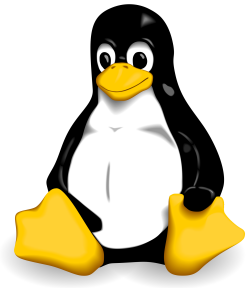
Linux Professional Institute



Linux & LPIC
Quick Reference Guide

2nd ed. 2014-09

Foreword

This guide stems from the notes I have been taking while working with Linux and preparing the LPIC-1 and LPIC-2 certifications. As such, it includes quite a good amount of topics for these exams, some subjects in more details than others. I started writing this guide in 2013 and it is my aim to update and integrate it periodically. Please check the edition number and date at the bottom of any page to ensure you're reading the latest release.

Happy Linux hacking,

Daniele Raffo

Suggested readings

- Adam Haeder et al., *LPI Linux Certification in a Nutshell*, O'Reilly
- Evi Nemeth et al., *UNIX and Linux System Administration Handbook*, O'Reilly
- Heinrich W. Klöpping et al., *The LPIC-2 Exam Prep*, `http://lpic2.unix.nl/`
- Mendel Cooper, *Advanced Bash-Scripting Guide*, `http://tldp.org/LDP/abs/html/`
- `http://www.gnu.org/manual/`
- `http://www.commandlinefu.com/`
- Linux man pages

Index

Logical Volume Management (LVM) introduces an abstraction between physical and logical storage that permits a more versatile use of filesystems.
LVM makes use of 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.
Physical Extents are mapped one-to-one to Logical Extents.
Logical Extents are grouped into Logical Volumes, on which filesystems are created.

How to create a Logical Volume

1.	`pvcreate /dev/hda2 /dev/hdb5`	Initialize one or more Physical Volumes to be used with LVM. Devices must be of partition type 0x8E
2.	`vgcreate -s 8M myvg0 /dev/hda2 /dev/hdb5`	Create a Volume Group and define the size of Physical Extents e.g. to 8 Mb (4 Mb by default)
3.	`lvcreate -L 1024M -n mydata myvg0`	Create a Logical Volume
4.	`mkfs -t ext3 /dev/myvg0/mydata`	Create a filesystem on the Logical Volume
5.	`mount /dev/myvg0/mydata /mydata`	The Logical Volume can now be mounted and used

How to extend a Logical Volume

1.	`vgextend myvg0 /dev/hdc`	Extend the Volume Group
2.	`lvextend -L 2048M /dev/myvg0/mydata`	Extend the Logical Volume
3.	`resize2fs /dev/myvg0/mydata`	Extend the filesystem

How to reduce a Logical Volume

1.	`resize2fs /dev/myvg0/mydata 900M`	Shrink the filesystem
2.	`lvreduce -L 900M /dev/myvg0/mydata`	Shrink the Logical Volume

Note: extension/shrinking of a Logical Volume are possible only if the underlying filesystem permits it.

How to snapshot and backup a Logical Volume

1.	`lvcreate -s -L 1024M -n snapshot0 /dev/myvg0/mydata`	Create the snapshot just like another Logical Volume
2.	`tar cvzf snapshot0.tar.gz snapshot0`	Backup the snapshot with any backup tool
3.	`lvremove /dev/mvvg0/snapshot0`	Delete the snapshot

`pvs`	Report information about Physical Volumes
`pvck`	Check Physical Volume metadata
`pvdisplay`	Display Physical Volume attributes
`pvscan`	Scan all disks for Physical Volumes
`pvremove`	Remove a Physical Volume
`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
`vgs`	Report information about Volume Groups
`vgck`	Check Volume Group metadata
`vgmerge`	Merge two Volume Groups
`vgimport`	Import a Volume Group into a system
`vgexport`	Export a Volume Group from a system
`vgchange`	Change Volume Group attributes

`lvs`	Report information about Logical Volumes
`lvchange`	Change Logical Volume attributes
`lvscan`	Scan all disks for Logical Volumes

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 will panic.

Startup	The system loads startup scripts and runlevel scripts.
X Server	(Optional) The X Display Manager starts the X Server.

Some newer systems use UEFI (Unified Extensible Firmware Interface). 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 the GRUB.

OS startup sequence (SysV)	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 on boot is to add it to 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*

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 the service supports this option, otherwise restart the service	
`condrestart` `try-restart`	Restart the service only if already running	Optional
`reload`	Reload service configuration	

`/etc/init.d/`*service* `start`
`service` *service* `start` (Red Hat)
`rc`*service* `start` (SUSE)

Start a service

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

The default runlevels are **2 3 4 5**

`runlevel`
`who -r`

Display the previous and the current runlevel

`init` *`runlevel`*
`telinit` *`runlevel`*

Change runlevel

`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.
This command can be run only by the root user and by those users (if any) listed in `/etc/shutdown.allow`

`shutdown -h 16:00` *`message`*

Schedule a shutdown for 4 PM and send a warning message to all logged in users

`shutdown -a`

Non-root users that are listed in `/etc/shutdown.allow` can use this command to shut down the system

`shutdown -f`

Skip fsck on reboot

`shutdown -F`

Force fsck on reboot

`shutdown -c`

Cancel an already running shutdown

`update-rc.d` *service* `defaults` (Debian) `chkconfig --add` *service* (Red Hat)	Add 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 S symlinks for runlevels 2 3 4 5
`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`	Start the service on runlevels 2 4 5	
`chkconfig` *service* `on`	Start the service on default runlevels (via the `xinetd` super server)	
`chkconfig` *service* `off`	Stop the service on default runlevels	
`chkconfig` *service* `reset`	Reset the on/off state of the service for all runlevels to whatever is specified in the init script *	
`chkconfig` *service* `resetpriorities`	Reset the start/stop priorities of the service for all runlevels to whatever is specified in the init script *	
`chkconfig --list` *service*	Display current configuration of service (its status and the runlevels in which it is active)	
`chkconfig --list`	List all active services and their current configuration	

* The Linux Standard Base (LSB) defines a format to specify the 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 be also specified as such:

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

/etc/inittab

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

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

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

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

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

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

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

Filesystem Hierarchy Standard (FHS)	
/bin	Essential command binaries
/boot	Bootloader files (e.g. OS loader, kernel image, initrd)
/dev	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 the 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
/tmp	Temporary files
/usr	User utilities and applications
/usr/bin	Non-essential command binaries (for all users)
/usr/lib	Libraries for the binaries in /usr/bin and /usr/sbin
/usr/sbin	Non-essential system binaries (daemons and services)
/usr/src	Source code
/var	Variable files (e.g. logs, caches, mail spools)

`/dev/hda, /dev/hdb, /dev/hdc`	first, second, third IDE hard drive
`/dev/sda, /dev/sdb, /dev/sdc`	first, second, third SATA hard drive
`/dev/sda1, /dev/sda2, /dev/sda3`	first, second, third partition of the first SATA drive

Partitioning limits for Linux:
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

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.
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.

`fdisk /dev/sda`	Disk partitioning interactive tool
`cfdisk`	Text-based UI fdisk
`gparted`	GUI fdisk
`fdisk -l /dev/sda`	List the partition table of `/dev/sda`
`partprobe`	After fdisk operations, this command must be run to notify the OS of partition table changes. Otherwise, these 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` `mke2fs`
	`mkfs.ext3` `mke3fs`
	`mkfs.ext4`
	`mkfs.msdos` `mkdosfs`
	`mkfs.reiserfs` `mkreiserfs`
	`mkfs.jfs`
	`mkfs.xfs`

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

`mke2fs -j /dev/sda`
`mkfs.ext3 /dev/sda`
`mke3fs /dev/sda`
	Create a 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 /etc/mtab`
`cat /proc/mounts`
	Display the currently mounted filesystems.
	`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 msdos /dev/fd0 /mnt`	Mount a MS-DOS filesystem floppy disk to mount point `/mnt` (this directory must exist)
`mount /dev/fd0`	Mount a floppy disk; `/etc/fstab` must contain an entry for `/dev/fd0`
`umount /dev/fd0` `umount /mnt`	Unmount a floppy disk that was mounted on `/mnt` (must not be busy to unmount)
`umount -l /dev/fd0`	Unmount the floppy disk as soon as it is not in use anymore
`mount -o remount,rw /`	Remount the root directory as read-write (supposing it was mounted read-only). Used 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 the 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

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.

