Intro to Linux Command-Line: The File System

Introduction: Introduce users to typing commands using the command line to work with the Linux operating system. Focusing on hands-on exercises, it will introduce the basic structure and use of the file system, and how to find help.

Course Goals

- To introduce users (*who have never used Linux*) to the Linux OS and command line environment.
- Use basic Linux commands from a command line interface (CLI) within a terminal.
- How to find help on a particular command.
- Understand what a File System is and be able to navigate around, list folder contents, create folders, move, copy and delete files/folders.
- Introduce file/folder permissions and ownership.

Sections

1. Getting Started with Linux: Introduction to what Linux is.
2. Using the Terminal: Introduction to using a Terminal.
3. The Linux File System: Introduction to the Linux File System, its structure and how to navigate around it, as well as creating, moving and copying files and folders.
4. Next Steps and Summary: Point to resources and trainings to continue learning Linux. Summarize what's been covered.

Getting Started with Linux

Goal: Introduction to what Linux is.

What is Linux and Linux Distributions (distro)

What is an Operating System?

- When you turn your device on, it boots up the operating system, which manages the communication/interface between your applications and the hardware it is running on.
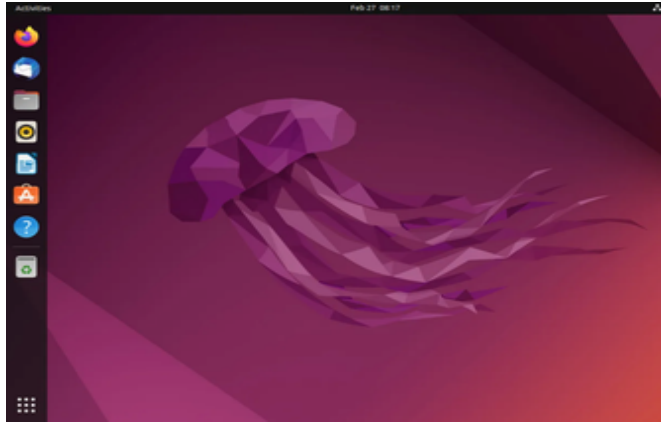
What is Linux?

- Linux is an Operating Systems – similar to Windows, Mac OS, iOS, Android.
- Linux is open-source – freely available – so you can download, modify and redistribute.

Varieties of Linux Distributions (distros):

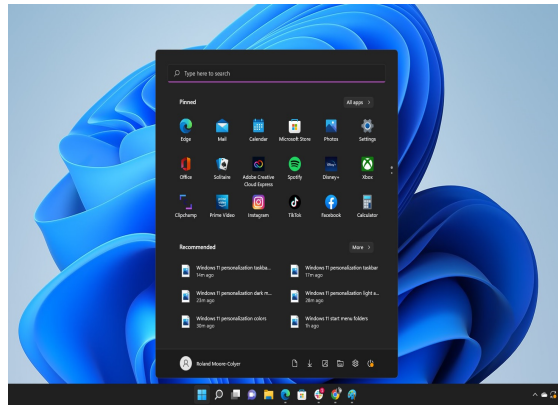|	
Debian • **Ubuntu (based on Debian)**	
Fedora • Amazon Linux 2 • Commercial: Red Hat (which we are using today) • Rocky Linux A lot of commonality across distros.	

Types of Environments

Desktop: *Windows* **type Graphical User Interface (GUI) - Graphics based interactions between the user and the operating system.** • **Mouse pointer, click, icons, windows, drag and drop.**	

Terminal: Program that opens a window to runs a:
 	
• **Shell** which is a *command interpreter* that processes the typed commands.
   o Interface to the OS.
   o Provides a **Command-Line Interface (CLI)** – written/text-based input and output.
   o Different Shells share common commands, but syntax and behavior can be different.	

Using the Terminal

Goals: Introduction to using a Terminal

- What does a prompt look like?
- General syntax of shell command.
- Commands/options are case sensitive.
- Getting Help:
    - Man pages (`man`)
    - Options: `<command> --help`

Login

1. Open up Chrome

2. Navigate to: MedicineBow OnDemand
3. Type in your provided username and password. Usually this will be your UWYO username and password, unless you are using an assigned training account.
4. Authenticate using your preferred 2 factor method (expandable directions below):

⌄ Duo Mobile push:

If you usually get a two-factor push to your phone, just hit enter after entering your username and password, then complete authentication by approving the push on your device.
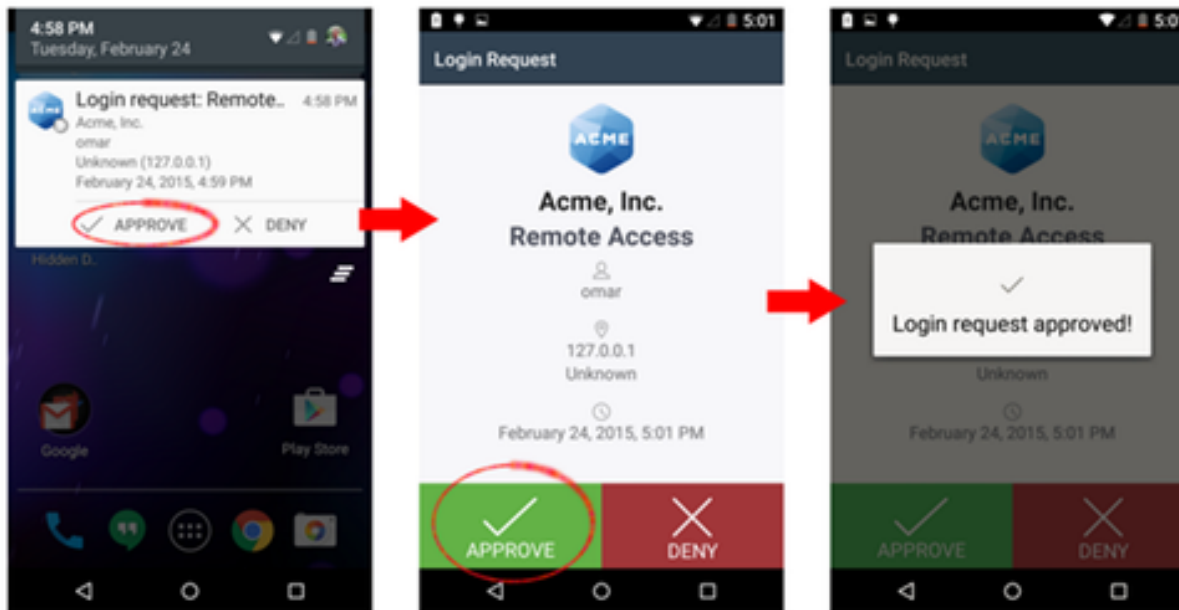
⌄ Phone Call:

Without hitting enter after typing in your username and password, in the password text box, append a comma (,) to the end of your password, then append phone as shown in the screenshot below:

arcc-t01

password,phone

☒ Remember me

Sign In

You should get a phone call on your main phone # associated with your two factor account. Answer this call and hit # to approve access.

