Linux
Quick Reference Guide

6th edition   August 2018

Foreword

This guide stems from the notes I have been taking both while working as a Linux sysadmin and while preparing 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). It contains a good amount of topics for these certification exams, with some subjects handled in more details than others, plus other useful information about standards and tools for Linux system administration. Unless otherwise specified, shell commands and operations refer to Bash.

This is an independent publication and is not affiliated with, authorized by, sponsored by, or otherwise approved by LPI or Red Hat. You can freely use and share this whole guide or the single pages, both in electronic or in printed form, provided that you distribute them unmodified and not for profit.

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

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
- 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 physical or virtual disk to the machine	
2.	`lsblk`	Check that the new disk is being 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 LVM 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 which is now ready to be used

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

1.	Add a new physical or virtual disk to the machine; this will provide the extra disk space	
2.	`fdisk /dev/sdc`	Partition the new disk
3.	`pvcreate /dev/sdc`	Initialize the Physical Volume
4.	`vgextend myvg0 /dev/sdc`	Add the Physical Volume to an existing Volume Group
5.	`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
6.	`resize2fs /dev/myvg0/mylv`	Extend the filesystem

How to reduce the size of a Logical Volume (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 snapshot0 /dev/myvg0/mylv`	Create the snapshot like a Logical Volume
2.	`tar cvzf snapshot0.tar.gz snapshot0`	Backup the snapshot with your preferred backup tool
3.	`lvremove /dev/mvvg0/snapshot0`	Delete the snapshot

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`	Resize a disk or partition in use with LVM			`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	
`lvmdiskscan`	Scan the system for disks and partitions usable by LVM
`dmsetup command`	Perform low-level LVM operations

`/dev/mapper/vgname-lvname` `/dev/vgname/lvname`	Mapping of Logical Volumes in the filesystem

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.

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/rcN.d/`	`/etc/rc.d/rcN.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.

Service operation parameters supported by the init scripts		
`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
`rm /etc/machine-id && \` `systemd-machine-id-setup` (Red Hat)	Initialize the machine ID
`/etc/securetty`	List of TTYs from which the root user is allowed to login

`/etc/issue`

Message that will be printed before the login prompt.
Can contain the following 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`	Architecture identifier of machine	`\U`	"*n* users" logged in
`\n`	Nodename aka hostname	`\v`	OS version and build date

`/etc/issue.net`	Message that will be printed before the login prompt on a remote session
`/etc/motd`	Message that will be printed after a successful login, before execution of the login shell
`/etc/nologin`	If this file exists, `login` and `sshd` deny login to the system. Useful to prevent users to log in when doing system maintenance

To prevent a user to log in, their shell can be set either as:
- `/bin/false` (user will be forced to exit immediately)
- `/sbin/nologin` (user will be prompted a message, then forced to exit; message is "This account is currently not available" or, if the file `/etc/nologin.txt` exists, the contents of that file)

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

Runlevel (SysV)	Target (Systemd)	Debian	Red Hat
0		Shutdown	
1		Single user / maintenance mode	
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)	

(The left side of rows 2–5 is labelled "default runlevels".)

Systemd's target `runleveln.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 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

`init 0`
`telinit 0`
`shutdown -h now`
`halt`
`poweroff`
	Halt the system

`init 6`
`telinit 6`
`shutdown -r now`
`reboot`
	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

System V	Systemd	Action

`/etc/init.d/`*`service operation`*
 	
`service `*`service operation`*   (Red Hat)
 	
`rc`*`service operation`*   (SUSE)
	`systemctl `*`operation service`*	Perform one of these operations on the specified service:
		`start`
		`stop`
		`restart`
		`status`
		`force-reload`
		`condrestart`
		`try-restart`
		`reload`

`update-rc.d `*`service`*` defaults` (Debian) `chkconfig --add `*`service`* (Red Hat)		Add a service at boot
`update-rc.d -f `*`service`*` remove` (Debian) `chkconfig --del `*`service`* (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; create 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`	`systemctl enable `*`service`*	Add the service on default runlevels
`chkconfig `*`service`*` off`	`systemctl disable `*`service`*	Remove the service on default runlevels
`chkconfig `*`service`*	`systemctl is-enabled `*`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`	`systemctl list-unit-files \` `--type=service`	List all active services and their current configuration
`ls /etc/rc`*`n`*`.d` (Debian)		List services started on runlevel *n*
	`systemctl`	List loaded and active units
	`systemctl --all`	List all units, including inactive ones
	`systemctl -t target`	List targets

```
                                    /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	

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 done

Filesystem Hierarchy Standard (FHS)	
`/bin`	Essential command binaries
`/boot`	Bootloader files (e.g. OS loader, kernel image, initrd)
`/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
`/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 actual 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.

`/dev/hda`	IDE hard drive
`/dev/sda`	SCSI, PATA, or SATA hard drive
`/dev/vda`	Virtual disk for KVM-based virtual machines
`/dev/hda, /dev/hdb, /dev/hdc ...`	First, second, third ... hard drive
`/dev/sda1, /dev/sda2, /dev/sda3 ...`	First, second, third ... partition of the first hard drive

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).
Most modern filesystems use journaling; in a journaling filesystem, the journal logs changes before committing them to the filesystem, which ensures faster recovery and less 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   Partition numbers: 1-4
Max 11 logical partitions (inside the extended partition) per hard disk   Partition numbers: 5-15
Max disk size is 2 Tb

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

`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`	This command can be run after fdisk operations to notify the OS of 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

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, given its name

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 < 32M)
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 <32M	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` provides the above list of partition IDs and names.

Most used Linux-supported filesystems

ext2	Linux default filesystem, offering the best performances
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

In Linux, the swap space is a virtual memory area (a file or a partition) used as RAM extension. Usually a partition is preferred because of better performances concerning fragmentation and disk speed. Although listed as filesystem type 0x82, the swap partition is not a filesystem but a raw addressable memory with no structure; therefore it is not shown in the output of `mount` or `df` commands.
The `fdisk` tool can be used to create a swap partition.

`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    Filesystems information
# <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
```

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: `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 (i.e. access time) information for the filesystem. This can improve performances because the system does not need anymore to do filesystem writes for files which are just being read
	`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 size of each file inside *directory*
`du -s` *directory*	Report the sum of all files contained inside *directory*
`du -sh` *directory*	Report the sum of all files contained inside *directory* in human-readable output
`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
`blockdev --getbsz /dev/sda1`	Get the block size of the specified partition
`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 /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 (which hence is defined as the filesystem root)
`mknod /dev/sda`	Create a directory allocating the proper inode. Useful during system recovery when experiencing filesystem problems
`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. Potentially dangerous!
`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 (e.g. number of mounts, last checks, UUID)
`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. For this, there must be at least one spare new disk partition available. 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 lowest level of 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. Enable 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 permits 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 subdir. ```# mount point map options` `/net -hosts` `/- /etc/auto.direct` `/misc /etc/auto.misc` `/home /etc/auto.home --timeout=60`  The `-hosts` map tells AutoFS to mount/unmount automatically any export from the NFS server *nfsserver* when the directory `/net/`*nfsserver*`/` is accessed.

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 subdir 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

Non-GRUB bootloaders		
LILO (Linux Loader)		Obsolete. Small bootloader that can be placed in the MBR or the boot sector of a partition. The configuration file is `/etc/lilo.conf` (run `/sbin/lilo` afterwards to validate changes).
SYSLINUX	SYSLINUX	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.

	ISOLINUX	Able to boot from CD-ROM ISO 9660 filesystems.
		Used for Live CDs and bootable install CDs.
		
		The CD must contain the following files:
		
		`isolinux/isolinux.bin` ISOLINUX image, from the SYSLINUX distro
		
		`boot/isolinux/isolinux.cfg` ISOLINUX configuration
		
		`images/` Floppy images to boot
		
		`kernel/memdisk`
		
		The CD can be burnt with the command:
		`mkisofs -o output.iso -b isolinux/isolinux.bin -c isolinux/boot.cat \`
		`-no-emul-boot -boot-load-size 4 -boot-info-table CDrootdir`

	PXELINUX	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:
		
		`/tftpboot/pxelinux.0` PXELINUX image, from the SYSLINUX distro
		
		`/tftpboot/pxelinux.cfg/` 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:
		`01-88-99-aa-bb-cc-dd`
		`C000025B`
		`C000025`
		`C00002`
		`C0000`
		`C000`
		`C00`
		`C0`
		`C`
		`default`

	EXTLINUX	General-purpose bootloader like LILO or GRUB. Now merged with SYSLINUX.

GRUB (Grand Unified Bootloader) is the standard boot manager on modern Linux distros. 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, 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.

