

GNU/Linux most wanted

Summary of most useful commands

©Copyright 2014-2005, Free Electrons.
Free to share under the terms of the Creative Commons
Attribution-ShareAlike 3.0 license
(<http://creativecommons.org>)

Sources:
<http://git.free-electrons.com/training-materials>
Updates:
<http://free-electrons.com/doc/training/embedded-linux>
Translations, command and concepts details:
http://free-electrons.com/training/intro_unix_linux

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

Latest update: Jun 19, 2016

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+rX dir (recursive)
```

Comparing files and directories

Comparing 2 files:

```
diff file1 file2
```

Comparing 2 files (graphical):

```
gvimdiff file1 file2
tkdiff file1 file2
meld 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)
lzma file (.lzma format, best compression)
xz file (.xz format, best for code)
```

Uncompress a file:

```
gunzip file.gz
bunzip2 file.bz2
unlzma file.lzma
```

`unxz file.xz`

Archiving

Create a compressed archive (`tape archive`):

```
tar zcvf archive.tar.gz dir
tar jcvf archive.tar.bz2 dir
tar Jcvf archive.tar.xz dir
tar --lzma -cvf archive.tar.lzma
```

Test (list) a compressed archive:

```
tar tvf archive.tar.[gz|bz2|lzma|xz]
```

Extract the contents of a compressed archive:

```
tar xvf archive.tar.[gz|bz2|lzma|xz]
```

`tar` options:

```
c: create
t: test
x: extract
j: on the fly bzip2 (un)compression
J: on the fly xz (un)compression
z: on the fly gzip (un)compression
```

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 printer)
```

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.doc
lpr doc.ps
```

View a PostScript file:

```
ps2pdf doc.doc
lpr 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 - (switch user)
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:

```
sudo chown -R newuser.newgroup dir
```

Reboot the machine in 5 minutes:

```
sudo shutdown -r +5
```

Shutdown the machine now:

```
sudo shutdown -h now
```

Display all available network interfaces:

```
ifconfig -a
```

Assign an IP address to a network interface:

```
sudo ifconfig eth0 207.46.130.108
```

Bring down a network interface:

```
sudo ifconfig eth0 down
```

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

```
sudo route add default gw 192.168.0.1
```

Delete the default route:

```
sudo 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)
sudo mount /dev/ubal1 /mnt/usbdisk
```

Mount a filesystem image (loop device):

```
sudo mount -o loop fs.img /mnt/fs
```

Unmount a filesystem:

```
sudo umount /mnt/usbdisk
```

Check the system kernel version:

```
uname -a
```

