Information Security

Lab – 1

Name: Atmik Ajoy

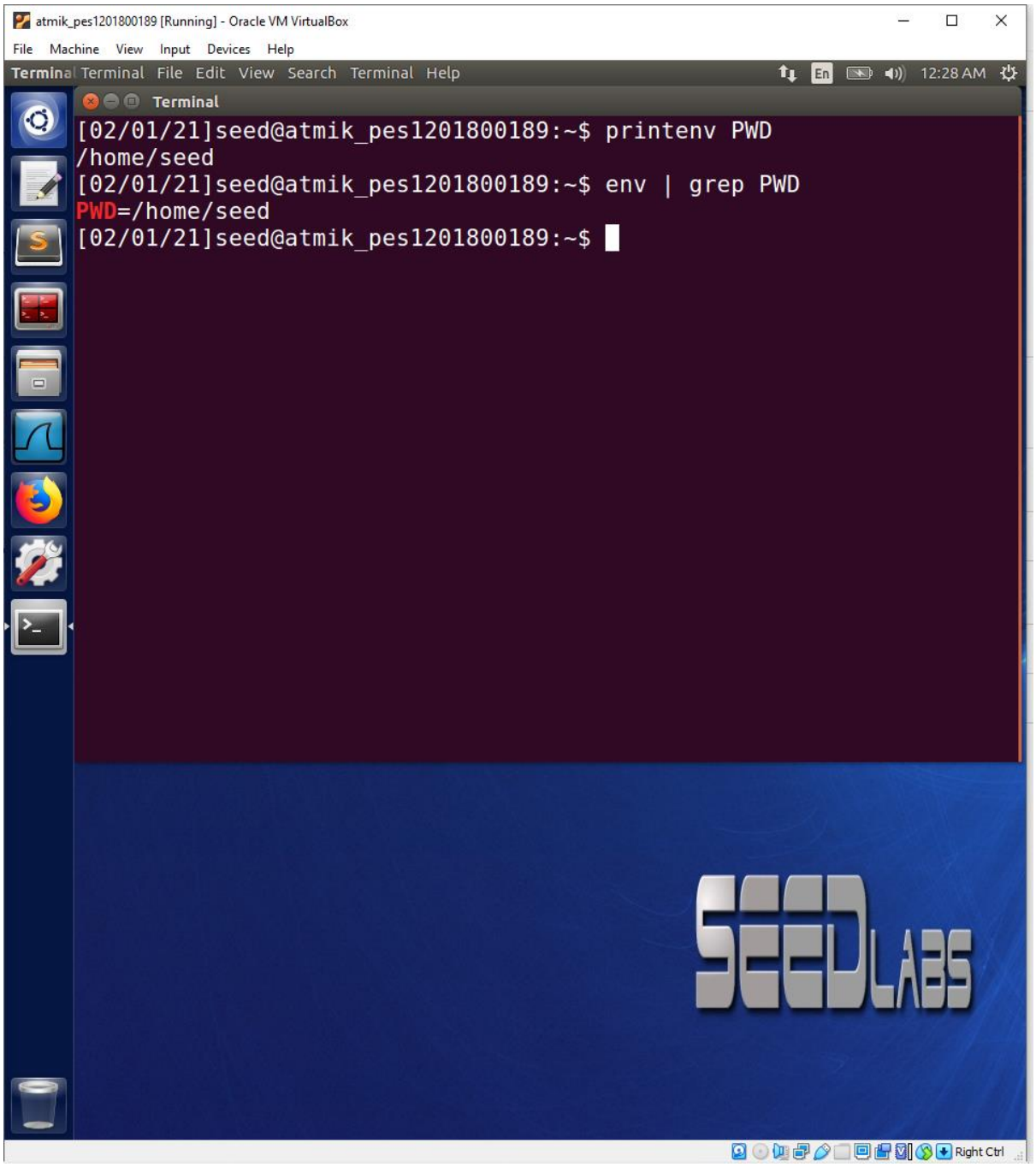
SRN: PES1201800189

Section – A

Task1

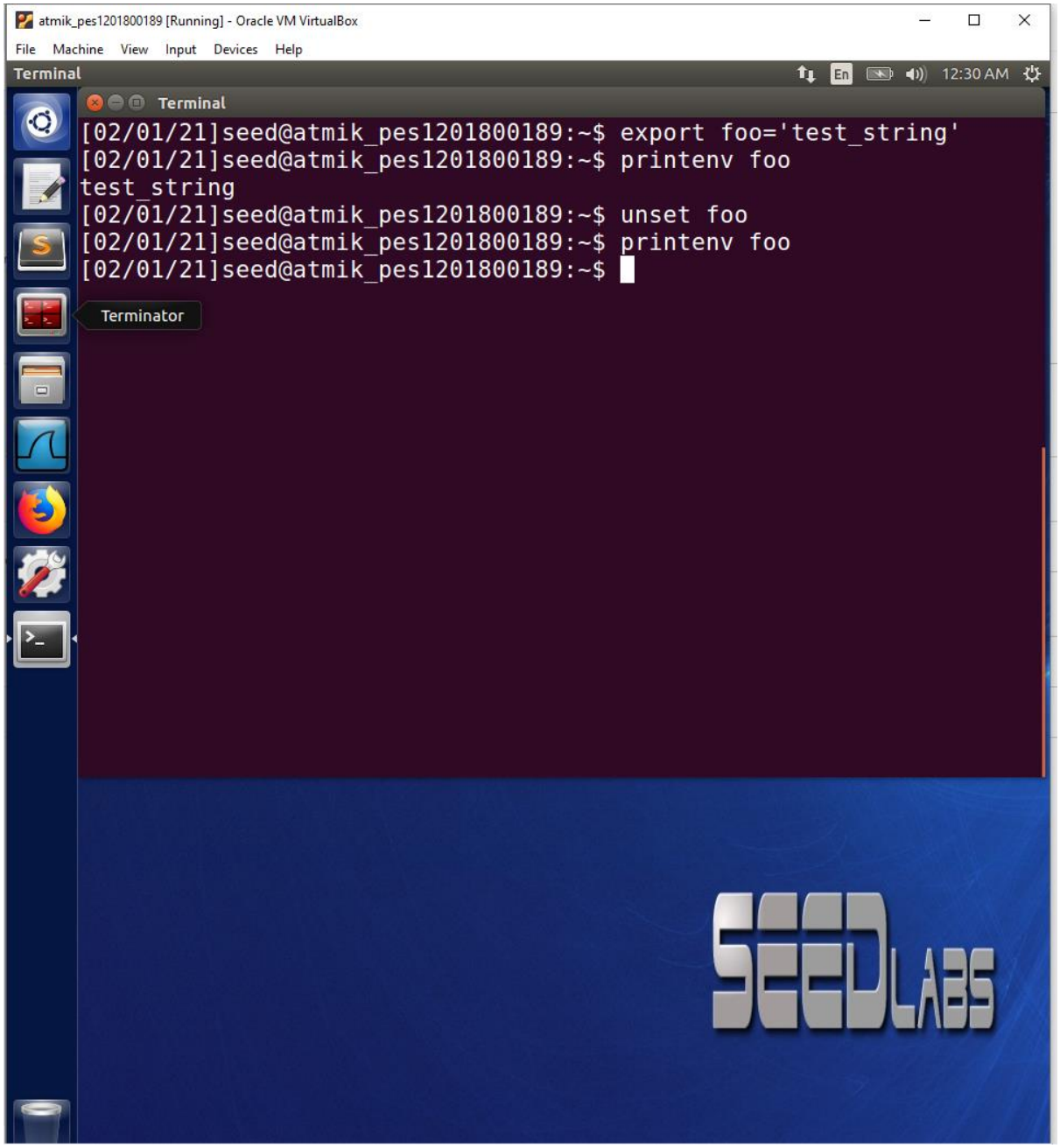
Step1

- printenv *variable_name* prints the value of the environment variable
- printenv PWD prints the value of the current working directory which in out case is /home/seed



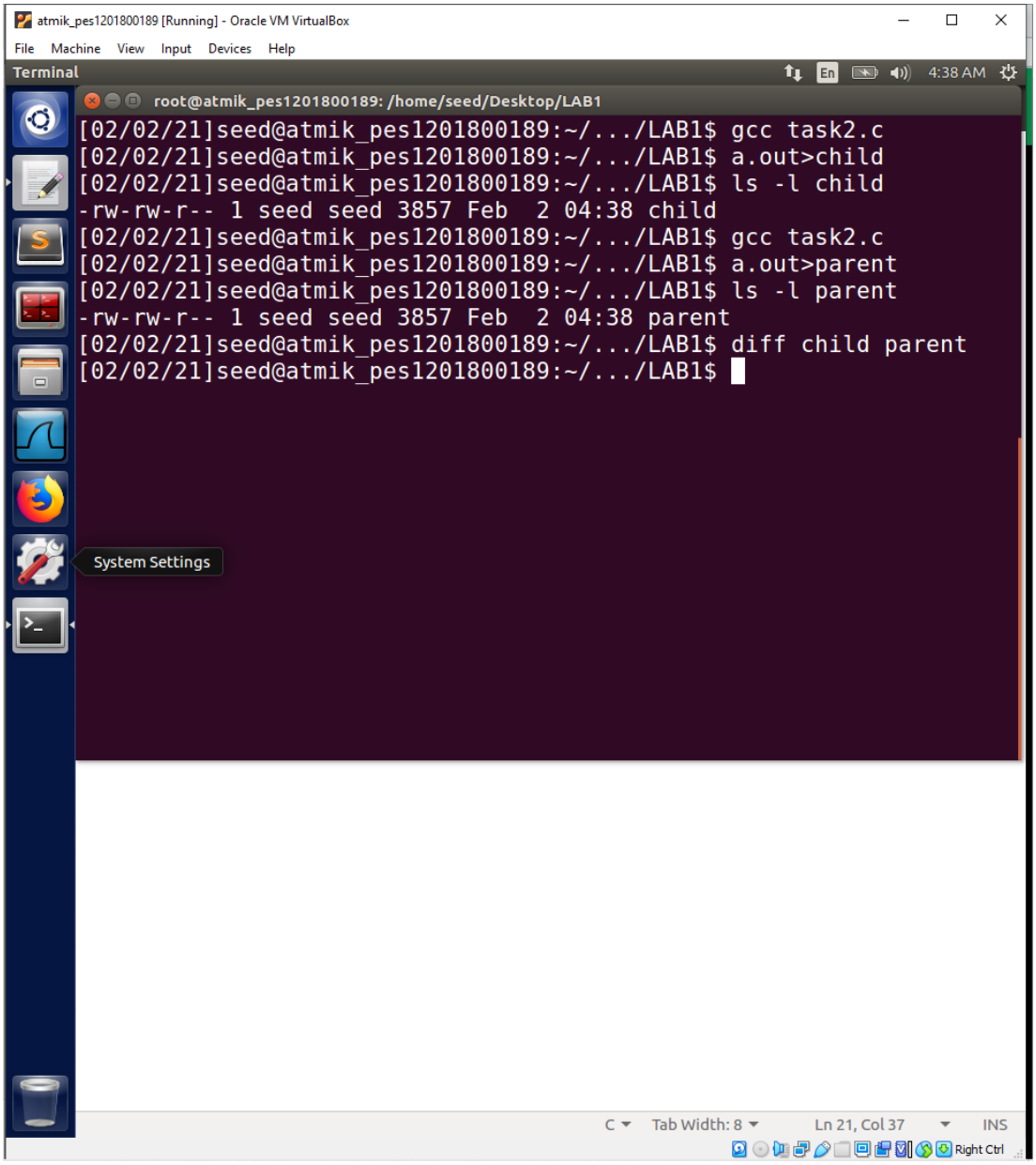
Step 2

- the export command is used to set a variable in the current shell and for all the processes running in said shell, here we have created a variable foo and set it to ‘test_string’
- the unset command is used to clear the variable that we just set (using export), here we unset foo and hence when we printenv after export it prints the string stored but after unsetting its empty



