Foreword

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

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

This document has been composed with Apache OpenOffice.

Happy Linux hacking,

Daniele Raffo

Version history

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

Bibliography and suggested readings

- Evi Nemeth et al., UNIX and Linux System Administration Handbook, O'Reilly
- Adam Haeder et al., LPIC Linux Certification in a Nutshell, O'Reilly
- Heinrich W. Klöpping et al., The LPIC-2 Exam Prep, http://lpic2.unix.nl
- Ellen Siever et al., Linux in a Nutshell, O'Reilly, http://archive.oreilly.com/linux/cmd
- Christoph Braun, Unix System Security Essentials, Addison-Wesley
- A-Z index of Bash command line, http://ss64.com/bash
- Shell command line snippets, http://www.commandlinefu.com
- Bash command line snippets, http://www.bashoneliners.com
- Regular expressions tester, http://www.regextester.com
- Bash pitfalls, http://mywiki.wooledge.org/BashPitfalls
Index

- LVM........................................... 1
- LVM commands............................ 2
- System boot.................................. 3
- SysV startup sequence.................... 4
- Login....................................... 5
- Runlevels................................... 6
- SysV service management.................. 7
- Systemd service management................ 8
- /etc/init.d................................. 9
- Filesystem Hierarchy Standard.......... 10
- Partitions and filesystem................ 11
- mount....................................... 12
- Filesystem types........................... 13
- Swap......................................... 14
- /etc/fstab................................. 15
- Filesystem operations..................... 16
- Filesystem maintenance.................... 17
- XFS, ReiserFS, CD-ROM fs................ 18
- AutoFS...................................... 19
- RAID......................................... 20
- Bootloader.................................. 21
- GRUB 2 configuration..................... 22
- GRUB 2 operations......................... 23
- GRUB Legacy................................... 24
- dpkg-apt..................................... 25
- rpm......................................... 26
- yum.......................................... 27
- yum repositories........................... 28
- Backup...................................... 29
- Archive formats........................... 30
- Documentation............................... 31
- Shell usage.................................. 32
- Text filters................................... 33
- Advanced text filters........................ 34
- Regular expressions........................ 35
- File management............................ 36
- Directory management..................... 37
- File operations.............................. 38
- I/O streams.................................. 39
- read and echo................................ 40
- Processes.................................... 41
- Signals...................................... 42
- Resource monitoring....................... 43
- vmstat....................................... 44
- free......................................... 45
- File permissions............................ 45
- File attributes................................ 47
- ACLs.......................................... 48
- Links......................................... 49
- Find system files........................... 50
- Shell variables............................. 51
- Shell operations............................ 52
- Shell options................................ 53
- Shell scripting............................. 54
- getopts...................................... 55
- Command execution........................ 56
- Tests......................................... 57
- Operators.................................... 58
- Flow control.................................. 59
- less.......................................... 60
- Vi commands................................... 62
- Vi options.................................... 63
- SQL.......................................... 64
- SQL SELECT................................. 65
- SQL JOIN..................................... 66
- MySQL........................................ 67
- MySQL tools.................................. 68
- MySQL syntax................................ 69
- MySQL status............................... 70
- MySQL recipes............................... 71
- MySQL operations........................... 72
- PostgreSQL................................. 73
- X............................................. 74
- X tools....................................... 75
- X keysim codes.............................. 76
- /etc/passwd................................... 77
- User management............................ 78
- Group management........................... 79
- UID and GID.................................. 80
- sudo and su.................................. 81
- Terminals.................................... 82
- Messaging.................................... 83
- cron......................................... 84
- at............................................. 85
- Utilities..................................... 86
- Localization.................................. 87
- System time................................... 88
- syslog....................................... 89
- E-mail........................................ 90
- SMTP......................................... 91
- Sendmail..................................... 92
- Exim.......................................... 93
- Postfix....................................... 94
- Postfix configuration...................... 95
- Procmail..................................... 96
- Courier POP configuration................ 97
- Courier IMAP configuration.............. 98
- Dovecot...................................... 99
- Dovecot mailbox configuration........... 100
- Dovecot POP/IMAP configuration......... 101
- Dovecot authentication.................... 102
- FTP........................................... 103
- vsftpd....................................... 104
- CUPS......................................... 105
- IP addressing............................... 106
- Subnetting................................... 107
- Network services............................ 108
- Network configuration commands........ 109
- Network configuration files.............. 110
- Red Hat vs Debian network config.... 111
- nmcli........................................ 112
- Teaming and bridging...................... 113
- Wireless networking....................... 114
- Network tools............................... 115
- Network monitoring....................... 116
- tcppedump................................... 117
- netcat....................................... 118
- TCP Wraper.................................. 119
- Routing...................................... 120
- iptables..................................... 121
- iptables rules............................... 122
- iptables NAT routing....................... 123
- firewall..................................... 124
- firewall rules............................... 125
- SSH.......................................... 126
- SSH tools.................................... 127
- SSH operations............................. 128
- SSH configuration......................... 129
- X.509........................................ 130
- OpenSSL...................................... 131
- CA.pl......................................... 132
- GnuPG........................................ 133
- OpenVPN...................................... 134
- Security tools.............................. 135
- Key bindings - terminal............... 136
- Key bindings - X Window................. 137
- udev........................................ 138
- Kernel........................................ 139
- Kernel management......................... 140
- Kernel compile and patching............... 141
- Kernel modules............................ 142
- /proc........................................ 143
- /dev........................................ 144
- System recovery............................ 145
- DNS.......................................... 146
- DNS configuration.......................... 147
- DNS zone file............................... 148
- HTTP response codes....................... 149
- Apache....................................... 150
- Apache server configuration............. 151
- Apache main configuration.............. 152
- Apache virtual hosts...................... 153
- Apache authorization...................... 154
- Apache SSL/TLS............................. 155
- Apache proxy................................ 156
- Tomcat....................................... 157
- Samba server................................. 158
- Samba client................................ 159
- Samba global configuration.............. 160
- Samba share configuration.............. 161
- Samba access configuration.............. 162
- Samba setup................................ 163
- NFS.......................................... 164
- /etc/exports................................. 165
- NFS setup.................................... 166
- iscsi......................................... 167
- iscsi setup................................. 168
- DHCP......................................... 169
- PAM.......................................... 170
- LDAP.......................................... 171
- Idamch....................................... 172
- OpenLDAP.................................... 173
- SELinux....................................... 174
- semanage.................................... 175
- sealert....................................... 176
- Kickstart.................................... 177
- Red Hat Satellite 6....................... 178
- KVM.......................................... 179
- Git............................................ 180
- Git search and configuration............. 181
- Vagrant...................................... 182
- Puppet........................................ 183
- Ansible...................................... 184
- HTML 4.01 components.................... 185
- HTML 4.01 text................................ 186
- HTML 4.01 images......................... 187
- HTML 4.01 tables........................... 188
- 7-bit ASCII table........................... 189
Logical Volume Management (LVM) introduces an abstraction between physical and logical storage, allowing a more versatile use of filesystems. LVM uses the Linux device mapper feature (/devmapper).

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

**How to create a Logical Volume**

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

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

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

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

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

**How to snapshot and backup a Logical Volume**

1. `lvcreate -s -L 1024M -n mysnaps /dev/myvg0/mylv`  
   Create the snapshot like a Logical Volume
2. `tar cvzf mysnaps.tar.gz mysnaps`  
   Backup the snapshot with any backup tool
3. `lvremove /dev/mvvg0/mysnap`  
   Delete the snapshot
<table>
<thead>
<tr>
<th>PV commands</th>
<th>VG commands</th>
<th>LV commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvs</td>
<td>vgs</td>
<td>lvs</td>
</tr>
<tr>
<td>pvs&lt;sup&gt;sc&lt;/sup&gt;an</td>
<td>vg&lt;sup&gt;sc&lt;/sup&gt;an</td>
<td>lv&lt;sup&gt;sc&lt;/sup&gt;an</td>
</tr>
<tr>
<td>pvdisplay</td>
<td>vgdisplay</td>
<td>lvdisplay</td>
</tr>
<tr>
<td>pvck</td>
<td>vgck</td>
<td>lv&lt;sup&gt;c&lt;/sup&gt;hange</td>
</tr>
<tr>
<td>pvcreate</td>
<td>vgcreate</td>
<td>lv&lt;sup&gt;c&lt;/sup&gt;reate</td>
</tr>
<tr>
<td>pvchange</td>
<td>vgchange</td>
<td>lv&lt;sup&gt;c&lt;/sup&gt;hange</td>
</tr>
<tr>
<td>pv&lt;sup&gt;r&lt;/sup&gt;move</td>
<td>vg&lt;sup&gt;r&lt;/sup&gt;move</td>
<td>lv&lt;sup&gt;r&lt;/sup&gt;move</td>
</tr>
<tr>
<td>pv&lt;sup&gt;r&lt;/sup&gt;size</td>
<td>vg&lt;sup&gt;r&lt;/sup&gt;merge</td>
<td>lv&lt;sup&gt;r&lt;/sup&gt;size</td>
</tr>
<tr>
<td>pvmove</td>
<td>vg&lt;sup&gt;+&lt;/sup&gt;merge</td>
<td></td>
</tr>
</tbody>
</table>

### LVM global commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dmsetup command</td>
<td>Perform low-level LVM operations</td>
</tr>
<tr>
<td>lvm command</td>
<td>Perform LVM operations. May also be used as an interactive tool</td>
</tr>
<tr>
<td>lvmsar</td>
<td>LVM system activity reporter. Unsupported on LVM2</td>
</tr>
<tr>
<td>lvmdiskscan</td>
<td>Scan the system for disks and partitions usable by LVM</td>
</tr>
<tr>
<td>lvmconfig</td>
<td>Show the current LVM disk configuration</td>
</tr>
</tbody>
</table>

### Filesystem Mapping

- `/dev/mapper/vgname-lvname`: Mapping of Logical Volumes in the filesystem
- `/dev/vgname/lvname`: Directory containing Volume Groups metadata backups
- `/etc/lvm/archive/`:
## System boot

<table>
<thead>
<tr>
<th>Boot sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>POST</strong> (Power-On Self Test)</td>
</tr>
<tr>
<td><strong>BIOS</strong> (Basic I/O System)</td>
</tr>
<tr>
<td><strong>Chain loader</strong> (GRand Unified Bootloader)</td>
</tr>
<tr>
<td><strong>Linux kernel</strong></td>
</tr>
<tr>
<td><strong>init</strong></td>
</tr>
<tr>
<td><strong>Startup</strong></td>
</tr>
<tr>
<td><strong>Login</strong></td>
</tr>
</tbody>
</table>

Newer systems use UEFI (Unified Extensible Firmware Interface) instead of BIOS. UEFI does not use the MBR boot code; it has knowledge of partition table and filesystems, and stores its application files required for launch in a EFI System Partition, mostly formatted as FAT32.

After the POST, the system loads the UEFI firmware which initializes the hardware required for booting, then reads its Boot Manager data to determine which UEFI application to launch. The launched UEFI application may then launch another application, e.g. the kernel and `initramfs` in case of a boot loader like GRUB.

Information about the boot process can be found in the manpages `man 7 boot` and `man 7 bootup`. 
SysV startup sequence

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

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   18 Nov 28 14:52 S07iscsid -> ../init.d/iscsid
lrwxrwxrwx. 1 root root   18 Nov 28 14:42 S08iptables -> ../init.d/iptables
```

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

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

To add or remove services at boot sequence:

- `update-rc.d service defaults`
- `update-rc.d -f service remove`
- `chkconfig --add service`
- `chkconfig --del service`

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

**Supported service operations**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>Start the service</td>
</tr>
<tr>
<td>stop</td>
<td>Stop the service</td>
</tr>
<tr>
<td>restart</td>
<td>Restart the service (stop, then start)</td>
</tr>
<tr>
<td>status</td>
<td>Display daemon PID and execution status</td>
</tr>
<tr>
<td>force-reload</td>
<td>Reload configuration if service supports it, otherwise restart</td>
</tr>
<tr>
<td>condrestart</td>
<td>Restart the service only if already running</td>
</tr>
<tr>
<td>try-restart</td>
<td>Reload the service configuration</td>
</tr>
</tbody>
</table>

**Linux Standard Base (LSB)**

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

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

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

```bash
# chkconfig: 2345 85 15
# description: Foo service
```
Start the specified number of terminals at bootup via getty, which manages physical or virtual terminals (TTYs)

Control appearance and functioning of the system during bootup

Randomly-generated machine ID. The machine ID can be safely regenerated by deleting this file and then running the command `systemd-machine-id-setup`

List of TTYs from which the root user is allowed to login

Message printed before the login prompt. Can contain these escape codes:

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

Message printed before the login prompt on a remote session

Message Of The Day, printed after a successful login, but before execution of the login shell

If this file exists, `login` and `sshd` deny login to all unprivileged users. Useful when doing system maintenance

Logfile containing user logins (both successful and failed) and authentication mechanisms

Logfile containing failed authentication attempts

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

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

Print the list of users logged into the system

Print the list of users logged into the system, and what they are doing

Print the list of users that logged in and out. Searches through the file `/var/log/wtmp`

Print the list of bad login attempts. Searches through the file `/var/log/btmp`

Temporarily ban IP addresses (via firewall rules) that have too many failed password logins. This information is taken from authentication logs

Deny access to users that have too many failed logins

Turn process accounting on or off

Print statistics about connect time of users

Print information about previously executed commands

Print summarized information about previously executed commands
### Runlevels

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

#### Systemd’s target

*runleveln.target* **emulates a SysV’s runlevel n.**

- `runlevel`
- `who -r`
- `init runlevel`
- `telinit runlevel`
- `systemctl get-default`
- `systemctl set-default target`
- `systemctl isolate target`
- `systemctl emergency`
- `systemctl rescue`
- `systemctl -t target`
- `init 0`
- `telinit 0`
- `shutdown -h now`
- `halt`
- `poweroff`
- `systemctl isolate shutdown.target`

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

- `shutdown`
- `shutdown -a`
- `shutdown -h 16:00 message`
- `shutdown -f`
- `shutdown -F`
- `shutdown -c`

- **Display the previous and the current runlevel**
- **Change to runlevel**
- **Get the default target**
- **Set target as the default target**
- **Change to target**
- **Change to maintenance single-user mode with only /root filesystem mounted**
- **Change to maintenance single-user mode with only local filesystems mounted**
- **List targets**
- **Halt the system**
- **Reboot the system**
- **Shut down the system in a secure way: all logged-in users are notified via a message to their terminal, and login is disabled. Can only be run by the root user**
- **Non-root users that are listed in /etc/shutdown.allow can use this command to shut down the system**
- **Schedule a shutdown for 4 PM and send a warning message to all logged-in users**
- **Skip fsck on reboot**
- **Force fsck on reboot**
- **Cancel a shutdown that has been already initiated**
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/etc/init.d/service operation</code></td>
<td>(Red Hat) Perform the specified operation (start, stop, status, etc.) on the specified service</td>
</tr>
<tr>
<td><code>rcservice operation</code></td>
<td>(SUSE)</td>
</tr>
<tr>
<td><code>update-rc.d service defaults</code></td>
<td>(Debian) Add a service at boot</td>
</tr>
<tr>
<td><code>chkconfig --add service</code></td>
<td>(Red Hat)</td>
</tr>
<tr>
<td><code>update-rc.d -f service remove</code></td>
<td>(Debian) Remove a service at boot</td>
</tr>
<tr>
<td><code>chkconfig --del service</code></td>
<td>(Red Hat)</td>
</tr>
<tr>
<td><code>update-rc.d -f service \start 30 2 3 4 5 . stop 70 0 1 6 .</code></td>
<td>Add a service on the default runlevels; creates S30 symlinks for starting the service and K70 symlinks for stopping it</td>
</tr>
<tr>
<td><code>chkconfig --levels 245 service on</code></td>
<td>Add the service on runlevels 2 4 5</td>
</tr>
<tr>
<td><code>chkconfig service on</code></td>
<td>Add the service on default runlevels</td>
</tr>
<tr>
<td><code>chkconfig service off</code></td>
<td>Remove the service on default runlevels</td>
</tr>
<tr>
<td><code>chkconfig service reset</code></td>
<td>Check if the service is enabled on the current runlevel</td>
</tr>
<tr>
<td><code>chkconfig service resetpriorities</code></td>
<td>Reset the on/off state of the service for all runlevels to whatever the LSB specifies in the init script</td>
</tr>
<tr>
<td><code>chkconfig --list service</code></td>
<td>Display current configuration of service (its status and the runlevels in which it is active)</td>
</tr>
<tr>
<td><code>chkconfig</code></td>
<td></td>
</tr>
<tr>
<td><code>chkconfig --list</code></td>
<td>List all active services and their current configuration</td>
</tr>
<tr>
<td><code>ls /etc/rcn.d</code></td>
<td>(Debian) List services started on runlevel \n</td>
</tr>
</tbody>
</table>
### Systemd service management

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>systemctl operation service</code></td>
<td>Perform the specified operation (start, stop, status, etc.) on the specified service (unit file)</td>
</tr>
<tr>
<td><code>systemctl enable service</code></td>
<td>Add the service on the current target</td>
</tr>
<tr>
<td><code>systemctl disable service</code></td>
<td>Remove the service on the current target</td>
</tr>
<tr>
<td><code>systemctl is-enabled service</code></td>
<td>Check if the service is enabled on the current target</td>
</tr>
<tr>
<td><code>systemctl mask service</code></td>
<td>Mask the service on the current target. This prevents the service to be enabled or started</td>
</tr>
<tr>
<td><code>systemctl unmask service</code></td>
<td>Unmask the service on the current target</td>
</tr>
<tr>
<td><code>systemctl list-unit-files --type=service</code></td>
<td>List all active services and their current configuration</td>
</tr>
<tr>
<td><code>systemctl</code></td>
<td>List loaded and active units</td>
</tr>
<tr>
<td><code>systemctl --all</code></td>
<td>List all units, including inactive ones</td>
</tr>
</tbody>
</table>
# 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.
10:0:wait:/etc/init.d/rc 0
11:1:wait:/etc/init.d/rc 1
12:2:wait:/etc/init.d/rc 2
13:3:wait:/etc/init.d/rc 3
14:4:wait:/etc/init.d/rc 4
15:5:wait:/etc/init.d/rc 5
16:6: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/initab 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.

<table>
<thead>
<tr>
<th>id</th>
<th>1-4 characters, uniquely identifies an entry. For gettys and other login processes it should be equal to the suffix of the corresponding tty</th>
</tr>
</thead>
<tbody>
<tr>
<td>runlevels</td>
<td>Runlevels for which the specified action must be performed. If empty, action is performed on all runlevels</td>
</tr>
<tr>
<td>action</td>
<td>Process to execute. If prepended by a +, utmp and wtmp accounting will not be performed</td>
</tr>
</tbody>
</table>

### Action
- **respawn**: Process will be restarted when it terminates
- **wait**: Process is executed at system boot. Runlevels field is ignored
- **once**: Process is executed once at the specified runlevel
- **boot**: Process is executed when an on-demand runlevel (A, B, C) is called
- **bootwait**: Process is executed when an on-demand runlevel (A, B, C) is called
- **off**: Process is executed at system boot, before any boot or bootwait entries. Runlevels field is ignored
- **ondemand**: Process is executed at system boot, before any boot or bootwait entries. Runlevels field is ignored
- **inittab**: Process is executed when init receives a SIGINT via **CTRL**, **ALT**, **DEL**
- **sysinit**: Process is executed when power goes down and an UPS kicks in. init will not wait for its termination
- **powerfail**: Process is executed when power goes down and an UPS kicks in. init will 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 a special key combination is pressed on console
- **kbdrequest**: Process is executed when init receives a SIGINT via **CTRL**, **ALT**, **DEL**
### Filesystem Hierarchy Standard (FHS)

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

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

The superblock contains information relative to the filesystem e.g. filesystem type, size, status, metadata structures. The Master Boot Record (MBR) is a 512-byte program located in the first sector of the hard disk; it contains information about hard disk partitions and has the duty of loading the OS. On recent systems, the MBR has been replaced by the GUID Partition Table (GPT).

Almost all modern filesystems use journaling; in a journaling filesystem, the journal logs changes before committing them to the filesystem, which ensures faster recovery and less risk of corruption in case of a crash.

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

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

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

- fdisk /dev/sda
  
- fdisk -l /dev/sda
  
- parted
  
- sfdisk /dev/sda
  
- cfdisk
  
- gparted
  
- gnome-disks
  
- partprobe device
  
- hdparm -z device

- mkfs -t fstype device

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

- mkfs -t ext2 /dev/sda
  
- mke2fs /dev/sda
  
- mkfs.ext2 /dev/sda

- mke2fs -j /dev/sda
  
- mkfs.ext3 /dev/sda
  
- mke3fs /dev/sda

- mkfs -t msdos /dev/sda
  
- mkdosfs /dev/sda
  
- Create an ext2 filesystem on /dev/sda
  
- Create an ext3 filesystem (ext2 with journaling) on /dev/sda
  
- Create a MS-DOS filesystem on /dev/sda
Display the currently mounted filesystems.
The commands `mount` and `umount` maintain in `/etc/mtab` a database of currently mounted filesystems, but `/proc/mounts` is authoritative.

`mount -a` Mount all devices listed in `/etc/fstab`, except those indicated as `noauto`.

`mount -t ext3 /dev/sda /mnt` Mount a Linux-formatted disk. The mount point (directory) must exist.

`mount -t msdos /dev/fd0 /mnt` Mount a MS-DOS filesystem floppy disk to mount point `/mnt`.

`mount -o nolock 10.7.7.7:/export /mnt/nfs` Mount a NFS share without running NFS daemons. Useful during system recovery.

`mount -t iso9660 -o ro,loop=/dev/loop0 cd.img /mnt/cdrom` Mount a CD-ROM ISO9660 image file like a CD-ROM (via the loop device).

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

`umount /mnt` Unmount the floppy disk as soon as it is not in use anymore.

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

`eject /mnt` Eject a removable media device.

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

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

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

`blkid /dev/sda1` Print the UUID of the specified partition.

`blkid -L /boot` Print the UUID of the specified partition, given its label.

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

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

`findfs LABEL=/boot` Print the name of the specified partition, given its label.

`e2label /dev/sda1` Print the label of the specified partition.
The command `sfdisk -T` prints the above list of partition IDs and names.

### Most used Linux-supported filesystems

- **ext2**: The oldest Linux ext filesystem, without journaling
- **ext3**: ext2 with journaling
- **ext4**: Linux journaling filesystem, an upgrade from ext3
- **Reiserfs**: Journaling filesystem
- **XFS**: Journaling filesystem, developed by SGI
- **JFS**: Journaling filesystem, developed by IBM
- **Btrfs**: B-tree filesystem, developed by Oracle
- **msdos**: DOS filesystem, supporting only 8-char filenames
- **umsdos**: Extended DOS filesystem used by Linux, compatible with DOS
- **fat32**: MS-Windows FAT filesystem
- **vfat**: Extended DOS filesystem, with support for long filenames
- **ntfs**: Replacement for fat32 and vfat filesystems
- **minix**: Native filesystem of the MINIX OS
- **iso9660**: CD-ROM filesystem
- **cramfs**: Compressed RAM disk
- **nfs**: Network filesystem, used to access files on remote machines
- **SMB**: Server Message Block, used to mount Windows network shares
- **proc**: Pseudo filesystem, used as an interface to kernel data structures
- **swap**: Pseudo filesystem, Linux swap area
The **swap** space is an area on disk (a file or a partition) used as a RAM extension. When there is not enough free physical RAM for a process, inactive pages in memory are temporarily swapped out of memory to disk, to later be swapped in to memory when RAM resources are available again. If both RAM and swap space become nearly full, the system may get clogged by spending all the time paging blocks of memory back and forth between RAM and swap (thrashing). The amount of RAM plus the swap is defined as the **virtual memory**.

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

A swap partition can be created via any partitioning tool e.g. `fdisk`.

```
  dd if=/dev/zero of=/swapfile bs=1024 count=512000
  Create a 512-Mb swap file

  mkswap /swapfile
  Initialize a (already created) swap file or partition

  swapon /swapfile
  Enable a swap file or partition, thus telling the kernel that it can use it now

  swapoff /swapfile
  Disable a swap file or partition

  swapon -s
  Show the sizes of total and used swap areas

  cat /proc/swaps
  cat /proc/meminfo
  free
  top
```

---

**How to extend a LVM swap partition**

1. `lvs`
   Determine the name of the swap Logical Volume

2. `swapoff /dev/volgroup0/swap_lv`
   Turn off the swap volume

3. `lvresize -L+1G /dev/volgroup0/swap_lv`
   Extend the swap volume with an additional 1 Gb of space

4. `mkswap /dev/volgroup0/swap_lv`
   Format the swap volume

5. `swapon /dev/volgroup0/swap_lv`
   Turn on the swap volume
**/etc/fstab**

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>Mount Point</th>
<th>Type</th>
<th>Options</th>
<th>Dump</th>
<th>Pass</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/sda2</td>
<td>/</td>
<td>ext2</td>
<td>defaults</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/dev/sdb1</td>
<td>/home</td>
<td>ext2</td>
<td>defaults</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/dev/cdrom</td>
<td>/media/cdrom</td>
<td>auto</td>
<td>ro,noauto,user,exec</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/dev/fd0</td>
<td>/media/floppy</td>
<td>auto</td>
<td>rw,noauto,user,exec</td>
<td></td>
<td></td>
</tr>
<tr>
<td>proc</td>
<td>/proc</td>
<td>proc</td>
<td>defaults</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/dev/hda1</td>
<td>swap</td>
<td>swap</td>
<td>pri=42</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nfsserver:/dirs</td>
<td>/mnt</td>
<td>nfs</td>
<td>intr</td>
<td></td>
<td></td>
</tr>
<tr>
<td>//smbserver/jdoe/shares/jdoe</td>
<td>/shares/jdoe</td>
<td>cifs</td>
<td>auto,credentials=/etc/smbcreds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LABEL=/boot</td>
<td>/boot</td>
<td>ext2</td>
<td>defaults</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UUID=652b786e-b87f-49d2-af23-8087ced0c667</td>
<td>/test</td>
<td>ext4</td>
<td>errors=remount-ro,noatime</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

/\etc/fstab\ contains information about filesystems, including all filesystems that must be automatically mounted at bootup.

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

- **dump**: Options for the `dump` backup utility. 0 = do not backup
- **pass**: Order in which the filesystem must be checked by `fsck`. 0 = do not check
Filesysten operations

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>df</td>
<td>Report filesystem disk space usage</td>
</tr>
<tr>
<td>df -h</td>
<td>Report filesystem disk space usage in human-readable output</td>
</tr>
<tr>
<td>df directory</td>
<td>Shows on which device the specified directory is mounted</td>
</tr>
<tr>
<td>du directory</td>
<td>Report disk usage, as the size of each file contained in directory, in Kb</td>
</tr>
<tr>
<td>du -s directory</td>
<td>Show the total sum of the sizes of all files contained in directory</td>
</tr>
<tr>
<td>du -h directory</td>
<td>Report disk usage in human-readable output</td>
</tr>
<tr>
<td>du -hs *</td>
<td>sort -hr</td>
</tr>
<tr>
<td>du -a /path</td>
<td>sort -nr</td>
</tr>
<tr>
<td>find /path -type f -exec du -Sh {} + \</td>
<td>sort -hr</td>
</tr>
<tr>
<td>ncdu</td>
<td>Disk usage analyzer with Ncurses UI</td>
</tr>
<tr>
<td>resize2fs options device size</td>
<td>Resize an ext2/ext3/ext4 filesystem</td>
</tr>
<tr>
<td>lsblk</td>
<td>List information about all available block devices</td>
</tr>
<tr>
<td>lsscsi</td>
<td>List information about all SCSI devices</td>
</tr>
<tr>
<td>sync</td>
<td>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</td>
</tr>
<tr>
<td>chroot /path/to/newrootdir command</td>
<td>Run a command in a chroot jail (i.e. in a new root directory). The command process will be unable to access files outside the chroot jail</td>
</tr>
<tr>
<td>chroot /mnt/sysimage</td>
<td>Start a shell with /mnt/sysimage as filesystem root. Useful during system recovery when the machine has been booted from a removable media; this device is defined as the filesystem root and often needs to be changed to perform operations on the machine</td>
</tr>
<tr>
<td>mknod /dev/sga</td>
<td>Create a directory allocating the proper inode. Useful if experiencing filesystem problems during system recovery</td>
</tr>
<tr>
<td>multipath options device</td>
<td>Detect and aggregate multiple I/O paths (SAN connections) to a device</td>
</tr>
<tr>
<td>hdparm</td>
<td>Get/set drive parameters for SATA/IDE devices</td>
</tr>
<tr>
<td>hdparm -g /dev/hda</td>
<td>Display drive geometry (cylinders, heads, sectors) of /dev/hda</td>
</tr>
<tr>
<td>hdparm -i /dev/hda</td>
<td>Display identification information for /dev/hda</td>
</tr>
<tr>
<td>hdparm -tT /dev/hda</td>
<td>Perform disk read benchmarks on the /dev/hda drive</td>
</tr>
<tr>
<td>hdparm -p 12 /dev/hda</td>
<td>Reprogram IDE interface chipset of /dev/hda to mode 4. Warning: using an unsupported mode can cause filesystem corruption</td>
</tr>
<tr>
<td>sdparm</td>
<td>Access drive parameters for SCSI devices</td>
</tr>
</tbody>
</table>
Filesystem maintenance

**fsck device**

Check and repair a Linux filesystem (which must be unmounted).

Corrupted files will be placed into the `/lost+found` directory of the partition.

The exit code returned is the sum of the following conditions:

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

Fsck is a wrapper utility for the actual filesystem-specific checker commands:

- `fsck.ext2` aka `e2fsck`
- `fsck.ext3` aka `e2fsck`
- `fsck.ext4` aka `e2fsck`
- `fsck.msdos`
- `fsck.vfat`
- `fsck.cramfs`

**fsck**

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`

Check an ext2 filesystem, running the `badblocks` command to mark all bad blocks and add them to the bad block inode so they will not be allocated to files or directories

**touch /forcefsck** (Red Hat)

Force a filesystem check after next reboot

**tune2fs options device**

Adjust tunable filesystem parameters on ext2/ext3/ext4 filesystems

**tune2fs -l** `/dev/sda1`

List the contents of the filesystem superblock

**tune2fs -j** `/dev/sda1`

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

**tune2fs -m 1** `/dev/sda1`

Reserve 1% of the partition size to privileged processes. This space (5% by default, but can be reduced on modern filesystems) is reserved to avoid filesystem fragmentation and to allow privileged processes to continue to run correctly when the partition is full

**tune2fs -C 7** `/dev/sda1`

Set the mount count of the filesystem to 7

**tune2fs -c 20** `/dev/sda1`

Set the filesystem to be checked by fsck after 20 mounts

**tune2fs -i 15d** `/dev/sda1`

Set the filesystem to be checked by fsck each 15 days

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

**dumpe2fs options device**

Dump ext2/ext3/ext4 filesystem information

**dumpe2fs -h** `/dev/sda1`

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

**dumpe2fs /dev/sda1 | grep -i superblock**

Display locations of superblock (primary and backup) of filesystem

**dumpe2fs -b** `/dev/sda1`

Display blocks that are marked as bad in the filesystem

**debugfs device**

Interactive ext2/ext3/ext4 filesystem debugger

**debugfs -w** `/dev/sda1`

Debug `/dev/sda1` in read-write mode (by default, debugfs accesses the device in read-only mode)

Many hard drives feature the **Self-Monitoring, Analysis and Reporting Technology (SMART)** whose purpose is to monitor the reliability of the drive, predict drive failures, and carry out different types of drive self-tests.

The `smartctl` 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, ReiserFS, CD-ROM fs

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

xfs_info /dev/sdal
Print XFS filesystem geometry

xfs_growfs -n /dev/sdal

xfs_check options device
Check XFS filesystem consistency

xfs_repair options device
Repair a damaged or corrupt XFS filesystem

xfsdump -v silent -f /dev/tape /
Dump the root of a XFS filesystem to tape, with the lowest verbosity.
Incremental and resumed dumps are stored in the inventory database
/var/lib/xfsdump/inventory

xfsrestore -f /dev/tape /
Restore a XFS filesystem from tape

xfsdump -J - / | xfsrestore -J - /new
Copy the contents of a XFS filesystem to another directory, without
updating the inventory database

reiserfstune options device
Adjust tunable filesystem parameters on ReiserFS filesystem

dbgeiserfs device
Interactive ReiserFS filesystem debugger

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

CD-ROM filesystems

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

CD-ROM filesystem extensions

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Rock Ridge</td>
<td>Contains the original file information (e.g. permissions, filename) for MS Windows 8.3 filenames</td>
</tr>
<tr>
<td>MS Joliet</td>
<td>Used to create more MS Windows friendly CD-ROMs</td>
</tr>
<tr>
<td>El Torito</td>
<td>Used to create bootable CD-ROMs</td>
</tr>
</tbody>
</table>
AutoFS is a client-side service that allows automounting of filesystems, even for nonprivileged users. AutoFS is composed of the *autofs* kernel module that monitors specific directories for attempts to access them; in this case, the kernel module signals the *automount* userspace daemon, which mounts the directory when it needs to be accessed and unmounts it when it is no longer accessed. Mounts managed by AutoFS should not be mounted/unmounted manually or via `/etc/fstab`, to avoid inconsistencies.

<table>
<thead>
<tr>
<th>AutoFS configuration files</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/etc/sysconfig/autofs</code></td>
<td>AutoFS configuration file.</td>
</tr>
<tr>
<td><code>/etc/auto.master</code></td>
<td>Master map file for AutoFS. Each line is an indirect map, and each map file stores the configuration for the automounting of the subdirectory. The <code>-hosts</code> map tells AutoFS to mount/unmount automatically any export from the NFS server <code>nfsserver</code> when the directory <code>/net/nfsserver/</code> is accessed.</td>
</tr>
<tr>
<td></td>
<td><code># mount point  map         options</code></td>
</tr>
<tr>
<td></td>
<td><code>/net  -hosts</code></td>
</tr>
<tr>
<td></td>
<td><code>/-    /etc/auto.direct</code></td>
</tr>
<tr>
<td></td>
<td><code>/misc  /etc/auto.misc</code></td>
</tr>
<tr>
<td></td>
<td><code>/home  /etc/auto.home      --timeout=60</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AutoFS map files</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/etc/auto.direct</code></td>
<td>Direct map file for automounting of a NFS share.</td>
</tr>
<tr>
<td></td>
<td><code># dir  filesystem</code></td>
</tr>
<tr>
<td></td>
<td><code>/mydir  nfsserver1.foo.org:/myshare</code></td>
</tr>
<tr>
<td><code>/etc/auto.misc</code></td>
<td>Indirect map file for automounting of directory <code>/misc</code>.</td>
</tr>
<tr>
<td></td>
<td><code># subdir options            filesystem</code></td>
</tr>
<tr>
<td></td>
<td><code>public  -ro,soft,intr       ftp.example.org:/pub</code></td>
</tr>
<tr>
<td></td>
<td><code>cd      -fstype=iso9660,ro,nosuid,nodev  :/dev/cdrom</code></td>
</tr>
<tr>
<td><code>/etc/auto.home</code></td>
<td>Indirect map file for automounting of directory <code>/home</code> on a NFS share. The <code>*</code> wildcard matches any subdirectory the system attempts to access, and the <code>&amp;</code> variable takes the value of the match.</td>
</tr>
<tr>
<td></td>
<td><code># subdir options            filesystem</code></td>
</tr>
<tr>
<td></td>
<td><code>*      -rw,soft,intr         nfsserver2.bar.org:/home/&amp;</code></td>
</tr>
</tbody>
</table>
### RAID levels

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

---

mdadm -C /dev/md0 -l 5 \ -n 3 /dev/sdb1 /dev/sdc1 /dev/sdd1 \ -x 1 /dev/sde1  
Create a RAID 5 array from three partitions and a spare. Partitions type must be set to 0xFD. Once the RAID device has been created, it must be formatted e.g. via mke2fs -j /dev/md0  

mdadm --manage /dev/md0 -f /dev/sdd1  
Mark a drive as faulty, before removing it  

mdadm --manage /dev/md0 -r /dev/sdd1  
Remove a drive from the RAID array. The faulty drive can now be physically removed  

mdadm --manage /dev/md0 -a /dev/sdd1  
Add a drive to the RAID array. To be run after the faulty drive has been physically replaced  

mdadm --misc -Q /dev/sdd1  
Display information about a device  

mdadm --misc -D /dev/md0  
Display detailed information about the RAID array  

mdadm --misc -o /dev/md0  
Mark the RAID array as readonly  

mdadm --misc -w /dev/md0  
Mark the RAID array as read & write  

/etc/mdadm.conf  
Configuration file for the mdadm command  

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

cat /proc/mdstat  
Display information about RAID arrays and devices
# Bootloader

## Non-GRUB bootloaders

<table>
<thead>
<tr>
<th>Bootloader</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LILO</strong></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><strong>SYSLINUX</strong></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>
</tbody>
</table>
| **ISOLINUX**| Able to boot from CD-ROM ISO 9660 filesystems. Used for Live CDs and bootable install CDs. The CD must contain the following files:  

- `isolinux/isolinux.bin` ISOLINUX image, from the SYSLINUX distro  
- `boot/isolinux/isolinux.cfg` ISOLINUX configuration  
- `images/` Floppy images to boot  

and can be burnt with the command:  

```
mkisofs -o output.iso -b isolinux/isolinux.bin -c isolinux/boot.cat 
-no-emul-boot -boot-load-size 4 -boot-info-table cd_root_dir
``` |
| **PXELINUX**| Able to boot from PXE (Pre-boot eXecution Environment). PXE uses DHCP or BOOTP to enable basic networking, then uses TFTP to download a bootstrap program that loads and configures the kernel. Used for Linux installations from a central server or network boot of diskless workstations. The boot TFTP server must contain the following files:  

- `/tftpboot/pxelinux.0` PXELINUX image, from the SYSLINUX distribution  
- `/tftpboot/pxelinux.cfg/*` Directory containing a configuration file for each machine. A machine with Ethernet MAC address 88:99:aa:bb:cc:dd and IP address 192.0.2.91 (C000025B in hexadecimal) will search for its configuration filename in this order:  

01-88-99-aa-bb-cc-dd  
C000025B  
C000025  
C00002  
C0000  
C000  
C00  
C  
default |
| **EXTLINUX**| General-purpose bootloader like LILO or GRUB. Now merged with SYSLINUX. |
GRUB (Grand Unified Bootloader) is the standard boot manager on Linux distributions. The latest version is GRUB 2; the older version is GRUB Legacy.

GRUB Stage 1 (446 bytes), as well as the partition table (64 bytes) and the boot signature (2 bytes), is stored in the 512-byte MBR. It then accesses the GRUB configuration and commands available on the filesystem, usually on `/boot/grub`.

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

### GRUB 2 configuration file

```bash
# 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
}
```

Common kernel parameters:

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

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

---

**GRUB 2 operations**

- `grub2-mkconfig -o /boot/grub2/grub.cfg` (BIOS) Regenerate GRUB configuration file
- `grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg` (EFI)
- `grub-install /dev/sda` Install GRUB on first SATA drive
- `grub` Access the GRUB shell
- `grub2-set-default 1` Set GRUB to automatically boot the second entry in the GRUB menu
- `grub2-editenv list` Display the current GRUB menu entry that is automatically booted

---

**/boot/grub/device.map**

This file can be created to map Linux device filenames to BIOS drives

```plaintext
(fd0)  /dev/fd0
(hd0)  /dev/hda
```
GRUB Legacy shell commands

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

GRUB Legacy configuration file

```
# Section 0: Linux boot
title Debian   # Menu item to show on GRUB bootmenu
croot /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
croot (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
ckernel /memdisk # boot a floppy disk image
initrd /floppy-img-7.7.7
```

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

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

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

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

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

Dselect

Package manager with text interface, front-end to dpkg. Obsolete

Aptitude

Package manager with Ncurses UI, front-end to apt

Synaptic

Package manager with Gtk+ UI, front-end to apt
rpm is the low-level package manager for Red Hat. It uses the RPM package format, which is cpio-compressed.

