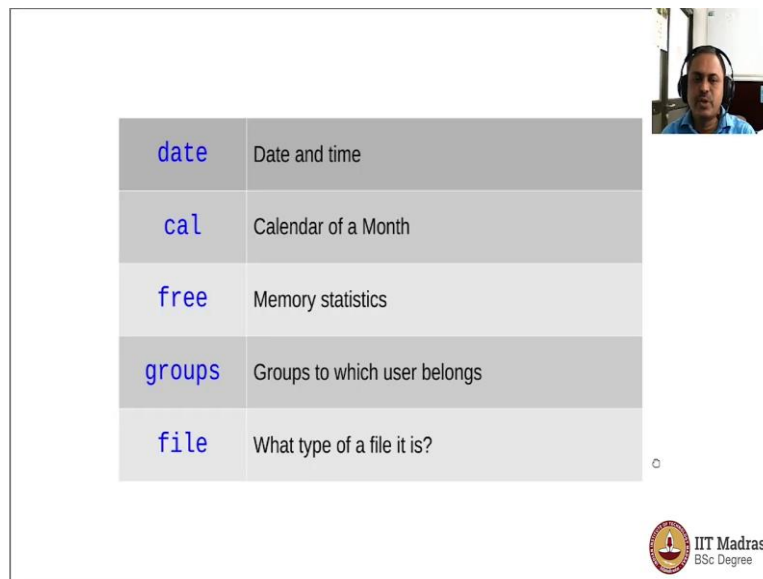


System Commands
Online Degree Programme
B. Sc in Programming and Data Science
Diploma Level
Prof. Gandham Phanikumar
Department of Materials Engineering
Indian Institute of Technology- Madras


Lecture 3
Simple Commands in Linux - 1

Let us look at some simple commands. So, that we can use in the Linux command line environment, we will make ourselves familiar with the command line environment by trying out some simple commands and these are not going to change the system in any manner and therefore you can try this out without anybody.

(Refer Slide Time: 00:31)



<code>date</code>	Date and time
<code>cal</code>	Calendar of a Month
<code>free</code>	Memory statistics
<code>groups</code>	Groups to which user belongs
<code>file</code>	What type of a file it is?



The commands that we are going to look at are the following to know the date and time at this moment. The calendar of this month some statistics about the memory that is available which groups does the current user belong to in the directory listing that we are seeing in our home directory what type of file each entry corresponds to.

(Video Start: 00:51)

Now let us explore some of these commands by using the command line environment. So, we are using the terminal emulator called terminal in Ubuntu and the batch shell is what is actually responding to our commands. So, where are we first we are in the home

directory will verify that we are in the home directory and we could then change the directory using the shortcuts. So, we use the shortcut double dot to go to the parent directory and we can do that once more to go to the root folder.

And then we can go down the hierarchy let us say to user and then to bin directory. Now from here we can go back to the home directory by simply pressing `cd` and then enter and let us say we want to go and visit the directory again we do not have to type `user bin` we can just say `cd` and then a hyphen to indicate the previous directory we were in and we will be placed in the previous directory that we were working in.

Sometimes if the directory names are very long this could be quite useful. And when we proceed we come to the home directory and as you would notice if you press `cd` space dot then nothing will happen because that is like changing the directly to the current directory which does not change anything. Now wherever we are let us say we go to `user bin` we could go to the home directory in multiple ways.

There is a shortcut tilde which also will take you to the home directory. So, you verify that you are in the home directory. So, there are some characters that are actually indicating that there are aliases for certain directory entries for example dot is for the current directory double dot is for the parent directory and the tilde is for the home directory and dash is for the directory that we were in the previous command.

So, I press now control l and clear the screen. Now we will have some useful commands to look at. Let us look at a command called `date` it tells what is a current time stamp for this moment. So, if you run it again and again you will see that the second position number is changing. Now obviously this time need not be accurate because it depends upon what you have configured and you can also change it.

We will learn how to do that at some point later. We could also have the output of the `date` command in multiple manners. So, if you give it in with a comment option `-capital R` it would give you the command output in the format called `rfc 5322` standard. This is a standard that is used for email communications and if you want you could also display the output in many other formats.

