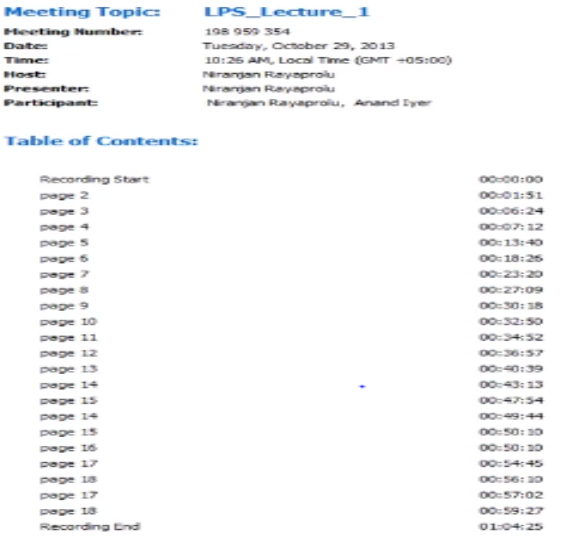
(Refer Slide Time: 00:00)



Meeting Topic: LPS_Lecture_1
Meeting Number: 198 959 354
Date: Tuesday, October 29, 2013
Time: 10:26 AM, Local Time (GMT +05:00)
Host: Niranjan Rayaprolu
Presenter: Niranjan Rayaprolu
Participant: Niranjan Rayaprolu, Anand Iyer

Table of Contents:

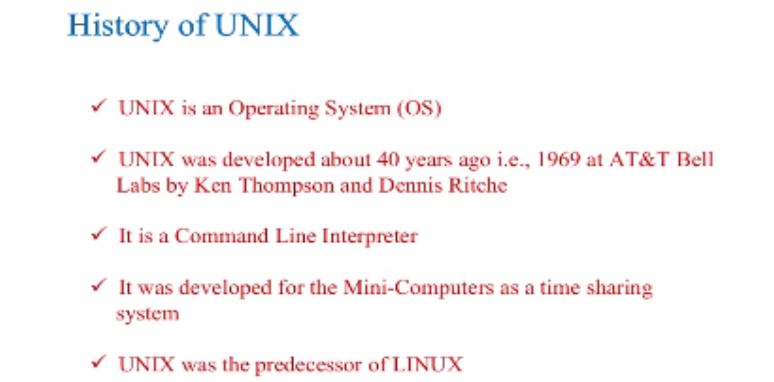
Recording Start	00:00:00
page 2	00:01:51
page 3	00:06:24
page 4	00:07:12
page 5	00:13:40
page 6	00:18:26
page 7	00:23:20
page 8	00:27:09
page 9	00:30:18
page 10	00:32:50
page 11	00:34:52
page 12	00:36:57
page 13	00:40:39
page 14	00:43:13
page 15	00:47:54
page 14	00:49:44
page 15	00:50:10
page 16	00:50:10
page 17	00:54:45
page 18	00:56:10
page 17	00:57:02
page 18	00:59:27
Recording End	01:04:25

SEER AKADEMI
Linux Basics

Hello everyone again welcome to this Eric Adam ease course on Linux and programming languages today we are going to begin our lecture with the Linux basics my name is Anand and I will be their instructor for this course, I will be today like I mean I am going to mainly talk about the Linux basics some of the history of Linux how it evolved into where it is today one why commercially.

It is very successful and we will see why it is very successful commercially and then we will also start working on the Linux with how it is structured means we will study about the file structures file systems and the various programs that are embedded in Linux. And then I will be taking you through how to interact with Linux we will begin with some of the commands the basic commands that we will give you and then I think you recover as much as many common core possible today and then we leave it for the next lecture.

(Refer Slide Time: 01:54)



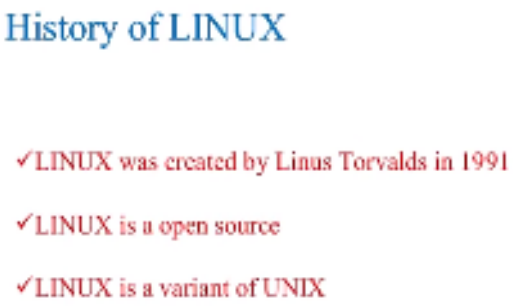
So let us begin you let us really talk about the history of Linux the history of Linux the is not complete without talking about the history of UNIX itself I think many of you are familiar with UNIX some having worked on it in different times at different times so far UNIX is an operating system very similar to Windows or any other operating system that he may send a loop another thing that comes to mind today so it is widely used is the OS is walking system building ball Apple I Phones Windows is very popular that is the operating system supported by Microsoft it is widely used in almost all go PCS .

Today probably it has a market share or a go Predecessor also UNIX is mainly used in the Pentad computation world which includes the LSI design and many other areas ,we will see why it is widely moving books on computing world the UNIX itself was developed more than 40 years ago 1969 at the AT&T Bell Labs the two pioneers came Thompson and Dennis Ritchie where the main people developing this operating system and in fact if you look at the timeline actually predates even the Windows operating system.

Which is for Google into being in 70s one thing to note about UNIX is the command line interpreter what that means is as you type the commands that is how it understands what you are typing and body of the customer will need services what you are requesting it is developed for mini-computers form as you know the computing world itself evolved right in the from 1930s when the big main trims were invented by IBM then the IBM the then the mainframes actually went into mini computers.