	`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
Common kernel parameters:	`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 choosing 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 it)
P	Bring up the GRUB password prompt (necessary if a GRUB password has been set)

`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
```

Low-level package managers	Debian	Red Hat

Install a package file	`dpkg -i package.deb`	`rpm -i package.rpm`
		`rpm -i ftp://host/package.rpm`
		`rpm -i http://host/package.rpm`

Remove a package	`dpkg -r package`	`rpm -e package`
Upgrade a package (and remove old versions)		`rpm -U package.rpm`
Upgrade a package (only if an old version is already installed)		`rpm -F package.rpm`
List installed packages and their state	`dpkg -l`	`rpm -qa`
List installed packages and their installation date, from newest to oldest		`rpm -qa --last`
List the content of an installed package	`dpkg -L package`	`rpm -ql package`
List the content of a package file	`dpkg -c package.deb`	`rpm -qpl package.rpm`
Show the package containing a specific file	`dpkg -S file`	`rpm -qf file`
Verify an installed package		`rpm -V package`
Reconfigure a package	`dpkg-reconfigure package`	
Install a package source file		`rpm -i package.src.rpm`
Compile a package source file		`rpm -ba package.spec`

High-level package managers	Debian	Red Hat
Install a package	`apt-get install package`	`yum install package`
Install a package file		`yum install package.rpm` `yum localinstall package.rpm`
Remove a package	`apt-get remove package`	`yum remove package`
Upgrade an installed package		`yum update package`
Upgrade all installed packages	`apt-get upgrade`	`yum update`
Upgrade all installed packages and handle dependencies with new versions	`apt-get dist-upgrade`	
Replace a package with another		`yum swap packageout packagein`
Get the source code for a package	`apt-get source package`	
Check for broken dependencies and update package cache	`apt-get check`	
Fix broken dependencies	`apt-get install -f`	
Update information on available packages	`apt-get update`	
List all installed and available packages		`yum list`
List installed and available packages that match the search term		`yum list searchterm`
List installed packages		`yum list installed`
List packages available for install		`yum list available`
Search for a package	`apt-cache search package`	
Search for packages that match the search term in the package name or summary		`yum search searchterm`
Search for packages that match the search term in the package name, summary, or description		`yum search all searchterm`
Show package dependencies	`apt-cache depends package`	`yum deplist package`
Show package records	`apt-cache show package`	`yum list package`
Show information about a package	`apt-cache showpkg package`	`yum info package`
Show the installation history (installs, updates, etc.)		`yum history` `yum history list`
Show the installation history about a package		`yum history package package` `yum history list package package`
Update information about package contents	`apt-file update`	
List the content of an uninstalled package	`apt-file list package`	
Show which package provides a specific file	`apt-file search file`	`yum whatprovides file`
Add a CD-ROM to the sources list	`apt-cdrom add`	
Download package and all its dependencies		`yumdownloader --resolve package`
Show URLs that would be downloaded		`yumdownloader --urls package`
Try to complete unfinished or aborted package installations		`yum-complete-transaction`
Execute the command but only considering a specific repository		`yum command --disablerepo="*"` `--enablerepo="repository"`
Print list of available repositories	`cat /etc/apt/sources.list`	`yum repolist` `cat /etc/yum.repos.d/*.repo`
Package format	compressed with `ar`	compressed with `cpio`

High-level package managers are able to install remote packages and automatically solve dependencies.

GUI package managers	Debian	Red Hat
Manage packages and dependencies using a graphical or text-based UI	`aptitude`	`pirut`
	`dselect`	
	`synaptic`	

Package management utilities	Debian	Red Hat
Convert a RPM package to DEB and install it. May break the package system!	`alien -i `*`package`*`.rpm`	
Convert a RPM package to a cpio archive		`rpm2cpio `*`package`*`.rpm`
Add a key to the list of keys used to authenticate packages	`apt-key add `*`keyfile`*	
Create an XML file of repository metadata from the set of RPMs contained in *directory*		`createrepo `*`directory`*
Show a tree with all dependencies of *package*		`repoquery --tree-requires `*`package`*
Register a system to the RHSM (Red Hat Subscription Management) portal		`subscription-manager register`
Attach a RHSM subscription to a registered system		`subscription-manager attach`

`/etc/yum.repos.d/foobar.repo` Configuration file for a "foobar" repository (Red Hat)	
`[foobar]`	Repository ID
`name=Foobar $releasever - $basearch`	Repository name

`baseurl=http://download.foobarproject.org/pub/linux/\`
`releases/$releasever/Everything/$basearch/os/`
`        http://foo.org/linux/$releasever/$basearch/`
`        http://bar.org/linux/$releasever/$basearch/`
	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're listed. By default, if more than one baseurl is specified, yum chooses one randomly
`metalink=https://mirrors.foobarproject.org/metalink?repo=\` `foobar-$releasever&arch=$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-foobar-\` `$releasever-$basearch`	ASCII-armored GPG public key file of the repository

The manpage `man yum.conf` lists all repository configuration options.

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

It is recommended not to use `dd` on a mounted block device because of write cache issues.

`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. For all transfers subsequent to the first, rsync only copies the blocks that have changed, making it a very efficient backup solution in terms of speed and bandwidth
`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
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 a command
man 7 *command*	Show section 7 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 (usually only for root) 9 - Kernel routines
mandb	Generate or refresh the search database for manpage entries. This must be done after installing new packages, in order to obtain meaningful results from apropos or man -k
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 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*

`command <<< 'string'`	Create a Here String, passing *string* as input to *command*
`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
`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 (named `xaa`, `xab`, `xac`, and so on)
`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
`od file`	Dump a file into octal (or other formats)
`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 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 allow 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
`tidy`	Validate, 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

`strings `*`file`*	Show all printable character sequences at least 4-character long that are inside a file
`antiword `*`docfile`*	Show text and images from a MS Word document
`catdoc `*`docfile`*	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, and so on 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 (i.e. prepended with a backslash character `\`).

`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
`touch file`	Change access timestamp and modify timestamp of a file as now. If the file does not exist, it is created
`mktemp`	Create a temporary file or directory, using `tmp.XXXXXXXXXX` as filename template
`ls`	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
`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
`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 a period of time
`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 all processes using an unlinked file. These processes, until killed or restarted, hold the file open preventing it from being deleted (and freeing disk space)
`lslocks`	List information about all currently held file locks

`cd directory`	Change to the specified directory
`cd -`	Change to the previously used directory
`pwd`	Print the current working directory
`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
`pushd dir`	Add a directory 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
`dirs`	Display the directory stack (i.e. the list of remembered directories)
`dirname file`	Output the directory path in which the file is located, stripping any non-directory suffix from the filename

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

File-naming wildcards (globbing)	
`*`	Matches zero or more characters
`?`	Matches one character
`[kxw]`	Matches k, x, or w
`[!kxw]`	Matches any character except k, x, or w
`[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"

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`

`cat /etc/passwd | wc -l`   Pipe the stdout of command `cat` to the stdin of command `wc` (hence printing the number of accounts in the system). Piped commands run concurrently

`ls > `*`file`*
`ls 1> `*`file`*   Redirect the stdout of command `ls` to *file* (hence writing on *file* the content of the current directory). This overwrites *file* if it already exists, unless the Bash noclobber option is set (via `set -o noclobber`). The redirection is handled by the shell, not by the command invoked

`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* (hence writing any error encountered by the command to *file*)

`ls 2>> `*`file`*   Append the stderr of command `ls` to *file*

`ls 2> /dev/null`   Silence any error coming from command `ls`

`mail user@foo.com < `*`file`*   Redirect *file* to the stdin of command `mail` (hence sending via e-mail the contents of *file* to the specified email address)

`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`
`ls &> `*`file`* †   Redirect both stdout and stderr of command `ls` to *file*
`ls >& `*`file`* †   † = non-POSIX standard and therefore not recommended

`> `*`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* (hence writing content of 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*

`ls foo* | xargs cat`   `xargs` calls the `cat` command multiple times for each argument found on stdin (hence printing the content of every file whose name starts by "foo")

```
while read -r line
do
   echo "Hello $line"
done < file
```
Process a text file line by line, reading from *file*.
If *file* is `/dev/stdin`, reads from standard input instead

```
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)

```
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, or script that runs on the system is a process. 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 info; 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` *process*	Show PID of *process*
`top`	Monitor processes in real-time
`htop`	Monitor processes in real-time (ncurses UI)
`ipcs`	Show IPC facilities information (shared memory, message queues, and semaphores)
`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"
`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
`pkill -9 -u root sshd`	Kill processes whose name is "sshd" and are owned by root
`xkill`	Interactive program to kill a process by its X GUI resource
`strace` *command*	Trace the execution of *command*, intercepting and printing the system calls called by a process and the signals received by a process
`jobs`	List all jobs (i.e. processes whose parent is a Bash shell)
CTRL Z	Suspend a job, putting it in the stopped state (send a SIGTSTP)
`bg %1`	Put job #1 in the background (send a SIGCONT)
`fg %1`	Resume job #1 in the foreground and make it the current job (send a SIGCONT)
`kill %1`	Kill job #1

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 a 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
`( `*command*` )& pid=$!; sleep `*n*`; kill -9 $pid`	Run a command and kill it after *n* seconds
`:(){ :|:& };:`	Fork bomb: starts a process that continually replicates itself, slowing down or crashing the system because of resource starvation

Most frequently used signals		
Signal number	**Signal name**	**Meaning**
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

`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 2 5`
`iostat 2 5`
`mpstat 2 5`
	Print the relevant report every 2 seconds, for 5 times

`iotop`	Display I/O usage by processes in the system
`atop`	Advanced system monitor that displays the load on CPU, RAM, disk, and network
`free`	Show the amount of free and used memory in the system
`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. 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 -n DEV`	Show reports about network activity (received and transmitted packets per second)
`sar -f /var/log/sa/s19 \` `-s 06:00:00 -e 06:30:00`	Show reports for system activity from 6 to 6:30 AM on the 19th of the month
`powertop`	Power consumption and power management diagnosis tool
`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

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

```
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)	
	free	Free memory (idle)	in Kb
	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)	
	sy	Time spent running system code (kernel)	
	id	Time spent idle	in percentage of total CPU time
	wa	Time spent waiting for I/O	
	st	Time stolen from a virtual machine	

```
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.

