Introduction to Linux/Unix (Part 1)

Tinghua Chen
IT Research CyberInfrastructure (IT-RCI)

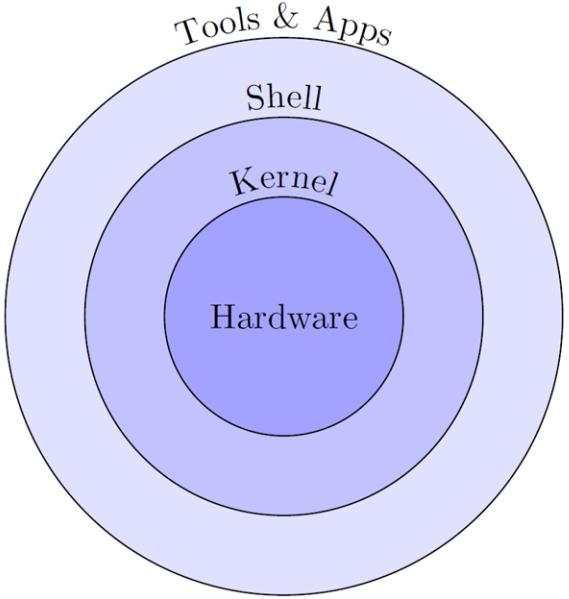
University of Delaware

Workshop Overview

- Linux/Unix Operating System
- Command-line Interface
- Files and Directories
- Input/Output Redirection

Linux/Unix Operating System

- Kernel:
  - Core part of Linux OS
  - Manage resources
  - Between hardware and shell
- Shell:
  - Interface between user and kernel
  - Interpret commands and execute them
  - Type of Shell
    - Bash, zsh, csh, etc.