How do we find out what are the various ways? So, we take the man manual page of a data command and look at that and here you have some help to tell you which fields correspond to which part of the daytime stamp and then you can configure that to print the date in any matter that you would like. So, I come out of this help by pressing q. Now you could also have more information I would clear the screen and you could actually just type cal and you would see the calendar for the current month.

You could also use this command to find out the calendar of any month. Let us say for the august month of 1947 let us look at the calendar and it shows that we began our independent India on Friday. And you could also give the month in other formats. So, it would give you with the same output you could also perhaps use a capital letter for the August and that would also still work.

So, this is a pretty quick way of checking the calendar of any month of any year and you may want to find out which day of the week we were born in using this command. Now you could also display the calendar in other formats. So, you have the direction change now ncal would be a command which would give you calendar in a different orientation which may be more useful for you depending upon the way you are used to looking at the calendar for maths.

And the options would still work the same way ncal and then let us say august 1947 even if it would show you the calendar for that month in the different display orientation. There are some more commands that you would be interested to run to just to check the statistics about the machine and that there is a command called free and this shows what is the amount of memory that is available for free in this computer right now and you can see that these numbers are not very readable.

So, we would actually type free and then hyphen hedge as an option to make it human readable and you would see that there is uh that the total memory of this particular computer is 31 gigabytes and out of that only 1.2 gigabytes is used 26 gigabytes is free. So, we are now not running very much. So, most of the memory is free. And there is something called swap memory which is basically used whenever the physical memory is exceeded by the number of programs that are running.

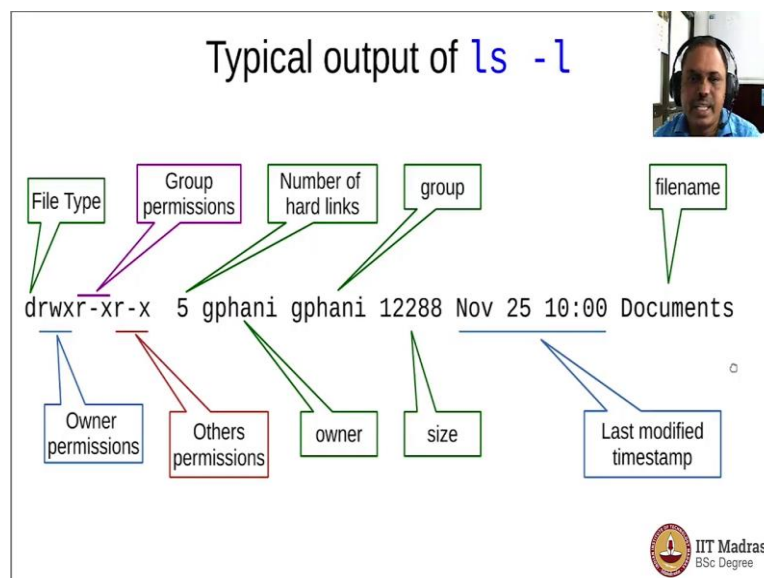
And one can also look at whether that is being occupied and you see that we are not occupying anything in this swap memory which is a good thing because most of the usage if it is in the physical memory then it will be fast and if there is any requirement to use a swap memory then it would be drastically slow because swap memory is actually stored on the disk and the disk is much slower than the actual physical memory also referred to as random access memory.

There is a command called `groups` if you run you would see what are the groups to which you belong to, the current user is belonging to. So, this current user is belonging to a group called `sudo` which means that the current user is also a super user. And this can be changed by adding ourselves to more groups. So, there is a group by the same name as a user which is also to ensure that there is a privacy with the certain files that nobody else can actually use. In this computer there are not many users and therefore there are not many groups either.

(Video End: 07:36)

Now let us look at the long listing of the directory structure. So, we have learned it earlier and if you put uh `-l` option to the `ls` command then you will see the long listing.

(Refer Slide Time: 07:49)



So, here is the typical output of the `ls -l` command and you would see that there is this output which is given for as an example. So, the very first character tells you what type of file it is. So, the `d` will say that it is a directory that we are talking about. And then

there are nine characters which would be rwx with any of the three positions replaced with the hyphen in case that particular permission is not available.

The first three characters refer to the permissions of the owner. The second set of three characters are permissions of the group the third set of characters are for the others that is any other user in the particular Linux system whoever is logged on can be considered as part of the others group. And then the number after that corresponds to the number of hard links what do we mean by a hard link?

