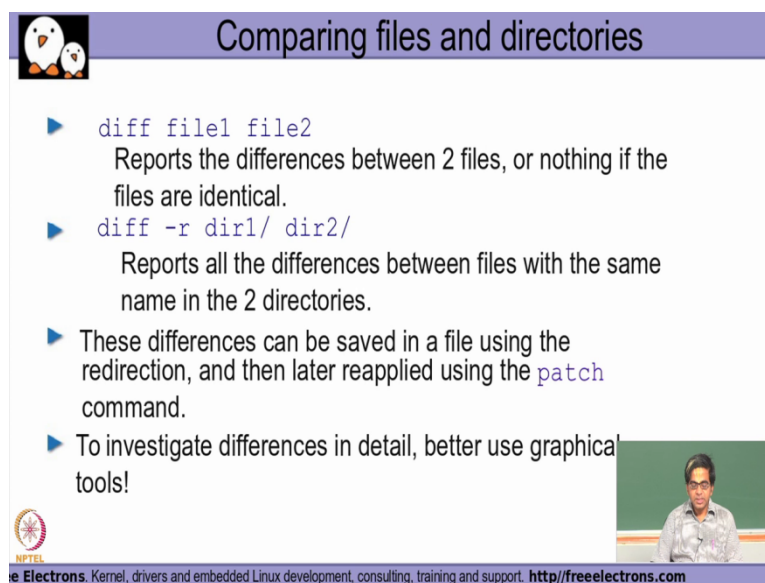


Information Security
Sri Vasam V S Principal Consultant
Department of Computer Science and Engineering
Indian Institute of Technology, Madras
Module 24
Linux File Comparison

In this module we will actually be looking at a very important set of tools that is used whenever we are dealing with different types of files and trying to compare the contents of the files on a Linux System.

(Refer Slide Time: 00:28)



Comparing files and directories

- ▶ `diff file1 file2`
Reports the differences between 2 files, or nothing if the files are identical.
- ▶ `diff -r dir1/ dir2/`
Reports all the differences between files with the same name in the 2 directories.
- ▶ These differences can be saved in a file using the redirection, and then later reapplied using the `patch` command.
- ▶ To investigate differences in detail, better use graphical tools!

NPTEL
Free Electrons. Kernel, drivers and embedded Linux development, consulting, training and support. <http://freeelectrons.com>

So one of the very very basic commands that is used for comparing two different files on a Unix system is what is called as a diff command, so if I give two arguments diff command file1 and file2 it reports the differences between the two files right up to the point where the differences are present, so it will report the difference in the line number in which these two files are getting to be different.

On the other hand, if there are no differences at all between the two given files they are exact replicas of each other, then the diff command will not report anything as part of the output. So if the user finds that diff is not reporting any output the user has to interpret it as saying that the the the content of the two files that is actually been given to the diff command is exactly identical. Now if I want to really compare the two directories I could actually make use of the minus r option where I will supply the arguments of two directories that I want to differentiate to the diff command and it will all it will report basically all the differences in

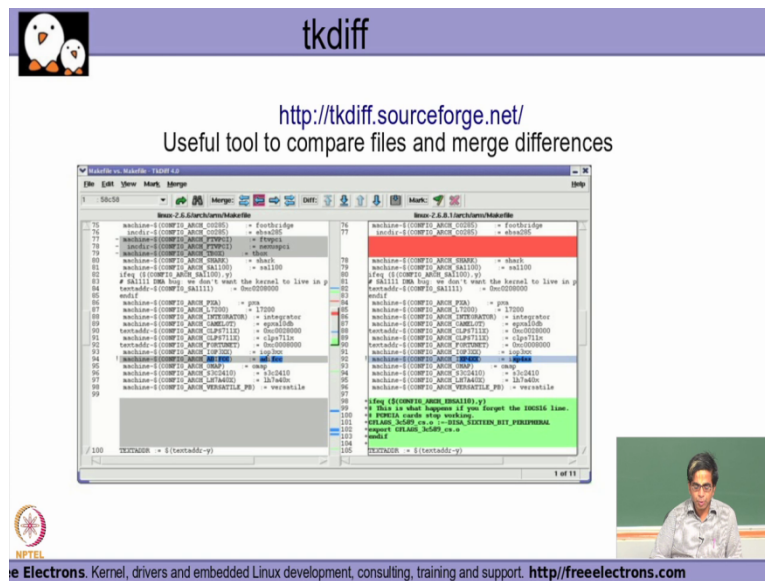
the files between the dir 1 and the dir 2 that has actually been supplied as arguments to the diff command.

