Tutorial 1

Registration Code for CS108-T3 SAFE course → 03RXPXSI

1) Basic commands list:-

pwd : present working directory

ls : list directory contents

mkdir : make new directory

cd : change directory

mv : move

cp : copy

rm : remove

rmdir : remove directory

man : manual documentation page

cat : concatenate

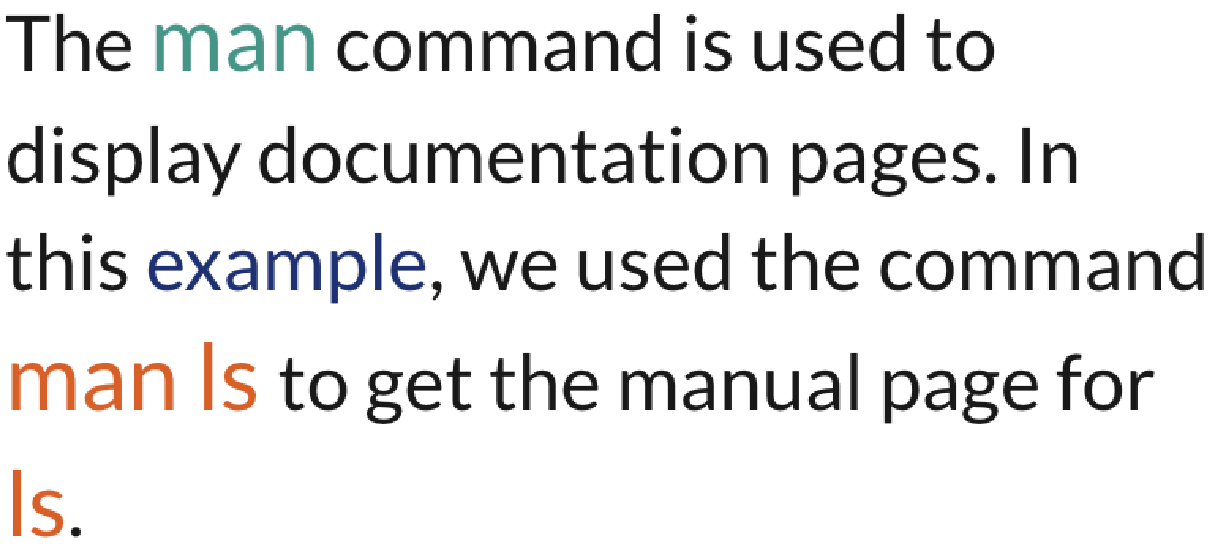
clear : clear the terminal screen

echo : display the text passed in as an argument

head : display first lines of a file

tail : display the last part of a file

2) man command:-

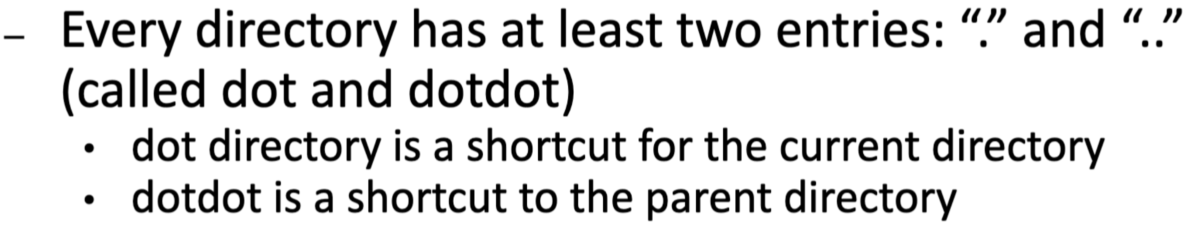


The man command is used to display documentation pages. In this example, we used the command man ls to get the manual page for ls.

man man

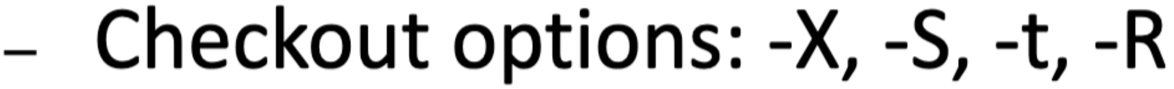
3) ls command:-

1. The ls command is used to display a listing of files and directories.
2. If no arguments are given, then provides the list of files and directories in the current location.
3. If argument is given, then provides the list of files and directories within the specified path.
4. Additionally, various options can be used with ls to modify the output or gather more detailed information about the files and directories.
   For example:
   -a : lists hidden files/directories as well
   -l : list files in the long format



