ROBOTICS CORNER

botics Corner

botics Corner

Linux

Linux is an open Source operating system that is widely used in most of the companies.

01 File archiving and compression (tar, gzip)

02 Text editors and text manipulation

03 Processes and process management

04 Running and managing background processes

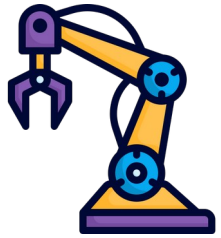
05 Redirecting input and output

ROBOTICS CORNER

Linux

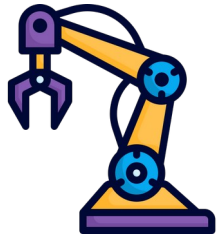
File Operations and Processes

Linux



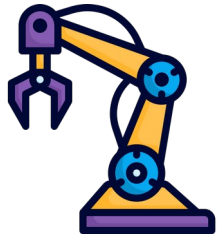
File archiving and compression

- To compress a file in Linux: tar -zcvf file.tar.gz [PATH]
- ex: tar -zcvf file.tar.gz file.cpp
- -c: Create a new archive.
- -z: Compress the archive using gzip.
- -v: Verbose mode (optional, shows the progress).
- -f: Specifies the archive file name.



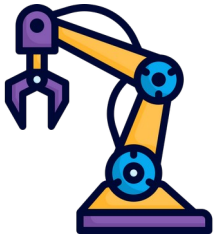
exclude directories and files

- Compress files but excluding files: tar -zcvf archive.tar.gz --exclude=’[DIRECTORY]’
- Compress files but excluding directories: tar -czvf /nfs/backup.tar.gz --exclude=" DIRECTORY " /home/vivek/



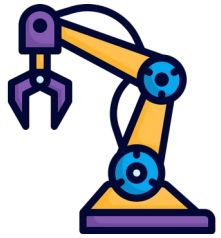
Extract a file

- tar -xzvf file.tar.gz
- tar -xjvf file.tar.bz2
- extract the contents of the archive into a specific directory: tar -xzvf my.tar.gz -C [DIRECTORY]
- tar -xjvf archive.tar.bz2 -C [DIRECTORY]



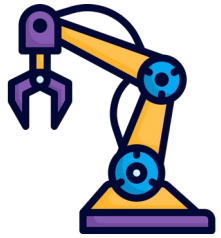
Extract a file

- -x : Extract files from an archive
- -t : List the contents of an archive
- -v : Verbose output
- -f file.tar.gz : Use archive file
- -C DIR : Change to DIR before performing any operations
- --exclude : Exclude files matching PATTERN/DIR/FILENAME



Text Editors in Linux

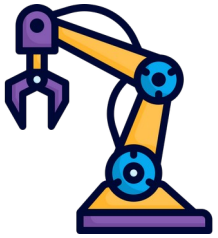
- Text Editor: nano
- If you already have a txt file: nano _text_file_name.txt
- CRTL + O ◊ to save
- CTRL + X ◊ to exit



Nano

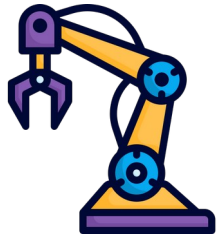
Flag	Description	Example
`-B`	Makes a backup of the current file before saving changes.	`nano -B myfile.txt`
`-I`	Enables automatic indentation.	`nano -I myfile.txt`
`-N`	No conversion from DOS/Mac format.	`nano -N myfile.txt`
`-T`	Sets the size of a tab to the given number of spaces.	`nano -T 4 myfile.txt`
`-U`	Enables undo functionality.	`nano -U myfile.txt`
`-Y`	Syntax highlighting.	`nano -Y sh myfile.sh`
`-c`	Constantly show the cursor position.	`nano -c myfile.txt`
`-i`	Automatically indents new lines.	`nano -i myfile.txt`
`-k`	Toggle cut so it cuts from cursor position.	`nano -k myfile.txt`
`-m`	Enable mouse support.	`nano -m myfile.txt`

ROBOTICS CORNER



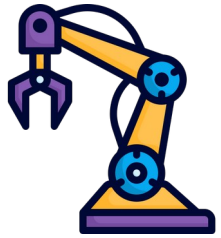
Other text editors

- While the nano command is a powerful and user-friendly text editor, Linux provides other text editors that might better suit your needs, particularly as you become more experienced. Two of the most popular alternatives are Vi and Emacs.



