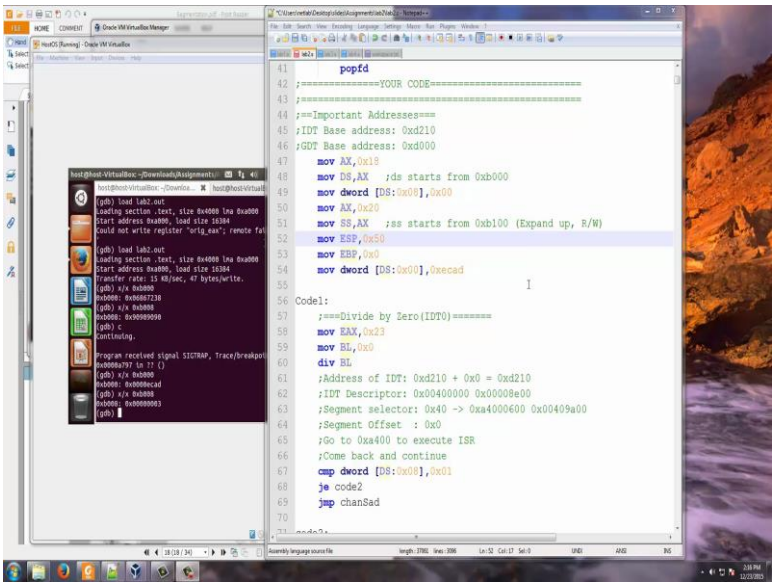
**Information Security - II**
**Prof. V. Kamakoti**
**Department of Computer Science and Engineering**
**Indian Institute of Technology, Madras**

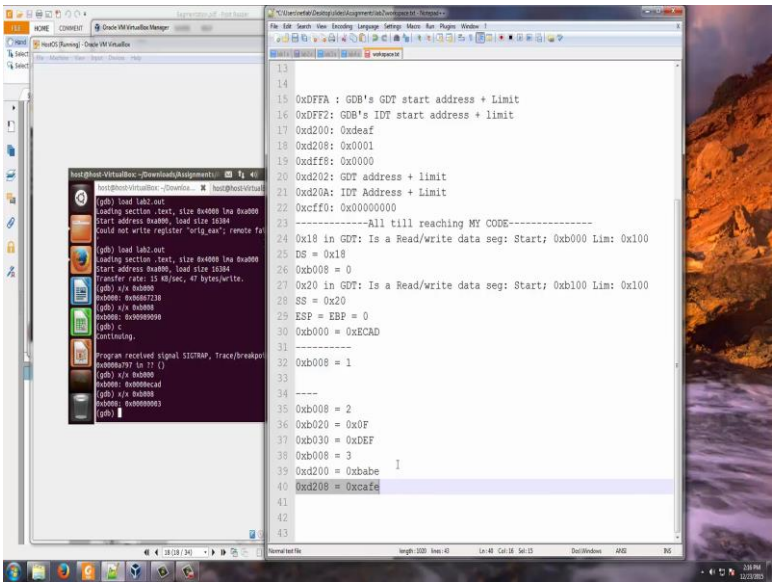
**Lecture - 26**
**Lab2 Part 2 - Week 5**

(Refer Slide Time: 00:09)



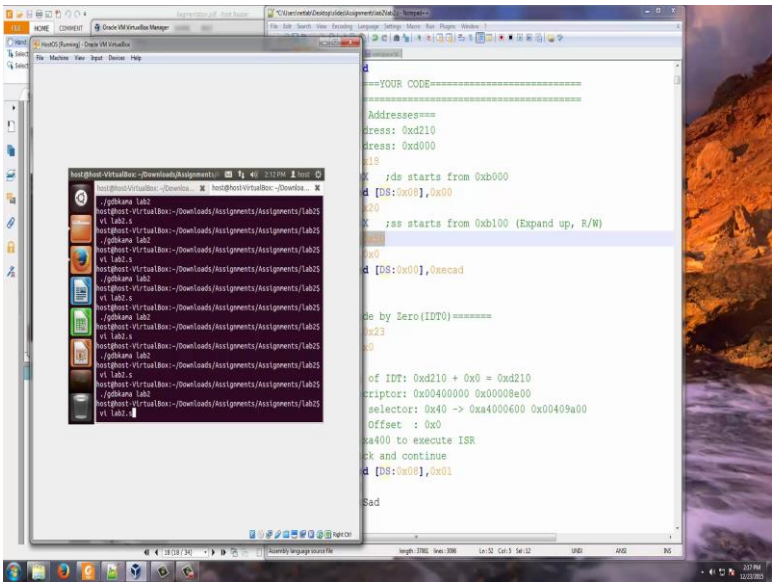

So, in the previous demo, we find that the code did not work.

