

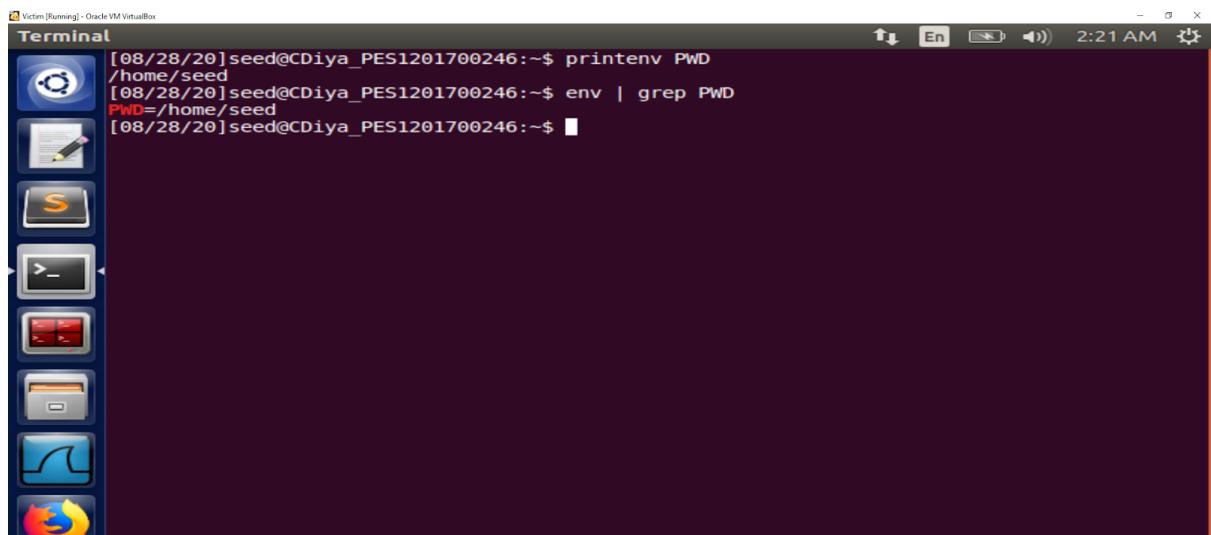
Information Security

LABORATORY 1

Environment Variable and Set-UID Program Lab

C Diya
PES1201700246

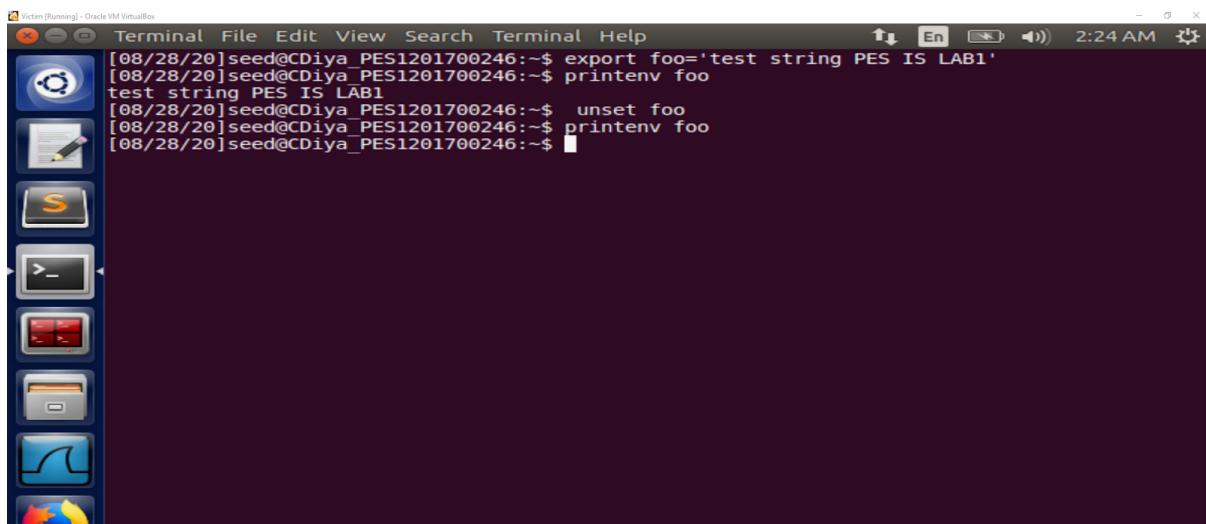
TASK 1: Manipulating environment variables



The screenshot shows a terminal window titled "Terminal" running on a Linux desktop. The desktop environment includes icons for a terminal, file manager, browser, and other applications. The terminal window displays the following command-line session:

```
[08/28/20]seed@CDiya_PES1201700246:~$ printenv PWD  
/home/seed  
[08/28/20]seed@CDiya_PES1201700246:~$ env | grep PWD  
PWD=/home/seed  
[08/28/20]seed@CDiya_PES1201700246:~$ █
```

Observation: The printenv or the env command is used to print the environment variables. It can be clearly seen from the screenshot above that when the printenv PWD command is executed, the value of PWD is printed. PWD is an environment variable which stores the path of the current directory. The env command can be used in a similar manner by using grep to find the environment variable.

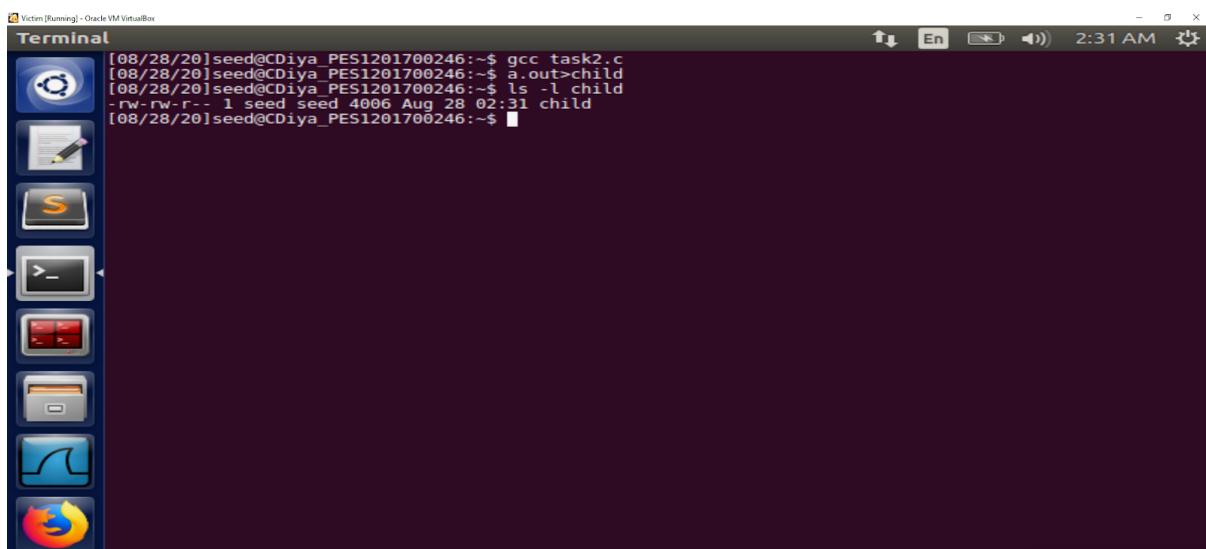
A screenshot of an Ubuntu desktop environment. A terminal window is open in the top panel, showing the following command-line session:

```
[08/28/20]seed@CDiya_PES1201700246:~$ export foo='test string PES IS LAB1'  
[08/28/20]seed@CDiya_PES1201700246:~$ printenv foo  
test string PES IS LAB1  
[08/28/20]seed@CDiya_PES1201700246:~$ unset foo  
[08/28/20]seed@CDiya_PES1201700246:~$ printenv foo  
[08/28/20]seed@CDiya_PES1201700246:~$
```

The desktop panel includes icons for Dash, Home, Screenshot, Terminal, Dash Home, Dash Help, and the Dash search bar.

Observation: The screenshot below shows the setting and unsetting of environment variables. The export command can be used for this. A test environment variable foo is set with a variable and printenv verifies that it is now an environment variable. The unset command can be used to unset foo as an environment variable.

TASK 2: Inheriting environment variables from parents

A screenshot of an Ubuntu desktop environment. A terminal window is open in the top panel, showing the following command-line session:

```
[08/28/20]seed@CDiya_PES1201700246:~$ gcc task2.c  
[08/28/20]seed@CDiya_PES1201700246:~$ a.out>child  
[08/28/20]seed@CDiya_PES1201700246:~$ ls -l child  
-rw-rw-r-- 1 seed seed 4006 Aug 28 02:31 child  
[08/28/20]seed@CDiya_PES1201700246:~$
```

The desktop panel includes icons for Dash, Home, Screenshot, Terminal, Dash Home, Dash Help, and the Dash search bar.

Observation: the program compiled above uses the fork() unix call to create a child process from a parent process. The screenshot above prints the environment variables of the child process using the printenv command and the output is saved in a file

```
[08/28/20]seed@CDiya_PES1201700246:~$ gedit task2.c
[08/28/20]seed@CDiya_PES1201700246:~$ gcc task2.c
[08/28/20]seed@CDiya_PES1201700246:~$ a.out>parent
[08/28/20]seed@CDiya_PES1201700246:~$ ls -l parent
-rw-rw-r-- 1 seed seed 4066 Aug 28 02:34 parent
[08/28/20]seed@CDiya_PES1201700246:~$
```

Observation: The screenshot above prints the environment variables of the parent process and the result is saved on an output file.

