

OpenLDAP

OpenLDAP is an open source implementation of LDAP, and was initially developed together with the LDAP protocol. Its related service is `slapd`, the Standalone OpenLDAP daemon.
`sssd`, the System Security Services Daemon, can be used to provide access to OpenLDAP as an authentication and identity provider.

`/var/lib/ldap/`	Files constituting the OpenLDAP database
`/etc/openldap/slapd.conf` `/usr/local/etc/openldap/slapd.conf`	OpenLDAP configuration file (deprecated)
`/usr/local/etc/openldap/slapd.d/`	From v2.3 onwards, directory containing the LDIF database that stores the OpenLDAP configuration. These LDIF files must not be edited by hand

`slapcat -b cn=config`
`ldapsearch -Y EXTERNAL -H ldapi:/// -b cn=config`
	Show the OpenLDAP configuration

`slaptest -u`	Verify that the OpenLDAP configuration is correct
`slapcat -l file.ldif`	Dump the contents of an OpenLDAP database to an LDIF file
`slapadd -l file.ldif`	Import an OpenLDAP database from an LDIF file
`slapindex`	Regenerate OpenLDAP's database indexes
`yum install openldap openldap-clients authconfig \` `sssd nss-pam-ldapd authconfig-gtk` (RHEL 7)	Install the OpenLDAP client
authconfig --enableldap --enableldapauth \\ --ldapserver=ldap://*ldapserver* \\ --ldapbasedn="dc=example,dc=org" \\ --enablesssd --update (RHEL 7)	Set up the LDAP client to connect to a *ldapserver*. This will update the configuration files `/etc/sssd/sssd.conf` and `/etc/openldap/ldap.conf`
`authselect select sssd --force` (RHEL 8)	Set up LDAP client authentication via `sssd`
`authconfig-gtk` `system-config-authentication`	OpenLDAP configuration GUI
getent group *groupname*	Get entries about *groupname* from NSS libraries

Security-Enhanced Linux (SELinux) is a Linux kernel security module that provides a mechanism for supporting access control security policies.

SELinux implements a Mandatory Access Control framework that allows the definition of fine-grained permissions for how **subjects** (i.e. processes) access **objects** (i.e. other processes, files, devices, ports, sockets); this improves security with respect to the traditional Discretionary Access Control, which defines accesses based on users and groups.
Processes, files, and users have a **security context** structured as *user:role:type:level* e.g. `unconfined_u:object_r:user_home_t:s0`. The third field defines a *type* for files or a *domain* for processes.
The decisions SELinux takes about allowing or disallowing access are stored in the **AVC (Access Vector Cache)**.

SELinux creates a pseudo filesystem (SELinuxfs) containing commands used by the kernel for its operations; this filesystem is usually mounted on `/selinux/` or `/sys/fs/selinux/`.

setenforce 0 echo 0 > /*selinuxfs*/enforce	Enter permissive mode (SELinux must be enabled)
setenforce 1 echo 1 > /*selinuxfs*/enforce	Enter enforcing mode (SELinux must be enabled)
getenforce cat /*selinuxfs*/enforce	Display current mode
sestatus -v	Show SELinux mode, SELinuxfs mount point, etc.

SELinux state can be configured permanently in `/etc/selinux/config` (symlinked in `/etc/sysconfig/selinux`):

mode	SELINUX=	enforcing	SELinux fully enforces security policies
		permissive	SELinux does not enforce security policies, but logs violations
		disabled	SELinux security policies are disabled
policy	SELINUXTYPE=	targeted	SELinux protects targeted daemons
		strict	(up to RHEL 6) SELinux fully protects the system
		minimum	(RHEL 7 and later) SELinux only protects selected processes
		mls	(RHEL 7 and later) Multi Level Security protection

ls -Z	List files and their security context. The security context of a file is stored in its extended attributes
ps -eZ	List processes and their security context
tar --selinux *otherargs* star -xattr -H=exustar *otherargs*	Create or extract archives that retain the security context of the original files

