OHTS Lab 2 – Linux Shell Coding

To write the shell codes, knowledge of Assembly and C is needed. Also need to know how the stack works.

Next I google the website called as https://0x00sec.org/t/linux-shellcoding-part-1-0/289 , in order to go to that website in order to get the shell code.

Then, I started the Kali Linux in the Virtual Box, and typed as follows, which is shown in the screen shot (Figure 1) below.

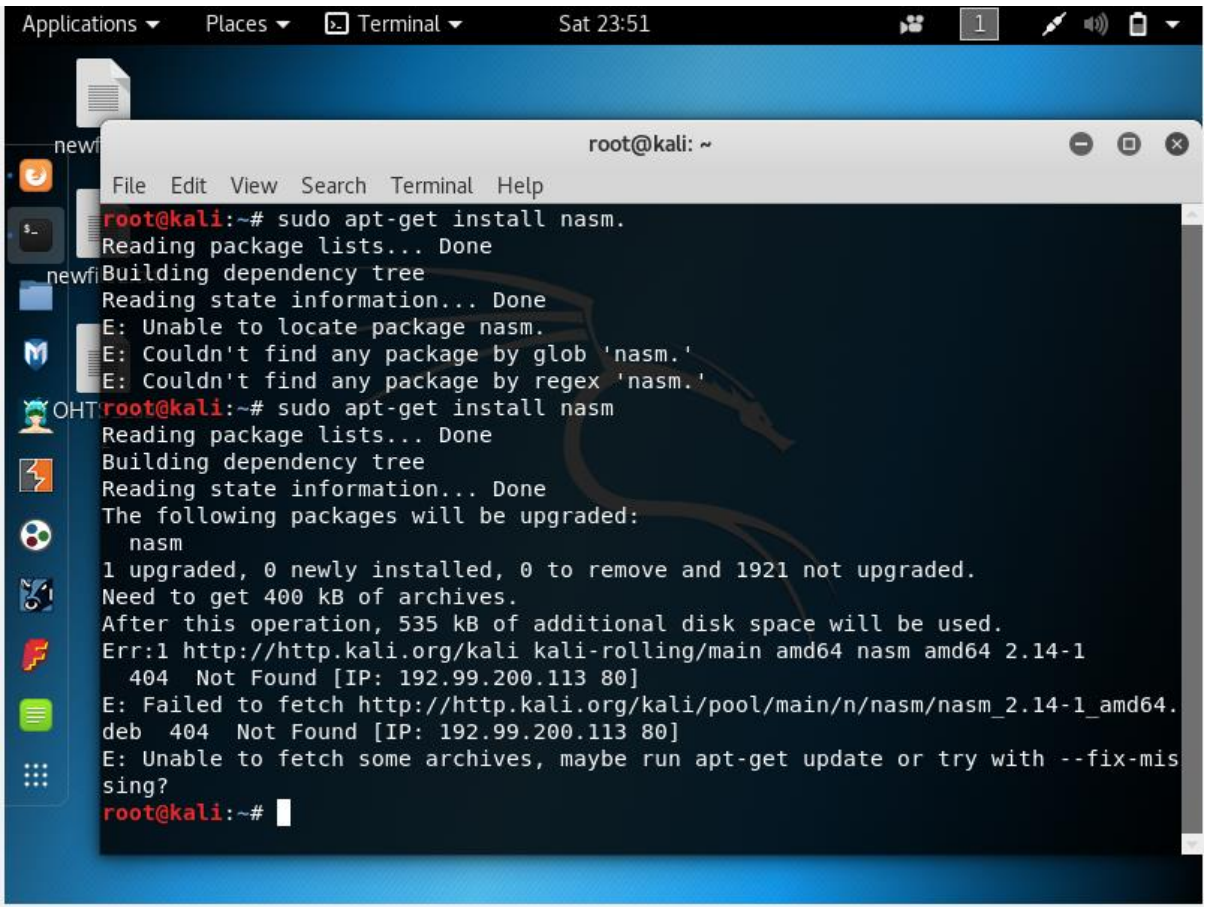


Figure 1: Install nasm

Then, I copy paste the assembly program in the text editor and save it as shell.asm. This is shown in (Figure 2).