The screenshot shows a Linux desktop environment with a terminal window titled "Terminal". The terminal displays the following command and output:

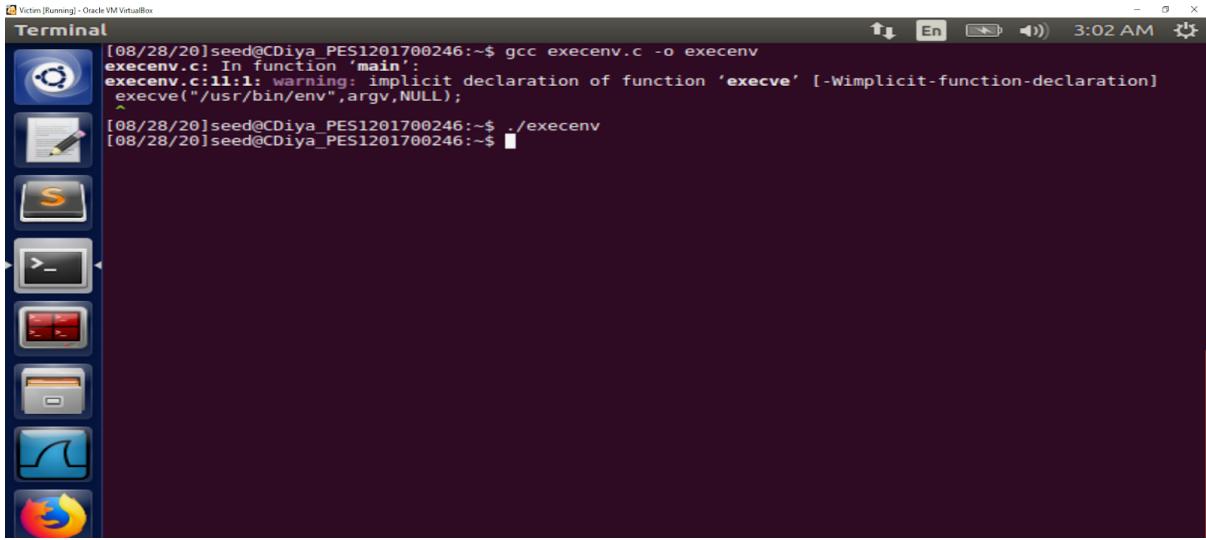
```
[08/28/20]seed@CDiya_PES1201700246:~$ diff parent child
[08/28/20]seed@CDiya_PES1201700246:~$
```

The desktop interface includes a vertical dock on the left containing icons for various applications like a file manager, terminal, and browser. The top bar shows the window title "Victim [Running] - Oracle VM VirtualBox", system status icons, and the current time "2:54 AM".

Final Observation:

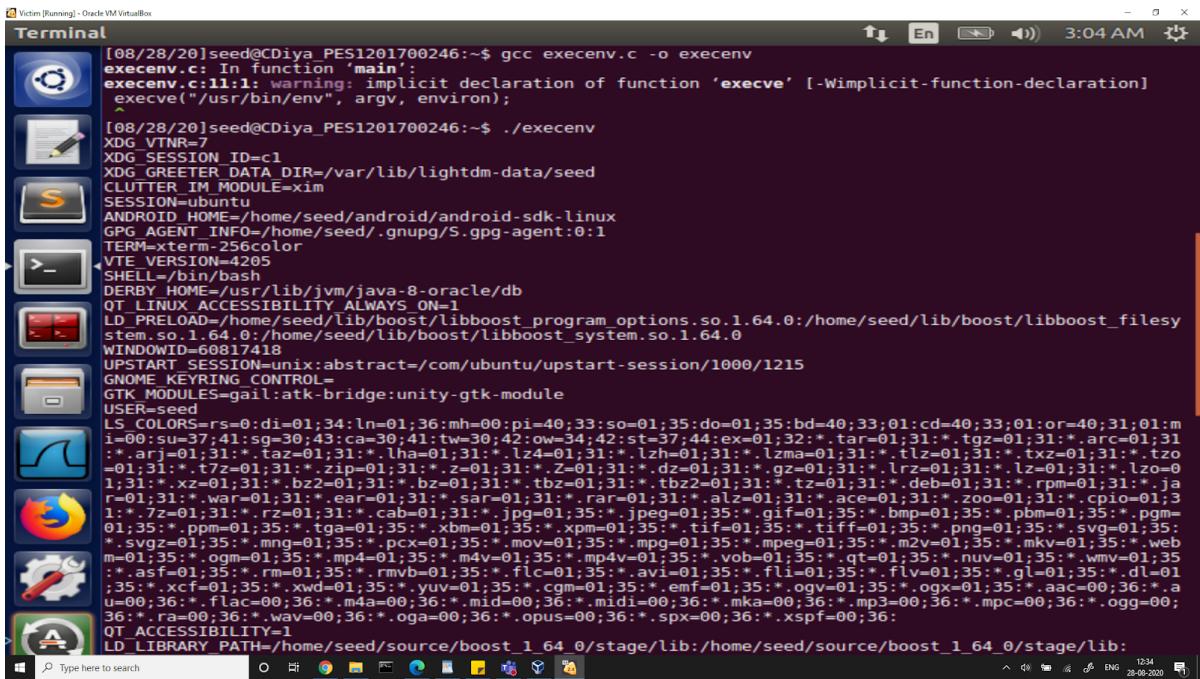
The diff command is used to compare the output from the parent and child. Thus, it can be observed that nothing gets printed since there are no changes between parent and child. This is because fork() causes child processes to inherit parent's environment variables and no difference can be observed.

TASK 3: Environment variables and execve()



```
[08/28/20]seed@CDiya_PES1201700246:~$ gcc execenv.c -o execenv
execenv.c: In function 'main':
execenv.c:11:1: warning: implicit declaration of function 'execve' [-Wimplicit-function-declaration]
execve("/usr/bin/env", argv, NULL);
[08/28/20]seed@CDiya_PES1201700246:~$ ./execenv
[08/28/20]seed@CDiya_PES1201700246:~$
```

Observation: The screenshot above compiles the program that uses the execve() command. The execve() runs the new program inside the calling process. Since NULL is passed as a parameter to the execve() command, nothing gets printed as no output is generated. The environment variables of the process are not visible.

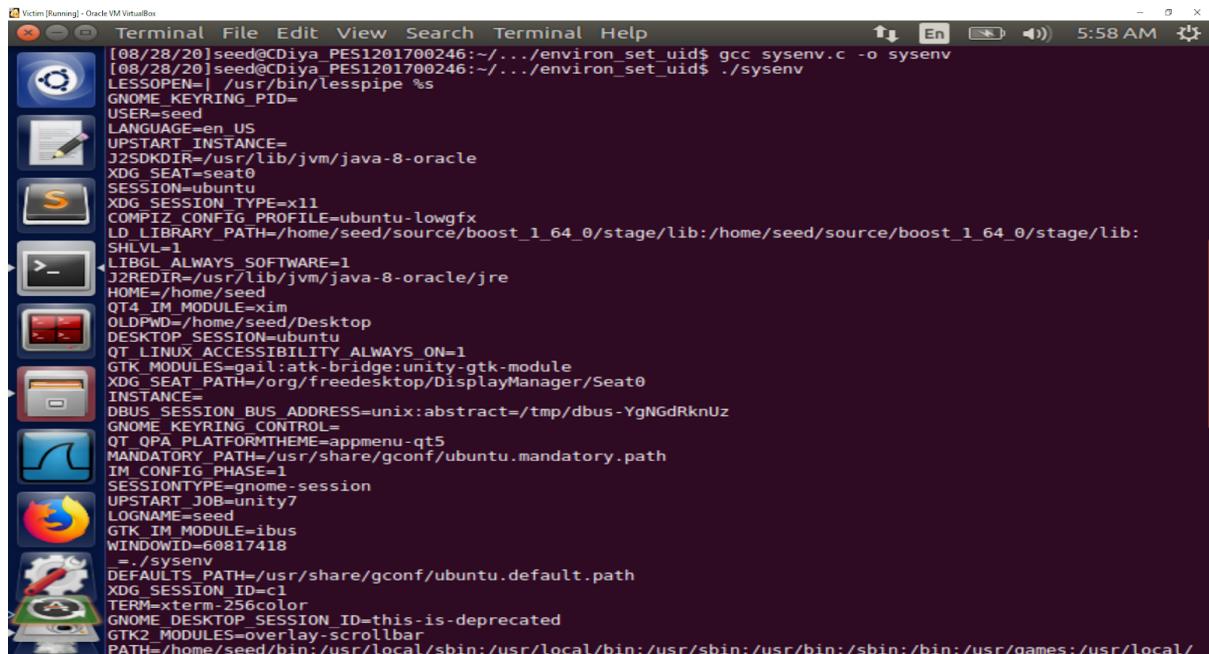


```
[08/28/20]seed@CDiya_PES1201700246:~$ gcc execenv.c -o execenv
execenv.c: In function 'main':
execenv.c:11:1: warning: implicit declaration of function 'execve' [-Wimplicit-function-declaration]
execve("/usr/bin/env", argv, environ);
[08/28/20]seed@CDiya_PES1201700246:~$ ./execenv
XDG_VTNR=7
XDG_SESSION_ID=c1
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed
CLUTTER_IM_MODULE=xim
SESSION=ubuntu
ANDROID_HOME=/home/seed/android/android-sdk-linux
GPG_AGENT_INFO=/home/seed/.gnupg/S.gpg-agent:0:1
TERM=xterm-256color
VTE_VERSION=4205
SHELL=/bin/bash
DERBY_HOME=/usr/lib/jvm/java-8-oracle/db
QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1
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/lib/boost/libboost_system.so.1.64.0
WINDOWID=60817418
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1215
GNOME_KEYRING_CONTROL=
GTK_MODULES=gail:atk-bridge:unity-gtk-module
USER=seed
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33:01:cd=40;33:01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.bz=01;31:*.bz2=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.ja=r=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogg=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:
```