We will come to in a moment and then there are two names that are given here the first name correspond to the owner. So, who is owning this particular entry documents that owner is displayed here the second entry is for the group. So, sometimes you may want to have the ownership of any folder to be a group to which your part. So, that a set of users can collaborate to work on a particular directory together.

So, you should understand that Unix or Linux come from a time when uh multiple users are locked down to a mainframe computer and are sharing files and working hard collaboratively and therefore the file system has these features built in already. By default there is always a group by the same name as the user and therefore such folders are basically private to particular user.

Then this number here corresponds to the size in terms of bytes how much size is occupied by that particular folder. And then come the time stamp. So, when was the last time it was modified this directory was modified that is given as a time stamp and in the case the year is different the year also would be appearing here. And then lastly the name of the file is given. So, we will now dissect and look at each of these possibilities in detail.

(Refer Slide Time: 10:06)

File Types	
-	Regular file
d	Directory
l	Symbolic link
c	Character file
b	Block file
s	Socket file
p	Named pipe

So, in the very first character if you see a hyphen it means that it is a regular file. Regular file as in it could be a script it could be a text file it could be an executable and so on. So, what we normally refer to as a file is what it is if there is a hyphen in the very first entry of the long listing of the directory structure. Now if the first character happens to be d it means it is a directory.

So, under that there will be other files or other territories usually most of our work in the Linux command line environment would be restricted to only these two types of files most of the time. Now the third character that you would be alerted is about the l which is a symbolic link we have already seen that once, c indicate that it is a character file and b indicates is a block file.

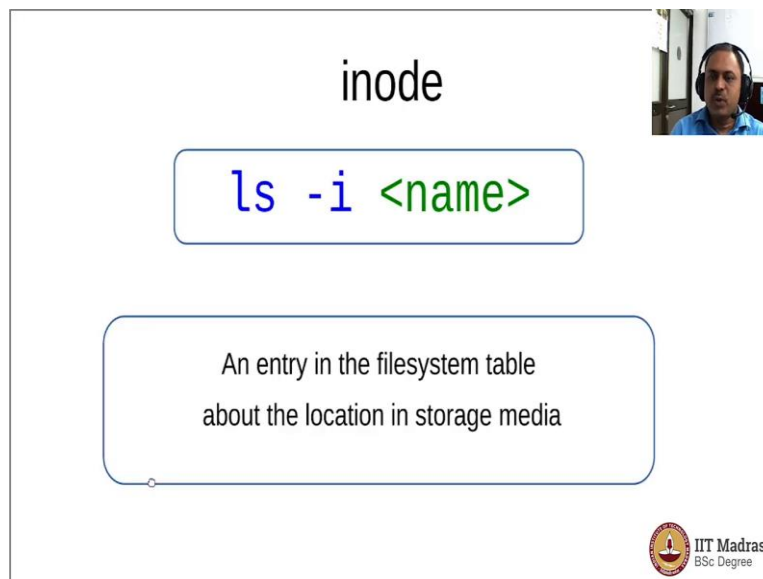
So, block file and character files are typically devices. So, the character file would be the terminal typically and the block file would be the hard disk and we normally do not mess around with those files unless we know what we are doing. If the first character happens to be s that means it is a socket file that means it is actually a process that is actually running behind you can have two way communication with that particular file which means that a network device would open a socket and listen to the incoming connection.

And we will also be able to send out the data a two way communication is possible and it would be actually a process that is doing this activity behind it is not actually a static file and this is indicated by the character s in the first position of the long listing of the directory structure. If the first character happens to be p it means it is a named pipe that

means there is a process behind which is acting to communicate and it would be a one way communication and you could actually see that you cannot send the data but you could actually read the data that is output by that particular process.

So, whenever you see any direct listing if the first character happens to be a hyphen or a d that means it is in a familiar directory as what we are normally used to. If the first character happens to be c or b or s or p then you should be careful with such files because you should know what you are doing when you try to modify those files.

