Crio Sprint: JAVA-112

Session 1 - Introduction to Linux

Agenda

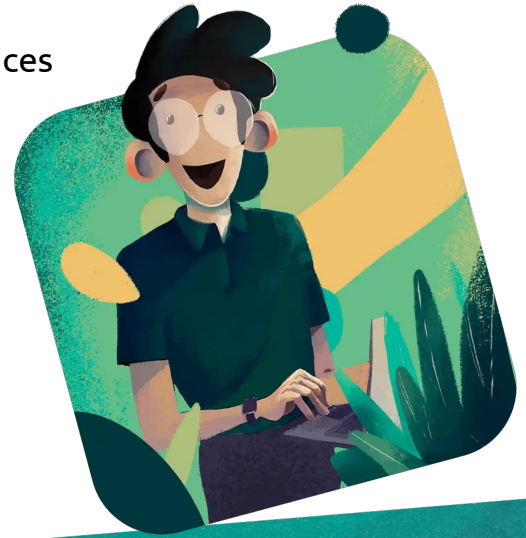
- Importance of command line
- Navigation using terminal
- Arguments and options
- Linux directory structure
- Absolute and relative path
- Working with files
- Bash script

Why Linux?

- It’s inevitable
- Good Skill for an IT professional
- Getting comfortable with command line gives you a head start

Real World - you are using linux servers on a daily basis, whether you realize it or not

- Cloud - Dropbox/Google drive, Other SaaS, Online streaming/gaming services
- Mobiles, TVs, Cars, Smartwatches, Routers, Supercomputers
- Almost all Enterprise machines

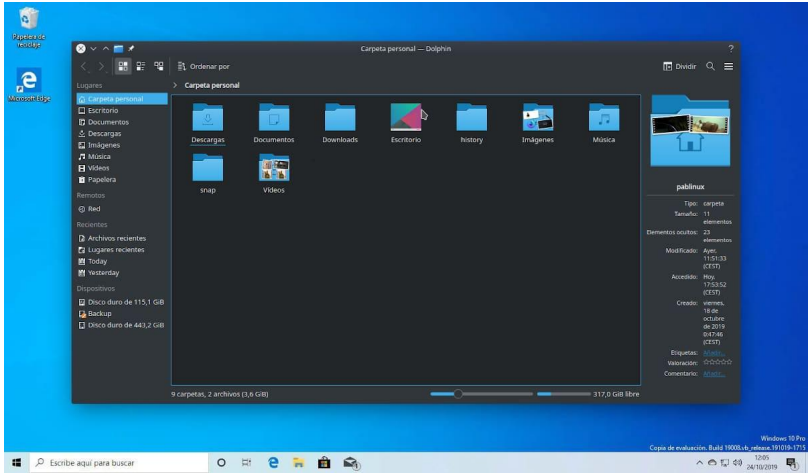


Command Line (Terminal)

- Text-based user interface
- All GUI (Graphical User Interface) based actions eventually translate to a command which gets executed
- The Terminal gives a direct way to run these commands
- Popular CLIs (Command Line Interface):
  - Powershell: Windows
  - Bash: Linux & MacOS

Activity: What are basic tasks you do on your Desktop?

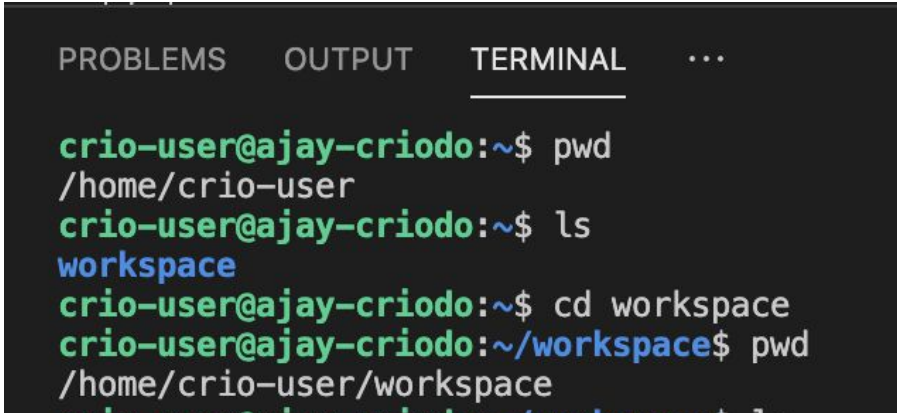
- Create a file or directory
- Delete a file or directory
- Copy/Paste a file or a directory
- Move a file



