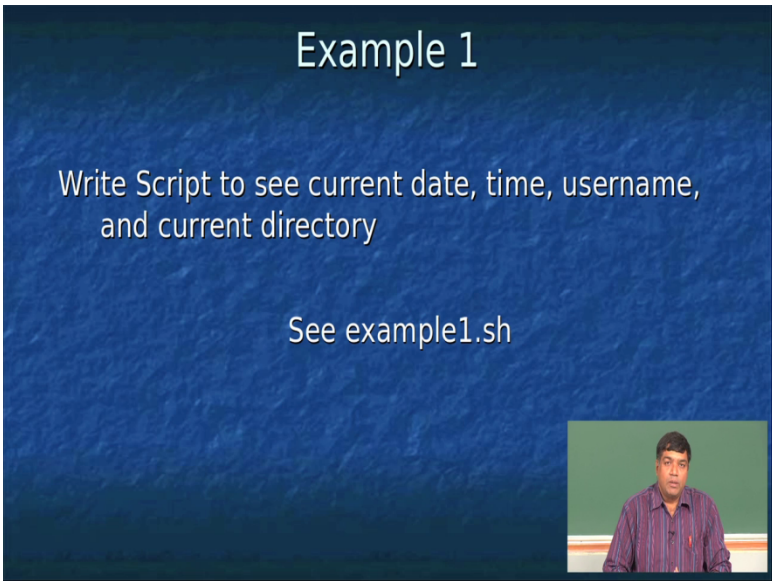
Information Security 3

Sri M J Shankar Raman,
Consultant of Computer Science and Engineering
Indian Institute of Technology Madras

Module 32
Shell Examples

Hi welcome to this session on shell scripting. Until now we had seen about basic shell commands we had also seen some of the shell constructs which includes the shell variables we had seen how relational operators, arithmetic operators and Boolean operators work. In this session we will see some examples, so that we understand how to use all these operators in a program. We also see a part of the if statement and what we will do it will combine all these things together to do some programming using shell.

(Refer Slide Time: 01:00)



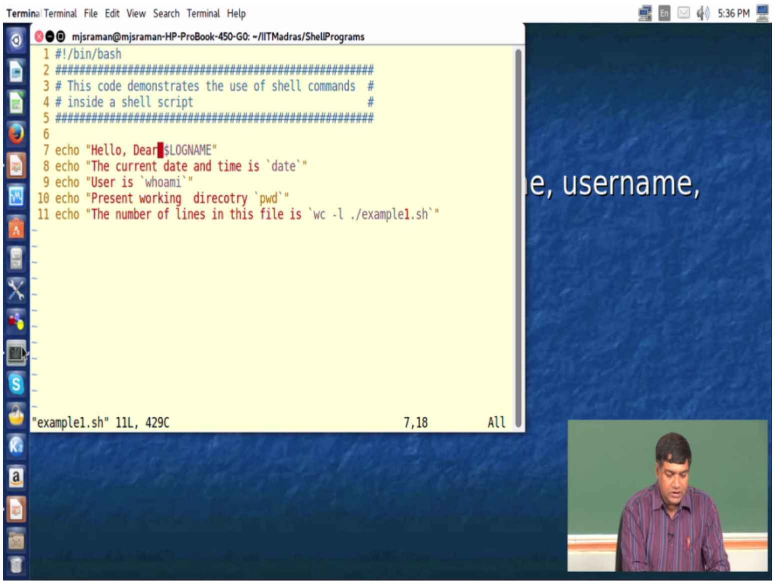
Example 1

Write Script to see current date, time, username, and current directory

See example1.sh

So let's start with some example usually we will be complete a task using the Shell scripting. So if you look at the first example what we need to do is we have to write a script that shows the current date, time, user name and the current working directory of the user. So how do you go about writing the script?

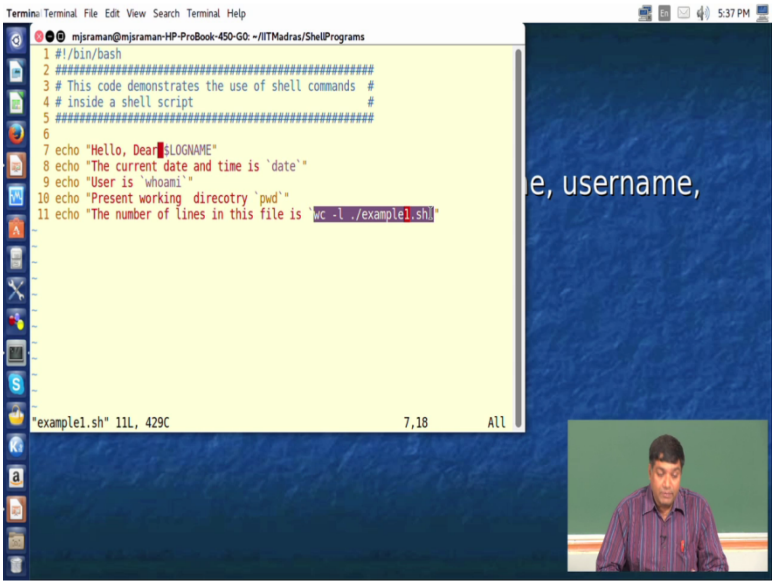
(Refer Slide Time: 01:22)