TASK 2

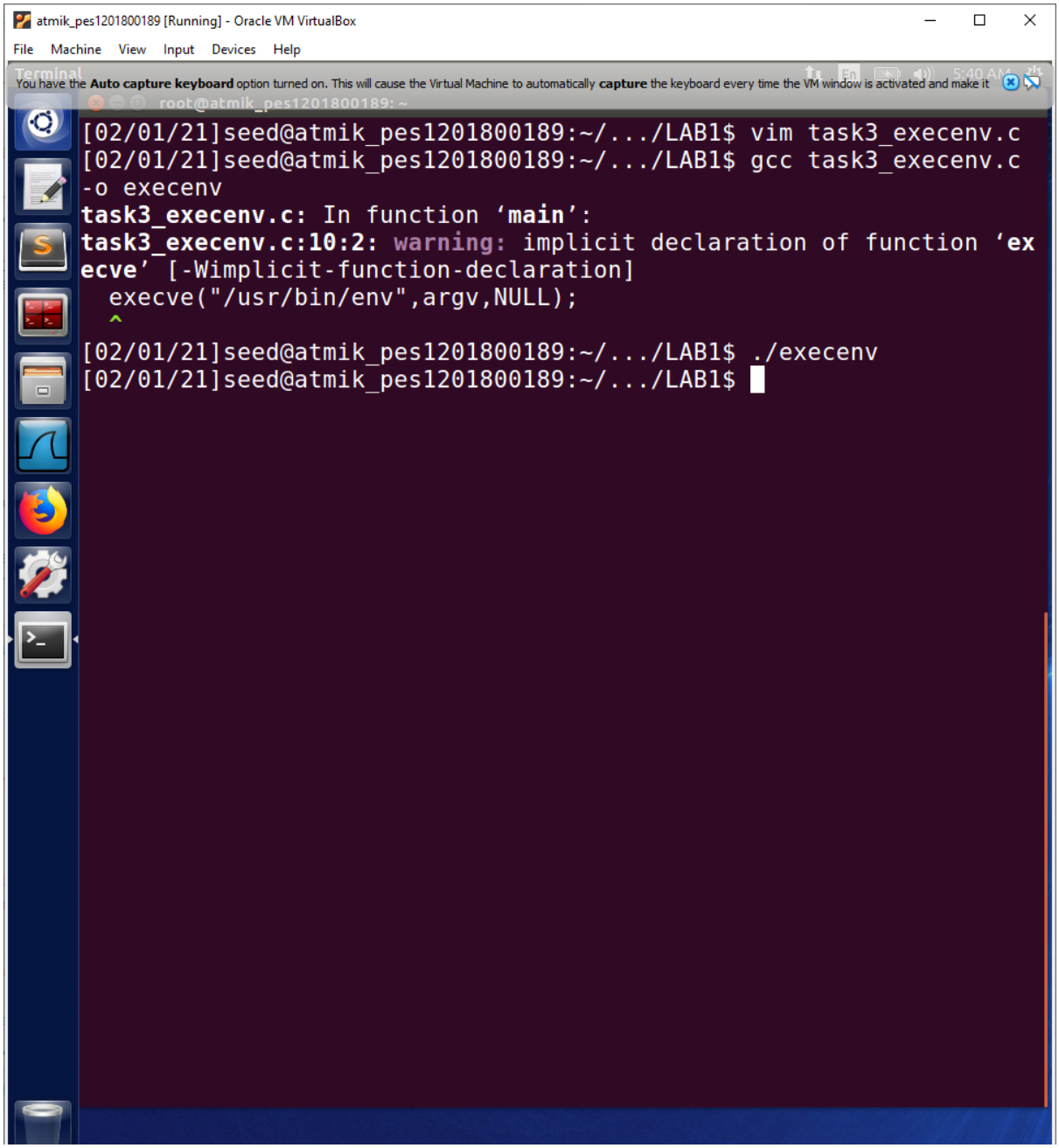
- values of environment variables for child are stored in child and same for parent
- diff child parent shows that all the environmental variables are inherited by child process
- but each process has its own environment hence, any changes made in environment variables will not be visible in child and vice versa



Task 3

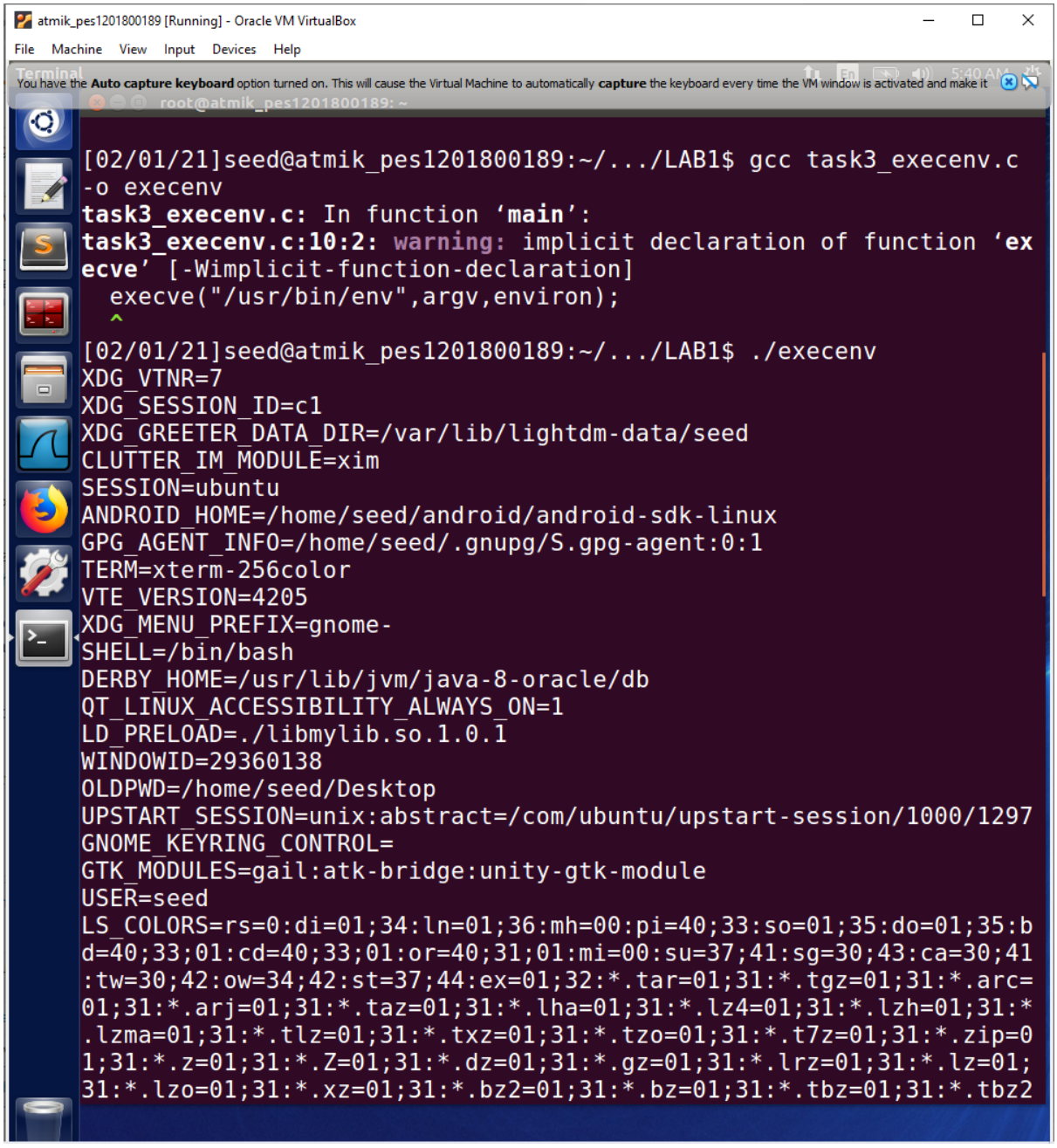
Step 1:

- since execve passes NULL to new program , ./execenc doesn’t print anything



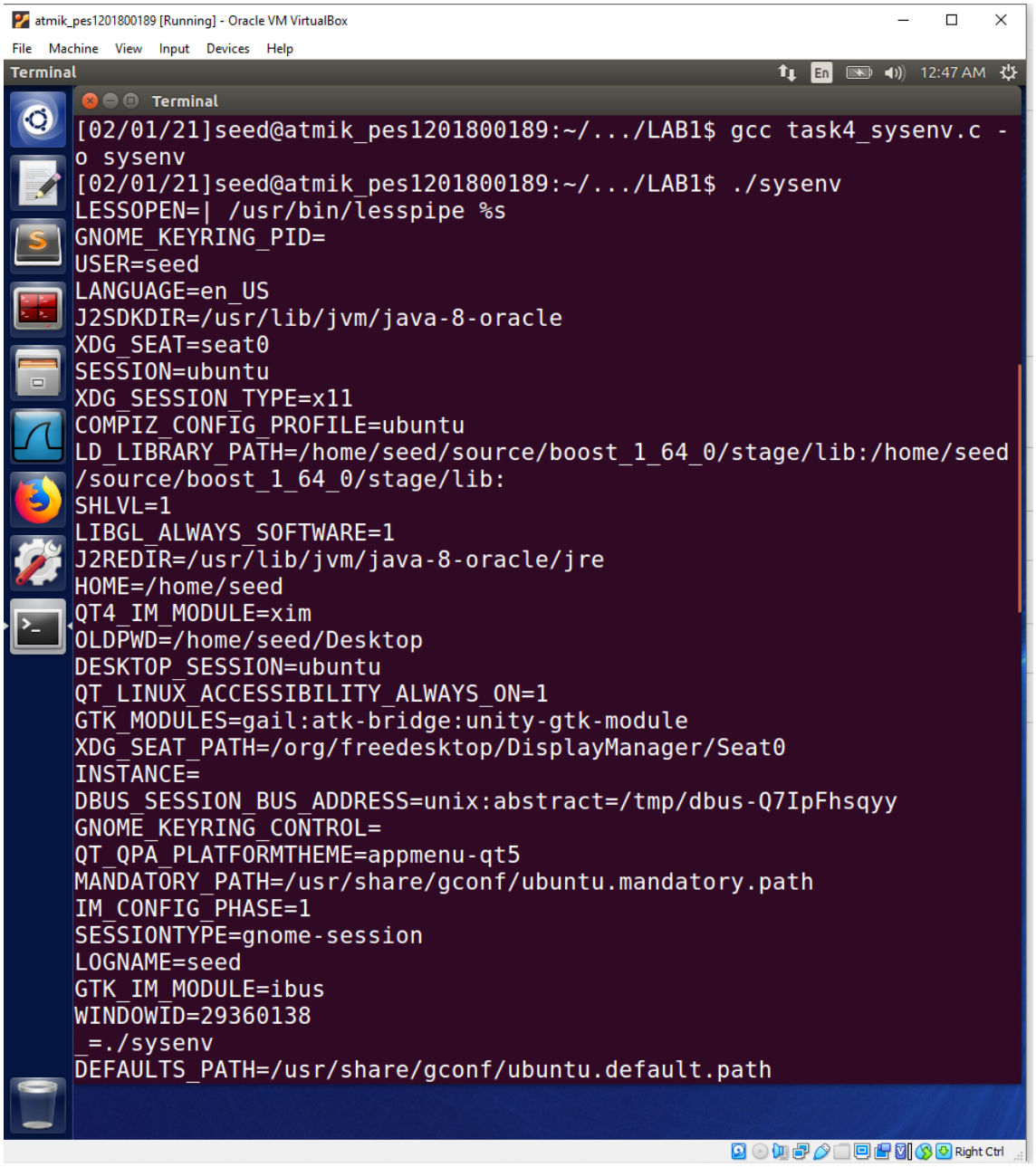
Step 2:

- since we pass environ(which is a pointer to a string array having key value pairs of environment variables) to the new program, it prints a full list of env variables acquired by program

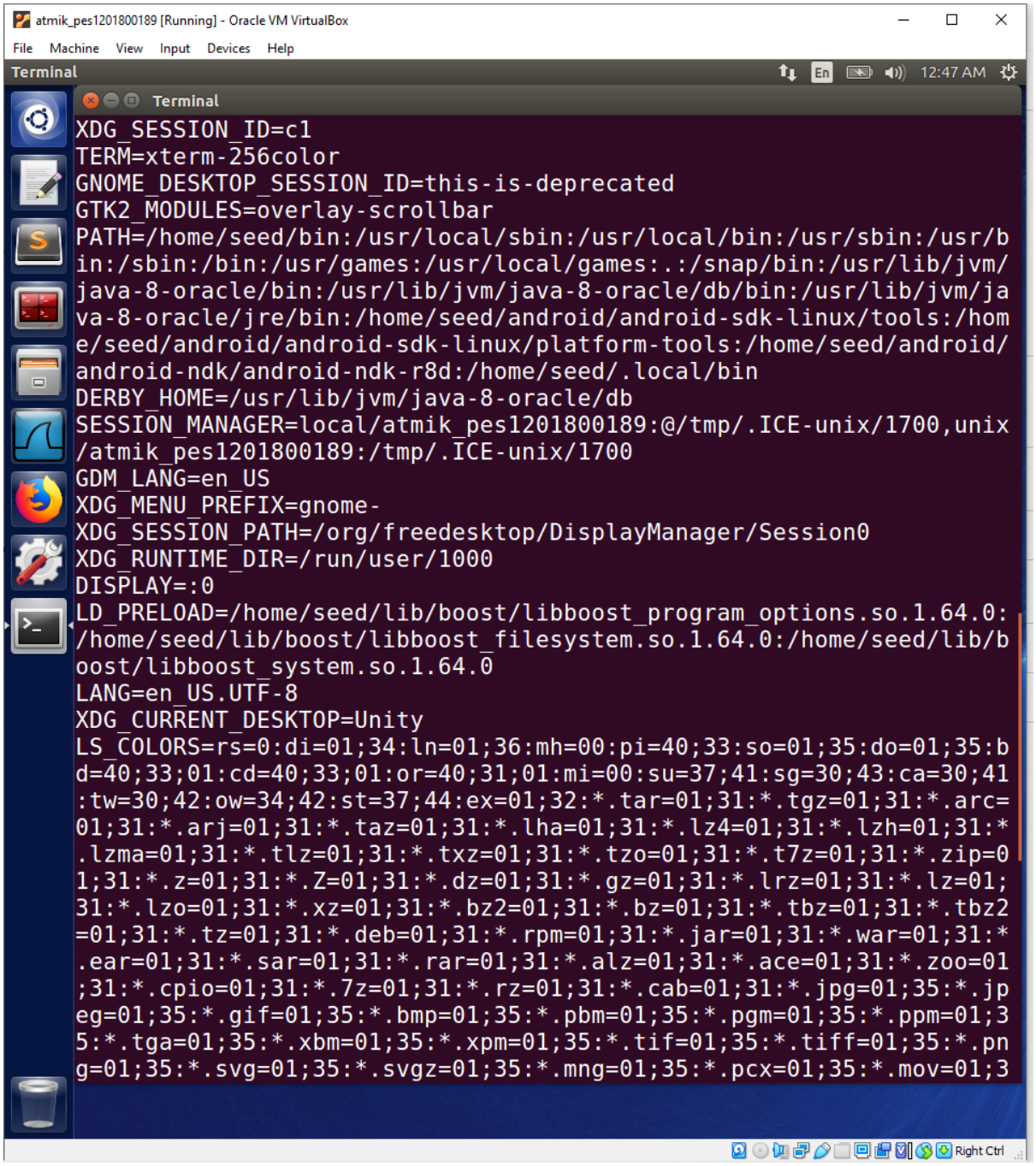


Task 4

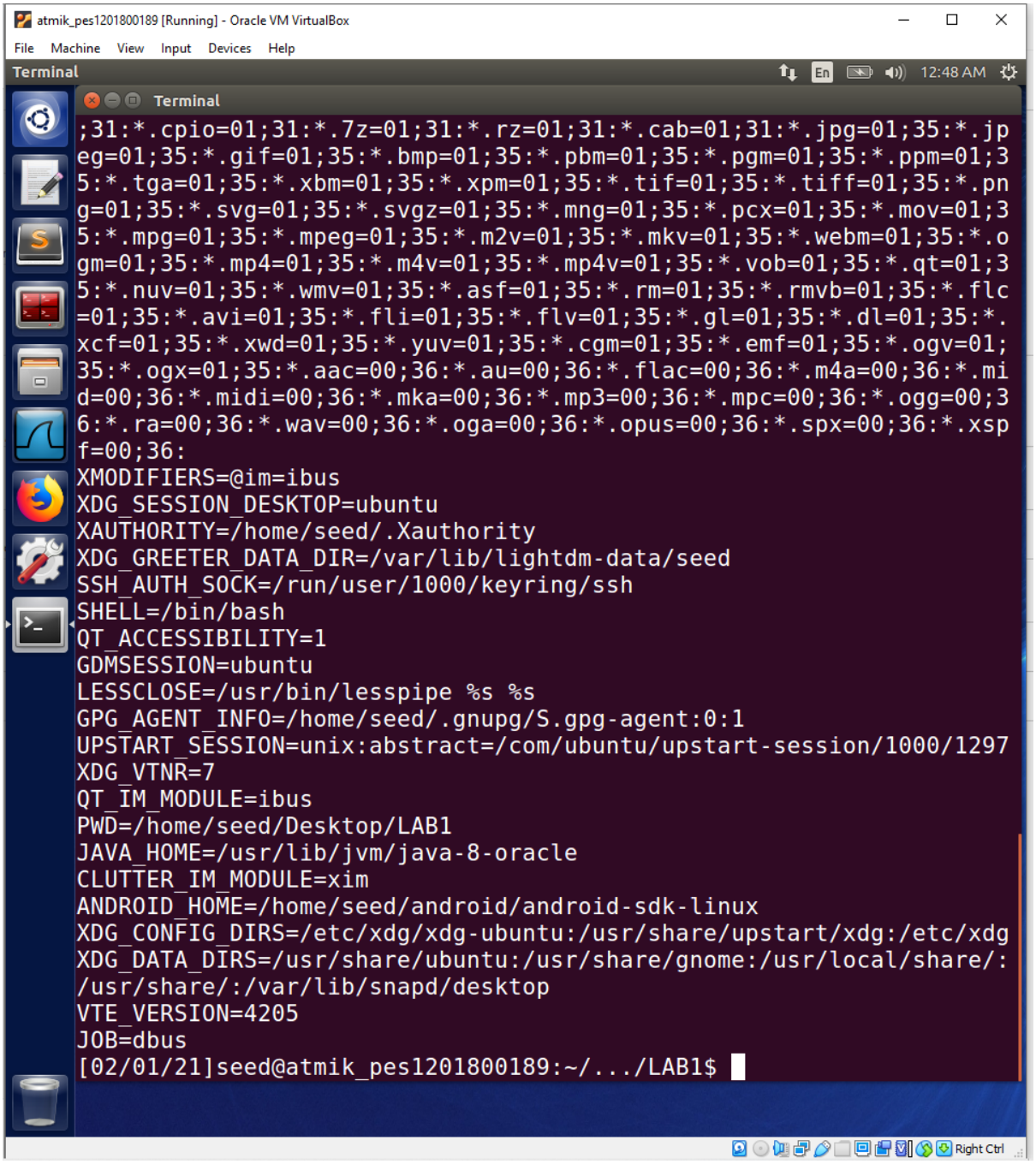
- Environment variables of the program is passed to /bin/sh by execve() which is called by execl() and system()
- ./sysenv displays all the environment variables as it runs the executable



4.1


atmik_pes1201800189 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
12:47 AM
Terminal
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/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
DERBY_HOME=/usr/lib/jvm/java-8-oracle/db
SESSION_MANAGER=local/atmik_pes1201800189:@/tmp/.ICE-unix/1700,unix/atmik_pes1201800189:/tmp/.ICE-unix/1700
GDM_LANG=en_US
XDG_MENU_PREFIX=gnome-
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_RUNTIME_DIR=/run/user/1000
DISPLAY=:0
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
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: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:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=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:*.jar=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;3
Right Ctrl

4.2

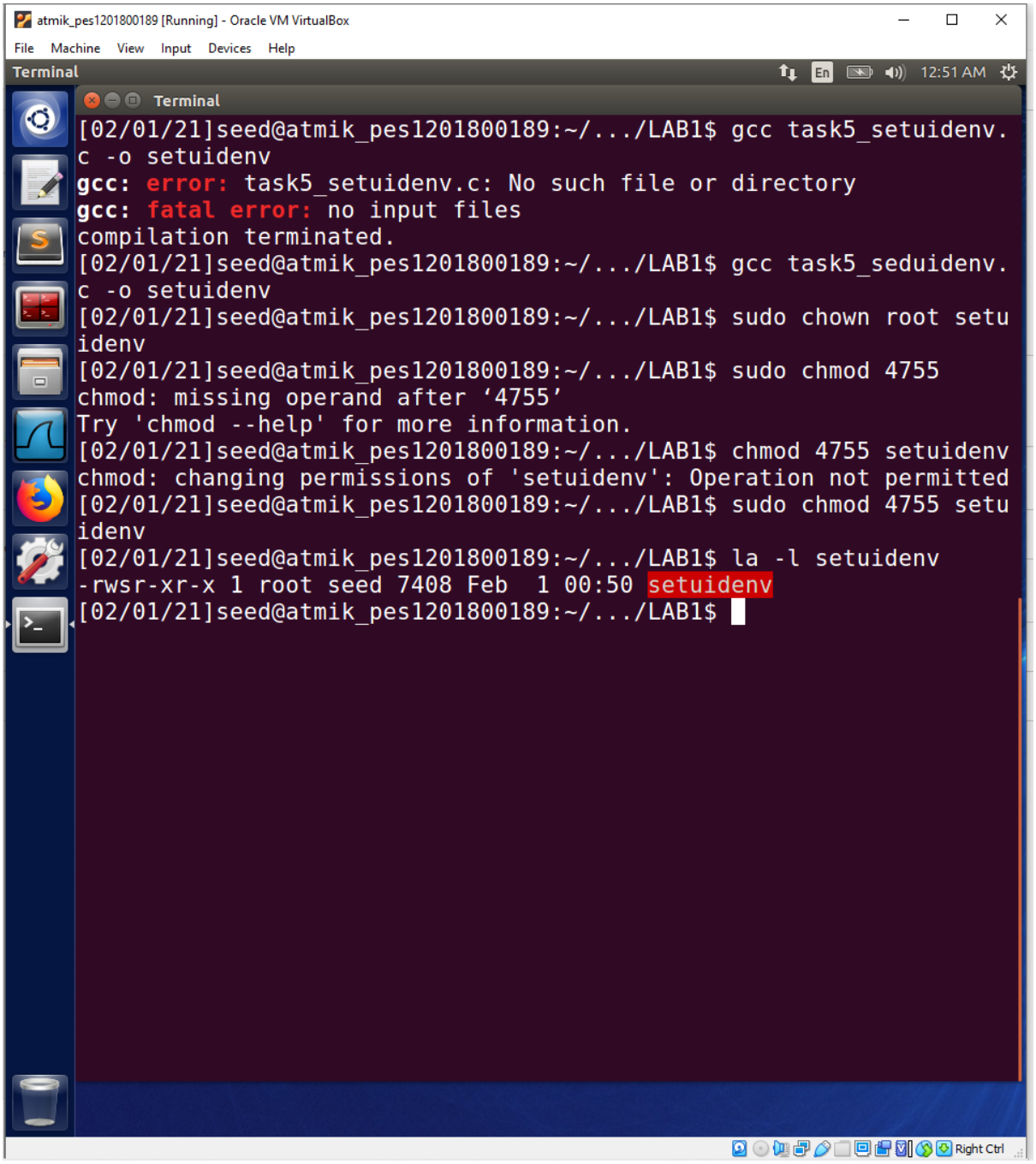

atmik_pes1201800189 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
12:48 AM
Terminal
;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.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:*.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:
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
QT_ACCESSIBILITY=1
GDMSESSION=ubuntu
LESSCLOSE=/usr/bin/lesspipe %s %s
GPG_AGENT_INFO=/home/seed/.gnupg/S.gpg-agent:0:1
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1297
XDG_VTNR=7
QT_IM_MODULE=ibus
PWD=/home/seed/Desktop/LAB1
JAVA_HOME=/usr/lib/jvm/java-8-oracle
CLUTTER_IM_MODULE=xim
ANDROID_HOME=/home/seed/android/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=dbus
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$
Right Ctrl

