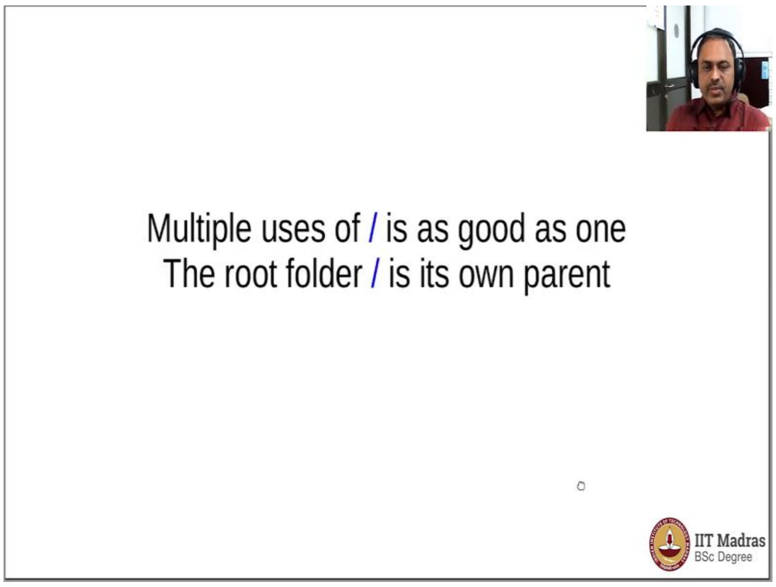
System Commands

Online Degree Programme
B. Sc in Programming and Data Science
Diploma Level

Prof. Gandham Phanikumar
Department of Materials Engineering
Indian Institute of Technology- Madras

Lecture 4
Simple Commands in Linux - 2

(Refer Slide Time: 00:18)



Multiple uses of / is as good as one
The root folder / is its own parent

IIT Madras
BSc Degree

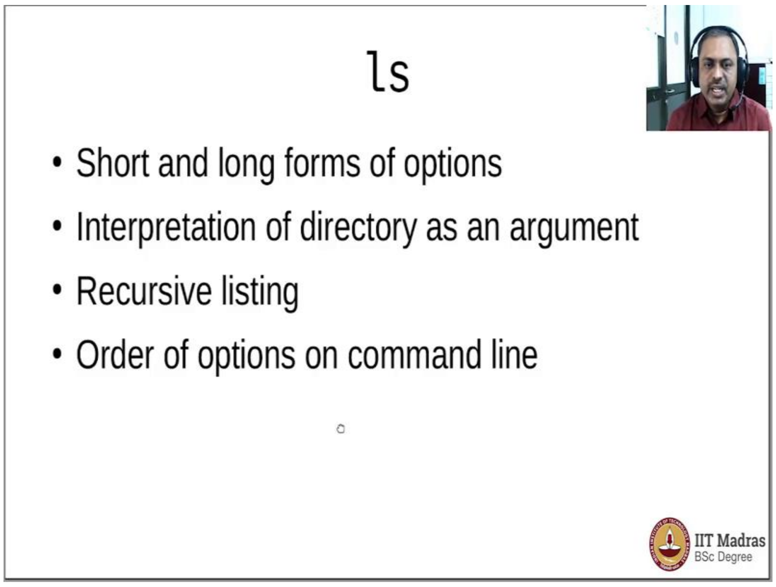
Welcome to the second session on simple commands in Linux. So, here we explore some more commands in the Linux operating system and it is hoped that at the end of the session you will be more comfortable with the command line environment and also looking around in the file system and knowing about the machine a little more than earlier. As you know we have been introduced to the forward slash which is the character used to separate the directories within the file system hierarchy.

And it is also the name of the root file system the starting point of the file system in the Linux environment. We will see that we can use the forward slash multiple times and it is as good as and if we try to access the parent folder while in the root folder we would actually stay in the root folder which actually means that the root folder is its own parent.

(Video Start: 01:10)

So, as he said multiple users of the forward slash is as good as a single usage. So, I want to visit the user bin. So, I will type multiple times and you would see that I am in the same directory as slash user slash bin I. Now go to the root folder and try to access the parent folder I see that I am in the root folder still and you can do it multiple times and it has implies that the root folder is its own parent.

(Refer Slide Time: 01:41)



ls

- Short and long forms of options
- Interpretation of directory as an argument
- Recursive listing
- Order of options on command line

IIT Madras
BSc Degree

The ls command has both short and long forms of the options and when you give a directory name as an argument the way it is interpreted is something that we have seen in a moments. And you can actually recursively list the files and directories within a particular folder using the -r option of ls and the sequence of options to be given in the command line of ls command is also something that is very flexible and we can look at it in the demo.