```
-  r w x  r w x  r w x
```

| **- = regular file**
d = directory
l = symbolic link
s = Unix domain socket
p = named pipe
c = character device file
b = block device file | **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 |
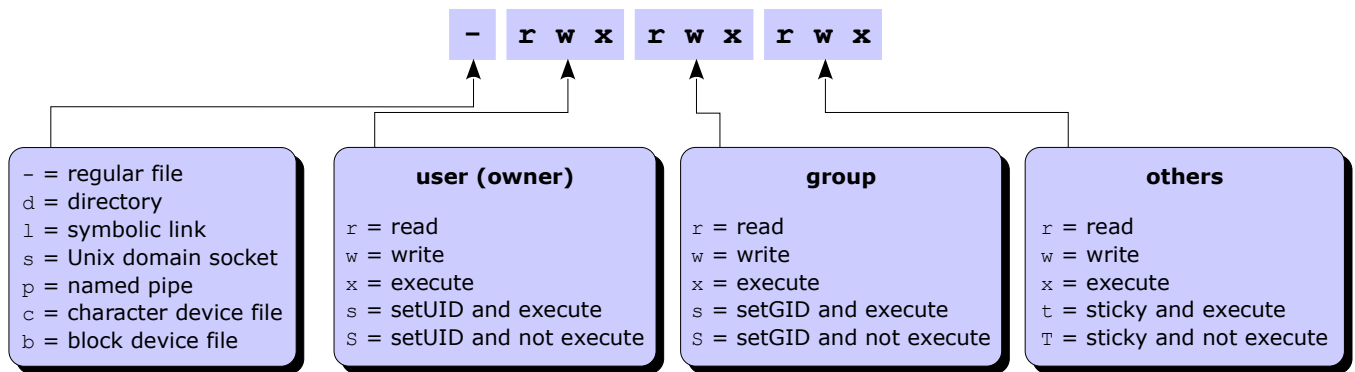
Permission	Octal value	Command	Effect on file	Effect on directory
Read	user: 400	`chmod u+r`	Can open and read the file	Can list directory content
	group: 40	`chmod g+r`		
	others: 4	`chmod o+r`		
Write	user: 200	`chmod u+w`	Can modify the file	Can create, delete, and rename files in the directory
	group: 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

`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

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 `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

Timestamp	Value tracked	Command to show
mtime	Time of last **modification** to file contents (data itself)	`ls -l`
ctime	Time of last **change** to file contents or 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`	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 is an IRIX-compatibility command
`man acl`	Show the manpage about ACLs

A Linux directory contains a list of structures which are associations between a filename and an inode.
An inode contains 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.
An inode does not contain the name of the file; this information is stored in the directory where the file is.

`ls -i`	Show a listing of the directory with the files' inode numbers
`df -i`	Report filesystem inode usage

	Hard link	**Soft or 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 references the inode the original file pointed to)	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 new inode number

`find / -name "foo*"` `find / -name "foo*" -print`	Find all files, starting from the root dir, whose name start with foo
`find / -name "foo*" -exec chmod 700 {} \;`	Find all files whose name start with "foo" and apply permission 700 to all of them
`find / -name "foo*" -ok chmod 700 {} \;`	Find all files whose name start with "foo" and apply permission 700 to all of them, asking for confirmation before each file
`find / -size +128M`	Find all files larger than 128 Mb
`find / -ctime +10`	Find all files created more than 10 days ago
`find / -perm -4000 -type f`	Find all files of type file (i.e. not directories) and with SUID set (a possible security risk, because a shell with SUID root is a backdoor)
`find / -perm -2000 -type f`	Find all files with SGID set
`find /home/jdoe/path -type f \` `-newermt "May 4 14:50" -delete`	Find and delete all files newer than the specified datetime. Using `-delete` is preferable to using `-exec rm {} \;`
`find . -type f -print -exec cat {} \;`	Print all files in the current directory with a filename header
`locate `*`command`* `slocate `*`command`*	Locate *command* 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 rebuilding 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*.

`set`	Display all variables
`env`	Display all environment variables
`export MYVAR`	Export a variable, making it an environment variable

`MYVAR=`*value*
`((MYVAR=`*value*`))`
`let "MYVAR=`*value*`"`
	Set a variable

`echo $MYVAR` `echo ${MYVAR}`	Use a variable (in this case, echo it to screen). If other characters follow the variable name, it is necessary to specify the boundaries of the variable name via `{}` to make it unambiguous
command `"$MYVAR"`	Pass a variable as argument to *command*. It is recommended to double quote a variable when referencing it, to prevent interpretation of special characters (except `\ $ ` `` ` ``), and avoid word splitting if the variable contains spaces

`MYVAR=$((2+2))`
`MYVAR=$[2+2]`
`FOO=$((BAR + 42))`
``FOO=`expr $BAR + 42` ``
	Evaluate a numeric expression and assign the result to another variable

``MYVAR=`date` `` `MYVAR=$(date)`	Assign to a variable the output resulting from a command

`for i in /`*path*`/*`
`do`
`   echo "Filename: $i"`
`done`
	Loop and operate through all the output tokens (in this case, files in the *path*).
	Note: looping over the output of `$(ls)` is unnecessary and harmful, as filenames containing whitespace or glob characters may have unintended results

`unset MYVAR`	Delete a variable
`set ${MYVAR:=`*value*`}` `MYVAR=${MYVAR:-`*value*`}`	Set a variable, only if it is not already set (i.e. does not exist) or is null
`echo ${MYVAR:-`*message*`}`	If variable exists and is not null, print its value, otherwise print *message*
`echo ${MYVAR:+`*message*`}`	If variable exists and is not null, print *message*, otherwise print nothing
`echo ${MYVAR,,}`	Print a string variable in lowercase

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

Bash shell options	
`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`	Enable printing of shell input lines as they are read
`set -x` `set -o xtrace`	Enable printing of command traces before execution of each command (debug mode)
`set -u` `set -o nounset`	Treat expansion of unset variables as an error
To run a script with a Bash option enabled, do one of the following: - Run the script with `bash -option scriptfile.sh` - Specify the shebang line as `#!/bin/bash -option` - Add the command `set -option` at the beginning of the script	

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`	

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

Script execution	
`source myscript.sh` `. myscript.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 myscript.sh` `./myscript.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, non-zero 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

`if [ $? -eq 0 ]`
`then`
   `echo "Success"`
`else`
   `echo "Failure"`
`fi`
	Evaluate whether the last executed command 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. An advantage of functions over aliases is that functions can be passed arguments
`myfunc arg1 arg2 ...`	Call a function
`typeset -f`	Show functions defined in the current Bash session

`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
`usleep 5000`	Pause for 5000 microseconds
`getopts`	Parse positional parameters in a shell script
`script`	Generate a typescript of a terminal session
`expect`	Dialogue with interactive programs according to a script, analyzing what can be expected from the interactive program and replying accordingly
`parallel command`	Run a command in parallel. This is used to operate on multiple inputs, similarly to `xargs`
`zenity`	Display GTK+ graphical dialogs for user messages and input

```
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		**File operators**		**Expression operators**	
`-eq`	Equal to	`-e` or `-a`	Exists	`-a`	Logical AND
`-ne`	Not equal to	`-d`	Is a directory	`-o`	Logical OR
`-lt`	Less than	`-b`	Is a block special file	`!`	Logical NOT
`-le`	Less than or equal to	`-c`	Is a character special file	`\( \)`	Priority
`-gt`	Greater than	`-f`	Is a regular file		
`-ge`	Greater than or equal to	`-r`	Is readable		
String operators		`-w`	Is writable		
`-z`	Is zero length	`-x`	Is executable		
`-n` or nothing	Is non-zero length	`-s`	Is non-zero length		
`=` or `==`	Is equal to	`-u`	Is SUID		
`!=`	Is not equal to	`-g`	Is SGID		
`<`	Is alphabetically before	`-k`	Is sticky		
`>`	Is alphabetically after	`-h`	Is a symbolic link		

Evaluation operators					
`=`	Equal to	`+`	Plus	`string : regex` `match string regex`	String matches regex
`!=`	Not equal to	`-`	Minus		
`<`	Less than	`\*`	Multiplied by	`substr string pos length`	Substring
`<=`	Less than or equal to	`/`	Divided by	`index string chars`	Index of any chars in string
`>`	Greater than	`%`	Remainder	`length string`	String length
`>=`	Greater than or equal to				

`expr "$MYVAR" = "39 + 3"` `MYVAR=$((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

Tests	
| ```
if [test 1]
then
    [command block 1]
elif [test 2]
then
    [command block 2]
else
    [command block 3]
fi
``` | ```
case $STRING in
    pattern1)
        command1
        command1bis
        ;;
    pattern2)
        command2
        ;;
    *)
        defaultcommand
        ;;
esac
``` |

Loops		
| ```
while [test]
do [command block]
done
``` | ```
until [test]
do [command block]
done
``` | ```
for I in [list]
do [command block]
done
``` |
| The *command block* executes as long as *test* is true | The *command block* executes as long as *test* is false | The *command block* executes for each I 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)
`emacs`	GNU Emacs, a GUI text editor
`gedit`	GUI text editor
`ed`	Line-oriented text editor
`more`	Text pager (obsolete)
`less`	Text pager

`less` **pager commands**	
`h`	Help
`g`	Go to the first line in the file
`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
`:n`	When reading multiple files, go to the next file
`:p`	When reading multiple files, go to the previous file
`q`	Quit
`less` **pager options**	
`--follow-name`	Attempts periodically to reopen the file by name. Useful to keep reading, via the `F` command, 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		
`I`	Insert text after line	and go to Insert mode	
`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 a 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

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 largest size of changes on status line
`ro`	Make file readonly
`scroll=12`	Set screen size as 12 lines
`sh=/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`	Warn "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*
Options can also be permanently set by including them in `~/.exrc` (Vi) or `~/.vimrc` (Vim)

`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)

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