(Refer Slide Time: 12:26)



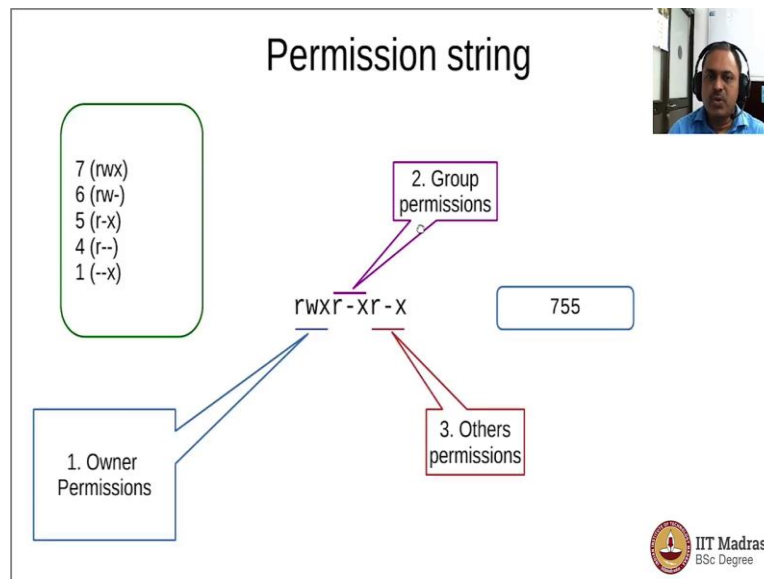
The slide is titled "inode". It features a central box containing the command `ls -i <name>`, where `ls` is blue, `-i` is blue, and `<name>` is green. Below this, another box contains the text: "An entry in the filesystem table about the location in storage media". In the top right corner, there is a small video feed of a man wearing a headset. In the bottom right corner, there is a logo for "IIT Madras BSc Degree".

Now to understand what is a hard link we need to know what is an inode. So, you can actually find out the inode numbers for every file entry by using the option -i when we are running the ls commander normally in Linux and Unix environments whenever we need to give as an argument that argument is listed in the text as this manner with the angular brackets and some entry there.

So, it is an indication that you need to provide something there as an input it is not as if you have to actually use angular brackets it is just a placeholder for the argument that you are going to give. So, I will illustrate that with the comment line just a moment. So, inode it is actually an entry in the file system table about the location in the storage medium and it will be unique for every unique file.

So, if there are multiple entries in the file system that has the same ignored it means that they are on hard links.

(Refer Slide Time: 13:24)



Now a little bit details about the permission string also before we go to the comment line. So, the permissions can be changed but the way we understand the permission strings are the first character r indicates that you have the read permission the second character if it is w it means that you can write to that particular file and x indicates that it can have executable permission.

And if the file happens to be a directory then x would mean that you can change to the directory ok and the first three characters are meant for the owner the second set of three characters are meant for the group the third set of three characters are meant for the others. Now there is a numerical way that you can also represent this particular permission string and here I am writing it as 755.

Why I am doing that is because the three characters can be represented as on and off switches in binary format. So, you would have one here two here and then a four here. So, if you take four it means you are giving read permission if you are writing five that means you are giving read and executable permission and if you are giving seven you are giving all three permissions.

So, the details are here typical permissions that you set are indicated by these numbers here from 1 to 7 and the permission string that is written here is indicated by 755. We would use this a numerical value when we are changing the permission.

(Video Start: 14:48)

Let us try to explore the permissions how to change them and the ability to create files delete them moving them around, organizing them into folder etcetera in this brief exercise. So, we opened the terminal emulator and the batch shell is running responding to our commands. So, where are we now we are in the home directly. Now when I press `ls -l` we see the long format and we see that there are a bunch of folders here.

What permissions do I have for my home directory? To know that I need to go one level up and look at the permissions. So I go `cd ..` and then type `ls -l` and we see that for the folder which is named after my user name the permission string says `rwx` for the owner the owner is same as my user ID and the `w` says that it is having right permission which means that I can create files or folders within my home directory which is expected.

Also you would notice that the rest of the permissions for the group or for others is all empty which means that I have not given permissions for any other user to do anything in my home directory which is also a standard practice. So, I now go back to my home directory with the `cd` and clear the screen with `control l` and let us try to create some files to see how we are able to use this permission to organize files.

So, let me make a directory `mkdir` and then let us say level one. So, I am creating a directory called level one let me check what I have done just now. So, you see that there is a new folder here level one and it is actually a directory and there are some default permissions that have been given okay it has some permissions that are given for the group and for others.