How would you do all these without GUI? 🤔

Some Directory related commands

1. **pwd**
   - **P**rint **W**orking **D**irectory
   - Displays the current directory you are in.
2. **ls**
   - List down files and sub-directories in a directory
3. **cd**
   - **C**hange **D**irectory
   - Change the current working directory



Code Demo: Navigation using terminal

Create a folder named **me_linux** inside

/home/crio-user/workspace/bytes/ → *using VScode UI.*

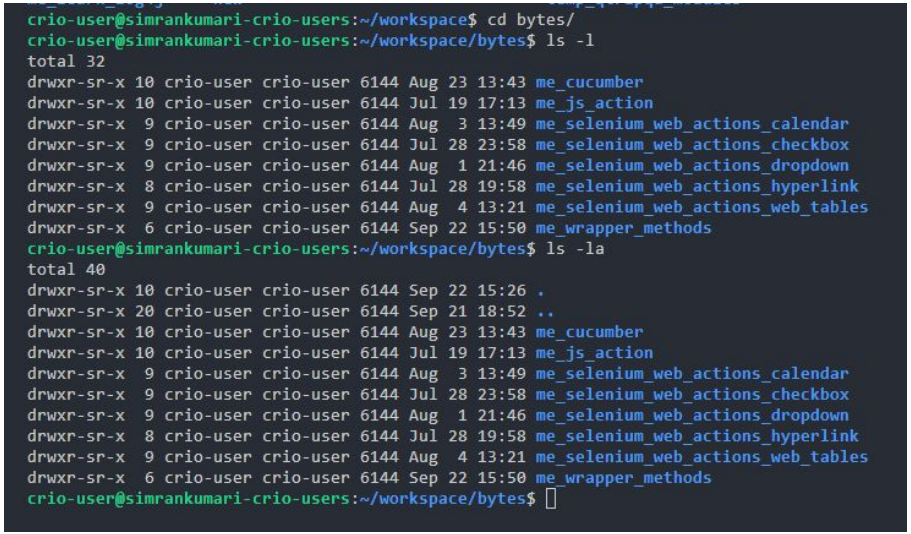
Carry out the following steps using the terminal:

1. Check if the folder is created.
2. Check whether we are in the created folder.
3. Move to the newly created folder.
4. Again check if we are in the created folder.

Argument vs Options

Arguments

- It is an input given as part of terminal commands.
- Example: `cd <directory_name>`
- There can be one or more arguments.



Options

- Options modifies the behavior of a command.
- Must start with a '-'; otherwise, it is assumed to be an argument.
- Example : `ls -l`
- We can add multiple options together.

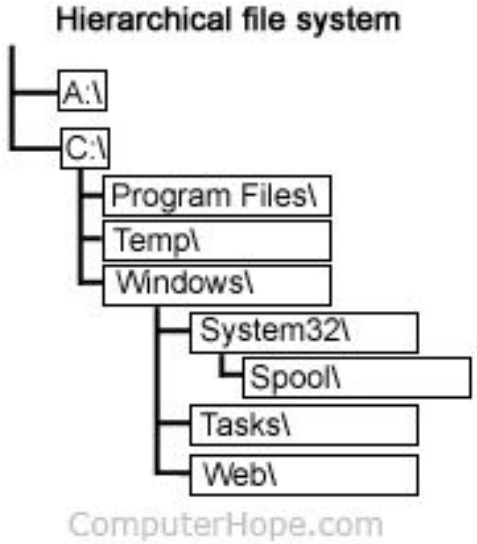
Code Demo: Useful Shortcuts

Try using the following shortcuts in the terminal window:

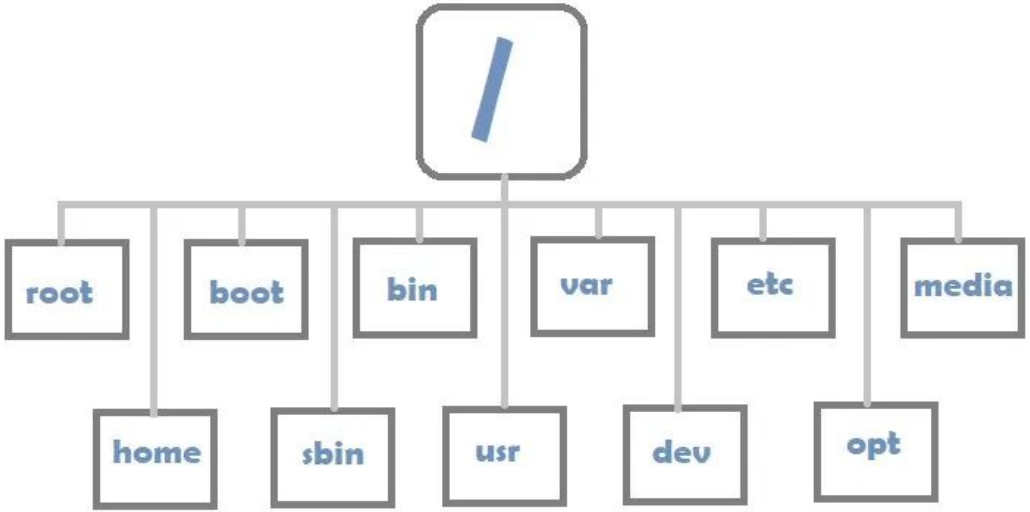
- **`up arrow` key** → will bring up the last command that was executed. Each press will bring up the previous command executed.
- **`history`** → will print out a list of the previous commands executed.
- **`tab` key** → pressing the tab key can be used to auto-complete the directory and file names while typing paths or filenames.

Windows file system

Looks like this:



Linux directory structure



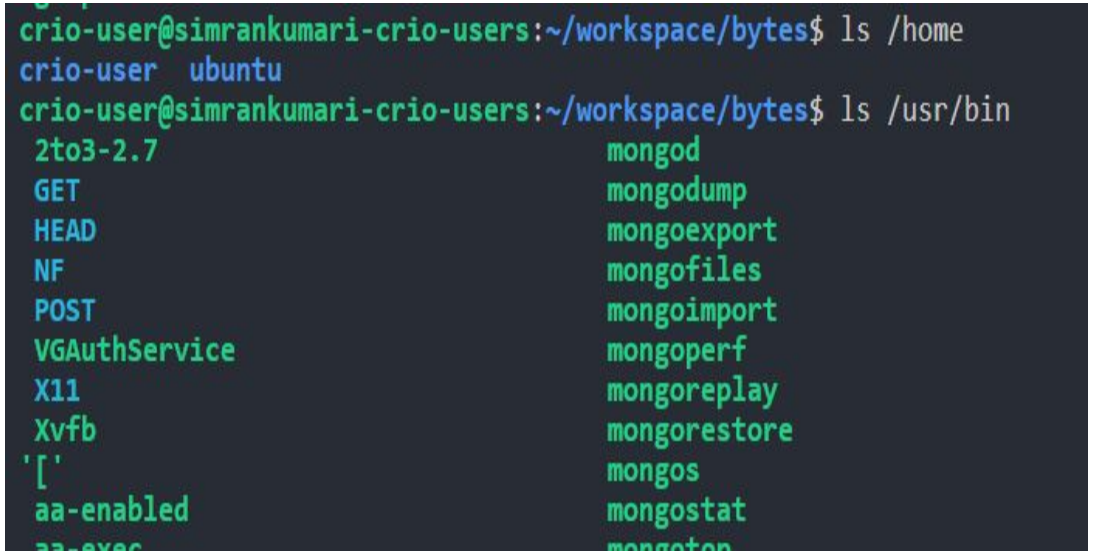
Know about some common files

Path	Notes
/	This is the root directory
/home	Where all the user home directories are saved
/boot	Files required to start (boot) a Linux machine
/bin	Contains the executable files
/var	Contains variable data like logs, databases, website content etc

Code demo: Try different file paths

Try to run the following commands in the terminal:

- ls
- ls /home
- ls /usr/bin
- ls /var



Absolute path

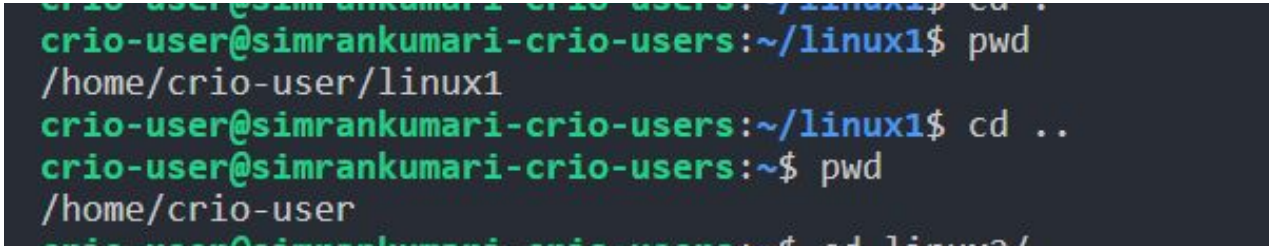
In the previous demo , we observed that :

- Irrespective of the present working directory, we ran `**ls /var**` and it listed out the content of `**/var**` folder.
- This is called using the *absolute path* where we provided the full path to a directory right from the root of the filesystem: `**/**`.

Activity: Figuring out relative path

Typing out the absolute path can be cumbersome at times.

Do we have any way to move to the parent folder, without using the absolute path?



Relative path

Relative paths are *relative* to the present working directory

Relative Path	Description	Example	Notes
.	present working directory	ls .	list contents of current directory
..	parent directory	cd ..	go one level up to the parent directory
-	previous working directory	cd -	go back to the previous working directory

Code Demo: Relative path and Absolute path

Using the VS Code Ui, create two folder linux1 and linux2 inside the **me_linux/**. Within each folder create two txt files. Now, use the terminal to try out the following steps.

1. Move to linux1 folder
2. Check the contents of linux2 folder using absolute path
3. Move to parent directory using relative path.
4. Move to previous working directory

Work with files using terminal part 1

4. mkdir

- Stands for “make directory”.
- It creates each directory specifed on the command line in the order given.

5 .touch

- It is used to create a file without any content i.e an empty file

6. cat

- It reads data from the file and gives their content as output.
- `-n` options prints the content along with line numbers

7. cp

- This command is used to copy the contents of one file to another.

Code Demo: Working with files using terminal

Use the terminal to execute the following steps.

1. Create a directory named dir1 inside **me_linux/**
2. Within the directory create an empty file name empty.txt
3. Create another file named hello.txt which contains “Hello world”

   using VS code UI.
4. Confirm the contents of the file
5. Copy the contents of hello.txt file to empty.txt file

Work with files using terminal part-2

8. mv

- mv stands for move.
- mv is used to move one or more files or directories from one place to another

9. rm

- rm stands for remove here.
- rm command is used to remove files.

10. rmdir

- rmdir stands for remove directory here.
- rm command is used to remove folders.

Code Demo: Working with files using terminal

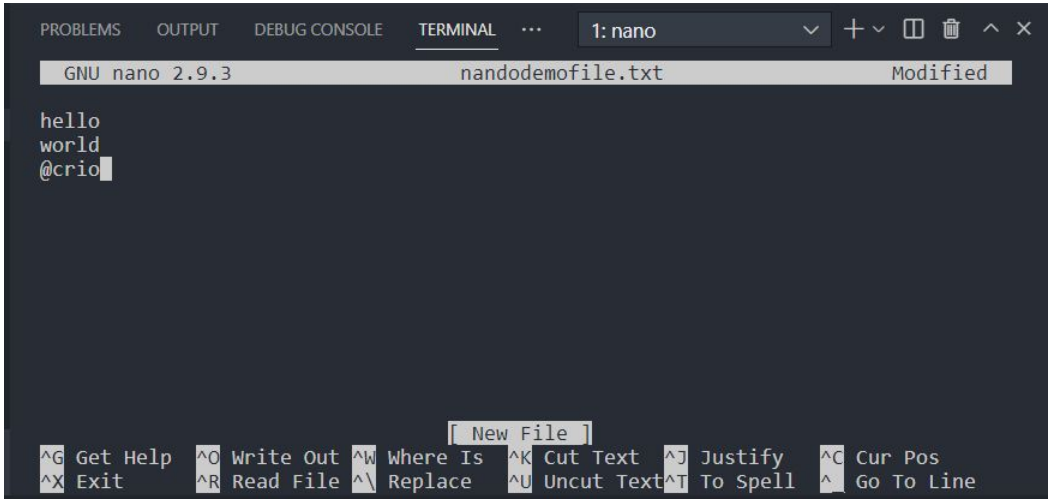
With respect to the previous activity , use the terminal to execute the

following steps.

