**Computing Basics – The UNIX Shell and Command Line Basics**

**Hands-on activities**

In this workshop you will learn how to:

* Get a terminal or Unix Command Line Interface (CLI) shell
* Use some basic Unix commands to explore the CLI
* Create and manage Unix files/folders

--------------------------------------------------------------------------------------------------------------------

**Activity 1- How to get CLI Shell**

Similar to how Windows and Mac are computer Operating Systems (OS), Unix is the Operating System for Linux computers. The Unix command line (CLI) shell or terminal is the user interface for interacting with the OS in Linux.

Linux and MacOS computers have built-in CLI shell, it’s also available on Virtual machines and Linux machines running in the cloud. Windows computers can get CLI shell or terminal from a Windows Subsystem for Linux (WSL) installation or from other 3rd party software such as MobaXterm or Putty.

Demonstrations in this workshop are run in MobaXterm.

**Open CLI terminal on Windows**: The free edition of MobaXterm can be found here: <https://mobaxterm.mobatek.net/download.html>, install the MobaXterm Home Edition (Installer edition), launch the application and click Start local terminal.

**Open CLI terminal on MacOS:** In the Mac search bar, -> type terminal.

OR, click Finder and select:

Applications -> Utilities -> Terminal

The terminal displays a command prompt when it is ready for commands to be typed in. A command prompt is a sequence of characters that may include information like the pathname of the current working location (folder/directory) or a hostname for a remote machine and usually ends with a symbol. Command prompt symbols are not unique, different types of symbols exist. Here are some popular ones: $, %, #, :, >.

Below is the command prompt for my MobaXterm terminal on Windows



Here is the command prompt for my Mac terminal



What information is on your command prompt?

To execute or run Unix commands, type the specific text command(s) next to the prompt symbol, then hit Enter (Windows Users) or Return (MacOS Users) on the computer keyboard. When a command is finished executing, the prompt may change and any output that is generated gets displayed below the prompt. Also, the command prompt returns ready for another command to be typed in.

---------------------------------------------------------------------------------------------------------------------

**Activity 2: How to use some basic UNIX commands to navigate CLI**

Now that we have a CLI shell or terminal, let’s use some basic Unix commands to explore the CLI environment. Several Unix commands are very simple and only two letters long. Commands may have powerful options and may require arguments. The general format for using a Unix command is:

command option1 option2 argument1 argument2 ...

Unix commands are case sensitive, and must be typed in lowercase with spaces included between commands, options and arguments. Files can serve as arguments as shown in the following command syntax below, where a file called filename serves as the argument.

command option(s) filename

Exercise 1 - getting help and man pages

Basic help information for Unix commands can be displayed by running the command name and the --help option or by running the man command

Run command --help on MobaXterm and man command on mac.

For example, let’s get the options for the ls command by running:

ls –-help

Go ahead and review the output for the ls –-help command and use some of the ls options (such as –l, or –a) to explore further.

What is the output when you run ls with these two options: –l and –a?

* ls –l displays the long listing of files in that location
* ls –a displays the list of all files in that location, including dot . files

A command’s manual pages provides a description of the command, its options and how to use it.

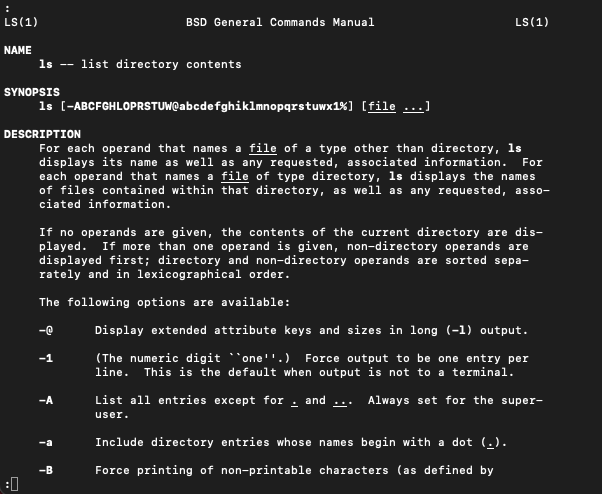
To display manual pages for a command, run:

man command

Display the pages for the ls command by running:

man ls

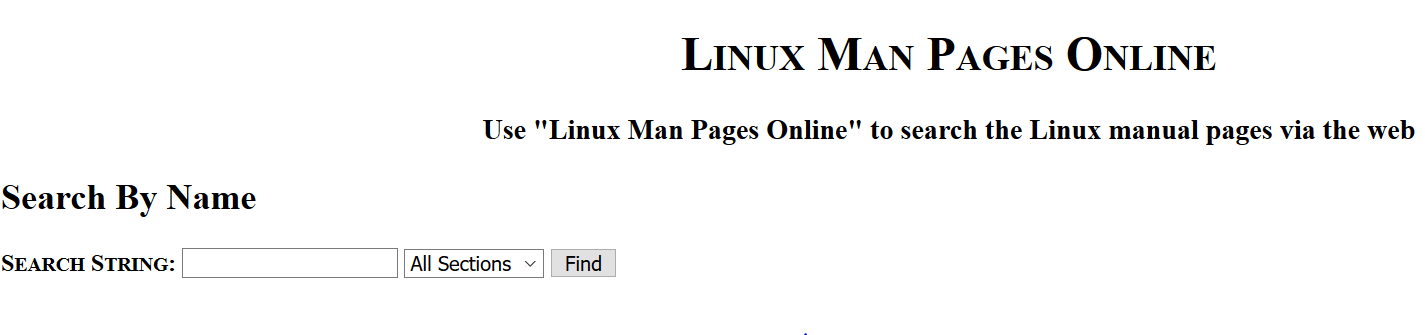
Below is the output from a mac terminal



The up/down arrow or space bar on your computer keyboard can be used to scroll through and browse the manual pages. When finished reading the man page, press ‘q’ on your keyboard to quit the ‘man’ page and return the prompt.

Note that Linux and MacOS terminals will display manual pages, but MobaXterm has limited Unix features and lacks some manual pages. This is not a major problem for using MobaXterm when it comes to implementing features for the many other functionalities needed for working in a CLI shell.

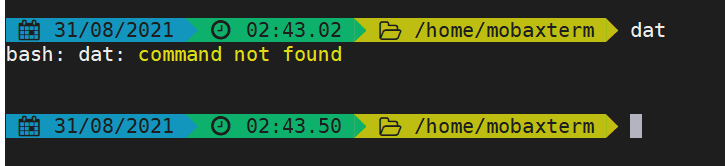
If needed, Linux man pages are available online at <http://man.he.net/> - enter the command into the search box and clicking Find.



Check out this [BasicUnixCommands](https://wustl.box.com/s/55jzm979p1ejxr6ayw38ujmqkuh044ux) document for information on the commands used in this workshop.

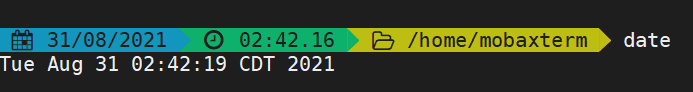
Exercise 2 – dealing with errors

Misspelling a command or failure to include spaces between commands, options and arguments, could generate an error message or not return the prompt. For example, a misspelling of the date command generates an error message as shown below.



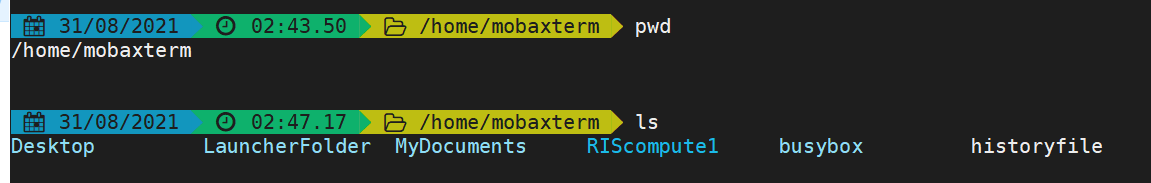
For help with error messages, read the error message and look up an explanation for the error here - <https://tldp.org/LDP/abs/html/mkexitcodes.html> or search for it on Google.