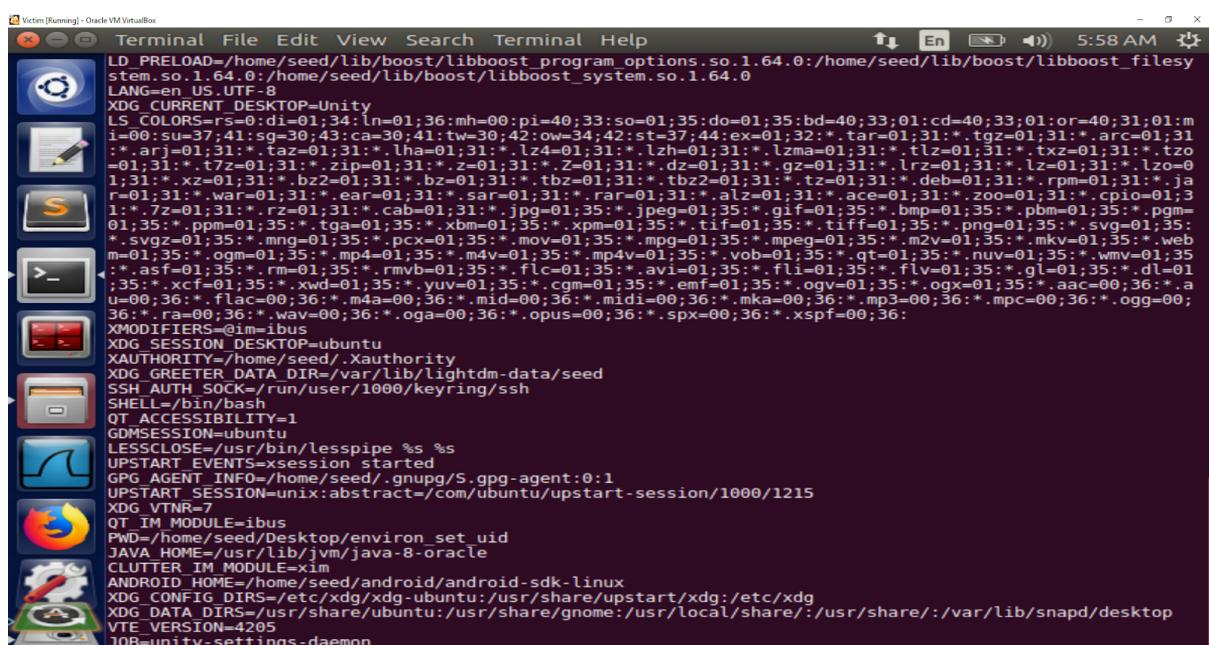
QT_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:

Observation: On changing the program above to pass environ as a parameter to the execve() command, the environment variables of the process are printed. Thus, execve() runs the new program inside the calling process and the environment variables of the program are generated as the output.

TASK 4: Environment variables and system()



```
[08/28/20]seed@Diya:~/Desktop$ gcc sysenv.c -o sysenv
[08/28/20]seed@Diya:~/Desktop$ ./sysenv
LESSOPEN=-| /usr/bin/lesspipe %
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
LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/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
OLDPWD=/home/seed/Desktop
DESKTOP_SESSION=ubuntu
QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1
GTK_MODULES=gail:atk-bridge:unity-gtk-module
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
INSTANCE=
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-YgNGdRknUz
GNOME_KEYRING_CONTROL=
QT_QPA_PLATFORMTHEME=appmenu-qt5
MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path
IM_CONFIG_PHASE=1
SESSIONTYPE=gnome-session
UPSTART_JOB=unity7
LOGNAME=seed
GTK_IM_MODULE=ibus
WINDOWID=60817418
/`/sysenv
DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path
XDG_SESSION_ID=c1
TERM=xterm-256color
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GTK2_MODULES=overlay-scrollbar
PATH=/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/
```



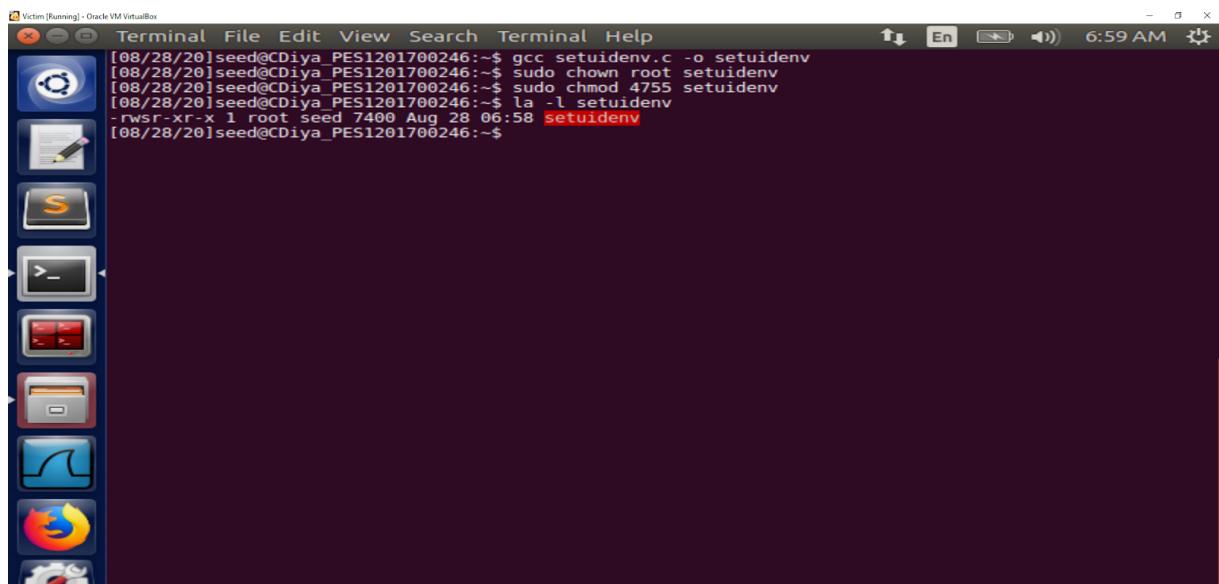
```
LD_PRELOAD=/home/seed/lib/boost/libboost_program_options.so.1.64.0:/home/seed/lib/boost/libboost_fileystem.so.1.64.0:/home/seed/lib/boost/libboost_system.so.1.64.0
LANG=en_US.UTF-8
XDG_CURRENT_DESKTOP=Unity
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33:01:cd=40;33:01:or=40;31:01:m
i=00:su=37:1sq=03:43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31
*:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo
=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.ja
r=01;31:*.war=01;31:*.ear=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;3
1:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=
01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35
*:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.web
m=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35
*:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.flv=01;35:*.gl=01;35:*.dl=01
*:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.flv=01;35:*.gl=01;35:*.dl=01
*:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogg=01;35:*.aac=00;36:*.a
u=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00
*:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:
XMODIFIERS=@im=ibus
XDG_SESSION_DESKTOP=ubuntu
XAUTHORITY=/home/seed/.Xauthority
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
SHELL=/bin/bash
GDMSESSION=ubuntu
LESSCLOSE=/usr/bin/lesspipe %s %
UPSTART_EVENTS=xsession started
GPG_AGENT_INFO=/home/seed/.gnupg/S.gpg-agent:0:1
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1215
XDG_VTNR=7
QT_IM_MODULE=ibus
PWD=/home/seed/Desktop/environ_set_uid
JAVA_HOME=/usr/lib/jvm/java-8-oracle
CLUTTER_IM_MODULE=xim
ANDROID_HOME=/home/seed/android-sdk-linux
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/usr/share/upstart/xdg:/etc/xdg
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/share/gnome:/usr/local/share/:/usr/share/:/var/lib/snapd/desktop
VTE_VERSION=4205
JOB=unity-settings-daemon
```

Observation:

The system() executes a child process that executes the shell command specified by calling /bin/sh -c command, and returns after the command has been completed. Thus, it can be seen from the screenshots above that the system() passes down the environment array to the new program which gets printed.

TASK 5: Environment variable and Set-UID Programs

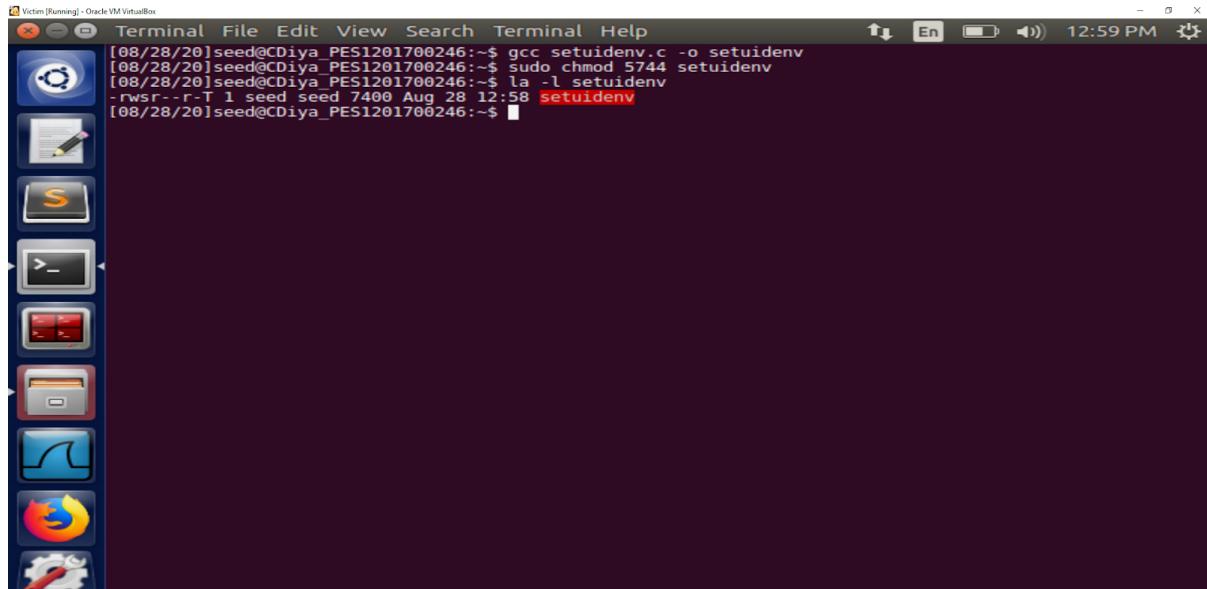
STEP 1:

A screenshot of a Linux desktop environment titled "Victim [Running] - Oracle VM VirtualBox". The desktop has a dark theme with a dock on the left containing icons for a terminal, file manager, and web browser. A terminal window is open, showing the following command-line session:

```
[08/28/20]seed@CDiya_PES1201700246:~$ gcc setuidenv.c -o setuidenv
[08/28/20]seed@CDiya_PES1201700246:~$ sudo chown root setuidenv
[08/28/20]seed@CDiya_PES1201700246:~$ sudo chmod 4755 setuidenv
[08/28/20]seed@CDiya_PES1201700246:~$ la -l setuidenv
-rwsr-xr-x 1 root seed 7400 Aug 28 06:58 setuidenv
[08/28/20]seed@CDiya_PES1201700246:~$
```

Observation: It can be inferred from the screenshot above that the setuid command is used to set the program owner to root. The following chmod 4755 command is used to sets read, write, and execute permissions for users, and sets read and execute permissions for Group, but not write permission.