`chcon context file`	Change the security context of *file* to the specified *context*
`chcon --reference=file0 file`	Change the security context of *file* to be the same as *file0*
`restorecon -f file`	Restore the security context of *file* to the system default
`getsebool boolean`	Get the value of a SELinux boolean
`setsebool boolean=value`	Set the value of a SELinux boolean
`semanage`	Manage SELinux policies
`semanage fcontext -l`	List files and their assigned SELinux labels
`semanage fcontext -a -t label file`	Assign the SELinux *label* to *file*. It is then necessary to apply the label via `restorecon -f file`
`semanage login -l`	List mappings between users and SELinux users
`semanage port -l`	List port numbers and their assigned SELinux type definitions
`semanage port -a -t portlabel -p tcp n`	Assign the SELinux *portlabel* to TCP port *n*
`semanage port -a -t http_port_t -p tcp 8888`	Allow a local webserver to serve content on port 8888
`semanage port -d -t http_port_t -p tcp 8888`	Remove the binding of `http_port_t` port label to TCP 8888
`semanage port -m -t http_cache_port_t -p tcp 8888`	Modify the port label bound to TCP 8888
`semanage permissive -a auditd_t`	Add `auditd_t` to the list of permissive types/domains. In this case, SELinux allows the `auditd` daemon all access while logging its AVC violations
`semanage permissive -d auditd_t`	Delete `auditd_t` from the list of permissive types/domains
`semanage permissive -l`	List all permissive types/domains
`sepolicy`	Inspect a SELinux policy
`sepolicy manpage -a -p /usr/local/man/man8 && mandb`	Generate all SELinux policy manpages
`seinfo`	Query the components of a SELinux policy

`/var/log/audit/audit.log`	Logfile containing AVC denials, if `auditd` is running
`/var/log/messages`	Logfile containing AVC denials, if `rsyslogd` is running. AVC denials can also be seen via `dmesg`

`sealert -a` *`logfile`*	Analyze a SELinux logfile and display verbosely SELinux policy violations. SELinux violation events are logged as `type=AVC msg=audit(`*`timest.amp:id`*`): avc:  denied` (...)
`grep` *`timest.amp:id logfile`* `| audit2why`	Diagnostic a specific AVC denial event entry (identified by a *timestamp* and an *id*) from a SELinux *logfile*
`audit2why -d`	Read AVC violations from the output of `dmesg`
`ausearch -a` *`id`*	Query the SELinux log for event *id*
`audit2allow -i` *`inputfile`* `-M` *`module`*	Generate a loadable *module* containing the appropriate SELinux policy from a denied operation stored in *inputfile*
`ausearch -c '`*`(exe)`*`' --raw | audit2allow -M` *`module`*	Generate a loadable module to allow access on an executable which caused an AVC violation

`semodule -l`	List installed SELinux policy modules
`semodule -X` *`n`* `-i` *`module.pp`*	Install a SELinux policy module at priority *n*. Installed modules are not removed after reboot. Module files have usually the suffix `.pp` (policy package)
`semodule -X` *`n`* `-r` *`module`*	Remove a SELinux policy module at priority *n*. Modules must be removed at the same priority at which they were installed

Kickstart is a method to perform automatic installation and configuration of RHEL machines.
This can be done by specifying `inst.ks=hd:/dev/sda:/root/path/ksfile` either as a boot option, or an option to the kernel command in GRUB 2.

`/root/anaconda-ks.cfg`	Kickstart file describing the current system. This file is automatically generated during the installation
`system-config-kickstart`	GUI tool to create a Kickstart file
`ksvalidator` *`ksfile`*	Check the validity of a Kickstart file
`ksverdiff -f RHEL6 -t RHEL7`	Show the differences in the Kickstart syntax between RHEL 6 and RHEL 7

Red Hat Satellite 6

Red Hat **Satellite** is a system management software that allows provisioning and configuration of RHEL machines.
Repository content is provided via Red Hat Subscription Management (RHSM).
Satellite 5 was based on Spacewalk, an open source system management software for Linux machines. Satellite 6 is a complete overhaul of it and is composed of:
- **Foreman**, an open source lifecycle management tool able to provision servers via Kickstart and Puppet;
- **Katello**, a tool that handles Red Hat repository management (via the **Pulp** service) and subscription management (via the **Candlepin** service).