Linux first released in 1991

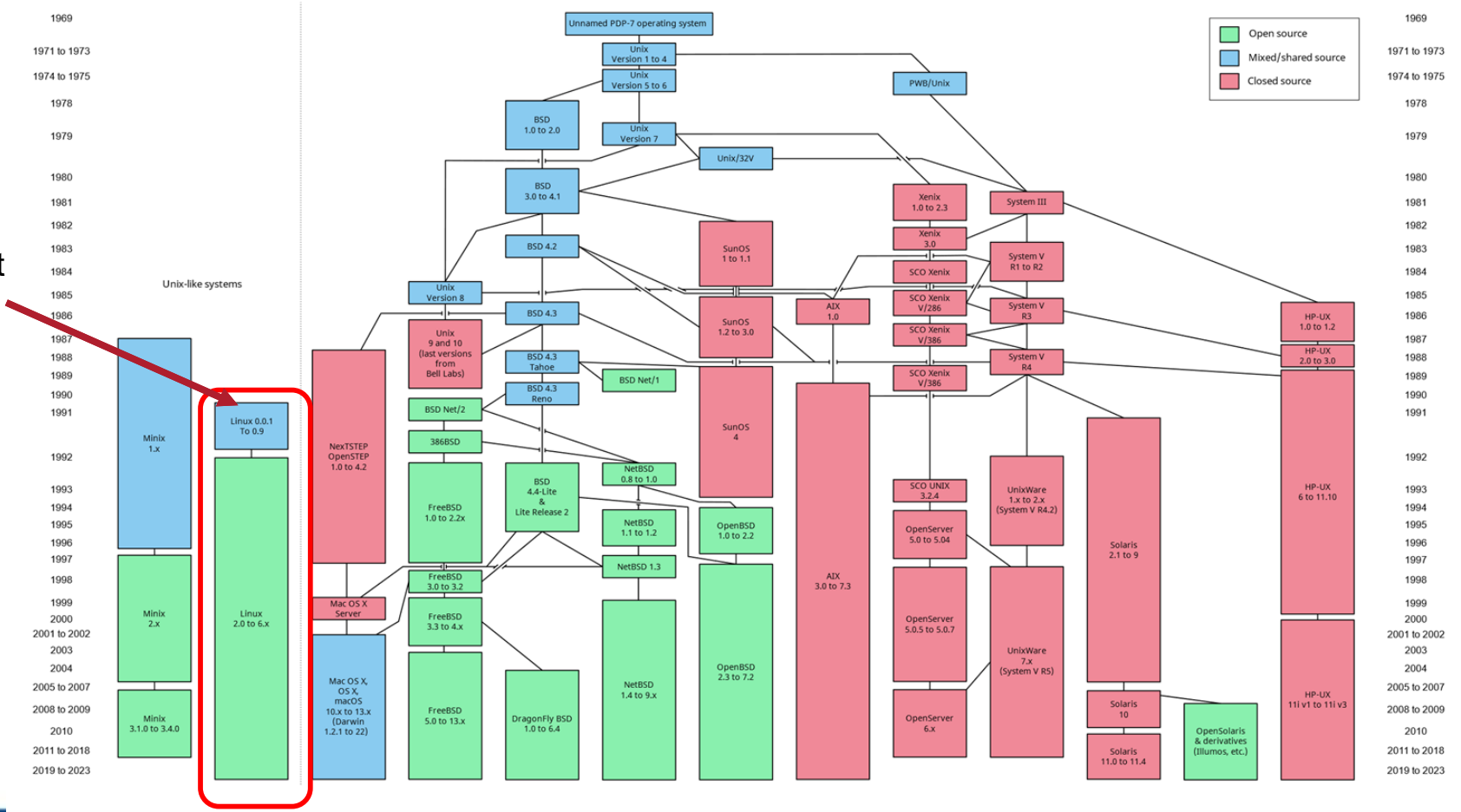
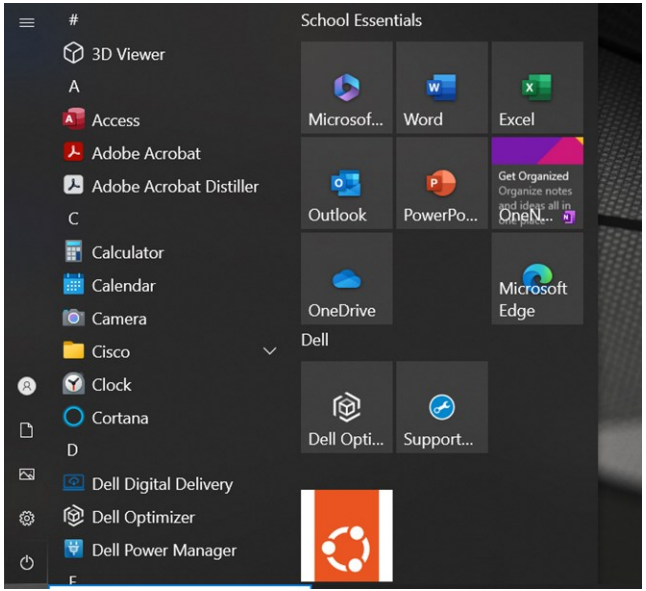
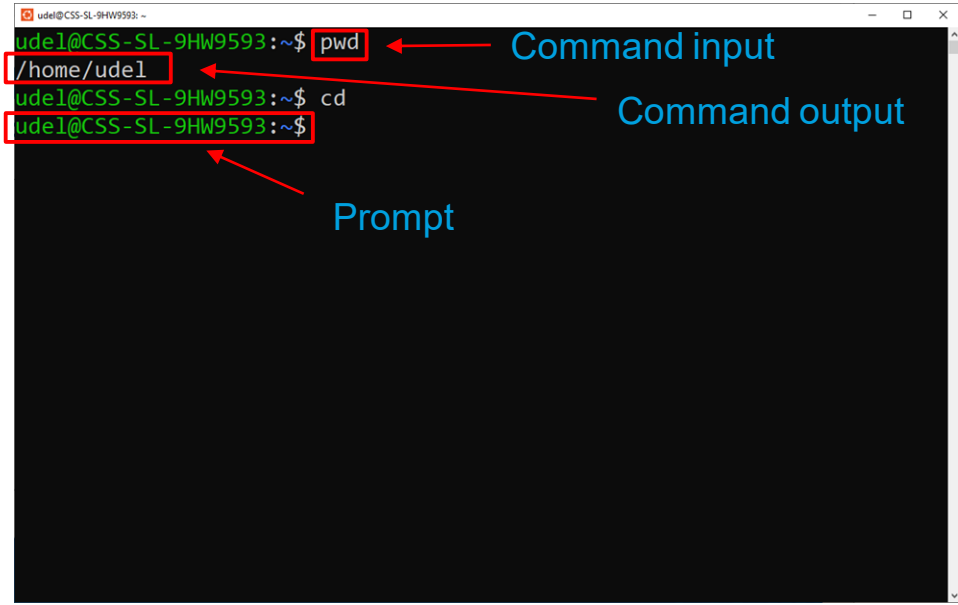


Image by Eraserhead1, Infinity0, Sav_vas.
https://commons.wikimedia.org/wiki/File:Uix_history-simple.svg

Graphical User Interface (GUI)



Command-line Interface (CLI)



General Command Syntax

- `command option(s) argument(s)`
- There may be zero or more options
- Example commands:
  - `pwd`
  - `echo 'Hello World'`
  - `echo -n "Hello World"`
  - `ls -la`

Finding Help for Commands

- `man <command>`
  - Open the manual pages
- `<command> --help`
- `info <command>`
  - Display information in the document format
- `apropos <keyword>`
  - Search the descriptions for the installed command
- Google