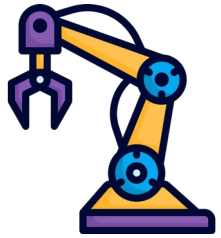
VI

- vi myfile.txt
- a. press 'i' to enter insert mode
- ex: This is some text.
- c. press 'Esc' to exit insert mode
- d. type ':wq' to save and quit
- e. Output: 'myfile.txt' 1L, 18C written



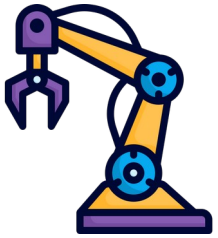
Nano vs. Vi vs. Emacs

Feature	Nano	Vi	Emacs
Learning Curve	Easy	Moderate	Hard
User Interface	Simple and intuitive	Modal, requires learning commands	Extensible, requires learning commands
Extensibility	Limited	Yes, with Vimscript	Yes, with Emacs Lisp
Accessibility	Universally available	Universally available	Not always installed by default



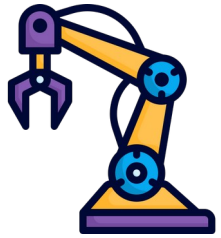
Basic text manipulation

- 1. grep
- 2. cat
- 3. less
- 4. head
- 5. tail



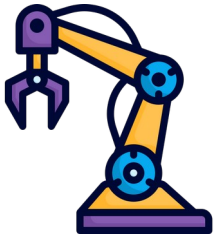
grep

- 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 printout).
- grep [options] pattern [files]
- [options]: These are command-line flags that modify the behavior of grep.
- [pattern]: This is the regular expression you want to search for.



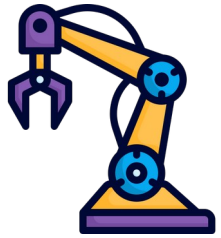
grep

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
-w	Match whole word



Cat, less

- The cat command is the simplest way to view the contents of a file. It displays the contents of the file(s) specified on to the output terminal. Used for simple and short files: cat FILENAME
- The less command is similar to the more command but provides extensive features. One important one is that it allows backward as well as forward movement in the file: less FILENAME
- Use g and G to go up or down, q to exit

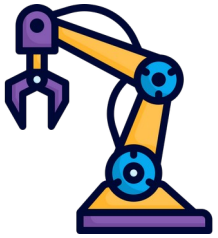


Head, tail

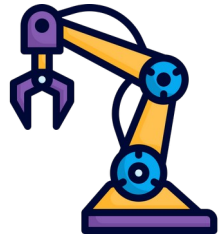
- As their names imply, the head command will output the first part of the file, while the tail command will print the last part of the file. Both commands write the result to standard output: head [OPTIONS] FILES
- Both allow the display of a certain number of lines
- tail [OPTIONS] FILES

ROBOTICS CORNER

Processes and process management

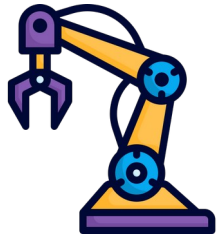


- Processes and process management are fundamental concepts in Linux and other operating systems. Understanding and effectively managing processes is crucial for the stability, performance, and security of a Linux system.
- A process is an instance of a program currently running on a computer system. In Linux, processes are managed by the operating system's kernel, which allocates system resources and schedules processes to run on the CPU. Understanding and managing processes is a critical skill for Linux administrators and developers.



ps

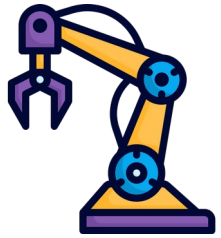
- Sometimes, we want to close unresponsive programs or check if a background process has started in our Linux system.
- The ps command in Linux is used for displaying information about processes. It provides a snapshot of the currently running processes on the system. The ps command has various options that allow you to customize the output according to your requirements: ps [options]
- PID – the process id
- TTY – terminal associated with the process
- TIME – elapsed CPU utilization time for the process
- CMD – the executable command



top, kill

- Top is used to list all running Linux processes on your system: top
- To kill a process: kill [PID]