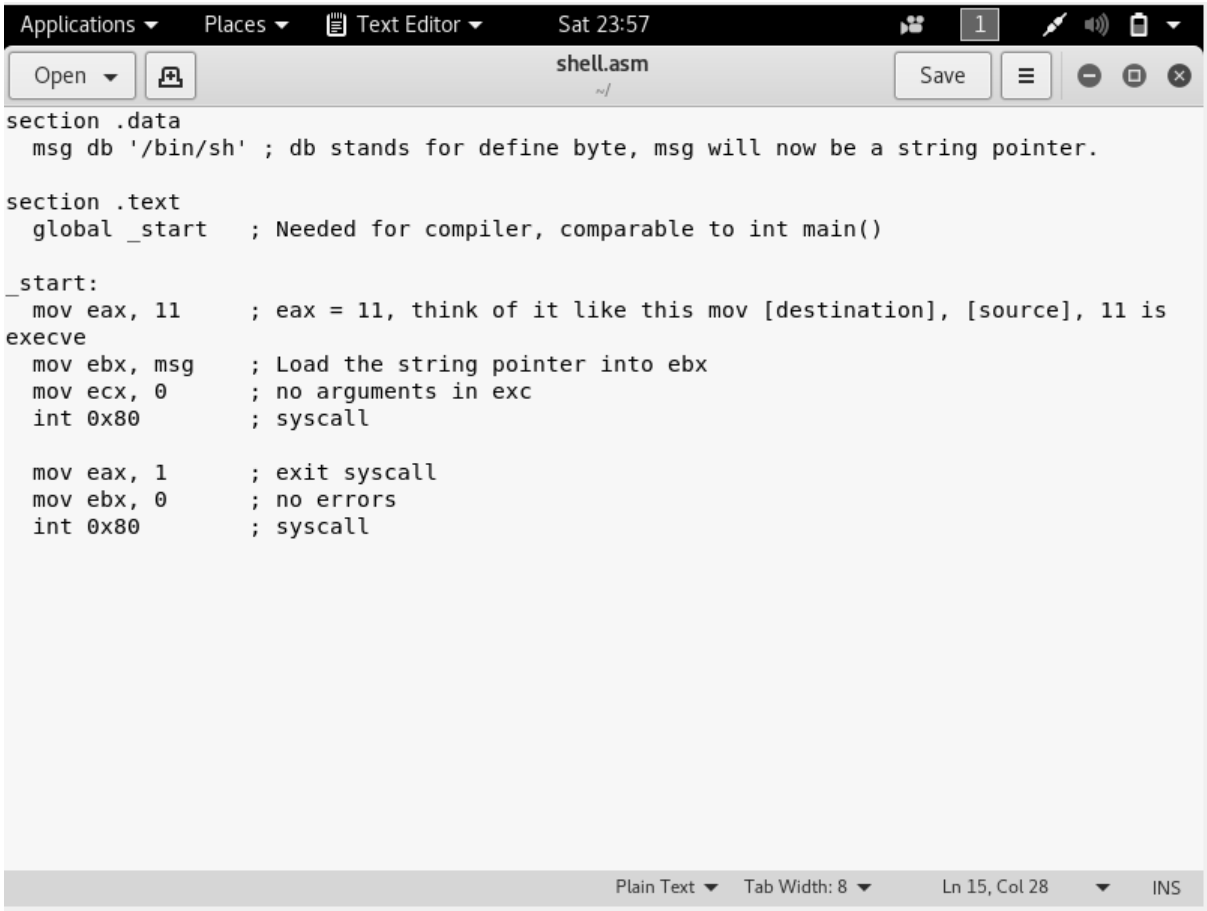


Figure 2: Save it as shell.asm

Next, to compile this, I typed the following commands. But, it showed me error. So, next I went to Stack overflow to correct the error. Then, I typed as shown in the screen shot (Figure 3).

