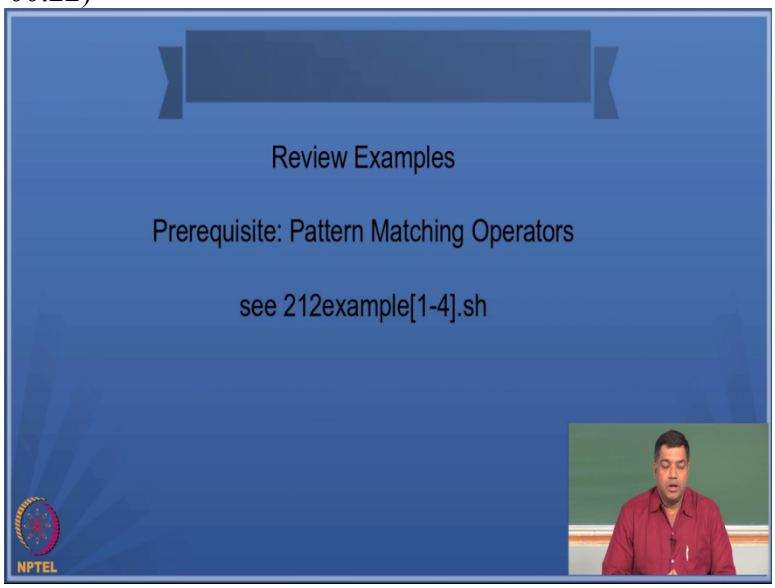
Information Security 3
Sri M J Shankar Raman,
Consultant Department of Computer Science and Engineering,
Indian Institute of Technology Madras
Module 43
Shell Examples 2

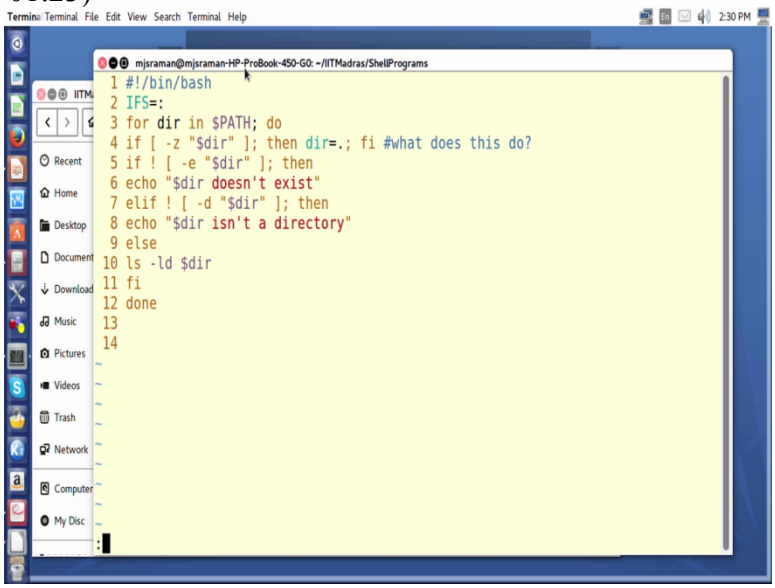
Welcome to this session on shell programming, in this session we will take examples and then

(Refer Slide Time: 00:22)



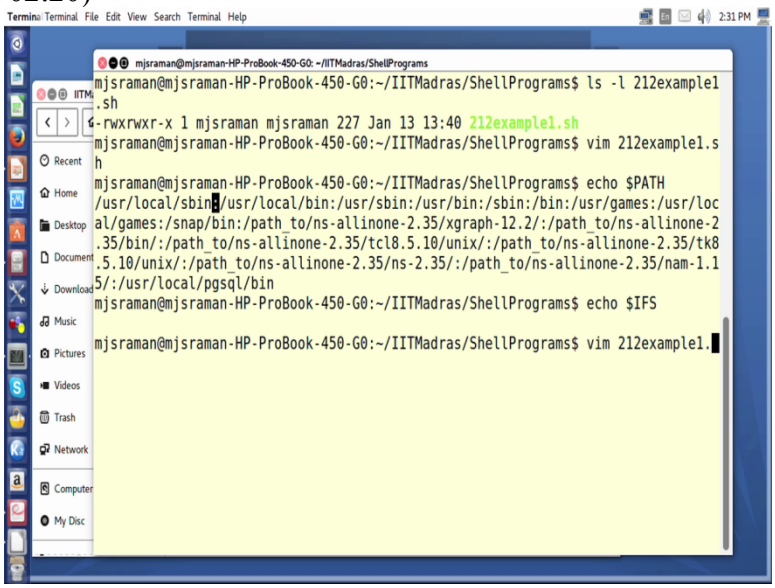
Review our knowledge of shell scripting. Since we are coming moral is coming to the end of the codes, one of the problems that you might be facing is that there are some kinds of new syntaxes that appears in shell scripts and there are certain syntaxes for which I am not able to understand what it means by reading the code, etc. So we will try to identify if such things happen how can we understand shell scripting just by execution, so by execution is one of the ways by which we can understand and if you see the use of that minus x that we discussed in the initial few classes will actually help you understand how the shell script gets executed. So in this regard we will review examples so we'll see in two or three areas ok, some examples of shell scripts and this will help you to get gain confidence into understanding the syntax of shell script. So let us start with the first example, ok?