`fdisk` 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
```

Any of these commands can be used to show the sizes of total and used swap areas

Most used Linux-supported filesystems		
Filesystem	**Properties**	**Partition type**
ext2	Linux default filesystem, offering the best performances	0x83
ext3	ext2 with journaling	
ext4	Linux journaling filesystem, 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	0x82

/etc/fstab – Information about filesystems

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

/dev/sda2          /                 ext2      defaults                 1 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, its label, or its UUID (Universal Unique Identifier) which is a 128-bit hash number that is associated to the partition at its initialization	
mount point	Directory on which the partition must be mounted	
type	Filesystem type, or `auto` if detected automatically	
options	`defaults`	Use the default options: `rw, suid, dev, exec, auto, nouser, async`
	`ro`	Mount read-only
	`rw`	Mount read-write
	`suid`	Permit SUID and SGID bit operations
	`nosuid`	Do not permit SUID and SGID bit operations
	`dev`	Interpret block special devices on the filesystem
	`nodev`	Do not interpret block special devices on the filesystem
	`auto`	Mount automatically at bootup, or when the command `mount -a` is given
	`noauto`	Mount only if explicitly demanded
	`user`	Partition can be mounted by any user
	`nouser`	Partition can be mounted only by the root user
	`exec`	Binaries contained on the partition can be executed
	`noexec`	Binaries contained on the partition cannot be executed
	`sync`	Write files immediately to the partition
	`async`	Buffer write operations and commit them later, or when device is unmounted
	`rsize=`*`nnn`*	NFS: Size for read transfers (from server to client)
	`wsize=`*`nnn`*	NFS: Size for write transfers (from client to server)
	`nfsvers=`*`n`*	NFS: Version of NFS to use for transport
	`retry=`*`n`*	NFS: Time to keep retrying a mount attempt before giving up, in minutes
	`timeo=`*`n`*	NFS: Time after a mount attempt times out, in tenths of a second
	`intr`	NFS: User can interrupt a mount attempt
	`nointr`	NFS: User cannot interrupt a mount attempt (default)
	`hard`	NFS: The system will try a mount indefinitely (default)
	`soft`	NFS: The system will try a mount until an RPC timeout occurs
	`bg`	NFS: The system will try a mount in the foreground, all retries occur in the background
	`fg`	NFS: All mount attempts occur in the foreground (default)
	`tcp`	NFS: Connect using TCP
	`udp`	NFS: Connect using UDP
dump	Dump (backup utility) options. `0` = do not backup	
pass	Fsck (filesystem check utility) options. Defines in which order the filesystems should be checked; `0` = do not check	

`df`	Report filesystem disk space usage
`df -h`	Report filesystem disk space usage in human-readable output
`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`	Creates a directory allocating the proper inode. Useful during system recovery when experiencing filesystem problems

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

`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 benchmarks on the /dev/hda drive
`hdparm -p 12 /dev/hda`	Reprogram IDE interface chipset of /dev/hda to mode 4. Use with caution!
`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` 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 actual filesystem-specific checker commands:

```
fsck.ext2      e2fsck
fsck.ext3
fsck.ext4
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 a ext2 filesystem, running the `badblocks` command to mark all bad blocks and add them to the bad block inode to prevent them from being allocated to files or directories

`tune2fs` *[options] device*

Adjust tunable filesystem parameters on ext2/ext3/ext4 filesystems

`tune2fs -j /dev/sda1`

Add a journal to this ext2 filesystem, making it a ext3

`tune2fs -C 4 /dev/sda1`

Set the mount count of the filesystem to 4

`tune2fs -c 20 /dev/sda1`

Set the filesystem to be checked by fsck after 20 mounts

`tune2fs -i 15d /dev/sda1`

Set the filesystem to be checked by fsck each 15 days

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

`dumpe2fs` *[options] device*

Dump ext2/ext3/ext4 filesystem information

`dumpe2fs -h /dev/sda1`

Display filesystem's superblock information (number of mounts, last checks, UUID, ...)

`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)

Most 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 (there must be at least one spare new disk partition available)

```
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 (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 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, and in this case signals the `automount` userspace daemon which mounts the directory when it needs to be accessed and unmounts it when no longer accessed.

`/etc/auto.master`

Primary configuration file for AutoFS.
Each line is an indirect map; each map file stores the configuration for subdirs automounting

```
# mount point   map              options
/misc           /etc/auto.misc
/home           /etc/auto.home   --timeout=60
```

`/etc/auto.misc`

Configuration 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`

Configuration file for automounting of directory `/home` .
The `*` wildcard matches any subdir the system attepts to access, and the `&` variable takes the value of the match

```
# subdir   options                          filesystem
*          -rw,soft,intr                   nfsserver.example.org:/home/&
```

The `/net/`*`nfsserver`*`/` tree allows nonprivileged users to automatically access any *nfsserver*.

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
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 `mdadm`

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

```
cat /proc/mdstat
```

Display information about RAID arrays and devices

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

The CD must contain the following files:

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

The CD can be burnt with the command:

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

The boot TFTP server must contain the following files:

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

GRUB (Grand Unified Bootloader) is the standard boot manager on modern Linux distros, which may use either version: GRUB Legacy or GRUB 2.
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` .

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

```
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
```

Common kernel parameters:

`root=`	Specify the location of the filesystem root. Required parameter
`ro`	Mount read-only on boot
`quiet`	Disable non-critical kernel messages during boot
`debug`	Enable kernel debugging
`splash`	Show splash image
`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. 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`

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

```
# 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
}
```

This file must not be edited manually. Instead, edit the files in `/etc/grub.d/` (they 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` .

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

ENTER	Boot the 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 GRUB entry once it has been modified
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

`/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

Package management		Debian	Red Hat
Low-level tools	Install a package file	`dpkg -i package.deb`	`rpm -i 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 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 tools (can install remote packages, automatically solve dependencies)	Install a package	`apt-get install package`	`yum install package`
	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`	
	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 about available packages	`apt-get update`	
	List all available packages		`yum list`
	Search for a package	`apt-cache search package`	`yum search package`
	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`
	Update information about package contents	`apt-file update`	
	List the content of an uninstalled package	`apt-file list package`	
	Show the package containing a specific file	`apt-file search file`	`yum provides file`
	Add a CD-ROM to the list of available sources	`apt-cdrom add`	
	Download package and resolve dependencies		`yumdownloader \ --resolve package`
	List the URLs that would be downloaded		`yumdownloader \ --urls package`
Text-based UI or graphical tools	Manage packages and dependencies	`aptitude`	
		`dselect`	
Other tools	Convert a RPM package to DEB and installs it. Might break the package system!	`alien -i package.rpm`	
	Convert a RPM package to cpio archive		`rpm2cpio package.rpm`
Miscellaneous information	List of available sources	`/etc/apt/sources.list`	`/etc/yum.repos.d`
	Package format	compressed with `ar` (package `binutils`)	compressed with `cpio`

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

```
dd if=/dev/sda1 of=sda1.img
```
Create the image of a partition

```
dd if=/dev/cdrom of=cdrom.iso bs=2048
```
Create an ISO file from a CD-ROM, using a block size of 2 Kb

```
rsync -rzv /home  /tmp/bak
rsync -rzv /home/ /tmp/bak/home
```
Synchronize the content of the home directory with the temporary backup directory. Use compression, verbosity, and recursion. 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 (operates recursively and preserves owner, group, permissions, timestamps, and symlinks)

```
ls | cpio -o > myarchive.cpio
ls | cpio -oF myarchive.cpio
```
Create an archive of all files that are on the current directory

```
find /home/ | cpio -o > homedirs.cpio
```
Create an archive of all users' home directories

```
cpio -id < myarchive.cpio
```
Extract all files from the archive, recreating the structure of directories

```
cpio -i -t < myarchive.cpio
```
List the contents of an archive file without extracting it

```
gzip myfile
```
Compress a file with gzip

```
gunzip myfile.gz
```
Decompress a gzip-compressed file

```
zcat myfile.gz
```
Read a gzip-compressed text file

```
bzip2 myfile
```
Compress a file with bzip2

```
bunzip2 myfile.bz2
```
Decompress a bzip2-compressed file

```
bzcat myfile.bz2
```
Read a bzip2-compressed text file

```
tar cvzf myarc.tar.gz mydir/
tar xvzf myarc.tar.gz
```
Create/extract a tarred gzip-compressed archive

```
tar cvjf myarc.tar.bz2 mydir/
tar xvjf myarc.tar.bz2
```
Create/extract a tarred bzip2-compressed archive

```
tar cvJf myarc.tar.xz mydir/
tar xvJf myarc.tar.xz
```
Create/extract a tarred xz-compressed archive

```
tar tvf myarc.tar
```
List the contents of the tarred archive without extracting it

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)

`man 7 command` — Show man page 7 for a command

`man man` — Show information about man pages' content:
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

`cd directory` — Change to the specified directory

`cd -` — Change to the previously used directory

`pwd` — Print the current directory

`history` — Show the history of command lines executed up to this moment.
Commands prepend by a space will be executed but won't 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` — Delete command line history

`uname -a` — Print system information

`vlock`
`away` — Lock the virtual console (terminal)

Almost all Linux commands accept the option `-v` (verbose), and many commands also accept the option `-vv` (very verbose).

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

`cat myfile`	Print a text file
`cat myfile1 myfile2 > myfile3`	Concatenate text files
`head myfile` `head -n 10 myfile`	Print the first 10 lines of a text file
`tail myfile` `tail -n 10 myfile`	Print the last 10 lines of a text file
`tail -f myfile`	Output appended data as the text file grows; useful to read logs in realtime
`tac myfile`	Print a text file in reverse, from last line to first line
`fmt -w 75 myfile`	Format a text file so that each line has a max width of 75 chars
`pr myfile`	Format a text file for a printer
`nl myfile`	Prepend line numbers to a text file
`wc myfile`	Print the number of lines, words, and bytes of a text file
`join myfile1 myfile2`	Join lines of two text files on a common field
`paste myfile1 myfile2`	Merge lines of text files
`split -l 1 myfile`	Split a text file into 1-line files (named `xaa`, `xab`, `xac`, ...)
`uniq myfile`	Print the unique lines of a text file, omitting consecutive identical lines
`sort myfile`	Sort alphabetically the lines of a text file
`expand myfile`	Convert tabs into spaces
`unexpand myfile`	Convert spaces into tabs
`od myfile`	Dump a file into octal
`cut -d: -f3 myfile`	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
`sed s/foo/bar/ myfile`	Stream Editor: Replace the first occurrence of foo with bar
`sed s/foo/bar/g myfile`	Replace all occurrences of foo with bar
`tr a-z A-Z <myfile` `tr [:lower:] [:upper:] <myfile`	Translate characters: Convert all lowercase into uppercase in a text file
`tr -d 0-9 <myfile` `tr -d [:digit:] <myfile`	Delete all digits from a text file

