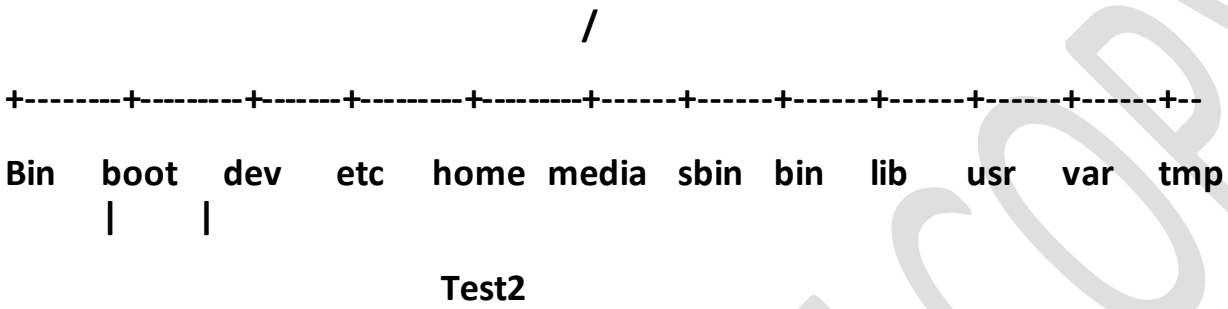
Commands are in black



Above is a view of a typical Linux file structure from the root (/)

Getting help

The man command allows you to view manual pages in Linux just like F1 in Windows. Please make note of the following when issuing the man command or any other command that has one page per screen displayed at a time. Enter or up and down arrow allows you to scroll, space bar allow you to scroll one screen (page) at a time and q allows you to quit.

```
Manual page whois(1) line 1 (press h for help or q to quit)
```

1. In the terminal window enter the following command

`man ls` and then press Enter

2. What is displayed?

3. In the terminal window enter the following command

```
man whois
```

4. What does this command do? **Hint look at the description**

5. In the terminal window enter the following commands and then look at the top of the page? What do you see is the difference?

```
man 1 chmod
```

and then

```
man 2 chmod
```

6. The following other command can also be used to get help (`--help`) for example

```
chmod --help
```

. For more practice issue these other commands

```
useradd --help
groupadd --help
passwd --help
touch --help
```

Using the su (switch user) command

Before you begin this lab create a user called jack. Remember Linux is case sensitive

Exercise 1-1

1. Create a user called jack

```
root@localhost:~# adduser jack
```

2. Switch to the use jack

```
root@localhost:~# su jack
```

3. What prompt is displayed?
4. Switch back to the root user, you must enter the root password

```
$ su
```

5. What prompt is displayed?
6. Type the command below

```
root@localhost:~# clear
```

7. What happened?

Switch back to user jack

Displaying the Date and Time

Using the table below issue the command that is in bold. For example

```
$ date
```

Take note of what happens…

Issue the following commands	*Explanation*
date 101909302010	Changing the date and time

date	Displays current date, time and time zone
date '+%A %B %d %G'	Display day, month , day of the month, year
date '+Today is %F'	Add word to the date output
date - -date=' 6 weeks'	Display date 6 weeks from today
date - - date='5 months'	Display date 5 months from today
date - - date='5 months 5 days'	Display date 5 months and 5 days from today
date - - date='25 Dec' +%A	Display the date Christmas falls on

User Information

Using the table below issue the command that is in bold. Note the $ represents the prompt, you do not have to type it.

Commands	*Description*
$last	List the most recent successful logins
$last –a	Makes it easier to read
$lastb	List the most recent unsuccessful logins
$who	List who is currently logged in
$who –u	Same as who command but in long format
$users	Same as who but in short format
$id	Gives your identity
$whoami	Same as who
$w	Give more details about who is logged in
$finger	Display a list of users logged into the system
$finger –s jack	Give user information short
$finger –l root	Give user information in long format

Absolute and Relative Path

Absolute path begins from the root (**/**) or starts from the root of the file system. If you will be writing the absolute path the following must be adhered to:

- Start at the root directory (/) and work down.
- Write a slash (/) after every directory name (last one is optional)

For example

What is the absolute path to the etc/apt folder

`/etc/apt` Note when issuing a command with the absolute path location is not relevant.

Relative Path

Relative path start with your present working directory (pwd) or you current directory(present location). For example the relative path for `/etc/apt` is the **apt** folder.

Exercise 1-2

1. Clear your screen, what command did you use?
2. Type the command `cd /`
3. Type `pwd` what is your absolute path?
4. Install the tree command in Kali. Type the command below, the clear the screen.

```
apt-get install tree
```

We will be using this command later

User and Group Management

You will have ability to create users and groups

What is used to create users

Remember we are working with Kali but all distributions use the useradd command

```
useradd
```

You can also use the

```
adduser
```

Example:

```
root@localhost:~# adduser john
Adding user `john' ...
Adding new group `john' (1002) ...
Adding new user `john' (1002) with group `john' ...
Creating home directory `/home/john' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
```

Task 1:1 Listing users and groups:

Fill in at least 3 username and other information

User Login Name	User Full Name	Room Location of User	Group User is a Member of
robertp	Robert Phillips	Tech 211A	faculty
			student
			student
			security