```
rpm -i package.rpm
rpm -i ftp://host/package.rpm
rpm -i http://host/package.rpm
rpm -e package
rpm -U package.rpm
rpm -F package.rpm
rpm -qa
rpm -qa --last
rpm -ql package
rpm -qpl package.rpm
rpm -qf file
rpm -V package
rpm -i package.src.rpm
rpm -ba package.spec
```

Install a package file
Remove a package
Upgrade a package (and remove old versions)
Upgrade a package (only if an old version is already installed)
List installed packages and their state
List installed packages and their installation date, from newest to oldest
List the content of an installed package
List the content of a package file
Show the package containing a specific file
Verify an installed package
Install a package source file
Compile a package source file

```
rpm2cpio package.rpm
creatererepo directory
```

Convert a RPM package to a cpio archive
Create an XML file of repository metadata from the set of RPMs contained in `directory`

pirut

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

yum install package
yum install package.rpm
yum localinstall package.rpm
yum remove package
yum update package
yum update
yum swap packageout packagein
yum list
yum list searchterm
yum list installed
yum list available
yum search searchterm
yum search all searchterm
yum deplist package
yum list package
yum info package
yum history
yum history list
yum history list n
yum history info n
yum history package package
yum history list package package
yum whatprovides file
yum cmd --disablerepo="*" --enablerepo="repo"

yum repolist
cat /etc/yum.repos.d/*.repo
yum clean all
rmdir /var/cache/yum

yumdownloader --resolve package
yumdownloader --urls package
yum-complete-transaction
repoquery --tree-requires package

Install a package
Install a package file
Remove a package
Upgrade an installed package
Upgrade all installed packages
Replace a package with another
List all installed and available packages
List installed and available packages matching the search term
List installed packages
List packages available for install
Search for packages that match the search term in the package name or summary
Search for packages that match the search term in the package name, summary, or description
Show package dependencies (recursively)
Show package records
Show information about a package
Show the installation history (installs, updates, etc.)
Show item n of the installation history
Show detailed information on item n of the installation history (begin and end times, packages altered, etc.)
Show the installation history about a package
Show which package provides a specific file
Execute the yum command but only considering a specific repository repo
Print list of available repositories
Delete temporary files for repositories

Download package and all its dependencies
Show URLs that would be downloaded
Try to complete unfinished or aborted package installations
Show a tree with all dependencies of package
Configuration of a Fedora repository (Red Hat)

```
[fedora]
name=Fedora $releasever - $basearch
baseurl=http://download.fedoraproject.org/pub/fedora/linux/releases/$releasever/Everything/$basearch/os/
        http://foo.org/linux/$releasever/$basearch/os/
        http://bar.org/linux/$releasever/$basearch/os/

enabled=1
gpgcheck=1
failovermethod=priority
metalink=https://mirrors.fedoraproject.org/metalink?repo=fedora-$releasever&arch=$basearch

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$releasever-$basearch
```

Repository ID
Repository name
List of URLs to the repository's repodata directory. Can be any of these types:
file:/// local file
file:// NFS
http:// HTTP
https:// HTTPS
ftp:// FTP

Whether this repository is enabled
Whether to perform a GPG signature check on the packages downloaded from this repository
Makes yum try the baseurls in the order they are listed. By default, if more than one baseurl is specified, yum chooses one randomly
URL to a metalink file that specifies the list of mirrors to use. Can be used with or in alternative to a baseurl
ASCII-armored GPG public key file of the repository

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

How to install a package on an offline machine

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

On the online machine:

1. Install the package and all its dependencies in a local directory
   
   mkdir /tmp/repo
   yum --downloadonly --downloaddir=/tmp/repo install package

2. Create a local yum repository
   
   createrepo /tmp/repo
   chown -R root:root /tmp/repo
   chmod -R 755 /tmp/repo

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

On the offline machine:

4. Create a yum repo file /etc/yum.repos.d/local.repo for the new repository
   
   [local]
   name=Local
   baseurl=file:///tmp/repo
   enabled=1
   gpgcheck=0
   protect=1

5. Install the package from the local repository
   
   yum install package
Backup

**dd**

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

```
dd if=/dev/sda of=/dev/sdb
```

Copy the content of one hard disk over another

```
cat /dev/sda > /dev/sdb
```

Generate the image file of a partition

```
dd if=/dev/sda1 of=sda1.img
```

Generate the image file of a partition

```
dd if=/dev/cdrom of=cdrom.iso bs=2048
```

Create an ISO file from a CD-ROM, using a block size transfer of 2 Kb

```
dd if=install.iso of=/dev/sdc bs=512k
```

Write an installation ISO file to a device (e.g. a USB thumb drive)

**ddrescue**

Tool for data recovery. Like `dd`, but with high tolerance for read errors

```
rsync
```

Tool for local and remote file synchronization. For all copies subsequent to the first, copies only the blocks that have changed, making it a very efficient backup solution in terms of speed and bandwidth

```
rsync -rv/home /tmp/bak
rsync -rv /home/ /tmp/bak/home
rsync -avz /home root@10.0.0.7:/backup/
```

Synchronize the content of the home directory with the temporary backup directory. Use recursion, compression, and verbosity

Synchronize the content of the home directory with the backup directory on the remote server, using SSH. Use archive mode (i.e. operates recursively and preserves owner, group, permissions, timestamps, and symlinks)

**burp**

Backup and restore program

<table>
<thead>
<tr>
<th>Devices</th>
<th>First SCSI tape device</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/st0</td>
<td>First SCSI tape device (no-rewind device file)</td>
</tr>
<tr>
<td>/dev/nst0</td>
<td></td>
</tr>
</tbody>
</table>

**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
mtx -f /dev/sg1 load 3
mtx -f /dev/sg1 unload
mtx -f /dev/sg1 transfer 3 4
mtx -f /dev/sg1 inventory
mtx -f /dev/sg1 inquiry
```

Display status of tape library

Load tape from slot 3 to drive 0

Unload tape from drive 0 to original slot

Transfer tape from slot 3 to slot 4

Force robot to rescan all slots and drives

Inquiry about SCSI media device (Medium Changer = tape library)
## Archive formats

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cpio</strong></td>
<td>Create a cpio archive of all files in the current directory.</td>
</tr>
<tr>
<td></td>
<td>Create a cpio archive of all users' home directories.</td>
</tr>
<tr>
<td></td>
<td>Extract all files, recreating the directory structure.</td>
</tr>
<tr>
<td></td>
<td>List the contents of a cpio archive file.</td>
</tr>
<tr>
<td>gzip</td>
<td>Compress a file with gzip.</td>
</tr>
<tr>
<td></td>
<td>Compress a file with gzip, leaving the original file into place.</td>
</tr>
<tr>
<td></td>
<td>Decompress a gzip-compressed file.</td>
</tr>
<tr>
<td></td>
<td>Test the integrity of a gzip-compressed file.</td>
</tr>
<tr>
<td></td>
<td>Read a gzip-compressed text file.</td>
</tr>
<tr>
<td></td>
<td>grep for a gzip-compressed text file.</td>
</tr>
<tr>
<td></td>
<td>less for a gzip-compressed text file.</td>
</tr>
<tr>
<td></td>
<td>more for a gzip-compressed text file.</td>
</tr>
<tr>
<td></td>
<td>Parallel, multicore-optimized gzip.</td>
</tr>
<tr>
<td>bzip2</td>
<td>Compress a file with bzip2.</td>
</tr>
<tr>
<td></td>
<td>Decompress a bzip2-compressed file.</td>
</tr>
<tr>
<td></td>
<td>Read a bzip2-compressed text file.</td>
</tr>
<tr>
<td>7-Zip</td>
<td>Create a 7-Zip archive (has the highest compression ratio).</td>
</tr>
<tr>
<td>xz</td>
<td>Compress a file with xz.</td>
</tr>
<tr>
<td></td>
<td>Decompress a xz-compressed file.</td>
</tr>
<tr>
<td></td>
<td>Read a xz-compressed file.</td>
</tr>
<tr>
<td>LZMA</td>
<td>Compress a file with LZMA.</td>
</tr>
<tr>
<td></td>
<td>Decompress a LZMA-compressed file.</td>
</tr>
<tr>
<td></td>
<td>Read a LZMA-compressed file.</td>
</tr>
<tr>
<td>rar</td>
<td>Create a RAR archive.</td>
</tr>
<tr>
<td></td>
<td>Extract a RAR archive.</td>
</tr>
<tr>
<td>tar</td>
<td>Create a tarred archive (bundles multiple files in a single one).</td>
</tr>
<tr>
<td></td>
<td>Create a tarred gzip-compressed archive.</td>
</tr>
<tr>
<td></td>
<td>Extract a tarred gzip-compressed archive.</td>
</tr>
<tr>
<td></td>
<td>Create a tarred bzip2-compressed archive.</td>
</tr>
<tr>
<td></td>
<td>Extract a tarred bzip2-compressed archive.</td>
</tr>
<tr>
<td></td>
<td>Create a tarred xz-compressed archive.</td>
</tr>
<tr>
<td></td>
<td>Extract a tarred xz-compressed archive.</td>
</tr>
<tr>
<td></td>
<td>List the contents of a tarred archive.</td>
</tr>
<tr>
<td>star</td>
<td>Create a star archive.</td>
</tr>
<tr>
<td></td>
<td>Extract a star archive.</td>
</tr>
</tbody>
</table>
man command

man n command

man man

man n intro

mandb

yum whatprovides /usr/share/man/mann/command.n.gz

yum install man-pages (Red Hat)

yum install man-db (Red Hat)

apropos keyword
man -k keyword

apropos -r regex
man -k regex

man -K regex

whatis command

info command

help
help command

Show the manpage for command

Show section n of the command manpage

Show information about manpages' sections:
1  - Executable programs or shell commands
2  - System calls (functions provided by the kernel)
3  - Library calls (functions within program libraries)
4  - Special files
5  - File formats and conventions
6  - Games
7  - Miscellaneous
8  - System administration commands (only for root)
9  - Kernel routines

Show an introduction to the contents of section n

Generate or refresh the search database for manpage entries. This must be done after installing new packages, in order to obtain results from apropos or man -k

Find which package provides section n of the command manpage

Install a large number of manpages from the Linux Documentation Project

Install various manpage commands and utilities

Show the commands whose manpage's short description matches the keyword. Inverse of the whatis command

Show the commands whose manpage's short description matches the regex

Show the commands whose manpage's full text matches the regex

Show the manpage's short description for a command

Show the Info documentation for a command

Show the list of available shell commands and functions

Show help about a shell command or function
Shell usage

history
Show the history of command lines executed up to this moment.
Commands prepended by a space will be executed but will not show up in the history.
After the user logs out from Bash, history is saved into ~/.bash_history

!n
Execute command number n in the command line history

history -c
Clear the command line history

history -d n
Delete command number n from the command line history

alias ls='ls -lap'
Set up an alias for the ls command

alias
Show defined aliases

unalias ls
Remove the alias for the ls command

\ls
Run the non-aliased version of the ls command

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

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

grep -- -i file
Search for the string "-i" in file

rn -- -rf
Delete a file called "-rf"

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

Text filters

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cut -d: -f3 file</code></td>
<td>Cut the lines of a file, considering <code>:</code> as the delimiter and printing only the 3rd field</td>
</tr>
<tr>
<td><code>cut -d: -f1 /etc/passwd</code></td>
<td>Print the list of local user accounts in the system</td>
</tr>
<tr>
<td><code>cut -c3-50 file</code></td>
<td>Print character 3 to 50 of each line of a file</td>
</tr>
<tr>
<td><code>sed 's/foo/bar/\' file</code></td>
<td>Stream Editor: Replace the first occurrence on a line of &quot;foo&quot; with &quot;bar&quot; in file, and print on stdout the result</td>
</tr>
<tr>
<td><code>sed -i 's/foo/bar/' file</code></td>
<td>Replace &quot;foo&quot; with &quot;bar&quot;, overwriting the results in file</td>
</tr>
<tr>
<td><code>sed 's/foo/bar/g' file</code></td>
<td>Replace all occurrences of &quot;foo&quot; with &quot;bar&quot;</td>
</tr>
<tr>
<td><code>sed -n '7,13p' file</code></td>
<td>Replace &quot;foo&quot; with the value of variable $var. The double quotes are necessary for variable expansion</td>
</tr>
<tr>
<td><code>sed 's/foo/$var/' file</code></td>
<td>Translate characters: Convert all lowercase into uppercase in a text file</td>
</tr>
<tr>
<td><code>tr a-z A-Z &lt; file</code></td>
<td>Delete all digits from a text file</td>
</tr>
<tr>
<td><code>tr \[:lower:\] \[:upper:\] &lt; file</code></td>
<td>Interpreter for the AWK programming language, designed for text processing and data extraction</td>
</tr>
<tr>
<td><code>awk</code></td>
<td>Print the lines of a file containing &quot;foo&quot;</td>
</tr>
<tr>
<td><code>grep foo file</code></td>
<td>Print the lines of a file not containing &quot;foo&quot;</td>
</tr>
<tr>
<td><code>grep -v foo file</code></td>
<td>Print the lines of a file containing &quot;foo&quot; or &quot;bar&quot;</td>
</tr>
<tr>
<td>`grep -E 'foo</td>
<td>bar' file`</td>
</tr>
<tr>
<td><code>grep -v -e foo -e bar file</code></td>
<td>Output appended data as the text file grows, printing only the lines containing &quot;foo&quot;</td>
</tr>
<tr>
<td><code>grep -E regex file</code></td>
<td>Run command with modified stdin, stdout, or stderr buffering</td>
</tr>
<tr>
<td><code>egrep regex file</code></td>
<td>Replace strings in a file</td>
</tr>
<tr>
<td><code>rpl oldstring newstring file</code></td>
<td>Correct and tidy up the markup of HTML, XHTML, and XML files</td>
</tr>
<tr>
<td><code>tidy</code></td>
<td>Validate the syntax of a JSON file</td>
</tr>
<tr>
<td><code>tidy -asxml -xml -indent -wrap 2000 \ -quiet --hide-comments yes file.xml</code></td>
<td>Pretty format a JSON file</td>
</tr>
<tr>
<td><code>json_verify &lt; file.json</code></td>
<td>Strip out comments from an XML file</td>
</tr>
<tr>
<td><code>json_reformat &lt; file.json</code></td>
<td>Show all printable character sequences at least 4-characters long that are contained in file</td>
</tr>
<tr>
<td><code>strings file</code></td>
<td>Show text and images from a MS Word document</td>
</tr>
<tr>
<td><code>antiword file.doc</code></td>
<td>Output plaintext from a MS Word document</td>
</tr>
<tr>
<td><code>catdoc file.doc</code></td>
<td></td>
</tr>
</tbody>
</table>
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-zA-Z]  Any of the characters in the specified range
[^abc]  Any character except those specified
*  Zero or more times the preceding regex
+  One or more times the preceding regex
?  Zero or one time the preceding regex
{5}  Exactly 5 times the preceding regex
{5,}  5 times or more the preceding regex
{5,10}  At most 10 times the preceding regex
{5,10}  Between 5 and 10 times the preceding regex
( )  Grouping, to be used for back-references. \1 expands to the 1st match, \2 to the 2nd, etc. until \9

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

- **cp file file2**  
  Copy a file

- **cp file dir/**  
  Copy a file to a directory

- **cp -ar /dir1/. /dir2/**  
  Copy a directory recursively

- **mv file file2**  
  Rename a file

- **mv file dir/**  
  Move a file to a directory

- **rm file**  
  Delete a file

- **pv file > file2**  
  Copy a file, monitoring the progress of data through a pipe

- **rename str1 str2 file**  
  Rename a file, replacing the first occurrence of string *str1* with *str2*

- **unlink file**  
  Remove the hard link to a file (equivalent to *rm*).

- **touch file**  
  Change access timestamp and modify timestamp of a file as now. If the file does not exist, it is created.

- **truncate -s size file**  
  Shrink or extend a file to the specified size. If the file is larger than the specified size, it is truncated; if the file is shorter, the extra space is filled with zeros.

- **mktemp**  
  Create a temporary file or directory, using *tmp.XXXXXXXXXX* as filename template.

- **fdupes dir**  
  Examines a directory for duplicate files in it. To consider files a duplicate, first compares file sizes and MD5 signatures, then compares the file contents byte-by-byte.

- **tmpwatch**  
  Remove files which have not been accessed for some time.

- **od file**  
  Dump a file into octal (or other formats).

- **hexdump options file**  
  Dump a file into hexadecimal (or other formats e.g. octal, decimal, ASCII).

- **xxd options file**  
  Convert a file from binary to hexadecimal, or vice versa.

### File-naming wildcards (globbing)

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Matches zero or more characters</td>
</tr>
<tr>
<td>?</td>
<td>Matches one character</td>
</tr>
<tr>
<td>[abc]</td>
<td>Matches a, b, or c</td>
</tr>
<tr>
<td>![abc]</td>
<td>Matches any character except a, b, or c</td>
</tr>
<tr>
<td>[a-z]</td>
<td>Matches any character between a and z</td>
</tr>
</tbody>
</table>

### Brace expansion

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp foo.[txt,bak]</td>
<td>Copy file &quot;foo.txt&quot; to &quot;foo.bak&quot;</td>
</tr>
<tr>
<td>touch foo_{a,b,c}</td>
<td>Create files &quot;foo_a&quot;, &quot;foo_b&quot;, &quot;foo_c&quot;</td>
</tr>
<tr>
<td>touch foo_{a..c}</td>
<td>Create files &quot;foo_a&quot;, &quot;foo_b&quot;, &quot;foo_c&quot;</td>
</tr>
</tbody>
</table>
### Directory management

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

### Bash directory shortcuts

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>Current directory</td>
</tr>
<tr>
<td>..</td>
<td>Parent directory</td>
</tr>
<tr>
<td>~</td>
<td>Home directory of current user</td>
</tr>
<tr>
<td>-user</td>
<td>Home directory of user</td>
</tr>
<tr>
<td>--</td>
<td>Previously used directory</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>stat file</code></td>
<td>Display file or filesystem status</td>
</tr>
<tr>
<td><code>stat -c %A file</code></td>
<td>Display file permissions</td>
</tr>
<tr>
<td><code>stat -c %s file</code></td>
<td>Display file size, in bytes</td>
</tr>
<tr>
<td><code>shred /dev/hda</code></td>
<td>Securely wipe the contents of a device</td>
</tr>
<tr>
<td><code>shred -u file</code></td>
<td>Securely delete a file</td>
</tr>
<tr>
<td><code>lsof</code></td>
<td>List all open files</td>
</tr>
<tr>
<td><code>lsof -u user</code></td>
<td>List all files currently open by <code>user</code></td>
</tr>
<tr>
<td><code>lsof -i</code></td>
<td>List open files and their sockets (equivalent to <code>netstat -ap</code>)</td>
</tr>
<tr>
<td><code>lsof -i :80</code></td>
<td>List connections of local processes on port 80</td>
</tr>
<tr>
<td><code>lsof -i@10.0.0.3</code></td>
<td>List connections of local processes to remote host 10.0.0.3</td>
</tr>
<tr>
<td><code>lsof -i@10.0.0.3:80</code></td>
<td>List connections of local processes to remote host 10.0.0.3 on port 80</td>
</tr>
<tr>
<td><code>lsof -c mysql</code></td>
<td>List all files opened by <code>mysql</code>, the MySQL daemon</td>
</tr>
<tr>
<td><code>lsof file</code></td>
<td>List all processes using a specific <code>file</code></td>
</tr>
<tr>
<td><code>lsof +L1</code></td>
<td>List open files with a link count of 0 i.e. that have been unlinked. These files are not accessible but take up disk space. A process holding such a file prevents the system from deleting it (thus freeing disk space), until the process is killed or restarted.</td>
</tr>
<tr>
<td><code>fuser</code></td>
<td>Show the name of processes using a specific file, directory, or socket</td>
</tr>
<tr>
<td><code>fuser -v file</code></td>
<td>Show the name of the process using <code>file</code></td>
</tr>
<tr>
<td><code>fuser -v -n tcp 443</code></td>
<td>Show the name of the process running on port 443</td>
</tr>
<tr>
<td><code>lslocks</code></td>
<td>List information about all currently held file locks</td>
</tr>
<tr>
<td><code>aide</code></td>
<td>Advanced Intrusion Detection Environment. HIDS tool that makes a snapshot of the filesystem state and records it in a database, to check integrity of files at a later time</td>
</tr>
</tbody>
</table>
In Linux, everything is (displayed as) a file. File descriptors are automatically associated to any process launched.

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Type</th>
<th>Default device</th>
<th>Device file</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Standard input (stdin)</td>
<td>Input text stream</td>
<td>Keyboard</td>
<td>/dev/stdin</td>
</tr>
<tr>
<td>1</td>
<td>Standard output (stdout)</td>
<td>Output text stream</td>
<td>Terminal</td>
<td>/dev/stdout</td>
</tr>
<tr>
<td>2</td>
<td>Standard error (stderr)</td>
<td>Output text stream</td>
<td>Terminal</td>
<td>/dev/stderr</td>
</tr>
</tbody>
</table>

mail user@email < file

Redirect file to the stdin of command mail (in this case, send via e-mail the contents of file to the email address user@email).

Redirection is handled by the shell, not by the command invoked. The space after the redirect operator is optional.

ls > file

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

ls >| file

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

ls >> file

Append the stdout of command ls to file.

ls 2> file

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

ls 2>> file

Append the stderr of command ls to file.

ls 2> /dev/null

Silence any error coming from the command ls.

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

Redirect file1 to the stdin and file2 to the stdout of the command cat (in this case, copy file1 to file2).

cat >file2 <file1 also works but is not recommended, because it truncates file2 if file1 cannot be opened.

cat /etc/passwd | wc -l

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

Piped commands run concurrently.

echo "$\{sort file\}" > file
echo "$\{sort file\}" > file

Sort the contents of file and write the output to the file itself.

sort file > file would not produce the desired result, because the stdout destination is created (and therefore the content of the preexisting file is deleted) before the sort command is run.

ls 2>${}

Redirect stderr of command ls to stdout.

ls > file 2>${}

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

> file

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

ls | tee file

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

ls | tee -a file

tee reads from stdin and appends both to stdout and file.
read MYVAR
read -n 8 MYVAR
read -t 60 MYVAR
read -s MYVAR

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

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

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

echo $MYVAR
echo -n "message"
printf "message"
echo -e '\a'

pv -qL10 <<< "message"
Any application, program, script, or service that runs on the system is a **process**. Processes whose parent is a shell are called **jobs**. **Signals** are used for inter-process communication. Each process has a unique PID (Process ID) and a PPID (Parent Process ID); when a process spawns a child, the process PID is assigned to the child’s PPID. The `/sbin/init` process, run at bootup, has PID 1. It is the ancestor of all processes and becomes the parent of any orphaned process. It is also unkillable; should it die, the kernel will panic. When a child process dies, its status becomes EXIT_ZOMBIE and a SIGCHLD is sent to the parent. The parent should then call the `wait()` system call to read the dead process' exit status and other information; until that moment, the child process remains a zombie.

**Commands and Options**

- `ps -ef` (UNIX options)  
  List all processes

- `ps aux` (BSD options)

- `pstat PID`  
  Display all processes in hierarchical format.  
  The process tree is rooted at `PID`, or at `init` if `PID` is omitted

- `pidof processname`  
  Show PIDs of processes with name `processname`

- `pidof -s processname`  
  Show PID of process with name `processname`, returning a single result

- `pgrep sshd`  
  Show processes whose name is "sshd"

- `pgrep -u root sshd`  
  Show processes whose name is "sshd" and are owned by root

- `pmap PID`  
  Display the memory map of process `PID`

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

- `killall -9 sshd`  
  Kill processes whose name is "sshd"

- `pkill -9 -u root sshd`  
  Kill processes whose name is "sshd" and are owned by root

- `pkill -9 -u user`  
  Kill all processes owned by `user`, forcing him to log out

- `skill`  
  Send a signal to a process or show process status. Obsolete

- `xkill`  
  Kill a process by its X GUI resource. Pops up a cursor to select a window

- `jobs`  
  List all jobs

  - `CTRL + Z`  
    Suspend a job, putting it in the stopped state (send a SIGTSTP)

  - `bg %n`  
    Put job # `n` in the background (send a SIGCONT)

  - `fg %n`  
    Resume job # `n` in the foreground and make it the current job (send a SIGCONT)

  - `kill %n`  
    Kill job # `n`

  - `disown %n`  
    Remove job # `n` from the table of active jobs

  - `disown -h %n`  
    Prevent job # `n` from receiving a SIGHUP if the shell receives that signal

The niceness value ranges from -20 to 19, and a newly created process has a default niceness of 0. Unprivileged users can modify a process’ niceness only within the range from 1 to 19.

- `nice -n -5 command`  
  Start `command` with a niceness of -5. If niceness is omitted, a default value of 10 is used

- `renice -5 command`  
  Change the niceness of a running `command` to -5

- `snice`  
  Change the niceness of a process. Obsolete
### Most frequently used signals

<table>
<thead>
<tr>
<th>Signal number</th>
<th>Signal name</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SIGHUP</td>
<td>Used by many daemons to reload their configuration</td>
</tr>
<tr>
<td>2</td>
<td>SIGINT</td>
<td>Interrupt, stop</td>
</tr>
<tr>
<td>9</td>
<td>SIGKILL</td>
<td>Kill unconditionally (this signal cannot be ignored)</td>
</tr>
<tr>
<td>15</td>
<td>SIGTERM</td>
<td>Terminate gracefully</td>
</tr>
<tr>
<td>18</td>
<td>SIGCONT</td>
<td>Continue execution</td>
</tr>
<tr>
<td>20</td>
<td>SIGTSTP</td>
<td>Stop execution</td>
</tr>
</tbody>
</table>

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

- `kill -l` List all available signal names
- `kill -l n` Print the name of signal number `n`

- `trap action condition` Trap a signal

- `strace command` Trace the execution of `command`, intercepting and printing system calls called by a process and signals received by a process

- `ipcs` Show IPC facilities information (shared memory, message queues, and semaphores)

- `:(){ :||& };:` Fork bomb: starts a process that continually replicates itself, slowing down or crashing the system because of resource starvation. Dangerous!

- `(` `command `)& `pid=$!; sleep ` `n` `; kill `-9 ` `Spid` Run `command` and kill it after `n` seconds
### Resource monitoring

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>top</strong></td>
<td>Monitor processes in real-time</td>
</tr>
<tr>
<td><strong>htop</strong></td>
<td>Monitor processes in real-time (Ncurses UI)</td>
</tr>
<tr>
<td><strong>iotop</strong></td>
<td>Display I/O usage by processes in the system</td>
</tr>
<tr>
<td><strong>atop</strong></td>
<td>Advanced system monitor that displays the load on CPU, RAM, disk, and network</td>
</tr>
<tr>
<td><strong>powertop</strong></td>
<td>Power consumption and power management diagnosis tool</td>
</tr>
<tr>
<td><strong>uptime</strong></td>
<td>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</td>
</tr>
<tr>
<td><strong>time command</strong></td>
<td>Execute <code>command</code> and, at its completion, write to stderr timing statistics about the run: elapsed real time between invocation and termination, user CPU time, system CPU time</td>
</tr>
<tr>
<td><strong>sar</strong></td>
<td>Show reports about system activity (including reboots). Reports are generated from data collected via the cron job <code>sysstat</code> and stored in <code>/var/log/sa/sn</code>, where <code>n</code> is the day of the month</td>
</tr>
<tr>
<td><code>sar -f /var/log/sa/sa13 -s 06:00:00 -e 09:00:00</code></td>
<td>Show reports for system activity from 6 to 9 AM on the 13th of the month</td>
</tr>
<tr>
<td><code>sar -u n m</code></td>
<td>Show real-time CPU activity, every <code>n</code> seconds for <code>m</code> times</td>
</tr>
<tr>
<td><code>sar -n DEV</code></td>
<td>Show real-time network activity (received and transmitted packets per second)</td>
</tr>
<tr>
<td><strong>sysbench</strong></td>
<td>Multi-threaded benchmark tool able to monitor different OS parameters: file I/O, scheduler, memory allocation, thread implementation, databases</td>
</tr>
<tr>
<td><strong>inxi</strong></td>
<td>Debugging tool to rapidly and easily gather system information and configuration</td>
</tr>
<tr>
<td><strong>stress-ng</strong></td>
<td>Tool for CPU and RAM stress tests</td>
</tr>
</tbody>
</table>

### Linux monitoring tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>collectd</strong></td>
<td>System statistics collector</td>
</tr>
<tr>
<td><strong>Nagios</strong></td>
<td>System monitor and alert</td>
</tr>
<tr>
<td><strong>MRTG</strong></td>
<td>Network load monitor</td>
</tr>
<tr>
<td><strong>Cacti</strong></td>
<td>Network monitor</td>
</tr>
<tr>
<td><strong>Munin</strong></td>
<td>System and network monitor and alert</td>
</tr>
<tr>
<td><strong>Zabbix</strong></td>
<td>System and network monitor and alert</td>
</tr>
<tr>
<td><strong>Centreon</strong></td>
<td>System and network monitor and alert</td>
</tr>
<tr>
<td><strong>netdata</strong></td>
<td>Real-time performance and health monitor</td>
</tr>
</tbody>
</table>
vmstat

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

iostat

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

mpstat

Print a report about processor activities

vmstat \textit{n} \textit{m}
iostat \textit{n} \textit{m}
mpstat \textit{n} \textit{m}

Print the relevant report every \textit{n} seconds for \textit{m} times

### Output of command `vmstat`

<table>
<thead>
<tr>
<th>proc</th>
<th>r</th>
<th>b</th>
<th>Memory used (swap)</th>
<th>memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>proc</td>
<td></td>
<td></td>
<td>Virtual memory used (swap)</td>
<td></td>
</tr>
<tr>
<td>proc</td>
<td></td>
<td></td>
<td>Free memory (idle)</td>
<td></td>
</tr>
<tr>
<td>proc</td>
<td></td>
<td></td>
<td>Memory used as buffers</td>
<td></td>
</tr>
<tr>
<td>proc</td>
<td></td>
<td></td>
<td>Memory used as cache</td>
<td></td>
</tr>
<tr>
<td>proc</td>
<td></td>
<td></td>
<td>Memory swapped in from disk</td>
<td></td>
</tr>
<tr>
<td>proc</td>
<td></td>
<td></td>
<td>Memory swapped out to disk</td>
<td></td>
</tr>
<tr>
<td>proc</td>
<td></td>
<td></td>
<td>Blocks received in from a block device</td>
<td></td>
</tr>
<tr>
<td>proc</td>
<td></td>
<td></td>
<td>Blocks sent out to a block device</td>
<td></td>
</tr>
<tr>
<td>proc</td>
<td></td>
<td></td>
<td>Number of interrupts</td>
<td></td>
</tr>
<tr>
<td>proc</td>
<td></td>
<td></td>
<td>Number of context switches</td>
<td></td>
</tr>
<tr>
<td>proc</td>
<td></td>
<td></td>
<td>Time spent running user code (non-kernel)</td>
<td></td>
</tr>
<tr>
<td>proc</td>
<td></td>
<td></td>
<td>Time spent running system code (kernel)</td>
<td></td>
</tr>
<tr>
<td>proc</td>
<td></td>
<td></td>
<td>Time spent idle</td>
<td></td>
</tr>
<tr>
<td>proc</td>
<td></td>
<td></td>
<td>Time spent waiting for I/O</td>
<td></td>
</tr>
<tr>
<td>proc</td>
<td></td>
<td></td>
<td>Time stolen from a virtual machine</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output of command <code>vmstat</code></th>
<th>procs</th>
<th>memory</th>
<th>swap</th>
<th>io</th>
<th>system</th>
<th>cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td>procs r b swpd free buff cache si so bi bo in cs us sy id wa st</td>
<td>0 0 0 296724 267120 3393400 0 0 17 56 0 3 2 2 95 1 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **procs:**
  - \textit{r} \ Number of runnable processes (running or waiting for run time)
  - \textit{b} \ Number of processes in uninterruptible sleep

- **memory:**
  - \textit{swpd} \ Virtual memory used (swap)
  - \textit{free} \ Free memory (idle)
  - \textit{buff} \ Memory used as buffers
  - \textit{cache} \ Memory used as cache

- **swap:**
  - \textit{si} \ Memory swapped in from disk
  - \textit{so} \ Memory swapped out to disk

- **io:**
  - \textit{bi} \ Blocks received in from a block device
  - \textit{bo} \ Blocks sent out to a block device

- **system:**
  - \textit{in} \ Number of interrupts
  - \textit{cs} \ Number of context switches

- **cpu:**
  - \textit{us} \ Time spent running user code (non-kernel)
  - \textit{sy} \ Time spent running system code (kernel)
  - \textit{id} \ Time spent idle
  - \textit{wa} \ Time spent waiting for I/O
  - \textit{st} \ Time stolen from a virtual machine
free

Show the amount of free and used memory in the system

<table>
<thead>
<tr>
<th>Output of command <code>free</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>total</strong></td>
</tr>
<tr>
<td>Mem:</td>
</tr>
<tr>
<td>Swap:</td>
</tr>
</tbody>
</table>

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

**Mem**

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

**-/+ buffers/cache**

- **used**: Memory used by kernel buffers
- **free**: Memory available for new applications (without using swap)

**Swap**

- **total**: Total configured amount of swap space
- **used**: Used swap space
- **free**: Free swap space

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

#### Permissions

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

- chmod 777 file
- chmod u=rwx,go=x file
- chmod u+wx file
- chmod -x file

- chmod -R g+x /path
- find /path -type d \\ -exec chmod g+x {} \\ ;

- chown user file
- chown user:group file
- chown :group file
- chgrp group file

- umask 022

Set read, write, and execute permission to user; set execute permission to group and others.

Add write and execute permission to user.

Remove execute permission from everybody (user, group, and others).

Set the group execute bit recursively on path and every dir and file underneath.

Set the group execute bit recursively on path and every dir, but not file, underneath.

Change the owner of the file to user.

Change the owner of the file to user, and group ownership of the file to group.

Change group ownership of the file to group.

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.
## File attributes

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

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

### Timestamp

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Value tracked</th>
<th>Displayed via</th>
</tr>
</thead>
<tbody>
<tr>
<td>mtime</td>
<td>Time of last modification to file contents (data itself)</td>
<td><code>ls -1</code></td>
</tr>
<tr>
<td>ctime</td>
<td>Time of last change to file contents or file metadata (owner, group, or permissions)</td>
<td><code>ls -1c</code></td>
</tr>
<tr>
<td>atime</td>
<td>Time of last access to file for reading contents</td>
<td><code>ls -lu</code></td>
</tr>
</tbody>
</table>

The POSIX standard does not define a timestamp for file creation. Some filesystems (e.g. ext4, JFS, Btrfs) store this value, but currently there is no Linux kernel API to access it.
Access Control Lists (ACLs) provide a fine-grained set of permissions that can be applied to files and directories. An access ACL is set on an individual file or directory; a default ACL is set on a directory, and applies to all files and subdirs created inside it that don't have an access ACL. The final permissions are the intersection of the ACL with the chmod/umask value. A partition must have been mounted with the acl option in order to support ACLs on files.

```
setfacl -m u:user:permissions file
setfacl -m g:group:permissions file
setfacl -m m:permissions file
setfacl -m o:permissions file
setfacl -x u:user file
setfacl -x g:group file
```

Set an access ACL on a file for an user
Set an access ACL on a file for a group
Set the effective rights mask on a file
Set the permissions on a file for other users
Remove an access ACL from a file for an user
Remove an access ACL from a file for a group

The permissions are standard Unix permissions specified as any combination of rwx.

```
setfacl -m d:u:user:permissions dir
setfacl -d -m u:user:permissions dir
```

Same as above, but set a default ACL instead of an access ACL. This applies to all commands above

```
getfacl file
getfacl file1 | setfacl --set-file= file2
getfacl --access dir | setfacl -d -M- dir
```

Display the access (and default, if any) ACL for a file
Copy the ACL of file1 and apply it to file2
Copy the access ACL of a directory and set it as default ACL

```
chacl options
```

Change an ACL. This command exists to provide compatibility with IRIX

```
man acl
```

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

A directory contains a list of mappings between filenames and inodes.

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

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

<table>
<thead>
<tr>
<th>Hard link</th>
<th>Symbolic link</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Definition</strong></td>
<td>A link to an already existing inode</td>
</tr>
<tr>
<td><strong>Command to create it</strong></td>
<td><code>ln file hardlink</code></td>
</tr>
<tr>
<td><strong>Link is still valid if the original file is moved or deleted</strong></td>
<td>Yes (because the link still references the inode to which the original file pointed)</td>
</tr>
<tr>
<td><strong>Can link to a file in another filesystem</strong></td>
<td>No (because inode numbers make sense only within a determinate filesystem)</td>
</tr>
<tr>
<td><strong>Can link to a directory</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Link permissions</strong></td>
<td>Reflect the original file’s permissions, even when these are changed</td>
</tr>
<tr>
<td><strong>Link attributes</strong></td>
<td><code>-</code> (regular file)</td>
</tr>
<tr>
<td><strong>Inode number</strong></td>
<td>The same as the original file</td>
</tr>
</tbody>
</table>

```bash
ls -i
ls -l
df -i
find / -inum n
find / -samefile file
```

Show a listing of the directory with the inode number for each file

Show a listing of the directory with the link count for each file

Report filesystem inode usage

Find all files linked to the same inode n

Find all files linked to the same inode as file
Find system files

`find /path -name "foo"`
`find /path -name "foo" -print`
`find / -name "foo" -exec chmod 700 {} \;`
`find / -name "foo" -ok chmod 700 {} \;`
`find / -size +128M`
`find / -type f -ctime +10`
`find / -type f -perm -4000`
`find / -type f -newermt "May 4 2:55" -delete`
`find . -type f -print -exec cat {} \;`
`find . ! -name ".gz" -type f -exec gzip {} \;`
`find / -xdev -type f -exec ls -lah {} \;`

Locate file
`slocate file`

`updatedb`

`which command`
`which -a command`

`whereis command`
`whereis -b command`
`whereis -s command`
`whereis -m command`

`type command`

`file file`

Find all files and dirs, in the directory tree rooted at /path, whose name starts with "foo"
Find all files and dirs whose name start with "foo" and apply permission 700 to all of them
Find all files and dirs whose name start with "foo" and apply permission 700 to all of them, asking for confirmation
Find all files larger than 128 Mb
Find all files last changed more than 10 days ago
Find all files with SUID set (a possible security risk, because a shell with SUID root is a backdoor)
Find and delete all files newer than the specified timestamp. Using -delete is preferable to using -exec rm {} \;
Print all files, in the current directory and under, prepending them with a filename header
Find all files, in the current directory and under, which do not have the gz extension, and compress them
Find all files larger than 100 Mb in the current filesystem only and display detailed information about them

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

Locate a binary executable command within the PATH
Locate all matches of a command, not only the first one
Locate the binary, source, and manpage files for a command
Locate the binary files for a command
Locate the source files for a command
Locate the manpage files for a command

Determine if a command is a program or a built-in (i.e. an internal feature of the shell)

Analyze the content of a file or directory, and display the kind of file (e.g. executable, text file, program text, swap file)
The scope of variables is the current shell only, while environment variables are visible within the current shell as well as within all subshells and Bash child processes spawned by the shell. Environment variables are set in /etc/environment in the form variable=value. Conventionally, variable names are lowercase while environment variable names are uppercase.

- `set` Display all variables
- `env` Display all environment variables
- `readonly -p` Display all variables that are read-only
- `VAR=value` Set the value of a variable. There must be no spaces around the = sign. It is possible to add space around ( and )
- `readonly VAR=value` Set a variable making its value unchangeable
- `set ${VAR:=value}` Set a variable only if it is not already set (i.e. does not exist) or is null
- `unset VAR` Unset (i.e. delete) a variable
- `export VAR` Export a variable, making it an environment variable
- `command $VAR` Pass a variable as argument to command. If other characters follow the variable name, it is necessary to specify the boundaries of the variable name via () to make it unambiguous. It is recommended to double quote the variable when referencing it, to prevent interpretation of special characters (except \ $ `) and word splitting (in case the variable value contains whitespaces), both of which will have unintended results
- `VAR=$((5 + 37))` Evaluate a numeric expression and assign the result to another variable
- `VAR=` Command substitution. Assign to a variable the standard output resulting from command (which is executed in a subshell)
- `for i in /path/* do echo "Filename: $i" done` Loop and operate through all the output tokens (in this case, files in the path). The equivalent construct for i in $(ls /path/) is unnecessary and harmful, because filenames containing whitespaces or glob characters will cause unintended results
- `echo ${VAR:-message}` If variable exists and is not null, print its value, otherwise print message
- `echo ${VAR:+message}` If variable exists and is not null, print message, otherwise print nothing
- `echo ${VAR,,}` Print a string variable in lowercase
- `TOKENS=($STRING)` String tokenizer. Splits a string stored in the variable STRING into tokens, according to the content of the shell variable $IFS, and stores them in the array TOKENS
- `echo ${TOKENS[n]}` Print the token number n
- `echo ${TOKENS[*]}` Print all tokens
### Bash built-in variables

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

### Bash shell event

<table>
<thead>
<tr>
<th>Event</th>
<th>Files run</th>
</tr>
</thead>
<tbody>
<tr>
<td>When a login shell is launched</td>
<td><code>/etc/profile</code> <code>/etc/profile.d/*.sh</code> <code>~/.bash_profile</code> <code>~/.bash_login</code> <code>~/.profile</code></td>
</tr>
<tr>
<td>When a login shell exits</td>
<td><code>~/.bash_logout</code></td>
</tr>
<tr>
<td>When a non-login shell is launched</td>
<td><code>/etc/bash.bashrc</code> <code>/etc/bashrc</code> <code>~/.bashrc</code></td>
</tr>
</tbody>
</table>
Shell options

```
set -o longoption

set +o longoption
```

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>set -o</code></td>
<td>Show the status of all Bash options</td>
</tr>
<tr>
<td><code>set -v</code></td>
<td>Print shell input lines as they are read</td>
</tr>
<tr>
<td><code>set -x</code></td>
<td>Print command traces before execution of each command (debug mode)</td>
</tr>
<tr>
<td><code>set -e</code></td>
<td>Exit the script immediately if a command fails. Recommended option</td>
</tr>
<tr>
<td><code>set -u</code></td>
<td>Treat expansion of unset variables as an error. This avoids unintended results</td>
</tr>
</tbody>
</table>

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

```
shopt

shopt -s shelloption
shopt -u shelloption
```

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>shopt</code></td>
<td>Display the list of all shell options with their current value (on or off)</td>
</tr>
<tr>
<td><code>shopt -s shelloption</code></td>
<td>Set (enable) a specific shell option</td>
</tr>
<tr>
<td><code>shopt -u shelloption</code></td>
<td>Unset (disable) a specific shell option</td>
</tr>
</tbody>
</table>
Bash shell scripts must start with the shebang line `#!/bin/bash` indicating the location of the script interpreter.