Now if I use some very specific operations specific options in the diff command like the minus e option it basically gives me the difference in a format which I can make use of later on with the patch command. Now what exactly is a patch command is if I want to sort of patch a file in some very specific lines alone among my entire file then that is when I use the patch Command.

So when I when I run the diff with the minus e option instead of it generating the output exactly in lines which have which in which the content is differing between the two files it also gives me the context of the difference in which line which ever visible in the difference is coming in, so let us say that line number is different in file1 and file2 I when I use the minus e option to diff instead of just printing the line number 10 alone it also prints me 3 lines before line number 10 and 3 lines after line number 10,

So essentially from line number 7 to line number 13 which is very useful for applying it with the patch command later on if I want to just update only certain portion of my file, right. So that is again a very powerful option to the diff command which is commonly used whenever you deal with very large source code files in which require some sort of a frequent updates as patches to the individual code files.

(Refer Slide Time: 03:39)



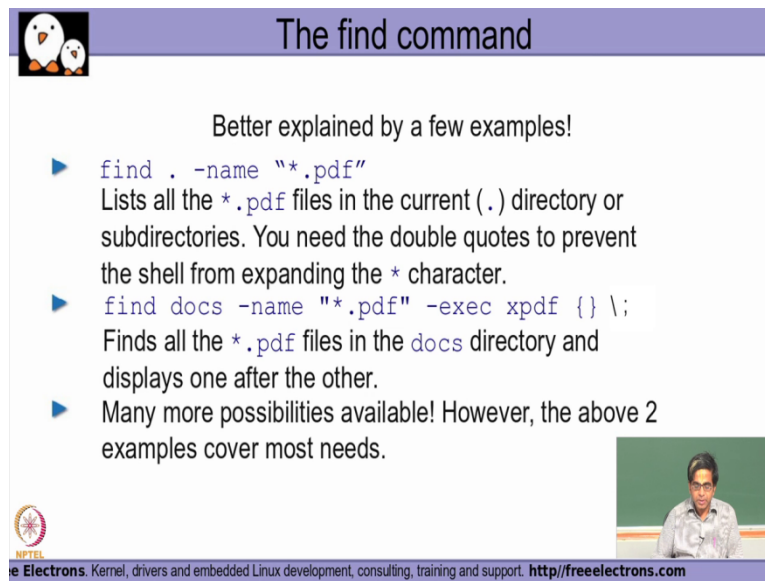
So there are different versions the gui version of the diff command that is also available like one example is a tk diff, so here it gives you a graphical interface and tells you mechanism by which it shows the differences very clearly here.

For example if you see that you have these three lines added in the first file and these three lines are not available in this particular second file that you've given as an argument which has been indicated with a particular colour right, on the other hand in the first you don't have certain lines are there in the second file this has been indicated in a different colour by this tool.

So basically this tool is also running a diff command but for very interactive and very intuitive way by which user can easily come to know the differences, the differences are actually projected in a graphical format for the user to quickly understand where the differences are in the contents between the two given files.

so another application that does something very similar is what is called as a compare, which basically tries to compare the differences and optionally merges the differences also if you explicitly ask it to merge, so this a, this is typically a tool that comes by default in the fedora core distribution of the Linux OS.

(Refer Slide Time: 05:06)



The find command

Better explained by a few examples!

- ▶ `find . -name "*.pdf"`
Lists all the *.pdf files in the current (.) directory or subdirectories. You need the double quotes to prevent the shell from expanding the * character.
- ▶ `find docs -name "*.pdf" -exec xpdf {} \;`
Finds all the *.pdf files in the docs directory and displays one after the other.
- ▶ Many more possibilities available! However, the above 2 examples cover most needs.

Free Electrons. Kernel, drivers and embedded Linux development, consulting, training and support. <http://freeelectrons.com>

Now next command that is very very commonly used when you are dealing with files and trying to understand the locations where the files are present is what is called as a find command and this is a very very powerful command and it it is having lots of options and lots of features available inside that which could be very very handy in a lot of situations, we will just take a look at a couple of examples here alone and then if you look at further details in the main page of this command you will be able to understand why we are referring to this is a very very powerful command, now if I want to basically find a file with a particular pattern.