⌄ Duo Passcode

If you prefer to use a 2 factor passcode from your Duo Mobile app, without hitting enter after typing in your username and password, in the password text box, append a comma (,) to the end of your password, then append the multi digit passcode found in duo mobile as shown in the screenshot below:

arcc-t01

password,12345678

☒ Remember me

Sign In

⌄ Yubikey:

Type in the account password, then, without hitting enter, append a comma (,) to the end of the password, then touch the light on the yubikey as shown in the screenshot and photo below:

arcc-t01

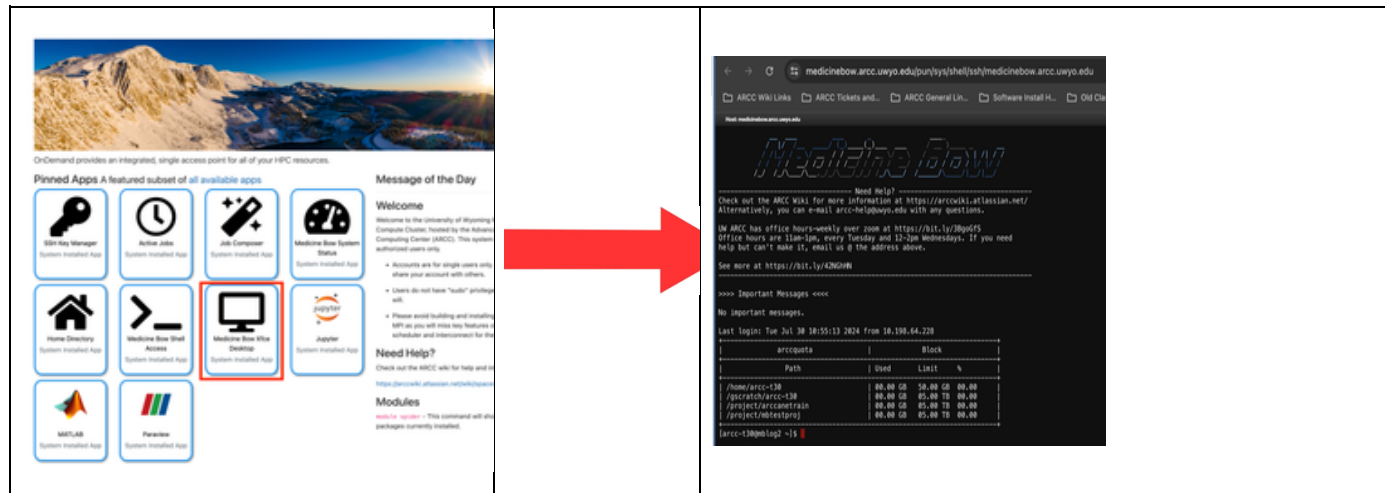password,   👁̸

☐ Remember me

Sign In

Then hit the green light on your yubikey to authenticate:
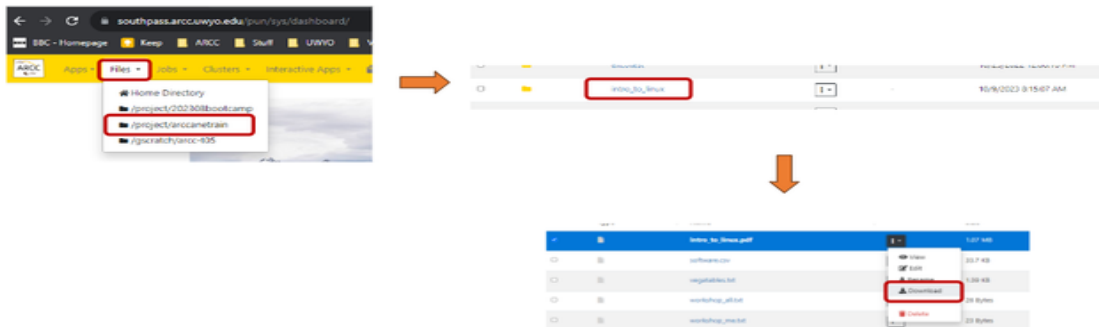
Start *MedicineBow Shell Access*

Download Slides

The Command-Line Prompt

[arcc-t05@blog1 ~]$

[username@hostname 'current working directory']<user type>

Syntax of a Shell Command

$ ls −al ~

[prompt command option(s) argument(s)

Case Sensitive

```
# Lists what is in the current location.
[arcc-t05@blog1 ~]$ ls
Desktop  Documents  Downloads
# Throws an error.
[arcc-t05@blog1 ~]$ LS
-bash: LS: command not found
Filename  ≠  FiLeNaMe  ≠  FILENAME
```

Filename ≠ FiLeNaMe ≠ FILENAME

Getting Help: man

```
[arcc-t05@blog1 ~]$ man ls
LS(1)                                   User Commands                                   LS(1)
NAME
       ls - list directory contents
SYNOPSIS
       ls [OPTION]... [FILE]...
DESCRIPTION
       List  information about the FILEs (the current directory by default).  Sort entries alpha-
       betically if none of -cftuvSUX nor --sort is specified.
       Mandatory arguments to long options are mandatory for short options too.
       -a, --all
              do not ignore entries starting with .
       -A, --almost-all
              do not list implied . and ..
       ...
Manual page ls(1) line 1 (press h for help or q to quit)
```

Getting Help: <command --help>

```
[arcc-t05@blog1 ~]$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
Mandatory arguments to long options are mandatory for short options too.
  -a, --all                  do not ignore entries starting with .
  -A, --almost-all           do not list implied . and ..
      --author               with -l, print the author of each file
  -b, --escape               print C-style escapes for nongraphic characters
      --block-size=SIZE      with -l, scale sizes by SIZE when printing them;
                               e.g., '--block-size=M'; see SIZE format below
  -B, --ignore-backups       do not list implied entries ending with ~
```

Getting Help: Options

Typically, options can have a:

- short-name: "-a":
  - Single letter following a single "-"
- long-name: "--all":
  - More *descriptive* word after two dashes "--"

Short options can be grouped:

- "ls -a -l" can be shortened to "ls -al"

Single vs Multiple Lines

```
[arcc-t05@blog1 ~]$ ls -al ~
[arcc-t05@blog1 ~]$ ls \
> -al \
> ~
```

Exercises

Questions:

1. Is there a difference between running `ls` versus `ls -al`?
2. How can you find out what the `-al` options do?
3. What does the `pwd` command do?
4. From the command line, what happens if you press the up/down arrow keys?