STEP 2:

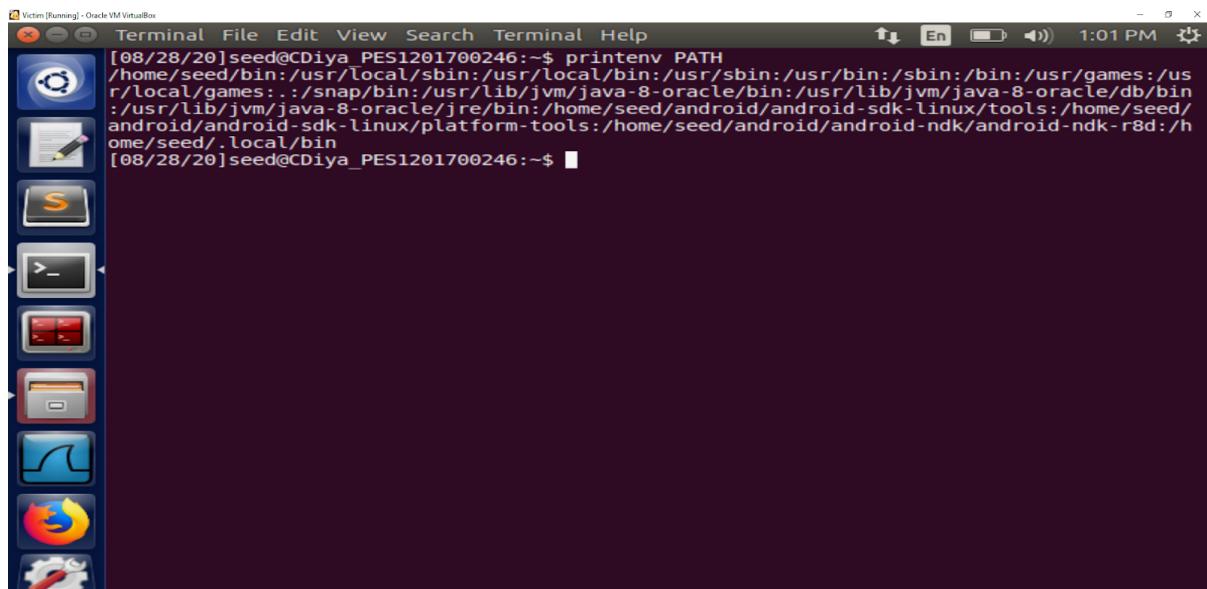
A screenshot of an Ubuntu desktop environment. A terminal window is open in the top-left corner. The terminal shows the following command-line session:

```
[08/28/20]seed@CDiya_PES1201700246:~$ gcc setuidenv.c -o setuidenv
[08/28/20]seed@CDiya_PES1201700246:~$ sudo chmod 5744 setuidenv
[08/28/20]seed@CDiya_PES1201700246:~$ la -l setuidenv
-rwsr--r-T 1 seed seed 7400 Aug 28 12:58 setuidenv
[08/28/20]seed@CDiya_PES1201700246:~$
```

The terminal window has a dark background with light-colored text. The desktop interface includes a dock with icons for various applications like Nautilus, Dash, and the Dash icon.

Observation: the chmod 5755 ensures that the owner of the program can read, write and execute but the group now has only read permissions.

STEP 3

A screenshot of an Ubuntu desktop environment. A terminal window is open in the top-left corner. The terminal shows the following command-line session:

```
[08/28/20]seed@CDiya_PES1201700246:~$ 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/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/android/android-ndk/android-ndk-r8d:/home/seed/.local/bin
[08/28/20]seed@CDiya_PES1201700246:~$
```

The terminal window has a dark background with light-colored text. The desktop interface includes a dock with icons for various applications like Nautilus, Dash, and the Dash icon.

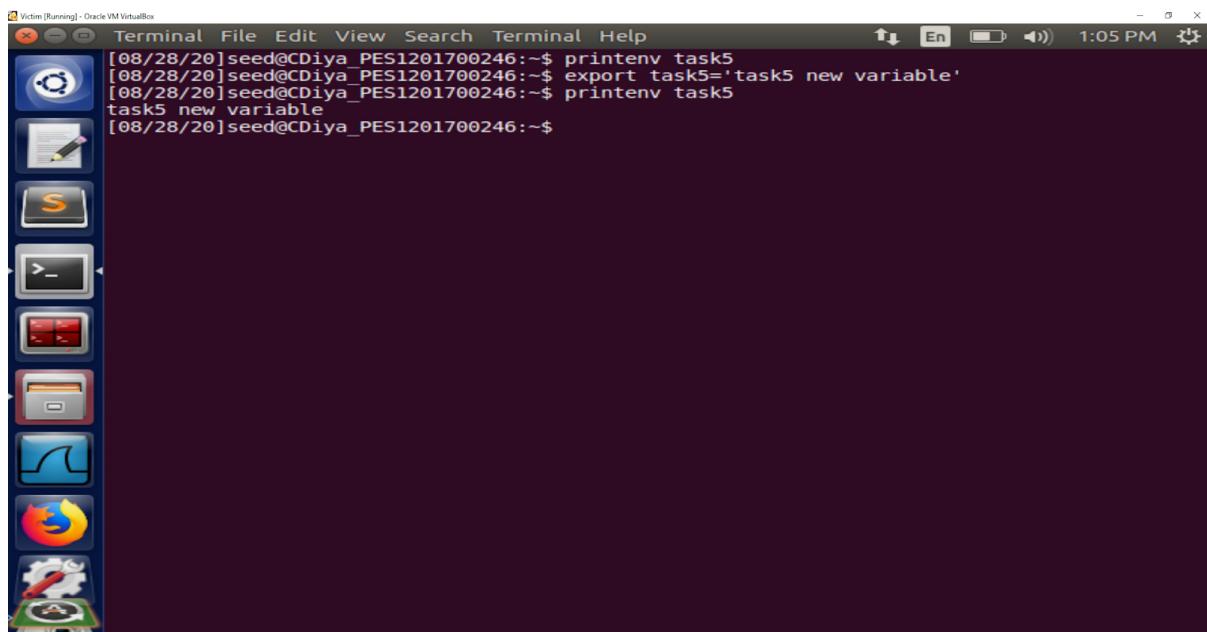
Observation: prints the value present in the PATH environment variable

```
[08/28/20]seed@CDiya_PES1201700246:~$ printenv LD_LIBRARY_PATH  
/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:  
[08/28/20]seed@CDiya_PES1201700246:~$
```

Observation: prints the value present in the LD_LIBRARY_PATH environment variable

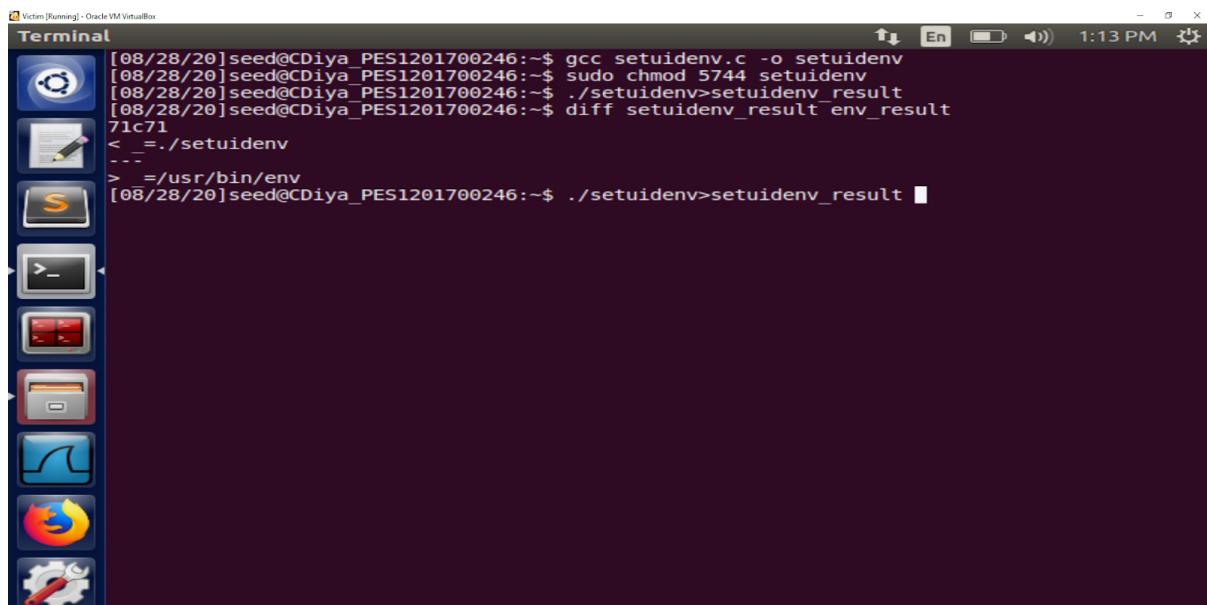
```
[08/28/20]seed@CDiya_PES1201700246:~$ printenv LD_LIBRARY_PATH  
/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:  
[08/28/20]seed@CDiya_PES1201700246:~$ export LD_LIBRARY_PATH=/home/seed:$LD_LIBRARY_PATH  
[08/28/20]seed@CDiya_PES1201700246:~$ printenv LD_LIBRARY_PATH  
/home/seed:/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/l  
ib:  
[08/28/20]seed@CDiya_PES1201700246:~$
```

Observation: The screenshot below shows the setting of the LD_LIBRARY_PATH environment variable using the export command.

