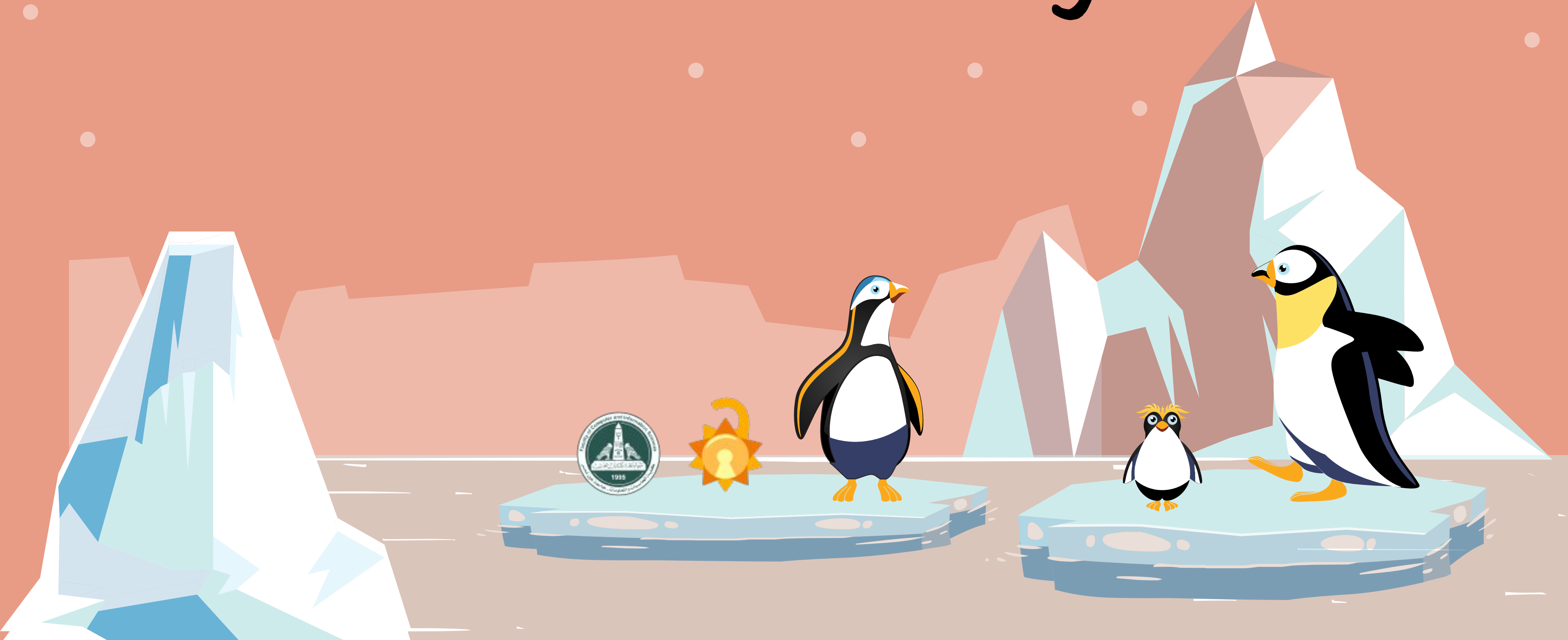
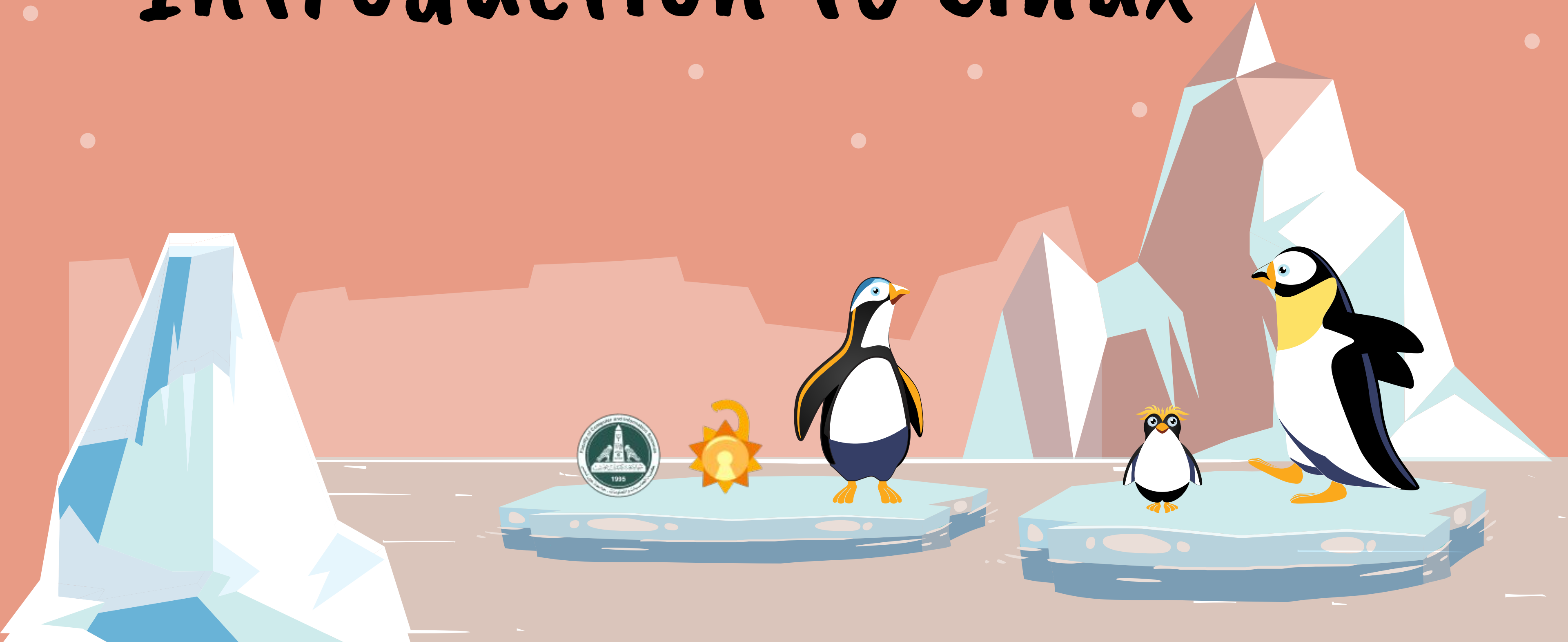
# Linux Summer Training 23



# Introduction to Linux



# Agenda

01 What is Linux?

02 Installing Linux

03 Terminal & Shells

04 File Systems

05 Commands

06 Text Editing with nano

# What is Linux?

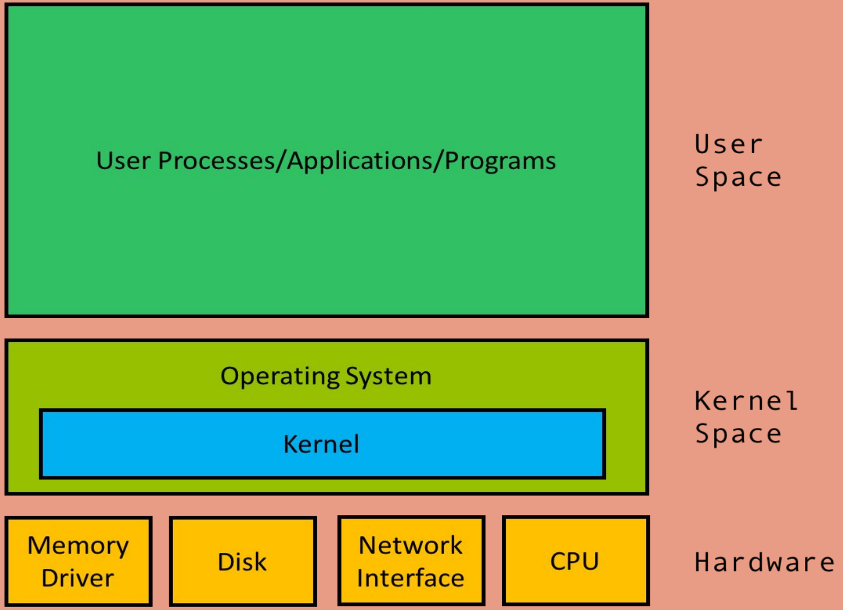
Linux is not an operating system

It’s a kernel

# So what is a kernel? 🤔

- It's the main layer between the OS and underlying computer hardware.
- it helps with tasks such as process, memory management and file systems.

It's the **Heart** of the OS



# Once upon a time...

There was an OS called **UNIX**

- Stable, Secure and Reliable.
- First to introduce hierarchical file system.
- First to introduce CLI within an OS.
- EXPENSIVE.

Richard Stallman

# The GNU project

Developed many essential software components, such as compilers, editors, and utilities.

Missing one **CRUCIAL** component!

Linus Torvalds

# Linux Kernel

“A hobby project”

It's called ...

# GNU/Linux

GNU's not UNIX

But you can still call it Linux 😁

# Over 500 Distributions

Powered by:
archlinux
debian
fedora
foresight linux
gentoo linux
Mandriva
linux Mint
kubuntu
openSUSE
PCLinuxOS
redhat
sabayon
slackware
slax
ubuntu
xubuntu

# What makes Linux great? and why to use it?

- Open source, secure and free.
- Compatible with old machines
- Fully Customizable.

# Agenda

01 What is Linux?

02 Installing Linux

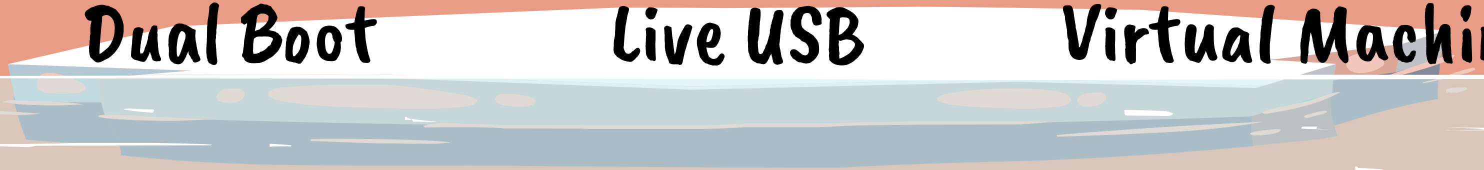
03 Terminal & Shells

04 File Systems

05 Commands

06 Text Editing with nano

# Installing Linux



Dual Boot

Live USB

Virtual Machine

# Dual Boot

- Installing Linux alongside windows on the same or a different disk.
  - The same disk **NOT** the same partition.
- **DOWNSIDS:**
  - You could accidentally erase your entire disk 😕.

