

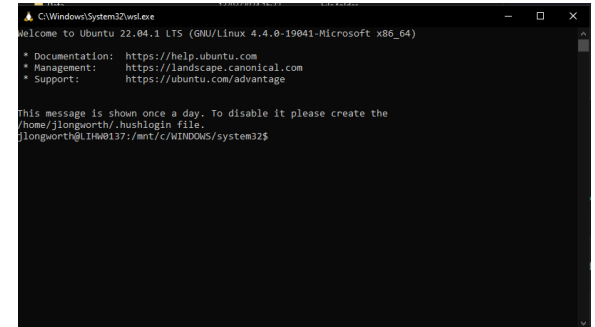
# Command line

terminal, command, prompt, shell, Bash

Joseph Longworth

# Why Use the Command Line?

- **Efficiency & Automation:** Execute complex tasks quickly with single commands. Automate repetitive actions with scripts.
- **Granular Control:** Leverage powerful tools built for bioinformatics not readily available through GUIs.
- Windows has a **CMD**, **terminal**, **powershell** but most access for a biologist is gained using the **Unix/Linux** based Command line.



```
C:\Windows\System32\cmd.exe
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 4.4.0-19041-Microsoft x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:   https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This message is shown once a day. To disable it please create the
/home/jlongworth/.hushlogin file.
jlongworth@LIH40137: /mnt/c:/WINDOWS/system32$
```

# What is Linux?

Linux is a free and open-source operating system (OS). Unlike Windows or macOS, it's not owned by a single company. Instead, it's developed by a community of programmers around the world.



Image of Linux penguin

# Who Created Linux?



Image of Linus Torvalds smiling

- Linus Torvalds, created Linux Kernel in 1991.
- Linux as a free alternative to the Unix operating system.
- **Linux distro:** Linux comes in a variety of versions know as a distribution ( distro ). Some popular distros include Ubuntu, Mint, and Fedora.

# Why use Linux?

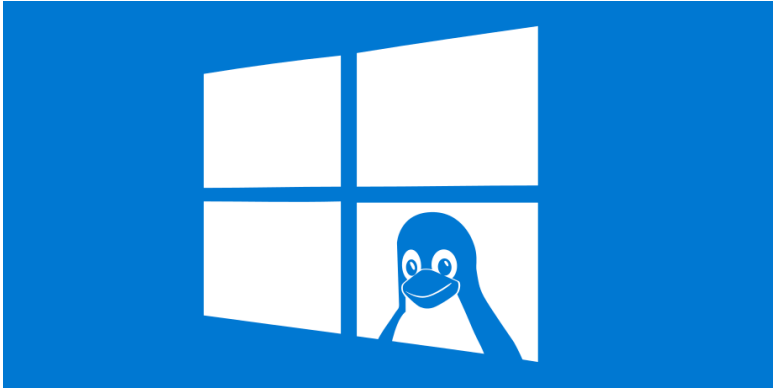
There are many reasons to use Linux, including:

- **Free and open-source:**
- **Good Security:**
- **Flexibility:** Used on a wide range of devices, from small embedded systems to powerful High performance computer **HPC**.
- **Large community:** Linux has a large and active community of users and **bioinformaticians**.

# Getting started with Linux

- **Install a linux based operating system:** This can be with a dedicated graphical user interface phase of simply text based.
- **Try it in a virtual machine (VM):** a software program that allows you to run another operating system on your computer.
- **Install it on a dual-boot system:** You can install Linux on a separate partition of your hard drive alongside your existing operating system.
- **WSL:** Windows subsystem for Linux .

