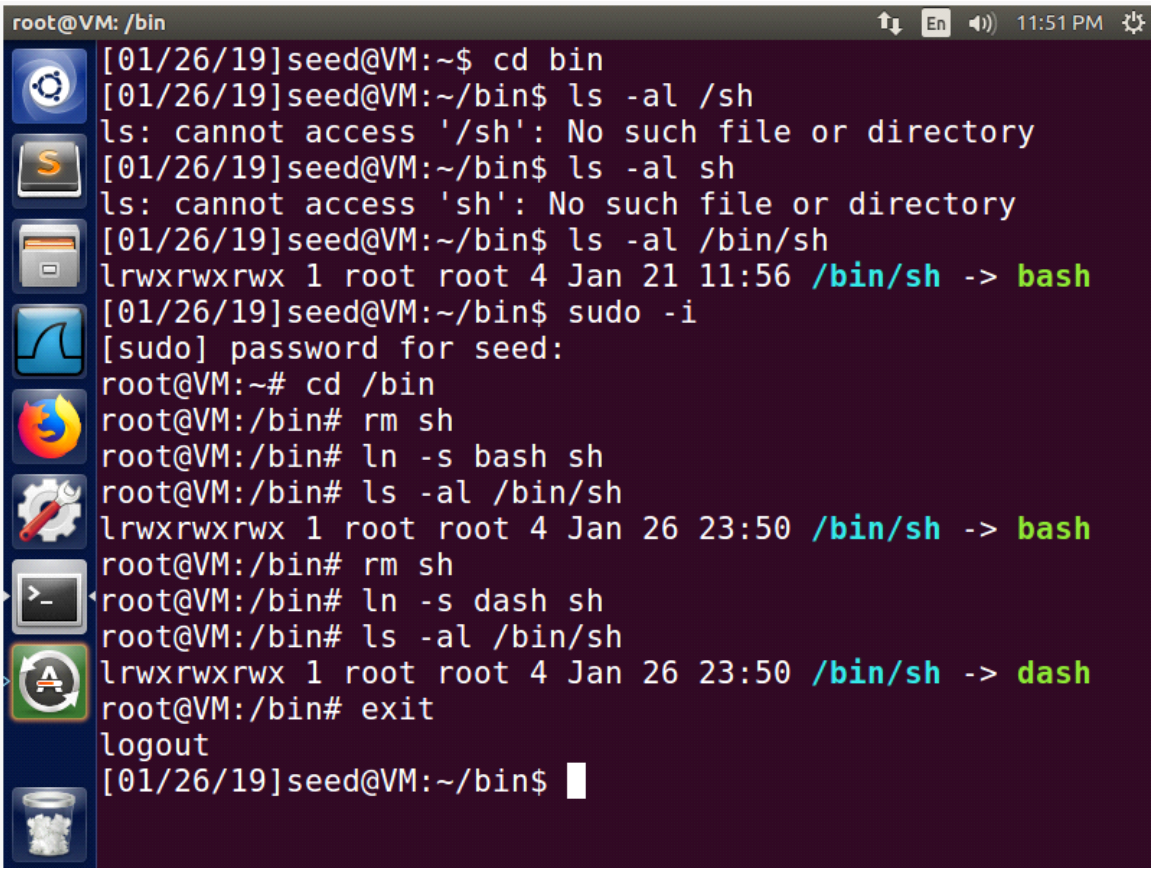
Assignment 2

Name: Arunika yadav

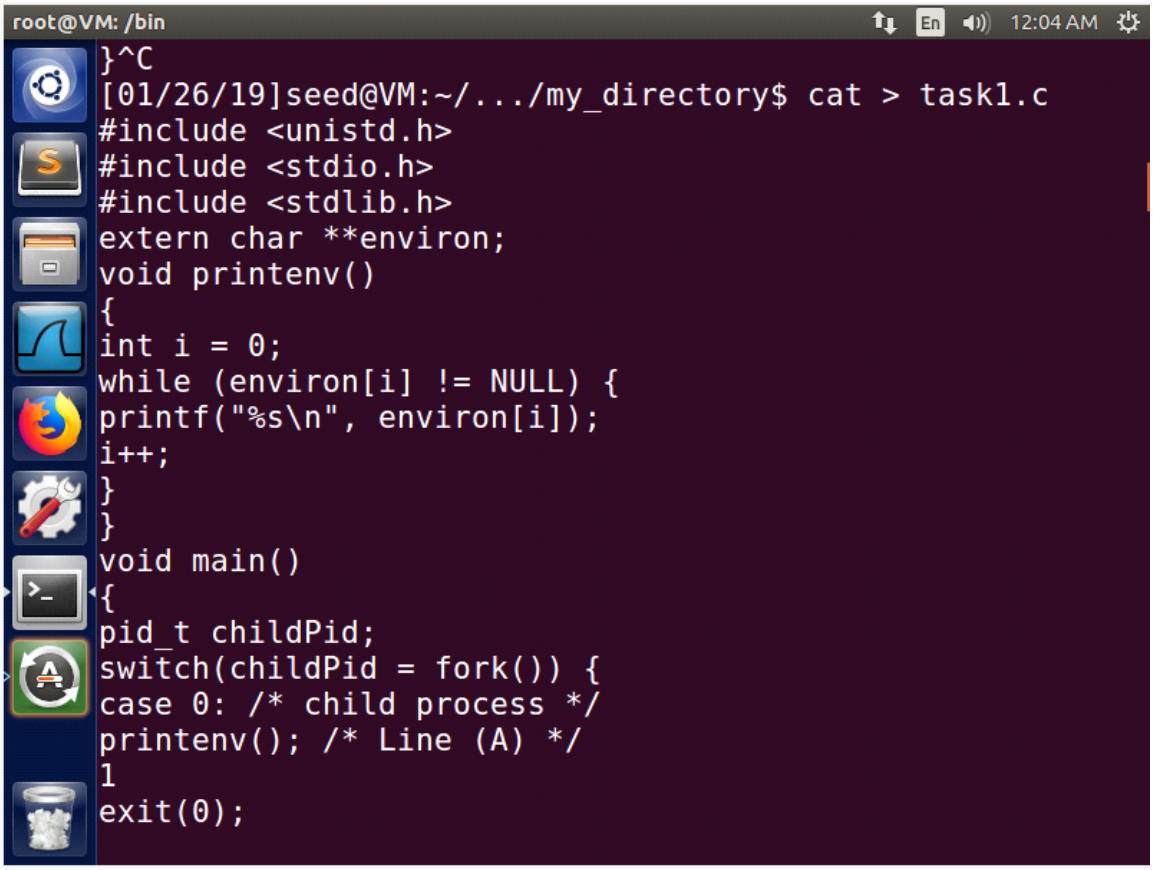
Roll Number:1601CS56

Question 1

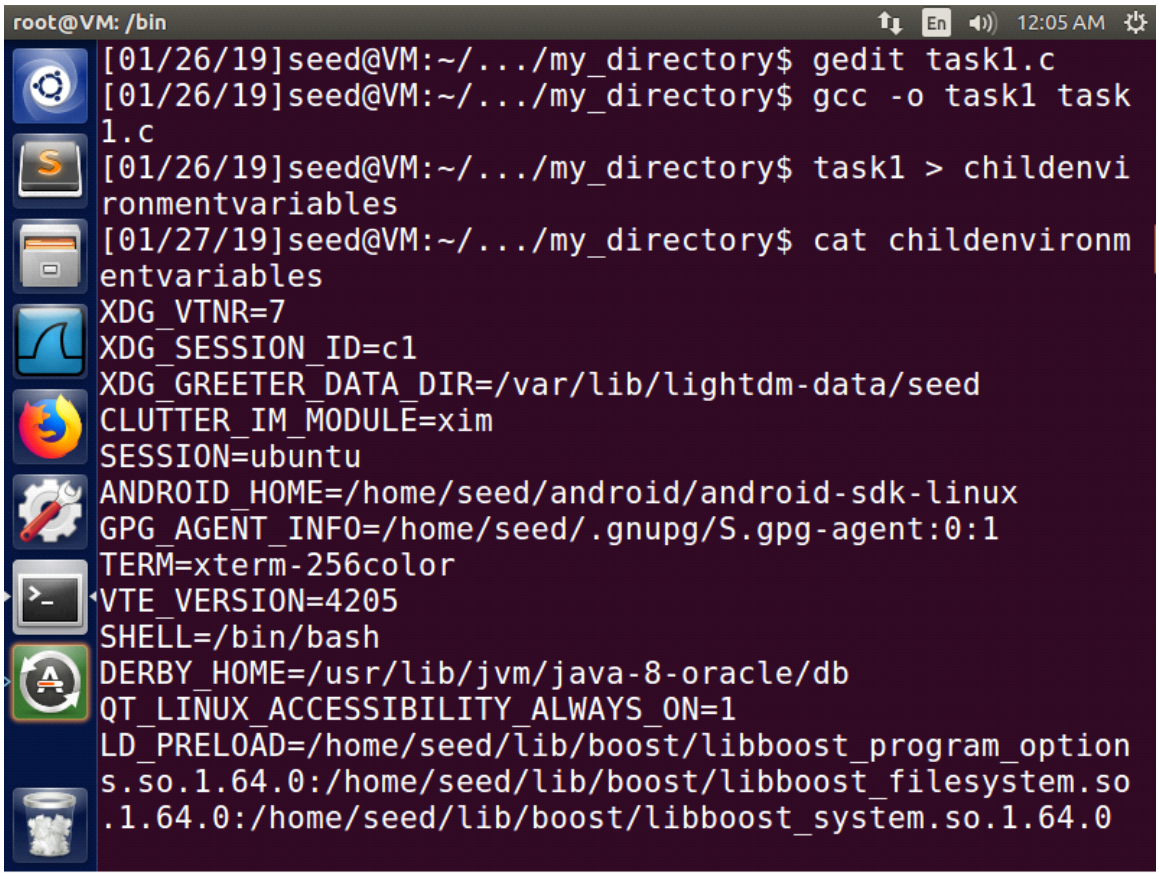