```
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 = '00789';`	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
`DROP DATABASE CompanyDatabase;`	Delete a database
`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

`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 WHERE zipcode = '00123';`	Select first and last name of customers whose zip code is 00123
`SELECT firstname, lastname FROM customers WHERE zipcode IS NOT NULL;`	Select first and last name of customers with a recorded 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 but 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 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 = '00123'`
`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 = '00123'`
`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 = '00123'`
`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 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. Recommended
`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 -ppassword`	Login to MySQL as root with the specified *password*
`mysql -u root -p -h host -P port`	Login to the specified remote MySQL server 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 > alldbs.sql`	Backup all databases to a dump file
`mysqldump -u root -p MyDatabase > mydb.sql`	Backup a database to a dump file
`mysqldump -u root -p --databases MyDb1 MyDb2 > dbs.sql`	Backup several databases to a dump file
`mysqldump -u root -p MyDatabase t1 t2 > tables.sql`	Backup some tables of a database to a dump file
`mysql -u root -p < alldbsbak.sql`	Restore all databases from a dump file (which contains a complete dump of a MySQL server)
`mysql -u root -p MyDatabase < mydbbak.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 options`	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 'john'@'localhost' IDENTIFIED BY 'p4ssw0rd';
```
Create a MySQL user and set his password

```
DROP USER 'john'@'localhost';
```
Delete a MySQL user

```
SET PASSWORD FOR 'john'@'localhost' = PASSWORD('p4ssw0rd');
SET PASSWORD FOR 'john'@'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 'john'@'localhost';
```
Show permissions for a user

```
GRANT ALL PRIVILEGES ON MyDatabase.* TO 'john'@'localhost';
```
Grant permissions to a user

```
REVOKE ALL PRIVILEGES ON MyDatabase.* FROM 'john'@'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

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

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

```
USE MyDatabase; SOURCE mydbbak.sql;
```
Restore a database from a dump file

```
USE MyDatabase; LOAD DATA LOCAL INFILE 'foofile' INTO TABLE foo;
```
Populate a table with data from 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 a SQL statement to the server

```
statement\G
```
Display result in vertical format, showing each record in multiple rows

```
SELECT /*!99999 comment*/ * FROM MyDatabase.mytable;
```
Insert a comment

```
SELECT /*!n statement*/ * FROM MyDatabase.mytable;
```
The commented *statement* is executed only if MySQL is version *n* 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 they're 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 > /root/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 /root/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('s3cr3t');
```

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 setup 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.	Create the *user* PostgreSQL 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.	Grant the necessary privileges:	`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                          Start the appropriate Display Manager
/etc/init.d/gdm start
/etc/init.d/kdm start
```

`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		Latin 1				Latin 2	
BackSpace	ff08	space	0020	questiondown	00bf	Aogonek	01a1
Tab	ff09	exclam	0021	Agrave	00c0	breve	01a2
Linefeed	ff0a	quotedbl	0022	Aacute	00c1	Lstroke	01a3
Clear	ff0b	numbersign	0023	Acircumflex	00c2	Lcaron	01a5
Return	ff0d	dollar	0024	Atilde	00c3	Sacute	01a6
Pause	ff13	percent	0025	Adiaeresis	00c4	Scaron	01a9
Scroll_Lock	ff14	ampersand	0026	Aring	00c5	Scedilla	01aa
Sys_Req	ff15	apostrophe	0027	AE	00c6	Tcaron	01ab
Escape	ff1b	quoteright	0027	Ccedilla	00c7	Zacute	01ac
Delete	ffff	parenleft	0028	Egrave	00c8	Zcaron	01ae
		parenright	0029	Eacute	00c9	Zabovedot	01af
Cursor control		asterisk	002a	Ecircumflex	00ca	aogonek	01b1
Home	ff50	plus	002b	Ediaeresis	00cb	ogonek	01b2
Left	ff51	comma	002c	Igrave	00cc	lstroke	01b3
Up	ff52	minus	002d	Iacute	00cd	lcaron	01b5
Right	ff53	period	002e	Icircumflex	00ce	sacute	01b6
Down	ff54	slash	002f	Idiaeresis	00cf	caron	01b7
Prior	ff55	0 - 9	0030 – 0039	ETH	00d0	scaron	01b9
Page_Up	ff55	colon	003a	Eth	00d0	scedilla	01ba
Next	ff56	semicolon	003b	Ntilde	00d1	tcaron	01bb
Page_Down	ff56	less	003c	Ograve	00d2	zacute	01bc
End	ff57	equal	003d	Oacute	00d3	doubleacute	01bd
Begin	ff58	greater	003e	Ocircumflex	00d4	zcaron	01be
		question	003f	Otilde	00d5	zabovedot	01bf
Misc functions		at	0040	Odiaeresis	00d6	Racute	01c0
Select	ff60	A - Z	0041 - 005a	multiply	00d7	Abreve	01c3
Print	ff61	bracketleft	005b	Oslash	00d8	Lacute	01c5
Execute	ff62	backslash	005c	Ooblique	00d8	Cacute	01c6
Insert	ff63	bracketright	005d	Ugrave	00d9	Ccaron	01c8
Undo	ff65	asciicircum	005e	Uacute	00da	Eogonek	01ca
Redo	ff66	underscore	005f	Ucircumflex	00db	Ecaron	01cc
Menu	ff67	grave	0060	Udiaeresis	00dc	Dcaron	01cf
Find	ff68	quoteleft	0060	Yacute	00dd	Dstroke	01d0
Cancel	ff69	a - z	0061 - 007a	THORN	00de	Nacute	01d1
Help	ff6a	braceleft	007b	Thorn	00de	Ncaron	01d2
Break	ff6b	bar	007c	ssharp	00df	Odoubleacute	01d5
Mode_switch	ff7e	braceright	007d	agrave	00e0	Rcaron	01d8
script_switch	ff7e	asciitilde	007e	aacute	00e1	Uring	01d9
Num_Lock	ff7f	nobreakspace	00a0	acircumflex	00e2	Udoubleacute	01db
		exclamdown	00a1	atilde	00e3	Tcedilla	01de
Modifiers		cent	00a2	adiaeresis	00e4	racute	01e0
Shift_L	ffe1	sterling	00a3	aring	00e5	abreve	01e3
Shift_R	ffe2	currency	00a4	ae	00e6	lacute	01e5
Control_L	ffe3	yen	00a5	ccedilla	00e7	cacute	01e6
Control_R	ffe4	brokenbar	00a6	egrave	00e8	ccaron	01e8
Caps_Lock	ffe5	section	00a7	eacute	00e9	eogonek	01ea
Shift_Lock	ffe6	diaeresis	00a8	ecircumflex	00ea	ecaron	01ec
Meta_L	ffe7	copyright	00a9	ediaeresis	00eb	dcaron	01ef
Meta_R	ffe8	ordfeminine	00aa	igrave	00ec	dstroke	01f0
Alt_L	ffe9	guillemotleft	00ab	iacute	00ed	nacute	01f1
Alt_R	ffea	notsign	00ac	icircumflex	00ee	ncaron	01f2
Super_L	ffeb	hyphen	00ad	idiaeresis	00ef	odoubleacute	01f5
Super_R	ffec	registered	00ae	eth	00f0	rcaron	01f8
Hyper_L	ffed	macron	00af	ntilde	00f1	uring	01f9
Hyper_R	ffee	degree	00b0	ograve	00f2	udoubleacute	01fb
		plusminus	00b1	oacute	00f3	tcedilla	01fe
		twosuperior	00b2	ocircumflex	00f4	abovedot	01ff
		threesuperior	00b3	otilde	00f5		
		acute	00b4	odiaeresis	00f6		
		mu	00b5	division	00f7		
		paragraph	00b6	oslash	00f8		
		periodcentered	00b7	ooblique	00f8		
		cedilla	00b8	ugrave	00f9		
		onesuperior	00b9	uacute	00fa		
		masculine	00ba	ucircumflex	00fb		
		guillemotright	00bb	udiaeresis	00fc		
		onequarter	00bc	yacute	00fd		
		onehalf	00bd	thorn	00fe		
		threequarters	00be	ydiaeresis	00ff		