Let us explore the home directory a little further. So, I am not in the home directory and when we give the -l option to the ls command by default it is showing you all the files and directories within the current directory. Now if I were to give a folder name let us say level 1 then it would show what are the files and directories within the directory level 1. So, it would go below the level 1 and then show you what is there inside.

Now I want to only look at the permission string and other details in the long listing format of the level 1 directory but not what is inside that. So, for that what we need to do is give additional option d. So, that it would show you a string giving you the long format a further level 1 folder as it is also shown along with other files when we did it using the command ls -l.

So, you see that this -d option is basically triggering that you should not traverse inside the directory and show the listing. We could also look at multiple options in one go. So, -ldi level 1 and you will see that we are seeing the inode number and we are also seeing the long listing format and also further directory which is level 1. Now we may give these options in any order.

So, let me clear the screen ls -l level 1 and then -di and then I would change the sequence -t level 1 -li ls -i level 1 -ld l level 1 -l d I or idl. So, you can see that I am giving the options in very many ways but the output is identical it just means that the ls command really does not bother about the sequence of the options it would take all the three options that have been given to it and then produce the output which is basically the long listing for the folder level 1 along with the inode number to be displayed on the screen.

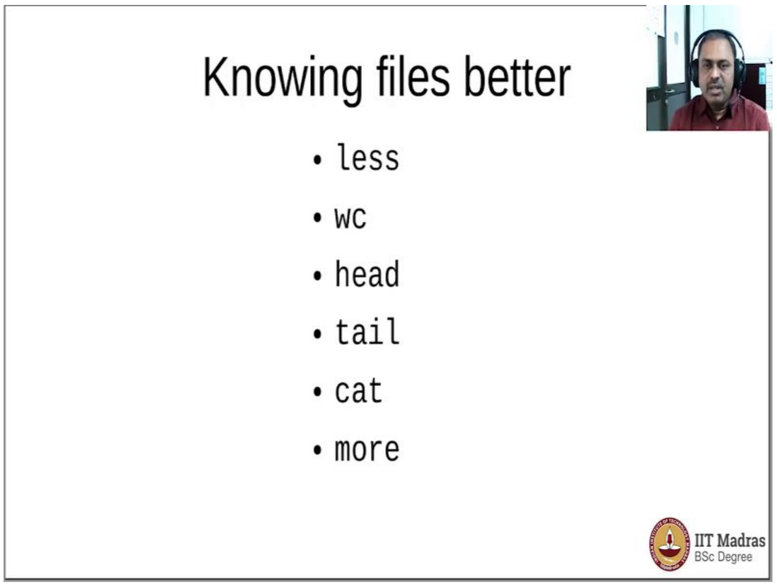
Now this flexibility in terms of giving the option either before or after the argument is not available with every command because sometimes some of the options require the argument to immediately follow them in which case the sequence of how you provide them on the command line would be quite important. So, generally it is a good idea to follow the sequence in this following manner.

Where the options are in the middle and collected together as far as possible and then the arguments are following after that okay. So, this is a very standard way of writing a command. If you see the manual page for the ls command you would notice that there are some options which have a long format also that is a long option also is available and they are basically equivalent and some of them do not have that.

For example –l it does not have a long form whereas if you look at the -i it has a long farm -inode and then if you look at it, it has a long form called directory. So, which means that I can actually type in this manner -id l and then level 1 I could also type in this manner -l --directory --inode level 1. So, you could see that the output is same except that in this form if you write the intention of the command is very clear because the long form of the option is a little more human readable.

However it is only for those who wish to type a lot but it is a good idea to also learn the short form because the beauty of the command line environment is also to be able to write a very short command. So, that to do a lot of work.

(Refer Slide Time: 06:28)



Knowing files better

- less
- wc
- head
- tail
- cat
- more

IIT Madras
BSc Degree

So, there are some commands available which help you know the content of a text file a little more. So, we have already seen the less command and. Now we would see the wc command the head command the tail command the cat command and the more command all of these let you inspect the contents of a text file. We will take the example of slash it is his slash profile to use these commands.

So, we go to the slash etc folder which contains the configurations and there is one configuration which will be very important for anyone who uses a shell and that is called the etc profile. So, we will understand what is in that profile later as we learn about the shell scripts etcetera for now let us look at the content ls -l profile is there any file like that and you see that yes there is a file like that.

Now what are the contents of it? So, earlier we learned about the less command. So, if you type a less profile and you will see the contents of it and when the contents are scrolled at the end you can actually see that available as bracket end which means that the contents are over and you may. Now press q to come out of it. So, this is a short file fitting in one screen itself.