04 Running and managing background processes

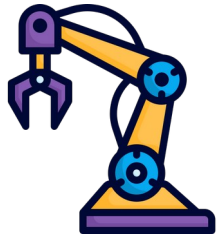


```
roboticscorner@linux: ~ 80x24
oboticscorner@linux:~$ sleep 50 &
5] 2960
oboticscorner@linux:~$ sleep 50 &
6] 2961
oboticscorner@linux:~$ sleep 50 &
7] 2962
oboticscorner@linux:~$ sleep 70 &
8] 2963
oboticscorner@linux:~$ fg 5
leep 50
```

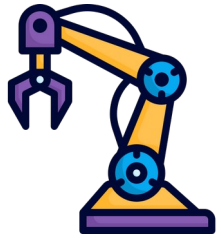
```
roboticscorner@linux: ~
roboticscorner@linux: ~ 80x24
roboticscorner@linux:~$ sleep 50 &
[5] 2960
roboticscorner@linux:~$ sleep 50 &
[6] 2961
roboticscorner@linux:~$ sleep 50 &
[7] 2962
roboticscorner@linux:~$ sleep 70 &
[8] 2963
roboticscorner@linux:~$ fg 5
sleep 50
^Z[1]   Done                    sleep 50
[2]   Done                    sleep 50
[3]   Done                    sleep 50
[4]   Done                    sleep 70

[5]+  Stopped                 sleep 50
roboticscorner@linux:~$
```

ROBOTICS CORNER

04 Running and managing background processes



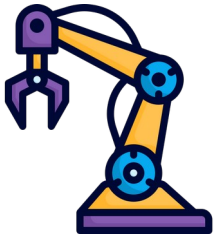
```
roboticscorner@linux: ~
roboticscorner@linux: ~ 80x24
roboticscorner@linux:~$ sleep 50 &
[5] 2960
roboticscorner@linux:~$ sleep 50 &
[6] 2961
roboticscorner@linux:~$ sleep 50 &
[7] 2962
roboticscorner@linux:~$ sleep 70 &
[8] 2963
roboticscorner@linux:~$ fg 5
sleep 50
^Z[1]   Done                    sleep 50
[2]   Done                    sleep 50
[3]   Done                    sleep 50
[4]   Done                    sleep 70

[5]+  Stopped                 sleep 50
roboticscorner@linux:~$ jobs
[5]+  Stopped                 sleep 50
[6]   Done                    sleep 50
[7]   Done                    sleep 50
[8]-  Running                 sleep 70 &
roboticscorner@linux:~$
```



echo

- The echo command is a way to communicate with your Linux terminal. It allows you to send text, variables, and special characters to the standard output, which is usually the terminal screen. The echo command is like a messenger that delivers your words to the terminal.

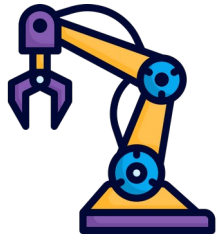
Redirecting input and output



- The ‘>‘ symbol is used for output (STDOUT) redirection overriding the file: > : ls –al > record.txt
- If the contents of the file are to be added and not overwritten: >> : ls –n text >> record.txt
- The ‘<‘ symbol is used for input(STDIN) redirection : Mail -s "Subject" to-address < Filename

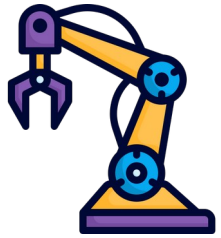
ROBOTICS CORNER

Redirecting input and output



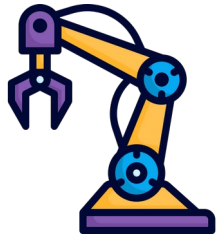
|

- In Linux, the vertical bar (|) is known as a "pipe" and is used to connect the output of one command to the input of another. This allows you to create a pipeline of commands, where the output of one command becomes the input for the next command. The | symbol is commonly used in the command line to enhance the functionality and flexibility of Linux commands.



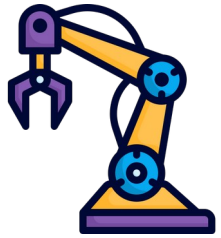
Word count

- wc [file]: returns is the file's number of lines, words, and characters, followed by its path.



Diff command

- Its primary purpose is to compare the contents of two files and display their differences. diff [FILE1] [FILE2]
- Can compare 2 versions of the same file



Thank You

Do you have any questions?

ROBOTICS CORNER

ROBOTICS CORNER

01211626904

www.roboticscorner.tech

Robotics Corner

Robotics Corner

Robotics Corner

Robotics Corner