(**Part 1**)The symbolic link is made to point to dash shell program.



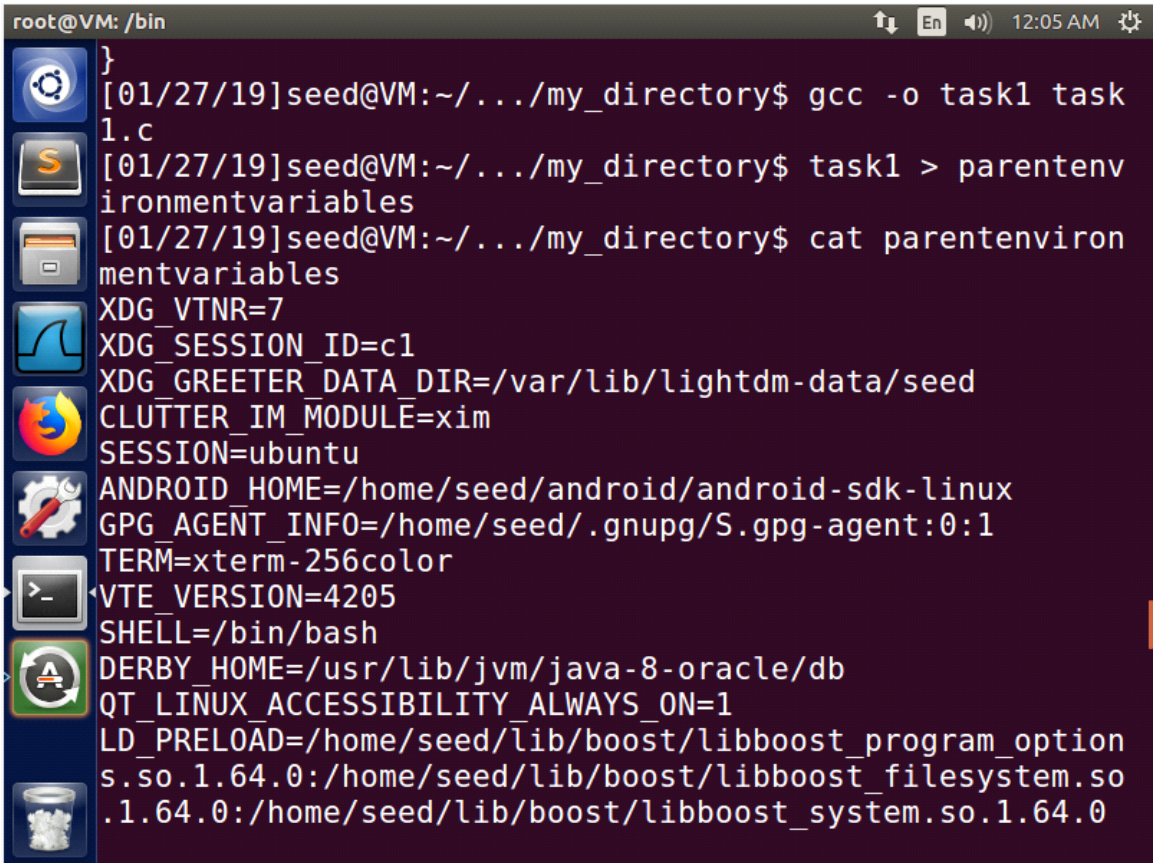
(**Part2**)The program given in the question is copied to a file task1.c