Now there are also other ways of looking at the content. So, cat is one such command. So, it is to cat it or concatenate the text onto the screen. So, if you type cat profile it would just dump the contents of the file onto the screen and then exit. So, there is no more prompting that is required for you to come out by pressing a queue or anything like that. The disadvantage of cat is that you cannot actually move back and forth to look at the content page by page and you cannot also come out halfway through when the text is being displayed on the screen whereas in the case of ls you have such features.

So, sometimes if the file is very long cat is not the best way to look at the contents because the file would keep scrolling on the screen and depending upon how many lines are stored by your terminal emulator you may not actually be able to see the initial content by the time it finishes displaying the whole file on the screen. Now there are some more such commands available.

So, let me clear the screen now. More is also one command available. So, more is going to behave similar to less it would also allow you to look at the contents of a file page by page if there are multiple pages. Now you may also look at it just the first few lines or the last few lines. So, there is a command called head and if you type head profile you will see the first ten lines.

So, let us look at the man page of health and it shows you here that it actually prints the first ten lines of the file but if you are interested in less number of lines you could also do that using -n and then the number of lines. So, let us do that I would say type -n and then just the five lines of the file called profile and you see that exactly five lines have been displayed.

There is another command called tail okay as the name indicates you can view the last few lines of the text file tail profile so you can see that the last ten lines are there let us look at the man page for tail and it tells you that it prints the last ten lines of the file to the screen. So, just like a head command the tail command also supports reducing or increasing the number of lines that are to be displayed. So, here I am trying to show the last five lens of the file profile and here are the last five lines.

As you can see there are multiple ways to inspect the contents of any text file within the next environment before you open the file it may be a good idea to inspect how many lines are there? How big is a file? So, there is a command for that called wc. Look at the main page of wc it tells you that it will show you how many new lines are there which means how many lines are there and how many words are there and how many bites are there in each of the text files that we are giving.

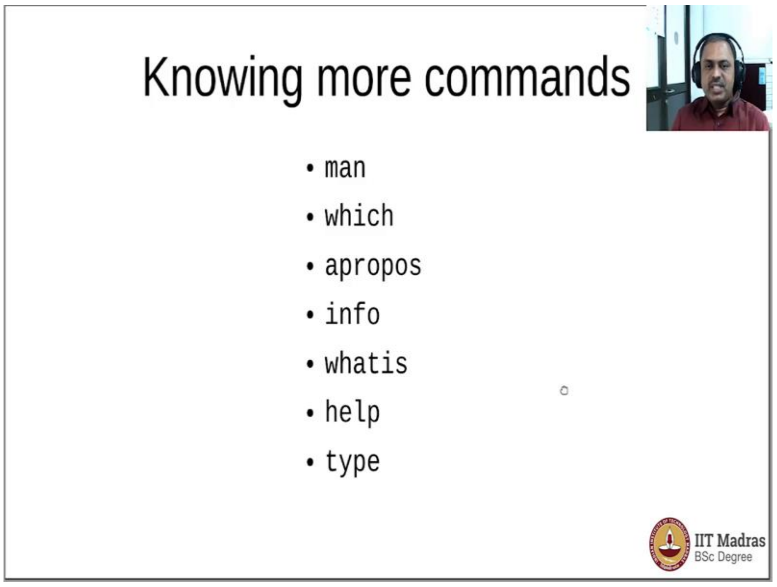
So, let us go ahead and plan it on the file profile. So, wc file I said that there are 27 lines in that particular file and there are 97 words and then 581 bites it is occupying on the disk. So, if you look at the content in the long listing it would show you that five hundred and eighty one is the size of the file called etc profile. Now sometimes we are usually interested only the number of lines for the purpose of trimming some of them displaying a specific line etcetera.

So, for that we may use option -l okay. So, it just gives you the number of lines. Now we may be interested in knowing where are these commands and where are they located etcetera. So, there is a command called which okay. So, we say which less and it will tell you the location of that particular command it is sitting in user bin. So, we may want to know what is this which itself.

So, man which and it tells you that you can actually locate the okay. So, there is another command called whatis and using that you will have a very brief description of the command okay. So, man page is showing you a complete manual page but what is would show you just the first line of the man page where the brief description of that particular command is available.

Now you will see that there is also another command called more and you will see that this command is also in user bin. So, let us see these two how big they are ls -l and then I give the part to know more about them okay and ls -l user bin. Now you could see that the command called more actually occupies less space it is a smaller command and the command called less is actually a bigger executable you can see that it occupies 180 kilobytes. The joke about this in Linux is that less is more.

(Refer Slide Time: 13:00)


Knowing more commands

- man
- which
- apropos
- info
- whatis
- help
- type

IIT Madras
BSc Degree

We have seen that the manual pages which come bundled along with Linux help us in learning more about a command and often we may want to discover more commands for a specific purpose. So, you could search for them not necessarily going to the internet but within the Linux environment itself at the batch prompt by using the apropos command.