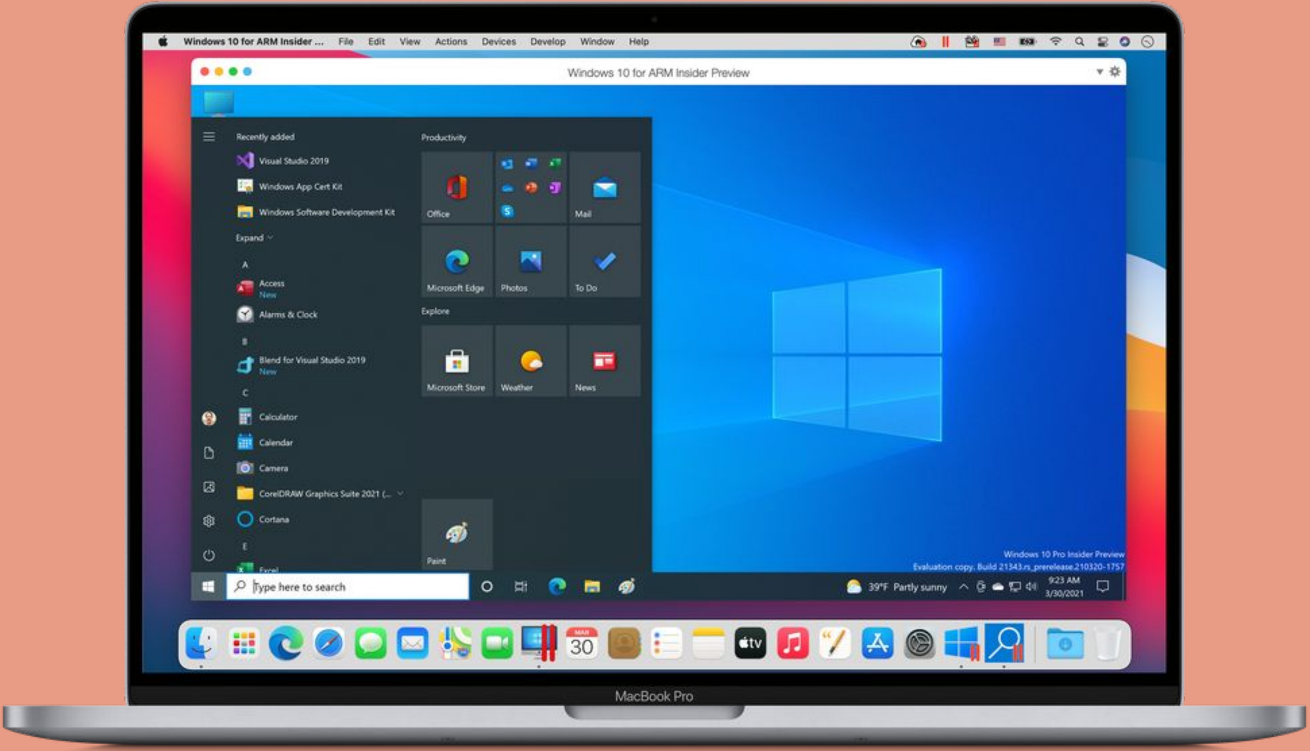
# Live USB

- Burn Linux iso image to a flash drive and boot directly from it.
- DOWNSIDES:
  - Anything you install, edit or save is gone on reboot.

# Virtual Machine

- A computer inside of a computer.
- Running 2 operating systems by sharing resources.
- DOWNSIDES:
  - Poor performance.

# Virtual Machine



# Agenda

02

Installing Linux

03

Terminal & Shells

04

File Systems

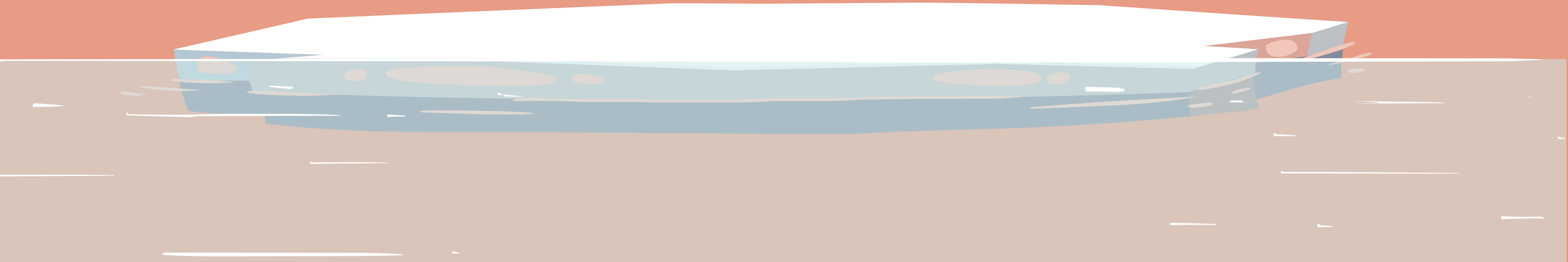
05

Commands

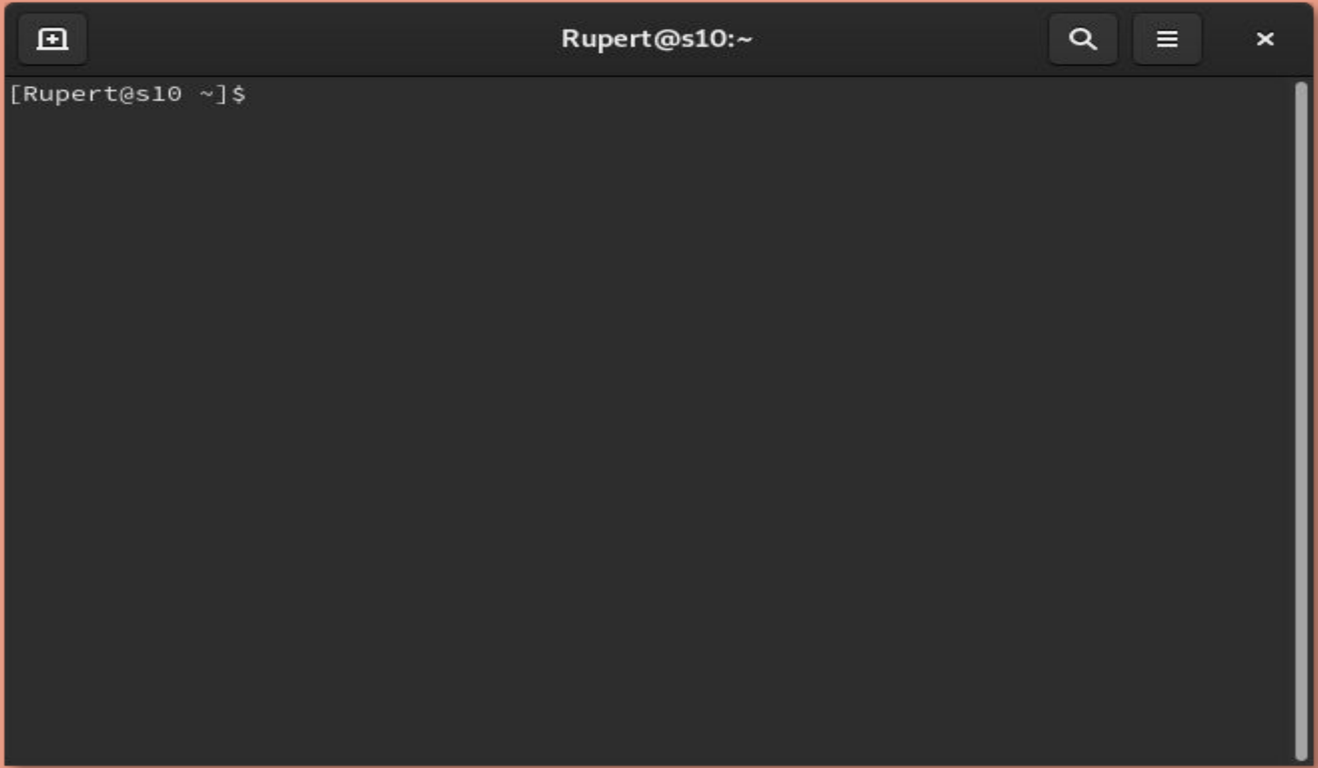
06

Text Editing with nano

# Terminals & Shells

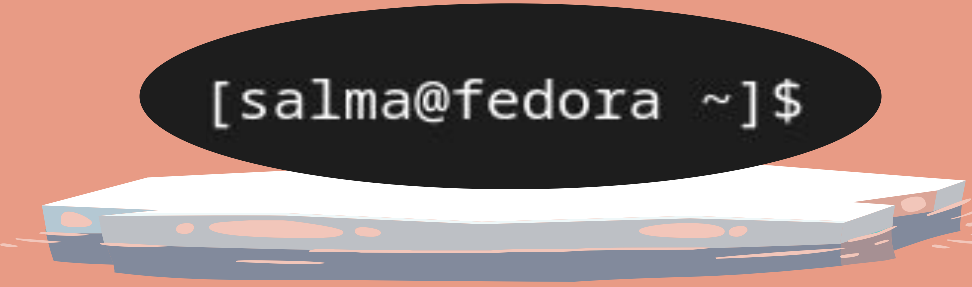


It's a text based command line interface (CLI) that runs instruction on a Linux machine.