With the correct spelling of date command, the output is:



Exercise 3 – navigate the CLI using pwd, ls and cd commands

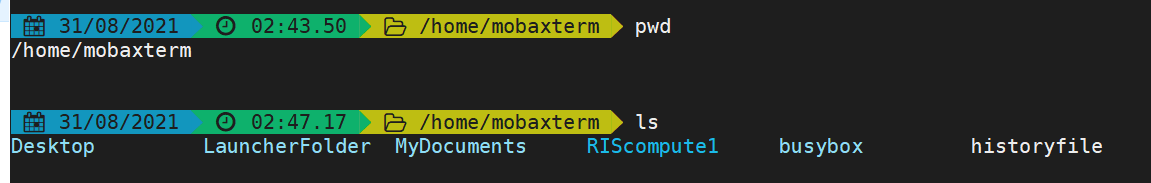
Let's start with the pwd command, it can tell us the location or the folder we’re in. Start by running the pwd command and reviewing the output:



This output indicates that my current or present working location is /home/mobaxterm,

what is yours?

Next, we’ll run ls command, remember that it lists the names of files/folders in the current location (/home/mobaxterm)



We can use the ls and supply it with a pathname or folder name argument to list names of files/folder for a specific sub location. Let's supply Desktop to the ls command to list the names of files/folder in the Desktop location.



Are you curious about what's in the root directory, if so, use ls and the forward slash / option to list the names of all files and folders in root. The forward slash / option alone refers to the root of the Unix file tree.



We can change from the current working location using the cd command. In Unix CLI, directory (folders, subfolders) locations and files are referenced by either absolute or relative pathnames.

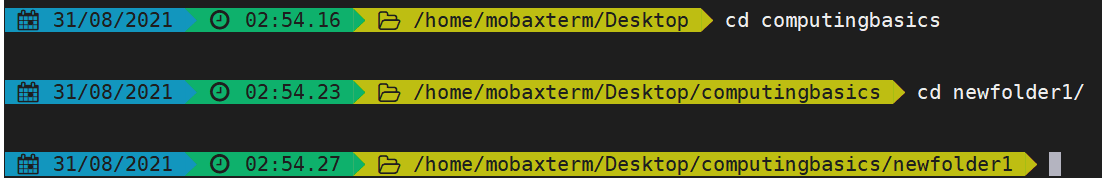
File and folder names in a pathname are separated by a forward slash /. When we run the cd command with a pathname argument or directory name (current working directory), it changes the current working directory location:

cd pathname

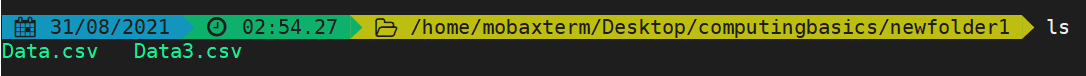
OR

cd foldername

The current location below is /home/mobaxterm/Desktop let’s change location and move into newfolder1 location within the computingbasics folder. Notice that no output is generated as shown below, however the command prompt changes to add the new location to the pathname.

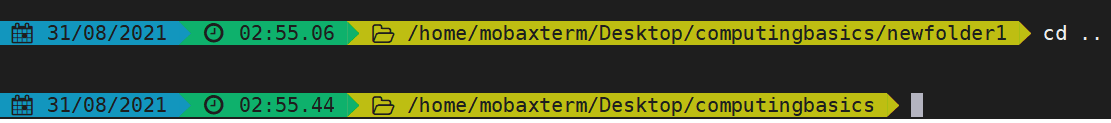


We can run ls to view the files in the new location.

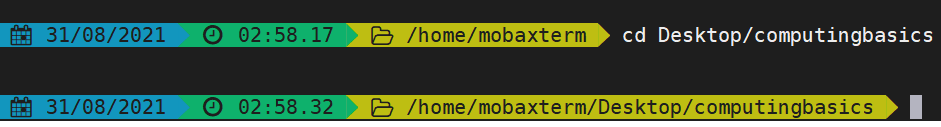


If we wanted to return to the folder that is located one step/level up in the Unix file tree, we'll use the cd .. (cd double dots) command. The double dots .. refer to the folder above the current folder.