So, that they are able to actually read and change over to the directory are and `x` permission are given. So, this depends upon what is called a `umask` which we will come to later. Now I want to change these permissions. So, that they look like rest of the folders also. So, what I would do is first I would like to knock off the write permissions for the group. So, I would use `ch mode` and then a group `-w` which for the group remove the right permission and for which file for the folder called level one.

So, I can see whether the group permissions have been changed. So, let us look at the directory again and you can now see the difference here. So, there is a w here in the second set of permission string for the level one folder and here you could see that that permission has been removed using this command `ch mode group -w g -w level 1`. Now let us remove the executable permission also and for that what I would do is `ch mode` and `ch mode g -x level 1`.

And I now need to look at the output once more long listing of the files in this folder and you could now see that for the group the executable permission has also been removed for the folder calls level 1. Now we can actually change the permissions by using this string where `ch mode` and then the first string would be u if it is for the user or g if it is for the group or o if it is for the others.

And then if you use plus that means give the permission to use minus then remove the permission after that the permission can be `r r r r w r x` whichever it is okay now what I am doing is that others I am removing the executable permission for the folder level 1 let us say and let us look at the output and you would see that here for others executable permission was originally there but it is now removed.

Now you can actually do this one permission at a time but as we have seen in the slides you could actually you can give a numerical value as an option to the `ch mode` command with the argument which is the folder for which you have changed the permissions in such a way that in one go you can make the permissions appear as you would like. So, let us say I want to give the permission such that read, write and executable are there for the owner and no permission is there for others are for the group.

Then I would write as `ch mode 700` then level 1 okay and let us look at the output now and you would see that all the folders are looking with the same permission and here you can see for the folder level 1 I have removed permissions. So, we know now we have created a folder called level one using `make dir` and then we have adjusted the permissions as per our vision.

Now we can go into the folder and see what is there, `cd level one` and you can see that it is empty because we have not put anything else in that folder. Now let us create a file an

empty file using the touch command. Touch command is used actually to change the time stamp of last modified uh information about that particular file or folder. And in case the file does not exist then it will be created and that is one way by which you can create empty files.

So, using the touch command we will create. So, let us say file one okay touch file one and let us look at the output now you would see that there is a file there and it has just been created and the name of the file one. And there are some default permission that were also given. Now we will make the default permissions again change according to the ch mode. So, let us say ch mode and the letters say 700 file one and you will see that the permissions have been modified.

So, like this we can actually adjust the permissions of any file or folder. We have seen that we have done it for the folder called level one and for a file called file one. We can actually make a copy of this file using the cp command. Copy the file to a new name let us say new name is file2 and now let us look at it. So, there are two files of course both are empty but we have made a copy of the first file using the command the cp and then old file name followed by the new file name the three fields are separated by a blank space between them.

Now you want to move this around. Let us say you want to move the file to one level up to the home directory. So, you could just say mv that is to move file to double dot double dot is a short name for the parent directory. So, if you do this, what happens is the file2 is going to go into the home directory. Let us check if it has been done. So, first of all the file is not available in this particular folder.

Let us go to one level above and the check and sure enough the file has been found in the home directory because we have moved it one level up from the folder called level one. Now you apart from making a copy you can also use the mv command to move the files around and also you can mv command can also be used to rename files. So, let us rename the file2 as something else you can give a new name here.

So, we can save file2a and let us look at the listing and you can see that the file name has been changed from file2 to file2a. So, you can change it whichever way you like. Now

sometimes you may want to have a space between two different words as a part of the file name in which case you have to use the quotation symbol and let us also do that here. So, I will create a new file here called let us say file1 and then I change the permission and then you can see that the permissions have been changed.

And now I want to rename it to do something else. So, I will use mv command file1 and I want to rename it such that there is a space between two words say file 1. Now let us see how it looks like you can see that the file has been renamed and there is a quotation symbol to indicate that these two are separate words of the same file name and there is a space between them.

So, when we want to now change permissions we cannot use this file name without using the quotation symbols because then it will have a mistake if we use it without the codes. Because it would think that the space between the two words is actually a field separator for two different files how is it let us just check this out. So, let us say I want to change the permission ch mode and let us say 700 the same permission and let us say file1.

