

| | | |
|---|---|--|
| File system | Move (rename) file | <code>mv /path/to/source /path/to/destination</code> |
| | Copy file to directory | <code>cp /path/to/source /path/to/destination_directory</code> |
| | Copy the directory with all the files inside it | <code>cp -r /path/to/source_directory /path/to/destination_directory</code> |
| | Copy several files | <code>cp /path/to/file1 /path/to/file2 /path/to/destination_directory</code> |
| | Delete a specific file | <code>rm /path/to/file</code> |
| | Delete files with extension <code>.txt</code> | <code>rm -rf *.txt</code> |
| | Delete directory with all files inside (flag <code>-f</code> for deletion without confirmation) | <code>rm -rf /path/to/directory</code> |
| | Change file owner | <code>chown user:group /path/to/file</code> |
| | Change the owner of the directory and all files inside it | <code>chown -R user:group /path/to/directory</code> |
| | Give all users the permission to read and write to the directory and all files inside it | <code>chmod -R a+rw /path/to/directory</code> |
| | Make the file permission the same as the other file | <code>chmod --reference=/path/to/source /path/to/destination</code> |
| | Mount disk <code>/dev/sdb1</code> with mount point <code>/mnt/usb</code> | <code>mount /dev/sdb1 /mnt/usb</code> |
| | Mount device with file system <code>ext4</code> only for reading | <code>mount /dev/sdb1 /mnt/usb -t ext4 -o ro -o noexec</code> |
| | Unmount mount point | <code>umount /mnt/usb</code> |
| | Reading logs | Force unmount file system |
| Copy blocks of one device to another | | <code>dd if=/path/to/input of=/path/to/output</code> |
| Write the image to the device | | <code>dd bs=4M if=/path/to/linux.iso of=/dev/sdx</code> |
| Display the first 20 lines from the file | | <code>head -n 20 access.log</code> |
| Display the last 30 lines of the file | | <code>tail -n 30 error.log</code> |
| File system search | Launch <code>tail</code> in the tracking mode of new line | <code>tail -f access.log</code> |
| | Open file for paginated output | <code>less access.log</code> |
| | Open file with line number displaying | <code>less -N access.log</code> |
| | Find files with name <code>netdata.conf</code> | <code>find -name 'netdata.conf'</code> |
| | Find all files with extension <code>.conf</code> | <code>find / -name '*.conf'</code> |
| | Find all files with name <code>apache2</code> | <code>find / -type f -name 'apache2'</code> |
| | Find all directories with a name <code>nginx</code> | <code>find / -type d -name 'nginx'</code> |
| | Find all files larger than 100MB in your home directory | <code>find ~ -size +100M</code> |
| | Find in the home directory all files with a size less than 100MB | <code>find ~ -size -100M</code> |
| | Find all empty files in the home directory | <code>find ~ -empty</code> |
| | Delete all empty files in the home directory (<code>{}</code> replaced by the file name) | <code>find ~ -empty -exec rm -rf {} \;</code> |
| | Find word <code>Forbidden</code> in file <code>error.log</code> | <code>grep 'Forbidden' error.log</code> |
| | Find word <code>forbidden</code> in file <code>error.log</code> (case-insensitive search) | <code>grep -i 'forbidden' error.log</code> |
| | Display the number of matches found | <code>grep -c 'forbidden' error.log</code> |
| | Display an additional 2 lines after the match | <code>grep -i -A2 'forbidden' error.log</code> |
| Display an additional 2 lines before the match | <code>grep -i -B2 'forbidden' error.log</code> | |
| Display an additional 2 lines before and after the match | <code>grep -i -C2 'forbidden' error.log</code> | |
| Find a phrase <code>access denied</code> in all files in the folder <code>~/pm2/logs</code> | <code>grep -i -r 'access denied' ~/.pm2/logs</code> | |