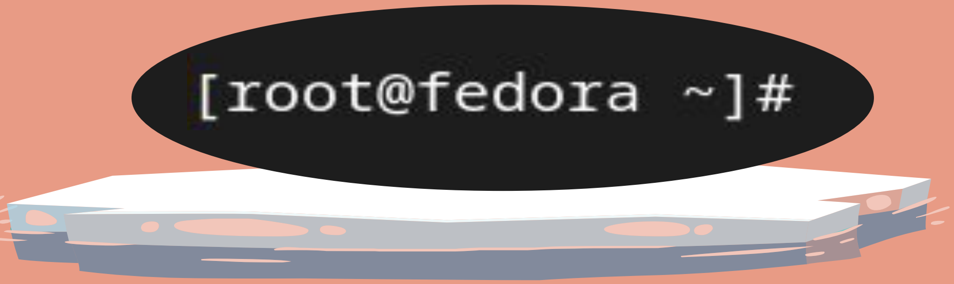
```
[Rupert@s10 ~]$
```

$ → Normal user



```
[salma@fedora ~]$
```

# → Root(admin)



```
[root@fedora ~]#
```

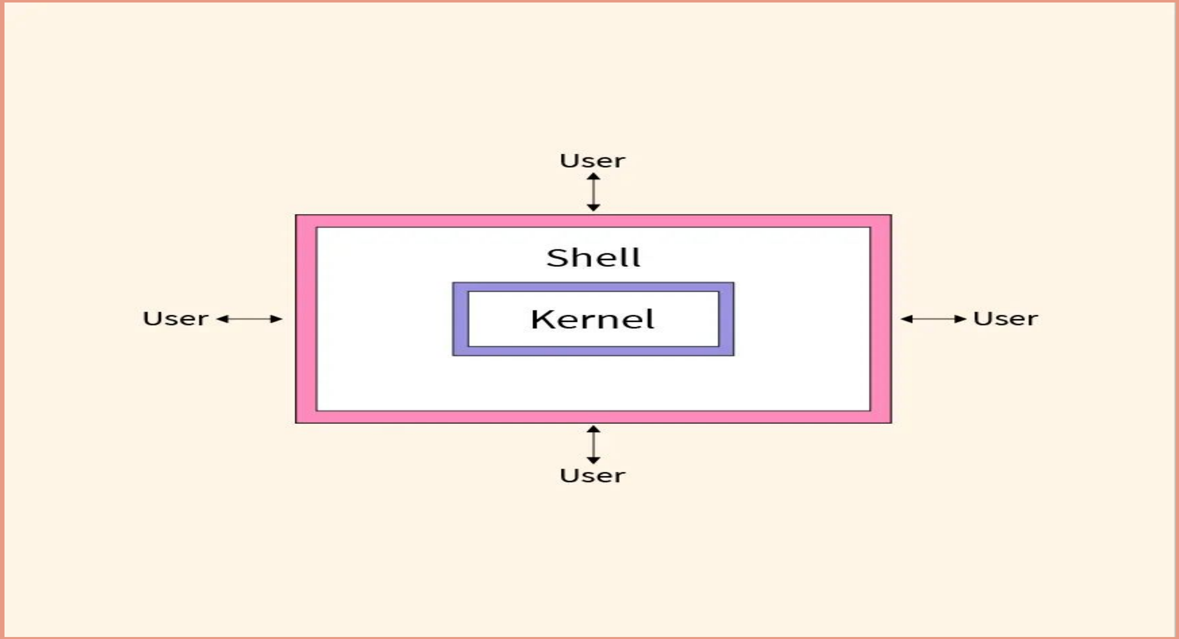
**Username@Hostname  Working_Directory($/#)**

# Why use terminal over GUI?

- Faster than GUI.
- More functionality.

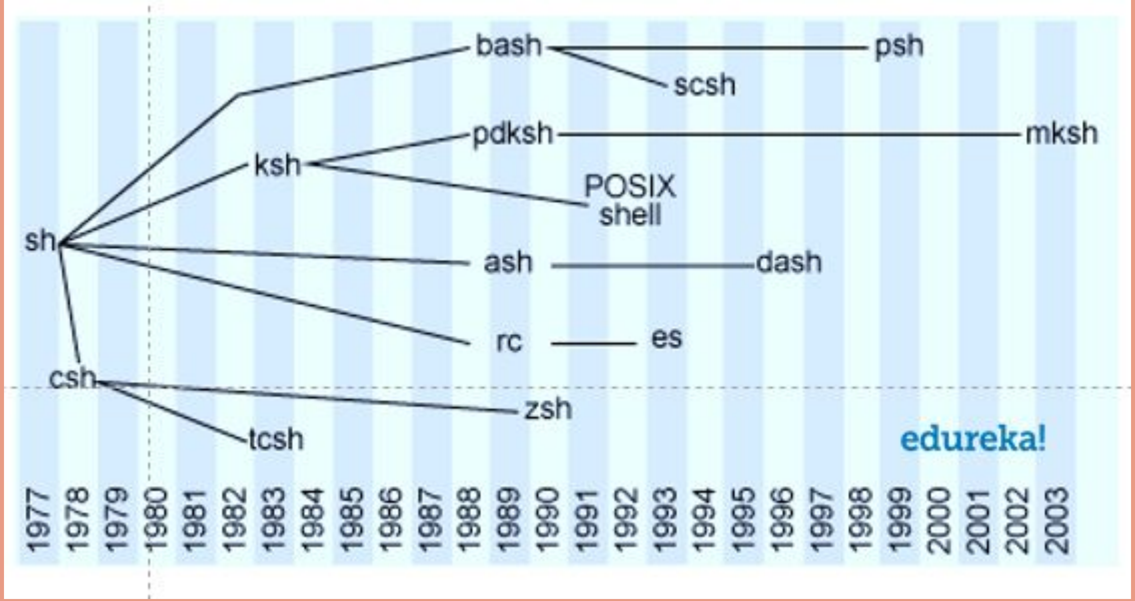
# What is a shell?

A program that runs inside a terminal and takes commands from the user and gives it the OS to execute.



# Types of shells

Differ in the look and feel and have different functionality.