```
cp myfile myfile2
```
Copy a file

```
cp myfile mydir/
```
Copy a file to a directory

```
mv myfile myfile2
```
Rename a file

```
mv myfile mydir/
```
Move a file to a directory

```
rm myfile
```
Delete a file

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

```
mkdir mydir
```
Create a directory

```
mkdir -m 777 mydir
```
Create a directory with 777 permission

```
mkdir -p /tmp/mydir1/mydir2
```
Create a directory, and the parent directories if they don't exist

```
rmdir mydir
```
Delete an empty directory

```
touch myfile
```
Change access/modification timestamp on a file, creating it if it doesn't exist

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 myfile.{txt,bak}`	Copy myfile.txt to myfile.bak
`touch myfile_{a,b,c}`	Create myfile_a, myfile_b, myfile_c
`touch {a..h}`	Create 8 files named a b c d e f g h

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

File descriptors			
#	**Name**	**Type**	**Default device**
0	Standard input (stdin)	Input text stream	Keyboard
1	Standard output (stdout)	Output text stream	Terminal
2	Standard error (stderr)	Output text stream	Terminal

```
ls | sort
```
Pipe the stdout of command `ls` to stdin of command `sort`
(i.e. generate a sorted list of the files on the current directory)

```
ls > myfile
ls 1> myfile
```
Redirect the stdout of command `ls` to a file
(i.e. write on a file the content of the current directory).
File is overwritten if it already exists; to prevent this, set the Bash noclobber option via `set -o noclobber`

```
ls >| myfile
```
Redirect the stdout of command `ls` to a file, even if noclobber is set

```
ls >> myfile
ls 1>> myfile
```
Append the stdout of command `ls` to a file

```
df 2> myfile
```
Redirect the stderr of command `df` to a file
(i.e. write any error encountered by the command `df` to a file)

```
df 2>> myfile
```
Append the stderr of command `df` to a file

```
mail root@example.com < myfile
```
Redirect a file to the stdin of command `mail`
(i.e. mail a file to the specified email address)

```
ls > myfile 2>&1
ls &> myfile
```
Redirect both stdout and stderr of command `ls` to a file

```
ls | tee myfile
```
`tee` reads from stdin and writes both to stdout and a file
(i.e. write content of current directory to screen and to a file at the same time)

```
ls | tee -a myfile
```
`tee` reads from stdin and appends both to stdout and a file

```
ls foo* | xargs cat
```
`xargs` calls the `cat` command multiple times for each argument found on stdin
(i.e. print the content of every file whose filename starts by foo)

Any application/program/script that runs on the system is a process. Signals are used for inter-process communication.
Each process has an 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
`top` `htop`	Monitor processes in realtime

`kill -9 1138`	Send a signal 9 (SIGKILL) to process 1138, hence killing it
`killall -9 sshd`	Kill 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

(`pgrep` and `pkill` accept the same options)

`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

When a Bash shell is terminated cleanly via `exit`, its jobs will became child of the Bash's parent and will continue running.
When a Bash is killed instead, it issues a SIGHUP to his children which will terminate.

`nohup myscript.sh`	Prevent a process from receiving a SIGHUP (hence terminating) when its parent Bash dies

To each process is associated a niceness value: the lower the niceness, the higher 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

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

`man 7 signal` Manual page about signals

`kill -l` List all available signal names

`kill -l 1` Print the name of signal number 1

`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

`vmstat`
Print a report about process usage, virtual memory, blocks I/O, interrupts, and CPU time

`vmstat 1 5`
Print a report every second, for 5 times

`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

`sar`
Show reports about system activity.
Reports are generated from data collected via the cron job `sysstat` and stored in `/var/log/sa/sn`, where *n* is the day of the month

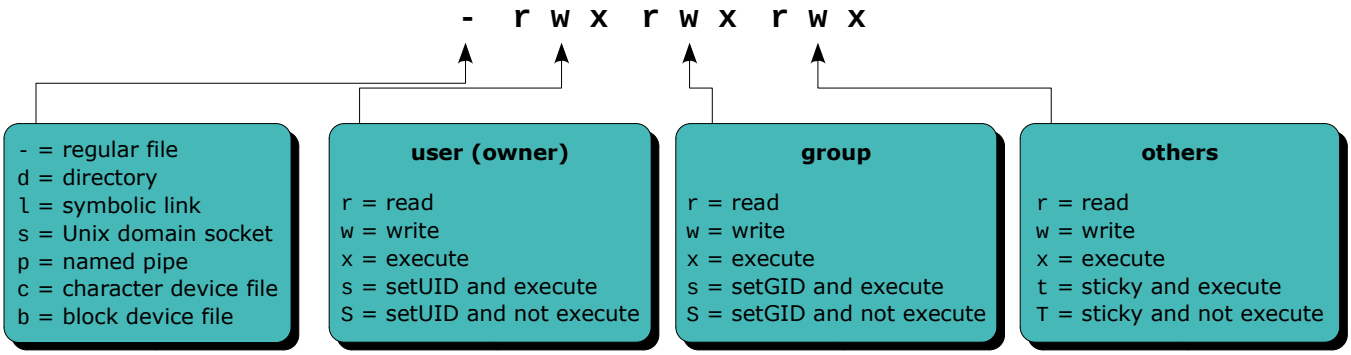
`sar -n DEV`
Show reports about network activity (received and transmitted packets per second)

```
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

`iotop`
Display I/O usage by processes in the system

Monitoring tools	
collectd	System statistics collector
Nagios	System monitor and alert
MRTG	Network load monitor
Cacti	Network monitor

Regular expressions	
^	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
{3,6}	Between 3 and 6 times the preceding regex
|	The regex either before or after the vertical bar
( )	Grouping, to be used for back-references. \1 expands to the first match, \2 for the second, and so on until \9



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 access the directory
	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	Only the file's or the directory's owner can delete or rename a file inside

`chmod 710 file` `chmod u=rwx,g=x file`	Set read, write, and execute permission to user; set execute permission to group
`chmod ug=rw file` `chmod 660 file`	Set read and write permission to user and group
`chmod +wx file`	Add write and execute permission to everybody (user, group, and others)
`chmod -R o+r file`	Add recursively read permission to others
`chmod o-x file`	Remove execute permission from others

`chown root file`	Change the owner of file to root
`chown root:mygroup file`	Change the owner of file to root, and the group of file to mygroup

`chgrp mygroup file`	Change the group of file to mygroup

The `chmod`, `chown`, and `chgrp` commands accept the option `-R` to recursively change properties of files and directories.

`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.

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 the file is in.

`ls -i` Show a listing of the directory with the files' inode numbers

`df -i` Report filesystem inode usage

	Hard link	Symbolic or soft link
What it is	A link to an already existing inode	A path to a filename; a shortcut
How to create it	`ln` *`myfile hardlink`*	`ln -s` *`myfile symlink`*
Is the link still valid if the original file is moved or deleted	Yes (because the link references the inode the original file pointed to)	No (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 all files, starting from the root dir, whose name start with foo
`find / -name "foo*" -print`	Find all files whose name start with foo and print their path
`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 / -perm -4000 -type f`	Find all files 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
`locate ls` `slocate ls`	Locate the command `ls` by searching the file index, not by actually walking the filesystem. The search is quick but will only held results relative to the last rebuilding of the file index (`/etc/updatedb.conf`)
`updatedb`	Build the file index (`/etc/updatedb.conf`)
`which command`	Locate a binary executable `command` within the PATH
`which -a command`	Locate all matches of `command`, not only the first one
`whereis command`	Locate the binary, source, and manpage files for `command`
`whereis -b command`	Locate the binary files for `command`
`whereis -s command`	Locate the source files for `command`
`whereis -m command`	Locate the manpage files for `command`
`file myfile`	Analyse the content of a file or directory
`type command`	Determine if `command` is a program or a builtin (i.e. a feature internal to the shell)

Bash shell event	Files run	

When a login shell is launched	`/etc/profile`
	`~/.bash_profile`
	`~/.bash_login`
	`~/.profile`
		The shell executes the system-wide profile file, 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` `~/.bashrc`	

