

Where is Linux Used?

- ♣ 75% of respondents were already using Linux and another 14% were evaluating it
- ♣ 43% of all web sites use Linux servers running the Apache Web server

Basics

Shell

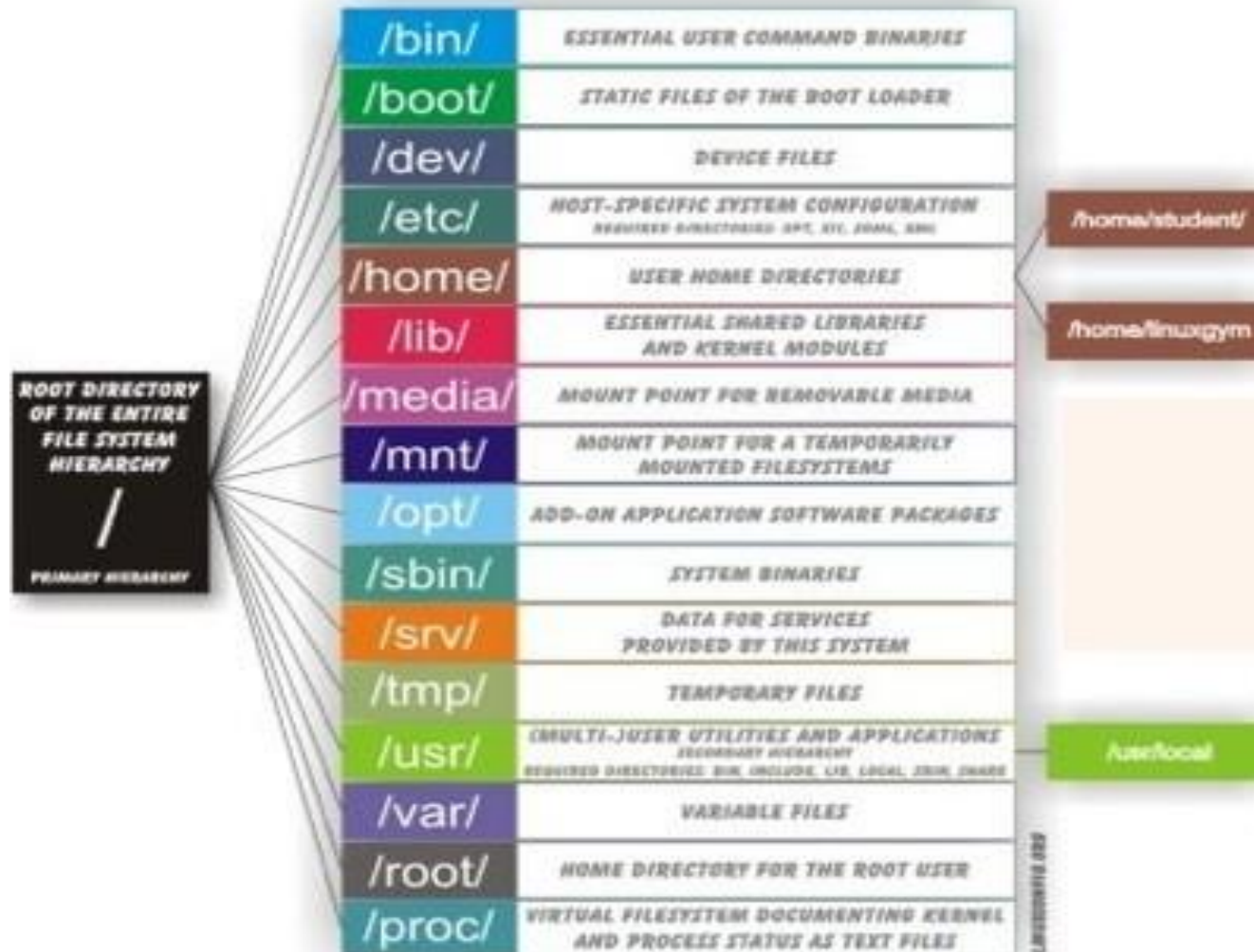
- Shell is nothing but a Command prompt like in windows, When you enter the commands, you get your work done
- The Shell is a command interpreter, it takes each command and passes it to the OS kernel to be acted upon. It will displays the results of your operations on screen.
- We have 'n' no.of shells available on any Unix Machine. each one has it's own strengths and weaknesses.

