w

Week 7 Practical Session 2023 - 2024 Computer Crime and Digital Evidence **Command Line Practical**

### Date Issued: 05th October 2023

Issued by Jay Murphy



**Contents**

1. [Aim and Objectives 2](#_TOC_250007)
2. [Introduction 2](#_TOC_250006)
3. [Task 1: Basic navigation commands 2](#_TOC_250005)
   1. [Identify yourself! 5](#_TOC_250004)
   2. [Absolute and relative paths 7](#_TOC_250003)
4. [Task 2: Learn Basic and more advanced commands 10](#_TOC_250002)
5. [Conclusions 11](#_TOC_250001)

[References 11](#_TOC_250000)

# Aim and Objectives

The aim of today’s lab session is to make you familiar with the use of the Terminal on Linux systems.

#### Objectives

By the end of this session you should be able to:

* + Navigate in a Unix hierarchical system using the command line.
  + Perform basic tasks like creating and moving folders and files.
  + Change file permissions.
  + Use more advanced commands (such as to redirect the output).
  + Know what a bash script is.

# Introduction

*Linux* has a graphical user interace (GUI) but in general it is considered that an advanced computer user should be familiar with the use of the command line (Terminal). You will use your UWE Linux VM during this lab session. You do not need the Forensic Windows VM today.

Launch the Linux VM and open the ‘Terminal’ (Figure 1).

**Disclaimer**: Screenshots in this lab sheet might differ from the view you will get in your VM due to the fact that you are using the most current Kali version.

# Task 1: Basic navigation commands

The command line interface establishes a text-based communication with the system.

Usually you issue *commands* to tell the computer what you want it to do.

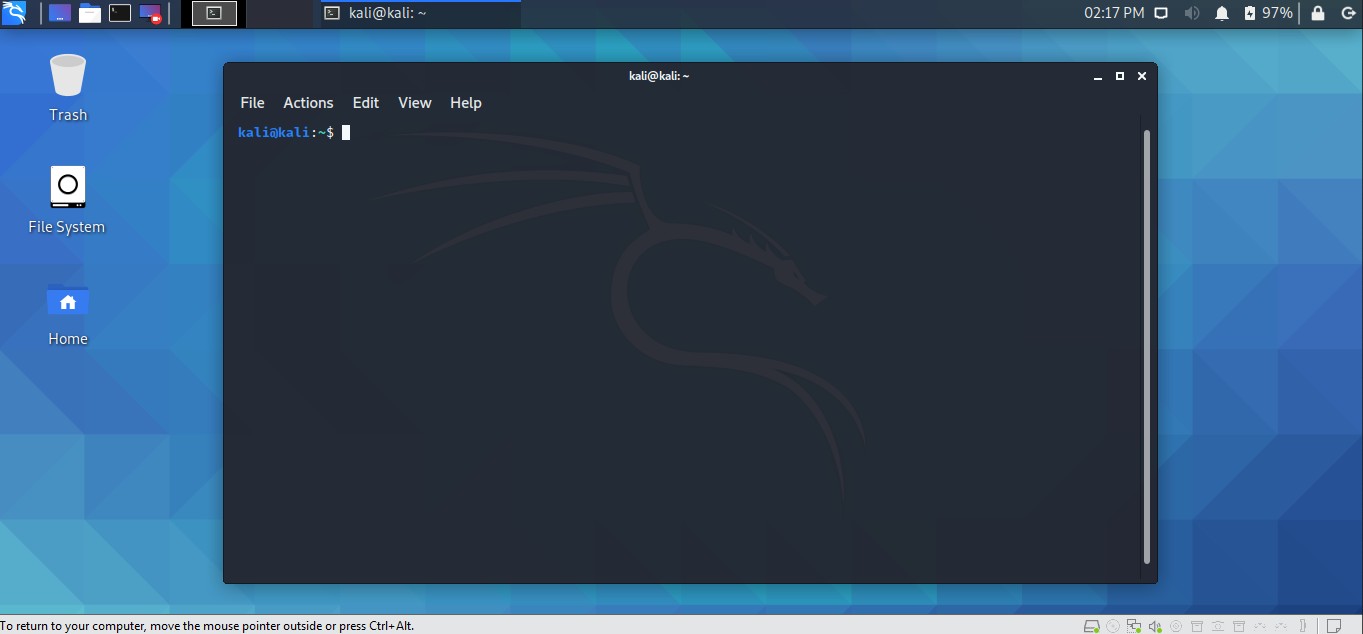


Figure 1: Open the Terminal.

