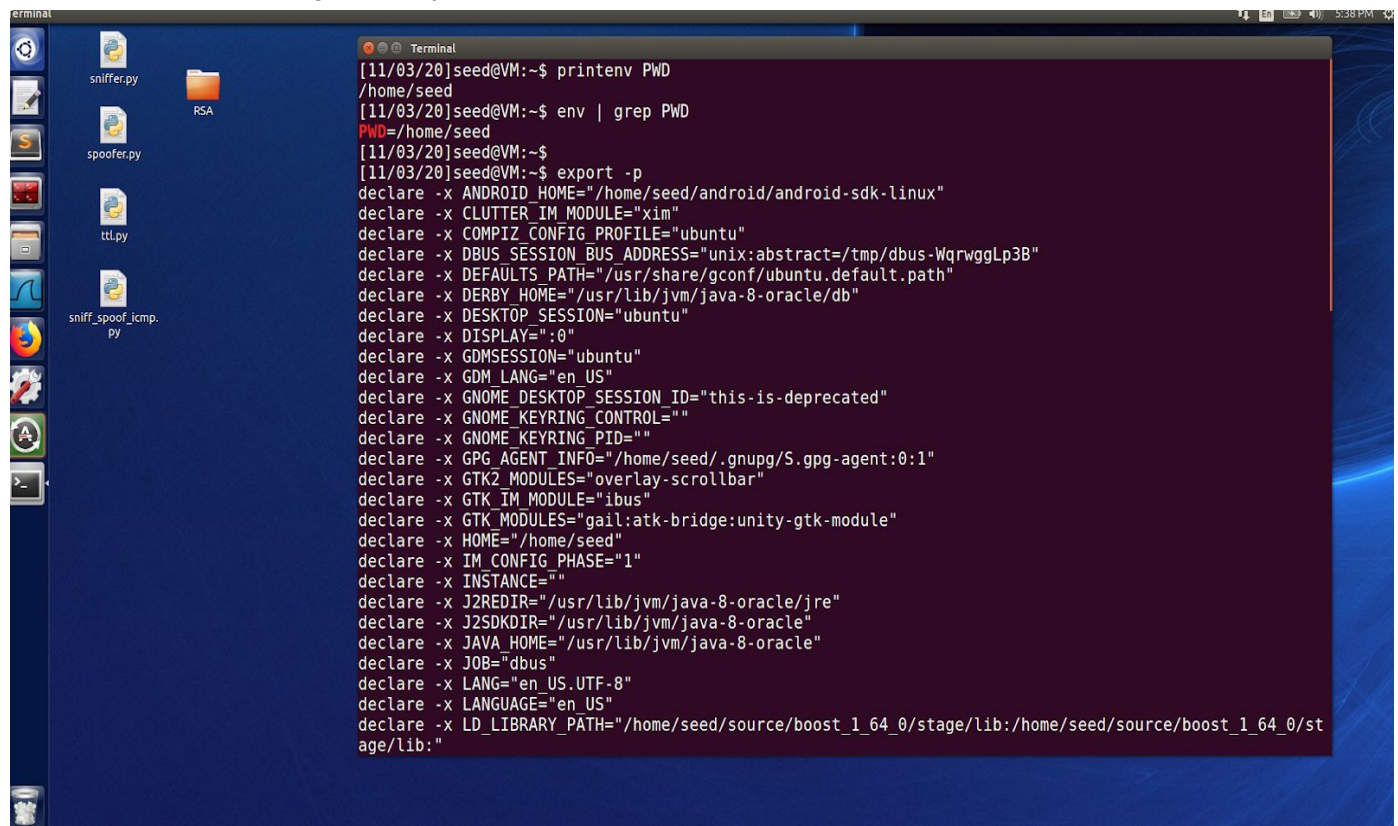


Task 1: Manipulating Environment Variables

Printenv and env are commands that allow us to print out environment variables and export or unset allows us to change/ modify them here is a screenshot to show them used.



```
[11/03/20]seed@VM:~$ printenv PWD
/home/seed
[11/03/20]seed@VM:~$ env | grep PWD
PWD=/home/seed
[11/03/20]seed@VM:~$ export -p
declare -x ANDROID_HOME="/home/seed/android/android-sdk-linux"
declare -x CLUTTER_IM_MODULE="xim"
declare -x COMPIZ_CONFIG_PROFILE="ubuntu"
declare -x DBUS_SESSION_BUS_ADDRESS="unix:abstract=/tmp/dbus-WqrwggLp3B"
declare -x DEFAULTS_PATH="/usr/share/gconf/ubuntu.default.path"
declare -x DERBY_HOME="/usr/lib/jvm/java-8-oracle/db"
declare -x DESKTOP_SESSION="ubuntu"
declare -x DISPLAY=":0"
declare -x GDMSESSION="ubuntu"
declare -x GDM_LANG="en_US"
declare -x GNOME_DESKTOP_SESSION_ID="this-is-deprecated"
declare -x GNOME_KEYRING_CONTROL=""
declare -x GNOME_KEYRING_PID=""
declare -x GPG_AGENT_INFO="/home/seed/.gnupg/S.gpg-agent:0:1"
declare -x GTK2_MODULES="overlay-scrollbar"
declare -x GTK_IM_MODULE="ibus"
declare -x GTK_MODULES="gail:atk-bridge:unity-gtk-module"
declare -x HOME="/home/seed"
declare -x IM_CONFIG_PHASE="1"
declare -x INSTANCE=""
declare -x J2REDIR="/usr/lib/jvm/java-8-oracle/jre"
declare -x J2SDKDIR="/usr/lib/jvm/java-8-oracle"
declare -x JAVA_HOME="/usr/lib/jvm/java-8-oracle"
declare -x JOB="dbus"
declare -x LANG="en_US.UTF-8"
declare -x LANGUAGE="en_US"
declare -x LD_LIBRARY_PATH="/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:"
```

For example I can do export NEW = 1 will set a new variable and give it a value of 1.