Continued...

— The most commonly available Shell(s) are:

1. Bourne Shell (sh)
2. Korn Shell (ksh)
3. C Shell (csh)
4. Bourne Again Shell (bash)
5. TC Shell (tcsh)

Unix Directory Structure/File System Architecture



Logging into the System

- To login to your account type your user name and at the login prompt. Unix is Case Sensitive.
- When the password prompt appears, type your password. Your password will never displayed on screen for security Measure. It is also case sensitive
- Then You will get [username@hostname]# or [username@hostname <pwd>]\$.
 - #, means it is administrative User Login
 - \$, means it is Normal user Login
- And for Administrator always Username is 'root', and the User ID and Group Id is 0.

Basic Commands

Commands and their options

- | Each and every command will have generally some options to get more information about the respective outputs.
- | Options will be used by preceding “-”. Options will be given after entering the command by giving some space

Working with Basic Commands

Uname: Prints system information

- a print all information
- s print the kernel name
- r print the kernel release
- v print the kernel version
- m print the machine hardware name
- p print the processor type
- o print the operating system

```
root@ubuntu:~# uname -a
Linux ubuntu 2.6.38-8-generic #42-Ubuntu SMP Mon Apr 11 03:31:50 UTC 2011 i686 i686 i386 GNU/Linux
```

Working with Basic Commands

ls – Listing of files and directories

```
[root@localhost ~]# ls
bin    dev    home  lost+found  misc  net  proc  sbin    srv  tmp  var
boot  etc    lib   media      mnt   opt  root  selinux  sys  usr
```

Options:-

-l – long listing of files

```
[root@localhost ~]# ls -l
total 60
-rw----- 1 root root  803 Jan 29 06:22 anaconda-ks.cfg
drwxr-xr-x 2 root root 4096 Jan 29 13:01 Desktop
-rw-r--r-- 1 root root 34457 Jan 29 06:22 install.log
-rw-r--r-- 1 root root  3747 Jan 29 06:22 install.log.syslog
```


1 2 3 4 5 6 7 8
-rw-r--r-- 1 root root 34457 Jan 29 06:22 install.log

1 – file type
- Normal file
d Directory

l Link file

c character file

b block file

2– File/Dir Permissions

3 – Owner

4 – Group membership

5 – File/Dir Size

6- File/Dir creation

/modification Date

7 – Time modified/created

8 – File name

Working with Basic Commands

-a – Listing of normal files/dirs and as well as hidden files/dirs

```
[root@localhost ~]# ls -a
.          .bashrc   .gconfd   .ICEauthority  .tcshrc
..         .cshrc    .gnome     install.log     .Trash
anaconda-ks.cfg Desktop    .gnome2    install.log.syslog
.bash_history .dmrc     .gnome2_private .metacity
.bash_logout  .egg cups .gstreamer-0.10 .nautilus
.bash_profile .gconf    .gtkrc-1.2-gnome2 .redhat
```

-r – Print the output in reverse order

-t – Print the output in based on time

-h – Print the file/dir sizes in human-readable format

-l – Print the file/dir output line by line

Working with Basic Commands

- **man – manual pages for commands or help for the commands**
- Example :- man ls
- Output will be no. of pages.... To view the information line wise or page wise follow these...
- To see the output line by line press “Enter “
- To see the output page by page press “Spacebar”
- To quit from the manual page press the “q” key
- The above three options are useful in all other Unix commands, whenever this kind of scenario occurs.
- **clear – to clear the screen.**

Working with Basic Commands

uptime - Tells how long the system has been running

date - To know present date and time

cal - To know the present month calendar.