So, what it would do is for a keyword it would show you all the commands which would have that keyword in description. And you could also actually browse through various sections and various commands in those sections using a system called info. And once you find out the name of any command you can learn more about it obviously from the man pages but also to know where is it located and what type of a command is it a script or is it an alias or is it a binary file sitting in one of the binary directories like user bin etcetera.

So, we can explore the commands all by ourselves without having to go to the internet using these set of features that are available with the Linux operating system. Let us have a demonstration of how to discover new commands by using the features that we have just now mentioned. So, apropos and you can give a keyword which would be searched for. So, let us give a keyword like who.

So, I want to know who are logged onto the system etcetera okay. So, when you type apropos who it will give you a set of commands which will help you to know who are logged on and one of them is the who itself and of course there is another command called who am I which we have used in one of the earlier sessions to print the name of the user who is locked on okay.

Now obviously to know more about it you can type a man who and learn more about it from the manual pages and what is this command apropos? you could actually ask for it by typing which apropos and you would see that it is sitting in user bin apropos. There is another command called whatis and let us say who. So, if there is a command called who I want a brief description of whatis that command and what is actually do that okay.

So, you could say whatis ls, whatis ps whatis wc whatis less what is more. So, you could actually ask what are those command and what is would give you a very brief description of that command. Now you want to see where is this command located namely the whatis command. So, which whatis would tell you that it is located in user bin. So, let us see whether these are both same or not.

So, if you type ls -l user bin apropos it appears that user bin apropos is actually a symbolic link to whatis command. So, it means that actually when you type whatis or apropos is the same executable which is running however the output is different. For example if I type whatis who and apropos who then the output is different. So, how is this possible? The reason is that in Linux every executable would know in what name it has been invoked.

And depending on the name that it is invoked it could actually have different behaviour okay. So, you could actually have these features tested out as part of your own baskets once you start learning that as part of this course. Now apropos is also having the same output as man -k and you will see that the identical output is coming. So, searching for keyword using the manual pages is the same as what apropos is doing.

Now there are also some more features such as available. For example you can type for help and what actually help does is show you certain keywords that are actually reserved for the shell which you are running which shell are we running? So, we are running back shell okay you can see here you are running batch shell and the help command shows you but all the special words that are used by the batch environment.

I will show you the info also feature you could see that you could go to any a section and you can browse through the commands and you can put the cars on any of the commands for example chmode I will go there and then press enter and you would see the information about the chmode that is shown here. And we can actually go back by using the left angle bracket which is same as shift plus comma okay you come to the main window.

And you could then go to some other command and read about that and then again come back to the man. So, you can actually almost like a browser text based browser you can actually browse through the commands that are available in the info system that came along with the Linux operator. Now sometimes you may want to know what type of a community. So, type is a command to tell you what type of a command.

So, let us say type type. So, it says it is a shell bulletin. What it means is that the command type is actually being offered to you from the shell and not from the operating system okay. So, type ls. Now it says that ls is alias to type command. So, which means that it is actually coming from the command ls after having been alias with certain option and which ls will tell you that it is actually coming from the operating system because there is executable available can we check about that.

You can see that it is an executable and it is sitting in the user bin directory we may even find more about it by using the file command and you will see that it is an ELF 64 bit executable for X86 architecture and so on and occupying about 142 kilo bites and therefore ls is actually an executable and therefore offered from the operating system whereas type is actually coming from the shell.

So, similarly help is also a shell bulletin. So, sometimes when you are running a command it is a good idea to know is it being offered by the shell or by the operating system or is it an alias. So, that brings the context to aliases? What are aliases? Aliases are nothing but nicknames that are given to certain commands sometimes you may want it for convenience.