And you see that it is complaining say that there are no such files however when we try to do this way by giving a quotation symbol then it works. And whenever a Linux command works fine it actually does not give you any output on the screen because the philosophy of Unix which has been passed on to Linux ease of nonsense namely do not throw any output if everything worked fine.

Now let us say we want to remove this file. So, there is a command called rm which we can remove the files rm is a file okay rm file1 and it is asking me whether I should remove it or not and I say yes okay. Now normally if you use it very first time then it does not ask because the default behaviour is not to ask. I have made an alias for the remove command where I pass on the option -i okay. So, let us look at the help for the command if you see here -i option is there -i option is there.

So, that for every removal there is a prompting okay. So, it is a problem if you have a lot of files to be removed if it keeps prompting but for new bees it is a good idea to let it be there. So, that you are careful. Now how do you set the alias okay. So, that is done by

using a command called alias `alias rm = rm -i` okay. Now the alias actually is to make it ask every time.

If you use this command then it will ask every time okay. So, you can try it out. So, that you can check whether it prompts you when you want to remove files always tried to do these commands on empty files that you have created using the touch command. So, that there is no loss of data if you tried it on any other files that was useful for you. Now let us go forward to explore more features.

So, let me clear the screen now. Now the file2a is in the home directory. Now let us say I want to move it back to the level one. So, we could do that by using the mv command `mv file2a /level one` and you see that it is disappeared from the home directory and I change the directory to level one and I see the output and it is there in the level one folder one level below.

So, we have never been able to create empty files and move them around copy them and also delete them if you do not need them. Now one concept called hardlink is something that we need to get familiarized just to understand what is this number that is coming here in this string? So, that is actually the number of hardlinks and we mentioned about the inode number in the context of it and we can actually look at the inode numbers to understand what is that.

So, look at the folder called level one it has two hard links. So, what are those two hard links? So, to understand that we should get the inode number. So, I would do that here `ls -li` and then you can see that there are some additional numbers that are printed in front of the long listing of the file system here and for level one if you see here this is the inode number. This is inode number we will note it down and keep it safe to compare.

Now let us go into the level one directory okay and let us look at the long listing with the inode number but also with every file that is there including the dot and double dot okay. So, `-a` option is to have dot and double dot. Now you can combine options with one single hyphen that is quite acceptable okay. Now you see that the dot which is there which is the current directory has this is the same inode number.

You can see here same inode number as level one here. A location for the dictionary level one has the inode number which is highlighted here okay and also here and it has two interests in the file system one for the directory called level one which is in the home directory and then there is another directory called dot which is under the level one. So, there are basically two hard links.

Why are we saying that there are two hard links because they are basically having the same inode number and there is a reason why when you look at this number here it says two okay. Now what would happen when we look at the subdirectories this number will keep increasing as the number of subdirectories get created. So, let us look at that now. So, in the level one I make a directory called level two.

Now I look at the same option and I see that the number of hard links has increased to three. So, where is the third hard link? Well you can understand that when you go inside the level two and look at that the parent directory from there is the same inode number as the level one. So, the third inode is basically here the double dot under the subdirectory. So, as the number of subdirectories increases the number of hard links also keeps increasing.

And one needs to basically look at this number to understand how many different entries are there in the file system. Now most of the simple files regular files that are having a dash in the front of the long listing of the file we actually have only one hard link which means that this is the only file and if you delete this that means that particular file has disappeared.

Now if there are more hard links it means that if you delete one such entry there are other entries which may still be existing and for regular files such a possibility is very much there and we will come to the use of that in backup systems at a later point of time. For now let us now go back to the home directory and as we have discussed earlier tilde is a alias for the home director I can just press tilde and go to the home directory. So, where am I? I am in the home directory.

I can actually now go back to the same directory with a dash and we will go back to the level two directory again. So, the dash is a alias for the previous directory that we were

using and tilde is alias for the home directory that we started from. And now I clear the screen with control l. So, till now we have been seeing where we have permissions namely in the home directory and where we were able to create files and also do some reorganization of files.

Now let us see whether we get errors when we try to do that in a folder where we are not supposed to try that out. So, let us go to the system folder do not be afraid you can just try it out and then see how the system is actually preventing you from doing any damage to the operated system inadvertently. So, let me go to the root folder where am I? I am in the root folder.

