What is markdown?

- is a markup language that lets you write plain text documents with a few light weight formatting options.

What is Git?

- Version control system. Its a command.

Github is a website trunk based developer. Hosted Git in cloud or website

what is Slack?

- Slack a messaging app, but built for work.

Lecture 2 Introduction to Linux Notes

- What is an Operating System? An operating system provides all fundamental software features of a computer. An OS enables you to use the computer's hardware providing you the basic tools that make the computer useful.All of those features relay on the OS's kernel. Other OS features are owed to additional programs that run atop the kernel.
- What is a kernel? An OS kernel is a software component that's responsible for managing low-level features of the computer, including the following managing system hardware, memory allocation, CPU time, and program to program interaction.
- Which part aside from the kernel identify an OS? Command - line shells - Graphical User Interfaces - Utility and Productivity Programs- Libraries
- What is linux and linux distribution? Linux is a Unix Operating System popular in academic and business environment
- List at least 4 linux characteristics: open source software-free of charge- highly scalable- you can use on almost any system
- What is Debian? Debian is an all volunteer organization dedicated to developing free software and promoting the ideals of the Free Software community.
- List and define the different types licensing agreements open source- the software may be distributed for a fee or free. The source code is distributed with the source code. The user is restricted from modifying the code. closed source- the software is not distributed with the source code. The user is restricted from modifying the code. freeware- the software is free but the source code is not available. Shareware- the software is free on a trial basis
- What is Free software? Define the 4 freedoms. the software is distributed with the source code. The software can be free of charge or obtained. Freedom 0- use the software for any purpose. Freedom 1 examine the source code and modify it as you see fit. Freedom 2- redistribute the software. Freedom 3- redistribute your modified software.

What is a graphical user interface (GUI)? a graphical user interface is a of program that allows a user to with a computer system via icons window and various other virtual elements.

> What is a desktop environment? is an implementation of the desktop metaphor made ofa bundle of programs running on top of a computer operating system which shares a common GUI sometimes described aa a graphical shell.

is the command line interface (CLI)? the user interacts with the computer by inputting commands, the commands are interpreted by a shell, the shell can be considered a programming language

How do I access the command line interface (CLI)? Terminal Emulator and Linux console

What is a virtual console? a terminal system that runs in Linux system memory.

What is a terminal emulator? allows you to access Linux CLI when using the GUI

What is bash? a program that provides interactive access to the Linux system

What is the shell prompt? When you launch a terminal

Definition, usage, and examples of the following commands:

---

clear clears the terminal window/screen

echo Prints/displays text to screen

- date displays current date and time
- free Displays the amount of RAM/ memory used and free
- uname sgiws basic information about your system such as Kernel, OS, architecture
- history display a history of every command you run so farPrints/displays text to screen
- man Shows/displays manual page for a given command
- tldr similar to cheat sheet but it is available to APT and keeps the local cache cheat a collection of cheat sheets hostname Shows the system's hostname
- df Displays the amount of disk space available/used on entire file systems/partitions
    - du dispalys the amount of space used by specific files or directories figlet displays text in ASCII letters

echo plus option plus string

Display/print a line of text to the screen

echo "Hello World"

Display/print 2 lines of text to the screen in a single echo command

echo -e "line1\nLine2"

Displays/print a line of text to the screen supressing the new line

echo -n "Hello World"

Display/print a line of text to the screen with a tab

echo -e "\tHello World"

-n Do not append a newline at the end of the output. ex. echo -n "Hello"

-e Enable interpretation of backlash escapes (like \n,\t,etc)

echo -E "Hello\nWorld

-E Disable interpretation of backlash escapes (this is the default behavior)

echo -E "Hello\nWorld"

Escape Sequence Meaning

-e Enable interpretation of backlash escapes (like \n,\t,etc) (use -e after echo to perform the followig commands)

```
Example Output
```

\n Newline Hello\nWorld \t Horizontal tab Hello\tWorld \ Backlash Hello\World \b Backspace Hello\bWorld \a Alert (bell) Triggers sound alert

```
echo -e "line1\nline2"
```

```
echo "Current date UTC"
date -u
echo "Disk space human readable -h"
df -h
echo "memory"
free -h
echo "hostname"
uname -n
echo "operating system"
uname -op
```

example

```
echo: to display text
date: to manipulate dates
df: disk space usage
free: memory usage
uname: basic system information
clear: clear screen

- How to install and remove software using the APT command?
```