4.3

Task 5

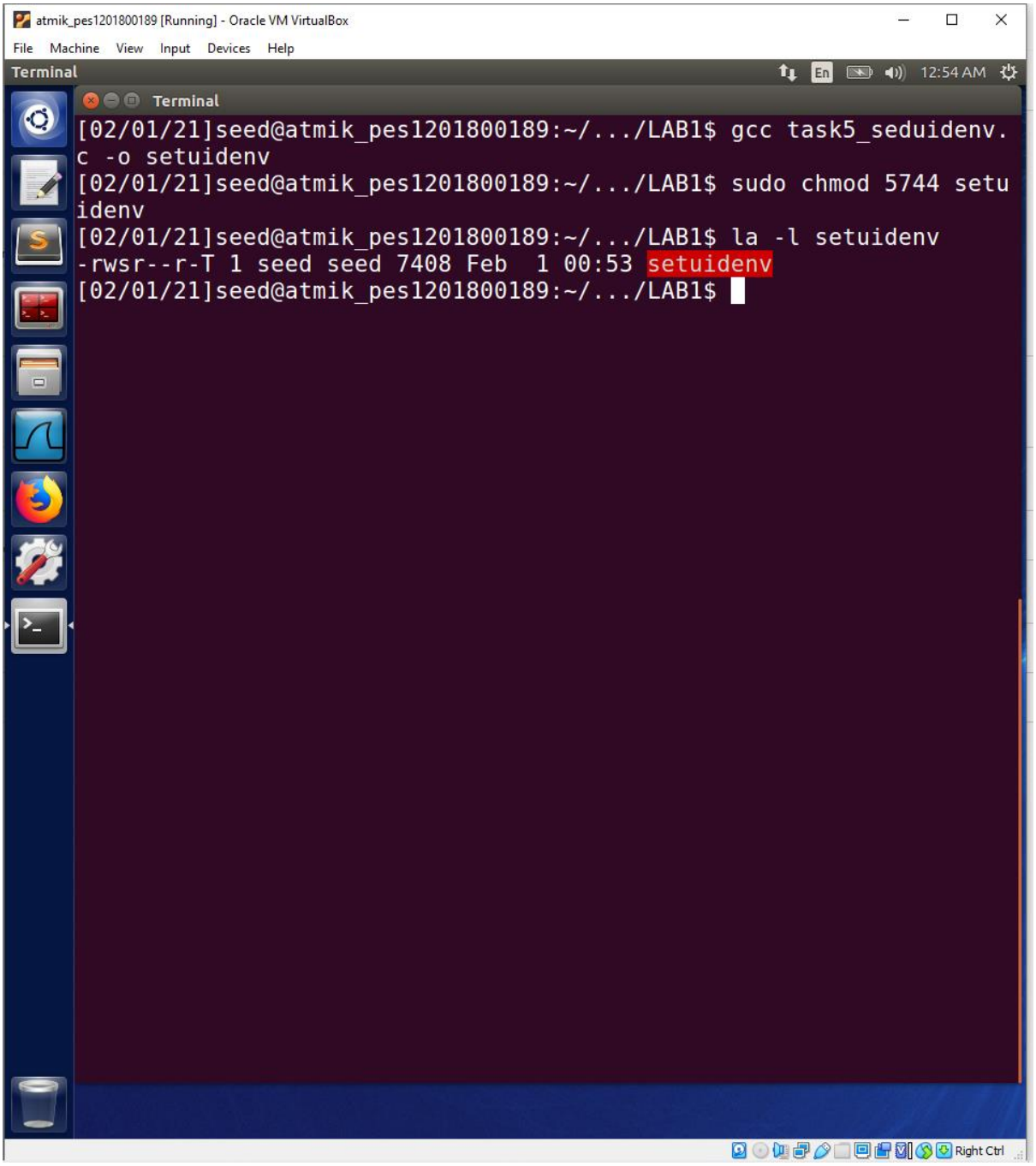
Step 1:

- chown changes the ownership of the setuidenv executable to root
- chmod 4755 sets the write, read and execute permissions, the permissions are then displayed using la -l command



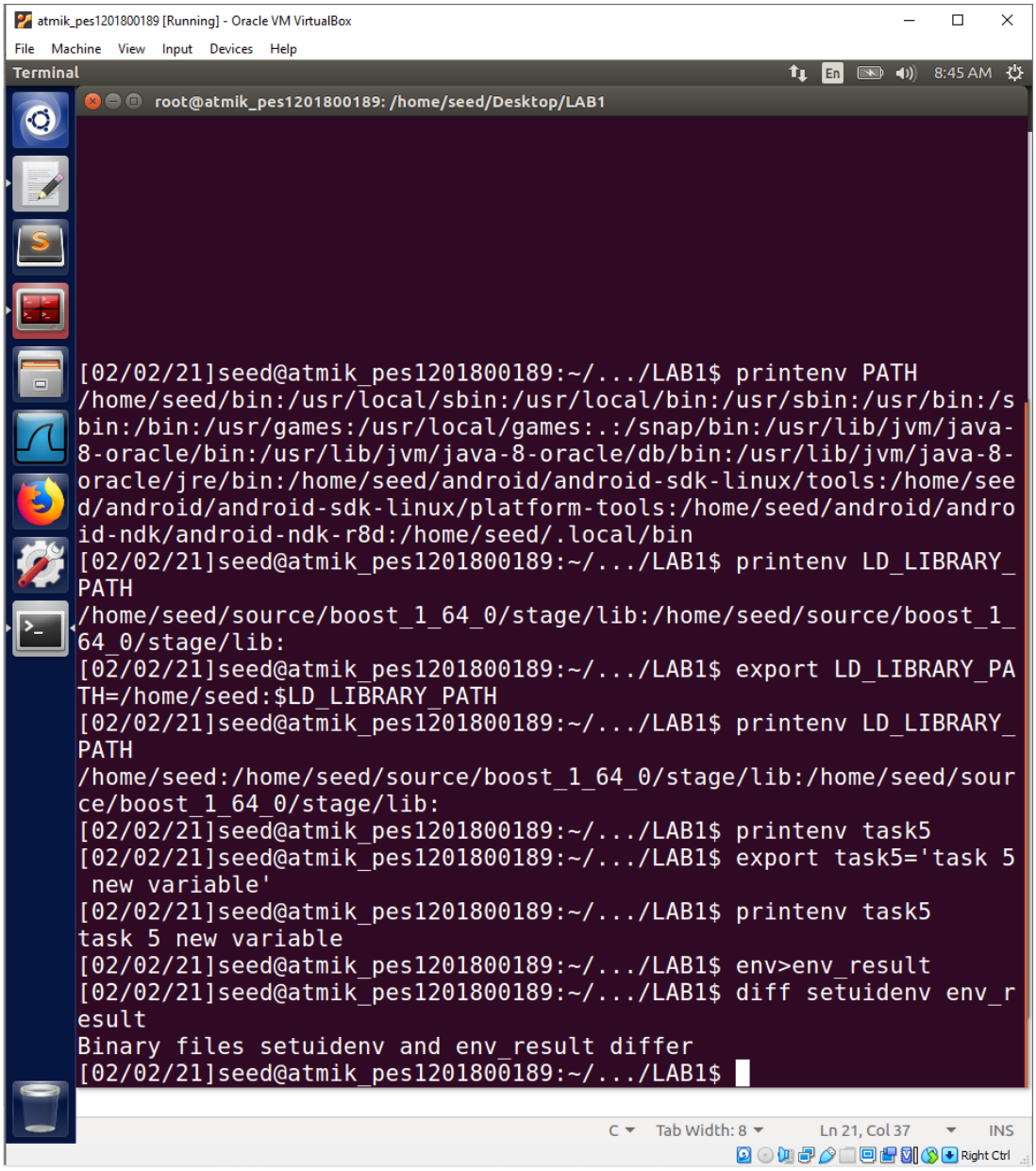
Step 2:

- chmod 5744 sets the permissions of setuid executable to current working directory which in our case is Desktop/LAB1



Step 3:

- it is observed from the screenshots that all the env variables set in setuidenv are also set in the child process env_result meaning that the program acquires the values of the environment variables based on the env its running on