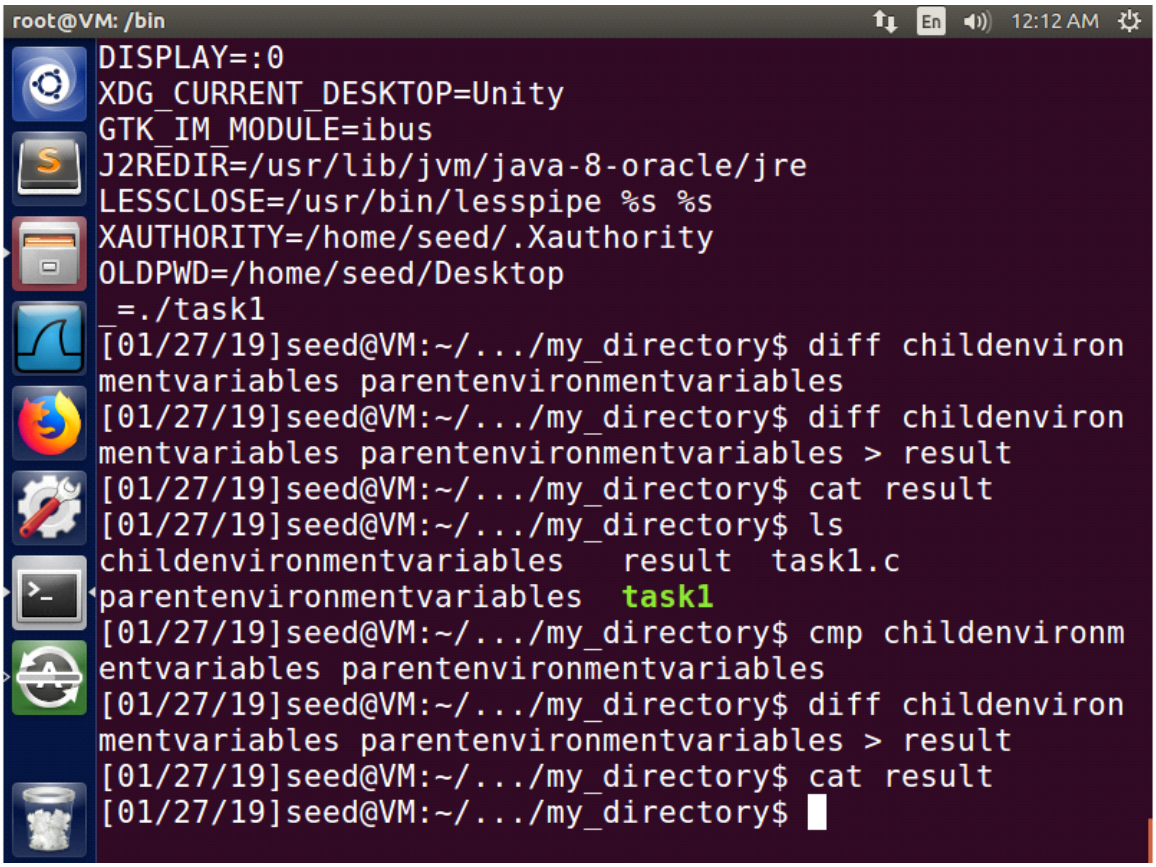
task1.c is executed and an executable task1 is created.task1 on executing runs the program and stores the result in a file called childenvironmentvariables.When we use the cat command the content of the same file is displayed as follows.



(**part3**)The above job is repeated for a file called parentenvironmentvariables after uncommenting the default case body.The child and the parent processes have different pids.

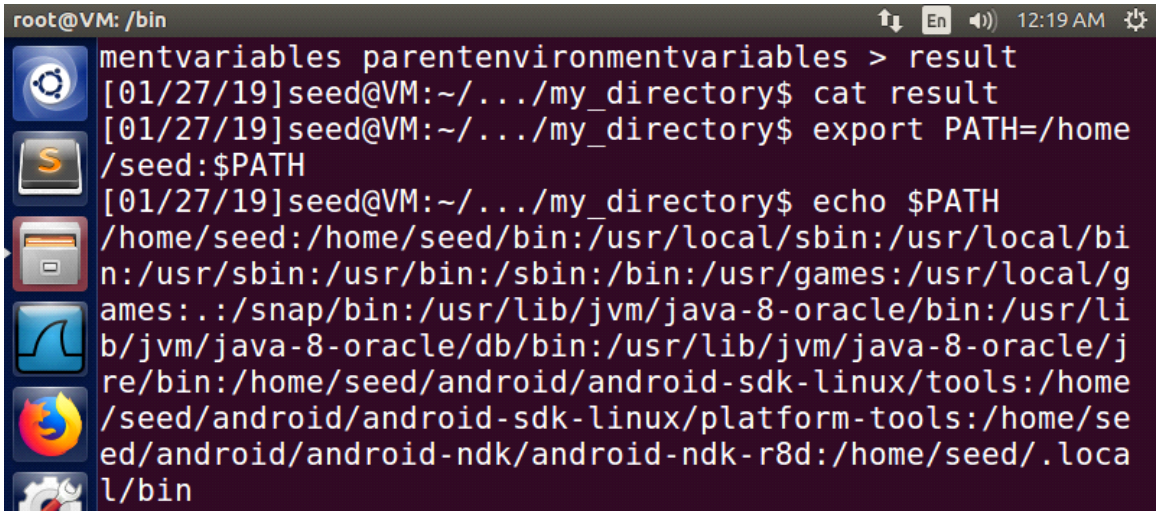


(**Part4**) After using the diff command to compare the two files parentenvironmentvariables and childenvironmentvariables, and storing the result of the same in a file called result, we observe that the result file is empty which indicates that the child process gets all the environment variables of the parent process on being created.thus, the child process has a new PID when it is forked but the same kind of environment as its parent process.