<table>
<thead>
<tr>
<th>Script execution</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>source script.sh</code></td>
<td>Script execution takes place in the same shell. Variables defined and exported in the script are seen by the shell when the script exits</td>
</tr>
<tr>
<td><code>. script.sh</code></td>
<td></td>
</tr>
<tr>
<td><code>bash script.sh</code></td>
<td>Script execution spawns a new shell</td>
</tr>
<tr>
<td><code>./script.sh</code> (file must be executable)</td>
<td></td>
</tr>
</tbody>
</table>

- `command &`  
  Execute `command` in the background
- `command1; command2`  
  Execute `command 1` and then `command 2`
- `command1 && command2`  
  Execute `command 2` only if `command 1` executed successfully (exit status = 0)
- `command1 || command2`  
  Execute `command 2` only if `command 1` did not execute successfully (exit status > 0)
- `(command1 && command2)`  
  Group commands together for evaluation priority
- `command1 ||| command2`  
  Run `command` in a subshell. This is used to isolate `command`'s effects, as variable assignments and other changes to the shell environment operated by `command` will not remain after `command` completes
- `exit`  
  Terminate a script
- `exit n`  
  Terminate a script with the specified exit status number `n`. By convention, a 0 exit status is used if the script executed successfully, a non-zero value otherwise
- `/bin/true`  
  Do nothing and return immediately a status code of 0 (indicating success)
- `/bin/false`  
  Do nothing and return immediately a status code of 1 (indicating failure)
- `if command then echo "Success" else echo "Failure" fi`  
  Run a command, then evaluate whether it exited successfully or failed
- `function myfunc { commands } myfunc() { commands }`  
  Define a function. A function must be defined before it can be used in a Bash script. Argument number `n` is accessed in the body of the function via `${n}`. An advantage of functions over aliases is that functions can be passed arguments
- `readonly -f myfunc`  
  Call a function
- `readonly -f -t myfunc`  
  Mark an already defined function as read-only, preventing it to be redefined
- `typeset -f`  
  Show functions defined in the current Bash session
- `readonly -p -f`  
  Show functions which are read-only
- `expect`  
  Dialogue with interactive programs according to a script, analyzing what can be expected from the interactive program and replying accordingly
- `zenity`  
  Display GTK+ graphical dialogs for user messages and input
getopts

Parse positional parameters in a shell script

<table>
<thead>
<tr>
<th>getopts syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>while getopts abc:d: OPT do case $OPT in</td>
<td>Definition of accepted options</td>
</tr>
<tr>
<td>a)</td>
<td></td>
</tr>
<tr>
<td>command_a</td>
<td>Matches option -a.</td>
</tr>
<tr>
<td></td>
<td>Executes a command</td>
</tr>
<tr>
<td>b)</td>
<td></td>
</tr>
<tr>
<td>command_b</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>c)</td>
<td></td>
</tr>
<tr>
<td>command_c $OPTARG exit 0 ;;</td>
<td>Matches option -c argument.</td>
</tr>
<tr>
<td></td>
<td>Executes a command with argument</td>
</tr>
<tr>
<td>d)</td>
<td></td>
</tr>
<tr>
<td>command_d $OPTARG exit 0 ;;</td>
<td></td>
</tr>
<tr>
<td>*)</td>
<td></td>
</tr>
<tr>
<td>default_command exit 1 ;;</td>
<td>Command to execute if none of above options applies</td>
</tr>
<tr>
<td>esac</td>
<td></td>
</tr>
<tr>
<td>done</td>
<td></td>
</tr>
</tbody>
</table>
Command execution

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

test "$MYVAR" operator "value" && command
[ "$MYVAR" operator "value" ] && command
if [ "$MYVAR" operator "value" ]; then command; fi
```

### Test operators

<table>
<thead>
<tr>
<th>Integer operators</th>
<th>File operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>-eq value</td>
<td>-e or -a file</td>
</tr>
<tr>
<td>-ne value</td>
<td>-f file</td>
</tr>
<tr>
<td>-lt value</td>
<td>-d file</td>
</tr>
<tr>
<td>-le value</td>
<td>-b file</td>
</tr>
<tr>
<td>-gt value</td>
<td>-c file</td>
</tr>
<tr>
<td>-ge value</td>
<td>-r file</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Numeric operators</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>= value</td>
<td>-e or -a file</td>
</tr>
<tr>
<td>!= value</td>
<td>-f file</td>
</tr>
<tr>
<td>&lt; value</td>
<td>-d file</td>
</tr>
<tr>
<td>&lt;= value</td>
<td>-b file</td>
</tr>
<tr>
<td>&gt; value</td>
<td>-c file</td>
</tr>
<tr>
<td>&gt;= value</td>
<td>-r file</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression operators</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>expr1 -a expr2</td>
<td>-e or -a file</td>
</tr>
<tr>
<td>expr1 -o expr2</td>
<td>-f file</td>
</tr>
<tr>
<td>! expr1</td>
<td>-d file</td>
</tr>
<tr>
<td>( expr1 )</td>
<td>-b file</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String operators</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-z</td>
<td>-e or -a file</td>
</tr>
<tr>
<td>-n or nothing</td>
<td>-f file</td>
</tr>
<tr>
<td>= or == string</td>
<td>-d file</td>
</tr>
<tr>
<td>!= string</td>
<td>-b file</td>
</tr>
<tr>
<td>&lt; string</td>
<td>-c file</td>
</tr>
<tr>
<td>&gt; string</td>
<td>-r file</td>
</tr>
<tr>
<td>substr string pos len</td>
<td>Is non-zero length</td>
</tr>
<tr>
<td>length string</td>
<td>-s file</td>
</tr>
<tr>
<td>string : regex</td>
<td>-t file</td>
</tr>
<tr>
<td>or match string regex</td>
<td>Is owned by the Effective UID</td>
</tr>
</tbody>
</table>

### Example:

```bash
expr "$MYVAR" = "39 + 3"  # Evaluate an expression (in this case, assigns the value 42 to the variable)
expr string : regex        # Return the length of the substring matching the regex
expr string : \(regex\)   # Return the substring matching the regex
```
<table>
<thead>
<tr>
<th>Mathematical operators</th>
<th>Logical operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>!</td>
</tr>
<tr>
<td>Addition</td>
<td>Logical negation</td>
</tr>
<tr>
<td>−</td>
<td>&amp;&amp;</td>
</tr>
<tr>
<td>Subtraction</td>
<td>Logical AND</td>
</tr>
<tr>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Multiplication</td>
<td>Logical OR</td>
</tr>
<tr>
<td>/</td>
<td>Bitwise operators</td>
</tr>
<tr>
<td>Division</td>
<td>^</td>
</tr>
<tr>
<td>%</td>
<td>Bitwise negation</td>
</tr>
<tr>
<td>Remainder</td>
<td>&lt;&lt;</td>
</tr>
<tr>
<td>**</td>
<td>Bitwise AND</td>
</tr>
<tr>
<td>Exponentiation</td>
<td>&gt;&gt;</td>
</tr>
<tr>
<td>++</td>
<td>Bitwise OR</td>
</tr>
<tr>
<td>Pre/post increment</td>
<td></td>
</tr>
<tr>
<td>--</td>
<td>Bitwise XOR</td>
</tr>
<tr>
<td>Pre/post decrement</td>
<td></td>
</tr>
<tr>
<td>Assignment operators</td>
<td></td>
</tr>
<tr>
<td>=</td>
<td>&lt;&lt;</td>
</tr>
<tr>
<td>Assignment</td>
<td>Right bitwise shift</td>
</tr>
<tr>
<td>op=</td>
<td></td>
</tr>
<tr>
<td>Operation and assignment</td>
<td></td>
</tr>
</tbody>
</table>
## Flow control

### Tests

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

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

### Loops

<table>
<thead>
<tr>
<th>Loop Type</th>
<th>Description</th>
<th>Code Example</th>
</tr>
</thead>
</table>
| while | The command block executes as long as test is true | i=0
while [ $i -le 7 ]
do
  echo $i
  let i++
done |
| until | The command block executes as long as test is false | i=0
until [ $i -gt 7 ]
do
  echo $i
  let i++
done |
| for | The command block executes for each item in list | for i in 0 1 2 3 4 5 6 7
do
  echo $i
done |
| for | | for i in {0..7}
do
  echo $i
done |
| for | | start=0
end=7
for i in $(seq $start $end)
do
  echo $i
done |
| for | | start=0
end=7
for ((i = start; i <= end; i++))
do
  echo $i
done |
<p>| break | Exit a loop | |
| continue | Jump to the next iteration | |</p>
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vi</td>
<td>Vi, text editor</td>
</tr>
<tr>
<td>vim</td>
<td>Vi Improved, an advanced text editor</td>
</tr>
<tr>
<td>gvim</td>
<td>Vim with GUI</td>
</tr>
<tr>
<td>vimgdiff file1 file2</td>
<td>Compare two text files in Vim</td>
</tr>
<tr>
<td>pico</td>
<td>Pico, simple text editor</td>
</tr>
<tr>
<td>nano</td>
<td>Nano, simple text editor (a GNU clone of Pico)</td>
</tr>
<tr>
<td>rnano</td>
<td>Restricted version of Nano: does not allow the user access the filesystem (except for files specified as argument) or a command shell</td>
</tr>
<tr>
<td>emacs</td>
<td>GNU Emacs, a GUI text editor</td>
</tr>
<tr>
<td>gedit</td>
<td>GUI text editor</td>
</tr>
<tr>
<td>ed</td>
<td>Line-oriented text editor</td>
</tr>
<tr>
<td>hexedit</td>
<td>Hexadecimal and ASCII editor</td>
</tr>
<tr>
<td>more</td>
<td>Text pager (obsolete)</td>
</tr>
<tr>
<td>less</td>
<td>Text pager</td>
</tr>
<tr>
<td>most</td>
<td>Text pager with advanced features (screen split, binary viewer, etc.)</td>
</tr>
<tr>
<td>Key</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>G</td>
<td>Go to the first line in the file</td>
</tr>
<tr>
<td>ng</td>
<td>Go to line number ( n )</td>
</tr>
<tr>
<td>G</td>
<td>Go to the last line in the file</td>
</tr>
<tr>
<td>F</td>
<td>Go to the end of the file, and move forward automatically as the file grows</td>
</tr>
<tr>
<td></td>
<td>Stop moving forward</td>
</tr>
<tr>
<td>-N</td>
<td>Show line numbers</td>
</tr>
<tr>
<td>-n</td>
<td>Don't show line numbers</td>
</tr>
<tr>
<td>=</td>
<td>Show information about the file</td>
</tr>
<tr>
<td>⌃+G</td>
<td>Show current and total line number, byte, and percentage of the file read</td>
</tr>
<tr>
<td>/pattern</td>
<td>Search ( \text{pattern} ) forward</td>
</tr>
<tr>
<td>?pattern</td>
<td>Search ( \text{pattern} ) backwards</td>
</tr>
<tr>
<td>&amp;pattern</td>
<td>Display only lines matching ( \text{pattern} )</td>
</tr>
<tr>
<td>n</td>
<td>Search next occurrences forward</td>
</tr>
<tr>
<td>N</td>
<td>Search next occurrences backwards</td>
</tr>
<tr>
<td>:n</td>
<td>When reading multiple files, go to the next file</td>
</tr>
<tr>
<td>:p</td>
<td>When reading multiple files, go to the previous file</td>
</tr>
<tr>
<td>R</td>
<td>Repaint the screen</td>
</tr>
<tr>
<td>V</td>
<td>Show version number</td>
</tr>
<tr>
<td>h</td>
<td>Help</td>
</tr>
<tr>
<td>q</td>
<td>Quit</td>
</tr>
</tbody>
</table>

```
less +command file
```
Open file for reading, applying `command` (see list above)

```
less +F --follow-name file
```
Move forward, attempting periodically to reopen file by name; useful to keep reading a logfile that is being rotated. Note that, by default, `less` continues to read the original input file even if it has been renamed.
### Vi commands

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

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>ai</td>
<td>Turn on auto indentation</td>
</tr>
<tr>
<td>all</td>
<td>Display all options</td>
</tr>
<tr>
<td>ap</td>
<td>Print a line after the commands</td>
</tr>
<tr>
<td>aw</td>
<td>Automatic write on commands :n ! e# ^^ :rew ^} :tag</td>
</tr>
<tr>
<td>bf</td>
<td>Discard control characters from input</td>
</tr>
<tr>
<td>dir=tmpdir</td>
<td>Set tmpdir as directory for temporary files</td>
</tr>
<tr>
<td>eb</td>
<td>Precede error messages with a bell</td>
</tr>
<tr>
<td>ht=8</td>
<td>Set terminal tab as 8 spaces</td>
</tr>
<tr>
<td>ic</td>
<td>Ignore case when searching</td>
</tr>
<tr>
<td>lisp</td>
<td>Modify brackets for Lisp compatibility</td>
</tr>
<tr>
<td>list</td>
<td>Show tabs and EOL characters</td>
</tr>
<tr>
<td>set listchars=tab:&gt;&amp;-</td>
<td>Show tab as &gt; for the first char and as – for the following chars</td>
</tr>
<tr>
<td>magic</td>
<td>Allow pattern matching with special characters</td>
</tr>
<tr>
<td>mesg</td>
<td>Enable UNIX terminal messaging</td>
</tr>
<tr>
<td>nu</td>
<td>Show line numbers</td>
</tr>
<tr>
<td>opt</td>
<td>Speed up output by eliminating automatic Return</td>
</tr>
<tr>
<td>para=LL1PLPPQPbP</td>
<td>Set macro to start paragraphs for { } operators</td>
</tr>
<tr>
<td>prompt</td>
<td>Prompt : for command input</td>
</tr>
<tr>
<td>re</td>
<td>Simulate smart terminal on dumb terminal</td>
</tr>
<tr>
<td>remap</td>
<td>Accept macros within macros</td>
</tr>
<tr>
<td>report</td>
<td>Show the largest size of changes on status line</td>
</tr>
<tr>
<td>ro</td>
<td>Make file readonly</td>
</tr>
<tr>
<td>scroll=12</td>
<td>Set screen size as 12 lines</td>
</tr>
<tr>
<td>shell=/bin/bash</td>
<td>Set shell escape to /bin/bash</td>
</tr>
<tr>
<td>showmode</td>
<td>Show current mode on status line</td>
</tr>
<tr>
<td>slow</td>
<td>Postpone display updates during inserts</td>
</tr>
<tr>
<td>sm</td>
<td>Show matching parentheses when typing</td>
</tr>
<tr>
<td>sw=8</td>
<td>Set shift width to 8 characters</td>
</tr>
<tr>
<td>tags=/usr/lib/tags</td>
<td>Set path for files checked for tags</td>
</tr>
<tr>
<td>term</td>
<td>Print terminal type</td>
</tr>
<tr>
<td>terse</td>
<td>Print terse messages</td>
</tr>
<tr>
<td>timeout</td>
<td>Eliminate 1-second time limit for macros</td>
</tr>
<tr>
<td>tl=3</td>
<td>Set significance of tags beyond 3 characters (0 = all)</td>
</tr>
<tr>
<td>ts=8</td>
<td>Set tab stops to 8 for text input</td>
</tr>
<tr>
<td>wa</td>
<td>Inhibit normal checks before write commands</td>
</tr>
<tr>
<td>warn</td>
<td>Display the warning message &quot;No write since last change&quot;</td>
</tr>
<tr>
<td>window=24</td>
<td>Set text window as 24 lines</td>
</tr>
<tr>
<td>wm=0</td>
<td>Set automatic wraparound 0 spaces from right margin</td>
</tr>
</tbody>
</table>

Options can also be permanently set by including them in ~/.exrc (Vi) or ~/.vimrc (Vim)
<table>
<thead>
<tr>
<th>SQL Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHOW DATABASES;</td>
<td>Show all existing databases</td>
</tr>
<tr>
<td>USE CompanyDatabase;</td>
<td>Select a database to use</td>
</tr>
<tr>
<td>SELECT DATABASE();</td>
<td>Show which database is currently selected</td>
</tr>
<tr>
<td>DROP DATABASE CompanyDatabase;</td>
<td>Delete a database</td>
</tr>
<tr>
<td>SHOW TABLES;</td>
<td>Show all tables from the selected database</td>
</tr>
<tr>
<td>CREATE TABLE customers ( cusid INT NOT NULL AUTO_INCREMENT PRIMARY KEY, firstname VARCHAR(32), lastname VARCHAR(32), dob DATE, city VARCHAR(24), zipcode VARCHAR(5));</td>
<td>Create tables</td>
</tr>
<tr>
<td>CREATE TABLE payments ( payid INT NOT NULL AUTO_INCREMENT PRIMARY KEY, date DATE, fee INT, bill VARCHAR(128), cusid INT, CONSTRAINT FK1 FOREIGN KEY (cusid) REFERENCES customers(cusid));</td>
<td></td>
</tr>
<tr>
<td>INSERT INTO customers (firstname,lastname,dob) VALUES ('Arthur','Dent',1959-08-01), ('Trillian','','',1971-03-19);</td>
<td>Insert new records in a table</td>
</tr>
<tr>
<td>DELETE FROM customers WHERE firstname LIKE 'Zaphod';</td>
<td>Delete some records in a table</td>
</tr>
<tr>
<td>UPDATE customers SET city = 'London' WHERE zipcode = 'L1 42HG';</td>
<td>Modify records in a table</td>
</tr>
<tr>
<td>CREATE INDEX lastname_index ON customers(lastname);</td>
<td>Create an index for faster searches</td>
</tr>
<tr>
<td>ALTER TABLE customers ADD INDEX lastname_index (lastname);</td>
<td></td>
</tr>
<tr>
<td>DESCRIBE customers;</td>
<td>Describe the columns of a table</td>
</tr>
<tr>
<td>SHOW CREATE TABLE customers;</td>
<td>Show the code used to create a table</td>
</tr>
<tr>
<td>SHOW INDEXES FROM customers;</td>
<td>Show primary key and indexes of a table</td>
</tr>
<tr>
<td>DROP TABLE customers;</td>
<td>Delete a table</td>
</tr>
<tr>
<td>ALTER TABLE customers MODIFY city VARCHAR(32);</td>
<td>Modify the type of a column</td>
</tr>
<tr>
<td>CREATE VIEW cust_view AS SELECT * FROM customers WHERE city != 'London';</td>
<td>Create a view. Views are used similarly to tables</td>
</tr>
<tr>
<td>COMMIT;</td>
<td>Commit changes to the database</td>
</tr>
<tr>
<td>ROLLBACK;</td>
<td>Rollback the current transaction, canceling any changes done during it</td>
</tr>
<tr>
<td>START TRANSACTION; BEGIN;</td>
<td>Disable autocommit for this transaction, until a COMMIT or ROLLBACK is issued</td>
</tr>
</tbody>
</table>

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

SELECT firstname, lastname FROM customers LIMIT 5;

SELECT firstname, lastname FROM customers LIMIT 1000,5;
SELECT firstname, lastname FROM customers OFFSET 1000 LIMIT 5;

SELECT firstname, lastname FROM customers WHERE zipcode = 'L1 42HG';
SELECT firstname, lastname FROM customers WHERE zipcode IS NOT NULL;

SELECT * FROM customers ORDER BY lastname, firstname;
SELECT customers ORDER by zipcode DESC;

SELECT city, COUNT(*) FROM customers GROUP BY city;
SELECT cusid, SUM(fee) FROM payments GROUP BY cusid;
SELECT cusid, AVG(fee) FROM payments GROUP BY cusid HAVING AVG(fee)<50;
SELECT cusid, SUM(fee) FROM payments GROUP BY cusid HAVING AVG(fee)<50;
SELECT firstname, lastname, TIMESTAMPDIFF(YEAR,dob,CURRENT_DATE) AS age FROM customers;
SELECT city, COUNT(*) FROM customers GROUP BY city;
SELECT * FROM customers WHERE lastname IN (SELECT lastname FROM customers GROUP BY lastname HAVING COUNT(lastname) > 1);
SELECT cusid FROM payments WHERE fee > ALL (SELECT fee FROM payments WHERE cusid = 4242001);
SELECT cusid, SUM(fee) FROM payments GROUP BY cusid HAVING AVG(fee)<50;
SELECT cusid, AVG(fee) FROM payments GROUP BY cusid HAVING AVG(fee)<50;
SELECT * FROM customers WHERE firstname LIKE 'Trill%';
SELECT * FROM customers WHERE firstname REGEXP '^Art.*r$';

SELECT firstname, lastname FROM customers WHERE zipcode = 'L1 42HG'
UNION
SELECT firstname, lastname FROM customers WHERE cusid > 4242001;
SELECT firstname, lastname FROM customers WHERE zipcode = 'L1 42HG'
INTERSECT
SELECT firstname, lastname FROM customers WHERE cusid > 4242001;
SELECT firstname, lastname FROM customers WHERE zipcode = 'L1 42HG'
EXCEPT
SELECT firstname, lastname FROM customers WHERE cusid > 4242001;

Select all columns from the customers table
Select first and last name of customers, showing 5 records only
Select first and last name of customers, showing 5 records only
Select first and last name of customers, skipping the first 1000 records only
Select first and last name of customers, skipping the first 1000 records only
Select first and last name of customers with an existing zip code
Select customers in alphabetical order by last name, then first name
Select customers, sorting them by zip code in reverse order
Select first name, last name, and calculated age of customers
Show all cities, retrieving each unique output record only once
Show all cities and the number of customers in each city. NULL values are not counted
Show all fee payments grouped by customer ID, summed up
Show the average of fee payments grouped by customer ID, where this average is less than 50
Show the highest fee in the table
Show how many rows are in the table
Show the customer ID that pays the highest fee (via a subquery)
Show the customer ID that pays the highest fee (via a user set variable)
Show the customers which have same last name as other customers
Show the customer IDs that pay fees higher than the highest fee paid by customer ID 4242001
Show the customer IDs that pay fees higher than the highest fee paid by customer ID 4242001

Select customers whose first name matches the expression:
% any number of chars, even zero
_ a single char
Select customers whose first name matches the regex
Select customers that satisfy any of the two requirements
Select customers that satisfy both of the two requirements
Select customers that satisfy the first requirement but not the second
### SQL JOIN

<table>
<thead>
<tr>
<th>SQL</th>
<th>MySQL</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT customers.name, payments.bill FROM customers, payments WHERE customers.cusid = payments.cusid;</td>
<td>SELECT customers.name, payments.bill FROM customers INNER JOIN payments ON customers.cusid = payments.cusid;</td>
<td>Perform a join (aka inner join) of two tables to select data that are in a relationship</td>
</tr>
<tr>
<td>SELECT customers.name, payments.bill FROM customers NATURAL JOIN payments;</td>
<td>SELECT customers.name, payments.bill FROM customers INNER JOIN payments USING (cusid);</td>
<td></td>
</tr>
<tr>
<td>SELECT customers.name, payments.bill FROM customers JOIN payments USING (cusid);</td>
<td>SELECT customers.name, payments.bill FROM customers JOIN payments;</td>
<td></td>
</tr>
<tr>
<td>SELECT customers.name, payments.bill FROM customers CROSS JOIN payments;</td>
<td>SELECT customers.name, payments.bill FROM customers JOIN payments;</td>
<td>Perform a cross join (aka Cartesian product) of two tables</td>
</tr>
<tr>
<td>SELECT customers.name, payments.bill FROM customers LEFT JOIN payments ON customers.cusid = payments.cusid;</td>
<td>SELECT customers.name, payments.bill FROM customers JOIN payments;</td>
<td>Perform a left join (aka left outer join) of two tables, returning records matching the join condition and also records in the left table with unmatched values in the right table</td>
</tr>
<tr>
<td>SELECT customers.name, payments.bill FROM customers RIGHT JOIN payments ON customers.cusid = payments.cusid;</td>
<td>SELECT customers.name, payments.bill FROM customers JOIN payments;</td>
<td>Perform a right join (aka right outer join) of two tables, returning records matching the join condition and also records in the right table with unmatched values in the left table</td>
</tr>
</tbody>
</table>
MySQL is the most used open source RDBMS (Relational Database Management System). It runs on TCP port 3306. On RHEL 7 and later it is replaced by its fork MariaDB, but the names of the client and of most tools remain unchanged.

**mysqld_safe**

Start the MySQL server (`mysqld`) with safety features such as restarting the server if errors occur and logging runtime information to the error logfile. This is the recommended command.

**mysql_install_db (deprecated)**

Initialize the MySQL data directory, create system tables, and set up an administrative account. To be run just after installing the MySQL server.

**mysql_secure_installation**

Set password for root, remove anonymous users, disable remote root login, and remove test database. To be run just after installing the MySQL server.

**mysql -u root -p**

Login to MySQL as root and prompt for the password.

**mysql -u root -p**

Login to MySQL as root with the specified password.

**mysql -u root -p -h host -P port**

Login to the specified remote MySQL host and port.

**mysql -u root -p -eNB'SHOW DATABASES'**

Run a SQL command via MySQL. Flags are:
- `e` Run in batch mode
- `N` Do not print table header
- `B` Do not print table decoration characters `+-|`

**mysqldump -u root -p --all-databases > dump.sql**

Backup all databases to a dump file.

**mysqldump -u root -p db > dump.sql**

Backup a database to a dump file.

**mysqldump -u root -p --databases db1 db2 > dump.sql**

Backup multiple databases to a dump file.

**mysql -u root -p db table1 table2 > dump.sql**

Backup some tables of a database to a dump file.

**mysql -u root -p db < dump.sql**

Restore all databases from a dump file (which contains a complete dump of a MySQL server).

**mysql -u root -p db < dump.sql**

Restore a specific database from a dump file (which contains one database).

**mysql_upgrade -u root -p**

Check all tables in all databases for incompatibilities with the current version of MySQL.

**mysqlcheck**

Perform table maintenance. Each table is locked while is being processed. Options are:
- `--check` Check table for errors (default)
- `--analyze` Analyze table
- `--optimize` Optimize table
- `--repair` Repair table: can fix almost all problems except unique keys that are not unique

**mysqlcheck --check db table**

Check the specified table of the specified database.

**mysqlcheck --check --databases db1 db2**

Check the specified databases.

**mysqlcheck --check --all-databases**

Check all databases.
MySQL tools

mysqlslap

Tool for MySQL stress tests

mysqltuner.pl

Review the current MySQL installation configuration for performances and stability

mysqlreport (obsolete)

Generate a user-friendly report of MySQL status values

mytop

Monitor MySQL processes and queries

innnotop

Monitor MySQL InnoDB transactions

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

Perform an operation on each database name
**MySQL syntax**

- **List all MySQL users**
- **Create a MySQL local user and set his password**
- **Delete a MySQL user**
- **Set a password for a MySQL user. The password can be specified either in plaintext or by its hash value**
- **Show permissions for a user**
- **Grant permissions to a user**
- **Revoke permissions from a user; must match the already granted permission on the same database or table**
- **Create a MySQL user and set his grants at the same time**
- **Reload and commit the grant tables; must be run after any GRANT command**
- **Export a table to a CSV file**
- **Restore a database from a dump file**
- **Populate a table with data from a file (one record per line, values separated by tabs)**
- **Sleep for n seconds**
- **Enable profiling**
- **Send an SQL statement to the server**
- **Display result in vertical format, showing each record in multiple rows**
- **Insert a comment**
- **The commented statement is executed only if MySQL is version v or higher**
- **Cancel current input**
- **Run a shell command**
- **Log all I/O of the current MySQL session to the specified logfile**

```sql
SELECT Host, User FROM mysql.user;
CREATE USER 'user'@'localhost' IDENTIFIED BY 'p4ssw0rd';
DROP USER 'user'@'localhost';
SET PASSWORD FOR 'user'@'localhost' = PASSWORD('p4ssw0rd');
SET PASSWORD FOR 'user'@'localhost' = '*7E684A3DF6273CD1B6DE53';
SHOW GRANTS FOR 'user'@'localhost';
GRANT ALL PRIVILEGES ON database.* TO 'user'@'localhost';
REVOKE ALL PRIVILEGES ON database.* FROM 'user'@'localhost';
GRANT SELECT ON database.* TO 'john'@'localhost' IDENTIFIED BY 'p4ssw0rd';
GRANT SELECT ON database.* TO 'john'@'localhost' IDENTIFIED BY PASSWORD '*7E684A3DF6273CD1B6DE53';
FLUSH PRIVILEGES;

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

USE database; SOURCE dump.sql;
USE database; LOAD DATA LOCAL INFILE 'file' INTO TABLE table;

DO SLEEP(n);
SELECT SLEEP(n);

SET PROFILING=1;
SHOW PROFILE;

statement;
statement\g
statement\G

SELECT /*!99999 comment*/ * FROM database.table;
SELECT /*!v statement*/ * FROM database.table;
\c

\! command

TEE logfile
SHOW VARIABLES;
SHOW SESSION VARIABLES;
SHOW LOCAL VARIABLES;
SHOW GLOBAL VARIABLES;
SHOW VARIABLES LIKE '%query%';
SHOW VARIABLES LIKE 'hostname';
SELECT @@hostname;

SET sort_buffer_size=10000;
SET SESSION sort_buffer_size=10000;
SET LOCAL sort_buffer_size=10000;
SET @@session.sort_buffer_size=10000;
SET @@local.sort_buffer_size=10000;
SET GLOBAL sort_buffer_size=10000;
SET @@global.sort_buffer_size=10000;

SHOW STATUS;
SHOW SESSION STATUS;
SHOW LOCAL STATUS;
SHOW GLOBAL STATUS;
SHOW STATUS LIKE '%wsrep%';

SHOW WARNINGS;
SHOW ERRORS;

SHOW TABLE STATUS;
SHOW ENGINE INNODB STATUS;
SELECT * FROM information_schema.processlist;
SHOW FULL PROCESSLIST;
SELECT * FROM information_schema.processlist WHERE user='you';

SHOW CREATE TABLE table;
SHOW CREATE VIEW view;

SELECT VERSION();
SELECT CURDATE();
SELECT CURRENT_DATE;
SELECT CURTIME();
SELECT CURRENT_TIME;
SELECT NOW();
SELECT USER();

Print session variables (affecting current connection only)
Print global variables (affecting global operations on the server)
Print session variables that match the given pattern
Print a session variable with the given name
Set a session variable
Set a global variable
Print session status (concerning current connection only)
Print global status (concerning global operations on the server)
Print session status values that match the given pattern
Print warnings, errors and notes resulting from the most recent statement in the current session that generated messages
Print errors resulting from the most recent statement in the current session that generated messages
Print information about all tables of the current database e.g. engine (InnoDB or MyISAM), rows, indexes, data length
Print statistics concerning the InnoDB engine
Print the list of threads running in your local session; if run as root, print the list of threads running on the system
Print the list of threads running in your local session and all your other logged-in sessions
Print the CREATE statement that created table or view
Print the version of the MySQL server
Print the current date
Print the current time
Print the current date and time
Print the current user@hostname that is logged in
Print status information about server and current connection
SELECT table_schema AS "Name",
SUM(data_length+index_length)/1024/1024 AS "Size in Mb"
FROM information_schema.tables GROUP BY table_schema;

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

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

Display the size of database

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

Display data and index size of all tables of database

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

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

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

Display the amount of InnoDB data in all databases

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

Print name and engine of all tables in database

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

Kill all connections belonging to user

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

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

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

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

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

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

SELECT MAX(CHAR_LENGTH(field)) FROM table;

Display the longest string stored in field

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

Display the list of views in database

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

Display the differences between the contents of two tables table1 and table2 (assuming the tables are composed of 3 columns each)
How to resync a master-slave replication

1. On the master, on terminal 1:
   ```
   mysql -u root -p
   RESET MASTER;
   FLUSH TABLES WITH READ LOCK;
   SHOW MASTER STATUS;
   ```
   Note the values of MASTER_LOG_FILE and MASTER_LOG_POS; these values will need to be copied on the slave

2. On the master, on terminal 2:
   ```
   mysqldump -u root -p --all-databases > /path/to/dump.sql
   ```
   It is not necessary to wait until the dump completes

3. On the master, on terminal 1:
   ```
   UNLOCK TABLES;
   ```

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

5. On the slave:
   ```
   mysql -u root -p
   STOP SLAVE;
   SOURCE /path/to/dump.sql;
   RESET SLAVE;
   CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin.nnnnnn', MASTER_LOG_POS=mm;
   START SLAVE;
   SHOW SLAVE STATUS;
   ```

How to recover the MySQL root password

1. Stop the MySQL server

2. Restart the MySQL server skipping the grant tables
   ```
   mysqld_safe --skip-grant-tables --skip-networking &
   ```

3. Connect to the MySQL server passwordlessly
   ```
   mysql -u root
   ```

4. Reload the grant tables
   ```
   FLUSH PRIVILEGES;
   ```

5. Change the root password
   ```
   SET PASSWORD FOR 'root'@'localhost' = PASSWORD('newpassword');
   ```

6. Stop the MySQL server and restart it normally
PostgreSQL (aka Postgres) is an open source object-relational database. By default it listens for connections on TCP port 5432.

```
\list
\l
\list+
\l+
\connect database
\c database
\q
```

List all databases

List all databases, displaying database size and description

Connect to database

Quit

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

1. Set up PostgreSQL
   ```
   postgresql-setup initdb
   ```

2. Change the password of the postgres shell user
   ```
   passwd postgres
   ```

3. Create the user shell user
   ```
   useradd user
   ```

4. Switch to the postgres shell user and connect to PostgreSQL
   ```
   su - postgres
   psql -U postgres
   ```

5. In PostgreSQL, create the user
   ```
   CREATE ROLE user WITH LOGIN;
   \password user
   \q
   ```

6. Create a database owned by user
   ```
   createdb -E utf8 -l C -T template0 database -O user
   ```

7. Switch to the postgres shell user and connect to PostgreSQL
   ```
   su - postgres
   psql -U postgres
   ```

8. In PostgreSQL, grant the necessary privileges on database to user
   ```
   GRANT ALL PRIVILEGES ON DATABASE database TO user;
   \q
   ```

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

<table>
<thead>
<tr>
<th>Display Manager</th>
<th>Configuration files</th>
<th>Display Manager greeting screen</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>xdm</code></td>
<td><code>/etc/x11/xdm/Xaccess</code></td>
<td>Control inbound requests from remote hosts</td>
</tr>
<tr>
<td></td>
<td><code>/etc/x11/xdm/Xresources</code></td>
<td>Configuration settings for X applications and the login screen</td>
</tr>
<tr>
<td></td>
<td><code>/etc/x11/xdm/Xservers</code></td>
<td>Association of X displays with local X server software, or with X terminals via XDMCP</td>
</tr>
<tr>
<td></td>
<td><code>/etc/x11/xdm/Xsession</code></td>
<td>Script launched by xdm after login</td>
</tr>
<tr>
<td></td>
<td><code>/etc/x11/xdm/Xsetup_0</code></td>
<td>Script launched before the graphical login screen</td>
</tr>
<tr>
<td></td>
<td><code>/etc/x11/xdm/xdm-config</code></td>
<td>Association of all xdm configuration files</td>
</tr>
<tr>
<td><code>gdm</code></td>
<td><code>/etc/gdm/gdm.conf or /etc/gdm/custom.conf</code></td>
<td>Configured via gdmsetup</td>
</tr>
<tr>
<td><code>kdm</code></td>
<td><code>/etc/kde/kdm/kdmrc</code></td>
<td>Configured via kdm_config</td>
</tr>
</tbody>
</table>

```
/etc/init.d/xdm start  
/etc/init.d/gdm start  
/etc/init.d/kdm start
```

Start the appropriate Display Manager

```
xorgconfig  
Xorg -configure
xorgcfg      
system-config-display
X -version
xdpyinfo
xwininfo
xhost + 10.3.3.3  
xhost - 10.3.3.3  
switchdesk gde
 gnome-shell --version
```

Configure X (text mode)

Configure X (graphical mode)

Show which version of X is running

Display information about the X server

Display information about windows

Add or remove 10.3.3.3 to the list of hosts allowed to make X connections to the local machine

Switch to the GDE Display Manager at runtime

Show which version of GNOME is running

```
/etc/X11/xorg.conf  
~/.Xresources  
$DISPLAY
```

Configuration file for X

Configuration settings for X applications, in the form `program*resource: value`

Environment variable defining the display name of the X server, in the form `hostname:displaynumber.screennumber`

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

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

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

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

xdotool getwindowfocus

xdotool selectwindow

xdotool key --window 12345678 Return

xprop

xprop | grep WM_CLASS

xrandr

xrandr -q

xrandr --output eDP1 --right-of VGA1

xsel

xsel -b < file

xsel -b -a < file

xsel -b -o

cat file | xclip -i

mkfontdir

xset fp+ /usr/local/fonts

xfs

fc-cache

X automation tool

Get the ID of the currently focused window (if run in command line, it is the terminal where this command is typed)

Pop up an X cursor and get the ID of the window selected by it

Simulate a RETURN keystroke inside window ID 12345678

X property displayer. Pops up a cursor to select a window

Get process name and GUI application name of the selected window

Show screen(s) size and resolution

Extend the screen on an additional VGA physical screen situated to the left

Manipulate the X selection (primary, secondary, and clipboard)

Copy the contents of a file to the X clipboard

Append the contents of a file to the X clipboard

Output onscreen the contents of the X clipboard

Copy the contents of a file to the X clipboard

Catalog the newly installed fonts in the new directory

Dynamically add new installed fonts in /usr/local/fonts to the X server

Start the X font server

Install fonts and build font information cache
<table>
<thead>
<tr>
<th>Key</th>
<th>Main</th>
<th>Latin 1</th>
<th>Latin 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>BackSpace</td>
<td>ff08</td>
<td>space</td>
<td>0020</td>
</tr>
<tr>
<td>Tab</td>
<td>ff09</td>
<td>exclam</td>
<td>0021</td>
</tr>
<tr>
<td>Linefeed</td>
<td>ff0a</td>
<td>quotedbl</td>
<td>0022</td>
</tr>
<tr>
<td>Clear</td>
<td>ff0b</td>
<td>numbersign</td>
<td>0023</td>
</tr>
<tr>
<td>Return</td>
<td>ff0d</td>
<td>dollar</td>
<td>0024</td>
</tr>
<tr>
<td>Pause</td>
<td>ff13</td>
<td>percent</td>
<td>0025</td>
</tr>
<tr>
<td>Scroll_Lock</td>
<td>ff14</td>
<td>ampersand</td>
<td>0026</td>
</tr>
<tr>
<td>Sys_Req</td>
<td>ff15</td>
<td>apostrophe</td>
<td>0027</td>
</tr>
<tr>
<td>Escape</td>
<td>ff1b</td>
<td>quoteright</td>
<td>0027</td>
</tr>
<tr>
<td>Delete</td>
<td>ffff</td>
<td>parenleft</td>
<td>0028</td>
</tr>
<tr>
<td></td>
<td></td>
<td>asterisk</td>
<td>0029</td>
</tr>
<tr>
<td></td>
<td></td>
<td>greater</td>
<td>003e</td>
</tr>
<tr>
<td></td>
<td></td>
<td>question</td>
<td>003f</td>
</tr>
<tr>
<td></td>
<td></td>
<td>select</td>
<td>0060</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cent</td>
<td>0061</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sterling</td>
<td>0063</td>
</tr>
<tr>
<td></td>
<td></td>
<td>currency</td>
<td>0064</td>
</tr>
<tr>
<td></td>
<td></td>
<td>yen</td>
<td>0065</td>
</tr>
<tr>
<td></td>
<td></td>
<td>yencanbar</td>
<td>0066</td>
</tr>
<tr>
<td></td>
<td></td>
<td>section</td>
<td>007a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>nosign</td>
<td>008d</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hyphen</td>
<td>00ad</td>
</tr>
<tr>
<td></td>
<td></td>
<td>registered</td>
<td>00ae</td>
</tr>
<tr>
<td></td>
<td></td>
<td>macron</td>
<td>00af</td>
</tr>
<tr>
<td></td>
<td></td>
<td>degree</td>
<td>00b0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>plusminus</td>
<td>00b1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>twosuperior</td>
<td>00b2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>threesuperior</td>
<td>00b3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>acute</td>
<td>00b4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>acute</td>
<td>00b5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>acute</td>
<td>00b6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>plus</td>
<td>00b7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>periodcentered</td>
<td>00b7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>periodcentered</td>
<td>00b7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00bd</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00be</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00bf</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00c0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00c1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00c2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00c3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00c4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00c5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00c6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00c7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00c8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00c9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00ca</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00cb</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00cc</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00cd</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00ce</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00cf</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00d0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00d1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00d2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00d3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00d4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00d5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00d6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00d7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00d8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00d9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00da</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00db</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00dc</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00dd</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00de</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00df</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00e0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00e1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00e2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00e3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00e4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00e5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00e6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00e7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00e8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00e9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00ea</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00ec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00ed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00ee</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00ef</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00f0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00f1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00f2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00f3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00f4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00f5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00f6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00f7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00f8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00f9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00fa</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00fb</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00fc</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00fd</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00fe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>percent</td>
<td>00ff</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Login name</th>
<th>Hashed password (obsolete), or x if password is in /etc/shadow</th>
<th>UID – User ID</th>
<th>GID – Default Group ID</th>
<th>GECOS field – Information about the user: Full name, Room number, Work phone, Home phone, Other</th>
<th>Home directory of the user</th>
<th>Login shell (if set to /sbin/nologin or /bin/false, user will be unable to log in)</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>x:0:0:/root:/bin/bash</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bin</td>
<td>x:1:1:/bin:/bin/bash</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jdoe</td>
<td>x:500:100:John Doe,,555-1234,,:/home/jdoe:/bin/bash</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### /etc/shadow

<table>
<thead>
<tr>
<th>Login name</th>
<th>Hashed password (Œ if account is disabled, ! or !! if no password is set, prefixed by ! if the account is locked). Composed of the following subfields separated by $:</th>
<th>Days before password may be changed; if 0, user can change the password at any time</th>
<th>Days after which password must be changed</th>
<th>Days before password expiration that user is warned</th>
<th>Days after password expiration that account is disabled</th>
<th>Date of account disabling (in number of days since 1 January 1970)</th>
<th>Reserved field</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>$6$qk8JmJHf$X9GF0Z/i9LZP4Kldu6.D3cx2pXA:15537:0:99999:7:::</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bin</td>
<td>!$6:15637:0:99999:7:::</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jdoe</td>
<td>!$6$YOihl0tQ$KxeeuKHEK8e3jCUdw9Rxy3Wu53:15580:0:99999:7::15766:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### /etc/group

<table>
<thead>
<tr>
<th>Group name</th>
<th>Encrypted password, or x if password is in /etc/gshadow</th>
<th>GID – Group ID</th>
<th>Group members (if this is not their Default Group)</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>x:root:root</td>
<td></td>
<td></td>
</tr>
<tr>
<td>jdoe</td>
<td>x:501</td>
<td></td>
<td></td>
</tr>
<tr>
<td>staff</td>
<td>x:530:jdoe,asmith</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### /etc/gshadow

<table>
<thead>
<tr>
<th>Group name</th>
<th>Encrypted password, or ! if no password is set (default)</th>
<th>Group administrators</th>
<th>Group members</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>x:root:root</td>
<td></td>
<td></td>
</tr>
<tr>
<td>jdoe</td>
<td>!:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>staff</td>
<td>0cfz7IpLhWL9i:root,jdoe</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

```
useradd -m user
useradd -mc "Name Surname" user
useradd -ms /bin/ksh user
useradd -D
```

Create a user account, creating and populating his homedir from `/etc/skel`

Create a user account, specifying his full name

Create a user account, specifying his login shell

Show default values for user account creation, as specified in `/etc/login.defs` and `/etc/default/useradd`

```
usermod -c "Name Surname" user
usermod -L user
usermod -U user
```

Modify the GECOS field of a user account

Lock a user account

Unlock a user account

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

```
userdel -r user
```

Delete a user and his homedir

```
chfn user
chsh user
```

Change the GECOS field of a user

Change the login shell of a user

```
passwd user
passwd -l user
passwd -S user
```

Change the password of a user

Lock a user account

Show information about a user account: username, account status (L=locked, P=password, NP=no password), date of last password change, min age, max age, warning period, inactivity period in days

```
chage -E 2022-02-14 user
chage -d 13111 user
chage -d 0 user
chage -M 30 user
chage -m 7 user
chage -W 15 user
chage -I 3 user
chage -l user
```

Change the password expiration date; account will be locked at that date

Change the date (in number of days since 1 January 1970) of last password change

Force the user to change password at his next login

Change the max number of days during which a password is valid

Change the min number of days between password changes

Change the number of days before password expiration that the user will be warned

Change the number of days after password expiration before the account is locked

List password aging information for a user

```
chpasswd
```

Tool for batch update of passwords. Reads from stdin a list of `username:password`

```
vipw
vigr
```

Edit manually `/etc/passwd`, `/etc/shadow`, `/etc/group`, or `/etc/gshadow`

```
adduser
deluser
```

User-friendly front-end commands for user management

```
system-config-users (Red Hat)
```

GUI for user and group management
groupadd group

Create a group

groupmod -n newname oldname

Change a group name

groupdel group

Delete a group

gpasswd group

Set or change the password of a group

gpasswd -a user group

Add a user to a group

gpasswd -d user group

Delete a user from a group

gpasswd -A user group

Add a user to the list of administrators of the group

addgroup
delgroup

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

* as recommended by the Linux Standard Base core specifications

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

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

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

* `whoami` Print your username (as effective UID)
* `id` Print your real and effective UID and GID, and the groups you are a member of
* `id -u` Print your effective UID
* `id user` Print UID, GID, and groups information about *user*
Sudo is a mechanism that allows running a command as another user. Sudo access rights are defined in the sudoers files `/etc/sudoers` and `/etc/sudoers.d/*`; these files must be edited only via `visudo`. Commands run by sudo users are logged via syslog on `/var/log/auth.log` (Debian) or `/var/log/secure` (Red Hat).

```
sudo -u user command                     Run command as user
sudo command                            Run command as root
sudo -u root command                    Login on an interactive shell as root
sudo su -                               Login as user with a shell, even if the user's shell is `/sbin/nologin` or similar
sudo -l                                 List the allowed commands for the current user
sudo !!                                 Run again the last command, but this time as root

sudoedit /etc/passwd                     Edit safely a file (in this case, `/etc/passwd`) according to security policies. It is recommended to allow users use this command instead of sudoing text editors as root on protected files, because the editor might be able to spawn a shell, causing security issues
sudo -e /etc/passwd                      Run again the last command, but this time as root

visudo                                  Edit safely the sudoers file
visudo -c                                Check the sudoers file for syntax errors, unused aliases, etc.

su user                                  Run a shell as user
su                                       Run a shell as root
su root                                  Pass a single command to the shell
su -                                      Ensure that the spawned shell is a login shell, hence running login scripts and setting the correct environment variables. Recommended option
su -l                                     

gksudo -u root command                    GUI front-ends to `su` and `sudo` used to run an X Window command or application as root. Pops up a requester prompting the user for root's password
gksu -u root -l                           

runuser -u user command                   Run command as user. Can be launched only by root
```
**Terminals**

- **chvt n**  
  Make `/dev/tty` the foreground terminal
  
- **vlock**  
  Lock the virtual console (terminal)

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

- **stty -ixon**  
  Disable XON/XOFF flow control

- **clear**  
  Clear the terminal screen

- **tmux**  
  Terminal multiplexer

- **nohup script.sh**  
  Prevent a process from terminating (receiving a SIGHUP) when its parent Bash dies. When a Bash shell is terminated cleanly via `exit`, its jobs will become child of the Bash's parent and will continue running. When a Bash shell is killed instead, it issues a SIGHUP to its children which will terminate

**Screen**  

Screen manager that multiplexes a single virtual VT100/ANSI terminal between multiple processes or shells.