(Refer Slide Time: 01:23)



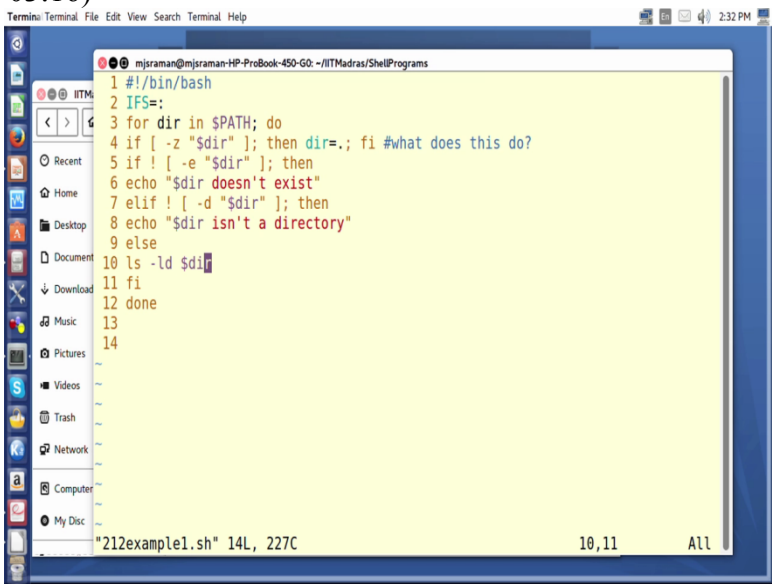
So let's start with the first example, we'll try to understand the shell scripts so my session is that whenever I edit the file ok please go ahead and try to understand what this file will do and then let's execute the programs and then see what the file does, ok? So here is the first shell script, now you see that this mostly makes use of the if statement and it has got something like ifs and and something like that so we'll try to decipher what this is all about. You do not need to understand such programs) the first thing that you've to do is so you've to look at what are all the variables. So if you look at this there are two variables that this shell script is using, the first is the dollar path. And the second is the ifs, so let's try to understand what this dollar path does and then what this ifs does ok,

(Refer Slide Time: 02:20)



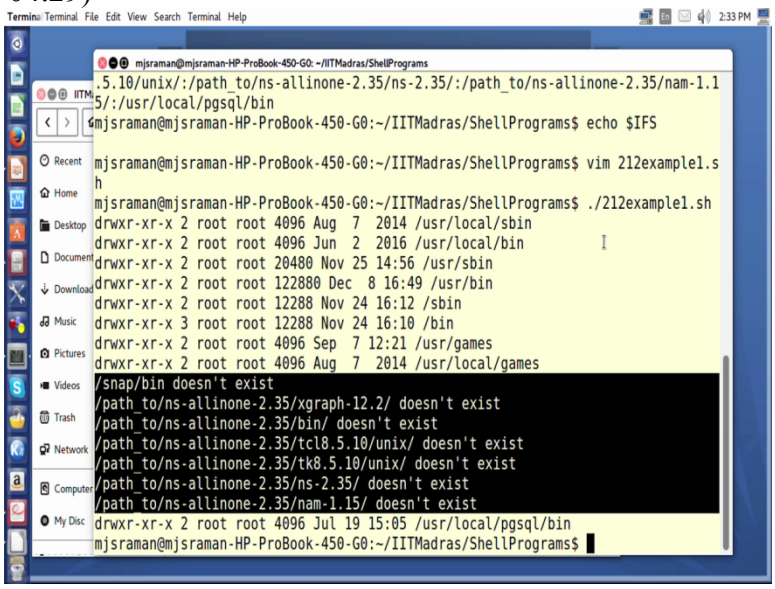
So let us go and echo these two echo dollar path ok and it prints you the path where you've to search for an executable and if you look at this, so this has something known as a colon. This colon separates two directories so for example slash user local sbin is one directory and slash user local bin is another directory and these two are separated by a colon, so ifs actually means the internal field separator. So this is actually a predefined variable so if you put echo ifs it's actually turns out to be null ok, so what we will do is we will just edit this program and then now we will try to understand what exactly the program will do, ok?