This is an excerpt of `keysymdef.h` which defines keysym codes (i.e. characters or functions associated with each key in X11) as `XK_key` and the *key* hex value. These keys can be used as argument for 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 7 8    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            3 4
```

1	Group name
2	Encrypted password, or `!` if no password 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 jdoe`	Create a user account, creating and populating his homedir from `/etc/skel`
`useradd -mc "John Doe" jdoe`	Create a user account, specifying his full name
`useradd -ms /bin/ksh jdoe`	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 "Jonas Doe" jdoe`	Modify the GECOS field of a user account
`usermod -L jdoe`	Lock a user account
`usermod -U jdoe`	Unlock a user account

Most options for `usermod` and `useradd` are the same.

`userdel -r jdoe`	Delete a user and his homedir
`chfn jdoe`	Change the GECOS field of a user
`chsh jdoe`	Change the login shell of a user
`passwd jdoe`	Change the password of a user
`passwd -l jdoe`	Lock a user account
`passwd -S jdoe`	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 jdoe`	Change the password expiration date; account will be locked at that date
`chage -d 13111 jdoe`	Change the date (in number of days since 1 January 1970) of last password change
`chage -d 0 jdoe`	Force the user to change password at his next login
`chage -M 30 jdoe`	Change the max number of days during which a password is valid
`chage -m 7 jdoe`	Change the min number of days between password changes
`chage -W 15 jdoe`	Change the number of days before password expiration that the user will be warned
`chage -I 3 jdoe`	Change the number of days after password expiration before the account is locked
`chage -l jdoe`	List password aging information for a user
`groupadd staff`	Create a group
`groupmod -n newstaff staff`	Change a group name
`groupdel staff`	Delete a group
`gpasswd staff`	Set or change the password of a group
`gpasswd -a jdoe staff`	Add a user to a group
`gpasswd -d jdoe staff`	Delete a user from a group
`gpasswd -A jdoe staff`	Add a user to the list of administrators of the group

`adduser`
`deluser`
`addgroup`   (Debian)
`delgroup`
	User-friendly front-end commands for user and group management

`system-config-users` (Red Hat)	GUI for user and group management

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, and so on) 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 a user
`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`	Scan authentication logs and temporarily ban IP addresses (via firewall rules) that have too many failed password logins
`/var/log/auth.log`	Logfile containing user logins and authentication mechanisms
`/var/log/pwdfail`	Logfile containing failed authentication attempts

`runuser -u user command`	Run *command* as *user*. Can be launched only by the superuser
`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
`sudo -uuser command`	Run *command* as *user*
`sudo command` `sudo -uroot command`	Run *command* as root
`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 a protected file. It is recommended to use this instead of allowing users to sudo text editors as root, which will cause security problems if the editor spawns a shell
`visudo`	Edit `/etc/sudoers`, the configuration file that specifies access rights to sudo

Sudo commands are logged via syslog on `/var/log/auth.log` (Debian) or `/var/log/secure` (Red Hat).

`sudo su -` `sudo -i`	Login on an interactive shell as the superuser
`gksu -u root -l` `gksudo -u root guicommand`	GUI front-ends to `su` and `sudo` used to run an X Window command as root. Pops up a requester prompting the user for root's password

`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
`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 his 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 you 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` *sessionowner/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

1.	**CTRL** **Z**	Suspend the job
2.	`bg`	Send the job to background
3.	`jobs`	Show the number (say *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)

Now, when the terminal is closed, the job will not be killed.

`write` *`user`*	Write interactively a message to the terminal of *user* (must be logged in)
`wall`	Write interactively a message to the terminal of all logged in users
`echo "Hello"	write` *`user`*	Write a message to the terminal of *user* (must be logged in)
`echo "Hello"	wall`	Write a message to the terminal of all logged in users
`talk` *`user`*	Open an interactive chat session with *user* (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` is used for repeated scheduled execution of commands.

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.

It is not necessary to restart `crond` after the modification of a crontab file, as the changes will be reloaded automatically.

`crontab -e`	Edit your user crontab file
`crontab -l`	List the contents of your crontab file
`crontab -e -u jdoe`	Edit the crontab file of another user (command available only to the superuser)
`/etc/crontab`	System-wide crontab file; this is the list of commands to execute periodically
`/etc/cron.d/`	Directory containing commands to execute periodically, one command per file (which must have the same syntax as `/etc/crontab`)

`/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*

/etc/crontab	
`# m  h  dom mon dow  user  command`	
` 25  6   *   *   1  root  foo.sh`	every Monday at 6:25 AM
`*/5 16   *   *   *  root  /opt/myscript.sh`	from 4:00 to 4:55 PM every 5 minutes everyday
`0,30 7  25  12   *  jdoe  /home/jdoe/bar.sh`	at 7:00 and 7:30 AM on 25th December
` 3 17   *   *  1-5 root  baz.sh`	at 5:03 PM everyday, 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 checks `/etc/crontab` every minute and runs the command as the specified user at the specified times.
Each user may also set his own crontab scheduling, which will result in a file `/var/spool/cron/user`; this user's crontab file has the same format as the system-wide crontab file, except that the `user` field is not present.

/etc/anacrontab	
`# period  delay  job-identifier  command`	
`   7      10    cron.weekly    /opt/myscript.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

Anacron jobs are run by `crond`, and permit 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).
The file `/var/spool/anacron/job_identifier` contains the date of the last execution of the specified anacron job.

`at` is used for scheduled execution of commands that must run only once.

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 root can access the service.

```
at 5:00pm tomorrow myscript.sh
at -f mylistofcommands.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

`bc`	Calculator
`factor`	Print the prime factors of an integer 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`	Password generator
`uuidgen`	Generate a UUID value, random or time-based
`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`	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`   (Debian)
`timedatectl set-timezone` *timezone* (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 a slow adjusting
`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 of 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

Syslog logging facility:

```
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 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 syslogd with facility "auth" and priority "info"

`logrotate`   Rotate logs. It gzips, renames, and eventually deletes old logfiles according to the configuration file `/etc/logrotate.conf`

`tail -f `*logfile*
`less +F `*logfile*   Display the logs in real-time. Prints the end of the log file, showing new entries and moving forward in the file as soon as they appear

`/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

`/var/log/secure`   Information about failed authentication and authorization (e.g. sshd failed logins)

MUA (Mail User Agent) mailclient of sender	MTA (Mail Transfer Agent) SMTP server of sender	MTA (Mail Transfer Agent) remote host	MDA (Mail Delivery Agent) mailserver of recipient	MUA (Mail User Agent) mailclient of recipient
e.g. Pine, Mutt	e.g. Sendmail, Exim, Postfix, qmail		e.g. Procmail, SpamAssassin	

`~/.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`	Commands to send mail

`mailx -s "`*Subject*`" \`
`-S smtp="mailserver.foobar.com:25" \`
`jdoe@example.org < `*messagefile*
	Send a mail message to jdoe@example.org, using an external SMTP server

`uuencode `*binaryfile*`	mail jdoe@example.org`	Send a binary file to jdoe@example.org (not recommended because many mailclients will display the received attachment inline)
`mutt -a `*binaryfile*` -- jdoe@example.org < /dev/null`	Send a binary file to jdoe@example.org 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.	`$HOME/Mail/`*folder*
Maildir	Each mail folder is a directory, and contains the subdirectories `/cur`, `/new`, and `/tmp`. Each email message is stored in its own file with a unique filename ID. The process that delivers an email message writes it to a file in the `tmp/` directory, and then moves it to `new/`. The moving is commonly done by hard linking the file to `new/` and then unlinking the file from `tmp/`, which guarantees that a MUA will not see a partially written message as it never looks in `tmp/`. When the MUA finds mail messages in `new/` it moves them to `cur/`. Advantages: fast location/retrieval/deletion of a specific mail message, no file locking needed, can be used with NFS. Disadvantages: some filesystems may not efficiently handle a large number of small files, searching text inside all mail messages is slow	`$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
```

Command	Description
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 a 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.

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

The `.cf` configuration files must not be edited by hand and are generated from editable `.mc` text files via the `m4` command, e.g. `m4 /etc/mail/submit.mc > /etc/mail/submit.cf`

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

The `.db` database files must not be edited by hand and are generated from editable text files via the `makemap` command, e.g. `makemap hash /etc/mail/access.db < /etc/mail/access`

`/var/spool/mqueue/`	Temporary mailqueue files (where *nnn* is the Message ID):	
	`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
`/var/spool/postfix/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
`/var/spool/postfix/deferred`	Deferred queue. A message is placed here when all its deliverable recipients are delivered, and for some recipients delivery failed for a transient reason. The queue manager scans this queue periodically and puts some messages into the active queue for a retry
`/var/spool/postfix/bounce`	Message delivery status report about why mail is bounced (non-delivered mail)
`/var/spool/postfix/defer`	Message delivery status report about why mail is delayed (non-delivered mail)
`/var/spool/postfix/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 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. The `#` is not a comment delimiter when not placed at the beginning of a line.

