Shell Commands

Commands

- Scroll up and down for command in history log..

Commands

- **clear** command for cls

Commands

- history :list all
- history <count> ex history 10

Commands

- !! Repeats last command

Commands

- cat : to see content of file on screen without editor
  - ex: cat my.sh
- cat > filename
  - add data to file create + save data in a file
- cat old file > new file
  - content gets copied
- cat file2 >> file1
  - file1=file1+file2
- Cat file1 file2 file 3 >> final file

Commands

tail command

it's pretty similar to cat, but only print last 10 lines by default, but it can be changed by adding -n.

ex

tail -15 amar.txt

Commands

- head Command
- This one is complementary to the tail command. head outputs the first 10 lines of a text file, but you can set any number of lines you want to display with the -n flag:
  - head long.txt
  - head -n 5 long.txt

Commands

- **htop** command
- interactive process viewer that lets you manage your machine's resources directly from the terminal. note, it's not installed by default certain times.

Commands

- grep
- The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression (grep stands for global search for regular expression and print out).
- grep [options] pattern [files]

- **Options Description**
- **-c : This prints only a count of the lines that match a pattern**
- **-h : Display the matched lines, but do not display the filenames.**
- **-i : Ignores, case for matching**
- **-l : Displays list of a filenames only.**
- **-n : Display the matched lines and their line numbers.**
- **-v : This prints out all the lines that do not matches the pattern**
- **-e exp : Specifies expression with this option. Can use multiple times.**
- **-f file : Takes patterns from file, one per line.**
- **-E : Treats pattern as an extended regular expression (ERE)**
- **-w : Match whole word**
- **-o : Print only the matched parts of a matching line, with each such part on a separate output line.**
- **-A n : Prints searched line and nlines after the result.**
- **-B n : Prints searched line and n line before the result.**
- **-C n : Prints searched line and n lines after before the result.**

Commands

- pwd Command
- The pwd command stands for “print working directory,” and it outputs the absolute path of the directory you’re in.

Commands

- cd
  - Go to the home folder
    - cd
  - 2. Move a level up
    - cd ..
  - 3. Return to the previous directory
    - cd -

Commands

- cp Command
- It's so easy to copy files and folders directly in the Linux terminal that sometimes it can replace conventional file managers.
  - cp file_to_copy.txt new_file.txt
- You can also copy entire directories by using the recursive flag:
  - cp -r dir_to_copy/ new_copy_dir/
- Remember that in Linux, folders end with a forward slash (/).

Commands

- cd
  - Go to the home folder
    - cd
  - 2. Move a level up
    - cd ..
  - 3. Return to the previous directory
    - cd -

Commands

- rm Command
- Now that you know how to copy files, it’ll be helpful to know how to remove them.
- You can use the rm command to remove files and directories. Be careful while using it, though, because it’s very difficult (yet not impossible) to recover files deleted this way.
  - rm file_to_copy.txt
- If you want to delete an empty directory, you can use the recursive (-r) flag:
  - rm -r dir_to_remove/
- On the other hand, to remove a directory with content inside of it, you need to use the force (-f) and recursive flags:
  - rm -rf dir_with_content_to_remove/

Commands

- mv Command
- You use the mv command to move (or rename) files and directories through your file system.
- To use this command, you'd type its name with the source and destination files:
  - mv source_file destination_folder/
  - mv command_list.txt commands/
- To utilize absolute paths, you'd use:
  - mv /home/kinsta/BestMoviesOfAllTime ./
  - …where ./ is the directory you're currently in.
- You also can use mv to rename files while keeping them in the same directory:
  - mv old_file.txt new_named_file.txt

Commands

- mkdir Command
- To create folders in the shell, you use the mkdir command. Just specify the new folder's name, ensure it doesn't exist, and you're ready to go.
- For example, to make a directory to keep all of your images, just type:
  - mkdir images/
- To create subdirectories with a simple command, use the parent (-p) flag:
  - mkdir -p movies/2004/

Commands

- man Command
- Another essential Linux command is man. It displays the manual page of any other command (as long as it has one).
- To see the manual page of the mkdir command, type:
  - man mkdir

