Welcome To:
Module 4

Linux Fundamentals

Commands Syntax

- Command options and arguments

Commands typically have the syntax:

command option(s) argument(s)

Options:

Modify the way that a command works

Usually consist of a hyphen or dash followed by a single letter

Some commands accept multiple options which can usually be grouped together after a single hyphen

Arguments:

Most commands are used together with one or more arguments

Some commands assume a default argument if none is supplied

Arguments are optional for some commands and required by others

File Permissions

- UNIX is a multi-user system. Every file and directory in your account can be protected from or made accessible to other users by changing its access permissions. Every user has responsibility for controlling access to their files.

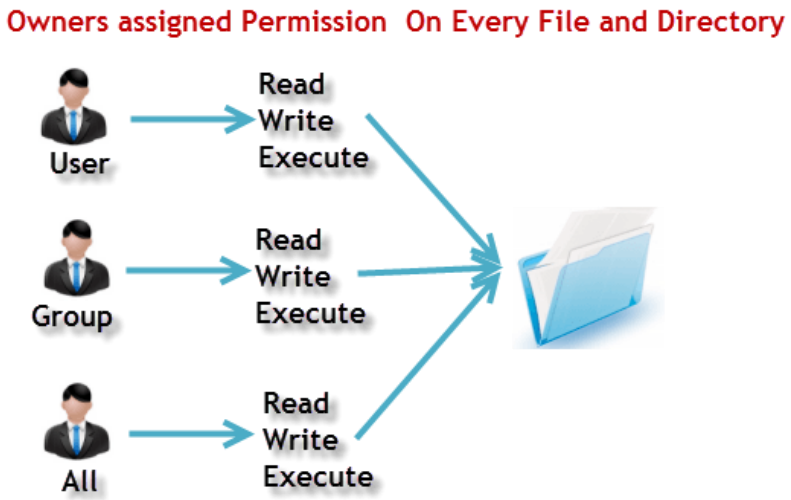
- Permissions for a file or directory may be restricted to by types
- There are 3 type of permissions
  - r - read
  - w - write
  - x - exeawke = running a program
- Each permission (rwx) can be controlled at three levels:
  - u - user = yourself
  - g - group = can be people in the same project
  - o - other = everyone on the system
- File or Directory permission can be displayed by running ls –l command
  - -rwxrwxrwx
- Command to change permission
  - chmod

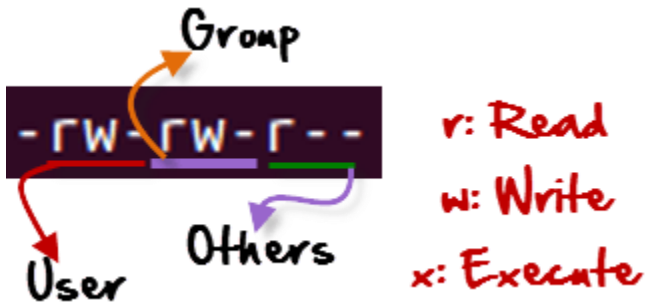
Permission Using Numeric Mode

- Permission to a file and directory can also be assigned numerically
  - **`chmod ugo+r FILE`**

  OR
  - **`chmod 444 FILE`**

`-r--r--r--`





Permission Using Numeric Mode

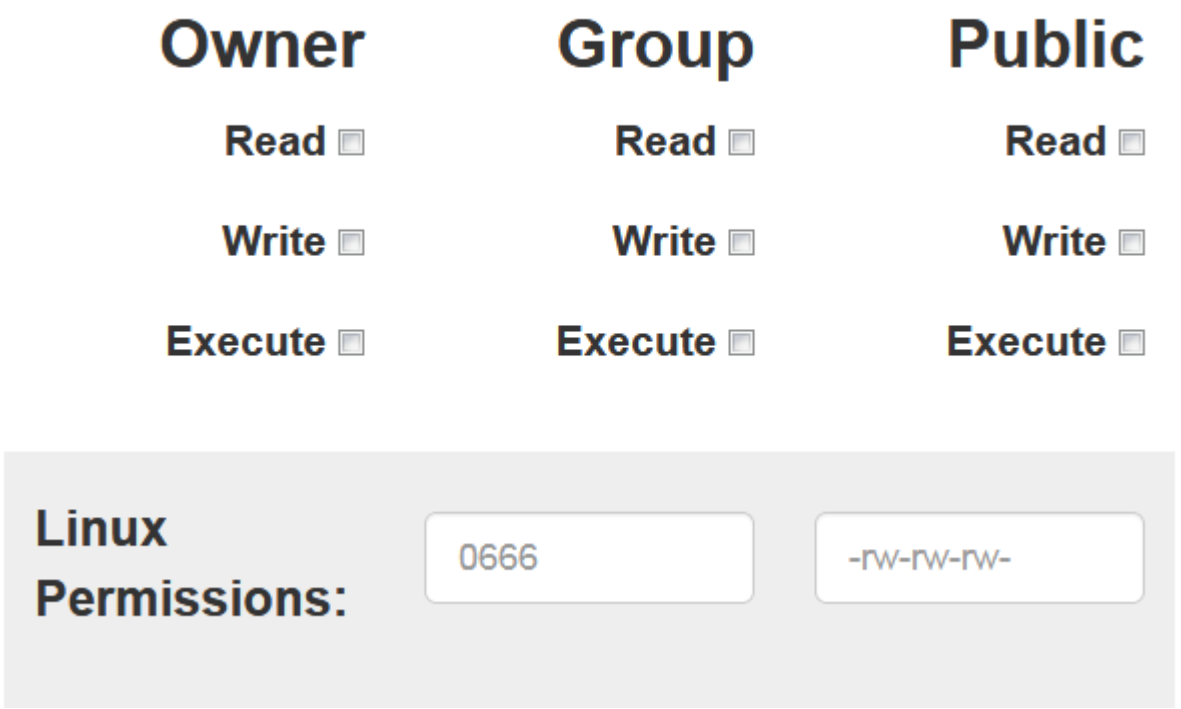
- The table below assigns numbers to permissions types

