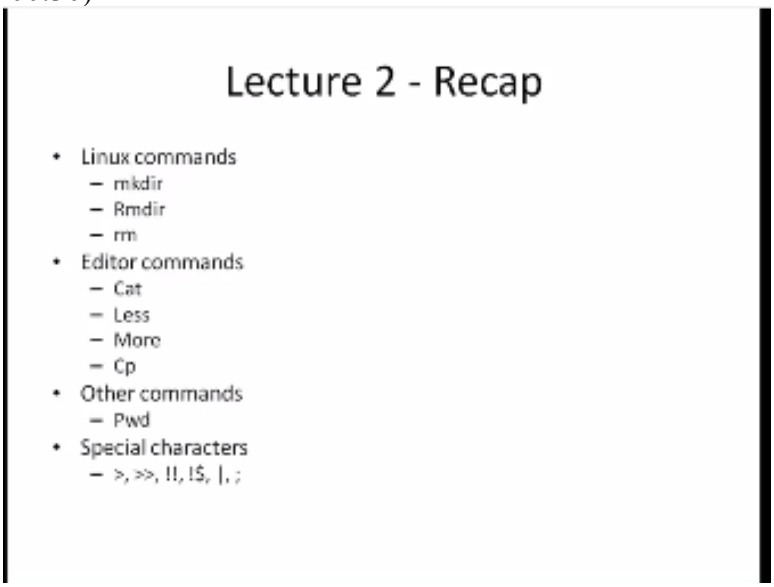
SEER AKADEMI
Linux Basics – Lecture 3

The course on the Linux and programming languages today we will be continuing our discussion on the Linux basics today is the lecture 3 of 4 part series so let us before we start of a discussion for today let us review what we did last time.

(Refer Slide Time: 00:30)



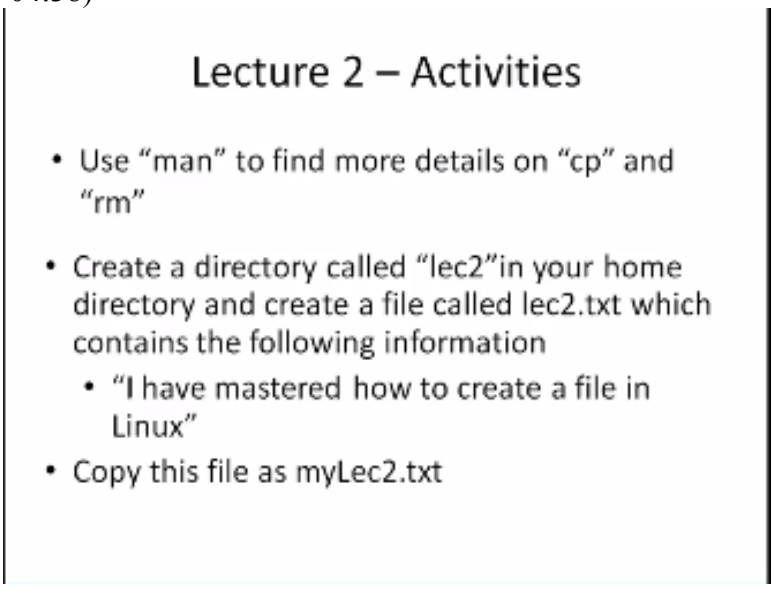
So just a recap the lecture 1 we talked about the basics of Linux we then to like them of the file system the kernel the shell in the last lecture we started talking about some commands specifically We covered several command for me to make there to create the directory are RMDR to remove a directory then RM to remove a file we also saw the Associated options and arguments and we also saw some dangerous commands that we do not want to really use it.

Unless we know exactly what we are trying to do for example the RM - RF which is a recursive and forced delete if you do an RM- R * or * that * and if we do it from your from the root directory we will flash the boom or pretty much like everything can be wiped out and then Usually in Linux so the once the files are removed its permanently erased so it is very difficult to get back.

So we saw that and then we also saw some editor command like CAT or at the ordered how to actually edit the file and then you will be also like look at the commands like less and more then we also did the CP command copy a copying file from one location to another we also like saw the variations of these copying commands so that we copy directly over or just a file of what are the arguments for the things like moves.

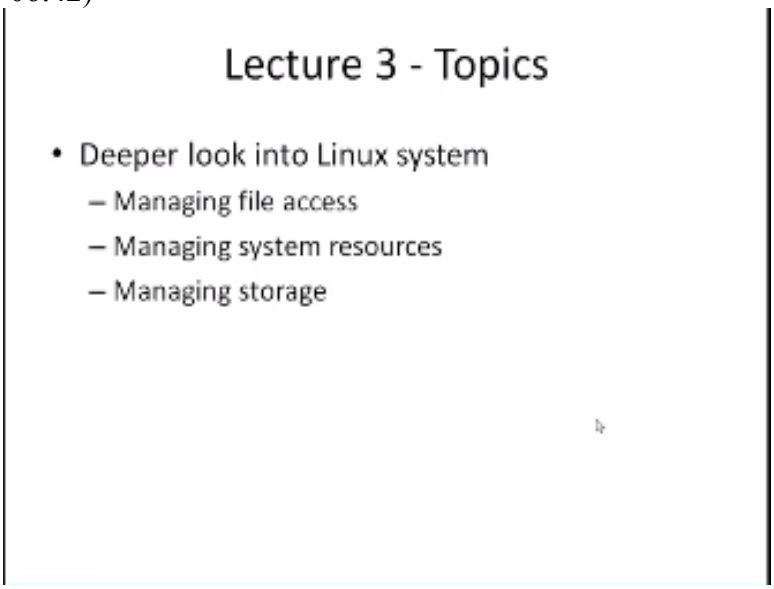
We also saw PWD or the present working directory to kind of like the in a map it says you are here similarly like I may need to do PWD to tell you are here there exactly we also saw some

usage of special characters like the >the >> !!, !$, in this type command I do not know whether B is that they have not we will do it for next section and then also ; so you know that the > the single > symbol of command to actually add it into a file.
So we could do like a CAT in > in a file name to write into a CAT into a file then open called even then with all the fat for that whatever we write on the terminal we go into the that particular file name the >> is more open to the file !! Is the exit to the previous command in general like! Denotes that it is a command needs be executed we will see this! In another other places too like I mean we will talk about that and then we will see.
How we can do it and then also we saw this ; or that is basically a command separator so in a line you can have multiple comma separated by semicolons and or command interpreter will understand that has a different command from to the serious action so pretty much now we know how to execute implement it in serial fashion also the various useful various special characters and except that.
(Refer Slide Time: 04:38)