Terminal Terminal File Edit View Search Terminal Help
5:36 PM
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
e, username,

```
 1 #!/bin/bash
 2 #####################################################
 3 # This code demonstrates the use of shell commands  #
 4 # inside a shell script                             #
 5 #####################################################
 6
 7 echo "Hello, Dear $LOGNAME"
 8 echo "The current date and time is `date`"
 9 echo "User is `whoami`"
10 echo "Present working  direcotry `pwd`"
11 echo "The number of lines in this file is `wc -l ./example1.sh`"
~
"example1.sh" 11L, 429C                              7,18          All
```

Let us take a look at the example 1 so is this file example 1 that we have. So I will edit this file example1 dot sh and if you look at this file the first thing you should you note is how the code is commented. So it is mandatory that you comment whatever you right now because understanding Shell scripting can at times become very complicated. One due to syntax and other due to the logic that you follow. So in this quote the very first very first line invokes the bash shell and if you see that line numbers 2 3 4 and 5 or the comments that we want to write for this code and if you look at line number 7 we have used the echo statement and we have used one of the shell variables called Log name, ok?

(Refer Slide Time: 02:20)


Terminal Terminal File Edit View Search Terminal Help
5:37 PM
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
e, username,

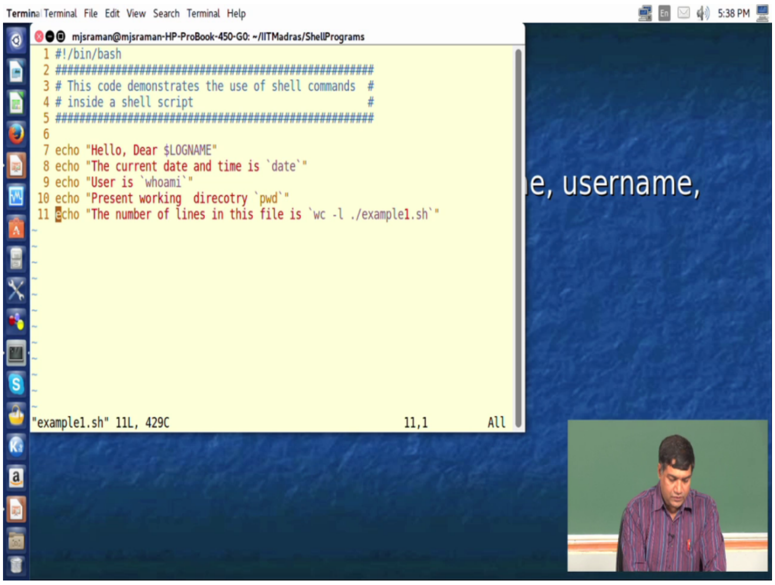
```
 1 #!/bin/bash
 2 #####################################################
 3 # This code demonstrates the use of shell commands  #
 4 # inside a shell script                             #
 5 #####################################################
 6
 7 echo "Hello, Dear $LOGNAME"
 8 echo "The current date and time is `date`"
 9 echo "User is `whoami`"
10 echo "Present working  direcotry `pwd`"
11 echo "The number of lines in this file is `wc -l ./example1.sh`"
~
"example1.sh" 11L, 429C                              7,18          All
```

So this will answer your question on journal how to current date, time, the username. So this will the log name actually the shell variable log name contains the username and therefore the first line will print the username. And then if you see the line number 8 if you see this it says that the current date and time is and then you have the date command and the date command is presented within the back code.

The back code tells us that the output of the date command will be printed in this case whatever is the current date and as well as the time. The line number 9 actually tells you who am I? Actually it tells you who is the username? ok? And line number 10 actually tells you the current working directory of the user again if you see we had put the back code which tells you that the output of the PWD command should be printed by the echo statement. And finally we can also see that we can include other shell commands within this shell scripts so if we look at this WC is the actually counts the number of lines in this program.

So in this case what we are doing is we are counting the number of lines in the current program itself the current program is the program that we are going to run right now.

