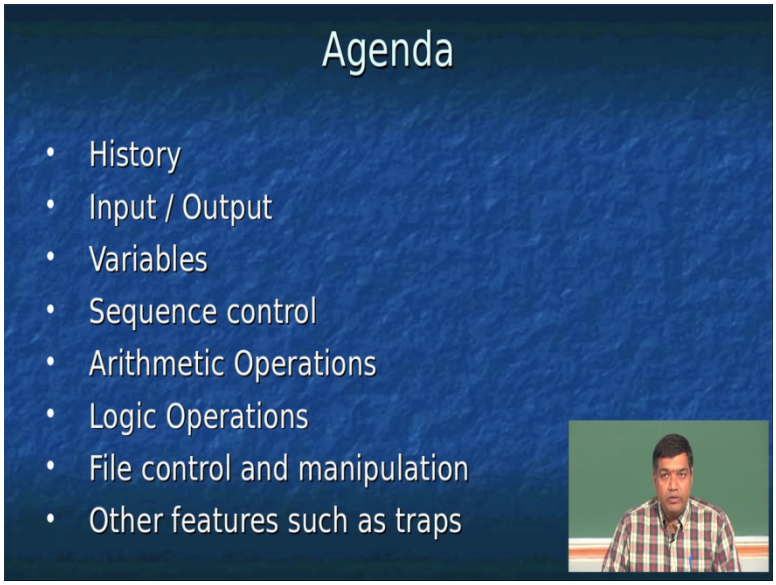
Information Security
Madras Sri M J Shankar Raman ,
Consultant of Computer Science and Engineering
Indian Institute of Technology
Module 04 Lecture 27
Basics of Shell Programming

Welcome to this addition of basics of Shell programming I think until now you would have covered the principles of operating system as well as some basic commands with respect to shell, in this context you would have got practice by typing lot of Shell commands individually what we are going to do right now is, in order to accomplish a particular task you might need more than one set of Shell commands to be executed.

In this case you could have also use the and and and other kinds of a piping or other kinds of mechanisms to execute more than one command but if I want to include or do a series of activities one after the other or in parallel ok? Shell provides you a programming language or supports a programming language using which you can accomplish the task using a series of steps.

(Refer Slide Time: 01:32)



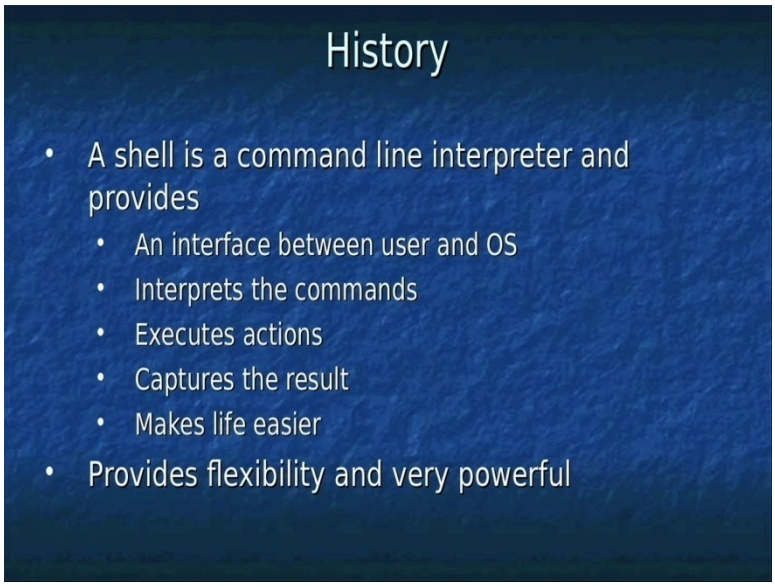
So what we're going to see in the next few classes or sessions are that we'll be looking at the history of Shell. Specially we're going to concentrate on Bash which is available with the Linux operating system and is a variant of Bourne shell, we will look at this what a shell is definition of a shell and what are the basic commands that you can use in the Shell what are all the variables etc, also see how to get data from the user as well as give information back to

the user, we'll look at what are the types of variables in the Shell support actually shells are do not support what should I say I mean data type I mean they are type plus therefore they are called these variables are actually type plus so every variable is interpreted as a stream and we have to give special sequences or type special notations to show that its a number etc, then we'll look at sequence control essentially this is a concept of while do loops far loops etc and we'll also look at what are the arithmetic operations still support almost all the basic arithmetic operations that are used in programming languages as well as logic operations.