The prompt changes to include pathname for the current location.



What if you wanted to change folders and access a different folder that’s not one step away? Running the cd command with the pathname of interest will change the current working directory to the new pathname.



The prompt changes to include the pathname of the new location.

Feel free to experiment further in exploring the CLI terminal on your computer

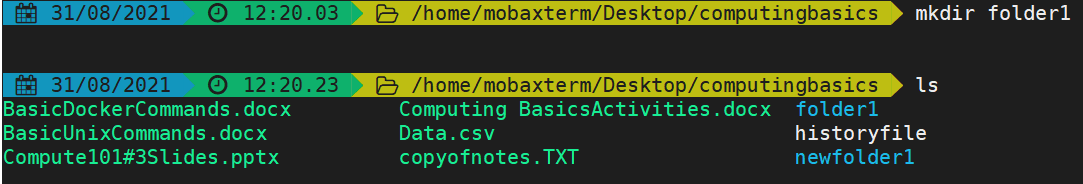
--------------------------------------------------------------------------------------------------------------------

**Activity 3 – How to create UNIX folders and files**

Exercises 1 – create folders and files

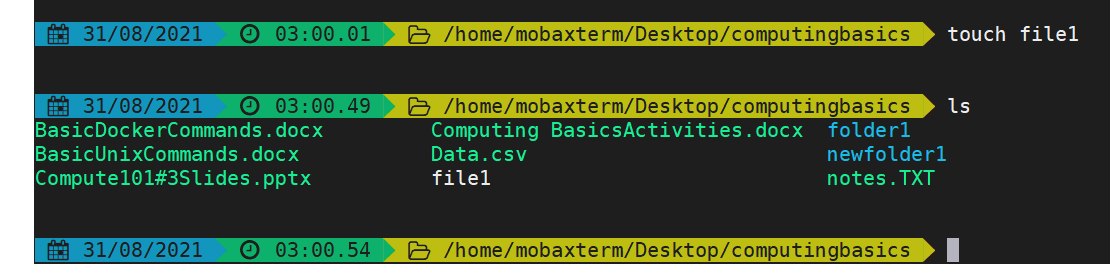
Creating folders – The mkdir command with a folder name creates a new empty folder

mkdir folder1 means, make a new folder called folder1



Creating files - There are a number of ways to create a new file in Unix, here, we are going to use the touch command to create a new empty file

touch file1 means, make a new empty file called file1



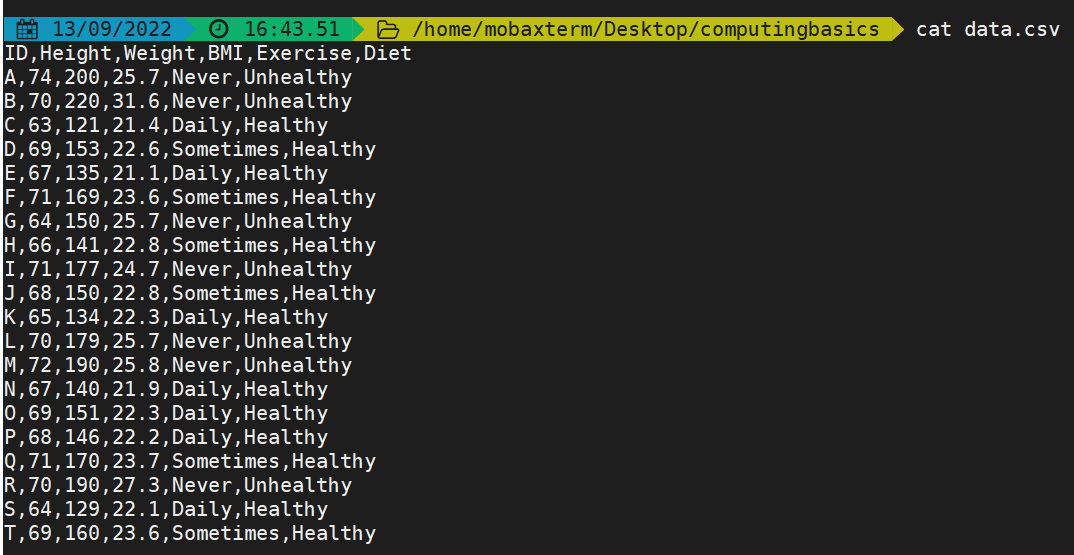
Exercise 2 – displaying file contents

Run the cat command with file name to display file contents on the screen

cat filename

In this exercise, we’ll use the data.csv file found in the computingbasics folder. Let’s now use the cat command and the data.csv file to display the file contents on to the screen

cat data.csv



The head and tail commands are great for viewing contents of files as well.

