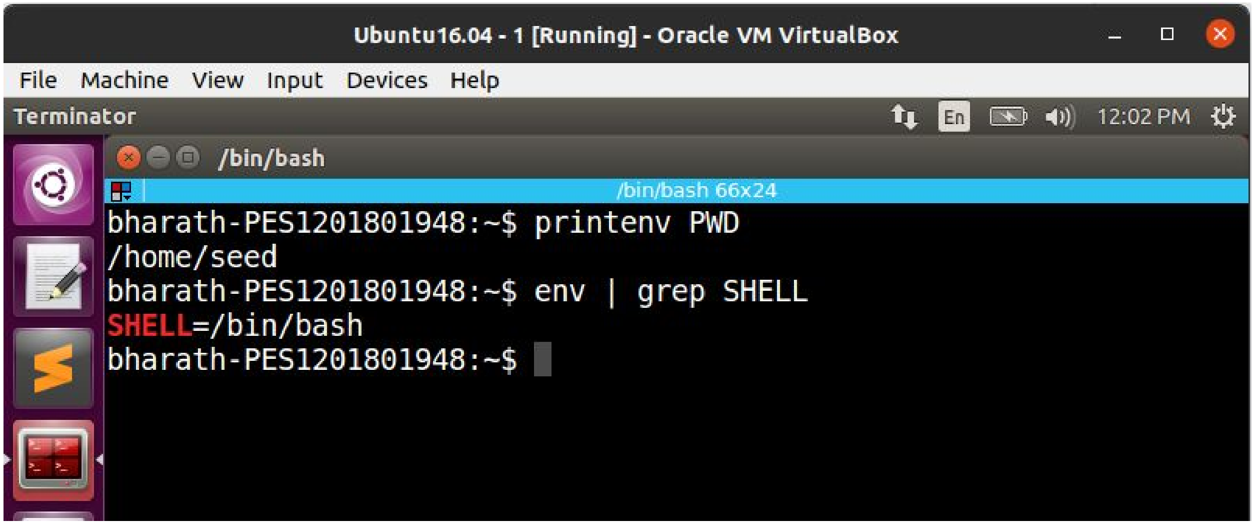
Information Security
Lab 1
Set-UID and Environment Variables

PES1201801948
Bharath S Bhambore
Section H - 6th sem

Lab Setup :

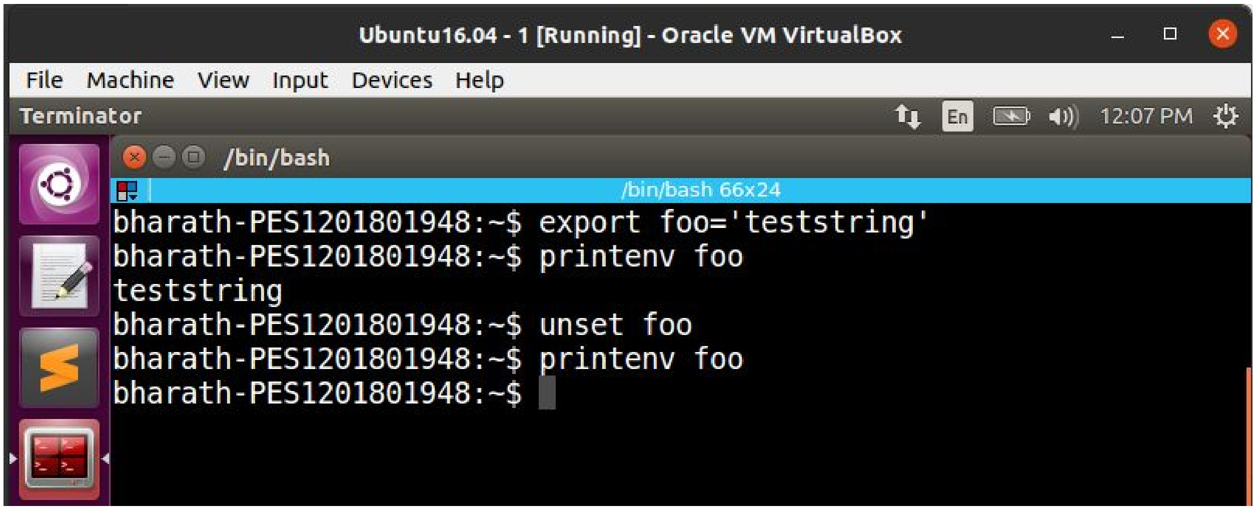
Machine : Seed Ubuntu 16.04
IP address : 10.0.2.9

Task 1 : Manipulating Variables


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
En
12:02 PM
/bin/bash
/bin/bash 66x24

```
bharath-PES1201801948:~$ printenv PWD
/home/seed
bharath-PES1201801948:~$ env | grep SHELL
SHELL=/bin/bash
bharath-PES1201801948:~$
```

“env” command lists out the environmental variables.
“printenv” prints the required environmental variable.


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
En
12:07 PM
/bin/bash
/bin/bash 66x24

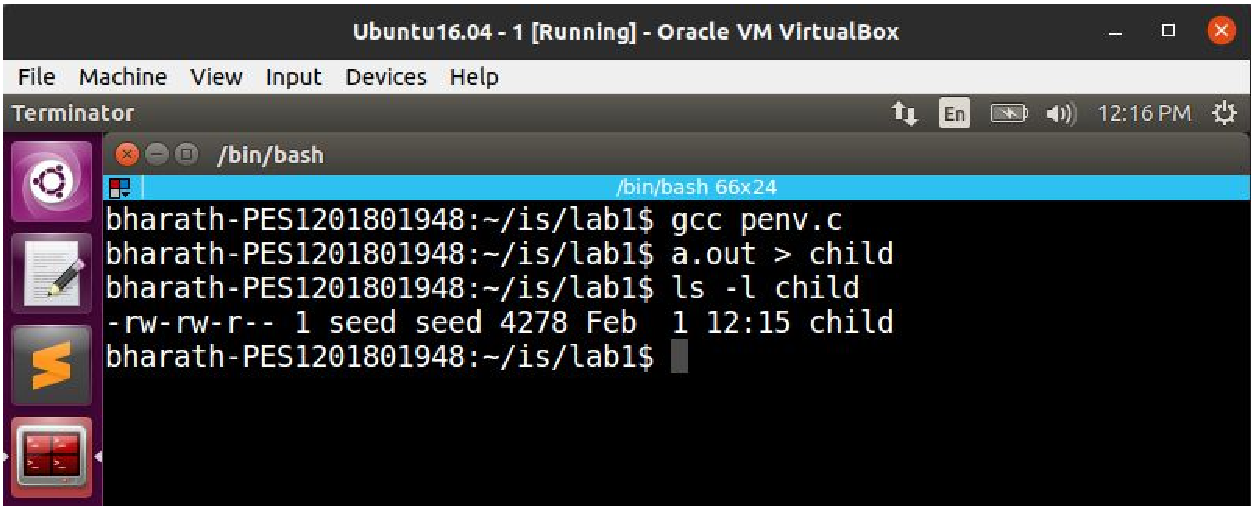
```
bharath-PES1201801948:~$ export foo='teststring'
bharath-PES1201801948:~$ printenv foo
teststring
bharath-PES1201801948:~$ unset foo
bharath-PES1201801948:~$ printenv foo
bharath-PES1201801948:~$
```

export command is a SHELL BUILTIN command, it is used to set an environmental variable. It marks an environmental variable to be exported to any newly forked child processes, implying the child process is allowed to inherit the marked variables.

unset is a command that is used to remove a variable from the list of variables that are being tracked.
“unsetting” a variable means you can not access the value stored in that variable.

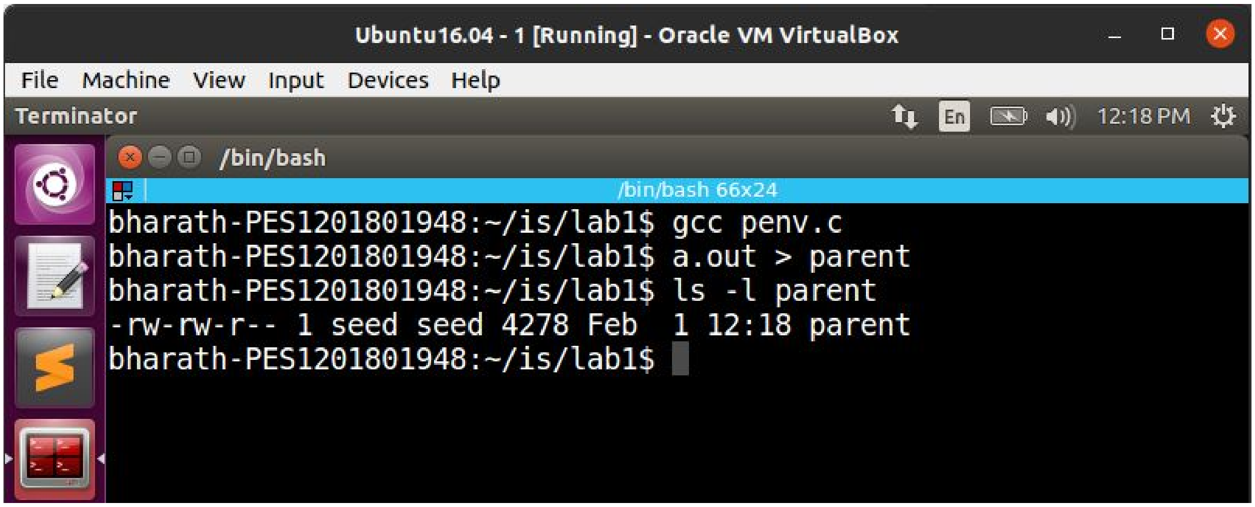
Task 2 : Inheriting environment variables from Parents

Printenv in child case :


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
En
12:16 PM
/bin/bash
/bin/bash 66x24

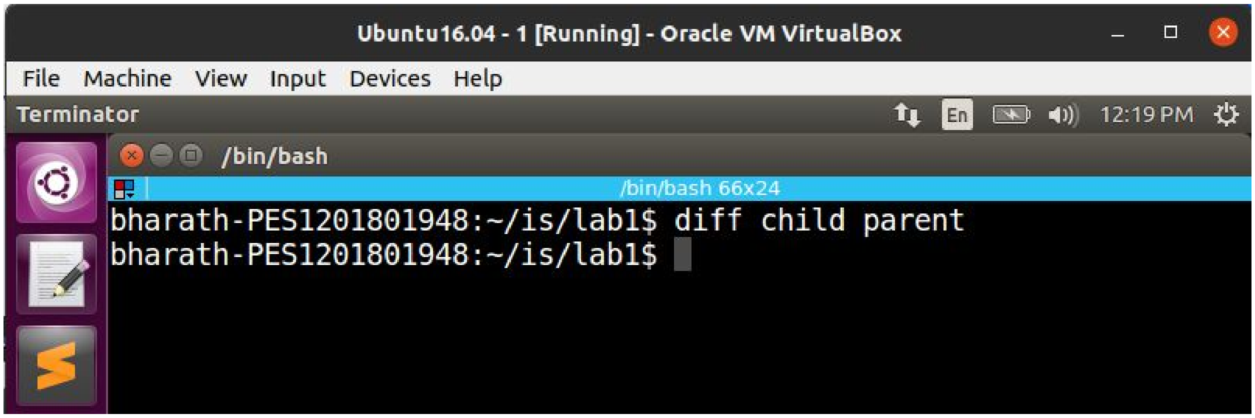
```
bharath-PES1201801948:~/is/lab1$ gcc penv.c
bharath-PES1201801948:~/is/lab1$ a.out > child
bharath-PES1201801948:~/is/lab1$ ls -l child
-rw-rw-r-- 1 seed seed 4278 Feb  1 12:15 child
bharath-PES1201801948:~/is/lab1$
```

Printenv in parent case :


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
12:18 PM
/bin/bash
/bin/bash 66x24

```
bharath-PES1201801948:~/is/lab1$ gcc penv.c
bharath-PES1201801948:~/is/lab1$ a.out > parent
bharath-PES1201801948:~/is/lab1$ ls -l parent
-rw-rw-r-- 1 seed seed 4278 Feb  1 12:18 parent
bharath-PES1201801948:~/is/lab1$ 
```

Running diff, to compare the files :

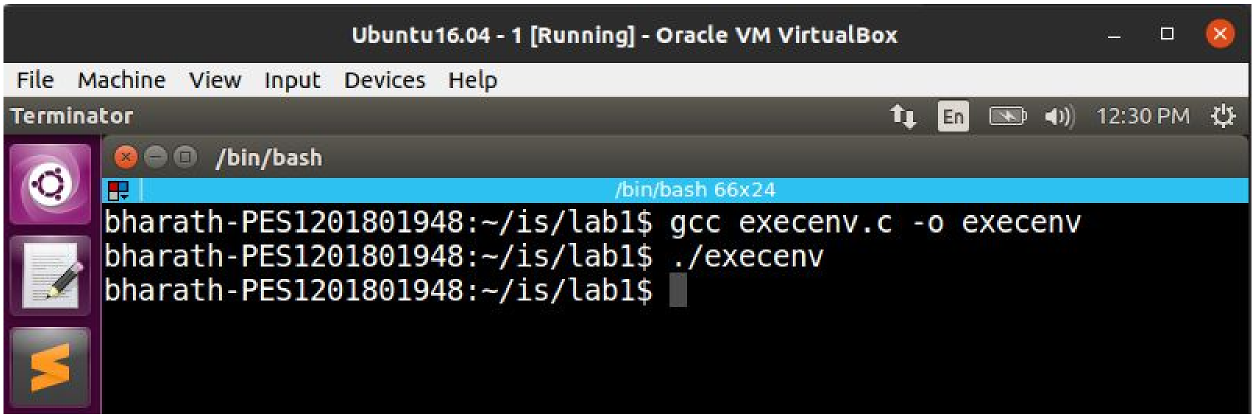

Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
12:19 PM
/bin/bash
/bin/bash 66x24

```
bharath-PES1201801948:~/is/lab1$ diff child parent
bharath-PES1201801948:~/is/lab1$ 
```

There is no difference between the 2 files.
The variable “environ” points to an array of pointers to strings called the environment.
When a child process is created using fork(), it inherits a copy of the parent’s environment. Thus there is no difference in the child and parent files in the above program.