And you see that this particular folder everything is owned by root you can see that it is everything is owned by a root. Now who am I am actually a different user I am not the root user who am I? You can see that the user name is printed there. So, the user name is not matching the owner here and therefore this folder for example I cannot do much. So, let us try to create an empty file in this folder by let us say you know file1 and you see that there is an error that comes permission denied because we do not have write permission.

And how do I know that ls -la and for the current directory which is root directory there are no write permissions for others and my user name is not matching the root therefore I am other user and therefore I do not have a right permission. So, I cannot create a file in this particular folder. So, that is how the denial happens for the creation of files. Now even for accessing the folders also there could be permission denials and we will explore that in the var folder.

So, let us go to the var folder and then to the log folder and this is a place where the access logs system logs etc are all kept. So, here you see that most of the files are owned by root are syslog process and in a different group etcetera. So, here there are many files which are actually not owned there are many files which do not have a read permission for others and in this case we are other users who we do not have read permission.

So, syslog we do not have read permission let us scroll up and see with the mouse button I scroll up and see whether I have a root read permission in any other files. So, I have a

file here alternatives dot log in which the others will have a read permission. So, let us go ahead and see whether we can read it okay. So, what we do is there is a command called less which actually allows you to read a text file page by page and at the end of the content you can actually come out of it by pressing q.

And we are using bash. So, we could actually type a few characters of a file name and press tab and it will fill the file name for you. So, you do not have to type the entire file name accurately. Now I just press here and you can see some output that is coming onto the screen from the file alternatives dot log and which means that I am able to read this file and I have come to the bottom of the page and it shows here end and I pressed q to come out of it.

So, I have a read permission and therefore I am able to read this file using less command. Now let me see if I can change the permission with the write permission. So, let us see let us ch mode others I want to give write permission for this file. I am another user. So, I want to give a write permission and you see that it is denied. Why is it denied because when we go and scroll up and see it here it is owned by root okay only root can change the permissions of files owned by root.

So, anybody can change permissions of a file only when their owners of that particular file. So, here the ownership is given to the root and that is one of the safety mechanisms by which an ordinary user cannot actually change permissions to do any modifications to that particular file however read permissions are given. Now there are some files for which read permissions are not given.

So, let us see if we can actually read them okay. So, let us say less and ubuntu advantage dot log okay I want to read this file here you see this okay there is a log file which the permissions are not available and I want to try and read it and you can see that it does not get through because the breed permissions are not there. Now this is a folder what are the kind of permissions in this folder let us see.

The folder itself the current directory does not have any right permission for others also. So, which means that I cannot create any folder here either make dir level one failed ok I cannot create folders there because this folder is owned by root. Touch file1 I cannot

create an empty file here because the folder does not have write permissions for others. So, you can see that there is a mechanism in place where we cannot actually create files or change permissions or create directories in locations where they are belonging to the system and the operating system.

And we cannot actually override those settings unless we are super users So, how to override it and doing something knowingly and carefully what is required for system administration is something that we will come to know as we go along in this course. For now I want to also show you one more permission that can be denied. If you see here this particular folder unattended upgrades unattended upgrades you can see that there is no x permission given for others.

Which means that I cannot even change directory to that particular folder okay I cannot even cd to that. So, let me try that cd unattended and I just feel that with a tab character and try it out and says permission then I cannot even enter the directory. So, this is how the upper system has secured itself from ordinary users. So, that there are folders which you cannot enter.

There are folders which you cannot write to and there are files that you cannot read and whatever is available to you with read permission that is all that you can play with by reading but there is nothing more that you can do. So, naturally you do not have to worry much as a novice user when you are running around with the commands simple commands like ls or cd or cp and such things you really cannot do too much of any damage to operating system.

And you can go ahead and explore the file system hierarchy and try to read text files wherever you find them. Now how do I know which file is a text file and which file is actually a binary file. So, for that let me go to the user bin okay and there are a lot of files here and I just pick up the last one and just to see what is that file. So, there is a command called file where it will tell you what kind of a file that is.

So, in user bin directory the file called znew happens to be a ASCII text and it is an executable string executable script okay. So, ASCII text means so that I can read it on the screen. So, let me try that out and you see that I am able to read it ok it just opens

and shows you the script. So, we will understand the script by the end of this course. So, do not worry for now we are able to read it.

