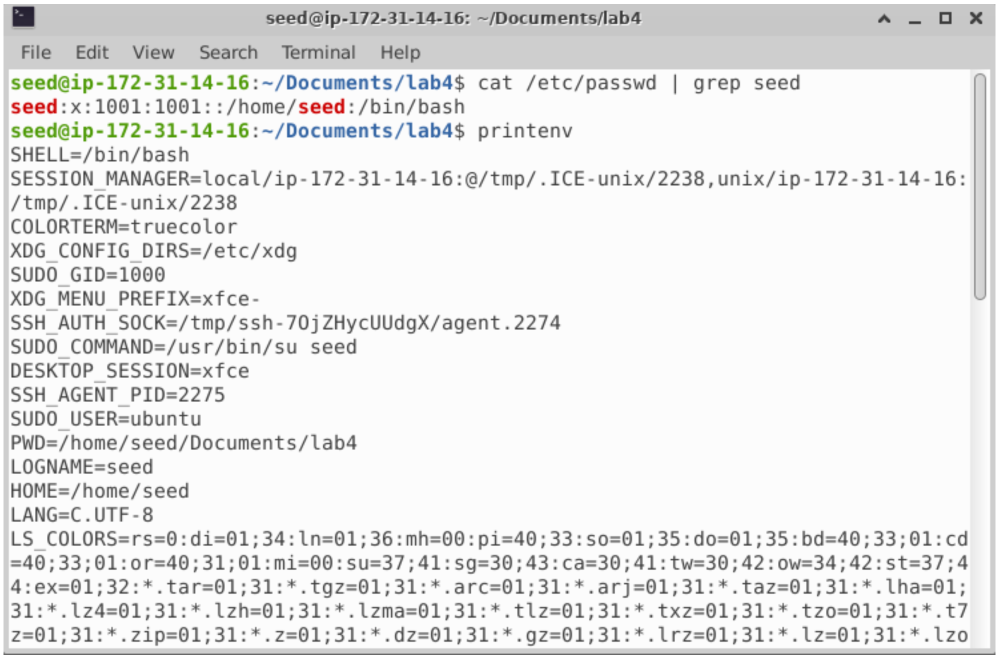
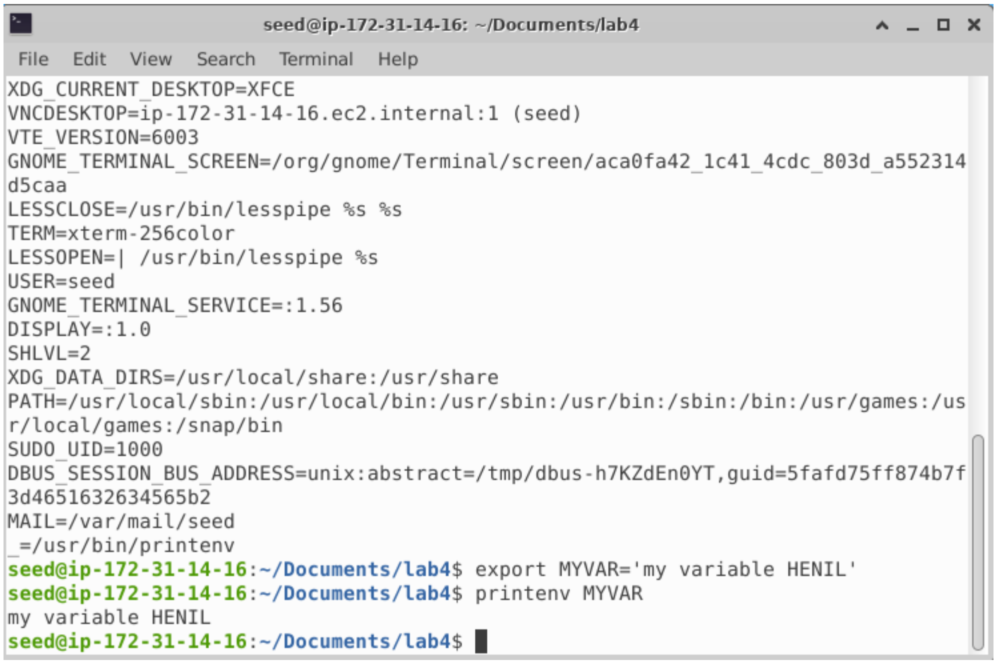
COMPUTER SECURITY : SET UID AND ENVIRONMENT VARIABLES LAB : LAB-04

