

GNU/Linux most wanted

Summary of most useful commands

© Copyright 2006-2005, Free Electrons.
Free to share under the terms of the Creative Commons
Attribution-ShareAlike 2.5 license
(<http://creativecommons.org>)

Sources, translations, updates, command and concepts
details on our free training materials:
http://free-electrons.com/training/intro_unix_linux

Thanks to Michel Blanc, Hermann J. Beckers and Thierry
Grellier.

Latest update: Sep 15, 2008

Handling files and directories

Create a directory:

`mkdir dir`

Create nested directories:

`mkdir -p dir1/dir2`

Changing directories:

`cd newdir`

`cd ..` (parent directory)

`cd -` (previous directory)

`cd` (home directory)

`cd ~bill` (home directory of user `bill`)

Print the working (current) directory:

`pwd`

Copy a file to another:

`cp source_file dest_file`

Copy files to a directory:

`cp file1 file2 dir`

Copy directories recursively:

`cp -r source_dir dest_dir`

`rsync -a source_dir/ dest_dir/`

Create a symbolic link:

`ln -s linked_file link`

Rename a file, link or directory:

`mv source_file dest_file`

Remove files or links:

`rm file1 file2`

Remove empty directories:

`rmdir dir`

Remove non-empty directories:

`rm -rf dir`

Listing files

List all “regular” files (not starting with `.`) in
the current directory:
`ls`

Display a long listing:

`ls -l`

List all the files in the current directory,
including “hidden” ones (starting with `.`):
`ls -a`

List by time (most recent files first):

`ls -t`

List by size (biggest files first)

`ls -S`

List with a reverse sort order:

`ls -r`

Long list with most recent files last:

`ls -ltr`

Displaying file contents

Concatenate and display file contents:

`cat file1 file2`

Display the contents of several files (stopping
at each page):

`more file1 file2`

`less file1 file2` (better: extra features)

Display the first 10 lines of a file:

`head -10 file`

Display the last 10 lines of a file:

`tail -10 file`

File name pattern matching

Concatenate all “regular” files:

`cat *`

Concatenate all “hidden” files:

`cat .*`

Concatenate all files ending with `.log`:

`cat *.log`

List “regular” files with `bug` in their name:

`ls *bug*`

List all “regular” files ending with `.` and a single
character:

`ls *.*?`

Handling file contents

Show only the lines in a file containing a given
substring:

`grep substring file`

Case insensitive search:

`grep -i substring file`

Showing all the lines but the ones containing a
substring:

`grep -v substring file`

Search through all the files in a directory:

`grep -r substring dir`

Sort lines in a given file:

`sort file`

Sort lines, only display duplicate ones once:

`sort -u file` (unique)

Changing file access rights

Add write permissions to the current user:

`chmod u+w file`

Add read permissions to users in the file group:

`chmod g+r file`

Add execute permissions to other users:

`chmod o+x file`

Add read + write permissions to all users:

`chmod a+rw file`

Make executable files executable by all:

`chmod a+X *`

Make the whole directory and its contents
accessible by all users:

`chmod -R a+X dir` (recursive)

Comparing files and directories

Comparing 2 files:

`diff file1 file2`

Comparing 2 files (graphical):

`gvimdiff file1 file2`

`tkdiff file1 file2`

`kompere file1 file2`

Comparing 2 directories:

`diff -r dir1 dir2`

Looking for files

Find all files in the current (`.`) directory and its
subdirectories with `log` in their name:

`find . -name “*log”`

Find all the `.pdf` files in `dir` and subdirectories
and run a command on each:

`find . -name “*.pdf” -exec xpdf {} \;`

Quick system-wide file search by pattern
(caution: index based, misses new files):

`locate “*pub”`

Redirecting command output

Redirect command output to a file:

`ls *.png > image_files`

Append command output to an existing file:

`ls *.jpg >> image_files`

Redirect command output to the input of
another command:

`cat *.log | grep error`

Job control

Show all running processes:

`ps -ef`

Live hit-parade of processes (press `P`, `M`, `T`: sort
by Processor, Memory or Time usage):

`top`

Send a termination signal to a process:

