Linux & Git cheat sheet

Sachin Singh
@sachin_Singh

📁 File Operations:

- **ls:** Lists all files and directories in the present working directory
- **ls -R:** Lists files in sub-directories as well
- **ls -a:** Shows hidden files
- **ls -al:** Lists files and directories with detailed information 📄📁
- **cd** directoryname: Changes the directory 📁
- **cd ..**: Moves one level up ↗
- **pwd**: Displays the present working directory 🗺️
- **cat > filename**: Creates a new file ✏️📄
- **cat filename**: Displays the file content 📄👀
- **touch filename**: Creates or modifies a file 🛠️📄
- **rm filename**: Deletes a file ❌📄
- **cp source destination:** Copies files from source to destination ➡️📄
- **mv source destination**: Moves files from source to destination 🔄📄
- **find / -name filename**: Finds a file or a directory by its name 🔍📄📁
- **file filename**: Determines the file type 📄❓
- **less filename**: Views the file content page by page 📄📄
- **head filename**: Views the first ten lines of a file 📄TOP
- **tail filename**: Views the last ten lines of a file 📄END
- **du -h --max-depth=1**: Shows the size of each directory 📁📏

📁 Directory Operations:

- **mkdir directoryname**: Creates a new directory 📁🆕
- **rmdir directoryname**: Deletes a directory ❌📁
- **cp -r source destination**: Copies directories recursively ➡️📁
- **mv olddir newdir**: Renames directories 🔄📁
- **find / -type d -name directoryname**: Finds a directory starting from root 🔍📁

🔄 Process Operations:

- **ps**: Displays your currently active processes 🔄
- **top**: Displays all running processes 💻🔄
- **kill pid**: Kills the process with given pid ❌🔄

- **pkill name**: Kills the process with the given name ❌🔄
- **bg**: Resumes suspended jobs without bringing them to foreground ▶️🔄
- **fg**: Brings the most recent job to foreground ▶️
- **renice +n [pid]**: Change the priority of a running process 🔄⚙️
- **&>filename**: Redirects both the stdout and the stderr to the file 🔄📄
- **1>filename**: Redirect the stdout to file 🔄📄
- **2>filename**: Redirect stderr to file 🔄📄

🔒 File Permissions:

- **chmod octal filename:** Change the permissions of file to octal 🔐📄
- **chown ownername filename:** Change file owner 👤📄
- **chgrp groupname filename:** Change group owner 👥📄

🔗 Networking:

- **ping host:** Ping a host and outputs results 🌐
- **netstat -pnltu:** Display various network related information 🔌🌐
- **ssh user@host:** Remote login into the host as user 🌐
- **wget url:** Download files from the web 🌐⬇️
- **curl url:** Sends a request to a URL and returns the response 🌐🔃

🗄️ Archives and Compression:

- **tar cf file.tar files:** Create a tar named file.tar containing files 📦📄
- **tar xf file.tar:** Extract the files from file.tar 📦📂
- **gzip file:** Compresses file and renames it to file.gz 📄➡️📄.gz
- **gzip -d file.gz:** Decompresses file.gz back to file 📄.gz➡️📄
- **zip -r file.zip files:** Create a zip archive named file.zip 📦📄
- **unzip file.zip:** Extract the contents of a zip file 📦📂

🔍 Text Processing:

- **grep pattern files:** Search for pattern in files 🔍📄
- **grep -r pattern dir:** Search recursively for pattern in dir 🔍📂
- **echo 'text':** Prints text 🗣️
- **sed 's/string1/string2/g' filename:** Replaces string1 with string2 in filename 🔀📄
- **di file1 file2:** Compares two files and shows the di erences ↔️📄
- **wc filename:** Count lines, words, and characters in a file 📄🔢
- **awk:** A versatile programming language for working on files 📄🐺
- **sed -i 's/string1/string2/g' filename:** Replace string1 with string2 in filename 🔀📄 (In-place edit)
- cut -d':' -f1 /etc/passwd: Cut out the first field of each line in /etc/passwd, using colon as a field delimiter 🔪📄

💾 Disk Usage:

- **df:** Shows disk usage 📀📏
- **du:** Shows directory space usage 📂📏
- **free:** Show memory and swap usage 📈🔀

- **whereis app**: Show possible locations of app 🔍📂

💻 System Info:

- **date:** Show the current date and time 📅🕒
- **cal:** Show this month's calendar 📅
- **uptime:** Show current uptime ⏱️
- **w:** Display who is online 👤
- **uname -a:** Show kernel information 💻🔍

📦 Package Installations:

- **sudo apt-get update:** Updates package lists for upgrades 🔄📦
- **sudo apt-get upgrade:** Upgrades all upgradable packages 🔄📦
- **sudo apt-get install pkgname:** Install pkgname 📦
- **sudo apt-get remove pkgname:** Removes pkgname 📦❌

🔧 Others (mostly used in scripts):

- **command | grep pattern:** Pipe the output of command to grep for searching ⏩🔍

🔍 Search and Find:

- **locate filename:** Find a file by its name 🔍📄
- **whereis programname:** Locate the binary, source, and manual page files for a command 🔍📂
- **which commandname:** Shows the full path of (shell) commands 🔍🚀

📦 Compression / Archives:

- **tar -cvf archive.tar dirname/:** Create a tar archive 📦📂
- **tar -xvf archive.tar:** Extract a tar archive 📦📄
- **tar -jcvf archive.tar.bz2 dirname/:** Create a compressed bz2 archive 📦📂
- **tar -jxvf archive.tar.bz2:** Extract a bz2 archive 📦📄