Effectively using Linux Command Line

- Linux is case-sensitive
- It is a good practice to avoid SPACES in filenames
- Tab completion
  - Automatically complete filenames, directory names, and commands

```
udel@CSS-SL-9HW9593:~/test-1$ ls
myfile.txt  Myfile.txt
udel@CSS-SL-9HW9593:~/test-1$ ls myFile.txt
ls: cannot access 'myFile.txt': No such file or directory
udel@CSS-SL-9HW9593:~/test-1$ ls myfile.txt
myfile.txt
udel@CSS-SL-9HW9593:~/test-1$
```

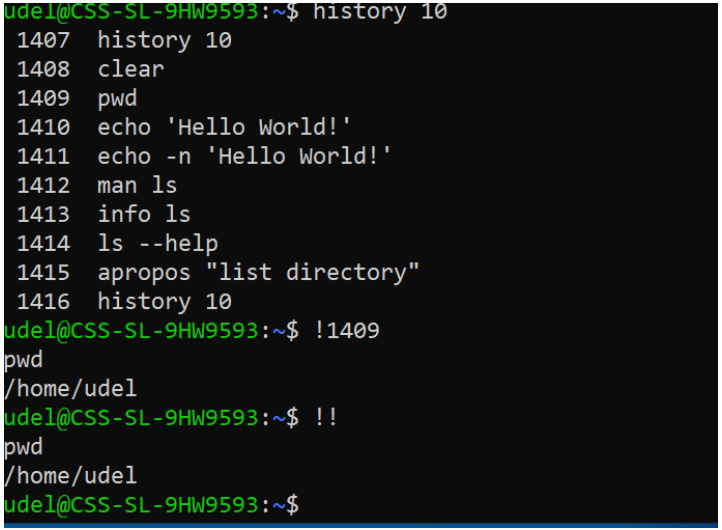
Exercise 1

- Try typing the following commands in the terminal:
  - `pwd`
  - `echo 'Hello World'`
  - `echo -n "Hello World"`
  - `ls -la`
  - `man ls`

- **`ls`**
  - List files and directories
  - `'-l'`: Long format, providing detailed information about each file or directory
  - `'-a'`: Lists all files, including hidden ones (starting with a dot)
  - `'-h'`: Human-readable format, displaying file sizes in a human-readable format

```
udel@CSS-SL-9HW9593:~/demo$ ls -lah
total 16K
drwxr-xr-x  3 udel udel 4.0K Apr  1 11:55 .
drwxr-x--- 19 udel udel 4.0K Apr  1 11:56 ..
drwxr-xr-x  2 udel udel 4.0K Apr  1 11:55 dir1
-rw-r--r--  1 udel udel   52 Apr  1 11:54 list.txt
lrwxrwxrwx  1 udel udel    8 Apr  1 11:55 sym_link -> list.txt
```

- Use command `history [option]`
  - i.e., `history 10`, displays the 10 previous commands in the history
  - Exclamation mark (`!`) followed by the command number in history to rerun the command
  - Double exclamation mark (`!!`) to rerun the last command



Keyboard Shortcuts

- Ctrl + a: Move to the beginning of the line.
- Ctrl + e: Move to the end of the line.
- Ctrl + c: Interrupt the current command.
- Ctrl + u: Delete text from cursor to line start.
- Ctrl + k: Delete text from cursor to line end.
- Up/Down Arrow: Recall the previous command.

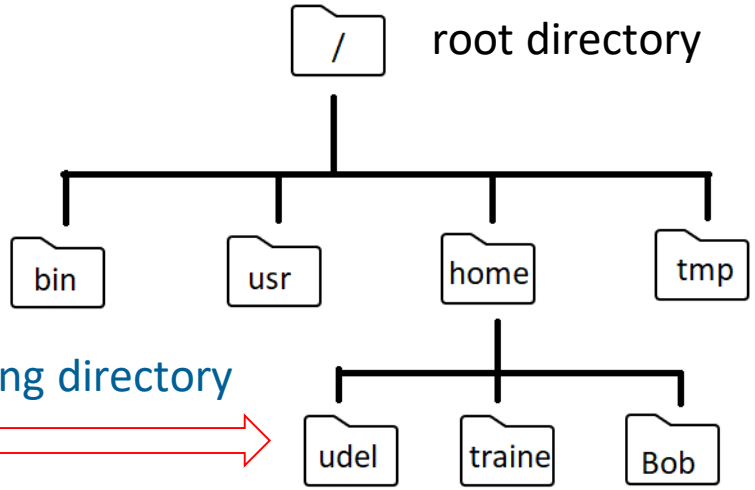
Exercise 2

- The command `touch` can be used to create an empty file. Try to create an empty file **`ex2.txt`**
- Try `touch Ex 2.txt`. Does this command create the desired file? The command `ls` can be used to view the contents of a directory.
- Use the `history` command to show last 5 commands used.
- The command `sleep 300` will pause the terminal for 300 seconds. How do you stop it to get back to the shell prompt?