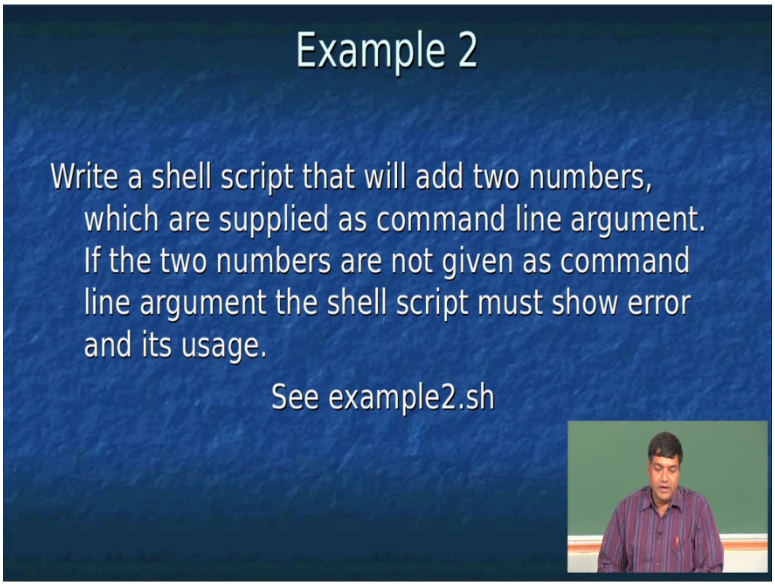
(Refer Slide Time: 03:36)


Terminal Terminal File Edit View Search Terminal Help
5:38 PM
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
e, username,

```
 1 #!/bin/bash
 2 #####################################################
 3 # This code demonstrates the use of shell commands  #
 4 # inside a shell script                             #
 5 #####################################################
 6
 7 echo "Hello, Dear $LOGNAME"
 8 echo "The current date and time is `date`"
 9 echo "User is `whoami`"
10 echo "Present working  direcotry `pwd`"
11 echo "The number of lines in this file is `wc -l ./example1.sh`"
~
"example1.sh" 11L, 429C                              11,1          All
```

So once I run this program you can see that it prints the log name the person who is logged in and then the current date and the time and if you see here it is the output of the date command that is getting printed and then the user name is m j s Raman,, and then it tells you the current directory where I am having these programs, ok? And the last lines say that the number of lines in the file is 11. So WC commands actually tells you how much you can do the word count, you can do the number of characters etc. using this program this command shell command. And in our case we have given WC space minus L which tells you the number of lines in the program it tells you that examples dot sh has about 11 lines. So this is a very small program that tells you ok? How to write the shell script a small shell script to just get some information and this makes use of the shell variables to display the values.

(Refer Slide Time: 04:45)



Example 2

Write a shell script that will add two numbers, which are supplied as command line argument. If the two numbers are not given as command line argument the shell script must show error and its usage.

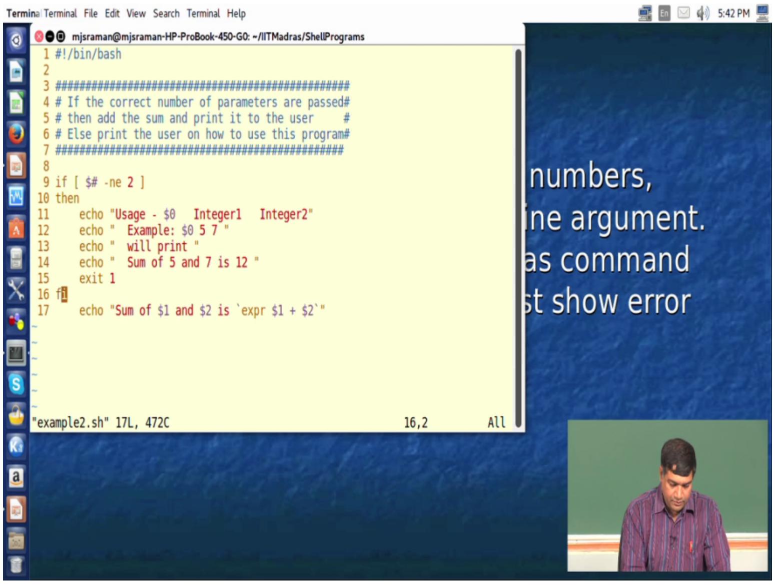
See example2.sh

Now let us go to the next program. Ok in this program what we will try to do is we will add two numbers and we will make use of the command line arguments ok? And if the two numbers are not given in the command line the shell scripts must throw out an error regarding its usage. This is a convention that is followed when you are writing any kind of scripts if the usage the way to use the scripts must be told to the user, how it can be done? I mean there are many ways in UNIX as you would know how you could create man pages or you could give a info command etc but when you are writing a shell script it is convention that if the user does not type the parameters properly you are supposed to tell the user how to use the command, ok? So let us go and take a look at this example to see how this can be accomplished?