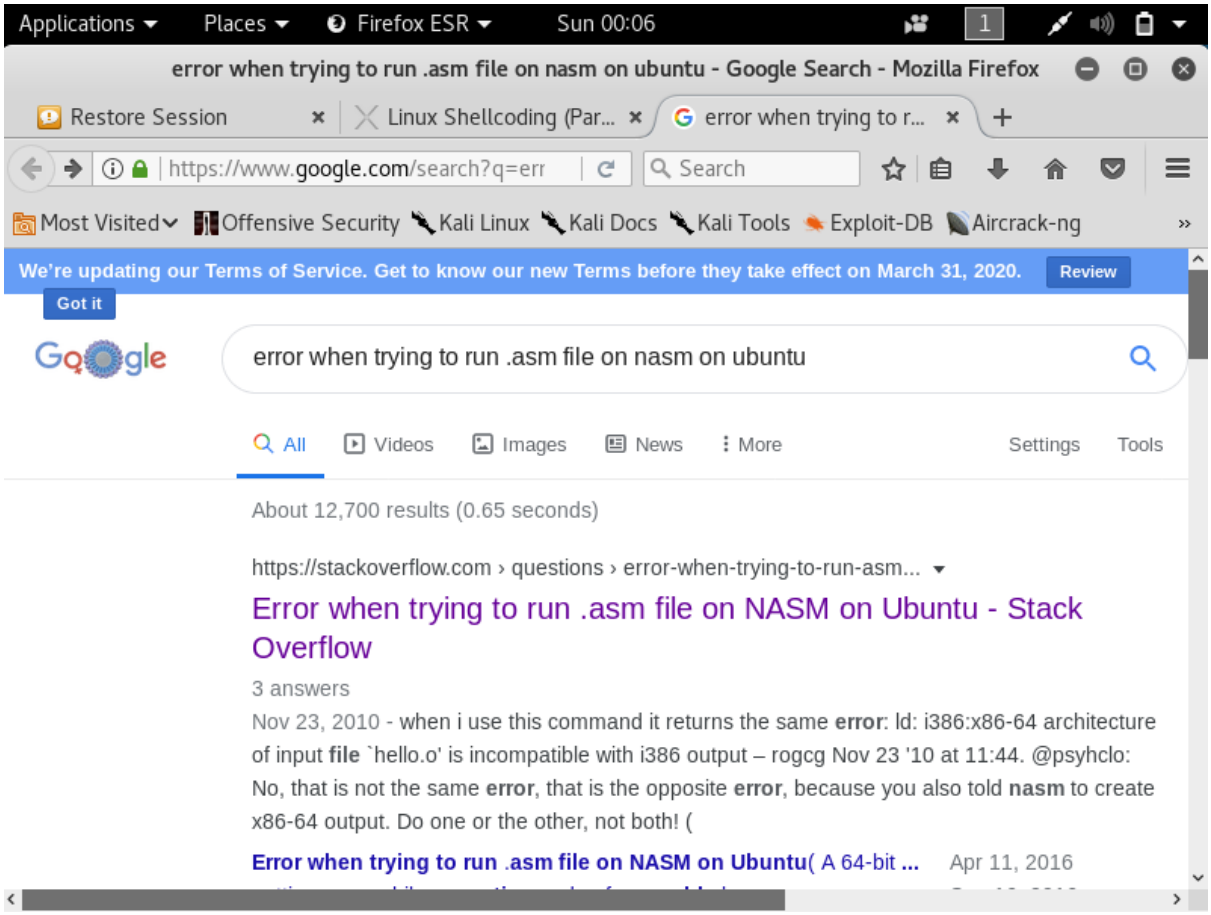


Figure 3: Searching for the error to resolve

Then, as shown in (Figure 4) I typed in the terminal.