Remember how to use the --help option to get information for a command? With that in mind, what are some options for using the head and tail commands? Find out by running the following commands:

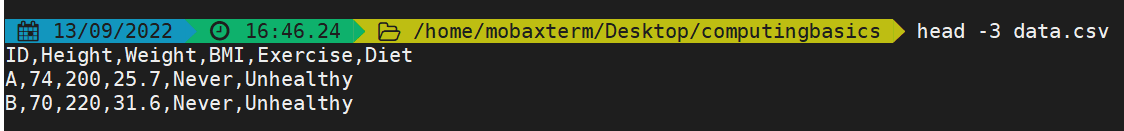
head –-help

tail –-help

Review the list of options for the head and tail commands and display the first 3 lines of the data.csv file:

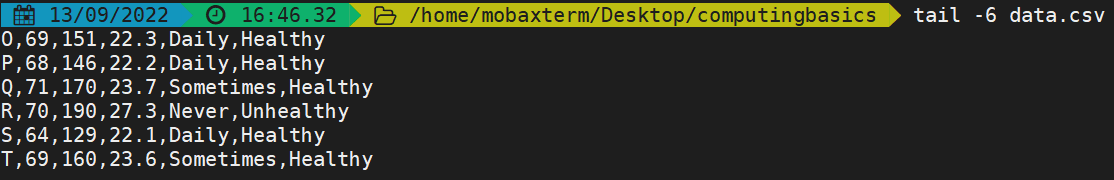
The general command is as follows: head –n filename, where –n is the number of lines in the file. To display the first 6 lines for the data.csv file, run:

head -3 data.csv



Similarly, the general command to display the last lines for a file is tail -n filename. Let’s display the last 6 lines of for the data.csv file using follow command:

tail -6 data.csv



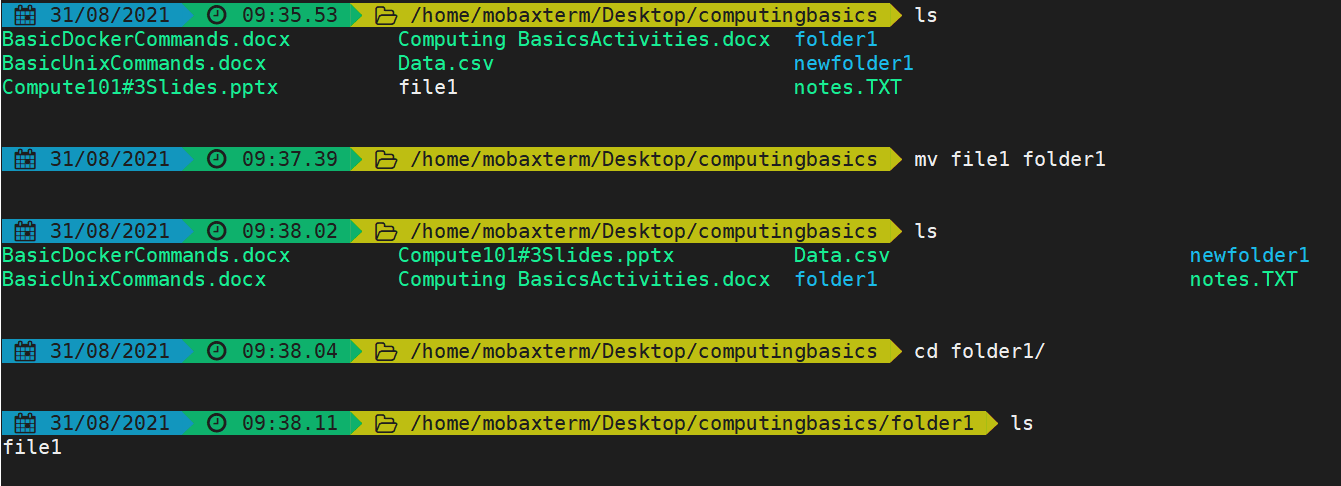
Exercise 3 – moving files around and copying files