So we have to accomplish two tasks. The first task is that if the user does not give the command properly or does not invoke the current shell script properly we should throw out an error and tell the user how to use the command. If the user gives all the commands properly or gives the input parameters properly then we should be able to add the two numbers and print the result. So this process of validating the inputs is very important in not only shell scripting but also in any programming language.

So usually it is referred to as garbage in garbage out. And if you give garbage inand then the computer is going to throw garbage out therefore its always better that you validate the parameters that are given for a function or a program. Not only shell script but even if you are going to code in a C programming language. Now let us take a look at this example shell script, ok? We will invoke this example we will first understand the example shell script and then we will run it.

(Refer Slide Time: 06:36)



Terminal Terminal File Edit View Search Terminal Help
5:42 PM
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
numbers,
ine argument.
as command
st show error

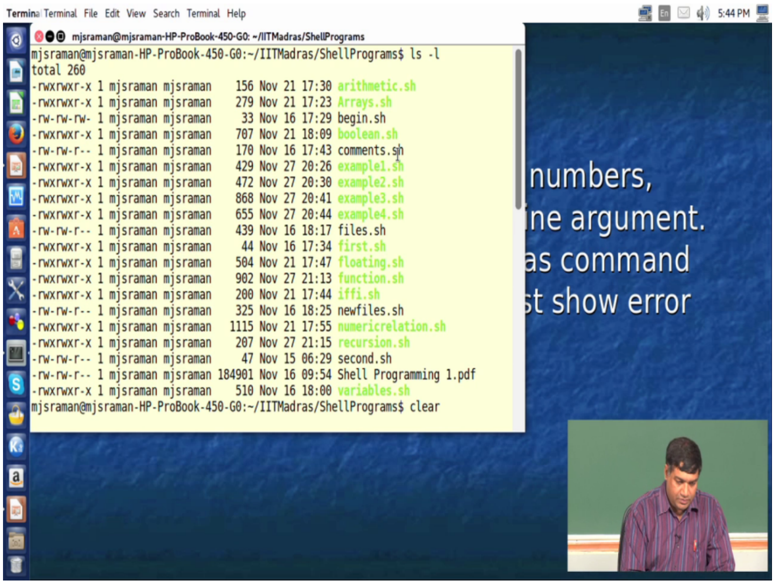
```
 1 #!/bin/bash
 2
 3 #################################################
 4 # If the correct number of parameters are passed#
 5 # then add the sum and print it to the user     #
 6 # Else print the user on how to use this program#
 7 ################################################
 8
 9 if [ $# -ne 2 ]
10 then
11     echo "Usage - $0   Integer1   Integer2"
12     echo "  Example: $0 5 7 "
13     echo "  will print "
14     echo "  Sum of 5 and 7 is 12 "
15     exit 1
16 fi
17     echo "Sum of $1 and $2 is `expr $1 + $2`"
"example2.sh" 17L, 472C                          16,2          All
```

So as usual we are commenting the code and so take this line I mean line number 9 if you remember we are told that dollar hash represents the number of parameters ok that is going to be input to the shell script, ok? And if the number of parameters is not equal to 2 then it tells you how to use this command, ok? So usually you put these usages then the command name dollar 0 refers to the command name and then you have to give integer 1 and integer 2 for example I mean it is also better you give an example because it is slightly criptic so we can save command and then if you give number 5 and 7 then your program will print sum of 5 and 7 is 12.

And then you should exit remember when you write the code properly or when the code execute properly you are supposed to return exit of 0 and if there is any error in the code or any error in the input or program does not terminate properly because of some issue. Then you should always assign a positive value so in this case if you looked at it I have assigned a value of 1.

Now after this if the input gets validated let us say that you have passed the proper parameters then what I do is I just use the expression command ok and then calculate dollar sum of dollar 1 and dollar 2 where dollar 1 refers to the first parameter which corresponds to integer 1 and dollar 2 refers to the second parameter that corresponds to integer 2. Obviously this command takes only integers and I hope you remember that in order to calculate floating point we have to use some other command in the Shell shell script. So let us try to now execute this command.