A screenshot of an Ubuntu desktop environment. On the left is a dock with icons for Dash, Home, Applications, Places, System, and Dash to Dock. A terminal window is open in the center, showing the following command-line session:

```
[08/28/20]seed@CDiya_PES1201700246:~$ printenv task5  
[08/28/20]seed@CDiya_PES1201700246:~$ export task5='task5 new variable'  
[08/28/20]seed@CDiya_PES1201700246:~$ printenv task5  
task5 new variable  
[08/28/20]seed@CDiya_PES1201700246:~$
```

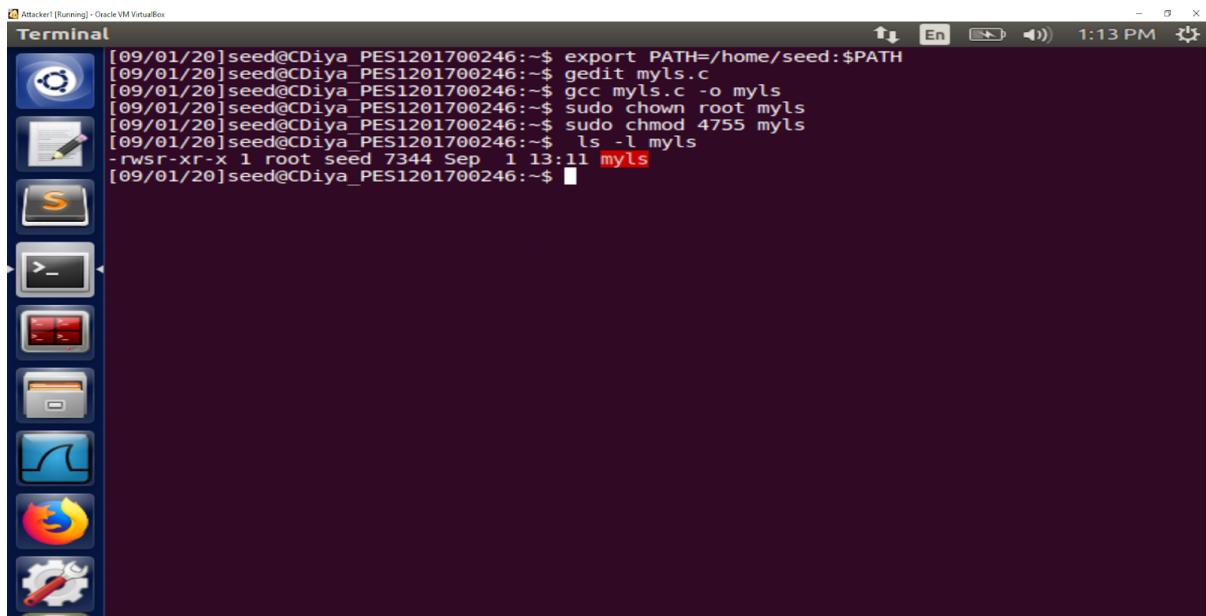
Observation: The screenshot below shows the setting of environment variables using the export command.

A screenshot of an Ubuntu desktop environment. On the left is a dock with icons for Dash, Home, Applications, Places, System, and Dash to Dock. A terminal window is open in the center, showing the following command-line session:

```
[08/28/20]seed@CDiya_PES1201700246:~$ gcc setuidenv.c -o setuidenv  
[08/28/20]seed@CDiya_PES1201700246:~$ sudo chmod 5744 setuidenv  
[08/28/20]seed@CDiya_PES1201700246:~$ ./setuidenv>setuidenv_result  
71c71  
< ./setuidenv  
---  
> =/usr/bin/env  
[08/28/20]seed@CDiya_PES1201700246:~$ ./setuidenv>setuidenv_result
```

Observation: When the Set-UID program from Step 2 is run in the shell, the shell forks a child process, and uses the child process to run the program. There is an addition of a new environment that can be seen above and thus, all the environment variables that were set in the shell process (parent) get into the Set-UID child process.

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

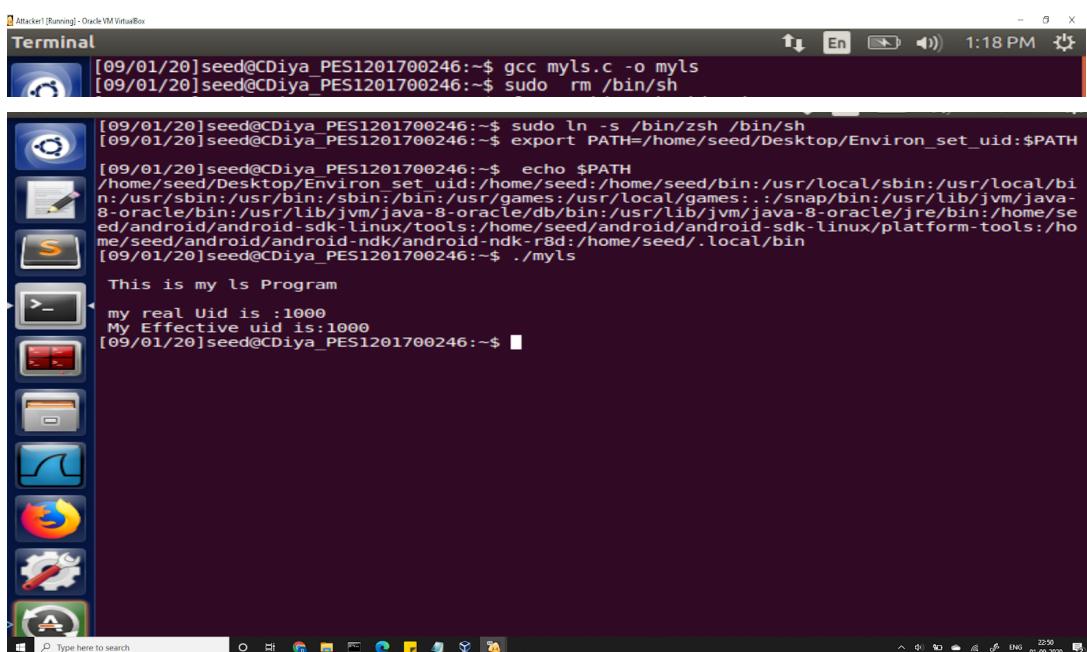


```
[09/01/20]seed@CDiya_PES1201700246:~$ export PATH=/home/seed:$PATH
[09/01/20]seed@CDiya_PES1201700246:~$ gedit myls.c
[09/01/20]seed@CDiya_PES1201700246:~$ gcc myls.c -o myls
[09/01/20]seed@CDiya_PES1201700246:~$ sudo chown root myls
[09/01/20]seed@CDiya_PES1201700246:~$ sudo chmod 4755 myls
[09/01/20]seed@CDiya_PES1201700246:~$ ls -l myls
-rwsr-xr-x 1 root seed 7344 Sep 1 13:11 myls
[09/01/20]seed@CDiya_PES1201700246:~$
```

Observation:

Can you let this Set-UID program run your code instead of /bin/ls? If you can, is your code running with the root privilege?

Answer: The screenshot above shows that the SET-UID program can be used in place of /bin/ls. It also shows that the program runs with root privileges which can be inferred that the root is the owner of the program.



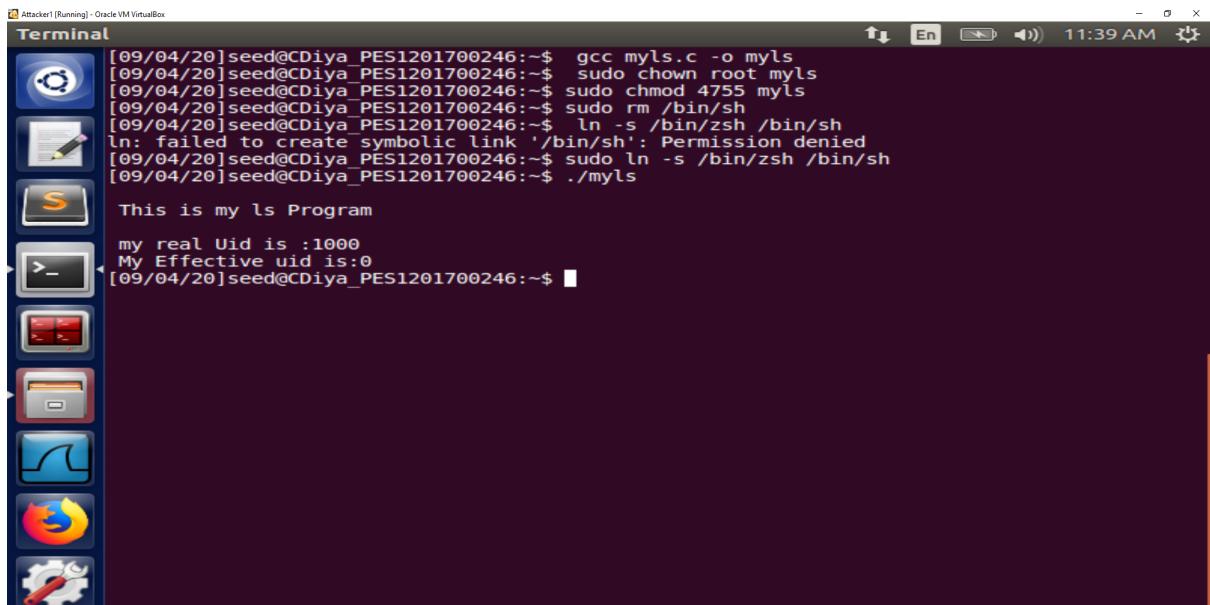
```
[09/01/20]seed@CDiya_PES1201700246:~$ gcc myls.c -o myls
[09/01/20]seed@CDiya_PES1201700246:~$ sudo rm /bin/sh
[09/01/20]seed@CDiya_PES1201700246:~$ sudo ln -s /bin/zsh /bin/sh
[09/01/20]seed@CDiya_PES1201700246:~$ export PATH=/home/seed/Desktop/Environ_set_uid:$PATH
[09/01/20]seed@CDiya_PES1201700246:~$ echo $PATH
/home/seed/Desktop/Environ_set_uid:/home/seed:/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:./snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm/java-8-oracle/jre/bin:/home/seed/android/android-sdk-linux/tools:/home/seed/android/android-sdk-linux/platform-tools:/home/seed/android/android-ndk/android-ndk-r8d:/home/seed/.local/bin
[09/01/20]seed@CDiya_PES1201700246:~$ ./myls
This is my ls Program
my real Uid is :1000
My Effective uid is:1000
[09/01/20]seed@CDiya_PES1201700246:~$
```

Observation:

Real UserID : It is an account of the owner of this process. It defines which files that this process has access to.