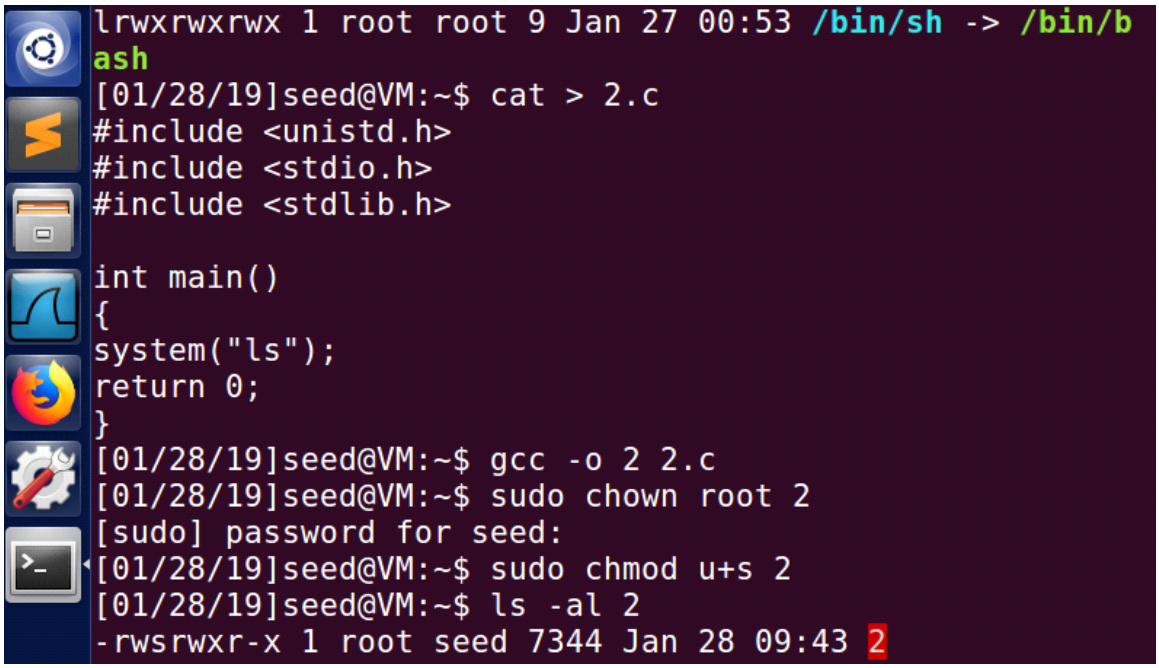
Question 2

The PATH environment variable is modified as shown below:



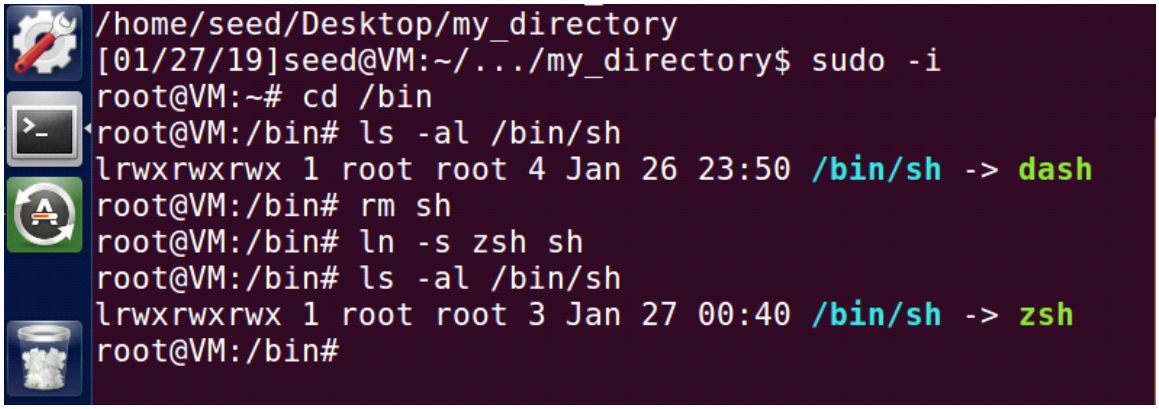
A file 2.c is created with the content specified by the question.It is compiled and the executable 2 is

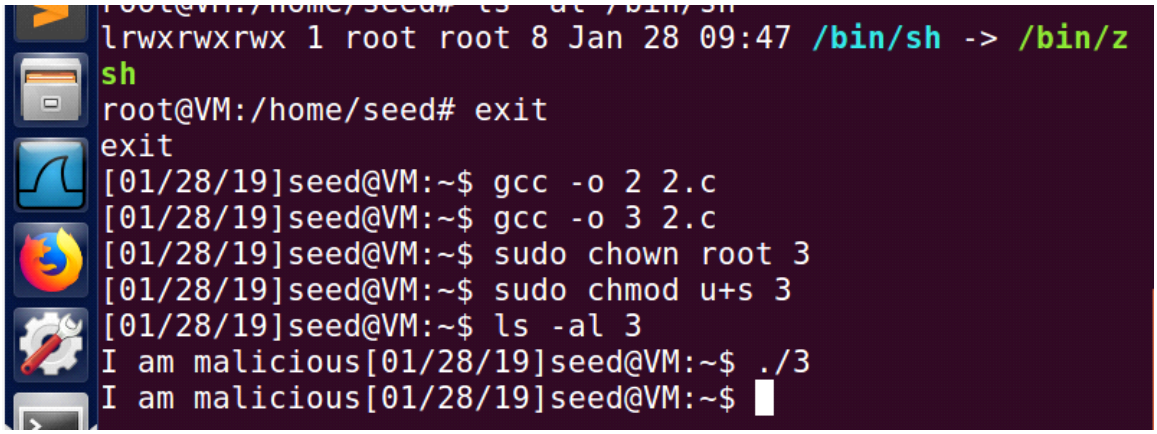
made.The owner of the 2 file is changed to root usign chown and the program is made setuid using the chmod command.Also a program ls.c is written which contains the malicious content and we have modified the PATH variable to include the path of the ls executable of the ls.c file in the variable.We also use the eport keyword which passes on the environment variable to any child processes of the executable 2.