- Every directory has at least two entries: "." and ".." (called dot and dotdot)
  - dot directory is a shortcut for the current directory
  - dotdot is a shortcut to the parent directory

ls -l (detailed)
ls -a (hidden)
ls -a -l OR ls -al (hidden and detailed)

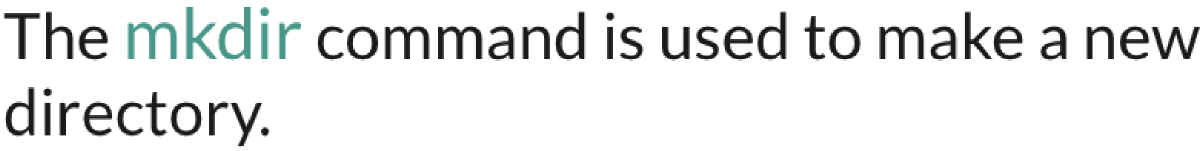


- Checkout options: -X, -S, -t, -R

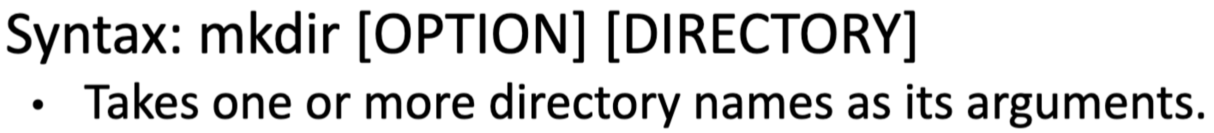
If you look at the man page of ls (man ls) then we can find -X, -S, -t, -R
ls -R (recursive listing) ****Important**
ls -t (sort by time, newest one first)
ls -r (different than ls -R) (reverse sort)
ls -X (sorts alphabetically for a particular extension)
ls -Rl OR ls -R -l (recursive and detailed listing)

****Intersting** → ls -d */ (lists all level 1 subdirectories in a directory)

4) mkdir command:-



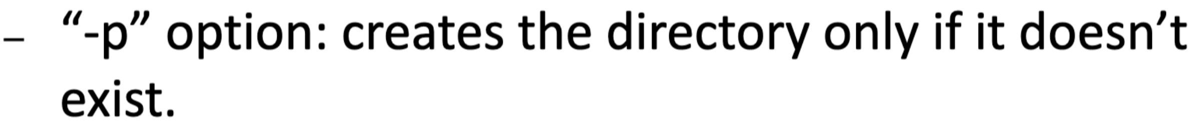
The mkdir command is used to make a new directory.



Syntax: mkdir [OPTION] [DIRECTORY]

- Takes one or more directory names as its arguments.

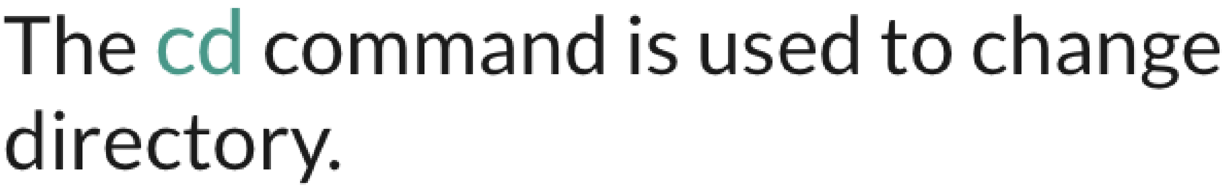
mkdir dir1
mkdir dir1/dir2
mkdir my folder (2 different directories "my" and "folder")
mkdir my\ folder OR mkdir "my folder" OR mkdir 'my folder'
mkdir folder1/folder2 (Error if folder1 does not exist)