Task 2: Creating the users

Using the adduser command create the user in Task 1:1

```
root@localhost:~# adduser robertp
Adding user `robertp' ...
Adding new group `robertp' (1005) ...
Adding new user `robertp' (1005) with group `robertp' ...
Creating home directory `/home/robertp' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for robertp
Enter the new value, or press ENTER for the default
        Full Name []: Robert Phillips
        Room Number []: Tech 211A
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
```

Task 3: Creating Groups

Two commands can be used to create groups: `addgroup` and `groupadd`

```
root@localhost:~# addgroup faculty
Adding group `faculty' (GID 1006) ...
Done.
```

```
root@localhost:~# groupadd student
```

Finish creating the security group

Task 4: Adding user to Groups

Adding the users to groups

```
root@localhost:~# usermod -G faculty robertp
```

Add the other users to their respective groups

Task 5: Verifying and looking at User and Groups configuration files

Default user settings	/etc/login.defs
Stores user information	/etc/passwd
Stores group information	/etc/group
Stores password information in encrypted form	/etc/shadow

Fields in the /etc/passwd file

```
robertp:x:1005:1005:Robert Phillips,Tech 211A,,:/home/robertp:/bin/bash
```

User name	robertp
Encrypted password	x
User ID number (UID)	1005
User's group ID number (GID)	1005
Full name of the user	Robert Phillips
User home directory	/home/robertp
Login shell	/bin/bash

Fields in /etc/shadow:

```
robertp:$6$ZDDr3OTE$i3l4D2O/1pHzgyODmJrHIIk5JwRV2VZorSKN0zqmHEiRHmZloTEWz5flDs1Bavi1uBMcoKCk4Qn3C9zKe7ocP1:17787
:0:99999:7:::
```

1. username:
2. encrypted_password:

Review the file for other options—check the internet

Now logout and login as robertp

Task 6: Understanding Present Working Directory

1. Go to the shell prompt. What is your shell prompt

2. Type pwd. What is your present working directory

3. Enter the command: `cd ~`
   What is your current directory now?

4. Your current directory is /home/robertp
   You enter the command: `cd`
   What is your current directory now?

5. Your current directory is /home/robertp
   You enter the command: `cd ..`
   What is your current directory now?

6. Your current directory is /home/Robertp
   You enter the command: `cd /`
   What is your current directory now?

Task 7: Listing Directory Contents and Navigating the file system

Understanding files and folders

The **ls** command is used to show the contents of a directory. When the content of the directory is shown it is important to note:

- **Black represents a file**
- **Red represents backup or zip files**
- **Blue represents a folder**
- **Green represents executable**
- File and directory that begin with a period (.) are hidden.

ls	List the contents of the current directory
ls /home	List the contents of the /etc directory
ls -a	Displays all entries in the current directory including those that are hidden.
ls -l	Display a long or detailed listing of the current directory, one line of data for each entry.
ls -al /home	Display a long listing of the /etc directory including "hidden" files or subdirectories.
ls -t	Files are sorted by date modified.
ls -tr	Files are sorted by date modified in reverse order
ls -F	Shows a character after every entry. Executables (*), directories (/), FIFO (I), symbolic link (@).
ls -R	Output is recursive, including all subdirectories
ls -u	Sorted by date of last access.

Remember these commands

pwd - print working directory

cd - change directory

You must fill in the commands that allow you to complete the task

A sample directory entry:

```
drwxr-xr-x 15 robertp robertp 4096 Sep 13 01:58 .
```

- `d` This is an directory
- `rwx` The permissions for the owner (robertp): read, write, execute
- `r-x` The permissions for the group (robertp): read, no write, execute
- `r-x` The permissions for others: read, not write, execute
- `15` The number of links to the file
- `robertp` The owner
- `robertp` The group
- `4096` The size (in bytes)
- `Sep 13 01:58` The last time that the file was modified

1. Change to the root of the file structure?
2. List at least 2 folders in this directory
3. Change to robertp home directory
4. List at least two folders in this directory
5. Change to the /etc from your current location(remember absolute path)
6. Change to the /bin directory from your current location
7. View the content of the /home directory from your current location
8. View the content of the /sbin directory from your current location
9. Do you have hidden files in the /home/robertp folder
10. Do you have any executable files in the /etc folder
11. Do you have any links in the /sbin folder

Task 8: Creating and working with Directories

Switch to the root user

mkdir - This command creates a new directory. Examples:

```
mkdir test

mkdir /etc/test123
```

rmdir -This command removes an existing directory. The directory must be empty before it can be removed. Example:

`rmdir test` This removes a directory but it must be empty

`rm -rf test` This removes a directory and everything within it

Create the following folder Structure

1. Go to the home folder of robertp and create a folder call parent

```
root@localhost:~# su robertp
robertp@localhost:/root$ cd
robertp@localhost:~$ mkdir parent
robertp@localhost:~$ cd parent
```

2. Inside the parent folder create the following structure below.

```
robertp@localhost:~/parent$ tree
.
├── childA
│   └── foldera
├── childB
│   └── folderb
├── childC
│   └── folderc
└── ChildD

7 directories, 0 files
```