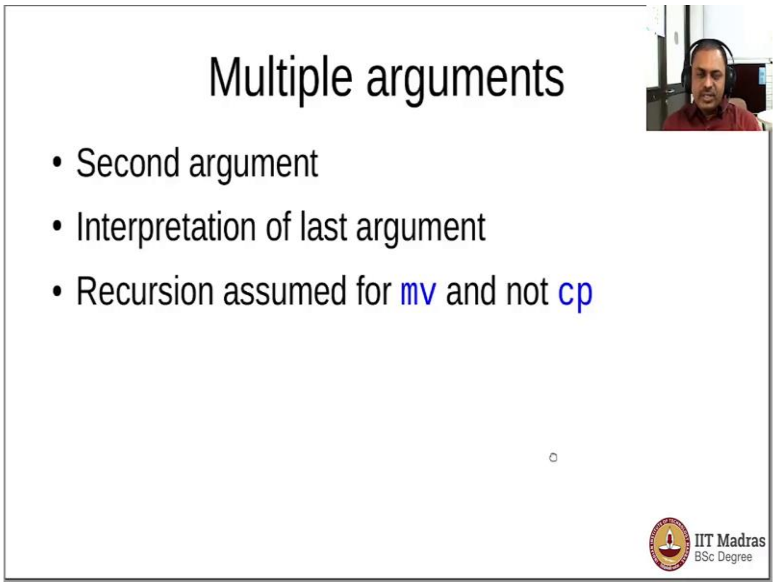
For example you may want to very frequently look at the long listing of your directory ok and you may want to give that as an alias. So, I may say alias and let us save long listing is equal to ls -l. Now you see that l l is a new command that you have created and if you type l l you actually get the long listing of the current directory. Now you can ask what is this l l? And it tells you that l l is alias asked to ls -l.

So, you understand it is not a command coming from the operating system is alias coming from what you have configured just a while back. Now you do not want that you can remove alias ll. You may want to find out what are alias that have been configured for you in the batch and you logged on and just type alias and then you would see as list of alias that are there.

Some of these are actually created by the user and some of them may be created by the administrator on behalf of the user by putting them in etc profile or some such a script file which is executed whenever the batch environment starts. We will learn about that also a little later okay. Now you can always remove some of these aliases. So, I want to remove let us say this alias.

So, I could just say un alias l okay. Now I will again type alias and you could see that that particular entry has been missing. Now what happens is that if I type just l obviously this must have come in but earlier if you were typed then it would actually give you the output which is corresponding to ls -C F as options.

(Refer Slide Time: 21:38)



Multiple arguments

- Second argument
- Interpretation of last argument
- Recursion assumed for mv and not cp

IIT Madras
BSc Degree

So, by now you are familiar with the concept of arguments and options. So, options are basically enhanced features of the command that we are giving and arguments are specific names of files or directories that we are giving on the command line. So, that the command would act on those respect files are territories as requirement needs. Now when we are writing a command we could actually have multiple arguments on the command line and the way those arguments are interpreted depends upon the command itself.

So, we should definitely read the man page of every command that we are going to use frequently to understand how this interpretation is taking place. So, if you look at the commands like copying using cp or moving a file using mv. Then if you give two arguments what happens is something that we have already seen and let us see how this is actually changing when the second argument is directory or if it is a last argument among many many arguments that are given on the command line.

And sometimes the recursion is sometimes assumed for some command line and sometimes you have to explicitly state it. So, let us look at that difference also in a moment. So, we are. Now in our home directory and we will see how the number of arguments can be specified. So, I will take the example of touch command to create some files. Let me just see that there are no files in my home directory. So, I will create some empty files for this practice.

So, I would say touch and let us say file1. Now if you give multiple files all of them will be created. So, this is the idea of a multiple arguments okay. So, if I give like this it means that all the three files will be created in one go and the time stamp will also be identical okay. So, we have seen that these three files have been created. Now to illustrate the way the directory will be interpreted as one of the arguments in the copy file we will create an empty directory mkdir my director okay.

So, then I would say copy file1 to my directory and what happens is that the file1 has been copied over to my directory. So, if you put ls you would see that the file1 is actually there in the home directory but if you go to the directory mydir and there is a file called file1 there also which means you have made a copy of that okay. Now let us say I type copy file1 to file2.

Now what happens is that if the second argument is a file then you are trying to override that file by this copy command and naturally it would ask you to confirm if you have given with the -i option and let us say I do not want to do that I said no and I have given earlier for cp which is -i and if you look at the man page of cp -i would correspond to interactive which means that it should prompt whenever there is an overwriting.

So, it is a good idea to have that alias kept. Now let me go to mydir and remove this file and come out and then I can. Now remove the directory also. So, our mydir will remove an empty directory. Now let me go back and create it again and see if I can do it without having to go and remove the files before I remove the directory. So, I create the directory again. Now I go to that directory and I create some file there well you see that there is a file that is available though it is a empty file but it is still a file that is there inside and I want to remove this directory.

And now you see that it is not empty and therefore you cannot remove it automatically. So, you may want to force the removal and you could actually give an option called -. So, rm -r mydir and if then go ahead and descend into the director and remove whatever is there in that. So, it would ask each time because I have asked for that the alias -i option I have given in the allies. So, you can see what is alias I have r -i.

So, because I have set the interactive option already it would ask me if I do not have that alias it would not ask me let us test it out already okay. So, I will alias rm then I will again see what is alias for arm there is no such alias. Now which rm it says rm is a command. So, you now see that this rm command is quite dangerous because it would not ask you while removing the directory okay.