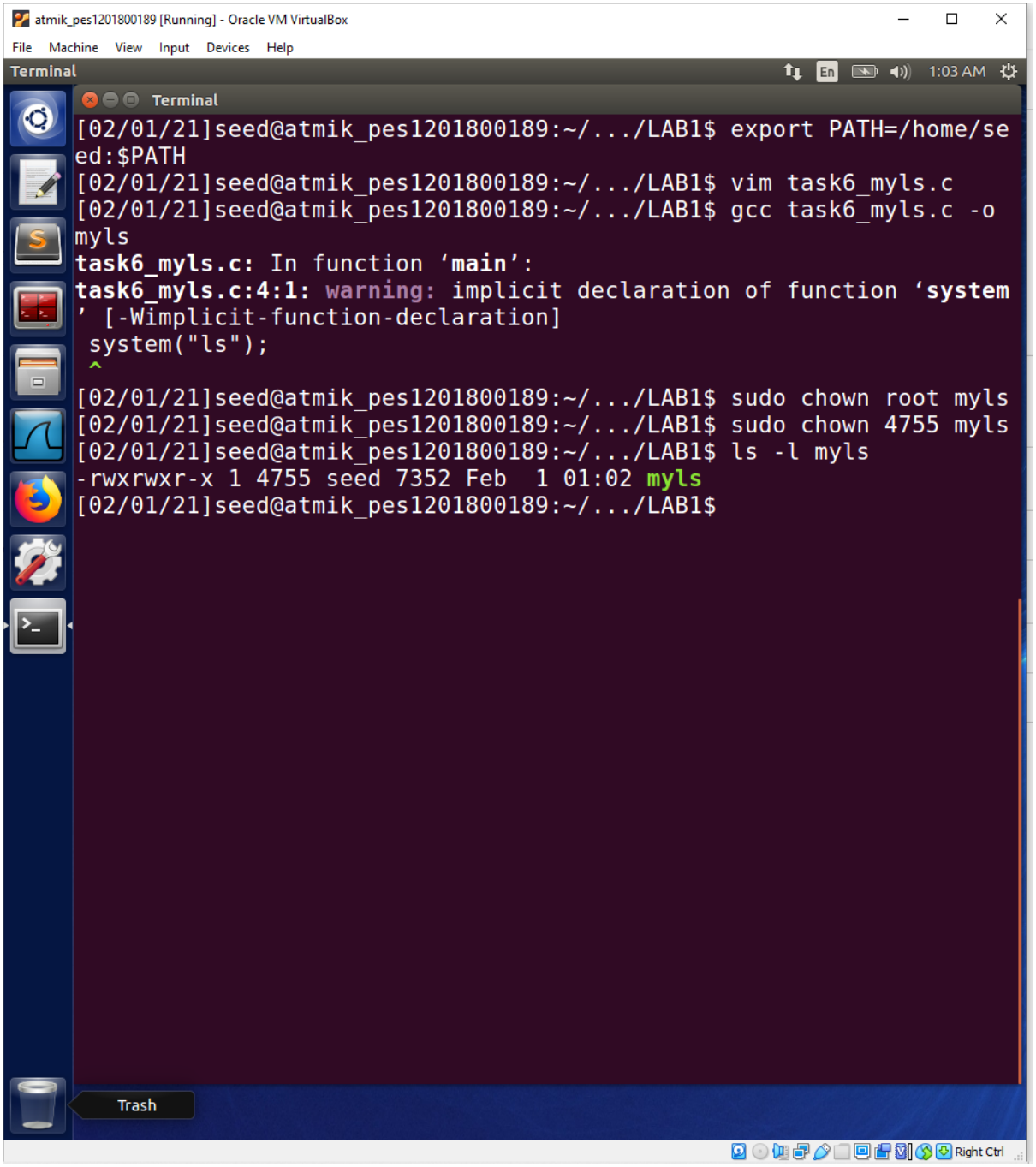


5.3 1

Task 6

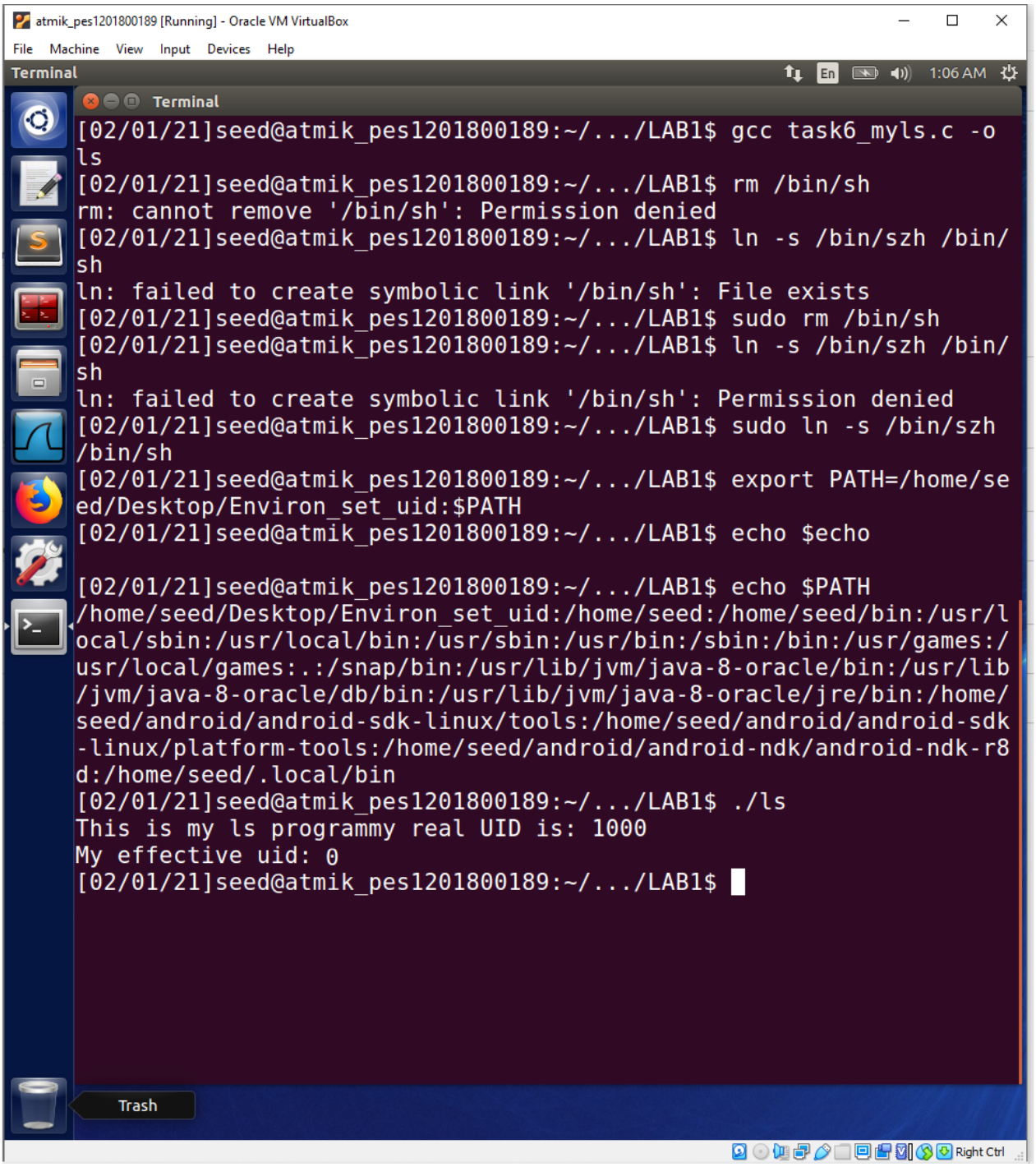
Step 1:

- Export appends /home/seed to the existing path
- As mentioned before, chown and chmod change ownership to root and permissions of the executable respectively



Step 2:

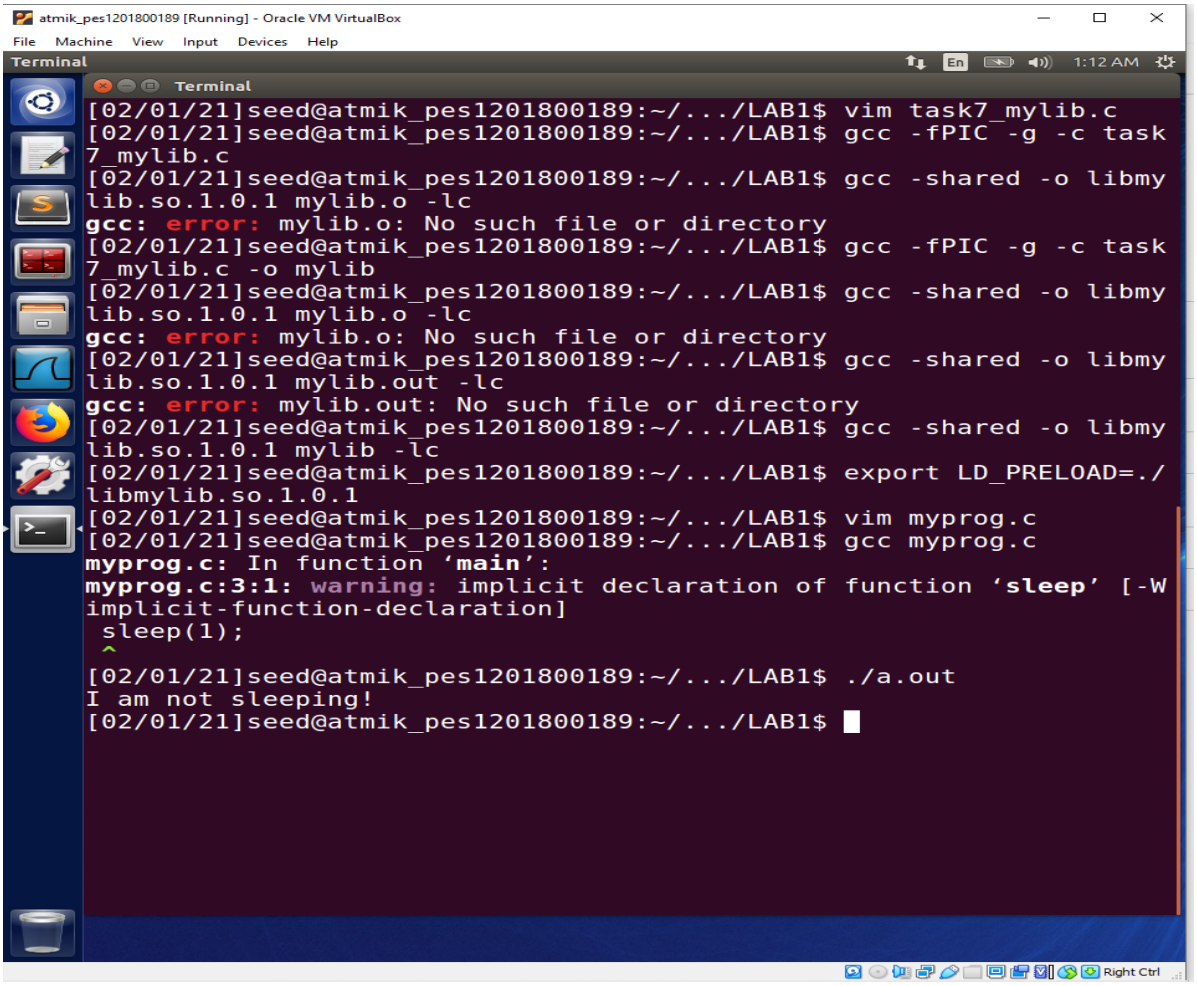
- rm /bin/sh and ln -s command will replace /bin/sh with the new z shell located at /bin/zsh to execute new ls program
- ./myls runs the executable from the program which executes the system(ls) function call, the results printed indicate that the command is running in superuser



Task 7

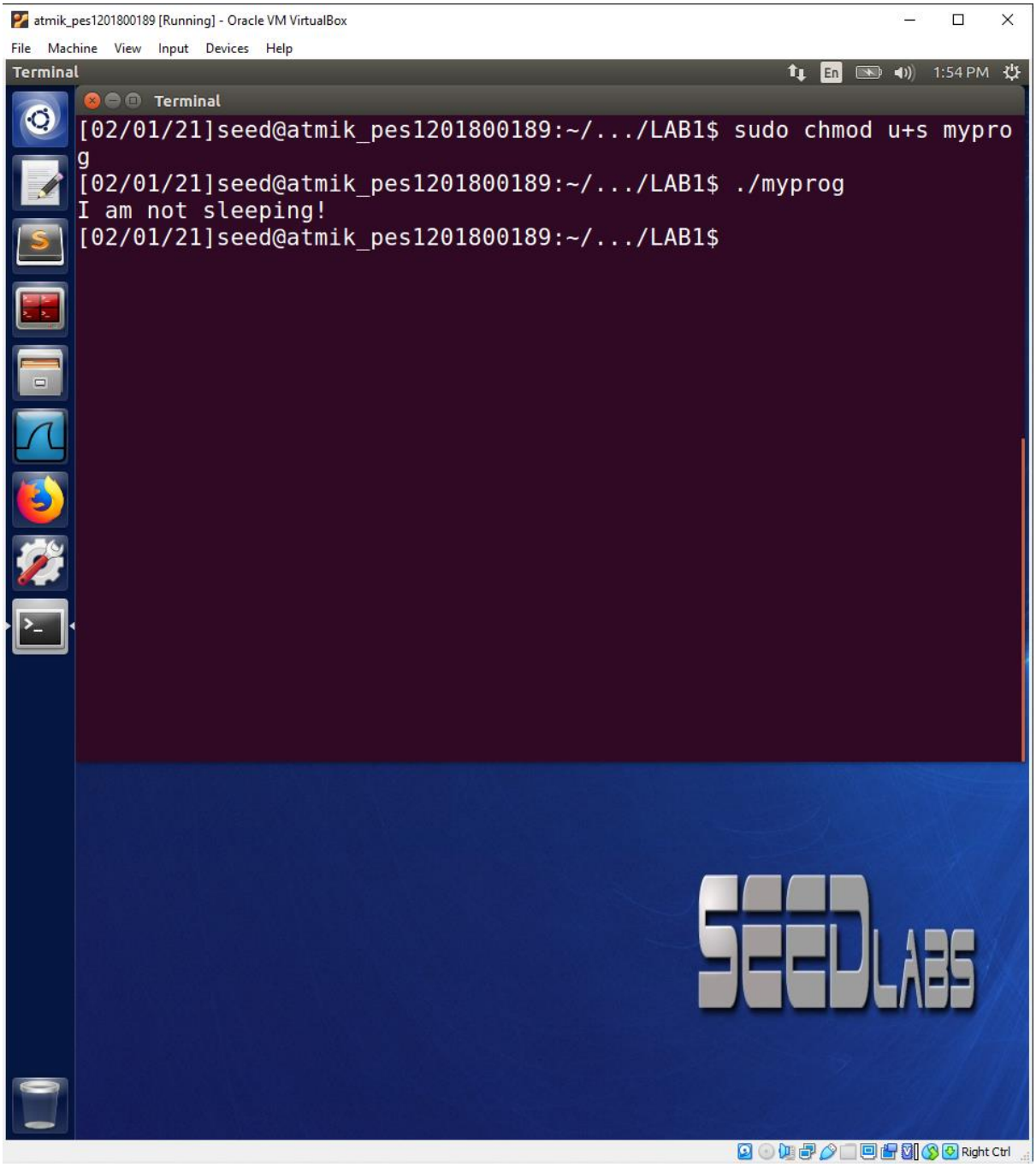
Step 1:

- fPIC option helps in generating position independent code that can be used with shared libraries (which is done in the next command) and would be suitable for dynamic linking (which is better than fpic option as here there is no limit on the global offset table size)
- shared option produces an object which can the be linked with other objects to form a executable, in this case libmylib.so.1.0.1 (dynamic lib)
- export LD_PRELOAD makes the variable point to the dynamic linker/loader created with the -shared step
- we then compile the program in the same directory as the dynamic library libmylib.so.1.0.1 to get required output

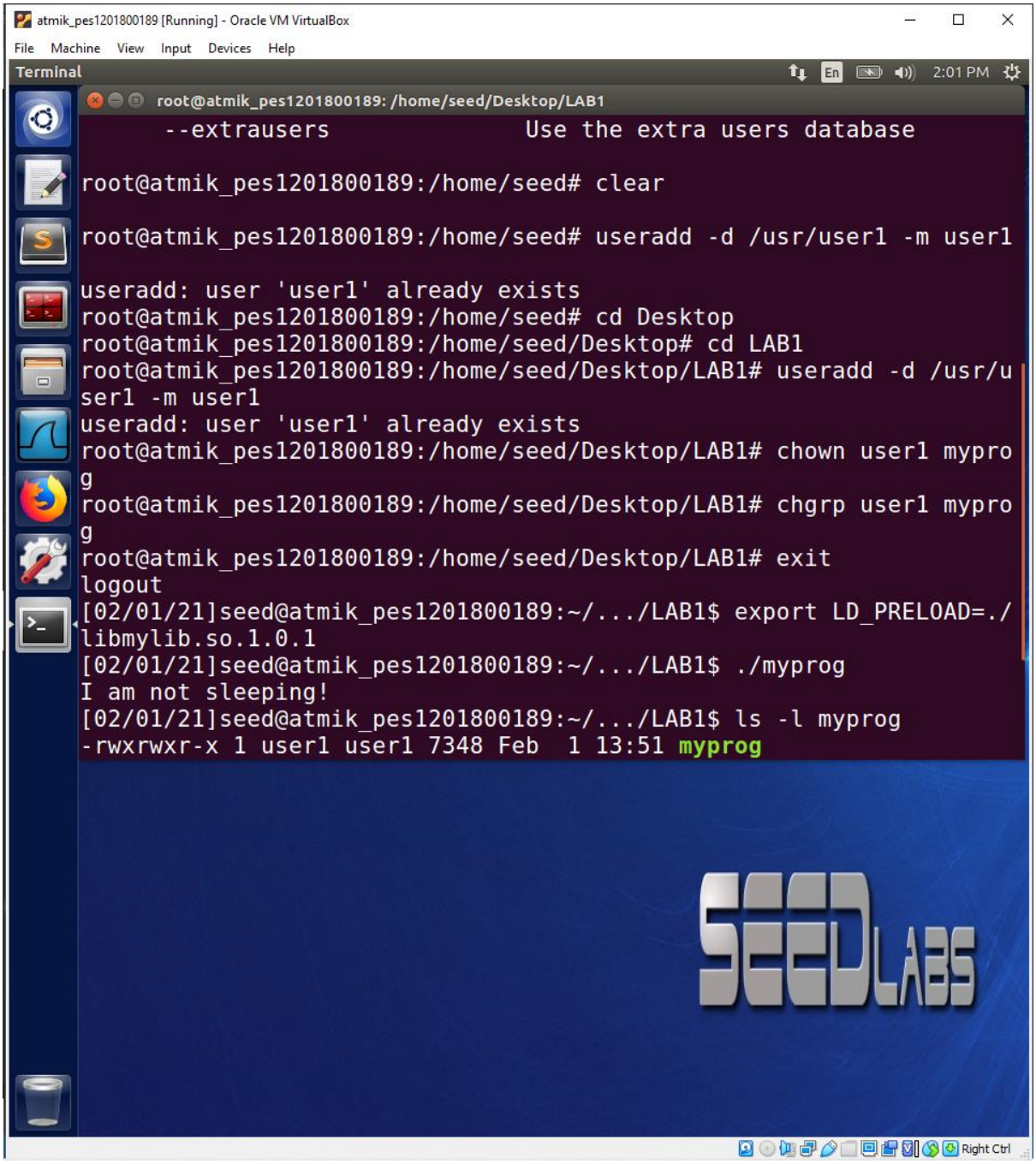


Step 2:

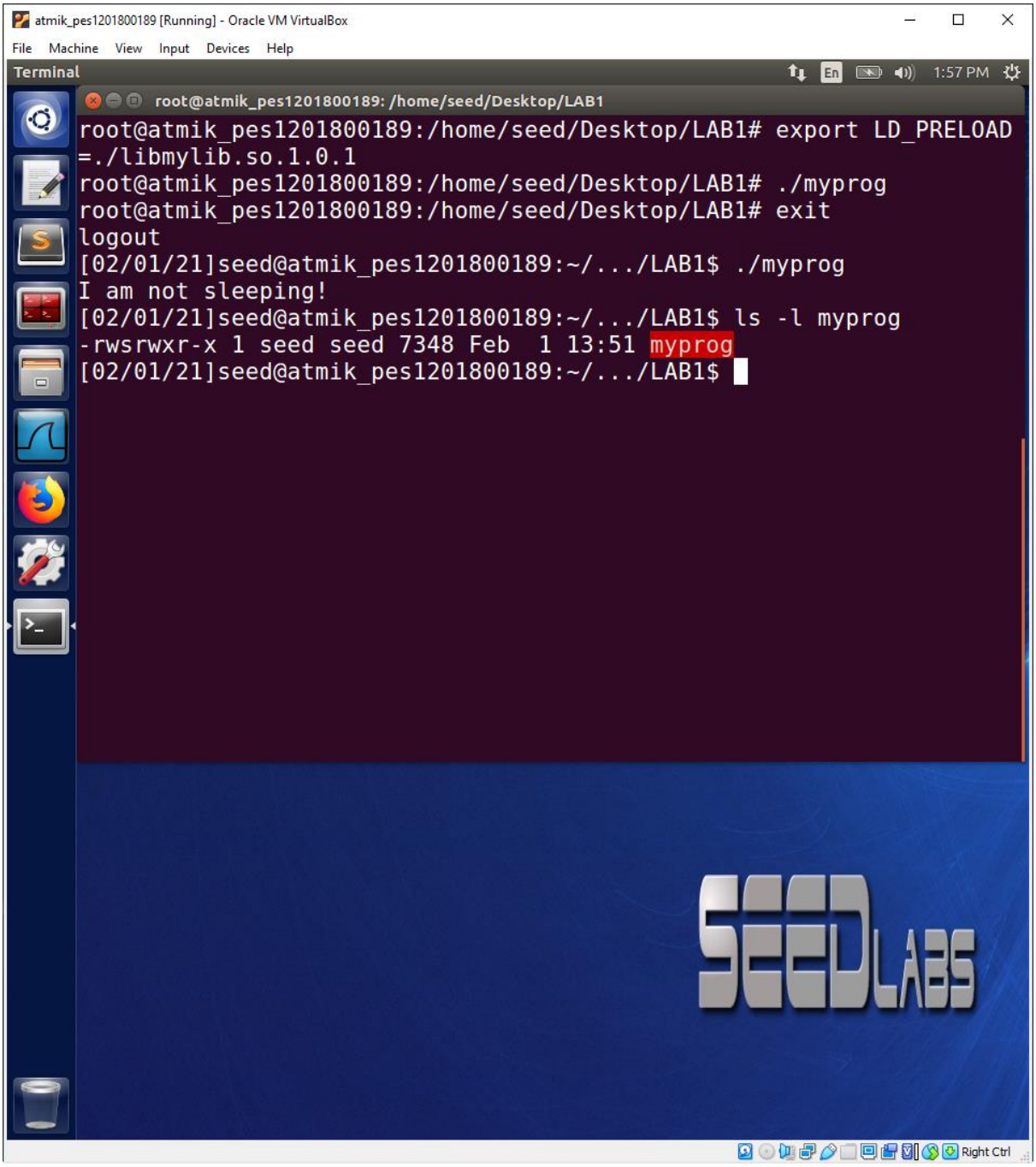
- we execute under 4 different behaviours here, the way we initialise these behaviours/configurations is detailed in the previous tasks
    - run myprog as regular user
        - LD_PRELOAD loads the function defined in the libmylib.so.1.0.1 into RAM from main memory and executed the function defined in c file which would be library for sleep func.
    - make myprog a setuid root program and run as normal user
        - LD_PRELOAD has the value ./libmylib.so.1.0.1 as the program is running in the environment that it was created in. hence we know that environment variables will be the same for everything running in /Desktop/LAB1. This will then execute similar to first configuration with the UID of the owner and not current user.
    - make myprog a setuid root program, export the LD_PRELOAD environment variable in root once again and run
        - since we are in a different environment (root user) hence we need to export LD_PRELOAD again as it isn't defined here. Once we do that we can execute the program and it will work as before
    - make it set uid user1 program and run as non root user
        - the permissions are assigned only to user1 and since the /Desktop/LAB1 is the superuser , it could access myprog and LD_PRELOAD loaded and hence run the executable.


atmik_pes1201800189 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
1:54 PM
Terminal
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$ sudo chmod u+s mypro
g
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$ ./myprog
I am not sleeping!
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$
SEEDLABS
Right Ctrl

7.2 1


atmik_pes1201800189 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
2:01 PM
root@atmik_pes1201800189: /home/seed/Desktop/LAB1
--extrausers Use the extra users database
root@atmik_pes1201800189:/home/seed# clear
root@atmik_pes1201800189:/home/seed# useradd -d /usr/user1 -m user1
useradd: user 'user1' already exists
root@atmik_pes1201800189:/home/seed# cd Desktop
root@atmik_pes1201800189:/home/seed/Desktop# cd LAB1
root@atmik_pes1201800189:/home/seed/Desktop/LAB1# useradd -d /usr/user1 -m user1
useradd: user 'user1' already exists
root@atmik_pes1201800189:/home/seed/Desktop/LAB1# chown user1 myprog
root@atmik_pes1201800189:/home/seed/Desktop/LAB1# chgrp user1 myprog
root@atmik_pes1201800189:/home/seed/Desktop/LAB1# exit
logout
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$ export LD_PRELOAD=./libmylib.so.1.0.1
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$ ./myprog
I am not sleeping!
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$ ls -l myprog
-rwxrwxr-x 1 user1 user1 7348 Feb 1 13:51 myprog
SEEDLABS
Right Ctrl