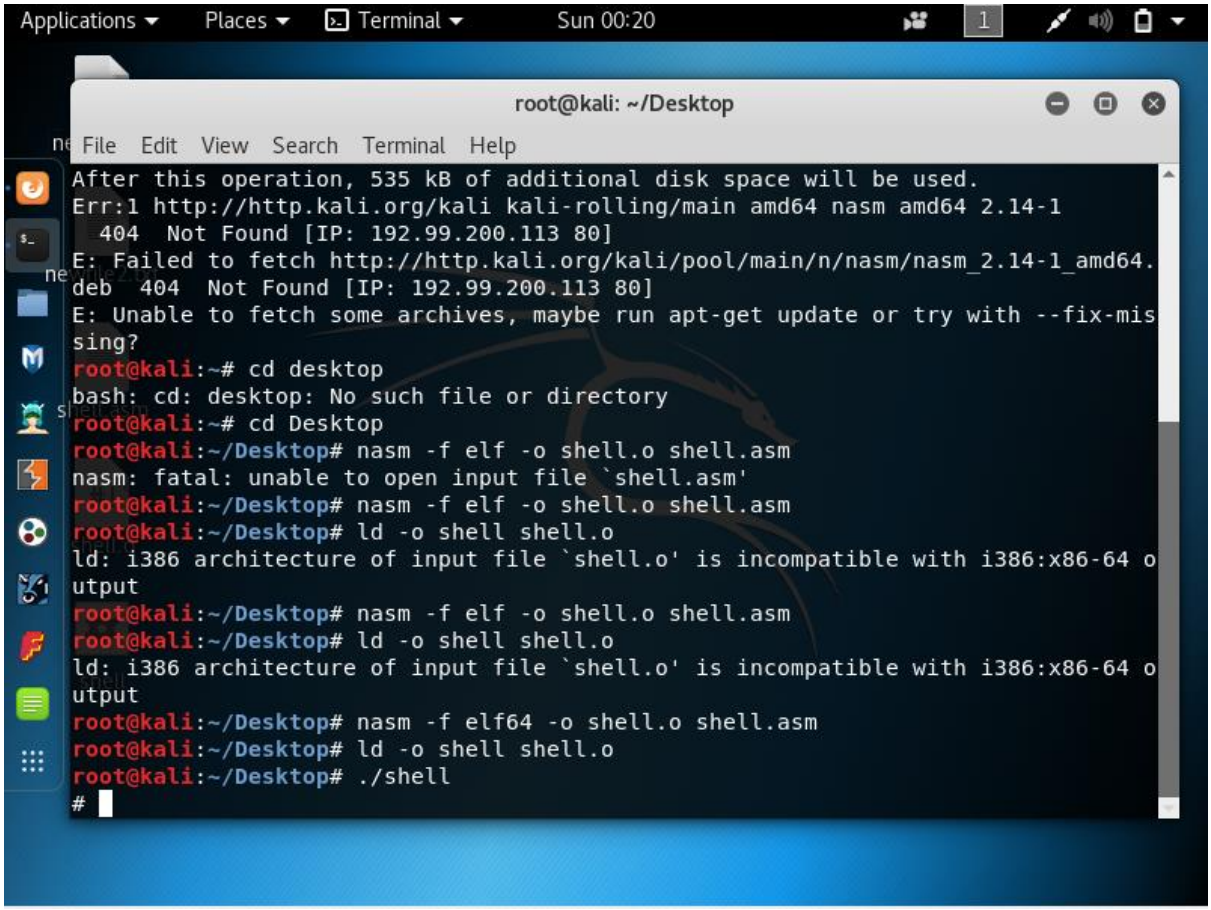
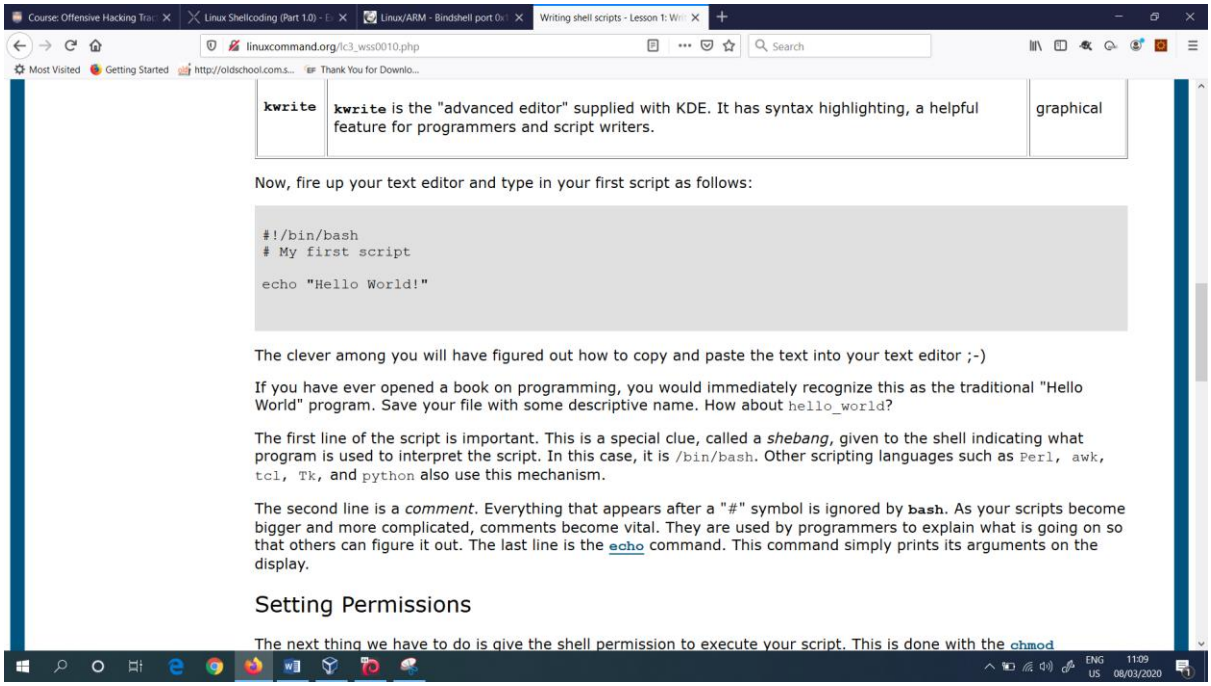