When the connection to a terminal is lost (e.g. because the terminal is closed manually, the user logs out, or the remote SSH session goes into timeout), a SIGHUP is sent to the shell and from there to all running child processes which are therefore terminated. The `screen` command starts an interactive shell screen session, to which the user will be able to reattach later

- **screen -S sessionname**  
  Start a screen session with the specified session name

- **screen command**  
  Start the specified command in a screen session; session will end when the command exits

- **screen -list**  
  Show the list of detached screen sessions

- **screen -r pid.tty.host**  
  Resume a detached screen session

- **screen -r owner/pid.tty.host**  
  Resume the last detached screen session

- **screen -d -R**  
  Detach a remote screen session and reattach your current terminal to it

**How to detach an already running job that was not started in a screen session**  

(These commands detach the job from its parent shell, so that the job will not be killed when the terminal is closed)

1. **CTRL`A`**  
   Suspend the job

2. **bg**  
   Send the job to background

3. **jobs**  
   Show the number (let's assume is `n`) of the backgrounded job

4. **disown -h %n**  
   Mark job `n` so it will not receive a SIGHUP from its parent shell

or

1. **screen**  
   Start a screen session

2. **reptyr pid**  
   Attach the job with process ID `pid` to the new terminal (screen session)
write user

Write interactively a message to the terminal of user (which must be logged in)

echo "Message" | write user

Write a message to the terminal of user (which must be logged in)

wall

Write interactively a message to the terminal of all logged in users

echo "Message" | wall

Write a message to the terminal of all logged in users

talk user

Open an interactive chat session with user (which must be logged in)

mesg y
chmod g+w $(tty)

Allow the other users to message you via write, wall, and talk

mesg n
chmod g-w $(tty)

Disallow the other users to message you via write, wall, and talk

mesg

Display your current message permission status

mesg works by enabling/disabling the group write permission of your terminal device, which is owned by system group tty. The root user is always able to message users.
cron is a job scheduler, allowing the repeated execution of commands specified in crontab files. The crond daemon checks the crontab files every minute and runs the command as the specified user at the specified times. It is not necessary to restart crond after the modification of a crontab file, as the changes will be reloaded automatically.

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

```
/etc/crontab
/etc/cron.d/*
/etc/cron.hourly/
/etc/cron.daily/
/etc/cron.weekly/
/etc/cron.monthly/
/var/spool/cron/
```

System-wide crontab files

```
/var/spool/cron/user
```

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

```
crontab -e
```

Edit your user crontab file

```
crontab -l
```

List the contents of your crontab file

```
crontab -e -u user
```

Edit the crontab file of another user (command available only to the superuser)

<table>
<thead>
<tr>
<th>m</th>
<th>h</th>
<th>dom</th>
<th>mon</th>
<th>dow</th>
<th>user</th>
<th>command</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>6</td>
<td>*</td>
<td>*</td>
<td>1</td>
<td>root</td>
<td>/opt/script1.sh</td>
</tr>
</tbody>
</table>

every Monday at 6:25 AM

| */5 | 16 | *   | *   | *   | root | /opt/script2.sh |

from 4:00 to 4:55 PM every 5 minutes every day

| 0,30 | 7 | 25  | 12  | *   | jdoe | /home/jdoe/foo.sh |

at 7:00 and 7:30 AM on 25th December

| 3   | 17 | *   | 1-5 | root | /root/bar.sh |

at 5:03 PM every day, from Monday to Friday

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

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

```
/var/spool/anacron/jobid
```

Date of the last execution of the anacron job identified by jobid

```
/etc/anacrontab
```

<table>
<thead>
<tr>
<th>period</th>
<th>delay</th>
<th>job-identifier</th>
<th>command</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>10</td>
<td>cron.weekly</td>
<td>/opt/script3.sh</td>
</tr>
</tbody>
</table>

If the job has not been run in the last 7 days, wait 10 minutes and then execute the command

<table>
<thead>
<tr>
<th>period</th>
<th>delay</th>
<th>job-identifier</th>
<th>command</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

period, in days, during which the command was not executed
delay, to wait, in minutes, before execution of the command
job-identifier job identifier in anacron messages; should be unique for each anacron job
command command that will be executed
**at** is used for scheduled execution of commands that must run only once. Execution of these commands is the duty of the **atd** daemon.

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

```bash
at 5:00pm tomorrow script.sh
at -f listofcommands.txt 5:00pm tomorrow
echo "rm file" | at now+2 minutes
```

Execute a command once at the specified time (absolute or relative)

```bash
at -l
atq
at -d 3
atrm 3
```

List the scheduled jobs
Remove job number 3 from the list

**batch**

Schedule execution of a command for when the system is not too charged. Reads a command from stdin and runs it when the system’s load average falls below 0.8
<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bc</td>
<td>Calculator</td>
</tr>
<tr>
<td>dc</td>
<td>Calculator featuring unlimited precision arithmetic</td>
</tr>
<tr>
<td>factor</td>
<td>Finds the prime factors of a number</td>
</tr>
<tr>
<td>units</td>
<td>Converter of quantities between different units</td>
</tr>
<tr>
<td>cal</td>
<td>Calendar</td>
</tr>
<tr>
<td>banner</td>
<td>Print a text in large letters made of the character #</td>
</tr>
<tr>
<td>figlet</td>
<td>Print a text in large letters, in a specific font</td>
</tr>
<tr>
<td>toilet</td>
<td>Print a text in large colorful letters, in a specific font</td>
</tr>
<tr>
<td>lolcat</td>
<td>Print a text in rainbow coloring</td>
</tr>
<tr>
<td>fortune</td>
<td>Print a random aphorism, like those found in fortune cookies</td>
</tr>
<tr>
<td>sensors</td>
<td>Print sensor chips information (e.g. temperature)</td>
</tr>
<tr>
<td>beep</td>
<td>Produce a beep from the machine's speakers</td>
</tr>
<tr>
<td>speaker-test</td>
<td>Speaker test tone generator for the ALSA (Advanced Linux Sound Architecture) framework</td>
</tr>
<tr>
<td>on_ac_power</td>
<td>Return 0 (true) if machine is connected to AC power, 1 (false) if on battery. Useful for laptops</td>
</tr>
<tr>
<td>ipcalc</td>
<td>IP addresses calculator</td>
</tr>
<tr>
<td>pwgen</td>
<td>Random password generator</td>
</tr>
<tr>
<td>pwgen</td>
<td>Random password generator with controllable quality</td>
</tr>
<tr>
<td>uuidgen</td>
<td>Generator of UUIDs (random or time-based)</td>
</tr>
<tr>
<td>haveged</td>
<td>Generator of random numbers via the HAVEGE (Hardware Volatile Entropy Gathering and Expansion) algorithm. Can be run as a daemon to automatically replenish /dev/random whenever the supply of random bits in the random device gets too low</td>
</tr>
<tr>
<td>aspell</td>
<td>Spell checker</td>
</tr>
<tr>
<td>cloc</td>
<td>Count lines of source code</td>
</tr>
<tr>
<td>gnome-terminal</td>
<td>GNOME shell terminal</td>
</tr>
<tr>
<td>conky</td>
<td>Highly configurable system monitor widget with integration for audio player, email, and news</td>
</tr>
<tr>
<td>gkrellm</td>
<td>System monitor widget</td>
</tr>
</tbody>
</table>
Localization

<table>
<thead>
<tr>
<th>Locale environment variables</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LANG</strong></td>
</tr>
<tr>
<td><strong>LANGUAGE</strong></td>
</tr>
<tr>
<td><strong>LC_NUMERIC</strong></td>
</tr>
<tr>
<td><strong>LC_TIME</strong></td>
</tr>
<tr>
<td><strong>LC_COLLATE</strong></td>
</tr>
<tr>
<td><strong>LC_MONETARY</strong></td>
</tr>
<tr>
<td><strong>LC_MESSAGES</strong></td>
</tr>
<tr>
<td><strong>LC_PAPER</strong></td>
</tr>
<tr>
<td><strong>LC_NAME</strong></td>
</tr>
<tr>
<td><strong>LC_ADDRESS</strong></td>
</tr>
<tr>
<td><strong>LC_TELEPHONE</strong></td>
</tr>
<tr>
<td><strong>LC_MEASUREMENT</strong></td>
</tr>
<tr>
<td><strong>LC_IDENTIFICATION</strong></td>
</tr>
<tr>
<td><strong>LC_ALL</strong></td>
</tr>
</tbody>
</table>

The values of these locale environment variables are in the format `language_territory.encoding` e.g. `en_US.UTF-8`. The list of supported locales is stored in `/usr/share/i18n/SUPPORTED`.

locale

locale-gen it_IT.UTF-8

apt-get install manpages-it language-pack-it (Debian)

iconv -f IS6937 -t IS8859 filein > fileout

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.
**System time**

**date**

- `date` Show current date and time
- `date -d "9999 days ago"` Calculate a date and show it
- `date -d "1970/01/01 + 4242"`
- `date +"%F %H:%M:%S"`
- `date +"%s"`
- `date -s "20130305 23:30:00"`
- `date 030523302013`

**timedatectl**

- `timedatectl` Show current date and time
- `timedatectl set-time 2013-03-05`
- `timedatectl set-time 23:30`
- `timedatectl list-timezones`

**tzdump GMT**

- `tzdump` Show current date and time in the GMT timezone

**tzselect tzconfig dpkg-reconfigure tzdata timedatectl set-timezone timezone** (Debian)

**/etc/timezone (Debian)**

- `Timezone`

**/etc/localtime (Red Hat)**

- `Timezone, a symlink to the appropriate timezone file in /usr/share/zoneinfo/`

**ntpd**

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

**ntpq -p timeserver**

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

**ntpdate timeserver**

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

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

**chronyd chronyc**

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

**hwclock --show hwclock -r**

- `hwclock --show` Show the hardware clock
- `hwclock -r` Set the system time from the hardware clock

**hwclock --hctosys hwclock -s**

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

**hwclock --systohc hwclock -w**

- `hwclock --systohc` Indicate that the hardware clock is kept in Coordinated Universal Time
- `hwclock -w` Indicate that the hardware clock is kept in local time
syslogd
rsyslogd (Ubuntu 14)

daemon logging events from user processes

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

auth or security†
authpriv
cron
daemon
dean
kern
lpr
mail
mark
news
syslog
user
uucp
local0 ... local7 (custom)

### Level
Severity of the message

emerg or panic† (highest)
alert
crit
err or error†
warning or warn†
otice
info
debug (lowest)
none (facility disabled)

### Action
Destination of the message

file
@host
user1,user2,user3
*

† = deprecated

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

`logger -p auth.info "Message"`

Send a message to syslog with facility "auth" and priority "info"

`logrotate`

Rotate logs. It gzips, renames, and eventually deletes old logfiles according to the configuration files `/etc/logrotate.conf` and `/etc/logrotate.d/*`. It is usually scheduled as a daily cron job

`/var/log/messages`

Global system logfile

`/var/log/dmesg`

Kernel ring buffer information

`/var/log/kern.log`

Kernel log

`/var/log/boot.log`

Information logged during boot
Mailbox formats

<table>
<thead>
<tr>
<th>mbox</th>
<th>Each mail folder is a single file, storing multiple email messages.</th>
<th>Advantages: universally supported; fast search inside a mail folder. Disadvantages: issues with file locking; possible mailbox corruption.</th>
<th>$HOME/Mail/folder</th>
</tr>
</thead>
</table>

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HELO</td>
<td>Initiate the conversation and identify client host to server.</td>
</tr>
<tr>
<td>EHLO</td>
<td>Like HELO, but tell server to use Extended SMTP.</td>
</tr>
<tr>
<td>MAIL</td>
<td>Specify mail sender.</td>
</tr>
<tr>
<td>RCPT</td>
<td>Specify mail recipient.</td>
</tr>
<tr>
<td>DATA</td>
<td>Specify data to send. Ended with a dot on a single line.</td>
</tr>
<tr>
<td>QUIT</td>
<td>Disconnect.</td>
</tr>
<tr>
<td>RSET</td>
<td>List all available commands.</td>
</tr>
<tr>
<td>NOOP</td>
<td>Empty command.</td>
</tr>
<tr>
<td>VRFY</td>
<td>Verify the existence of an e-mail address (this command should not be implemented, for security reasons).</td>
</tr>
<tr>
<td>EXPN</td>
<td>Check mailing list membership.</td>
</tr>
</tbody>
</table>

### SMTP response codes

<table>
<thead>
<tr>
<th>First digit</th>
<th>Description</th>
<th>Second digit</th>
<th>Description</th>
<th>Third digit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Command accepted, but not processed until client sends confirmation.</td>
<td>0</td>
<td>Syntax error or command not implemented.</td>
<td>Specifies further the response.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Command successfully completed.</td>
<td>1</td>
<td>Informative response in reply to a request for information.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Command accepted, but not processed until client sends more information.</td>
<td>2</td>
<td>Connection response in reply to a data transmission.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Command failed due to temporary errors.</td>
<td>3</td>
<td>Status response in reply to a mail transfer operation.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Command failed due to permanent errors.</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Examples:**

**HELO xyz.linux.org**

**EHLO xyz.linux.org**

**MAIL FROM: alice@linux.org**

**RCPT TO bob@foobar.com**

**RCPT TO carol@quux.net**

**DATA**

**QUIT**

**HELP**

**NOOP**

**VRFY alice@linux.org**

**EXPN mailinglist**
Sendmail is an MTA distributed as a monolithic binary file. Previous versions used to run SUID root, which caused many security problems; recent versions run SGID smmsp, the group that has write access on the mail queue.

Sendmail uses smrsh, a restricted shell, to run some external programs.

Configuration files (must not be edited by hand):

<table>
<thead>
<tr>
<th>etc/mail/</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>submit.cf</td>
<td>Sendmail local mail transfer configuration file</td>
</tr>
<tr>
<td>sendmail.cf</td>
<td>Sendmail MTA configuration file</td>
</tr>
</tbody>
</table>

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

Database files (must not be edited by hand):

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

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

Temporary mailqueue files (where m is the Message ID):

<table>
<thead>
<tr>
<th>/var/spool/mqueue/</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>dfmnn</td>
<td>Mail body</td>
</tr>
<tr>
<td>qfnnn</td>
<td>Message envelope with headers and routing information</td>
</tr>
<tr>
<td>Qfnnn</td>
<td>Message envelope if abandoned</td>
</tr>
<tr>
<td>hfnnn</td>
<td>Message envelope if held / quarantined by a milter (i.e. mail filter)</td>
</tr>
<tr>
<td>tfnnn</td>
<td>Temporary file</td>
</tr>
<tr>
<td>lfnnn</td>
<td>Lock file</td>
</tr>
<tr>
<td>nfnnn</td>
<td>Backup file</td>
</tr>
<tr>
<td>xfnnn</td>
<td>Transcript of delivery attempts</td>
</tr>
</tbody>
</table>

newaliases
sendmail -bi  Update the aliases database. Must be run after any change to /etc/aliases
mailq
sendmail -bp  Examine the mail queue
sendmail -bt  Run Sendmail in test mode
sendmail -q   Force a queue run

hoststat
purgetstat     Clear statistics about remote host usage
mailstats      Print statistics about the mailserver
praliases      Display email aliases
Exim is a free MTA, distributed under open source GPL license.

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

exim4 -bp  Examine the mail queue
exim4 -M messageId  Attempt delivery of message
exim4 -Mrm messageId  Remove a message from the mail queue
exim4 -Mvh messageId  See the headers of a message in the mail queue
exim4 -Mvb messageId  See the body of a message in the mail queue
exim4 -Mvc messageId  See a message in the mail queue
exim4 -qf domain  Force a queue run of all queued messages for a domain
exim4 -Rff domain  Attempt delivery of all queued messages for a domain
exim4 -bV  Show version and other info
exinext  Give the times of the next queue run
exigrep  Search through Exim logfiles
exicyclog  Rotate Exim logfiles
Postfix is a fast, secure, easy to configure, open source MTA intended as a replacement for Sendmail. It is implemented as a set of small helper daemons, most of which run in a chroot jail with low privileges. The main ones are:

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

### /var/spool/postfix/

<table>
<thead>
<tr>
<th>Queue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>incoming</td>
<td>Incoming queue. All new mail entering the Postfix queue is written here by the cleanup daemon. Under normal conditions this queue is nearly empty</td>
</tr>
<tr>
<td>active</td>
<td>Active queue. Contains messages ready to be sent. The queue manager places messages here from the incoming queue as soon as they are available</td>
</tr>
<tr>
<td>deferred</td>
<td>Deferred queue. A message is placed here when all its deliverable recipients are delivered, and delivery failed for some recipients for a transient reason. The queue manager scans this queue periodically and puts some messages back into the active queue to retry sending</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Queue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bounce</td>
<td>Message delivery status report about why mail is bounced (non-delivered mail)</td>
</tr>
<tr>
<td>defer</td>
<td>Message delivery status report about why mail is delayed (non-delivered mail)</td>
</tr>
<tr>
<td>trace</td>
<td>Message delivery status report (delivered mail)</td>
</tr>
</tbody>
</table>

**postfix reload**  
Reload configuration

**postconf -e 'mydomain = example.org'**  
Edit a setting in the Postfix configuration

**postconf -l**  
List supported mailbox lock methods

**postconf -m**  
List supported database types

**postconf -v**  
Increase logfile verbosity

**postmap dbtype:textfile**  
Manage Postfix lookup tables, creating a hashed map file of database type dbtype from textfile

**postmap hash:/etc/postfix/transport**  
Regenerate the transport database

**postalias**  
Convert /etc/aliases into the aliases database file /etc/aliases.db

**postsuper**  
Operate on the mail queue

**postqueue**  
Unprivileged mail queue manager
### Postfix Configuration

**/etc/postfix/main.cf** | Postfix main configuration file
---|---
mydomain = example.org | This system's domain
myorigin = $mydomain | Domain from which all sent mail will appear to originate
myhostname = foobar.$mydomain | This system's hostname
inet_interfaces = all | Network interface addresses that this system receives mail on. Value can also be `localhost`, `all`, or `loopback-only`
proxy_interfaces = 1.2.3.4 | Network interface addresses that this system receives mail on by means of a proxy or NAT unit
mynetworks = 10.3.3.0/24 !10.3.3.66 | Domains from which Postfix will accept received mail. Value can also be a lookup database file e.g. a hashed map
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 | 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:
virtual_alias_maps = /etc/postfix/virtual or
virtual_alias_domains = hash:/etc/postfix/virtual
mailbox_command = /usr/bin/procmail | Use Procmail as MDA

A line beginning with whitespace or tab is a continuation of the previous line.
A line beginning with a `#` is a comment. A `#` not placed at the beginning of a line is not a comment delimiter.

**/etc/postfix/master.cf** | Postfix master daemon configuration file
---|---
# Service | 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
```

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/etc/procmailrc</td>
<td>System-wide recipes</td>
</tr>
<tr>
<td>~/.procmailrc</td>
<td>User's recipes</td>
</tr>
<tr>
<td>procmail -h</td>
<td>List all Procmail flags for recipes</td>
</tr>
<tr>
<td>formail</td>
<td>Utility for email filtering and editing</td>
</tr>
<tr>
<td>lockfile</td>
<td>Utility for mailbox file locking</td>
</tr>
<tr>
<td>mailstat</td>
<td>Utility for generation of reports from Procmail logs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Common parameters, nonspecific to Procmail</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATH=$HOME/bin:/usr/bin:/bin:/usr/sbin:/sbin</td>
</tr>
<tr>
<td>MAILDIR=$HOME/Mail</td>
</tr>
<tr>
<td>DEFAULT=$MAILDIR/Inbox</td>
</tr>
<tr>
<td>LOGFILE=$HOME/.procmaillog</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Procmail recipes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:0h: or :0:</td>
<td>Flag: match headers (default) and use file locking (highly recommended when writing to a file or a mailbox in mbox format)</td>
</tr>
<tr>
<td>* &quot;From: .*{alice</td>
<td>bob}@foobar.[org]</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>Condition: match the header specifying the sender address</td>
</tr>
<tr>
<td>$DEFAULT</td>
<td>Destination: default mailfolder</td>
</tr>
</tbody>
</table>

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

# Blacklisted by SpamAssassin
| :0                       | Flag: file locking not necessary because blackholing to /dev/null           |
| * "X-Spam-Status: Yes   |
| /dev/null                |

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

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

| :0:                      | Condition: match messages larger than 256 Kb                               |
| > 256000                 |
| | /root/myprogram         |

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

| :0c                      | Flag: copy the message and proceed with next recipe                        |
| * "Subject:.*administration |
| ! secretary@domain.com    |

| :0:                      | Destination: forward to specified email address, and (this is ordered by the next recipe) save in the specified mailfolder |
| $MAILDIR/Forwarded       |
The Courier MTA provides modules for ESMTP, IMAP, POP3, webmail, and mailing list services in a single framework. To use Courier, it is necessary first to launch the courier-authlib service, then launch the desired mail service e.g. courier-imap for the IMAP service.

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
</table>
| /usr/lib/courier-imap/etc/ or /etc/courier/ | Courier IMAP daemon configuration
| /usr/lib/courier-imap/etc/pop3d | Courier POP3 daemon configuration

Directory for public and private keys

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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

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

**/usr/lib/courier-imap/etc/imapd**  
Courier IMAP configuration file

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDRESS=0</td>
<td>Address on which to listen. 0 means all addresses</td>
</tr>
<tr>
<td>PORT=127.0.0.1.900,192.168.0.1.900</td>
<td>Port number on which connections are accepted. In this case, accept connections on port 900 on IP addresses 127.0.0.1 and 192.168.0.1</td>
</tr>
<tr>
<td>AUTHSERVICE143=imap</td>
<td>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</td>
</tr>
<tr>
<td>MAXGAEMONS=40</td>
<td>Maximum number of IMAP servers started</td>
</tr>
<tr>
<td>MAXPERIP=20</td>
<td>Maximum number of connections to accept from the same IP address</td>
</tr>
<tr>
<td>PIDFILE=/var/run/courier/imapd.pid</td>
<td>PID file for couriertcpd</td>
</tr>
<tr>
<td>TCPDOPTS=&quot;-nodnslookup -noidentlookup&quot;</td>
<td>Miscellaneous couriertcpd options. Should not be changed</td>
</tr>
<tr>
<td>LOGGEROPTS=&quot;-name=imapd&quot;</td>
<td>Options for courierlogger</td>
</tr>
<tr>
<td>DEFDOMAIN=&quot;@example.com&quot;</td>
<td>Optional default domain. If the username does not contain the first character of DEFDOMAIN, then it is appended to the username. If DEFDOMAIN and DOMAINESEP are both set, then DEFDOMAIN is appended only if the username does not contain any character from DOMAINESEP</td>
</tr>
<tr>
<td>IMAP_CAPABILITY=&quot;IMAP4rev1 UIDPLUS / CHILDREN NAMESPACE THREAD=ORDEREDSUBJECT \ THREAD=REFERENCES SORT QUOTA IDLE&quot;</td>
<td>Specifies what most of the response should be to the CAPABILITY command</td>
</tr>
<tr>
<td>IMAP_KEYWords=1</td>
<td>Enable or disable custom IMAP keywords. Possible values are: 0 disable keywords 1 enable keywords 2 enable keywords with a slower algorithm</td>
</tr>
<tr>
<td>IMAP_ACL=1</td>
<td>Enable or disable IMAP ACL extension</td>
</tr>
<tr>
<td>SMAP_CAPABILITY=SMAP1</td>
<td>Enable the experimental Simple Mail Access Protocol extensions</td>
</tr>
<tr>
<td>IMAP_PROXY=0</td>
<td>Enable or disable proxying</td>
</tr>
<tr>
<td>IMAP_PROXY_FOREIGN=0</td>
<td>Proxying to non-Courier servers. Resends the CAPABILITY command after logging in to remote server. May not work with all IMAP clients</td>
</tr>
<tr>
<td>IMAP_IDLE_TIMEOUT=60</td>
<td>How often, in seconds, the server should poll for changes to the folder while in IDLE mode</td>
</tr>
<tr>
<td>IMAP_CHECK_ALL_FOLDERS=0</td>
<td>Enable or disable server check for mail in every folder</td>
</tr>
<tr>
<td>IMAP_umask=022</td>
<td>Set the umask of the server process. This value is passed to the umask command. Mostly useful for shared folders, where file permissions of the messages may be important</td>
</tr>
<tr>
<td>IMAP_ULIMITD=131072</td>
<td>Set the upper limit of the size of the data segment of the server process, in Kb. This value is passed to the ulimit -d command. Used as an additional safety check to stop potential DoS attacks that exploit memory leaks to exhaust all the available RAM on the server</td>
</tr>
<tr>
<td>IMAP_USELOCKS=1</td>
<td>Enable or disable dot-locking to support concurrent multiple access to the same folder. Strongly recommended when using shared folders</td>
</tr>
<tr>
<td>IMAP_SHAREDINDEXFILE=&quot;/etc/courier/shared/index&quot;</td>
<td>Index of all accessible folders. This setting should normally not be changed</td>
</tr>
<tr>
<td>IMAP_TRASHFOLDERNAME=Trash</td>
<td>Trash folder</td>
</tr>
<tr>
<td>IMAP_EMPTYTRASH=Trash:7,Sent:30</td>
<td>Purge folders i.e. delete all messages from the specified folders after the specified number of days</td>
</tr>
<tr>
<td>IMAP_MOVE_EXPUNGE_TO_TRASH=0</td>
<td>Enable or disable moving expunged messages to the trash folder (instead of directly deleting them)</td>
</tr>
<tr>
<td>HEADERFROM=X-IMAP-Sender</td>
<td>Save the return address ($SENDER) in the X-IMAP-Sender mail header. This header is added to the sent message, but not in the copy of the message saved in the folder</td>
</tr>
<tr>
<td>MAILDIRPATH=Maildir</td>
<td>Mail directory</td>
</tr>
</tbody>
</table>
Dovecot is an open source, security-hardened, fast, and efficient IMAP and POP3 server. It implements its own high-performance dbox mailbox format. By default, it uses PAM authentication. The script `mkcert.sh` can be used to create self-signed SSL certificates.

<table>
<thead>
<tr>
<th><code>/etc/dovecot.conf</code></th>
<th>Dovecot configuration file</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>base_dir = /var/run/dovecot/</code></td>
<td>Base directory where to store runtime data</td>
</tr>
<tr>
<td><code>protocols = imaps pop3s</code></td>
<td>Protocols to serve. If Dovecot should use <code>dovecot-auth</code>, this can be set to <code>none</code></td>
</tr>
<tr>
<td><code>listen = * , [::]</code></td>
<td>Network interfaces on which to accept connections. In this case, listen to all IPv4 and IPv6 interfaces</td>
</tr>
<tr>
<td><code>disable_plaintext_auth = yes</code></td>
<td>If yes, disable LOGIN command and all other plaintext authentications unless SSL/TLS is used (LOGINDISABLED capability)</td>
</tr>
<tr>
<td><code>shutdown_clients = yes</code></td>
<td>If yes, kill all IMAP and POP3 processes when Dovecot master process shuts down; if no, Dovecot can be upgraded without forcing existing client connections to close</td>
</tr>
<tr>
<td><code>log_path = /dev/stderr</code></td>
<td>Log file to use for error messages, instead of sending them to syslog. In this case, log to stderr</td>
</tr>
<tr>
<td><code>info_log_path = /dev/stderr</code></td>
<td>Log file to use for informational and debug messages. Default value is the same as <code>log_path</code></td>
</tr>
<tr>
<td><code>syslog_facility = mail</code></td>
<td>Syslog facility to use, if logging to syslog</td>
</tr>
<tr>
<td><code>login_dir = /var/run/dovecot/login</code></td>
<td>Directory where the authentication process places authentication UNIX sockets. The login process needs to be able to connect to these sockets</td>
</tr>
<tr>
<td><code>login_chroot = yes</code></td>
<td>Chroot login process to the <code>login_dir</code></td>
</tr>
<tr>
<td><code>login_user = dovecot</code></td>
<td>User for the login process and for access control in the authentication process. This is not the user that will access mail messages</td>
</tr>
<tr>
<td><code>login_process_size = 64</code></td>
<td>Maximum login process size, in Mb</td>
</tr>
<tr>
<td><code>login_process_per_connection = yes</code></td>
<td>If yes, each login is processed in its own process (more secure); if no, each login process processes multiple connections (faster)</td>
</tr>
<tr>
<td><code>login_processes_count = 3</code></td>
<td>Number of login processes to keep for listening for new connections</td>
</tr>
<tr>
<td><code>login_max_processes_count = 128</code></td>
<td>Maximum number of login processes to create</td>
</tr>
<tr>
<td><code>login_max_connections = 256</code></td>
<td>Maximum number of connections allowed per each login process. This setting is used only if <code>login_process_per_connection = no</code>; once the limit is reached, the process notifies master so that it can create a new login process</td>
</tr>
<tr>
<td><code>login_greeting = Dovecot ready.</code></td>
<td>Greeting message for clients</td>
</tr>
<tr>
<td><code>login_trusted_networks = 10.7.7.0/24 10.8.8.0/24</code></td>
<td>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. <code>disable_plaintext_auth</code> is also ignored for these networks</td>
</tr>
<tr>
<td><code>mbox_read_locks = fcntl</code></td>
<td>Locking methods to use for locking mailboxes in mbox format. Possible values are:</td>
</tr>
<tr>
<td><code>mbox_write_locks = dotlock fcntl</code></td>
<td><code>dotlock</code> Create <code>mailbox.lock</code> file; oldest and NSF-safe method</td>
</tr>
<tr>
<td><code>maildir_stat_dirs = no</code></td>
<td>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</td>
</tr>
<tr>
<td><code>dbox_rotate_size = 2048</code></td>
<td>Maximum and minimum file size, in Kb, of a mailbox in dbox format until it is rotated</td>
</tr>
<tr>
<td><code>dbox_rotate_min_size = 16</code></td>
<td>Include configuration file</td>
</tr>
<tr>
<td><code>include /etc/dovecot/conf.d/*.conf</code></td>
<td>Include optional configuration file, and do not report an error if file is not found</td>
</tr>
<tr>
<td><code>include_try /etc/dovecot/extra.conf</code></td>
<td>Include configuration file</td>
</tr>
</tbody>
</table>
### Dovecot mailbox configuration

**/etc/dovecot.conf**

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
</table>
| `mail_location` | Mailbox location, in mbox or Maildir format. Variables: `

| `%u` | username |
| `%n` | user part in `user@domain`, same as `%u` if there is no domain |
| `%d` | domain part in `user@domain`, empty if there is no domain |
| `%h` | home directory |
|
| `namespace shared {` | Definition of a shared namespace, for accessing other users' mailboxes that have been shared. Private namespaces are for users' personal emails. Public namespaces are for shared mailboxes managed by root user |
| `separator = /` | Hierarchy separator to use. It should be the same for all namespaces, and depends on the underlying mail storage format |
| `prefix = shared/%%u/` | Prefix required to access this namespace; must be different for each. In this case, mailboxes are visible under `shared/user@domain/`; the variables `%%n`, `%%d`, and `%%u` are expanded to the destination user |
| `location = maildir:%%h/Maildir:\ INDEX=~/Maildir/shared/%%u` | Mailbox location for other users' mailboxes; it is in the same format as `mail_location` which is also the default for it. `%variable` and `~` expand to the logged in user's data; `%%variable` expands to the destination user's data |
| `inbox = no` | Define whether this namespace contains the INBOX. Note that there can be only one INBOX across all namespaces |
| `hidden = no` | Define whether the namespace is hidden i.e. not advertised to clients via NAMESPACE extension |
| `subscriptions = no` | Namespace handles its own subscriptions; if set to no, the parent namespace handles them and Dovecot uses the default namespace for saving subscriptions. If `prefix` is empty, this should be set to yes |
| `list = children` | Show the mailboxes under this namespace with LIST command, making the namespace visible for clients that do not support the NAMESPACE extension. In this case, lists child mailboxes but hide the namespace prefix; list the namespace only if there are visible shared mailboxes |
| `mail_uid = 666` | UID and GID used to access mail messages |
| `mail_gid = 666` | Group to enable temporarily for privileged operations. Currently this is used only with INBOX when its initial creation or a dotlocking fails |
| `mail_access_groups = tmpmail` | Supplementary groups to with grant access for mail processes. Used typically to set up access to shared mailboxes |
| `lock_method = fcntl` | Locking method for index files. Can be `fcntl`, `flock`, or `dotlock` |
| `first_valid_uid = 500` | 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 |
| `last_valid_uid = 0` | Valid GID range for users; default is non-root. Users with invalid primary GID are not allowed to login |
| `max_mail_processes = 512` | Maximum number of running mail processes. When this limit is reached, new users are not allowed to login |
| `mail_process_size = 256` | Maximum mail process size, in Mb |
| `valid_chroot_dirs =` | List of directories under which chrooting is allowed for mail processes |
| `mail_chroot =` | Default chroot directory for mail processes. Usually not needed as Dovecot does not allow users to access files outside their mail directory |
| `mailbox_idle_check_interval = 30` | Minimum time, in seconds, to wait between mailbox checks. When the IDLE command is running, mailbox is checked periodically for new mails or other changes |
# Dovecot POP/IMAP configuration

<table>
<thead>
<tr>
<th>/etc/dovecot.conf</th>
<th>Dovecot configuration file</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>protocol pop3 {</strong></td>
<td>Block with options for the POP3 protocol</td>
</tr>
<tr>
<td>listen = *:110</td>
<td>Network interfaces on which to accept POP3 connections</td>
</tr>
<tr>
<td>login_executable = /usr/libexec/dovecot/pop3-login</td>
<td>Location of the POP3 login executable</td>
</tr>
<tr>
<td>mail_executable = /usr/libexec/dovecot/pop3</td>
<td>Location of the POP3 mail executable</td>
</tr>
<tr>
<td>pop3_no_flag_updates = no</td>
<td>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</td>
</tr>
<tr>
<td>pop3_lock_session = no</td>
<td>Defines whether to keep the mailbox locked for the whole POP3 session</td>
</tr>
<tr>
<td>pop3_uidl_format = %08Xu%08Xv</td>
<td>POP3 UIDL (Unique Mail Identifier) format to use</td>
</tr>
<tr>
<td><strong>}</strong></td>
<td></td>
</tr>
<tr>
<td><strong>protocol imap {</strong></td>
<td>Block with options for the IMAP protocol</td>
</tr>
<tr>
<td>listen = *:143</td>
<td>Network interfaces on which to accept IMAP and IMAPS connections</td>
</tr>
<tr>
<td>ssl_listen = *:993</td>
<td>Location of the IMAP login executable</td>
</tr>
<tr>
<td>login_executable = /usr/libexec/dovecot/imap-login</td>
<td>Location of the IMAP mail executable</td>
</tr>
<tr>
<td>mail_executable = /usr/libexec/dovecot/imap</td>
<td>Maximum number of IMAP connections allowed for a user from each IP address</td>
</tr>
<tr>
<td>mail_max_userip_connections = 10</td>
<td>Waiting time, in seconds, between &quot;OK Still here&quot; notifications when client is IDLE</td>
</tr>
<tr>
<td>imap_idle_notify_interval = 120</td>
<td></td>
</tr>
<tr>
<td><strong>}</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ssl = yes</strong></td>
<td>SSL/TLS support. Possible values are yes, no, required</td>
</tr>
<tr>
<td><strong>ssl_cert_file = /etc/ssl/certs/dovecot-cert.pem</strong></td>
<td>Location of the SSL certificate</td>
</tr>
<tr>
<td><strong>ssl_key_file = /etc/ssl/private/dovecot-key.pem</strong></td>
<td>Location of private key</td>
</tr>
<tr>
<td><strong>ssl_key_password = p4ssw0rd</strong></td>
<td>Password of private key, if it is password-protected. Since /etc/dovecot.conf is usually world-readable, it is better to place this setting into a root-owned 0600 file instead and include it via the setting /!include_try /etc/dovecot/dovecot-passwd.conf. Alternatively, Dovecot can be started with dovecot -p p4ssw0rd</td>
</tr>
<tr>
<td><strong>ssl_ca_file = /etc/dovecot/cafile.pem</strong></td>
<td>List of trusted SSL certificate authorities. This file contains CA certificates followed by CRLs</td>
</tr>
<tr>
<td><strong>ssl_verify_client_cert = yes</strong></td>
<td>Request client to send a certificate</td>
</tr>
<tr>
<td><strong>ssl_cipher_list = ALL:!LOW:!SSLv2</strong></td>
<td>List of SSL ciphers to use</td>
</tr>
<tr>
<td><strong>verbose_ssl = yes</strong></td>
<td>Show protocol level SSL errors</td>
</tr>
</tbody>
</table>
### Dovecot authentication

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

#### auth default

- **mechanisms = plain login cram-md5**
- **passdb passwd-file {**
  - **args = /etc/dovecot.deny**
  - **deny = yes**
- **}**
- **passdb pam {**
  - **args = cache_key=%u%r dovecot**
- **}**
- **passdb passwd {**
  - **blocking = yes**
  - **args =**
- **}**
- **passdb shadow {**
  - **blocking = yes**
  - **args =**
- **}**
- **passdb bsdauth {**
  - **cache_key = %u**
  - **args =**
- **}**
- **passdb sql {**
  - **args = /etc/dovecot/dovecot-sql.conf**
- **}**
- **passdb ldap {**
  - **args = /etc/dovecot/dovecot-ldap.conf**
- **}**

#### socket listen

- **master {**
  - **path = /var/run/dovecot/auth-master**
  - **mode = 0600**
  - **user = dovecot**
  - **group = dovecot**
- **}**
- **client {**
  - **path = /var/run/dovecot/auth-client**
  - **mode = 0660**
- **}**
FTP (File Transfer Protocol) is a client-server unencrypted protocol for file transfer. Secure alternatives are FTPS (FTP secured with SSL/TLS) and SFTP (SSH File Transfer Protocol). It can operate either in active or in passive mode.

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

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

### FTP servers

<table>
<thead>
<tr>
<th>Very Secure FTP</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pure-FTP</strong></td>
<td>Free and easy-to-use FTP server</td>
</tr>
<tr>
<td></td>
<td><strong>pure-ftpd</strong>                   Pure-FTP daemon</td>
</tr>
<tr>
<td></td>
<td><strong>pure-ftpwho</strong>                 Show clients connected to the Pure-FTP server</td>
</tr>
<tr>
<td></td>
<td><strong>pure-mrtginfo</strong>               Show connections to the Pure-FTP server as a MRTG graph</td>
</tr>
<tr>
<td></td>
<td><strong>pure-statsdecode</strong>            Show Pure-FTP log data</td>
</tr>
<tr>
<td></td>
<td><strong>pure-pw</strong>                    Manage Pure-FTP virtual accounts</td>
</tr>
<tr>
<td></td>
<td><strong>pure-pwconvert</strong>             Convert the system user database to a Pure-FTP virtual accounts database</td>
</tr>
<tr>
<td></td>
<td><strong>pure-quotacheck</strong>            Manage Pure-FTP quota database</td>
</tr>
<tr>
<td></td>
<td><strong>pure-uploadscript</strong>          Run a command on the Pure-FTP server to process an uploaded file</td>
</tr>
</tbody>
</table>

### FTP clients

<table>
<thead>
<tr>
<th>FTP</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ftp</strong></td>
<td>Standard FTP client</td>
</tr>
<tr>
<td></td>
<td><code>ftp ftpserver.domain.com</code> Connect to an FTP server</td>
</tr>
<tr>
<td><strong>lftp</strong></td>
<td>Sophisticated FTP client with support for HTTP and BitTorrent</td>
</tr>
<tr>
<td></td>
<td><code>lftp ftpserver.domain.com</code> Connect to an FTP server and try an anonymous login</td>
</tr>
</tbody>
</table>
The `/etc/vsftpd/vsftpd.conf` file contains settings for the `vsftpd` FTP server. Here are some key settings and what they mean:

- **listen=NO**: Run `vsftpd` in standalone mode (i.e., not via `inetd`).
- **local_enable=YES**: Allow local system users (i.e., in `/etc/passwd`) to log in.
- **chroot_local_user=YES**: Chroot local users in their home directory.
- **write_enable=YES**: Allow FTP commands that write on the filesystem (i.e., `STOR`, `DELE`, `RNFR`, `RNTO`, `MKD`, `RMD`, `APPE`, and `SITE`).
- **anonymous_enable=YES**: Allow anonymous logins? If yes, `anonymous` and `ftp` are accepted as logins.
- **anon_root=/var/ftp/pub**: Directory to go after anonymous login.
- **anon_upload_enable=YES**: Allow anonymous uploads.
- **chown_uploads=YES**: Change ownership of anonymously uploaded files.
- **chown_username=ftp**: User to whom set ownership of anonymously uploaded files.
- **anon_world_readable_only=NO**: Allow anonymous users to only download world-readable files.
- **ssl_enable=YES**: Enable SSL.
- **force_local_data_ssl=NO**: Encrypt local data.
- **force_local_logins_ssl=YES**: Force encrypted authentication.
- **allow_anon_ssl=YES**: Allow anonymous users to use SSL.
- **ssl_tlsv1=YES**, **ssl_tlsv2=NO**, **ssl_tlsv3=NO**: Allowed SSL/TLS versions.
- **rsa_cert_file=/etc/pki/tls/certs/vsftpd.pem**: Location of certificate file.
- **rsa_private_key_file=/etc/pki/tls/certs/vsftpd.pem**: Location of private key file.
In Linux, printers are managed by `cupsd`, the CUPS (Common Unix Printing System) daemon. Printers are administered via a web interface on the URL http://localhost:631.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/etc/cups/cupsd.conf</td>
<td>CUPS configuration file</td>
</tr>
<tr>
<td>/etc/cups/printers.conf</td>
<td>Database of available local CUPS printers</td>
</tr>
<tr>
<td>/etc/printcap</td>
<td>Database of printer capabilities, for old printing applications</td>
</tr>
<tr>
<td>/var/spool/cups/</td>
<td>Printer spooler for data awaiting to be printed</td>
</tr>
<tr>
<td>/var/log/cups/error_log</td>
<td>CUPS error log</td>
</tr>
<tr>
<td>/var/log/cups/page_log</td>
<td>Information about printed pages</td>
</tr>
<tr>
<td>/etc/init.d/cupsys start</td>
<td>Start the CUPS service</td>
</tr>
</tbody>
</table>

To manage printers:

- **Run the CUPS Manager graphical application**: `gnome-cups-manager`
- **Enable a CUPS printer**: `cupsenable printer0`
- **Disable a CUPS printer**: `cupsdisable printer0`
- **Accept a job sent on a printer queue**: `cupsaccept printer0`
- **Reject a job sent on a printer queue, with an informational message**: `cupsreject -r "Message" printer0`
- **Test the conformance of a PPD file to the format specification**: `cupstestppd LEXC510.ppd`
- **Export a printer to Samba (for use with Windows clients)**: `cupsaddsmb printer0`

For compiling and linking:

- **Show the necessary compiler options**: `cups-config --cflags`
- **Show the default CUPS data directory**: `cups-config --datadir`
- **Show the necessary linker options**: `cups-config --ldflags`
- **Show the necessary libraries to link to**: `cups-config --libs`
- **Show the default CUPS binaries directory that stores filters and backends**: `cups-config --serverbin`
- **Show the default CUPS configuration file directory**: `cups-config --serverroot`

For administering or viewing print queues:

- **Show CUPS status information**: `lpsstat`
- **Administer CUPS printers**: `lpadmin`
- **Specify a PPD (Adobe PostScript Printer Description) file to associate to a printer**: `lpadmin -p printer0 -P LEXC750.ppd`
- **Print a file on the specified printer**: `lp -d printer0 file`
- **View the default print queue**: `lpq`
- **View a specific print queue**: `lpq -P printer0`
- **View the print queue of a specific user**: `lpq user`
- **Delete a specific job from a printer queue**: `lprm -P printer0 jobnumber`
- **Delete all jobs from a specific user from a printer queue**: `lprm -P printer0 user`
- **Delete all jobs from a printer queue**: `lprm -P printer0 -`
- **Manage print queues**: `lp`

For converting files:

- **Convert a text file to PostScript**: `a2ps file.txt`
- **Convert a file from PostScript to PDF**: `ps2pdf file.ps`
- **Print a PostScript document on multiple pages per sheet on a PostScript printer**: `mpage file.ps`
- **View a PostScript document (the gv software is a derivation of GhostView)**: `gv file.ps`
An IPv4 address is 32-bit long, and is represented divided in four octets (dotted-quad notation), e.g. 193.22.33.44.

There are approximately $4 \times 10^{16}$ total possible IPv4 addresses.

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

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

There are approximately $3 \times 10^{38}$ total possible IPv6 addresses.

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

<table>
<thead>
<tr>
<th>Region</th>
<th>RIR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Africa</td>
<td>AFRINIC</td>
</tr>
<tr>
<td>US, Canada, Antarctica</td>
<td>ARIN</td>
</tr>
<tr>
<td>Asia and Oceania</td>
<td>APNIC</td>
</tr>
<tr>
<td>Latin America</td>
<td>LACNIC</td>
</tr>
<tr>
<td>Europe, Middle East, and Russia</td>
<td>RIPE NCC</td>
</tr>
</tbody>
</table>

---

**IPv4 addressing**

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

---

**IPv6 addressing**

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

---

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>254 hosts each</td>
<td>254 hosts each</td>
<td>1 subnet</td>
<td>2 subnets</td>
<td>8 subnets</td>
<td>16 subnets</td>
<td>64 subnets</td>
</tr>
<tr>
<td>254 total hosts</td>
<td>254 total hosts</td>
<td>2 subnets</td>
<td>4 subnets</td>
<td>8 subnets</td>
<td>16 subnets</td>
<td>64 subnets</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>.0</td>
<td>.0</td>
<td>.16</td>
<td>.32</td>
<td>.48</td>
<td>.64</td>
<td>.96</td>
</tr>
<tr>
<td>.0</td>
<td>.0</td>
<td>.16</td>
<td>.32</td>
<td>.48</td>
<td>.64</td>
<td>.96</td>
</tr>
<tr>
<td>.0</td>
<td>.0</td>
<td>.16</td>
<td>.32</td>
<td>.48</td>
<td>.64</td>
<td>.96</td>
</tr>
<tr>
<td>.0</td>
<td>.0</td>
<td>.16</td>
<td>.32</td>
<td>.48</td>
<td>.64</td>
<td>.96</td>
</tr>
<tr>
<td>.0</td>
<td>.0</td>
<td>.16</td>
<td>.32</td>
<td>.48</td>
<td>.64</td>
<td>.96</td>
</tr>
<tr>
<td>.0</td>
<td>.0</td>
<td>.16</td>
<td>.32</td>
<td>.48</td>
<td>.64</td>
<td>.96</td>
</tr>
<tr>
<td>.0</td>
<td>.0</td>
<td>.16</td>
<td>.32</td>
<td>.48</td>
<td>.64</td>
<td>.96</td>
</tr>
<tr>
<td>.0</td>
<td>.0</td>
<td>.16</td>
<td>.32</td>
<td>.48</td>
<td>.64</td>
<td>.96</td>
</tr>
</tbody>
</table>

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

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

1-1023: privileged ports, used server-side

1024-65535: unprivileged ports, used client-side

/`etc/services` lists all well-known ports.

Many network services are run by the `xinetd` super server.

### ISO/OSI and TCP/IP protocol stack models

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