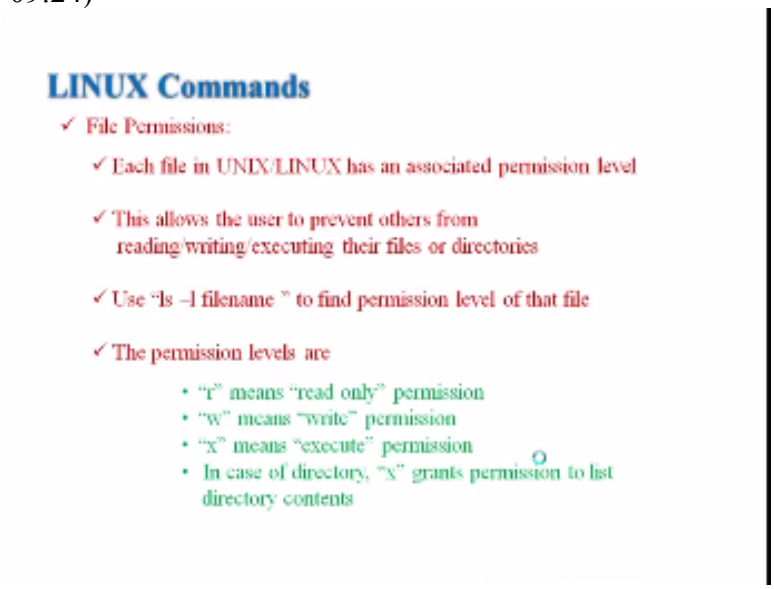
So now let us see some activities that we can do based on the lecture 2 and just assigning you couple of activities one is try to use man we learnt about man in the first lecture use man to find more details on the CP command and also the RM command you know laxity is the copy described man team NRT in terms of activity create a directory called the lec-2 in your home directory and create a file called mylec2.txt and it should contain the following region.
I have mastered how to create a file in Linux this you can do it in two ways for by both base and then also the copy this file over to another file on my Lec- 2. txt so you see here in my lec-2 the way that we write it is basically starting to load this character and then whenever there is a word

boundary the take the next the starting letter of the second word or the next word as an uppercase and then we continue all.
This particular way of writing is called the camel takes form I think this will be preferred way to actually code in variables or file names etc you see this more and more future discussions go into the programming itself it is a it is easy to read people also do like separate the words using on the score that is another way to do it but I prefer this camel case so it is also like very equivalent this lot of programmers today.
(Refer Slide Time: 06:42)



So now let us look at the topics for today is discussion now we will continue our discussion and we will do some more deeper look into the Linux system because today we will learn about how to manage the file axis what are the commands for doing that and what each of the what do we each characters or various types what do they determine and then we will also go into managing the system which is this is another big topic as to how do we control the program execution.
We would not go into a lot of detail in today is discussion we will only see how to access those the system resources we will also talk in the next lecture has evolved really going to controlling the resources themselves and then finally we will finish today is discussion with how to manage storage which is also another key point because in a general Linux machine operating system can assume that storage assistant.
Today the programs are becoming bigger and bigger and as you know like the operating system once we start working on the Linux operating system and if it is a Linux 64-bit operating system you have like all the 2 to power 64 bits as the address space which is like the trillion or number

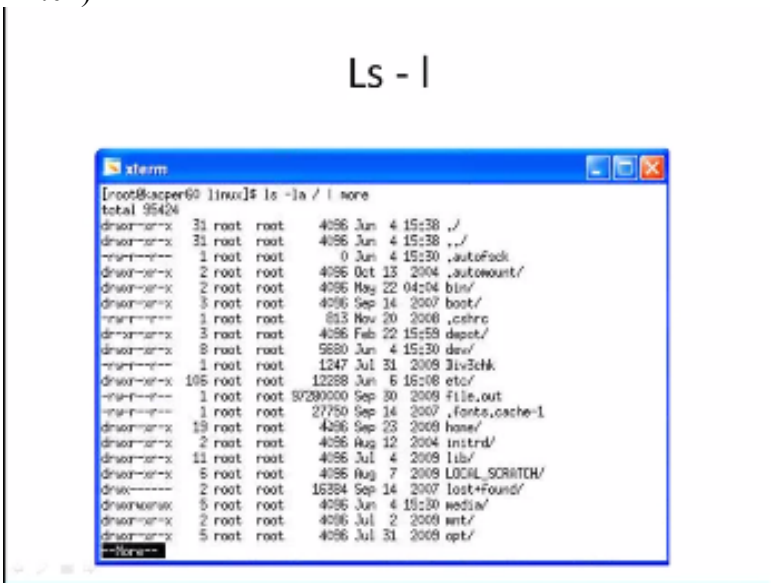
of bits to be addressed and so it almost looks as it is like an infinite storage space but in fact in reality like a limited storage.
So then we need to know how to control the storage and how do we get information about how much storage is left so that we can plan and manage according you because in reality the system will have one terabyte this so how do we make sure that we have enough space and even programs and we also look into liquid types of memory and the typically like I mean so you Heard about hardware resources for the RAM and the main memory things like that We will see like how those terms to translate into the Linux system.
(Refer Slide Time: 09:24)