Figure 4: Typed in the terminal

Next, the command which is in the given website https://0x00sec.org/t/linux-shellcoding-part-1-0/289 gave me error, I changed the "Hello World" command as shown below (Figure 6).

Then, I typed as "linuxcommand.org/lc3_wss0010.php" in the google. And get the command in order to write the first script. This website is shown in (Figure 5).



kwrite	kwrite is the "advanced editor" supplied with KDE. It has syntax highlighting, a helpful feature for programmers and script writers.	graphical

Now, fire up your text editor and type in your first script as follows:

```
#!/bin/bash
# My first script

echo "Hello World!"
```

The clever among you will have figured out how to copy and paste the text into your text editor ;-)

If you have ever opened a book on programming, you would immediately recognize this as the traditional "Hello World" program. Save your file with some descriptive name. How about hello_world?

The first line of the script is important. This is a special clue, called a *shebang*, given to the shell indicating what program is used to interpret the script. In this case, it is /bin/bash. Other scripting languages such as Perl, awk, tcl, Tk, and python also use this mechanism.

The second line is a *comment*. Everything that appears after a "#" symbol is ignored by bash. As your scripts become bigger and more complicated, comments become vital. They are used by programmers to explain what is going on so that others can figure it out. The last line is the echo command. This command simply prints its arguments on the display.