The next part of file control and manipulation actually becomes very easy with shell and if you take a programming language like C you have to open a file then you should have specific Syntax etc but in shell all this task is very much relieve in fact you will be surprised to know that when the Linux mission is booting and the system is shutting down the amount of shell script that is used to boot the system and shutting the system is enormous if you are very curious you can take a look into the directory call ETC me dot D and inside those directory there are tremendous amount of Shell Scripts usually they start with k or capital K or capital S ok?

So such is the role played by shell scripts in the Linux operating system we'll also look at the other features that shell provides it essentially a signalling mechanism or capture of signals etc using command such as trap, I will be practically showing some of this command so it's better you you have your laptop that is running Linux next to you so that you can try out some of these commands and see for yourself how it works, let's start with the history of Shell ok?

(Refer Slide Time: 04:14)



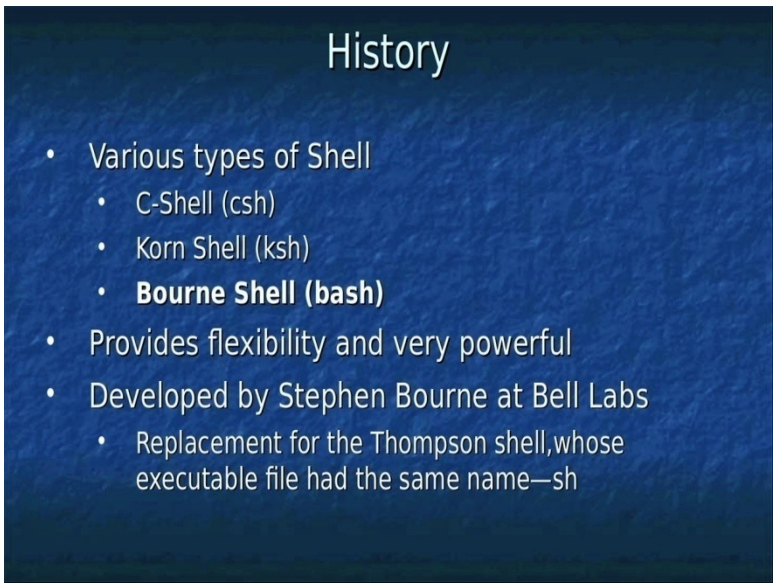
The shell is actually a command-line interpreter okay many of your work with beautiful nice graphical user interfaces in your operating systems but then why should we go for a command line interpretation which is actually you might think it's sort of very primitive when you are working with the computer and you have to type lot of stuff on the keyboard etc, well the reason why people you still use command line interpreter is that it's very powerful I mean you can if you if you're typing speed is very fast you can accomplished lot of things very fast, that's one of the reasons ok command line interpreters have been very I mean command lines when you you type a line of command in the Shell.

And if you look at very advanced administrators of computer systems many of them especially in the area of to manage network providers they prefer command line over any nice GUI because you get your task done extremely fast with a command line as they would have already told you a shell is a access interface between the user and the operating system.

So it one of its task is to understand what the user is typing in from the keyboard and then after the user types in it tries to identify whether there is any kind of syntax error in the command that the user has given its so it tells it tells as such that there is an error in your command then it executes the command by forking a process and then you can capture the results I mean if the command a command that you execute out of the shell can either succeeds or fail actually the shell keeps track of what are the commands you are executing in fact any command that you type in the Shell actually becomes a parent process ok?