Typically the first thing you type in the command line is the command, followed by a space and then a number of options (or *arguments*) which typically start with a dash (-).

Type *echo %SHELL* to see what happens. Now type *echo* $*SHELL* to see which SHELL is used in your system (Figure 2). The shell is a program that takes commands from the keyboard and gives them to the operating system to execute. Which shell are you using in this VM? Note that the use of any symbol can alter the outcome of a command (here % and $ gave a different result, because actually the $ symbol is a special symbol that is used to retrieve environment variables.)

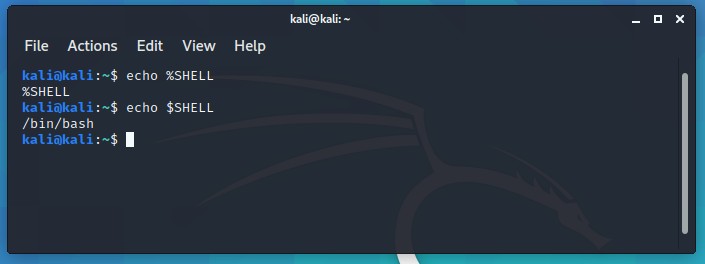


Figure 2: Which shell your system uses?

Use the up and down arrow keys to see previously typed commands and use them as shortcuts, instead of writing again the same commands.

*→*

*→* Use the *clear* command to clear your screen (Figure 3).

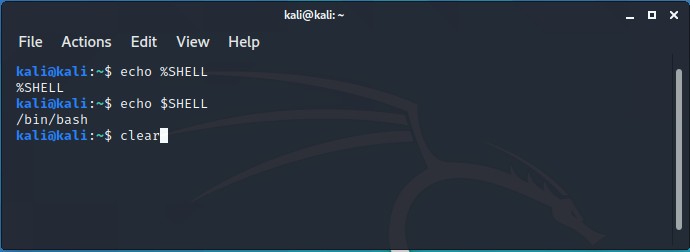


Figure 3: Type clear to clear your screen.

Use the tab key to automatically fill in your command. Hit the tab twice to see suggestions when you have started writing commands. For example, while you are at your current working directory, type *cd D* and hit tab twice. You will see which files/folders begin with the letter **D**.

*→*

Now type again *cd De* and hit the tab once. What happened? (Figure 4). Use tab to autofill your commands and save yourself some time and effort from typing!

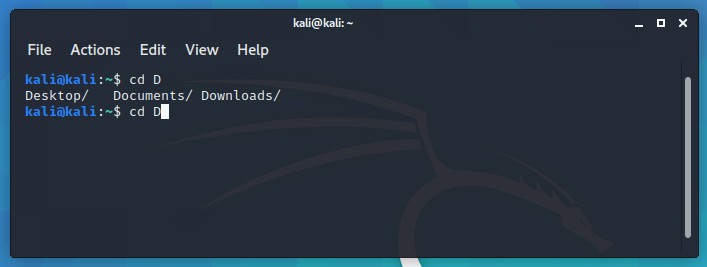


Figure 4: Use the arrow keys and the tab key as shortcuts.

## Identify yourself!

Type the following commands, one at a time, and keep a record (keep some notes) of what you get back from the system:

* + - whoami
    - logname
    - id
    - who

Use the Internet (i.e. a web search machine) to find out what these commands do...

Type *man [command]* to see the manual page of the [command] you are interested in. Type *man who* to get the manual page of the *who* command. This is a very useful command.

Now type the following commands, one at a time, and keep a record of what you get back from the system. You have probably already used extensively some of these commands. Use the Internat or the *man* pages to figure out what these commands do.

* + - cd
    - pwd
    - ls
    - ls -l

Note that when you use the ls command with the -l argument, you get the permissions, number of hard links, the owner and the group of the file, its size, the last-modified date and its filename (Figure 5).

If you type *ls -al* you will also see the hidden files in the current working directory (Figure 6). These files are marked with a period in front of their names.