All these components above need a PostgreSQL database, except Pulp which needs a MongoDB database.
As a separate component, **Capsule** servers act as a proxy for many of the main Satellite functions e.g. repository storage.
A Capsule is also integrated in each Satellite server.

`subscription-manager register`	Register a system to the RHSM portal
`subscription-manager attach`	Attach a RHSM subscription to a registered system
`foreman-maintain service list`	List all Satellite services

`foreman-maintain service status`
`foreman-maintain service start`
`foreman-maintain service stop`
`foreman-maintain service restart`
	Display status or start, stop, restart all Satellite services.
	Performed via `systemctl`

`foreman-maintain backup`	Make a backup of Satellite
`foreman-rake` *`command:option`*	Perform various administrative tasks
`hammer`	CLI tool for Foreman
`pulp-admin-client`	Tool to administer the Pulp server
`virt-who`	Agent for reporting virtual guest IDs and hypervisors to a Satellite server
`foreman-debug`	Collect Satellite configuration, log, and backend data for debug purposes
`sosreport`	Collect diagnostic and configuration data for technical support
`citellus.py` *`sosreportfile`*	Perform some automated checks for troubleshooting a system

KVM (Kernel-based Virtual Machine) is a virtualization infrastructure for the Linux kernel that allows it to function as a hypervisor.

`/etc/libvirt/qemu/`	Directory containing the XML files that define VMs properties. `libvirtd` must be restarted after modifying an XML file
`/var/lib/libvirt/`	Directory containing files related to the VMs

`virt-manager`	KVM GUI

`virt-install --prompt`	Interactive command-line program to create a VM

`virt-install -n` *vmname* `-r 2048 \`
`--disk path=/var/lib/libvirt/images/vmname.img \`
`-l /root/vmstuff/inst/ \`
`-x "ks=/root/vmstuff/kickstart.cfg"`
	Create a VM with 2 Gb of RAM, specifying path of virtual disk, location of installation files, and (as extra argument) the Kickstart configuration to use

`virt-clone --prompt`	Interactive command-line program to clone a VM. A VM must be shut off or paused before it can be cloned
`virt-clone -o` *vmname* `-n` *vmclonename*	Clone a VM

`virsh`	Interface for VM management
`virsh list --all`	List all VMs present on the system
`virsh start` *vmname*	Start a VM
`virsh destroy` *vmname*	Brutally shut down a VM
`virsh shutdown` *vmname*	Gracefully shut down a VM
`virsh autostart` *vmname*	Set a VM to be automatically started when the system boots. Done by symlinking the VM to `/etc/libvirt/qemu/autostart/`
`virsh autostart --disable` *vmname*	Disable the autostart of a VM at system boot
`virsh edit` *vmname*	Edit the XML file defining a VM's properties

`virt-what`	Detect whether the current machine is a VM

Git is an open source version control system with a small footprint and very high performances. A Git directory is a complete repository with full history and version tracking abilities, independent of any remote repository.
Git commits are identified by a 40-hex-digits hash number, usually shortened to 7 digits, or even less if unambiguous.