TASK 1:

In order to print all environment variables, I type printenv. The output shows that environment Next, I type in printenv PWD, which returns only the value of the variable PWD. In order to find out all the variables that consist of a substring PWD, I use env | grep PWD, which gets all the variables and values that contains PWD as a substring within them. The output is in the format of variable = value. The unset command helps to delete a particular environment variable, as is seen in the output. Once we unset PWD and then try to find it using env command, it returns nothing because there is no variable PWD. Using the export command, we can set the environment variable and value, as seen in the output. This command can be used to create or edit a particular environment variable. The output shows a demo of these commands: variables are just variable = value pairs. Entering env would give a similar output:


seed@ip-172-31-14-16: ~/Documents/lab4
File Edit View Search Terminal Help

```
seed@ip-172-31-14-16:~/Documents/lab4$ cat /etc/passwd | grep seed
seed:x:1001:1001::/home/seed:/bin/bash
seed@ip-172-31-14-16:~/Documents/lab4$ printenv
SHELL=/bin/bash
SESSION_MANAGER=local/ip-172-31-14-16:@/tmp/.ICE-unix/2238,unix/ip-172-31-14-16:
/tmp/.ICE-unix/2238
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg
SUDO_GID=1000
XDG_MENU_PREFIX=xfce-
SSH_AUTH_SOCK=/tmp/ssh-7OjZHycUUdgX/agent.2274
SUDO_COMMAND=/usr/bin/su seed
DESKTOP_SESSION=xfce
SSH_AGENT_PID=2275
SUDO_USER=ubuntu
PWD=/home/seed/Documents/lab4
LOGNAME=seed
HOME=/home/seed
LANG=C.UTF-8
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;4
4: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:*.t7
z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo
```

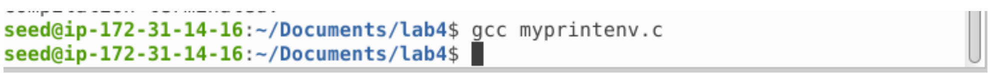

seed@ip-172-31-14-16: ~/Documents/lab4
File Edit View Search Terminal Help

```
XDG_CURRENT_DESKTOP=XFCE
VNCDESKTOP=ip-172-31-14-16.ec2.internal:1 (seed)
VTE_VERSION=6003
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/aca0fa42_1c41_4cdc_803d_a552314
d5caa
LESSCLOSE=/usr/bin/lesspipe %s %s
TERM=xterm-256color
LESSOPEN=| /usr/bin/lesspipe %s
USER=seed
GNOME_TERMINAL_SERVICE=:1.56
DISPLAY=:1.0
SHLVL=2
XDG_DATA_DIRS=/usr/local/share:/usr/share
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/us
r/local/games:/snap/bin
SUDO_UID=1000
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-h7KZdEn0YT,guid=5fafd75ff874b7f
3d4651632634565b2
MAIL=/var/mail/seed
_=/usr/bin/printenv
seed@ip-172-31-14-16:~/Documents/lab4$ export MYVAR='my variable HENIL'
seed@ip-172-31-14-16:~/Documents/lab4$ printenv MYVAR
my variable HENIL
seed@ip-172-31-14-16:~/Documents/lab4$ 
```