So, let me just try that out make dir mydir go to mydir and touch a file okay then come back to the home directory rm -r mydir and you see that it actually deleted the directory without even asking you. So, which means that it goes by the philosophy of Linux of no nonsense and you intended to remove the directory and that is what has been done. So, it is a good idea for novice users to use the aliases with the –i so, that unintentionally you will not remove and the prompting actually makes you think once more before you say yes.

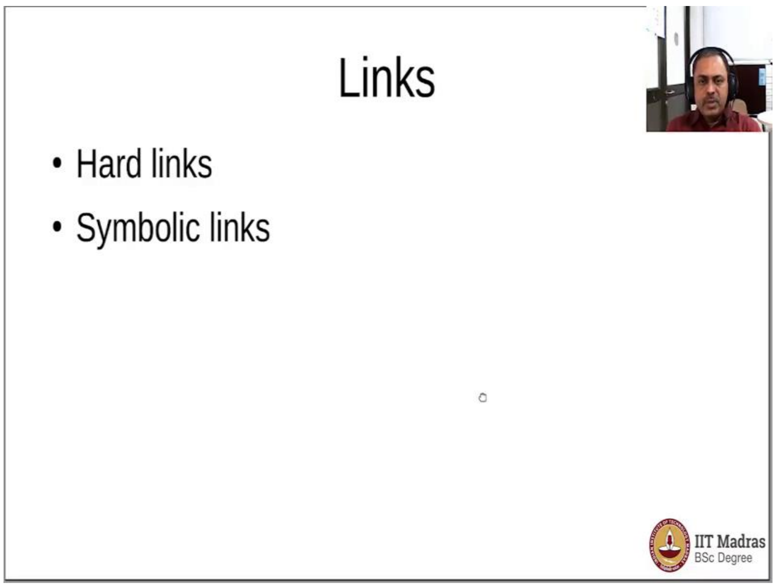
Now some options assume recursion and some options do not assume recursion okay. So, here I will just show you we make a directory and we go into the directory and create let us say two files in them come back to the home directory and I want to look at the contents of mydir. So, you can see that there are two files in them. Now I want to rename the mydir I would actually just use a move command.

But let us I want to copy to do some other name. So, what would happen? Copy mydir and I went to copy to another directory the second argument is actually a directory that does not exist and we would like to make a copy and that does not seem to help because it is not a directory that is existing right now. So, what you could do is copy -r mydir to mydir. So, if you give the -r option it would actually work because it now has created the second directory let us go and see that.

So, it has the files that we have created. So, which means that for the copy command the recursion is not assumed while copying it is asking whether you want it to be recursive or not. Now mv command does not ask for that you will see that let us say I want to move the mydir2 mydir3 and you see that it just renamed it and therefore there is no recursion that is being asked for.

So, what it in place is that some commands assume recursion some commands do not assume recursive and one must be aware of that while playing around with those commands.

(Refer Slide Time: 29:04)



Links

- Hard links
- Symbolic links

IIT Madras
BSc Degree

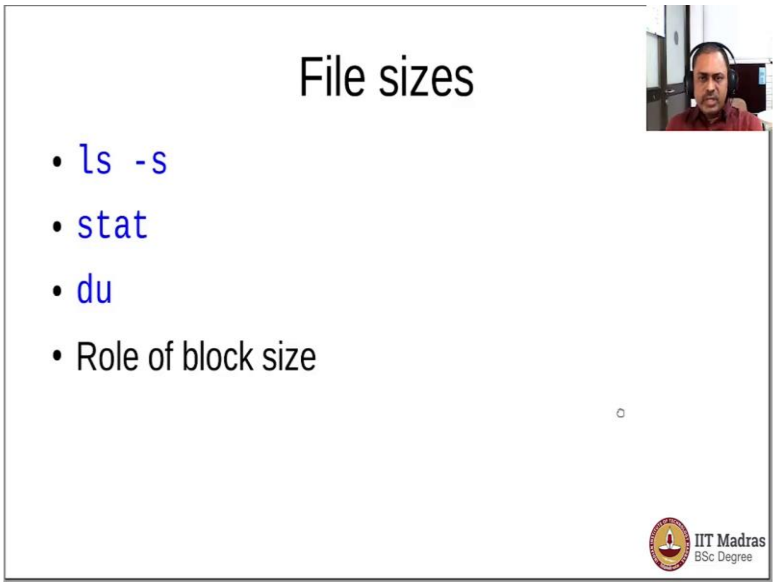
A little bit of discussion about the links is worthwhile. So, we have seen that both hand link and symbolic links are possible and we can create them for a given file and looking at the inode numbers we can actually determine whether they are hard links or symbolic links and also from the long listing of the file system also you could actually see whether their hard links are symbolic links okay.