So let us begin our discussion the number one is the file active or permission so in one of the key determiners of the security within Linux is because it has a very strict control on who can access the various files so when you create a file it is not that it is easily portable and basically as you know the Linux system it is a multi-user system so even others can actually come and see your directories and things like that.
But are they really able to see the see the file or be able to write into your files or they can publish into and they can execute from space so those are the key things that we will talk about so essentially like the file permission the permissions that you grant so when you create a file you are the owner of a file and you cannot grant permissions to others the level of access that we know that it is possible.
So a typical way to find out what the level of access from a given file is to use the electronic components and here we will go to the next slide and then come back to this and talk about the

permission levels but I want to you to read this R means the read-only right executable execute permission.

(Refer Slide Time: 11:01)



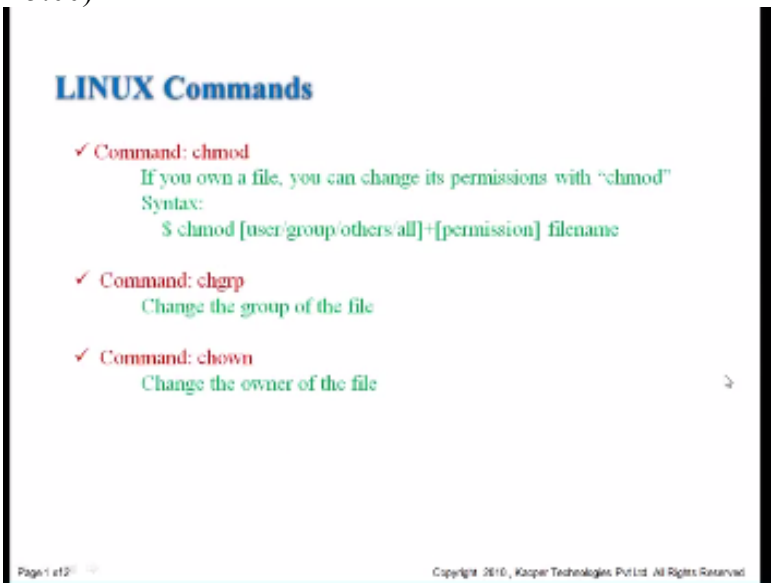
And then let us look at this so here if the LS- L command or more particular directory and then it saturates LS - L or AL which also now gives us the hidden files that start with the part then the pending so you look at on the left hand side we see that there are some characters so the way to read this is the first one denotes the directory basically if it is D then we know that that is the directory.

And if it is let blank then that we know that that is a fine then you have three sets of three characters each and each one is RWX and this is in or this one particularly this are - X and then the third set is also R-X and this is always the same basic in our WX and this could be our WX2 and this also can be RWX for the way to read this is essentially like the first one as I said it is mostly then do not see what is it is a directional mode.

The first set of this RWX characters that denote the user permission meaning what permission do you have as an owner of this file to access this particular one so for example the auto FSK you only have read and write permission so R stands for read W stands for write X stands for execute so in this particular auto FSK you have just read and write function and then the second three the second set of our RWX that denote.

What is the access permission for people who belong to your same group so in this particular instance you see the third column and the fourth column third column denotes the owner of this particular or if it is called a user of this particular directory and then the second column or

actually like one two three fourth column is actually be the group that this particular users belong to.
So again the six second set of the three characters is the permission that you are giving to the group and finally the last three steps you are saying that it for everyone else.
So for example in this particular is the particular directory is readable writable and executable there are any executable by this particular user and this group is also called group the group get the permission to read as well as execute.
That means that you can copy any files from this directly and everyone else also gets the read and equitable permission so that they can also copy as well as execute in this section so the same thing like if you look at this particular example will log out you can tell me what it is again it is a file you know and then the this person who is man if the user gets the read/write functions and then all the others get just the read function the really simple example this media and gets read and write a group of everyone.
(Refer Slide Time: 15:00)



So let us look at how we can control the axis then the we have like the others accessing this form these files so the particular command called change mode is allows the users to change the permissions on for a given user so in the change mode again you can specify various options which are essentially the user group others of all and then what is the permission and then the argument is the file the file name and the directory name.
Actually like the other yeah most it has a file name and we see like how we can change it yeah and then you can change the group permission of a given file just by changing by using the change group and change ownership is essentially if you want to transfer the ownership of file from you to somebody else another user so then you can use the change one or the change one.

(Refer Slide Time: 16:25)

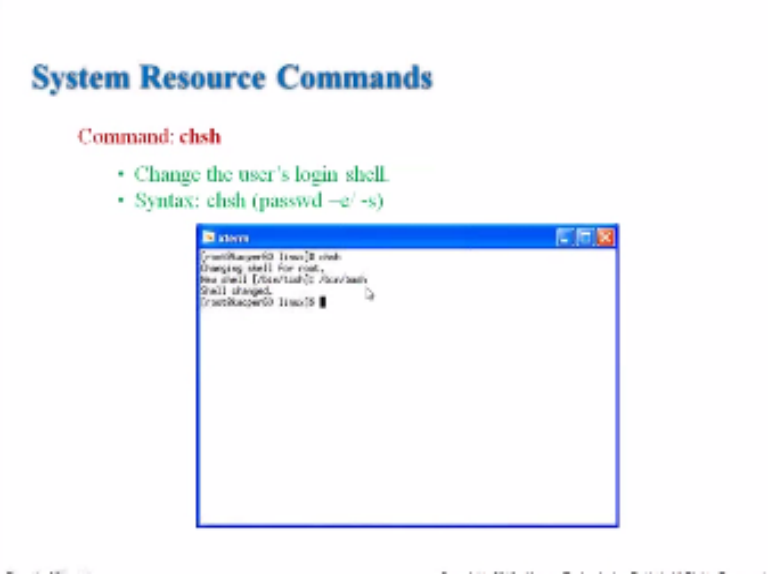
So let us look at an example here the change mode it gives 750 and file name so one thing that I mentioned was for the first set of three second set of three in third set of three these numbers can be expressed in binary option so you can think of this as a one bit screen essentially starting from the first RWX because the D are actually denoted by just the directory or whether it is directly or not.