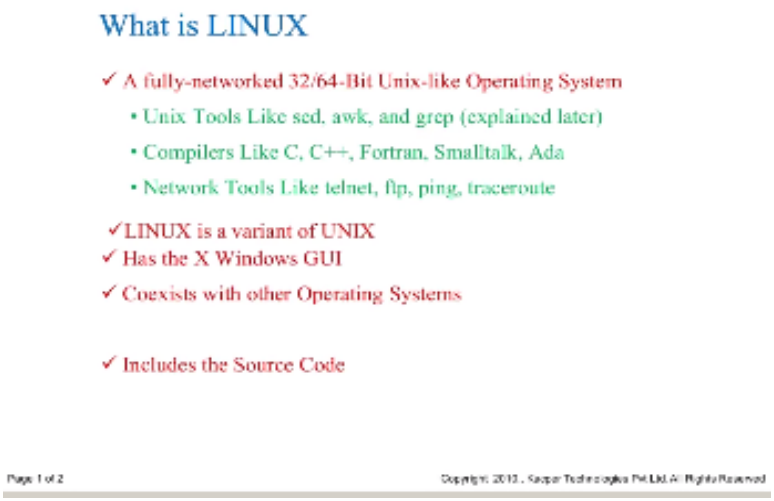
And then in the 1970s the PC revolution started with the PCs becoming coming into the focus for the mainframe stencils the operating systems they are preparing and he will use to interact with the computers using punch cards and ladies other types of programming not really developing

any kind of personal programming, interfaces so it will be like more like a batch program so everybody will be writing program they will be submitting into a punch card machine which turns the program into punch cards and then one person collect.
All these punch cards go to the computer he runs those punch cards and then the computer will produce some results. I mean all punch card or some kind of a printout and then that printout is distributed to everyone who wrote the program and they can look at the result and then they do not go and modify the mini computers really started the revolution there the computers started interacting with people on a one-on-one basis so this is again one of the key aspects and this also prompted what is the time sharing which means that every person gets a fixed time on the computer.
I hope in your experience too you must have taken these kind of computing forces where you actually sit in this terminal do a fixed time or time of work and then move on and then somebody else will take your spot and kind of a time sharing you can also think of time sharing in a much more granular form that we will talk about in the later stages of course and again as I mentioned the UNIX is the predecessor for Linux and we will see like how Linux came into being and what are the key aspects of Linux and why is it popular.
(Refer Slide Time: 06:26)



Today so Linux it serves Linux was created by Linus Torvalds 1991 it is created as an open-source when more line is actually proposed LINUX said that this is the lightweight operating put them and this needs to be an open system, and it is a variant of UNIX in the central optic uses predominantly the core basics of Linux and then it adds some more so one of the key aspects of

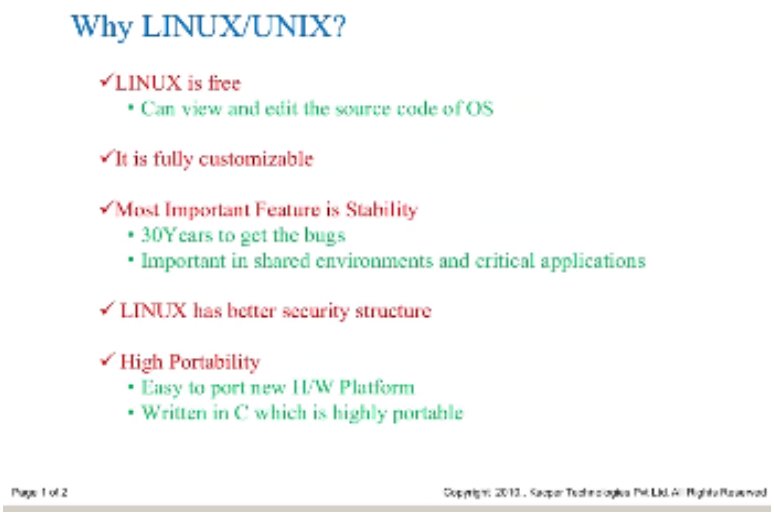
Linux is this open source and we will see like what is an open source and why is it important, so before we look at that let us look at the what exactly is Linux.
(Refer Slide Time: 07:14)



The Linux is the fully networked 32-bit as the lesser 64-bit Unix-like operating system what this means is it uses UNIX tools like set arc and grip will again talk about these things later on in the course form just to remember these terms said all can rip you will see what they are it has built-in compilers or various programming languages C C++ Fortran small talk and ADA these are the two other languages also supported these are the expectation like languages we will we would not be learning about these languages but just remember that it has a support or multiple languages as well and then it also supports networking tools like telnet FTP ping trace route etc.
I hope you guys are already familiar with some of these terms for example telnet is telnet allow users to communicate to computers which are removed remotely located so they can actually use telnet as a protocol to login into a computer which is situated namely make a positive rule and then basically use it as a server to service whatever their programming needs FTP is it is an acronym or five Transfer Protocol again this is mainly used for transferring one file to another one file from one location to another location again.
We will talk about some more concepts on the FTP as to how we can leave that thing either just a remote system there you can actually warm whether a particular machine is alive or dead using the spring program and more we will talk about all those things in a mulch only through all stage again as I mentioned the Linux provides a way or multiple uses to communicate or multi uses to use so it is a multi user multitasking and multiprocessor system, why is this important multi-user.