Directory Structure

Linux's principle: Everything is a file

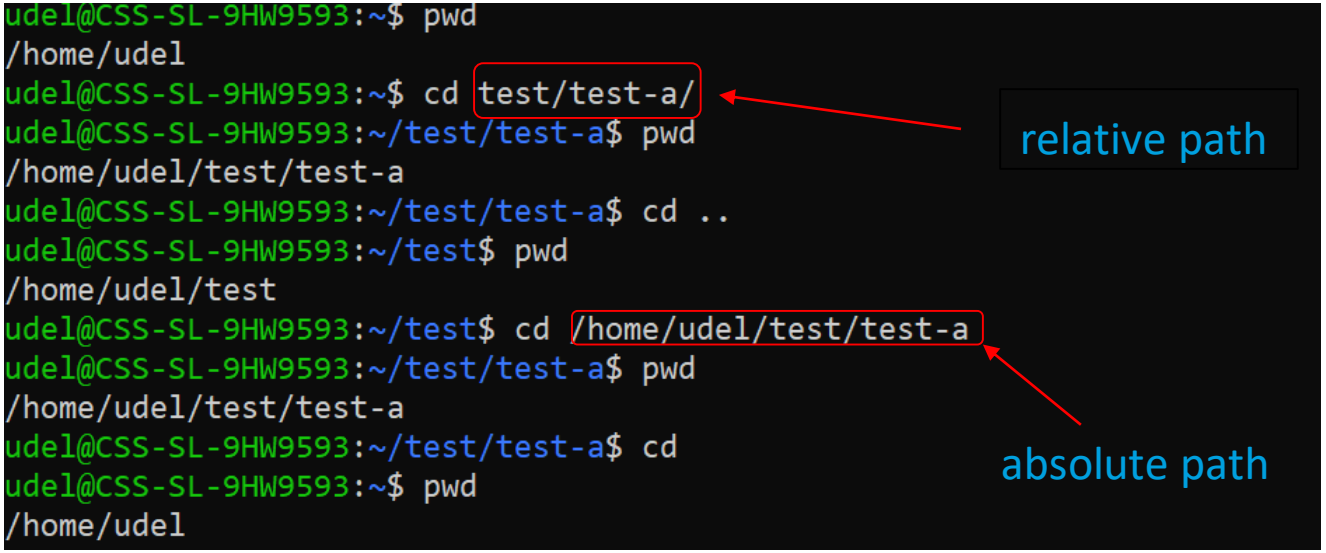
- absolute path
  - begin from / of the file system
- relative path
  - begin from the current working directory
  - Two dots (..) represent parent directory
  - Single dot (.) represents the current working directory



Basic File Operations

- **mkdir/rmdir**
  - Create/delete directories
  - Syntax: `mkdir [option] <directory>`
  - `'-p'`: create parent directory if needed, e.g., `mkdir -p train_data/data`
  - Syntax: `rmdir <directory>`, e.g., `rmdir train_data`

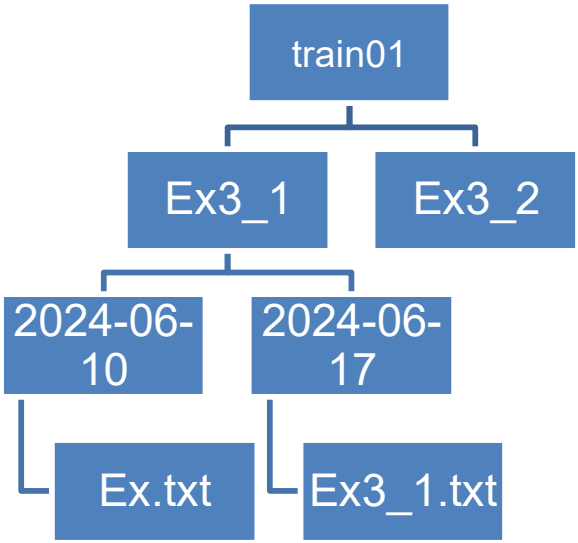
```
udel@CSS-SL-9HW9593:~/test-1$ mkdir dir1
udel@CSS-SL-9HW9593:~/test-1$ ls dir1/
udel@CSS-SL-9HW9593:~/test-1$ rmdir dir1/
udel@CSS-SL-9HW9593:~/test-1$ ls dir1
ls: cannot access 'dir1': No such file or directory
```



- **cd**
  - Change directories
  - Syntax: `cd <directory>`, e.g., `cd /home/Bob`
  - Back to parent directory: `cd ..`
  - Back to the previous directory: `cd -`