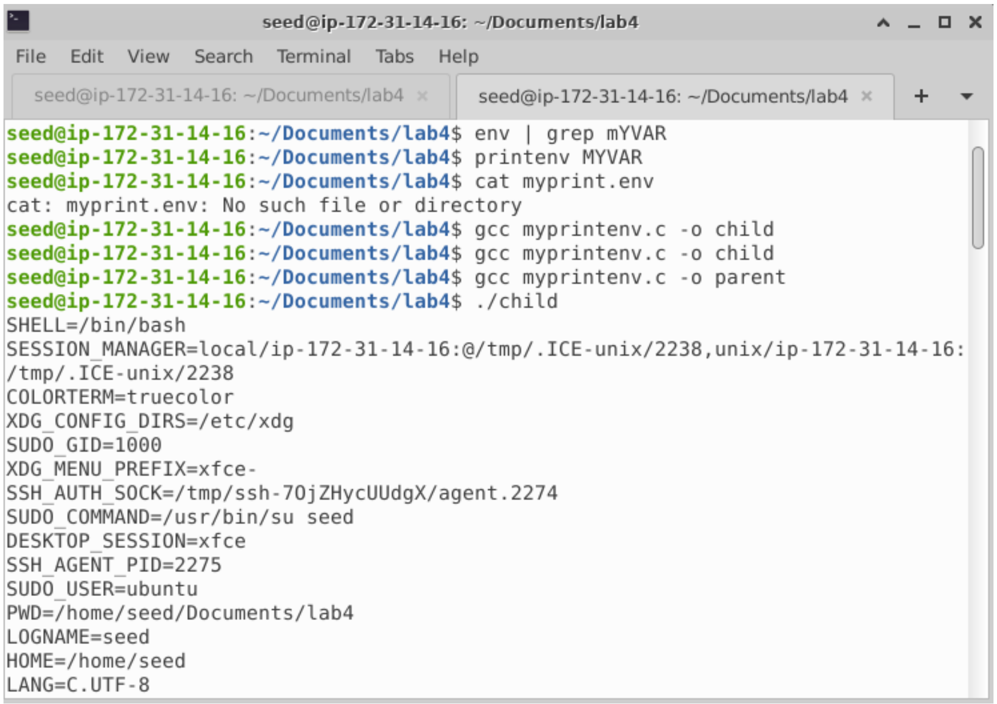
TASK 2:

The content of the output of parent – child prog containing child process with print env is stored in file named output

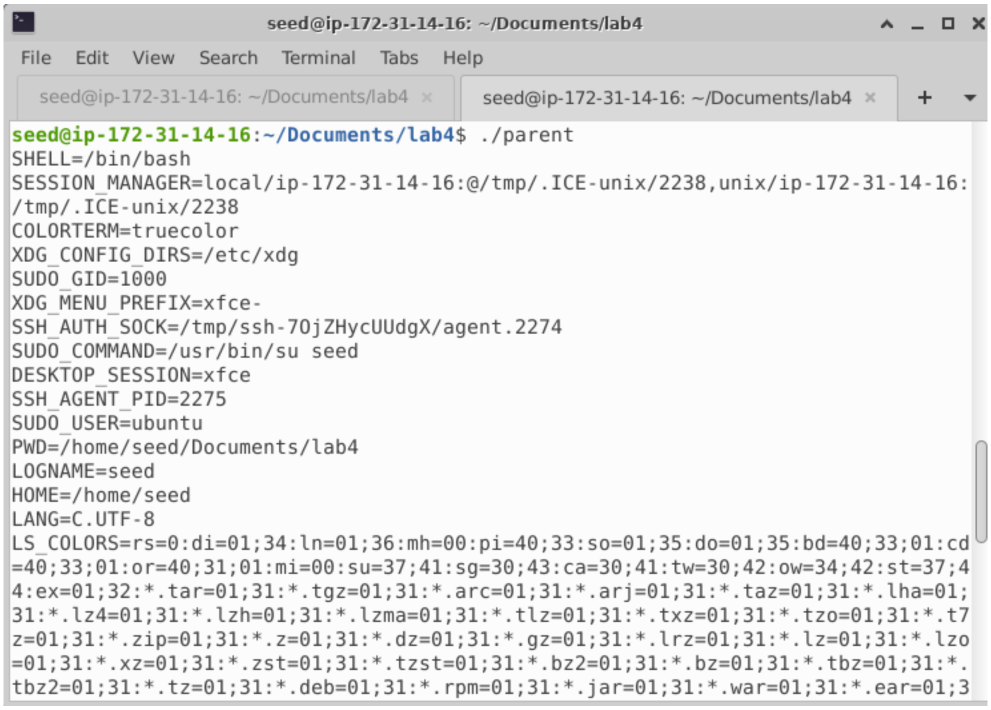
file. It displays all the environment variables of the child process.