Moving files – a file can be moved from one folder to a new folder location - The mv command moves files between folders. Remember that file and folder names in a pathname are separated with the slash /. Below is an example scenario:

mv /folder1/file1 /folder2 means, move file1(located in folder1)into folder2

Let’s move file1 from the computingbasics folder into folder1 using the command:

mv file1 folder1



You can move folders in the same way. Type the command to move folder1 into newfolder1 too.

Copying files - cp(make a copy of a file) - If we wanted to make a copy of a file, we'll use the cp command:

cp file1 file2 means, make a copy of file1, name it file2 and save it in the current working location

In this example, we’ll make a copy of the data.csv file in computingbasics folder and name it copyofdata.csv. It’ll get saved in the current directory which is the computingbasics folder.

cp data.csv copyofdata.csv

How can you confirm that copyofdata.csv is really a copy of data.csv? Run:

cat copyofdata.csv

In the following scenario, cp is used to make a new copy of a file in a new folder.

cp file1 folder1/ means, make a copy of file1 and save it in folder1. The dot . indicates that folder1 is in the current directory. If we wanted to make a copy of the data.csv file and save it in folder1, the command will look like this:

cp data.csv folder1/.

How would you confirm that a copy of data.csv is now in folder1? One way to do so is to run the ls command with the folder name:

ls folder1

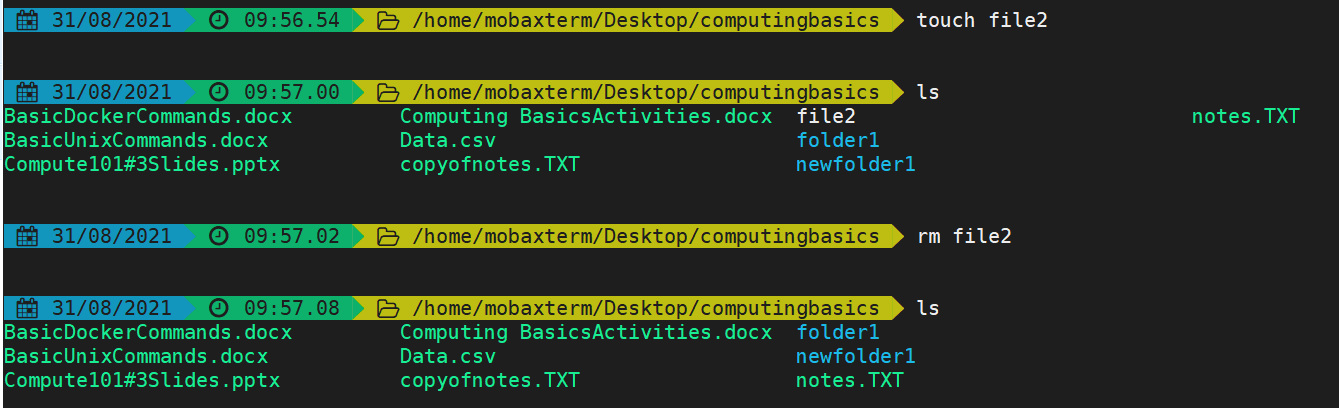
Alternatively, we can run cd command with folder1 to change to the folder1 location, and then run ls to confirm that the data.csv file was copied and saved in folder1.