`MYVAR=myvalue` `((MYVAR=myvalue))`	Set a variable
`((MYVAR++))`	Post-increment a numeric variable (C-style)
`unset MYVAR`	Delete a variable
`export MYVAR`	Export a variable so it can be seen by Bash child processes
`echo $MYVAR`	Print the value of a variable
`echo ${MYVAR:-mymessage}`	If variable exists and is not null, print its value, otherwise print a message
`echo ${MYVAR:+mymessage}`	If variable exists and is not null, print a message, otherwise print nothing
`set ${MYVAR:=myvalue}`	Set a variable only if it does not exist or is null
`set`	Display all Bash variables
`set -o`	Show the status of all Bash options
`set -o` *`option`*	Enable a Bash option
`set +o` *`option`*	Disable a Bash option
`env`	Display all environment variables
`typeset -f`	Show functions defined in the current Bash session
`alias ls='ls -lap'`	Set up an alias for the `ls` command
`alias`	Show defined aliases
`\ls` `/bin/ls`	Run the non-aliased version of the `ls` command

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 a 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
`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
`function myfunc { commands }` `myfunc() { commands }`	Define a function
`myfunc arg1 arg2 ...`	Call a function
`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 $MYVAR`	Print a variable on screen
`echo -n "mymessage"`	Print on screen without a trailing line feed
`MYVAR=\`date\`` `MYVAR=$(date)`	Assign to a variable the output resulting from a command
`zenity`	Display GTK+ graphical dialogs for user messages and input

Bash built-in variables	
`$0`	Script name
`$1, $2, ...`	First, second, ... argument passed to the script or function
`$#`	Number of arguments passed to the script or function
`$?`	Exit status of the last executed command
`$$`	PID of the script in which this variable is called

```
test $MYVAR = "myvalue" && mycommand
[ $MYVAR = "myvalue" ] && mycommand
if [ $MYVAR = "myvalue" ]; then mycommand; fi
```

Perform a test; if it holds true, the 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		

`expr $MYVAR = "39 + 3"`	Evaluate an expression; the variable will hold the value 42
`expr` *`string`* `:` *`regex`*	Return the length of the substring matching the regex
`expr` *`string`* `: \(`*`regex`*`\)`	Return the substring matching the regex

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				

Tests

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

```
case $VAR in
   [pattern 1]) [command 1] ;;
   [pattern 2]) [command 2] ;;
   *) [command 3]
esac
```

Loops

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

```
for $I in [list]
do
   [command operating on $I]
done
```

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

SQL syntax	
`USE MyDatabase;`	Choose which database to use
`SHOW DATABASES;`	Show all existing databases
`SHOW TABLES;`	Show all tables from the selected database
`DESC tableCustomers;`	Describe the columns of a table
`SELECT * FROM tableCustomers;`	Select all columns from the table
`SELECT * FROM tableCustomers ORDER BY columnLastname LIMIT 5;`	Select only the first 5 records of customers as ordered by last name
`SELECT columnFirstname, columnLastname FROM tableCustomers WHERE columnZipcode = 00123;`	Select only first and last name of customers whose zip code is 00123
`SELECT columnCustomerID, SUM(columnSalary) FROM tablePayments GROUP BY columnCustomerID;`	Select all salary payments grouped by customer ID, summed up
`SELECT tableCustomers.columnLastname, tablePayments.columnAmount FROM tableCustomers, tablePayments WHERE tableCustomers.columnCustomerID = tablePayments.columnCustomerID;`	Perform a join by selecting data from two tables that are linked
`INSERT INTO tableCustomers (columnFirstname,columnLastname,columnDOB) VALUES (Arthur,Dent,1959-08-01);`	Insert new data
`UPDATE tableCustomers SET columnCity = 'London' WHERE columnZipcode = 00789;`	Modify data
`SHOW GRANTS FOR 'user'@'localhost';`	Show permissions for a user
`GRANT ALL PRIVILEGES ON MyDatabase.* TO 'user'@'localhost';`	Grant permissions to a user
`REVOKE ALL PRIVILEGES FROM 'user'@'localhost';`	Revoke permissions from a user
`SELECT Host,User FROM mysql.user;`	List MySQL users
`CREATE USER 'user'@'localhost' IDENTIFIED BY 'p4ssw0rd';`	Create a MySQL user
`SET PASSWORD FOR 'user'@'localhost' = PASSWORD('p4ssw0rd');`	Set a password for a MySQL user

MySQL command line syntax	
`mysql -u root -p`	Login to MySQL as root, prompting for the password
`mysql -u root -ps3cr3t`	Login to MySQL as root with password s3cr3t
`mysql -u root -p -e 'CREATE DATABASE NewDatabase'`	Create a new database by passing a SQL command to MySQL
`mysql -u root -p NewDatabase < newdb.sql`	Create a new database from an external file (`.sql` files are composed of SQL commands)
`mysqldump -u root -p MyDatabase > backup.sql`	Backup a database on an external file

Display Managers				
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 following line: xlogin*greeting: \ Debian GNU/Linux (CLIENTHOST)
		/etc/x11/xdm/Xresources	Configuration settings for X applications and the login screen	
		/etc/x11/xdm/Xservers	Association of X displays with local X server software, or with X terminals via XDMCP	
		/etc/x11/xdm/Xsession	Script launched by xdm after login	
		/etc/x11/xdm/Xsetup_0	Script launched before the graphical login screen	
		/etc/x11/xdm/xdm-config	Association of all xdm configuration files	
gdm	GNOME Display Manager	/etc/gdm/gdm.conf or /etc/gdm/custom.conf		Configured via gdmsetup
kdm	KDE Display Manager	/etc/kde/kdm/kdmrc		Configured via kdm_config

/etc/init.d/xdm start /etc/init.d/gdm start /etc/init.d/kdm start	Start the X Display Manager
xorgconfig	Configure X (text mode) (Debian)
Xorg -configure	Configure X (text mode) (Red Hat)
xorgcfg	Configure X (graphical mode) (Debian)
system-config-display	Configure X (graphical mode) (Red Hat)
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	Add 10.3.3.3 to the list of hosts allowed to make X connections to the local machine
xhost - 10.3.3.3	Remove 10.3.3.3 from the list of hosts allowed to make X connections to the local machine
mkfontdir	Catalog the newly installed fonts in the new directory
xset fp+ /usr/local/fonts	Dynamically add the newly 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
switchdesk gde	Switch to the GDE Display Manager at runtime

/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*

/etc/inittab instructs init to launch XDM at runlevel 5:

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

/etc/sysconfig/desktop defines GNOME as the default Display Environment and Display Manager:

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

/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	Encrypted password (obsolete), or `x` if password is in `/etc/shadow`
③	UID – User ID (UID 0 is superuser; by convention UIDs 1-99 are system accounts, UIDs above are regular users)
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 (can be set to `/bin/false` to prevent a user from logging in)

/etc/shadow User passwords (file is readable only by root)

```
root:fZPe54/Kldu6D32pl0X/A:15537:0:99999:7:::
bin:*:15637:0:99999:7:::
jdoe:!hsp\8e3jCUdw9Ru53:15580:0:99999:7::15766:
 ①          ②                  ③  ④  ⑤  ⑥⑦  ⑧  ⑨
```

①	Login name
②	Encrypted password (a `!` prefix if the account is locked), `*` if account is disabled, `!` or `!!` if no password
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 (file is readable only by root)

```
root::root:root
jdoe:!::
staff:0cfz7IpLhGW19i::root,jdoe
 ①      ②            ③   ④
```

①	Group name
②	Encrypted password, or `!` if no password set (default)
③	Group administrators
④	Group members

`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 (specified in `/etc/login.defs`) for user account creation
`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

(`usermod` accepts many `useradd` options)

`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
`chage -E 2013-02-14 jdoe`	Change the password expiration date, locking the account 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`
`delgroup`
	User-friendly front-ends for user and group management (Debian)

User control	
`who am i` `whoami`	Print your effective user ID
`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
`fail2ban`	Scan authentication logs and temporarily ban IP addresses (via firewall rules) that have too many failed password logins
`/var/log/auth.log`	Log containing user logins and authentication mechanisms
`/var/log/pwdfail`	Log containing failed authentication attempts
`/etc/nologin`	If this file exists, `login` and `sshd` deny login to the system

su and sudo	
`su jdoe`	Run a shell as the specified user. If user is not specified, assume 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 fdisk -l`	Run a command as root. Sudo commands are logged via syslog
`sudo -ujdoe fdisk -l`	Run a command as another user
`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 arise security problems if the editor spawns a shell
`visudo`	Edit `/etc/sudoers`, the configuration file that specifies access rights to sudo

`echo "Message" | write jdoe`	Write a message to the terminal of user jdoe
`echo "Message" | wall`	Write a message to the terminal of all logged in users
`talk jdoe`	Open an interactive chat session with user jdoe

`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 superuser is always able to message users.

`echo $(tty)`	Print your terminal device (e.g. `/dev/tty1`, `/dev/pts/1`)

`/etc/issue` — Message to be printed before the login prompt. Can contain these escape codes:

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

`/etc/issue.net`	Message to be printed before the login prompt on a remote session
`/etc/motd`	Message to be printed after a successful login, before execution of the login shell

cron – repeated scheduled execution

/etc/crontab

```
# m  h  dom mon dow  user  command
  25 6   *   *   1   root  myscript.sh
```

```
  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)
```

```
25    6  *  *   1 = every Monday at 6:25 AM
*/5  16  *  *   * = from 4:00 to 4:55 PM every 5 mins, everyday
0,30  7 25 12   * = on 25th December at 7:00 and 7:30 AM
3    17  *  * 1-5 = at 5:03 PM everyday, from Monday to Friday
```

The crond daemon checks the /etc/crontab system-wide file every minute and executes command as user at the specified times.
Each user may also set his own crontab scheduling, which will result in a file /var/spool/cron/*username*. A user' crontab file has the same format, except that the user field is not present.