I would basically used this command saying find dot minus name and give the pattern, now what we are basically saying is find in this particular location so what is this location since you've mentioned it as a dot that means I am asking it to find it in the current directory in a recursive manner because find by default always does it recursively and we don't need to explicitly tell it with any specific option saying that you try to find it in a recursive manner. So find from the current directory that is a dot directory in a recursive manner what to find we specify it with a minus name option to the find star.PDF so here we are basically saying all files that are ending with the dot PDF directory try to find it in the current directory in a recursive manner,

So how many our sub directories are present from my current directory go take a look and see if there are any files that are ending with dot PDF and if they are present just list me the

contents of the path name, so not the contents just list me the path names where those files are present from the current directory so it will give me the relative paths of all those files which are the extension of dot PDF at its end listed down as part of this command, right. Now let take a look at the next example `find docs -name star.pdf -exec xpdf open {} \;` braces close braces, then there should be a back slash here followed by a semicolon, right now look at this here `find docs` so we are basically telling that in this particular example inside this docs directory find files what files it has to find minus 10 star.PDF,

So again as in the previous example find all the files which are ending with dot pdf that is why it is star dot PDF so find all the star dot PDF files in the docs directory and whenever you find a dot PDF file what should the find command do that is basically what we are specifying with the minus exec option, right. So minus exec xpdf so it is basically an instruction to the find command to say whenever you find the dot PDF file execute this particular command xpdf, right. So xpdf as we all know is basically the command application that is used to open up a a PDF file,right.

Now what is this open braces and close braces brackets that you see here, the find command will automatically replace the open braces and the close braces with the path name where this particular PDF file has been found, because the xpdf command has to be given the path name in which this particular file is present otherwise xpdf command will fail, right and when we run the find command we wouldn't know in which path it is going to be finding because that is basically why we are running a find command itself in the first place.

So whenever it finds the dot PDF file recursively inside the docs directory for each of those dot PDF files this minus exec option will open up an xpdf command and give that particular relative path name, relative file path name for that found.pdf file because we have given open braces and close braces here, so that substitution of open braces and close braces

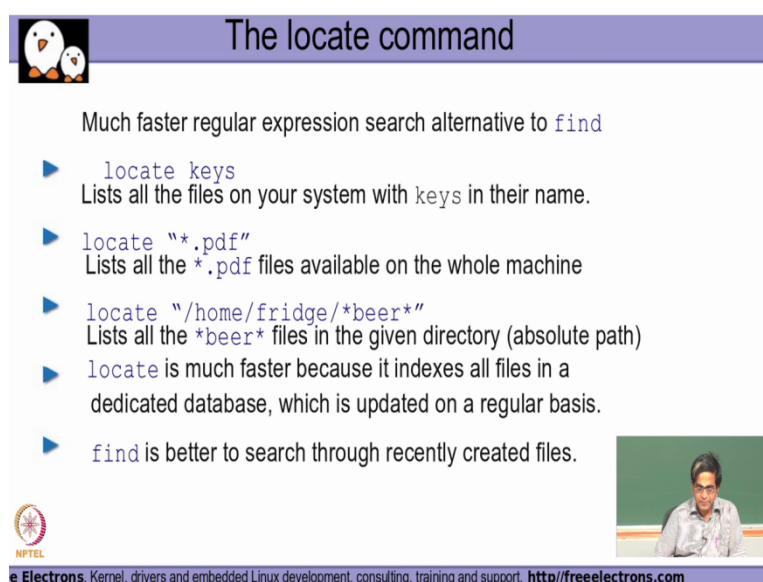
will be done automatically and replaced with the relative path name by the find command whenever it finds a particular xpdf dot PDF file right, now if I for example find that there are three dot PDF files, right so there are three files,

The files name that are actually ending with dot PDF in all the sub directories and the files in docs directory because we've run the find command in this with these options there will be 3 xpdf executions that will happen each for those dot PDF file since we have 3 pdf files it will

open up 3 xpdf application instances in which each of those PDF files will be opened up, right.