Because now it is not like your PC where you are actually sitting and you have your own system your own window you just communicate to it and then basically like it serves you and so for example if you are editing a word document that word document is so personal that only you can do it you unless you put it on the web or you use a mail program to send it to somebody nobody else is will be able to see what you are trying to do but as in a multi-user situation.
Actually you can do some things and then your friend can come and add on top of it things like that so again the multi-user is key aspect why it is preferred in this computing world in the scientifically with multitasking again you can simultaneously work on many tasks within the Linux operating system this is another key aspect today in personal computing world like them if you contrast it against the Windows operating system, Windows also allows you to actually have multiple programs open.
But you still can only work on one program at a time here you can actually see multiple programs as one program run you can also work on one of the one things like that it prompts with multiple windows in the inside the same operating system, and multiprocessor again that is the important because now you can actually add on processes to increase the throughput of a given program and work simultaneously with these multiple processes again contrasting to Windows operating system.
Windows provides the maybe multiple CPUs in the machine but it is still us working on only one of the aspects of it whether it is multi CPU multi-core is it still servicing only one use or one program it has an X Windows GUI again X Windows is so different from your normal Microsoft Windows in the sense that now you can open up multiple terminals inside the same window and then work in each of these terminals differently it also coexists with other operating system one of the three things that you will notice that today's x86 world.
Actually now you can put two operating systems at the same time for example in your PC's you can create a partition where you can have a Linux operating system on one side on partition and then the Windows operating system on the other side, it also runs on multiple platforms again this is another key thing as I mentioned now the Linux works on x86 it also works on SPARC processors HP's that you can name it like.
I mean all these platforms can support later and the important aspect that I mentioned in the previous slide was that Linux is an open source and yes so it includes the source code as part of the operating system so even though we call it an open source you have to think of this as providing this source code for majority of the programs and we will see like I mean how this has evolved into even better and why people use it.

(Refer Slide Time: 13:43)



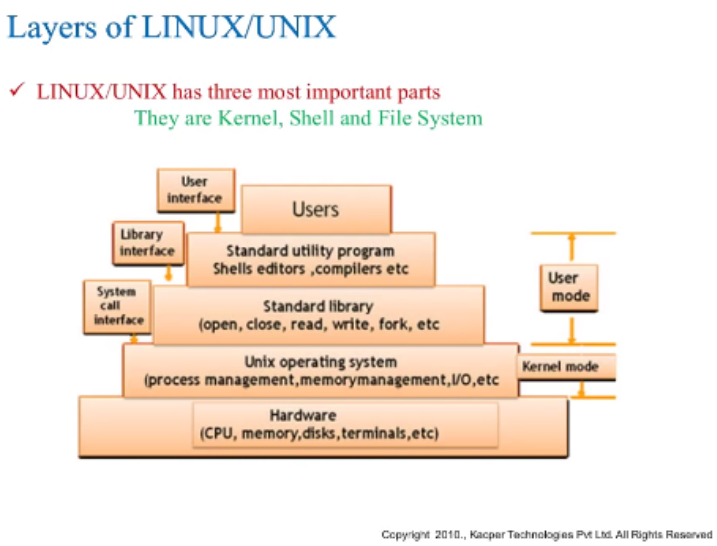
So let us see like why Linux is free. I mean Linux is used today by men lot of fun company number one is Linux is free so you can view and edit the source code very easily the source code comes shipped as part of the executable itself as a part of the bugs even though I call it like Linux is free in reality, it is really not free there are commercial systems like the red hat fedora who actually produce Linux or commercial consumption the thing is even though Linus Torvalds original vision walls have this Linux has a free operating system it is a lot of purposes but at the same time.

You know support became an issue and some of these companies took over the support and so that they can provide the bug fixes and more additional updates to certain features that the users are requesting in a more streamlined fashion , so that is the reason why there are commercially available Linux installations and of course the commercial installations are attractive for the companies like ones who work on the design or VLSI design for example, because they at least get a reliable installation of Linux rather than looking at the open source.

And see like then some of the bugs may be fixed or something and it is fully customizable so again people can write their own custom scripts and custom tools on top of Linux, and with for example ,if you have a real-time operating system or also a real-time processor then you want to make it a real-time operating system, you can put your own customizations on top of the Linux operating system to make it real time one of the key features is the stability which is which again comes from the open source.

Because once it is open source it is tested and verified by large community of users and they actually can constantly test the new code as well as they provide the bug system so it is basically

about 30 years of testing has gone into this operating system and hence ,we get a very reliable and a stable installation again this is critical for critical applications even VLSI design for example is a critical application and so we need these kind of highly stable environment and it is also good for the shared environment as it opens.
Today Linux also has a better security structure we noticed that actually in Windows we come across several Trojan attacks and virus attacks but for the Linux side we seldom see any of these kind of attacks by the spurious program , the reason is the security infrastructure that is already built in as part of the Linux ,since all these way that it work is very reliable and very consistent it is very hard to break it into this zone and it is also highly portable we can if we have a new processor or a hardware platform.
It is very easy to actually extend into that platform and it is also written actually it is written in C so that actually makes it more portable across these multiple arm platforms, so now that we have explained the why Linux is used so a lot of companies today let us look at what are the main features for LINUX.
(Refer Slide Time: 18:27)



So that we can familiarize ourselves with , so Linux has three most important parts one is called the kernel the other is the shell and the third one is the file system, so let us begin from the bottom this is the hardware , so again the goal of the Linux is to actually to for the users to communicate with the with the hardware so on the bottom we have the hardware on the top we have the users, so the users actually communicate with the hardware so let us see how they do it.
So the first layer that eunuchs the Linux office is the operating system itself which is essentially as the process management memory management etc , they are the very low level programs

which goes into the hardware resources and they try to control the hardware resources okay, on top of these programs, we have the standard library which are used or opening/closing read/write fork etcetera these are the various commands the hole that support or that uses these posit management programs they call these programs to do.