/etc/anacrontab

```
# period  delay  job-identifier  command
  7       10     cron-weekly     myscript.sh
```

```
        period = period in days
         delay = delay in minutes
job-identifier = job identifier in anacron messages
```

Anacron jobs are run by crond, and permit the execution of periodic jobs on a machine that is not always running, such as a laptop.
If the job has not been executed in the last period, the system waits for delay and then executes command.

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.

`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 (only root can do this)

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

at – scheduled execution 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

Locale environment variables		
LANG LANGUAGE	Language, stored in `/etc/default/locale` . When scripting, `LANG=C` should be set because this specifies the minimal locale environment for C translation, and guarantees a standard collation and formats for the execution of scripts	These locale 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`
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	

`locale`	Show locale environment variables
`locale-gen it_IT.UTF-8`	Generate a locale by compiling a list of locale definition files
`apt-get install manpages-it language-pack-it`	Install a different locale (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.

`tzselect`
`tzconfig`
`dpkg-reconfigure tzdata` (Debian)
	Set the timezone, stored in `/etc/timezone`

Timezone is also set as a symbolic link from `/etc/localtime` to the correct timezone file in `/usr/share/zoneinfo/`

`date`	Show current date and time
`date -d "9999 days ago"`	Show a different, calculated date
`date -d "1970/01/01 + 14662"`	Convert number of days since 1 January 1970 (e.g. 14662) in a canonical date
`date +"%F %H:%M:%S"`	Show date in the format specified
`date -s "20130305 23:30:00"`	Set the date
`date 030523302013`	Set the date, in the format *MMDDhhmmYYYY*
`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 -n -g -q`	Start NTP as a non-daemon, force set the clock, and quit
`ntpq -p` *`timeserver`*	Query the time server for a list of peers
`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
`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`	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
```

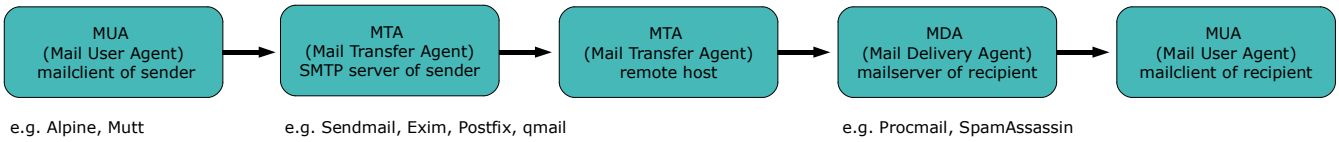
Facility Creator of the message	**Level** Severity of the message	**Action** Destination of the message	

`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)
		`filename`
		
		`@hostname`
		
		`user1,user2,user3`
		
		`*`
			message is written into a logfile
			
			message is sent to a logger server (via UDP port 514)
			
			message is sent to users' consoles
			
			message is sent to all logged-in users' consoles

† deprecated			

`logger -p auth.info "Message"`	Send a message to syslogd with the specified facility and priority
`man 3 syslog`	Syslog manpage listing facilities and levels
`logrotate`	Rotate logs (by gzipping, renaming, and eventually deleting old logfiles) according to `/etc/logrotate.conf`
`tail -f /var/log/messages`	Print the last lines of the message log file, moving forward as the file grows (i.e. read logs in real-time)
`zgrep` *`grep_options file`*	Grep search in a gzipped *file*
`zcat /var/log/messages.1.gz`	Print a gzipped file on stdout

`/var/log/messages`
`/var/log/syslog`
`/var/log/kern.log`
	System and kernel logfiles



`~/.forward`	Mail address(es) to forward the user's mail to, 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 -s "Subject" -c "jdoe@example.org" <` *`bodyfile`*	Send a mail message

`newaliases` `sendmail -bi`	Update the aliases database; must be run after any change to `/etc/aliases`
`mailq` `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

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

SMTP commands		
220 smtp.example.com ESMTP Postfix HELO abc.example.org 250 Hello abc.example.org, glad to meet you MAIL FROM: alice@example.org 250 Ok RCPT TO bob@foobar.com 250 Ok RCPT TO eve@foobar.com 250 Ok DATA 354 End data with <CR><LF>.<CR><LF> From: Alice <alice@example.org> To: Bob <bob@foobar.com> Cc: Eve <eve@foobar.com> Date: Wed, 13 August 2014 18:02:43 -0500 Subject: Test message This is a test message. . 250 OK id=1OjReS-0005kT-Jj QUIT 221 Bye	HELO abc.example.org EHLO abc.example.org MAIL FROM: alice@example.org RCPT TO: bob@foobar.com DATA QUIT RSET HELP NOOP VRFY jdoe@example.org EXPN mailinglist	Initiate the conversation and identify client host to server Like HELO, but tell server to use Extended SMTP Specify mail sender Specify mail recipient Specify data to send. Ended with a dot on a single line Disconnect List all available commands Empty command Verify the existence of an e-mail address (this command should not be implemented, for security reasons) 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 distributed as a monolithic binary file.
It used to run SUID `root`, which caused many security problems; recent versions runs 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 are generated from edited `.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 are generated from edited text files via the `makemap` command, e.g.
`makemap hash /etc/mail/access.db < /etc/mail/access`

`sendmail -bt`	Run Sendmail in test mode
`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

`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`*	Create a hashed map file of database type *dbtype* from *textfile*
`postalias` `newaliases`	Convert `/etc/aliases` into the aliases database file `/etc/aliases.db`

/etc/postfix/main.cf	Postfix 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 if it is 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

Courier POP configuration

The Courier MTA provides modules for ESMTP, IMAP, POP3, webmail, and mailing list services in a single framework.

The `courier-authlib` service must be launched first, then 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 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
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. Re-sends 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 straight deleting them)
HEADERFROM=X-IMAP-Sender	Make the return address, $SENDER, being saved in the X-IMAP-Sender mail header. This header gets 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 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
`}`	
`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
`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 user contains a character not listed in here, the login automatically fails. This is to prevent an 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`	Accepted authentication mechanisms

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

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

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

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

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

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

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

`socket listen {`
`master {`
`path = /var/run/dovecot/auth-master`
`mode = 0600`
`user =`
`group =`
`}`
`client {`
`path = /var/run/dovecot/auth-client`
`mode = 0660`
`}`
`}`
	Export the authentication interface to other programs.
	Master socket provides access to userdb information; 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

Very Secure FTP is 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.

vsftpd.conf	
`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 to allow

`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

Pure-FTP is 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

`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	
`193.22.33.44`	32-bit divised in 4 octects (dotted-quad) 4 billion addresses

IPv6	

`2130:0000:0000:0000:0007:0040:15bc:235f`
 	
`2130:0:0:0:7:40:15bc:235f`
 	
`2130::7:40:15bc:235f`
	128-bit
	divised in 8 16-bit sections
	
	3×10^{38} addresses

IPv4 classful addressing, as assigned by IANA

		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

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

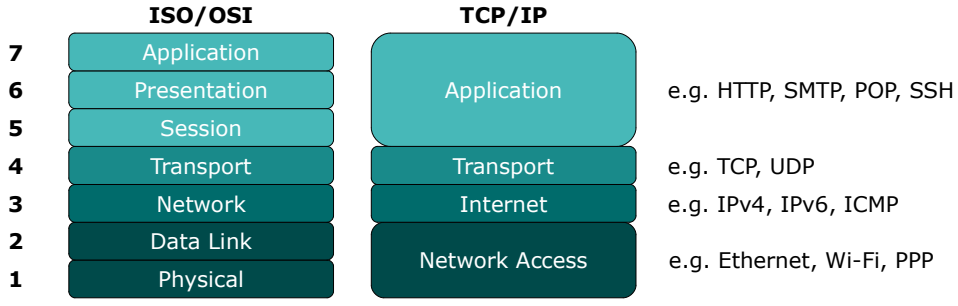
Each block of a column identifies a subnet, whose range of valid hosts addresses is [network address +1 — broadcast address -1] inclusive.
The network address of the subnet is the number shown inside a block.
The broadcast address of the subnet is the network address of the block underneath -1 or, for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

Most frequently used 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
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

The full list of well-known ports is in `/etc/services`

Protocol stack models



```
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 a ARP entry

```
ip neigh flush all
```
Delete the ARP table for all interfaces

`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
`hostname`	Get the hostname (stored in `/etc/hostname`)
`hostname -f`	Get the FQDN (Fully Qualified Domain Name)
`hostname mylinuxbox`	Set the hostname
`/etc/init.d/networking` `/etc/init.d/network`	Initialize network services