Now every time that PDF file is found by the find command the open braces and close braces will actually replace that relative path name and then give that relative path as an argument to the xpdf command and then the xpdf will be getting started up, right. So these are the two very common scenarios in which we use this kind of a very powerful find command and then there are more powerful ways of using it which we can actually take a look at by looking at the main page of this find command.

(Refer Slide Time: 11:36)



The locate command

Much faster regular expression search alternative to `find`

- ▶ `locate keys`
Lists all the files on your system with `keys` in their name.
- ▶ `locate "*.pdf"`
Lists all the `*.pdf` files available on the whole machine
- ▶ `locate "/home/fridge/*beer*"`
Lists all the `*beer*` files in the given directory (absolute path)
- ▶ `locate` is much faster because it indexes all files in a dedicated database, which is updated on a regular basis.
- ▶ `find` is better to search through recently created files.

NPTEL

Electrons. Kernel, drivers and embedded Linux development, consulting, training and support. <http://freeelectrons.com>

So alternatively other than the find command we also have the locate command with which we will be able to find the the location where a particular file is available so that is another mechanism by which to sort of get the location where the where the file that has been given is an argument is available,

So now what is the difference between the find command and the locate command if both of them are really trying to find out the location of the given file name, right. Now what locate does is, locate basically tries to find the location from a database that is actually created first time, right. Now first I will have to have a database that is build and whenever you give any file name locate tries to quickly take a look at the database and sort of indexes directly into if into the location where a that particular file will be found and then immediately displays the location of the file, ok.

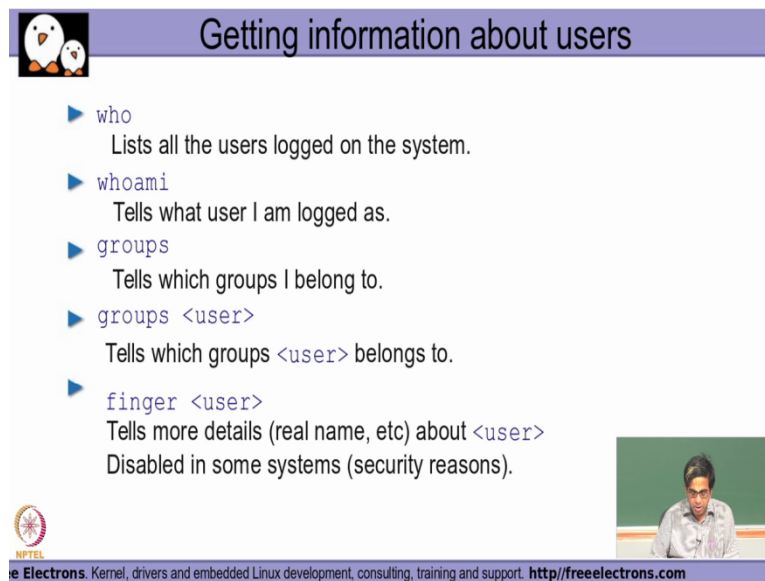
Whereas find is useful whenever I dynamically insert newer files into that my target directory location it will be able to even find those files that has actually been added just a few units of time before right,

The problem with locate is that if I have added any new files into my directory in which I am trying to find any particular file I will have to re generate the database that is being used by the locate application before the locate command is used is successfully able to find that particular file path, right. So but the advantages of locate command is it is extremely fast right.

On the other hand find will take its own time depending on how big my directory is in which I am trying to find the file but find is very successful is very useful in finding even the files that I have added very very recently so both find and locate has his own advantages and disadvantages and depending on whether we are trying to find a file with which has only been added very earlier and not something which is constantly getting updated or we're trying to find which find a file which is actually might be also very recently updated the user is expected to use the either the locate command or the find command appropriately, right.

So find is very slow if the directory that is being used for finding is is is containing a very large number of files but the advantage with this is it is very efficient even in trying to find the files that has been recently added on the other hand locate is extremely fast because it is using an internally a database that the flip side is that if it all i have added any new files after the database has been generated, I'll have to regenerate the database again then only the locate command will be successful, right. So that's basically the difference between the find command and the locate command.