💻 Shell Scripting:

- **#!/bin/bash:** Shebang line to specify the script interpreter 🔄🔽
- **$0, $1, ..., $9, ${10}, ${11}:** Script arguments 🔄🔢
- **if [condition]; then ... fi:** if statement in bash scripts 🔄🔒
- **for i in {1..10}; do ... done:** for loop in bash scripts 🔄🔁
- **while [condition]; do ... done:** while loop in bash scripts 🔄🔁
- **function name() {...}:** Define a function 🔄🎯

🔧 Environment Variables:

- **env:** Display all environment variables 🌐🔧
- **echo $VARIABLE:** Display the value of an environment variable 🗣️🔧
- **export VARIABLE=value:** Set the value of an environment variable 🔄🔧
- **alias new_command='old_command options':** Create a new command that executes the old command with the specified options 🔤🆕🔧

- **echo $PATH:** Print the PATH environment variable🗣️📂🔧
- **export PATH=$PATH:/new/path:** Add /new/path to the PATH🔄📂🔧

🔄 Others (mostly used in scripts):

- **command1 ; command2:** Run command1 and then command2 🔄🔁
- **command1 && command2:** Run command2 if command1 is successful 🔄✅
- **command1 || command2:** Run command2 if command1 is not successful 🔄❌
- **command &:** Run command in background 🔄🕛

Remember, you can always use the **man command** (**e.g., man ls**) to get more information about each command. Happy coding! 🚀

🖥️ System Monitoring and Performance:

- **iostat:** Reports Central Processing Unit (CPU) statistics and input/output statistics for devices, partitions, and network filesystems 🔄📈
- **vmstat**: Reports information about processes, memory, paging, block IO, traps, disks, and CPU activity 🔄📈
- **htop**: An interactive process viewer for Unix systems 🔄💻

📀 Disk Usage:

- **dd if=/dev/zero of=/tmp/output.img bs=8k count=256k:** Create a file of a certain size for testing disk speed 📀🚓
  - This command will create a file named output.img in the /tmp directory with a size of approximately 2 GB (8 KB * 256 KB). It will be filled with zeros.

example:

$ dd if=/dev/zero of=/tmp/output.img bs=8k count=256k
262144+0 records in
262144+0 records out

2147483648 bytes (2.1 GB, 2.0 GiB) copied, 2.18953 s, 981 MB/s

Note: The output of the dd command may vary based on your system's performance.

- **hdparm -Tt /dev/sda:** Measure the read speed of your hard drive 📀📶
  - This command will measure the read speed of your hard drive (/dev/sda) and display the result in MB/sec.

example:

$ **sudo hdparm -Tt /dev/sda**

/dev/sda:

Timing cached reads: 20124 MB in 2.00 seconds = 10076.39 MB/sec
Timing bu ered disk reads: 588 MB in 3.01 seconds = 195.47 MB/sec

💡 Others:

- **yes > /dev/null &**: Use this command to push a system to its limit 💡🏁
  - Running this command will continuously print the letter "y" and redirect the output to /dev/null, which discards it. The process will run in the background.
- **example:**

$ yes > /dev/null &
[1] 12345
The number in brackets [1] and the PID 12345 represent the background job's ID and process ID, respectively.

- **:(){ :|:& };:**: A fork bomb – handle with care. Do not run this command on a production system 💣🚫
  - :(){ :|:& };:
    Warning: Do not run this command on a production system or any system you care about. It is a fork bomb and can quickly consume system resources, causing the system to become unresponsive or crash.

A fork bomb is a self-replicating program that creates a large number of child processes, overwhelming the system. It can be executed as follows:
$:(){ :|:& };:

If you accidentally run this command, you may need to restart your system to recover from its e ects.

🗓️ Cron Jobs:

- **crontab -l**: List all your cron jobs 🗒️🕰️
- **crontab -e**: Edit your cron jobs ✏️🗒️
- **crontab -r**: Remove all your cron jobs ❌🗒️
- **crontab -v**: Display the last time you edited your cron jobs 🕰️🗒️
- **crontab file:** Install a cron job from a file 🗒️🔄
- **@reboot command**: Schedule a job to run at startup 🗒️⏰🔃

Git Cheat sheet

🔄 Git Commands:

- **git init**: Initialize a local git repository 🔄📦
- **git clone url**: Create a local copy of a remote repository 🔄🔗📦

git add filename: Add a file to the staging area 🔄📄➕
git commit -m "Commit message": Commit changes with a message 🔄✍️📦
git status: Check the status of the working directory 🔄📦
git pull: Pull latest changes from the remote repository 🔄🔄📦
git push: Push changes to the remote repository 🔄🔄📦
git branch: List all local branches 🔄🌳
git branch branchname: Create a new branch 🔄🌳➕
git checkout branchname: Switch to a branch 🔄🌳⬅️➡️
git merge branchname: Merge a branch into the active branch 🔄🌳🔄
git stash: Stash changes in a dirty working directory 🔄🔒
git stash apply: Apply changes from a stash 🔄🔒 ➡️
git log: View commit history 🔄📜
git reset: Reset your HEAD pointer to a previous commit 🔄⬅️BACK
git rm filename: Remove a file from version control 🔄📄❌
git rebase: Reapply commits on top of another base tip 🔄🔄

🔄 **Version Control (Git commands)**:

git revert: Create a new commit that undoes all of the changes made in a particular commit, then apply it to the current branch 🔄⬅️BACK📦
git cherry-pick commitID: Apply the changes introduced by some existing commits 🔄🍒📦