`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`
`dig @10.7.7.7 -t MX example.org`	Perform a DNS lookup for the MX record of the domain example.org, querying nameserver 10.7.7.7
`dig -x 203.0.113.1`	Perform a reverse DNS lookup for the IP address 203.0.113.1
`host example.org`	Perform a DNS lookup for the specified domain or hostname. Does honor `/etc/resolv.conf`
`host example.org 10.7.7.7`	Perform a DNS lookup for the domain example.org, querying nameserver 10.7.7.7
`host 192.168.13.13`	Perform a reverse DNS lookup for the IP address 192.168.13.13
`nslookup example.org` (deprecated)	Perform a DNS lookup for the specified domain or hostname
`whois example.org`	Query the WHOIS service for an Internet resource, usually a domain name
`ping 10.0.0.2`	Test if a remote host can be reached and measure the round-trip time to it (by sending an ICMP ECHO_REQUEST datagram and expecting an ICMP ECHO_RESPONSE)
`fping -a 10.0.0.2 10.0.0.7 10.0.0.8`	Ping multiple hosts in parallel and report which ones are alive
`traceroute 10.0.0.3`	Print the route, hop by hop, packets trace to a remote host (by sending a sequence of ICMP ECHO_REQUEST datagrams with increasing TTL values, starting with TTL=1)
`tracepath 10.0.0.3`	Simpler `traceroute`
`mtr 10.0.0.3`	`traceroute` and `ping` combined
`telnet 10.0.0.4 23`	Establish a telnet connection to the specified host and port (if port is omitted, use default port 23)
`ftp 10.0.0.5`	Establish an interactive FTP connection with host 10.0.0.5
`wget –-no-clobber –-html-extension \ --page-requisites --convert-links \ --recursive --domains example.org \ --no-parent www.example.org/foobar`	Download a whole website www.example.org/foobar
`nc` `netcat` (SUSE)	Netcat, the Swiss Army knife of networking, a very flexible generic TCP/IP client/server
`nc -l -p 25`	Listen for connections on port 25 (i.e. mimic a SMTP server). Send any input on stdin to the connected client and dump on stdout any data received from the client
`nc 10.0.0.7 389 < myfile`	Push the content of a 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 command
`while true; \ do nc -l -p 80 -q 1 < mypage.html; done`	Start a web server, serving the specified HTML page to any connected client
`nc -z 10.0.0.7 22`	Scan for a listening SSH daemon on remote host 10.0.0.7
`nc -v -n -z -w1 -r 10.0.0.7 1-1023`	Run a TCP port scan against remote host 10.0.0.7. Probe 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

`netstat`	Display network connections
`netstat –-tcp`	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 (similar to `netstat`)
`ss -t -a`	Display all TCP sockets
`nmap 10.0.0.1` `nmap -sS 10.0.0.1`	Scan for open ports (TCP SYN scan) on remote host 10.0.0.1
`nmap -sP 10.0.0.1`	Do a ping sweep (ICMP ECHO probes) on remote host
`nmap -sU 10.0.0.1`	Scan UDP ports on remote host
`nmap -sV 10.0.0.1`	Do a service and version scan on open ports
`nmap -p 1-65535 10.0.0.1`	Scan all ports (1-65535) on remote host, not only the common ports
`nmap -O 10.0.0.1`	Find which operating system is running on remote host (OS fingerprinting)
`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 with that MAC address
`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
`lsof`	List all open files
`lsof -u jdoe`	List all files currently open by user jdoe
`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 the MySQL daemon
`lsof /var/run/mysqld/mysqld.sock`	List all processes which are using a specific file
`iptraf`	IP LAN monitor (ncurses UI)

`/sys/class/net` List of all network interfaces in the system

`/etc/hosts` Mappings between IP addresses and hostnames, for simple name resolution

```
127.0.0.1  localhost  localhost.localdomain
10.2.3.4   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` Specification of the domain names that must be appended to bare hostnames and of the 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

`/etc/ethers` ARP mappings (MAC to IP addresses)

`/etc/inetd.conf` Configuration file for inetd, the super-server Internet daemon

`/etc/hostname` Hostname of the local machine

`/etc/network/interfaces` List and configuration of all network interfaces

`/etc/sysconfig/network-scripts/ifcfg-eth0` (RedHat) Configuration file for network interface eth0.
This file is read by the `ifup` and `ifdown` scripts

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
NETMASK=255.255.255.0
IPADDR=10.2.3.4
USERCTL=no
```

`/etc/sysconfig/network-scripts/ifcfg-eth0:0` (RedHat)
`/etc/sysconfig/network-scripts/ifcfg-eth0:1`
`/etc/sysconfig/network-scripts/ifcfg-eth0:2`
Configuration files for different interface aliases.
This makes possible to bind multiple IP addresses to a single NIC

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

Host access control files used by the TCP Wrapper system.

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

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

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

`sshd: 10.0.1.0/24`
`sshd: 10.0.1.`
`sshd: 10.0.1.0/255.255.255.0`
	SSH to 10.0.1.0/24 (all commands are equivalent)

`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

```
ip route
route -en
route -F
netstat -rn
```

Display IP routing table

Gateway:

host	gateway name
*	no gateway
-	rejected 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 show cache
route -C
```

Display kernel routing cache

```
ip route add default via 10.1.1.254
route add default gw 10.1.1.254
```

Add a default gateway

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

Add a route for a host

```
ip route add 10.2.0.0/16 via 10.2.0.254
route add -net 10.2.0.0 netmask 255.255.0.0 gw 10.2.0.254
```

Add a route for a network

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

Delete a route for a host

```
ip route flush all
```

Delete the routing table for all interfaces

```
/etc/sysconfig/network-scripts/route-eth0  (RedHat)
```

Static route configuration for eth0

```
ADDRESS=10.2.0.0
NETMASK=255.255.0.0
GATEWAY=10.2.0.254
```

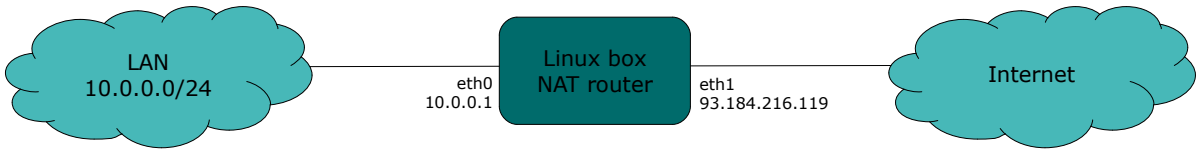
The Netfilter framework provides firewalling capabilities in Linux. It is implemented by `iptables` (which replaced `ipchains`, which itself replaced `ipfwadm`). The IPv6 equivalent of iptables is `ip6tables`.

Tables contain sets of chains, which contain sets of rules.
The filter table contains chains INPUT, FORWARD, OUTPUT (built-in chains).
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, and take no further action
`iptables -D INPUT -s 10.0.0.9 -j LOG`	Delete a rule
`iptables -D INPUT 42`	Delete rule 42 of the INPUT chain
`iptables -F INPUT`	Flush all rules of the INPUT chain
`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 -P INPUT -j DROP`	Define the chain policy, which takes effect when no rule matches and the end of the rules list is reached
`iptables -A OUTPUT -d 10.7.7.0/24 -j DROP`	Add a rule to drop all packets with destination 10.7.7.0/24
`iptables -A FORWARD -i eth0 -o eth1 -j LOG`	Add a rule to log all packets entering the system via eth0 and exiting via eth1
`iptables -A INPUT -p 17 -j DROP` `iptables -A INPUT -p udp -j DROP`	Add a rule to drop all incoming UDP traffic (protocol numbers are defined in `/etc/protocols`)
`iptables -A INPUT --sport 1024:65535 --dport 53 \` `-j ACCEPT`	Add a rule to accept all packets coming from any unprivileged port and with destination port 53
`iptables -A INPUT -p icmp --icmp-type echo-request \` `-m limit --limit 1/s -i eth0 -j ACCEPT`	Add a rule to accept incoming pings through eth0 at a maximum rate of 1 ping/second
`iptables -A INPUT -m state --state ESTABLISHED \` `-j ACCEPT`	Load the module for stateful packet filtering, and add a rule to accept all packets that are part of a communication already tracked by the state module
`iptables -A INPUT -m state --state NEW -j ACCEPT`	Add a rule to accept all packets that are not part of a communication already tracked by the state module
`iptables -A INPUT -m state --state RELATED -j ACCEPT`	Add a rule to accept all packets that are related (e.g. ICMP responses to TCP or UDP traffic) to a communication already tracked by the state module
`iptables -A INPUT -m state --state INVALID -j ACCEPT`	Add a rule to accept all packets that do not match any of the states above

`iptables-save > fwrules.saved`	Save iptables configuration to a file
`iptables-restore < fwrules.saved`	Restore a iptables configuration from a file
`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)



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

```
ssh root@remotehost
```
Connect to a remote host via SSH (Secure Shell) and login as the superuser

```
ssh -p 2222 root@remotehost
```
Login as the superuser to a remote host via SSH using port 2222 instead of standard port 22

```
ssh root@remotehost /root/myscript.sh
```
Execute a command on a remote host

```
sftp root@host.foo.com
```
FTP-like tool for secure file transfer

```
scp myfile root@host.foo.com:/tmp/myfile2
scp root@host.foo.com:/tmp/myfile2 myfile
scp jdoe@host1:/tmp/myfile root@host2:/root/myfile2
```
Non-interactive secure file copy.
Able of transferring files from local to remote, from remote to local, or between two remote systems