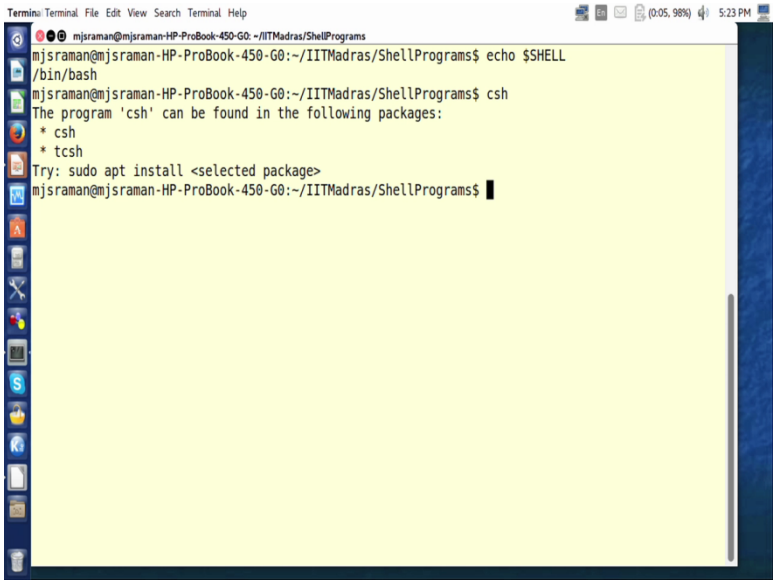
and finally it makes life easier so initially it's very difficult to understand shell commands because it's very strict with respect to syntax I mean if you have an extra space are you have a tab or you don't type a space shell actually speaks out lot of error messages and as I told you the most important aspect of shell is that it provides is flexibility and its extremely powerful we'll see the power of shell and you will recognise the power of shell as far as we complete these sessions moving on there are various types of shells that are available I mean if you are a C programmer then you would prefer using C shell okay

(Refer Slide Time: 06:37)



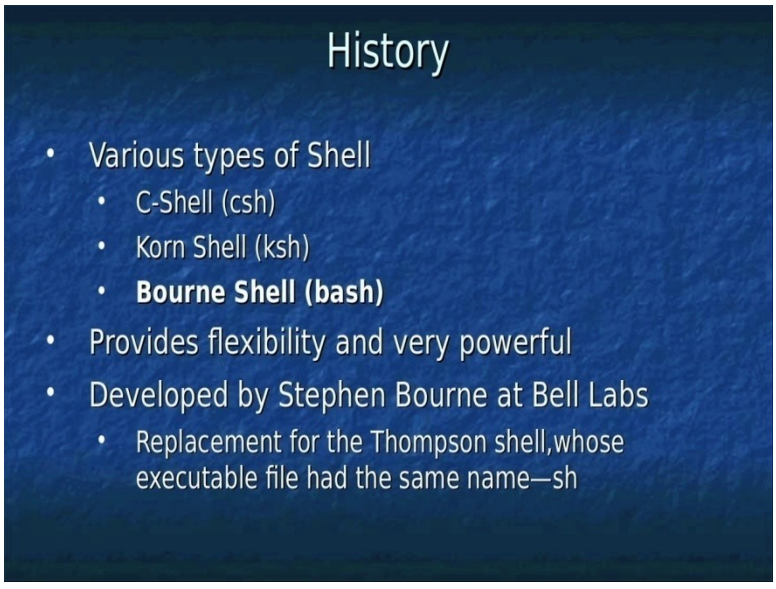
Korn Shell has some advanced features but what we're going to look is a Bourne shell or a Bourne again shell what they they call it to be pure, we call it as Bourne Again shell and will be working on Bourne Again shell many of the command that are available if its a Unix command it'll work with any other cell but the way the syntax is might slightly differ so you need to be very careful when you are trying to programming Shell if you write in bash shell and there are times where it it won't work on a C shell and it show Syntax error etc so you need to be very careful and so you initially you have to identify what type of Shell you are using ok?

(Refer Slide Time: 07:19)



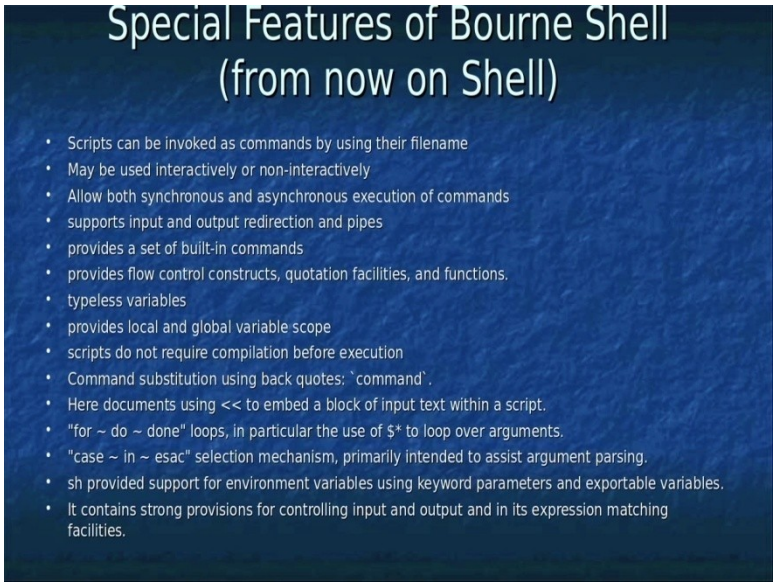
To in order to identify it you can use a very simple command ok for example you should type the command Echo dollar shell so it tells you that you are planning to use Bash shell. So in this way you can identify what type of suppose I want to have C shell ok so in this my case I have not installed C shell therefore it shows error that C shell can't be found otherwise it will it can actually install C shell then it will actually go to C shell so moving from one type of shell to another type of shell is as simple as typing a command in this case.