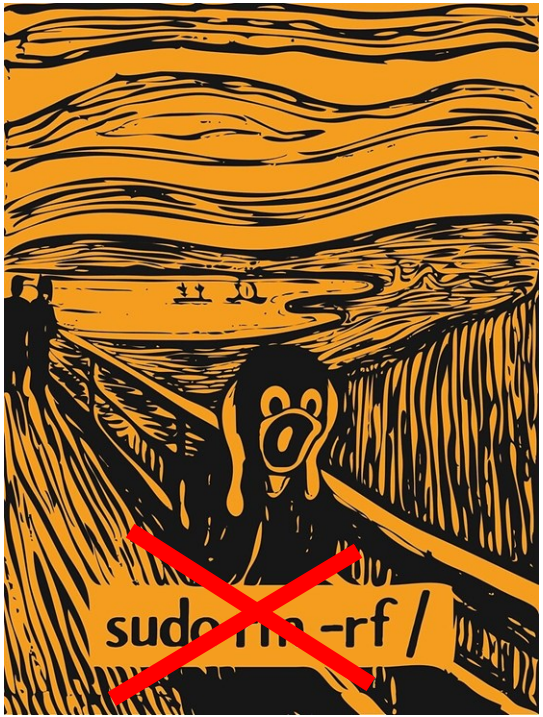
Exercise 3.1

- Use `mkdir`, `touch` commands to create the directories and files as the filesystem diagram shown on the right. (`/home/train01` represents the user's home directory)
- Starting from the `/home/train01/Ex3_1/2024-06-17`, which of the following commands could you use to navigate to the home directory, which is `/home/train01`?
  - `cd .`
  - `cd /`
  - `cd /home/train01`
  - `cd ../..`



- **cp**
  - Copy files and directories
  - Syntax: `cp [option] <source> <destination>`
  - e.g., `cp sample.txt sample_dir`
  - `'-a'`: copy files recursively and preserve symbolic links and date/time stamps
  - e.g., `cp -a <source> <destination>`
- **mv**
  - Move files/directories to a new location
  - Syntax: `mv [option] <file1> <destination>`
  - e.g., `mv sample.txt /home/train01`
  - Rename files/directories
  - e.g., `mv sample.txt sample_1.txt`

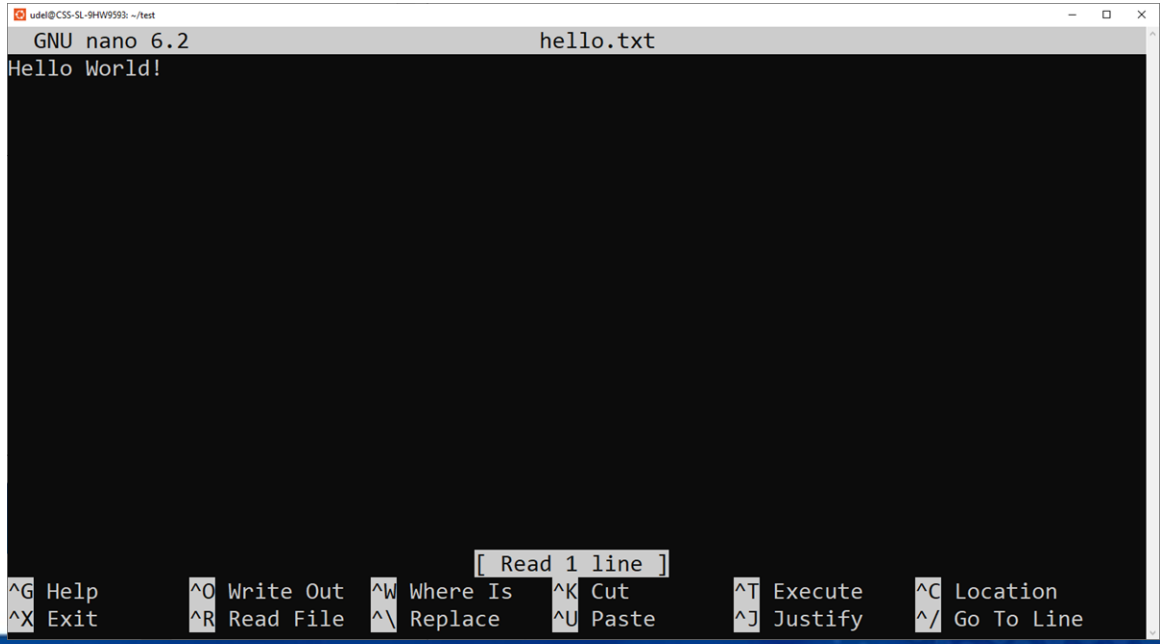
- **rm**
  - Removes files or directories, be caution to use
  - Syntax: `rm [option] <file1>`
  - e.g., `rm sample.txt`
  - `'-i'`: add interactive flag for confirmation
  - `'-r'`: remove file recursively, can be dangerous
  - e.g., use with `'-i'` option, `rm -ri <dir1>`