Task 2: Passing Environment Variables from Parent Process to Child Process

In this process we will see how a child process gets its environment variables from its parent. fork() creates a new process by duplicating the calling process and is then called a child and the original is called the parent. You can see the screenshot on the next page.

```

Terminal
PWD=/home/seed/Desktop/SETLAB
JOB=dbus
XMODIFIERS=@im=ibus
JAVA_HOME=/usr/lib/jvm/java-8-oracle
GNOME_KEYRING_PID=
LANG=en_US.UTF-8
GDM_LANG=en_US
MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path
COMPIZ_CONFIG_PROFILE=ubuntu
IM_CONFIG_PHASE=1
GDMSESSION=ubuntu
SESSIONTYPE=gnome-session
GTK2_MODULES=overlay-scrollbar
SHLVL=1
HOME=/home/seed
XDG_SEAT=seat0
LANGUAGE=en_US
LIBGL_ALWAYS_SOFTWARE=1
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
XDG_SESSION_DESKTOP=ubuntu
LOGNAME=seed
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-WqrrggLp3B
J2SDKDIR=/usr/lib/jvm/java-8-oracle
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/share/gnome:/usr/local/share:/usr/share:/var/lib/snapd/desktop
QT4_IM_MODULE=xim
LESSOPEN=| /usr/bin/lesspipe %s
INSTANCE=
XDG_RUNTIME_DIR=/run/user/1000
DISPLAY=:0
XDG_CURRENT_DESKTOP=Unity
GTK_IM_MODULE=ibus
J2REDIR=/usr/lib/jvm/java-8-oracle/jre
LESSCLOSE=/usr/bin/lesspipe %s %s
XAUTHORITY=/home/seed/.Xauthority
OLDPWD=/home/seed/Desktop
= ./first
[11/03/20]seed@VM:~/.../SETLAB$

Terminal
PWD=/home/seed/Desktop/SETLAB
JOB=dbus
XMODIFIERS=@im=ibus
JAVA_HOME=/usr/lib/jvm/java-8-oracle
GNOME_KEYRING_PID=
LANG=en_US.UTF-8
GDM_LANG=en_US
MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path
COMPIZ_CONFIG_PROFILE=ubuntu
IM_CONFIG_PHASE=1
GDMSESSION=ubuntu
SESSIONTYPE=gnome-session
GTK2_MODULES=overlay-scrollbar
SHLVL=1
HOME=/home/seed
XDG_SEAT=seat0
LANGUAGE=en_US
LIBGL_ALWAYS_SOFTWARE=1
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
XDG_SESSION_DESKTOP=ubuntu
LOGNAME=seed
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-WqrrggLp3B
J2SDKDIR=/usr/lib/jvm/java-8-oracle
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/share/gnome:/usr/local/share:/usr/share:/var/lib/snapd/desktop
QT4_IM_MODULE=xim
LESSOPEN=| /usr/bin/lesspipe %s
INSTANCE=
XDG_RUNTIME_DIR=/run/user/1000
DISPLAY=:0
XDG_CURRENT_DESKTOP=Unity
GTK_IM_MODULE=ibus
J2REDIR=/usr/lib/jvm/java-8-oracle/jre
LESSCLOSE=/usr/bin/lesspipe %s %s
XAUTHORITY=/home/seed/.Xauthority
OLDPWD=/home/seed/Desktop
= ./second
[11/03/20]seed@VM:~/.../SETLAB$

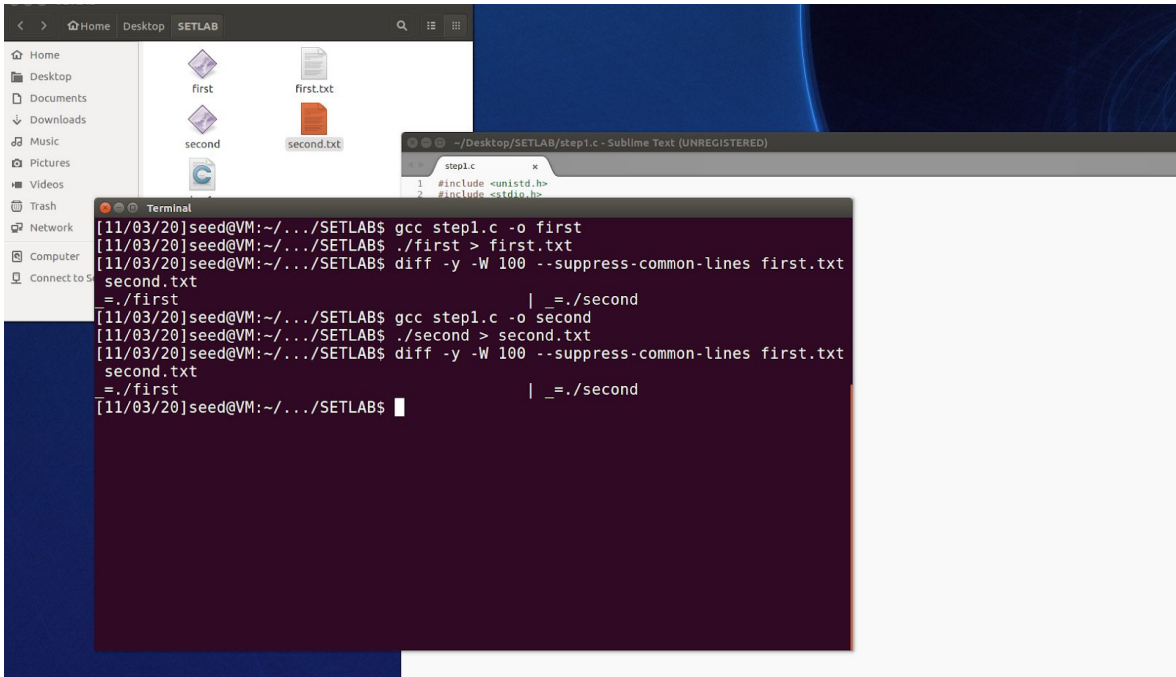
```

We changed the code and in the left it printed with the print env being enabled in child but in the right it was not and only in the parent. The printouts look identical and it is hard to find the differences so we save the files and use the diff command to help us find where the two outputs are different. We ran this command `diff -y -W 100 --suppress-common-lines first.txt second.txt` to tell us where the differences are