When the executable 2 is executed then the Path variable first looks in the current directory for the ls program since it is specified first in the environment variable and when it finds that the file exists in the home directory, it runs this ls instead of the shell program's ls in the /bin folder.The ls file thus replaces the functionality of the default ls command and this can be used as a malicious file.**Hence, we can use setuid programs to run mailicious files with root privileges if the Path variable is altered**.

Question 3

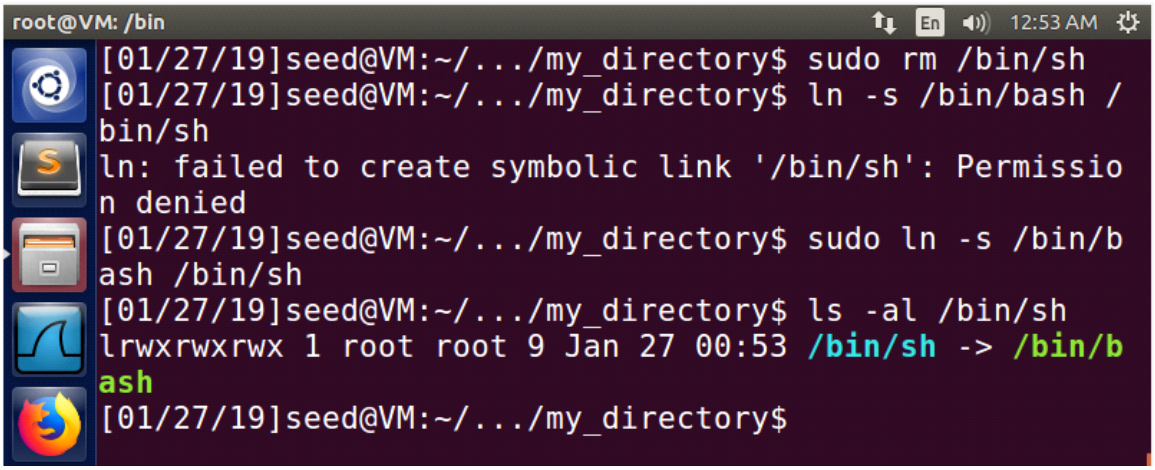




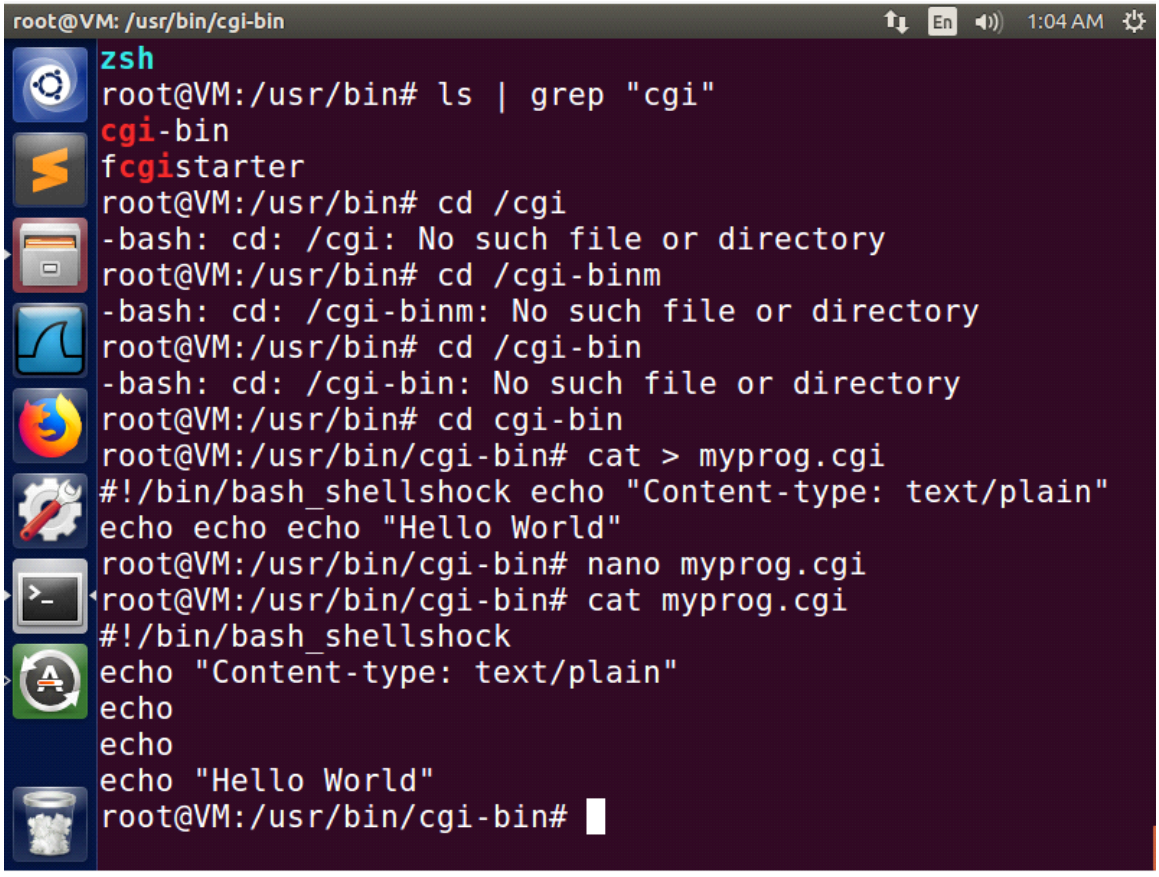
The "2" executable in the previous question is replaced by "3", which is changed to a root owned setuid program as shown above.We had seen in assignment 1 that the zsh shell is more vulnerable to attacks by a leaky setuid program as compared to a bash shell, hence here also when we execute the 3 executable then because of the change in the PATH environment variable , it performs the same as in the previous question and the attack succeds.