Commands

- apt, yum, pacman commands
- No matter which Linux distribution you're using, it's likely that you use package managers to install, update, and remove the software you use every day.
- You can access these package managers through the command line, and you'd use one or another depending on the distro your machine is running.
- The following examples will install GIMP, a free and open source software usually available in most package managers:
- 1. Debian-based (Ubuntu, Linux Mint)
  - sudo apt install gimp
- 2. Red Hat-based (Fedora, CentOS)
  - sudo yum install gimp
- 3. Arch-based (Manjaro, Arco Linux)
  - sudo pacman -S gimp

Commands

- ps Command
- With ps, you can take a look at the processes your current shell session is running. It prints useful information about the programs you're running, like process ID, TTY (TeleTYpewriter), time, and command name.

- ps

Commands

- kill Command
- It's annoying when a program is unresponsive, and you can't close it by any means. Fortunately, the kill command solves this kind of problem.
- Simply put, kill sends a TERM or kill signal to a process that terminates it.
- You can kill processes by entering either the PID (processes ID) or the program's binary name:
  - kill 533494
  - kill firefox

Commands

- ping Command
- ping is the most popular networking terminal utility used to test network connectivity. ping has a ton of options, but in most cases, you'll use it to request a domain or IP address:

- ping google.com

- ping 8.8.8.8

Commands

- passwd Command
- passwd allows you to change the passwords of user accounts. First, it prompts you to enter your current password, then asks you for a new password and confirmation.

- It's similar to any other change of password you've seen elsewhere, but in this case, it's directly in your terminal:

- passwd

Commands

- which Command
- The which command outputs the full path of shell commands. If it can't recognize the given command, it'll throw an error.
- For example, we can use this to check the binary path for Python and the Brave web browser:
  - which python
  - # /usr/bin/python
  - which brave
  - # /usr/bin/brave

Commands

- whoami Command
- The whoami command (short for "who am i") displays the username currently in use:
  - whoami
- You would get the same result by using echo and the environmental variable $USER:
  - echo $USER

Commands

- whatis Command
- whatis prints a single-line description of any other command, making it a helpful reference:
  - whatis python
  - # python (1) - an interpreted, interactive, object-oriented programming language
  - whatis whatis
  - # whatis (1) - display one-line manual page descriptions

Commands

- wc Command
- Wc stands for “word count,” and as the name suggests, it returns the number of words in a text file:
- wc long.txt
- # 37 207 1000 long.txt

Commands

- uname Command
- uname(short for “Unix name”) prints the operative system information, which comes in handy when you know your current Linux version.
- Most of the time, you’ll be using the -a (–all) flag, since the default output isn’t that useful:
  - uname
    - # Linux
  - uname -a
    - # Linux kinstamanjaro 5.4.138-1-MANJARO #1 SMP PREEMPT Thu Aug 5 12:15:21 UTC 20

Commands

- neofetch Command
- Neofetch is a CLI (command-line interface) tool that displays information about your system — like kernel version, shell, and hardware — next to an ASCII logo of your Linux distro:

Commands

- find Command
- The find command searches for files in a directory hierarchy based on a regex expression. To use it, follow the syntax below:

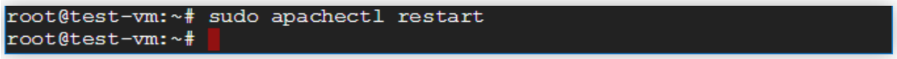
- find [flags] [path] -name [expression]
- To search for a file named long.txt in the current directory, enter this:

- find ./ -name "long.txt" # ./long.txt
- To search for files that end with a .py (Python) extension, you can use the following command:

- find ./ -type f -name "*.py" ./get_keys.py ./github_automation.py ./binarysearch

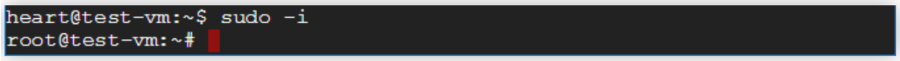
Commands

sudo



```
root@test-vm:~# sudo apachectl restart
root@test-vm:~#
```

The **`sudo`** command is used to run a single command with root permissions. In the example above, the command **`sudo`** is used before the command **`apachectl restart`** in order to grant the user the level of permissions required to restart the Apache server.