| | | |
|--|---|--|
| Dealing with processes | Display a list of all processes | <code>ps aux</code> |
| | Display only <i>node</i> processes | <code>ps aux grep node</code> |
| | Display processes as a tree, show only <i>pid</i> and <i>command</i> | <code>ps -e -o pid,args -forest</code> |
| | Send signal <i>SIGTERM</i> (sent by default) to the process with <i>pid 8888</i> | <code>kill -SIGTERM 8888</code> |
| | Send <i>SIGKILL</i> (force terminate the process) to the process with <i>pid 8888</i> | <code>kill -9 8888</code> |
| | Stop all processes named <i>node</i> | <code>killall node</code> |
| | Display processes whose parent is process with <i>pid 3607</i> | <code>pgrep -P 3607</code> |
| | Display the processes that opened the file <i>/etc/hosts</i> | <code>lsuf /etc/hosts</code> |
| | Find the process which took port 80 | <code>lsuf -i :80</code> |
| Running processes in the background | Run the process in the background | <code>ping google.com &</code> |
| | Run several processes in the background | <code>ping google.com & nmap 192.168.1.* &</code> |
| | Display a list of background processes | <code>jobs -l</code> |
| | Get access to the process (put it into priority mode) | <code>%1</code> |
| | Bring the process back to the background | <code>CTRL+Z</code> <code>%1 &</code> |
| | Start a new screen session | <code>screen</code> |
| | Transfer session to detached mode | <code>CTRL+A+D</code> |
| | Start the process in a new session in detached mode | <code>screen -d -m ping google.com</code> |
| | View the list of sessions | <code>screen -ls</code> |
| | Attach screen session | <code>screen -R [session id]</code> |
| Deal with the network | Run <i>HEAD</i> request (get headers only) | <code>curl -I http://google.com</code> |
| | Run <i>POST</i> request with data sending | <code>curl -d 'first_name=John&last_name=Doe' http://google.com</code> |
| | Send <i>JSON</i> to server | <code>curl -d '{"name":"John"}' -H 'content-type: application/json' http://google.com</code> |
| | Download file (similar to using <i>wget</i>) | <code>curl -O http://google.com/1.png</code> |
| | Track packets that were sent from the local machine | <code>tcpdump src 127.0.0.1</code> |
| | Track packets that came to the local machine through a specific network interface | <code>tcpdump dst 127.0.0.1 -i eth0</code> |
| | Display a list of network interfaces | <code>tcpdump --list-interfaces</code> |
| | Track packages that are gone from the local machine to <i>google.com</i> | <code>tcpdump src 127.0.0.1 and dst google.com -n</code> |
| | Display captured packets in <i>ASCII</i> | <code>tcpdump dst google.com and port 80 -A</code> |
| | Track packets to a specific <i>IP</i> and <i>port</i> | <code>tcpdump dst google.com and port 443 -n</code> |
| | Scan a specific server | <code>nmap -sP 217.160.0.201</code> |
| | Scan local network | <code>nmap -sP 192.168.1.*</code> |
| | Try to determine the server OS | <code>nmap -O 192.168.1.8</code> |
| | Scan server ports (use <i>-sV</i> to determine the version of the service) | <code>nmap -Pn 192.168.1.8</code> |
| | Run a quick scan (scanning of the common services) | <code>nmap -F 192.168.1.8</code> |
| | Scan specific ports (use <i>-open</i> to display only open ports) | <code>nmap -Pn -p 80,443 192.168.1.8</code> |
| Scan port combination (<i>U</i> – UDP port, <i>T</i> – TCP port, <i>21-25</i> – port range) | <code>nmap -p U:53,111,137,T:21-25,80,139,8080 192.168.1.1</code> | |
| Scan 10 top ports (<i>ssh, ftp, http ...</i>) | <code>nmap -Pn --top-ports 10 192.168.1.8</code> | |



Linux utilities that every developer should know (part 3)

| | | |
|-------------------------|---|---|
| System monitoring | Launch interactive process monitor | top htop |
| | Run I/O monitor | iostat |
| | Display space usage | du /path/to/directory |
| | Analyze space usage | ncdu /path/to/directory |
| View system performance | Display OS name and version | lsb_release -a |
| | See the full list of all devices | lshw |
| | See processor information | lscpu |
| | Display RAM Information | free -h |
| | See information about all mount points | df -h |
| | Display information about all available block devices | lsblk |
| Other utilities | Start the process with update every 500 ms | watch -n0.5 ls -laS |
| | Reset terminal | reset |
| | Display the calendar for the current year | cal -j |
| | Display calendar for June 2021 | cal -j 6 2021 |
| | Display information about the current user | id |
| | Display current user name | whoami |
| | Convert string to base64 | echo Hello base64 |
| | Decode base64 string | echo "SGVsbG8K" base64 -d |
| | Format json | echo '{"status":1}' jq curl https://opinionated-quotes-api.gigalixirapp.com/v1/quotes jq |