|: shows if line is changed in second file

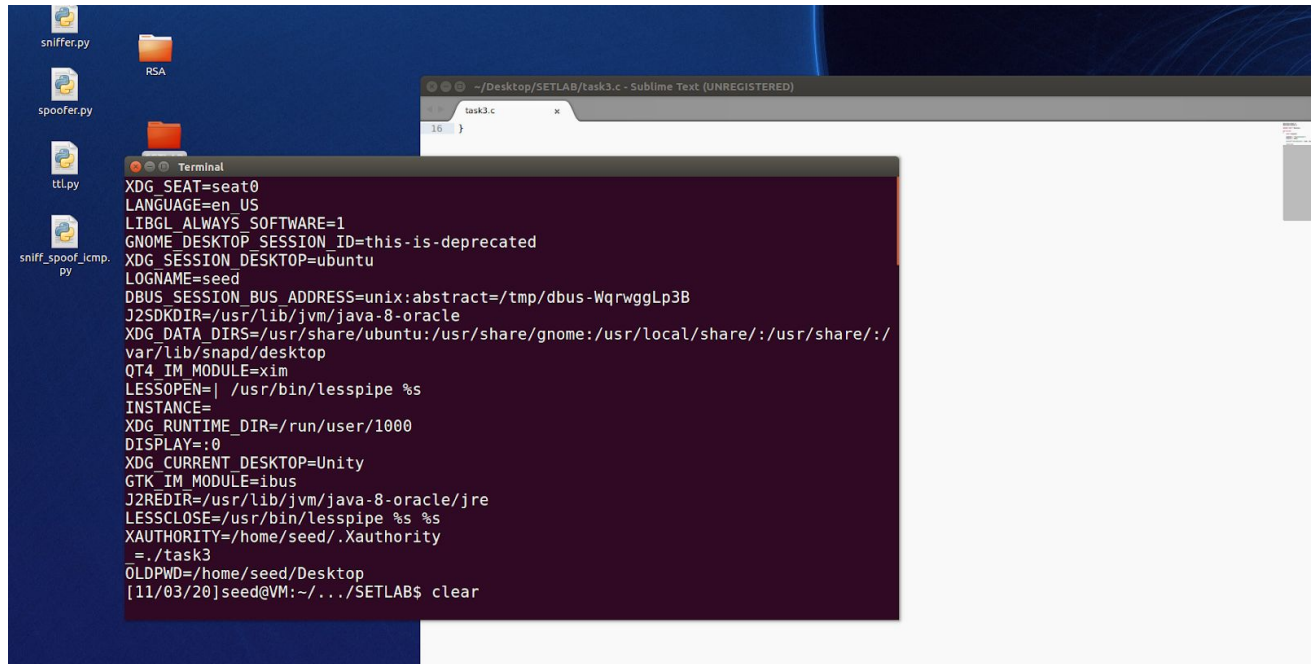
<: shows line was deleted in second file

>: shows line was added to second file and not in first If you look at the picture you will notice that the second file just has one line that is changed according to the diff command.



Task 3: Environment Variables and execve()

The program is supposed to print out current environmental variables of the current process. When the argument was null it was empty but when we changed the argument to environ then we saw something very similar to what happened with the previous tasks in what was printed.