Answers(1)

1: Is there a difference between running `ls` versus `ls -al`?

```
[arcc-t05@blog1 ~]$ ls
Desktop   Documents   Downloads
[arcc-t05@blog1 ~]$ ls -al
total 76
drwxr-x---    8 arcc-t05 arcc-t05   4096 Oct  3 13:57 .
drwxr-xr-x 925 root     root      32768 Sep 27 16:21 ..
-rw-------    1 arcc-t05 arcc-t05    212 Sep 12 15:44 .bash_history
-rw-r--r--    1 arcc-t05 arcc-t05     18 Aug 10 17:00 .bash_logout
-rw-r--r--    1 arcc-t05 arcc-t05    141 Aug 10 17:00 .bash_profile
-rw-r--r--    1 arcc-t05 arcc-t05    376 Aug 10 17:00 .bashrc
drwx------    3 arcc-t05 arcc-t05   4096 Sep 12 11:36 .config
drwxr-xr-x    2 arcc-t05 arcc-t05   4096 Aug 10 17:00 Desktop
```

```
drwxr-xr-x    2 arcc-t05 arcc-t05   4096 Aug 10 17:00 Documents
```

2: How can you find out what the `-al` options do?

- Use `man ls` or `ls --help`
- `-a, --all do not ignore entries starting with .`
- `-l use a long listing format`
- Options are also case sensitive:
- `[arcc-t05@blog1 ~]$ ls -A`
- `.bash_history  .bash_profile  .config  Documents  .emacs     .kshrc     .mozilla  .zshrc`
  `.bash_logout    .bashrc          Desktop  Downloads  .esd_auth  .lesshst  .sshWhat does the pwd command do?`

3: What does the `pwd` command do?

- Use `man pwd` or `pwd --help`
- `pwd - print name of current/working directory`

4: From the command line, what happens if you press the up/down arrow keys?

- Steps through the previous commands you've typed.

The Linux File System

Goals: Introduction to the Linux File System, its structure and how to navigate around it, as well as creating, moving and copying files and folders.

- What the file system is, and a typical organization / hierarchy.
- Some high-level comparison to that of Windows.
- Absolute vs relative paths.
- Commands: `pwd`, `cd`, `ls`, `mv`, `cp`, `mkdir`, `rmdir`, `rm`
- History: `history`
- File Ownership and Permissions.

- [Some high-level comparison to that of Windows](#)
- [Linux OS General Structure](#)
- [Linux Hierarchical Structure: Example](#)
- [Absolute Path: /home/arcc-t05/](#)
- [Relative Path: workshop/projects/p01/etc/](#)
- [Ex: Starting at / (root), what is the absolute path to the bits folder?](#)
- [Ex: Starting in the home folder what is the relative path to the Jan folder?](#)
- [Commands: <command --help>](#)
- [pwd: Print the name of the current working directory](#)
- [cd: Change the shell working directory](#)
- [ls: List information about the FILEs (cwd by default)](#)
- [mkdir: Create the DIRECTORY(ies), if they do not already exist](#)
- [mv: Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY](#)
- [cp: Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY](#)
- [rmdir: Remove the DIRECTORY(ies), if they are empty](#)
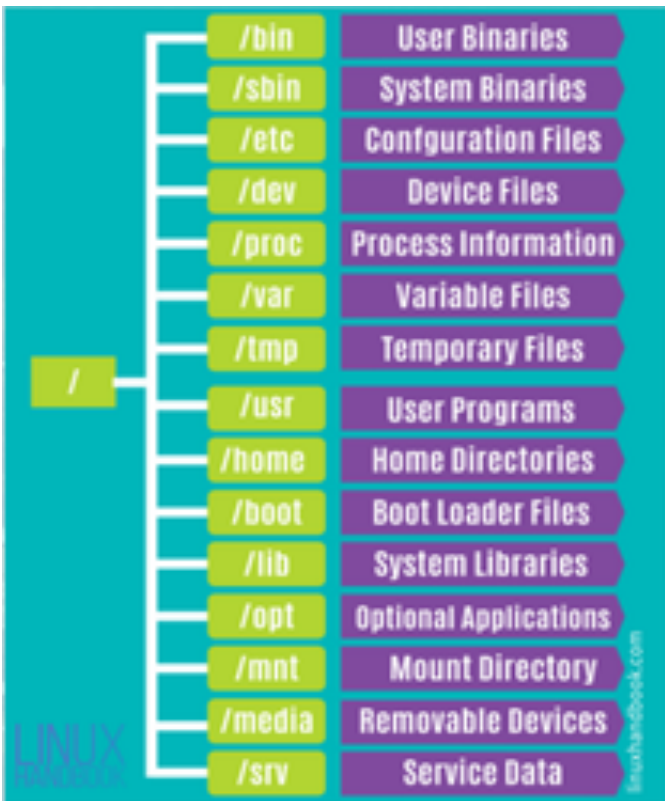- [rm: Remove (unlink) the FILE(s)](#)
- [rm: WARNING](#)
- [history](#)

Some high-level comparison to that of Windows

	Windows	Linux
Structure	Uses (data) drives C:, D:, E:…	Uses a tree hierarchy starting at "/" Known as the **root** directory.
	`<drive>\<folder01>\<folder02>`	`/<folder01>/<folder02>`
Syntax	Uses the back slash: "\"	Uses the forward slash: "/"
Home folder	`C:\Users\<username>`	`/home/<username>`
Application Install	`C:\Program Files\`	`/usr/`
Folder and Filenames	Case insensitive: FoLdEr = FOLDER	Case sensitive: FoLdEr ≠ FOLDER
		Wherever you are within the hierarchy is known as your **current working directory** (cwd)