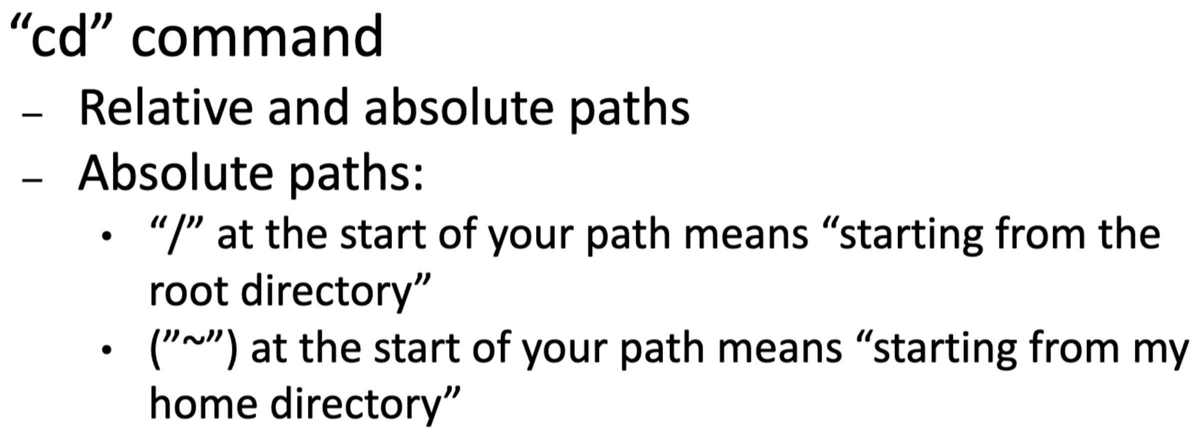
- "-p" option: creates the directory only if it doesn't exist.

mkdir -p folder1/folder2 (Creates folder1 if it did not exist before)
mkdir -p dir1/dir2/dir3 (Creates dir1, then inside of it, dir2, and inside of dir2, dir3 is made
****Imp. → But if it existed, still no error. It would then behave as "mkdir folder1/folder2"**

5) cd command:-



The cd command is used to change directory.



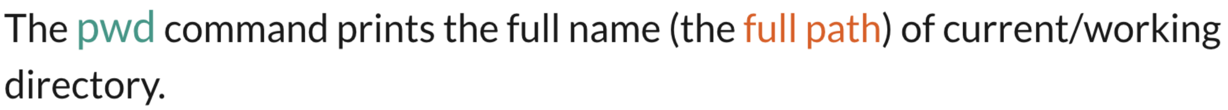
“cd” command

- Relative and absolute paths
- Absolute paths:
  - “/” at the start of your path means “starting from the root directory”
  - (”~”) at the start of your path means “starting from my home directory”

****Imp. → For a root user, home directory is /root or “~” is equivalent to /root**

cd (takes to home directory)
cd - (takes to previous directory)
cd ~ (same as cd)
cd .. (parent directory)
cd ../.. (parent of parent directory)

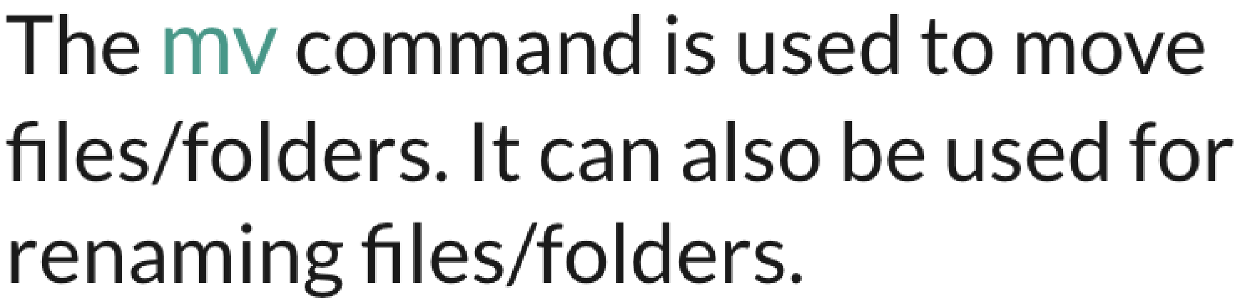
6) pwd command:-



The pwd command prints the full name (the full path) of current/working directory.