So now we if you look at this one it is all like it assigns one bit at a time so there are 1 , 2, 3, 4 ,5, 6 ,7 ,8, 9 bits through and each set of 3 can be then go from 000 to 111 so now tell me how many combinations are possible so as you know since you are all like electronic engineer actually a combinations are possible so you can represent numbers from 0 to 10 so for each of them you can represent either 0 to 7.

We do not combine it and we do not use any other thing we will be like go up to radix eight notation so the change mode has a static state notation so here let me stay like 750 that means that the user gets the complete permission which is read write execute and then comes the five for the group which is only the read and execute so the middle number is popping basically and then finally the others cannot even read or write or it is no permissions.

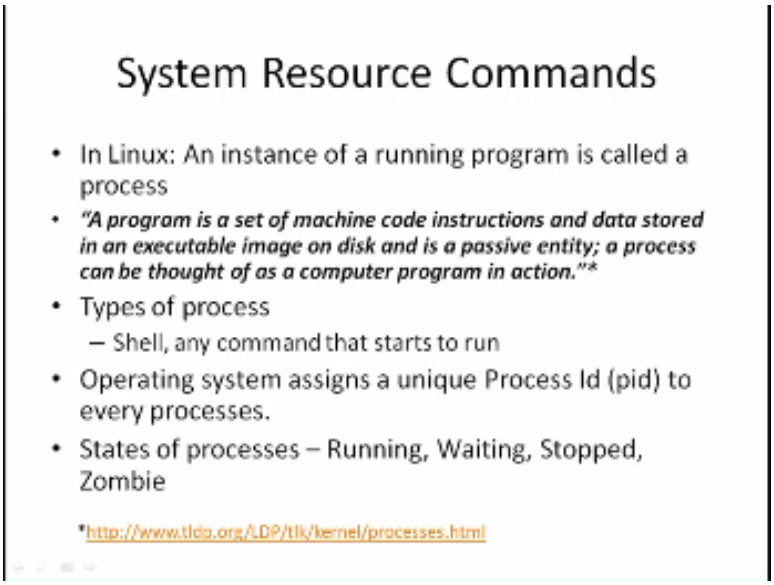
So here essentially into so we can we can look at various things and see what the corresponding numbers so can you tell me like I mean what the number for copy directories you look at it is 755 what about a desktop or the desktop it is actually 6 because it is 110 right and then 6 or 4 and 4 is enough we will give you some more practice ones that you can go through this exercise and we will see how to build useful page mode.

And this particular shell also offers from essentially lagging we talked about this in the first lecture of second lecture will be where it actually color codes the file names So that you know what is what for example the AXIRPL is actually it is a direction and then AXIRTL dot zip is like an executable wall it is a binary file and then the green is the text file things like that that we already know about this one so we will see as to how the operating system determines what type of file it is and then we can actually put together actually in this case like the make file is actually it is more like descriptive writing if it is for lack of them.
(Refer Slide Time: 20:23)



So let us look at that and now we go in to take some more system resource commands for one for how to report a date so the command is just the date you give the date it tells you like I mean what is the current date at the time so gives you the day the date and also time and it gives you the time zone and the year so a simple command essentially this is used in programs then you want to print the start point with this issue a command date.
And then once the program finishes you if you understand and date then now you can actually compare in between almost time the program took and then you can take meaningful actions based on that information okay and then the next one is the change shell command essentially this is to change the users login so it so you can actually like issue the thing and then this system will prompt or war the new shell is if you just type enter.
Then it retains it to exchange otherwise it is change to dash for example so it is an easy command to move from one shell to another sometimes like some programs are um they do want a particular shell and they want to execute in a particular shell demand certain resources and

certain control of resources so this is the easy way to actually change shell it now let us look at how can we change a password this is one of the important commands.
So here the command is just PASSWD or password that is good and then you can give options I would like you to find what are the options available and password or something like that but in general when you type password the system will prompt you to enter the new password and it checks against certain pre specified rules in this case it is actually it is checking against dictionary.
So if you specify a dictionary word it rejects that password because it does not want a password to be in dictionary so that somebody can copy it in the system administrator can set up various rules for governing the password so he can demand certain length of the password and demand certain special characters including password today most passwords the minimum of 8 characters needs a special character and number and the case difference.
So that that kind of gives like you are more authentication so for the user so once it you type in a valid password then it updates the existing password and there is a password file there all these passwords are kept and I think like it you can tell me like what will be the permissions for that one so if you have if you are listening to earlier slide the permission for that file that is protected only the protected and it is only system administrators can actually change that value.
So in general what happens is when you type password the password command actually gets control of that particular file and then it let us you edit that file only for your info and then once you enter then it is again put back and then the permission reverted back and then only the system administrator or root user and really access that point it is a not read not write not execute I mean maybe execute.
Only for the user so in terms of the numbers that we saw earlier it could be like the 00 or actually it should be 11 actually sorry I will take it back it is 711 so as a group person or as in others you can only do this execute because you do not want to see others password and even want others to see your password.
(Refer Slide Time: 25:44)



Now we go into our second topic problem take a long time today is discussion we already started talking about this but I wanted to highlight what is a process in terms of in the in the context of Linux so one key things when we talk about Linux we talked about the kernel we talked about the shell we talk about file system another key thing is also is a process in Linux any instance of the running program is all the process here this is the exact quote from the Linux manual.