Setting Permissions

The next thing we have to do is give the shell permission to execute your script. This is done with the chmod

Figure 5: Get the first script

Next, typed the above shown command in the Kali Linux terminal in order to verify whether the same shell code is derived or not (Figure 6).

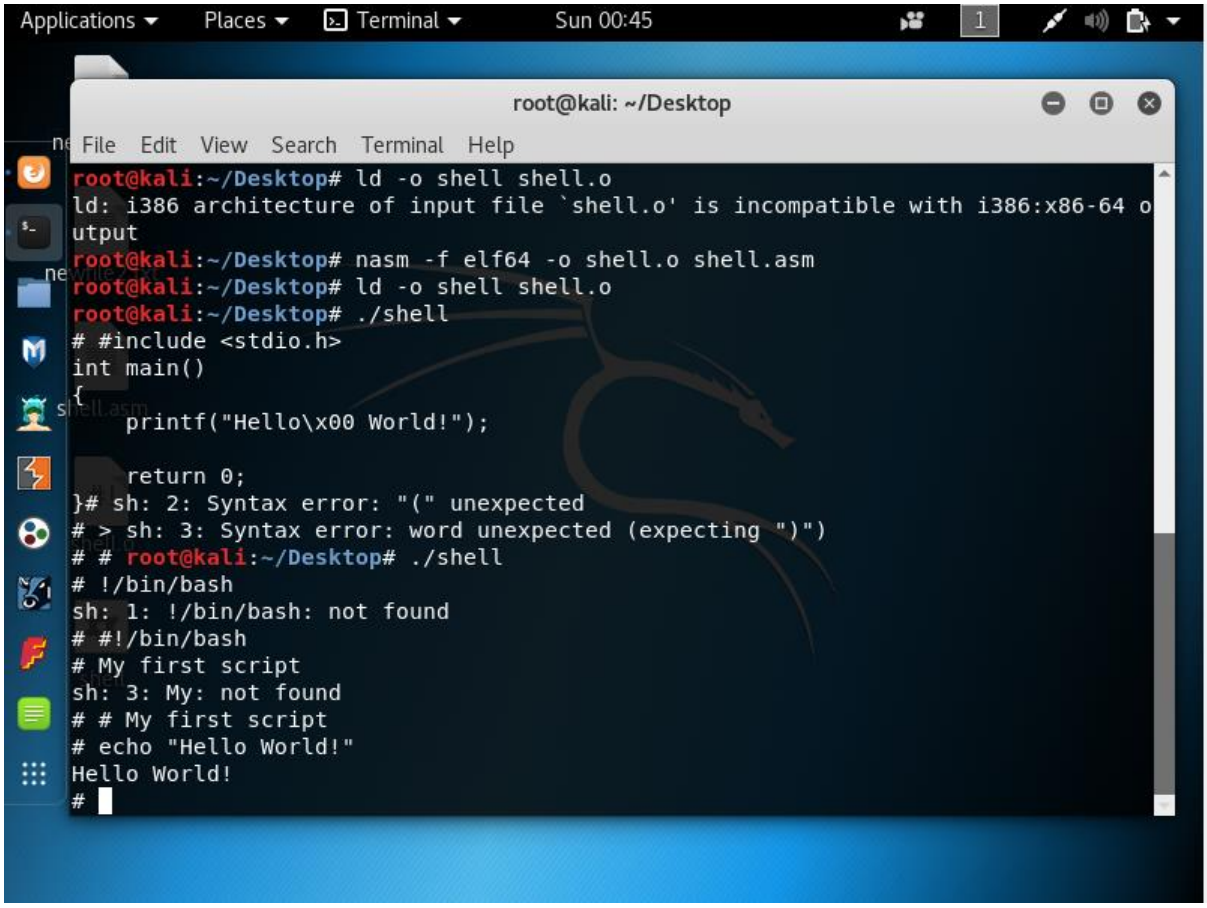


Figure 6: Typed the above command in Kali Linux terminal

Then I typed the command `objdump -M intel -d shell`. Next, it shows me the script code for the file which I saved earlier. The following (Figure 7) shows the following.