`kill <pid>` (number found in `ps` output)

Have the kernel kill a process:

`kill -9 <pid>`

Kill all processes (at least all user ones):

`kill -9 -1`

Kill a graphical application:

`xkill` (click on the program window to kill)

File and partition sizes

Show the total size on disk of files or
directories (`disk usage`):

`du -sh dir1 dir2 file1 file2`

Number of bytes, words and lines in file:

`wc file` (`word count`)

Show the size, total space and free space of the
current partition:

`df -h`

Display these info for all partitions:

`df -h`

Compressing

Compress a file:

`gzip file` (`.gz` format)

`bzip2 file` (`.bz2` format, better)

Uncompress a file:

`gunzip file.gz`

`bunzip2 file.bz2`

Archiving

Create a compressed archive (**tape** archive):

`tar zcvf archive.tar.gz dir/`

`tar jcvf archive.tar.bz2 dir/` (better)

Test (list) a compressed archive:

`tar tzvf archive.tar.gz`

`tar jtvf archive.tar.bz2`

Extract the contents of a compressed archive:

`tar zxvf archive.tar.gz`

`tar jxvf archive.tar.bz2`

`tar` options:

c: create

t: test

x: extract

j: on the fly `bzip2` (un)compression

z: on the fly `gzip` (un)compression

Using 7-zip: (better compression than `bzip2`!)

`7z a archive.7z <files>` (add: create)

`7z l archive.7z` (list)

`7z x archive.7z` (extract)

7-zip compressed tar archive
(keeps user and group information)

`tar cf - dir | 7z a -si dir.tar.7z` (create)

`7z x -so dir.tar.7z | tar xf -` (extract)

Handling zip archives

`zip -r archive.zip <files>` (create)

`unzip -t archive.zip` (test / list)

`unzip archive.zip` (extract)

Printing

Send PostScript or text files to `queue`:

`lpr -Pqueue f1.ps f2.txt` (local **p**rinter)

List all the print jobs in `queue`:

`lpq -Pqueue`

Cancel a print job number in `queue`:

`cancel 123 queue`

Print a PDF file:

`pdf2ps doc.pdf`

`lpr doc.ps`

View a PostScript file:

`ps2pdf doc.ps`

`xpdf doc.pdf`

User management

List users logged on the system:

`who`

Show which user I am logged as:

`whoami`

Show which groups `user` belongs to:

`groups user`

Tell more information about `user`:

`finger user`

Switch to user `hulk`:

`su -hulk`

Switch to super user (`root`):

`su -` (**s**witch **u**ser)

`su` (keep same directory and environment)

Time management

Wait for 60 seconds:

`sleep 60`

Show the current date:

`date`

Count the time taken by a command:

`time find_ charming_prince -cute -rich`

Command help

Basic help (works for most commands):

`grep --help`

Access the full manual page of a command:

`man grep`

Misc commands

Basic command-line calculator

`bc -l`

Basic system administration

Change the owner and group of a directory and
all its contents:

`chown -R newuser:newgroup dir`

Reboot the machine in 5 minutes:

`shutdown -r +5`

Shutdown the machine now:

`shutdown -h now`

Display all available network interfaces:

`ifconfig -a`

Assign an IP address to a network interface:

`ifconfig eth0 207.46.130.108`

Bring down a network interface:

`ifconfig eth0 down`

Define a default gateway for packets to
machines outside the local network:

`route add default gw 192.168.0.1`

Delete the default route:

`route del default`

Test networking with another machine:

`ping 207.46.130.108`

Create or remove partitions on the first IDE
hard disk:

`fdisk /dev/hda1`

Create (format) an ext3 filesystem:

`mkfs.ext3 /dev/hda1`

Create (format) a FAT32 filesystem:

`mkfs.vfat -v -F 32 /dev/hda2`

Mount a formatted partition:

`mkdir /mnt/usbdisk` (just do it once)

`mount /dev/uba1 /mnt/usbdisk`

Mount a filesystem image (loop device):

`mount -o loop initrd.img /mnt/initrd`

Unmount a filesystem:

`umount /mnt/usbdisk`

Check the system kernel version:

`uname -a`