http://www.redbubble.com/people/uman

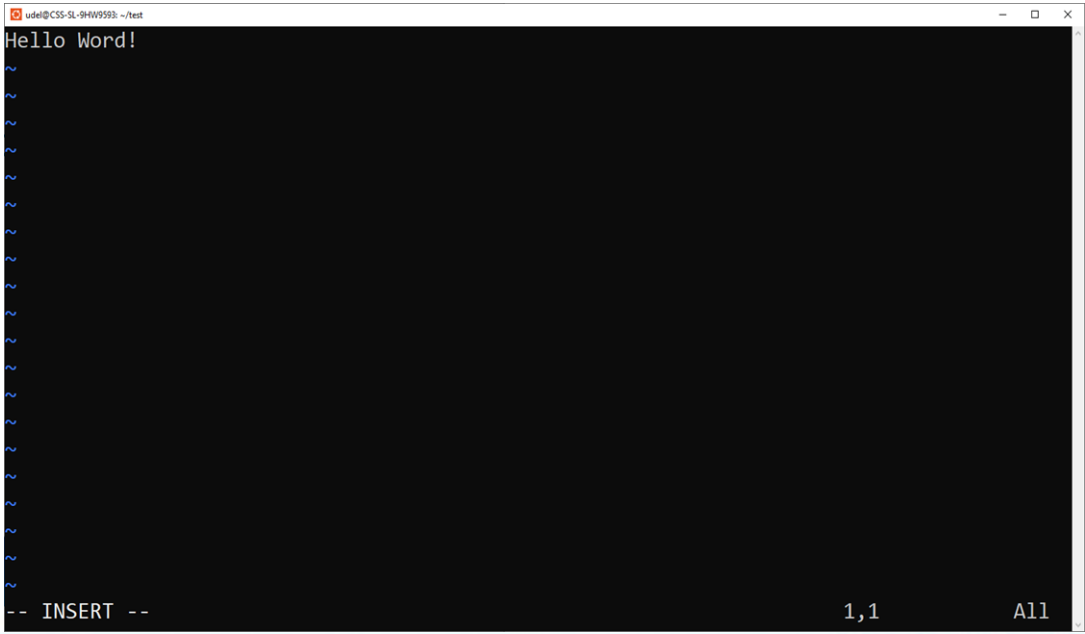
Text Editor: nano

- `nano <filename>`
  - Ctrl + x: exit
  - Ctrl + g: open the help menu
  - Ctrl + w: search the word
  - Ctrl + k: cut the text
  - Ctrl + u: paste the text



Text Editor: vim

- `vim <filename>`
  - Insert mode to add or edit text, by using `'i'`
  - Esc + `':wq'`: save the changes and exit
  - Esc + `':q!'`: quit without saving the last operation
  - Esc + `':w'`: save the file



- **cat**
  - Display file contents, e.g., `cat sample.txt`
- **ln**
  - Make a link to a file
  - `'-s'` creates symbolic link (soft link), e.g., `ln -s sample.txt sample_link`

Exercise 3.2

- Copy `/home/train01/Ex3_1/2024-06-17/Ex3_1.txt` to the directory **`Ex3_2`** and rename it as **`Ex3_2.txt`**
- Use the text editor to add the line "`This is the first line.`" to the file **`Ex3_2.txt`**
- Make a symbolic link **`Example3_2`** for the file **`Ex3_2.txt`**. Use the text editor to change the contents of **`Ex3_2.txt`**, will the contents of the symbolic link **`Example3_2`** change? If you delete the file **`Ex3_2.txt`**, will it delete the symbolic link **`Example3_2`**?

Wildcards

- *(asterisk)
  - Represent any number of characters
  - e.g., `ls a*.txt`
- ?(question mark)
  - Represent any single character, e.g., `ls a?.txt`
- [](square brackets)
  - Represent any single character within the specified range or set
  - e.g., `ls a[a-e].txt`
- {}(curly brackets)
  - Used for multiple matches, e.g., `mv file{1,2,3}.txt dir1/`

Exercise 4

- Create the directory **Ex4** and use `touch` to create the files: **doc1.txt, doc2.txt, doc3.txt, note1.txt, note2.txt**
- List and verify all files were created
- Create a directory **dir2.** Copy all files that start with “doc” to **dir2**
- Use the `rm` command with wildcards to remove the files: **doc1.txt, doc2.txt, doc3.txt**.