(Refer Slide Time: 07:56)



This Bourne shell was developed by Stephen Bourne The initial version of the Bourne shell initially it was Kemp Thomson who had the executable by the same name sh ok so they changed his name to Bash ok?

(Refer Slide Time: 08:09)



So some special features of Bourne shell I mean from now I'll use the word shell or Bourne shell interchangeably, so some of the special features of Bourne shell is that as a (08:19 initial) shell scripts can be invoked as commands by using the filename I mean this I think you have been doing quite regularly when you are trying to interact with the operating system the first shell that you would have land up is the Bourne shell and you would have been interacting with the operating system by typing lot of commands like cpls, etc now the best thing about shell is until now you are been using it in a interactive fashion.

That means you type some commands and the shell responds back so and then you wait for the response to come in or you put it in as a background process by putting an and symbol and so on what you can do is you can have the series of commands typed in and then they can also be non interactive in the sense you need not interact with and write to a file or read from a file and then based on what you write or read from a file you can take certain actions etc all these things can be done.

So this is other advantage of a Bourne shell and as I told your making the program run in the background all these features can be done within the Shell scripting language also the most important characteristics of Shell is this input output redirection I think you would be very

familiar with this all those input output redirection can be used within a shell scripts and the notice that the shell provides a bunch of built-in commands ok? We'll see some of those built-in commands during the course and the most important in order to accomplish a certain task you have to sequence bunch of operations you have to provide control and you have to iterate many times you have to choose from various paths.

So all these functions are provided by the shell scripting language ok it has flow constructs it has and if any one define functions just like a C programming language you are you define some functions you can provide you can define functions within the shell and you cannot there's something known as a code facility and we'll get to know of it when the course promises and as I mentioned before the variables are type less.

So it's very easy to manage the scope can either be local that is within a particular shell or Global which can be exported to other cells etc ok and the most important is Shell works as an entrepreneur the advantage of interpretation is that especially when you are learning something new programming language or something is that you can type in the command you can run it and then you can see in front of you if there is an error you can fix it, of course it's very slow but you have to pay a price for sex ability, the bar codes each of the keyboard characters has special meaning in the shell.

So especially the back quotes saying that you can substitute the values etc we'll see some examples of them in due course you can also write a shell script on the terminal ok and then save it as a file and then execute commands, it provides as I told you for iteration it does far do while etc and for it provides the case statement for branching flow of control etc and then there are certain parameters these are all called environmental variables ok so when a shell is executing its needs to know who is executing etc.

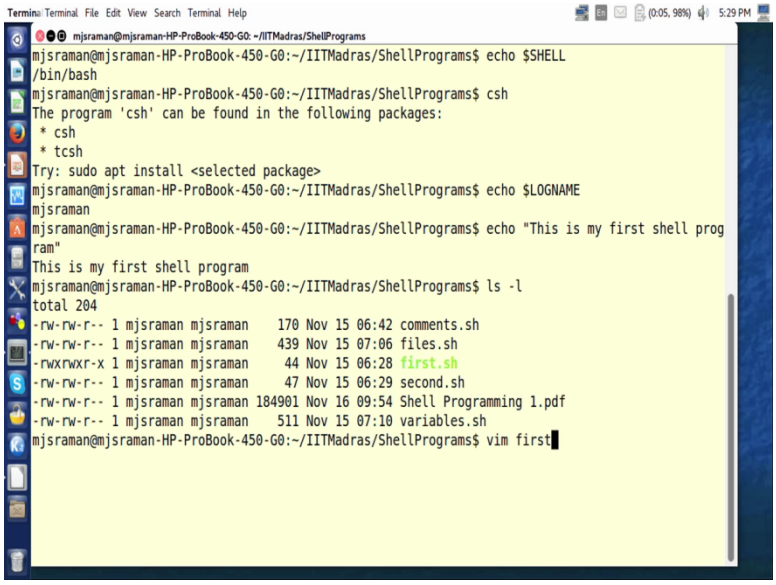
So for example let's say from the shell you want to find out who is the person who is lock them ok so what you can do is you can go ahead and type a command called echo dollar log me in and it tells you the name of the person who is logged in ok and of course you can ask me why I already is there the screen why do you want to see but I just want to show you a demo of how shell stores some variables ok for example this logged in parameter stored by the shell so it it it also has a lot of expression matching facilities of Islam in it should be there because I am in you know you learnt about the grep command in it provides a pattern matching facilities so similarly the expression matching facilities provided by the shell.