**Activity 4 – Explore and manage UNIX folders and files**

Exercise 1 – deleting files and folders

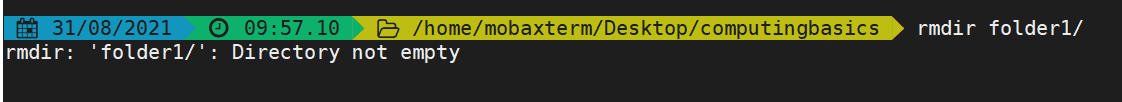
Deleting files – the rm command with the filename, deletes the file.

rm file2 means remove or delete file2 as demonstrated in the image below:



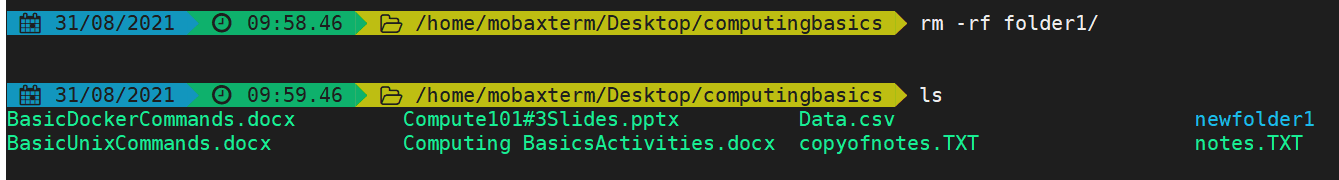
Deleting empty folders – the rmdir command and the folder name, deletes the folder.

rmdir folder1 means remove folder1 (the folder must be empty for it to work). Note that the folder must be empty otherwise you will get an error message and the folder will not be deleted, see the demonstration below.



For rmdir to delete a folder, first delete all the files in the folder, once the folder is empty, use the rmdir command and the folder name to delete the empty folder.

Use the rm -rf command to delete a folder that is not empty. It recursively (r) goes through the directory and deletes all files encountered and the folder. Basically, it will Force (f) the removal of the folder and all of its contents. Note that deleted files/folders can’t be recovered, so be absolutely sure that you really want to delete everything in the folder before using rm-rf command.



Exercise 2 – more with history and grep commands

Get a history of previously run commands - the CLI shell keeps track of the commands that have been run in the past, run the history command to display a list of those commands. Then direct that history content into a new file:

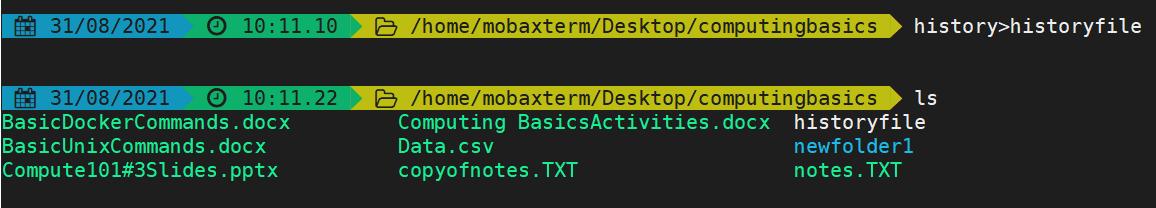
First type history at the prompt and run it to see a list of previously run commands:

history

Now, use the > symbol to direct the output from the history command into a file. In Unix CLI, the > symbol takes the output from one command and puts it into a file. If the file doesn't exist, a new file gets created, but if the file exists, it is overwritten.

Here's the command to direct the output of history command into a file called historyfile:

history>historyfile

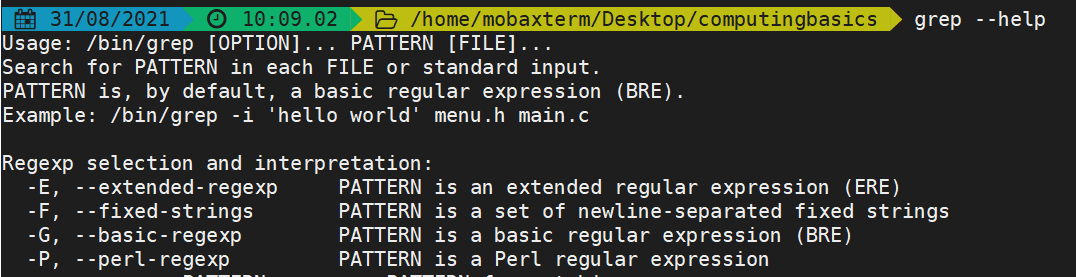


Note: the >> symbol takes the output of a command and adds it into a file, if the file exist, the output is added to the end of the file

Find and print lines in a file using the grep command. Grep stands for Global Regular Expression Print. The grep command finds and prints lines in files that match a pattern. This is a common sequence of operations in early Unix text editors. In this exercise, we’ll use grep on the historyfile to select lines that match a simple pattern

First, let’s look up the options for grep?

grep --help



The grep command is used to find and print lines in files that match a pattern.

