Introduction to Linux
=====================

Linux has four major parts:

The kernel

The low-level operating system, handling files, disks, networking, and other necessities we take for granted. Most users rarely notice the kernel.

Supplied programs

Thousands of programs for file manipulation, text editing, mathematics, web browsing, audio, video, computer programming, typesetting, encryption, DVD burning...you name it.

The shell

A user interface for typing commands, executing them, and displaying the results. Linux has various shells: the Bourne shell, Korn shell, C shell, and others. bash, the Bourne-Again Shell,which is often the default for user accounts. However, all these shells have similar basic functions.

X

A graphical system that provides windows, menus, icons, mouse support, and other familiar GUI elements. More complex graphical environments are built on X; the most popular are KDE and GNOME. We'll discuss only a few programs that open X windows to run.

Basics of Linux File System
===========================

File holds data. In order to distinguish the contents, we name the files.

Directory is a special file which holds together other file(s).

Note: Directory equivalent to Folder in Windows

Concept of a sub-directory

If a directory named as P contains a file called as S and if S is also directory on its own then S is called subdirectory of P.
P in this case is called Parent directory.

Top most directory in Linux is denoted by / and is called root directory.

All the files descent from root directory in Linux environment.

In order to refer file(s), we have two mechanisms to address::
i) Absolute path
ii) Relative path

Absolute path starts with root directory. E.g., P s a directory and contains sub-directory called as S and S contains a file called as F then

Absolutely path for referring F would be

/P/S/F

Relative Path concept emerges based on your current position in the file system.

Current position in the file system is denoted by .(period)

.. refers to one level above w.r.t. current position.

Filesystem in Linux/Unix/UNIX is Hierarchical

Demonstrate FS on the Laptop

```
/ /bin /usr/bin /etc /var
```

mkdir

mkdir -p

cd

rmdir

rm

rm -rf

recursively deletes a given directory

Concept of home directory and shell variable HOME

Directory traversing and file access demo using absolute path and relative path.

cat

ls

more

```
Standard files and I/O Redirection, Pipes
=========================================
                   fd (File Descriptor)
                   ====================

  stdin       0   standard input
  stdout      1   standard output
  stderr      2   standard error

  I/O Redirection
  ---------------
   >    redirect output
   <    redirect input
   2>   redirect error
   >>   append
```

Concept of piping:

Output of a process is input to another process

```
cat /etc/passwd | more
```

```
=============================================================
```

In order to try out the unix commands you have many options.

Option 1:: Install Linux on your PC/Laptop either on top of Windows or side

by side.
This is difficult as it is likely to disturb/destroy windows and/or linux both if you are not careful

Option 2:: Install Vmware or VirtualBox in your windows OS and then install Linux as Virtual OS within Vmware or VirtualBox.

Option 3:: Easiest option
Refer to https://itsfoss.com/online-linux-terminals/
Try out JSLinux (Command and Graphical both)
Try Copy.sh

==

Users

Normal end users
Super user (root, root privileges)

Access Control (Permissions) on the FS

```
      r     rEAD
      w     wRITE
      x     ExECUTE
```

Concept of owner, group and others

```
 r 4
 w 2
 x 1
```

How to see ownership and permissions?

```
      ls -l
      ls -ld
```

Output Columns

```
1              File type:
            - = file, d = directory, l = symbolic link, p = named pipe,
                c = character device, b = block device

2–4         Read, write, and execute permissions for the file’s owner
5–7         permission (rwx) for  file’s group members
8–10        permission (rwx) for other users
```

chmod, chown and chgrp commands
----

chmod octal_value_for_all file

```
E.g., $ chmod 700 file1
      $ chmod 575 file2
      $ chmod 540 file3

file1 owner:rwx, group:nil others:nil
file2 owner:r-x, group:rwx others:r-x
file3 owner:r-x, group:r-- others:nil

chmod [ugoa...][[-+=][perms...]...] file...
```

Same as octal mode result using symbolic form

```
chmod u+xrw,g-rwx,o-rxw file1
chmod u+xr-w,g+rwx,o+rx-w file2
chmod u+xr-w,g+r-wx,o-rxw file3
======================================================================
```

Basic Commands

```
                  ONLINE Manuals at http://man.he.net
cd
cat
cp
mv
rm
rmdir
clear
pwd
man
chown
chmod
wc
cut
tr
diff
sort
wget
echo
passwd
su / sudo
history
vi
```