Ok so having see if if you want to see look at other advantages or more advantages of shall I would suggest that you go through the Internet there are lot of documents that are written on the shell the various features that are been added etc but for us its very important we get hands-on. so let us try to write our very first program with the shell okay the program is essentially prints this is my first shell program .

(Refer Slide Time: 12:43)

So in this case what we do is.

(Refer Slide Time: 12:45)

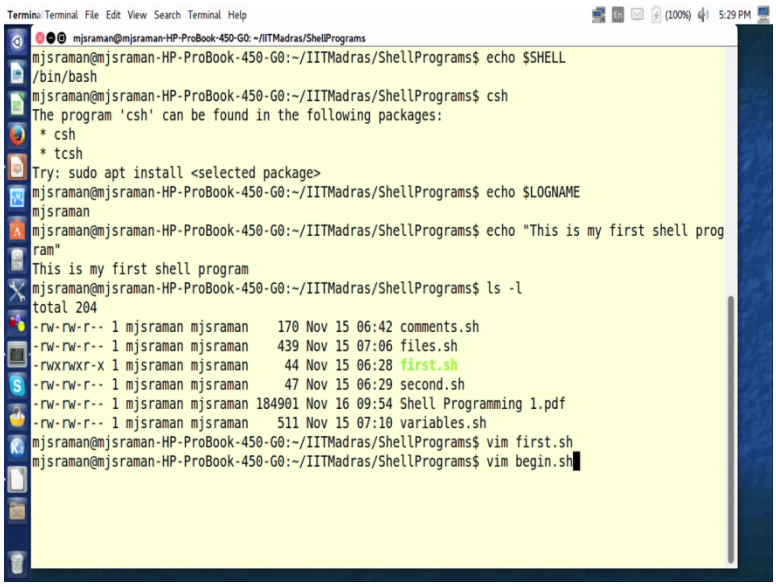


The first way of doing it is I can just print it like this this is my first shell program and what the shell does is it just prints whatever is given after the Echo and that is there within the double quotes, now I can convert this into a shell program see this is just a command I have typed in now what I can do is I can convert this into a shell program so if you look at this and I've already typed and using a editor called VI improved editor ok?

(Refer Slide Time: 13:26)

And what are you done is forget about the first line you see what though I type is already present in this line, line number 2 ok? So what I have to do is I can just write a program using a text editor something like this so probably I will write it in for you first OK?

(Refer Slide Time: 13:44)

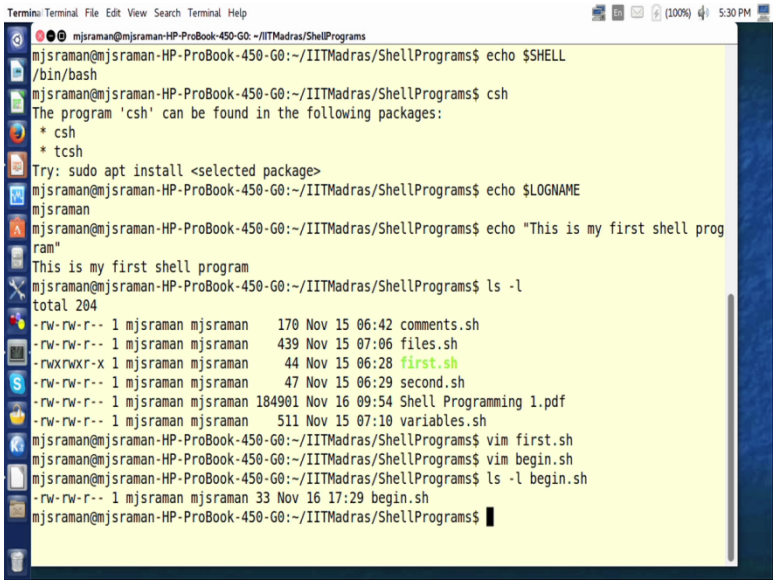


So I'll write a program called say let us say begin dot sh ok.