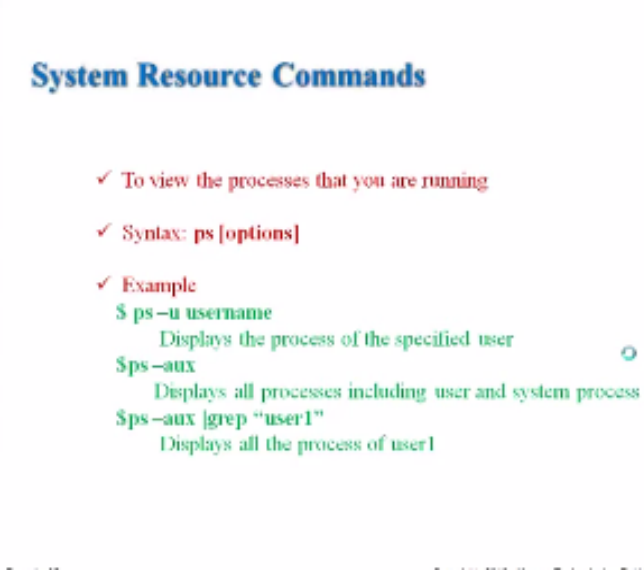
Which is actually listed here the reference it is a program or way they define it is as follows a program is a set of machine code instructions and data stored in an executable image on a disk and it is a passive entity so a program they call it as a passive entity this be there in the disk has not been executed but a process in this part of a computer program in action so when the and the program moves to the shell and starts executing that is what is called the process.

So what are the types of processes or example the shell by the nature it is there and it is getting executed that itself is a process and any command that you type in and then as soon as you press enter that is a process that becomes a process so how do we know that what a processes so the Operating system itself assigns a unique process ID the process ID is also called PID to every process so it is a number that the operating system assigns to the particular process.

So as soon as I mean internally what happens is form as soon as you type in a command and press Enter the operating system takes over it looks at that command it first assigns the process the process is specific set of resources required to run the program and then it starts that process and I assign that process ID so that it can track at any point what is happening to the process because the states of process is very important.

The state would be running the process or it is a gating process or it is a stopped process or a zombie process so we will see how we can distinguish these processes in much later stage.

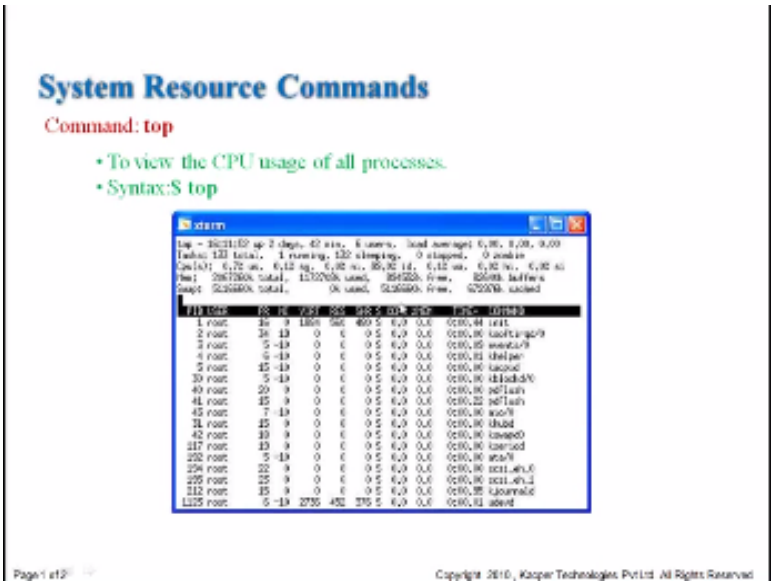
But at least like I want you to understand what way the commands are run is by using these processes so one of the key things is how do we control these processes and what we do for these processes that is what we will be learning books few slides.
(Refer Slide Time: 28:58)



So to view the processes that you are running the simple command is called PS process show essentially if you do a PS - R and PS-A that displays all the processes that are running on that particular CPU again this is a multi CPU system Linux but the way and it is also multi-user showing and multitasking so the multi-user and the multitask is all represented in PS so you can just type PS -A and in this case actually PS –AUX.
Which is essentially which stands for the processes that are but all the processes including the user processes and the processes that are executing in the background is without a terminal display all the information that is a PUX and then you can also specifically say like PS- U and then the user name then it will display the processes specific to that particular and here we use some command this we will learn in the next section called the grab.
And I also mentioned earlier that this type command which is essentially the way to do inter process communication the inter process communication we use many things one is this type anthem and the type command gets the output of this PS - AUX process and then feeds it into the next command so we know like semicolon separates for the various commands so far we have been calling them as commands book once they start executing like.
Now that we know term called processes we should really call them as processes so the semicolon is that process separator where separates from one process to another.
Whereas the type command actually takes the output of one of the processes and feeds it into then the next one so here also like the program execution is still going to be a video but output

can be feed into the next one we will also like encounter other forms of with inter process communication there are something called T and semaphore which we will learn about in the later stage So let us look at how this PS command is issued and what is what are the outputs.
(Refer Slide Time: 31:55)

In a simple scenario when you type PS this is a single user particular machine is used this ball this particular session is used by this one user and now it displays essentially like this information which is the PIDI mentioned the process ID it is 30380 that is assigned to this the shell so it starts from this particular terminal ETS is five and then it started at time zero and the command is itself is the shell itself.
And then since we type the PS that P is actually started another process that ID is three zero four nine four and the process command is so simple enough and we can it can be more complicated than this but at least like we will know how to display these processes now the process display is a passive display meaning it is given at the point of time but how come how can we look at least the system resources on an ongoing basis.
We want to see how things change on a continuous basis and then we are running a program you want to know how much the program is taking because there is no point in actually like running PS every time we have to do it like every micro second I said.
(Refer Slide Time: 33:30)