```
ip a
ip addr
ip addr show
ifconfig -a

ip link show eth0
ifconfig eth0

ip addr add dev eth0 10.1.1.3/24
ifconfig eth0 10.1.1.3 netmask 255.255.255.0 broadcast 10.1.1.255
ifconfig eth0 hw ether 45:67:89:ab:cd:ef

ip link set eth0 up
ifconfig eth0 up
ifup eth0

ip link set eth0 down
ifconfig eth0 down
ifdown eth0

dhcpcd eth0
dhcpcd eth0 (SUSE)

ip neigh
arp -a

ip neigh show 10.1.1.4
arp 10.1.1.4

ip neigh add 10.1.1.5 lladdr 01:23:45:67:89:ab dev eth0
arp -s 10.1.1.5 01:23:45:67:89:ab

ip neigh del 10.1.1.5 dev eth0
arp -d 10.1.1.5

ip neigh flush all

hostname
hostname -f
hostname mybox
hostnamectl set-hostname --static "mybox" (Red Hat)
hostnamectl (Red Hat)

/etc/init.d/networking restart (Debian)
/etc/init.d/network restart (Red Hat)

ethtool option device
ethtool eth0
```

Display configuration of all network interfaces

Display configuration of eth0

Configure IP address of eth0

Configure MAC address of eth0

Activate eth0

Shut down eth0

Request an IP address via DHCP

Show the ARP cache table (containing mappings of MAC to IP addresses)

Show the ARP cache entry for a host

Add a new ARP entry for a host

Delete an ARP entry

Delete the ARP table for all interfaces

Get the hostname

Get the FQDN (Fully Qualified Domain Name)

Get the hostname, OS, and other information

Restart network services

Query or control network driver and hardware settings

View hardware settings of eth0
Network configuration files

/etc/hosts

Mappings between IP addresses and hostnames, for name resolution

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

/etc/nsswitch.conf

Sources that must be used by various system library lookup functions

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

/etc/host.conf

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

order hosts,bind
multi on

/etc/resolv.conf

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

search domain1.org domain2.org
nameserver  192.168.3.3
nameserver  192.168.4.4

/etc/networks

Mappings between network addresses and names

loopback  127.0.0.0
mylan     10.2.3.0

/etc/services

List of service TCP/UDP port numbers

/etc/protocols

List of available protocols

/sys/class/net

List of all network interfaces in the system
### Red Hat network configuration

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

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

- **Multiple configuration files for a single eth0 interface, which allows binding multiple IP addresses to a single NIC**
  - `/etc/sysconfig/network-scripts/ifcfg-eth0:0`
  - `/etc/sysconfig/network-scripts/ifcfg-eth0:1`
  - `/etc/sysconfig/network-scripts/ifcfg-eth0:2`

- **Static route configuration for eth0**
  - `/etc/sysconfig/network-scripts/route-eth0`
    - `default 10.2.3.4 dev eth0`
    - `10.7.8.0/24 via 10.2.3.254 dev eth0`
    - `10.7.9.0/24 via 10.2.3.254 dev eth0`

- **Ethernet frame types. Lists various Ethernet protocol types used on Ethernet networks**
  - `/etc/ethertypes`

### Debian network configuration

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

- **Hostname of the local machine**
  - `/etc/hostname`

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

```
nmcli device status
```

Show all network devices

```
nmcli device disconnect iface
```

Disconnects the device `iface`.

This command should be used instead of `nmcli connection down connection` because if **connection** is set to autoconnect, Network Manager will bring the connection (and the device) up again short time later.

```
nmcli connection show
```

Show all connections.

```
nmcli connection show --active
```

Connections with an empty device entry are inactive

```
nmcli connection show connection
```

Show active connections

```
nmcli connection add con-name connection \ type ethernet ifname iface ipv4.method manual \ ipv4.addresses 10.0.0.13/24 ipv4.gateway 10.0.0.254
```

Configure a new **connection** that uses the Ethernet interface `iface` and assigns it an IPv4 address and gateway

```
nmcli connection modify connection options
```

Modify the configuration of **connection**

```
nmcli connection up connection
```

Brings up a **connection**

```
nmcli connection reload
```

Reload any manual change made to the files `/etc/sysconfig/network-scripts/ifcfg-*`

The **manpage** `man nmcli-examples` contains examples of network configuration.
Teaming and bridging

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

How to set up a teaming connection

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

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

3. `nmcli connection add type team-slave ifname iface \ master teamcon`
   
   Add an existing device `iface` as a slave of team `teamcon`. The slave connection will be automatically named `team-slave-iface`.

4. Repeat the previous step for each slave interface.

```
teamdctl teamif state
```

Show the state of the team interface `teamif`

```
teamnl teamif command
```

Debug a team interface `teamif`

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

How to set up a bridge connection

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

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

3. `nmcli connection add type bridge-slave ifname iface \ master brcon`
   
   Add an existing device `iface` as a slave of bridge `brcon`. The slave connection will be automatically named `bridge-slave-iface`.

4. Repeat the previous step for each slave interface.

```
brctl show brif
```

Display information about the bridge interface `brif`

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

iwlist wlan0 scan
List all wireless devices in range, with their quality of signal and other information

iwlist wlan0 freq
Display transmission frequency settings

iwlist wlan0 rate
Display transmission speed settings

iwlist wlan0 txpower
Display transmission power settings

iwlist wlan0 key
Display encryption settings

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

iwconfig wlan0
Display configuration of wireless interface wlan0

iwconfig wlan0 option
Configure wireless interface wlan0

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

rfkill list
List installed wireless devices

rfkill unblock n
Enable wireless device number n

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

**dig example.org**
Perform a DNS lookup for the specified domain or hostname. Returns information in BIND zone file syntax; uses an internal resolver and hence does not honor `/etc/resolv.conf`

**host example.org**
Perform a DNS lookup for the specified domain or hostname. Does honor `/etc/resolv.conf`

**nslookup example.org** (deprecated)
Perform a DNS lookup for the specified domain or hostname. Does honor `/etc/resolv.conf`

**dig @nameserver -t MX example.org**
Perform a DNS lookup for the MX record of the specified domain, querying nameserver

**host -a example.org**
Get all DNS records for a domain

**dig -x a.b.c.d**
Perform a reverse DNS lookup for the IP address a.b.c.d

**whois example.org**
Query the WHOIS service for an Internet resource (usually a domain name)

**ping host**
Test if a remote host can be reached and measure the round-trip time to it. This is done by sending an ICMP Echo Request datagram and awaiting an ICMP Echo Response

**fping -a host1 host2 host3**
Ping multiple hosts in parallel and report which ones are alive

**bing host1 host2**
Calculate point-to-point throughput between two hosts

**traceroute host**
Print the route, hop by hop, packets trace to a remote host. This is done by sending a sequence of ICMP Echo Request datagrams with increasing TTL values, starting with TTL=1, and expecting ICMP Time Exceeded datagrams

**tracepath host**
Simpler traceroute

**mtr host**
traceroute and ping combined

**redir --laddr=ip1 --lport=port1 \ --raddr=ip2 --rport=port2**
Redirect all connections coming to local IP address ip1 and port port1, to remote IP address ip2 and port port2

**telnet host port**
Establish a telnet connection to the specified host and port number. If port is omitted, uses default port 23

**wget --no-clobber --html-extension \ --page-requisites --convert-links \ --recursive --domains example.org \ --no-parent www.example.org/path**
Download a whole website www.example.org/path

**curl www.example.org/file.html -o myfile.html**
Download a file via HTTP and save it locally under another name

**curl -u user:password 'ftp://ftpserver/path/file'**
Download a file via FTP, after logging in to the server

**curl -XPUT webserver -d' data'**
Send an HTTP PUT command with data to webserver

**hping3 options host**
Send a custom TCP/IP packet to host and display the reply
Network monitoring

`netstat` Display network connections
`netstat --tcp` Display active TCP connections
`netstat -t` Display only listening sockets
`netstat -l` Display only listening sockets
`netstat -a` Display all listening and non-listening sockets
`netstat -n` Display network connections, without resolving hostnames or portnames
`netstat -p` Display network connections, with PID and name of program to which each socket belongs
`netstat -i` Display network interfaces
`netstat -s` Display protocol statistics
`netstat -r` Display kernel routing tables (equivalent to `route -e`)
`netstat -c` Display network connections continuously

`ss` Display socket statistics (similarly to `netstat`)
`ss -t -a` Display all TCP sockets

`nmap host` Scan for open TCP ports (TCP SYN scan) on remote host
`nmap -sS host` Do a ping sweep (ICMP ECHO probes) on remote host
`nmap -SP host` Scan for open UDP ports on remote host
`nmap -sV host` Do a service and version scan on open ports
`nmap -p 1-65535 host` Scan all ports (1-65535), not only the common ports, on remote host
`nmap -O host` Find which operating system is running on remote host (OS fingerprinting)

`arp-scan` Scan all hosts on the current LAN. Uses ARP (Layer 2) packets and is therefore able to find even the hosts configured to drop all IP or ICMP traffic; for the same reason it cannot scan hosts outside the same LAN

`ngrep` Filter data payload of network packets matching a specified regex

`dhcpdump -i eth0` Sniff all DHCP packets on interface `eth0`

`nload` Display a graph of the current network usage

`iptraf` IP LAN monitor (Ncurses UI)
`iptraf-ng` IP LAN monitor (Ncurses UI)

`netserver` Run a network performance benchmark server
`netperf` Do network performance benchmarks by connecting to a netserver

`iperf -s` Run a network throughput benchmark server
`iperf -c server` Perform network throughput tests in client mode, by connecting to an iperf server
Tcpdump is a packet analyzer (aka packet sniffer). A GUI equivalent is Wireshark, previously called Ethereal.

```
tcpdump -ni eth0
Sniff all network traffic on interface eth0, suppressing DNS resolution

tcpdump ip host 10.0.0.2 tcp port 25
Sniff network packets on TCP port 25 from and to 10.0.0.2

tcpdump ether host '45:67:89:ab:cd:ef'
Sniff traffic from and to the network interface having MAC address 45:67:89:ab:cd:ef

tcpdump 'src host 10.0.0.2 and (tcp port 80 or tcp port 443)'
Sniff HTTP and HTTPS traffic having as source host 10.0.0.2

tcpdump -ni eth0 not port 22
Sniff all traffic on eth0 except that belonging to the SSH connection

tcpdump -vvvn -i eth0 arp
Sniff ARP traffic on eth0, on maximum verbosity level, without converting host IP addresses and port numbers to names

tcpdump ip host 10.0.0.2 and not 10.0.0.9
Sniff IP traffic between 10.0.0.2 and any other host except 10.0.0.9
```
Netcat is "the Swiss Army knife of networking", a very flexible generic TCP/IP client/server. Depending on the distribution, the binary is called `nc`, `ncat` (Red Hat), or `netcat` (SUSE).

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

Scan for a listening SSH daemon on remote host 10.0.0.7

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

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

```
nc 10.0.0.7 389 < file
```

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

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

Connect to web server 10.0.0.7 and issue a HTTP GET

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

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

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

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

```
echo "\n" | 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
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.

<table>
<thead>
<tr>
<th>/etc/hosts.allow and /etc/hosts.deny lines syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL: ALL</td>
<td>All services to all hosts</td>
</tr>
<tr>
<td>ALL: .example.edu</td>
<td>All services to all hosts of the example.edu domain</td>
</tr>
<tr>
<td>ALL: .example.edu EXCEPT host1.example.edu</td>
<td>All services to all hosts of example.edu, except host1</td>
</tr>
<tr>
<td>in.fingerd: .example.com</td>
<td>Finger service to all hosts of example.com</td>
</tr>
<tr>
<td>in.tftpd: LOCAL</td>
<td>TFTP to hosts of the local domain only</td>
</tr>
<tr>
<td>sshd: 10.0.0.3 10.0.0.4 10.1.1.0/24</td>
<td>SSH to the hosts and network specified</td>
</tr>
<tr>
<td>sshd: 10.0.1.0/24</td>
<td>SSH to 10.0.1.0/24</td>
</tr>
<tr>
<td>sshd: 10.0.1.0/255.255.255.0</td>
<td>Send a finger probe to hosts attempting TFTP and notify root user via email</td>
</tr>
<tr>
<td>in.tftpd: ALL: spawn (/safe_dir/safe_finger \</td>
<td>When a client attempts a RPC request via the portmapper (NFS access), echo a message to the terminal and notify the root user via email</td>
</tr>
<tr>
<td>-l @%h</td>
<td>/bin/mail -s %d-%h root) &amp;</td>
</tr>
<tr>
<td>portmap: ALL: (echo Illegal RPC request \</td>
<td></td>
</tr>
<tr>
<td>from %h</td>
<td>/bin/mail root) &amp;</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Output of command `route -en`

<table>
<thead>
<tr>
<th>Kernel IP routing table</th>
<th>Destination</th>
<th>Gateway</th>
<th>Genmask</th>
<th>Flags</th>
<th>Metric</th>
<th>Ref</th>
<th>Use</th>
<th>Iface</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>192.168.3.0</td>
<td>0.0.0.0</td>
<td>255.255.255.0</td>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>eth0</td>
</tr>
<tr>
<td></td>
<td>0.0.0.0</td>
<td>192.168.3.1</td>
<td>0.0.0.0</td>
<td>UG</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>eth0</td>
</tr>
</tbody>
</table>

### Destination
- **network or host**: destination network or host
- **0.0.0.0**: default route

### Gateway
- **host**: gateway
- **0.0.0.0**: no gateway needed, network is directly connected
- *****: rejected route

### Genmask
- **network mask**: network mask to apply for the destination network
- **255.255.255.255**: destination host
- **0.0.0.0**: default route

### Flags
- **U**: route is up
- **G**: use gateway
- **H**: target is host
- **!**: rejected route
- **D**: dynamically installed by daemon
- **M**: modified from routing daemon
- **R**: reinstate route for dynamic routing

---

**ip route**
Display IP routing table

**route -en**

**route -F**

**netstat -rn**

**ip route show cache**
Display kernel routing cache

**route -C**

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

**route add default gw 10.1.1.254**

**ip route add 10.2.0.1 dev eth0**
Add a route for a host 10.2.0.1

**ip route add 10.2.0.1 via 10.2.0.254**

**route add -host 10.2.0.1 gw 10.2.0.254**

**ip route add 10.2.0.0/16 via 10.2.0.254**
Add a route for a network 10.2.0.0/16

**route add -net 10.2.0.0 netmask 255.255.0.0 gw 10.2.0.254**

**ip route delete 10.2.0.1 dev eth0**
Delete a route for a host 10.2.0.1

**route del -host 10.2.0.1 gw 10.2.0.254**

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

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

In Ubuntu, the firewall is managed by the `ufw` (Uncomplicated Firewall) service, with `iptables` as backend.

/proc/sys/log/hostname

`/etc/sysconfig/iptables`: Default file containing the firewall rules

`iptables-restore < file`: Load into `iptables` the firewall rules specified in the `file`

`iptables-save > file`: Save into `iptables` the firewall rules specified in the `file`

<table>
<thead>
<tr>
<th>iptables rules file</th>
<th>Delete all rules and open the firewall to all connections</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>*filter</code></td>
<td></td>
</tr>
<tr>
<td>:INPUT ACCEPT [0:0]</td>
<td></td>
</tr>
<tr>
<td>:FORWARD ACCEPT [0:0]</td>
<td></td>
</tr>
<tr>
<td>:OUTPUT ACCEPT [0:0]</td>
<td></td>
</tr>
<tr>
<td>COMMIT</td>
<td></td>
</tr>
</tbody>
</table>
Iptables uses **tables** containing sets of **chains**, which contain sets of **rules**. Each rule has a **target** (e.g. ACCEPT). The "filter" table contains chains INPUT, FORWARD, OUTPUT (built-in chains); this is the default table to which all iptables commands are applied, unless another table is specified via the `-t` option.

The "nat" table contains chains PREROUTING, OUTPUT, POSTROUTING.

The "mangle" table contains chains PREROUTING, OUTPUT.

When a packet enters the system, it is handed to the INPUT chain. If the destination is local, it is processed; if the destination is not local and IP forwarding is enabled, the packet is handed to the FORWARD chain, otherwise it is dropped.

An outgoing packet generated by the system will go through the OUTPUT chain.

If NAT is in use, an incoming packet will pass at first through the PREROUTING chain, and an outgoing packet will pass last through the POSTROUTING chain.

```
iptables -A INPUT -s 10.0.0.6 -j ACCEPT
Add a rule to accept all packets from 10.0.0.6

iptables -A INPUT -s 10.0.0.7 -j REJECT
Add a rule to reject all packets from 10.0.0.7 and send back a ICMP response to the sender

iptables -A INPUT -s 10.0.0.8 -j DROP
Add a rule to silently drop all packets from 10.0.0.8

iptables -A INPUT -s 10.0.0.9 -j LOG
Add a rule to log (via syslog) all packets from 10.0.0.9

iptables -D INPUT -s 10.0.0.9 -j LOG
Delete a specific rule

iptables -D INPUT 42
Delete rule 42 of the INPUT chain

iptables -F INPUT
Flush all rules of the INPUT chain

iptables -F
Flush all rules, hence disabling the firewall

iptables -t mangle -F
Flush all rules of the "mangle" table

iptables -t mangle -X
Delete all user-defined (not built-in) rules in the "mangle" table

iptables -L INPUT
List the rules of the INPUT chain

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

iptables -N mychain
Define a new chain

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

iptables -A OUTPUT -d 10.7.7.0/24 -j DROP
Add a rule to drop all packets with destination 10.7.7.0/24

iptables -A FORWARD -i eth0 -o eth1 -j DROP
Add a rule to drop all incoming UDP traffic (protocol numbers are defined in /etc/protocols)

iptables -A FORWARD -i eth0 -o eth1 -j LOG
Add a rule to drop all packets entering the system via eth0 and exiting via eth1

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

iptables -A INPUT -p udp -j DROP
Add a rule to drop all packets coming from any unprivileged port and with destination port 53

iptables -A INPUT --sport 1024:65535 --dport 53 \ 
--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 --limit 1/s -i eth0 -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 --state ESTABLISHED \ 
--state RELATED \ 
--state INVALID -j ACCEPT
Add a rule to accept all packets that are not part of a communication already tracked by the state module

iptables -A INPUT -m state --state RELATED -j ACCEPT
Add a rule to accept all packets that are related (e.g. ICMP responses to TCP or UDP traffic) to a communication already tracked by the state module

iptables -A INPUT -m state --state INVALID -j ACCEPT
Add a rule to accept all packets that do not match any of the states above
```
SNAT (Source Network Address Translation)

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

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

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

DNAT (Destination Network Address Translation)

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

PAT (Port Address Translation)

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

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

sysctl -w net.ipv4.ip_forward=1
echo 1 > /proc/sys/net/ipv4/ip_forward
Enable IP forwarding; necessary to set up a Linux machine as a router. (This command causes other network options to be changed as well.)
In firewalld, a network interface (aka interface) or a subnet address (aka source) can be assigned to a specific zone. To determine to which zone a packet belongs, first the zone of the source is analyzed, then the zone of the interface; if no source or interface matches, the packet is associated to the default zone (which is "public", unless set otherwise). If the zone is not specified (via --zone=zone), the command is applied to the default zone. By default, commands are temporary; adding the --permanent option to a command sets it as permanent, or shows permanent settings only. Temporary commands are effective immediately but are canceled at reboot, firewall reload, or firewall restart. Permanent commands are effective only after reboot, firewall reload, or firewall restart.

<table>
<thead>
<tr>
<th>Firewalld zones (as obtained by firewall-cmd --get-zones)</th>
</tr>
</thead>
<tbody>
<tr>
<td>block</td>
</tr>
<tr>
<td>dmz</td>
</tr>
<tr>
<td>drop</td>
</tr>
<tr>
<td>external</td>
</tr>
<tr>
<td>home</td>
</tr>
<tr>
<td>internal</td>
</tr>
<tr>
<td>public</td>
</tr>
<tr>
<td>trusted</td>
</tr>
<tr>
<td>work</td>
</tr>
</tbody>
</table>

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

firewall-config
Firewall management GUI

firewall-cmd --reload
Reload firewall configuration; this applies all permanent changes and cancels all temporary changes. Current connections are not terminated

firewall-cmd --complete-reload
Reload firewall configuration, stopping all current connections

firewall-cmd --runtime-to-permanent
Transform all temporary changes to permanent

firewall-cmd --list-all-zones
List all zones and their full settings

firewall-cmd --get-default-zone
Show the default zone

firewall-cmd --set-default-zone=home
Set "home" as the default zone

firewall-cmd --get-active-zones
Show the active zones i.e. zones bound to either an interface or a source

firewall-cmd --get-zones
Show all available zones

firewall-cmd --get-zone-of-interface=eth0
Show the zone assigned to eth0

firewall-cmd --new-zone=test
Create a new zone called "test"

firewall-cmd --zone=home --change-interface=eth0
Assign eth0 to the "home" zone

firewall-cmd --zone=home --list-all
List temporary settings of the "home" zone

firewall-cmd --zone=home --list-all --permanent
List permanent settings of the "home" zone

firewall-cmd --zone=home --add-source=10.1.1.0/24
Assign 10.1.1.0/24 to the "home" zone i.e. route all traffic from that subnet to that zone

firewall-cmd --zone=home --list-sources
List sources bound to the "home" zone
**firewalld rules**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>firewall-cmd --zone=trusted --add-service=ssh</code></td>
<td>Add the SSH service to the &quot;trusted&quot; zone</td>
</tr>
<tr>
<td><code>firewall-cmd --zone=trusted --add-port=22/tcp</code></td>
<td>Add the SSH, HTTP, and HTTPS services to the &quot;trusted&quot; zone</td>
</tr>
<tr>
<td><code>firewall-cmd --zone=trusted --add-service={ssh,http,https}</code></td>
<td>Show temporary and permanent services bound to the &quot;trusted&quot; zone</td>
</tr>
<tr>
<td><code>firewall-cmd --zone=trusted --list-services</code></td>
<td>Show temporary and permanent ports open on the &quot;trusted&quot; zone</td>
</tr>
<tr>
<td><code>firewall-cmd --get-services</code></td>
<td>List all predefined services</td>
</tr>
<tr>
<td><code>Predefined services are configured in /usr/lib/firewalld/services/service.xml.</code></td>
<td></td>
</tr>
<tr>
<td>User-defined services are configured in /etc/firewalld/services/service.xml.</td>
<td></td>
</tr>
<tr>
<td><code>firewall-cmd --get-icmptypes</code></td>
<td>Show all known types of ICMP messages</td>
</tr>
<tr>
<td><code>firewall-cmd --add-icmp-block=echo-reply</code></td>
<td>Block a specific ICMP message type</td>
</tr>
<tr>
<td><code>firewall-cmd --query-icmp-block=echo-reply</code></td>
<td>Tell if a specific ICMP message type is blocked</td>
</tr>
<tr>
<td><code>firewall-cmd --list-icmp-block</code></td>
<td>Show the list of blocked ICMP message types</td>
</tr>
<tr>
<td><code>firewall-cmd --add-rich-rule='richrule'</code></td>
<td>Set up a rich rule (for more complex and detailed firewall configurations)</td>
</tr>
<tr>
<td><code>firewall-cmd --add-rich-rule='rule family=ipv4 source address=10.2.2.0/24 service name=tftp log prefix=tftp level=info limit value=3/m accept'</code></td>
<td>Set up a rich rule to allow tftp connections from subnet 10.2.2.0/24 and log them via syslog at a rate of 3 per minute</td>
</tr>
<tr>
<td><code>firewall-cmd --list-rich-rules</code></td>
<td>List all rich rules</td>
</tr>
<tr>
<td>The manpage <code>man firewalld.richlanguage</code> contains several examples of rich rules.</td>
<td></td>
</tr>
<tr>
<td><code>firewall-cmd --direct --add-rule directrule</code></td>
<td>Set up a direct rule (in iptables format)</td>
</tr>
<tr>
<td><code>firewall-cmd --direct --add-rule --ipv4 filter INPUT 0 -p tcp --dport 22 -j ACCEPT</code></td>
<td>Set up a direct rule to allow SSH connections</td>
</tr>
<tr>
<td><code>firewall-offline-cmd directrule</code></td>
<td>Set up a direct rule when firewalld is not running</td>
</tr>
<tr>
<td><code>firewall-cmd --direct --get-all-rules</code></td>
<td>Show all direct rules</td>
</tr>
<tr>
<td>The manpage <code>man firewalld.direct</code> documents the syntax of direct rules. User-defined direct rules are stored in /etc/firewalld/direct.xml.</td>
<td></td>
</tr>
<tr>
<td><code>firewall-cmd --zone=zone --add-masquerade</code></td>
<td>Set up masquerading for hosts of zone; packets originating from zone will get the firewall's IP address on the &quot;external&quot; zone as source address</td>
</tr>
<tr>
<td><code>firewall-cmd --zone=zone --add-rich-rule='rule family=ipv4 source address=10.2.2.0/24 masquerade'</code></td>
<td>Set up masquerading only for those hosts of zone located in subnet 10.2.2.0/24</td>
</tr>
<tr>
<td><code>firewall-cmd --zone=zone --add-forward-port=port=22:proto=tcp:toport=2222:toaddr=10.7.7.7</code></td>
<td>Set up port forwarding for hosts of zone; incoming connections to port 22 for hosts of zone will be forwarded to port 2222 on host 10.7.7.7</td>
</tr>
</tbody>
</table>
Secure Shell (SSH) is a protocol (not a shell) for encrypted secure communications. It is mostly used as a replacement to Telnet to securely login to a remote server’s terminal, but can be applied to any network protocol. Some of the most common applications of SSH are Secure Copy (SCP) and SSH File Transfer Protocol (SFTP).

```
ssh user@host
```

Connect to a remote host via SSH and login as user.

Options:
- `-v` `-vv` `-vvv` Increasing levels of verbosity
- `-p n` Use port `n` instead of standard port 22

```
ssh user@host command
```

Execute a command on a remote host.

```
autossh user@host
```

Connect to a remote host, monitoring the connection and restarting it automatically if it dies.

```
sshpas -p password ssh user@host
```

Connect to a remote host using the specified password.

```
pssh -i -H "host1 host2 host3" command
```

Execute a command in parallel on a group of remote hosts.

```
ssh-keygen -t rsa -b 2048
```

Generate interactively a 2048-bit RSA key pair; will prompt for a passphrase.

```
ssh-keygen -t dsa
```

Generate a DSA key pair.

```
ssh-keygen -p -t rsa
```

Change passphrase of the private key.

```
ssh-keygen -q -t rsa -f keyfile -N '' -C ''
```

Generate a RSA key with no passphrase (for non-interactive use) and no comment.

```
ssh-keygen -lf keyfile
```

View key length and fingerprint of a public or private key.

```
< keyfile.pub awk '{print $2}' | base64 -d | openssl hashfunction
```

View fingerprint of a key, calculated using `hashfunction`. RSA keys fingerprint use `sha1` (deprecated) or `md5`.

```
ssh-keyscan host >> ~/.ssh/known_hosts
```

Get the public key of `host` and add it to the user’s known hosts file.

```
ssh-agent
```

Echo to the terminal the environment variables that must be set in order to use the SSH Agent.

```
eval `ssh-agent`
```

Start the SSH Agent daemon that caches decrypted private keys in memory; also shows the PID of ssh-agent and sets the appropriate environment variables. Once `ssh-agent` is started, the keys to cache must be added via the `ssh-add` command; cached keys will then be automatically used by any SSH tool e.g. `ssh`, `sftp`, `scp`.

```
ssh-agent bash -c 'ssh-add keyfile'
```

Start ssh-agent and cache the specified key.

```
ssh-add
```

Add the default private keys to the ssh-agent cache.

```
ssh-add keyfile
```

Add a specific private key to the ssh-agent cache.

```
ssh-copy-id user@host
```

Use locally available keys to authorize, via public key authentication, login of user on a remote host. This is done by copying the user’s local public key 
`~/.ssh/id_rsa.pub` to `~/.ssh/authorized_keys` on the remote host.
SSH tools

scp /path1/file user@host:/path2/
scp user@host:/path1/file /path2/
scp user1@host1:/path1/file user2@host2:/path2/

Non-interactive secure file copy via SSH. Can transfer files from local to remote, from remote to local, or between two remote hosts

sftp user@host

SSH FTP-like tool for secure file transfer

scponly

SSH wrapper pseudo-shell providing access to remote users for secure file transfer, but without execution privileges
**SSH port forwarding (aka SSH tunneling)**

```bash
ssh -L 2525:mail.foo.com:25 user@mail.foo.com
```

Establish a SSH encrypted tunnel from localhost to remote host mail.foo.com, redirecting traffic from local port 2525 to port 25 of remote host mail.foo.com.

Useful if the local firewall blocks outgoing port 25. In this case, port 2525 is used to go out; the application must be configured to connect to localhost on port 2525 (instead of mail.foo.com on port 25).

```bash
ssh -L 2525:mail.foo.com:25 user@login foo.com
```

Establish a SSH encrypted tunnel from localhost to remote host login.foo.com.

Remote host login.foo.com will then forward, unencrypted, all data received over the tunnel on port 2525 to remote host mail.foo.com on port 25.

**SSH reverse forwarding (aka SSH reverse tunneling)**

```bash
ssh -R 2222:localhost:22 user@login foo.com
```

Establish a SSH encrypted reverse tunnel from remote host login.foo.com back to localhost, redirecting traffic sent to port 2222 of remote host login.foo.com back towards local port 22.

Useful if the local firewall blocks incoming connections so remote hosts cannot connect back to local machine. In this case, port 2222 of login.foo.com is opened for listening and connecting back to localhost on port 22; remote host login.foo.com is then able to connect to the local machine on port 2222 (redirected to local port 22).

**SSH as a SOCKS proxy**

```bash
ssh -D 33333 user@login foo.com
```

The application supporting SOCKS must be configured to connect to localhost on port 33333. Data is tunneled from localhost to login.foo.com, then unencrypted to destination.

**X11 Forwarding**

```bash
ssh -X user@login foo.com
```

Enable the local display to execute locally a X application stored on a remote host login.foo.com.

**How to enable public key authentication**

1. On remote host, set `PubkeyAuthentication yes` in `/etc/ssh/sshd_config`
2. On local machine, do `ssh-copy-id you@remotehost` (or copy your public key to the remote host by hand)

**How to enable host-based authentication amongst a group of trusted hosts**

1. On all hosts, set `HostbasedAuthentication yes` in `/etc/ssh/sshd_config`
2. On all hosts, create `/etc/ssh/shosts.equiv` and enter in this file all trusted hostnames
3. Connect via SSH manually from your machine on each host so that all hosts' public keys go into `~/.ssh/known_hosts`
4. Copy `~/.ssh/known_hosts` from your machine to `/etc/ssh/ssh_known_hosts` on all hosts

**How to enable X11 Forwarding**

1. On remote host 10.2.2.2, set `X11Forwarding yes` in `/etc/ssh/sshd_config`, and make sure that `xauth` is installed
2. On local host 10.1.1.1, type `ssh -X 10.2.2.2`, then run on remote host the graphical application e.g. `xclock &`

It is also possible to enable X11 Forwarding via telnet (but this is insecure and obsolete, and therefore not recommended):

1. On remote host 10.2.2.2, type `export DISPLAY=10.1.1.1:0.0`
2. On local host 10.1.1.1, type `xhost +`
3. On local host 10.1.1.1, type `telnet 10.2.2.2`, then run on remote host the graphical application e.g. `xclock &`
### SSH Configuration

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/etc/ssh/sshd_config</code></td>
<td>SSH server daemon configuration file</td>
</tr>
<tr>
<td><code>/etc/ssh/ssh_config</code></td>
<td>SSH client global configuration file</td>
</tr>
<tr>
<td><code>/etc/ssh/sshd_host_key</code></td>
<td>Host's private key (should be mode 0600)</td>
</tr>
<tr>
<td><code>/etc/ssh/sshd_host_key.pub</code></td>
<td>Host's public key</td>
</tr>
<tr>
<td><code>/etc/ssh/ssh_known_hosts</code></td>
<td>Names of trusted hosts for host-based authentication</td>
</tr>
<tr>
<td><code>~/.ssh/</code></td>
<td>User's SSH directory (must be mode 0700)</td>
</tr>
<tr>
<td><code>~/.ssh/config</code></td>
<td>SSH client user configuration file</td>
</tr>
<tr>
<td><code>~/.ssh/id_rsa</code></td>
<td>User's RSA or DSA private key, as generated by <code>ssh-keygen</code></td>
</tr>
<tr>
<td><code>~/.ssh/id_dsa</code></td>
<td>User's RSA or DSA public key, as generated by <code>ssh-keygen</code></td>
</tr>
<tr>
<td><code>~/.ssh/known_hosts</code></td>
<td>Host public keys that were previously accepted as legitimate</td>
</tr>
<tr>
<td><code>~/.ssh/authorized_keys</code></td>
<td>Trusted public keys; the corresponding private keys allow the user to</td>
</tr>
<tr>
<td></td>
<td>authenticate on this host</td>
</tr>
<tr>
<td><code>~/.ssh/authorized_keys2</code></td>
<td>(obsolete)</td>
</tr>
</tbody>
</table>

#### `/etc/ssh/sshd_config` - SSH server configuration file

- **PermitRootLogin**
  - yes: Control superuser login via SSH. Possible values are:
    - yes: Superuser can login
    - no: Superuser cannot login
    - without-password: Superuser cannot login with password
    - forced-commands-only: Superuser can only run commands in SSH command line
  - no: Superuser cannot 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`
- **AllowGroups** `geeks`
- **DenyGroups** `*`
- **PasswordAuthentication** yes
- **PubKeyAuthentication** yes
- **HostbasedAuthentication** yes
- **Protocol 1,2**
- **X11Forwarding** yes

#### `/etc/ssh/config` and `~/.ssh/config` - SSH client configuration file

- **Host** `*`
  - List of hosts to which the following directives will apply, or `*` for all hosts
- **StrictHostKeyChecking** yes
  - Ask before adding new host keys to the `~/.ssh/known_hosts` file, and refuse to connect if the key for a known host has changed. This prevents MITM attacks
- **GSSAPIAuthentication** yes
  - Support authentication using GSSAPI
- **ForwardX11Trusted** yes
  - Allow remote X11 clients to fully access the original X11 display
- **IdentityFile** `~/.ssh/id_rsa`
  - User identity file for authentication. Default values are:
    - `~/.ssh/identity` for protocol version 1
    - `~/.ssh/id_rsa` and `~/.ssh/id_dsa` for protocol version 2
The X.509 standard defines the format of public key certificates and other related files. It includes cryptographic standards and protocols such as SSL/TLS, PKCS7, PKCS12, and OCSP. The Public Key Infrastructure X.509 (PKIX) is described in RFC 5280.

### X.509 file formats

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DER</td>
<td>Binary-encoded certificate</td>
</tr>
</tbody>
</table>
| PEM    | ASCII-armored Base64-encoded certificate, included between these two lines:  

```
-----BEGIN X.509_FILE_TYPE-----
-----END X.509_FILE_TYPE-----
```

DER and PEM are also used as file extensions for different types of files; see below.