(Refer Slide Time: 13:45)

And I just type in echo this is my first program.

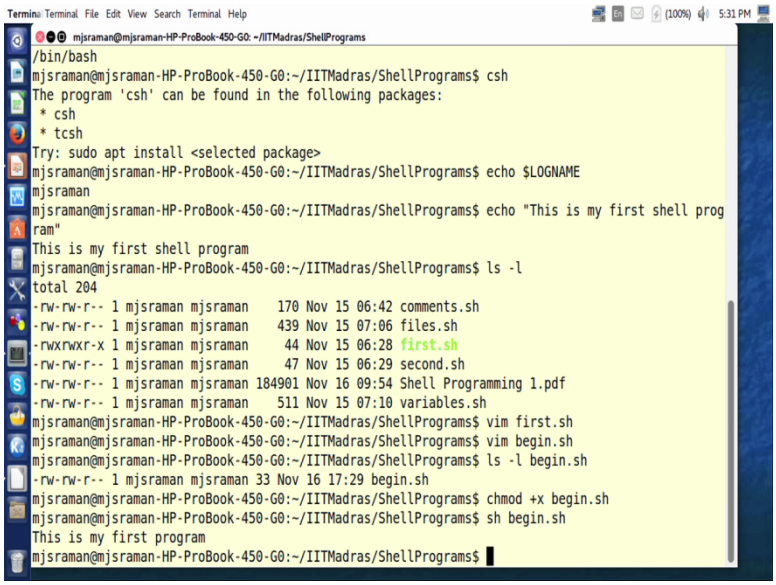
(Refer Slide Time: 13:59)



And obviously I mean I I store these files so if you look at this I so i stored a file called begin dot sh is my now the question is usually the convection says that you should store this file with dot sh okay extension ok so we are stored it with dot sh extension. So if you look at this the storing task begin dot sh. Now what we will do right, now is the question is if I put within that sh you see that this file does not have any executable permissions, so you would have

known about this that all these executable permission is needed to execute any file. So what I can do is I can just get the executable permissions ready for this program by using this command ok? And once I get the executive permission ready now what I can do is I can start executing this program ok?

(Refer Slide Time: 15:00)

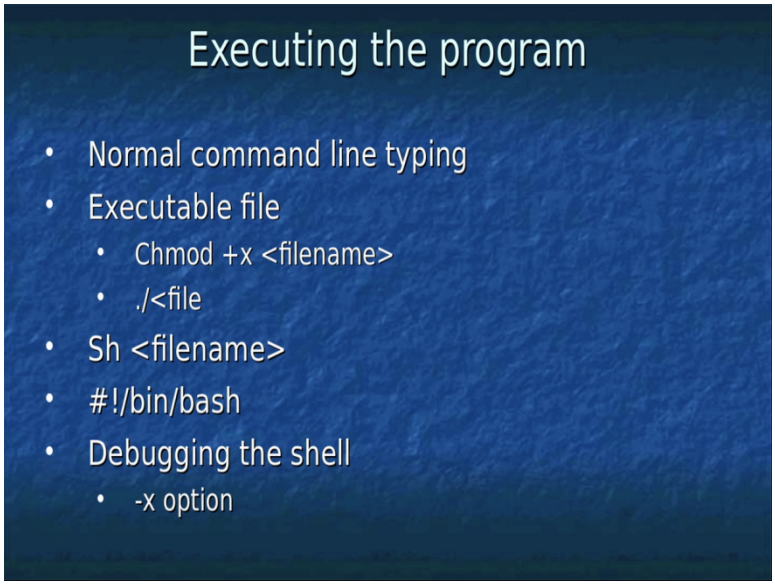


So what I can do is I can say shell I mean this is one way I can call the executive shell program we will see the other ways of doing it the first way that we can do is we can just write this shell and then I can say begin dot sh and you see that it is printing this is the first program, ok? So unlike with this time where we had actually accord it what we are done as we had put this echo statement inside a file called begin dot sh then we change the file to executable more so you can see that the file has been changed to executable mode and once I execute so I can execute in many ways the first way I am trying to execute is SH and it says begin dot sh ok?