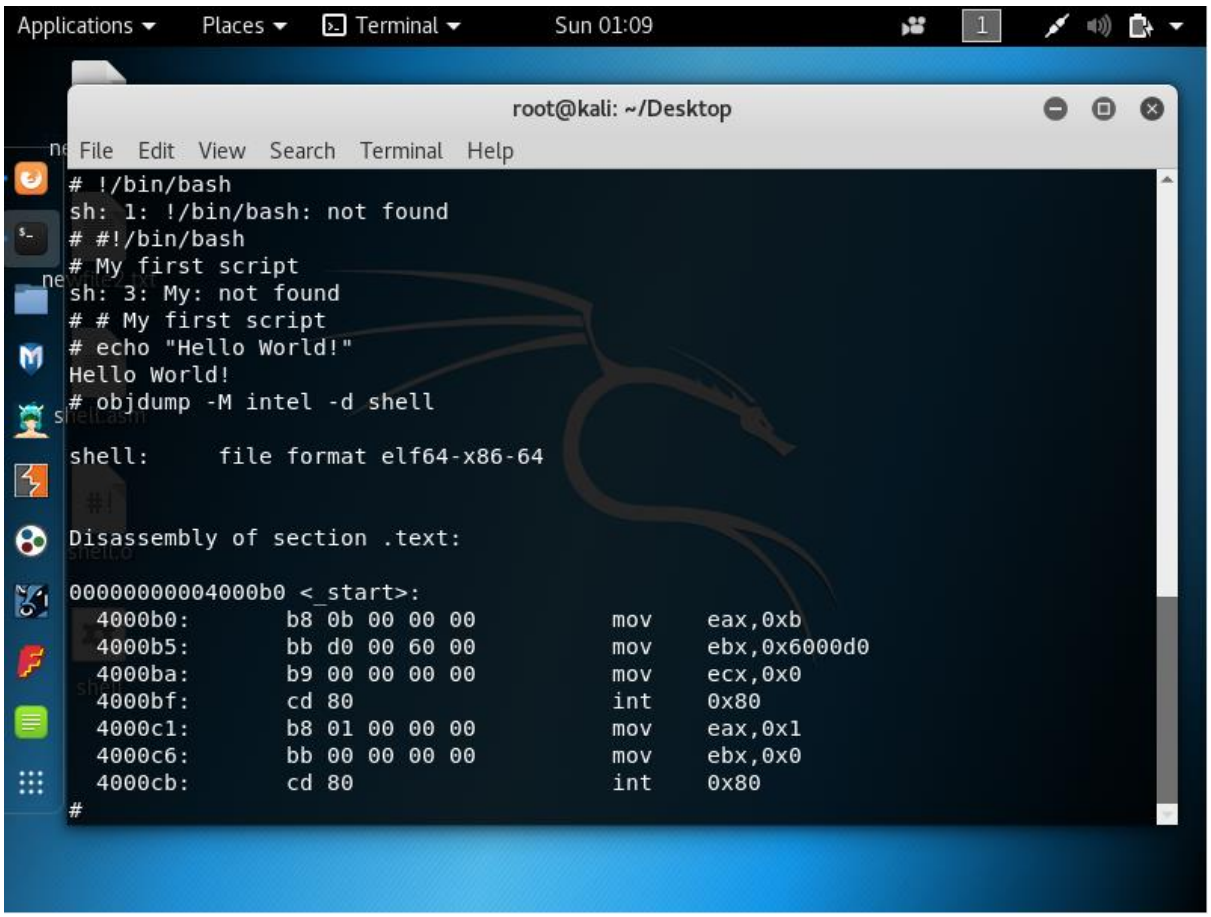


Figure 7: Got the shell code

This shell code which I got was as same which is in the https://0x00sec.org/t/linux-shellcoding-part-1-0/289 website.

So, finally I reverse engineered and I got the same code.