(Refer Slide Time: 08:22)



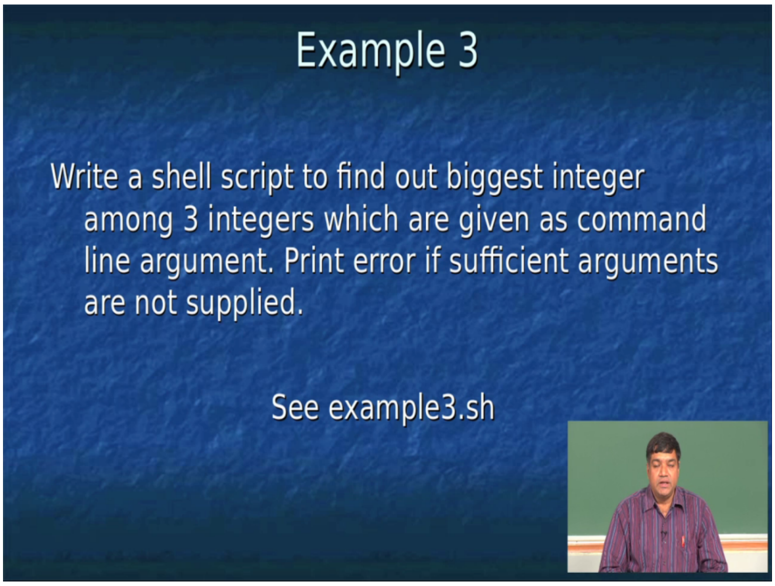
Terminal Terminal File Edit View Search Terminal Help
5:44 PM
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
numbers,
ne argument.
as command
st show error

```
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ls -l
total 260
-rwxrwxr-x 1 mjsraman mjsraman    156 Nov 21 17:30 arithmetic.sh
-rwxrwxr-x 1 mjsraman mjsraman    279 Nov 21 17:23 Arrays.sh
-rw-rw-rw- 1 mjsraman mjsraman     33 Nov 16 17:29 begin.sh
-rwxrwxr-x 1 mjsraman mjsraman    707 Nov 21 18:09 boolean.sh
-rw-rw-r-- 1 mjsraman mjsraman    170 Nov 16 17:43 comments.sh
-rwxrwxr-x 1 mjsraman mjsraman    429 Nov 27 20:26 example1.sh
-rwxrwxr-x 1 mjsraman mjsraman    472 Nov 27 20:30 example2.sh
-rwxrwxr-x 1 mjsraman mjsraman    868 Nov 27 20:41 example3.sh
-rwxrwxr-x 1 mjsraman mjsraman    655 Nov 27 20:44 example4.sh
-rw-rw-r-- 1 mjsraman mjsraman    439 Nov 16 18:17 files.sh
-rwxrwxr-x 1 mjsraman mjsraman     44 Nov 16 17:34 first.sh
-rwxrwxr-x 1 mjsraman mjsraman    504 Nov 21 17:47 floating.sh
-rwxrwxr-x 1 mjsraman mjsraman    902 Nov 27 21:13 function.sh
-rwxrwxr-x 1 mjsraman mjsraman    200 Nov 21 17:44 iffi.sh
-rw-rw-r-- 1 mjsraman mjsraman    325 Nov 16 18:25 newfiles.sh
-rwxrwxr-x 1 mjsraman mjsraman   1115 Nov 21 17:55 numericrelation.sh
-rwxrwxr-x 1 mjsraman mjsraman    207 Nov 27 21:15 recursion.sh
-rw-rw-r-- 1 mjsraman mjsraman     47 Nov 15 06:29 second.sh
-rw-rw-r-- 1 mjsraman mjsraman 184901 Nov 16 09:54 Shell Programming 1.pdf
-rwxrwxr-x 1 mjsraman mjsraman    510 Nov 16 18:00 variables.sh
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ clear
```

Now if you look at this whatever we had given hash 2 if you look at this hash 2 that means you don't give the number of parameters so I this command expect an integer 1 and integer 2 if it do not give integer 1 and integer 2 you see that we have to give it like this ok? So let us try to give the command says that how to print this.

So let us say I give five and I do not give the second input again it tells you that this is not the way to use a command and now I give 7 so now the command works properly saying that the number of inputs are sufficient to do the calculation. So we will also try some other number ok? So this works for all integer numbers and if you look at this we have learnt some good way to tell the users that you have to do validation input validation before you type any command any Shell Script ok!