```
ssh-keygen -t rsa -b 2048
```
Generate interactively a 2048-bit RSA key pair, prompting 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/ssh_host_key \
-N '' -C ''
```
Generate a RSA key with no passphrase (to be used by a server host, not a user) and no comment

```
ssh-keygen -l -f /etc/ssh/ssh_host_key
```
View fingerprint of a public key

```
ssh-agent
```
Start the SSH Agent daemon that caches decrypted private keys in memory; also echoes to the terminal the environment variables that must be set. The cached keys are automatically used by SSH tools `ssh`, `sftp` and `scp`

```
eval `ssh-agent`
```
Show the PID of ssh-agent and set appropriate environment variables

```
ssh-add ~/.ssh/id_rsa
```
Add a private key to the ssh-agent cache

```
ssh -L 2525:mail.foo.com:25 user@mail.foo.com
```
SSH port forwarding (aka SSH tunneling)
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 -R 2222:localhost:22 user@login.foo.com
```
SSH reverse forwarding (aka SSH reverse tunneling)
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 -D 33333 user@login.foo.com
```
SSH as a SOCKS proxy
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

```
ssh -X user@login.foo.com
```
X11 Forwarding
Enable the local display to execute locally a X application stored on a remote host login.foo.com

SSH files	
`/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	
`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

How to enable public key authentication
1. Set `PubkeyAuthentication yes` in `/etc/ssh/sshd_config` of remote server
2. Append your public key `~/.ssh/id_rsa.pub` to the file `~/.ssh/authorized_keys` on the remote server

How to enable host-based authentication amongst a group of trusted hosts
1. Set `HostbasedAuthentication yes` in `/etc/ssh/sshd_config` on all hosts
2. Create `/etc/ssh/shosts.equiv` on all hosts, and enter there all 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 SSH Agent
1. Type `eval `ssh-agent``
2. Type `ssh-add` to add the private key to cache, and enter the key's passphrase

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 &`

How to enable X11 Forwarding via telnet (insecure and obsolete)
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 &`

`gpg --gen-key`	Generate a key pair
`gpg --import alice.asc`	Import Alice's public key 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_backup.gpg`	Export your whole keyring to a file
`gpg --export-secret-key -a "You" -o private.key`	Export your private key (username You) to a file
`gpg --export-public-key -a "Alice" -o alice.pub`	Export Alice's public key to a file
`gpg --edit-key "Alice"`	Sign Alice's public key
`gpg -e -u "You" -r "Alice" file.txt`	Encrypt a file (to Alice i.e. with Alice's public key), signing it with your private key
`gpg -d file.txt.gpg`	Decrypt a file (with your own public key)

`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	RIGHT ARROW	Move cursor forward one char
CTRL B	LEFT ARROW	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 buffer
SHIFT PAGE DOWN		Scroll down the buffer
CTRL L		Clear screen (same as `clear`)
CTRL P	UP ARROW	Previous command in history
CTRL N	DOWN ARROW	Next command in history
CTRL R		Reverse history search
TAB		Autocomplete file and directory names
CTRL J	RETURN	Line feed
CTRL M		Carriage return
CTRL S		Pause trasfer to terminal
CTRL Q		Resume transfer to terminal
CTRL Z		Send a SIGTSTP to put the current job in background
CTRL C		Send a SIGINT to stop the current process
CTRL D		Send a EOF to current process (same as `logout`)
CTRL ALT DEL		Send a SIGINT to reboot the machine (same as `shutdown -r now`)*
CTRL ALT F1 ... F6		Switch between text consoles
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		Change between X Window tasks
CTRL ALT BACKSPACE		Reboot the X Window server

* as specified in `/etc/inittab` and `/etc/init/control-alt-delete`

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 dynamically generates the device nodes in `/dev/` for devices present on the system. udev 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. `fd`*n*); place device node in `/dev/floppy/`*n* and create a symlink `/dev/fd`*n* to it
`SUBSYSTEM=="block", ATTR{size}=="41943040", SYMLINK+="mydisk"`	Match a block device with a size attribute of 41943040; create a symlink `/dev/mydisk`
`KERNEL=="fd[0-9]*", OWNER="jdoe"`	Match all floppy disk drives; give ownership of the device file to user jdoe
`KERNEL=="sda", PROGRAM="/bin/mydevicenamer %k", SYMLINK+="%c"`	Match a device named by the kernel as `sda`; to name the device, use the defined program which takes on stdin the kernel name and output on stdout e.g. *name1 name2*. Create symlinks `/dev/`*name1* and `/dev/`*name2* pointing to `/dev/sda`
`KERNEL=="sda", ACTION=="add", RUN+="/bin/myprogram"`	Match a device named by the kernel as `sda`; run the defined program when the device is connected
`KERNEL=="sda", ACTION=="remove", RUN+="/bin/myprogram"`	Match a device named by the kernel as `sda`; run the defined program when the device is disconnected
%n = kernel number (e.g. = 3 for `fd3`) %k = kernel name (e.g. = `fd3` for `fd3`) %c = device name as output from program	

A kernel version number has the form *major.minor.patchlevel*.
Kernel images are usually gzip-compressed and can be of two types: zImage (max 520 Kb) and bzImage (no size limit).
Kernel modules can be loaded dynamically into the kernel to provide additional functionalities on demand, instead of being included when the kernel is compiled; this reduces memory footprint.
`kerneld` (daemon) and `kmod` (kernel thread) facilitate the dynamic loading of kernel modules.

`/lib/modules/X.Y.Z/*.ko`	Kernel modules for kernel version *X.Y.Z*
`/lib/modules/X.Y.Z/modules.dep`	Modules dependencies. This file needs to be recreated (via the command `depmod -a`) after a reboot or a change in module dependencies
`/etc/modules.conf` `/etc/conf.modules` (deprecated)	Modules configuration file
`/usr/src/linux/`	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]`	Create a `initrd` image file (Red Hat)
`mkinitramfs`	Create a `initrd` image file according to the configuration file `/etc/initramfs-tools/initramfs.conf` (Debian)
`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

`lsdev`	List information about the system's hardware
`lspci`	List PCI devices
`lspci -d 8086:`	List all Intel hardware present. PCI IDs are stored in `/usr/share/misc/pci.ids` (Debian) or `/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`
`dmesg`	Print the logs of the kernel ring buffer
`dmesg -n 1`	Set the logging level to 1 (= only panic messages)
`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

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`	To remove a patch, you can either apply the patch again or use this command (reverse patch)
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`/`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 pseudo filesystem		
File	**Information stored**	**Equivalent command to `cat`**
`/proc/n/`	Information about process with PID *n*	`ps n`
`/proc/n/cmdline`	Command line the process was launched by	
`/proc/n/environ`	Values of environment variables of process	
`/proc/n/status`	Status of process	
`/proc/n/root`	Symlink to process' filesystem root	
`/proc/n/exe`	Symlink to process' executable	
`/proc/n/cwd`	Symlink to process' working directory	
`/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/filesystems`	Filesystems supported by the system	
`/proc/partitions`	Drive partition information	
`/proc/mdstat`	Information about RAID arrays and devices	
`/proc/swaps`	Size of total and used swap areas	`swapon -s`
`/proc/mounts`	Mounted partitions	`mount`
`/proc/devices`	Drivers currently loaded	
`/proc/modules`	Kernel modules currently loaded	`lsmod`
`/proc/bus`	Buses (e.g. PCI, USB, PC Card)	
`/proc/ioports`	I/O addresses in use	
`/proc/dma`	DMA channels in use	
`/proc/interrupts`	Current interrupts	
`/proc/cpuinfo`	CPUs information	
`/proc/meminfo`	Total and free memory	`free`
`/proc/version`	Linux version	`uname -a`

`/proc/sys` is the only writable branch of `/proc` and can be used to tune kernel parameters on-the-fly.
All changes will be lost after system shutdown.

```
sysctl fs.file-max
cat /proc/sys/fs/file-max
```

Get the maximum allowed number of open files

```
sysctl -w "fs.file-max=100000"
echo "100000" > /proc/sys/fs/file-max
```

Set the maximum allowed number of open files to 100000

```
sysctl -a
```

List all available kernel tuning options

```
sysctl -p
```

Apply all tuning settings listed in `/etc/sysctl.conf` .
This command is usually run at boot by the system initialization script and therefore allows permanent changes to the kernel