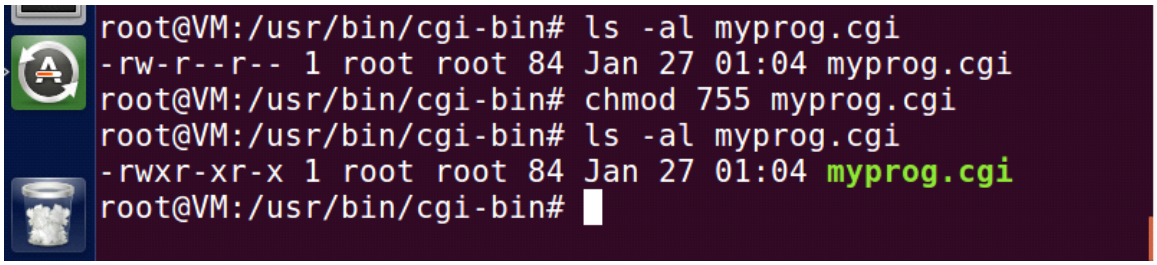
Question 4

The default symbolic changed to bash as shown below:

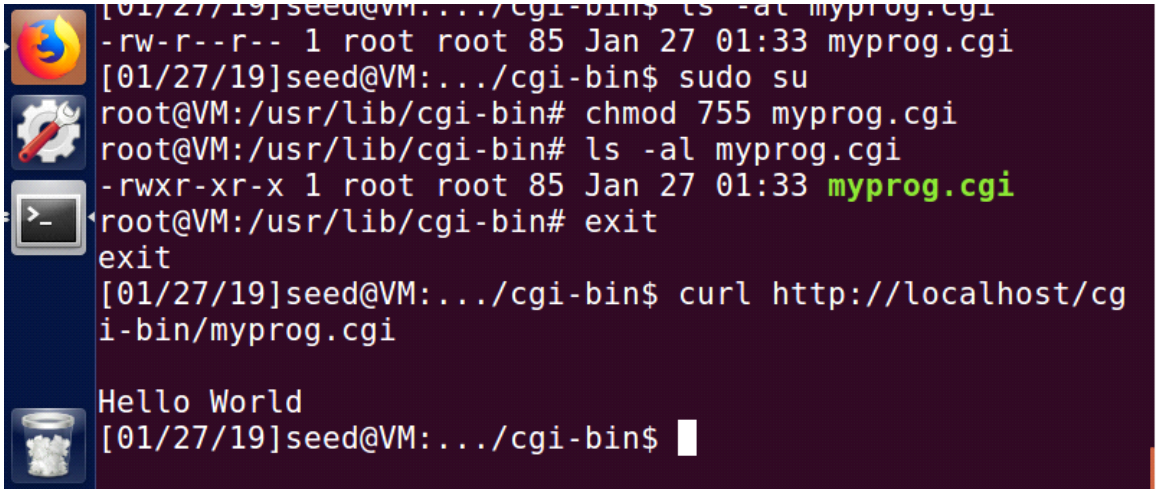


The cgi-bin is made and the myprog.cgi program is loaded with the content required to make a cgi program.The root owner does this because the /usr/bin folder is editable only by the root.





When we use the curl command to send a request to the localhost server , it runs the myprog.cgi program and gives the output through the echo command.



After changing the contents of myprog.cgi to :

#!/bin/bash_shellshock

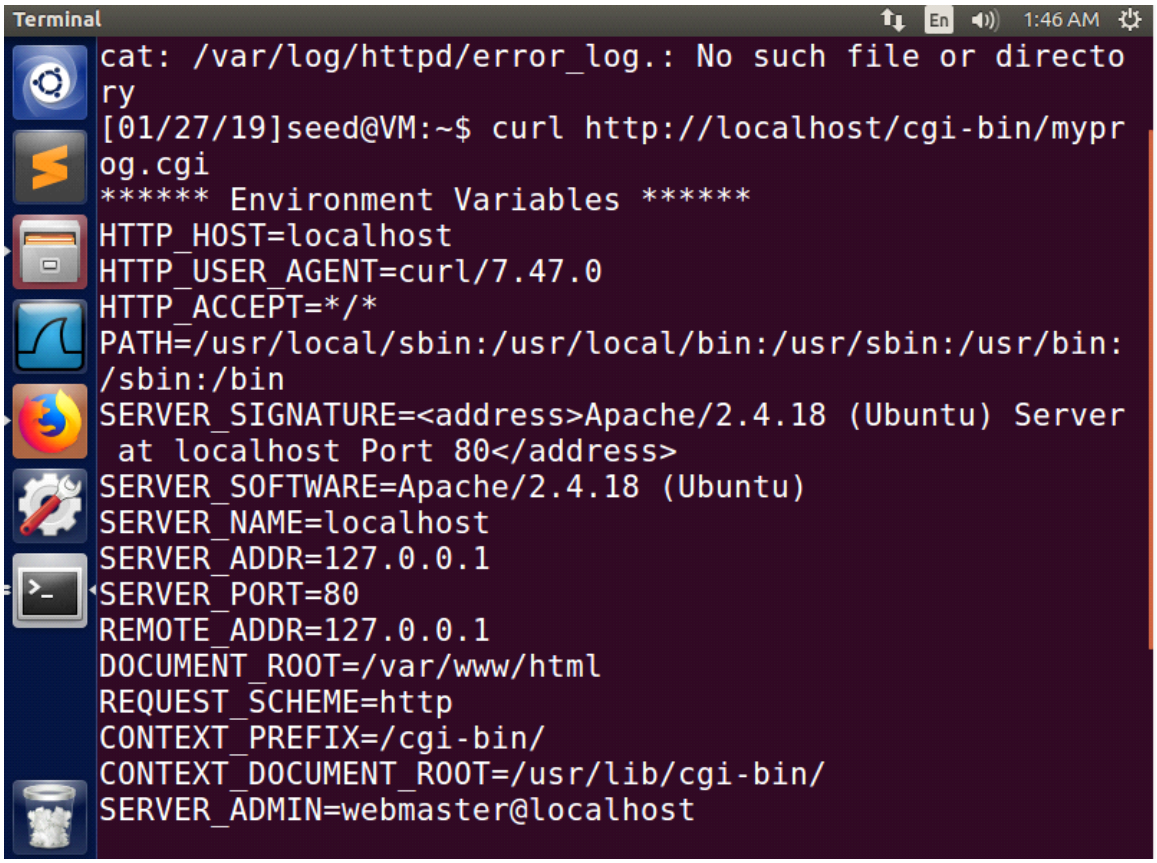
echo "Content-type: text/plain"