### X.509 file type extensions

<table>
<thead>
<tr>
<th>Extension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRT</td>
<td>Certificate or certificate chain</td>
</tr>
<tr>
<td>CER</td>
<td>Certificate Signing Request</td>
</tr>
<tr>
<td>CSR</td>
<td>Certificate Signing Request</td>
</tr>
<tr>
<td>KEY</td>
<td>Private key</td>
</tr>
<tr>
<td>CRL</td>
<td>Certificate Revocation List</td>
</tr>
<tr>
<td>DER</td>
<td>Certificate; DER-encoded</td>
</tr>
<tr>
<td>PEM</td>
<td>Certificate (including or not the private key), certificate chain, or Certificate Signing Request; PEM-encoded</td>
</tr>
</tbody>
</table>

### Other file type extensions

<table>
<thead>
<tr>
<th>Extension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P12</td>
<td>Certificate (including or not the private key), certificate chain, or Certificate Signing Request; bundled in a PKCS#12 archive file format</td>
</tr>
<tr>
<td>PFX</td>
<td>Certificate (including or not the private key), certificate chain, or Certificate Signing Request; bundled in a PKCS#12 archive file format</td>
</tr>
</tbody>
</table>
OPENSSL

openssl x509 -text -in cert.crt -noout
openssl req -text -in cert.csr -noout
openssl req -new -key cert.key -out cert.csr
openssl x509 -req -text -in cert.crt -noout
openssl req -new -keyout cert.key -out cert.csr -newkey rsa:2048 -nodes
openssl x509 -req -in cert.crt -CAcreateserial -CA ca.crt -CAkey ca.key -out cert.crt -days validity
openssl req -x509 -key cert.key -out cert.crt
openssl ca -config ca.conf -in cert.crt -out cert.crt -days validity -verbose
openssl ca -config ca.conf -generate -revoke cert.crt -crl_reason why
openssl ca -config ca.conf -generate -out list.crl

openssl x509 -in cert.pem -outform DER -out cert.der
openssl pkcs12 -export -in cert.pem -inkey cert.key -out cert.pfx -name friendlyname
openssl pkcs12 -in cert.p12 -out cert.crt -clcerts -nokeys
openssl pkcs12 -in cert.p12 -out cert.key -nocerts -nodes
openssl pkcs12 -in cert.p12 -out ca.crt -cacerts
cat cert.crt cert.key > cert.pem

openssl dgst -hashfunction -out file.hash file
openssl dgst -hashfunction -sign private.key -out file.sig file
openssl dgst -hashfunction -verify public.key -signature file.sig file
openssl enc -e -cipher -in file -out file.enc -salt
openssl enc -d -cipher -in file.enc -out file

openssl genpkey -algorithm RSA -cipher 3des -pkeyopt rsa_keygen_bits:2048 -out keypair.pem
openssl pkey -text -in private.key -noout
openssl pkey -in old.key -out new.key -cipher
openssl pkey -in old.key -out new.key

1. openssl s_client -connect www.site.com:443 > tmpfile
2. CTRL-C
3. openssl x509 -in tmpfile -text

openssl list-message-digest-commands
openssl list-cipher-commands

Read a certificate
Read a Certificate Signing Request
Generate a Certificate Signing Request, given a private key
Generate a Certificate Signing Request, creating also a 2048-bit RSA key pair (unencrypted, for non-interactive use)
Sign a certificate as a CA, given a Certificate Signing Request
Generate a self-signed root certificate, and create a new CA private key
Sign a certificate
Revoke a certificate

Convert a certificate from PEM to DER
Convert a certificate from PEM to PKCS#12 including the private key
Convert a certificate from PKCS#12 to PEM
Extract the private key from a PKCS#12 certificate
Extract the CA certificate from a PKCS#12 certificate
Create a PEM certificate from CRT and private key
Generate the digest (hash) of a file
Check the hash of a file; no output means OK
Sign a file
Verify the signature of a file
Encrypt a file
Decrypt a file
Generate a 2048-bit RSA key pair protected by a TripleDES-encrypted passphrase
Examine a private key
Change the passphrase of a private key
Remove the passphrase from a private key
Inspect an SSL certificate from a website
List all available hash functions
List all available ciphers
CA.pl -newca  
Create a Certification Authority hierarchy

CA.pl -newreq  
Generate a Certificate Signing Request

CA.pl -newreq-nodes  
Generate a Certificate Signing Request, creating also a key pair (unencrypted, for non-interactive use)

CA.pl -signreq  
Sign a Certificate Signing Request

CA.pl -pkcs12 "Certificate name"  
Generate a PKCS#12 certificate from a Certificate Signing Request

CA.pl -newcert  
Generate a self-signed certificate

CA.pl -verify  
Verify a certificate against the Certification Authority certificate for "demoCA"
GnuPG aka GPG (GNU Privacy Guard) is a well-known implementation of the OpenPGP standard described in RFC 4880. The OpenPGP standard derives from PGP (Pretty Good Privacy), the first tool for strong encryption available to the general public.

```
gpg --gen-key
Generate a key pair

gpg --import alice.asc
Import Alice's public key alice.asc into your keyring

gpg --list-keys
List the keys contained into your keyring

gpg --list-secret-keys
List your private keys contained into your keyring

gpg --list-public-keys
List the public keys contained into your keyring

gpg --export -o keyring.gpg
Export your whole keyring to a file keyring.gpg

gpg --export-secret-key -a "You" -o private.key
Export your private key to a file private.key

gpg --export-public-key -a "Alice" -o alice.pub
Export Alice's public key to a file alice.pub

gpg --edit-key "Alice"
Sign Alice's public key

gpg -e -u "You" -r "Alice" file
Sign file (with your private key) and encrypt it to Alice (with Alice's public key)

gpg -d file.gpg -o file
Decrypt file.gpg (with your own private key) and save the decrypted file to file
```
OpenVPN is an open source software that implements a Virtual Private Network (VPN) between two endpoints. The encrypted VPN tunnel uses UDP port 1194.

```
openvpn --genkey --secret keyfile
```

Generate a shared secret keyfile for OpenVPN authentication. The keyfile must be copied on both server and client.

```
openvpn server.conf
```

Start the VPN on the server side

```
openvpn client.conf
```

Start the VPN on the client side

```
/etc/openvpn/server.conf
```

Server-side configuration file:

```
dev tun
ifconfig server_IP client_IP
keepalive 10 60
ping-timer-rem
persist-tun
persist-key
secret keyfile
```

```
/etc/openvpn/client.conf
```

Client-side configuration file:

```
remote server_public_IP
dev tun
ifconfig client_IP server_IP
keepalive 10 60
ping-timer-rem
persist-tun
persist-key
secret keyfile
```
Security tools

md5sum  
sha1sum  
sha224sum  
sha256sum  
sha384sum  
sha512sum  
shasum

Print or check the digest of a file generated by a specific hashing algorithm

stunnel

TLS encryption wrapper. Can be used to secure any client-server protocol
## Key bindings - terminal

<table>
<thead>
<tr>
<th>Key</th>
<th>Alternate key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTRL + F</td>
<td>←</td>
<td>Move cursor forward one character</td>
</tr>
<tr>
<td>CTRL + B</td>
<td>←</td>
<td>Move cursor backward one character</td>
</tr>
<tr>
<td>CTRL + A</td>
<td>HOME</td>
<td>Move cursor to beginning of line</td>
</tr>
<tr>
<td>CTRL + E</td>
<td>END</td>
<td>Move cursor to end of line</td>
</tr>
<tr>
<td>CTRL + H</td>
<td>BACKSPACE</td>
<td>Delete character to the left of cursor</td>
</tr>
<tr>
<td>CTRL + W</td>
<td></td>
<td>Delete word to the left of cursor</td>
</tr>
<tr>
<td>CTRL + U</td>
<td></td>
<td>Delete all characters to the left of cursor</td>
</tr>
<tr>
<td>CTRL + K</td>
<td></td>
<td>Delete all characters to the right of cursor</td>
</tr>
<tr>
<td>CTRL + T</td>
<td></td>
<td>Swap current character with previous one</td>
</tr>
<tr>
<td>ESC + T</td>
<td></td>
<td>Swap current word with previous one</td>
</tr>
<tr>
<td>SHIFT + PAGE UP</td>
<td></td>
<td>Scroll up the screen buffer</td>
</tr>
<tr>
<td>SHIFT + PAGE DOWN</td>
<td></td>
<td>Scroll down the screen buffer</td>
</tr>
<tr>
<td>CTRL + L</td>
<td></td>
<td>Clear screen (same as <code>clear</code>)</td>
</tr>
<tr>
<td>CTRL + P</td>
<td>↑</td>
<td>Previous command in history</td>
</tr>
<tr>
<td>CTRL + N</td>
<td>↓</td>
<td>Next command in history</td>
</tr>
<tr>
<td>CTRL + R</td>
<td></td>
<td>Reverse history search</td>
</tr>
<tr>
<td>CTRL + I</td>
<td>TAB</td>
<td>Autocomplete commands, filenames, and directory names</td>
</tr>
<tr>
<td>ALT + I</td>
<td></td>
<td>Autocomplete filenames and directory names only</td>
</tr>
<tr>
<td>CTRL + ALT + E</td>
<td></td>
<td>Expand the Bash alias currently entered on the command line</td>
</tr>
<tr>
<td>CTRL + J</td>
<td>RETURN</td>
<td>Line feed</td>
</tr>
<tr>
<td>CTRL + M</td>
<td></td>
<td>Carriage return</td>
</tr>
<tr>
<td>CTRL + S</td>
<td></td>
<td>Pause transfer to terminal</td>
</tr>
<tr>
<td>CTRL + Q</td>
<td></td>
<td>Forward history search (if XON/XOFF flow control is disabled)</td>
</tr>
<tr>
<td>CTRL + Z</td>
<td></td>
<td>Resume transfer to terminal</td>
</tr>
<tr>
<td>CTRL + C</td>
<td></td>
<td>Send a SIGTSTP to put the current job in background</td>
</tr>
<tr>
<td>CTRL + D</td>
<td></td>
<td>Send a SIGINT to stop the current process</td>
</tr>
<tr>
<td>CTRL + ALT + DEL</td>
<td></td>
<td>Send a EOF to current process (same as <code>logout</code> if process is a shell)</td>
</tr>
<tr>
<td>CTRL + ALT + F1 ... F6</td>
<td></td>
<td>Send a SIGINT to reboot the machine (same as <code>shutdown -r now</code>), as specified in <code>/etc/inittab</code> and <code>/etc/init/control-alt-delete</code></td>
</tr>
<tr>
<td>CTRL + ALT + F1 ... F6</td>
<td></td>
<td>Switch between text consoles (same as <code>chvt n</code>)</td>
</tr>
<tr>
<td>Key</td>
<td>Alternate key</td>
<td>Function</td>
</tr>
<tr>
<td>-----</td>
<td>---------------</td>
<td>----------</td>
</tr>
<tr>
<td>CTRL</td>
<td>ALT</td>
<td>F7 ... F11</td>
</tr>
<tr>
<td>CTRL</td>
<td>ALT</td>
<td>↑</td>
</tr>
<tr>
<td>CTRL</td>
<td>ALT</td>
<td>↓</td>
</tr>
<tr>
<td>CTRL</td>
<td>TAB</td>
<td></td>
</tr>
<tr>
<td>CTRL</td>
<td>ALT</td>
<td>←</td>
</tr>
<tr>
<td>CTRL</td>
<td>ALT</td>
<td>→</td>
</tr>
<tr>
<td>CTRL</td>
<td>ALT</td>
<td>BACKSPACE</td>
</tr>
</tbody>
</table>

**GNOME**

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALT</td>
<td>TAB</td>
</tr>
<tr>
<td>SUPER</td>
<td>Show activities overview</td>
</tr>
<tr>
<td>SUPER</td>
<td>↓</td>
</tr>
<tr>
<td>SUPER</td>
<td>↑</td>
</tr>
<tr>
<td>SUPER</td>
<td>←</td>
</tr>
<tr>
<td>SUPER</td>
<td>→</td>
</tr>
<tr>
<td>SUPER</td>
<td></td>
</tr>
<tr>
<td>SUPER</td>
<td></td>
</tr>
<tr>
<td>ALT</td>
<td>F2</td>
</tr>
<tr>
<td>CTRL</td>
<td>↑</td>
</tr>
<tr>
<td>CTRL</td>
<td>↓</td>
</tr>
</tbody>
</table>
The Hardware Abstraction Layer (HAL) manages device files and provides plug-and-play facilities. The HAL daemon hald maintains a persistent database of devices.

udev is the device manager for the Linux kernel. It dynamically generates the device nodes in /dev/ for devices present on the system; it also provides persistent naming for storage devices in /dev/disk.

When a device is added, removed, or changes state, the kernel sends an uevent received by the udevd daemon which will pass the uevent through a set of rules stored in /etc/udev/rules.d/*.rules and /lib/udev/rules.d/*.rules.

```
udevadm monitor
udevmonitor
udevadm info --attribute-walk --name=/dev/sda
cat /sys/block/sda/size
udevadm test /dev/sdb
gnome-device-manager
```

- Show all kernel uevents and udev messages
- Print all attributes of device /dev/sda in udev rules key format
- Print the size attribute of disk sda in 512-byte blocks. This information is retrieved from sysfs
- Simulate an udev event run for the device and print debug output
- Browser for the HAL device manager

<table>
<thead>
<tr>
<th>/etc/udev/rules.d/<em>.rules and /lib/udev/rules.d/</em>.rules</th>
<th>udev rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>KERNEL=&quot;hda&quot;, NAME=&quot;mydisk&quot;</td>
<td>Match a device which was named by the kernel as hda; name the device node as &quot;mydisk&quot;. The device node will be therefore /dev/mydisk</td>
</tr>
<tr>
<td>KERNEL=&quot;hdb&quot;, DRIVER=&quot;ide-disk&quot;, SYMLINK=&quot;mydisk myhd&quot;</td>
<td>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</td>
</tr>
</tbody>
</table>
| KERNEL="fd[0-9]*", NAME="floppy/%n", SYMLINK="%k" | Match all floppy disk drives (i.e. fdn); place device node in /dev/floppy/n and create a symlink /dev/fd
| 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
/lib/modules/X.Y.Z/modules.dep
```

Kernel modules for kernel version X.Y.Z

```
/lib/modules/X.Y.Z/*.ko
/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/
/usr/src/linux/.config
```

Directory containing the kernel source code to be compiled

Kernel configuration file

```
freeramdisk
```

Free the memory used for the initrd image. This command must be run directly after unmounting /initrd

```
mkinitrd initrd_image kernel_version (Red Hat)
mkinitramfs (Debian)
dracut
```

Create an initrd image file

```
dbus-monitor
```

Monitor messages going through a D-Bus message bus

```
dbus-monitor --session
```

Monitor session messages (default)

```
dbus-monitor --system
```

Monitor system messages

```
kexec -l kernel_image --append=options \  --initrd=initrd_image && kexec -e
```

Load a kernel image file into memory and boot it. This allows running a different kernel without rebooting the machine

The runtime loader ld.so loads the required shared libraries of the program into RAM, searching in this order:

1. LD_LIBRARY_PATH Environment variable specifying the list of dirs where libraries should be searched for first
2. /etc/ld.so.cache Cache file
3. /lib and /usr/lib Default locations for shared libraries

Shared library locations (other than the default ones /lib and /usr/lib) can be specified in the file /etc/ld.so.conf.

```
ldconfig
```

Create a cache file /etc/ld.so.cache of all available dynamically linked libraries. This command should be run when the system complains about missing libraries

```
ldd program_or_lib
```

Print library dependencies
Kernel management

lspci
List PCI devices
lspci -d 8086:
List all Intel hardware present. PCI IDs are stored in:
    /usr/share/misc/pci.ids (Debian)
    /usr/share/hwdata/pci.ids (Red Hat)

lsusb
List USB devices
lsusb -d 8086:
List all Intel USB devices present. USB IDs are stored in:
    /var/lib/usbutils/usb.ids (Debian)
    /usr/share/hwdata/usb.ids (Red Hat)

lsdev
List information about the system hardware

lshw
List system hardware

lscpu
List information about the CPU architecture

uname
Print system information. Values that can be printed are:
    -s  Kernel name
    -n  Network node hostname
    -r  Kernel release number X.Y.Z
    -v  Kernel version number
    -m  Machine hardware name
    -p  Processor type
    -i  Hardware platform
    -o  Operating system
    -a  All the above information, in that order

evtest
Monitor and query input device events in /dev/input/eventn

dmesg
Print the messages of the kernel ring buffer. Options are:
    -T  Print human-readable timestamps
    -n 1  Set the logging level to 1 (= only panic messages)

journalctl
Display the Systemd journal, which contains the kernel logs
    journalctl -n n
    journalctl --since "1 hour ago"
    journalctl -x
    journalctl -f
    journalctl -u crond.service
    journalctl _SYSTEMD_UNIT=crond.service

mkdir -p /var/log/journal/ && 
systemctl restart systemd-journald

Enable persistent storage of logs in /var/log/journal/ (by default, journalctl stores the logfiles in RAM only)
## Kernel compile

<table>
<thead>
<tr>
<th>Download</th>
<th>Download the kernel source code <a href="http://www.kernel.org">linux-X.Y.Z.tar.bz2</a> from <a href="http://www.kernel.org">http://www.kernel.org</a> to the base of the kernel source tree /usr/src/linux</th>
</tr>
</thead>
</table>
| Clean    | **make clean** Delete most generated files  
|          | **make mrproper** Delete all generated files and kernel configuration  
|          | **make distclean** Delete temporary files, patch leftovers, and similar files |
| Configure| **make config** Terminal-based (options must be set in sequence)  
|          | **make menuconfig** Ncurses UI  
|          | **make xconfig** GUI  
|          | **make gconfig**  
|          | **make oldconfig** Create a new configuration file, based on the options in the old configuration file and in the source code |
| 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 |
| 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

<table>
<thead>
<tr>
<th>Download</th>
<th>Download and decompress the patch to /usr/src</th>
</tr>
</thead>
</table>
| Patch    | **patch -p1 < file.patch** Apply the patch  
|          | **patch -Rp1 < file.patch** Remove (reverse) a patch. Alternatively, applying the patch again reverses it |
| Build    | Build the patched kernel as explained above |
| Install  | Install the patched kernel as explained above |
Kernel modules allow the kernel to access functions (symbols) for kernel services e.g. hardware drivers, network stack, or filesystem abstraction.

**lsmod** List the modules that are currently loaded into the kernel

**insmod module** Insert a module into the kernel. If the module requires another module or if it does not detect compatible hardware, insertion will fail

**rmmod module** Remove a module from the kernel. If the module is in use by another module, it is necessary to remove the latter first

**modinfo module** Display the list of parameters accepted by the module

**depmod -a** Probe all modules in the kernel modules directory and generate the file that lists their dependencies

It is recommended to use `modprobe` instead of `insmod` and `rmmod`, because it automatically handles prerequisites when inserting modules, is more specific about errors, and accepts just the module name instead of requiring the full pathname.

**modprobe module option=value** Insert a module into the running kernel, with the specified parameters. Prerequisite modules will be inserted automatically

**modprobe -a** Insert all modules

**modprobe -t directory** Attempt to load all modules contained in the directory until a module succeeds. This action probes the hardware by successive module-insertion attempts for a single type of hardware, e.g. a network adapter

**modprobe -r module** Remove a module

**modprobe -c module** Display module configuration

**modprobe -l** List loaded modules

### Configuration of device drivers

Device drivers support the kernel with instructions on how to use that device.

<table>
<thead>
<tr>
<th>Device driver compiled into the kernel</th>
<th>Device driver provided as a kernel module</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Configure the device driver by passing a kernel parameter in the GRUB menu:</strong></td>
<td><strong>Edit module configuration in <em>/etc/modprobe.conf</em> or <em>/etc/modprobe.d/</em> (Red Hat):</strong></td>
</tr>
<tr>
<td>kernel /vmlinuz ro root=/dev/vg0/root vga=0x33c</td>
<td>alias eth0 3c59x Specify that eth0 uses the 3c59x.ko driver module</td>
</tr>
<tr>
<td>options 3c509 irq=10,11 Assign IRQ 10 and 11 to 3c509 devices</td>
<td></td>
</tr>
</tbody>
</table>
**/proc** is a pseudo filesystem that gives access to process data held in the kernel.

<table>
<thead>
<tr>
<th>File</th>
<th>Information stored (can be viewed via <code>cat</code>)</th>
<th>Equivalent command</th>
</tr>
</thead>
<tbody>
<tr>
<td>/proc/bus</td>
<td>Buses (e.g. PCI, USB, PC Card)</td>
<td></td>
</tr>
<tr>
<td>/proc/cpuinfo</td>
<td>CPUs information</td>
<td></td>
</tr>
<tr>
<td>/proc/devices</td>
<td>Drivers currently loaded</td>
<td></td>
</tr>
<tr>
<td>/proc/dma</td>
<td>DMA channels in use</td>
<td></td>
</tr>
<tr>
<td>/proc/filesystems</td>
<td>Filesystems supported by the system</td>
<td></td>
</tr>
<tr>
<td>/proc/interrupts</td>
<td>Current IRQs (Interrupt Requests)</td>
<td>procinfo</td>
</tr>
<tr>
<td>/proc/ioports</td>
<td>I/O addresses in use</td>
<td></td>
</tr>
<tr>
<td>/proc/kcore</td>
<td>Memory allocatable by the kernel</td>
<td></td>
</tr>
<tr>
<td>/proc/loadavg</td>
<td>System load averages</td>
<td>uptime</td>
</tr>
<tr>
<td>/proc/mdstat</td>
<td>Information about RAID arrays and devices</td>
<td></td>
</tr>
<tr>
<td>/proc/meminfo</td>
<td>Total and free memory</td>
<td>free</td>
</tr>
<tr>
<td>/proc/modules</td>
<td>Kernel modules currently loaded</td>
<td>lsmod</td>
</tr>
<tr>
<td>/proc/mounts</td>
<td>Mounted partitions</td>
<td>mount</td>
</tr>
<tr>
<td>/proc/net/dev</td>
<td>Network interface statistics</td>
<td></td>
</tr>
<tr>
<td>/proc/partitions</td>
<td>Drive partition information</td>
<td>fdisk -l</td>
</tr>
<tr>
<td>/proc/swaps</td>
<td>Size of total and used swap areas</td>
<td>swapon -s</td>
</tr>
<tr>
<td>/proc/sys/</td>
<td><code>sysfs</code>: exposes tunable kernel parameters</td>
<td></td>
</tr>
<tr>
<td>/proc/sys/kernel/</td>
<td>Kernel information and parameters</td>
<td></td>
</tr>
<tr>
<td>/proc/sys/net/</td>
<td>Network information and parameters</td>
<td></td>
</tr>
<tr>
<td>/proc/uptime</td>
<td>Time elapsed since boot</td>
<td>uptime</td>
</tr>
<tr>
<td>/proc/version</td>
<td>Linux version</td>
<td>uname -a</td>
</tr>
<tr>
<td>/proc/n/</td>
<td>Information about process with PID <code>n</code></td>
<td>ps <code>n</code></td>
</tr>
<tr>
<td>/proc/n/cmdline</td>
<td>Command by which the process was launched</td>
<td></td>
</tr>
<tr>
<td>/proc/n/cwd</td>
<td>Symlink to process' working directory</td>
<td></td>
</tr>
<tr>
<td>/proc/n/environ</td>
<td>Values of environment variables of process</td>
<td></td>
</tr>
<tr>
<td>/proc/n/exe</td>
<td>Symlink to process' executable</td>
<td></td>
</tr>
<tr>
<td>/proc/n/fd</td>
<td>Files currently opened by the process</td>
<td>ls -l</td>
</tr>
<tr>
<td>/proc/n/root</td>
<td>Symlink to process' filesystem root</td>
<td></td>
</tr>
<tr>
<td>/proc/n/status</td>
<td>Status of process</td>
<td></td>
</tr>
</tbody>
</table>

**/proc/sys** is the only writable branch of **/proc** and can be used to tune kernel parameters on-the-fly. All changes are lost after system shutdown, unless applied via `sysctl -p`.

```bash
sysctl fs.file-max
get the maximum allowed number of open files

cat /proc/sys/fs/file-max

sysctl -w "fs.file-max=100000"
set the maximum allowed number of open files to 100000

echo "100000" > /proc/sys/fs/file-max

sysctl -a
list all available kernel tuning options

sysctl -p
apply all tuning settings listed in `/etc/sysctl.conf`. This command is usually run at boot by the system initialization script, to make permanent changes to kernel parameters
```
The directory **/dev** contains the device files to access all devices in the system.

<table>
<thead>
<tr>
<th>File</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/sda</td>
<td>SCSI, PATA, or SATA hard drive</td>
</tr>
<tr>
<td>/dev/hda</td>
<td>IDE hard drive</td>
</tr>
<tr>
<td>/dev/pda</td>
<td>Parallel port IDE hard drive</td>
</tr>
<tr>
<td>/dev/vda</td>
<td>Virtual disk for KVM-based virtual machines</td>
</tr>
<tr>
<td>/dev/sda, /dev/sdb, /dev/sdc ...</td>
<td>First, second, third ... hard drive</td>
</tr>
<tr>
<td>/dev/sda1, /dev/sda2, /dev/sda3 ...</td>
<td>First, second, third ... partition of the first hard drive</td>
</tr>
<tr>
<td>/dev/md0</td>
<td>Metadisk group, for use with RAID</td>
</tr>
<tr>
<td>/dev/sr0</td>
<td>SCSI CD-ROM</td>
</tr>
<tr>
<td>/dev/pcd0</td>
<td>Parallel port CD-ROM</td>
</tr>
<tr>
<td>/dev/cdrom</td>
<td>CD-ROM. Usually symlinked to /dev/sr0</td>
</tr>
<tr>
<td>/dev/fd0</td>
<td>Floppy disk drive</td>
</tr>
<tr>
<td>/dev/ht0</td>
<td>IDE tape drive</td>
</tr>
<tr>
<td>/dev/pt0</td>
<td>Parallel port tape drive</td>
</tr>
<tr>
<td>/dev/sg0</td>
<td>Generic SCSI device</td>
</tr>
<tr>
<td>/dev/loop0</td>
<td>Loopback device</td>
</tr>
<tr>
<td>/dev/autofs</td>
<td>AutoFS device</td>
</tr>
<tr>
<td>/dev/fuse</td>
<td>FUSE device</td>
</tr>
<tr>
<td>/dev/dsp</td>
<td>Digital Signal Processor device. Interfaces with the soundcard</td>
</tr>
<tr>
<td>/dev/fb0</td>
<td>Framebuffer device. Interfaces with the graphics hardware</td>
</tr>
<tr>
<td>/dev/1p0</td>
<td>Parallel port printer device</td>
</tr>
<tr>
<td>/dev/parport0</td>
<td>Raw parallel port device</td>
</tr>
<tr>
<td>/dev/mem</td>
<td>Physical memory</td>
</tr>
<tr>
<td>/dev/kmem</td>
<td>Kernel virtual memory</td>
</tr>
<tr>
<td>/dev/core</td>
<td>Obsolete. Symlink to /proc/kcore</td>
</tr>
<tr>
<td>/dev/stdin</td>
<td>Standard Input</td>
</tr>
<tr>
<td>/dev/stdout</td>
<td>Standard Output</td>
</tr>
<tr>
<td>/dev/stderr</td>
<td>Standard Error</td>
</tr>
<tr>
<td>/dev/null</td>
<td>Null device, aka blackhole or bit bucket. Discards any received data</td>
</tr>
<tr>
<td>/dev/zero</td>
<td>Zero device. Outputs an infinite stream of zero bytes (NUL) on reads</td>
</tr>
<tr>
<td>/dev/full</td>
<td>&quot;Always full&quot; device. Similar to /dev/zero, and also returns an error &quot;No space left on device&quot; (ENOSPC) on writes</td>
</tr>
<tr>
<td>/dev/random</td>
<td>Non-deterministic random number generator. Gathers entropy from the system to generate randomness; once the entropy pool is depleted, the device blocks all reads until it can collect more entropy</td>
</tr>
<tr>
<td>/dev/urandom</td>
<td>Pseudo random number generator. Faster but unsafe for cryptographic purposes</td>
</tr>
<tr>
<td>/dev/console</td>
<td>System console</td>
</tr>
<tr>
<td>/dev/tty</td>
<td>Terminal for current process</td>
</tr>
<tr>
<td>/dev/tty0</td>
<td>Current virtual console</td>
</tr>
<tr>
<td>/dev/ttyS0</td>
<td>Serial port, usually used for modem connections</td>
</tr>
<tr>
<td>/dev/pty0</td>
<td>Pseudo-TTY master</td>
</tr>
<tr>
<td>/dev/ttyP0</td>
<td>Pseudo-TTY slave</td>
</tr>
</tbody>
</table>
If the kernel has been booted in emergency mode and init has not been run, some initial configuration is necessary e.g.

```
mount /proc
mount -o remount,rw /
mount -a
```

If mounting the filesystems fails:

```
mknod /dev/sda
mknod /dev/sda1
fdisk -l /dev/sda
fsck -y /dev/sda1
mount -t ext3 /dev/sda1 /mnt/sysimage
chroot /mnt/sysimage
```

To install a package using an alternative root directory (useful if the system has been booted from a removable media):

```
rpm -U --root /mnt/sysimage package.rpm
```

To install GRUB on the specified directory (which must contain /boot/grub/):

```
grub-install --root-directory=/mnt/sysimage /dev/sda
```

Alternative method:

```
chroot /mnt/sysimage
grub-install /dev/sda
```

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

**How to reset the root password (RHEL 7 and 8)**

1. Power up the system and, once on the GRUB 2 boot screen, press `CTRL+X` to edit the current entry
2. On the kernel line that mentions `linux16`, remove the `rhgb` and `quiet` parameters and add `rd.break` at the end
3. Press `CTRL+X`; the system will boot on the initramfs switch_root prompt
4. Remount the filesystem as writable
```
mount -o remount,rw /sysroot
```
```
chroot /sysroot
```
```
passwd root
```
```
touch /autoconf
```
```
mount -o remount,ro /sysroot
```
5. Change the filesystem root
6. Modify the root password
7. Force SELinux to relabel context on next boot
8. Remount the filesystem as readonly (not strictly necessary)
9. Exit the chroot environment
10. Resume system boot
```
# DNS

## DNS implementations

<table>
<thead>
<tr>
<th>Platform</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIND</td>
<td>Berkeley Internet Name Domain system, is the standard DNS server for UNIX</td>
</tr>
<tr>
<td>Unbound</td>
<td>Standard DNS server in RHEL 7</td>
</tr>
<tr>
<td>dnsmasq</td>
<td>Lightweight DNS, DHCP and TFTP server for a small network</td>
</tr>
<tr>
<td>djbdns</td>
<td>Security-hardened DNS server that also includes DNS debugging tools</td>
</tr>
<tr>
<td>PowerDNS</td>
<td>Alternative open-source DNS server</td>
</tr>
</tbody>
</table>

- **named**: BIND Name Daemon
- **ndc**: Name Daemon Controller for BIND 8
- **rndc**: Remote 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 `ffffff`. This will create two key files:
  - `Kdns1.example.org.+nnn+ffffff.key`
  - `Kdns1.example.org.+nnn+ffffff.private`
  which contain a key number that must be inserted both in `/etc/named.conf` and `/etc/rndc.conf`

- `rndc-confgen -a`: Generate a `/etc/rndc.key` key file:

  ```plaintext```
  ```
  key "rndc-key" {
    algorithm hmac-md5;
    secret "vy2gL3tPhsnA57e4LT0Ek==";
    }
  ```

  ```plaintext```
  ```
  options { 
    default-key "rndc-key";
    default-server 127.0.0.1;
    default-port 953;
  }
  ```

  This file is automatically read both by *named* and *rndc*

- `dnssec-signzone example.org`: Sign the zone `example.org`

- `named -u named -g named`: Run BIND as user/group "named" (must be created if needed) instead of root
- `named -t /var/cache/bind`: Run BIND in a chroot jail `/var/cache/bind`
  (actually it is the chroot command that starts the *named* server)
### /etc/named.conf