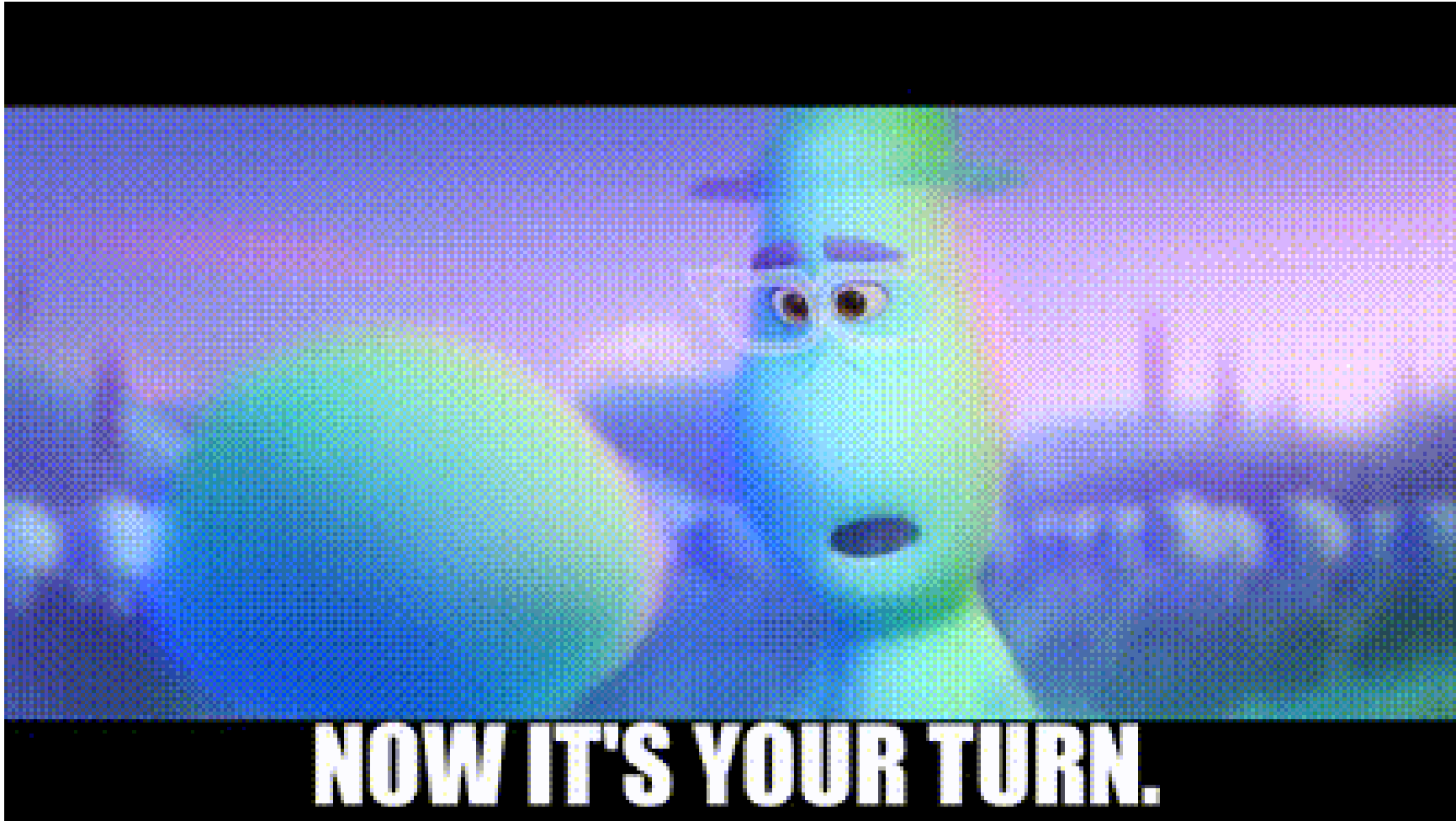
# Run Linux on Windows with WSL



- **Linux experience on Windows:** Allows you to execute a Linux environment directly within Windows 10 or 11.
- **No virtual machine needed:** Unlike traditional methods, WSL doesn't require setting up a separate virtual machine, simplifying the process.
- **Integration with Windows:** WSL integrates seamlessly with the Windows file system, enabling you to access and work with files from both environments.