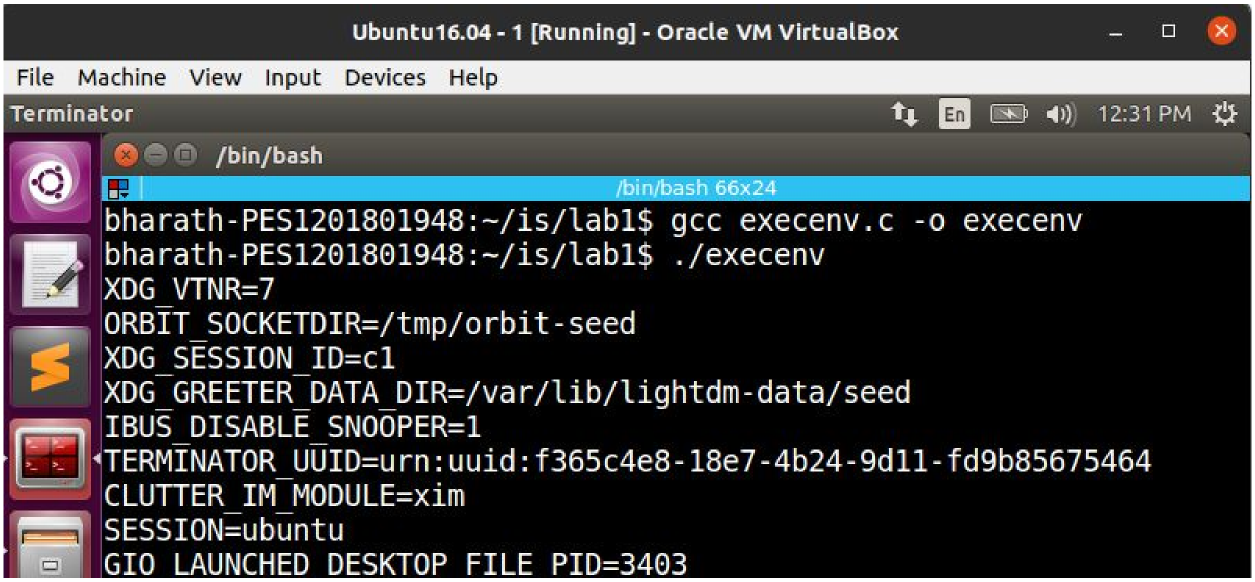
Task 3 : Environment variables and execve()

envp = NULL


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
12:30 PM
/bin/bash
/bin/bash 66x24

```
bharath-PES1201801948:~/is/lab1$ gcc execenv.c -o execenv
bharath-PES1201801948:~/is/lab1$ ./execenv
bharath-PES1201801948:~/is/lab1$
```

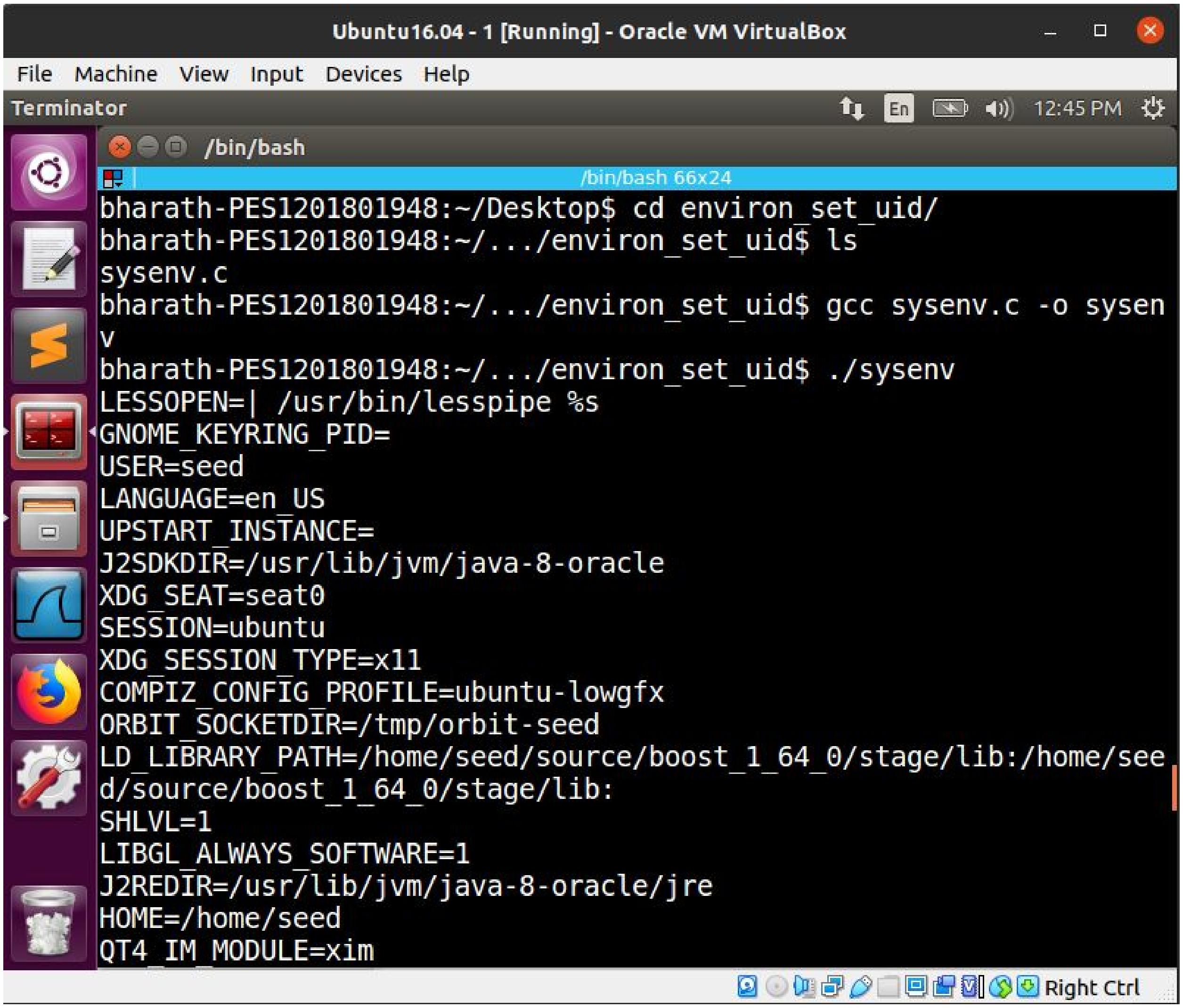
envp = environ


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
12:31 PM
/bin/bash
/bin/bash 66x24

```
bharath-PES1201801948:~/is/lab1$ gcc execenv.c -o execenv
bharath-PES1201801948:~/is/lab1$ ./execenv
XDG_VTNR=7
ORBIT_SOCKETDIR=/tmp/orbit-seed
XDG_SESSION_ID=c1
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed
IBUS_DISABLE_SNOOPER=1
TERMINATOR_UUID=urn:uuid:f365c4e8-18e7-4b24-9d11-fd9b85675464
CLUTTER_IM_MODULE=xim
SESSION=ubuntu
GIO_LAUNCHED_DESKTOP_FILE_PID=3403
```

envp is an array of strings, generally in the form of key=value.
In the first screenshot, envp was NULL, hence why the execution produced no output.
While in the second screenshot, envp was given the environ variable which points to an array of pointers to strings called the environment.
Hence why, the output of this is the list of environmental variables.

Task 4 : Environment Variables and system()



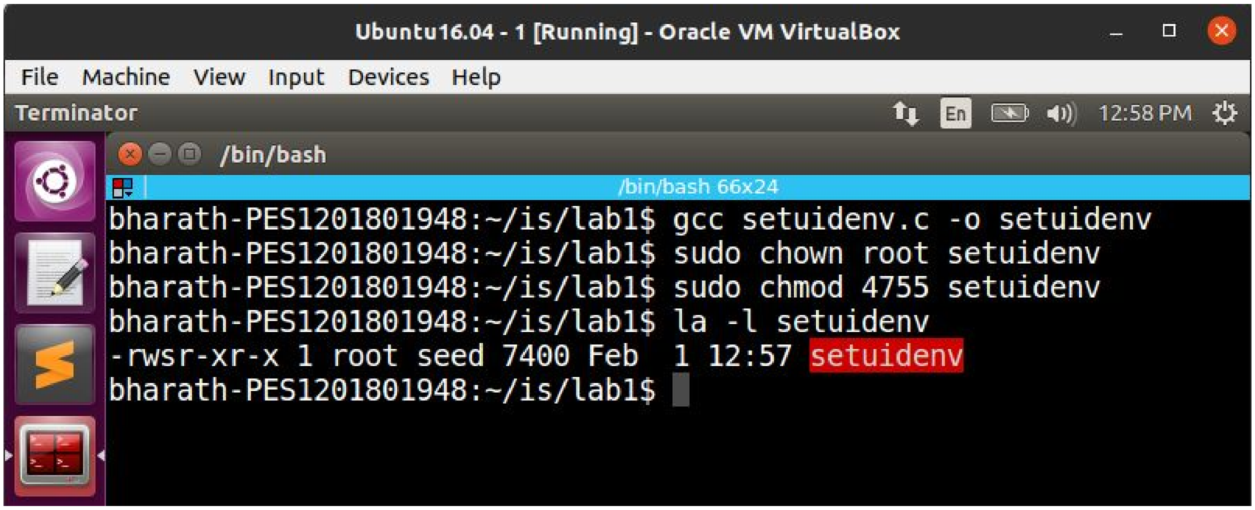
Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
En
12:45 PM
/bin/bash
/bin/bash 66x24
Right Ctrl

```
bharath-PES1201801948:~/Desktop$ cd environ_set_uid/
bharath-PES1201801948:~/.../environ_set_uid$ ls
sysenv.c
bharath-PES1201801948:~/.../environ_set_uid$ gcc sysenv.c -o sysen
v
bharath-PES1201801948:~/.../environ_set_uid$ ./sysenv
LESSOPEN=| /usr/bin/lesspipe %s
GNOME_KEYRING_PID=
USER=seed
LANGUAGE=en_US
UPSTART_INSTANCE=
J2SDKDIR=/usr/lib/jvm/java-8-oracle
XDG_SEAT=seat0
SESSION=ubuntu
XDG_SESSION_TYPE=x11
COMPIZ_CONFIG_PROFILE=ubuntu-lowgfx
ORBIT_SOCKETDIR=/tmp/orbit-seed
LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/see
d/source/boost_1_64_0/stage/lib:
SHLVL=1
LIBGL_ALWAYS_SOFTWARE=1
J2REDIR=/usr/lib/jvm/java-8-oracle/jre
HOME=/home/seed
QT4_IM_MODULE=xim
```

system() uses fork() to create a child process to execute the command specified using execl()