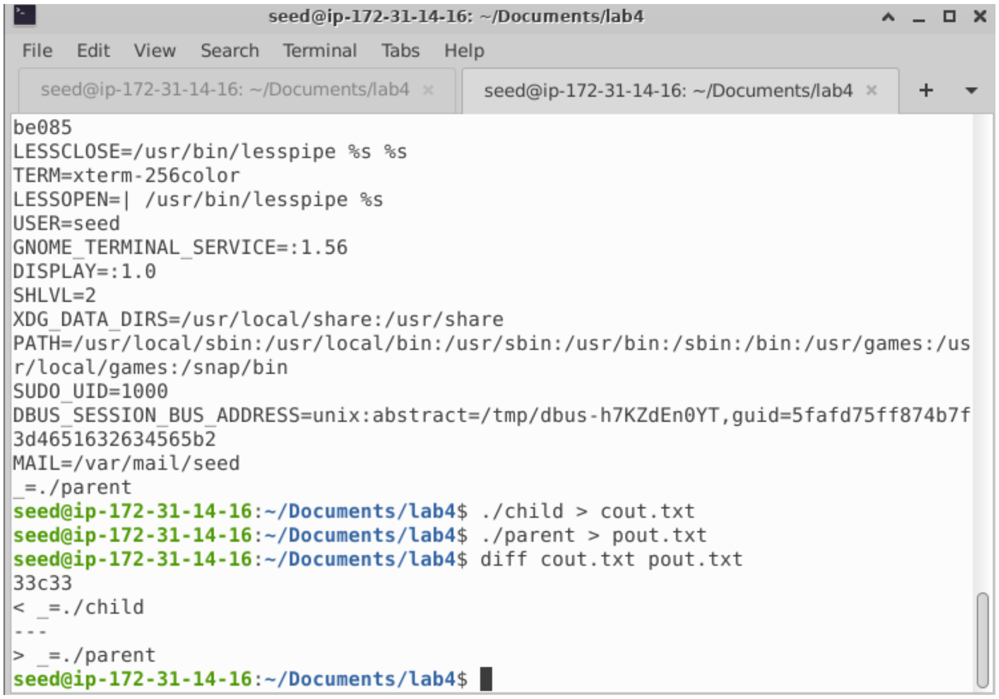
```
seed@ip-172-31-14-16:~/Documents/lab4$ gcc myprintenv.c
seed@ip-172-31-14-16:~/Documents/lab4$ 
```


seed@ip-172-31-14-16: ~/Documents/lab4
File Edit View Search Terminal Tabs Help
seed@ip-172-31-14-16: ~/Documents/lab4
seed@ip-172-31-14-16: ~/Documents/lab4

```
seed@ip-172-31-14-16:~/Documents/lab4$ env | grep mYVAR
seed@ip-172-31-14-16:~/Documents/lab4$ printenv MYVAR
seed@ip-172-31-14-16:~/Documents/lab4$ cat myprint.env
cat: myprint.env: No such file or directory
seed@ip-172-31-14-16:~/Documents/lab4$ gcc myprintenv.c -o child
seed@ip-172-31-14-16:~/Documents/lab4$ gcc myprintenv.c -o child
seed@ip-172-31-14-16:~/Documents/lab4$ gcc myprintenv.c -o parent
seed@ip-172-31-14-16:~/Documents/lab4$ ./child
SHELL=/bin/bash
SESSION_MANAGER=local/ip-172-31-14-16:@/tmp/.ICE-unix/2238,unix/ip-172-31-14-16:
/tmp/.ICE-unix/2238
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg
SUDO_GID=1000
XDG_MENU_PREFIX=xfce-
SSH_AUTH_SOCK=/tmp/ssh-70jZHycUUdgX/agent.2274
SUDO_COMMAND=/usr/bin/su seed
DESKTOP_SESSION=xfce
SSH_AGENT_PID=2275
SUDO_USER=ubuntu
PWD=/home/seed/Documents/lab4
LOGNAME=seed
HOME=/home/seed
LANG=C.UTF-8
```