(Refer Slide Time: 00:15)



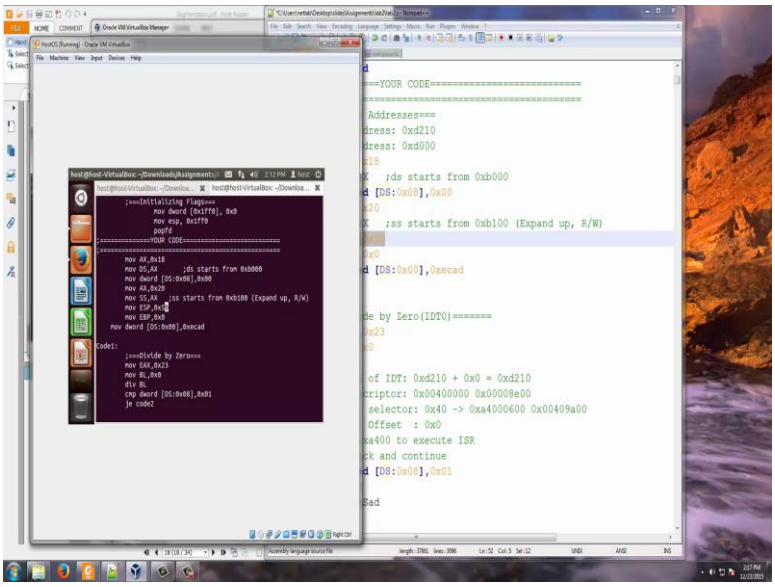

For example, if you have seen my b 1008 should have been 3 and my b 1000 should have been e cad and d200 as should be babe d208 should be c a f e all this did not happen. So, why it did not happen? So, let us go, just an exercise we had initialized stack pointer to 0 and it is a segment. When you try to push in then, if the stack pointer is 0 then, when you try to push, it basically gives us stack fault and that is why it was not able to and when we actually have an interrupt that stack is being pushed into and so essentially nothing could move in into the stack and that is the reason. Why the program did not work? Now, we go to line number 52 and make it 0 x 50.

(Refer Slide Time: 01:26)



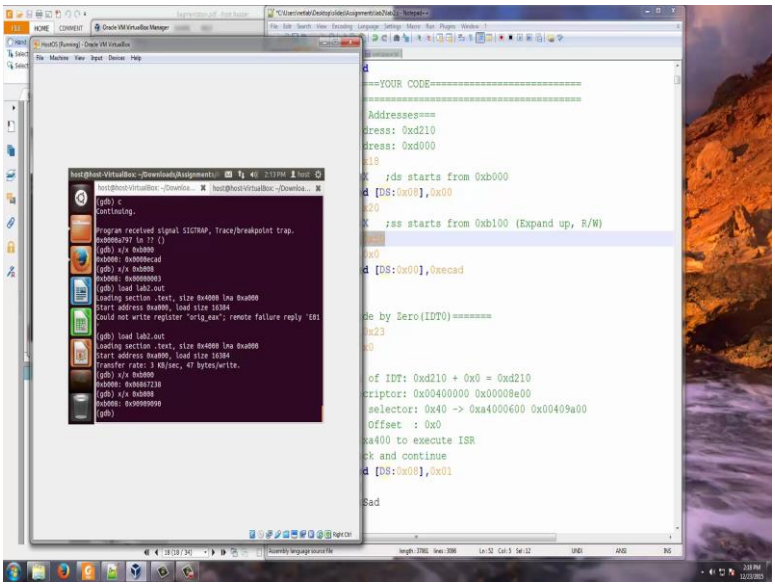
We will also see, in another window here, lab two dot s and go here and see that this is.