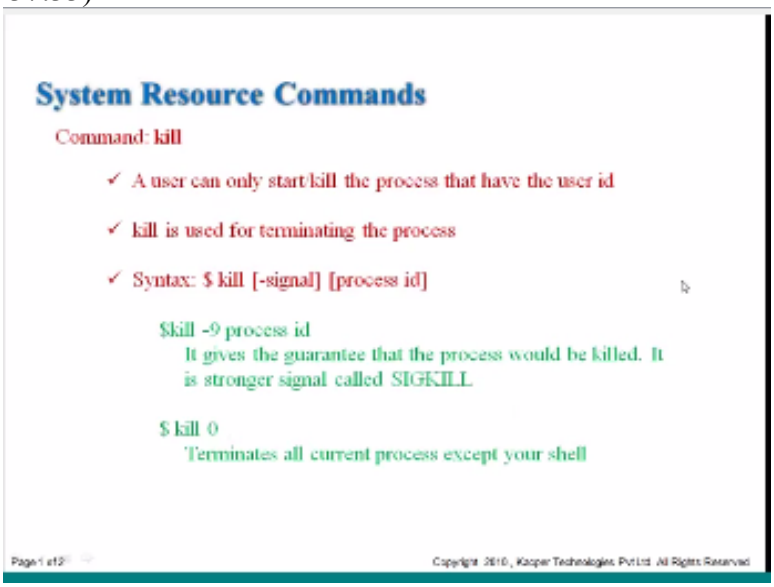
So there is another comment for that which is the top the top command is used to view the CPU usage of all the processes it is basically just type in top and then this process does not stop at this keeps running so unless you press Q to actually fit from that so here you can see much more details and on number one it tells you what is the time that this particular CP U to this particular operating system is up and running.

So in this case it is running two days 42 minutes and it has multiple uses six users in total and there are 133 tasks total of which one is one is run 132 are sleeping and then nothing is top and there is no terminal zombie processes so the task is synonymous with the processes and as I mentioned like the states are running slipping stop at one so how much is the seed with the CPU utilization here it is only like .7% I have by the system and then .1% is for this 30.7% is a user self when .1% is the system process and various others basically.

And then here we also have a concept of the memory so usually there are two or three types of memories actually one is the main the memory that is available there is also some concept of virtual memory which we talked about and then finally the swap the fox space so the you so here we see that basically this has about 200, 206 megabytes of which about 117 actually it is about 2.6 gist are available.

Total and average Gist about 1.1 gig is used in another 894 and negative three and then there are buffer allocated already and then there is a swap space which is a separate one which we go for another five game and then everything is free that is all swap space and now you see actually the real meet of the command which is it displays the process ID with the owner of the process and then the various rotation this we will talk about it later.

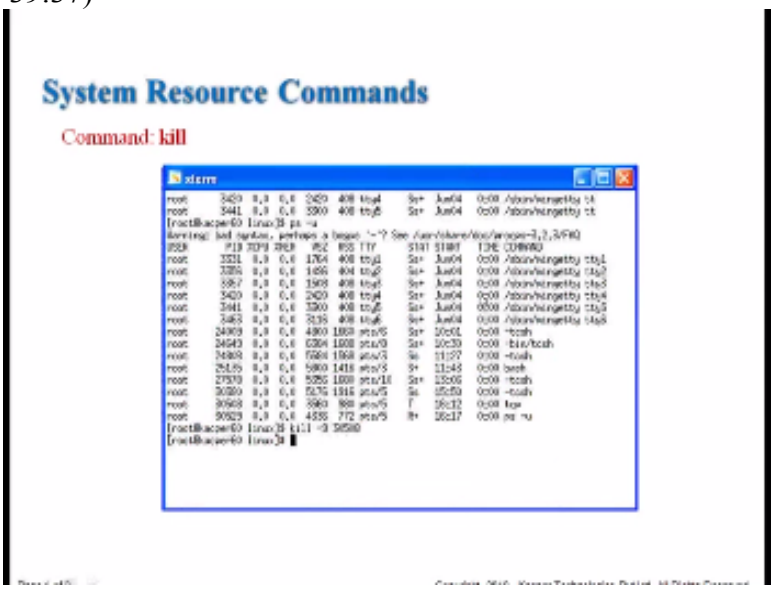
So here there is also like the virtual memory which is displayed as to how much version of is taking how much is resident memory it is taking and then this is the shared memory and then finally the CPU percentage and then how much time it is run so and this particular display is updated very frequently it is not like it displays ones and then it comes out basically it keeps changing continuously it is very similar to the task manager.
That we start in both windows for program and it also displays on a continuing basis also how the CPU consume same thing here you can see that there are multiple processes and various parameters somewhat and PR denotes a priority then this is the niceness basically so we will talk about these things for additional thing in the next lecture.
(Refer Slide Time: 37:33)



So now an important thing basically once you know like oh you know how to run the process of the startup process essentially you give a command but now how do you kill the process this is also another key important things essentially the command is essentially kill so the kill is terminated process and you can use the process ID as a way to identify bitch process to fail and then there are there are options available for this particular command.
Again the argument in the box ID and then the options are essentially the signals so here the kill -9 means nine is the highest level of form the particular signal which means that immediately kill do not even worry about anything like nothing just kill the process so though they essentially generates this signal called signal indicating tilde process or shore there is no questions asked skill set the kill 0 essentially.
It means terminates all the current processes except yourself so anything else is running like this is lag even becoming mortal just kill 0 and then it will a long process that are running on that