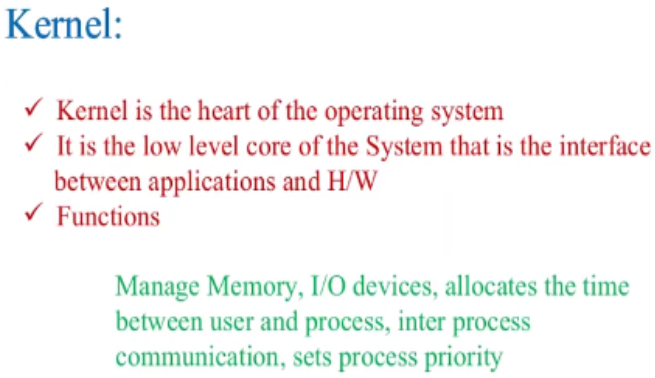
And on top of that is the standard utility programs like shell editors, compilers, etc and they actually call these other programs to facilitate how to do certain tasks so as a user you interact with these selectors and the components so as I mentioned earlier the three main systems are the three main parts of Linux are the kernel, shell and the file system, so you interact with what is the what is known as the shell and then the shell then pills have worked with the kernel which are the various management functions.

The process management memory management etc for and that in turn works with the hardware to satisfy our your requests , so let us look at some more of the terminologies themselves so when the users communicate with the shell editors that aspect is known as the user interface ,so as I mentioned the various terminals that you open within the windows are all the user interfaces and that part that forms the shell and you will also see like what is the shell the shell is actually driving it is the program that is driving these terminals.

So that you will be able to interact with the arm with the computer then the shell then shell itself works with the standard library and me this qualifies the library interface and then finally the libraries are populated using these for the management commands Process Management memory management, except for which manages the hardware resources and that is what is called the system call interface that interface between the libraries and the operating system on the other hand we see that actually though all these programs that really work on the process management memory managers etc.

In the cursor they are all the kernel and then anything above kernel is the user mode or the shell and going in between is the file system, which we will see how the file system is organized etc so let us look at the kernel in more details.

(Refer Slide Time: 23:21)



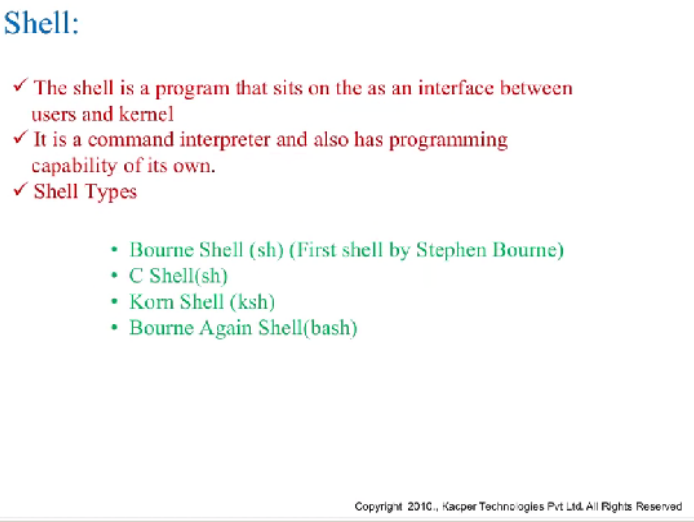
So the kernel is the heart of the operating system it is the lower level core of the system that is the interface between the application and the hardware so you can think of the kernel as essentially there are programs that calls various hardware resources , and they also make sure that your programs finally are getting executed in time on certain processors or on certain more the hardware resources and with all the resources intact the required resources so the functions that if performs are managing the memory managing.

The I/O devices and then one of the key things that we talked about was this share the time sharing between the processes so again it provides it allocates time between the user and the process it also does the inter process communication which becomes very key when we talk about whether we want to interrupt the process whether a process communicates with another process ending over things and it also steps process priorities ,so let us look at the brief example for example you if you have a writing a program say like you are starting a big array so it knows how to fetch that away from a memory and then you are actually interacting through the I/O devices.

So it also watches either the work results there are any additional inputs come coming in and meanwhile another user can start a program to compile a program or he can start a program to actually go through a dictionary and sort and find a particular word so it again timeshares between these two processes ,so that it knows like okay works on this or sometimes. I go starting program while it goes in like such as the dictionary for some other time and then meanwhile you may say that okay, you know what I forget this program.

I want to kill this process and then so I press the ctrl C will see or the commands in the later stage but they you press the ctrl C and then now suddenly that becomes an interrupt and it

understands that interrupt and basically kill that process and say like we want a process to communicate to another one essentially like. I mean you have started a process now for reading the memory or writing into the memory and that you want your answers to go into that process and it basically understands.
Which processes that need to communicate between each other and then it is correspondingly does that and when you are doing all these things suddenly like the memory management process comes along and says that more trees all the programs that you are running now I want to see like I mean how the various sectors are there .I want to compress some of the sectors and then I will fear so that process gets a higher priority and that also set by the term we can think of the kernel as the main thing that all the things and making sure that everything works properly.
(Refer Slide Time: 27:11)



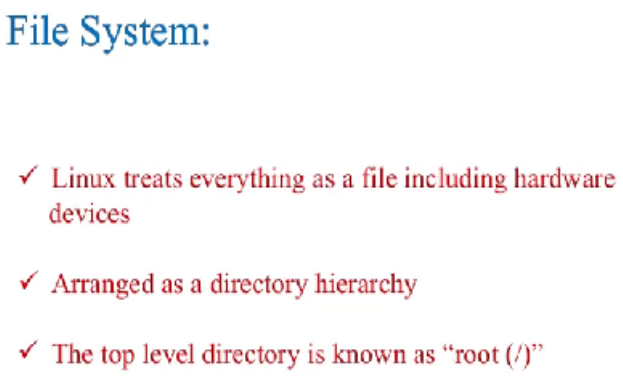
But now let us look at the shell what does that provide the shell program is the it is the topmost program or compost interface between the user and the kernel and it usually contains a comment interpreter and it also has its own programming capability, so what this means is a shell is something that you see in the terminal then it comes to a prompt the prompt is represented as a dollar or greater than sign then it comes to that it actually means that that the command interpreter is active.
Now whatever you type in is interpreted as a command and once you press enter that command gets executed the command gets executed in the background the shell understand this command it take that command it executes the command and then it spits out the result and then again it goes back to that prompt and sits and waits for you come at the next command so you can think of shell as your own personal slave is ready to do whatever it takes to terrify the master.