git init	Initialize the current directory as a repository
git clone *repo*	Clone a remote repository. *repo* can be an URL (SSH, HTTP, HTTPS, FTP, FTPS, Git) or a local path e.g. ssh://user@example.com:8888/path/to/repo.git git://example.com:9999/path/to/repo.git /path/to/repo.git
git checkout *branch*	Start working into an already existing *branch*
git checkout -B *branch*	Create *branch* and start working into it
git checkout -- *file*	Discard local changes done to *file*
git checkout *branch file*	Copy *file* from *branch* to the current branch, and add it to the staging area
git pull	Pull the changes from the remote repository branch to the local branch
git add *file*	Add *file* to the staging area (i.e. content staged for the next commit), hence starting to track it
git add .	Add all modified files to the staging area
git rm *file*	Remove *file* from the content staged for the next commit
git status	See the status (e.g. files changed but not yet staged) of the current branch
git commit -m "*Message*"	Commit all staged files in the current branch
git commit -am "*Message*"	Add all changed files to the staging area in the current branch, and commit them
git merge *branch*	Merge changes made on *branch* to the master branch
git push	Push the local commits from the current branch to the remote repository
git push origin *branch*	Push the local commits from *branch* to the remote repository
git revert *commit*	Revert a specific commit
git branch	Show local branches
git branch -r	Show remote branches
git branch -a	Show remote and local branches
git branch -a --contains *commit*	Show on which branch was done a specific commit number
git branch -d *branch*	Delete a local branch (which must have been merged in its upstream branch)
git branch -D *branch*	Delete a local branch (irrespective of its merged status)

Git search and configuration

git diff *commit1 commit2*	Show the differences between two commits
git diff *branch1 branch2*	Show the differences between two branches
git diff *branch1 branch2 file*	Show the differences between two branches for a specific file
git log --all -- *file*	Show the commits which involved *file*, across all branches
git log -p --all -S '*string*' git log -p --all -G '*regex*'	Show the commits whose added or deleted lines contain a specific word
git grep *string* `git show-ref --heads`	Search for *string* across all branches' heads (i.e. in the latest content only, and not in all the previous commits)
git config --list	Get all currently set options and their values in the Git configuration
git config *option*	Get the value of *option*
git config user.name *name*	Set your username
git config user.email *email*	Set your email address

Vagrant is an open source software that allows building and maintaining lightweight and portable virtual environments for software development. It relies on an underlying virtualization solution e.g. VirtualBox.

`vagrant -h`	Print the list of commands recognized by Vagrant
`vagrant` *`command`* `-h`	Print help about the Vagrant *command*
`vagrant init hashicorp/precise64`	Initialize the current directory as a specific Vagrant environment (in this case, Ubuntu 12.04 64-bit) by creating a Vagrantfile on it
`vagrant up` *`vmname`*	Start a guest virtual machine and do a first provisioning according to the Vagrantfile
`vagrant provision` *`vmname`*	Provision a virtual machine
`vagrant ssh` *`vmname`*	Connect via SSH to a virtual machine
`vagrant halt` *`vmname`*	Shut down the virtual machine
`vagrant destroy` *`vmname`*	Delete the virtual machine and free any resource allocated to it
`vagrant status`	Print the status of the virtual machines currently managed by Vagrant
`vagrant global-status`	Print the status of all Vagrant environments on the system, by reading cached data. Completes quickly but results may be outdated
`vagrant global-status --prune`	Print the status of all Vagrant environments on the system, after rebuilding the environment information cache. Results are always correct but completion takes longer

The directory containing the Vagrantfile on the host can be accessed on the guest via `/vagrant`.

Puppet is a software configuration management tool. It is based on a client-server architecture, where a **Puppet agent** (client, running as `root` on each managed node) periodically gathers information (**facts**) about the local node state via the **Facter** tool, then communicates this information to the **Puppet master** (server, running as the `puppet` user and listening on TCP port 8140). The Puppet master then sends back to the Puppet agent a **catalog** containing the desired configuration for that node. The Puppet agent applies the needed changes so that the node's configuration converges with the desired configuration, and sends back a report to the Puppet master. Puppet changes are idempotent.

Puppet configurations are based on **resources** (e.g. "package", "service", "file", "user" ...). For each resource, a list of **attributes** is specified, with the desired value for each attribute.
Each resource type is implemented through **providers** (e.g. `yum`, `rpm`, `apt`, `opkg` ... for the resource "package").
Resources managed together as a single unit can be grouped into **classes**; classes are contained in **manifests** which are files with the `.pp` extension.
Modules are directories containing self-contained pieces of configuration and classes for a specific complex setting, e.g. an Apache webserver or a MySQL server.