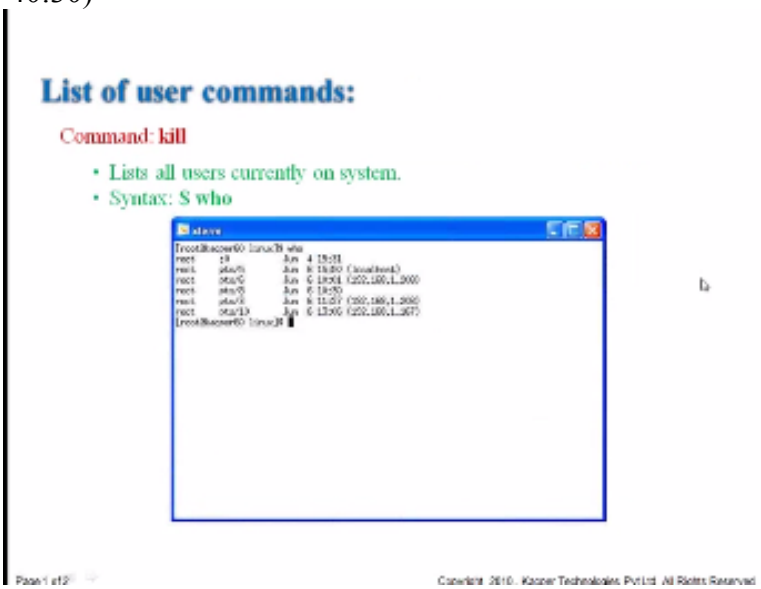
particular CPU so once we also like know how to communicate and how to use the network you will see like other commands for to work with the load balancing systems probably we will learn about that in the fourth lecture.

(Refer Slide Time: 39:37)



So here this is the top comment or it actually it is a PS -U where it displays all the processes running by this particular user and you have the process ID so you kill this one it is actually right here the particular the 3050 in cycle and at the top among so back one taken to scale using both kill -1 okay so now let us go into some more commands to identify to use it for anything for boxes.

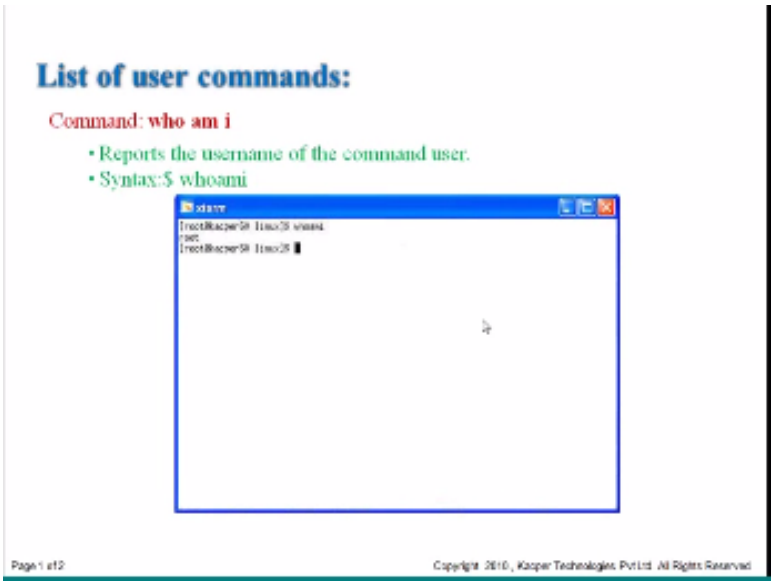
(Refer Slide Time: 40:30)



So the number one is who as we mentioned the Linux is a multi user system but you can know who are all logged in that who command system and it gives you the name of the user and also

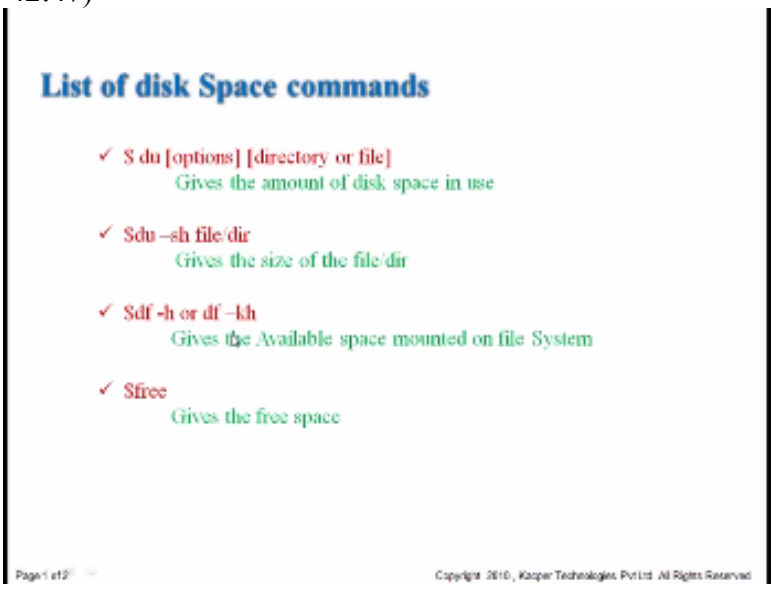
which terminal that they are using a useful command to see if your friend is logged in your System also you can see if there are any spurious persons logged in into your system so just typing this who will get you like although.
All the active windows and from where they are connected to and then now so these are their IP addresses and then basically like when did this start and then you can also again very similar to the PWD which is person working directory where I want to know where exactly I am in this whole PI system similarly you can also ask a who am I essentially.
(Refer Slide Time: 41:40)

And the command is this Who am I and that gives you the details about that particular window and that particular user who is using this one sometimes what happens is there are systems that are left by somebody else may be login wrong windows that are open which you are not sure whether it is you are the owner and you need to kill it or ready to need it go so who am I a mystic man to figure out it is indeed you who login into operating system in particular window or was it somebody else.
(Refer Slide Time: 42:17)