Linux OS General Structure

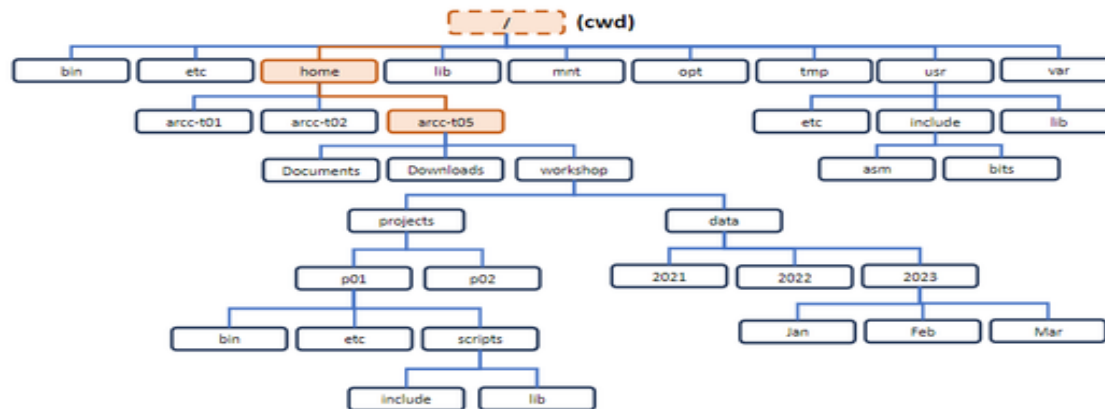| /		
/bin	User Binaries
/sbin	System Binaries
/etc	Confguration Files
/dev	Device Files
/proc	Process Information
/var	Variable Files
/tmp	Temporary Files
/usr	User Programs
/home	Home Directories
/boot	Boot Loader Files
/lib	System Libraries
/opt	Optional Applications
/mnt	Mount Directory
/media	Removable Devices
/srv	Service Data

LINUX HANDBOOK

linuxhandbook.com

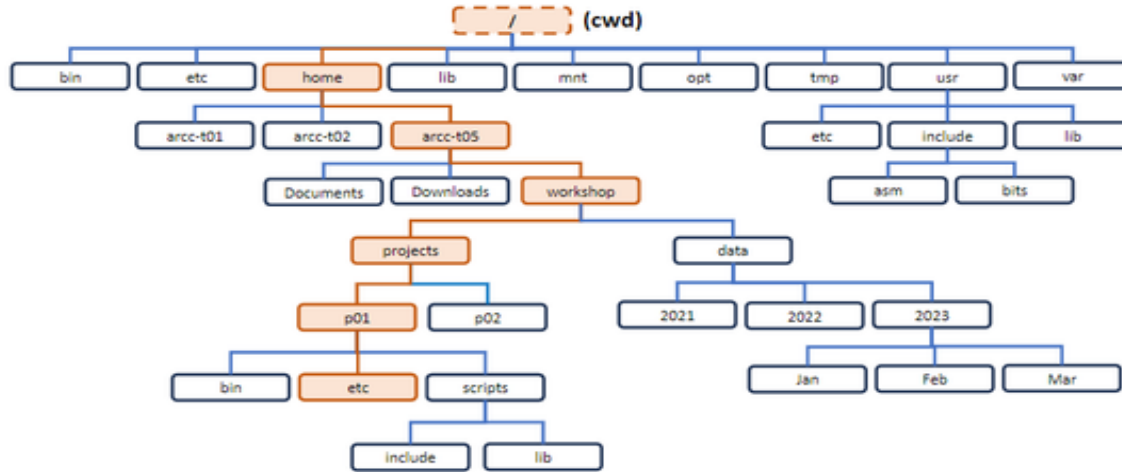# Linux Hierarchical Structure: Example

---

Absolute Path: /home/arcc-t05/
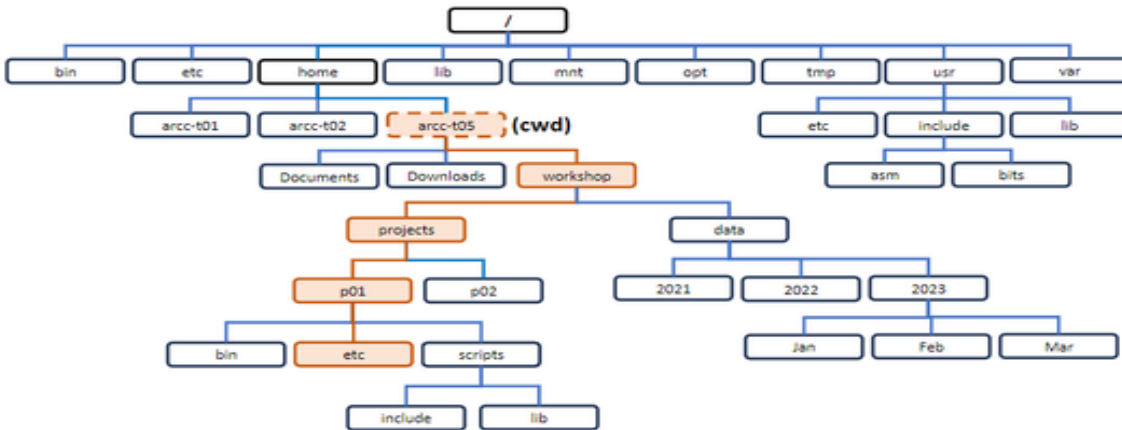
Path starts with a "/"