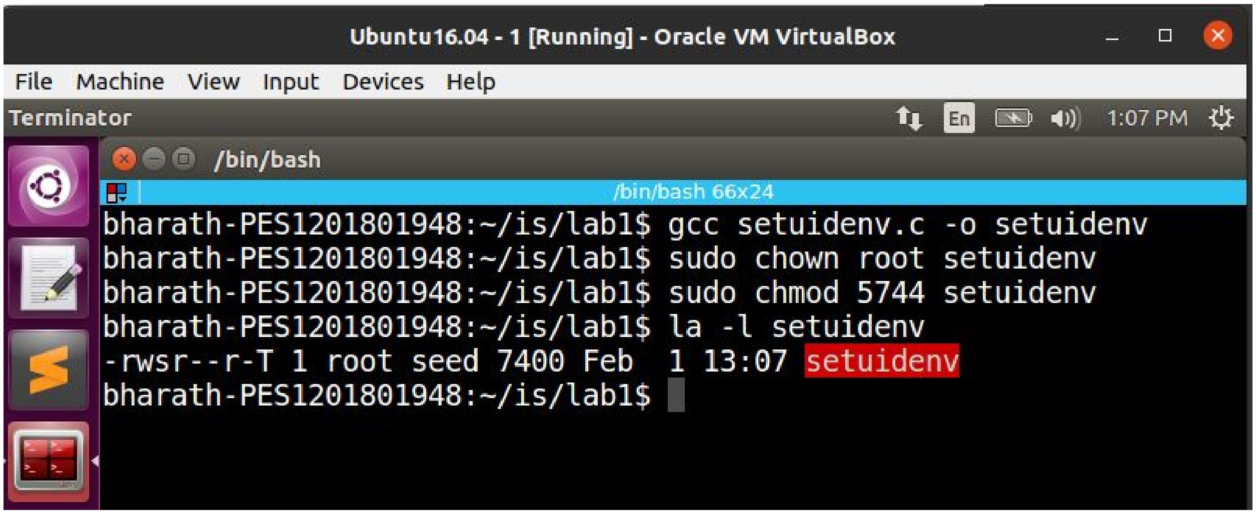
It is run as :
execl("/bin/sh", "sh", "-c", command, (char *) NULL);

Task 5 : Environment Variables and Set-UID programs


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
12:58 PM
/bin/bash
/bin/bash 66x24

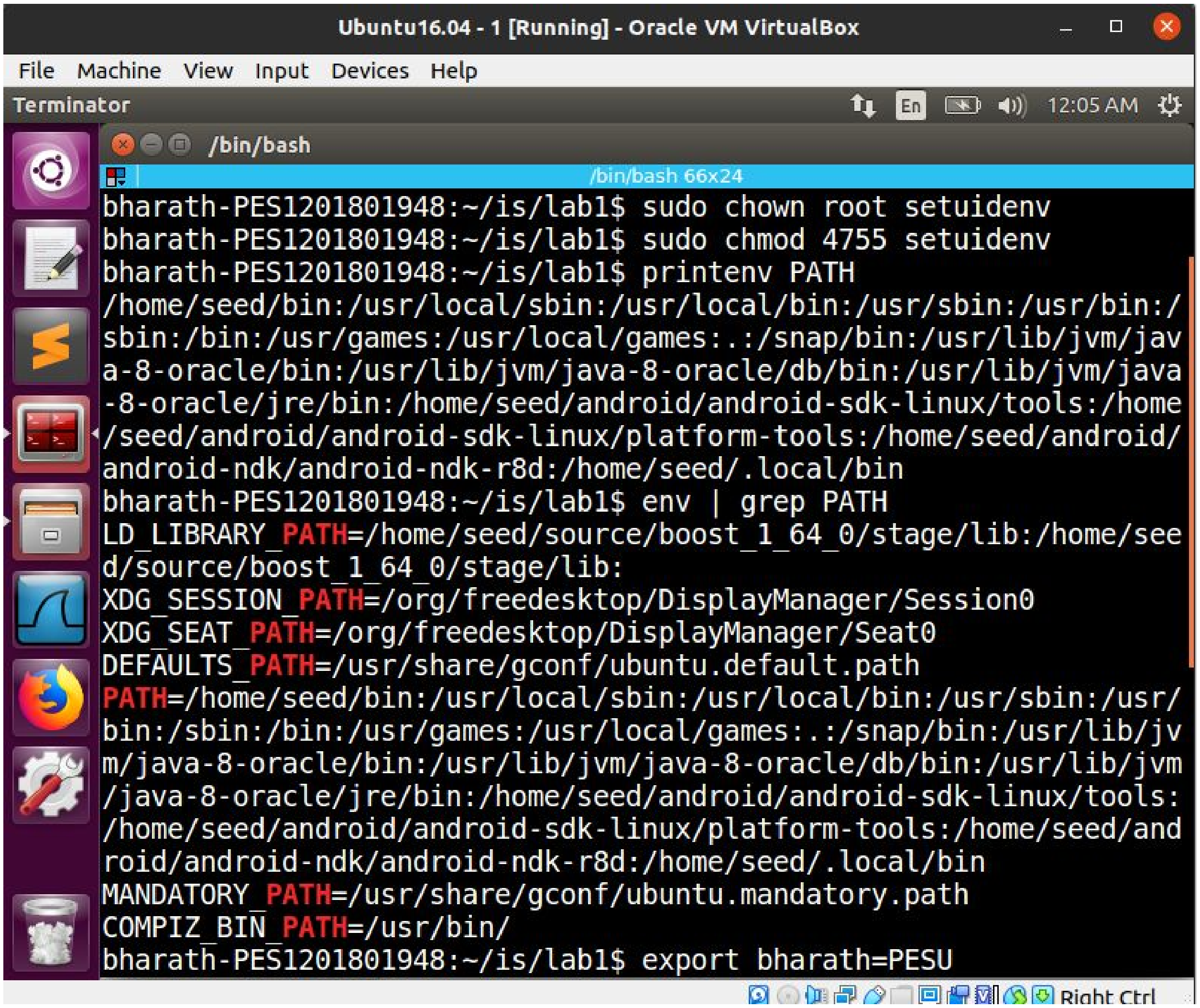
```
bharath-PES1201801948:~/is/lab1$ gcc setuidenv.c -o setuidenv
bharath-PES1201801948:~/is/lab1$ sudo chown root setuidenv
bharath-PES1201801948:~/is/lab1$ sudo chmod 4755 setuidenv
bharath-PES1201801948:~/is/lab1$ la -l setuidenv
-rwsr-xr-x 1 root seed 7400 Feb  1 12:57 setuidenv
bharath-PES1201801948:~/is/lab1$ 
```

The setuidenv executable ownership is changed to root, as well as its access permissions are changed using chmod.


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
1:07 PM
/bin/bash
/bin/bash 66x24

```
bharath-PES1201801948:~/is/lab1$ gcc setuidenv.c -o setuidenv
bharath-PES1201801948:~/is/lab1$ sudo chown root setuidenv
bharath-PES1201801948:~/is/lab1$ sudo chmod 5744 setuidenv
bharath-PES1201801948:~/is/lab1$ la -l setuidenv
-rwsr--r-T 1 root seed 7400 Feb  1 13:07 setuidenv
bharath-PES1201801948:~/is/lab1$ 
```

Owner is changed to root, but the chmod command makes it a set uid program

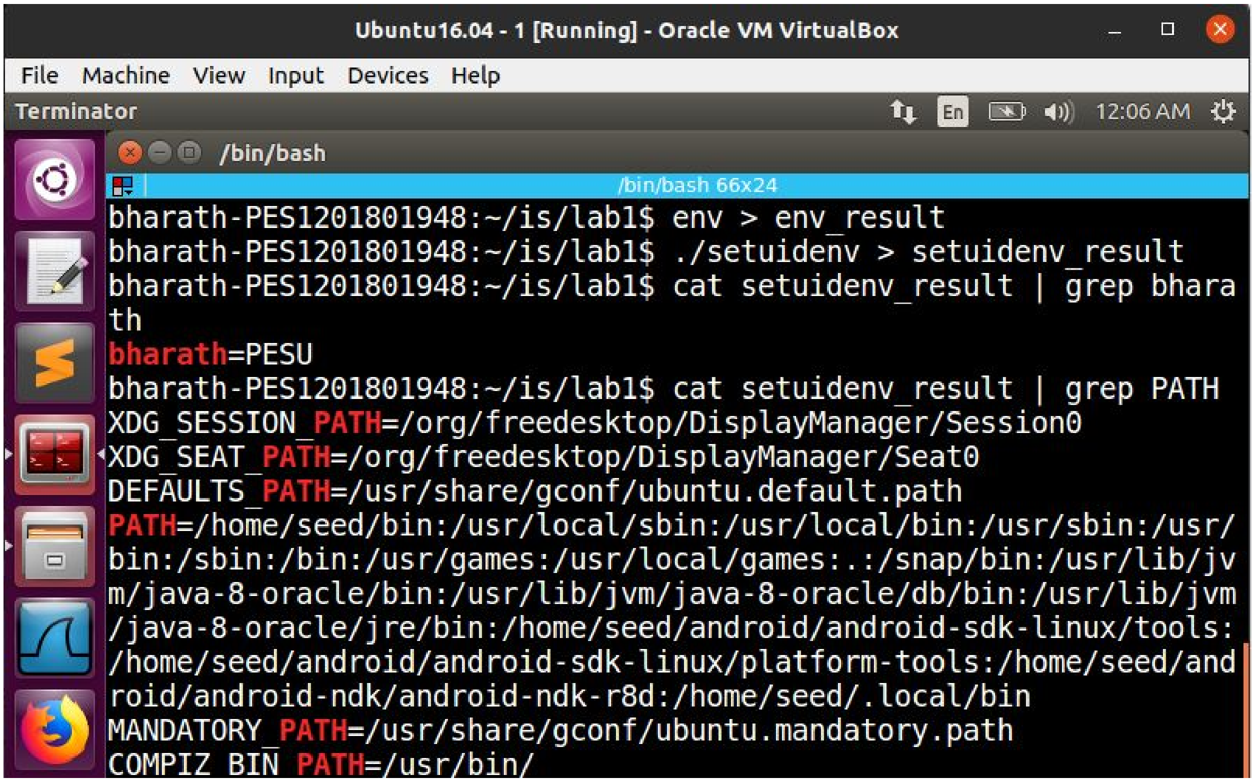

Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
En
12:05 AM
/bin/bash
/bin/bash 66x24
Right Ctrl

```
bharath-PES1201801948:~/is/lab1$ sudo chown root setuidenv
bharath-PES1201801948:~/is/lab1$ sudo chmod 4755 setuidenv
bharath-PES1201801948:~/is/lab1$ printenv PATH
/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/
sbin:/bin:/usr/games:/usr/local/games:.:/snap/bin:/usr/lib/jvm/jav
a-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm/java
-8-oracle/jre/bin:/home/seed/android/android-sdk-linux/tools:/home
/seed/android/android-sdk-linux/platform-tools:/home/seed/android/
android-ndk/android-ndk-r8d:/home/seed/.local/bin
bharath-PES1201801948:~/is/lab1$ env | grep PATH
LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/see
d/source/boost_1_64_0/stage/lib:
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path
PATH=/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/
bin:/sbin:/bin:/usr/games:/usr/local/games:.:/snap/bin:/usr/lib/jv
m/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm
/java-8-oracle/jre/bin:/home/seed/android/android-sdk-linux/tools:
/home/seed/android/android-sdk-linux/platform-tools:/home/seed/and
roid/android-ndk/android-ndk-r8d:/home/seed/.local/bin
MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path
COMPIZ_BIN_PATH=/usr/bin/
bharath-PES1201801948:~/is/lab1$ export bharath=PESU
```

PATH, LD_LIBRARY_PATH are already defined
bharath=PESU is the new env variable exported

The current environment variables are stored in the env_result file
Executing the program setuidenv, and storing the result in setuidenv_result.