(Refer Slide Time: 09:28)



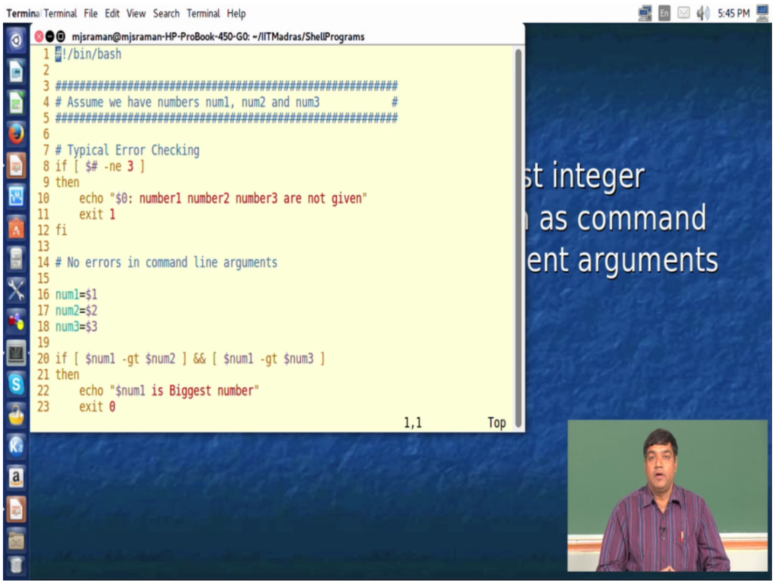
Example 3

Write a shell script to find out biggest integer among 3 integers which are given as command line argument. Print error if sufficient arguments are not supplied.

See example3.sh

Now moving on we will now go to slightly complicated example, now this example asks you to find out the biggest of the three integers which are given in the command line. So there are 2 things that we should do 1) Print the error message if sufficient arguments are not supplied and the second most important aspect is to prepare a logic for this program. When you do Shell scripting first of all we should understand that syntax of shell script is slightly difficult to follow. Second means other than that you need to also have a very solid logic when you hide programs. We will see that in this program what looks like a very simple program the logic is slightly complicated. So let us see how we will solve this problem first of all.

(Refer Slide Time: 10:25)

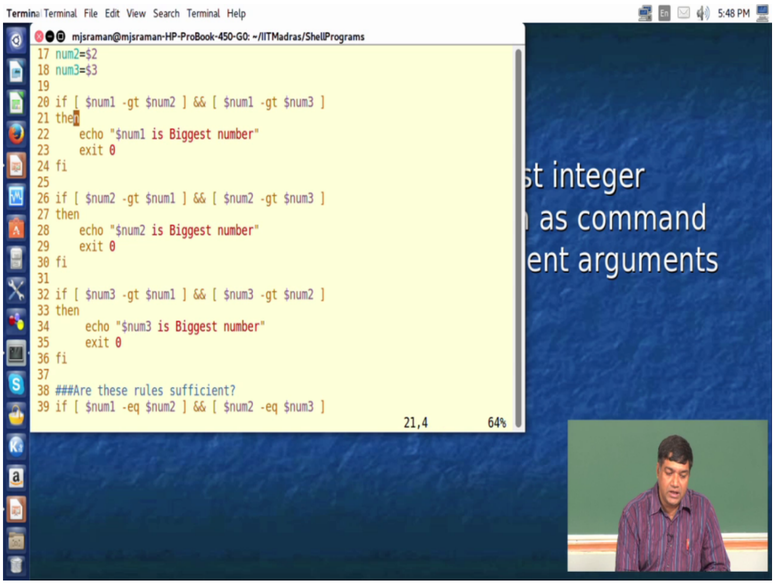


Termina Terminal File Edit View Search Terminal Help
5:45 PM
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
1,1 Top
st integer
as command
ent arguments

```
#!/bin/bash

##########################################################
# Assume we have numbers num1, num2 and num3             #
##########################################################

# Typical Error Checking
if [ $# -ne 3 ]
then
    echo "$0: number1 number2 number3 are not given"
    exit 1
fi

# No errors in command line arguments

num1=$1
num2=$2
num3=$3

if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]
then
    echo "$num1 is Biggest number"
    exit 0
```

Let us first take this example and see how we can identify the highest or the biggest of three numbers. Ok let us assume ok the algorithm that is given obviously this algorithm has an error so at the end of this you should try to find out what is the error and in case you are having your laptop please ensure that you debug this program and find out what is the right logic to be implemented. So in our case we are taken the slightly complicated example we tell you that not only your shell scripting is some technique is needed for writing a shell script you also need to identify a good logic to solve a problem

(Refer Slide Time: 11:00)



Termina Terminal File Edit View Search Terminal Help
5:48 PM
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
st integer
as command
ent arguments

```
17 num2=$2
18 num3=$3
19
20 if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]
21 then
22     echo "$num1 is Biggest number"
23     exit 0
24 fi
25
26 if [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]
27 then
28     echo "$num2 is Biggest number"
29     exit 0
30 fi
31
32 if [ $num3 -gt $num1 ] && [ $num3 -gt $num2 ]
33 then
34      echo "$num3 is Biggest number"
35      exit 0
36 fi
37
38 ###Are these rules sufficient?
39 if [ $num1 -eq $num2 ] && [ $num2 -eq $num3 ]
                                               21,4          64%
```