Formula

- sudo apt install package name
- you can install/remove multiple programs by adding the package name with a space between each package
- You can also remove packages by adding the - sign at the end of the package name
- You can add and remove package at the same time by using a +and - at the end of each package

```
- How to create a shell script step by step including screenshots and how
to run it. Try to be as detailed as possible.
- open a text editor
- save file as file_name.sh
- The first line in the file must be shebang or shell interpreter. In the
case of bash it would be:
- #!/bin/bash
- The rest of the script includes the commands you want the shell to
execute when the files is run.
- To run the script, use the following command:
- bash /path/to/script/scripts_name.sh
```

-

example on how to run shell script

- #!/bin/bash
- echo then "hello world"

ls - used for displaying all the files inside a given directory. When no directory is specified, ls

displays the files in the current working directory.

Examples

ls list the content of the present working directory

ls -a List all the files inside the current working directory including hidden files

ls -a ~/Pictures-

List all the files in a given directory recursively.

ls -lR ~/Pictures

man ls - to search for commnd also do grep

similar commands

dir, tree, exa

cd - used for changing the current working directory is given, cd changes the current user.

Change from your home directory to a Downloads directory.

cd Downloads

from anywhere in file system change to your Downloads directory

cd ~/Downloads

from anywhere in the file system change to your Documents directory.

cd /home/$USER/Documents

assuming that your present working directory is /usr/share/themes/ go to /usr/share

cd ../

If you want to go to your home directory you can use the following

cd cd ~ cd $Home If you want to go to the previous current working directory you can use cd - pwd- used for displaying the current work directory.

what is a variable?

is a container or placeholder for data allows you to temporarily store information within the shell script for use with the other commands in the script.

How do I use a variable? User="Bob" is read as

User contains the value "Bob" User has the value "Bob" User store the value "Bob" User has assinged the value "Bob"

What is an environment variable?

are used by the shell to track specific system information Some environment variable's value do not change from user to user, while user specific environment variable will change depending on the user logged in. you can use any of the following commands to see a list of environment variables: env, set, printenv

Environment varibales are typed in capital letters. This helps to differntuate them from user defined variables.

To use environment variables, type their name starting with a dollar sign

Example

echo $USER will display the current user.

What is an user defined variable?

are created by the user and exist only in the script and subshell that runs the script. User variables alllows you to temporarily store and use data and use it throughout the script.

Rules for creaitng vaiables can be any text string of up to 20 letters, digits, or underscore characters but they CANNOT start with a number.

User variables are case sensitive

Values are assigned using an equal sign no spaces (name='Peter')

The shell storess all values as text strings: Bash is essentially untyped

What the root directory?

- is the administrator of your system

What does "Parent Directory" mean?

- allows you to move forward to a subdirectory

What does "Current directory" mean?

- the directory you are at the moment

What is an absolute path?

- f Include an example full pathname absolute path always starts with the /home/tdw/music/song.mp3

What is relative path? Include an example starts with a subdirectory partial pathname for example:

- music/song.mp3

What is the difference between " Your home directory" and 'The home directory"?

-"Your directory" is my user's and "The home directory" is located in the root.

mkdir

- is used for creating a single directory or multiple directories. To create a directory with mkdir type: mkdir plus the name of the directory. To create multiple directories. separate each directory name with a space. You can create directories in the present working directory or in a different directory using an absolute path. You can create a directory with a space in its name using the escape character() or by surrounding the name in quotation marks.

touch

- is used for creating files.

rm removes files.

- to remove directory use rm with the -r option.To remove empty directories use the rmdir command. To remove non - empty directories use rm -r plus directory name or directory absolute path.

cp

- copies files/directories from a source to a destination. The cp command use the same mv command.cp plus files to copy plus destination. To copy directories use the -r option. cp -r plus directory to copy plus destination.

mv

- moves and renames directories. basic formula is mv plus source plus destination.Where source is the file or directory that you want to move and destination is where the directory or file is going. For renaming files/directories the formula remains the same: mv plus file/directory to rename plus new name.