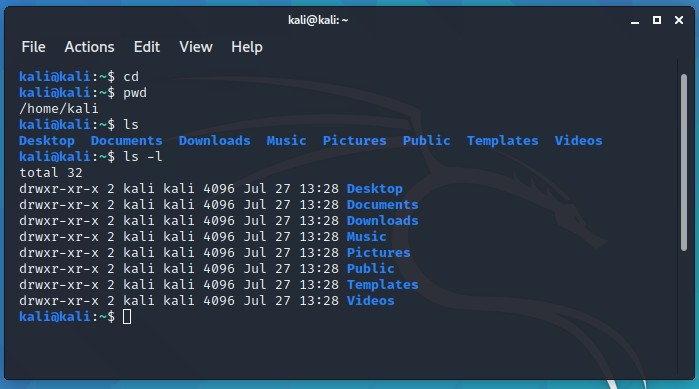


Figure 5: Basic commands.

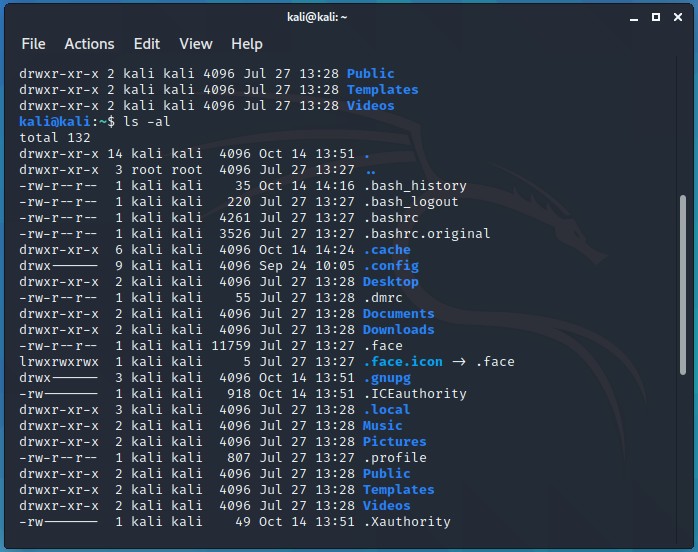


Figure 6: Hidden files/folders.

You can also see an entry with a filename as ‘.’ and another one with a filename as ‘..’. These are pointing to the current directory and the parent directory, respectively.

Type the following and notice how the single and the double dot work.

* + - cd
    - cd Desktop
    - cd .
    - cd ..

Type *ls -al* and then *cat .bash history* to see a list of commands that you previously used! That’s good information for a forensicator!

*→*

*→* Type *clear* to clear the screen.

## Absolute and relative paths

You will now see a tree structure of the root directory ‘/’. In order to do this you need to download the ‘tree’ utility. Type the following:

* + - cd
    - sudo apt update
    - sudo apt-get install tree

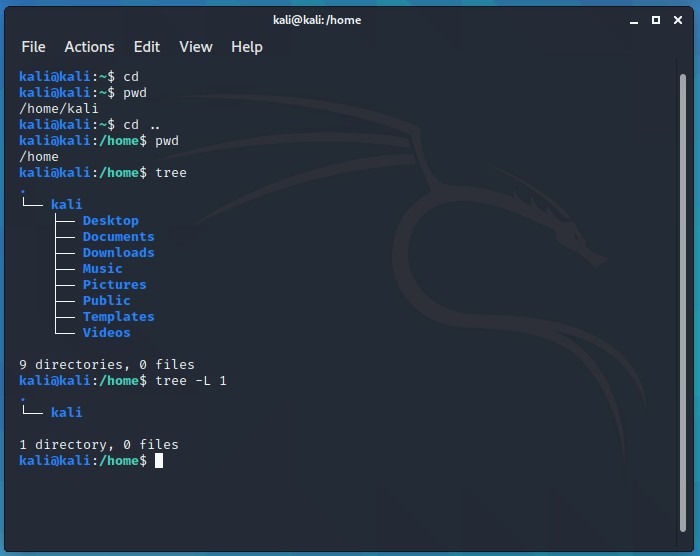


Figure 7: Tree command

Now type:

* + - cd
    - pwd
    - cd ..
    - pwd
    - tree
    - tree -L 1

You will notice (Figure 7) that if you don’t use the *-L 1* argument, then *tree*

will recursively show the contents of each folder in all levels. By using the

-L argument you are able to define the depth of your listing in the directory structure.