Now let us look at the hard link and soft link concept. So, we have got the home directory here. So, let us create an empty file okay and you would see that the empty file is here. Now we would create a symbolic link to this file. So, that is done using the command ln. So, we would look at the man page for ln. So, ln is making the links symbolic link is done by using the -s option here it is -s option okay.

So, symbolic link and the source for which we are making the link and the destination which is the link itself, so, ln –s source destination that is how the format looks like and now when we look at the long listing you would see that file2 is a symbolic link to file1. And let us look at the inode numbers you will see that file2 has an inode number which is different from the file1 which means that there are separate entries.

The file2 is a separate entry but it is just a shortcut to the file1 okay. Now let us go ahead and create a hard link also to file1. So, the hard links are created by with the link command without the -s option and now look at the inode numbers in the long listing and you will see that the file1 and the file3 have the same inode number which means that these two are basically the same file and you would also notice they are a number of hard links here it says two.

So, which means that these files are having 2 hard links in the file system whereas a link file2 has only one hard link which is the symbolic link itself so, this is the difference and you can actually note from this number here how many hard links and from there we can actually guess that there are multiple locations for the particular file.
(Refer Slide Time: 31:44)



File sizes

- `ls -s`
- `stat`
- `du`
- Role of block size

IIT Madras
BSc Degree

Now a brief note about the file sizes. So, we could actually get the file size from the long listing of the directory. We could also ask for the file size to be specified in the very first column using the -s option of the ls command there are also some more commands available to know more about the size occupied by the file on the disk using the stat command as well as by the du command and there is a role that the block size will play to understand that we will actually do both of the demo.

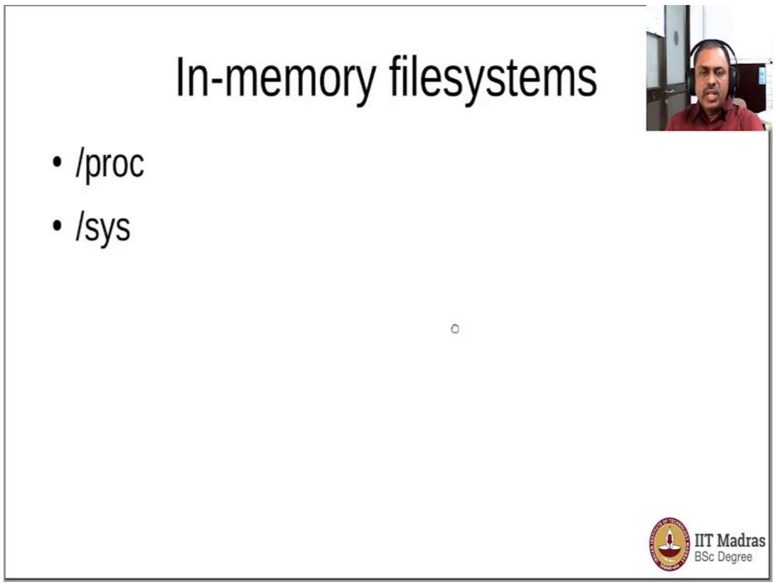
So, let us go and look at these sizes from one of the system folders. So, we will go to the user bin and type a ls there are amply number of files. So, we can inspect the sizes of these files here. So, there is a file called z new. So, let us look at the size of that. So, stat znew and you would see that it is telling you about the size and the size it says 4553 bites and how many blocks it is occupying and so on.

So, that new is occupying 4553 bites and it is actually occupying sixteen blocks and you have the time stamps available here which will tell you when was it accessed when was it modified and so on. Now look at the same information using the du command okay and it says that together 8 it is occupying about 8 kilobytes. So, we can actually use the -h command and it will tell you that you know in a human readable form it is about 8 kilobytes.

Now why is it 8 kilobytes as for the du output is concerned but the ls is showing you 4553? The reason is that the block size is 4 kilobytes and the size is just above 4 kilobytes. So, it actually goes on to the 8 kilobytes. It means that the files that are actually smaller than the block size would actually take up the whole block. So, you can have a block or more blocks but not a part of the block.

Now let us look at the zmore command which seems to be just less than one block okay and you understand that it would actually take the whole block okay and here you see. So, you could see that zmore a command which is occupying 4 kilograms and it would actually have the size which is actually less than one block only 1842 bytes but it is actually occupy 4 kilobytes because that is the size of the block.

(Refer Slide Time: 34:33)



In-memory filesystems

- /proc
- /sys

IIT Madras
BSc Degree

Now there are some directories which you would see in the root folder which are actually not sitting on the disk they are only in the memory and they are very special file systems. We should not be messing around with those territories but we can actually view a very important system information from those directories in a read only manner and let us explore those two directors now.