DNS server configuration file

```bash
controls {
    inet 127.0.0.1 allow {localhost;} keys {rndckey;};
};
key "rndc-key" {
    algorithm dsa;
    secret "HYZur46fftdU43BJKI093t4t78lkp";
};

acl "mynetwork" {10.7.0.0/24;};
// Built-in ACLs: any, none, localhost, localnets

options {
    directory "/var/named";
    version "0.0";
    listen-on port 53 {10.7.0.1; 127.0.0.1;};
    blackhole {172.17.17.0/24;};
    allow-query {mynetwork;};
    allow-query-on {any;};
    allow-query-cache {any;};
    allow-recursion {mynetwork;};
    allow-recursion-on {mynetwork;};
    allow-transfer {10.7.0.254;};
    allow-update {any;};
    recursive-clients 1000;
    dnssec-enable yes;
    dialup no;
    forward first;
    forwarders {10.7.0.252; 10.7.0.253;};

    // 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";
        masters {10.7.0.254;};
    }

    // Configure system to act as a slave server for the example2.org domain
    zone "example2.org" IN {
        type slave;
        file "slave/example2.org.zone";
    }

    zone "240.123.224.in-addr.arpa" IN {
        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";
    }

    zone "0.7.10.in-addr.arpa" IN {
        type slave;
        file "slave/10.7.0.revzone";
        masters {10.7.0.254;};
    }
}
```
**DNS zone file**

```
/var/named/master/example.org.zone

// DNS zone file

$TTL 86400 ; TTL (1 day)
$ORIGIN example.org.
example.org IN SOA dns1.example.org. help.example.org. (  
    2014052300 ; serial          ; Master DNS server is dns1.example.org
    28800 ; refresh (8 hours)    ; If problems, contact help@example.org
    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

$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

<table>
<thead>
<tr>
<th>Resource Record</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTL</td>
<td>How long to cache a positive response</td>
</tr>
<tr>
<td>ORIGIN</td>
<td>Suffix appended to all names not ending with a dot. Useful when defining multiple subdomains inside the same zone</td>
</tr>
<tr>
<td>SOA</td>
<td>Start Of Authority for the example.org zone</td>
</tr>
<tr>
<td></td>
<td>serial</td>
</tr>
<tr>
<td></td>
<td>refresh</td>
</tr>
<tr>
<td></td>
<td>retry</td>
</tr>
<tr>
<td></td>
<td>expire</td>
</tr>
<tr>
<td></td>
<td>negative TTL</td>
</tr>
<tr>
<td>A</td>
<td>Address: maps names to IP addresses. Used for DNS lookups.</td>
</tr>
<tr>
<td>PTR</td>
<td>Pointer: maps IP addresses to names. Used for reverse DNS lookups. Each A record must have a matching PTR record</td>
</tr>
<tr>
<td>CNAME</td>
<td>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</td>
</tr>
<tr>
<td>NS</td>
<td>Name Service: specifies the authoritative name servers for the zone</td>
</tr>
<tr>
<td>MX</td>
<td>Mailserver: specifies address and priority of the servers able to handle mail for the zone</td>
</tr>
</tbody>
</table>

Note: Glue Records are not really part of the zone; they delegate authority for other zones, usually subdomains.
## HTTP response codes

<table>
<thead>
<tr>
<th>Code Range</th>
<th>Category</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1XX</td>
<td>Informational</td>
<td>100</td>
<td>Continue</td>
</tr>
<tr>
<td></td>
<td></td>
<td>101</td>
<td>Switching Protocols</td>
</tr>
<tr>
<td>2XX</td>
<td>Success</td>
<td>200</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>201</td>
<td>Created</td>
</tr>
<tr>
<td></td>
<td></td>
<td>204</td>
<td>No Content</td>
</tr>
<tr>
<td></td>
<td></td>
<td>206</td>
<td>Partial Content</td>
</tr>
<tr>
<td>3XX</td>
<td>Redirection</td>
<td>301</td>
<td>Moved Permanently</td>
</tr>
<tr>
<td></td>
<td></td>
<td>302</td>
<td>Found</td>
</tr>
<tr>
<td></td>
<td></td>
<td>303</td>
<td>See Other</td>
</tr>
<tr>
<td></td>
<td></td>
<td>304</td>
<td>Not Modified</td>
</tr>
<tr>
<td></td>
<td></td>
<td>307</td>
<td>Temporary Redirect</td>
</tr>
<tr>
<td>4XX</td>
<td>Client Error</td>
<td>400</td>
<td>Bad Request</td>
</tr>
<tr>
<td></td>
<td></td>
<td>401</td>
<td>Unauthorized</td>
</tr>
<tr>
<td></td>
<td></td>
<td>403</td>
<td>Forbidden</td>
</tr>
<tr>
<td></td>
<td></td>
<td>404</td>
<td>Not Found</td>
</tr>
<tr>
<td></td>
<td></td>
<td>408</td>
<td>Request Timeout</td>
</tr>
<tr>
<td></td>
<td></td>
<td>409</td>
<td>Conflict</td>
</tr>
<tr>
<td></td>
<td></td>
<td>410</td>
<td>Gone</td>
</tr>
<tr>
<td></td>
<td></td>
<td>451</td>
<td>Unavailable for Legal Reasons</td>
</tr>
<tr>
<td>5XX</td>
<td>Server Error</td>
<td>500</td>
<td>Internal Server Error</td>
</tr>
<tr>
<td></td>
<td></td>
<td>501</td>
<td>Not Implemented</td>
</tr>
<tr>
<td></td>
<td></td>
<td>502</td>
<td>Bad Gateway</td>
</tr>
<tr>
<td></td>
<td></td>
<td>503</td>
<td>Service Unavailable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>504</td>
<td>Gateway Timeout</td>
</tr>
<tr>
<td></td>
<td></td>
<td>505</td>
<td>HTTP Version Not Supported</td>
</tr>
</tbody>
</table>
Apache is an open source and widespread HTTP server, originally based on the NCSA HTTPd server.

apachectl (Red Hat) Manage the Apache webserver
httpd (Red Hat)
apache2ctl (Debian)

apachectl start Start the Apache webserver daemon
apachectl status Display a brief status report
apachectl fullstatus Display a detailed status report
apachectl graceful Gracefully restart Apache; currently open connections are not aborted
apachectl graceful-stop Gracefully stop Apache; currently open connections are not aborted
apachectl configtest Test the configuration file, reporting any syntax error
apachectl -t
apachectl -M List all loaded and shared modules

/var/www/html Default document root directory
$HOME/public_html Default document root directory for users' websites

Web content must be readable by the user/group the Apache process runs as. For security reasons, it should be owned and writable by the superuser or the webmaster user/group (usually www-data), not the Apache user/group.

/etc/httpd/conf/httpd.conf
/etc/httpd/conf.d/*.conf (Red Hat) Apache configuration files
/etc/apache2/httpd.conf (Debian and SUSE)

The Apache webserver contains a number of MPMs (Multi-Processing Modules) which can operate following two methods:

prefork MPM A number of child processes is spawned in advance, with each child serving one connection. Highly reliable due to Linux memory protection that isolates each child process.

worker MPM Multiple child processes spawn multiple threads, with each thread serving one connection. More scalable but prone to deadlocks if third-party non-threadsafe modules are loaded.

HTTPS

HTTPS (i.e. HTTP over SSL/TLS) allows securing communications between the webserver and the client by encrypting all communications end-to-end between the two. A webserver using HTTPS hands over its public key to the client when the client connects to the server via port 443. The server's public key is signed by a CA (Certification Authority), whose validity is ensured by the root certificates stored into the client's browser.

The openssl command and its user-friendly CA.pl script are the tools of the OpenSSL crypto library that can be used to accomplish all public key crypto operations e.g. generate key pairs, Certificate Signing Requests, and self-signed certificates. Another user-friendly tool is genkey.

Virtual hosting with HTTPS requires assigning a unique IP address for each virtual host; this because the SSL handshake (during which the server sends its certificate to the client's browser) takes place before the client sends the Host: header (which tells to which virtual host the client wants to talk).

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

<table>
<thead>
<tr>
<th>Server configuration directives</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ServerName</strong> <a href="http://www.mysite.org:80">www.mysite.org:80</a></td>
<td>Name and port (if omitted, uses default HTTP port 80) of server</td>
</tr>
<tr>
<td><strong>ServerRoot</strong> /etc/httpd</td>
<td>Root directory for configuration and log files</td>
</tr>
<tr>
<td><strong>ServerAdmin</strong> <a href="mailto:webmaster@mysite.org">webmaster@mysite.org</a></td>
<td>Contact address that the server includes in any HTTP error messages to the client. Can be an email address or a URL</td>
</tr>
<tr>
<td><strong>StartServers</strong> 5</td>
<td>Number of servers to start initially</td>
</tr>
<tr>
<td><strong>MinSpareServers</strong> 5</td>
<td>Minimum and maximum number of idle child server processes</td>
</tr>
<tr>
<td><strong>MaxSpareServers</strong> 10</td>
<td></td>
</tr>
<tr>
<td><strong>MaxClients</strong> 256</td>
<td>(before 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</td>
</tr>
<tr>
<td><strong>MaxRequestWorkers</strong> 256</td>
<td>(v2.3.13 and later)</td>
</tr>
<tr>
<td><strong>ServerLimit</strong> 256</td>
<td>Prefork MPM: max configured value for MaxRequestWorkers. Worker MPM: in conjunction with ThreadLimit, max configured value for MaxRequestWorkers</td>
</tr>
<tr>
<td><strong>ThreadsPerChild</strong> 25</td>
<td>Worker MPM: number of threads created by each child process</td>
</tr>
<tr>
<td><strong>ThreadLimit</strong> 64</td>
<td>Worker MPM: max configured value for ThreadsPerChild</td>
</tr>
<tr>
<td><strong>MaxRequestsPerChild</strong> 16</td>
<td>(v2.2) Max number of connections allowed per child</td>
</tr>
<tr>
<td><strong>MaxConnectionsPerChild</strong> 16</td>
<td>(v2.4)</td>
</tr>
<tr>
<td><strong>LoadModule</strong> mime_module modules/mod_mime.so</td>
<td>Load the module mime_module by linking in the object file or library modules/mod_mime.so</td>
</tr>
<tr>
<td><strong>Listen</strong> 10.17.1.1:80</td>
<td>Make the server accept connections on the specified IP addresses (optional) and ports</td>
</tr>
<tr>
<td><strong>Listen</strong> 10.17.1.5:8080</td>
<td></td>
</tr>
<tr>
<td><strong>User</strong> nobody</td>
<td>User and group the Apache process runs as. For security reasons, this should not be root</td>
</tr>
<tr>
<td><strong>Group</strong> nobody</td>
<td></td>
</tr>
</tbody>
</table>
# Apache main configuration

<table>
<thead>
<tr>
<th>Apache configuration file</th>
<th>Main configuration directives</th>
</tr>
</thead>
<tbody>
<tr>
<td>DocumentRoot /var/www/html</td>
<td>Directory in filesystem that maps to the root of the website</td>
</tr>
<tr>
<td>Alias /image /mydir/pub/image</td>
<td>Map the URL <a href="http://www.mysite.org/image/">http://www.mysite.org/image/</a> to the directory /mydir/pub/image in the filesystem. This allows Apache to serve content placed outside of the document root</td>
</tr>
<tr>
<td>TypesConfig conf/mime.types</td>
<td>Media types file. The path is relative to ServerRoot</td>
</tr>
<tr>
<td>AddType image/jpeg jpeg jpg jpeg</td>
<td>Map the specified filename extensions onto the specified content type. These entries add to or override the entries from the media types file conf/mime.types</td>
</tr>
<tr>
<td>Redirect permanent /foo /bar</td>
<td>Redirect to a URL on the same host. Status can be: permanent return an HTTP status 301 - Moved Permanently temp return an HTTP status 302 - Found (default) seerter return an HTTP status 303 - See Other gone return an HTTP status 410 - Gone</td>
</tr>
<tr>
<td>Redirect /foo <a href="http://www.example.com/foo">http://www.example.com/foo</a></td>
<td>Redirect to a URL on a different host</td>
</tr>
<tr>
<td>AccessFileName .htaccess</td>
<td>Name of the distributed configuration file, which contains directives that apply to the document directory it is in and to all its subtrees</td>
</tr>
</tbody>
</table>

### Limited scope directives

<Directory "/var/www/html/foobar">
  AllowOverride AuthConfig Limit
</Directory>

<Directory "/var/www/html/foobar">
  [list of directives]
</Directory>

<Location /foobar>
  [list of directives]
</Location>

### Logging directives

| LogFormat "%h %l %u %t "%r" %>s %b" common | Specify the format of a log |
| LogFormat "%h %l %u %t "%r" %>s %b" common | Specify the nickname for a log format. In this case, specifies "common" for the CLF (Common Log Format) which is defined as such: %h IP address of the client host %l Identity of client as determined by identd %u User ID of client making the request %t Timestamp the server completed the request %r Request as done by the user %s Status code sent by the server to the client %b Size of the object returned, in bytes |
| CustomLog /var/log/httpd/access_log common | Set up a log filename, with the format or (as in this case) the nickname specified |
| TransferLog /var/log/httpd/access_log | Set up a log filename, with format determined by the most recent LogFormat directive which did not define a nickname |
| TransferLog "|rotatelogs access_log 86400" | Set log rotation every 24 hours |
| HostnameLookups Off | Disable DNS hostname lookup to save network traffic. Hostnames can be resolved later by processing the log file: logresolve <access_log >accessdns_log |
### Apache configuration file

**Virtual hosts directives**

| NameVirtualHost * (v2.2) | Specify which IP address will serve virtual hosting. The argument can be an IP address, an address:port pair, or * for all IP addresses of the server. The same argument need to be inserted in the relevant `<VirtualHost>` directive |
| `<VirtualHost *:80>` | 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>` | 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 *:80>` | Port-based virtual host answering to connections on port 8080. A `Listen 8080` directive must also be present |
| `<VirtualHost 10.17.1.5:80>` | IP-based virtual host answering to http://10.17.1.5 |
### Apache configuration file

<table>
<thead>
<tr>
<th>Authorization directives</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AuthName &quot;Protected zone&quot;</td>
<td>Name of the realm. The client will be shown the realm name and prompted to enter a user and password</td>
</tr>
<tr>
<td>AuthType Basic</td>
<td>Type of user authentication: Basic, Digest, Form, or None</td>
</tr>
<tr>
<td>AuthUserFile &quot;/var/www/.htpasswd&quot;</td>
<td>User database file. Each line has the format user:encryptedpassword. To add a user to the database file, use the command: htpasswd /var/www/.htpasswd user (will prompt for password)</td>
</tr>
<tr>
<td>AuthGroupFile &quot;/var/www/.htgroup&quot;</td>
<td>Group database file. Each line specifies a group followed by the usernames of all its members: group: user1 user2 user3</td>
</tr>
<tr>
<td>Require valid-user</td>
<td>Control who can access the protected resource. valid-user any user in the user database file user user only the specified user group group only the members of the specified group</td>
</tr>
<tr>
<td>Satisfy Any</td>
<td>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</td>
</tr>
<tr>
<td>Allow from 10.13.13.0/24 Deny from 10.13.14.0/24 (v2.2)</td>
<td>Control which host can or cannot access the protected resource</td>
</tr>
<tr>
<td>Order Allow,Deny (v2.2)</td>
<td>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</td>
</tr>
<tr>
<td><strong>Apache configuration file</strong></td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--</td>
</tr>
<tr>
<td><strong>SSL/TLS directives</strong> (mod_ssl module)**</td>
<td>**</td>
</tr>
<tr>
<td>SSLCertificateFile \ /etc/httpd/conf/ssl.crt/server.crt</td>
<td>SSL server certificate</td>
</tr>
<tr>
<td>SSLCertificateKeyFile \ /etc/httpd/conf/ssl.key/server.key</td>
<td>SSL server private key (for security reasons, this file must be mode 600 and owned by root)</td>
</tr>
<tr>
<td>SSLCACertificatePath \ /usr/local/apache2/conf/ssl.crt/</td>
<td>Directory containing the certificates of CAs. Files in this directory are PEM-encoded and accessed via symlinks to hash filenames</td>
</tr>
<tr>
<td>SSLCACertificateFile \ /usr/local/apache2/conf/ssl.crt/ca-bundle.crt</td>
<td>Certificates of CAs. Certificates are PEM-encoded and concatenated in a single bundle file in order of preference</td>
</tr>
<tr>
<td>SSLCertificateChainFile \ /usr/local/apache2/conf/ssl.crt/ca.crt</td>
<td>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</td>
</tr>
<tr>
<td>SSLEngine on</td>
<td>Enable the SSL/TLS Protocol Engine</td>
</tr>
<tr>
<td>SSLProtocol +SSLv3 +TLSv1.2</td>
<td>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)</td>
</tr>
<tr>
<td>SSLCipherSuite \ ALL:!aDH:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP</td>
<td>Cipher suite available for the SSL handshake (key exchange algorithms, authentication algorithms, cipher/encryption algorithms, MAC digest algorithms)</td>
</tr>
<tr>
<td>ServerTokens Full</td>
<td>Server response header field to send back to client. Possible values are: Prod sends Server: Apache Major sends Server: Apache/2 Minor sends Server: Apache/2.4 Minimal sends Server: Apache/2.4.2 OS sends Server: Apache/2.4.2 (Unix) Full sends Server: Apache/2.4.2 (Unix) \ PHP/4.2.2 MyMod/1.2 (default)</td>
</tr>
<tr>
<td>ServerSignature Off</td>
<td>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</td>
</tr>
<tr>
<td>SSLVerifyClient none</td>
<td>Certificate verification level for client authentication. Possible values are: none no client certificate is required require the client needs to present a valid certificate optional the client may present a valid certificate (this option is unused as it doesn't work on all browsers) optional_no_ca the client may present a valid certificate but it doesn't need to be successfully verifiable (this option is practically used only for SSL testing)</td>
</tr>
<tr>
<td>TraceEnable on</td>
<td>Enable TRACE requests</td>
</tr>
</tbody>
</table>
A **forward proxy** provides proxy services, typically web content caching and/or filtering, for clients located in a LAN. All outgoing requests from the clients, and the responses from the Internet, pass through the proxy. The clients must be manually configured (e.g. in the browser’s connection settings) to use the proxy.

**Apache configuration file**

<table>
<thead>
<tr>
<th>Forward proxy</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ProxyRequests On</td>
<td>Enable forward proxy requests</td>
</tr>
<tr>
<td>ProxyVia On</td>
<td>Add a <code>Via:</code> HTTP header line to every request and reply</td>
</tr>
<tr>
<td><code>&lt;Proxy &quot;***&quot;&gt;</code></td>
<td></td>
</tr>
<tr>
<td>Require ip 10.1.1</td>
<td>Serve only proxy requests coming from 10.1.1/24</td>
</tr>
<tr>
<td><code>&lt;/Proxy&gt;</code></td>
<td></td>
</tr>
</tbody>
</table>

A **reverse proxy** aka **gateway** allows to expose a single entry point for one or more webservers in a LAN. This improves security and simplifies management, as features (e.g. load balancing, firewalling, automatic redirection from HTTP to HTTPS, redirection on default ports) can be configured centrally. It is necessary to create a DNS A record that maps site.example.com to the public IP address of the proxy.

**Apache configuration file**

<table>
<thead>
<tr>
<th>Reverse proxy</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;VirtualHost *:80&gt;</code></td>
<td>Virtual host for HTTP</td>
</tr>
<tr>
<td>ServerName site.example.com</td>
<td>Define website name</td>
</tr>
<tr>
<td>RewriteEngine On</td>
<td>Redirect all HTTP requests to HTTPS</td>
</tr>
<tr>
<td>RewriteCond %{HTTPS} off</td>
<td>Alternatively:</td>
</tr>
<tr>
<td>RewriteRule &quot;.*&quot; https://%{HTTP_HOST}%{REQUEST_URI}</td>
<td>Redirect &quot;/&quot; &quot;<a href="https://10.2.2.73:443/">https://10.2.2.73:443/</a>&quot;</td>
</tr>
<tr>
<td><code>&lt;/VirtualHost&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>&lt;VirtualHost *:443&gt;</code></td>
<td>Virtual host for HTTPS</td>
</tr>
<tr>
<td>ServerName site.example.com</td>
<td>Define website name</td>
</tr>
<tr>
<td>ServerSignature On</td>
<td>Set a footer line under server-generated pages</td>
</tr>
<tr>
<td><code>&lt;/Proxy&gt;</code></td>
<td>Serve all proxy requests</td>
</tr>
<tr>
<td>Require all granted</td>
<td></td>
</tr>
<tr>
<td>SSLEngine on</td>
<td>Enable and configure SSL</td>
</tr>
<tr>
<td>SSLProtocol ALL -SSLv2 -SSLv3</td>
<td></td>
</tr>
<tr>
<td>SSLHonorCipherOrder on</td>
<td></td>
</tr>
<tr>
<td>SSLCipherSuite DEFAULT</td>
<td></td>
</tr>
<tr>
<td>SSLCertificateFile /etc/httpd/ssl/site.crt</td>
<td></td>
</tr>
<tr>
<td>SSLCertificateKeyFile /etc/httpd/ssl/site.key</td>
<td></td>
</tr>
<tr>
<td>SSLCACertificateFile /etc/httpd/ssl/site.ca.crt</td>
<td></td>
</tr>
<tr>
<td>ProxyPass &quot;.&quot; &quot;<a href="http://10.2.2.73:8080/">http://10.2.2.73:8080/</a>&quot;</td>
<td>Enable reverse proxying for server 10.2.2.73</td>
</tr>
<tr>
<td>ProxyPassReverse &quot;.&quot; &quot;<a href="http://10.2.2.73:8080/">http://10.2.2.73:8080/</a>&quot;</td>
<td></td>
</tr>
<tr>
<td><code>&lt;/VirtualHost&gt;</code></td>
<td></td>
</tr>
</tbody>
</table>
Tomcat

Apache Tomcat is an open source Java Servlet Container implementing several Java EE specifications, originally part of the Jakarta Project. It is composed of:
- Catalina, the core component and servlet container implementation;
- Coyote, an HTTP connector component, providing a pure Java webserver environment to run Java code;
- Jasper, a JSP (Java Server Pages) engine, which parses JSP files and compiles them into Java servlets.

Tomcat has been removed from RHEL 8; instead, it is suggested to use the JBoss Enterprise Application Platform, which includes Apache and Tomcat.

$JAVA_HOME Root of the Java installation e.g. /usr/lib/jvm/java-1.8.0-openjdk.x86_64/

$CATALINA_HOME Root of the Tomcat installation e.g. /usr/share/tomcat7/

$CATALINA_BASE Tomcat may also be configured for multiple instances by defining the variable $CATALINA_BASE for each instance. If a single instance of Tomcat is running, $CATALINA_BASE is the same as $CATALINA_HOME

<table>
<thead>
<tr>
<th>Tomcat global files</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CATALINA_BASE/conf/server.xml</td>
</tr>
<tr>
<td>$CATALINA_BASE/conf/web.xml</td>
</tr>
<tr>
<td>$CATALINA_BASE/conf/context.xml</td>
</tr>
<tr>
<td>$CATALINA_BASE/conf/tomcat-users.xml</td>
</tr>
<tr>
<td>$CATALINA_BASE/conf/catalina.policy</td>
</tr>
<tr>
<td>$CATALINA_BASE/conf/catalina.properties</td>
</tr>
<tr>
<td>$CATALINA_BASE/conf/logging.properties</td>
</tr>
<tr>
<td>$CATALINA_BASE/lib/</td>
</tr>
<tr>
<td>$JAVA_HOME/jre/lib/security/keystore.jks</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tomcat application-specific files</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CATALINA_BASE/webapps/appname/WEB-INF/</td>
</tr>
<tr>
<td>$CATALINA_BASE/webapps/appname/WEB-INF/web.xml</td>
</tr>
<tr>
<td>$CATALINA_BASE/webapps/appname/WEB-INF/classes/</td>
</tr>
<tr>
<td>$CATALINA_BASE/webapps/appname/WEB-INF/lib/</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tomcat log files</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CATALINA_BASE/logs/catalina.out</td>
</tr>
<tr>
<td>$CATALINA_BASE/logs/localhost.log</td>
</tr>
<tr>
<td>$CATALINA_BASE/logs/localhost_access.log</td>
</tr>
<tr>
<td>$CATALINA_BASE/logs/manager.log</td>
</tr>
<tr>
<td>$CATALINA_BASE/logs/host-manager.log</td>
</tr>
</tbody>
</table>

java -X Display all available -X options (nonstandard HotSpot JVM options)
java -XshowSettings:properties -version Print Java runtime settings
Samba is a free-software, cross-platform implementation of SMB/CIFS. SMB (Server Message Block) is Microsoft’s proprietary protocol for file and printer sharing, while CIFS (Common Internet File System) is the public version of SMB.

### Commonly used ports in Samba

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP/UDP 137</td>
<td>netbios-ns</td>
<td>NetBIOS name service requests and responses</td>
</tr>
<tr>
<td>TCP/UDP 138</td>
<td>netbios-dgm</td>
<td>NetBIOS datagram services e.g. server announcements</td>
</tr>
<tr>
<td>TCP/UDP 139</td>
<td>netbios-ssn</td>
<td>NetBIOS session service e.g. file and printer sharing</td>
</tr>
<tr>
<td>TCP 445</td>
<td>microsoft-ds</td>
<td>Active Directory; registration and translation of NetBIOS names, network browsing</td>
</tr>
<tr>
<td>TCP 389</td>
<td>LDAP</td>
<td></td>
</tr>
<tr>
<td>TCP 901</td>
<td>SWAT service</td>
<td></td>
</tr>
</tbody>
</table>

The full list of used ports can be found via the command `grep -i netbios /etc/services`.

### Samba Server

`smbd`  
Server Message Block daemon. Provides SMB file and printer sharing, browser services, user authentication, and resource lock. An extra copy of this daemon runs for each client connected to the server.

`nmbd`  
NetBIOS Name Service daemon. Handles NetBIOS name lookups, WINS requests, list browsing and elections. An extra copy of this daemon runs if Samba functions as a WINS server; another extra copy of this daemon runs if DNS is used to translate NetBIOS names.

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

### Configuration and Utilities

/etc/smb/  
Samba directory

/etc/samba/  
(RHEL 7)

/etc/samba/lmhosts  
Samba NetBIOS hosts file

/etc/samba/netlogon  
User logon directory

- `smbd -V`  
Show the version of the Samba server

- `smbclient -V`  
Check the Samba configuration file and report any error

- `smbpasswd user`  
Change the Samba password of `user`

- `smbpasswd -a user`  
Create a new Samba `user` and set his password

- `nmblookup smbservice`  
Look up the NetBIOS name of a server and map it to an IP address

- `nmblookup -U winsserver -R WORKGROUP#1B`  
Query recursively a WINS server for the Domain Master Browser for the specified workgroup

- `nmblookup -U winsserver -R WORKGROUP#1D`  
Query recursively a WINS server for the Domain Controller for the specified workgroup

- `net`  
Tool for administration of Samba and remote CIFS servers

- `net rpc shutdown -r -S smbserver -U root%password`  
Reboot a CIFS server

- `net rpc service list -S smbserver`  
List available services on a CIFS server

- `net status sessions`  
Show active Samba sessions

- `net status shares`  
Show Samba shares

- `net rpc info`  
Show information about the domain

- `net groupmap list`  
Show group mappings between Samba and Windows
Samba client

mount.cifs
smbmount

mount //smbserver/share1 /mnt/share1 -t cifs -o username=user

Mount a Samba share on a Linux filesystem, using the CIFS filesystem interface

Mount a Samba share as user

smbstatus

Display current information about shares, clients connections, and locked files

smbclient //smbserver/share1
smbclient -L //smbserver -W WORKGROUP -U user

Access a Samba share on a server (with an FTP-like interface)
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

<table>
<thead>
<tr>
<th>Samba mount options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username=user</td>
<td>Mount the share as user</td>
</tr>
<tr>
<td>password=password</td>
<td>Specify the mount user's password</td>
</tr>
<tr>
<td>credentials=credfile</td>
<td>Mount the share as the user defined in the credentials file credfile which must have this format: username=user password=password</td>
</tr>
<tr>
<td>multiuser</td>
<td>Mount the share in multiuser mode</td>
</tr>
<tr>
<td>sec=ntlmssp</td>
<td>Set the security level to NTLMSSP. This is required in RHEL 7 to enable multiuser mode</td>
</tr>
</tbody>
</table>
### Samba global configuration

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[global]</td>
<td>Global server settings: defines parameters applicable for the whole Samba server and sets the defaults</td>
</tr>
<tr>
<td>workgroup = MYWORKGROUP</td>
<td>Make Samba join the specified workgroup</td>
</tr>
<tr>
<td>server string = Linux Samba Server %L</td>
<td>Describe server to the clients</td>
</tr>
<tr>
<td>hosts allow = 10.9.9.0/255.255.255.0</td>
<td>Allow only the specified machines to connect to the server</td>
</tr>
<tr>
<td>security = user</td>
<td>Set up user-level authentication</td>
</tr>
<tr>
<td>encrypt passwords = yes</td>
<td>Use encrypted passwords</td>
</tr>
<tr>
<td>smb passwd file = /etc/samba/smbpasswd</td>
<td>Refer to the specified password file for user authentication.</td>
</tr>
<tr>
<td></td>
<td>A new user’s password will need to be set both in Linux and Samba by using these commands from shell prompt:</td>
</tr>
<tr>
<td></td>
<td>passwd newuser</td>
</tr>
<tr>
<td></td>
<td>smbpasswd newuser</td>
</tr>
<tr>
<td>unix password sync = yes</td>
<td>When the password of a client user (e.g. under Windows) is changed, change the Linux and Samba passwords accordingly</td>
</tr>
<tr>
<td>username map = /etc/samba/smbusers</td>
<td>Map each Samba server user name to client user name(s).</td>
</tr>
<tr>
<td></td>
<td>The file /etc/samba/smbusers has the following format:</td>
</tr>
<tr>
<td></td>
<td>root = Administrator Admin</td>
</tr>
<tr>
<td></td>
<td>jdoe = &quot;John Doe&quot;</td>
</tr>
<tr>
<td></td>
<td>kgreen = &quot;Kim Green&quot;</td>
</tr>
<tr>
<td>netbios name = Mysambabox</td>
<td>Set NetBIOS name and alias</td>
</tr>
<tr>
<td>netbios aliases = Mysambabox1</td>
<td>Make Samba play the role of a WINS server.</td>
</tr>
<tr>
<td>wins support = yes</td>
<td>Note: There should be only one WINS server on a network</td>
</tr>
<tr>
<td>logon server = yes</td>
<td>Enable logon support.</td>
</tr>
<tr>
<td>log file = /var/log/samba/log.%m</td>
<td>Logon script parameters will be defined in a [netlogon] section</td>
</tr>
<tr>
<td>max log size = 1000</td>
<td>Use a separate logfile for each machine that connects</td>
</tr>
<tr>
<td>syslog only = no</td>
<td>Maximum size of each logfile, in Kb</td>
</tr>
<tr>
<td>syslog = 0</td>
<td>Do not use only syslog to log</td>
</tr>
</tbody>
</table>
| panic action = \
| /usr/share/samba/panic-action %d                  | 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 |
|                             | Mail a backtrace to the sysadmin in case Samba crashes                                                  |
| [netlogon]                                                   | Section defining a logon script                                                                         |
| comment = Netlogon for Windows clients                      | Specifies a per-user script e.g. /home/netlogon/jdoe.bat will be called when user jdoe logs in.         |
| path = /home/netlogon                                      | It is also possible to specify a per-clientname script %m.bat, which will be called when a specific machine logs in. |
| logon script = %U.bat                                        | Guest access to the service (i.e. access without entering a password) is disabled                        |
| browseable = no                                             |                                                                                                         |
| writeable = no                                              |                                                                                                         |
| guest ok = no                                               |                                                                                                         |
| [Canon LaserJet 3]                                          | Section defining a printer accessible via the network                                                   |
| printer name = lp                                          |                                                                                                         |
| comment = Canon LaserJet 3 main printer                    |                                                                                                         |
| path = /var/spool/lpd/samba                                 |                                                                                                         |
| printable = yes                                            |                                                                                                         |
| writeable = no                                              |                                                                                                         |
**Samba share configuration**

<table>
<thead>
<tr>
<th>/etc/samba/smb.conf</th>
<th>Samba configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>[public]</td>
<td>Section defining a public share accessible on read/write by anyone</td>
</tr>
<tr>
<td>comment = Public Storage on %L</td>
<td>Describe the public share to users</td>
</tr>
<tr>
<td>path = /home/samba</td>
<td>Path of the public share on the server</td>
</tr>
<tr>
<td>browsable = yes</td>
<td>Show the public share when browsing</td>
</tr>
<tr>
<td>writeable = yes</td>
<td>Allow all users to write in this directory</td>
</tr>
<tr>
<td>[homes]</td>
<td>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 <em>path</em> variable is not set, by default is <em>path=/home/%U</em></td>
</tr>
<tr>
<td>comment = %U’s home directory on %L from %m</td>
<td>Describe the share to the user</td>
</tr>
<tr>
<td>browsable = no</td>
<td>Do not show the homes share when browsing</td>
</tr>
<tr>
<td>writeable = yes</td>
<td>Allow the user to write in his home directory</td>
</tr>
<tr>
<td>[foobar]</td>
<td>Section defining a specific share</td>
</tr>
<tr>
<td>path = /foobar</td>
<td>Path of the share on the server</td>
</tr>
<tr>
<td>comment = Share Foobar on %L from %m</td>
<td>Describe the share to users</td>
</tr>
<tr>
<td>browsable = yes</td>
<td>Show the share when browsing</td>
</tr>
<tr>
<td>writeable = yes</td>
<td>Allow the users to write in this share</td>
</tr>
<tr>
<td>valid users = jdoe, kgreen, +geeks</td>
<td>Allow access only to users “jdoe” and “kgreen”, and to local group “geeks”</td>
</tr>
<tr>
<td>invalid users = csmith</td>
<td>Deny access to user “csmith”</td>
</tr>
<tr>
<td>read list = bcameron</td>
<td>Allow read-only access to user “bcameron”</td>
</tr>
<tr>
<td>write list = fcastle</td>
<td>Allow read-write access to user “fcastle”</td>
</tr>
</tbody>
</table>
### Samba access configuration

#### /etc/samba/smb.conf

<table>
<thead>
<tr>
<th>Samba configuration</th>
<th>/etc/samba/smb.conf</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-level authentication</td>
<td></td>
</tr>
<tr>
<td>[global]</td>
<td></td>
</tr>
<tr>
<td>security = user</td>
<td>Set up user-level authentication</td>
</tr>
<tr>
<td>guest account = nobody</td>
<td>Map the guest account to the system user nobody (default)</td>
</tr>
<tr>
<td>map to guest = Never</td>
<td>Specify how incoming requests are mapped to the guest account:</td>
</tr>
<tr>
<td></td>
<td>Bad User</td>
</tr>
<tr>
<td></td>
<td>Bad Password</td>
</tr>
<tr>
<td></td>
<td>Never</td>
</tr>
<tr>
<td>Guest account</td>
<td>nobody</td>
</tr>
<tr>
<td>Map the guest account to the system user nobody (default)</td>
<td></td>
</tr>
<tr>
<td>Map to guest</td>
<td>Never</td>
</tr>
<tr>
<td>Specify how incoming requests are mapped to the guest account:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bad User</td>
</tr>
<tr>
<td></td>
<td>Bad Password</td>
</tr>
<tr>
<td></td>
<td>Never</td>
</tr>
</tbody>
</table>

#### Server-level authentication

<table>
<thead>
<tr>
<th>Samba configuration</th>
<th>/etc/samba/smb.conf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server-level authentication</td>
<td></td>
</tr>
<tr>
<td>[global]</td>
<td></td>
</tr>
<tr>
<td>security = server</td>
<td>Set up server-level authentication</td>
</tr>
<tr>
<td>password server = srv1 srv2</td>
<td>Authenticate to server srv1, or to server srv2 if the first one is unavailable</td>
</tr>
</tbody>
</table>

#### Domain-level authentication

<table>
<thead>
<tr>
<th>Samba configuration</th>
<th>/etc/samba/smb.conf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain-level authentication</td>
<td></td>
</tr>
<tr>
<td>[global]</td>
<td></td>
</tr>
<tr>
<td>security = ADS</td>
<td>Set up domain-level authentication as an Active Directory member server</td>
</tr>
</tbody>
</table>
| realm = KRB_REALM | Join the specified realm. Kerberos must be installed and an administrator account must be created: net ads join -U Administrator%

#### Share-level authentication

<table>
<thead>
<tr>
<th>Samba configuration</th>
<th>/etc/samba/smb.conf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Share-level authentication</td>
<td></td>
</tr>
<tr>
<td>[global]</td>
<td></td>
</tr>
<tr>
<td>security = share</td>
<td>Set up share-level authentication</td>
</tr>
<tr>
<td>[foobar]</td>
<td>Define a “foobar” share accessible to any user which can supply user's password. The user must be created on the system:</td>
</tr>
<tr>
<td>path = /foobar</td>
<td>useradd -c &quot;Foobar account&quot; -d /tmp -s /sbin/nologin user</td>
</tr>
<tr>
<td>username = user</td>
<td>and added to the Samba password file:</td>
</tr>
<tr>
<td>only user = yes</td>
<td>smbpasswd -a user</td>
</tr>
</tbody>
</table>

### Samba macros

<table>
<thead>
<tr>
<th>Macro</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%S</td>
<td>Username</td>
</tr>
<tr>
<td>%U</td>
<td>Session username (the username that the client requested, not necessarily the same as the one he got)</td>
</tr>
<tr>
<td>%G</td>
<td>Primary group of session username</td>
</tr>
<tr>
<td>%h</td>
<td>Samba server hostname</td>
</tr>
<tr>
<td>%m</td>
<td>Client hostname</td>
</tr>
<tr>
<td>%l</td>
<td>NetBIOS name of the server</td>
</tr>
<tr>
<td>%n</td>
<td>NetBIOS name of the client</td>
</tr>
<tr>
<td>%d</td>
<td>Process ID of the current server process</td>
</tr>
<tr>
<td>%a</td>
<td>Architecture of remote machine</td>
</tr>
<tr>
<td>%i</td>
<td>IP address of client machine</td>
</tr>
<tr>
<td>%l</td>
<td>Local IP address to which a client connected</td>
</tr>
<tr>
<td>%t</td>
<td>Current date and time</td>
</tr>
<tr>
<td>%d</td>
<td>Domain or workgroup of the current user</td>
</tr>
<tr>
<td>%w</td>
<td>Winbind separator</td>
</tr>
<tr>
<td>%($var)</td>
<td>Value of the environment variable var</td>
</tr>
<tr>
<td>%s</td>
<td>Name of the current service, if any</td>
</tr>
<tr>
<td>%p</td>
<td>Root directory of the current service, if any</td>
</tr>
<tr>
<td>%u</td>
<td>Username of the current service, if any</td>
</tr>
<tr>
<td>%g</td>
<td>Primary group name of username</td>
</tr>
<tr>
<td>%h</td>
<td>Home directory of username</td>
</tr>
<tr>
<td>%n</td>
<td>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</td>
</tr>
<tr>
<td>%p</td>
<td>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</td>
</tr>
</tbody>
</table>
Samba setup

This procedure allows sharing on read-write the local directory `/smbshare` on server 10.1.1.1 to client 10.2.2.2.

### Server setup:

1. Create the group for write access to the share
   
   `groupadd -r geeks`

2. Create the user and assign it to the group
   
   `useradd -G geeks jdoe`

3. Add the user to Samba.
   You will be prompted to enter a password
   
   `smbpasswd -a jdoe`

4. Assign correct ownership to the share
   
   `chgrp geeks /smbshare`

5. Set the SGID bit to the share
   
   `chmod 2775 /smbshare`

6. Set the correct SELinux label to the share
   
   `semanage fcontext -a -t samba_share_t '/smbshare'
   
   `restorecon -FR /smbshare`

7. Enable the SELinux boolean for write access to the share
   
   `setsebool -P samba_export_all_rw=on`

8. Add a section for the share on `/etc/samba/smb.conf`:
   
   ```
   [smbshare]
   path = /smbshare
   hosts allow = 10.2.2.2
   write list = @geeks
   ```

9. Ensure that the `smb` and `nmb` services are running

### Client setup:

1. Add an entry to `/etc/fstab` to mount the Samba share device automatically:
   
   ```
   //10.1.1.1/smbshare /mountpoint cifs username=jdoe,password=s3cr3t  0 0
   ```

### Client multiuser setup:

1. Add an entry to `/etc/fstab` to mount the Samba share device automatically in multiuser mode:
   
   ```
   //10.1.1.1/smbshare /mountpoint cifs username=jdoe,password=s3cr3t,multiuser,sec=ntlmssp  0 0
   ```

2. Login as another user (there must be a matching Samba user on the Samba server 10.1.1.1)

   ```
   su - ksmith
   ```

3. Store the Samba username and password in the kernel keyring for the current session

   ```
   cifscreds add 10.1.1.1
   ```
A Network File System (NFS) server makes filesystems available to remote clients for mounting.

NFS requires the portmapper to map incoming TCP/IP connections to the appropriate NFS RPC calls. Some Linux distributions use rpcbind instead of the portmapper.

For security reasons, the TCP Wrapper should be configured to limit access to the portmapper to NFS clients only:

file /etc/hosts.deny should contain portmap: ALL
file /etc/hosts.allow should contain portmap: IP_addresses_of_clients

NFS handles user permissions across systems by considering users with same UID and username as the same user. Group permission is evaluated similarly, by GID and groupname.

rpc.nfsd
rpc.mountd
rpc.lockd
rpc.statd

/etc/exports
/var/lib/nfs/xtab
/etc/exports

Export or reexport all directories.
When exporting, fills the kernel export table /proc/fs/nfs/exports.
When reexporting, removes the entries in /var/lib/nfs/xtab that are deleted from /etc/exports (therefore synchronizing the two files), and removes the entries from /proc/fs/nfs/exports that are no longer valid

Export or reexport all directories.
When exporting, fills the kernel export table /proc/fs/nfs/exports.
When reexporting, removes the entries in /var/lib/nfs/xtab that are deleted from /etc/exports (therefore synchronizing the two files), and removes the entries from /proc/fs/nfs/exports that are no longer valid

showmount
showmount --directories
showmount --exports
showmount --all
showmount -e nfsserver

Show the remote client hosts currently having active mounts
Show the directories currently mounted by a remote client host
Show the filesystems currently exported i.e. the active export list
Show both remote client hosts and directories
Show the shares a NFS server has available for mounting

rpcinfo -p
rpcinfo -t nfsserver
rpcinfo -u nfsserver nfs

Probe the portmapper on a NFS server and display the list of all registered RPC services there
Test a NFS connection by sending a null pseudo request (using TCP)
Test a NFS connection by sending a null pseudo request (using UDP)

nfsstat

Display NFS/RPC client/server statistics.

<table>
<thead>
<tr>
<th>Options:</th>
<th>NFS</th>
<th>RPC</th>
<th>both</th>
</tr>
</thead>
<tbody>
<tr>
<td>server</td>
<td>-sn</td>
<td>-sr</td>
<td>-s</td>
</tr>
<tr>
<td>client</td>
<td>-cn</td>
<td>-cr</td>
<td>-c</td>
</tr>
<tr>
<td>both</td>
<td>-n</td>
<td>-r</td>
<td>-nr</td>
</tr>
</tbody>
</table>

mount -t nfs nfsserver:/share /usr

Command to be run on a client to mount locally a remote NFS share.
NFS shares accessed frequently should be added to /etc/fstab e.g.

nfsserver:/share /usr nfs intr 0 0
### /etc/exports

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>Filesystem on the NFS server to be exported to clients</th>
</tr>
</thead>
<tbody>
<tr>
<td>/export/</td>
<td>10.3.3.3(rw)</td>
</tr>
<tr>
<td>/export2/</td>
<td>10.4.4.0/24</td>
</tr>
<tr>
<td>/export3/</td>
<td>*(ro,sync)</td>
</tr>
<tr>
<td>/home/ftp/pub</td>
<td>myhost(rw) <em>.</em>.example.org(ro)</td>
</tr>
<tr>
<td>/home/crew</td>
<td>@FOOWORKGROUP(rw) *(ro)</td>
</tr>
</tbody>
</table>

### Client Identity

Client systems permitted to access the exported directory. Can be specified by hostname, IP address, wildcard, subnet, or @NIS workgroup. Multiple client systems can be listed, and each one can have different options.

<table>
<thead>
<tr>
<th>Client Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ro</td>
<td>Read-only access (default)</td>
</tr>
<tr>
<td>rw</td>
<td>Read and write access. The client may choose to mount read-only anyway</td>
</tr>
<tr>
<td>sync</td>
<td>Reply to requests only after the changes made by these requests have been committed to stable storage</td>
</tr>
<tr>
<td>async</td>
<td>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</td>
</tr>
<tr>
<td>root_squash</td>
<td>Requests by user root on client will be done as user nobody on server (default)</td>
</tr>
<tr>
<td>no_root_squash</td>
<td>Requests by user root on client will be done as same user root on server</td>
</tr>
<tr>
<td>all_squash</td>
<td>Requests by a non-root user on client will be done as user nobody on server</td>
</tr>
<tr>
<td>no_all_squash</td>
<td>Requests by a non-root user on client will be attempted as same user on server (default)</td>
</tr>
</tbody>
</table>

### NFS Mount Options

<table>
<thead>
<tr>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>rsize=nnn</td>
</tr>
<tr>
<td>wsize=nnn</td>
</tr>
<tr>
<td>nfsvers=n</td>
</tr>
<tr>
<td>retry=n</td>
</tr>
<tr>
<td>timeo=n</td>
</tr>
<tr>
<td>intr</td>
</tr>
<tr>
<td>nointr</td>
</tr>
<tr>
<td>hard</td>
</tr>
<tr>
<td>soft</td>
</tr>
<tr>
<td>bg</td>
</tr>
<tr>
<td>fg</td>
</tr>
<tr>
<td>tcp</td>
</tr>
<tr>
<td>udp</td>
</tr>
<tr>
<td>sec=krb5p</td>
</tr>
<tr>
<td>v4.2</td>
</tr>
</tbody>
</table>
## NFS setup

This procedure allows sharing on read-write the local directory `/nfsshare` on server 10.1.1.1 to client 10.2.2.2.

### Server setup:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Ensure that the <code>nfs-server</code> service is running</td>
</tr>
<tr>
<td>2.</td>
<td>Change ownership of the share <code>chown nfsnobody /nfsshare</code></td>
</tr>
</tbody>
</table>
| 3.   | Add an entry for the share on `/etc/exports`:
      | `/nfsshare  10.2.2.2(rw)` |
| 4.   | Reload the exports file `exportfs -r` |

### Client setup:

1. Add an entry to `/etc/fstab` to mount the NFS share device automatically:

   ```
   10.1.1.1:/nfsshare  /mountpoint  nfs  defaults  0 0
   ```

## Secure NFS setup

This procedure allows sharing on read-write the local directory `/nfsshare` on server 10.1.1.1 to client 10.2.2.2, securely with Kerberos enabled.

### Server setup:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Install the appropriate server keytab on <code>/etc/krb5.keytab</code></td>
</tr>
<tr>
<td>2.</td>
<td>Ensure that the <code>nfs-secure-server</code> service is running</td>
</tr>
<tr>
<td>3.</td>
<td>Change ownership of the share <code>chown nfsnobody /nfsshare</code></td>
</tr>
</tbody>
</table>
| 4.   | Add an entry for the share on `/etc/exports`:
      | `/nfsshare  10.2.2.2(sec=krb5p,rw)` |
| 5.   | Reload the exports file `exportfs -r` |

### Client setup:

1. Install the appropriate client keytab on `/etc/krb5.keytab` |
2. Ensure that the `nfs-secure` service is running |
3. Add an entry to `/etc/fstab` to mount the NFS share device automatically:

   ```
   10.1.1.1:/nfsshare  /mountpoint  nfs  defaults,sec=krb5p  0 0
   ```
iSCSI (Internet Small Computer System Interface) is a network protocol that allows emulating an SCSI local storage device over a TCP/IP network. By default it uses TCP port 3260.
An iSCSI server can use a local block device (physical or virtual disk, disk partition, or Logical Volume), a file, a physical SCSI device, or a ramdisk as the underlying storage resource (backstore) and make it available by assigning it a LUN (Logical Unit Number). An iSCSI server provides one or more targets, each of which presents one or more LUNs and is able to accept connections from an iSCSI client (initiator).
Targets and initiators are called nodes and are identified by a unique IQN (iSCSI Qualified Name) e.g. iqn.2017-11.org.example.subdomain:foo:bar. The IP address and port of a node is called a portal.
A target accepts connections from an initiator via a TPG (Target Portal Group) i.e. its IP address and port. A TPG may have in place an ACL so to accept connections only from a specific initiator's IQN.

**targetcli**  
Target configurator (server side). Can be used as a command line tool or as an interactive shell.  
Configuration is saved to /etc/target/saveconfig.json

**iscsiadm**  
Administration tool for iSCSI devices (client side)
This procedure makes available the local disk /dev/sbd on server 10.1.1.1 to the client having IQN iqn.2017-11.org.example:client.

### Server (target) setup:
1. Ensure that the targetcli service is running
   ```bash
   targetcli
   ```
2. Enter the targetcli shell
   ```bash
cd /backstores/block
create mydisk /dev/sdb
   ```
3. Create a backstore
   ```bash
cd /iscsi/create iqn.2017-11.org.example:target
   ```
4. Create a IQN for the target. This automatically creates a TPG for the IQN
   ```bash
cd /iscsi/iqn.2017-11.org.example:target/tpg1/acls
create iqn.2017-11.org.example:client
   ```
5. On the TPG, create an ACL to allow connections from the initiator with a specific IQN
   ```bash
cd /iscsi/iqn.2017-11.org.example:target/tpg1/luns
create /backstores/block/mydisk
   ```
6. On the TPG, create a portal listening from the server's IP address
   ```bash
cd /iscsi/iqn.2017-11.org.example:target/tpg1/portals
delete 0.0.0.0 ip_port=3260
create 10.1.1.1
   ```
7. Verify the configuration
   ```bash
ls /
   ```
8. Exit the targetcli shell.
   ```bash
exit
   ```

### Client (initiator) setup:
1. Set the correct initiator IQN in the file /etc/iscsi/initiatorname.iscsi:
   ```bash
   InitiatorName=iqn.2017-11.org.example:client
   ```
2. Ensure that the iscsi service is running
3. Discover the iSCSI target(s) provided by the portal. This echoes the target(s) IQN found
   ```bash
   iscsiadm -m discovery -t sendtargets -p 10.1.1.1
   ```
4. Login to the target IQN found
   ```bash
   iscsiadm -m node -T iqn.2017-11.org.example:target -p 10.1.1.1 -l
   ```
   The iSCSI device is now locally available and can be formatted and mounted. Node records remain after logout or reboot; the system will login again to the target IQN automatically
5. Add an entry to /etc/Fstab to mount the iSCSI device automatically:
   ```bash
   UUID=nnnnnnnn-nnnn-nnnn-nnnn-nnnnnnnnnnn /mountpoint fstype _netdev 0 0
   ```
DHCP (Dynamic Host Configuration Protocol) is a protocol for network management that automatically provides a requesting host with an IP address and other network configuration parameters. It is based on BOOTP (Bootstrap Protocol). A DHCP server listens for requests on UDP port 67 and answers to UDP port 68. The assignment of an IP address to a host is done through a sequence of DHCP messages initiated by the client host: DHCP Discover, DHCP Offer, DHCP Request, and finally DHCP Acknowledgment.

Because DHCP Discover messages are broadcast and therefore not routed outside a LAN, a DHCP relay agent is necessary for those clients situated outside the DHCP server’s LAN. The DHCP relay agent listens to DHCP Discover messages and relays them in unicast to the DHCP server.

<table>
<thead>
<tr>
<th>/etc/dhcpd.conf</th>
<th>Configuration file for the DHCP server</th>
</tr>
</thead>
<tbody>
<tr>
<td>/etc/sysconfig/dhcrelay (SUSE)</td>
<td>Configuration file for the DHCP relay agent</td>
</tr>
<tr>
<td>/var/lib/dhcpd/dhcpd.leases</td>
<td>DHCP current leases</td>
</tr>
</tbody>
</table>

### /etc/dhcpd.conf

<table>
<thead>
<tr>
<th>Option Description</th>
<th>Value Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Option domain-name-servers 10.2.2.2;</td>
<td></td>
</tr>
<tr>
<td>Option smtp-servers 10.3.3.3;</td>
<td></td>
</tr>
<tr>
<td>Option pop-servers 10.4.4.4;</td>
<td></td>
</tr>
<tr>
<td>Option time-servers 10.5.5.5;</td>
<td></td>
</tr>
<tr>
<td>Option nntp-servers 10.6.6.6;</td>
<td></td>
</tr>
<tr>
<td>Shared-network geek-net {</td>
<td></td>
</tr>
<tr>
<td>Default-lease-time 86400;</td>
<td></td>
</tr>
<tr>
<td>Max-lease-time 172800;</td>
<td></td>
</tr>
<tr>
<td>Option routers 10.0.3.252;</td>
<td></td>
</tr>
<tr>
<td>Option broadcast-address 10.0.3.255;</td>
<td></td>
</tr>
<tr>
<td>Subnet 10.0.3.0 netmask 255.255.255.128 {</td>
<td></td>
</tr>
<tr>
<td>Range 10.0.3.1 10.0.3.101;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>Subnet 10.0.3.128 netmask 255.255.255.128 {</td>
<td></td>
</tr>
<tr>
<td>Range 10.0.3.129 10.0.3.229;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>Group {</td>
<td></td>
</tr>
<tr>
<td>Option routers 10.0.17.252;</td>
<td></td>
</tr>
<tr>
<td>Option broadcast-address 10.0.17.255;</td>
<td></td>
</tr>
<tr>
<td>Netmask 255.255.255.0;</td>
<td></td>
</tr>
<tr>
<td>Host linuxbox1 {</td>
<td></td>
</tr>
<tr>
<td>Hardware ethernet AA:BB:CC:DD:EE:FF;</td>
<td></td>
</tr>
<tr>
<td>Fixed-address 10.0.17.42;</td>
<td></td>
</tr>
<tr>
<td>Option host-name &quot;linuxbox1&quot;;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>Host linuxbox2 {</td>
<td></td>
</tr>
<tr>
<td>Fixed-address 10.0.17.66;</td>
<td></td>
</tr>
<tr>
<td>Option host-name &quot;linuxbox2&quot;;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
</tbody>
</table>

### Global parameters for DNS, mail, NTP, and news servers specification

- 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;

### Definition of a network
- Time, in seconds, that will be assigned to a lease if a client does not ask for a specific expiration time
- Maximum time, in seconds, that can be assigned to a lease if a client asks for a specific expiration time

### 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

### Definition of a group
- 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

<table>
<thead>
<tr>
<th>/etc/pam.d/service</th>
</tr>
</thead>
<tbody>
<tr>
<td>auth    requisite    pam_securetty.so</td>
</tr>
<tr>
<td>auth    required    pam_nologin.so</td>
</tr>
<tr>
<td>auth    required    pam_env.so</td>
</tr>
<tr>
<td>auth    required    pam_unix.so nullok</td>
</tr>
<tr>
<td>account required    pam_unix.so</td>
</tr>
<tr>
<td>session required    pam_unix.so</td>
</tr>
<tr>
<td>session optional    pam_lastlog.so</td>
</tr>
<tr>
<td>password required    pam_unix.so nullok obscure min=4 max=8</td>
</tr>
</tbody>
</table>

- **type**
  - **auth**: Authentication module to verify user identity and group membership
  - **account**: Authorization module to determine user's right to access a resource (other than his identity)
  - **password**: Module to update a user's authentication credentials
  - **session**: Module (run at end and beginning of a user session) to set up the user environment

- **control**
  - **optional**: Module is not critical to the success or failure of service
  - **sufficient**: If this module successes, and no previous module has failed, module stack processing ends successfully. If this module fails, it is non-fatal and processing of the stack continues
  - **required**: If this module fails, processing of the stack continues until the end, and service fails
  - **requisite**: If this module fails, service fails and control returns to the application that invoked service
  - **include**: Include modules from another PAM service file

<table>
<thead>
<tr>
<th>module</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAM module and its options, e.g.:</td>
</tr>
<tr>
<td>pam_unix.so  Standard UNIX authentication module via /etc/passwd and /etc/shadow</td>
</tr>
<tr>
<td>pam_nis.so  Module for authentication via NIS</td>
</tr>
<tr>
<td>pam_ldap.so  Module for authentication via LDAP</td>
</tr>
<tr>
<td>pam_fshadow.so  Module for authentication against an alternative shadow passwords file</td>
</tr>
<tr>
<td>pam_cracklib.so  Module for password strength policies (e.g. length, case, max number of retries)</td>
</tr>
<tr>
<td>pam_limits.so  Module for system policies and system resource usage limits</td>
</tr>
<tr>
<td>pam_listfile.so  Module to deny or allow the service based on an arbitrary text file</td>
</tr>
</tbody>
</table>
LDAP (Lightweight Directory Access Protocol) is a simplified version of the X.500 standard and uses TCP port 389. LDAP allows to organize hierarchically a database of entries, each one of which is identified by a unique DN (Distinguished Name). Each DN has a set of attributes, and each attribute has a value; an attribute may appear multiple times. Special attributes called objectClass define which attributes are allowed and which are required, and determine the schema of the LDAP.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Attribute with value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>cn</td>
<td>cn: John Doe</td>
<td>Common Name</td>
</tr>
<tr>
<td>dc</td>
<td>dc=example,dc=org</td>
<td>Domain Component</td>
</tr>
<tr>
<td>givenName</td>
<td>givenName: John</td>
<td>First name</td>
</tr>
<tr>
<td>sn</td>
<td>sn: Doe</td>
<td>Surname</td>
</tr>
<tr>
<td>mail</td>
<td>mail: <a href="mailto:jdoe@example.org">jdoe@example.org</a></td>
<td>Email address</td>
</tr>
<tr>
<td>telephoneNumber</td>
<td>telephoneNumber: +1 555 1234 567</td>
<td>Telephone number</td>
</tr>
<tr>
<td>uid</td>
<td>uid: jdoe</td>
<td>User ID</td>
</tr>
<tr>
<td>c</td>
<td>c: US</td>
<td>Country code</td>
</tr>
<tr>
<td>l</td>
<td>l: San Francisco</td>
<td>Locality</td>
</tr>
<tr>
<td>st</td>
<td>st: California</td>
<td>State or province</td>
</tr>
<tr>
<td>street</td>
<td>street: 42, Penguin Road</td>
<td>Street</td>
</tr>
<tr>
<td>o</td>
<td>o: The Example Foundation</td>
<td>Organization</td>
</tr>
<tr>
<td>ou</td>
<td>ou: IT Dept</td>
<td>Organizational Unit</td>
</tr>
<tr>
<td>manager</td>
<td>manager: cn=Kim Green,dc=example,dc=org</td>
<td>Manager</td>
</tr>
</tbody>
</table>

**LDIF (LDAP Data Interchange Format)**

```
dn: cn=John Doe, dc=example, dc=org
changetype: modify
replace: mail
mail: johndoe@otherexample.com
  add: jpegPhoto
  jpegPhoto:< file://tmp/jdoe.jpg
delete: description
```

This LDIF file will change the email address of user "jdoe", add a picture, and delete the description attribute for the entry.
All the LDAP commands below accept the following arguments, plus some extra arguments which are command-dependent.