(Refer Slide Time: 03:16)



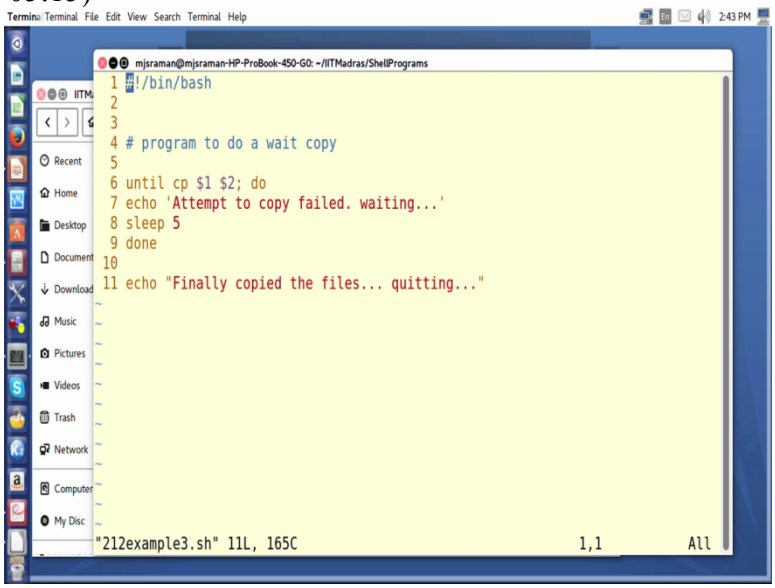
And then later we will go ahead and execute the program and see, so if you look at the first second line it tells you that the internal field separator is colon and now we know why they are using this internal field separator, so what we are doing is that first they are so if you put for dir in dollar path ok, what happens is that the whole string of dollar path gets assigned to dir and then it is followed by an internal field separator, so in this case for example so then it tries to find out whether this directory path is empty ok, if it is empty than you put that, you consider that current directory path, if the directory exits ok, then you say that the directory exits, if the directory does not exist ok it say that it is not a directory, if it's not a directory then you clearly say, so if you remember the use of minus d, minus e and minus z ok, and if you look at this then it says it's not a directory if the directory exits ok, then it actually list that directory, so that is what this program will do,

(Refer Slide Time: 04:29)



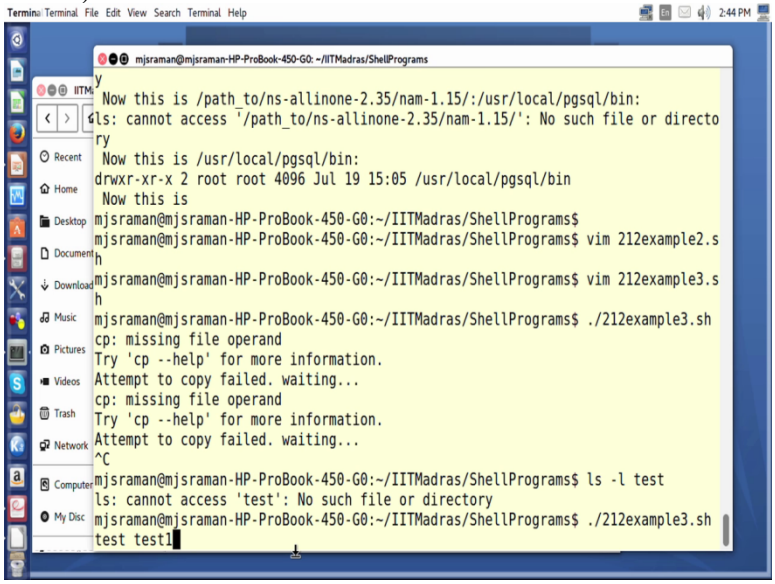
So let's run this program so if you look at this we will go ahead and execute this 1 2 example 1.sh, So if you look at this so it tells you that these are all the I mean if you if you remember the path variable consisted of these files sbin user local bin, user local bin and user local sbin. And then if you see in my directory for example, these files do not exist, ok? So even though these are path of my dollar path ok these files do not exist. So this example actually tries to identify those directories that exist and those directories that does not exist in the dollar path variable ok, so this is one way of writing this program, ok.