# Agenda

03

Terminal & Shells

04

File Systems

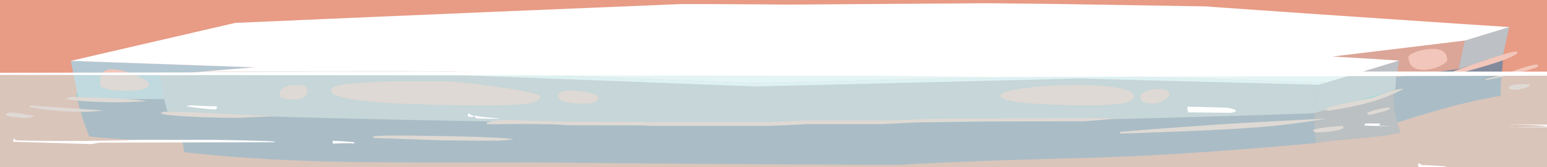
05

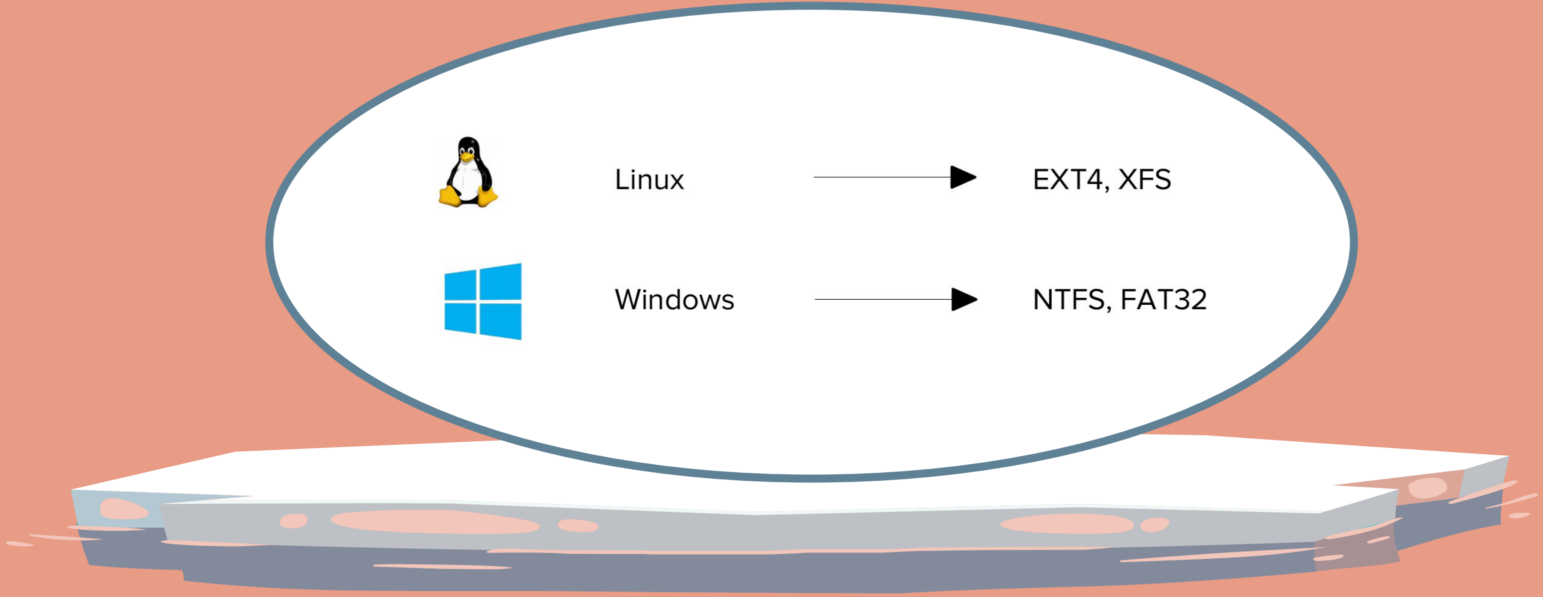
Commands

06

Text Editing with nano

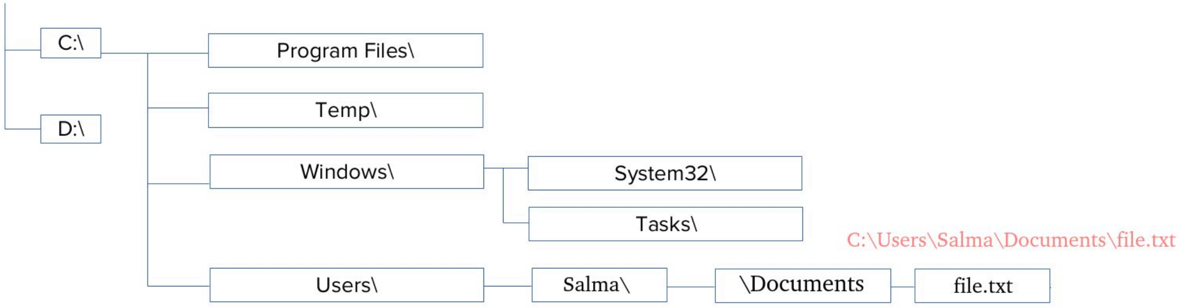
# File Systems





Linux → EXT4, XFS

Windows → NTFS, FAT32



C:\

D:\

Program Files\

Temp\

Windows\

System32\

Tasks\

C:\Users\Salma\Documents\file.txt

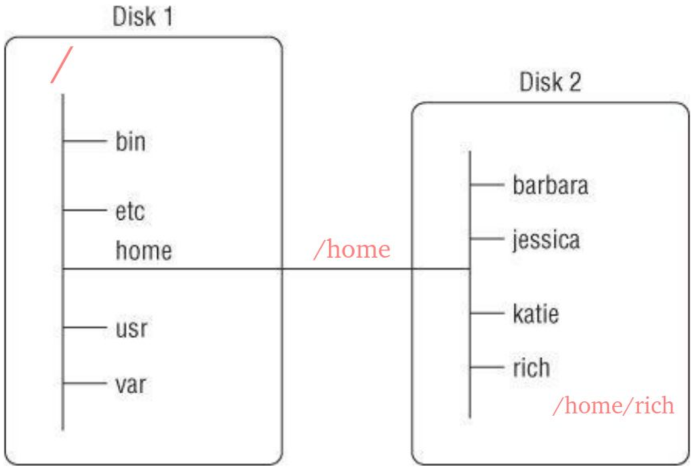
Users\

Salma\

\Documents

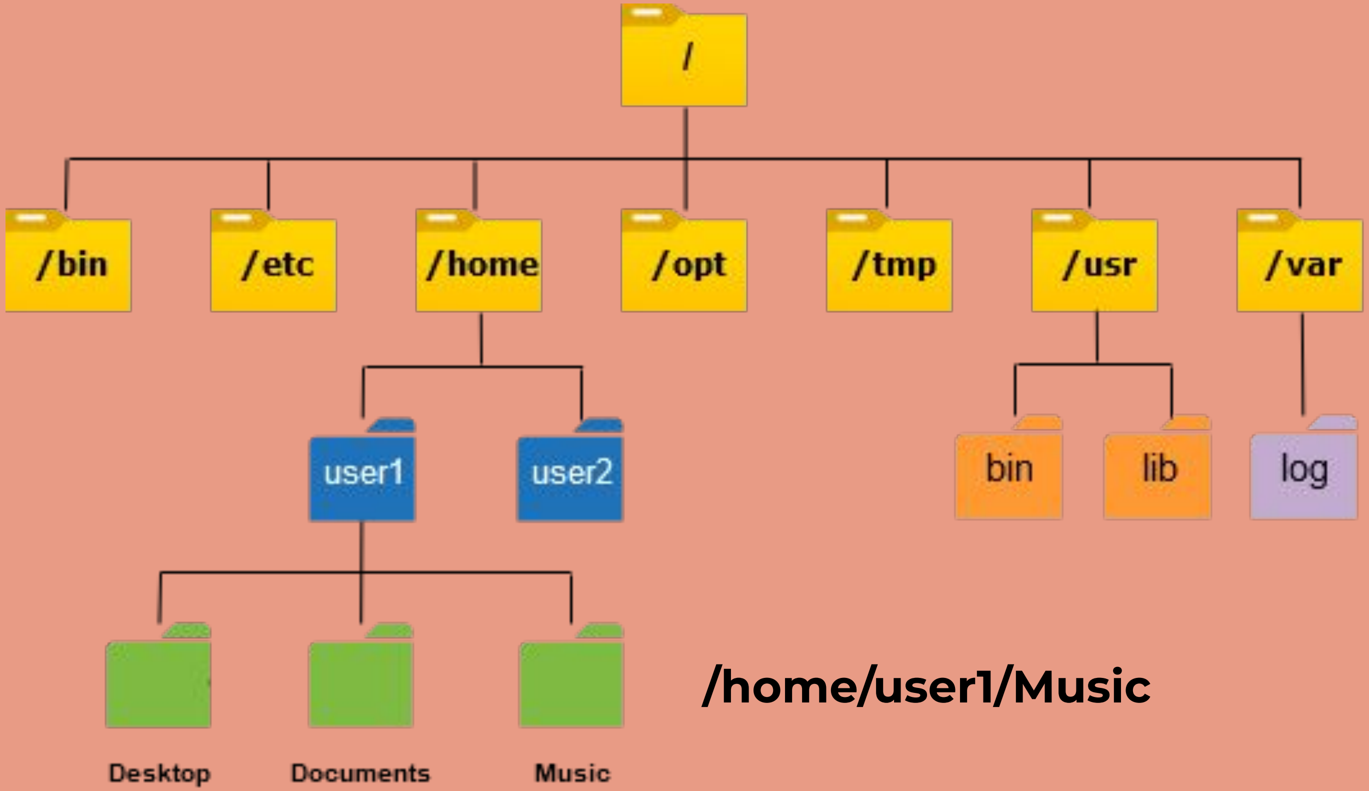
file.txt

# Windows Directory Structure



# Linux File System Hierarchy

# The root / Directory



/home/user1/Music

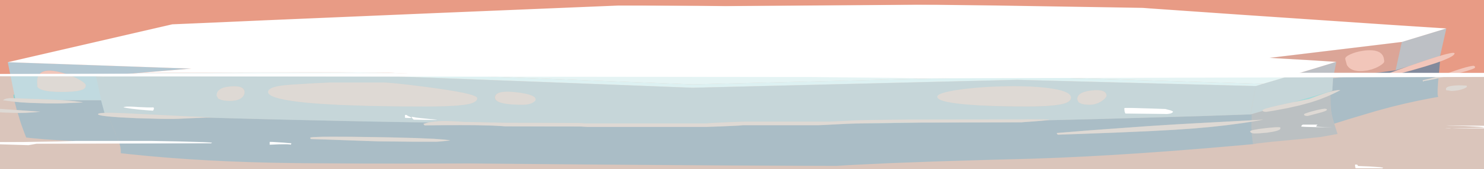
# Agenda

04 File Systems

05 Commands

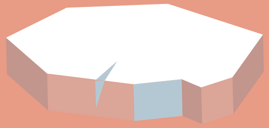
06 Text Editing with nano

# Commands

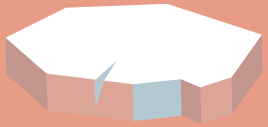


| [Command] | [OPTION] | [Arguments] |
|---|---|---|
| ls | -l, -a, -la | file or directory |
| rm | -r, -f, -rf | file or directory |

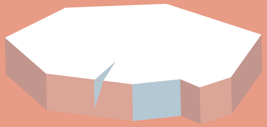
whoami



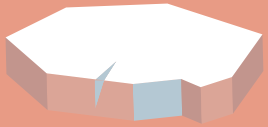
date



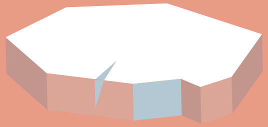
echo



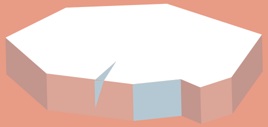
clear



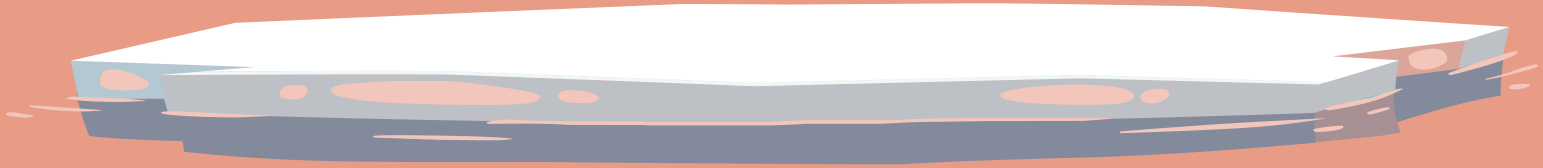
history

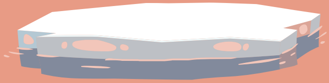
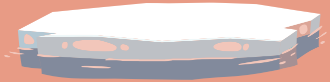
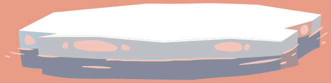


echo $0

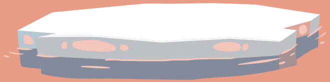


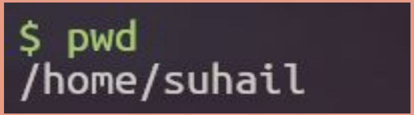
# Navigation Commands



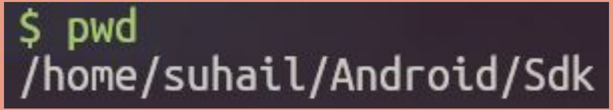
| | |
|---|---|
| pwd<br> | Print working directory |
| ls<br> | List the content of the directory |
| cd<br> | Change directory |

pwd



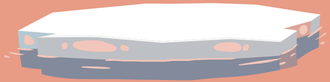


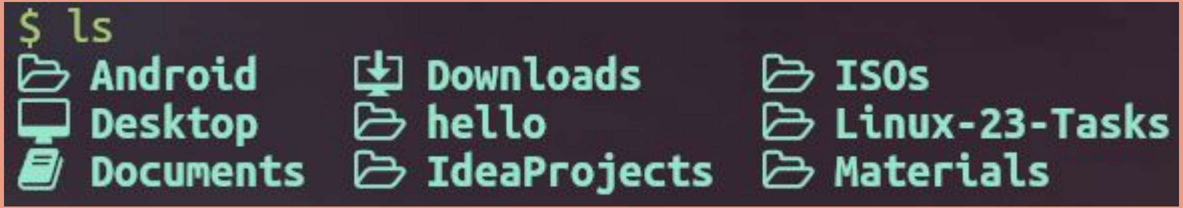
```
$ pwd
/home/suhail
```



```
$ pwd
/home/suhail/Android/Sdk
```

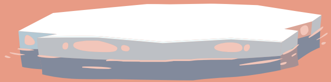
ls

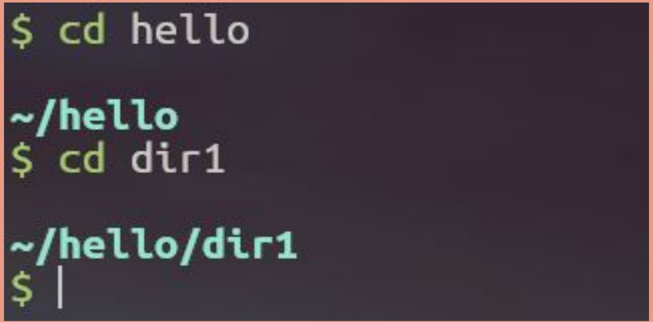




```
$ ls
  Android         Downloads        ISOs
  Desktop         hello            Linux-23-Tasks
  Documents       IdeaProjects     Materials
```

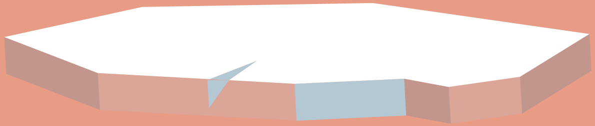
cd





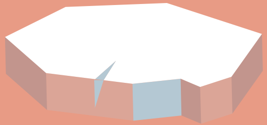
```
$ cd hello

~/hello
$ cd dir1

~/hello/dir1
$ |
```

cd /home/OSC

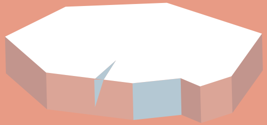


cd ~

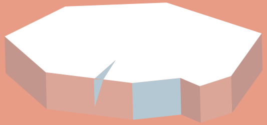
/home/OSC



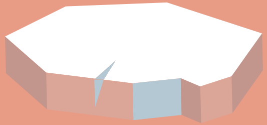
cd



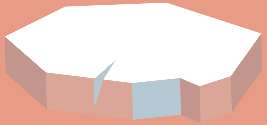
cd ..