(Refer Slide Time: 14:55)



Getting information about users

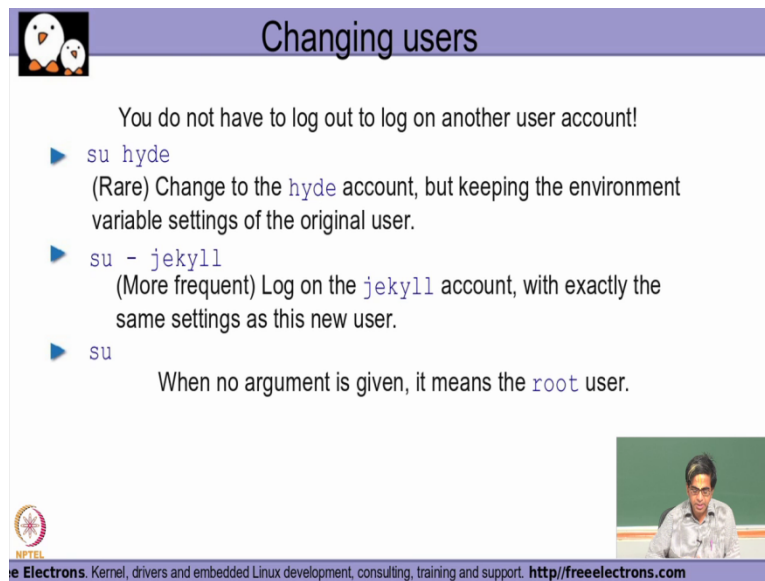
- ▶ `who`
Lists all the users logged on the system.
- ▶ `whoami`
Tells what user I am logged as.
- ▶ `groups`
Tells which groups I belong to.
- ▶ `groups <user>`
Tells which groups `<user>` belongs to.
- ▶ `finger <user>`
Tells more details (real name, etc) about `<user>`
Disabled in some systems (security reasons).

NPTEL
Free Electrons. Kernel, drivers and embedded Linux development, consulting, training and support. <http://freeelectrons.com>

So some of the other information command that I need to that I need to use whenever I am have a requirement to find about users is the `who` command will give me all the users who are logged into the system, `who am I` will give me my own user ID so as how my user ID is recognised by the system,

Groups will tell me to what are groups this currently logged in user ID is belonging to so whoever is running the command called `groups` who who that particular user is belonging to which groups will be reported by this command, on the other hand if I give a particular user name as an argument to `groups` it will tell me which all groups that particular user is belonging to and `finger` command is actually will give me details about the particular user and then quite a few system for security reasons you will find that this `finger` command is actually disabled.

(Refer Slide Time: 15:58)



Changing users

You do not have to log out to log on another user account!

- ▶ `su hyde`
(Rare) Change to the `hyde` account, but keeping the environment variable settings of the original user.
- ▶ `su - jekyll`
(More frequent) Log on the `jekyll` account, with exactly the same settings as this new user.
- ▶ `su`
When no argument is given, it means the `root` user.

Free Electrons. Kernel, drivers and embedded Linux development, consulting, training and support. <http://freeelectrons.com>

So if have a requirement which change user from one user to another user there is a command called su which is actually stands for switch user but that is sort of rarely used again for a simple reason that it is not very secure to make use of this so if at all I need to make use of it I.

I say su followed by the user ID to which I want to change, right. My own user ID to that particular user ID henceforth the system will be recognising me as that particular user name which I have given as an argument, so only thing is here when I actually used the su unless and until I am a Super user of the system I will be asked to enter the password of the user to which I am trying to change the user ID too, right.

So only if I am basically successful in giving the correct password for that user, the system will allow me to change myself to be of that user, right. If I uses su minus a user ID what happens is that the entire environment my current environment the shell environment will be getting changed to the environment that has actually been set for that particular user, so in the previous module we had looked at a while called dot bash rc for example which we were discussing basically gets executed every time the cell process gets started.