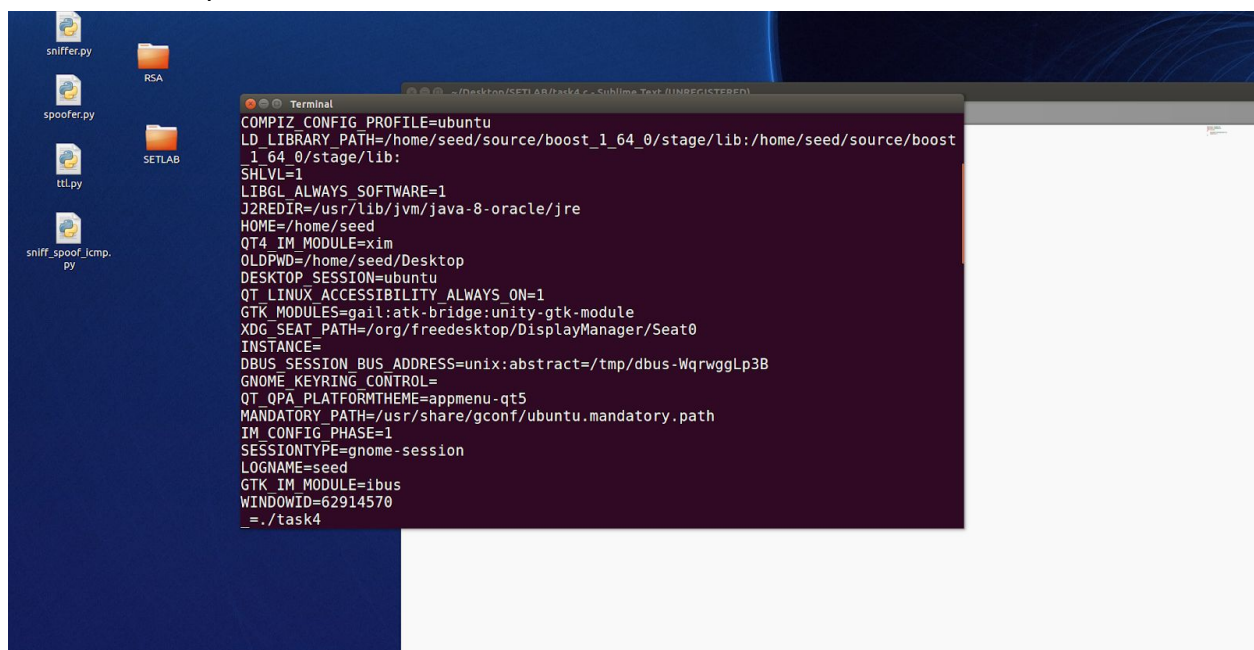


The screenshot shows a Linux desktop with a dark blue background. On the left, there are several icons: 'sniffer.py', 'spoofer.py', 'ttl.py', and 'sniff_spoof_icmp.py'. In the center, a terminal window is open, displaying a list of environment variables. To the right, a Sublime Text editor window is open, showing a file named 'task3.c' with a single line of code: '16 }'.

```
XDG_SEAT=seat0
LANGUAGE=en_US
LIBGL_ALWAYS_SOFTWARE=1
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
XDG_SESSION_DESKTOP=ubuntu
LOGNAME=seed
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-WqrggLP3B
J2SDKDIR=/usr/lib/jvm/java-8-oracle
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/share/gnome:/usr/local/share:/usr/share:/var/lib/snapd/desktop
QT4_IM_MODULE=xim
LESSOPEN=| /usr/bin/lesspipe %s
INSTANCE=
XDG_RUNTIME_DIR=/run/user/1000
DISPLAY=:0
XDG_CURRENT_DESKTOP=Unity
GTK_IM_MODULE=ibus
J2REDIR=/usr/lib/jvm/java-8-oracle/jre
LESSCLOSE=/usr/bin/lesspipe %s %s
XAUTHORITY=/home/seed/.Xauthority
./task3
OLDPWD=/home/seed/Desktop
[11/03/20]seed@VM:~/.../SETLAB$ clear
```

Task 4: Environment Variables and system()

In this task we will use system() instead of the previous command and see what happens. The same thing happened because system actually uses execv for some of the implementation therefore we expect similar results.

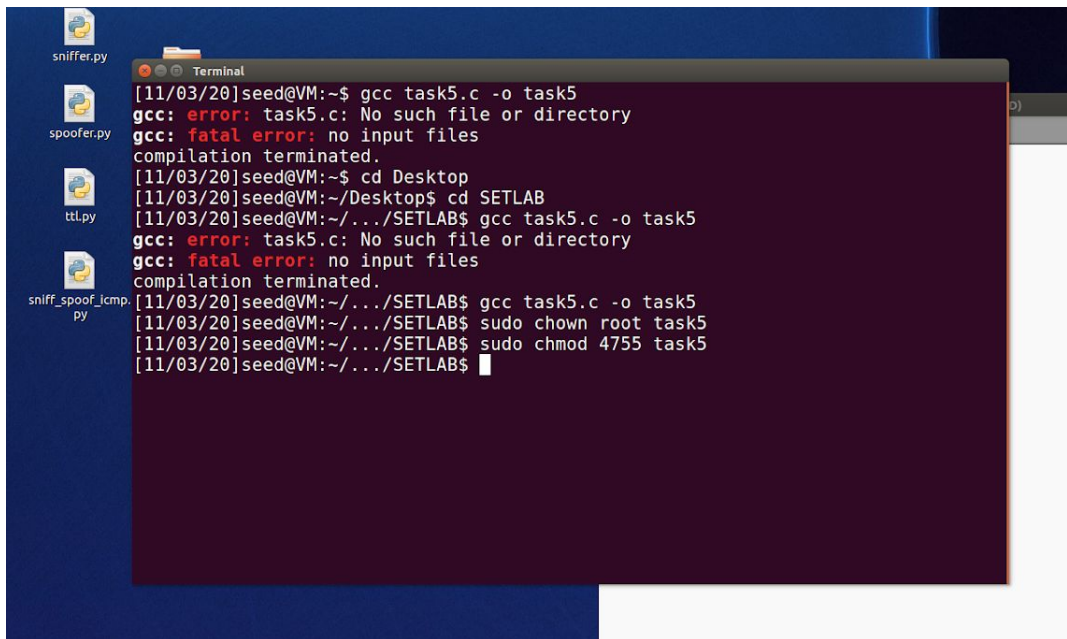


The screenshot shows a Linux desktop with a dark blue background. On the left, there are several icons: 'sniffer.py', 'spoofer.py', 'ttl.py', and 'sniff_spoof_icmp.py'. In the center, a terminal window is open, displaying a list of environment variables. To the right, a Sublime Text editor window is open, showing a file named 'task4.c' with a single line of code: '16 }'.