So you tell him one command because goes and runs the command gives you the output whether it is an error or yell output increases output and then he sits that and baked for your next column so he is really elegant dumb way to think of it and think of the shell that way and the shells are of multiple types one particular shell is called the Bourne shell it is denoted as this SH and he tore the shell made by Steven bond for this call this old name that is for SH and then the other one is the special.

It is the denoted as CSF there is a type of here this not on that it is actually CSH there is a modified the seashell powerful C shell and then there is something called Korn shell which is denoted as ASH and then there is a then another shell called Bourne-again shell which is also denoted as bash shell line torvalds first invented Linux operating system he actually coded the bash on top of the Linux so the first shell that was Gordon on the of the bash over the bond again shell.

So now that we looked at the second part of the operating system on the shell now we move on to the next one which is a file system so once you understand the kernel the shell and the file system then we can go on into the various commands.

(Refer Slide Time: 30:20)

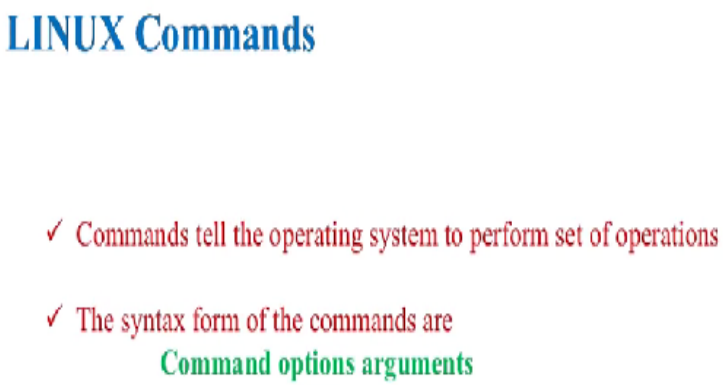


So let us look at the spy system one concept key concept of Linux is that Linux treats everything, whether it is a hardware resource or an I/O device whatever it is it treats them as files so for example ,if you are using a keyboard to enter the keyboard itself is a file so you can say like read keyboard which means that it opens a file called keyboard and it tries to read whatever you type, in so now my question is how do you actually write it into a terminal the terminal device is also a file so if you write from your program into a file or terminal at its displayed into the tournament.

Now the hardware resources are also like various files so they get executed we will see in the later sections as to how they are denoted usually they are because it has /dev or devices and one other key aspect of it is there is a fixed hierarchy or the file system and that is how the operating system finds there everything is residing for example, if you want to put the device
you will typically put it into a slash there or if you have other files will be putting more putting it into another location again.
The top level of the directory or the top level directory is known as the root and that is denoted as just the slash so when I take slash del that is the priority under that slash ,which contains some devices and then there are several other things you will see like slash etc slash bin various things which denotes specific functions and you will talk about that in a later stage.
So now that we understood the three main things we can start looking into some other commands and how to navigate through the lid open system.
(Refer Slide Time: 32:53)

So in order to get started we need a couple of things so when you start the device with Linux operating system typically we go to the computer you would turn it on now what happens you see like bunch of stuff going on your monitor scrolling through the monitor which is basically need initializes device or microphone your disks accessory and etc, once they are all initiated initialized then the system boots up and also it boots up the windows and then finally it comes up with a screen which contains which asks you for log.
So the login information typically is you need to provide the username password and the username is typically set by the system administrator it is a user specific it is a your own personal unique name and in Linux every keyboard every username everything is a sensitive, so an uppercase Anand will be different from a lowercase Anand two people can have different

usernames it is just the case differences and your password is also case sensitive also it you can change at any time and though you cannot change your username in time.
You need to go through the system administrator to change the reason we will see like how we can change the password in later section what password can be changed by the user I think so now that you have logged in into the system with the username and password now what happens so now the system actually comes up.
(Refer Slide Time: 34:53)



With the number of windows maybe one window or several other windows they are called the terms of X terms which stands for just a problem on the on top of there we and inside each of the terminal it also starts the shell, that we talked about and this could be one of the mini shells that that are supported that I talked about and then you know that the shell is active because the shell comes up with a prompt and as I mention the palm can be just a dollar sign a percentage sign or even gesture and a greater than symbol.
Once you have the prompt now you can ask it to perform the tasks that you want to get out of it so they are waterfall commands, so these commands tell the operating system to perform a set of operations so these operations we will see what kind of operations, and you ask the operating system to perform the typical syntax of the command is as follows the command followed by options followed by arguments again command options and finally arguments so how do we use these so this command .
Let us first talk about some basic amounts and then we will go more into all the more advanced commands and how we can string these commands together to perform ace operations you let us look at some of the basic command.
(Refer Slide Time: 36:59)

LINUX Commands

Control Keys:

- ✓ Control Keys performs special function
- ✓ The control keys used in LINUX are

^S	Pause Display
^Q	Restart Display
^C	Cancel Operation
^U	Cancel Line
^D	Signal end of file
^V	Treat following control character as normal character