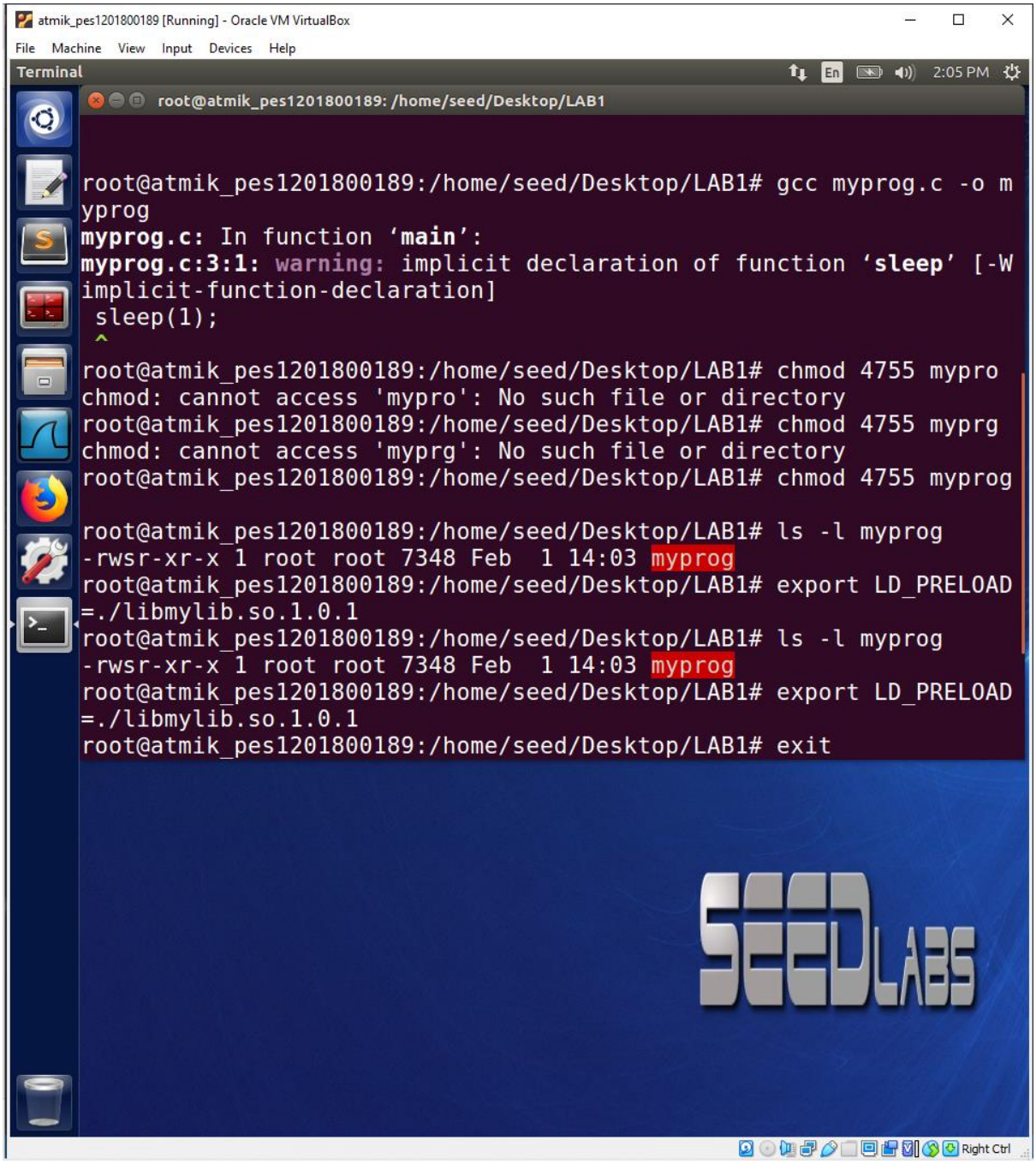
7.2 2


atmik_pes1201800189 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
1:57 PM
root@atmik_pes1201800189: /home/seed/Desktop/LAB1
root@atmik_pes1201800189:/home/seed/Desktop/LAB1# export LD_PRELOAD
=./libmylib.so.1.0.1
root@atmik_pes1201800189:/home/seed/Desktop/LAB1# ./myprog
root@atmik_pes1201800189:/home/seed/Desktop/LAB1# exit
logout
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$ ./myprog
I am not sleeping!
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$ ls -l myprog
-rwsrwxr-x 1 seed seed 7348 Feb 1 13:51 myprog
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$
SEEDLABS
Right Ctrl

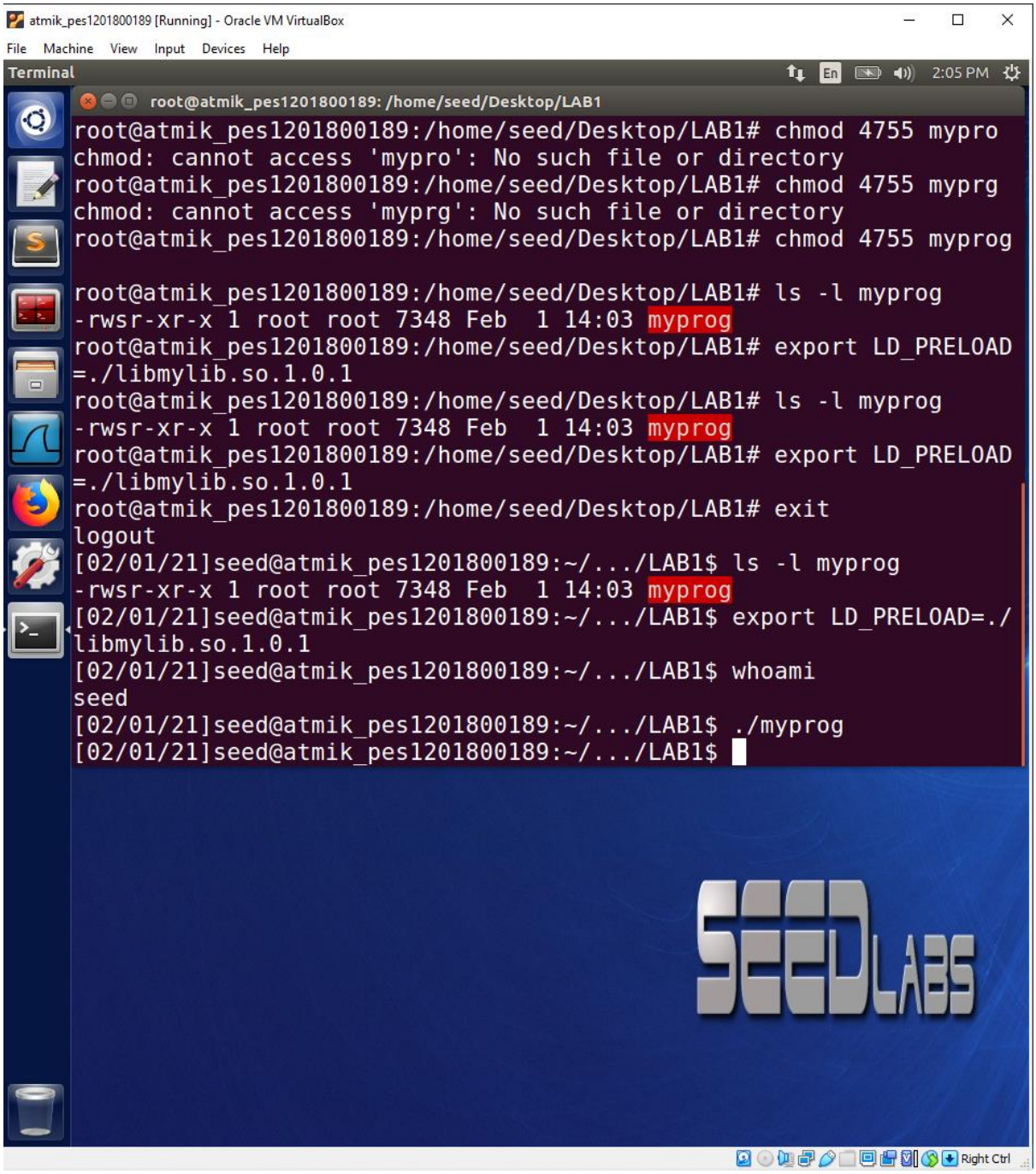
7.2 3

Step 3:

- The below screenshots enforce the reasonings for the above mentioned behaviours in step 2. We see that child processes don’t necessarily inherit environment variables LD_PRELOAD)



7.3 1


atmik_pes1201800189 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
2:05 PM
root@atmik_pes1201800189: /home/seed/Desktop/LAB1
root@atmik_pes1201800189:/home/seed/Desktop/LAB1# chmod 4755 mypro
chmod: cannot access 'mypro': No such file or directory
root@atmik_pes1201800189:/home/seed/Desktop/LAB1# chmod 4755 myprg
chmod: cannot access 'myprg': No such file or directory
root@atmik_pes1201800189:/home/seed/Desktop/LAB1# chmod 4755 myprog
root@atmik_pes1201800189:/home/seed/Desktop/LAB1# ls -l myprog
-rwsr-xr-x 1 root root 7348 Feb 1 14:03 myprog
root@atmik_pes1201800189:/home/seed/Desktop/LAB1# export LD_PRELOAD
=./libmylib.so.1.0.1
root@atmik_pes1201800189:/home/seed/Desktop/LAB1# ls -l myprog
-rwsr-xr-x 1 root root 7348 Feb 1 14:03 myprog
root@atmik_pes1201800189:/home/seed/Desktop/LAB1# export LD_PRELOAD
=./libmylib.so.1.0.1
root@atmik_pes1201800189:/home/seed/Desktop/LAB1# exit
logout
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$ ls -l myprog
-rwsr-xr-x 1 root root 7348 Feb 1 14:03 myprog
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$ export LD_PRELOAD=./
libmylib.so.1.0.1
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$ whoami
seed
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$ ./myprog
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$
SEEDLABS
Right Ctrl