/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, non specific 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 tell Procmail to 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 (as 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, you must first 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 to listen on. `0` means all addresses
`PORT=127.0.0.1.900,192.168.0.1.900`	Port number connections are accepted on. 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 that shouldn't be changed
`LOGGEROPTS="-name=pop3d"`	`courierlogger` options
`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`	Name of the 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. Accepts 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`	File where `couriertcpd` will save its process ID
`TCPDOPTS="-nodnslookup -noidentlookup"`	Miscellaneous `couriertcpd` options that shouldn't be changed
`LOGGEROPTS="-name=imapd"`	`courierlogger` options
`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. This feature is mostly useful for shared folders, where the 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. This feature is used as an additional safety check that should stop any potential DoS attacks that exploit any kind of a memory leak to exhaust all the available memory 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. Normally, this setting should not be changed
`IMAP_TRASHFOLDERNAME=Trash`	Name of the 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`	Make the return address, $SENDER, being saved 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`	Name of the mail directory

Dovecot is an open source, security-hardened, fast, and efficient IMAP and POP3 server.
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 to accept connections on. Here, listen to all IPv4 and IPv6 interfaces
`disable_plaintext_auth = yes`	Disable LOGIN command and all other plaintext authentications unless SSL/TLS is used (LOGINDISABLED capability)
`shutdown_clients = yes`	Kill all IMAP and POP3 processes when Dovecot master process shuts down. If set to 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. Here, 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, to which the login process needs to be able to connect
`login_chroot = yes`	Chroot login process to the `login_dir`
`login_user = dovecot`	User to use for the login process. This user is used to control access for authentication process, and not to 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, do not give error if file 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. Should be the same for all namespaces; it depends on the underlying mail storage format
`prefix = shared/%%u/`	Prefix required to access this namespace; must be different for each. Here, 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`	There can be only one INBOX, and this setting defines which namespace has it
`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. Here, 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 grant access to for mail processes; typically these are used 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/wheel. Users having non-valid 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`	When IDLE command is running, mailbox is checked once in a while to see if there are any new mails or other changes. This setting defines the minimum time to wait between these checks, in seconds

/etc/dovecot.conf **Dovecot configuration file**	
`protocol pop3 {`	Block with options for the POP3 protocol
`  listen = *:110`	Network interfaces to accept POP3 connections on
`  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`	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 to accept IMAP and IMAPS connections on
`  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`	How many seconds to wait 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 = b1gs3cr3t`	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 b1gs3cr3t`
`ssl_ca_file = /etc/dovecot/cafile.pem`	List of trusted SSL certificate authorities; the file contains the CA certificates followed by the 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`	Whether to log unsuccessful authentication attempts and the reasons why they failed
`auth_debug = no`	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`
 	
   `passdb passwd-file {`
      `args = /etc/dovecot.deny`
      `deny = yes`
   `}`
 	
   `passdb pam {`
      `args = cache_key=%u%r dovecot`
   `}`
 	
   `passdb passwd {`
      `blocking = yes`
      `args =`
   `}`
 	
   `passdb shadow {`
      `blocking = yes`
      `args =`
   `}`
 	
   `passdb bsdauth {`
      `cache_key = %u`
      `args =`
   `}`
 	
   `passdb sql {`
      `args = /etc/dovecot/dovecot-sql.conf`
   `}`
 	
   `passdb ldap {`
      `args = /etc/dovecot/dovecot-ldap.conf`
   `}`
 	
   `socket listen {`
      `master {`
         `path = /var/run/dovecot/auth-master`
         `mode = 0600`
         `user =`
         `group =`
      `}`
      `client {`
         `path = /var/run/dovecot/auth-client`
         `mode = 0660`
      `}`
   `}`
 	
`}`
	Accepted authentication mechanisms
	
	Deny login to the users listed in `/etc/dovecot.deny` (file contains one user per line)
	
	PAM authentication block.
	Enables authentication matching (username and remote IP address) for PAM
	
	System users e.g. NSS or `/etc/passwd`
	
	Shadow passwords for system users e.g. NSS or `/etc/passwd`
	
	PAM-like authentication for OpenBSD
	
	SQL database
	
	LDAP database
	
	Export the authentication interface to other programs. Master socket provides access to userdb information; it 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

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	A 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	A free, 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.
lftp	A sophisticated FTP client with support for HTTP and BitTorrent.
	`lftp ftpserver.domain.org` Connect to a FTP server and tries 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`	After anonymous login, go to directory `/var/ftp/pub`
`anon_upload_enable=YES`	Allow anonymous uploads?
`chown_uploads=YES`	Change ownership of anonymously uploaded files?
`chown_username=ftp`	Change ownership of anonymously uploaded files to user `ftp`
`anon_world_readable_only=NO`	Allow anonymous users to only download files which are world readable?
`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`
	Versions of SSL/TLS that are allowed

`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

`cupsd`	CUPS (Common Unix Printing System) daemon. Administration of printers is done via web interface on 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 "Rejected" 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 jdoe`	View the print queue of a specific user
`lprm -P printer0 5`	Delete a specific job from a printer queue
`lprm -P printer0 jdoe`	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 derived from 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

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

4×10^9 total 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

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

3×10^{38} total 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 and other local registries.

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 (e.g.)**	**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

Command	Description

`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 10.1.1.1/8` `ifconfig eth0 10.1.1.1 netmask 255.0.0.0 broadcast 10.255.255.255`	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`
`dhcpcd eth0` (SUSE)
	Request an IP address via DHCP

`ip neigh` `arp -a`	Show the ARP cache table
`ip neigh show 10.1.0.6` `arp 10.1.0.6`	Show the ARP cache entry for a host
`ip neigh add 10.1.0.7 lladdr 01:23:45:67:89:ab dev eth0` `arp -s 10.1.0.7 01:23:45:67:89:ab`	Add a new ARP entry for a host
`ip neigh del 10.1.0.7 dev eth0` `arp -d 10.1.0.7`	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 mylinuxbox` `hostnamectl set-hostname --static "mylinuxbox"` (RHEL 7)	Set the hostname
`hostnamectl` (RHEL 7)	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

`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 expecting 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 remote 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
`ftp host`	Establish an interactive FTP connection with the remote host

`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 a HTTP PUT command with *data* to *webserver*

`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 LAN. Uses ARP (Layer 2) packets and is therefore able to find hosts that drop all IP or ICMP traffic
`ngrep`	Filter data payload of network packets matching a specified regex
`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 -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
`dhcpdump -i eth0`	Sniff all DHCP packets on interface eth0

```
nc
ncat    (Red Hat)
netcat  (SUSE)
```
Netcat, "the Swiss Army knife of networking", a very flexible generic TCP/IP client/server

```
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><h1>WWW</h1></body></html>" \
| ncat -l -p 80; done
```
Start a minimal web server, serving the specified HTML page to any connected client

```
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`   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

`/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

`/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 (i.e. MAC to IP addresses)

In RHEL7 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 it up again shortly
`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 many network configuration examples.

Network teaming allows binding together two or more network interfaces to increase throughput or provide redundancy. RHEL7 implements 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` lists many examples of team configurations and runners.
The manpage `man nmcli-examples` contains, among others, examples of teaming and bridging configuration.

`/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`	SSH to 10.0.1.0/24
`sshd: 10.0.1.`	SSH to 10.0.1.0/24
`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 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                                          Display IP routing table
route -en
route -F
netstat -rn


ip route show cache                               Display kernel routing cache
route -C


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

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

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

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

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, iptables is replaced by the `firewalld` daemon. It is possible, but not recommended, to use iptables anyway by installing the package `iptables-services` (which provides a systemd interface for iptables) and disabling firewalld.
In Ubuntu, iptables is managed by the `ufw` service (Uncomplicated Firewall).

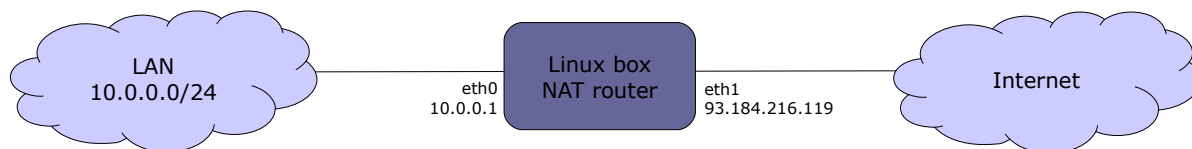|	
`/etc/sysconfig/iptables`	Default file containing the firewall rules
`iptables-restore < `*`file`*	Load into iptables the firewall rules specified in the *file*
`iptables-save > `*`file`*	Save into iptables the firewall rules specified in the *file*

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

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

`ssh `*`user@host`*	Connect to a remote *host* via SSH (Secure Shell) 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`*` /path/to/command`	Execute a command on a remote host
`sftp `*`user@host`*	FTP-like tool for secure file transfer
`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. Can transfer files from local to remote, from remote to local, or between two remote hosts
`sshpass -p `*`password`*` ssh `*`user@host`*	Connect to a remote host using the specified password
`pssh -i -H "`*`host1 host2 host3`*`" `*`/path/to/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 /etc/ssh/id_rsa -N '' -C ''`	Generate a RSA key with no passphrase (for non-interactive use) and no comment
`ssh-keygen -lf /etc/ssh/id_rsa.pub`	View key length and fingerprint of a public key
`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, one must add the keys to cache via the `ssh-add` command. The cached keys will then be automatically used by any SSH tool e.g. `ssh`, `sftp`, `scp`
`ssh-agent bash -c 'ssh-add `*`/path/to/keyfile`*`'`	Start ssh-agent and cache the specified key
`ssh-add`	Add the default private keys to the ssh-agent cache
`ssh-add `*`/path/to/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

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 tunneled 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

`openssl x509 -text -in certif.crt -noout`	Read a certificate
`openssl req -text -in request.csr -noout`	Read a Certificate Signing Request
`openssl req -new -key private.key -out request.csr`	Generate a Certificate Signing Request (in PEM format) for the public key of a key pair
`openssl req -new -nodes -keyout private.key \` `-out request.csr -newkey rsa:2048`	Create a 2048-bit RSA key pair and generate a Certificate Signing Request for it
`openssl req -x509 -newkey rsa:2048 -nodes \` `-keyout private.key -out certif.crt -days validity`	Generate a self-signed root certificate, and create a new CA private key
`openssl ca -config ca.conf -in request.csr \` `-out certif.crt -days validity -verbose`	Generate a self-signed certificate
`openssl ca -config ca.conf -gencrl -revoke certif.crt \` `-crl_reason why`	Revoke a certificate
`openssl ca -config ca.conf -gencrl -out crlist.crl`	Generate a Certificate Revocation List containing all revoked certificates so far
`openssl x509 -in certif.pem -outform DER \` `-out certif.der`	Convert a certificate from PEM to DER
`openssl pkcs12 -export -in certif.pem \` `-inkey private.key -out certif.pfx -name friendlyname`	Convert a certificate from PEM to PKCS#12 including the private key
`openssl pkcs12 -in certif.p12 -out certif.pem \` `-clcerts -nokeys`	Convert a certificate from PKCS#12 to PEM
`openssl pkcs12 -in certif.p12 -out private.key \` `-nocerts -nodes`	Extract the private key from a PKCS#12 certificate
`cat certif.crt private.key > certif.pem`	Create a PEM certificate from CRT and private key
`openssl dgst -hashfunction -out file.hash file`	Generate the digest of a file
`openssl dgst -hashfunction file | cmp -b file.hash`	Verify the digest of a file (no output means that digest verification is successful)
`openssl dgst -hashfunction -sign private.key \` `-out file.sig file`	Generate the signature of 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 TripleDES 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`	Retrieve and inspect a SSL certificate from a website
2. `CTRL` `C`	
3. `openssl x509 -in tmpfile -text`	
`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 -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 -newreq-nodes`	Generate a Certificate Signing Request, with unencrypted private key (for use in servers, because the private key must be accessed in non-interactive mode, without typing a passphrase)
`CA.pl -verify`	Verify a certificate against the Certification Authority certificate for "demoCA"

`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*

`md5sum`
`sha1sum`
`sha224sum`
`sha256sum`
`sha384sum`
`sha512sum`
`shasum`
	Print or check the digest of a file generated by a specific hashing algorithm

`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. The encrypted VPN tunnel uses UDP port 1194
`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
```