# CLASS ACTIVITY 1

- Follow along while we activate a WSL terminal on your PC using the LIH based approach.  
(UNI should be simply launched as a programme?)





# Navigating Your Linux Filesystem

The Linux file system is organized in a hierarchical tree structure., with everything rooted at the `/` directory:

- **No Drive Letters:** Unlike Windows, Linux doesn't use drive letters (C:, D:, etc.). Storage devices are mounted under specific directories within the file system.
- **Strong Conventions:**
  - **/etc** (configuration files): Stores system-wide configuration files for programs and services.
  - **/bin** (essential binaries): Contains essential command-line utilities readily available to all users.
  - **/usr** ("user system resources"): Houses a vast collection of user-installable programs, libraries, and documentation.
  - **/home** (user home directories): Each user has a personal home directory under `/home`, typically named after their username, for storing personal files and configurations.

# Navigating the Linux File System

Linux uses several commands to navigate and explore the file system. Here are three essential ones:

- **pwd:** Stands for “print working directory”. This command tells you the exact location of the directory you’re currently in.
- **ls:** Stands for “list”. This command displays a list of files and folders within the current directory.
- **cd:** Stands for “change directory”. This command allows you to move between directories. For example, `cd Desktop` would move you to your Desktop folder.

# Navigating the Linux File System

- ~ Takes you to your home directory.
- . Refers to the current directory.
- .. Moves you up one level to the parent directory.

## Example:

- `./report.txt` = Opens report.txt in the current Work directory.
- `../Downloads` = Moves you to the Downloads directory (one level up and down to a different location e.g. Downloads).

## CLASS ACTIVITY 2

- Follow along while we relocate our working directory to the provided files.



# Create

- **mkdir:** Creates a new directory (i.e. Folder).

```
mkdir project_name
```

- **touch:** Creates an empty file.

```
touch README.
```

# Relocate

- **cp:** Copies files and directories.

```
cp existing_file new_file
```

- You can also copy directories with the **-r** flag

```
cp -r directory_to_copy new_location).
```

- **mv:** Moves (or renames) files and directories.

```
mv project_name new_project_name
```

# Destroy

- **rm:** Removes files and directories.

`rm unwanted_file.`

**Caution:** Use `rm` with care, as deleted files are usually unrecoverable.

# Power Up Your Commands with Flags

Linux commands often accept flags (hyphen followed by single letters or words) to modify their behavior.

- **Examples:**

- `ls -l` (long listing): Shows detailed file information.
- `cp -r source destination` (recursive copy): Copies entire directory structures.
- `grep -i "pattern" file` (case-insensitive search): Finds patterns ignoring case.



- **Tip:** Use `--help` or `man` with a command to explore its available flags.



# Don't Get Lost in the Command Line!

- **man:** The manual pages are your comprehensive guide to most Linux commands.

`man <command_name>`

- **help:** Many commands also have built-in help documentation. To access it, simply use

`<command_name> --help` or `<command_name> -h`.

**Remember:** Don't be afraid to experiment! The command line can be a powerful tool, and these resources are here to assist you in your exploration.

# sudo: The Key to Admin Power

In Linux, certain tasks require administrative privileges (root access) to modify system settings or install software. This is where `sudo` “superuser do” comes in.

- **How to use sudo:**

- To use `sudo`, type `sudo` followed by the command

`sudo apt-get update`

- You’ll be prompted to enter your user password. Once verified, the command will run with root permissions.

# CLASS ACTIVITY 3

1. Take 5 min. Create a few files and directories.
2. Move them around and Destroy them.
3. What happens if you create 3 files called file1 file2 and file3 then run `rm file*`?



# Nano: Text Editor

Nano is a command-line text editor pre-installed on most Linux distributions. It's known for its:

- **Simplicity:** Easy-to-use interface with clear on-screen instructions.
- **User-friendliness:** Great for beginners due to its intuitive navigation and editing features.
- **Versatility:** Suitable for editing various file types, from configuration files to code.
- **Efficiency:** Lightweight and efficient, making it ideal for resource-constrained systems.