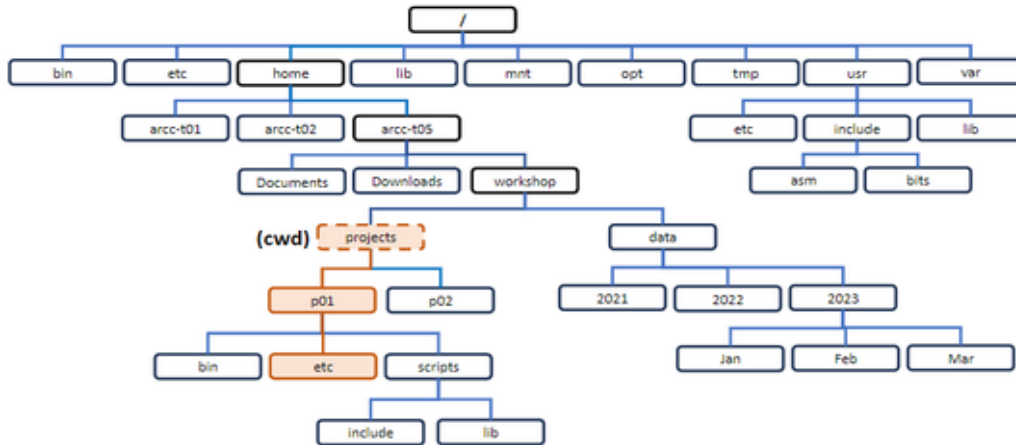Absolute Path: /home/arcc-t05/workshop/projects/p01/etc/

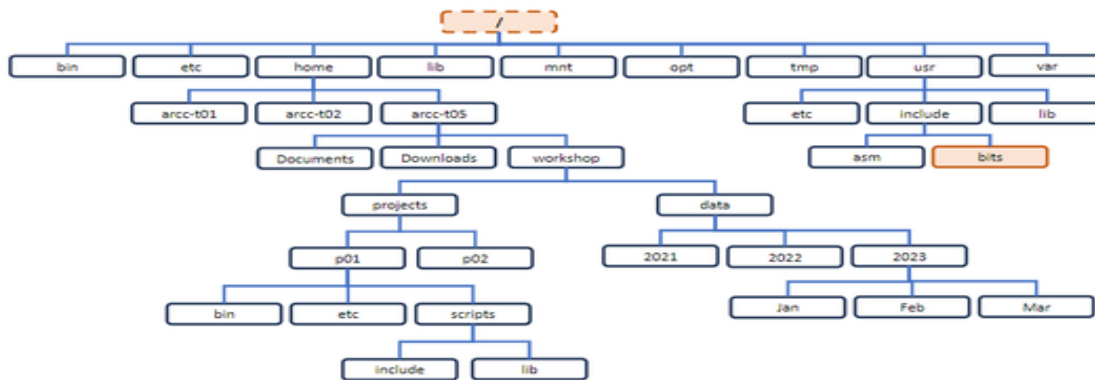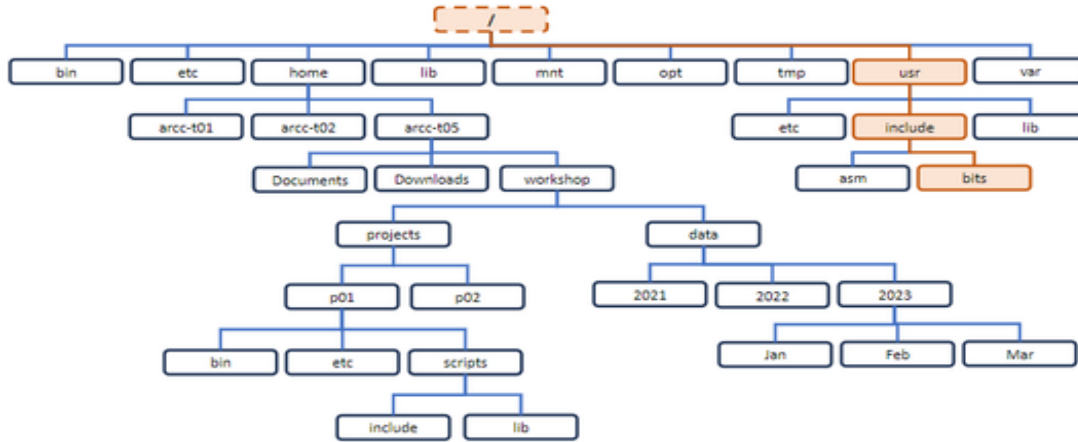# Relative Path: workshop/projects/p01/etc/

Path *does not* start with a "/"

Ex: Starting at / (root), what is the *absolute* path to the bits folder?

Ans: /usr/include/bits/

Ex: Starting in the home folder what is the *relative* path to the Jan folder?
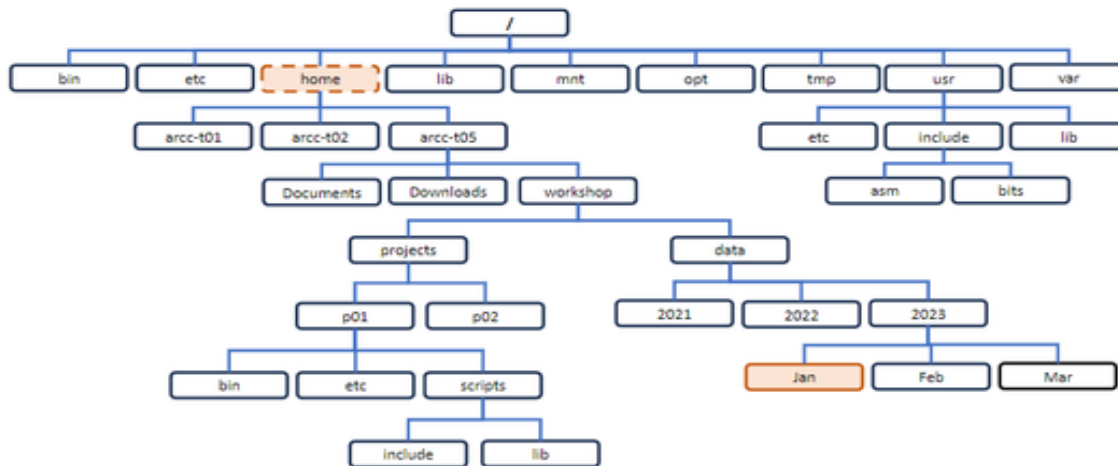
Ans: arcc-t05/workshop/data/2023/Jan/
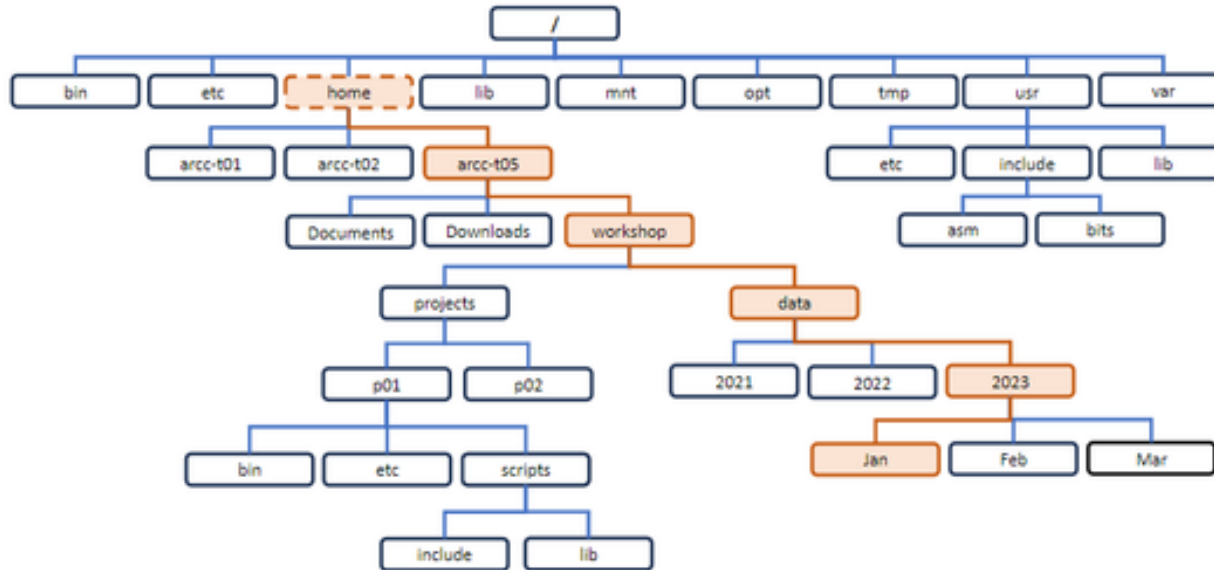
Commands: <command --help>

