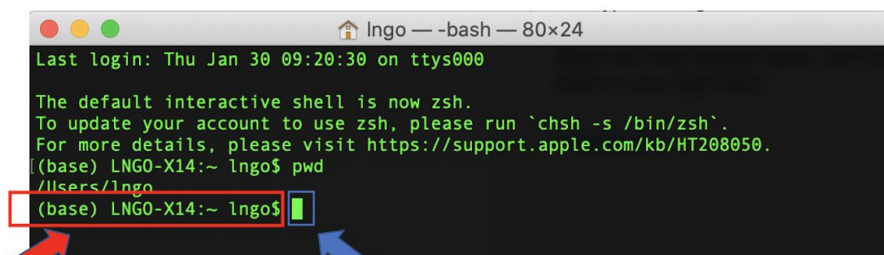This module is meant to be taught in a hands-on manner, with frequent pause between slides for students to practice. Assuming a novice level, students can easily get lost and fall behind, especially in activities that require navigation across different directories.

Linux Terminal (5 minutes):

- Remote access: Large-scale computational resources often need to be maintained in dedicated environments and shared among multiple researchers. Direct graphical access with mouse and keyboard in a manner similar to a traditional desktop/laptop is not possible. Interactions with these resources are often done via a command line interface (CLI), called the terminal (general term).
- Linux terminal is the CLI interface to interact with Linux-based systems.

- This portion is called the prompt, indicating the place where users can be gin typing in command.
- The last character of the prompt (before the user-input commands) is often a $ or # sign.

- This is the cursor. A flashing cursor indicating that it is now possible to type in commands.
- After finish typing a command, the user must hit the Enter key to tell the terminal to begin executing that command.

- In literature, the prompt is shortened to the last character, represented by either $ or #.
- Users interact with the system by typing in commands and hitting Enter. This process is called REPL:
  - The system READS the command.
  - The system EVALUATES the command.
  - The system PRINTS the result of the command (or error messages if this is not a correct command) to the screen.
  - The system LOOPS back to the terminal prompt again to wait for the next command.
- **Exercise:**
  - In a terminal, types **"Hello, World"** and hits Enter. Expect an error.

- In a terminal, types `echo "Hello, World"` and hits Enter. Observe the result.
- Make the terminal display another message.

Directories and file structures in Linux (10 minutes):

- **Path**: a sequence of directories connected together by the slash "/" character. A path is used in navigation (changing from one directory to another) or data access purpose (reading/writing to a file). The beginning of the path is the starting directory, and the ending of the path is the final destination (directory or files). Files can only be listed at the end of a path, and cannot be in the middle of the path.
    - The root directory of a Linux (or Unix-based) file system is also represented by a "/".
    - If the path begins with "/", it means that we start the path at the root of the file system. This is called an **absolute path**.
    - If the path does not begin with "/", it means that we start the path at the current directory. This is called a **relative path**.
- **Working directory**: When users open a terminal to interact with the system and run the "`ls`" command, some contents will be displayed. This is because users' terminal is "mapped" to a directory location within the file system. Typically, terminals initially placed users into their home directories. As users navigate to a different directory location, the content of "ls" will change as the terminal is now "mapped' to this new location. The directory that the terminal currently maps to is called the current working directory. The command to show the current working directory with respect to the root directory is "**pwd**".
- Special designations: There are three designations that can be used to refer to specific file system locations. They are: "~", ".", and ".." (*tilde*, *dot*, and *dot dot*)
    - "~": Users home directory
    - ".": The current directory.
    - "..": The directory one level above the current directory.
- To view directory and file contents, we can use the "**ls**" command (listing).
    - Running `ls` by itself
    - Running `ls` using one of the special designations
    - Running `ls` using a relative path
    - Running `ls` using an absolute path.
    - Running `ls` with the `-l` flag.
    - Running `man ls` to show documentation of the command.

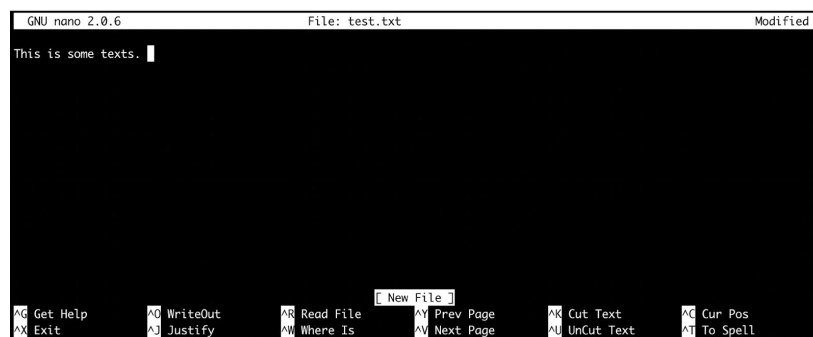## Navigating inside the Linux file system (5 minutes):

- The command to navigate from directory to directory is "**cd**" (change directory). Users enter the command and a path (absolute or relative). If the path is correct and accessible, the terminal will change its current directory to that path.
- Instructors show how absolute, relative, and special destinations can be used to navigate around. Combine with pwd to demonstrate directory changes.

Create directories:

- The command to create a new directory is "**mkdir**" (make directory). Users enter the command the path to where users want to create a new directory.
- It is expected that all directories on the path must exist.
- It is possible to create new directories on the path if they are not already existing by calling the **-p** flag

Create and edit files:

- There are several terminal-based text editors to help create and edit text files (vi/vim,emacs, nano…).
- Among the editors, nano is perhaps the one with the least steep learning curve.
- To use nano to create a file named test.txt, run the command "**nano test.txt**"

```
  GNU nano 2.0.6                      File: test.txt                          Modified

This is some texts. █









                                      [ New File ]
^G Get Help     ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text     ^C Cur Pos
^X Exit         ^J Justify      ^W Where Is     ^V Next Page    ^U UnCut Text   ^T To Spell
```

- To save the newly entered content, hit the key combination **Ctrl-X**, then **Y**, and finally **Enter**.

```
  GNU nano 2.0.6                         File: test.txt                              Modified

This is some texts.













Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?
 Y Yes
 N No              ^C Cancel
```

- To view the text file, you can use the cat command: "`cat test.txt`"

```
(base) LNGO-X14:~ lngo$ cat test.txt
This is some texts.
```

Delete files and directories:

- To delete the `test.txt` file, type "`rm test.txt`".

```
(base) LNGO-X14:~ lngo$ ls test.txt
test.txt
(base) LNGO-X14:~ lngo$ rm test.txt
(base) LNGO-X14:~ lngo$ ls test.txt
ls: test.txt: No such file or directory
(base) LNGO-X14:~ lngo$
```

- To delete directory, use "**rm -Rf directory_name**"
- Show error messages when try to delete empty directories