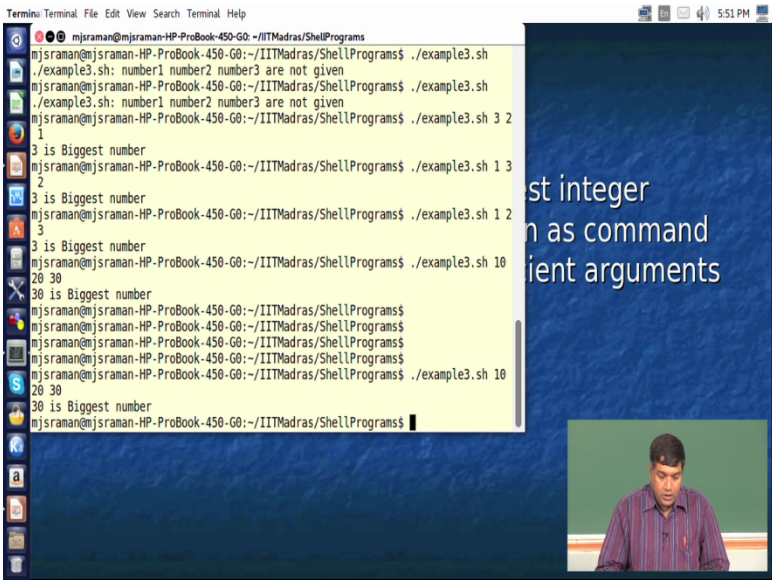
So let us assume that we have 3 numbers, ok? So as usual if the user does not give three numbers as arguments you have to give an error message probably you can also give the usage of this so in our case we have not given the usage of this. Now let us look at this logic, ok? So I take the first number as the first parameter as number 1 the second parameter as numbers 2 and the third parameter as number 3. Now let us go into the logic. So the logic says that if the first number is greater than the second number second number is greater than the sorry the first number is greater than the third number then we know that number one is the biggest number I think this is very straight forward.

Now the same logic can be applied to numbers 2 and 3 therefore if you look at line number 26 it says that if number 2 is Greater the number one and number 2 is greater than number 3 then we know that number 2 is the biggest of the numbers and if you find out the biggest of the numbers then I have to return it exit value of zero this is convention so because we have found out the biggest number. Similarly what would happen is the third number is a biggest of the other two therefore what we do is we do number 3 is greater than number one and number three is greater than number two. Remember you have to put an and condition because the given number must be greater than the other two numbers

Ok now having said a logic like this the question that we need to ask is? What happens if all the three number s are equal ok? OR if the two numbers are equal so let us see that if all the three numbers are equal then we can actually print it because we can say that if number 1 is equal to number 2 and number 2 is equal to number 3 ok? We say that all the numbers are equal and print now do you think this logic is sufficient ok? So we are trying to find out whether number 1 is equal to number 2 and number 2 is equal to number 3. So such questions you need to ask when you write your programs. It is not just you go and write your programs because such type of errors I mean if there is an error in this program it will slightly difficult to debug ok?

That is one of the reasons that we had given this program as an example and finally if I do not if I fail in any of these conditions because I am using an if condition, if I fail in any of these conditions then I say I am not smart enough to answer your question this is like giving up saying that look I cannot find out the maximum of three numbers but let us see whether this programs works first of all ok?

(Refer Slide Time: 14:16)



Terminal Terminal File Edit View Search Terminal Help
5:51 PM
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
st integer
n as command
ient arguments

```
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ./example3.sh
./example3.sh: number1 number2 number3 are not given
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ./example3.sh
./example3.sh: number1 number2 number3 are not given
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ./example3.sh 3 2
 1
3 is Biggest number
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ./example3.sh 1 3
 2
3 is Biggest number
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ./example3.sh 1 2
 3
3 is Biggest number
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ./example3.sh 10
20 30
30 is Biggest number
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ./example3.sh 10
20 30
30 is Biggest number
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ 
```