`/etc/puppet/puppet.conf`	Configuration file (Open Source Puppet)
`/etc/puppetlabs/puppet/puppet.conf`	Configuration file (Puppet Enterprise)
`facter`	Gather the facts about the managed node, returning a list of key-value pairs
`puppet agent`	Main Puppet client. Retrieves the node's desired configuration from the Puppet master and applies it
`puppet agent --enable`	Enable the Puppet agent on the node
`puppet agent --disable "`*`Reason for disabling`*`"`	Disable the Puppet agent on the node
`cat $(puppet config print vardir)/state/agent_disabled.lock`	Print the reason why the Puppet agent is currently disabled. If the Puppet agent is enabled instead, the lockfile does not exist
`puppet agent --noop`	Perform a dry run, displaying the changes that Puppet would have applied without actually applying them

`puppet --version`
`puppet agent --version`
`puppet master --version`
	Show version of different Puppet components

`puppet module list`	List all modules installed in Puppet
`puppet resource user` *`username`*	Inspect the state of the resource "user" with respect to *username*
`puppet resource service httpd enable=false`	Modify the state of the resource "service" (in this case, disable the HTTP server)
`puppet describe user`	Show information about the resource "user"
`puppet describe --list`	List all resource types
`puppet describe user --providers`	Return the list of providers for the resource "user"
`puppet apply` *`modulename`*`/init.pp`	Apply a manifest one time only
`puppet cert` *`operation`*	Manage the SSL certificates used for communications between master and agents

Ansible is an open source tool for configuration management and software provisioning. It is agentless and connects to the managed machines via SSH pubkey authentication. It only requires OpenSSH and Python to be installed on the managed nodes.
The configuration for managed nodes is specified in one or more **playbook**, written in YAML and containing a number of **tasks**. When a playbook is run, first it collects system and environment information (**facts**) which is then stored in multiple variables named `ansible_`*`varname`*.

`/etc/ansible/hosts`	Inventory file, containing the list of hosts managed by Ansible. Can be in INI or YAML format

`ansible` *`hosts`* `-m` *`module options`*	Apply the *options* concerning *module* to the specified *hosts*
`ansible-playbook` *`options playbook`*`.yml`	Apply the specified playbook

HTML 4.01 components

Tag		Attributes	
<h1> **...** <h6> **Heading**		align=left|center|right|justify	Heading alignment †
  Line break	Line break and carriage return		
<hr> **Horizontal line**		align=left|center|right	Line alignment †
		noshade	Solid rendering instead of 3D †
		size=*npixels*	Line height
		width=*npixels*|*percent*%	Line width
<p> **Paragraph** <div> **Section**		align=left|center|right|justify	Paragraph or section alignment †
<span> **Group**	Group of elements		
<a> **Anchor**	Hyperlink	charset=*encoding*	Character encoding of target URL
		coords=*left,top,right,bottom*|*cx,cy,radius*|*x1,y1,...,xn,yn*	Coordinates of region; depends on shape
		href=*url*	Target URL for the link
		hreflang=*language*	Language of document at the target URL
		name=*section*	Name of anchor for document bookmarking
		rel|rev=alternate|stylesheet|start|next|prev|contents|index|glossary|copyright|chapter|section|subsection|appendix|help|bookmark	Relationship between this document and the target URL (rel) or vice versa (rev)
		shape=rectangle|circle|polygon	Shape of region
		target=_blank|_parent|_self|_top	Destination of target URL
		type=*mimetype*	MIME type of target URL
<dl> **Definition list**			
<dt> **Definition term**			
<dd> **Definition description**	Description of a definition term		
<ol> **Ordered list**		compact=compact	List must be more compact †
		start=*firstnumber*	Number to start the list on †
		type=A|a|I|i|1	List numbers type †
<ul> **Unordered list**		compact=compact	List must be more compact †
		type=disc|square|circle	List type †
<li> **List item**		type=disc|square|circle|A|a|I|i|1	List item type †
		value=*itemno*	List item value †