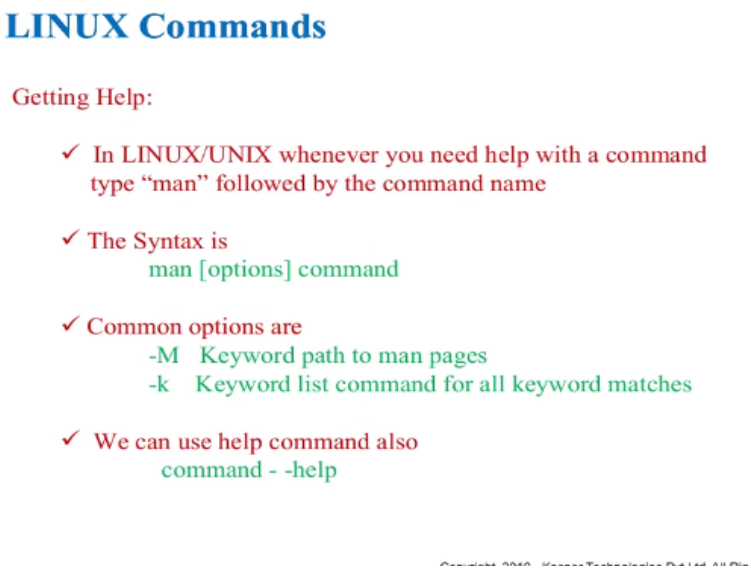
So these are all just shortcuts or control keys essentially so how do how can we use this control team to perform various functions so if you look at your P board the control key is oddly at the bottom most key and on your left hand side if this is CTL so we use that for doing certain operations, so in Linux control s means it will pause the display so once the display is filed actually even if you type anything nothing will show up on the terminal the control queue on the other hand will restart the display.

So that once you pause and you can restart using the purple cue the other one p control sequences or control see that cancels an operation, so you type in a long command sudden you find directly more. I do not want to execute this command let me cancel this one you can do a control C on it or say like you started recommend using like a the command interpreter and it is just going and it is running and you do not know like ,I mean after one hour come back and see that actually still running and you want to just cancel the command.

Because we know that you would not take more than an hour so again you do the control C and immediately it aborts that command and then comes back to the prompt so for cancelling an operation you can do the control C control-u cancels in line again the same thing once you start typing a line and then you just wanted to abort Midway and then because you may have done a typo then you basically do a control you control D is the signal the end of pipe and control V is to treat any following characters as a normal character.

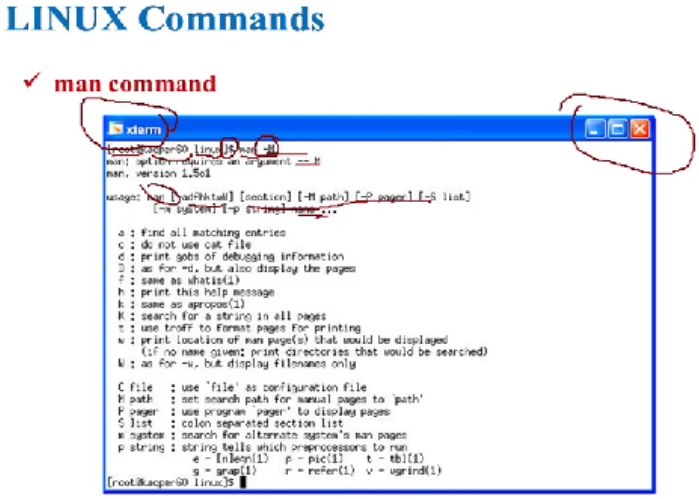
For example if you type control V and then control C it does not take the control state just some things that you just typed SST and then if this taxi other 50 control things are control a to go to the beginning of the line so for example you are typing a long command and suddenly you think that oh man I made a mistake in the command name and I want to change the command name.

So you can go to you can just press control a and then give me the cursor will you go to the first character of that command and then from there you can either retype it or you can use arrow keys to move left and right and then the other nifty one is also simple E which takes you to the end of the line which is also pretty useful.
When you want to come back after typing this particular or making a change in the beginning of the command now you want to go back to the end of the command and then start continuing to type the remainder of the command in control E is a very good way to do that you so now that we understand understood being the control key operations let us look at how we can get help in Linux.
(Refer Slide Time: 40:40)



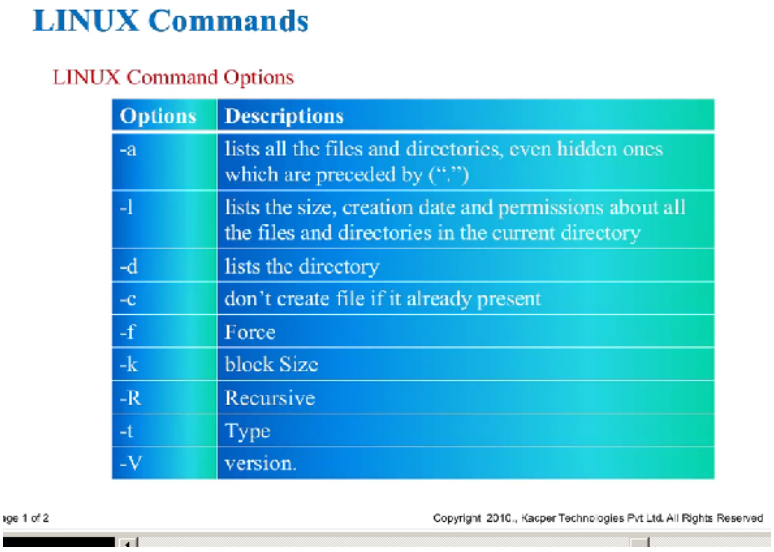
So in Linux you can get help in many ways one of the ways that you can get a get help on a particular command is to use what is called man the man is the short form or manual, so it is like a opening up a manual and then looking at what particular commanders, so the way to use this option is or will type man with some options or options are just optional and then followed by the command so the man options are this -M and for the keyword back to the man pages or - K you would list command or all the keyword matches.
So you can stay man - K and then you can save anything that you want maybe you want to just say dir and then it prints out all the keyword matches for a particular man comment so again you if you look at this one is also the man also follows the same syntax what we talked about that is the command ordered by options followed by arguments, so here the command that you are interested in to get the main pages becomes the argument or the man command itself, so again in