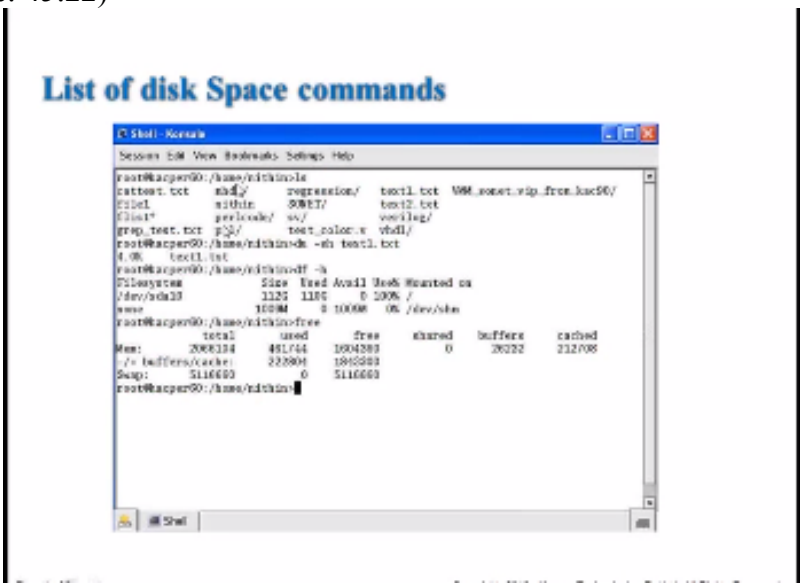
So there are two variations here one is agreement exactly this is this one in who that go space and space I and that gives you more details and if you continuously type in this Who am I in one word then that gives you your username does not give you like your details more than that.

(Refer Slide Time: 42:47)



Now we come to the third section which is how do we control the basic space or essentially so one thing to notice in Linux we said that basically everything is form through the directories and essentially even disks can be mounted in various directories so various Mon points are also Given that form victory so you can it is very easy to add additional disk space that if you want Or remove a particular disk awesome So at any point people could remove stuff or add more stuff into the system so one of the key things that we need to know is how do we what is the availability and how much disk space is there that we can use and also if we are exactly that this

located so the commands that we study right now are DU which is essentially sense for disk usage and gives you the options I mean you can give option for form of argument.
Argument is the directorial file if it gives you the exact amount of disk space in use essentially so we will see like how this command executed if the DU- SH the file or directory that gives the size of the file of the director and then the other one is DF command the D F - H and the DF - H this one gives you the available state mounted on a file system so the D U is more like the particular file the particular directory itself essentially.
As to how much it is getting how much space it will be goose is being used by that directory whereas a DR will give you more information regarding the whole disk itself as to what is the available scale or more where if it mounted and big than that And then free is another term ad which gives you amount of free space in the system let us look at an example.
(Refer Slide Time: 45:22)



So here so there is this form in which hand go various files in the directory so we do the LS and then you go all the things now you want to know like I mean what is the size of this text wonder text so you can save the human passage is the text or text picks on the text and then it gives you Its poking you can get this information also about this using element L that you already more about giving and that gives you the fifth column item will display the amount of space.
That the e to the file typically for directories it lists as 0 so the DU -HX set yet directory will also support will give you the amount of space taken by that direction then the DF- H here actually gives you the particular file system and as I mentioned the risks are even though they are Hardware objects they are they become directory which is in / - the 10 so /dev is all the disks are mounted on bottom and then here.

It is 112 d is the size out of which 110 is being used and the use is actually hundred percent because two can be represented so it is really hundred percent so you need to go and like thing and then there is a something called nun and then basically one gig is there then all the things are available and that is mounted on like the a section so now you also have this command free and that that gives you like.
I mean how much is a total memory and which is used and vomit extreme and then there's a shared room the amount of buffers that are there are special types of memory.
The cache area is another special room and then how much force equal wall space is essentially required for swapping program with either one so processes when you solve processes this song space so I think this is probably the end of the lecture 2 so again we talked about several things I just wanted to recap one more time the things on the lecture 3 is actually like we looked into more deeper into the Linux system.
We understood how to manage the file access the various permissions like our W and X and then the base groups the user group and everyone else so how do we give access as well as for giving change actions of various files we also talked about managing the system resources for from the perspective of what are the what is a process and how do we control the process so again we talked about one key aspect of the system.
Which is the item and essentially how to ask the output of one process into another process we will talk more about the processes in the next lecture where we go into some more additional commands of T and fork a fork is essentially lefty and then the other ones form the concept to is four which is a concept to understand how the really works we also talked about the storage before we associated between the memory.
The swap the virtual memory etc there are special memory elements like bubbles and dash areas which are locations but you cannot write into it because those are used by the system for writing and then we also saw how in control or at least get information regarding the usage of the resources from being so abusing the commands like BU and BS and freaking out command.
And uh one thing that I will add is we talked about the various colored representation by the always for wearing which by lifting cities so as you know in Linux or in with any the UNIX type of the system you have you can write scripts that we will see more and more and also we have regular times the platform so the way to user or write a script is essentially we need to prepare the script with whatever the shell that we want to use.
Usually in the form of hash-bang then we will specify which shell that we want you like may be user bin wish you have been this will also SH is also work so if we specify that line as the first

line file then the interpreter treats that file as a script and then it actually runs that the script itself we know that it can be straight.
You still need to give the file function as the function but the difference between a normal file or individual file is I mean executable file in form of the script the script is essentially it is a human readable program so usually that one we will have that that is the first line so once again I want to thank you all for listening this lecture and we will pick it up on the next one on this point thank you very much.