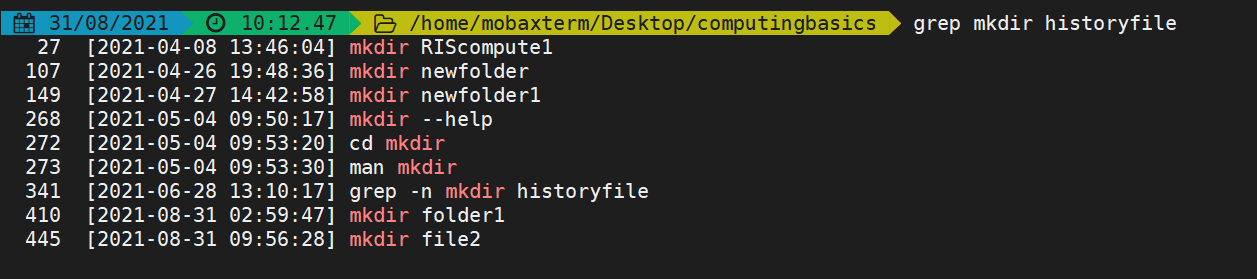
Usage: grep [OPTION]... PATTERN [FILE]...

The general command for using grep to find and print lines in a file that match a pattern is:

grep stringtomatch filename

Let’s find lines that contain the mkdir pattern in the historyfile file:

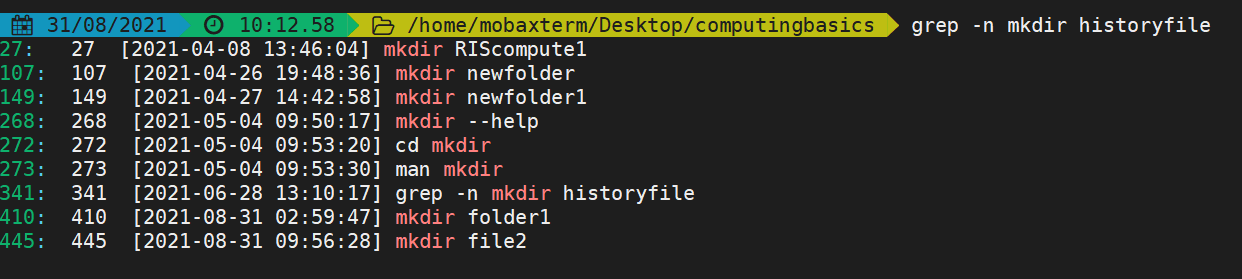
grep mkdir historyfile



Go ahead and search the historyfile for other patterns of your choice....for example ls, cd or pwd

Further, the -n option can be used to display the line in the historyfile from which a pattern appears, let’s try this with a pattern that has mkdir

grep –n mkdir historyfile



Grep can search for a character string in all files in a folder, to do this, use the –r option as shown below:

grep –r $PATTERN .

The period . above is very important, it denotes that grep is accessing files in the current or present folder location. We can search for a pattern in all files in the computingbasics folder using the command:

grep –r $PATTERN .

For example, search for the mkdir pattern in the computingbasics folder

grep –r mkdir .

**---------------------------------------------------------------------------------------------------------------------**

**Learn more from this RIS Computing 101 workshop** <https://docs.ris.wustl.edu/doc/compute/workshops/ris-compute-101.html>

**---------------------------------------------------------------------------------------------------------------------**

**Bonus Exercises:**

Navigate to your home directory, mine is home/mobaXterm/ yours might be different:

* list all the items in that directory
* list only the files with a specific file extension for example .csv, txt, .docx
* create a new folder and navigate to that new folder location
* create a new empty file