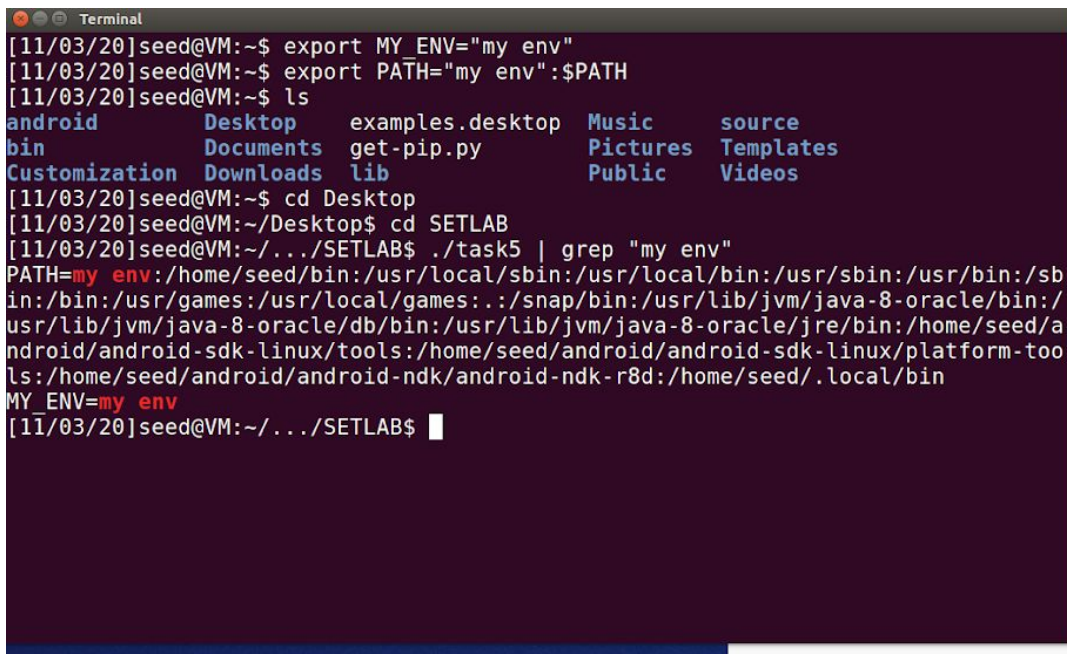
```
COMPIZ_CONFIG_PROFILE=ubuntu
LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:
SHLVL=1
LIBGL_ALWAYS_SOFTWARE=1
J2REDIR=/usr/lib/jvm/java-8-oracle/jre
HOME=/home/seed
QT4_IM_MODULE=xim
OLDPWD=/home/seed/Desktop
DESKTOP_SESSION=ubuntu
QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1
GTK_MODULES=gail:atk-bridge:unity-gtk-module
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
INSTANCE=
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-WqrggLP3B
GNOME_KEYRING_CONTROL=
QT_QPA_PLATFORMTHEME=appmenu-qt5
MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path
IM_CONFIG_PHASE=1
SESSIONTYPE=gnome-session
LOGNAME=seed
GTK_IM_MODULE=ibus
WINDOWID=62914570
./task4
```


Task 5: Environment Variable and Set-UID Programs

We are given a program to compile and change its ownership to root and then make it a SETUID program. And we did that in the below screenshot



```
[11/03/20]seed@VM:~$ gcc task5.c -o task5
gcc: error: task5.c: No such file or directory
gcc: fatal error: no input files
compilation terminated.
[11/03/20]seed@VM:~$ cd Desktop
[11/03/20]seed@VM:~/Desktop$ cd SETLAB
[11/03/20]seed@VM:~/.../SETLAB$ gcc task5.c -o task5
gcc: error: task5.c: No such file or directory
gcc: fatal error: no input files
compilation terminated.
[11/03/20]seed@VM:~/.../SETLAB$ gcc task5.c -o task5
[11/03/20]seed@VM:~/.../SETLAB$ sudo chown root task5
[11/03/20]seed@VM:~/.../SETLAB$ sudo chmod 4755 task5
[11/03/20]seed@VM:~/.../SETLAB$
```



```
[11/03/20]seed@VM:~$ export MY_ENV="my env"
[11/03/20]seed@VM:~$ export PATH="my env":$PATH
[11/03/20]seed@VM:~$ ls
android      Desktop      examples.desktop  Music        source
bin          Documents    get-pip.py        Pictures      Templates
Customization Downloads    lib               Public        Videos
[11/03/20]seed@VM:~$ cd Desktop
[11/03/20]seed@VM:~/Desktop$ cd SETLAB
[11/03/20]seed@VM:~/.../SETLAB$ ./task5 | grep "my env"
PATH=my env:/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
MY_ENV=my env
[11/03/20]seed@VM:~/.../SETLAB$
```

We made an environment and then run the program and the shell forks a process and uses the child to run it and the environment variables from the parent go to the child process which is running the program.