echo

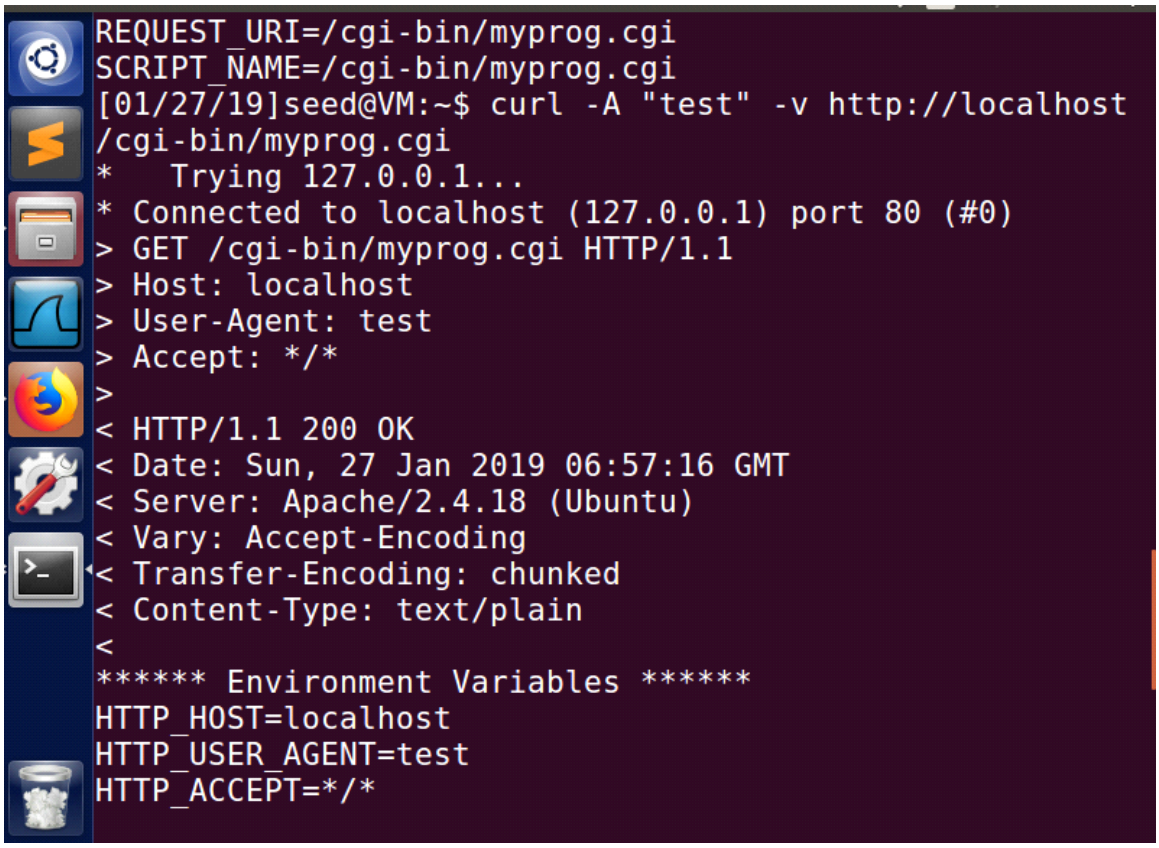
echo "**** Environment Variables ******"**

strings /proc/$$/environ

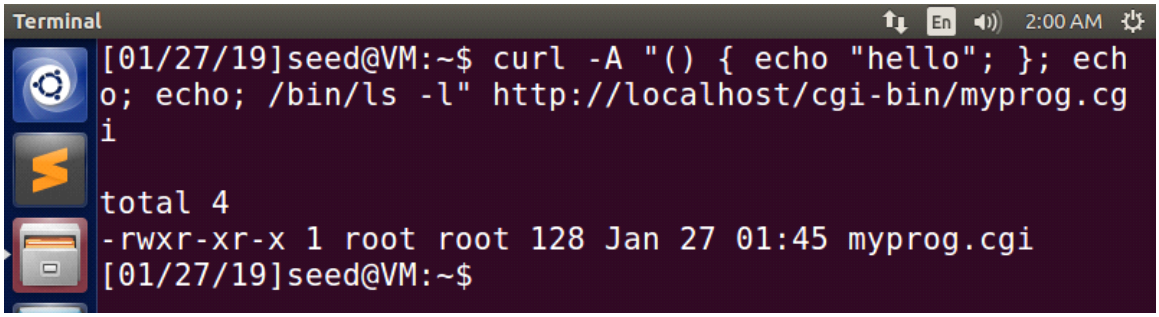
and raising a http request via the curl command, all of the environment variables of the current process get listed as a response from the server.We can use the -A attribute of the curl command to pass malicious content to the http request header and this content will influence the environment variables.