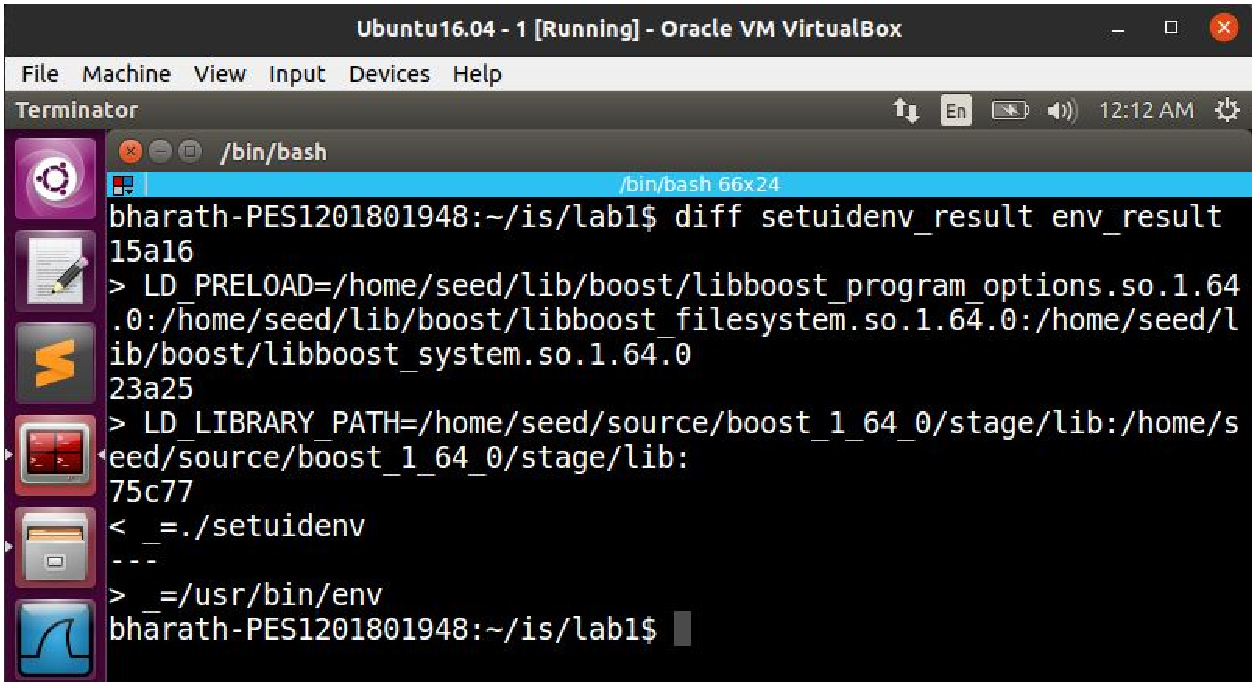
The PATH and the newly defined env variable bharath=PESU is found in the output of the setuidenv_result file, but the LD_LIBRARY_PATH env variable is not contained in it. Implying that the child process inherited the PATH and the new “bharath” variable but not the LD_LIBRARY_PATH.


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
En
12:06 AM
/bin/bash
/bin/bash 66x24

```
bharath-PES1201801948:~/is/lab1$ env > env_result
bharath-PES1201801948:~/is/lab1$ ./setuidenv > setuidenv_result
bharath-PES1201801948:~/is/lab1$ cat setuidenv_result | grep bhara
th
bharath=PESU
bharath-PES1201801948:~/is/lab1$ cat setuidenv_result | grep PATH
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path
PATH=/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/
bin:/sbin:/bin:/usr/games:/usr/local/games:.:/snap/bin:/usr/lib/jv
m/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm
/java-8-oracle/jre/bin:/home/seed/android/android-sdk-linux/tools:
/home/seed/android/android-sdk-linux/platform-tools:/home/seed/and
roid/android-ndk/android-ndk-r8d:/home/seed/.local/bin
MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path
COMPIZ_BIN_PATH=/usr/bin/
```

One reason for LD_LIBRARY_PATH not being inherited is that the real uid and the effective uid are different. As shown below,

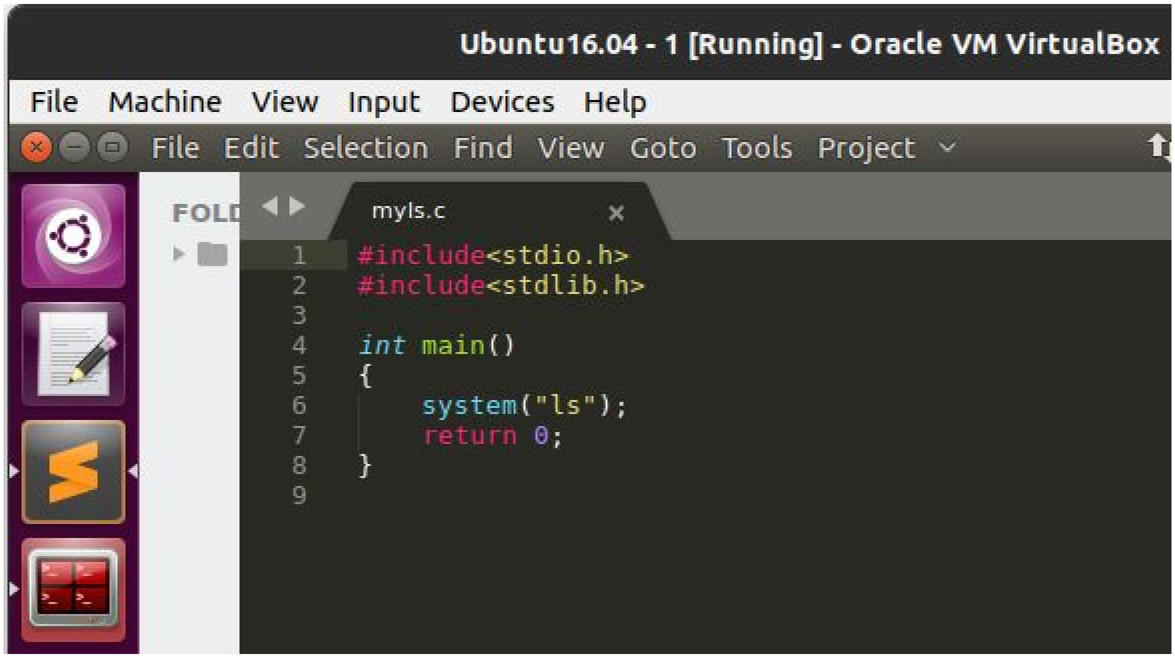
Running diff command on the 2 files,


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
En
12:12 AM
/bin/bash
/bin/bash 66x24

```
bharath-PES1201801948:~/is/lab1$ diff setuidenv_result env_result
15a16
> LD_PRELOAD=/home/seed/lib/boost/libboost_program_options.so.1.64
.0:/home/seed/lib/boost/libboost_filesystem.so.1.64.0:/home/seed/l
ib/boost/libboost_system.so.1.64.0
23a25
> LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/s
eed/source/boost_1_64_0/stage/lib:
75c77
< _=./setuidenv
---
> _=/usr/bin/env
bharath-PES1201801948:~/is/lab1$
```

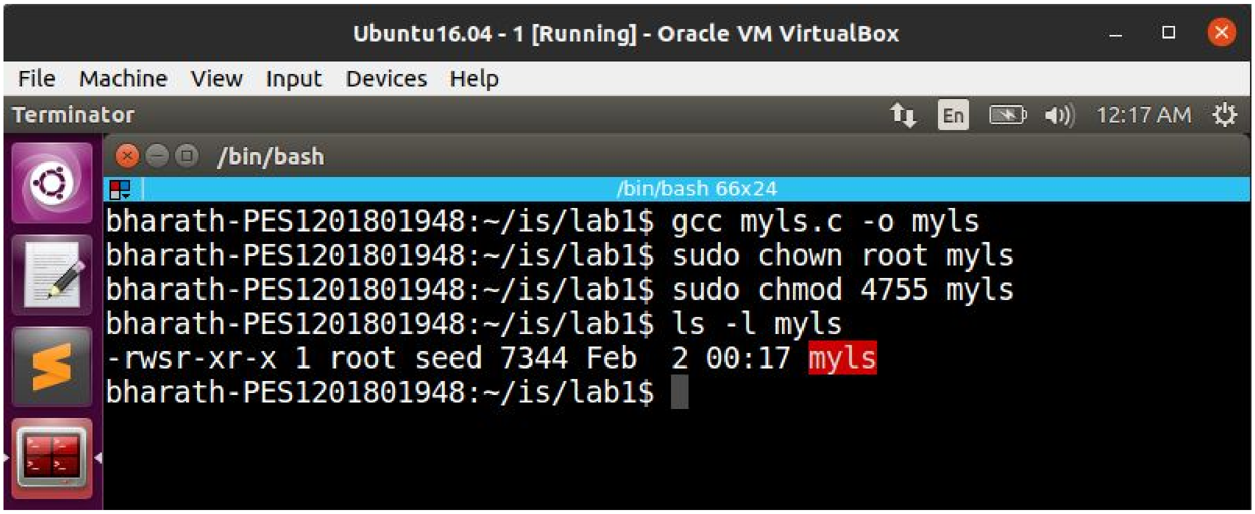
Task 6 : The PATH Environment variable and Set-UID Programs

myls.c


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Edit Selection Find View Goto Tools Project
FOLD
myls.c

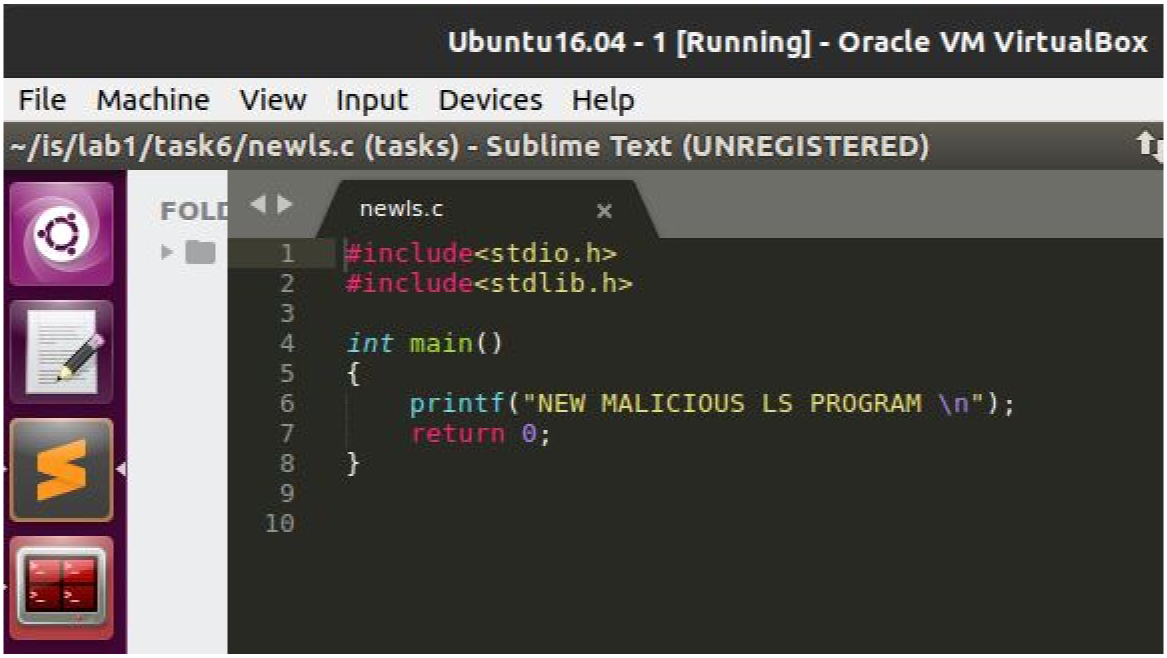
```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    system("ls");
    return 0;
}

```

Compiling
Changing its owner to root
Making it a setuid program


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
En
12:17 AM
/bin/bash
/bin/bash 66x24

```
bharath-PES1201801948:~/is/lab1$ gcc myls.c -o myls
bharath-PES1201801948:~/is/lab1$ sudo chown root myls
bharath-PES1201801948:~/is/lab1$ sudo chmod 4755 myls
bharath-PES1201801948:~/is/lab1$ ls -l myls
-rwsr-xr-x 1 root seed 7344 Feb  2 00:17 myls
bharath-PES1201801948:~/is/lab1$ 
```