Effective UserID : It is normally the same as Real UserID, but sometimes it is changed to enable a non-privileged user to access files that can only be accessed by root. SET UID is used to set this.

When the program is run without changing any privileges, the real and effective ID is the same(ie 1000). Additionally, if root access was enabled: the following screenshot is obtained.



The screenshot shows a terminal window titled "Terminal" with the following command history:

```
[09/04/20]seed@CDiya_PES1201700246:~$ gcc myls.c -o myls
[09/04/20]seed@CDiya_PES1201700246:~$ sudo chown root myls
[09/04/20]seed@CDiya_PES1201700246:~$ sudo chmod 4755 myls
[09/04/20]seed@CDiya_PES1201700246:~$ sudo rm /bin/sh
[09/04/20]seed@CDiya_PES1201700246:~$ ln -s /bin/zsh /bin/sh
ln: failed to create symbolic link '/bin/sh': Permission denied
[09/04/20]seed@CDiya_PES1201700246:~$ sudo ln -s /bin/zsh /bin/sh
[09/04/20]seed@CDiya_PES1201700246:~$ ./myls
```

Below the terminal, the user types:

```
This is my ls Program
my real Uid is :1000
My Effective uid is:0
```

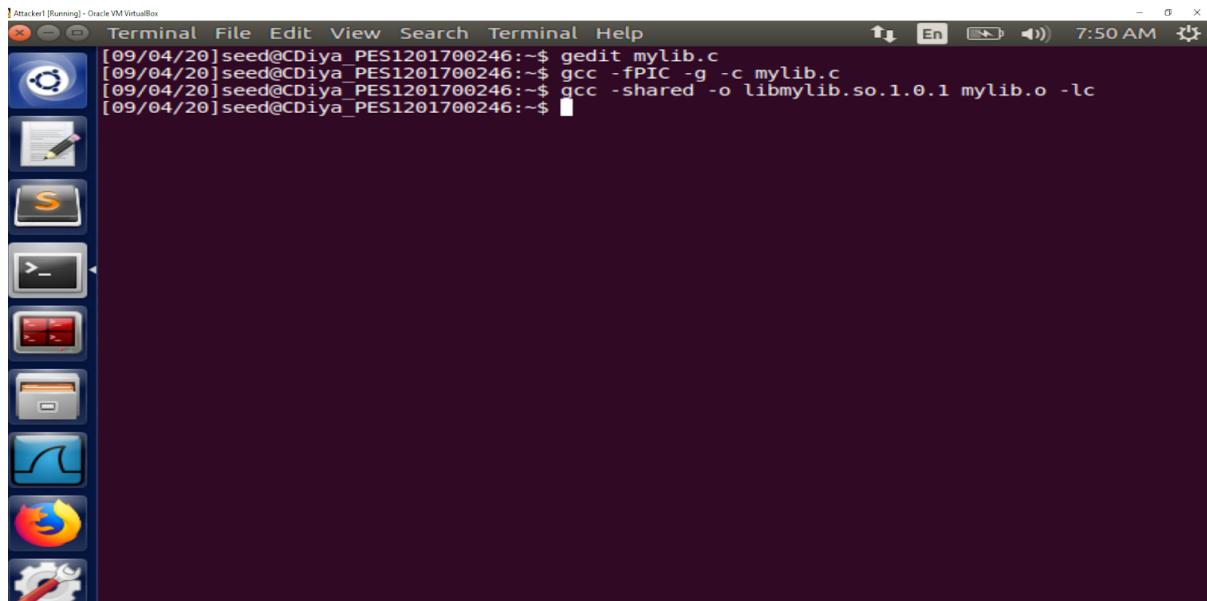
Observation:(real id=1000 and effective ID=0)

It can be observed above that if the owner is changed to root, the effective ID which was externally set becomes 0. 0 EFFECTIVE ID implies that the root is the owner and the program has root privileges

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

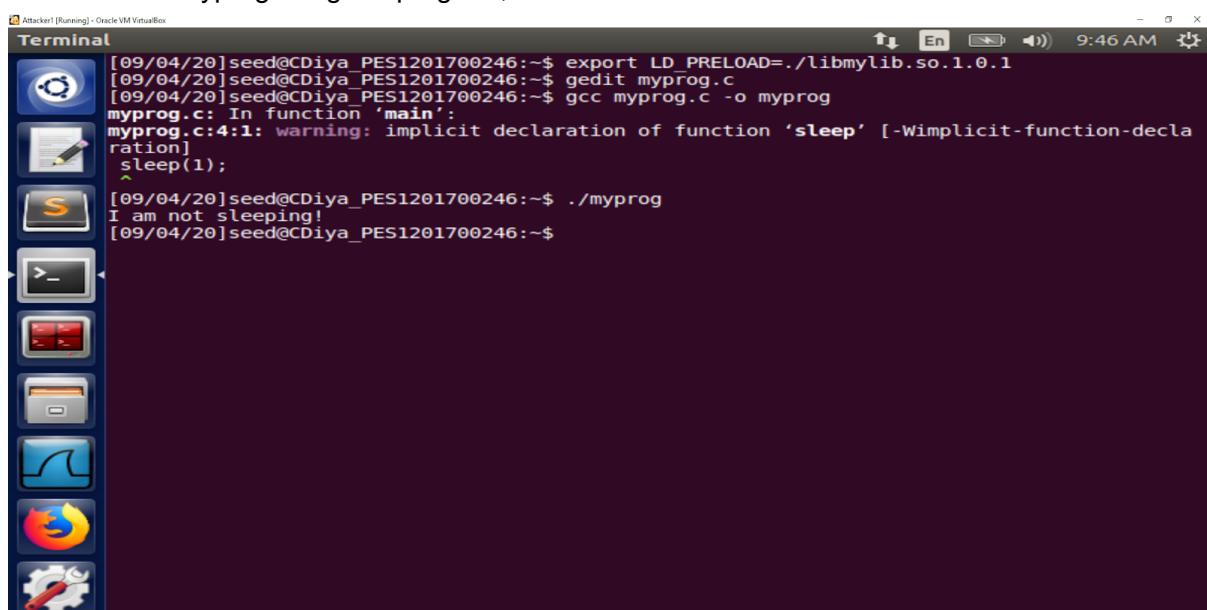
Step 1:

1. The program is compiled using the steps given.



```
[09/04/20]seed@CDiya_PES1201700246:~$ gedit mylib.c
[09/04/20]seed@CDiya_PES1201700246:~$ gcc -fPIC -g -c mylib.c
[09/04/20]seed@CDiya_PES1201700246:~$ gcc -shared -o libmylib.so.1.0.1 mylib.o -lc
[09/04/20]seed@CDiya_PES1201700246:~$
```

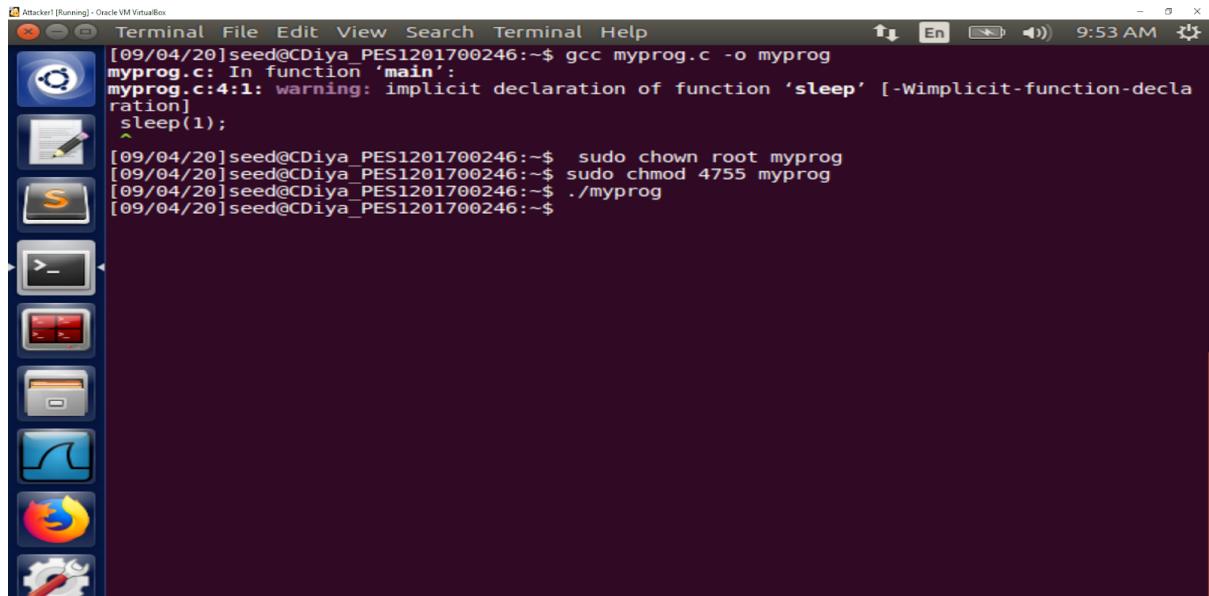
2. Make myprog a regular program, and run it as a normal user.



```
[09/04/20]seed@CDiya_PES1201700246:~$ export LD_PRELOAD=./libmylib.so.1.0.1
[09/04/20]seed@CDiya_PES1201700246:~$ gedit myprog.c
[09/04/20]seed@CDiya_PES1201700246:~$ gcc myprog.c -o myprog
myprog.c: In function 'main':
myprog.c:4:1: warning: implicit declaration of function 'sleep' [-Wimplicit-function-declaration]
  sleep(1);
  ^
[09/04/20]seed@CDiya_PES1201700246:~$ ./myprog
I am not sleeping!
[09/04/20]seed@CDiya_PES1201700246:~$
```

Observation: From the screenshot above, it can be inferred that when the given program is run, the output is printed “I am not sleeping”. This is because the root UID has not been set yet and it is run in a normal mode.