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

The shell program invoked is calling system() within the setuid program. The behavior of the shell program can be modified by the environment variables like PATH. You can change the variables and control the behavior of the program and can make it run with root privilege.

```
root@VM: /home/seed/Desktop/SETLAB
[11/03/20]seed@VM:~$ export PATH=/home/seed:$PATH
[11/03/20]seed@VM:~$ cd Desktop
[11/03/20]seed@VM:~/Desktop$ cd SETLAB
[11/03/20]seed@VM:~/.../SETLAB$ gcc task6.c -o task6
task6.c: In function 'main':
task6.c:3:5: warning: implicit declaration of function 'system' [-Wimplicit-func
tion-declaration]
    system("ls");
    ^
[11/03/20]seed@VM:~/.../SETLAB$ su
Password:
root@VM:/home/seed/Desktop/SETLAB# chmod u+s task6
root@VM:/home/seed/Desktop/SETLAB# exit
exit
[11/03/20]seed@VM:~/.../SETLAB$ ./task6
first      second      step1.c  task3.c  task4.c  task5.c  task6.c
first.txt  second.txt task3    task4    task5    task6
[11/03/20]seed@VM:~/.../SETLAB$
```

Task 7: The LD PRELOAD Environment Variable and Set-UID Programs

We will make a program that overrides the sleep() function in lib, then we compile and set the environment variable. Then we compile another program in the same environment variable. The same program will basically run but with different behaviors.