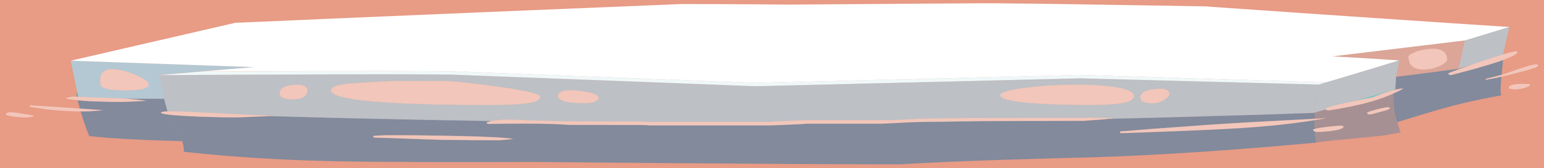
cd .

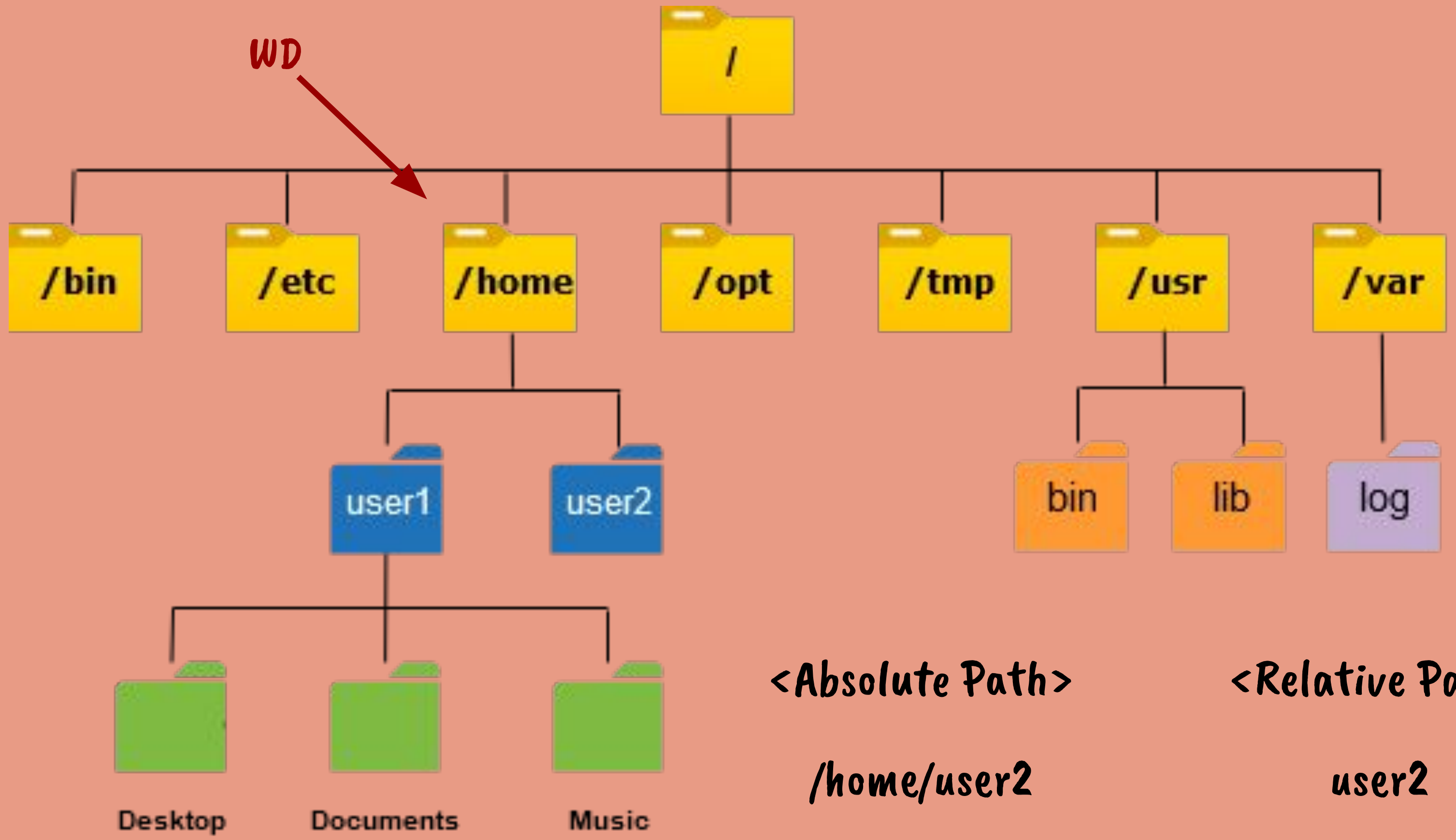


cd -



# Absolute and Relative Path



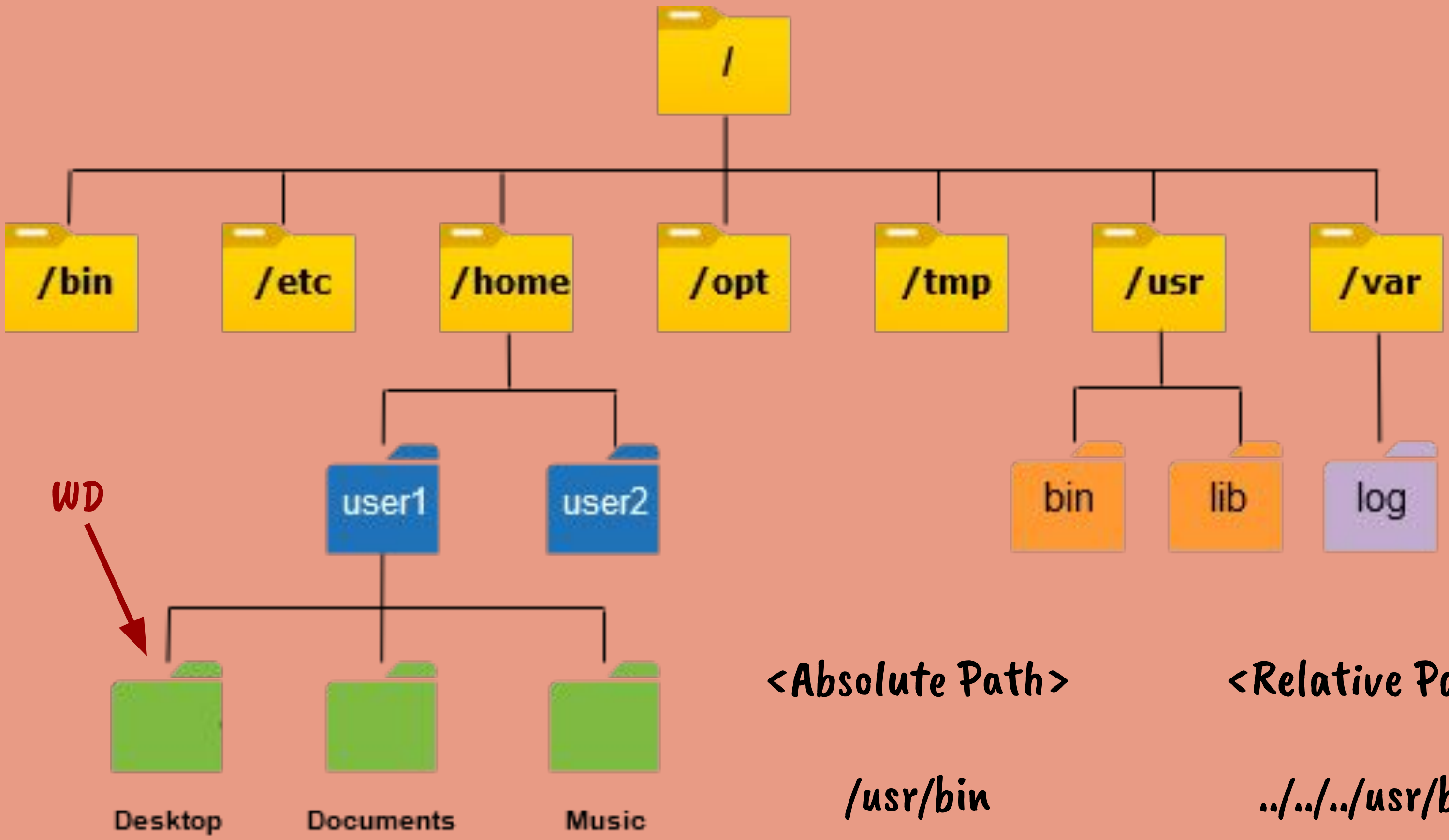


<Absolute Path>

/home/user2

<Relative Path>

user2



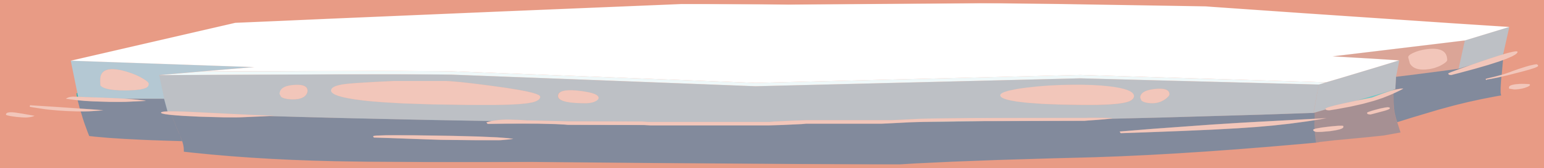
<Absolute Path>

/usr/bin

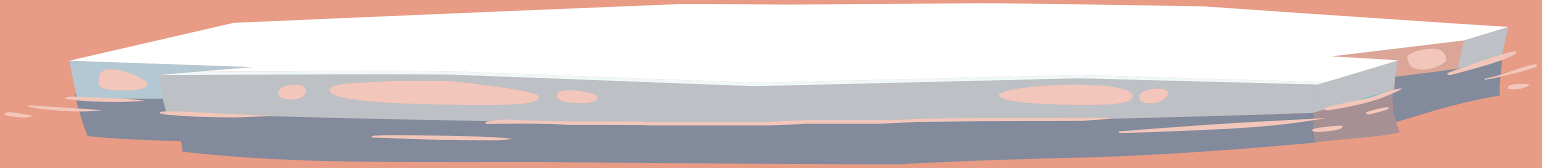
<Relative Path>

../../../usr/bin

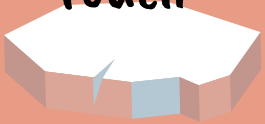
# BREAK

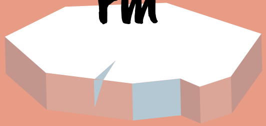


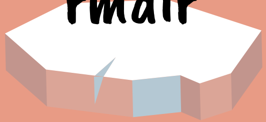
# File Management



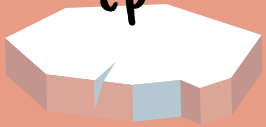
mkdir


touch

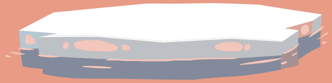

rm


rmdir

mv

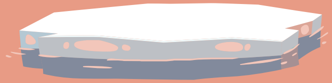

cp

mkdir



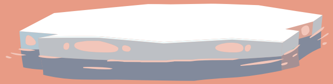
Create a directory

touch



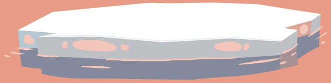
Create a file

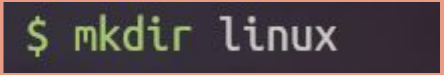
rm



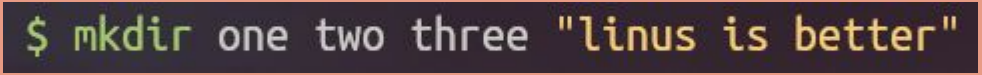
Removes files/directories

# mkdir

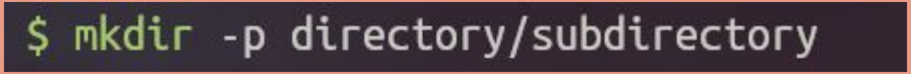




```
$ mkdir linux
```

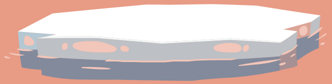


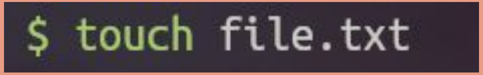
```
$ mkdir one two three "linus is better"
```



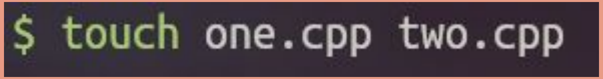
```
$ mkdir -p directory/subdirectory
```

# touch



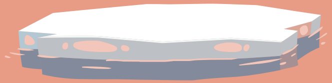


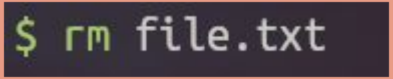
```
$ touch file.txt
```



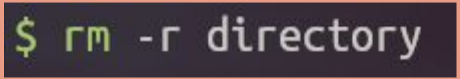
```
$ touch one.cpp two.cpp
```

# rm



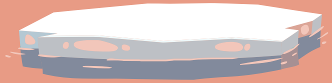


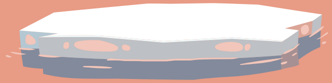
```
$ rm file.txt
```

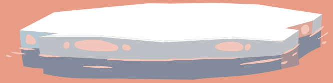


```
$ rm -r directory
```

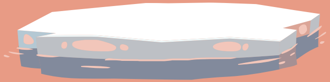
| | |
|---|---|
| **rmdir** | Delete empty directory |
| **mv** | Move/Rename files/directories |
| **cp** | Copy files/directories |

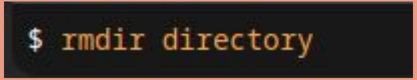






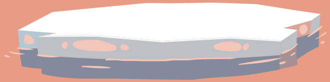
## rmdir

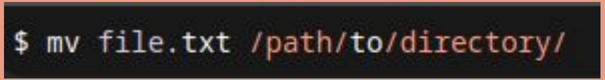




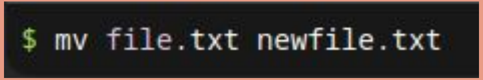
```
$ rmdir directory
```

## mv



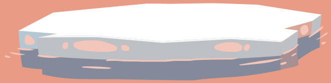


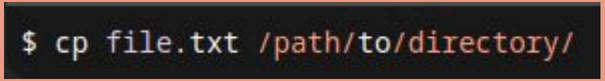
```
$ mv file.txt /path/to/directory/
```



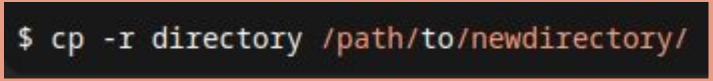
```
$ mv file.txt newfile.txt
```

## cp





```
$ cp file.txt /path/to/directory/
```



```
$ cp -r directory /path/to/newdirectory/
```

