[betterprogramming.pub](https://betterprogramming.pub/25-awesome-linux-command-one-liners-9495f26f07fb)

**25 Awesome Linux Command One-Liners - Better Programming**

Kesk -\*-

5-6 minutes

**Save time in your day-to-day work and have fun with it**

A person playing a video game

Description automatically generated with low confidence

Photo by [Anete Lusina](https://www.pexels.com/@anete-lusina?utm_content=attributionCopyText&utm_medium=referral&utm_source=pexels) from [Pexels](https://www.pexels.com/photo/crop-hacker-silhouette-typing-on-computer-keyboard-while-hacking-system-5240547/?utm_content=attributionCopyText&utm_medium=referral&utm_source=pexels)

I love working with the Linux shell. The shell lets you get things done in seconds, and no need to use any graphical environment.  
In this post, I compile 25 command one-liners that I find especially useful or funny. I hope you like them!

**1. Query Wikipedia via console**

#dig +short txt <keyword>.wp.dg.cx  
#For example:$dig +short txt usa.wp.dg.cx"a. English is the official language of at least 28 states\226\128\148some sources give higher figures, based on differing definitions of \"official\". English and Hawaiian are both official languages in the state of Hawaii. <http://en.wikipedia.org/wiki/United_States>"

**2. Set a console clock into the right corner**

Copy and paste the following sentence in a single line in your console:

$while sleep 1;do tput sc;tput cup 0 $(($(tput cols)-29));date;tput rc;done &

Text

Description automatically generated

A clock in the console right corner.

Binary clock:

Text

Description automatically generated

A binary clock.

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

**3. Mount a temporary RAM partition**

You can use it as a use and throw partition. However, the partition and all the data will be gone once you reboot the system.

For example, it can be used by programs or tasks that require many reading/writing operations to improve their performance.

$mount -t tmpfs tmpfs /mnt -o size=2048m

The above command will create a temporary partition size 2048MB in tmpfs and mount it under the /mnt directory.

**4. List numerical values for each of the 256 colors in bash**

$for code in {0..255}; do [echo](https://linuxcommandlibrary.com/man/echo) -e "\e[38;05;${code}m $code: Test"; done

Graphical user interface

Description automatically generated

Colors and numerical values.

**5. Run the last command as root**

$sudo !!

**6. Kill process by name**

$pkill -x chrome

**7. Create a complete compress image of a disk**

To create an image of a disk on your computer, you do not need any third-party tools or programs. Doing so is as simple as using the command:

if: origin disk, in this case, sda.

$dd if=/dev/sda | gzip -c > /path/to/my.disk.image.gz bs=1M

Note: If you want to clone the system disk, you will have to boot your computer with a live version, for example, with ubuntu live USB. Make sure no partitions are mounted from the source hard drive disk

bs: Sets the block size to 1M.

Restore de system:

$gunzip -c /path/to/my.disk.image.gz | dd of=/dev/sda

Warning: You should be very careful when using the dd command; it can destroy data. Remember the order of the input file: <if> and output file: <of>.

**8. Run the previous command and replacing**

$^aaa^bbb

**9. Limit the CPU usage for a process**

First, you have to knot the pid (Process identification number). For example, find out the chrome’s pid:

$pid of chrome  
#for example: 1543

Second, set the percentage to which you want to limit the CPU usage to the previous process, in this case, at 25 percent.

$sudo cpulimit -p 1543-l 25

**10. Search for a matching command previously typed in BASH**

$Ctrl-R <search-text>

**11. Processes per user counter**

$ps hax -o user | sort | uniq -c

**12. Replace one char with another in all filenames of the current directory**

You can use the Perl rename on Debian, Ubuntu, and derivatives (other distributions ship a different program as rename, and that program isn’t helpful here).

File 1  
File 2to File\_1  
File\_2$rename -v 's/ /\_/g' \*

With The ZSH or Z Shell:

autoload zmv  
zmv '(\*)' '${1// /\_}'

**13. SSH connection through a host in the middle**

$ssh -t reachable\_host ssh unreachable\_host

**14. Find all files larger than 5M and less than 10M**

Useful, for example, to search for a log file of a specific size.

$find / -type f -size +50M -size -10M

**15. Use the <rsync> command with a progress bar to copy**

$rsync -rv <src> <dst> --progress

**16. Check whether your computer is 32-bit or 64-bit**

$getconf LONG\_BIT

**17. Simulates typing on the screen**

You can simulate typing on the screen and look like a hacker by controlling the speed at which you type.

$echo "Lorem ipsum dolor sit amet, consectetuer adipiscing elit." | pv -qL 20

* q: Quiet do not output any transfer information at all.
* L: Rate-limit RATE limit transfer to RATE bytes per second.



Typing on the screen.

Note: you need to have installed the< pv> utility used to monitor data progress through a pipe.

$sudo apt install pv

**18. Show working directory of a process**

Note: See point 7 to obtain the PID of a process.

$pwdx pid

**19. Set password protection to a file using vim**

$vim -x file.txt

Graphical user interface

Description automatically generated with medium confidence

Prompt to set the password to the file.

**20. Remove lines in a file by a range**

$sed -i <file> -re '<start>,<end>d'

**21. Get my public IP address.**

$curl ifconfig.me$curl checkip.dyndns.org... or similars.



My IP address.

**22. Monitor the progress of a command**

$pv a\_big\_file.txt | gzip > a\_big\_file.log.gz

Note: you need to have installed the< pv> utility.

**23. List all bash shortcuts**

bind -P

Text

Description automatically generated

Shortcuts.

**24. Monitor memory usage in a human-readable way**

watch vmstat -sSM

Text

Description automatically generated

Memory usage.

**25. A forecast in the shell**

curl wttr.in/newyork

A screenshot of a computer

Description automatically generated with medium confidence

New York forecast by wttr.in

**Extra: Enter the matrix**

$perl -e '$|++; while (1) { print " ." x (rand(10) + 1), int(rand(2)) }'

A picture containing night sky

Description automatically generated

Ohh, the matrix in my computer!

If you enjoyed this article, consider subscribing to Medium through my [profile](https://kesk.medium.com/membership). Thank you!