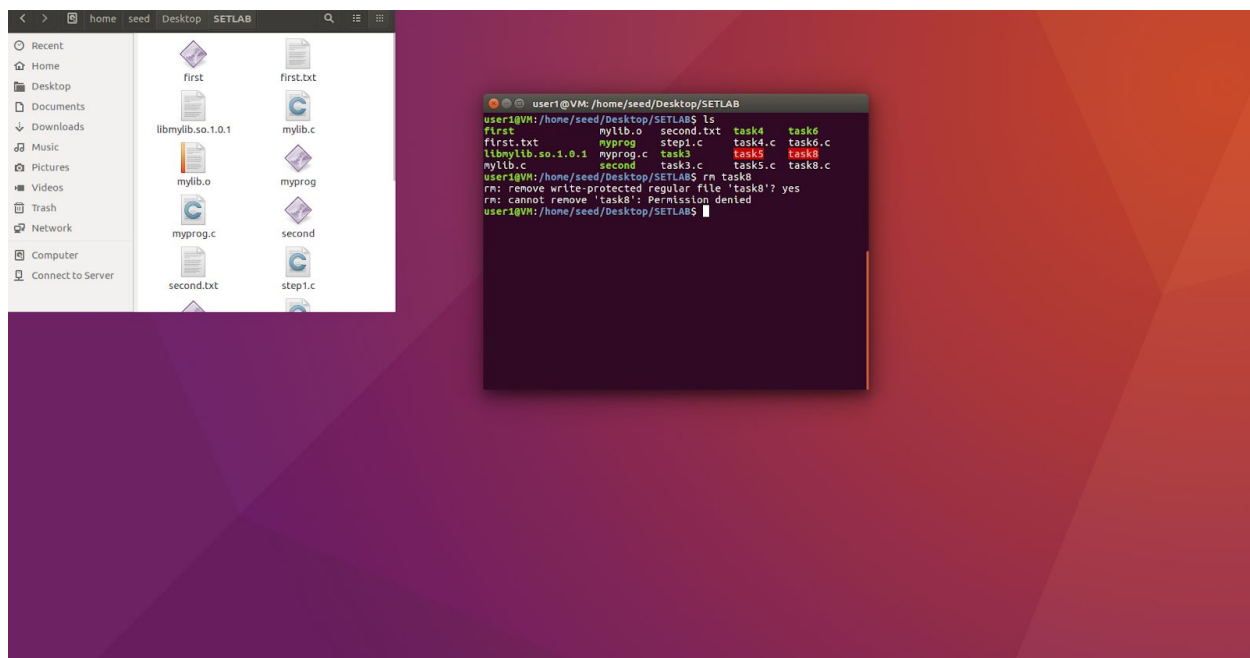
The program does not always give the same results, it is because the environment variable is missing in the time it does not give the same result.

```
terminal
root@VM: /home/seed/Desktop/SETLAB
[11/04/20]seed@VM:~$ cd Desktop
[11/04/20]seed@VM:~/Desktop$ cd SETLAB
[11/04/20]seed@VM:~/.../SETLAB$ gcc -fPIC -g -c mylib.c
[11/04/20]seed@VM:~/.../SETLAB$ gcc -shared -o libmylib.so.1.0.1 mylib.o -lc
[11/04/20]seed@VM:~/.../SETLAB$ export LD_PRELOAD=./libmylib.so.1.0.1
[11/04/20]seed@VM:~/.../SETLAB$ export PATH=/home/seed:$PATH
[11/04/20]seed@VM:~/.../SETLAB$ gcc -o myprog myprog.c
myprog.c: In function 'main':
myprog.c:4:5: warning: implicit declaration of function 'sleep' [-Wimplicit-function-declaration]
    sleep(1);
    ^
[11/04/20]seed@VM:~/.../SETLAB$ ./myprog
I am not sleeping!
[11/04/20]seed@VM:~/.../SETLAB$ sudo chmod u+s myprog
[11/04/20]seed@VM:~/.../SETLAB$ ./myprog
I am not sleeping!
[11/04/20]seed@VM:~/.../SETLAB$ sudo su
root@VM:/home/seed/Desktop/SETLAB# export LD_PRELOAD=./libmylib.so.1.0.1
root@VM:/home/seed/Desktop/SETLAB# ./myprog
root@VM:/home/seed/Desktop/SETLAB# exit
exit
[11/04/20]seed@VM:~/.../SETLAB$ ./myprog
I am not sleeping!
[11/04/20]seed@VM:~/.../SETLAB$ sudo su
root@VM:/home/seed/Desktop/SETLAB# useradd -d /usr/user1 -m user1
root@VM:/home/seed/Desktop/SETLAB# chown user1 myprog
root@VM:/home/seed/Desktop/SETLAB# chgrp user1 myprog
root@VM:/home/seed/Desktop/SETLAB# exit
exit
[11/04/20]seed@VM:~/.../SETLAB$ export LD_PRELOAD=./libmylib.so.1.0.1
[11/04/20]seed@VM:~/.../SETLAB$ ./myprog
I am not sleeping!