Create a directory in the present directory

- mkdir wallpapers

Create a directory in a different using relative path

-mkdir wallpapers/ocean

Create a directory in a differnt directory using absolute path

mkdir ~/wallpapers/forest

Create a directory with a space in the name

- mkdir wallpapers/new\cars
- mkdir wallpapers/'cities usa'

Create a directory with a single quote in the name

- mkdir wallpapers/"majora's mask"

Create multiple directories

- mkdir wallpapers/cars wallpapers/cities wallpapers/forest

Create a directory with a parent directory at the same time

-mmkdir -p wallpapers_others/movies

To create a file called list

- touch list

To create several files

- touch list_of_cars.txt script.py names.csv

To create a file using absolute path

- touch ~/Downloads/games2.txt

Yo create a file with a space in its name

- touch "list of foods.txt"

The Rm

- rm removes files
- rm by default does not removes directories. To remove a directory use rm with the -r option
- To remove non empty directories use the rmdir command
- To remove non empty directories use rm -r plus directory name or dilrectory absolute path

Remove a file

- rm list

Remove a file and prompt confirmation before removal

- rm -i lsit

Remove all the files inside a directory and ask before removing more than 3 files

- rm -I Downloads/games/*

Remove an empty directory

- rmdir Downloads/game

Remove an non empty directory

- rm -r Downloads/games

Notes7

How to use each of the wildcard.

* Matches zero to any number of characters

Example

- ls *.pdf

? Matches only one character

Example s

- ls program?.py

[set] Matches 1 character from a given set of characters

Example

- ls document [A-Z].doc

[!] The opposite of the given set

list all the files in a given directory

- ls Downloads/*

List all the text files in a given directory

- ls Downloads/*.txt

List all the text files in a given directory that start with letter f

- ls Downloads/f*.txt

List all the files that contain the word file in the name

- ls *file*

Example

- ls new-doc[!0-9].docx

How to use

Expansion to create entire directory structures.

- Start with an open brace
- With no spaces, type your string separating entries by a command.
- Close the brace
- example mkdir -pv example_site /{assets/large,docs/share,scripts/js}## Notes8

cat

command is used for displaying the content of a file. cat + option + File(s) to display

Basic Examples:

Display the content of a file located in ~/Documents/sample_files/

cat ~/Documents/sample_files/Code/helloWorld.py

Display the content of a file with line numbers

cat -n ~/Documents/sample_files/Code/helloWorld.py

Display the content of a file including non printing characrers and line endings.

cat -A ~/Documents/sample_files/Code/helloWorld.py

tac command is used for displaying the content of a file in reverse order

tac + option + file(s) to display

Basic Example

Display the content of a file located in ~/Documents/sample_files in reverse order

tac ~/Documents/sample_files/Code/helloWorld.py

Display the content of multiple files in reverser order

```
~/Documents/sample_files/Code/helloWorld.py
/Documents/sample_files/Code/helloWorld.py.sh
```

head command displays the top N of lines of a given file. By default, it prints the first 10 lines. If more than one file name is provided then data from each file is preceded by its file name.

head + option +file(s)

Basic Example

Display the first 10 lines of a file

head ~/Document/Book/dracula.txt

Display the first 5 lines of a file

head -5 ~/Documents/Book/dracula.txt

Displays the first 5 lines of multiple files

head -n 5 Txt/{dracula,war-and-peace}.txt

Displays the first line of multiple files using wildcards

head -n 1 Csv/*.csv Code/*.py

Displays a given number of lines of the output of a given command

ls -l ~/cis106/ | head -n 2

Displays the name of the file in the output

head -v -n 7 Json/joke.json

Displays a given number of bytes instead of lines

head -c 50Txt/dracula.txt

tail command displays the last N number of the lines of a given file. By default, it prints the last 10 lines. If more than one file name is provided then data from each file is preceeded by its file name.

tail + option + file(s)

Basic Example:

Displays the last 10 lines of a file

tail ~/Documents/sample_files/

displays the last 5 lines of a file

tail -5 /Documents/sample_files/

Displays the first 5 lines of multiple files

tail -n 5 Txt/{dracula,war-and-peace}.txt

Displays the first line of multiple files using wildcards

tail -n 1 Csv/*.csv Code/*.py

Displays a given number of lines of the output of a given command

ls -l ~/cis106/ | tail -n 2

Displays the name of the file in the output

tail -v n 7 Json/joke.json

Display a given number of bytes instead of lines

tail -c 50 Txt/dracula.txt

cut + optioN + file(s)