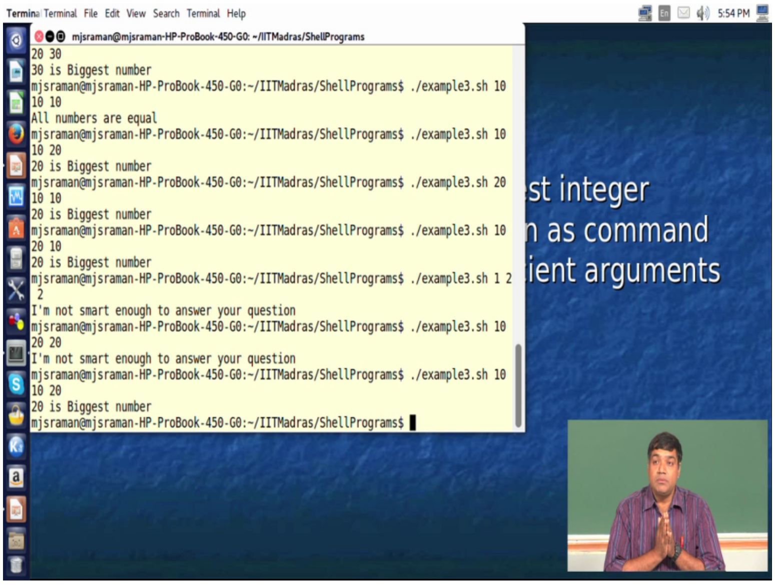
 So let us try to supply the inputs to the program ok? And if I do not supply inputs properly I mean this program will give an error out ok so now in this case you have to give the way you have to use it is number 1 number 2 and number 3 are not given so let us try to give the first case where I see number 1 ok is greater than the rest of the numbers so I can see example 3 dot sh I will give first number as 3 2 and 1 and program works correctly it says that 3 is the biggest of the numbers.

Let us type the second case I will give the inputs as 1 3 and 2 again this program works correctly and finds out that 3 is the largest of the numbers. Let us try the third case 1 2 and 3 and this program still says that 3 is the biggest of the number probably if you still have any confusion let us say if I put 10 20 and 30 so it says 30 is the biggest number. So let us now look at so we executed example 3 with three numbers the first number the third number is now the greatest of the three numbers and it works correctly if you remember these are the first three statements that we had design.

Now what we will go ahead and do is we will give equal numbers to see whether our logic of equal numbers works correctly so I give you numbers 10, 10 and 10 and it says all numbers are equal and it works correctly now this process of giving the correct inputs ok? And then testing the logic of your program is a part of unit testing.

So whenever you write a shells script we are also supposed to do a small portion of unit testing to see that your code works now unit testing not only consists of positive cases but it should also test one or two negative cases. So until now we have followed the logic of the program and then given the inputs. What we will do right now is we will now try to give inputs which does not follow the logic of the program. So let us try to see whether if we give such inputs the program works correctly the shell script whether it works correctly.

(Refer Slide Time: 16:42)



Terminal Terminal File Edit View Search Terminal Help
5:54 PM
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms

```
20 30
30 is Biggest number
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ./example3.sh 10
10 10
All numbers are equal
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ./example3.sh 10
10 20
20 is Biggest number
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ./example3.sh 20
10 10
20 is Biggest number
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ./example3.sh 10
20 10
20 is Biggest number
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ./example3.sh 1 2
 2
I'm not smart enough to answer your question
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ./example3.sh 10
20 20
I'm not smart enough to answer your question
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ./example3.sh 10
10 20
20 is Biggest number
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ 
```

st integer
n as command
ient arguments

So let us try this Example 3 dot sh I will give 10 10 and 20 wow this works correctly and we assume that it is going to work correctly for the rest of the inputs so what we will try to do is we will now try to make the last two numbers equal so what I will try to do is I will try to make 20 other I will give 10 and 10 and again this program finds out that 20 is the biggest number. So do you think that the logic is correct? ok! Let us see give some other inputs so let us start giving an input like 10 20 and 10 and again this says 20 is the biggest number so do you think that after passing all these tests your program is right?

Let us try some more inputs so probably what we should do is? We should try can you think of any other combination other than this? Ok let me now try to get a combination of 1 2 and 2 and it is so surprising that your program says I am not smart enough to answer your question. Now the challenge for you is, can you find out this bug? So it is so surprising that when I give 10 20 20 ok it says let us try to give 10 20 20 ok and what could be the bug in this program 10 20 20 this says I am not smart enough to answer your question whereas when I give 10 10 20 it is able to find the biggest number. Now these type of errors are extremely difficult to detect so you should have a very solid logic when you write your program.

Now we leave this open for you to identify the bug in this program and how to fix the bug in that way you will be able to gain more and more experience in shell scripting as well as debugging and you must also understand that not only you should do scripting, you should also do minimal amount of testing and these test cases must be carefully designed such that it can identify the errors in your program. Please remember making a mistake while writing a shell program is normal is only abnormal if you can write your shell program work correctly in the first shot.

Thank you!