Unix this is one of the key features basically you have the commands as arguments are or be these other comments.
Now another way to get help is to decide the command itself and then use the - help here the - help is the option there are no arguments and then the command is this you are a loop and here you can use like - help and - - valve interchangeably so you now let us look at an example of how the man is form use.
(Refer Slide Time: 43:14)



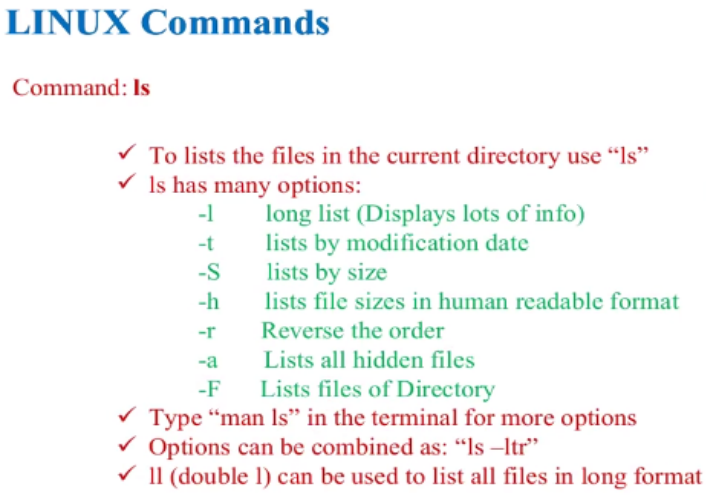
So here I will learn if you talk about various pieces first of all the X term you see here is the terminal that popped up after opening the Linux this is after you login and then, I need the this particular form but in the terminal and in the terminal you have all these various buttons, this is for minimizing maximum and then actually removing this X term itself there is also you can go to the border and actually warm click on it and drag and make the windows bigger or smaller for you to work with now let us look at here.
It says the root is the name of the user here and then this AC P R 60 is the name of the machine that this person logging so here essentially like this root does not mean the PI system root but it is the person who is logging in so the this person's login name is wood which is essentially a super user and then um the machine itself is this K ACPR 60 and the directory typically this is the Linux is the directory then the dollar sign denotes the shell here this is most like warm pas edge of the bash shell.
Now this person is now behavior giving this command so it is a man and then - M so beating this will work you can see that actually doing work because man as a command option but does not have an argument so it says this option requires an argument which is the - M and then this is the

version of the map that is that is getting installed actually remove some of is the annotation for the better and now it also provides you with the usage how to use the manager so here it is man is it all these are your options in the including this and then there is a impulsive thing.
Which is the name which is essentially the name of the command and then it goes on to actually explain to explain you as to what each of these options mean for example A is find all the matching entries, when we solve some of them to the uppercase M that we solve the pad the desert set the search path for the manual pages to that so if you give that M with the fact it uses that path to go and search for the man pages and then we also select the dash K which is the year it takes basically like in the same as a command.
So you know basically helping perform um the other options some of the interesting ones like the print location of the man pages S W in fact I typically use the man command with probably even without any arguments in just a man and in the command that works most of the cases unless you want to change some of these so this is one of the key things, so now you know how to get help within the system within four years every command so this is what we will see in the next slide.
(Refer Slide Time: 47:58)



So again it is a here it is described in more details the - a list all files and directories even the hidden ones that are preceded by dot so one key thing about Linux is some of these files you would not be able to see it when you do a test of display command unless you provide this – A option a stands for or and then - L is with the size creation date and the permissions the L actually stands for long so it to provide the long option essentially like I mean it gives you more details about various things then the dash D actually lists.

The directory and there C is it will not create a file if it is already present we will see in a cult use this dash X is typically means that the force and dash K is the block size - R is another P option this could be an appetizer or a load a form for stand for the and see let us see like how we can do perform, because it is useful for copying removing things like that to use this recursive nature actually the type and then finally - we typically -P of the version so if you want to know which version of a particular command is being used you can just simply say the command – B.
And then it which produce that version or example in the previous slide we saw that the Linux by default printed out the version as by 1.541 you but now we will go into various command I will start with the LS.
(Refer Slide Time: 50:19)



LS stands for list and this is mainly used to list the files in the current directory so there is a concept of a directory we saw the file structure with the root directory yes slash and then there you can also have a bunch of directories on the wall so every directory contains number of files and to list the number of files, we use this command called LS in fact there is an analog this command even in the windows world where you can display the number of or the name of all the files inside and electric and LS also has the mini option again.
I mentioned the - L that is one of them which is for the long list this way is lot more information - thing is list by modification date which is very useful because it starts by the modification bit and you can actually the - tea option gives the files from latest to the oldest modified we can do R to reverse that order so again you can combine like this some of these, I mean actually all these options together.

They are not you do not need to use them exclusively and forget s list by size H is less file sizes more human readable form and then I told you they are reverses the order L looks for all the hidden files in the file names that start with dot or hidden so if you just simply give an LS command upon show up unless you put an LS dash a and then uppercase S this file of the particular directory.