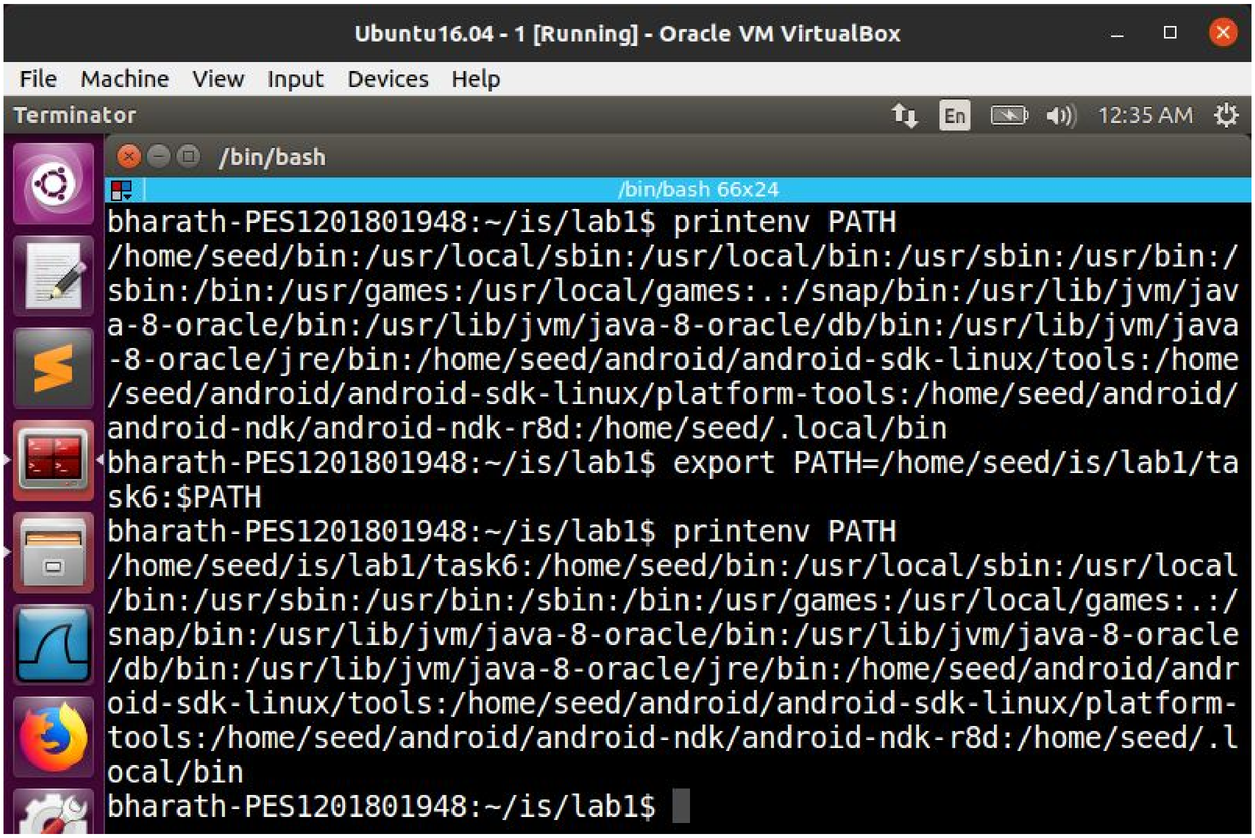
newls.c



Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
~/is/lab1/task6/newls.c (tasks) - Sublime Text (UNREGISTERED)
FOLD
newls.c

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    printf("NEW MALICIOUS LS PROGRAM \n");
    return 0;
}


```

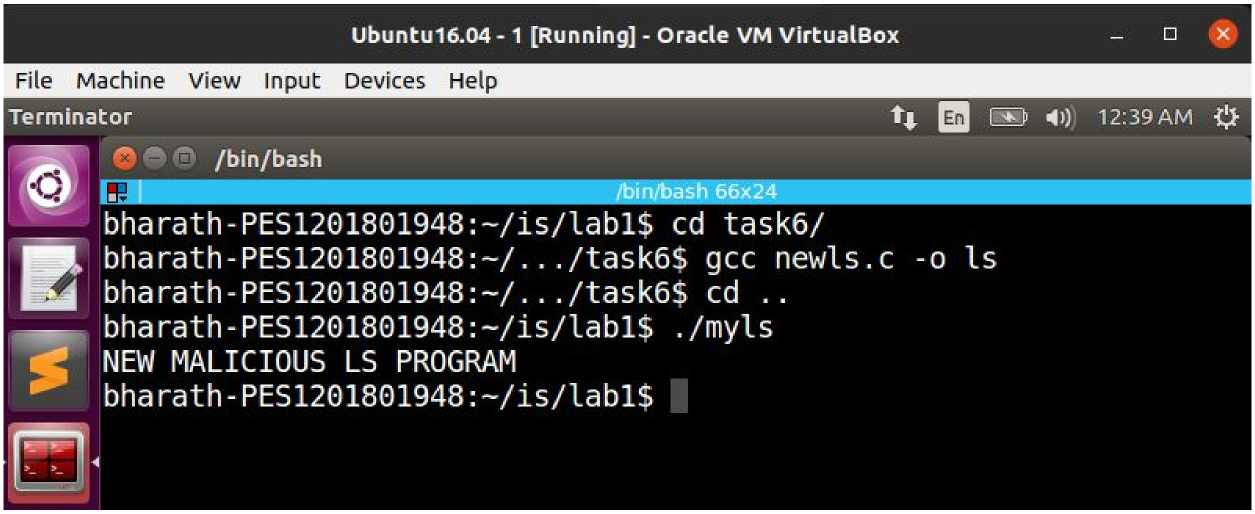
Changing the PATH variable to point to the task6 directory which contains the malicious file “newls.c“



Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
En
12:35 AM
/bin/bash
/bin/bash 66x24

```
bharath-PES1201801948:~/is/lab1$ printenv PATH
/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/
sbin:/bin:/usr/games:/usr/local/games:.:/snap/bin:/usr/lib/jvm/jav
a-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm/java
-8-oracle/jre/bin:/home/seed/android/android-sdk-linux/tools:/home
/seed/android/android-sdk-linux/platform-tools:/home/seed/android/
android-ndk/android-ndk-r8d:/home/seed/.local/bin
bharath-PES1201801948:~/is/lab1$ export PATH=/home/seed/is/lab1/ta
sk6:$PATH
bharath-PES1201801948:~/is/lab1$ printenv PATH
/home/seed/is/lab1/task6:/home/seed/bin:/usr/local/sbin:/usr/local
/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:.:/
snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle
/db/bin:/usr/lib/jvm/java-8-oracle/jre/bin:/home/seed/android/andr
oid-sdk-linux/tools:/home/seed/android/android-sdk-linux/platform-
tools:/home/seed/android/android-ndk/android-ndk-r8d:/home/seed/.l
ocal/bin
bharath-PES1201801948:~/is/lab1$ 
```

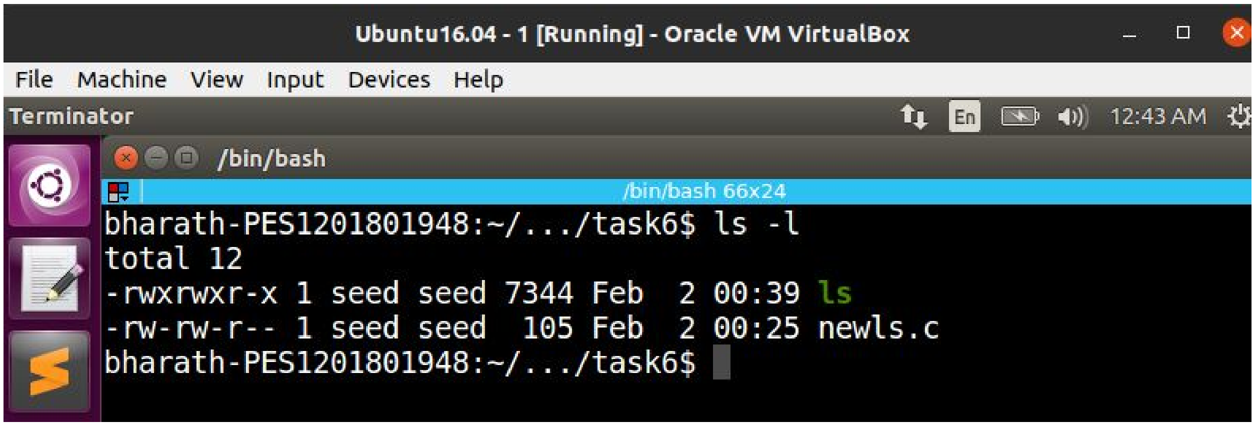
Compiling the malicious ls program “newls.c” and naming its executable “ls”


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
En
12:39 AM
/bin/bash
/bin/bash 66x24

```
bharath-PES1201801948:~/is/lab1$ cd task6/
bharath-PES1201801948:~/.../task6$ gcc newls.c -o ls
bharath-PES1201801948:~/.../task6$ cd ..
bharath-PES1201801948:~/is/lab1$ ./myls
NEW MALICIOUS LS PROGRAM
bharath-PES1201801948:~/is/lab1$
```

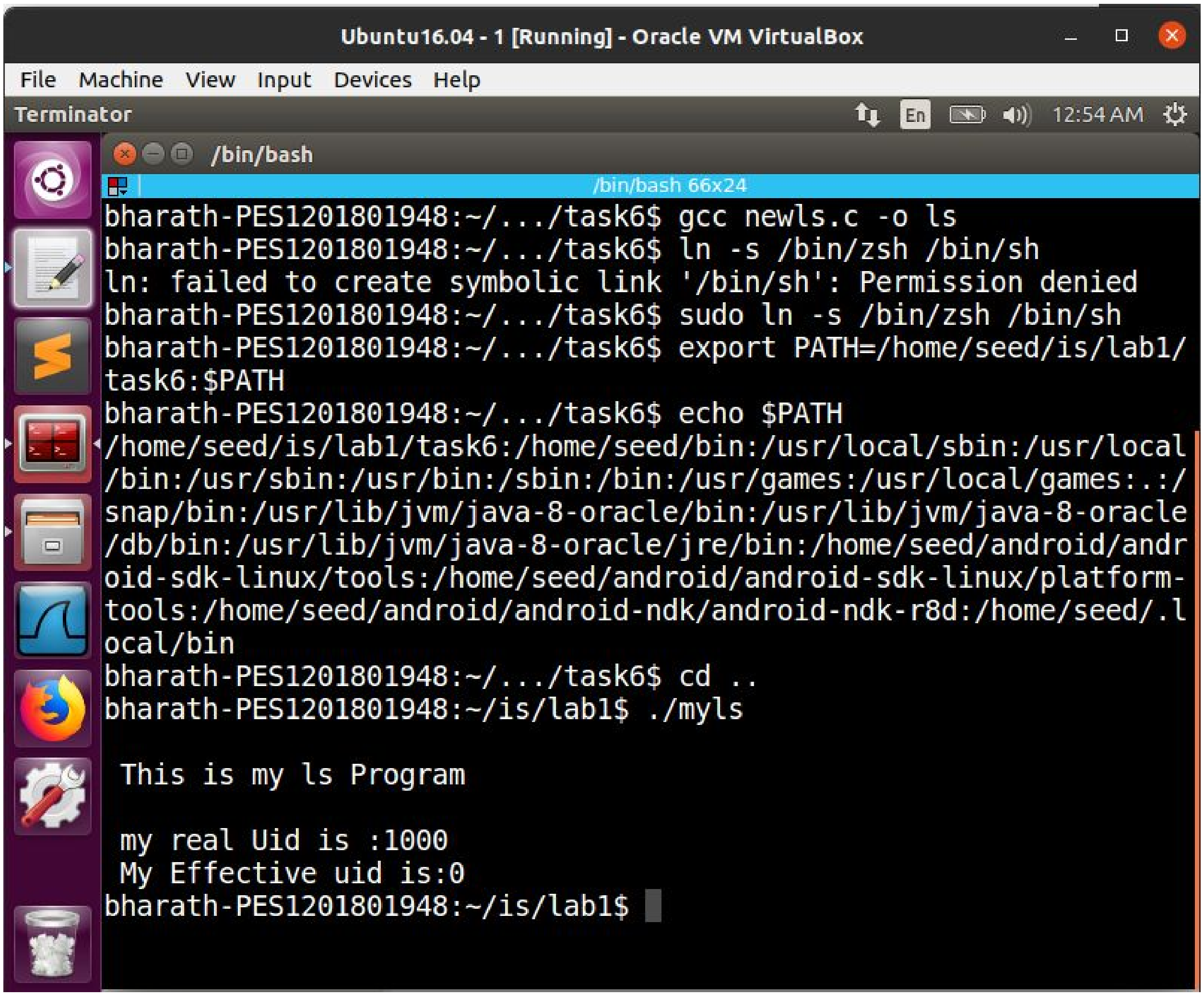
Running the original file, inturn runs the malicious file as the PATH variable was changed to point to the directory containing the malicious program. Implying, it will first search in that directory, and as the executable is named as ‘ls’, it is executed.

Checking the permissions of the newls.c and its executable,


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
En
12:43 AM
/bin/bash
/bin/bash 66x24