3. Make myprog a Set-UID root program, and run it as a normal user.



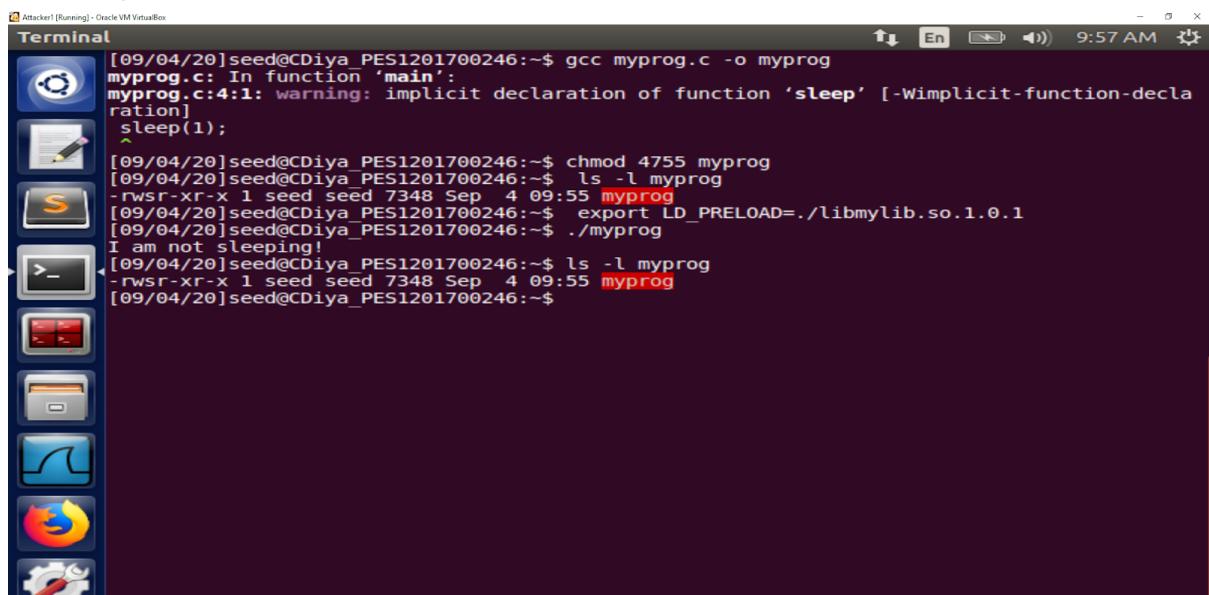
A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window titled "Terminal". The terminal shows the following command-line session:

```
[09/04/20]seed@CDiya_PES1201700246:~$ gcc myprog.c -o myprog
myprog.c: In function 'main':
myprog.c:4:1: warning: implicit declaration of function 'sleep' [-Wimplicit-function-declaration]
    sleep(1);
^
[09/04/20]seed@CDiya_PES1201700246:~$ sudo chown root myprog
[09/04/20]seed@CDiya_PES1201700246:~$ sudo chmod 4755 myprog
[09/04/20]seed@CDiya_PES1201700246:~$ ./myprog
[09/04/20]seed@CDiya_PES1201700246:~$
```

The desktop interface includes a dock with icons for various applications like Dash, Home, Applications, and the Dash search bar.

Observation: When the program runs after setting the UID root program it sleeps for 1 second.

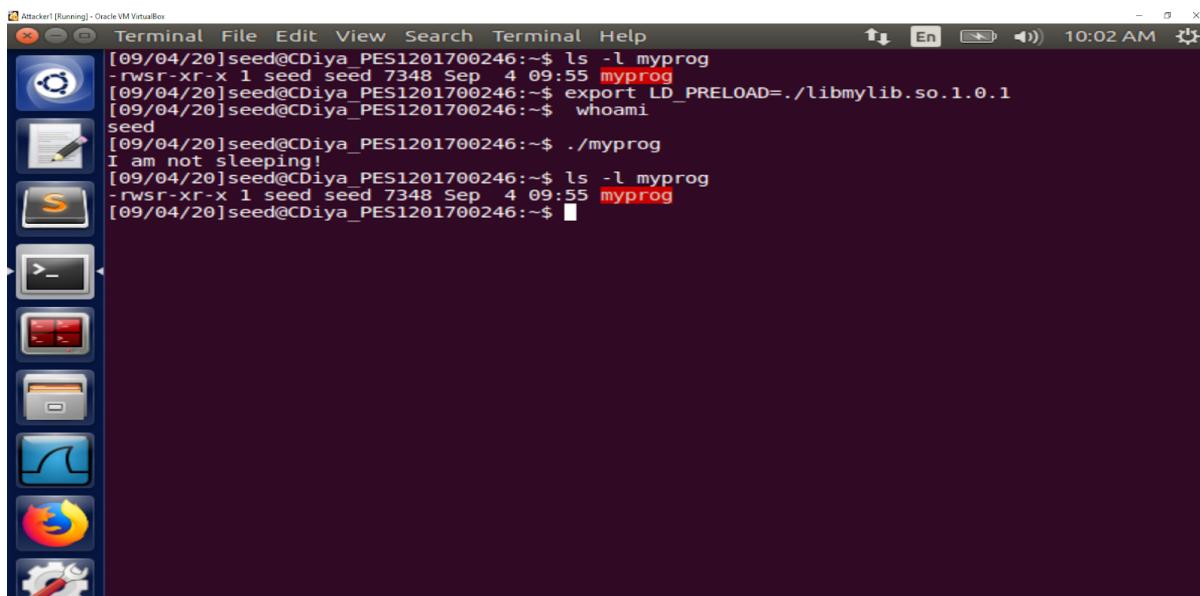
4. Make myprog a Set-UID root program, export the LD PRELOAD environment variable again in the root account and run it.



A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window titled "Terminal". The terminal shows the following command-line session:

```
[09/04/20]seed@CDiya_PES1201700246:~$ gcc myprog.c -o myprog
myprog.c: In function 'main':
myprog.c:4:1: warning: implicit declaration of function 'sleep' [-Wimplicit-function-declaration]
    sleep(1);
^
[09/04/20]seed@CDiya_PES1201700246:~$ chmod 4755 myprog
[09/04/20]seed@CDiya_PES1201700246:~$ ls -l myprog
-rwsr-xr-x 1 seed seed 7348 Sep 4 09:55 myprog
[09/04/20]seed@CDiya_PES1201700246:~$ export LD_PRELOAD=./libmylib.so.1.0.1
[09/04/20]seed@CDiya_PES1201700246:~$ ./myprog
I am not sleeping!
[09/04/20]seed@CDiya_PES1201700246:~$ ls -l myprog
-rwsr-xr-x 1 seed seed 7348 Sep 4 09:55 myprog
[09/04/20]seed@CDiya_PES1201700246:~$
```

The desktop interface includes a dock with icons for Dash, Home, Applications, and the Dash search bar.



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Attacker1 [Running] - Oracle VM VirtualBox". The terminal content shows the following commands and output:

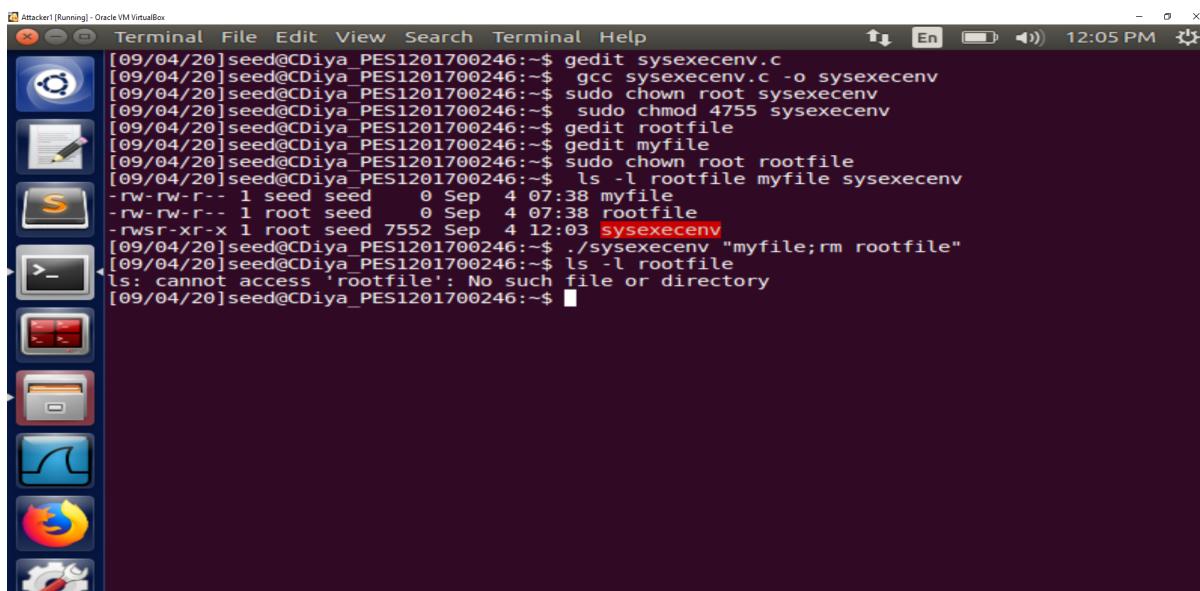
```
[09/04/20]seed@CDiya_PES1201700246:~$ ls -l myprog
-rwsr-xr-x 1 seed seed 7348 Sep 4 09:55 myprog
[09/04/20]seed@CDiya_PES1201700246:~$ export LD_PRELOAD=./libmylib.so.1.0.1
[09/04/20]seed@CDiya_PES1201700246:~$ whoami
seed
[09/04/20]seed@CDiya_PES1201700246:~$ ./myprog
I am not sleeping!
[09/04/20]seed@CDiya_PES1201700246:~$ ls -l myprog
-rwsr-xr-x 1 seed seed 7348 Sep 4 09:55 myprog
[09/04/20]seed@CDiya_PES1201700246:~$
```

Observation: From the above screenshots, when the seed is the user, the seed user would have all the execute permissions.

