bash, howto, tipps, 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,wsize=1048576,echo_interval=60,actim
eo=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 :

/etc/passwd is a good example.

Just specify the separator with -s parameter like this:

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

```
root              x  0      0      root                              /root                  /bin/bash
daemon            x  1      1      daemon                            /usr/sbin
/usr/sbin/nologin
bin               x  2      2      bin                               /bin
/usr/sbin/nologin
sys               x  3      3      sys                               /dev
/usr/sbin/nologin
sync              x  4      65534  sync                              /bin                   /bin/sync
games             x  5      60     games                             /usr/games
/usr/sbin/nologin
man               x  6      12     man                               /var/cache/man
/usr/sbin/nologin
lp                x  7      7      lp                                /var/spool/lpd
/usr/sbin/nologin
mail              x  8      8      mail                              /var/mail
/usr/sbin/nologin
news              x  9      9      news                              /var/spool/news
/usr/sbin/nologin
uucp              x  10     10     uucp                              /var/spool/uucp
```

```
/usr/sbin/nologin
proxy             x  13     13     proxy                              /bin
/usr/sbin/nologin
backup            x  34     34     backup                             /var/backups
/usr/sbin/nologin
list              x  38     38     Mailing List Manager               /var/list
/usr/sbin/nologin
gnats             x  41     41     Gnats Bug-Reporting System (admin)  /var/lib/gnats
/usr/sbin/nologin
nobody            x  65534  65534  nobody                             /nonexistent
/usr/sbin/nologin
_apt              x  100    65534  /nonexistent                       /usr/sbin/nologin
systemd-timesync  x  101    102    systemd Time Synchronization,,,    /run/systemd
/usr/sbin/nologin
systemd-network   x  102    103    systemd Network Management,,,      /run/systemd
/usr/sbin/nologin
systemd-resolve   x  103    104    systemd Resolver,,,                /run/systemd
/usr/sbin/nologin
messagebus        x  104    110    /nonexistent                       /usr/sbin/nologin
tss               x  105    111    TPM2 software stack,,,             /var/lib/tpm           /bin/false
dnsmasq           x  106    65534  dnsmasq,,,                         /var/lib/misc
/usr/sbin/nologin
usbmux            x  107    46     usbmux daemon,,,                   /var/lib/usbmux
/usr/sbin/nologin
rtkit             x  108    114    RealtimeKit,,,                     /proc
/usr/sbin/nologin
sshd              x  109    65534  /run/sshd                          /usr/sbin/nologin
pulse             x  110    118    PulseAudio daemon,,,               /var/run/pulse
/usr/sbin/nologin
avahi             x  112    120    Avahi mDNS daemon,,,               /var/run/avahi-daemon
/usr/sbin/nologin
saned             x  113    121    /var/lib/saned                     /usr/sbin/nologin
colord            x  114    122    colord colour management daemon,,,  /var/lib/colord
/usr/sbin/nologin
geoclue           x  115    123    /var/lib/geoclue                   /usr/sbin/nologin
hplip             x  116    7      HPLIP system user,,,               /var/run/hplip         /bin/false
sddm              x  117    124    Simple Desktop Display Manager     /var/lib/sddm          /bin/false
user              x  1000   1000   user,,,                            /home/user             /bin/bash
systemd-coredump  x  999    999    systemd Core Dumper                /
/usr/sbin/nologin
_rpc              x  118    65534  /run/rpcbind                       /usr/sbin/nologin
statd             x  119    65534  /var/lib/nfs                       /usr/sbin/nologin
nvpd              x  120    125    NVIDIA Persistence Daemon,,,       /var/run/nvpd/
/usr/sbin/nologin
uuidd             x  121    126    /run/uuidd                         /usr/sbin/nologin
```

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:

```
while true
do
  ping -c 1 heise.de > /dev/null 2>&1 && break
done
```

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:

```
ps aux | sort -nk 4
```

To sort by CPU usage:

```
ps aux | sort -nk 3
```

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

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.

```
cd –
```

Make non-interactive as interactive shell session

To do this, put our settings in ~/.bashrc from ~/.bash_profile.