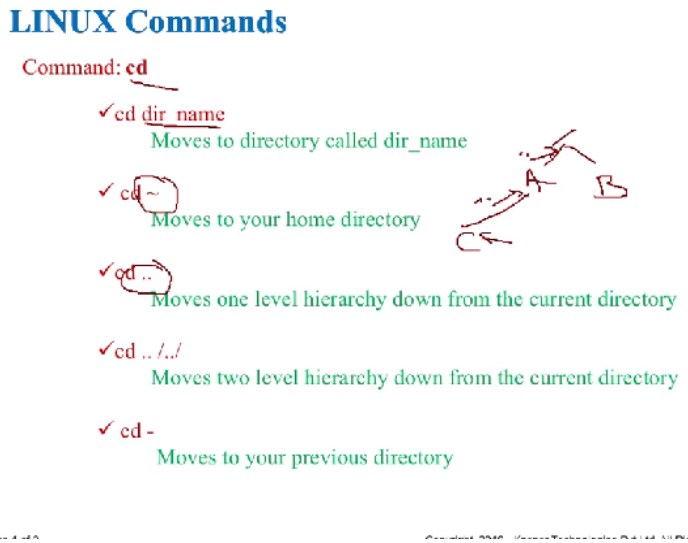
So one of the useful thing is here you can see that as I mentioned options can be combined and someone tell me what this command will written LS - n TR I think you are correct this returns the long form of the five long name it starts first by the modification date on the latest to the to the oldest but then because you see an R it reverses the order, so actually it displays the files from the oldest to the latest the latest files displayed at the very end of the list I think like should be clear.

I also urge you to actually type man - Ln our man LS to get the manual entry for the LS command itself it should be to print you like all these options and then it also tells you whether you need an argument or mark again. I want to ask you that question to this LS need an argument. I think again you are correct it does not need an argument because it just lists the files in this direction and double-l is to gain a short bond which stands for probably the LS - L.

So then these are what are known as aliases we will also look at some of these things and how to set a billion so people coming section but this will be a good introduction for LS command and then ,I want you to actually try it out and display with these commands as we go along and then see how they work and what you learn from them so now let us look at actually using the LS command in a particular direction.

(Refer Slide Time: 54:47)

So here again I do not have to remind you some x term and in this terminal all these other things that I mentioned the prompt and then now you type the LS so look at how it produces the pipes this is just mentioned the LS right so it produces all these things now tell me how these are presented to you they have this simply sorter down alphabetical order and then just present it to you in the path between your form, so you can see that one and if you have read it in various columns it is column 1 column 2 3 4 5 & 6 you can get the same thing with another problem also known as LS dash X.
That lots of print in this format you so now we will talk about one more command or that it is the CD.
(Refer Slide Time: 56:17)



CD stands for change directory so anyone guess how to use this demand so CD is used to move between various directories if you want to move from one directory to another you use this command Percy so the commands that are most popular in Linux are CD you being representative the elephant does it need an argument . I mentioned that it probably does not have an argument that is correct in most cases there you do not need to provide an oven but you can also provide an argument.
Which is you can type in LS and then space and then you can stay like I mean particularly then if that file name is found in that particular exactly it just returns the file that is not found then it will tell you that this file not found now there is a concept all the wild card , which we will learn in subsequent sections the wild card is typically the character, that kind of represents another variable or another name itself.

So for example ,you can say star or the asterisk symbol which denotes any character or a question mark will denote any character but only of one character length register can denote character of any length or character stream open in length same, so how can you use it you can say like LS star dot PL then it will only give you the files that have PL in as their last characters or PL the suffix because you put next are not here you put a star peel then it allows a return to you anything that has PL as the last two characters.
In the tribe which is slightly a bigger set than the star dot peel if it is a and if you say let us start txt then it only gives you the txt files which are in this case basically this copy file e^x one.txt x2 DX DF on 3x and maybe outdoor things so is useful in elect a man to actually filter or some files and the others, so this is again very useful confidential and now let us look at the CD command so the CD command essentially moves between the directories we will briefly touch upon this so the CD a directory name moves.
To a directory called the director name so the CD is the command name there are no options and the argument is the director The Reckoning it moves from wherever it is to this particular directory so you do not need to provide where you are moving from essentially but it knows that ok that is where you want any, so what does it mean by actually moving from moving to a different boat this means that any command that gets executed subsequently will start as from that directly as a reference.
So for example say you have a directory called slash A and you actually went to that directory initially and then go doing commands like LS then it produces the list of files within that directory called a but now when you do the CD to /p and then put an LS command now any time the new commandment only like looks at that particular action as it is written so now it produces the list of files in directed now there are characters which have special meaning for example the tilde the tilde this denotes your home directory.
So every user has a home directory where he has he or she has many other files and we will talk about how to pay the bomb directly what it means for the home directory in a subsequent slide so CD tilde moves the cursor to that particular home directory that your home directory and so now whenever you execute any subsequent home it always uses your home directory another one is this dot ,dot the two dots denote go to the higher level directory.
In that particular parity so if you have a slash a slash P and from that if you are issuing CD dot, dot it goes to /a a like I mean the slash indicates the hierarchy separator or the various directories so if you are doing CD dot , dot slash dot ,dot then it moves to level up the hand here we actually

like the terms are slightly confusing the hierarchy down it is actually like I mean the CD dot dot moves it and level up essentially.
So the way to think about it is like you have the slash as the root directory and then say a and then II then pee has the C&O move and you are here the dot, dot we will move you up to then the another dot, dot we're moving up to the flash so if you want to go from C to B there are two ways of doing it one is C D /b that will directly take it to the B or you can say CD dot, dot slash dot ,dot slash.
So that is another way to make tool and CD - actually typically moves using the ver ever you are you will come it goes back to that place for example if you say from C CD dot, dot slash dot, dot slash B so that now you are here and then just a CD - from B it takes you immediately to see or that is from the biggest action , so I think this should suffice for the today's lecture let us continue from this point the next time thank you very much.