While using the root user it can be inferred that the owner and group is of the root. Thus, access of the root library is possible

TASK 8: Invoking External Programs Using `system()` versus `execve()`

STEP 1:

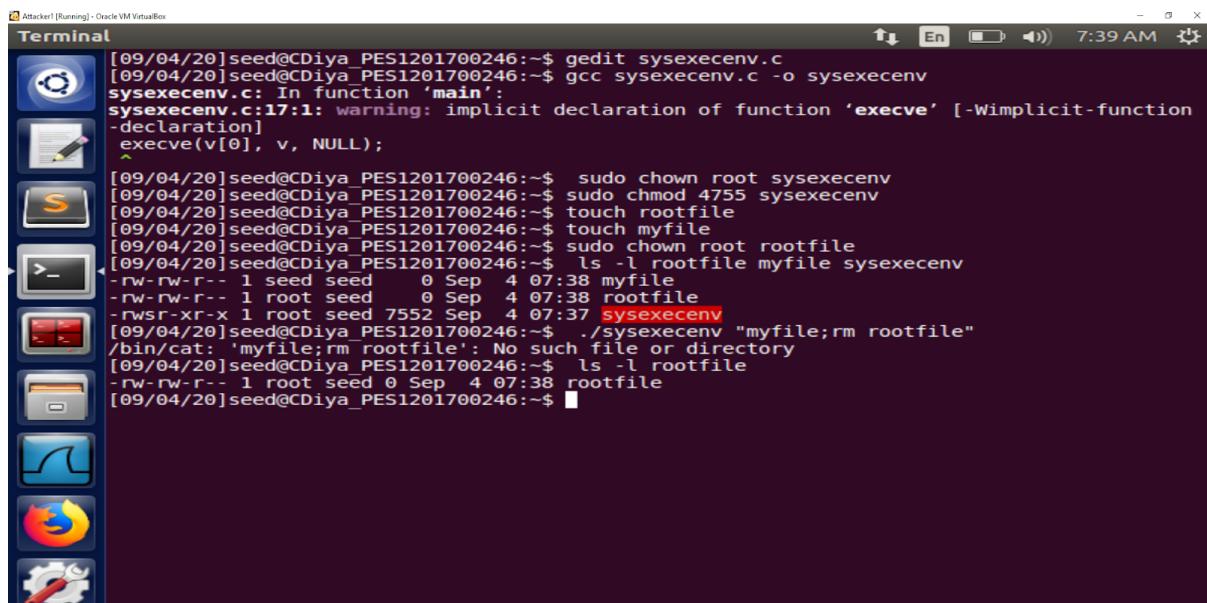


The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Attacker1 [Running] - Oracle VM VirtualBox". The terminal content shows the following commands and output:

```
[09/04/20]seed@CDiya_PES1201700246:~$ gedit sysexecenv.c
[09/04/20]seed@CDiya_PES1201700246:~$ gcc sysexecenv.c -o sysexecenv
[09/04/20]seed@CDiya_PES1201700246:~$ sudo chown root sysexecenv
[09/04/20]seed@CDiya_PES1201700246:~$ sudo chmod 4755 sysexecenv
[09/04/20]seed@CDiya_PES1201700246:~$ gedit rootfile
[09/04/20]seed@CDiya_PES1201700246:~$ sudo chown root rootfile
[09/04/20]seed@CDiya_PES1201700246:~$ ls -l rootfile myfile sysexecenv
-rw-rw-r-- 1 seed seed 0 Sep 4 07:38 myfile
-rw-rw-r-- 1 root seed 0 Sep 4 07:38 rootfile
-rwsr-xr-x 1 root seed 7552 Sep 4 12:03 sysexecenv
[09/04/20]seed@CDiya_PES1201700246:~$ ./sysexecenv "myfile;rm rootfile"
[09/04/20]seed@CDiya_PES1201700246:~$ ls -l rootfile
ls: cannot access 'rootfile': No such file or directory
[09/04/20]seed@CDiya_PES1201700246:~$
```

Observation: Two files were created, two files “myfile”(owner is seed) and “rootfile”(owner is root). Bob can compromise the integrity of the system by deleting the rootfile using the program given. The program executable is used to enter the rm command and the rootfile(root owner) which is passed to the system() command. Thus, despite the rootfile having root ownership, Bob can delete the file using the system(). Thus, system() is not safe because the PATH environment variable affects the behavior of system(), because the variable affects how the shell works.

STEP 2:



The screenshot shows a terminal window titled "Terminal" with the following session log:

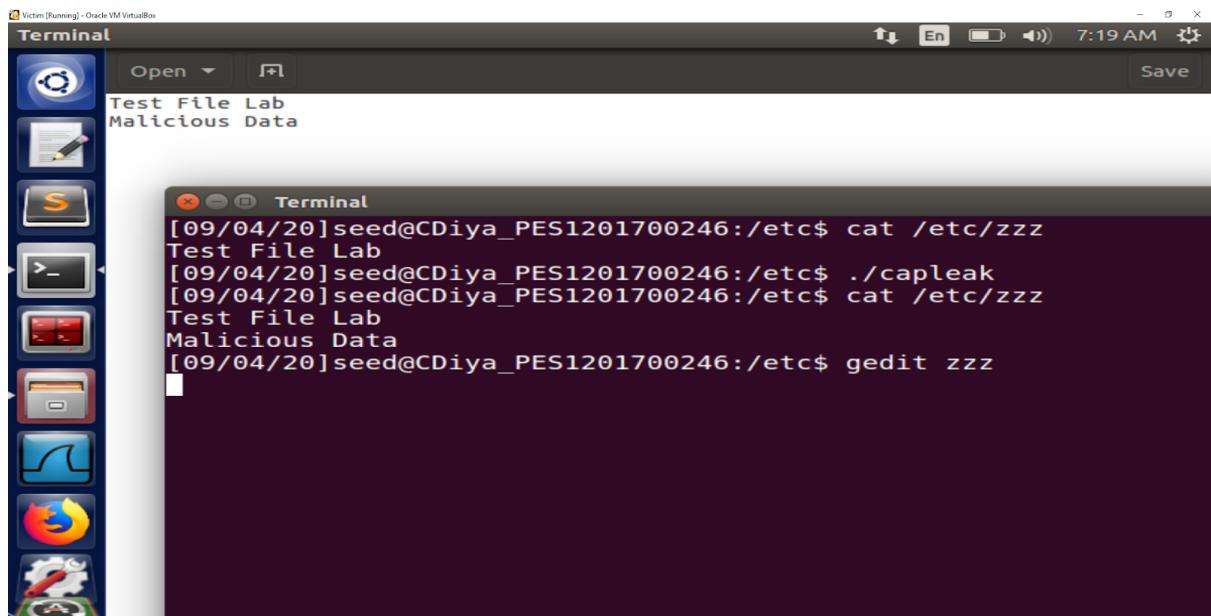
```
[09/04/20]seed@CDiya_PES1201700246:~$ gedit sysexecenv.c
[09/04/20]seed@CDiya_PES1201700246:~$ gcc sysexecenv.c -o sysexecenv
sysexecenv.c: In function 'main':
sysexecenv.c:17:1: warning: implicit declaration of function 'execve' [-Wimplicit-function-declaration]
    execve(v[0], v, NULL);
[09/04/20]seed@CDiya_PES1201700246:~$ sudo chown root sysexecenv
[09/04/20]seed@CDiya_PES1201700246:~$ sudo chmod 4755 sysexecenv
[09/04/20]seed@CDiya_PES1201700246:~$ touch rootfile
[09/04/20]seed@CDiya_PES1201700246:~$ touch myfile
[09/04/20]seed@CDiya_PES1201700246:~$ sudo chown root rootfile
[09/04/20]seed@CDiya_PES1201700246:~$ ls -l rootfile myfile sysexecenv
-rw-rw-r-- 1 seed seed 0 Sep 4 07:38 myfile
-rw-rw-r-- 1 root seed 0 Sep 4 07:38 rootfile
-rwsr-xr-x 1 root seed 7552 Sep 4 07:37 sysexecenv
[09/04/20]seed@CDiya_PES1201700246:~$ ./sysexecenv "myfile;rm rootfile"
/bin/cat: 'myfile;rm rootfile': No such file or directory
[09/04/20]seed@CDiya_PES1201700246:~$ ls -l rootfile
-rw-rw-r-- 1 root seed 0 Sep 4 07:38 rootfile
[09/04/20]seed@CDiya_PES1201700246:~$
```

Observation: system() is replaced by the execve(). When the program above is compiled and the executable is used to remove the rootfile, it can be seen that the rootfile does not get deleted. This is because execve() does not invoke the shell because PATH environment variable does not affect the behavior of execve(). Thus, it does not access environment variables due to which the file does not get removed and the integrity is not compromised.

TASK 9: Capability Leaking



The initial content of the file can be seen above:



Observation: “Malicious Data” has been added to the /etc/zzz file. Due to root privileges that were retained, read , write and execute privileges allowed the file to be written into.

If the root privileges are no longer needed, it's time to relinquish the root privileges permanently. When revoking this privilege, one of the common mistakes is capability leaking. The process may have gained some

privileged capabilities when it was still privileged; when the privilege is downgraded, if the program does not clean up those capabilities, they may still be accessible by the non-privileged process.