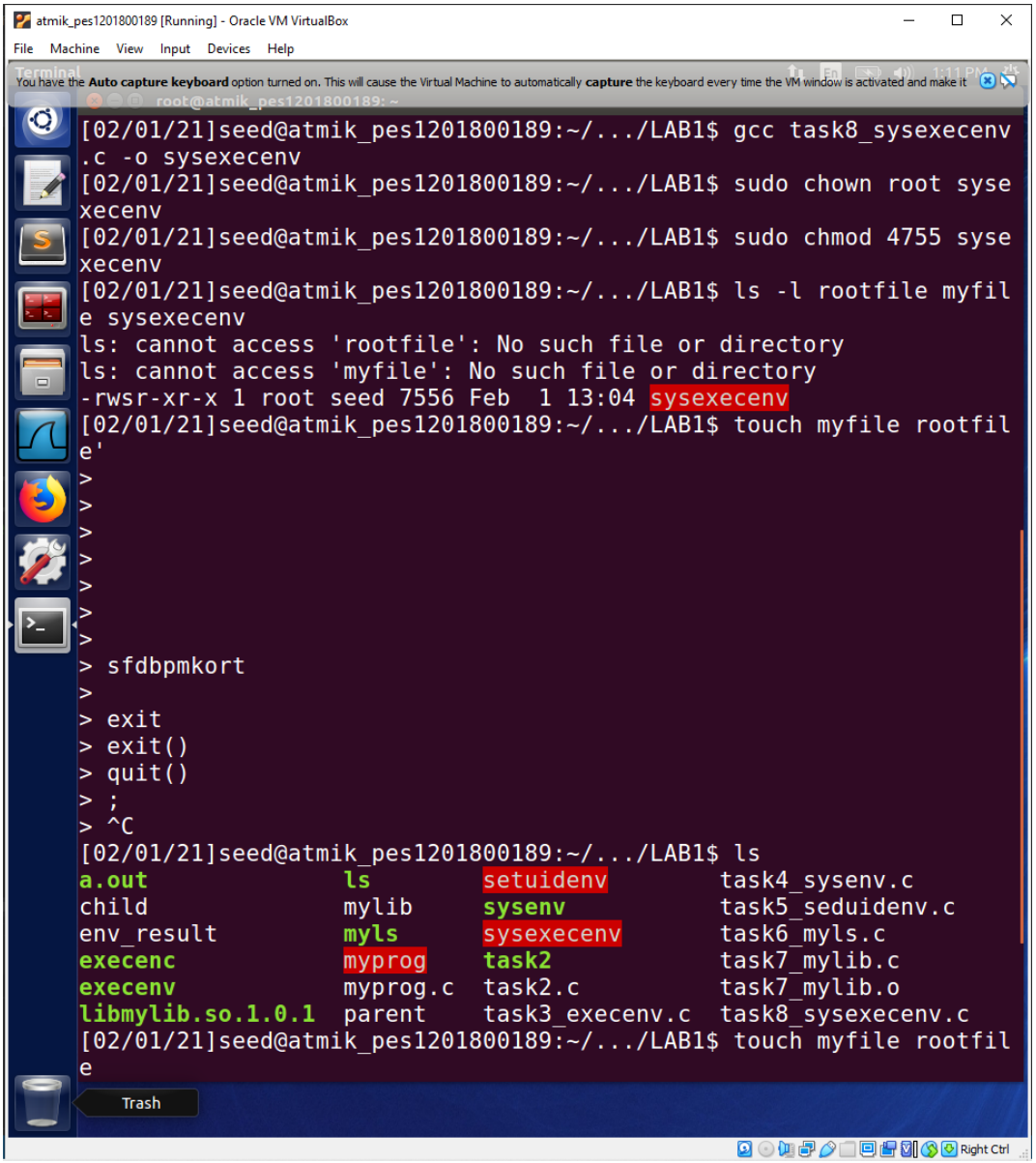
7.3 2

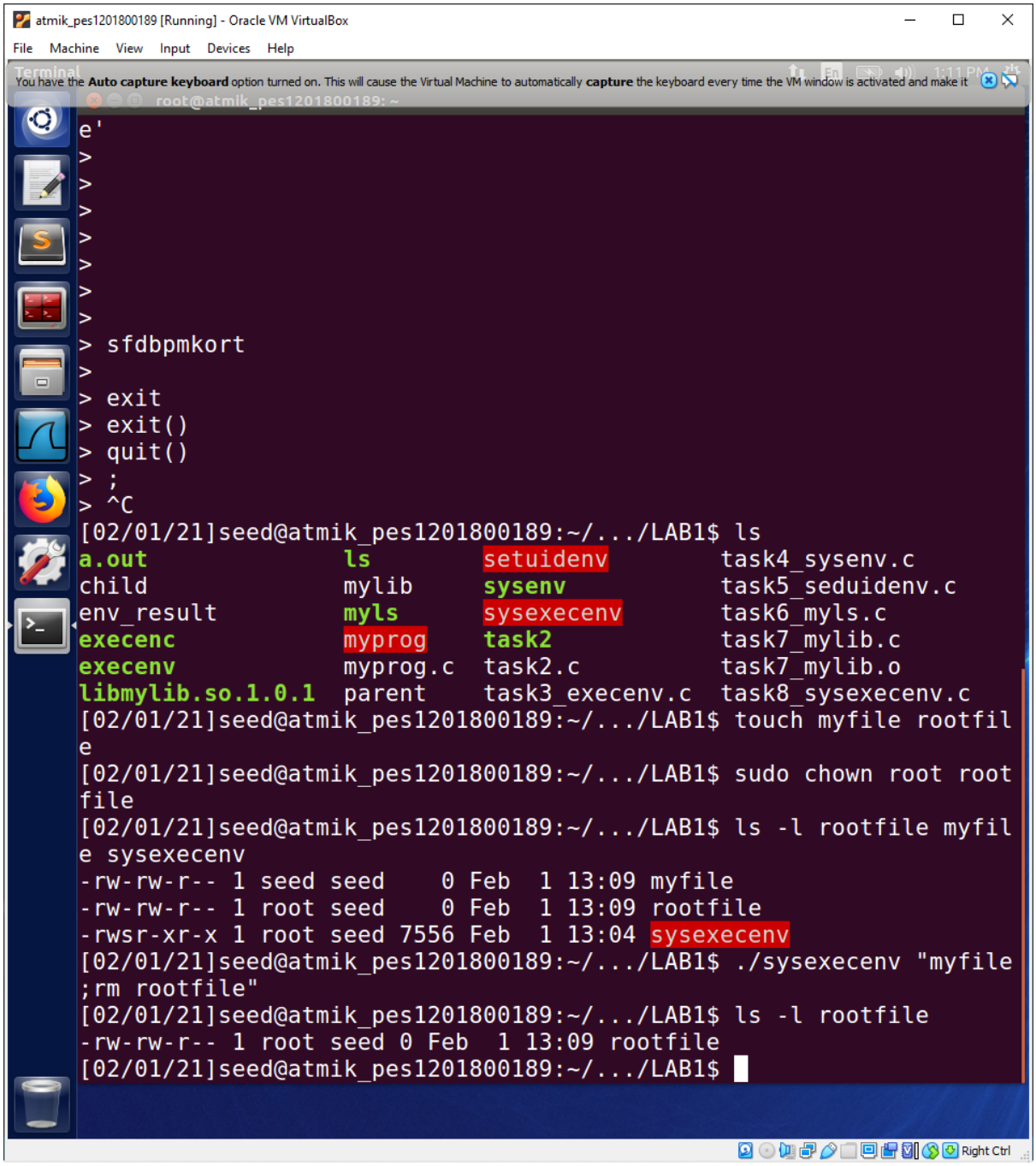
Task 8

Step 1:

- System() makes the /bin/cal file execute which inturn displays contents of myfile
- It also invokes the shell with root privileges as we are able to delete rootfile using rm rootfile which we wouldn't have been able to do if we didn't have those root privileges that system helps invoke

8.1 1

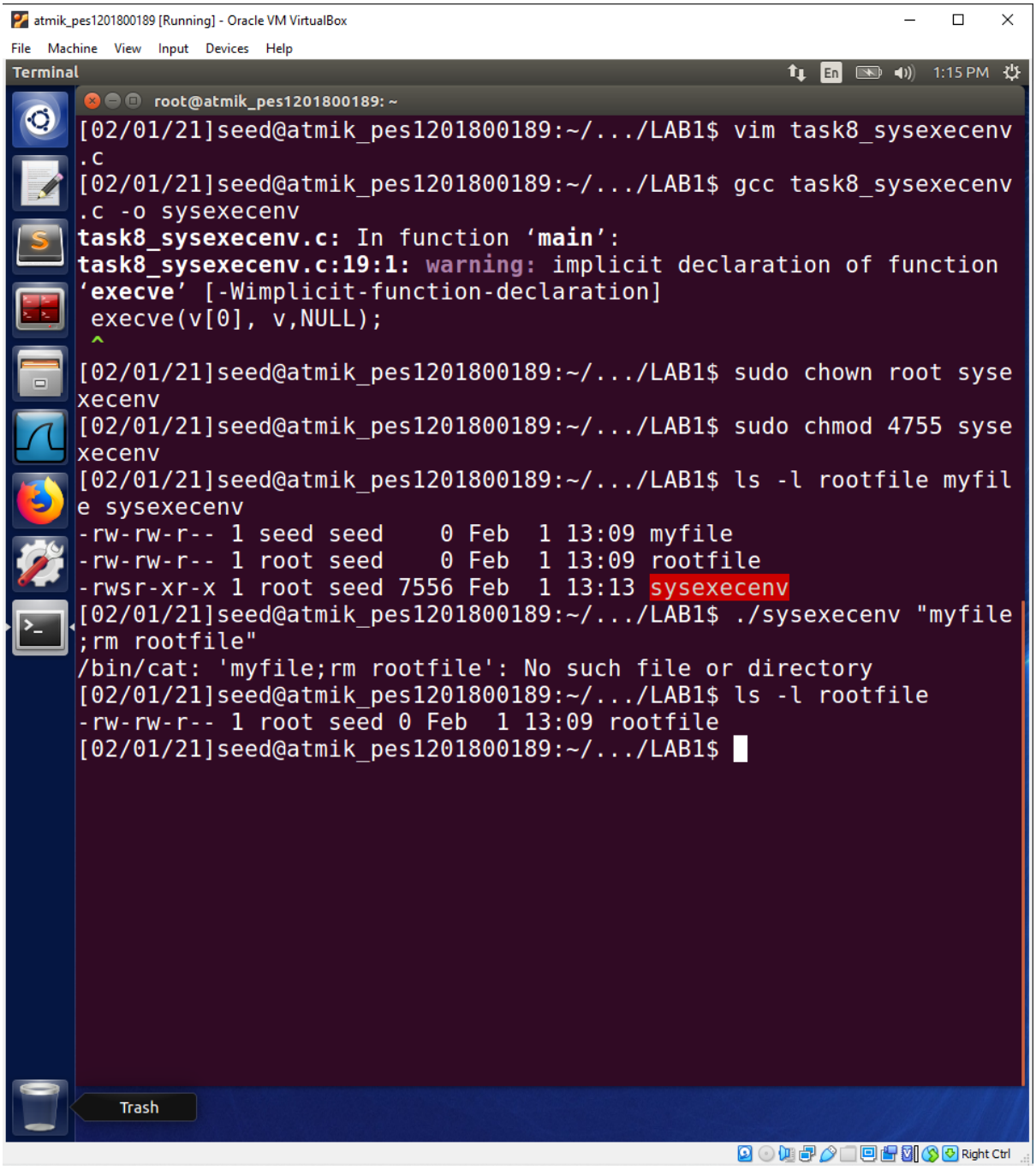



atmik_pes1201800189 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
You have the Auto capture keyboard option turned on. This will cause the Virtual Machine to automatically capture the keyboard every time the VM window is activated and make it
root@atmik_pes1201800189: ~
e'
>
>
>
>
>
>
>
> sfdbpmkort
>
> exit
> exit()
> quit()
> ;
> ^C
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$ ls
a.out ls setuidenv task4_sysenv.c
child mylib sysenv task5_seduidenv.c
env_result myls sysexecenv task6_myls.c
execenc myprog task2 task7_mylib.c
execenv myprog.c task2.c task7_mylib.o
libmylib.so.1.0.1 parent task3_execenv.c task8_sysexecenv.c
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$ touch myfile rootfil
e
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$ sudo chown root root
file
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$ ls -l rootfile myfil
e sysexecenv
-rw-rw-r-- 1 seed seed 0 Feb 1 13:09 myfile
-rw-rw-r-- 1 root seed 0 Feb 1 13:09 rootfile
-rwsr-xr-x 1 root seed 7556 Feb 1 13:04 sysexecenv
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$ ./sysexecenv "myfile
;rm rootfile"
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$ ls -l rootfile
-rw-rw-r-- 1 root seed 0 Feb 1 13:09 rootfile
[02/01/21]seed@atmik_pes1201800189:~/.../LAB1$
Right Ctrl

8.1 2

Step 2:

- Execve() doesn't invoke shell with root privileges and executes the /bin/cat executable which displays myfile and rootfile
- When we try to give same command as before now, it doesn't work as it is executed as /bin/cat 'myfile;rm rootfile' and not separately as before(/bin/cat myfile; rm rootfile)
- Hence this doesn't delete the rootfile and nothing happens , making it a safer option



Task 9

- Please note,I have changed permissions and executed the chown and chmod command before clearing, I cleared terminal screen by accident.
- We are able to modify cat/zzz through ./capleak and not manually as only capleak has root privileges and not us. Hence this enforces the principle of least privilege.