


[bash](#), [howto](#), [tips](#), [tricks](#), [column](#), [sort](#), [getconf](#), [processorarchitecture](#), [architecture](#), [multitail](#), [watch](#), [nohup](#), [yes](#), [dd](#), [sudo](#), [history](#), [script](#), [tr](#), [xargs](#)

Bash - Command Line Tricks

 Local copy of: <https://likegeeks.com/linux-command-line-tricks/>

 • **column**: Debian-package - util-linux

Display Output as a Table

Sometimes it's painful to read the output well due to the overcrowded strings, for example, the result of the mount command, what about viewing the output like a table? It is an easy job.

```
mount | column -t
```

```
sysfs                on /sys
type sysfs           (rw,nosuid,nodev,noexec,relatime)
proc                 on /proc
type proc            (rw,nosuid,nodev,noexec,relatime)
udev                 on /dev
type devtmpfs        (rw,nosuid,relatime,size=12319496k,nr_inodes=3079874,mode=755)
devpts               on /dev/pts
type devpts          (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs                on /run
type tmpfs           (rw,nosuid,noexec,relatime,size=2468204k,mode=755)
/dev/mapper/detlev--vg-root on /
type ext4            (rw,relatime,errors=remount-ro)
securityfs           on /sys/kernel/security
type securityfs     (rw,nosuid,nodev,noexec,relatime)
tmpfs                on /dev/shm
type tmpfs           (rw,nosuid,nodev)
tmpfs                on /run/lock
type tmpfs           (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs                on /sys/fs/cgroup
type tmpfs           (ro,nosuid,nodev,noexec,mode=755)
cgroup2              on /sys/fs/cgroup/unified
type cgroup2         (rw,nosuid,nodev,noexec,relatime,nsdelegate)
cgroup               on /sys/fs/cgroup/systemd
type cgroup          (rw,nosuid,nodev,noexec,relatime,xattr,name=systemd)
pstore               on /sys/fs/pstore
type pstore          (rw,nosuid,nodev,noexec,relatime)
bpf                  on /sys/fs/bpf
type bpf             (rw,nosuid,nodev,noexec,relatime,mode=700)
cgroup               on /sys/fs/cgroup/net_cls,net_prio
type cgroup          (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup               on /sys/fs/cgroup/pids
type cgroup          (rw,nosuid,nodev,noexec,relatime,pids)
cgroup               on /sys/fs/cgroup/blkio
type cgroup          (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup               on /sys/fs/cgroup/memory
type cgroup          (rw,nosuid,nodev,noexec,relatime,memory)
cgroup               on /sys/fs/cgroup/cpuset
type cgroup          (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup               on /sys/fs/cgroup/freezer
type cgroup          (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup               on /sys/fs/cgroup/cpu,cpuacct
type cgroup          (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup               on /sys/fs/cgroup/perf_event
type cgroup          (rw,nosuid,nodev,noexec,relatime,perf_event)
```

```
cgroup on /sys/fs/cgroup/devices
type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/rdma
type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)
debugfs on /sys/kernel/debug
type debugfs (rw,relatime)
hugetlbfs on /dev/hugepages
type hugetlbfs (rw,relatime,pagesize=2M)
mqueue on /dev/mqueue
type mqueue (rw,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc
type autofs (rw,relatime,fd=48,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=11783)
sunrpc on /run/rpc_pipefs
type rpc_pipefs (rw,relatime)
/dev/sda1 on /boot
type ext2 (rw,relatime)
/dev/mapper/1Tera01--vg-data01 on /mnt/data01
type ext4 (rw,relatime)
binfmt_misc on /proc/sys/fs/binfmt_misc
type binfmt_misc (rw,relatime)
tmpfs on /run/user/1000
type tmpfs (rw,nosuid,nodev,relatime,size=2468200k,mode=700,uid=1000,gid=1000)
gvfsd-fuse on /run/user/1000/gvfs
type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
fusectl on /sys/fs/fuse/connections
type fusectl (rw,relatime)
/dev/fuse on /run/user/1000/doc
type fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
/etc/auto.denker.wiretrip.de-cifs on /home/def/mnt/cifs
type autofs (rw,relatime,fd=6,pgrp=2327,timeout=60,minproto=5,maxproto=5,indirect,pipe_ino=38705)
/etc/auto.denker.wiretrip.de-cifs on /home/def/mnt/cifs/denker.wiretrip.de/austausch
type autofs (rw,relatime,fd=6,pgrp=2327,timeout=60,minproto=5,maxproto=5,offset,pipe_ino=38705)
(rw,relatime,vers=default,cache=strict,username=<USER>,uid=1000,forceuid,gid=1000,forcegid,addr=192.168.0.1,file_mode=0644,dir_mode=0755,soft,nounix,mapposix,rsize=1048576,wsiz=1048576,echo_interval=60,actimeo=1)
```

OK, in this example, we see the output is well formatted because the separator between them is spaces.

What if the separators are something else, like colons :

The /etc/passwd file is a good example.