Key	Alternate key	Function
CTRL **F**	**→**	Move cursor forward one char
CTRL **B**	**←**	Move cursor backward one char
CTRL **A**	**HOME**	Move cursor to beginning of line
CTRL **E**	**END**	Move cursor to end of line
CTRL **H**	**BACKSPACE**	Delete char to the left of cursor
CTRL **W**		Delete word to the left of cursor
CTRL **U**		Delete all chars to the left of cursor
CTRL **K**		Delete all chars to the right of cursor
CTRL **T**		Swap current char with previous char
ESC **T**		Swap current word with previous word
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
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 (if it's a shell, same as `logout`)
CTRL **ALT** **DEL**		Send a SIGINT to reboot the machine (same as `shutdown -r now`); 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 a 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. `fdn`); place device node in `/dev/floppy/n` and create a symlink `/dev/fdn` 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/`	Contains 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 a `initrd` image file
`mkinitramfs` (Debian)	Create a `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

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

`/etc/ld.so.conf`	Configuration file used to specify other shared library locations (other than the default ones `/lib` and `/usr/lib`)
`ldconfig`	Create a cache file `/etc/ld.so.cache` of all available dynamically linked libraries. To 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's hardware
`lshw`	List system hardware
`lscpu`	List information about the CPU architecture
`uname -s`	Print the kernel name
`uname -n`	Print the network node hostname
`uname -r`	Print the kernel release number *X.Y.Z*
`uname -v`	Print the kernel version number
`uname -m`	Print the machine hardware name
`uname -p`	Print the processor type
`uname -i`	Print the hardware platform
`uname -o`	Print the operating system
`uname -a`	Print 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
`dmesg -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

Download	Download 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 leftover files, and similar
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 config file, based on the options in the old config 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 `/usr/src/linux/.config` config file containing instructions for the compile	
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: 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) 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, you can apply the patch again to reverse it
Build	Build the patched kernel as explained previously	
Install	Install the patched kernel as explained previously	

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

Configuration of device drivers	
Device drivers support the kernel with instructions on how to use that device.	
Device driver compiled into the kernel	Configure the device driver by passing a kernel parameter in the GRUB menu: `kernel /vmlinuz ro root=/dev/vg0/root vga=0x33c`
Device driver provided as a kernel module	Edit module configuration in `/etc/modprobe.conf` or `/etc/modprobe.d/` (Red Hat): `alias eth0 3c59x` Specify that eth0 uses the `3c59x.ko` driver module — `options 3c509 irq=10,11` Assign IRQ 10 and 11 to 3c509 devices

`/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/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	
`/proc/swaps`	Size of total and used swap areas	`swapon -s`
`/proc/sys/`	sysfs: exposes tunable kernel parameters	
`/proc/sys/kernel/`	Kernel information and parameters	
`/proc/sys/net/`	Network information and parameters	
`/proc/uptime`	Time elapsed since boot	`uptime`
`/proc/version`	Linux version	`uname -a`
`/proc/`*n*`/`	Information about process with PID *n*	`ps` *n*
`/proc/`*n*`/cmdline`	Command by which the process was launched	
`/proc/`*n*`/cwd`	Symlink to process' working directory	
`/proc/`*n*`/environ`	Values of environment variables of process	
`/proc/`*n*`/exe`	Symlink to process' executable	
`/proc/`*n*`/fd`	Files currently opened by the process	`lsof -p` *n*
`/proc/`*n*`/root`	Symlink to process' filesystem root	
`/proc/`*n*`/status`	Status of process	

`/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                       Get the maximum allowed number of open files
cat /proc/sys/fs/file-max

sysctl -w "fs.file-max=100000"           Set the maximum allowed number of open files to 100000
echo "100000" > /proc/sys/fs/file-max


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
```

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)

1. Power up the system and, on the GRUB 2 boot screen, press **E** to edit the current entry.

2. Edit the kernel line that mentions `linux16`, removing the `rhgb` and `quiet` parameters and adding `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
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 has to 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 (both must be created if needed) instead of root
`named -t /var/cache/bind`	Run BIND in a chroot jail `/var/cache/bind` (actually 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;};
};
```

```
          /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                              ; For 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		

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).
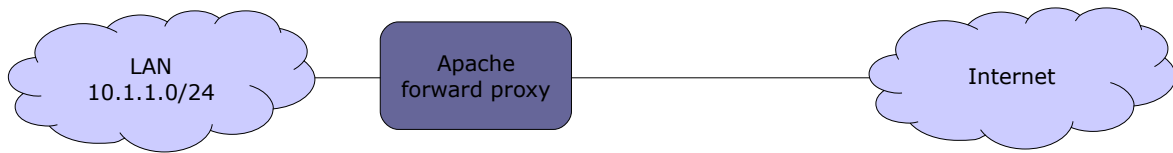A workaround for this is SNI (Server Name Indication) that makes the browser send the hostname in the first message of the SSL handshake. Another workaround is to have all multiple name-based virtual hosts use the same SSL certificate with a wildcard domain e.g. *.example.org.

`httpd.conf` **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 an 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`
`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
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 adds 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 a HTTP status 301 - Moved Permanently `temp` return a HTTP status 302 - Found (i.e. the resource was temporarily moved) `seeother` return a HTTP status 303 - See Other `gone` return a HTTP status 410 - Gone If status is omitted, default status `temp` is used
`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 a `.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

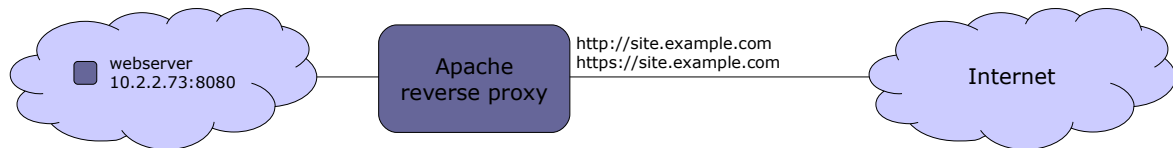| `httpd.conf` **Apache configuration file**	
Virtual hosts directives	
`NameVirtualHost *`	Specify which IP address will serve virtual hosting. The argument can be an IP address, an *address:port* pair, or `*` for all IP addresses of the server. The argument will be repeated in the relevant `<VirtualHost>` 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.
	In this case the config file must contain a `Listen 8080` directive

`<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`

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 (here, "common") for a log format. This one is the CLF (Common Log Format) 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`

`httpd.conf` Apache configuration file	
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

Directory protection directives	
`<Directory "/var/www/html/protected">`	
`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:
	`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
`Allow from 10.13.13.0/24`	Control which host can access the protected resource
`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
`Order Allow,Deny`	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
`</Directory>`	

httpd.conf **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` **send** `Server: Apache`
	`Major` **send** `Server: Apache/2`
	`Minor` **send** `Server: Apache/2.4`
	`Minimal` **send** `Server: Apache/2.4.2`
	`OS` **send** `Server: Apache/2.4.2 (Unix)`
	`Full` **send** `Server: Apache/2.4.2 (Unix)`
	`                  PHP/4.2.2 MyMod/1.2`
	If not specified, sends full header

`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 useless and is 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 to use the proxy.

`httpd.conf` 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.

`httpd.conf` Apache configuration file	
Reverse proxy	
`<VirtualHost *:80>`	Virtual host for HTTP
`   ServerName site.example.com`	Define website name
`   ProxyPass        / http://10.2.2.73:8080/` `   ProxyPassReverse / http://10.2.2.73:8080/`	Enable reverse proxying for server 10.2.2.73

`   RewriteEngine On`
`   RewriteCond %{HTTPS} off`
`   RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}`
	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 *>`
`      Order deny,allow`
`      Allow from all`
`   </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>`	

Tomcat is an open source Java Servlet Container implementing several Java EE specifications, and was 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.

`$JAVA_HOME`	Root of the Java installation e.g. `/usr/lib/jvm/java-1.8.0-openjdk.x86_64/`
`$CATALINA_HOME`	Root of the Tomcat installation e.g. `/usr/share/tomcat7`. 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`

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
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

`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.
WINS (Windows Internet Name Service) is a name service used to translate NetBIOS names to IP addresses.

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

`/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 jdoe`	Change the Samba password of user jdoe
`smbpasswd -a ksmith`	Create a new Samba user ksmith 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=jdoe`	Mount a Samba share as user jdoe
`smbstatus`	Display current information about shares, clients connections, and locked files
`smbclient //`*`smbserver`*`/share1`	Access a Samba share on a server (with a 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=`*`file`*	Mount the share as the user defined in the credentials *file* which must be formatted as such: `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

/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 password too

`  username map = /etc/samba/smbusers`	Map each Samba server user name to client user name(s).
	The file `/etc/samba/smbusers` is structured as follows:
	`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`	Whether to log only via Syslog
`  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`	Whether to show the public share when browsing
`writeable = yes`	Whether to 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`	Whether to show the homes share when browsing
`writeable = yes`	Whether to 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`	Whether to show the share when browsing
`writeable = yes`	Whether to allow the users to write in this share
`valid users = jdoe, kgreen, +geeks`	Allow access only to users jdoe and kgreen, and 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 srv1 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`
 	
`[foobar]`
`   path = /foobar`
`   username = quux`
`   only user = yes`
	Set up share-level authentication

	Define a foobar share accessible to any user which can supply quux's password.
	The user quux must be created on the system:
	`useradd -c "Foobar account" -d /tmp -m -s /sbin/nologin quux`
	and added to the Samba password file:
	`smbpasswd -a quux`

Samba macros			
`%S`	Username		The substitutes below apply only to the configuration options that are used when a connection has been established:
`%U`	Session username (the username that the client requested, not necessarily the same as the one he got)		
`%G`	Primary group of session username	`%S`	Name of the current service, if any
`%h`	Samba server hostname	`%P`	Root directory of the current service, if any
`%M`	Client hostname	`%u`	Username of the current service, if any
`%L`	NetBIOS name of the server	`%g`	Primary group name of username
`%m`	NetBIOS name of the client	`%H`	Home directory of username
`%d`	Process ID of the current server process	`%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
`%a`	Architecture of remote machine		
`%I`	IP address of client machine		
`%i`	Local IP address to which a client connected	`%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`
`%T`	Current date and time		
`%D`	Domain or workgroup of the current user		
`%w`	Winbind separator		
`%$(var)`	Value of the environment variable *var*		

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.

The portmapper is needed by NFS to map incoming TCP/IP connections to the appropriate NFS RPC calls. Some Linux distributions use rpcbind instead of the portmapper.
For security, 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 those entries in `/var/lib/nfs/xtab` that are deleted from `/etc/exports` (therefore synchronizing the two files), and removes those entries from `/proc/fs/nfs/exports` that are no longer valid
`exportfs -ua`	Unexport all directories. Removes from `/proc/fs/nfs/exports` all those 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      client1(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 a 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
   ```