(Refer Slide Time: 01:40)



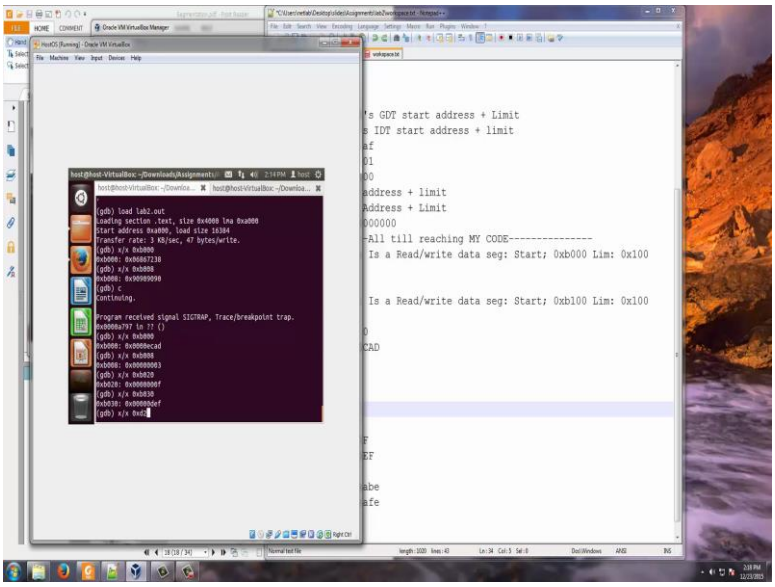
Go down to that particular line and you here see that your move sp comma 0 x 50. So, once you do this and compile this and then upload this here.

(Refer Slide Time: 01:57)



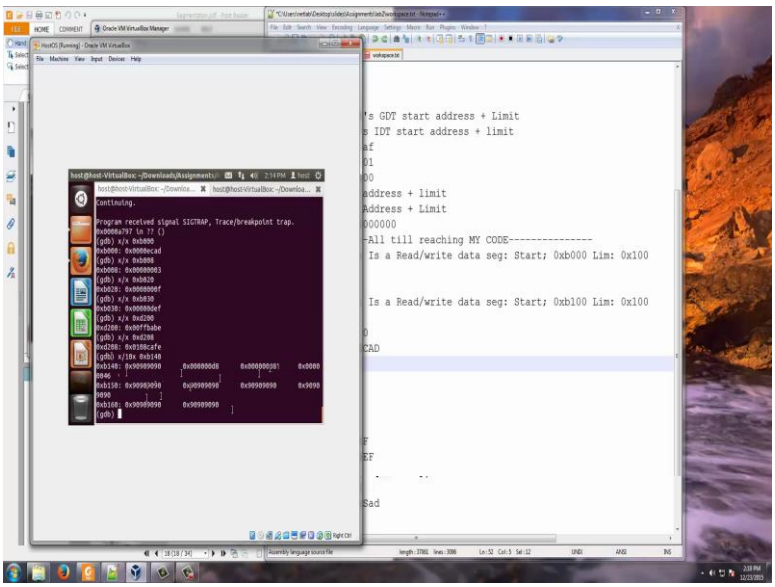

Now, let us go and check what is there in b 1000. There is some value and of course, all are null characters in b 1008.

(Refer Slide Time: 02:22)




Now, we continue, now you see what is there in b 1000, it is c cad b 1008 it is 3. That is, what we wanted and we will check the other values also as we see here b 1020, b 1020. That is been f 30 is d e f 8 is 3 we have already checked.

(Refer Slide Time: 03:05)



d200 is babe d208 cf. This basically worked and let us also now see where does this stack start. Essentially, the stack is starting at b 1100 and we are looking at a 50. So, let us see s slash 10 x, 0 x b 140. So, now, as you see here there as things that have been pushed here 46, 8, d 8. So, this was the code segment etcetera. So, the stack is now operational at v 150. So, the stack actually grows down. So, initially when we made esp as 0. It basically started growing down beyond b 1100 and essentially it became negative.

Now, I made esp as 50 it started working about 4 f and v 1 4 f. As you see here this is why the stack has been operational. So, what we have done in this assignment is we have created 3 faults, one is a hardware generated fault, divide by 0. The next one was an overflow; where there was an overflow created and then into instruction was executed to see the over flow and then there was software define in 31. The difference between the first and this last 2 was that when the interrupt service routine complete it is execution in the first case divided by it is 0. It came to the same instruction that created the fault. So, that we could actually rectify it but in the case of the other two it went it a next instruction that actually created the interrupt and this is very important to be noted.

So, interrupt service routines which can be potentially a big source of security vulnerability because, that actually causes multiple privilege changes as we will see in this subsequent assignment. But so, what we have done in this particular assignment is to basically demonstrate, how you write interrupt service routine. How you load the IDT's and how you configure this. So, what I suggest that is down the line, you please start generating more interrupts that you see in the Intel manual and then write some interesting interrupt service routines. Please note that the major part of a servicing device request comes from interrupt service routine. The device driver actually forms have measure component of the interrupt service routines. So, it is very important that you understand this particular assignment and I hope you have done so.

Thank you.