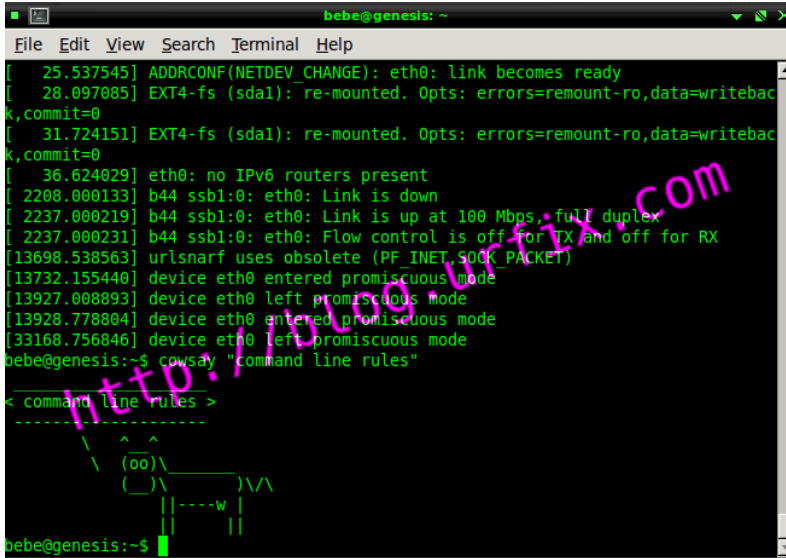


As a Linux user you'll come to learn and love certain commands. Remembering these commands is the toughest part.

A screenshot of a Linux terminal window. The window title is "bebe@genesis: ~". The terminal shows a series of network-related messages in green text on a black background, including "ADDRCONF(NETDEV\_CHANGE): eth0: link becomes ready", "EXT4-fs (sda1): re-mounted. Opts: errors=remount-ro,data=writeback", and "eth0: no IPv6 routers present". At the bottom, the user has entered the command "cowsay 'command line rules'", which has produced a cow ASCII art with the text "command line rules" above it. A large, semi-transparent pink watermark "http://www.prefix.com" is overlaid diagonally across the terminal output.

Some people use cheat-sheets some create scripts, and some just refer to website for their fix. Here I have posted the 25 top command line snippets.

## 25) **sshfs name@server:/path/to/folder /path/to/mount/point**

Mount folder/filesystem through SSH

Install SSHFS from <http://fuse.sourceforge.net/sshfs.html>

Will allow you to mount a folder security over a network.

## 24) **!!:gs/foo/bar**

Runs previous command replacing foo by bar every time that foo appears

Very useful for rerunning a long command changing some arguments globally.

As opposed to `^foo^bar`, which only replaces the first occurrence of foo, this one changes every occurrence.

## 23) **mount | column -t**

currently mounted filesystems in nice layout

Particularly useful if you're mounting different drives, using the following command will allow you to see all the filesystems currently mounted on your computer and their respective specs with the added benefit of nice formatting.

## 22) **<space>command**

Execute a command without saving it in the history

Prepending one or more spaces to your command won't be saved in history.

Useful for pr0n or passwords on the commandline.

21) **ssh user@host cat /path/to/remotefile | diff /path/to/localfile -**

Compare a remote file with a local file

Useful for checking if there are differences between local and remote files.

20) **mount -t tmpfs tmpfs /mnt -o size=1024m**

Mount a temporary ram partition

Makes a partition in ram which is useful if you need a temporary working space as read/write access is fast.

Be aware that anything saved in this partition will be gone after your computer is turned off.

19) **dig +short txt <keyword>.wp.dg.cx**

Query Wikipedia via console over DNS

Query Wikipedia by issuing a DNS query for a TXT record. The TXT record will also include a short URL to the complete corresponding Wikipedia entry.

18) **netstat -tlnp**

Lists all listening ports together with the PID of the associated process

The PID will only be printed if you're holding a root equivalent ID.

17) **dd if=/dev/dsp | ssh -c arcfour -C username@host dd of=/dev/dsp**

output your microphone to a remote computer's speaker

This will output the sound from your microphone port to the ssh target computer's speaker port. The sound quality is very bad, so you will hear a lot of hissing.

16) **echo "ls -l" | at midnight**

Execute a command at a given time

This is an alternative to cron which allows a one-off task to be scheduled for a certain time.

15) **curl -u user:pass -d status="Tweeting from the shell"**

**http://twitter.com/statuses/update.xml**

Update twitter via curl

14) **ssh -N -L2001:localhost:80 somemachine**

start a tunnel from some machine's port 80 to your local port 2001

now you can access the website by going to <http://localhost:2001/>

### 13) **reset**

Salvage a borked terminal

If you bork your terminal by sending binary data to STDOUT or similar, you can get your terminal back using this command rather than killing and restarting the session. Note that you often won't be able to see the characters as you type them.

### 12) **ffmpeg -f x11grab -s wxga -r 25 -i :0.0 -sameq /tmp/out.mpg**

Capture video of a linux desktop

### 11) **> file.txt**

Empty a file

For when you want to flush all content from a file without removing it (hat-tip to Marc Kilgus).

### 10) **\$ssh-copy-id user@host**

Copy ssh keys to user@host to enable password-less ssh logins.

To generate the keys use the command ssh-keygen

### 9) **ctrl-x e**

Rapidly invoke an editor to write a long, complex, or tricky command

Next time you are using your shell, try typing ctrl-x e (that is holding control key press x and then e). The shell will take what you've written on the command line thus far and paste it into the editor specified by \$EDITOR. Then you can edit at leisure using all the powerful macros and commands of vi, emacs, nano, or whatever.

### 8) **!whatever:p**

Check command history, but avoid running it

!whatever will search your command history and execute the first command that matches 'whatever'. If you don't feel safe doing this put :p on the end to print without executing.

Recommended when running as superuser.

### 7) **mtr google.com**

mtr, better than traceroute and ping combined

mtr combines the functionality of the traceroute and ping programs in a single network diagnostic tool.

As mtr starts, it investigates the network connection between the host mtr runs on and HOSTNAME. by sending packets with purposely low TTLs. It continues to send packets with low TTL, noting the response time of the intervening routers. This allows mtr to print the

response percentage and response times of the internet route to HOSTNAME. A sudden increase in packetloss or response time is often an indication of a bad (or simply over-loaded) link.

#### 6) **cp filename{,.bak}**

quickly backup or copy a file with bash

#### 5) **^foo^bar**

Runs previous command but replacing

Really useful for when you have a typo in a previous command. Also, arguments default to empty so if you accidentally run:

echo "no typozs"

you can correct it with

^z

#### 4) **cd -**

change to the previous working directory

#### 3):**w !sudo tee %**

Save a file you edited in vim without the needed permissions

I often forget to sudo before editing a file I don't have write permissions on. When you come to save that file and get the infamous "E212: Can't open file for writing", just issue that vim command in order to save the file without the need to save it to a temp file and then copy it back again.

#### 2) **python -m SimpleHTTPServer**

Serve current directory tree at http://\$HOSTNAME:8000/

#### 1) **sudo !!**

Run the last command as root

Useful when you forget to use sudo for a command. "!!" grabs the last run command.

[source](#)