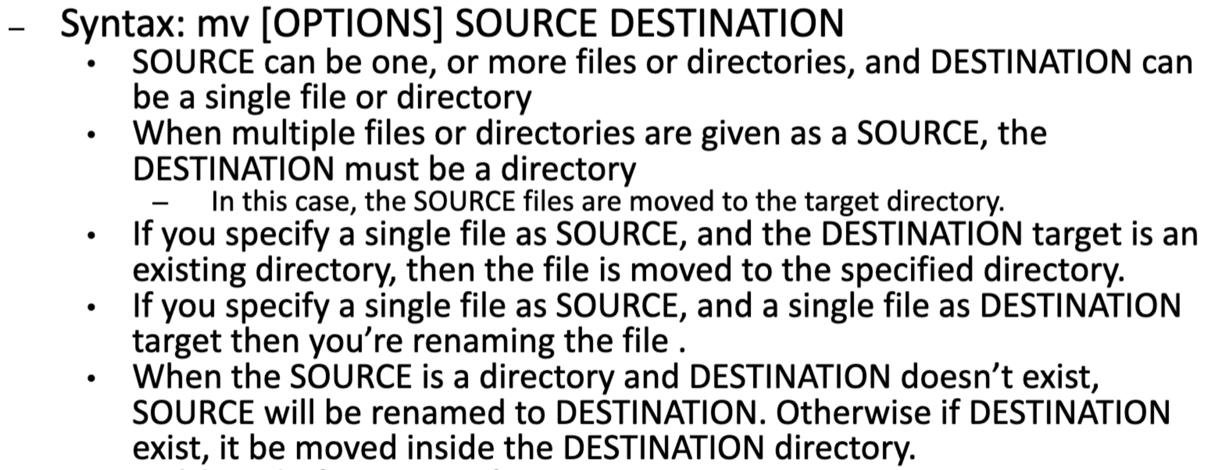
****Imp. → pwd shows complete absolute path**

7) mv and cp commands:-



The mv command is used to move files/folders. It can also be used for renaming files/folders.

****Imp. → mv is a way of renaming files/folders!**



- Syntax: mv [OPTIONS] SOURCE DESTINATION
  - SOURCE can be one, or more files or directories, and DESTINATION can be a single file or directory
  - When multiple files or directories are given as a SOURCE, the DESTINATION must be a directory
    - In this case, the SOURCE files are moved to the target directory.
  - If you specify a single file as SOURCE, and the DESTINATION target is an existing directory, then the file is moved to the specified directory.
  - If you specify a single file as SOURCE, and a single file as DESTINATION target then you're renaming the file .
  - When the SOURCE is a directory and DESTINATION doesn't exist, SOURCE will be renamed to DESTINATION. Otherwise if DESTINATION exist, it be moved inside the DESTINATION directory.

****Imp. → Very important image**

mv file1 file2 dir1 dir2 dir3 (Everything moved to dir3, assuming all things existed)

****Imp. → This can happen without error only if dir1, dir2 are empty directories**

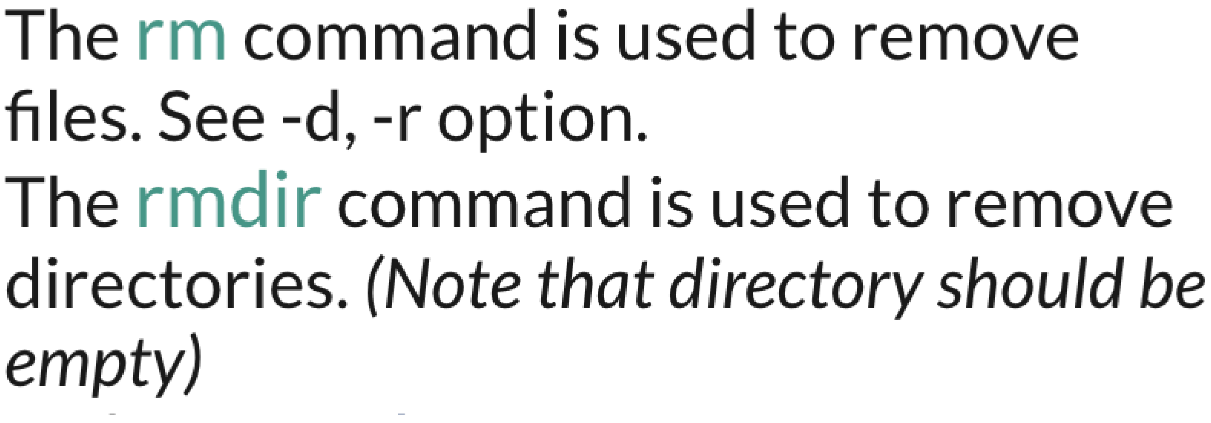
mv file1 file2 (file1 renamed to file2, even if file2 exists. If file2 existed before, then its content will be automatically deleted)

cp file1 file2 (Contents of file1 are transferred to file2, even if file2 didn't exist before) If file2 existed before, its previous contents will be deleted; otherwise, a new file2 will be created. file1 always remains.

cp -R dir1 dir2 (Copies all contents of dir1 **along with dir1)** into dir2)

****Peculiar** → cp -R dir1/ dir2 (Copies all contents of dir1 **without dir1)** into dir2)

8) rm and rmdir commands:-



The rm command is used to remove files. See -d, -r option.