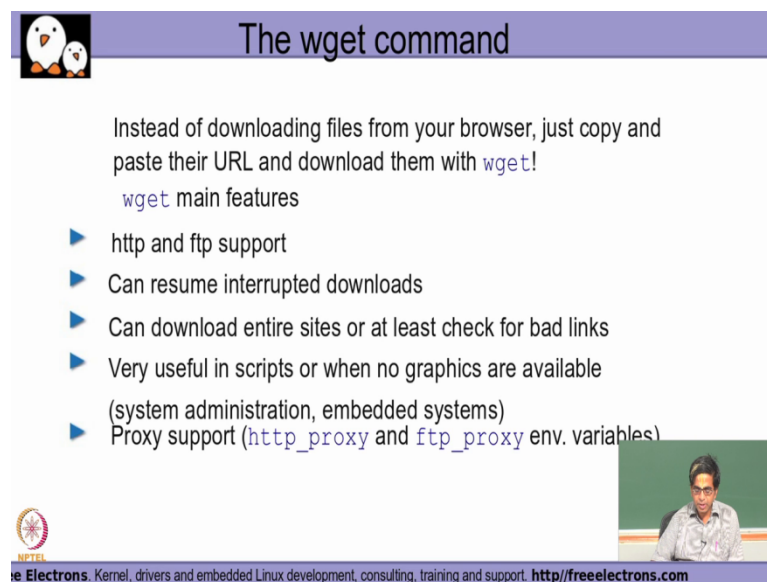
And we also discussed that this dot bash rc file is a hidden file available for every user in their home directory, right. So when we use minus option to the su command and then specify the user id, what is actually happens is that whatever user id is specified here, the password

for that user id will be queried assuming that I give the correct password the authentication is successful by the system for changing this is user ID the the shell that I get right?

Now will have the dot bash rc and the entire and the entire environment to be executed for this particular user ID execute so that my current shell environment will typically reflect whatever is the environment that has been set for this particular new user ID that I am trying to change myself too, right.

Now on other hand if I use su command without giving user ID the system will assume that it you are trying to change to the root user and then becoming a sort of a Super user and it will try to query you for the the root password in this particular scenario, right.

(Refer Slide Time: 18:28)



The wget command

Instead of downloading files from your browser, just copy and paste their URL and download them with `wget`!

`wget` main features

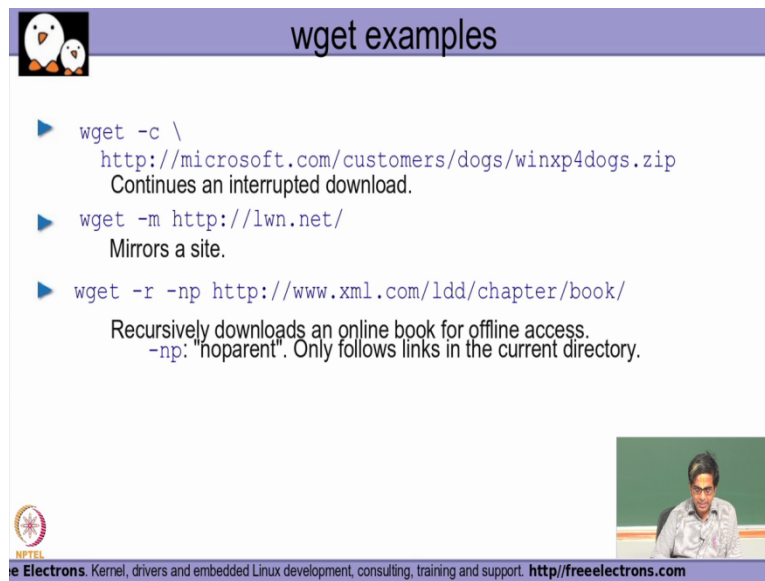
- ▶ http and ftp support
- ▶ Can resume interrupted downloads
- ▶ Can download entire sites or at least check for bad links
- ▶ Very useful in scripts or when no graphics are available (system administration, embedded systems)
- ▶ Proxy support (`http_proxy` and `ftp_proxy` env. variables)

Free Electrons. Kernel, drivers and embedded Linux development, consulting, training and support. <http://freeelectrons.com>

So another command that is very useful is what is called as the wget command, so the wget command is a sort of a command that you can use for simulating a browser functionality where if you for example give a URL in your browser you are a location the browser actually downloads that particular content of that URL and that displays it in your web page, right.

On the other hand if you want to have a a command line utility which simulates a browser then you can actually make use of wget command where an if you say wget and give a filename with the remote server name also available as part of the filename the wget command over the net will try to download the file from that particular username assuming that you have authenticated yourself successfully wherever required and then will get the contents for you and then display the contents, right.