So, we will go back to the root folder and see that these folders are available here you can see here this sys and then proc okay. So, what are these folders? These folders are actually not sitting on your hard disk proc and sys they are actually in the memory and these are all basically data structures available for the users to know more about their system and the proc system the slash proc file system is an older system from kernel 2.6 onwards the slash system has been used.

However even now the information about the various processes that are running is still stored in these slash proc directly itself. We can go in and look at that without much problem. Let us go there cd slash proc and you can see that it has many files which are actually zero size and that would be a little bit misleading the reason is that some of them actually have content that we can read but the size will be zero the reason is that it is only a representation for us to explore but actually they are not real files which are sitting on the hard disk.

So, let us look at some of the contents. So, there is a file called CPU info. So, which is actually occupying zero but it says but let us look at its content less CPU info and you see that there is some information that is coming out onto the screen and it seems to be information about the CPU okay. And we could also do cat CPU info and you would see that several lines have come and you can scroll with your mouse and read some of it.

So, it appears that it is giving you information about your CPU and when you type ls -l you would see this file is actually showing as if it is occupying zero bytes. So, which means that this is not real it is only in the memory and as a place holder for us to know about the system and you can actually now know more about your system by reading some other files also.

So, if you type cat version within the slash book directory you would see the information which is also available when you type uname -a which is basically telling you about the kernel that is being used by the operator system along with what kind of an architecture the particular mission is and what is the operator system it is running and so on. When you type meminfo okay cat meminfo will give you information about the memory and this information is coming in various forms how much of free memory is there and how much of the total memory is there etcetera.

And this is also available to us through other commands like for example free okay. So, it is more readable free -h will actually give you in a human readable form how much of memory is available for more applications to be launched on this particular operating system. And you can also do cat partitions I need to tell you how many partitions are there in this particular system if the hardest has multiple partitions it will show you and it will also show you the size of it in terms of number of blocks.

But this is not very helpful you could actually get the help using the df command and the df command will actually show you more information like what is a mount point in which directory is that particular partition made available to the user and so on. And if you type df -h it will give you the information in a human readable form with respect to the size. So, for example my root partition happens to be 183 gigabytes and of which only 30 gigabytes has been used.

And my home directory is having a size of 481 gigabytes out of it about 6.9 gigabytes has been used. Now if you go to slash proc and look at the files there is some file which actually looks rather big you know you can see this there is a k core file which occupies a huge size and it appears that this is a very big number. If you calculate it would be how many zeros if you calculate it would come out to be about 140 terabytes.

Obviously it cannot be a real file because our hard disk itself just one terabyte and therefore this cannot be coming from the hard disk. So, this basically is a an entry that tells you what is a maximum virtual memory the current Linux operating system is able to handle okay theoretically and this comes with 47 bits for the virtual memory and two respiratory comes to about 140 terabytes.

So, this directory is very magical because there are files which are having zero size but have content. There are files which have a huge size but actually do not exist on the. So, slash prop that way is a magical file system in which a lot of information is available for you to look at to know more about system. There are a lot of folders that are also going down with some names which are basically numbers these numbers correspond to what record the process Id’s.

For every process that is running in the operating system there is an id and using these folders. So, we can know more about those processes. Similarly the slash sys folder is also the same kind of a file system which is actually a much more well organized than slash block because it says like a newer development in the Linux and here in a very neatly organized directory structure you would see that the information about the system is all collected and available for you to read and process in your scripts or any other use that you may want to have.

So, let us go and explore some directory here. So, I will go to slash sys and then go to the bus folder and in the bath I will go to usb and then I will go to devices and you see that there are some there are some directories links that are available. So, these are all devices that are actually usb type of devices. Now let me go to the one dash one directories and there are some files that are here and I can just read about them.

So, like I say less manufacturer okay and it says JBL is a manufacturer okay and the less product and it says JBL Quantum 300 is a product map. So, instantly this particular usb device that we are looking at is this headphone that I am using right. Now to record my audio. So, the usb device is JBL Quantum 300. Similarly other devices also can be looked at. So, let us look at let us one dash and less manufacturer it is a pixel art is a manufacturer less product it says dell M116 usb optical mouse that is this mouse which is actually the usb device that we have looked at.

So, like this you know every device that is connected with the computer we can actually know more and more details about it just by simply exploring the file system that is already made available. So, slash sys and slash proc or two file systems which help us to know more about the hardware and we will learn to process that information as we go along. So I now encourage you to explore the file system and run the simple commands that we have discussed till now to understand your machine a little bit more.
(Video End: 42:43)