History of commands

```
      history          Displays the history of commands issued

      history -c  Deletes/Clears the history

      history N   Displays last N number of commands from your command history record

       !!               Execute Last command

      !$          Last parameter from the previous command
      !*          All parameters from the previous command

                  $ ls a b c
                  a b c
                  $ rm !$
                  $ ls c
                  c not found
                  $ ls x y z
                  x y z
                  $rm !*
                  $ ls x y z
                  x y z not found
```

Editor vi(m)

```
      Modes       Insert and command mode
```

```
                  i to insert
                  esc (Escape key) to get back to command mode

      Moving around     Arrow keys in command mode

      Beginning of file 1G
      End of file       G
      Any line number n       nG
      Beginning of Line 0
      End of Line       $
      w                 move forward by a word
      b                 back by a word
      n                 next
      j                 Down arrow
      k                 Up arrow
      l                 Right arrow
      h                 Left arrow
      /string                 Search for given string

        Command to ignore case  :set ignorecase OR :se ic
        To print line number    :set number OR :se nu

      Substitute        :s/old/new/

      c indicates intention to change, therefore, if you need to change a word,
command would be cw

      D           Delete upto end of line from where you are
      R           Replace until you wish from where you are
      x           Delete current character
      X           Delete previous character

      .           apply the last command once more

      u           undo last change

      dd          Delete current line

      ndd         Delete n number of lines

      yy          Copy current line in the unnamed buffer

      nyy         Copy n number of lines from current line into unnamed buffer
      p           Put the content of the buffer from the current position
onwards

      P           Put the content of the buffer just before the current position

      :w          Save changes to the current file

      :q          Quit the edit session, in case, changes are done and are not
to be saved use :q!


      Named registers   a b c so on upto z, 0-9, A-Z


      Show line numbers :set number
                        (In short, :se nu)

      Indent                  :set autoindent
                        :se ai

      All setting       :set all
```

```
                        Displays current settings

      Search at the
      beginning of line /^string

      Search at the
      end of line       /string$

      Blank line search /^$

      Deleting blank line     :map v /^$
dd
      First map v
      and now issue v command
```

==

System variables like – PS1, PS2 etc. How to set them

PS1 and PS2 system environment variables indicates prompt string 1 and prompt string 2.

to see the current values,

```
$ echo $PS1
$
$ echo $PS2
>
$

PS1="=> "
=>
=> PS2="# "
=> echo $PS2
#
=>
```

Network commands (telenet, ftp, ssh, sftp, finger)

Access control list::

```
   We have seen concept of rwx like permission for
   owner
   group members
   rest of the users
```

Consider a scenario, we have got 4 users

```
 user u1 group g1
 user u2 group g1

 user u3 group g2
 user u4 group g2
```

If a file f1 created by user has permissions like
 -rwxrwxrwx then it is clear that

u1 has rwx access as owner
u2 has rwx access as group member
u3, u4 has rwx access as others

What if u3 is to be provided with read access only and u4 no access at all?

The mode bits will not allow us to achieve the above. In such scenarios we resort to access control list.

We will need following commands to understand and use.

getfacl and setfacl

The setfacl has the following syntax

```
$ setfacl {-m, -x}  {u, g}:<name>:[r, w, x] <file, directory>
```

-m : modify one or several ACL entries on the file or directory.
-x : want to remove one or several ACL entries on a file or directory.
{u, g} : if you want to modify the ACL for a user or for a group.
name : this is an optional parameter, it can be omitted if you want to set the ACL entries for every user or for every group on your host.
[r, w, x] : in order to set read, write or execute permissions on the file or directory.

The getfacl can be used with the following syntax

```
$ getfacl file
```

Its output is divided into multiple categories, namely,

Filename, owner and group
Group permissions
Mask
Other permissions

Important Note::
Use -d option to create default acl entries for a directory
Use -k option to delete default acl entries for a directory
Use -R for recursive
Use -b to remove all acl entries
ACL mask different from mask set by umask for file creation

==

We have covered the following topics.

Introduction to Linux
- Working basics of file system
- Commands associated with files/directories & other basic commands.
- Operators like redirection, pipe
- What are file permissions and how to set them
- Permissions (chmod, chown, etc);
- System variables like – PS1, PS2 etc. How to set them
- network commands (telenet, ftp, ssh, sftp, finger)
- access control list;
- vi editor and history command

==