Basic Example

Display a list of all the users in your system

cut -d ':' -fl /etc/passwd

Displays a list of all the users in your system with their login shell

cut -d ':" -fl,7 /etc/passwd

Cut a range of bytes per line

cut -b 1-5 usernames.txt

cut -d ':' -f1,7 --output-delimiter=' " /etc/passwd

Cut a file excluding a given field

cut -d ',' -- complement -s -f3 users.txt

Cut the permissions from the output of Is

ls -l | cut -d ' ' -- complement -s -f1

sort

Sort a file and save the output to a new file

sort -o sorted.lst users.lst

Sort a file in reverse order

sort -r users.txt

Sort by column number

sort -k 2 users.txt

Sort a file with numeric data

sort -n codes.lst

Check if a file is sorted

sort -c sorted.lst

sort and remove duplicate entries

sort -u users.lst

wc command is used for printing the number of lines, characters and bytes in a file

wc + option + file(s)

Basic Example

Display the number of characters in a file

wc -m users.txt

Displays the number of lines in a file

wc -l users.txt

Display the number words in a file

wc -w users.txtSearch any line

Search any line that contains the word dracula regardless of case and with number line

```
grep -in 'dracula' ~/Documents/Books/dracula.txt
```

Search for all the lines that do not contain the word 'war'

```
grep -v 'war' ~/Documents/Books/war-and-peace.txt
```

Search and display only the matched string(pattern)

```
grep -o 'pride' ~/Documents/Books/war-and-peace.txt
```

Display a list of users with the/bin/bash login shell

```
grep -i "/bin/bash" /etc/passwd
```

Display your user's information as stored in the /etc/passwd

```
grep -i $USER /etc/passwd
```

Search for a given strings inside files in a given directory .

```
grep -iR 'conf' /etc/
```

Search and display the total number of times a given word appears in a file

```
grep -wc '/bin/bash /etc/passwd
```

The (caret) symbol matches the empty string at the the beginning of a line. Search for all the lines that start with a given word.

grep -ni '^dracula' ~/Documents/Books/dracula.txt

Search for all the lines that ends with the string "nonlogin"

grep -n '$nologin' /etc/passwd

-i Enables case insensitivity. (it wil match regardless of case) -n Displays line number for every line matched -E Treats the pattern (search criteria) as a basic regular expression -G Treats the pattern (search criteria) as a basic regular expression -v Inverts the search (looks for the opposite of the given criteria) -o Only displays the matched string -c Search and display the total number of times a pattern is mathced. -w Matches only given word (pattern) by itself -r, -R Matches recursively. Useful for searching files in a given directory.

```
awk awk + options   {awk command} + file + file to save
```

Basic Example

awk '{print $1}' ~/Documents/Csv/cars.csv

Print first field of /etc/passwd file (user command)

awk -F: '{print $1}' /etc/passwd

Print the last field of the /etc/passwd (login shell)

awk -F: '{print $NF}' /etc/passwd

Print the first and last field of the etc/passwd

awk -F: '{print $1, " = ", $NF }'

Print the first and 3 field with line numbers

awk -F: '{print NR,$1,$4}' /etc/passwd

Print the first and 4th field with a different field separator

awk -F: '{OFS="="}{print $1,$4}' /etc/passwd

Start printing a file from a given line (exclude the first 2 lines)

awk 'NR > 3{ Print }' /etc/passwd

sed command

```
sed options +ic sed script + file
```

Bas Example

Replacing a string in given file globally (replace false for true)

sed 's/false/true/g' ~/Documents/sample_files/Json/joke.john

To delete a particular line (line 5)

sed '3d' ~Documents/sample_files/Code/helloWorld.py

To delete the last line

sed '$d' Documents/sample_files/Code/helloWorld.py

To delete line from range x to y

sed '2,4d' Documents/sample_files/Code/helloWorld.pyT

To delete from a given number to last line

sed '3,$d' Documents/sample_files/Code/helloWorld.py

To delete pattern matching line in a file

sed '/fav/d' Documents/sample_files/Code/helloWorld.py

To insert one blank lined after each line

sed G helloWorld.c

To inset two blank lines

sed 'G;G' helloWorld.c

to delete balnk lines and insert one blank line after each line

sed '/^$d;G' helloWorld.c

Insert 5 spaces to the left of every line

sed 's/^/ /hehelloWorld.c

Explain how to use the pipe (|) for redirecting the output of a command to another. Include at least 3 examples

Usage command_1 | command_2 | command_3 | .... | command_N

Basic Examples

Use grep to look for a string in a particular man page

man ls | grep "human readable" man ls| grep "^[[:space:]]*[[:punct:]] man ls| grep "long" or "comma"

Explain how to save the output of a command to a file (>). Include at least 3 examples

Explain how to append the output of a command to a file. Include at least 3 examples

Final Notes

How to clone a GitHub repository.