Input/Output Redirection

- > (Output Redirection)
  - Redirects standard output to a file
  - e.g., `ls > output.txt`
- < (Input Redirection)
  - Redirects standard input from a file
  - e.g., `sort < input.txt`
- >> (Append Output)
  - Appends standard output to a file
  - e.g., `ls >> output.txt`

Input/Output Redirection

- 2> (Error Redirection)
  - Redirects the standard error
  - e.g., `ls file1.txt 2> error.txt`

```
udel@CSS-SL-9HW9593:~/test-1$ ls file1.txt
ls: cannot access 'file1.txt': No such file or directory
udel@CSS-SL-9HW9593:~/test-1$ ls file1.txt > error.txt
ls: cannot access 'file1.txt': No such file or directory
udel@CSS-SL-9HW9593:~/test-1$ ls file1.txt 2> error.txt
udel@CSS-SL-9HW9593:~/test-1$ cat error.txt
ls: cannot access 'file1.txt': No such file or directory
```

- **diff**
  - Compare two files and display the difference
  - Syntax: *`diff [option] file1 file2…`*
  - "<" refers to the content in file 1, ">" refers to the content in file 2
  - Line numbers corresponding to the file 1
  - a (add), c (change), d (delete)
- **wc**
  - Word count, e.g., *`wc hello.txt`*
  - `'-l'` count number of lines
  - `'-w'` count number of words

- **sort**
  - Sort file contents and print output
  - Syntax: *`sort [option] file`*
  - `'-u'` to eliminate duplicate entries, `'-f'` to sort case-insensitively
  - `'-r'` to sort in reverse order
- **grep**
  - Search text patterns within files
  - Syntax: *`grep [option] pattern [file]`*
  - `'-i'` for case-insensitively search, `'-w'` for match the whole word

Exercise 5

- Create the file **`fruits.txt`** containing a list of the following fruits: `apple`, `watermelon`, `grape`, `banana`, `cherry`, `blueberry`.
- Append "`pineapple`" to **`fruits.txt`**.
- Sort the file **`fruits.txt`** alphabetically and save the sorted list to **`fruits_sorted.txt`**.
- Using the `diff` command compare **`fruits.txt`** and **`fruits_sorted.txt`**
- Using pipes, display all lines of **`fruits_sorted.txt`** containing the string "`apple`", and count the number of occurrences.

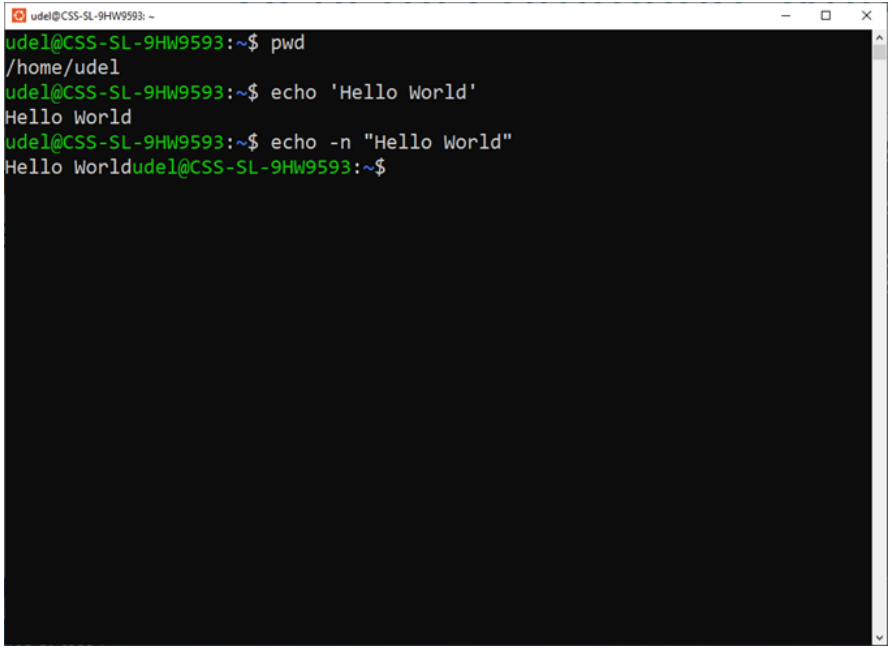
Bonus Exercise: Install Linux on Your Laptop

- For Windows Users: Try Installing WSL (Windows Subsystem for Linux)
- Enable WSL:
  - Open PowerShell as Administrator.
  - Run: `wsl --install`
- Install a Linux Distribution: Choose and install from the Microsoft Store (e.g., Ubuntu).
- Open the terminal and enjoy.

Try It Yourself First!

Before looking at the solutions for each exercise, please take a few moments to attempt to do the exercises on your own.