(Refer Slide Time: 15:45)

So know that you know and you know how to write the first shell program and execute and start executing it ok?

(Refer Slide Time: 15:54)



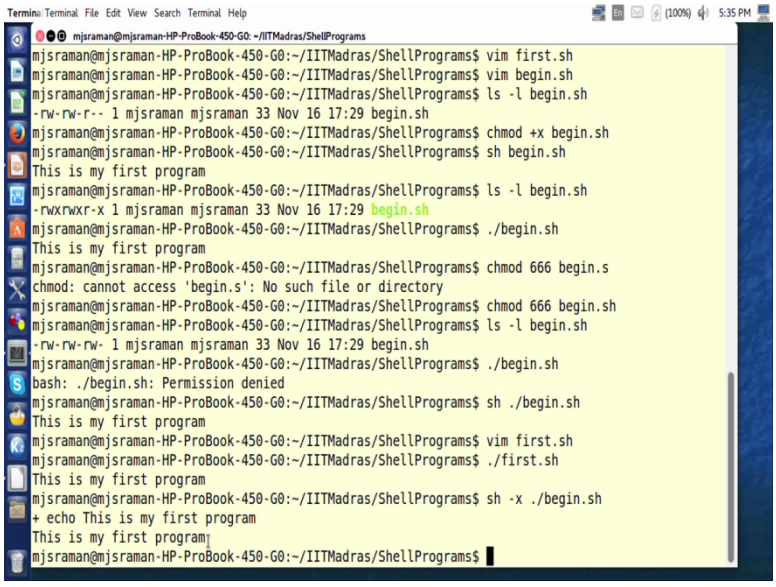
There are very many ways of executing the shell program okay the first way of executing the program as I had shown you is to type it in the Shell itself ok and then what we're done as we had executed it like this shell file name that also we have demonstrated we can then and I also told you that I had change the mode of the file to get executive information's .

Therefore what I can do is I can actually execute the file even like this because I had change the executable permissions ok let's see what happens if I remove the executable permission ok so suppose I made so let us say I made this permission now I try to execute this file begin dot s h it gives an error that permission denied then I can do is SH begin dot sh now the shell goes and executes so what you see is you can execute the shell after giving execute permission by typing in the command as it is so here is the example.

So if the program already has executable permissions like this then you can run the program by using this big in.ac the program does not have executable permission like this then you to give permission denied error if the program does not have executable permission still you can run the program by using this shell begin dot s.

So now we have seen first way of execution the second way of execution by converting it as an executable file and the Third way of execution there is yet another way of executing the shell program, this what you do is now whatever shell you are planning to execute the program on say for example in our case we're trying to put the Bourne shell Bourne Again shell be Bash, so this should be come if this becomes first line of your code.

(Refer Slide Time: 18:01)



So let me show with an example what I am talking about me vim let us say first dot sh if you look at this, this is the first line of the code and of the shell script and you should be very careful because this is should be typed as it is now this is true for even if you are going to invoke a program like Python, or pearl or whatever it is.

So this first line tells you that you are going to use this Bash shell to execute the programs or the commands that are for going to follow in the file so this is one way of doing it so once you do it what I can do is if you look at this I run this first dot sh now this is automatically be executed because the first line says that you had execute this, now there are some minor differences in this execution with respect to process ID and things like that we will be look at it later, but these are four ways of executing a shell program.

Now the next question that arises to us is so how do I debug a shell program? ok so what we can do is the easy way of debugging is as I told you if I put s h you just add minus x option and if I put this program say begin dot s h and hope you know that if I put a dot it means search from the current directory. So what happens is it puts a plus mark which tells you this is called tracing the program so this tells you that how you can trace the program so this tells you as soon as this program starts executing this line is executed and because this line is executed I am able to see this this is my first program.

I hope you enjoy this first session of knowing about what is a shell script about Born Again shell and then how to write a small Shell Script and then how to execute a shell script as well as debug the shell script we will now move on to other topics from the next session.