# Hands on

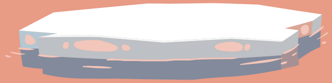
1. Create a directory named “Linux summer training”
2. Move into the new directory
3. Create a file with your first name and echo anything in it

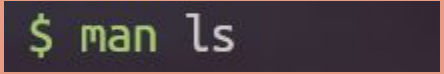
   (echo hello world > test)
4. Move out of this directory and copy it somewhere else
5. Go to the copied directory and delete everything on it
6. Move out of the copied directory and delete it

# Getting Help

| | |
|---|---|
| man | Open manual for any command |
| apropos | Search manual titles and description |
| whatis | Display one-line manual description |

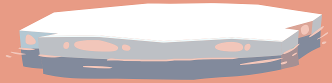
## man

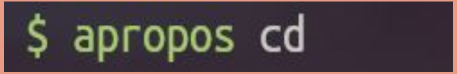




```
$ man ls
```

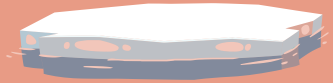
## apropos

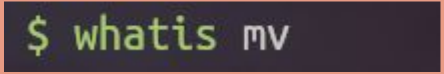




```
$ apropos cd
```

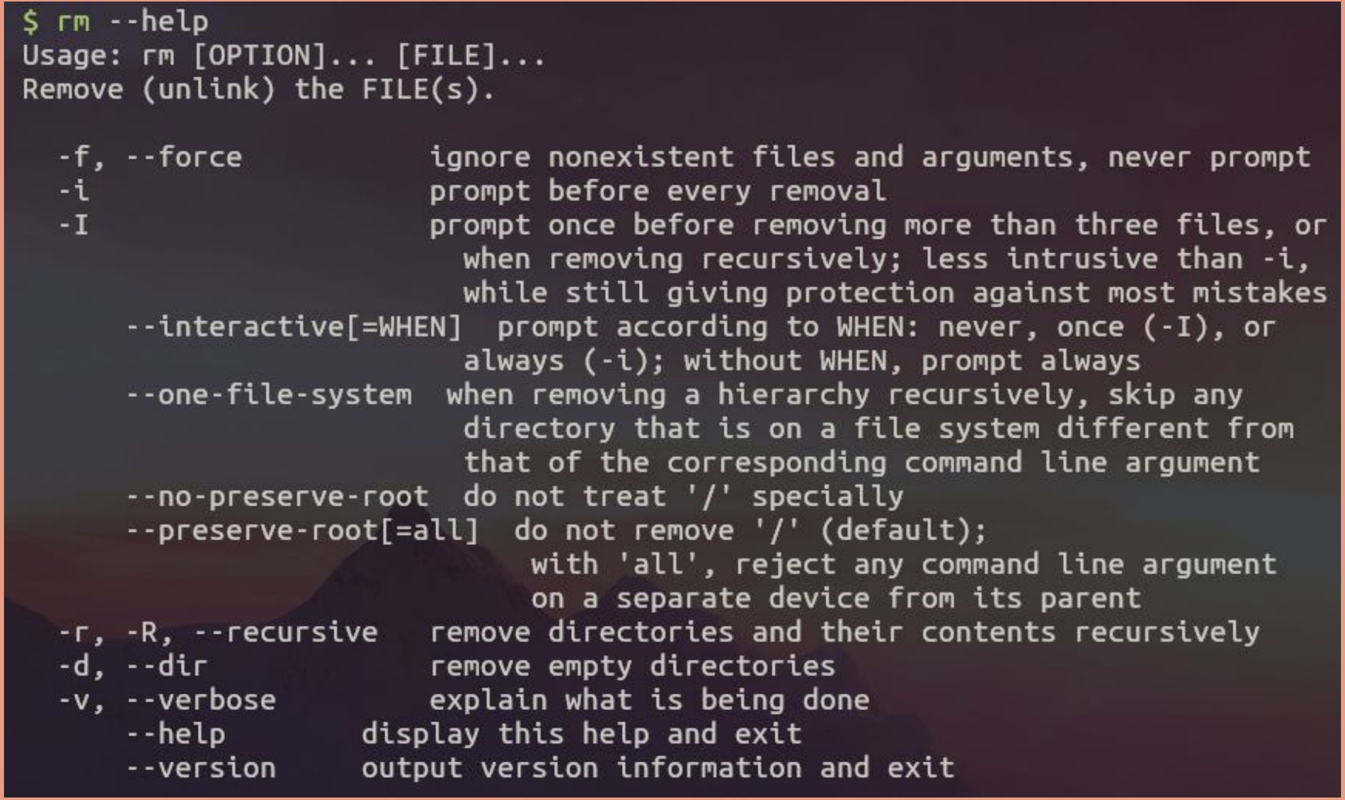
## whatis





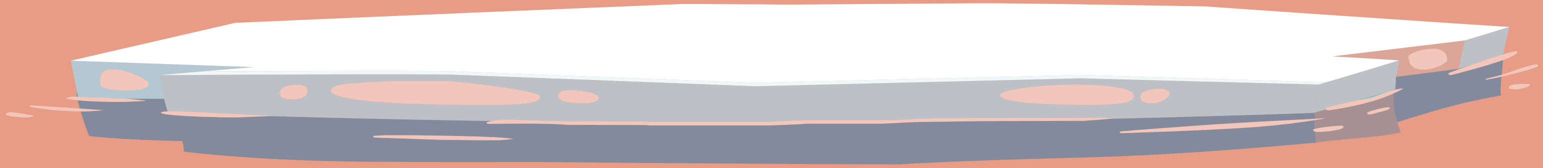
```
$ whatis mv
```

# --help

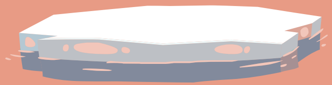


```
$ rm --help
Usage: rm [OPTION]... [FILE]...
Remove (unlink) the FILE(s).

  -f, --force           ignore nonexistent files and arguments, never prompt
  -i                    prompt before every removal
  -I                    prompt once before removing more than three files, or
                          when removing recursively; less intrusive than -i,
                          while still giving protection against most mistakes
      --interactive[=WHEN]  prompt according to WHEN: never, once (-I), or
                          always (-i); without WHEN, prompt always
      --one-file-system  when removing a hierarchy recursively, skip any
                          directory that is on a file system different from
                          that of the corresponding command line argument
      --no-preserve-root  do not treat '/' specially
      --preserve-root[=all]  do not remove '/' (default);
                              with 'all', reject any command line argument
                              on a separate device from its parent
  -r, -R, --recursive   remove directories and their contents recursively
  -d, --dir             remove empty directories
  -v, --verbose         explain what is being done
      --help        display this help and exit
      --version     output version information and exit
```