Exercise 1


udel@CSS-SL-9HW9593: ~
udel@CSS-SL-9HW9593:~$ pwd
/home/udel
udel@CSS-SL-9HW9593:~$ echo 'Hello World'
Hello World
udel@CSS-SL-9HW9593:~$ echo -n "Hello World"
Hello Worldudel@CSS-SL-9HW9593:~$

Exercise 2

```
udel@CSS-SL-9HW9593:~/example$ touch ex2.txt
udel@CSS-SL-9HW9593:~/example$ ls
ex2.txt
udel@CSS-SL-9HW9593:~/example$ touch Ex 2.txt
udel@CSS-SL-9HW9593:~/example$ ls
2.txt  Ex  ex2.txt
udel@CSS-SL-9HW9593:~/example$ history 5
 2045  touch ex2.txt
 2046  ls
 2047  touch Ex 2.txt
 2048  ls
 2049  history 5
```

Exercise 3_1

```
train01@craig-MacPro:~$ mkdir -p Ex3_1/2024-06-10 Ex3_1/2024-06-17 Ex3_2
train01@craig-MacPro:~$ ls
Ex3_1  Ex3_2  Ex4
train01@craig-MacPro:~$ cd Ex3_1/
train01@craig-MacPro:~/Ex3_1$ ls
2024-06-10  2024-06-17
train01@craig-MacPro:~/Ex3_1$ touch 2024-06-10/Ex.txt 2024-06-17/Ex3_1.txt
train01@craig-MacPro:~/Ex3_1$ cd 2024-06-17/
train01@craig-MacPro:~/Ex3_1/2024-06-17$ ls
Ex3_1.txt
train01@craig-MacPro:~/Ex3_1/2024-06-17$ cd ../..
train01@craig-MacPro:~$ pwd
/home/train01
train01@craig-MacPro:~$ cd -
/home/train01/Ex3_1/2024-06-17
train01@craig-MacPro:~/Ex3_1/2024-06-17$ cd /home/train01
train01@craig-MacPro:~$ pwd
/home/train01
train01@craig-MacPro:~$
```

Exercise 3_2

```
train01@craig-MacPro:~$ ls
Ex3_1  Ex3_2  Ex4
train01@craig-MacPro:~$ cp Ex3_1/2024-06-17/Ex3_1.txt Ex3_2/Ex3_2.txt
train01@craig-MacPro:~$ cd Ex3_2/
train01@craig-MacPro:~/Ex3_2$ ls
Ex3_2.txt
train01@craig-MacPro:~/Ex3_2$ nano Ex3_2.txt
train01@craig-MacPro:~/Ex3_2$ ln -s Ex3_2.txt Example3_2
train01@craig-MacPro:~/Ex3_2$ cat Example3_2
This is the first line.
train01@craig-MacPro:~/Ex3_2$ echo "Hello!" >> Ex3_2.txt
train01@craig-MacPro:~/Ex3_2$ cat Ex3_2.txt
This is the first line.
Hello!
train01@craig-MacPro:~/Ex3_2$ cat Example3_2
This is the first line.
Hello!
train01@craig-MacPro:~/Ex3_2$ rm Ex3_2.txt
train01@craig-MacPro:~/Ex3_2$ cat Example3_2
cat: Example3_2: No such file or directory
train01@craig-MacPro:~/Ex3_2$
```

Exercise 4

```
train01@craig-MacPro:~$ mkdir Ex4
train01@craig-MacPro:~$ cd Ex4
train01@craig-MacPro:~/Ex4$ touch doc{1..3}.txt note{1,2}.txt
train01@craig-MacPro:~/Ex4$ ls
doc1.txt  doc2.txt  doc3.txt  note1.txt  note2.txt
train01@craig-MacPro:~/Ex4$ mkdir dir2
train01@craig-MacPro:~/Ex4$ cp doc*.txt dir2/
train01@craig-MacPro:~/Ex4$ ls
dir2  doc1.txt  doc2.txt  doc3.txt  note1.txt  note2.txt
train01@craig-MacPro:~/Ex4$ rm doc*.txt
train01@craig-MacPro:~/Ex4$ ls
dir2  note1.txt  note2.txt
train01@craig-MacPro:~/Ex4$
```

Exercise 5

```
train01@craig-MacPro:~$ mkdir Ex5
train01@craig-MacPro:~$ cd Ex5
train01@craig-MacPro:~/Ex5$ nano fruits.txt
train01@craig-MacPro:~/Ex5$ echo "pineapple" >> fruits.txt
train01@craig-MacPro:~/Ex5$ sort < fruits.txt > fruits_sorted.txt
train01@craig-MacPro:~/Ex5$ diff fruits.txt fruits_sorted.txt
2,3d1
< watermelon
< grape
5d2
< cherry
6a4,5
> cherry
> grape
7a7
> watermelon
train01@craig-MacPro:~/Ex5$ cat fruits_sorted.txt | grep "apple" | wc -l
2
train01@craig-MacPro:~/Ex5$ cat fruits_sorted.txt
apple
banana
blueberry
cherry
grape
pineapple
watermelon
```

Need Help?

The Unix Shell
https://swcarpentry.github.io/shell-novice/

The Linux Command Line for Beginners
https://ubuntu.com/tutorials/command-line-for-beginners#1-overview

Linux/Unix Tutorial
https://www.geeksforgeeks.org/linux-tutorial/

HPC wiki
https://docs.hpc.udel.edu/

Google group: hpc-ask

Research Computing Help Request

Thank you!

For more information, contact:

askit@udel.edu