Now let us look at another such executable whether it is a script or is it a binary file let us look at that. Now it appears that the last file zoom is a symbolic link to something else okay. Now let us go there and see what it is. So, zoom launcher happens to be here and you see that it now gives a different kind of an output it says it is an ELF 64 bit okay LSB shared object for the architecture X86-64 and so on.

This actually means that this is a binary file okay we cannot read it. So, let us try if it is possible and what happens and you see that it is alerting you saying that this is a binary file. So, do you want to really see it? We will say no because it will actually appear in a garbled manner on a text screen and therefore we will not attempt that. So, you can see that you can actually find out what type of a file it is by using the file command and try it also on the directory.

File zoom and it says it is a directory go one level up file opt it says it is a directory come to home directory try it on come to home there try it on level one and it says it is a directory. Go to that level one try it on file1 and it says it is an empty file. So, you can see that you can actually find out what type of file it is and whether it is a directory or a regular file is a text file or executable binary etc using the file command.

And you can also explore what is there in the system for you to read using these commands. Now let us spend a few minutes about the time stamp also. So, the time stamps at this moment is given by the comment date and if you look at the directories some of them are actually quite old because they have been modified long back. So, if you look at the desktop folder for example it is modified on July 28th it says.

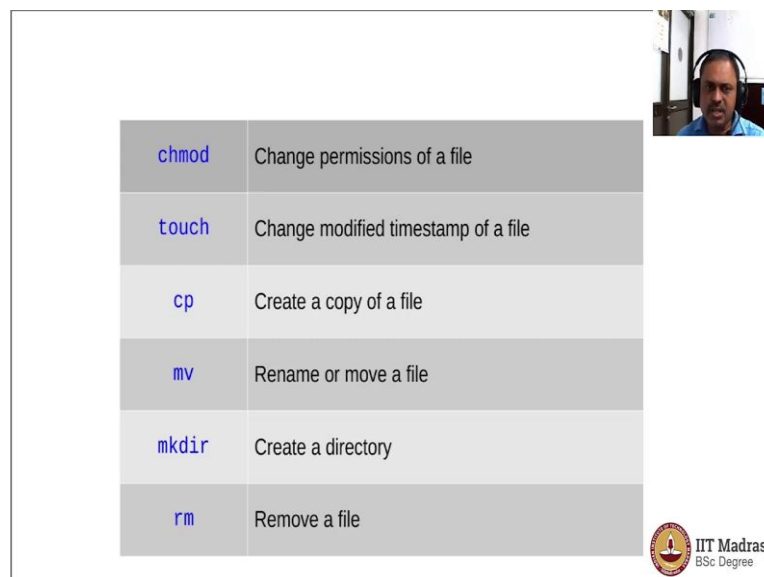
Now I want to change the time stamp. So, I could do the one thing I can just use touch command and then I can run `ls -l` again and you now see that that time stamp has changed you can see here that time stamp is 25th November 1650 and before the touch it is actually July 28th 2020. So, what it implies is that this file desktop folder was having a time stamp which was old and the command touch desktop has changed the last modified time stamp to the current time stamp.

So, that is how they time stamp has been changed. So, the command attached can actually make some of the files look like they are new or newly edited. And if you give a file name which does not exist then the touch command will actually create that file as an empty file and show it to you. So, let us try that out. So, you can see that I have just touched a file called xaa and there is no such file.

And what it does is it does not complain it just goes on to create an empty file with the latest time stamp and then shows that as a file in the list of files and then we have it as a part of the files in the home directory.

(Video End: 43:19)

(Refer Slide Time: 43:20)



<code>chmod</code>	Change permissions of a file
<code>touch</code>	Change modified timestamp of a file
<code>cp</code>	Create a copy of a file
<code>mv</code>	Rename or move a file
<code>mkdir</code>	Create a directory
<code>rm</code>	Remove a file

So, we have seen in this session some important commands which are very useful for a day to day operation such as ch mode to change the permissions of a file of course ls -l shows you what are those permissions and then ch mode allows you to change those and touch command is there to create empty files or to change the timestamp of any file, cp is to create a copy of a file, mv is to rename or relocate a file, mkdir is to create directories and rm is used to remove a file.