# Nano: Text Editor

## Common Uses:

- Creating and editing plain text files.
- Modifying configuration files.
- Writing basic scripts.

# CLASS ACTIVITY 4

1. Open a terminal window.
2. Type `nano filename.txt` (replace filename.txt with your desired file name).
3. Edit your text.
4. Use the on-screen guide for saving and exiting (usually Ctrl+O to save and Ctrl+X to exit).



# Unveiling File Contents

- **cat:** Displays the entire contents of a file in one go. Need to see the whole picture?

```
cat my_file.txt
```

- **head/tail:** Peek at specific sections of a file:

- **head:** Shows the beginning (default: 10 lines)

```
head my_file.txt.
```

- **tail:** Displays the ending (default: 10 lines)

```
tail my_file.txt.
```

- You can adjust the number of lines displayed with the **-n** flag

```
head -n 5 my_file.txt
```

# Unveiling File Contents

- `more/less`: Navigate file contents a page at a time:
- `grep`: Search for specific patterns within files. Find that elusive line of code?

```
grep "pattern" my_file.txt
```



# CLASS ACTIVITY 5

1. use `wget` to download the latest fasta of your choice 2, explore the fasta with `cat`, `nano`, `tail`, `head`, `more` and `less`



# apt-get: Your Software Package Manager

Linux distributions like Ubuntu and Debian rely on package managers for software installation and updates. `apt-get` is a widely used package manager that simplifies this process.

## Key `apt-get` Commands:

- `sudo apt-get update`
- `sudo apt-get install <package_name>`
- `sudo apt-get remove <package_name>`
- `sudo apt-get upgrade`

# CLASS ACTIVITY 6

1. Navigate to the exercise folder provided “Figure\_prep”
2. Unzip each inkscape file with gunzip.
3. use inkscape to convert the files to pdf `inkscape -o JPL1.pdf JPL_1_Figures.svg`
4. Use Ghostscript to reduce the resolution of the file `gs -sDEVICE=pdfwrite -dCompatibilityLevel=1.4 -dPDFSETTINGS=/prepress -dNOPAUSE -dQUIET -dBATCH -sOutputFile=JPL1_opt.pdf JPL1.pdf`
5. Use pdfunite (and help/man to see how to use it) to merge your optimised pdfs `pdfunite JPL1_opt.pdf JPL2_opt.pdf JPL3_opt.pdf JPL4_opt.pdf Merged.pdf`

# Controlling Access: Unix Permissions and chmod

- Each file and directory has three permission levels:
  - **User (u)**: Controls access for the file/directory owner.
  - **Group (g)**: Controls access for the group the file/directory belongs to.
  - **Others (o)**: Controls access for everyone else.
- Each permission level has three permissions:
  - **Read (r)**:
  - **Write (w)**:
  - **Execute (x)**:
- Permissions can be changed with chmod and a code describing level to grant:
  - 1 = execute
  - 2 = write
  - 4 = read

```
chmod 744 <filename>
```

# Automating Tasks with Bash Scripting

The Bash shell in Linux isn't just for single commands. You can harness its power to create scripts – a series of commands bundled into a file – to automate repetitive tasks.

## Simple Script Example:

```
#!/bin/bash  
  
sudo apt update  
  
sudo apt upgrade  
  
sudo apt autoclean  
  
echo "System updated and cleaned!"
```

- Save the script as a file (e.g., update\_script.sh).
- Make the script executable: `chmod +x update_script.sh`.
- Execute the script: `./update_script.sh`

# CLASS ACTIVITY 7

1. Ask you favorite LLM to create you a script to run the three lines from the previous exercise on all the files within a directory.
2. Try to understand the suggestion operations line by line
3. Use Nano to copy the script into your folder with the inkscape files
4. Does is work?
5. Try the version provided coalled conversion.sh