Command	Description
pwd	`pwd: pwd [-LP]` `Print the name of the current working directory.`
cd	`cd: cd [-L|[-P [-e]] [-@]] [dir]` `    Change the shell working directory.`
ls	`Usage: ls [OPTION]... [FILE]...` `List information about the FILEs (the current directory by default)`
mkdir	`Usage: mkdir [OPTION]... DIRECTORY...` `Create the DIRECTORY(ies), if they do not already exist.`

mv	Usage: mv [OPTION]... [-T] SOURCE DEST
	  or: mv [OPTION]... SOURCE... DIRECTORY
	  or: mv [OPTION]... -t DIRECTORY SOURCE...
	Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY

Commands: <command --help>

Command	Description

cp	Usage: cp [OPTION]... [-T] SOURCE DEST
	  or: cp [OPTION]... SOURCE... DIRECTORY
	  or: cp [OPTION]... -t DIRECTORY SOURCE...
	Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

rmdir	Usage: rmdir [OPTION]... DIRECTORY... Remove the DIRECTORY(ies), if they are empty.
rm	Usage: rm [OPTION]... [FILE]... Remove (unlink) the FILE(s).

pwd: Print the name of the current working directory

```
# Reset: Type the following:
[arcc-t05@blog1 ???]$ cd
[arcc-t05@blog1 ~]$
# The ~ "tilda" character represents your home directory.
[arcc-t05@blog1 ~]$ pwd
/home/arcc-t05
```

cd: Change the shell working directory

```
# Reset: cd
# Move up one level.
# Move into the folder's parent.
```

```
[arcc-t05@blog1 ~]$ cd ..
[arcc-t05@blog1 ~]$ pwd
/home
[arcc-t05@blog1 home]$ cd ..
[arcc-t05@blog1 /]$ pwd
/
# In the 'root' folder
[arcc-t05@blog1 /]$ ls
# Are we defining an absolute or relative path?
[arcc-t05@blog1 /]$ cd opt
[arcc-t05@blog1 opt]$ pwd
/opt
```

cd: Change the shell working directory

```
[arcc-t05@blog1 opt]$ cd
[arcc-t05@blog1 ~]$
# Are we defining an absolute or relative path?
[arcc-t05@blog1 ~]$ cd /usr/include/asm
[arcc-t05@blog1 asm]$ pwd
/usr/include/asm
[arcc-t05@blog1 asm]$ cd ../..
[arcc-t05@blog1 usr]$ pwd
/usr
[arcc-t05@blog1 usr]$ cd
[arcc-t05@blog1 ~]
```

ls: List information about the FILEs (cwd by default)

```
# Reset: cd
# List files in the user's home folder.
[arcc-t05@blog1 ~]$ ls
# List long format that includes ownership and permission details.
[arcc-t05@blog1 ~]$ ls -l
# List all files, including hidden files and folders start with ".".
[arcc-t05@blog1 ~]$ ls -a
# Notice how 'short-name' options are grouped.
# List all files with long format.
```

```
[arcc-t05@blog1 ~]$ ls -al
# List all files with long format, in reverse order.
[arcc-t05@blog1 ~]$ ls -alr
# List all files with long format, in reverse order, in human readable form.
[arcc-t05@blog1 ~]$ ls -alrh
```

mkdir: Create the DIRECTORY(ies), if they do not already exist

```
$ cd
[~]$ ls
Desktop  Documents  Downloads
[~]$ mkdir folder01
[~]$ ls
Desktop  Documents  Downloads  folder01
[~]$ mkdir folder01
mkdir: cannot create directory 'folder01': File exists
[~]$ cd folder01/
[folder01]$ pwd
/home/arcc-t05/folder01
```

mkdir: Create the DIRECTORY(ies), if they do not already exist

```
[folder01]$ mkdir folder02
[folder01]$ ls
folder02
[folder01]$ cd folder02/
[folder02]$ pwd
/home/arcc-t05/folder01/folder02
[folder02]$ cd ../..
[~]$ pwd
/home/arcc-t05
```

mv: Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY

```
$ cd
# Create an empty file.
[~]$ touch myfil.txt
[~]$ ls
Desktop  Documents  Downloads  folder01  myfil.txt
# Rename the file 'myfil.txt' to 'myfile.txt':
[~]$ mv myfil.txt myfile.txt
[~]$ ls
Desktop  Documents  Downloads  folder01  myfile.txt
```