[11/04/20]seed@VM:~/.../SETLAB$
```

Task 8: Invoking External Programs Using system() versus execve()

We make a program and make it owned by root and make it a setuid program, can we remove a file not writable to us? We logged into user1 that was created from the above task and used terminal to locate to the task8 program and were not able to delete it from user1.

```
Terminal
first.txt      mylib.o  second  task3  task4.c  task6
libmylib.so.1.0.1 myprog  second.txt task3.c task5    task6.c
[11/04/20]seed@VM:~/.../SETLAB$ clear
[11/04/20]seed@VM:~/.../SETLAB$ gcc task8.c -o task8
[11/04/20]seed@VM:~/.../SETLAB$ sudo chown root task8
[11/04/20]seed@VM:~/.../SETLAB$ sudo chmod u+s task8
[11/04/20]seed@VM:~/.../SETLAB$ ./task8
Please type a file name.
[11/04/20]seed@VM:~/.../SETLAB$ ls
first      mylib.c  myprog.c  step1.c  task4    task5.c  task8
first.txt  mylib.o  second    task3.c  task4.c  task6    task8.c
libmylib.so.1.0.1 myprog  second.txt task3.c  task5    task6.c
[11/04/20]seed@VM:~/.../SETLAB$ sudo passwd user1
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
[11/04/20]seed@VM:~/.../SETLAB$
```



Task 9: Capability Leaking

In this task we compile the given program and give it root privileges and make it a setuid program and run it and see if the file zzz was modified. And the file was modified because the file was open before being set to uid and therefore was modified.

```
root@VM: /home/seed/Desktop/SETLAB
[11/04/20]seed@VM:~/.../SETLAB$ ls -al zzz task9
-rwsr-xr-x 1 root root 7640 Nov  4 13:38 task9
-rw-rw-r-- 1 seed seed    0 Nov  4 13:45 zzz
[11/04/20]seed@VM:~/.../SETLAB$
```