(Refer Slide Time: 19:28)



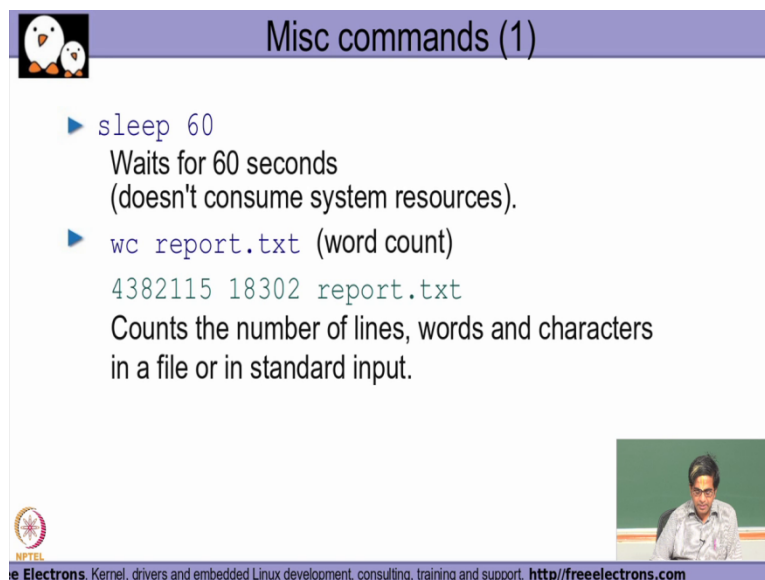
wget examples

- ▶ `wget -c \`
`http://microsoft.com/customers/dogs/winxp4dogs.zip`
Continues an interrupted download.
- ▶ `wget -m http://lwn.net/`
Mirrors a site.
- ▶ `wget -r -np http://www.xml.com/ldd/chapter/book/`
Recursively downloads an online book for offline access.
-np: "noparent". Only follows links in the current directory.

NPTEL
Free Electrons. Kernel, drivers and embedded Linux development, consulting, training and support. <http://freeelectrons.com>

So that's basically what the wget command actually does, so there are different ways of using the wget command which we can actually try it out over the network and then it's comfortable with using these different options.

(Refer Slide Time: 19:44)



Misc commands (1)

- ▶ `sleep 60`
Waits for 60 seconds
(doesn't consume system resources).
- ▶ `wc report.txt` (word count)
`4382115 18302 report.txt`
Counts the number of lines, words and characters
in a file or in standard input.

NPTEL
Free Electrons. Kernel, drivers and embedded Linux development, consulting, training and support. <http://freeelectrons.com>

So there are some other commands that we need to be aware of from the basic Linux point of view. One is that what is called as the sleep command, so if you say sleep followed by a certain number of seconds, the system will basically sleep for that particular duration of time that you have mentioned as an argument to the sleep command without doing anything and at the end of the

same time it will be guaranteed that during that sleep time the system will not consume any kind of a system resources like the processor or whatever it is, then another command that is very commonly used is a WC command.

WC basically stands for word count so if you give a filename as an argument to the WC command it will basically display you the different data so it will try to display you how much of words, how much of lines, how much of characters is present in that particular file appropriately as as the options that you've actually used in your command line.

So it has the capability to print the number of words, it has the capability to print the number of lines, it has the capability to print the number of characters along with the file name that you've actually given as an argument, so in in in in a scenario where you would want to know this kind of an information apart from just knowing the size of that file alone, then wc command comes in handy for it, right.

(Refer Slide Time: 21:11)

Misc commands (2)

- ▶ `bc` ("basic calculator?")
`bc` is a handy but full-featured calculator. Even includes a programming language! Use the `l` option to have floating point support.
- ▶ `date`
Returns the current date. Useful in scripts to record when commands started or completed.

NPTEL
Free Electrons. Kernel, drivers and embedded Linux development, consulting, training and support. <http://freeelectrons.com>

So this is a command which is equivalent to your calculator program so you could do all kinds of basic arithmetic operations with the the `bc` command, so if you use the minus `L` option to the `bc` command, it also helps you to do floating point, arithmetic operations with the decimal values also factored but if we, if the minus `L` option is not used to the command line, it will only do integer related arithmetic operations, then `date` is another command that it is very handy to give the current system date like a normal date it also prints the system date, the time.

It prints the time zone as part of its output, this is very useful and handy to be displayed when you are running shell scripts as part of the scripting to really understand when a script is actually started executing and to find out when a script is actually ended executing the difference of which will typically give you the total time that particular script has taken, so this is one example and there are lot of examples in which which could typically be used for running the date command.

Thank you.