The rmdir command is used to remove directories. *(Note that directory should be empty)*

rm -r dir1 OR rm -r dir1/ (Recursively deletes all contents of dir1 along with it)

rm -f (forceful removal)

rm -rf (deletes everything without a question)

rm -d (exactly same as rmdir)

rmdir dir1/dir2/dir3 (Deletes the empty folder dir3 only, not dir1 or dir2)

rmdir -p dir1/dir2/dir3 (Deletes all dir1, dir2, dir3 BUT there should be no other content !!)

9) echo command:-

- The echo command outputs whatever is given to it as argument.

echo “Hello World” (prints Hello World)
echo “Hello\nWorld (prints Hello\nWorld)
echo -e “Hello\nWorld” (prints Hello in one line and World in next line)

10) cat command:- {Comes from word ‘concatenate’}

Cat (concatenation): display the contents of a text file to screen

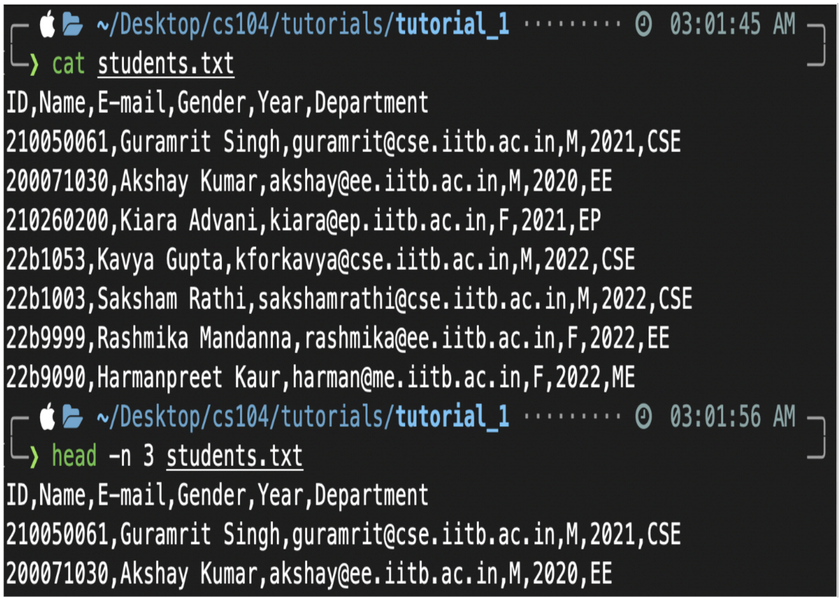
- Why named cat then? can combine outputs also
- “-n” option to display contents of a file with line numbers
- “-s” option to omit repeated empty output lines:

echo “Hello” > file1; echo “World” > file2; cat file1 file2
(prints Hello in one line and World in next line)

11) head and tail commands:-

head/tail: print n lines from head or tail

- head file; head -n 2 file
- Checkout: -n and -c option



```
╭─  ~/Desktop/cs104/tutorials/tutorial_1 ········· ◷ 03:01:45 AM ─╮
╰─❯ cat students.txt                                              ─╯
ID,Name,E-mail,Gender,Year,Department
210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE
200071030,Akshay Kumar,akshay@ee.iitb.ac.in,M,2020,EE
210260200,Kiara Advani,kiara@ep.iitb.ac.in,F,2021,EP
22b1053,Kavya Gupta,kforkavya@cse.iitb.ac.in,M,2022,CSE
22b1003,Saksham Rathi,sakshamrathi@cse.iitb.ac.in,M,2022,CSE
22b9999,Rashmika Mandanna,rashmika@ee.iitb.ac.in,F,2022,EE
22b9090,Harmanpreet Kaur,harman@me.iitb.ac.in,F,2022,ME
╭─  ~/Desktop/cs104/tutorials/tutorial_1 ········· ◷ 03:01:56 AM ─╮
╰─❯ head -n 3 students.txt                                        ─╯
ID,Name,E-mail,Gender,Year,Department
210050061,Guramrit Singh,guramrit@cse.iitb.ac.in,M,2021,CSE
200071030,Akshay Kumar,akshay@ee.iitb.ac.in,M,2020,EE
```

“-n x” prints the first x lines
“-c x” prints the first x characters