A DHCP (Dynamic Host Configuration Protocol) 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, 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 an user's authentication credentials
	`session`	Module (run at end and beginning of an 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 n 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 permits 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, each one of which has a value. An attribute may appear multiple times.

Most frequently used LDAP attributes		
Attribute	**Example**	**Meaning**
`dn`	`dn: cn=John Doe,dc=example,dc=org`	Distinguished Name (not an attribute; identifies the entry)
`cn`	`cn: John Doe`	Common Name
`dc`	`dc=example,dc=org`	Domain Component
`givenName`	`givenName: John`	Firstname
`sn`	`sn: Doe`	Surname
`mail`	`mail: jdoe@example.org`	Email address
`telephoneNumber`	`telephoneNumber: +1 505 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

```
ldapsearch -H ldap://ldapserver.example.org \
-s base -b "ou=people,dc=example,dc=com" \
"(sn=Doe)" cn sn telephoneNumber
```
Query the specified LDAP server for entries where surname=Doe, and print common name, surname, and telephone number of the resulting entries.
Output is shown in LDIF

```
ldappasswd -x -D "cn=Admin,dc=example,dc=org" \
-W -S "uid=jdoe,ou=IT Dept,dc=example,dc=org"
```
Authenticating as Admin, change the password of user jdoe in the OU called IT Dept, on example.org

```
ldapmodify -b -r -f /tmp/mods.ldif
```
Modify an entry according to the LDIF file `/tmp/mods.ldif`

```
ldapadd -h ldapserver.example.org \
-D "cn=Admin" -W -f /tmp/mods.ldif
```
Authenticating as Admin, add an entry by adding the content of the LDIF file `/tmp/mods.ldif` to the directory.
This command actually invokes `ldapmodify -a`

```
ldapdelete -v "uid=jdoe,dc=example,dc=org" \
-D "cn=Admin,dc=example,dc=org" -W
```
Authenticating as Admin, delete the entry of user jdoe

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 jdoe, add a picture, and delete the description attribute for the entry

`slapd`	Standalone OpenLDAP daemon
`/var/lib/ldap/`	Files constituting the OpenLDAP database
`/etc/openldap/slapd.conf` `/usr/local/etc/openldap/slapd.conf`	OpenLDAP configuration file
`slapcat -l file.ldif`	Dump the contents of an OpenLDAP database to a LDIF file
`slapadd -l file.ldif`	Import an OpenLDAP database from a LDIF file
`slapindex`	Regenerate OpenLDAP's database indexes
`yum install openldap openldap-clients \` `authconfig sssd nss-pam-ldapd authconfig-gtk`	Install the OpenLDAP client (on RHEL 7)

`authconfig --enableldap --enableldapauth \ --`
`ldapserver=ldap://`*ldapserver* `\ --`
`ldapbasedn="dc=example,dc=org" \`
`--enablesssd --update`
	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`

`getent group` *groupname*	Get entries about *groupname* from NSS libraries
`authconfig-gtk` `system-config-authentication`	OpenLDAP configuration GUI

`sssd` (the System Security Services Daemon) must be running to provide access to OpenLDAP as an authentication and identity provider.

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) interact with **objects** (i.e. other processes, files, devices, ports, sockets); this improves security with respect to the standard Discretionary Access Control, which defines accesses based on users and groups. The **security context** of a file is stored in its extended attributes.
The decisions SELinux takes about allowing or disallowing access are stored in the AVC (Access Vector Cache).

`setenforce 0` `echo 0 > /selinux/enforce`	Enter permissive mode
`setenforce 1` `echo 1 > /selinux/enforce`	Enter enforcing mode

`getenforce`
`cat /selinux/enforce`
`sestatus -v`
	Display current mode

SELinux mode can be configured permanently in `/etc/selinux/config` (symlinked in `/etc/sysconfig/selinux`):

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
# targeted - Only targeted network daemons are protected.
# strict - Full SELinux protection.
SELINUXTYPE=targeted
```

`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
`ls -Z`	List files and their security context
`ps -eZ`	List processes and their security context
`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*. You then need to apply the label via `restorecon -f` *file*
`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
`getsebool` *boolean*	Get the value of a SELinux boolean
`setsebool` *boolean=value*	Set the value of a SELinux boolean
`tar --selinux` *[other args]* `star -xattr -H=exustar` *[other args]*	Create or extract archives that retain the security context of the original files

`/selinux/`	Pseudo filesystem created by SELinux, containing commands used by the kernel for its operations
`/var/log/audit/audit.log`	Logfile containing AVC denials, if `auditd` is running
`/var/log/messages`	Logfile containing AVC denials, if `rsyslogd` is running
`sealert -a` *`logfile`*	Analyze a SELinux logfile and display SELinux policy violations
`grep` *`nnnnn.mmm:pp logfile`* `	audit2why`	Diagnostic a specific AVC event entry from a SELinux logfile: `type=AVC msg=audit(`*`nnnnn.mmm:pp`*`): avc:  denied  `(...)

KVM (Kernel-based Virtual Machine) is a virtualization infrastructure for the Linux kernel that allows it to function as an hypervisor.

`/etc/libvirt/qemu/`	Directory containing the XML files that define VMs properties. `libvirtd` must be restarted after modifying a 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

Kickstart	
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.	
`system-config-kickstart`	GUI tool to create a Kickstart file
`ksvalidator ksfile`	Check the validity of a Kickstart file
`/root/anaconda-ks.cfg`	Kickstart file describing the current system. This file was automatically generated during the installation of the current system
`ksverdiff -f RHEL6 -t RHEL7`	Show the differences in the Kickstart syntax between RHEL 6 and RHEL 7

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 from any remote repository.

`git init`	Initialize the current directory as a repository
`git clone `*`repoaddress`*	Clone a remote repository. *repoaddress* can be a 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 pull`	Pull the changes from the remote repository branch to the local branch
`git add `*`file`*	Add *file* to the content staged for the next commit (hence starting to track it)
`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 -am "`*`Message`*`"`	Commit all staged files in the current 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 merge `*`branch`*	Merge changes made on *branch* to the master branch
`git diff `*`checksum1 checksum2`*	Compare two commits
`git log -G`*`word`*	Show the commits whose added or deleted lines contain *word*
`git branch`	Show local branches
`git branch -r`	Show remote branches
`git branch -a`	Show remote and local branches

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. This command reads cached data, hence completes quickly but can print outdated results; use the `--prune` option to rebuild the cache and obtain correct results

The directory containing the Vagrantfile on the host can be accessed on the guest via `/vagrant`.

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

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

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.

The `ascii` command and its manpage `man ascii` can be used to display an ASCII table.