1. Move the empty.txt to new directory dir2 inside **me_linux/**
2. Delete file empty.txt
3. Delete dir2

Nano

- It is a text editor
- It is used to create and open new file.
- To save a file press `ctrl + o`
- To exit the editor use `ctrl +x`



In windows, we prefer to use notepad.

Code Demo: Nano editor

- Create a new file using nano editor.
- Add contents to the file
- Save the file and exit
- Use cat command to check if the file content was saved.

Activity: Brainstorming for Bash Script

What if we want to run some specific commands multiple times? Like we have functions in programming, which can be executed n number of time by calling them, similarly we have bash scripts.

- Bash scripts are essentially just a sequence of the same Linux commands that you would ordinarily use iteratively.

Creating Bash Script

Step 1: First create a new file with `.sh` extension, say `hello_world.sh` using nano command.

Step 2: Add the content to the script file.

```
#!/bin/bash

#use an echo statement
echo “The basch script is executed successfully”
```

Step 3: Execute the script

./hello_world.sh

Note: echo command is used to print content

Code Demo: Create Bash script

Create and execute a bash script inside me_linux/ , which does the

following:

1. Check present working directory
2. Create new directory named dir_new1
3. Delete the newly created directory.
4. Keep adding echo statement after each step.

Note: on executing the script you may get an error as shown below, we will resolve that later.

```
crio-user@simrankumari-crio-users:~/workspace/bytes$ ./run.sh
bash: ./run.sh: Permission denied
```

Demo: Resolve error

Let’s try to resolve the error we got in previous code:

- The error has something to do with permissions.
- Remember when we used `ls` to print out data in the long format earlier, it was also showing the file

```
crio-user@simrankumari-crio-users:~/workspace/bytes$ ls -l
total 36
drwxr-sr-x 10 crio-user crio-user 6144 Aug 23 13:43 me_cucumber
drwxr-sr-x 10 crio-user crio-user 6144 Jul 19 17:13 me_js_action
drwxr-sr-x  9 crio-user crio-user 6144 Aug  3 13:49 me_selenium_web_actions_calendar
drwxr-sr-x  9 crio-user crio-user 6144 Jul 28 23:58 me_selenium_web_actions_checkbox
drwxr-sr-x  9 crio-user crio-user 6144 Aug  1 21:46 me_selenium_web_actions_dropdown
drwxr-sr-x  8 crio-user crio-user 6144 Jul 28 19:58 me_selenium_web_actions_hyperlink
drwxr-sr-x  9 crio-user crio-user 6144 Aug  4 13:21 me_selenium_web_actions_web_tables
drwxr-sr-x  6 crio-user crio-user 6144 Sep 22 15:50 me_wrapper_methods
-rw-r--r--  1 crio-user crio-user    3 Sep 23 13:55 run.sh
```

Activity: In-session

Problem: Print the contents of the Path environment variables in your system.

Activity: In-session

Answer: echo $PATH$ or echo %path% (Windows)

To be covered

We will cover more in the byte like:

- grep command
- Output redirection
- System and process

Byte Overview

Linux Basics

Byte breakdown - Linux Basics

This Byte gets you familiar with the basics of Linux

- **Milestone 1** - Navigation using Terminal
- **Milestone 2** - Arguments vs options
- **Milestone 3** - File system and files
- **Milestone 4** - File and folder operations
- **Milestone 5** - Bash Script
- **Milestone 6** - Miscellaneous commands
- **Milestone 7** - System and processes

Let’s take a short quiz

Take home exercises

Complete these Bytes

- Linux Basics QA

Disclaimer and References

All images belongs to respective owners.

Crio.Do

www.crio.do