mv: Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY
```
# Move the file 'myfile.txt' into the directory 'folder01'
[~]$ mv myfile.txt folder01/
[~]$ ls
Desktop  Documents  Downloads  folder01
# We can 'ls' what is in a relative folder.
[~]$ ls folder01/
folder02  myfile.txt
```

cp: Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY

```
$ cd
# Use the touch command to create an empty file.
[~]$ touch myfile02.txt
[~]$ ls
Desktop  Documents  Downloads  folder01  myfile02.txt
# Copy (duplicate) a file.
[~]$ cp myfile02.txt myfile02b.txt
```

```
[~]$ ls
Desktop  Documents  Downloads  folder01  myfile02b.txt  myfile02.txt
# Copy a file into an existing folder.
[~]$ cp myfile02b.txt folder01/
[~]$ ls folder01/
folder02  myfile02b.txt  myfile.txt
```

cp: folders

```
[~]$ cp folder01
cp: missing destination file operand after 'folder01'
Try 'cp --help' for more information.
[~]$ ls folder01
folder02  myfile02b.txt  myfile.txt
[~]$ cp folder01 folder03
cp: -r not specified; omitting directory 'folder01'
# "recursively" copy a folder and all its contents.
[~]$ cp -r folder01 folder03
[~]$ ls
Desktop  Documents  Downloads  folder01  folder03 myfile02b.txt  myfile02.txt
[~]$ ls folder03
folder02  myfile02b.txt  myfile.txt
```

rmdir: Remove the DIRECTORY(ies), if they are empty

```
[~]$ ls
Desktop  Documents  Downloads  folder01  folder03  myfile02b.txt  myfile02.txt
[~]$ mkdir folder04
[~]$ ls
Desktop  Documents  Downloads  folder01  folder03  folder04  myfile02b.txt
myfile02.txt
# Can remove folder04 since it is empty.
[~]$ rmdir folder04
[~]$ ls
Desktop  Documents  Downloads  folder01  folder03  myfile02b.txt  myfile02.txt
[~]$ rmdir folder03/
rmdir: failed to remove 'folder03/': Directory not empty
```

rm: Remove (unlink) the FILE(s)

```
[~]$ cd
[~]$ cd folder03
[folder03]$ ls
folder02  myfile02b.txt  myfile.txt
[folder03]$ ls folder02/
[folder03]$
# 'folder02' is empty.
[folder03]$ rmdir folder02/
[folder03]$ ls
myfile02b.txt  myfile.txt
[folder03]$ rm myfile.txt
[folder03]$ ls
myfile02b.txt
[folder03]$ rm myfile02b.txt
```

rm: Remove (unlink) the FILE(s)

```
[folder03]$ ls
[folder03]$
# 'folder03' is now empty.
[folder03]$ cd ..
[~]$ rmdir folder03/
[~]$ ls
Desktop  Documents  Downloads  folder01  myfile02b.txt  myfile02.txt
# This has taken a lot of individual steps.
# Can we do this quicker?
```

rm: folders and file(s)

```
[~]$ cd
[~]$ rm folder01/
rm: cannot remove 'folder01/': Is a directory
[~]$ rm --help
Usage: rm [OPTION]... [FILE]...
Remove (unlink) the FILE(s).
...
  -r, -R, --recursive   remove directories and their contents recursively
...
```

```
[~]$ rm -r folder01/
[~]$ ls
Desktop  Documents  Downloads  myfile02b.txt  myfile02.txt
# Can remove multiple files.
[~]$ rm myfile02b.txt myfile02.txt
[~]$ ls
Desktop  Documents  Downloads
```

---

rm: WARNING

From the command-line there is NO trash bin.

Using `rm`/`rmdir` is FINAL!

---

history

```
[~]$ history --help
history: history [-c] [-d offset] [n] or history -anrw [filename] or history -ps arg [arg...]
    Display or manipulate the history list.
    ...
[~]$ history
  ...
  219  rm -f folder01/
  220  rm -r folder01/
  221  ls
  222  rm myfile02b.txt myfile02.txt
  223  ls
  224  history
# Repeat command '223'
[arcc-t05@blog1 ~]$ !223
ls
Desktop  Documents  Downloads
```

Exercises

Questions:

1. How can you return to your home folder?
2. What command do you use if you've forgotten where you are in the folder hierarchy?
3. How can you list what is in a folder as well as any subfolders?
4. Go back through the command related slides are try for yourself.

Answers

1: How can you return to your home folder?

- Use: `cd` or `cd ~`

2: What command do you use if you've forgotten where you are in the folder hierarchy?

- Use: `pwd`

3: How can you list what is a folder as well as any subfolders?

```
[]$ man ls
-R, --recursive
            list subdirectories recursively
[]$ ls -R
```

File Ownership and Permissions

What does the output of `ls -l` mean?

```
[arcc-t05@blog1 ~]$ cd /project/arccanetrain/intro_to_linux
[arcc-t05@blog1 intro_to_linux]$ ls -al
total 54
drwxrwsr-x   4 salexan5 arccanetrain  4096 Oct  6 08:09 .
drwxrws--- 40 root     arccanetrain  4096 Oct  6 08:09 ..
drwxrwsr-x   2 salexan5 arccanetrain  4096 Oct  5 11:19 clusters
drwxrwsr-x   6 salexan5 arccanetrain  4096 Oct  5 14:56 data
-rw-rw-r--   1 salexan5 arccanetrain   874 Oct  5 15:30 fruits.txt
-rw-rw-r--   1 salexan5 arccanetrain 34472 Oct  5 10:57 software.csv
-rw-rw-r--   1 salexan5 arccanetrain  1603 Oct  6 08:08 vegatables.txt
-rw-rw-r--   1 arcc-t05 arccanetrain    26 Oct  5 07:20 workshop_all.txt
-rw-------   1 arcc-t05 arccanetrain    23 Oct  5 07:20 workshop_me.txt
```
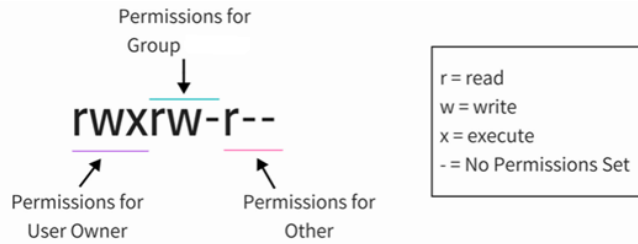
The first character on the left indicates if it is a directory "**d**" or a file "**-**".

```
drwxrwsr-x    clusters          # A folder.
-rw-rw-r--    workshop_all.txt  # A file.
```

File Ownership and Permissions

```
-rw-rw-r--    1 arcc-t05 arccanetrain    26 Oct  5 07:20 workshop_all.txt
```


- **User**: This is the owner of the file/folder. By default, the person who created it becomes its owner.
  - `arcc-t05` is the owner
- **Group**: A group is a collection of users. The primary purpose of the group is to define a set of privileges for a given resource that can be shared among the users within the group.
  - `arccanetrain` is the group.
  - All the `arcc-txx` users have been setup to be within this group.