```
bharath-PES1201801948:~/.../task6$ ls -l
total 12
-rwxrwxr-x 1 seed seed 7344 Feb  2 00:39 ls
-rw-rw-r-- 1 seed seed  105 Feb  2 00:25 newls.c
bharath-PES1201801948:~/.../task6$
```

It is the seed user, and not the root
But it still ran the myls executable whose owner is root, implying that using system() and relative path together in a setuid program can be very harmful and dangerous to the user system.


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
En
12:54 AM
/bin/bash
/bin/bash 66x24

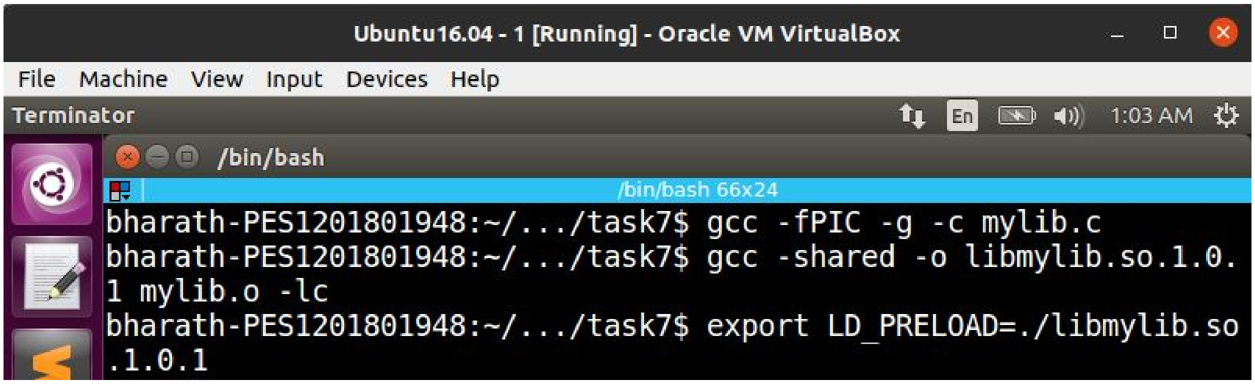
```
bharath-PES1201801948:~/.../task6$ gcc newls.c -o ls
bharath-PES1201801948:~/.../task6$ ln -s /bin/zsh /bin/sh
ln: failed to create symbolic link '/bin/sh': Permission denied
bharath-PES1201801948:~/.../task6$ sudo ln -s /bin/zsh /bin/sh
bharath-PES1201801948:~/.../task6$ export PATH=/home/seed/is/lab1/
task6:$PATH
bharath-PES1201801948:~/.../task6$ echo $PATH
/home/seed/is/lab1/task6:/home/seed/bin:/usr/local/sbin:/usr/local
/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:.:/
snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle
/db/bin:/usr/lib/jvm/java-8-oracle/jre/bin:/home/seed/android/andr
oid-sdk-linux/tools:/home/seed/android/android-sdk-linux/platform-
tools:/home/seed/android/android-ndk/android-ndk-r8d:/home/seed/.l
ocal/bin
bharath-PES1201801948:~/.../task6$ cd ..
bharath-PES1201801948:~/is/lab1$ ./myls

 This is my ls Program

 my real Uid is :1000
 My Effective uid is:0
bharath-PES1201801948:~/is/lab1$ 
```

This helps us understand that running the myls executable runs the malicious ls program newls.c.

Thereby running a file owned by root, even though the attacker just has user privileges.

Task 7 : The LD PRELOAD environment variable and Set-UID Programs

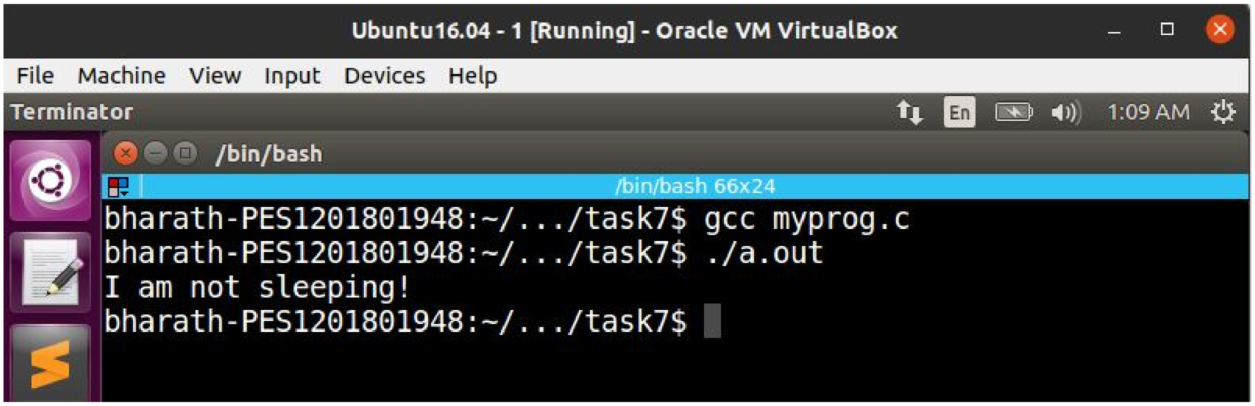
Setting up a new library for the sleep() function,



Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
En
1:03 AM
/bin/bash
/bin/bash 66x24

```
bharath-PES1201801948:~/.../task7$ gcc -fPIC -g -c mylib.c
bharath-PES1201801948:~/.../task7$ gcc -shared -o libmylib.so.1.0.
1 mylib.o -lc
bharath-PES1201801948:~/.../task7$ export LD_PRELOAD=./libmylib.so
.1.0.1
```

Compiling a program containing sleep()

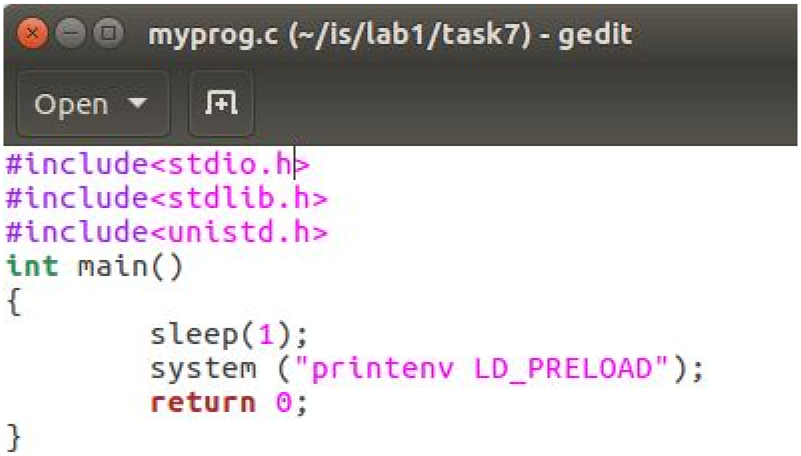


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
En
1:09 AM
/bin/bash
/bin/bash 66x24

```
bharath-PES1201801948:~/.../task7$ gcc myprog.c
bharath-PES1201801948:~/.../task7$ ./a.out
I am not sleeping!
bharath-PES1201801948:~/.../task7$
```

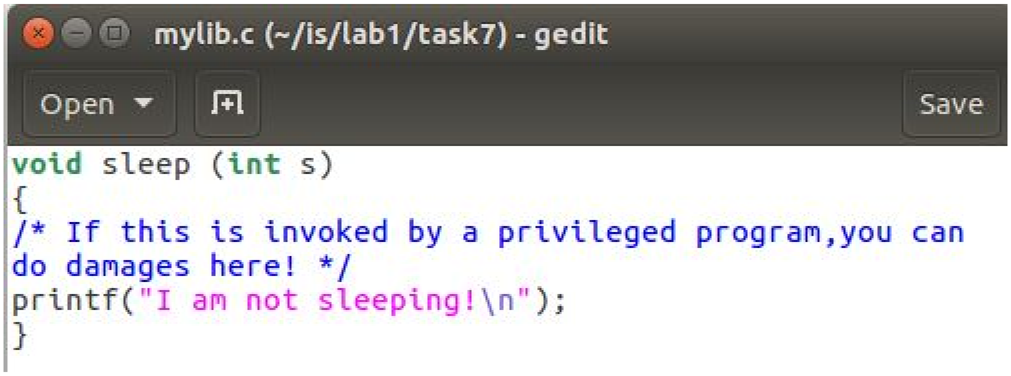
Therefore, it runs the sleep function in the library we provided

Running in different scenarios :
To understand the working better, i invoked the printenv LD_PRELOAD to check which library is loaded/used.
Program : myprog.c



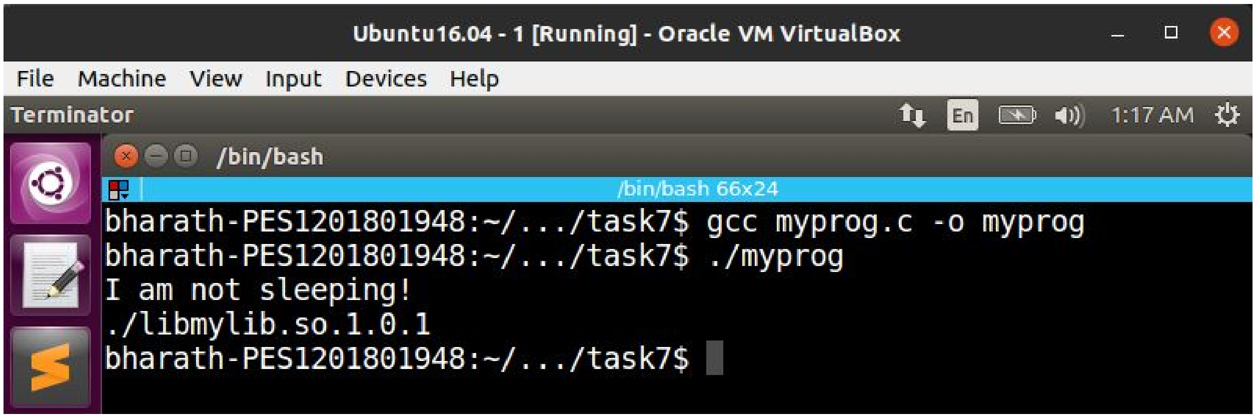
```c
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
int main()
{
        sleep(1);
        system ("printenv LD_PRELOAD");
        return 0;
}
```

Lib : mylib.c



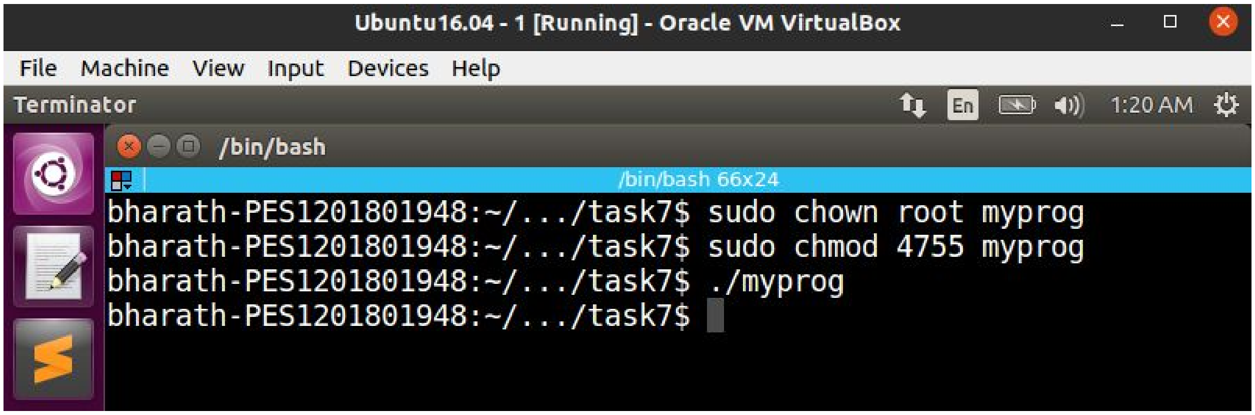
```c
void sleep (int s)
{
/* If this is invoked by a privileged program,you can
do damages here! */
printf("I am not sleeping!\n");
}
```

1. Myprog -> regular , Run as normal user



