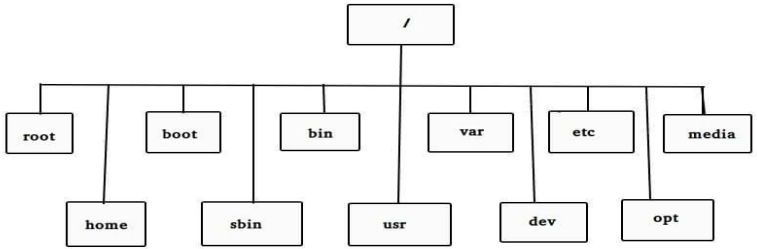
LINUX BASICS

File-system Hierarchy

- A file system is a logical collection of files on a partition or disk
- Linux uses a hierarchical file system structure, much like an upside-down tree
- Linux filesystem is a collection of files and directories.



- / - This is the root directory which should contain only the directories needed at the top level of the file structure.
- /root - home directory of root user.
- /home - Contains the home directory for users and other accounts.
- /bin - This is where the executable files are located
- /etc - Supervisor directory commands, configuration files, disk configuration files
- /tmp - Holds temporary files used between system boots

Users and Groups

- Three types of accounts on a Linux system:
  - **Root User:** Have complete and unfettered control of the system. Root user can run any commands without any restriction.
  - **System Users:** Those needed for the operation of system-specific components
  - **User accounts:** General users are typically assigned to these accounts and usually have limited access to critical system files and directories.
    - A normal user will only have access in their home directory.
- **Sudo**
  - Allows a permitted user to execute a command as the superuser or another user.
  - Add a normal user to sudoer file to get sudo permission.
    - **/etc/sudoers**
  - Append sudo before a command to run it as superuser.

❑ GROUP:

- ❑ Linux operating system is designed to allow more than one user to have access to the Linux system at a time.
- ❑ Group is logically groups a number of user accounts.
- ❑ When a user a created a group with same name as that of username is also created automatically. That user will only be the member of that group by default.

❑ Create a new user:

- ❑ **useradd** command adds a new user to the system.
- ❑ It creates a new user with userid(**UID**) and also a group with groupid (**GID**).
- ❑ A home directory for the user is created at **/home.**
- ❑ *$ useradd testuser*
- ❑ *$ id testuser*
  uid=1000(testuser) gid=1000(testuser)
- ❑ You can change the password of user using **passwd** command.
  $ *passwd username*
- ❑ **groupadd** command creates a new group in the system.
  $ *groupadd groupname*

Basic Commands

- **Linux Shell:** A shell is a program that receives commands from the user and gives it to the OS to process.
- Syntax
  - command [option] [arguments]
- Basic Commands:
  - **pwd** - command prints the absolute path to current working directory

    $ pwd

    /home/user
  - **echo** - This command will echo whatever you provide it.

    $ echo "linux"

    linux
  - **man** - To see a command's manual page

    $ man date

- **cd** - Change the current working directory to the directory provided as argument.If no argument is given to ‘cd’, it changes the directory to the user's home directory.

  $ pwd

  /home/raghu

  $ cd /usr/share/

  $ pwd

  /usr/share

- **ls** - List files and/or directories
  - **ls -l** - displays a long listing of the files.
  - **ls -a** - shows hidden files in directory.

    $ ls [files-or-directories]

    $ ls -l

    total 4

    drwxr-xr-x 2 user user 4096 2012-07-06 12:52 example

    -rw-r--r-- 1 user user 0 2012-07-06 12:52 file1.txt

- **mkdir** - To create a directory
  $ mkdir example
- **touch** - For creating an empty file
  $ touch file1
- **cp** - Copy files and directories
  $ cp usrlisting listing_copy.txt
- **mv** - Move files or directories
  $ mv source destination
- **rm** - remove files and directories
  $ rm files|directories
- **cat** - concatenator but can be used to view the contents of a file
  $ cat /etc/passwd
  root:x:0:0:root:/root:/bin/bash
  daemon:x:1:1:daemon:/usr/sbin:/bin/sh

- **head** - Displays the first few lines of a file.
  $ head /etc/passwd
  root:x:0:0:root:/root:/bin/bash
  daemon:x:1:1:daemon:/usr/sbin:/bin/sh
  bin:x:2:2:bin:/bin:/bin/sh
- **tail** - Shows the last 10 lines by default.
  $ tail -n 2 /etc/passwd
  raghu:x:1000:1000:Raghu Sharma,,,:/home/raghu:/bin/bash
  sshd:x:113:65534::/var/run/sshd:/usr/sbin/nologin
- **grep** - searches for a pattern in a file
  $ grep nologin /etc/passwd
  sshd:x:113:65534::/var/run/sshd:/usr/sbin/nologin
- **ln** - create links, links are a kind of shortcuts to other files.There are two types of links, soft links and hard links
  $ ln TARGET LINK_NAME

EDITORS

- **VI** : Standard editor in many Linux.
  - Default editor that comes with many Linux distributions.
    $ vi hello.txt
  - The vi editor has 3 modes in which it performs its functions.The default is **COMMAND mode**: in which tasks like copy, paste, undo etc can be performed.
    INSERT mode: in which whatever key you type is treated as a character and will be loaded into the file buffer. To enter this mode, press ‘i’ when in command mode.
  - **EX mode** or last line mode. The changes made in the buffer can be saved or discarded in this mode.To get into it, press Esc and then : (the colon)

- **Nano:** Nano is a modeless editor so you can start typing immediately to insert text
  - *$ nano filename*
  - to save the changes you've made, press **Ctrl** + **O**.
  - To exit nano, type **Ctrl**+ **X**. If you ask nano to exit from a modified file, it will ask you if you want to save it. Just press N in case you don't, or Y in case you do.
  - It will then ask you for a filename. Just type it in and press **Enter**
- **Gedit**: The default GUI text editor
  - *$ gedit filename*

FILE PERMISSIONS

- File ownership is an important component of Unix that provides a secure method for storing files.
- Every file in Unix has the following attributes:
  - **Owner permissions** − The owner's permissions determine what actions the owner of the file can perform on the file.
  - **Group permissions** − The group's permissions determine what actions a user, who is a member of the group that a file belongs to, can perform on the file.
  - **Other permissions** − The permissions for others indicate what action all other users can perform on the file.
- *$ ls -l /home/user/test.txt*
  - -rwxr-xr-- 1 ubuntu ubuntu 1024 Nov 2 00:10 *test.txt*
- The permissions are broken into groups of threes: read (r), write (w), execute (x)

❑ Changing Permissions:

- ❑ To change the file or the directory permissions, you use the **chmod** (change mode) command.
- ❑ *$ ls -l testfile*

  -rwxrwxr-- 1 user users 1024 Nov 2 00:10 testfile
- ❑ *chmod o+wx testfile*
- ❑ *$ ls -l testfile*

  -rwxrwxrwx 1 user users 1024 Nov 2 00:10 testfile

❑ Changing Owners and Groups:

- ❑ The **chown** command changes the ownership of a file

  $ *chown user file.txt*
- ❑ The value of the user can be either the name of a user on the system or the user id (uid) of a user on the system.
- ❑ The **chgrp** command changes the group ownership of a file.

  $ *chgrp groupname file.txt*
- ❑ The **usermod** command adds an Existing User to a Group.

  $ *usermod -a -G username groupname*

virtualenv

IDE

Assignment

https://docs.google.com/document/d/1OmZOud_DqXSNi-z5xau78Kpa7oVCkkytw-l48_r_F0U/edit