- git clone github website via tilx.

How to use the git commands?

- open up tilx: git pull git add . git commit -m "message" git push

How to write a Markdown file that contains images and proper formatting:

- to format any text as heading in Markdown? images
  - start the line with a # symbol then a space.

d. How to convert a Markdown file to PDF?

- right click on document and convert to Pdf.

1. How to compress (zip) a directory/folder in Debian.

- in tilx
- zip file1.zip file1.txt
- unzip file1.zip
- navigate to the folder amd right click on it.compress name and create it.
- cd into it in the terminal then hit zip

What are Absolute paths and relative paths?

Absolute Path

- the location of a file starting at the toot of the file system.

Example: /home/tdw/Downloads/song,mp3

Relatie Path

- the locations of a file starting from the current working directory or a directory that is located inside the current working directory.

Example: Downloads/song.mp3

(provide examples with commands. For example,

creating a file using an absolute path.)

cd /home/$USER/Documents

How to work with the manual pages (man command)?

- man then the command

How to parse (search) for specific words in the manual page.

- man ls | grep "specific command"

How to redirect output (>, >>, and |)

How and when to redirect the output of a command to another (pipes)

- The pipe line allows you to redirect standard output of a command to the standard input of another.

Usage

  - command_1 | command_2 | command_3 | ....| command_N |
  -

Basic Examples

- Use grep to look for a string in a particular man page
- man ls | grep "human readable"

How to append the output of a command to a file

example

- ls -la >> allmyfiles.lst
- > To redirect the output of a command to a file. Essentially saving the output of a command to a file

> > to append (add) the output of a command to a file | to pass the output of a command to another

Output redirection uses the file descriptors:

```
1 (STDOUT) Standard output = the regular output of a successfully executed
command
2 (STDERR) Standard error = any error message returned by a command


Write an echo command that displays the following text “All files in my
home directory”
Save the output of the echo command to a file called: file_report.txt
List all the files (including hidden but not the .. and .) in your home
directory in a comma-separated format and append the output to the
file_report.txt file
Display the content of the file_report.txt file
```

Example

- echo "all files in my home directory" > file_report.txt

How to use echo and output redirection to create a new file that contains same text

usage

- command output plus > plus file

save the output of a command to a file

- ls -lA ~ > all-file-in-home.txt

How to use wildcards (For copying and moving multiple files at the same time)

Move all files from one directory to another

- mv ~/downloads/Nature/* ~/Pictures/wallpapers
-

Copy specific files based solely on their file extension

- cp ~/Downloads/homework/*.pdf ~/Documents/*.txt ~/Projects/school/For creating entire directory structures in a single command)
- is a feature of the bash shell that generates atument strings.
- Start with open brace
- with no spaces, type your string separating entries by a command
- close the brace
-

Example

- mkdir -pv example_site/{assets/large,docs/share,scripts/js}

How to create a simple “hello world” shell script

echo "hello world"

How to use variables in a shell script

#!/bin/bash echo "The current User: $USER" echo -e "The Path VAr: \n $PATH"

awk

- is a scripting language used for processing and displaying text. Awk can work wirh a text file or from the standard output.

Usage

- awk plus options plus {awk commmand} plus file plus file to sav (optional)

Basic Example

Print the first column of every line of a file

- awk '{print $1}' ~/Documents/Csv/cars.csv

Print first field of /etc/passwd file

  - awk -f: '{print $1}' /etc/passwd

Print the last field of the /etc/passwd file

- awk -F: '{print $NF}' /etc/passwd

Print the first and last field of the /etc/passwd

- awk F:{print $1, " = ",$NF}
-

Print the first and 3 field wirth line numbers

- awk -F: '{print NR,$1,$3}' /etc/passwd

cat

usage: cat plus option plus fileS(s) to display

Basic example display the content in a file located in ~Documents/sample_file

~Documents/sample_file/Code/helloWorld.py

Display the content of the file with line numbers

cat -n ~/Documents/sample_files/Code/helloWorld.py

Display the content of a file including non printing characters and line endings

cat -A ~Documents/sample_file/Code/helloWorld.py

cp

- copies files/directories from a source to a destination

The cp command kuses the same structure as the mv command

- cp plus files to copy plus destination

To copy a file

- cp Downloads/wallpapers.zip Pictures/
-

To copy directories you must use the -r option

- cp -r directory to copy plus destinatnion

To copy a directory with absolute path

  - cp -r ~/downloads/wallpapers ~/Pictures/

cut

- The cucpt command is used to extract a specofic section of each line of a fiel and display it to the screen