(Refer Slide Time: 05:15)



And this is a different kind of a code, so in this code ok this program will go on running running running until you give the file to copy. Now look at the logic I mean it just says that the program to do a wait copy and then so I say until cp dollar1 dollar colons dollar2 do and then go to sleep, so now what should I, how does this program work. So what I do is I just run this program again, so I'll run it in the background ok?

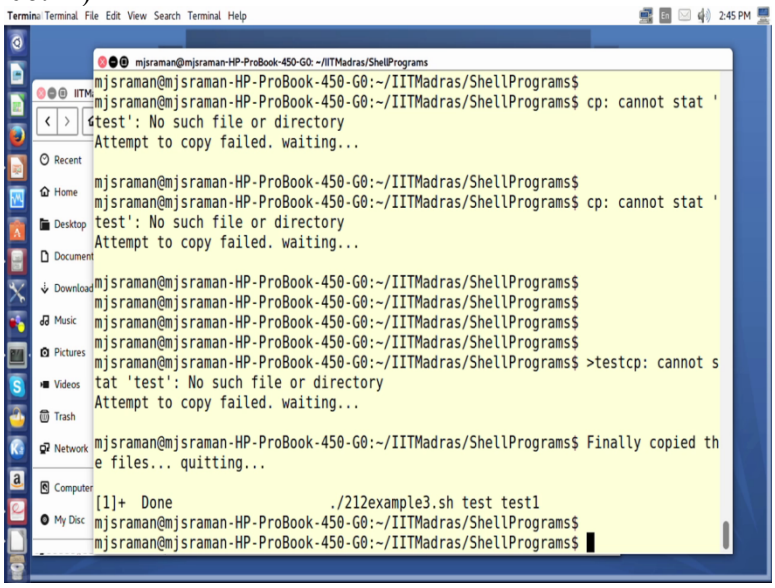
(Refer Slide Time: 05:51)



Because it's much more easier to understand so what I'll do is, I will do example3.sh, so essentially what this program is doing is so it says missing file operand ok, so so I have to give ok the files ok so I'll have to give say test and copy it to test1, so that is what the program is

expecting, now before that let me just show you that there is no program called test ls minus l test, there is no program called test, so what I do is now I try to give this example test and test1 ok, so I am trying to copy a file called test to test, now the test is still not it generated ok, so I'll just run it in the background for ease.

(Refer Slide Time: 06:44)

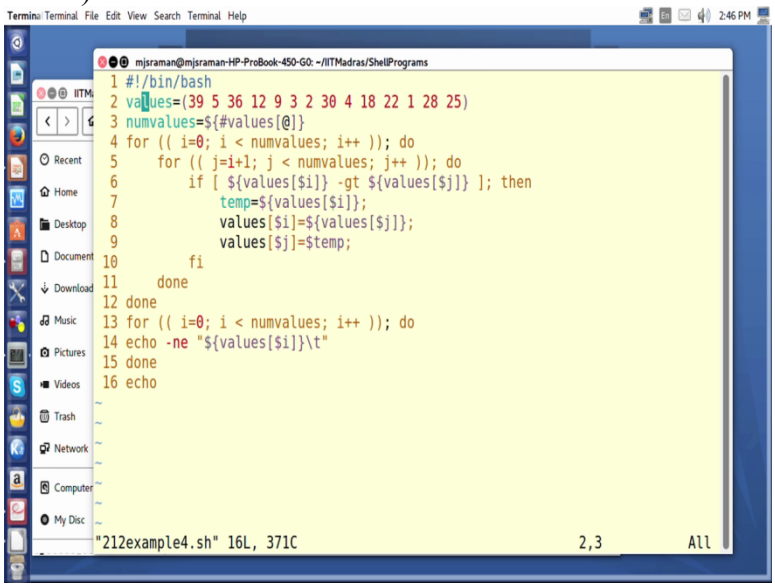


So now it says that so there is no such file so I attempted to copy failed and therefore it is waiting ok, so if you look at this it the program is running in the background waiting for the first file test file to get generated, now the test file is not getting generated. So what I'll do is now, I'll try to create the test file, this is one we have creating the test file, so once I create the test file the program sleeps and then it finally found out that the test file was generated therefore it just copied the test file and then it quit.

