I know how much you guys love top Linux commands. Lets consider this list a part two of my 25 More – Sick Linux Commands

Be sure to check out commandlinefu for more awesome gems.

1) Monitor progress of a command

pv access.log | gzip > access.log.gz

Pipe viewer is a terminal-based tool for monitoring the progress of data through a pipeline. It can be inserted into any normal pipeline between two processes to give a visual indication of how quickly data is passing through, how long it has taken, how near to completion it is, and an estimate of how long it will be until completion.
Source: http://www.catonmat.net/blog/unix-utilities-pipe-viewer/

2) Graphical tree of sub-directories

```
ls -R | grep ":$" | sed -e 's/:$//' -e 's/[^-][^\/]*\//--/g' -e 's/^/   /' -e 's/-/|/'
```

Prints a graphical directory tree from your current directory

3) Delete all files in a folder that don't match a certain file extension

```
rm !(*.foo|*.bar|*.baz)
```

Deletes all files in a folder that are NOT *.foo, *.bar or *.baz files. Edit the pattern inside the brackets as you like.

4) Easy and fast access to often executed commands that are very long and complex.

```
some_very_long_and_complex_command # label
```

When using reverse-i-search you have to type some part of the command that you want to retrieve. However, if the command is very complex it might be difficult to recall the parts that will uniquely identify this command. Using the above trick it’s possible to label your commands and access them easily by pressing ^R and typing the label (should be short and descriptive).

5) Define a quick calculator function

```
? () { echo "$*" | bc -l; }
```

defines a handy function for quick calculations from cli.

once defined:

? 10*2+3

6) Display a cool clock on your terminal

```
watch -t -n1 "date +%T|figlet"
```

This command displays a clock on your terminal which updates the time every second. Press Ctrl-C to exit.

A couple of variants:

A little bit bigger text:

watch -t -n1 "date +%T|figlet -f big"You can try other figlet fonts, too.

Big sideways characters:

watch -n 1 -t '/usr/games/banner -w 30 $(date +%M:%S)'This requires a particular version of banner and a 40-line terminal or you can adjust the width (“30” here).

7) intercept stdout/stderr of another process

```
strace -ff -e trace=write -e write=1,2 -p SOME_PID
```

8) Remove duplicate entries in a file without sorting.

```
awk '!x[$0]++' <file>
```

Using awk, find duplicates in a file without sorting, which reorders the contents. awk will not reorder them, and still find and remove duplicates which you can then redirect into another file.

9) Record a screencast and convert it to an mpeg

```
ffmpeg -f x11grab -r 25 -s 800x600 -i :0.0 /tmp/outputFile.mpg
```

Grab X11 input and create an MPEG at 25 fps with the resolution 800×600

10) Mount a .iso file in UNIX/Linux

```
mount /path/to/file.iso /mnt/cdrom -oloop
```

“-o loop” lets you use a file as a block device

11) Insert the last command without the last argument (bash)

```
!:-
```

/usr/sbin/ab2 -f TLS1 -S -n 1000 -c 100 -t 2 http://www.google.com/then

!:- http://www.urfix.com/is the same as

/usr/sbin/ab2 -f TLS1 -S -n 1000 -c 100 -t 2 http://www.urfix.com/

12) Convert seconds to human-readable format

```
date -d@1234567890
```

This example, for example, produces the output, “Fri Feb 13 15:26:30 EST 2009”

13) Job Control

```
^Z $bg $disown
```

You’re running a script, command, whatever.. You don’t expect it to take long, now 5pm has rolled around and you’re ready to go home… Wait, it’s still running… You forgot to nohup it before running it… Suspend it, send it to the background, then disown it… The ouput wont go anywhere, but at least the command will still run…

14) Edit a file on a remote host using vim

```
vim scp://username@host//path/to/somefile
```

15) Monitor the queries being run by MySQL

```
watch -n 1 mysqladmin --user=<user> --password=<password> processlist
```

Watch is a very useful command for periodically running another command – in this using mysqladmin to display the processlist. This is useful for monitoring which queries are causing your server to clog up.

More info here: http://codeinthehole.com/archives/2-Monitoring-MySQL-processes.html

16) escape any command aliases

```
\[command]
```

e.g. if rm is aliased for ‘rm -i’, you can escape the alias by prepending a backslash:

rm [file] # WILL prompt for confirmation per the alias

\rm [file] # will NOT prompt for confirmation per the default behavior of the command

17) Show apps that use internet connection at the moment. (Multi-Language)

```
ss -p
```

for one line per process:

ss -p | catfor established sockets only:

ss -p | grep STAfor just process names:

ss -p | cut -f2 -sd\"or

ss -p | grep STA | cut -f2 -d\"

18) Send pop-up notifications on Gnome

```
notify-send ["<title>"] "<body>"
```

The title is optional.

Options:

-t: expire time in milliseconds.

-u: urgency (low, normal, critical).

-i: icon path.

On Debian-based systems you may need to install the ‘libnotify-bin’ package.

Useful to advise when a wget download or a simulation ends. Example:

```
wget URL ; notify-send "Done"
```

19) quickly rename a file

```
mv filename.{old,new}
```

20) Remove all but one specific file

```
rm -f !(survivior.txt)
```

21) Generate a random password 30 characters long

```
strings /dev/urandom | grep -o '[[:alnum:]]' | head -n 30 | tr -d '\n'; echo
```

Find random strings within /dev/urandom. Using grep filter to just Alphanumeric characters, and then print the first 30 and remove all the line feeds.

22) Run a command only when load average is below a certain threshold

```
echo "rm -rf /unwanted-but-large/folder" | batch
```

Good for one off jobs that you want to run at a quiet time. The default threshold is a load average of 0.8 but this can be set using atrun.

23) Binary Clock

```
watch -n 1 'echo "obase=2;`date +%s`" | bc'
```

Create a binary clock.

24) Processor / memory bandwidthd? in GB/s

```
dd if=/dev/zero of=/dev/null bs=1M count=32768
```

Read 32GB zero's and throw them away.

How fast is your system?

25) Backup all MySQL Databases to individual files

```
for I in $(mysql -e 'show databases' -s --skip-column-names); do mysqldump $I | gzip > "$I.sql.gz"; done
```

That's all for today and I hope you enjoyed this list.