- **Other**: This is any other user who has access to the file/folder. This person has neither created the file, nor do they belong to a user group.

Permission Denied

```
[]$ cd /project/arccanetrain/
# drwxr-sr-x    2 arcc-t01 arccanetrain   4096 May 16 16:26 arcc-t01
# No one other than arcc-t01 has permission to write within this folder.
[arccanetrain]$ cd arcc-t01/
# Can arc-t05 create (write) a file within this folder?
[arcc-t01]$ touch text.txt
touch: cannot touch 'text.txt': Permission denied
[arcc-t01]$ cd ../intro_to_linux
# -rw-rw-r--    1 arcc-t05 arccanetrain     26 Oct  5 07:20 workshop_all.txt
# Anyone within the group can read/write this file.
[intro_to_linux]$ cat workshop_all.txt
Everybody can read this.
# Only arcc-t05 can read/write this file.
# -rw-------    1 arcc-t05 arccanetrain     23 Oct  5 07:20 workshop_me.txt
[intro_to_linux]$ cat workshop_me.txt
cat: workshop_me.txt: Permission denied
```

Exercise: Try it

```
[]$ cd /project/arccanetrain/
[arccanetrain]$ cd arcc-t05
[arcc-t05]$ touch test.txt
# Do you get a "Permission denied"?
# Navigate into the intro_to_linux folder.
[arcc-t05]$ cd ../intro_to_linux
[intro_to_linux]$ cat workshop_all.txt
Everybody can read this.
[intro_to_linux]$ cat workshop_me.txt
```

```
# Do you get a "Permission denied"?
# Can you cd into the /opt folder?
# Justify your answer.
# Can you cd into the /root folder?
# Justify your answer.
```

Answer

```
# Can you cd into the /opt folder?
[arcc-t05@blog1 ~]$ cd /opt
[arcc-t05@blog1 opt]$
# Can you cd into the /root folder?
[arcc-t05@blog1 ~]$ cd /root
-bash: cd: /root: Permission denied
# Justify your answer.
[arcc-t05@blog1 ~]$ ls -l /
...
# "other" has read permissions
drwxr-xr-x.    5 root root    43 Jun 26 11:47 opt
...
# No permission set for other read permissions
dr-xr-x---.   17 root root  4096 Oct  4 12:58 root
```

Intro to Linux Command-Line: Summary

Goal: Point to resources and trainings to continue learning Linux. Summarize what's been covered.

- Suggestions, Additional Resources
- Further Trainings: UWYO LinkedIn
- Request an Account with ARCC

- [Summary](#)
- [Next Steps](#)

Suggestions, Additional Resources

- Next Steps on using Linux:
    - Practicing using Linux online.
        - Tutorialspoint: [Online Linux Terminal](#)
        - [Best Online Linux Terminals and Online Bash Editors](#)
- Dual boot a Windows machine with Linux.
- Run a container image.
    - [Run Linux containers on Windows](#)
    - [How to dual-boot Linux and Windows](#)
- UW Researcher? Create a project on one of ARCC's clusters with your PI.

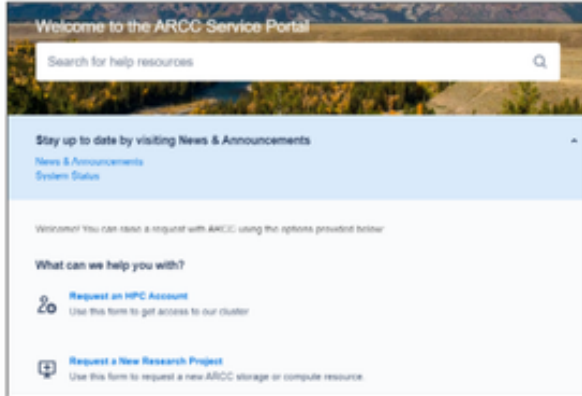# Further Trainings: UWYO LinkedIn
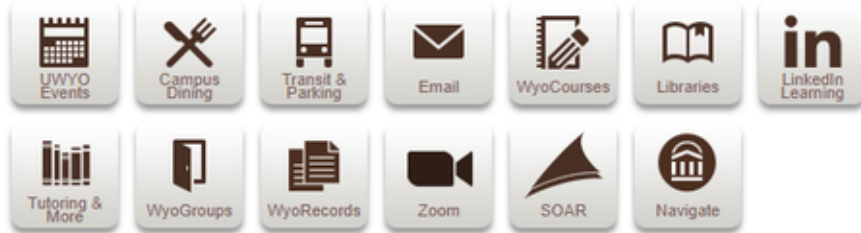
- Introduction to Linux
- Learning Linux Command Line
- Linux: Files and Permissions
- Linux: Over and Installation
- Learning Linux Shell Scripting

Request an Account with ARCC

- Web: [Advanced Research Computing Center | Overview](#)
- Wiki: [ARCC Wiki](#)
- Portal: https://arccwiki.atlassian.net/servicedesk/customer/portals

Summary

In this workshop we have:

- Introduced the basics of the Linux OS using a command-line interface.
- Taken a look at the hierarchical file system and how to navigate around it.
- Introduced the basics of file/folder permissions and ownership.
- How to view, create, update and delete files and folders.

Next Steps

Previous	Workshop Home
[The Linux File System](#)	[Intro to Linux CLI: The File System](#)

Use the following link to provide feedback on this training: https://forms.gle/RwtnWhK4EVHEN2rc9 or use the QR code below.