Now type the following (Figure 8):

* + - clear
    - cd
    - cd ..
    - tree -L 1

You are viewing the contents of the root directory /. Notice the *root* folder icon /.

Now, type the following:

* + - cd
    - pwd

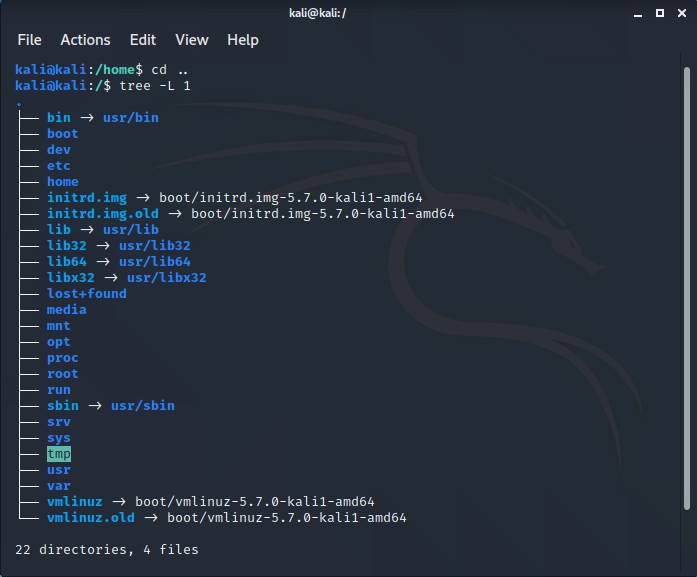


Figure 8: Hierachy under the ‘/’ directory.

Notice the ‘tilde’ symbol at the first line of your terminal in front of the cursor.

This indicates your ‘home’ directory.

The tilde points to the $HOME variable in UNIX (Linux). As a normal user on a Linux computer you should expect the tilde to point to a folder such as

/home/[username].

There exist two types of paths in Linux systems: Relative and Absolute paths.

*→*

*Relative path* is a file or directory location relative to where we currently are in the file system.

*Absolute path* is a file or directory location in relation to the root / of the file system.

*file* is a useful command because it shows you the type of the file that you used as an argument; if it is a text file, a folder, a binary etc...

*→*

Take a look at the files in the / directory. Be extra cautious because you don’t want to delete any files in this folder.

*→*

# Task 2: Learn Basic and more advanced commands

Now click on the following link and spend the rest of the time to do the tutorial. You can skip the ‘VI Text Editor’ part. *vi* is a primitive text editor and a lot of users find it intimidating. You can pick another Linux tutorial of your choice if you think that this one is not for you. You are not supposed to do the ‘Bonus material’ section of the proposed tutorial but you can try if you want. Try the scripting part!

https://ryanstutorials.net/linuxtutorial/aboutfiles.php

The link will land you on the 3rd Section of the tutorial. Try to go through as many sections as possible.

*You are advised to avoid interfering with the root directory (i.e. while you access the / directory). DO NOT write, remove, delete anything while you are at the / directory!*

*→*

BONUS! You can also work on the TryHackMe challenge. Take a look at the folder “Bonus Practical: TryHackMe: Linux Room” in this week’s Learning Materials.

*→*

# Conclusions

During this lab session you learned the basic commands that you can use when you have to deal with Linux - Unix (+ MacOS) machines and the GUI is not available. There exist various lists of additional commands if you want to experiment with them. Take a look at the References [1], [2], [3].

Next week you will be ready to initiate your forensic computing practice performing your first incident response task.

# References

1. An A-Z Index of the Bash command line for Linux. https://ss64.com/ bash/. Accessed: 2020-10-14.
2. List of Unix commands. https://en.wikipedia.org/wiki/List\_of\_ Unix\_commands. Accessed: 2020-10-14.
3. The Ultimate A To Z List of Linux Commands. https://fossbytes. com/a-z-list-linux-command-line-reference. Accessed: 2020-10- 14.