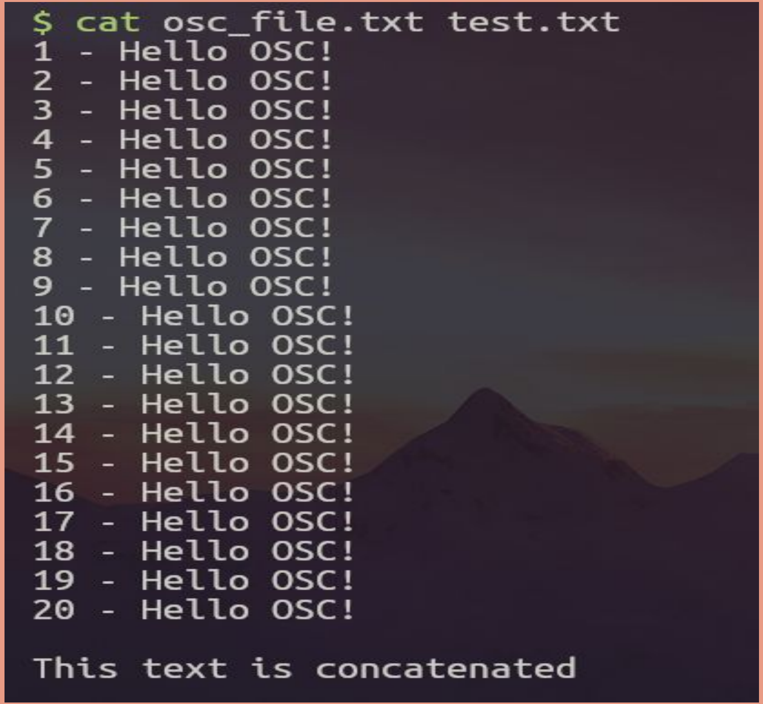
# Viewing File Contents



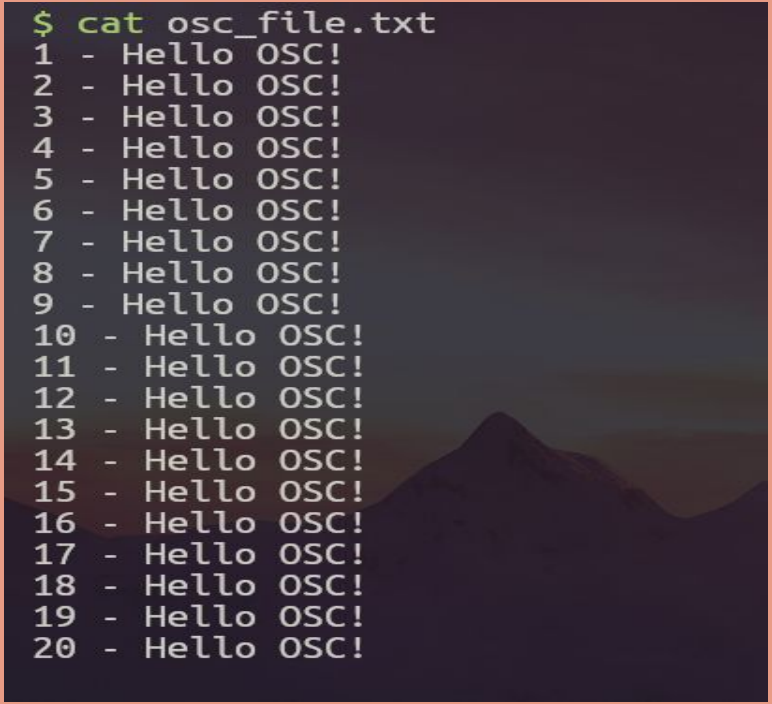
# cat



## Print and concatenate files.

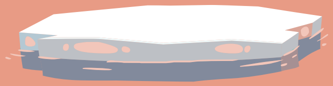


```
$ cat osc_file.txt test.txt
1 - Hello OSC!
2 - Hello OSC!
3 - Hello OSC!
4 - Hello OSC!
5 - Hello OSC!
6 - Hello OSC!
7 - Hello OSC!
8 - Hello OSC!
9 - Hello OSC!
10 - Hello OSC!
11 - Hello OSC!
12 - Hello OSC!
13 - Hello OSC!
14 - Hello OSC!
15 - Hello OSC!
16 - Hello OSC!
17 - Hello OSC!
18 - Hello OSC!
19 - Hello OSC!
20 - Hello OSC!

This text is concatenated
```

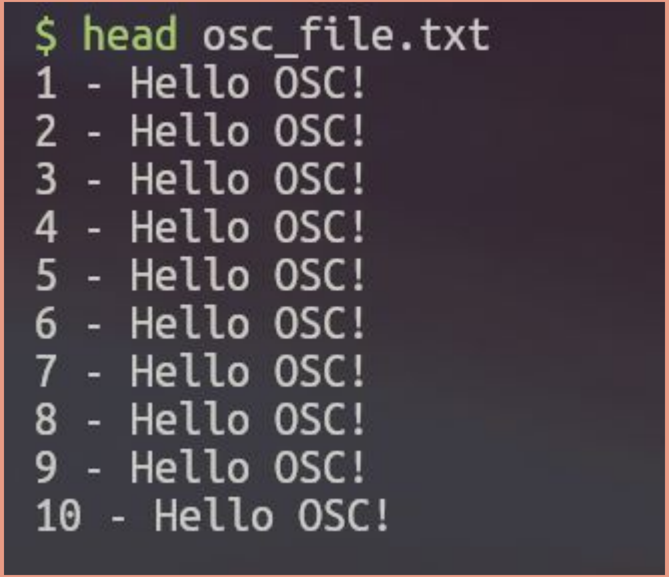


```
$ cat osc_file.txt
1 - Hello OSC!
2 - Hello OSC!
3 - Hello OSC!
4 - Hello OSC!
5 - Hello OSC!
6 - Hello OSC!
7 - Hello OSC!
8 - Hello OSC!
9 - Hello OSC!
10 - Hello OSC!
11 - Hello OSC!
12 - Hello OSC!
13 - Hello OSC!
14 - Hello OSC!
15 - Hello OSC!
16 - Hello OSC!
17 - Hello OSC!
18 - Hello OSC!
19 - Hello OSC!
20 - Hello OSC!
```

# head

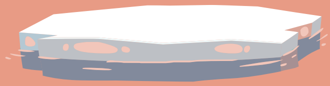


Display the first part of files.

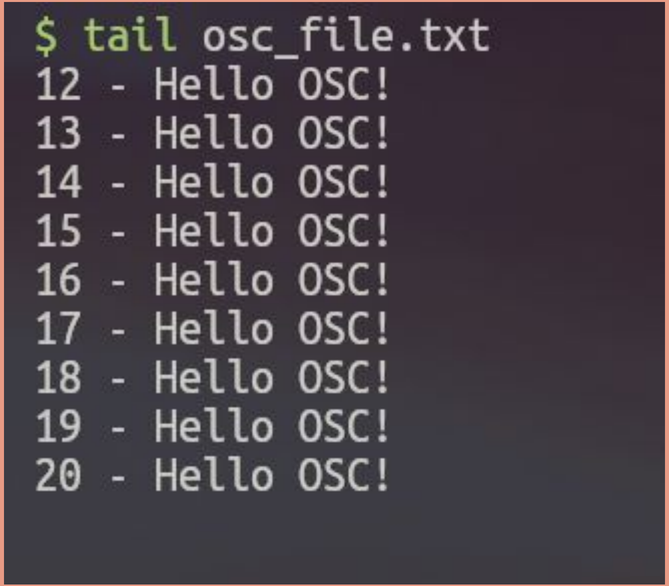


```
$ head osc_file.txt
1 - Hello OSC!
2 - Hello OSC!
3 - Hello OSC!
4 - Hello OSC!
5 - Hello OSC!
6 - Hello OSC!
7 - Hello OSC!
8 - Hello OSC!
9 - Hello OSC!
10 - Hello OSC!
```

# tail

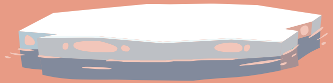


Display the last part of files.

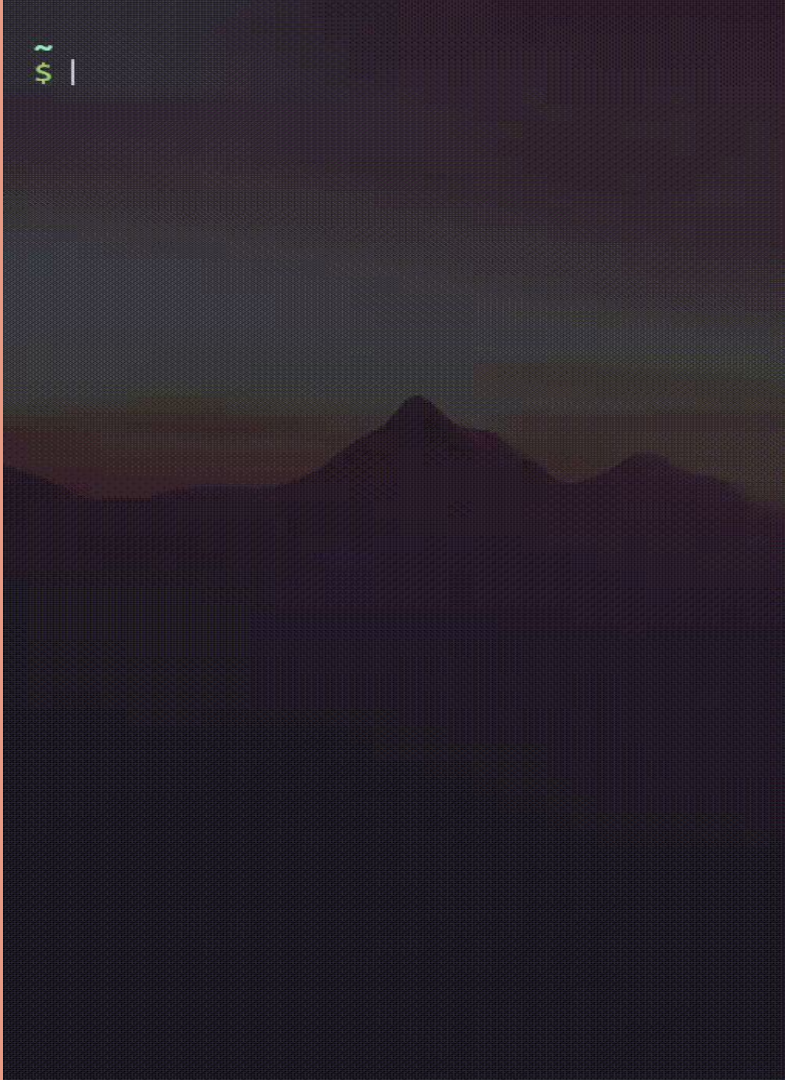


```
$ tail osc_file.txt
12 - Hello OSC!
13 - Hello OSC!
14 - Hello OSC!
15 - Hello OSC!
16 - Hello OSC!
17 - Hello OSC!
18 - Hello OSC!
19 - Hello OSC!
20 - Hello OSC!
```

# less



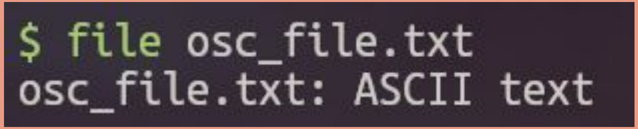
Open a file for interactive reading, allowing scrolling and search.