Just specify the separator with -s parameter like this:

```
cat /etc/passwd | column -t -s :
```

~\ownCloud\zim\Bilder\DEF\bash-command-line-tricks\002-users-tabular-view.png

Run Until Success

If you search google about that trick, you will find a lot of questions about people asking how to repeat the command till it returns success and runs properly, like ping the server till it becomes alive or check if a file with a specific extension is uploaded at specific directory or maybe check if a specific URL becomes available or maybe any geeky thing, the list is very long.

You can use the while true loop to achieve that:

~\ownCloud\zim\Bilder\DEF\bash-command-line-tricks\03-repeat-till-success.png

We use > /dev/null 2>&1 to redirect normal output and errors to /dev/null.

Actually, this is one of coolest Linux Command Line Tricks for me.

Sort Processes by Memory or CPU Usage

To sort by memory usage: [~\ownCloud\zim\Bilder\DEF\bash-command-line-tricks\04-sort-by-memory-usage.png](#)

To sort by CPU usage: [~\ownCloud\zim\Bilder\DEF\bash-command-line-tricks\05-sort-by-cpu-usage.png](#)

Check Your Architecture

getconf LONG_BIT

Monitor Multiple Log Files Concurrently

You can use the tail command to watch your logs and that's fine, but sometimes you may need to monitor multiple log files simultaneously to take some actions.

Using multital command which supports text highlighting, filtering, and many other features that you may need.

[~\ownCloud\zim\Bilder\DEF\bash-command-line-tricks\06-multitail.png](#)

You can install it if it is not found on your system like this:

```
apt-get install multitail
```

Return to Your Previous Directory

It's not a trick but some people forget it, others use it every minute.

Just type `cd -` and you will return back to the previous directory.

Watch Command Output

By using watch command, you can watch any output of any command, for example, you can watch the free space and how it is growing:

```
watch df -h
```

You can imagine what you can do with any variant data that you can watch using watch command.

Run Your Program After Session Killing

When you run any program in the background and close your shell, definitely it will be killed, what about if it continues running after closing the shell.

This can be done using the nohup command which stands for no hang up.

```
nohup wget site.com/file.zip
```

This command is really one of the most useful Linux command line tricks for most webmasters.

[~\ownCloud\zim\Bilder\DEF\bash-command-line-tricks\07-nohup-command.png](#)

A file will be generated in the same directory with the name nohup.out contains the output of the running program.

[~\ownCloud\zim\Bilder\DEF\bash-command-line-tricks\08-nohup-output.png](#)

Cool command right?

Answer bot Using Yes & No Commands

It's like an answer bot for those commands whose require the user to say yes.

That can be done using the yes command:

```
yes | apt-get update
```

Or maybe you want to automate saying no instead, this can be done using the following command:

yes no | command

~\ownCloud\zim\Bilder\DEF\bash-command-line-tricks\09-yes-command.png

Create a File With a Specific Size

Use the dd command to create a file with a specific size:

```
dd if=/dev/zero of=out.txt bs=1M count=10
```

This will create a file with 10-megabyte size filled with zeros. dd if=/dev/zero of=out.txt bs=1M count=10

Run Last Command as Root

Sometimes you forget to type sudo before your command that requires root privileges to run, you don't have to rewrite it, just type:

```
sudo !!
```

~\ownCloud\zim\Bilder\DEF\bash-command-line-tricks\11-sudo.png

Record your Command Line Session

If you want to record what you've typed in your shell screen, you can use the script command which will save all of your typings to a file named typescript.

```
script
```

Once you type exit, all of your commands will be written to that file so you can review them later.

Replacing Spaces with Tabs

You can replace any character with any other character using tr command which is very handy.

```
cat geeks.txt | tr ':[space]:' '\t' > out.txt
```

This command will replace the spaces with tabs.

~\ownCloud\zim\Bilder\DEF\bash-command-line-tricks\12-tr-command.png

Convert Character Case

```
cat my_file | tr a-z A-Z > output.txt
```

This command converts the content of the file to upper case using the tr command.

Powerful xargs Command

We can say that xargs command is one of the most important Linux command line tricks, you can use this command to pass outputs between commands as arguments, for example, you may search for png files and compress them or do anything with them.

```
find . -name "*.png" -type f -print | xargs tar -cvzf pics.tar.gz
```

Or maybe you have a list of URLs in a file and you want to download them or process them in a different way:

```
cat links.txt | xargs wget
```

The cat command result is passed to the end of xargs command.

What if your command needs the output in the middle?

Just use {} combined with -i parameter to replace the arguments in the place where the result should go like this:

```
ls /etc/*.conf | xargs -i cp {} /home/likegeeks/Desktop/out/
```

```
~\ownCloud\zim\Bilder\DEF\bash-command-line-tricks\13-xargs-command.png
```

From:

<https://wiki.nanoscopic.de/> - **nanoscopic wiki**

Permanent link:

<https://wiki.nanoscopic.de/doku.php/pages/howtos/bash/bash-command-line-tricks?rev=1612987254>

Last update: **2021/02/10 20:00**