```
-H ldap://srv
-H ldapi://
-D binddn
-w password
-W
-x
-v
```

**ldapsearch**

Query a LDAP server and return the output in LDIF

```
-ln base
-z n
-LLL
```

**filter**

Search filter. If not specified, uses the default filter `(objectClass=*)`

**attributes**

Attributes to return. If not specified, returns all attributes

**ldapmodify**

Modify a LDAP entry

```
-f file.ldif
```

**ldapadd**

Add a LDAP entry

```
-h ldap.example.org 
-D "cn=Admin,dc=example,dc=org" -W -f file.ldif
```

**ldapdelete**

Delete a LDAP entry

```
-h ldap.example.org 
-D "cn=Admin,dc=example,dc=org" -W 
"uid=jdoe,dc=example,dc=org"
```

**ldappasswd**

Change the password of a LDAP entry

```
-s password
-S
```

**Examples**

```
ldapsearch -H ldap://ldap.example.org -s base \\ -b "ou=people,dc=example,dc=com" "(sn=Doe)" \\ cn sn telephoneNumber
```

Query a LDAP server for entries in the OU "people" whose surname is "Doe"; print common name, surname, and telephone number of the entries found

```
ldapmodify -b -r -f file.ldif
```

Modify an entry according to the LDIF file specified

```
ldapadd -h ldap.example.org \\ -D "cn=Admin,dc=example,dc=org" -W -f file.ldif
```

Authenticating as "Admin", add an entry by adding the content of the specified LDIF file to the directory

```
ldapdelete -h ldap.example.org \\ -D "cn=Admin,dc=example,dc=org" -W \\ "uid=jdoe,dc=example,dc=org"
```

Authenticating as "Admin", delete the user "jdoe"

```
ldappasswd -h ldap.example.org \\ -D "cn=Admin,dc=example,dc=org" -W -x \\ -S "uid=jdoe,ou=IT Dept,dc=example,dc=org"
```

Authenticating as "Admin" on example.org, change the password of user "jdoe" in the OU "IT Dept"
OpenLDAP is an open source implementation of LDAP, and was initially developed together with the LDAP protocol. Its related service is *slapd*, the Standalone OpenLDAP daemon. *sssd*, the System Security Services Daemon, can be used to provide access to OpenLDAP as an authentication and identity provider.

```
/var/lib/ldap/
/etc/openldap/slapd.conf
/usr/local/etc/openldap/slapd.conf
/usr/local/etc/openldap/slapd.d/
```

Files constituting the OpenLDAP database

```
slapcat -b cn=config
ldapsearch -Y EXTERNAL -H ldapi:/// -b cn=config
```

Show the OpenLDAP configuration

```
slaptest -u
```

Verify that the OpenLDAP configuration is correct

```
slapcat -l file.ldif
```

Dump the contents of an OpenLDAP database to an LDIF file

```
slapadd -l file.ldif
```

Import an OpenLDAP database from an LDIF file

```
slapindex
```

Regenerate OpenLDAP's database indexes

```
yum install openldap openldap-clients authconfig
   sssd nss-pam-ldapd authconfig-gtk (RHEL 7)
```

Install the OpenLDAP client

```
authconfig --enableldap --enableldapauth
   --ldapserv=ldaps://ldapserver
   --ldapbasedn="dc=example,dc=org"
   --enablesssd --update (RHEL 7)
authselect select sssd --force (RHEL 8)
authconfig-gtk
system-config-authentication
```

Set up the LDAP client to connect to a *ldapserver*. This will update the configuration files `/etc/sssd/sssd.conf` and `/etc/openldap/ldap.conf`

Set up LDAP client authentication via *sssd*

```
getent group groupname
```

Get entries about *groupname* from NSS libraries
Security-Enhanced Linux (SELinux) is a Linux kernel security module that provides a mechanism for supporting access control security policies.

SELinux implements a Mandatory Access Control framework that allows the definition of fine-grained permissions for how subjects (i.e. processes) access objects (i.e. other processes, files, devices, ports, sockets); this improves security with respect to the traditional Discretionary Access Control, which defines accesses based on users and groups. Processes, files, and users have a security context structured as user:role:type:level e.g. unconfined_u:object_r:user_home_t:s0. The third field defines a type for files or a domain for processes. The decisions SELinux takes about allowing or disallowing access are stored in the AVC (Access Vector Cache).

SELinux creates a pseudo filesystem (SELinuxfs) containing commands used by the kernel for its operations; this filesystem is usually mounted on /selinux/ or /sys/fs/selinux/.

```bash
setenforce 0
echo 0 > /selinuxfs/enforce
```
Enter permissive mode (SELinux must be enabled)

```bash
setenforce 1
echo 1 > /selinuxfs/enforce
```
Enter enforcing mode (SELinux must be enabled)

```bash
getenforce
cat /selinuxfs/enforce
```
Display current mode

```bash
sestatus -v
```
Show SELinux mode, SELinuxfs mount point, etc.

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

<table>
<thead>
<tr>
<th>mode</th>
<th>SELINUX=</th>
<th>SELinux fully enforces security policies</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>enforcing</td>
<td>SELinux does not enforce security policies, but logs violations</td>
</tr>
<tr>
<td></td>
<td>permissive</td>
<td>SELinux security policies are disabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>policy</th>
<th>SELINUXTYPE=</th>
<th>SELinux protects targeted daemons</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>targeted</td>
<td>(up to RHEL 6) SELinux fully protects the system</td>
</tr>
<tr>
<td></td>
<td>strict</td>
<td>(RHEL 7 and later) SELinux only protects selected processes</td>
</tr>
<tr>
<td></td>
<td>minimum</td>
<td>(RHEL 7 and later) Multi Level Security protection</td>
</tr>
<tr>
<td></td>
<td>mls</td>
<td></td>
</tr>
</tbody>
</table>

```bash
ls -Z
```
List files and their security context. The security context of a file is stored in its extended attributes

```bash
ps -eZ
```
List processes and their security context

```bash
tar --selinux otherargs
star -xattr -H=exustar otherargs
```
Create or extract archives that retain the security context of the original files
chcon context file
chcon --reference=file0 file

restorecon -f file

getsebool boolean
setsebool boolean=value

semanage

semanage fcontext -l
semanage fcontext -a -t label file

semanage login -l

semanage port -l
semanage port -a -t portlabel -p tcp n
semanage port -a -t http_port_t -p tcp 8888
semanage port -d -t http_port_t -p tcp 8888
semanage port -m -t http_cache_port_t -p tcp 8888

semanage permissive -a auditd_t

semanage permissive -d auditd_t
semanage permissive -l

sepolicy
sepolicy manpage -a -p /usr/local/man/man8 && mandb

seinfo

Change the security context of file to the specified context
Change the security context of file to be the same as file0
Restore the security context of file to the system default
Get the value of a SELinux boolean
Set the value of a SELinux boolean
Manage SELinux policies
List files and their assigned SELinux labels
Assign the SELinux label to file. It is then necessary to apply the label via restorecon -f file
List mappings between users and SELinux users
List port numbers and their assigned SELinux type definitions
Assign the SELinux portlabel to TCP port n
Allow a local webserver to serve content on port 8888
Remove the binding of http_port_t port label to TCP 8888
Modify the port label bound to TCP 8888
Add auditd_t to the list of permissive types/domains. In this case, SELinux allows the auditd daemon all access while logging its AVC violations
Delete auditd_t from the list of permissive types/domains
List all permissive types/domains
Inspect a SELinux policy
Generate all SELinux policy manpages
Query the components of a SELinux policy
sealert -a logfile

```
sealert -a logfile
```

Analyze a SELinux logfile and display verbosely SELinux policy violations.

SELinux violation events are logged as

type=AVC msg=audit(timest.amp:id): avc: denied (...)

```
grep timest.amp:id logfile | audit2why
```

Diagnostic a specific AVC denial event entry (identified by a timestamp and an id) from a SELinux logfile

```
audit2allow -i inputfile -M module
```

Generate a loadable module containing the appropriate SELinux policy from a denied operation stored in inputfile

```
ausearch -c '(exe)' --raw | audit2allow -M module
```

Generate a loadable module to allow access on an executable which caused an AVC violation

```
semodule -l
```

List installed SELinux policy modules

```
semodule -X n -i module.pp
```

Install a SELinux policy module at priority n.

Installed modules are not removed after reboot. Module files have usually the suffix .pp (policy package)

```
semodule -X n -r module
```

Remove a SELinux policy module at priority n.

Modules must be removed at the same priority at which they were installed
Kickstart is a method to perform automatic installation and configuration of RHEL machines. This can be done by specifying `inst.ks=hd:/dev/sda:/root/path/ksfile` either as a boot option, or an option to the kernel command in GRUB 2.

- `/root/anaconda-ks.cfg`: Kickstart file describing the current system. This file is automatically generated during the installation.
- `system-config-kickstart`: GUI tool to create a Kickstart file.
- `ksvalidator ksfile`: Check the validity of a Kickstart file.
- `ksverdiff -f RHEL6 -t RHEL7`: Show the differences in the Kickstart syntax between RHEL 6 and RHEL 7.
Red Hat Satellite is a system management software that allows provisioning and configuration of RHEL machines. Repository content is provided via Red Hat Subscription Management (RHSM). Satellite 5 was based on Spacewalk, an open source system management software for Linux machines. Satellite 6 is a complete overhaul of it and is composed of:

- **Foreman**, an open source lifecycle management tool able to provision servers via Kickstart and Puppet;
- **Katello**, a tool that handles Red Hat repository management (via the Pulp service) and subscription management (via the Candlepin service).

All these components above need a PostgreSQL database, except Pulp which needs a MongoDB database.

As a separate component, **Capsule** servers act as a proxy for many of the main Satellite functions e.g. repository storage. A Capsule is also integrated in each Satellite server.

```
subscription-manager register
subscription-manager attach
foreman-maintain service list
foreman-maintain service status
foreman-maintain service start
foreman-maintain service stop
foreman-maintain service restart
foreman-maintain backup
foreman-rake command:option
hammer
pulp-admin-client
virt-who
foreman-debug
sosreport
citellus.py sosreportfile
```

- Register a system to the RHSM portal
- Attach a RHSM subscription to a registered system
- List all Satellite services
- Display status or start, stop, restart all Satellite services. Performed via systemctl
- Make a backup of Satellite
- Perform various administrative tasks
- CLI tool for Foreman
- Tool to administer the Pulp server
- Agent for reporting virtual guest IDs and hypervisors to a Satellite server
- Collect Satellite configuration, log, and backend data for debug purposes
- Collect diagnostic and configuration data for technical support
- Perform some automated checks for troubleshooting a system
KVM (Kernel-based Virtual Machine) is a virtualization infrastructure for the Linux kernel that allows it to function as a hypervisor.

/etc/libvirt/qemu/  
Directory containing the XML files that define VMs properties.  

/var/lib/libvirt/  
Directory containing files related to the VMs

virt-manager  
KVM GUI

virt-install --prompt  
Interactive command-line program to create a VM  
Create a VM with 2 Gb of RAM, specifying path of virtual disk, location of installation files, and (as extra argument) the Kickstart configuration to use

virt-install -n vmname -r 2048 \  
--disk path=/var/lib/libvirt/images/vmname.img \  
-1 /root/vmstuff/inst/ \  
-x "ks=/root/vmstuff/kickstart.cfg"

virt-clone --prompt  
Interactive command-line program to clone a VM.  
A VM must be shut off or paused before it can be cloned  
Clone a VM

virt-clone -o vmname -n vmclonename

virsh  
Interface for VM management

virsh list --all  
List all VMs present on the system  

virsh start vmname  
Start a VM  

virsh destroy vmname  
Brutally shut down a VM  

virsh shutdown vmname  
Gracefully shut down a VM  

virsh autostart vmname  
Set a VM to be automatically started when the system boots.  
Done by symlinking the VM to /etc/libvirt/qemu/autostart/  

virsh autostart --disable vmname  
Disable the autostart of a VM at system boot  

virsh edit vmname  
Edit the XML file defining a VM's properties  

virsh-what  
Detect whether the current machine is a VM
Git is an open source version control system with a small footprint and very high performances. A Git directory is a complete repository with full history and version tracking abilities, independent of any remote repository. Git commits are identified by a 40-hex-digits hash number, usually shortened to 7 digits, or even less if unambiguous.

```
git init
```
Initialize the current directory as a repository

```
git clone repo
```
Clone a remote repository.
`repo` can be an URL (SSH, HTTP, HTTPS, FTP, FTPS, Git) or a local path e.g.

```
ssh://user@example.com:8888/path/to/repo.git
```

```
ssh://example.com:9999/path/to/repo.git
```

```
/git://example.com:9999/path/to/repo.git
```
```
/path/to/repo.git
```

```
git checkout branch
```
Start working into an already existing `branch`

```
git checkout -B branch
```
Create `branch` and start working into it

```
git checkout -- file
```
Discard local changes done to `file`

```
git checkout branch file
```
Copy `file` from `branch` to the current branch, and add it to the staging area

```
git pull
```
Pull the changes from the remote repository branch to the local branch

```
git add file
```
Add `file` to the staging area (i.e. content staged for the next commit), hence starting to track it

```
git add .
```
Add all modified files to the staging area

```
git rm file
```
Remove `file` from the content staged for the next commit

```
git status
```
See the status (e.g. files changed but not yet staged) of the current branch

```
git commit -m "Message"
```
Commit all staged files in the current branch

```
git commit -am "Message"
```
Add all changed files to the staging area in the current branch, and commit them

```
git merge branch
```
Merge changes made on `branch` to the master branch

```
git push
```
Push the local commits from the current branch to the remote repository

```
git push origin branch
```
Push the local commits from `branch` to the remote repository

```
git revert commit
```
Revert a specific commit

```
git branch
```
Show local branches

```
git branch -r
```
Show remote branches

```
git branch -a
```
Show remote and local branches

```
git branch -a --contains commit
```
Show on which branch was done a specific commit number

```
git branch -d branch
```
Delete a local branch (which must have been merged in its upstream branch)

```
git branch -D branch
```
Delete a local branch (irrespective of its merged status)
git diff commit1 commit2
Show the differences between two commits

git diff branch1 branch2
Show the differences between two branches

git diff branch1 branch2 file
Show the differences between two branches for a specific file

git log --all -- file
Show the commits which involved file, across all branches

git log -p --all -S 'string'
Show the commits whose added or deleted lines contain a specific word

git grep string 'git show-ref --heads'
Search for string across all branches' heads (i.e. in the latest content only, and not in all the previous commits)

git config --list
Get all currently set options and their values in the Git configuration

Get the value of option

git config user.name name
Set your username

git config user.email email
Set your email address
Vagrant is an open source software that allows building and maintaining lightweight and portable virtual environments for software development. It relies on an underlying virtualization solution e.g. VirtualBox.

```
vagrant -h
vagrant command -h
vagrant init hashicorp/precise64
vagrant up vmname
vagrant provision vmname
vagrant ssh vmname
vagrant halt vmname
vagrant destroy vmname
vagrant status
vagrant global-status
vagrant global-status --prune
```

Print the list of commands recognized by Vagrant

Print help about the Vagrant command

Initialize the current directory as a specific Vagrant environment (in this case, Ubuntu 12.04 64-bit) by creating a Vagrantfile on it

Start a guest virtual machine and do a first provisioning according to the Vagrantfile

Provision a virtual machine

Connect via SSH to a virtual machine

Shut down the virtual machine

Delete the virtual machine and free any resource allocated to it

Print the status of the virtual machines currently managed by Vagrant

Print the status of all Vagrant environments on the system, by reading cached data. Completes quickly but results may be outdated

Print the status of all Vagrant environments on the system, after rebuilding the environment information cache. Results are always correct but completion takes longer

The directory containing the Vagrantfile on the host can be accessed on the guest via /vagrant.
Puppet is a software configuration management tool. It is based on a client-server architecture, where a Puppet agent (client, running as root on each managed node) periodically gathers information (facts) about the local node state via the Facter tool, then communicates this information to the Puppet master (server, running as the puppet user and listening on TCP port 8140). The Puppet master then sends back to the Puppet agent a catalog containing the desired configuration for that node. The Puppet agent applies the needed changes so that the node's configuration converges with the desired configuration, and sends back a report to the Puppet master. Puppet changes are idempotent.

Puppet configurations are based on resources (e.g. "package", "service", "file", "user" ...). For each resource, a list of attributes is specified, with the desired value for each attribute. Each resource type is implemented through providers (e.g. yum, rpm, apt, opkg ... for the resource "package"). Resources managed together as a single unit can be grouped into classes; classes are contained in manifests which are files with the .pp extension. Modules are directories containing self-contained pieces of configuration and classes for a specific complex setting, e.g. an Apache webserver or a MySQL server.

/etc/puppet/puppet.conf
/etc/puppetlabs/puppet/puppet.conf

facter

puppet agent

puppet agent --enable
puppet agent --disable "Reason for disabling"
cat $(puppet config print vardir)/state/agent_disabled.lock

puppet agent --noop

puppet --version
puppet agent --version
puppet master --version

puppet module list
puppet resource user username
puppet resource service httpd enable=false
puppet describe user
puppet describe --list
puppet describe user --providers
puppet apply modulename/init.pp

puppet cert operation

Configuration file (Open Source Puppet)
Configuration file (Puppet Enterprise)

Gather the facts about the managed node, returning a list of key-value pairs

Main Puppet client. Retrieves the node's desired configuration from the Puppet master and applies it

Enable the Puppet agent on the node

Disable the Puppet agent on the node

Perform a dry run, displaying the changes that Puppet would have applied without actually applying them

Show version of different Puppet components

List all modules installed in Puppet

Inspect the state of the resource "user" with respect to username

Modify the state of the resource "service" (in this case, disable the HTTP server)

Show information about the resource "user"

List all resource types

Return the list of providers for the resource "user"

Apply a manifest one time only

Manage the SSL certificates used for communications between master and agents
Ansible is an open source tool for configuration management and software provisioning. It is agentless and connects to the managed machines via SSH pubkey authentication. It only requires OpenSSH and Python to be installed on the managed nodes.

The configuration for managed nodes is specified in one or more playbook, written in YAML and containing a number of tasks. When a playbook is run, first it collects system and environment information (facts) which is then stored in multiple variables named ansible_varname.

/etc/ansible/hosts

Inventory file, containing the list of hosts managed by Ansible. Can be in INI or YAML format

ansible hosts -m module options

Apply the options concerning module to the specified hosts

ansible-playbook options playbook.yml

Apply the specified playbook
## HTML 4.01 components

<table>
<thead>
<tr>
<th>Tag</th>
<th>Attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;h1&gt;</code>-<code>&lt;h6&gt;</code></td>
<td>`align=left</td>
<td>center</td>
</tr>
<tr>
<td><code>&lt;br&gt;</code></td>
<td>`align=left</td>
<td>center</td>
</tr>
<tr>
<td><code>&lt;hr&gt;</code></td>
<td><code>noshade</code></td>
<td>Solid rendering instead of 3D †</td>
</tr>
<tr>
<td></td>
<td><code>size=npixels</code></td>
<td>Line height</td>
</tr>
<tr>
<td></td>
<td>`width=npixels</td>
<td>percent%`</td>
</tr>
<tr>
<td><code>&lt;p&gt;</code></td>
<td>`align=left</td>
<td>center</td>
</tr>
<tr>
<td><code>&lt;div&gt;</code></td>
<td>Group</td>
<td>Group of elements</td>
</tr>
<tr>
<td><code>&lt;a&gt;</code> Anchor</td>
<td><code>charset=encoding</code></td>
<td>Character encoding of target URL</td>
</tr>
<tr>
<td></td>
<td>`coords=left,top,right,bottom</td>
<td>cx,cy,radius</td>
</tr>
<tr>
<td></td>
<td><code>href=url</code></td>
<td>Target URL</td>
</tr>
<tr>
<td></td>
<td><code>hreffield=language</code></td>
<td>Language of document at the target URL</td>
</tr>
<tr>
<td></td>
<td><code>name=section</code></td>
<td>Name of anchor for document bookmarking</td>
</tr>
<tr>
<td></td>
<td>`rel</td>
<td>rev=alternate</td>
</tr>
<tr>
<td></td>
<td>`shape=rectangle</td>
<td>circle</td>
</tr>
<tr>
<td></td>
<td>`target=_blank</td>
<td>_parent</td>
</tr>
<tr>
<td></td>
<td><code>type=mimetype</code></td>
<td>MIME type of target URL</td>
</tr>
<tr>
<td><code>&lt;dl&gt;</code></td>
<td>Definition list</td>
<td>Description of a definition term</td>
</tr>
<tr>
<td><code>&lt;dt&gt;</code></td>
<td>Definition term</td>
<td>Description of a definition term</td>
</tr>
<tr>
<td><code>&lt;dd&gt;</code></td>
<td>Description of a definition term</td>
<td>Description of a definition term</td>
</tr>
<tr>
<td><code>&lt;ol&gt;</code></td>
<td><code>compact=compact</code></td>
<td>List must be more compact †</td>
</tr>
<tr>
<td></td>
<td><code>start=firstnumber</code></td>
<td>Number to start the list on †</td>
</tr>
<tr>
<td></td>
<td>`type=A</td>
<td>a</td>
</tr>
<tr>
<td><code>&lt;ul&gt;</code></td>
<td><code>compact=compact</code></td>
<td>List must be more compact †</td>
</tr>
<tr>
<td></td>
<td>`type=disc</td>
<td>square</td>
</tr>
<tr>
<td><code>&lt;li&gt;</code></td>
<td>`type=disc</td>
<td>square</td>
</tr>
<tr>
<td></td>
<td><code>value=itemno</code></td>
<td>List item value †</td>
</tr>
</tbody>
</table>

† = deprecated
<table>
<thead>
<tr>
<th>Tag</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;i&gt;</code> Italic</td>
<td></td>
</tr>
<tr>
<td><code>&lt;b&gt;</code> Bold</td>
<td></td>
</tr>
<tr>
<td><code>&lt;strike&gt;</code> Strike-through</td>
<td>Strike-through text †</td>
</tr>
<tr>
<td><code>&lt;u&gt;</code> Underlined</td>
<td>Underlined text †</td>
</tr>
<tr>
<td><code>&lt;big&gt;</code> Bigger</td>
<td></td>
</tr>
<tr>
<td><code>&lt;small&gt;</code> Smaller</td>
<td></td>
</tr>
<tr>
<td><code>&lt;sub&gt;</code> Subscript</td>
<td></td>
</tr>
<tr>
<td><code>&lt;sup&gt;</code> Superscript</td>
<td></td>
</tr>
<tr>
<td><code>&lt;tt&gt;</code> Teletype</td>
<td>Monospaced text</td>
</tr>
<tr>
<td><code>&lt;em&gt;</code> Emphasized</td>
<td></td>
</tr>
<tr>
<td><code>&lt;strong&gt;</code> Strong</td>
<td></td>
</tr>
<tr>
<td><code>&lt;del&gt;</code> Deleted</td>
<td>Deleted/inserted text</td>
</tr>
<tr>
<td><code>&lt;ins&gt;</code> Inserted</td>
<td></td>
</tr>
<tr>
<td><code>&lt;pre&gt;</code> Preformatted</td>
<td>Width=ncharacters Max number of characters per line †</td>
</tr>
<tr>
<td><code>&lt;code&gt;</code> Code</td>
<td>Source code text</td>
</tr>
<tr>
<td><code>&lt;samp&gt;</code> Sample</td>
<td>Sample code text</td>
</tr>
<tr>
<td><code>&lt;kbd&gt;</code> Keyboard</td>
<td>Keyboard key</td>
</tr>
<tr>
<td><code>&lt;var&gt;</code> Variable</td>
<td>Variable name</td>
</tr>
<tr>
<td><code>&lt;cite&gt;</code> Citation</td>
<td>Citation block</td>
</tr>
<tr>
<td><code>&lt;blockquote&gt;</code> Quotation</td>
<td></td>
</tr>
<tr>
<td><code>&lt;q&gt;</code> Short quotation</td>
<td>cite=url URL to document containing the quote</td>
</tr>
<tr>
<td><code>&lt;address&gt;</code> Address</td>
<td>Address block</td>
</tr>
<tr>
<td><code>&lt;abbr&gt;</code> Abbreviation</td>
<td></td>
</tr>
<tr>
<td><code>&lt;acronym&gt;</code> Acronym</td>
<td></td>
</tr>
<tr>
<td><code>&lt;dfn&gt;</code> Definition</td>
<td>Definition term</td>
</tr>
<tr>
<td><code>&lt;font&gt;</code> Font</td>
<td>Text color Text font Text size</td>
</tr>
<tr>
<td><code>&lt;bdo&gt;</code> Bidirectional override</td>
<td>dir=ltr</td>
</tr>
<tr>
<td><code>&lt;xmp&gt;</code> XMP</td>
<td>Non-formatted text † (ignores other HTML tags)</td>
</tr>
</tbody>
</table>

other tags Attributes common to almost all other tags

† = deprecated
<table>
<thead>
<tr>
<th>Tag</th>
<th>Attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;img&gt;</code></td>
<td>`align=top</td>
<td>bottom</td>
</tr>
<tr>
<td></td>
<td><code>alt=alternatetext</code></td>
<td>Description of the image for text-only browsers</td>
</tr>
<tr>
<td></td>
<td><code>border=npixels</code></td>
<td>Border width around the image †</td>
</tr>
<tr>
<td></td>
<td>`height=npixels</td>
<td>percent%`</td>
</tr>
<tr>
<td></td>
<td><code>hspace=npixels</code></td>
<td>Blank space on the left and right side of image †</td>
</tr>
<tr>
<td></td>
<td><code>ismap=url</code></td>
<td>URL for server-side image map</td>
</tr>
<tr>
<td></td>
<td><code>longdesc=url</code></td>
<td>URL containing a long description of the image</td>
</tr>
<tr>
<td></td>
<td><code>src=url</code></td>
<td>URL of the image</td>
</tr>
<tr>
<td></td>
<td><code>usemap=url</code></td>
<td>URL for client-side image map</td>
</tr>
<tr>
<td></td>
<td><code>vspace=npixels</code></td>
<td>Blank space on top and bottom of image †</td>
</tr>
<tr>
<td></td>
<td>`width=npixels</td>
<td>percent%`</td>
</tr>
<tr>
<td><code>&lt;map&gt;</code></td>
<td><code>id=id</code></td>
<td>Unique ID for the map tag</td>
</tr>
<tr>
<td></td>
<td><code>name=name</code></td>
<td>Unique name for the map tag</td>
</tr>
<tr>
<td><code>&lt;area&gt;</code></td>
<td><code>alt=alternatetext</code></td>
<td>Description of area for text-only browsers</td>
</tr>
<tr>
<td></td>
<td>`coords=left,top,right,bottom</td>
<td>cx, cy, radius</td>
</tr>
<tr>
<td></td>
<td><code>href=url</code></td>
<td>Target URL of area</td>
</tr>
<tr>
<td></td>
<td>`noreferrer=true</td>
<td>false`</td>
</tr>
<tr>
<td></td>
<td>`shape=rectangle</td>
<td>circle</td>
</tr>
<tr>
<td></td>
<td>`target=_blank</td>
<td>_parent</td>
</tr>
</tbody>
</table>

† = deprecated
<table>
<thead>
<tr>
<th>Tag</th>
<th>Attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;table&gt;</code></td>
<td>Table</td>
<td>Table alignment †</td>
</tr>
<tr>
<td></td>
<td>`align=left</td>
<td>center</td>
</tr>
<tr>
<td></td>
<td>`bgcolor=rgb(r,g,b)</td>
<td>#rrggbb</td>
</tr>
<tr>
<td></td>
<td><code>border=npixels</code></td>
<td>Space around the content of each cell</td>
</tr>
<tr>
<td></td>
<td>`cellpadding=npixels</td>
<td>percent%`</td>
</tr>
<tr>
<td></td>
<td>`cellspacing=npixels</td>
<td>percent%`</td>
</tr>
<tr>
<td></td>
<td>`frame=void</td>
<td>above</td>
</tr>
<tr>
<td></td>
<td>`rules=none</td>
<td>groups</td>
</tr>
<tr>
<td></td>
<td><code>summary=summery</code></td>
<td>Table width</td>
</tr>
<tr>
<td><code>&lt;tr&gt;</code></td>
<td>Table row</td>
<td>Horizontal text alignment</td>
</tr>
<tr>
<td></td>
<td>`align=left</td>
<td>center</td>
</tr>
<tr>
<td></td>
<td>`bgcolor=rgb(r,g,b)</td>
<td>#rrggbb</td>
</tr>
<tr>
<td></td>
<td><code>char=character</code></td>
<td>Alignment offset to first character, if <code>align=char</code></td>
</tr>
<tr>
<td></td>
<td>`charoff=npixels</td>
<td>percent%`</td>
</tr>
<tr>
<td></td>
<td>`valign=top</td>
<td>middle</td>
</tr>
<tr>
<td><code>&lt;td&gt;</code></td>
<td>Table cell</td>
<td>Abbreviated content in a cell</td>
</tr>
<tr>
<td></td>
<td><code>abbr=content</code></td>
<td>Horizontal text alignment</td>
</tr>
<tr>
<td></td>
<td>`align=left</td>
<td>center</td>
</tr>
<tr>
<td></td>
<td><code>axis=category</code></td>
<td>Cell background color †</td>
</tr>
<tr>
<td></td>
<td>`bgcolor=rgb(r,g,b)</td>
<td>#rrggbb</td>
</tr>
<tr>
<td></td>
<td><code>char=character</code></td>
<td>Alignment offset to first character, if <code>align=char</code></td>
</tr>
<tr>
<td></td>
<td>`charoff=npixels</td>
<td>percent%`</td>
</tr>
<tr>
<td></td>
<td><code>colspan=ncolumns</code></td>
<td>Cell header information for text-only browsers</td>
</tr>
<tr>
<td></td>
<td><code>headers=headerid</code></td>
<td>Cell height †</td>
</tr>
<tr>
<td></td>
<td><code>height=npixels</code></td>
<td>Text in cell stays on a single line †</td>
</tr>
<tr>
<td></td>
<td><code>nowrap</code></td>
<td>Number of rows this cell spans on</td>
</tr>
<tr>
<td></td>
<td><code>rowspan=nrows</code></td>
<td>Target for cell header information</td>
</tr>
<tr>
<td></td>
<td>`scope=col</td>
<td>colgroup</td>
</tr>
<tr>
<td></td>
<td>`valign=top</td>
<td>middle</td>
</tr>
<tr>
<td></td>
<td>`width=npixels</td>
<td>percent%`</td>
</tr>
</tbody>
</table>

† = deprecated
<table>
<thead>
<tr>
<th>Dec</th>
<th>Hex</th>
<th>Char</th>
<th>Dec</th>
<th>Hex</th>
<th>Char</th>
<th>Dec</th>
<th>Hex</th>
<th>Char</th>
<th>Dec</th>
<th>Hex</th>
<th>Char</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00</td>
<td>NUL</td>
<td>32</td>
<td>20</td>
<td>space</td>
<td>64</td>
<td>40</td>
<td>@</td>
<td>96</td>
<td>60</td>
<td>`</td>
</tr>
<tr>
<td>1</td>
<td>01</td>
<td>SOH</td>
<td>33</td>
<td>21</td>
<td>!</td>
<td>65</td>
<td>41</td>
<td>A</td>
<td>97</td>
<td>61</td>
<td>a</td>
</tr>
<tr>
<td>2</td>
<td>02</td>
<td>STX</td>
<td>34</td>
<td>22</td>
<td>&quot;</td>
<td>66</td>
<td>42</td>
<td>B</td>
<td>98</td>
<td>62</td>
<td>b</td>
</tr>
<tr>
<td>3</td>
<td>03</td>
<td>ETX</td>
<td>35</td>
<td>23</td>
<td>#</td>
<td>67</td>
<td>43</td>
<td>C</td>
<td>99</td>
<td>63</td>
<td>c</td>
</tr>
<tr>
<td>4</td>
<td>04</td>
<td>EOT</td>
<td>36</td>
<td>24</td>
<td>$</td>
<td>68</td>
<td>44</td>
<td>D</td>
<td>100</td>
<td>64</td>
<td>d</td>
</tr>
<tr>
<td>5</td>
<td>05</td>
<td>ENQ</td>
<td>37</td>
<td>25</td>
<td>%</td>
<td>69</td>
<td>45</td>
<td>E</td>
<td>101</td>
<td>65</td>
<td>e</td>
</tr>
<tr>
<td>6</td>
<td>06</td>
<td>ACK</td>
<td>38</td>
<td>26</td>
<td>&amp;</td>
<td>70</td>
<td>46</td>
<td>F</td>
<td>102</td>
<td>66</td>
<td>f</td>
</tr>
<tr>
<td>7</td>
<td>07</td>
<td>BEL</td>
<td>39</td>
<td>27</td>
<td>'</td>
<td>71</td>
<td>47</td>
<td>G</td>
<td>103</td>
<td>67</td>
<td>g</td>
</tr>
<tr>
<td>8</td>
<td>08</td>
<td>BS</td>
<td>40</td>
<td>28</td>
<td>(</td>
<td>72</td>
<td>48</td>
<td>H</td>
<td>104</td>
<td>68</td>
<td>h</td>
</tr>
<tr>
<td>9</td>
<td>09</td>
<td>HT</td>
<td>41</td>
<td>29</td>
<td>)</td>
<td>73</td>
<td>49</td>
<td>I</td>
<td>105</td>
<td>69</td>
<td>i</td>
</tr>
<tr>
<td>10</td>
<td>0A</td>
<td>LF</td>
<td>42</td>
<td>2A</td>
<td>*</td>
<td>74</td>
<td>4A</td>
<td>J</td>
<td>106</td>
<td>6A</td>
<td>j</td>
</tr>
<tr>
<td>11</td>
<td>0B</td>
<td>VT</td>
<td>43</td>
<td>2B</td>
<td>+</td>
<td>75</td>
<td>4B</td>
<td>K</td>
<td>107</td>
<td>6B</td>
<td>k</td>
</tr>
<tr>
<td>12</td>
<td>0C</td>
<td>FF</td>
<td>44</td>
<td>2C</td>
<td>,</td>
<td>76</td>
<td>4C</td>
<td>L</td>
<td>108</td>
<td>6C</td>
<td>l</td>
</tr>
<tr>
<td>13</td>
<td>0D</td>
<td>CR</td>
<td>45</td>
<td>2D</td>
<td>-</td>
<td>77</td>
<td>4D</td>
<td>M</td>
<td>109</td>
<td>6D</td>
<td>m</td>
</tr>
<tr>
<td>14</td>
<td>0E</td>
<td>SO</td>
<td>46</td>
<td>2E</td>
<td>.</td>
<td>78</td>
<td>4E</td>
<td>N</td>
<td>110</td>
<td>6E</td>
<td>n</td>
</tr>
<tr>
<td>15</td>
<td>0F</td>
<td>SI</td>
<td>47</td>
<td>2F</td>
<td>/</td>
<td>79</td>
<td>4F</td>
<td>O</td>
<td>111</td>
<td>6F</td>
<td>o</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>DLE</td>
<td>48</td>
<td>30</td>
<td>0</td>
<td>80</td>
<td>50</td>
<td>P</td>
<td>112</td>
<td>70</td>
<td>p</td>
</tr>
<tr>
<td>17</td>
<td>11</td>
<td>DC1</td>
<td>49</td>
<td>31</td>
<td>1</td>
<td>81</td>
<td>51</td>
<td>Q</td>
<td>113</td>
<td>71</td>
<td>q</td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td>DC2</td>
<td>50</td>
<td>32</td>
<td>2</td>
<td>82</td>
<td>52</td>
<td>R</td>
<td>114</td>
<td>72</td>
<td>r</td>
</tr>
<tr>
<td>19</td>
<td>13</td>
<td>DC3</td>
<td>51</td>
<td>33</td>
<td>3</td>
<td>83</td>
<td>53</td>
<td>S</td>
<td>115</td>
<td>73</td>
<td>s</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>DC4</td>
<td>52</td>
<td>34</td>
<td>4</td>
<td>84</td>
<td>54</td>
<td>T</td>
<td>116</td>
<td>74</td>
<td>t</td>
</tr>
<tr>
<td>21</td>
<td>15</td>
<td>NAK</td>
<td>53</td>
<td>35</td>
<td>5</td>
<td>85</td>
<td>55</td>
<td>U</td>
<td>117</td>
<td>75</td>
<td>u</td>
</tr>
<tr>
<td>22</td>
<td>16</td>
<td>SYN</td>
<td>54</td>
<td>36</td>
<td>6</td>
<td>86</td>
<td>56</td>
<td>V</td>
<td>118</td>
<td>76</td>
<td>v</td>
</tr>
<tr>
<td>23</td>
<td>17</td>
<td>ETB</td>
<td>55</td>
<td>37</td>
<td>7</td>
<td>87</td>
<td>57</td>
<td>W</td>
<td>119</td>
<td>77</td>
<td>w</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td>CAN</td>
<td>56</td>
<td>38</td>
<td>8</td>
<td>88</td>
<td>58</td>
<td>X</td>
<td>120</td>
<td>78</td>
<td>x</td>
</tr>
<tr>
<td>25</td>
<td>19</td>
<td>EM</td>
<td>57</td>
<td>39</td>
<td>9</td>
<td>89</td>
<td>59</td>
<td>Y</td>
<td>121</td>
<td>79</td>
<td>y</td>
</tr>
<tr>
<td>26</td>
<td>1A</td>
<td>SUB</td>
<td>58</td>
<td>3A</td>
<td>;</td>
<td>90</td>
<td>5A</td>
<td>Z</td>
<td>122</td>
<td>7A</td>
<td>z</td>
</tr>
<tr>
<td>27</td>
<td>1B</td>
<td>ESC</td>
<td>59</td>
<td>3B</td>
<td>:</td>
<td>91</td>
<td>5B</td>
<td>{</td>
<td>123</td>
<td>7B</td>
<td>l</td>
</tr>
<tr>
<td>28</td>
<td>1C</td>
<td>FS</td>
<td>60</td>
<td>3C</td>
<td>&lt;</td>
<td>92</td>
<td>5C</td>
<td>\</td>
<td>124</td>
<td>7C</td>
<td>l</td>
</tr>
<tr>
<td>29</td>
<td>1D</td>
<td>GS</td>
<td>61</td>
<td>3D</td>
<td>=</td>
<td>93</td>
<td>5D</td>
<td>}</td>
<td>125</td>
<td>7D</td>
<td>}</td>
</tr>
<tr>
<td>30</td>
<td>1E</td>
<td>RS</td>
<td>62</td>
<td>3E</td>
<td>&gt;</td>
<td>94</td>
<td>5E</td>
<td>^</td>
<td>126</td>
<td>7E</td>
<td>~</td>
</tr>
<tr>
<td>31</td>
<td>1F</td>
<td>US</td>
<td>63</td>
<td>3F</td>
<td>?</td>
<td>95</td>
<td>5F</td>
<td>_</td>
<td>127</td>
<td>7F</td>
<td>DEL</td>
</tr>
</tbody>
</table>

Characters 0-31 and 127 are non-printable.

ascii
man ascii

display an ASCII table

showkey -a
prompt for pressing a key and display its ASCII value in decimal, octal, and hex