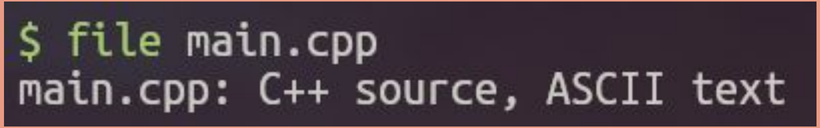
```
~
$ |
```

# file command

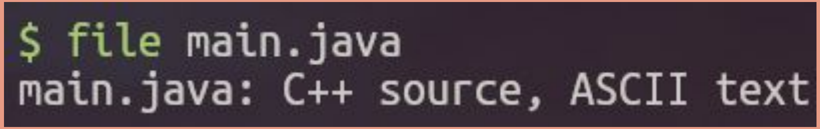
Give a description of the type of the specified file.



```
$ file osc_file.txt
osc_file.txt: ASCII text
```



```
$ file main.cpp
main.cpp: C++ source, ASCII text
```



```
$ file main.java
main.java: C++ source, ASCII text
```

# Hands on

1. create an empty file
2. concatenate it with the file from Hands-On 1
3. try man with any command
4. try apropos with any command
5. try whatis with any command
6. try file command on any file

# Agenda

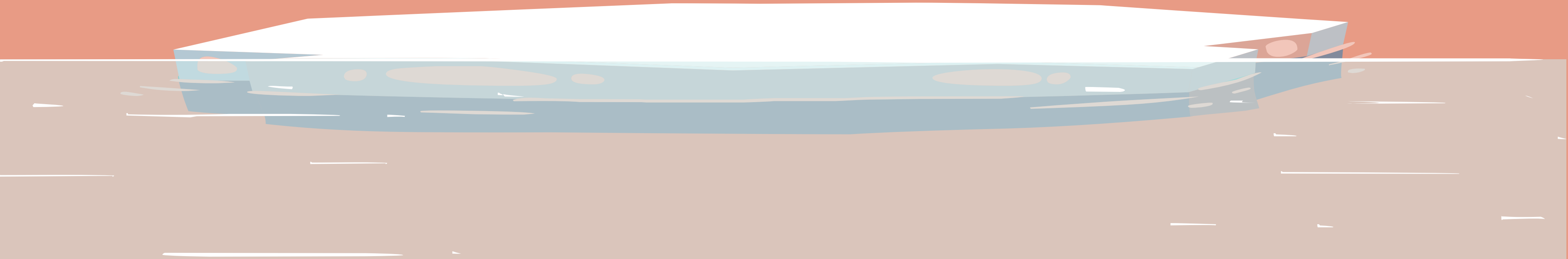
05

Commands

06

Text Editing with nano

# Types of Files



# Three Types

d

Directory Files

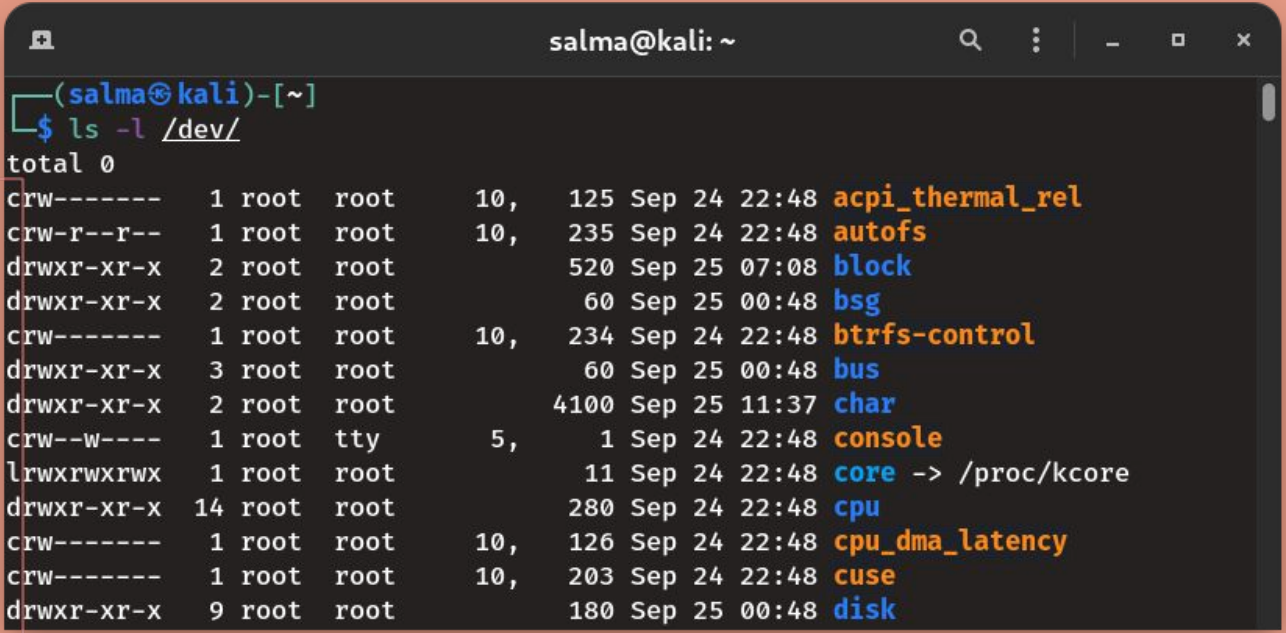
c / b / L

Special Files

-

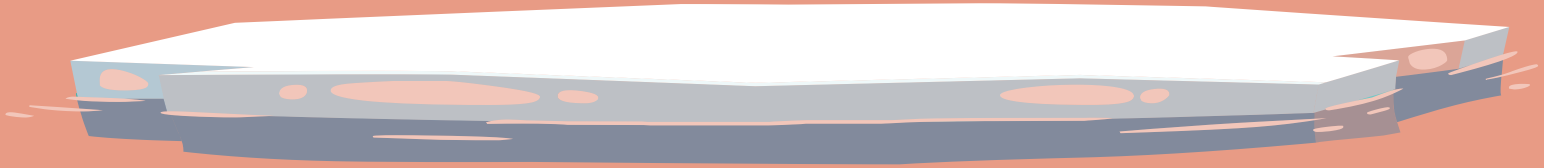
Regular Files

# Three Types



```
┌──(salma㉿kali)-[~]
└─$ ls -l /dev/
total 0
crw-------   1 root   root      10,   125 Sep 24 22:48 acpi_thermal_rel
crw-r--r--   1 root   root      10,   235 Sep 24 22:48 autofs
drwxr-xr-x   2 root   root            520 Sep 25 07:08 block
drwxr-xr-x   2 root   root             60 Sep 25 00:48 bsg
crw-------   1 root   root      10,   234 Sep 24 22:48 btrfs-control
drwxr-xr-x   3 root   root             60 Sep 25 00:48 bus
drwxr-xr-x   2 root   root           4100 Sep 25 11:37 char
crw--w----   1 root   tty        5,     1 Sep 24 22:48 console
lrwxrwxrwx   1 root   root             11 Sep 24 22:48 core -> /proc/kcore
drwxr-xr-x  14 root   root            280 Sep 24 22:48 cpu
crw-------   1 root   root      10,   126 Sep 24 22:48 cpu_dma_latency
crw-------   1 root   root      10,   203 Sep 24 22:48 cuse
drwxr-xr-x   9 root   root            180 Sep 25 00:48 disk
```

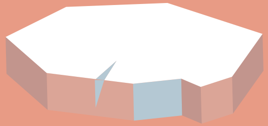
# File Extensions

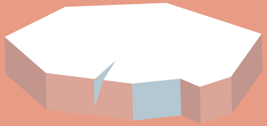


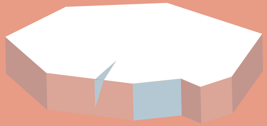
# Linux is extensionless

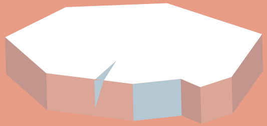
Unlike Windows, Linux does not care about the extension of your files. It looks into the file contents and will figure it out by its own.

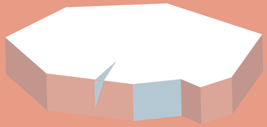
# Agenda



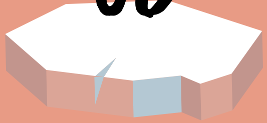






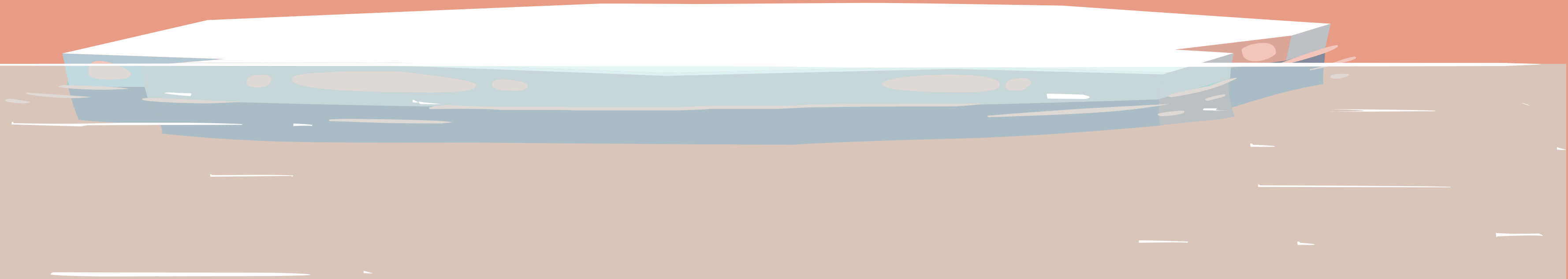


06



Text Editing with nano

nano



GNU nano is an easy to use command line text editor for Unix and Linux operating systems. It includes all the basic functionality you'd expect from a regular text editor, like syntax highlighting, multiple buffers, search and replace with regular expression support, spell checking, UTF-8 encoding, and more.

# Thanks!