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 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
```

An alternative metod is to `chroot /mnt/sysimage` before installing GRUB via `grub-install /dev/sda` .

Run `sync` and unmount filesystems before exiting the shell, to ensure that all changes have been written on disk.

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		

Methods of MPM (Multi-Processing Modules) operation of the Apache webserver:

prefork MPM	A number of child processes is spawned in advance, with each child serving exclusively 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

`apache2ctl start`	Start the Apache webserver daemon `httpd`
`apache2ctl status`	Display a brief status report
`apache2ctl fullstatus`	Display a detailed status report
`apache2ctl graceful`	Gracefully restart Apache; currently open connections are not aborted
`apache2ctl graceful-stop`	Gracefully stop Apache; currently open connections are not aborted
`apache2ctl configtest`	Test the configuration file, reporting any syntax error

`/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, not the Apache user/group.

`/etc/httpd/conf/httpd.conf`	(Red Hat)	Apache configuration file
`/etc/apache2/httpd.conf`	(Debian & SUSE)	

httpd.conf	
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 config 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` (after v2.3.13)	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

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"`	Organize 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	
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 an user and password
`AuthType Basic`	Type of user authentication: `Basic`, `Digest`, `Form`, or `None`
`AuthUserFile "/var/www/.htpasswd"`	User database file. Each line is in the format *`user`*`:`*`encrypted_password`* To add an user jdoe to the database file, use the command: `htpasswd -c /var/www/.htpasswd jdoe` (will prompt for his password)
`AuthGroupFile "/var/www/.htgroup"`	Group database file. Each line contains a groupname followed by all member usernames: `mygroup: jdoe ksmith mgreen`
`Require valid-user`	Control who can access the protected resource. `valid-user` any user in the user database file `user jdoe` only the specified user `group mygroup` 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>`	

A secure web server (using HTTP over SSL i.e. HTTPS) hands over its public key to the client when the latter connects to it 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, self-signed certificates.

Virtual hosting with HTTPS requires assigning an 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 which virtual host the client wants to talk to).
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 e.g. for a wildcard domain *.example.org.

`/etc/ssl/openssl.cnf`		Configuration file for OpenSSL
`/etc/httpd/conf.d/ssl.conf`	(Red Hat)	Configuration file for the mod_ssl module

httpd.conf

SSL/TLS directives (module mod_ssl)

`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 should be readable only by root)
`SSLCACertificatePath \` `/usr/local/apache2/conf/ssl.crt/`	Directory containing the certificates of CAs. Files in this directory are PEM-encoded and accessed via symlinks to hash filenames
`SSLCACertificateFile \` `/usr/local/apache2/conf/ssl.crt/ca-bundle.crt`	Certificates of CAs. Certificates are PEM-encoded and concatenated in a single bundle file in order of preference
`SSLCertificateChainFile \` `/usr/local/apache2/conf/ssl.crt/ca.crt`	Certificate chain of the CAs. Certificates are PEM-encoded and concatenated from the issuing CA certificate of the server certificate to the root CA certificate. Optional
`SSLEngine on`	Enable the SSL/TLS Protocol Engine

`SSLProtocol +SSLv3 +TLSv1.2`	SSL protocol flavors that the client can use to connect to server. Possible values are:
	`SSLv2` (deprecated)
	`SSLv3`
	`TLSv1`
	`TLSv1.1`
	`TLSv1.2`
	`All` (all the above protocols)

`SSLCipherSuite \` `ALL:!aDH:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP`	Cipher suite available for the SSL handshake (key exchange algorithms, authentication algorithms, cipher/encryption algorithms, MAC digest algorithms)

`ServerTokens Full`	Server response header field to send back to client. Possible values are:
	`Prod` sends `Server: Apache`
	`Major` sends `Server: Apache/2`
	`Minor` sends `Server: Apache/2.4`
	`Minimal` sends `Server: Apache/2.4.2`
	`OS` sends `Server: Apache/2.4.2 (Unix)`
	`Full` (or not specified) sends
	`Server: Apache/2.4.2 (Unix) PHP/4.2.2 MyMod/1.2`

`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 has not much purpose and is used only for SSL testing)
`TraceEnable on`	Enable TRACE requests

```
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 ca -config ca.conf -in request.csr \
-out certif.cer -days validity -verbose
```
Sign a CSR (to generate a self-signed certificate, the steps are creating a CSR and signing it)

```
openssl ca -config ca.conf -gencrl -revoke certif.cer \
-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 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 (if there is no output, then 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 key.pem
```
Generate a 2048-bit RSA key pair protected by TripleDES passphrase

```
openssl genrsa -des3 -out key.pem 2048
```
Generate a 2048-bit RSA key pair protected by TripleDES passphrase (older versions of OpenSSL)

```
openssl pkey -text -in private.key -noout
```
Examine a private key

```
openssl rsa -text -in private.key -noout
```
Examine a private key (older versions of OpenSSL)

```
openssl pkey -in old.key -out new.key -cipher
```
Change a private key's passphrase

```
openssl rsa -in old.key -out new.key -cipher
```
Change a private key's passphrase (older versions of OpenSSL)

```
openssl s_client -connect www.website.com:443 > tmpfile
```
CTRL C
```
openssl x509 -in tmpfile -text
```
Retrieve and inspect a SSL certificate from a website

```
openssl list-message-digest-commands
```
List all available hash functions

```
openssl list-cipher-commands
```
List all available ciphers

`CA.pl -newca`	Create a Certification Authority hierarchy
`CA.pl -newreq`	Generate a Certificate Signing Request
`CA.pl -signreq`	Sign a Certificate Signing Request
`CA.pl -pkcs12 "My certificate"`	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 an unencrypted private key (necessary for servers as the private key must be accessed)
`CA.pl -verify`	Verify a certificate against the Certification Authority certificate for "demoCA"

Samba is a cross-platform implementation of Microsoft's SMB (Server Message Block) protocol for file and printer sharing.
SMB is sometimes also referred to as CIFS (Common Internet File System).
WINS (Windows Internet Name Service) is a name service used to translate NetBIOS names to IP addresses.

Ports used:		
	TCP 137	name service requests and responses
	TCP 138	datagram services e.g. server announcements
	TCP 139	file and printer sharing
	UDP	registration and translation of NetBIOS names, network browsing

`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/lmhosts`	Samba NetBIOS hosts file
`/etc/smb/netlogon`	User logon directory

`mount.cifs //smbserver/share1 /mnt/shares/sh1 \`
`-o auto,credentials=/etc/smbcreds`
	Mount a Samba share on a Linux filesystem, using the CIFS filesystem interface.
	Access is checked upon a credentials file `/etc/smbcreds` (should be readable only by root) formatted as follows:
	`username = jdoe`
	`password = jd03s3cr3t`

`smbmount //smbserver/share1 /mnt/shares/sh1 \` `-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)
`smbpasswd jdoe`	Change the Samba password of the specified user
`smbpasswd -a ksmith`	Create a new Samba user and set his password
`nmblookup smbserver`	Look up the NetBIOS name of a server and map it to an IP address
`nmblookup -U winsserver -R WORKGROUP#1B`	Query recursively a WINS server for the Domain Master Browser for the specified workgroup
`nmblookup -U winsserver -R WORKGROUP#1D`	Query recursively a WINS server for the Domain Controller for the specified workgroup
`testparm`	Check for errors in the Samba configuration file
`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 service 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

/etc/smb/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/smb/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/smb/smbusers`	Map each Samba server user name to client user name(s).
	The file `/etc/smb/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`
`path = /home/netlogon`
`browseable = no`
`guest ok = no`
`writeable = no`
`logon script = %U.bat`
	Section defining a logon script.
	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.
	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/smb/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]`
`path = /foobar`
`comment = Share Foobar on %L from %m`
`browsable = yes`
`writeable = yes`
	Section defining a specific 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

Samba share access	
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`	Set up share-level authentication

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

Samba macros	
%S	Username
%U	Session username (the username that the client requested, not necessarily the same as the one he got)
%G	Primary group of session username
%h	Samba server hostname
%M	Client hostname
%L	NetBIOS name of the server
%m	NetBIOS name of the client
%d	Process ID of the current server process
%a	Architecture of remote machine
%I	IP address of client machine
%i	Local IP address to which a client connected
%T	Current date and time
%D	Domain or workgroup of the current user
%w	Winbind separator
%$(*var*)	Value of the environment variable *var*

The substitutes below apply only to the configuration options that are used when a connection has been established:	
%S	Name of the current service, if any
%P	Root directory of the current service, if any
%u	Username of the current service, if any
%g	Primary group name of username
%H	Home directory of username
%N	Name of the NIS home directory server as obtained from the NIS `auto.map` entry. Same as `%L` if Samba was not compiled with the `--with-automount` option
%p	Path of service's home directory as obtained from the NIS `auto.map` entry. The NIS `auto.map` entry is split up as `%N:%p`

A Network File System (NFS) server makes filesystems available to 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, synchronizes `/etc/exports` with `/var/lib/nfs/xtab` by removing those entries in `/var/lib/nfs/xtab` that are deleted from `/etc/exports`, and removes those entries from `/proc/fs/nfs/exports` that are no longer valid

`exportfs -ua`
Unexport all directories.
All entries listed in `/var/lib/nfs/xtab` are removed from `/proc/fs/nfs/exports`, and the file is cleared

`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

`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` :
`nfsserver:/share /usr nfs intr 0 0`

`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`

/etc/exports

```
/export/          10.3.3.3(rw)
/export/          *(ro,sync)
/home/ftp/pub     client1(rw)  *.example.org(ro)
/home/crew        @FOOBARWORKGROUP(rw)  (ro)
```

filesystem	Filesystem on the NFS server to be exported to clients	
client identity	Client systems allowed to access the exported directory. Can be identified 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)

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

```
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 an 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)
dc	dc=example,dc=org	Domain Component
cn	cn: John Doe	Common Name
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: Example Corporation	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.
Actually invokes the command `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@othercorp.org - 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

SSSD (the System Security Services Daemon) can be used to provide access to OpenLDAP as an authentication and identity provider.