† = deprecated

Tag		Attributes	
<i> **Italic**			
<b> **Bold**			
<s> <strike> **Strike-through**	Strike-through text †		
<u> **Underlined**	Underlined text †		
<big> **Bigger**			
<small> **Smaller**			
<sub> **Subscript**			
<sup> **Superscript**			
<tt> **Teletype**	Monospaced text		
<em> **Emphasized**			
<strong> **Strong**			
<del> **Deleted** <ins> **Inserted**	Deleted/inserted text	cite=*url*	URL to document explaining deletion/insertion
		datetime=*yyyy-mm-dd*	When the text was deleted/inserted
<pre> **Preformatted**		width=*ncharacters*	Max number of characters per line †
<code> **Code**	Source code text		
<samp> **Sample**	Sample code text		
<kbd> **Keyboard**	Keyboard key		
<var> **Variable**	Variable name		
<cite> **Citation**	Citation block		
<blockquote> **Quotation** <q> **Short quotation**		cite=*url*	URL to document containing the quote
<address> **Address**	Address block		
<abbr> **Abbreviation**			
<acronym> **Acronym**			
<dfn> **Definition**	Definition term		
<font> **Font**	Font †	color=rgb(*r*,*g*,*b*)|#*rrggbb*|*color*	Text color
		face=*fontname*	Text font
		size=*[1 ... 7]*|*[-6 ... +6]*	Text size
<bdo> **Bidirectional override**		dir=ltr|rtl	Direction of text: left-to-right or right-to-left
<xmp> **XMP**	Non-formatted text † (ignores other HTML tags)		
other tags	Attributes common to almost all other tags	class=*class*|*style*	Class of the element
		id=*id*	Unique ID of the element
		style=*styledef*	Inline style definition
		title=*tooltip*	Text of the tooltip to display
		dir=ltr|rtl	Direction of text: left-to-right or right-to-left
		lang=*language*	Language of the content
		accesskey=*character*	Keyboard shortcut for the element
		tabindex=*ntab*	N of tab for the element

† = deprecated

Tag	Attributes	
<img> **Image**	align=top|bottom|left|middle|right	Image alignment with respect to surrounding text †
	alt=*alternatetext*	Description of the image for text-only browsers
	border=*npixels*	Border width around the image †
	height=*npixels*|*percent*%	Image height
	hspace=*npixels*	Blank space on the left and right side of image †
	ismap=*url*	URL for server-side image map
	longdesc=*url*	URL containing a long description of the image
	src=*url*	URL of the image
	usemap=*url*	URL for client-side image map
	vspace=*npixels*	Blank space on top and bottom of image †
	width=*npixels*|*percent*%	Image width
<map> **Image map**	id=*id*	Unique ID for the map tag
	name=*name*	Unique name for the map tag
<area> **Area of image map**	alt=*alternatetext*	Description of area for text-only browsers
	coords=*left*,*top*,*right*,*bottom*| *cx*,*cy*,*radius*|*x1*,*y1*,...,*xn*,*yn*	Coordinates of clickable area; depends on shape
	href=*url*	Target URL of area
	nohref=true|false	Excludes or includes the area from image map
	shape=rectangle|circle|polygon	Shape of area
	target=_blank|_parent|_self|_top	Destination of target URL

† = deprecated

HTML 4.01 tables

Tag	Attributes	
`<table>` **Table**	align=left|center|right	Table alignment †
	bgcolor=rgb(*r*,*g*,*b*)|#*rrggbb*|*color*	Table background color †
	border=*npixels*	Border width
	cellpadding=*npixels*|*percent*%	Space around the content of each cell
	cellspacing=*npixels*|*percent*%	Space between cells
	frame=void|above|below|lhs|rhs|hsides|vsides|box|border	Visibility of sides of the table border
	rules=none|groups|rows|cols|all	Horizontal or vertical divider lines
	summary=*summary*	Summary of the table for text-only browsers
	width=*npixels*|*percent*%	Table width
`<tr>` **Table row**	align=left|center|right|justify|char	Horizontal text alignment
	bgcolor=rgb(*r*,*g*,*b*)|#*rrggbb*|*color*	Row background color †
	char=*character*	Character to align text on, if `align=char`
	charoff=*npixels*|*percent*%	Alignment offset to first character, if `align=char`
	valign=top|middle|bottom|baseline	Vertical text alignment