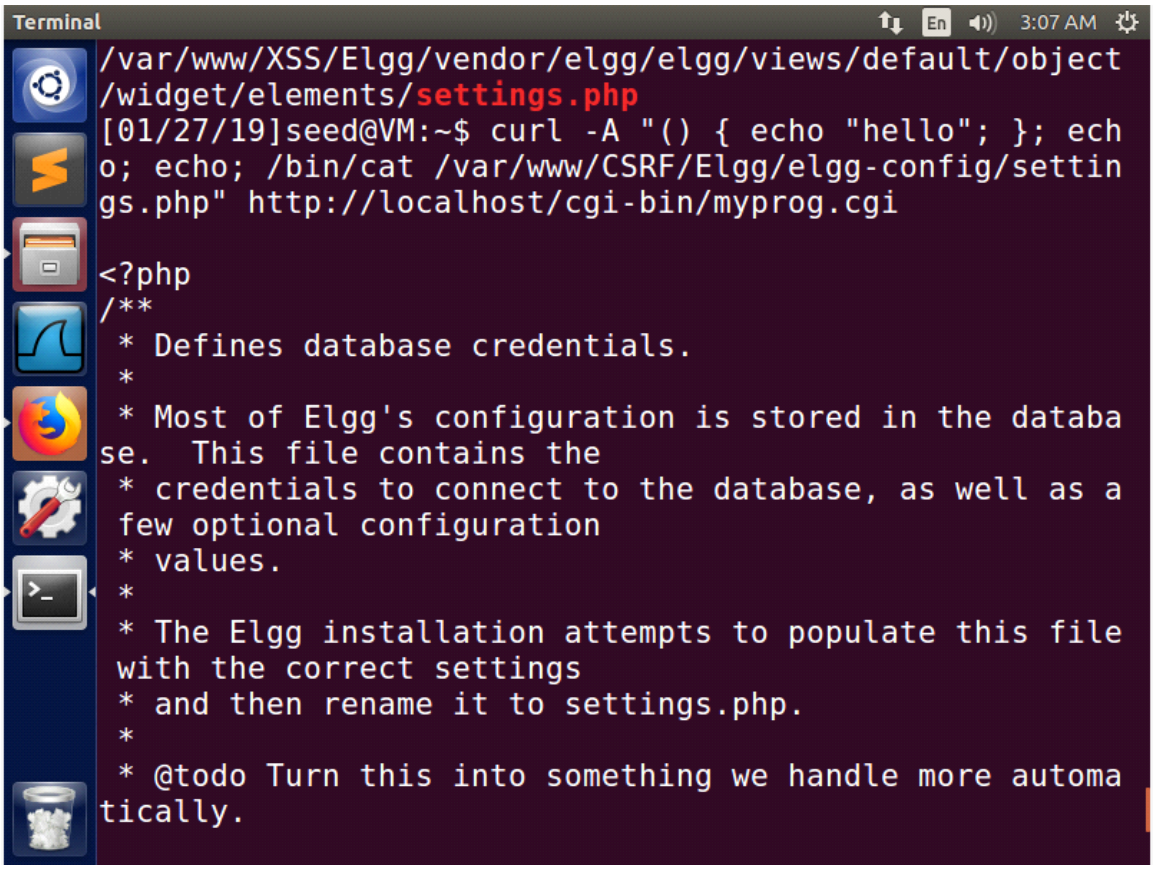
As mentioned above when we used the -A attribute(the user agent argument) and passed a string "test" to the http header, the HTTP_USER_AGENT environment variable was influenced.



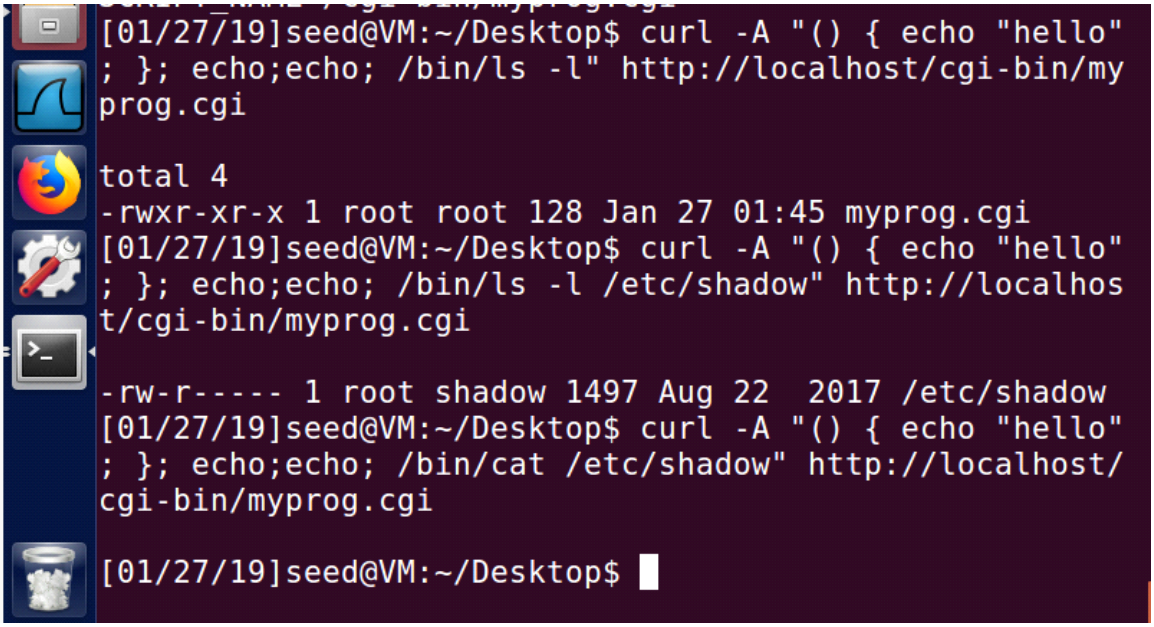
The remote user can now pass whatever command he wants to via this method and get confidential information from the server.In the screenshot below a number of shell commands separated by semi-colons were passed and each of them gets executed.We use the ls -l command to get the list of the contents of the present working directory.



In the http header below we used the cat command to get the contents of a configuration file of the server.The server receives the environment variables and creates a child process and all the malicious data from the http header gets passed on to the child process via the environment variables.the child process ultimately executes this data as shell commands and exposes the otherwise confidential data from the server side to the remote user.



If we know the path to world readable file on server side then we can steal the content of that file by using the user agent argument in the curl command and passing the cat command to the http header.



The /etc/shadow file is not world readable, hence when we pass the file with as an argument to the cat command via the curl user argument variable , still it does not show the contents of the /etc/shadow

file.hence we cannot access the contents of the /etc/shadow file using cgi.