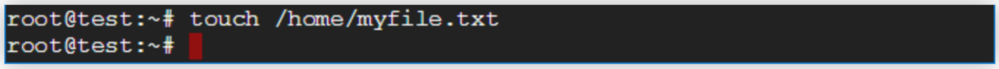


```
heart@test-vm:~$ sudo -i
root@test-vm:~#
```

In the example above, the **`sudo -i`** command is used to instill root permissions to the user for the entire duration of the SSH session. Notice how the username changes to 'root'.

Commands

touch

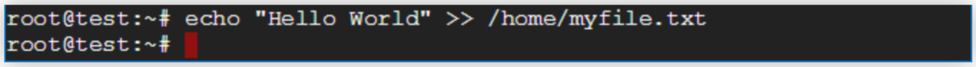


```
root@test:~# touch /home/myfile.txt
root@test:~#
```

The command `touch` is used to create new files. In the example above, the command `touch` is used to create a new file called myfile.txt in the home directory.

Commands

echo

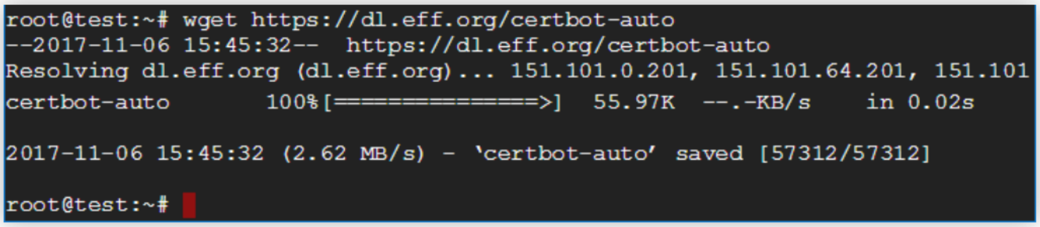


```
root@test:~# echo "Hello World" >> /home/myfile.txt
root@test:~#
```

The command `echo` is used to insert text into an existing file. In the example above, the command `echo "Hello World" >> /home/myfile.txt` is used to insert the text "Hello World" into the file called myfile.txt.

Commands

wget

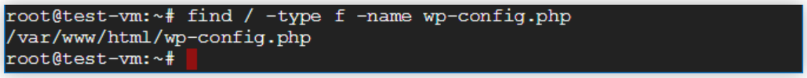


```
root@test:~# wget https://dl.eff.org/certbot-auto
--2017-11-06 15:45:32--  https://dl.eff.org/certbot-auto
Resolving dl.eff.org (dl.eff.org)... 151.101.0.201, 151.101.64.201, 151.101
certbot-auto        100%[=============>]  55.97K  --.-KB/s    in 0.02s

2017-11-06 15:45:32 (2.62 MB/s) - 'certbot-auto' saved [57312/57312]

root@test:~#
```

The command `wget` is used to download repositories to your Linux machine. In the example above, the command `wget https://dl.eff.org/certbot-auto` is used to retrieve the certbot-auto repository from eff.org and download it to my virtual machine.

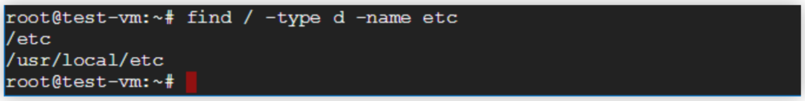
Commands

find



```
root@test-vm:~# find / -type f -name wp-config.php
/var/www/html/wp-config.php
root@test-vm:~#
```

FOR FILE: The command `find` is used to locate a file or directory. In the example above, the command `find / -type f -name wp-config.php` is used to locate the file called wp-config.php.

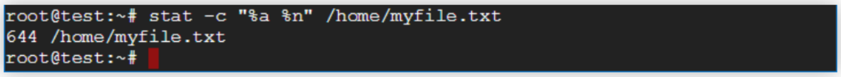


```
root@test-vm:~# find / -type d -name etc
/etc
/usr/local/etc
root@test-vm:~#
```

FOR DIRECTORY: In the example above, the command `find / -type d -name etc` is used to locate the directory called etc.