if you want specific month calendar in specific year

cal 3 1999 - will display the March 1999 calendar.

pwd - To get the present/current working directory.

mkdir - To create the directories

Ex:- mkdir dir1

To create multiple parent/child directories at a time

mkdir -p dir1/sdir1

Working with Basic Commands

cd – To change the working directory

```
cd <dir_name>      # Switches into specified  
cd directory        # Moves one directory up  
cd ../../          # Moves two directories up (and so on)  
cd /                # Move the / directory.  
cd -                # Go back to you were previously (before the last directory  
change)
```

touch – To create the empty file, if file is not there, otherwise it updates file timestamp to current timestamp.

```
touch f1
```

Working with Basic Commands

cat – To create/view the files.

cat > file1 - will create a file, it prompts for entering the data, after entering the data, save the file by pressing Ctrl-D.

cat >> file1 – Will append the data to the file, it also prompts for entering the data, save the file by pressing Ctrl-D

Behavior of cat command

cat	File not exist	File exists
>	Creates it	Overwrites it
>>	Creates it	Append it

You can't edit the files using the cat command...

To view the multiple files

cat file1 file2

Working with Basic Commands

cp – to copy and paste the files/directories

- Syntax to copy the files

`cp [options] <source> <destination>`

To copy file1 as file2 in the same location.

`cp file1 file2`

To copy file1 from your present working directory to some other directory

`cp file1 /opt/dir1/`

while copying you want to change the file name

`cp file1 /opt/dir1/file2`

Working with Basic Commands

Contd..

To copy file1 from some other directory to your present working directory

```
cp /opt/dir1/file1 .      ( . Means present working directory)
```

while copying you want to change the file name

```
cp /opt/dir1/file1 file2
```

To copy file1 from some directory to some where else in the system

```
cp /opt/dir1/file1 /home/user1/
```

while copying you want to change the file name

```
cp /opt/dir1/file1 /home/user1/file2
```


Working with Basic Commands

- Syntax to copy the directories
`cp -r [options] <srcdir> <destdir>`

Options:-

- f - to copy files/dir forcefully.. This option will be useful when we want to copy files/dirs, if already exists.
 - i – Interactively copying the files by asking question want to copy or not?
 - p – while copying files/dirs to preserve the file attributes like timestamp, permissions, ownership, groups.
- `cp -rf /opt/dir1 /home/user1/`

Working with Basic Commands

rm – removing of files/dirs

- Syntax to remove the files

rm [options] <filename>

- To remove a file

rm file1

- To remove all files starts with “f”

rm f*

- To remove all files ends “f”

rm *f

- Syntax to remove the directories

rm -r [options] <dirname>

Working with Basic Commands

mv – to move files/dirs from one location to another location...And also we can rename the files/dirs

-To rename a file/dir

```
mv file1 file2
```

```
mv dir1 dir2
```

-To move a file/dir

```
mv file1 /opt/dir1
```

```
mv dir1 /home/user1
```

-To move a file/dir and also rename

```
mv file1 /opt/dir1/file2
```

```
mv dir1 /home/user1/dir2
```

Working with Basic Commands

wc – To get the count of lines/words/characters

wc file1

To get the lines/words/characters output individually.

-l – no.of lines

wc -l file1

-w – no.of words

wc -w file1

-c – no.of characters

wc -c file1

Working with Basic Commands

- To view the file content page by page
 - more
 - less

`more <my_file>` # views text, use space bar to browse, hit 'q' to exit

`less <my_file>` # a more versatile text viewer than 'more', 'q' exits,

'G' moves to end of text, 'g' to beginning, '/' find forward, '?' find

backwards

Working with Basic Commands

- To view the file top lines based on numerical value

-head

head -n <filename>

by default you can see first 10 lines of the file

`head file1`

to view first 20 lines from file1

`head -20 file1`

- To view the file end lines based on numerical value

-tail

tail -n <filename>

by default you can see end 10 lines of the file

`tail file1`

to view last 20 lines from file1

`tail -20 file1`