Number	Permission Type	Symbol
0	No Permission	---
1	Execute	--x
2	Write	-w-
3	Execute + Write	-wx
4	Read	r--
5	Read + Execute	r-x
6	Read +Write	rw-
7	Read + Write +Execute	rwx

- chmod 764 FILE

Permission Using Numeric Mode

- Online calculators can be used as well



File Ownership

- There are 2 owners of a file or directory
  - User and group
- Command to change file ownership
  - chown and chgrp
    - chown changes the ownership of a file
    - chgrp changes the group ownership of a file
- Recursive ownership change option (Cascade)
  - -R

Help Commands

- There are 3 types of help commands
  - **`whatis`** `command`
  - `command` **`--help`**
  - **`man`** `command`

TAB Completion and Up Arrow

- Hitting TAB key completes the available commands, files or directories
  - **`chm TAB`**
  - **`ls j<TAB>`**
  - **`cd Des<TAB>`**

- Hitting up arrow key on the keyboard returns the last command ran.

Adding Text to Files (Redirects)

- 3 Simple ways to add text to a file
  - **`vi`**
  - **`Redirect command output > or >>`**
  - **`echo > or >>`**

Input and Output Redirects

- There are 3 redirects in Linux
  1. Standard input (**`stdin`**) and it has file descriptor number as 0
  2. Standard output (**`stdout`**) and it has file descriptor number as 1
  3. Standard error (**`stderr`**) and it has file descriptor number as 2

- Output (**stdout**) - 1
  - By default when running a command its output goes to the terminal
  - The output of a command can be routed to a file using > symbol
    - E.g. **`ls -l > listings`**
      `pwd > findpath`
  - If using the same file for additional output or to append to the same file then use >>
    - E.g. **`ls -la >> listings`**
      `echo "Hello World"  >> findpath.`

Input and Output Redirects

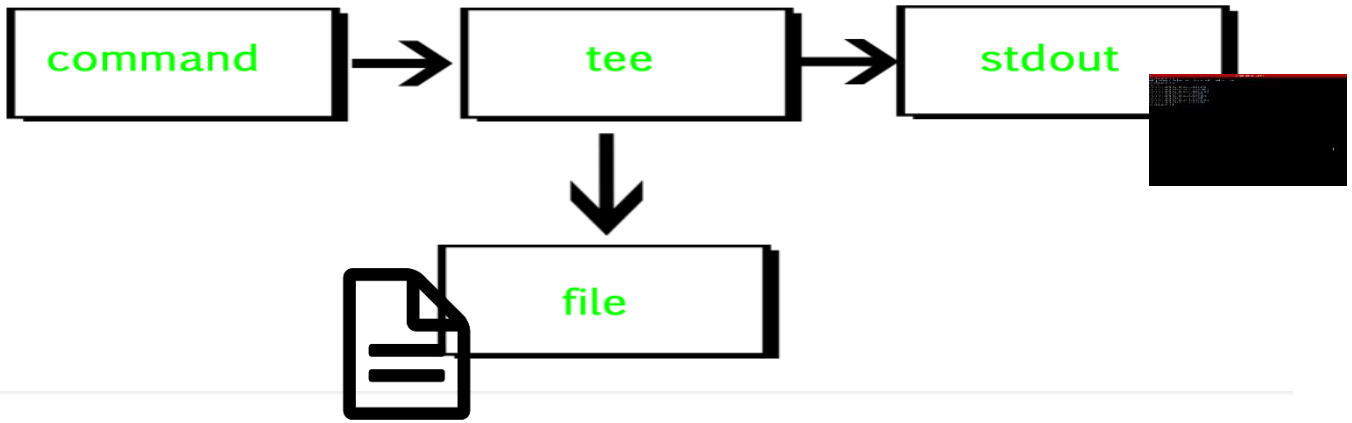
- Input (**stdin**) - 0
  - Input is used when feeding file contents to a file
    - E.g. **`cat < listings`**
      `mail –s "Office memo" allusers@abc.com < memoletter`

- Error (**stderr)** - 2
  - When a command is executed we use a keyboard and that is also considered (stdin -0)
  - That command output goes on the monitor and that output is (stdout – 1)
  - If the command produced any error on the screen then it is considered (stderr – 2)
    - We can use redirects to route errors from the screen
      - E.g **`ls –l /root 2> errorfile`**
        `telnet localhost 2> errorfile.`

Standard Output to a File (tee)

- “tee” command is used to store and view (both at the same time) the output of any command

- The command is named after the T-splitter used in plumbing. It basically breaks the output of a program so that it can be both displayed and saved in a file. It does both the tasks simultaneously, copies the result into the specified files or variables and also display the result.



Pipes

- A pipe is used by the shell to connect the output of one command directly to the input of another command.

The symbol for a pipe is the vertical bar (|). The command syntax is:

```
command1 [arguments]  |  command2 [arguments]
```

Pipes

```
ls -l          |          more
```

File Maintenance Commands

- `cp`
- `rm`
- `mv`
- `mkdir`
- `rmdir or rm -r`
- `chgrp`
- `chown`