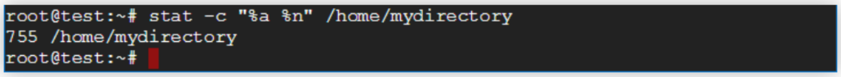
Commands

stat



```
root@test:~# stat -c "%a %n" /home/myfile.txt
644 /home/myfile.txt
root@test:~#
```

FOR FILE: The command `stat` is used to retrieve information about a file or directory. In the example above, the command `stat -c "%a %n" /home/myfile.txt` is used to find the permission level of the file called myfile.txt.

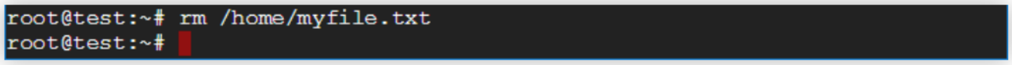


```
root@test:~# stat -c "%a %n" /home/mydirectory
755 /home/mydirectory
root@test:~#
```

FOR DIRECTORY: In the example above, the command `stat -c "%a %n" /home/mydirectory` is used to determine the permission level of the file called mydirectory.

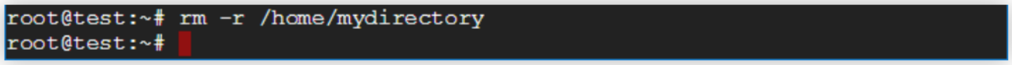
Commands

rm



```
root@test:~# rm /home/myfile.txt
root@test:~#
```

FOR FILE: The `rm` command is used to delete a file or directory. In the example above, the command `rm /home/myfile.txt` is used to delete the file called myfile.txt from the home directory.

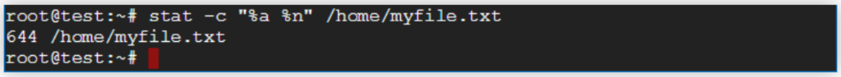


```
root@test:~# rm -r /home/mydirectory
root@test:~#
```

FOR DIRECTORY: In the example above, the `command rm -R /home/mydirectory` is used to delete the directory called mydirectory and all of it's contents.

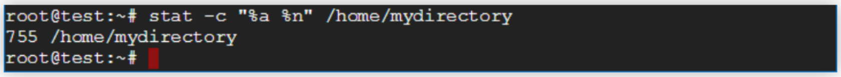
Commands

stat



```
root@test:~# stat -c "%a %n" /home/myfile.txt
644 /home/myfile.txt
root@test:~#
```

FOR FILE: The command `stat` is used to retrieve information about a file or directory. In the example above, the command `stat -c "%a %n" /home/myfile.txt` is used to find the permission level of the file called myfile.txt.

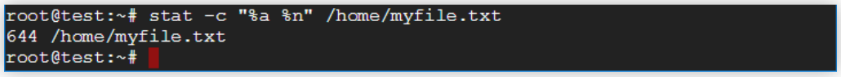


```
root@test:~# stat -c "%a %n" /home/mydirectory
755 /home/mydirectory
root@test:~#
```

FOR DIRECTORY: In the example above, the command `stat -c "%a %n" /home/mydirectory` is used to determine the permission level of the file called mydirectory.

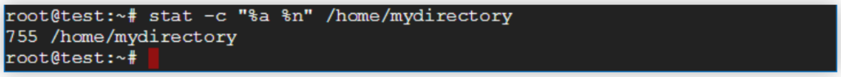
Commands

stat



```
root@test:~# stat -c "%a %n" /home/myfile.txt
644 /home/myfile.txt
root@test:~#
```

FOR FILE: The command `stat` is used to retrieve information about a file or directory. In the example above, the command `stat -c "%a %n" /home/myfile.txt` is used to find the permission level of the file called myfile.txt.



```
root@test:~# stat -c "%a %n" /home/mydirectory
755 /home/mydirectory
root@test:~#
```

FOR DIRECTORY: In the example above, the command `stat -c "%a %n" /home/mydirectory` is used to determine the permission level of the file called mydirectory.