Usage

cut plus option plus file(s)

Basic Example

Display a list of all the users on your system

- cut -d ':' -f1 /etc/passwd

grep

- is used to serach text given file. Grep works line by line basisi(it matches the search criteria in an linelby line basis)
-

Usage

grep plus search criteria plus fileSearch any line that conatains the word "dracula" in the given file:

Bais Example:

- grep 'dracula' ~/Documents/dracula.txt

Search any line that contains the word 'dracula' regardless of the case

- grep -i 'dracula'' ~/Documents/Books/draucla.txt

Display how many lines contain the matched string

- grep -c 'dracula' ~/Documents/Books/dracula.txt

Bais Example:

head

Usage head plus option plus file(s)

Basic Example:

Display the first 10 lines of a file

- head ~Documents/Book/dracula.txt

- # Dispkay the first 5 lines of a file

  - head -5 ~Documents/Book/dracula.txt
  -

  # Display the first 5 lines of multiple files

  - head -n 5 ~Txt/{dracula,war-and-peace}.txt
  -

  # Display the first line of multiple files using wildcards

  - head -n 1 Csv/*.csv Code/*.py

ls

lsit all the given files in a directory

- ls Downloads/*
-

List all the text files in a given directory

- ls Downloads/*txt

List all the text files in a given directory that start with letter f

- ls Downloads/f*.txt

List all files that contain the word file in the name

- ls *file*

List all hidden files in current working directory

- ls ./.??*

List all the hidden files in the parent directory

ls ../.??*

List all the files that have 2 charcters in the file name between letters b and k

- ls B??k*

List all the files that have a single character between letters f and I

- ls f?l*

man -

mkdir -

used for creating a single directory or multiple directories

mkdir plus the name of the directory

create a directory in the present working directory

- mkdir wallpapers

Creaate directory in a different directory using relative path

- mkdir wallpapers/ocean

Create a directory in a different directory using absolute path

- mkdir ~/wallpapers/forest

mv

- moves and renames directories

mv plus source plus destination

mv plus file/directory to rename plus new name

To move a file from a directory to another using relative path

- mv Downloads/homework.pdf Documents/

To move a directory from one directory to another using absolute path

- sudo mv ~/Downloads/theme /usr/share/themes

tac

Usage tac plus option file(s) to display

Basic Example:

Display the content of a file located in ~Documents/sample_files in reverse order

~Documents/sample_file/Code/helloWorld.py

- tac ~Documents/sample_file/Code/helloWorld.py

Display the content of multiple files in reversse order

- tac ~Documents/sample_file/Code/helloWorld.py
- ~Documents/sample_file/Code/helloWorld.sh

tail

Usage tail plus option Plus file (s)

Basic Example

Display the last 10 lines of a file

- tail ~/Documents/sample_files/
-

Display the last 5 lines of a file

- tail -5 ~/Documents/sample_files/

- Display the first 5 lines of multiple files

- tail -n 5 Txt/{dracula,war-and-peace}.txt
-

Display the first line of multiple files using wildcards

- tail -n 1 CSv/*.csv Code/*.py

touch

- is used for creating files

Examples

- To create a file called list
- touch lsit

To create several files:

touch list_of_cars.txt script.py names.csv

To create a file using absolute path:

- ~/Downloads/games.txt

To create a file using relative path (assumming you pwd is you home directory)

- touch Downlowads/games2.txt

tr

- The tr command is used for translating or deleting characters from standard output

Usage

- Standard output | tr plus option plus set plus set

Basic Example

- Translate one character to another (For example a period with a comma)
- cat file.txt | tr '.' ','

Translate white space into tabs

- cat program.py | tr "[:sapce:]" '\t'
-

Translate white space into tabs

- cat file.py | tr "[:space:]" ' '

tree

list all

- tree -a
-

Real path shows the absolute path

Example

- realpath kitchen/guest_room/attic/.key4