```
bharath-PES1201801948:~/.../task7$ gcc myprog.c -o myprog
bharath-PES1201801948:~/.../task7$ ./myprog
I am not sleeping!
./libmylib.so.1.0.1
bharath-PES1201801948:~/.../task7$
```

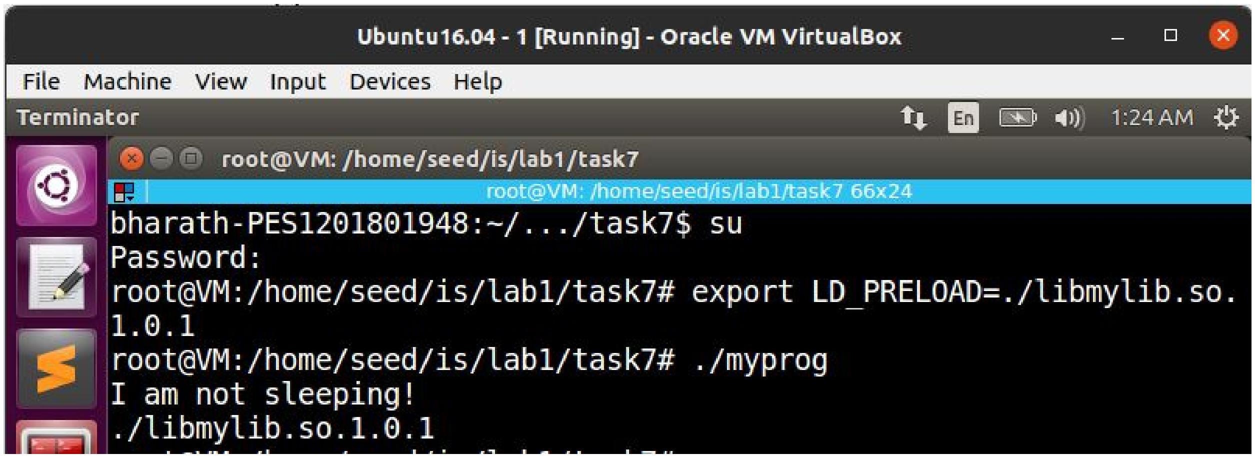
2. Myprog -> setuid, run as normal user


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
1:20 AM
/bin/bash
/bin/bash 66x24

```
bharath-PES1201801948:~/.../task7$ sudo chown root myprog
bharath-PES1201801948:~/.../task7$ sudo chmod 4755 myprog
bharath-PES1201801948:~/.../task7$ ./myprog
bharath-PES1201801948:~/.../task7$ 
```

Making myprog a setuid program and running it as normal user, it uses the system defined sleep() function

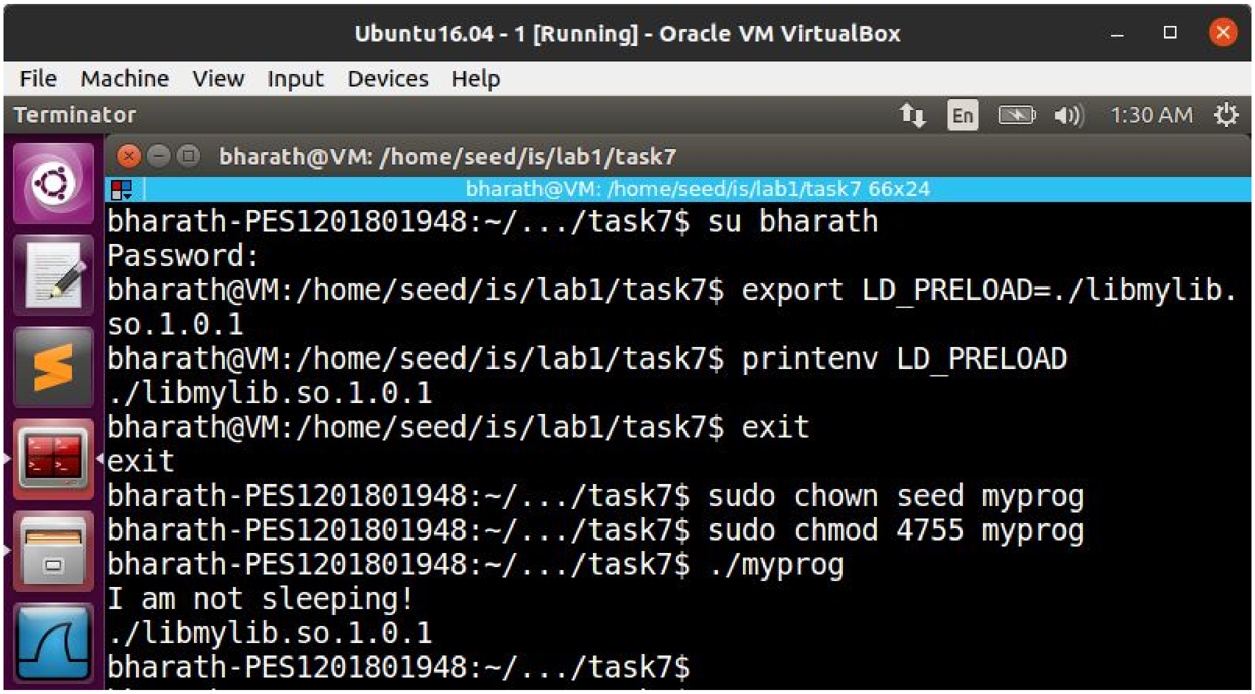
3. myprog a Set-UID root program, export the LD PRELOAD environment variable again in the root account and run it.


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
1:24 AM
root@VM: /home/seed/is/lab1/task7
root@VM: /home/seed/is/lab1/task7 66x24

```
bharath-PES1201801948:~/.../task7$ su
Password:
root@VM:/home/seed/is/lab1/task7# export LD_PRELOAD=./libmylib.so.
1.0.1
root@VM:/home/seed/is/lab1/task7# ./myprog
I am not sleeping!
./libmylib.so.1.0.1
```

The program uses the user-defined sleep() function

4.Make myprog a Set-UID user1 program (i.e., the owner is user1, which is another user account), export the LD PRELOAD environment variable again in a different user's account (not-root user) and run it.


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
En
1:30 AM
bharath@VM: /home/seed/is/lab1/task7
bharath@VM: /home/seed/is/lab1/task7 66x24

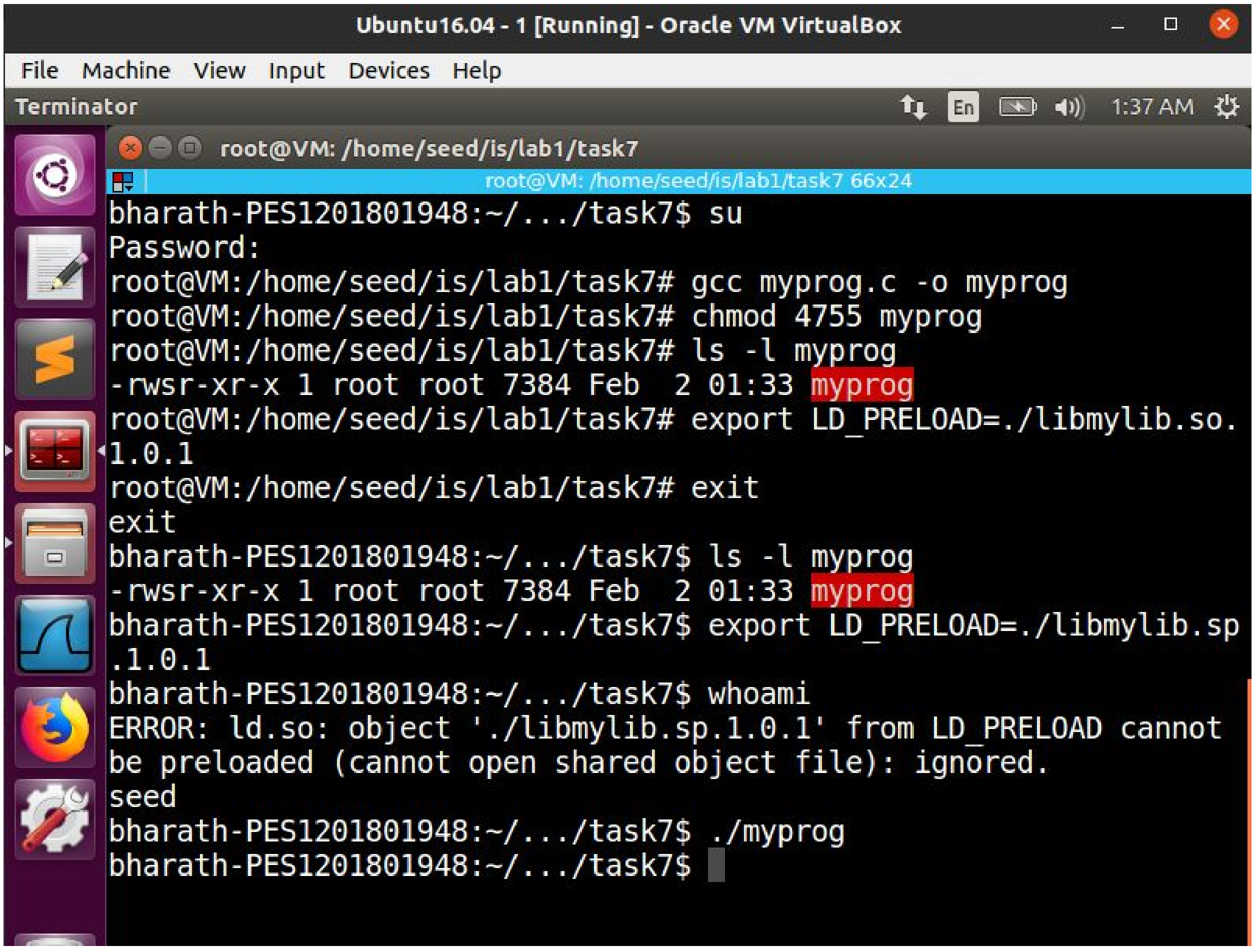
```
bharath-PES1201801948:~/.../task7$ su bharath
Password:
bharath@VM:/home/seed/is/lab1/task7$ export LD_PRELOAD=./libmylib.
so.1.0.1
bharath@VM:/home/seed/is/lab1/task7$ printenv LD_PRELOAD
./libmylib.so.1.0.1
bharath@VM:/home/seed/is/lab1/task7$ exit
exit
bharath-PES1201801948:~/.../task7$ sudo chown seed myprog
bharath-PES1201801948:~/.../task7$ sudo chmod 4755 myprog
bharath-PES1201801948:~/.../task7$ ./myprog
I am not sleeping!
./libmylib.so.1.0.1
bharath-PES1201801948:~/.../task7$
```

Exporting in the user account “bharath” , changing the owner to seed and running it, it uses the user-defined sleep() function.

We can understand the outputs in the following way :
1, 3, 4 examples use the library we defined, while 2, uses the system library

This happens due to the security mechanism of setuid.
Wherever the real uid and effective uid is the same, LD_PRELOAD is present.
In 1, 3, 4 the euid and ruid were always the same, hence why our user-defined sleep function is called.

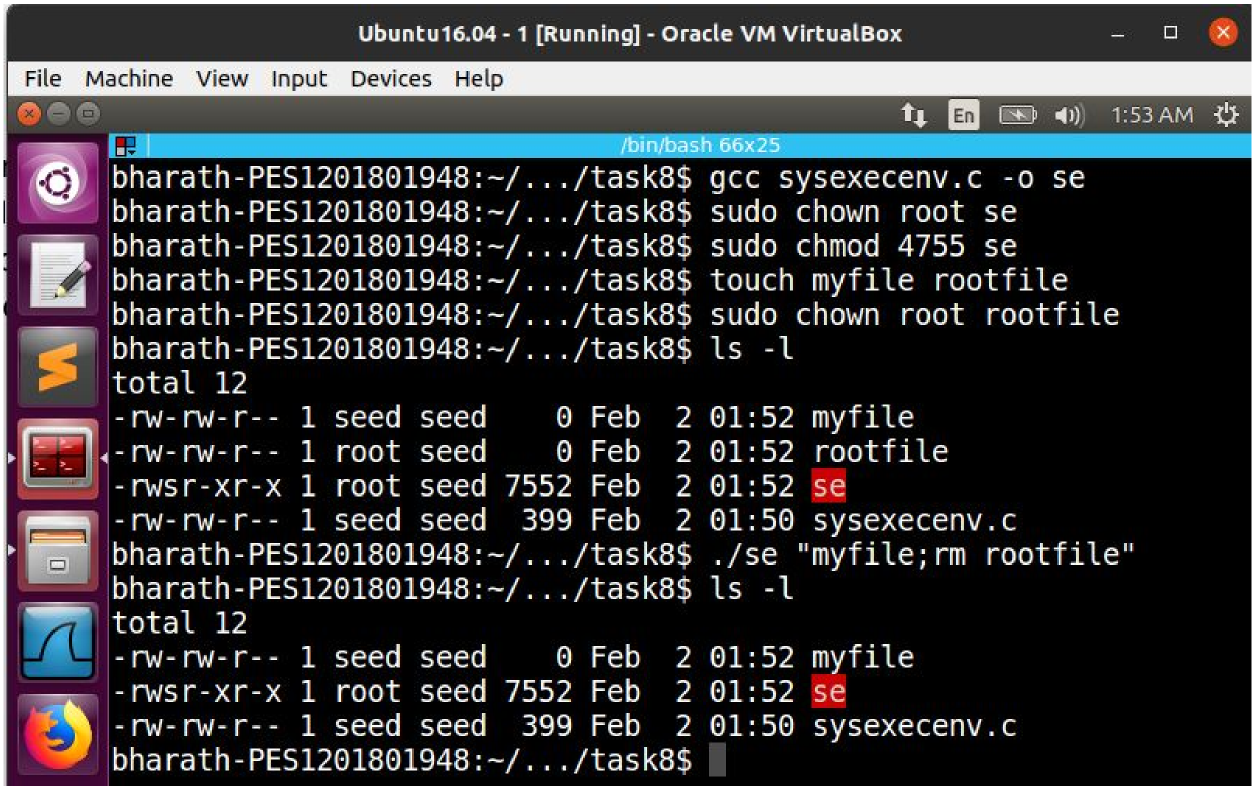
Whereas, in 2
euid and ruid is different, thereby the LD_PRELOAD env variable is dropped. Hence it uses the system library.


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
En
1:37 AM
root@VM: /home/seed/is/lab1/task7
root@VM: /home/seed/is/lab1/task7 66x24