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.

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

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
```

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 !!
```

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.

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 {} ~/tmp/out/
```

Redo last command as root

```
#as user
vim /etc/fstab
sudo !!
```

Keyboard Shortcuts

- https://en.wikipedia.org/wiki/GNU_Readline

Emacs editing mode key bindings are taken from the text editor Emacs.

On some systems, Esc must be used instead of Alt, because the Alt shortcut conflicts with another shortcut. For example, pressing Alt+f in Xfce's terminal emulator window does not move the cursor forward one word, but activates “File” in the menu of the terminal window, unless that is disabled in the emulator's settings.

- Tab ↹ : Autocompletes from the cursor position.
- Ctrl + A : Moves the cursor to the line start (equivalent to the key Home).
- Ctrl + B : Moves the cursor back one character (equivalent to the key ←).
- Ctrl + C : Sends the signal SIGINT via pseudoterminal to the current task, which aborts and closes it.[d]
- Ctrl + D
  - Sends an EOF marker, which (unless disabled by an option) closes the current shell (equivalent to the command exit). (Only if there is no text on the current line)
  - If there is text on the current line, deletes the current character (then equivalent to the key Delete).
- Ctrl + E : (end) moves the cursor to the line end (equivalent to the key End).
- Ctrl + F : Moves the cursor forward one character (equivalent to the key →).
- Ctrl + G : Abort the reverse search and restore the original line.
- Ctrl + H : Deletes the previous character (same as backspace).
- Ctrl + I : Equivalent to the tab key.

- Ctrl+J : Equivalent to the enter key.
- Ctrl+K : Clears the line content after the cursor and copies it into the clipboard.
- Ctrl+L : Clears the screen content (equivalent to the command clear).
- Ctrl+N : (next) recalls the next command (equivalent to the key ↓).
- Ctrl+O : Executes the found command from history, and fetch the next line relative to the current line from the history for editing.
- Ctrl+P : (previous) recalls the prior command (equivalent to the key ↑).
- Ctrl+R : (reverse search) recalls the last command including the specified characters. A second Ctrl+r recalls the next anterior command that corresponds to the search
- Ctrl+S : Go back to the next more recent command of the reverse search (beware to not execute it from a terminal because this command also launches its XOFF). If you changed that XOFF setting, use Ctrl+q to return.
- Ctrl+T : Transpose the previous two characters.
- Ctrl+U : Clears the line content before the cursor and copies it into the clipboard.
- Ctrl+V : If the next input is also a control sequence, type it literally (e. g. * Ctrl+v Ctrl+h types “^H”, a literal backspace.)
- Ctrl+W : Clears the word before the cursor and copies it into the clipboard.
- Ctrl+X Ctrl+E : Edits the current line in the $EDITOR program, or vi if undefined.
- Ctrl+X Ctrl+R : Read in the contents of the inputrc file, and incorporate any bindings or variable assignments found there.
- Ctrl+X Ctrl+U : Incremental undo, separately remembered for each line.
- Ctrl+X Ctrl+V : Display version information about the current instance of Bash.
- Ctrl+X Ctrl+X : Alternates the cursor with its old position. (C-x, because x has a crossing shape).
- Ctrl+Y : (yank) adds the clipboard content from the cursor position.
- Ctrl+Z : Sends the signal SIGTSTP to the current task, which suspends it. To execute it in background one can enter bg. To bring it back from background or suspension fg ['process name or job id'] (foreground) can be issued.
- Ctrl+_ : Incremental undo, separately remembered for each line.
- Alt+B : (backward) moves the cursor backward one word.
- Alt+C : Capitalizes the character under the cursor and moves to the end of the word.
- Alt+D : Cuts the word after the cursor.
- Alt+F : (forward) moves the cursor forward one word.
- Alt+L : Lowers the case of every character from the cursor's position to the end of the current word.
- Alt+R : Cancels the changes and puts back the line as it was in the history.
- Alt+U : Capitalizes every character from the cursor's position to the end of the current word.
- Alt+. : Insert the last argument to the previous command (the last word of the previous history entry).

~~DISCUSSION~~