So it's a very simple program ok which actually it automates the whole process, see sometimes what happens is you want to see whether a log file was generated, so what you do every two minutes you just open the terminal see ls minus l whether the log file was generated, instead of that you can write a very small shell script like this that whenever log file is generated ok, sent me a message or copy somewhere etc, and this is one small example of how this job of creating waiting for a file can be reduced by using this kind of a technique, I mean this is these are ways of solving a problem, I hope I mean this is the reason we just give you these examples and finally we've seen one pattern matching example we had seen one example of how to manipulate the files etc, and then finally we will see how some of the mathematical algorithms can be

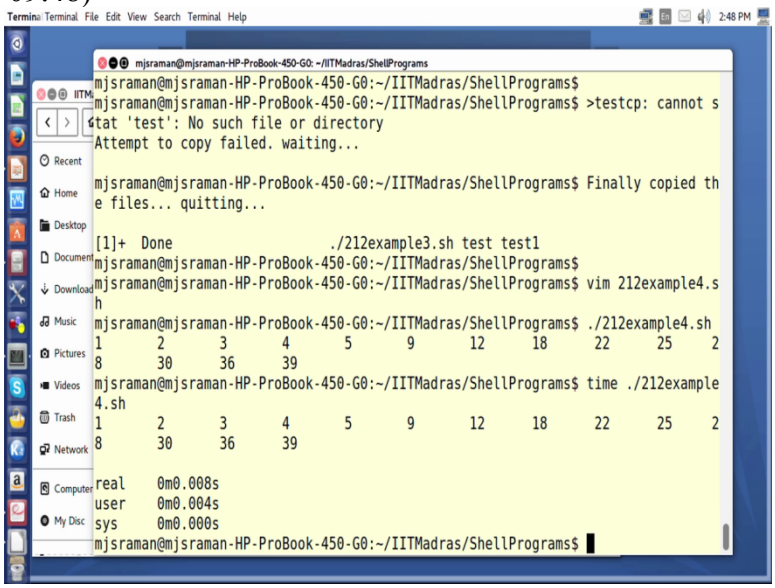
implemented in a shell script. So we'll look at the final example, the final example let's edit the file and then see what this example does, it's very simple example of a bubble sort algorithm ok, and you can make it efficient using the break statement etc.

(Refer Slide Time: 08:19)



So but then, so here is a bubble sort algorithm so and here we see I have declared an array of numbers ok and this is how you declare an array of numbers. So you can either declare, do the initialization something like that or you just declare an array declare minus a and then give the array name and then input the variables you can follow whatever method is useful or easy for you, so here is this way which is quite short ok, so I give an array of numbers and then I actually start I mean I just take the count of number of elements in the array so that can be done by this hash, hash gives you the count of numbers and then I put this count and then I this is if you see the next few lines if you are familiar with C programming or any other programming language, the code looks almost similar to what you write and see are other language, so this is the place where you try to exchange the value is greater, the first value is greater than the next value and so on, and then finally I print the arrays like this ok, (())(09:21) so if you look at this, I mean it is similar to C programming language ok, and the logic is something like this I mean if you had understood bubble sort and then you should know that the logic is very straight forward obviously you can optimize the code if there are no exchanges and things like that but we are not optimizing we just trying to show that even this kind of algorithms can be written using shell script. So shell scripts are extremely powerful so let's try to run this program and then see how it works.

(Refer Slide Time: 09:48)



So, sh so it essentially now sorts these numbers so I mean don't think that it's 2 and 8, it's actually 28, so it just sorts the numbers so we had given an array in a in a unsorted order and you run this program and then you can even time how much time this program takes ok so this could be of help to you to calculate, let's say I mean you want to see whether shell scripting is fast or slow I mean you can give increase the number of where numbers and then you can try to compare it if you know C or any other programming language, you can try to compare it with shell script and you'll see that shell script is slightly slower.

So that's why one of the disadvantage of shell scripts so it could run slightly slower but it's very flexible and it can automate a lot of your effort can be reduced, now even though we've talked about all these things there are certain things that you need to do as a homework or study ok, one of the first things that you need to do is you should be very familiar with pattern matching in shell scripts. Ok there, you you can use tools like grep and utilities that are there in shell, but finally if in the area of network security or in the area of security finally if you want to analyze ok you need to do lots of pattern matching's, so pattern matching with shell script is slightly easy and it has lot of tools so like arc and said and grep and all those tools so all these tools should be used in combination inside the shell scripting, so that will make shell scripting much more powerful. Thank you very much.