```
bharath-PES1201801948:~/.../task7$ su
Password:
root@VM:/home/seed/is/lab1/task7# gcc myprog.c -o myprog
root@VM:/home/seed/is/lab1/task7# chmod 4755 myprog
root@VM:/home/seed/is/lab1/task7# ls -l myprog
-rwsr-xr-x 1 root root 7384 Feb  2 01:33 myprog
root@VM:/home/seed/is/lab1/task7# export LD_PRELOAD=./libmylib.so.
1.0.1
root@VM:/home/seed/is/lab1/task7# exit
exit
bharath-PES1201801948:~/.../task7$ ls -l myprog
-rwsr-xr-x 1 root root 7384 Feb  2 01:33 myprog
bharath-PES1201801948:~/.../task7$ export LD_PRELOAD=./libmylib.sp
.1.0.1
bharath-PES1201801948:~/.../task7$ whoami
ERROR: ld.so: object './libmylib.sp.1.0.1' from LD_PRELOAD cannot
be preloaded (cannot open shared object file): ignored.
seed
bharath-PES1201801948:~/.../task7$ ./myprog
bharath-PES1201801948:~/.../task7$ 
```

Since LD_PRELOAD is already exported once in root, the export in the user space isnt allowed, implying again that the euid and ruid are differing, therefore it uses the system sleep() function.

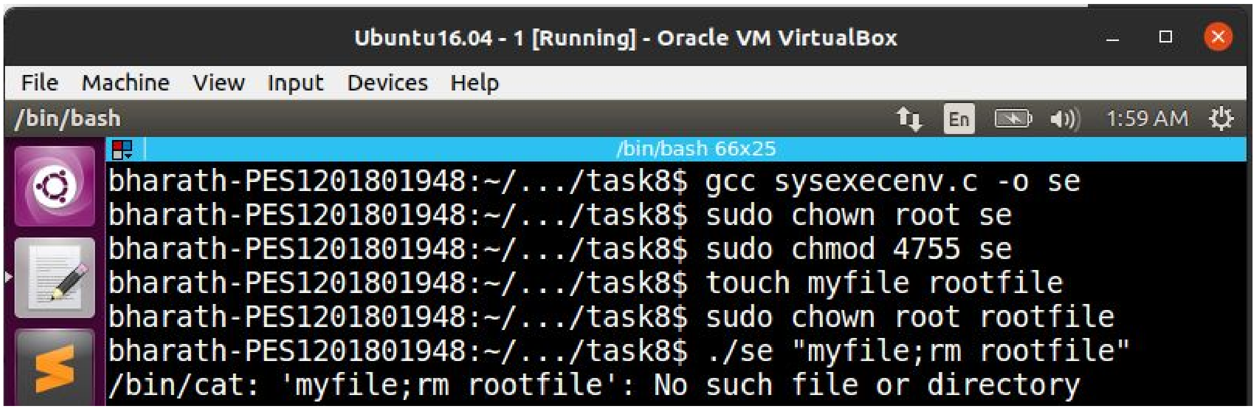
Task 8 : Invoking external programs using system() versus execve()


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1:53 AM
/bin/bash 66x25

```
bharath-PES1201801948:~/.../task8$ gcc sysexecenv.c -o se
bharath-PES1201801948:~/.../task8$ sudo chown root se
bharath-PES1201801948:~/.../task8$ sudo chmod 4755 se
bharath-PES1201801948:~/.../task8$ touch myfile rootfile
bharath-PES1201801948:~/.../task8$ sudo chown root rootfile
bharath-PES1201801948:~/.../task8$ ls -l
total 12
-rw-rw-r-- 1 seed seed    0 Feb  2 01:52 myfile
-rw-rw-r-- 1 root seed    0 Feb  2 01:52 rootfile
-rwsr-xr-x 1 root seed 7552 Feb  2 01:52 se
-rw-rw-r-- 1 seed seed  399 Feb  2 01:50 sysexecenv.c
bharath-PES1201801948:~/.../task8$ ./se "myfile;rm rootfile"
bharath-PES1201801948:~/.../task8$ ls -l
total 12
-rw-rw-r-- 1 seed seed    0 Feb  2 01:52 myfile
-rwsr-xr-x 1 root seed 7552 Feb  2 01:52 se
-rw-rw-r-- 1 seed seed  399 Feb  2 01:50 sysexecenv.c
bharath-PES1201801948:~/.../task8$
```

Yes, the rootfile was deleted, since we were using the system() command

Using the execv

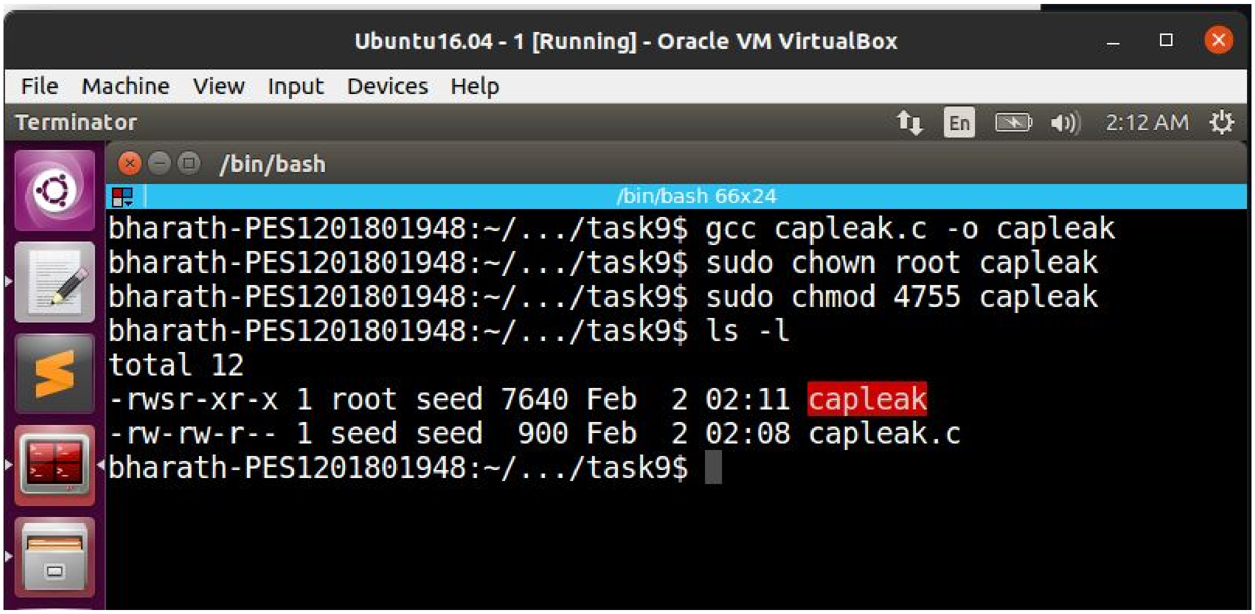

Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
/bin/bash
1:59 AM
/bin/bash 66x25

```
bharath-PES1201801948:~/.../task8$ gcc sysexecenv.c -o se
bharath-PES1201801948:~/.../task8$ sudo chown root se
bharath-PES1201801948:~/.../task8$ sudo chmod 4755 se
bharath-PES1201801948:~/.../task8$ touch myfile rootfile
bharath-PES1201801948:~/.../task8$ sudo chown root rootfile
bharath-PES1201801948:~/.../task8$ ./se "myfile;rm rootfile"
/bin/cat: 'myfile;rm rootfile': No such file or directory
```

No, we cannot remove the root file since the whole string is take as file input

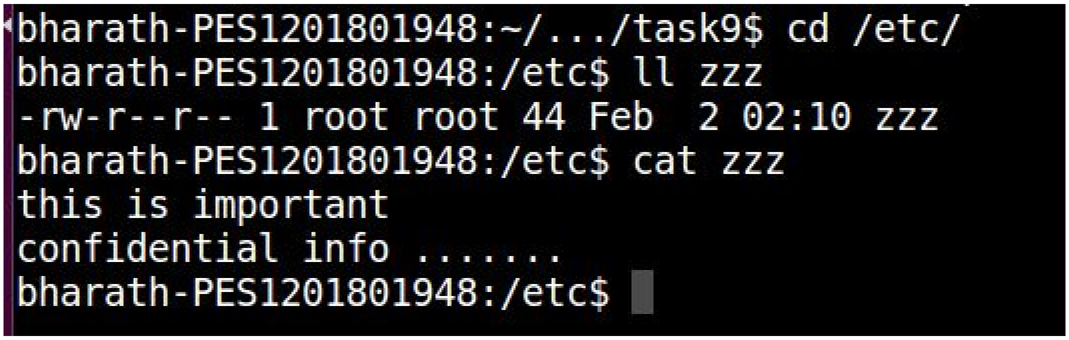
Therefore, we can conclude that system() has not input validation, whereas execve has some kind of minimal input validation.

Task 9 : Capability Leaking


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
En
2:12 AM
/bin/bash
/bin/bash 66x24

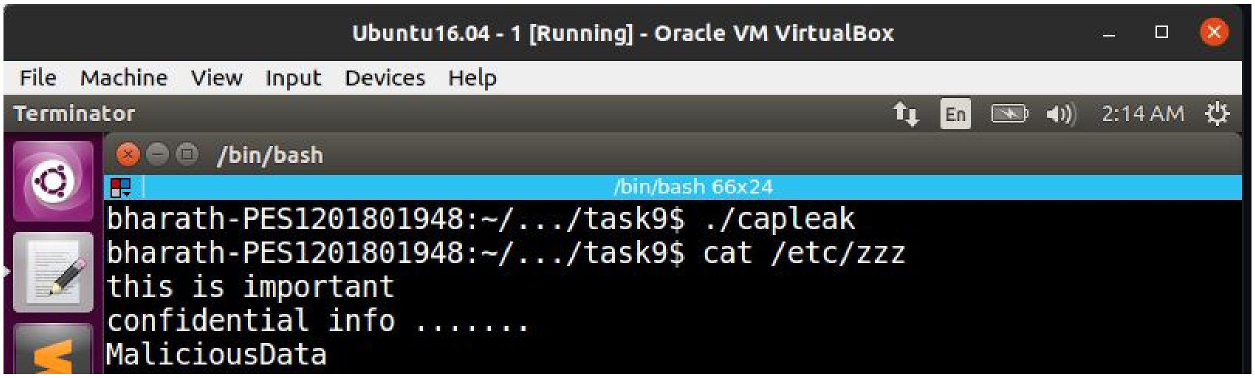
```
bharath-PES1201801948:~/.../task9$ gcc capleak.c -o capleak
bharath-PES1201801948:~/.../task9$ sudo chown root capleak
bharath-PES1201801948:~/.../task9$ sudo chmod 4755 capleak
bharath-PES1201801948:~/.../task9$ ls -l
total 12
-rwsr-xr-x 1 root seed 7640 Feb  2 02:11 capleak
-rw-rw-r-- 1 seed seed  900 Feb  2 02:08 capleak.c
bharath-PES1201801948:~/.../task9$
```

Compiling and making the file a setuid program



```
bharath-PES1201801948:~/.../task9$ cd /etc/
bharath-PES1201801948:/etc$ ll zzz
-rw-r--r-- 1 root root 44 Feb  2 02:10 zzz
bharath-PES1201801948:/etc$ cat zzz
this is important
confidential info .......
bharath-PES1201801948:/etc$
```

Created a file called zzz with confidential information


Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
En
2:14 AM
/bin/bash
/bin/bash 66x24

```
bharath-PES1201801948:~/.../task9$ ./capleak
bharath-PES1201801948:~/.../task9$ cat /etc/zzz
this is important
confidential info .......
MaliciousData
```

Running the file,we can see that the malicious data got appended to the confidential file, thereby it losing its integrity. This can be prevented by closing the file after its appropriate usage, to have the right permissions.