`<td>`
Table cell
`<th>`
Table header
	abbr=*content*	Abbreviated content in a cell

	align=left|center|right|justify|char	Horizontal text alignment
	axis=*category*	Cell name
	bgcolor=rgb(*r*,*g*,*b*)|#*rrggbb*|*color*	Cell background color †
	char=*character*	Character to align text on, if `align=char`
	charoff=*npixels*|*percent*%	Alignment offset to first character, if `align=char`
	colspan=*ncolumns*	Number of columns this cell spans on
	headers=*headerid*	Cell header information for text-only browsers
	height=*npixels*	Cell height †
	nowrap	Text in cell stays on a single line †
	rowspan=*nrows*	Number of rows this cell spans on
	scope=col|colgroup|row|rowgroup	Target for cell header information
	valign=top|middle|bottom|baseline	Vertical text alignment
	width=*npixels*|*percent*%	Cell width †

† = deprecated

7-bit ASCII table

Dec	Hex	Char		Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	
0	0	NUL	Null	32	20	space	64	40	@	96	60	`	
1	1	SOH	Start of heading	33	21	!	65	41	A	97	61	a	
2	2	STX	Start of text	34	22	"	66	42	B	98	62	b	
3	3	ETX	End of text	35	23	#	67	43	C	99	63	c	
4	4	EOT	End of transmission	36	24	$	68	44	D	100	64	d	
5	5	ENQ	Enquiry	37	25	%	69	45	E	101	65	e	
6	6	ACK	Acknowledge	38	26	&	70	46	F	102	66	f	
7	7	BEL	Bell	39	27	'	71	47	G	103	67	g	
8	8	BS	Backspace	40	28	(	72	48	H	104	68	h	
9	9	TAB	Horizontal tab	41	29	)	73	49	I	105	69	i	
10	A	LF	Line feed	42	2A	*	74	4A	J	106	6A	j	
11	B	VT	Vertical tab	43	2B	+	75	4B	K	107	6B	k	
12	C	FF	Form feed	44	2C	,	76	4C	L	108	6C	l	
13	D	CR	Carriage return	45	2D	-	77	4D	M	109	6D	m	
14	E	SO	Shift out	46	2E	.	78	4E	N	110	6E	n	
15	F	SI	Shift in	47	2F	/	79	4F	O	111	6F	o	
16	10	DLE	Data link escape	48	30	0	80	50	P	112	70	p	
17	11	DC1	Device control 1	49	31	1	81	51	Q	113	71	q	
18	12	DC2	Device control 2	50	32	2	82	52	R	114	72	r	
19	13	DC3	Device control 3	51	33	3	83	53	S	115	73	s	
20	14	DC4	Device control 4	52	34	4	84	54	T	116	74	t	
21	15	NAK	Negative ACK	53	35	5	85	55	U	117	75	u	
22	16	SYN	Synchronous idle	54	36	6	86	56	V	118	76	v	
23	17	ETB	End of Tx block	55	37	7	87	57	W	119	77	w	
24	18	CAN	Cancel	56	38	8	88	58	X	120	78	x	
25	19	EM	End of medium	57	39	9	89	59	Y	121	79	y	
26	1A	SUB	Substitute	58	3A	:	90	5A	Z	122	7A	z	
27	1B	ESC	Escape	59	3B	;	91	5B	[	123	7B	{	
28	1C	FS	File separator	60	3C	<	92	5C	\	124	7C	|	
29	1D	GS	Group separator	61	3D	=	93	5D	]	125	7D	}	
30	1E	RS	Record separator	62	3E	>	94	5E	^	126	7E	~	
31	1F	US	Unit separator	63	3F	?	95	5F	_	127	7F	DEL	Delete

Characters 0-31 and 127 are non-printable.

`ascii`
`man ascii`
	Display an ASCII table

`showkey -a`	Prompt for pressing a key and display its ASCII value in decimal, octal, and hex