seed@ip-172-31-14-16: ~/Documents/lab4
File Edit View Search Terminal Tabs Help
seed@ip-172-31-14-16: ~/Documents/lab4
seed@ip-172-31-14-16: ~/Documents/lab4

```
seed@ip-172-31-14-16:~/Documents/lab4$ ./parent
SHELL=/bin/bash
SESSION_MANAGER=local/ip-172-31-14-16:@/tmp/.ICE-unix/2238,unix/ip-172-31-14-16:
/tmp/.ICE-unix/2238
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg
SUDO_GID=1000
XDG_MENU_PREFIX=xfce-
SSH_AUTH_SOCK=/tmp/ssh-70jZHycUUdgX/agent.2274
SUDO_COMMAND=/usr/bin/su seed
DESKTOP_SESSION=xfce
SSH_AGENT_PID=2275
SUDO_USER=ubuntu
PWD=/home/seed/Documents/lab4
LOGNAME=seed
HOME=/home/seed
LANG=C.UTF-8
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;4
4: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:*.t7
z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo
=01;31:*.xz=01;31:*.zst=01;31:*.tzst=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;3
```

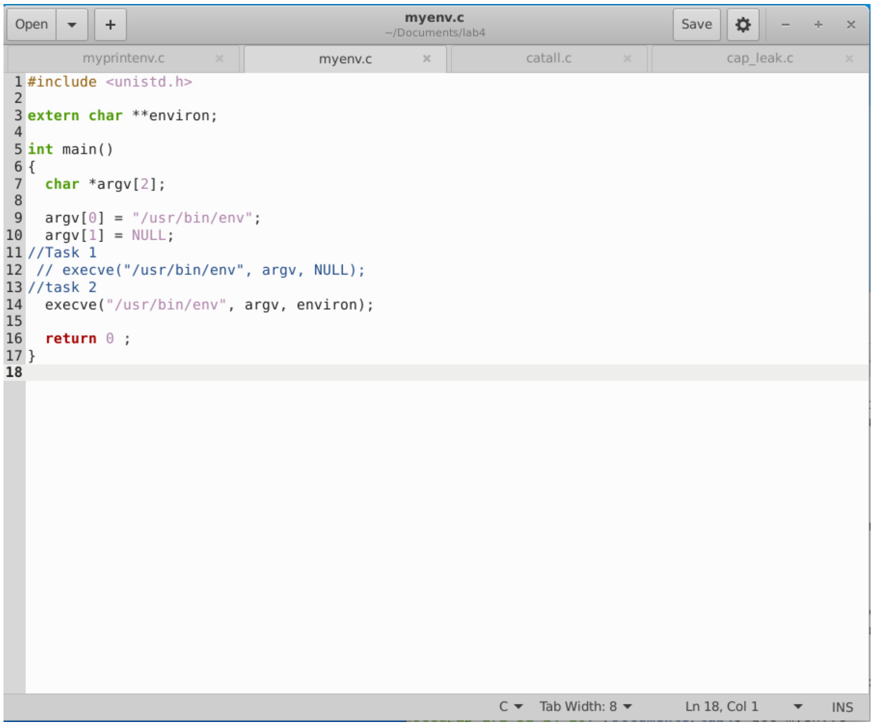

seed@ip-172-31-14-16: ~/Documents/lab4
File Edit View Search Terminal Tabs Help
seed@ip-172-31-14-16: ~/Documents/lab4
seed@ip-172-31-14-16: ~/Documents/lab4

```
be085
LESSCLOSE=/usr/bin/lesspipe %s %s
TERM=xterm-256color
LESSOPEN=| /usr/bin/lesspipe %s
USER=seed
GNOME_TERMINAL_SERVICE=:1.56
DISPLAY=:1.0
SHLVL=2
XDG_DATA_DIRS=/usr/local/share:/usr/share
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/us
r/local/games:/snap/bin
SUDO_UID=1000
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-h7KZdEn0YT,guid=5fafd75ff874b7f
3d4651632634565b2
MAIL=/var/mail/seed
_=./parent
seed@ip-172-31-14-16:~/Documents/lab4$ ./child > cout.txt
seed@ip-172-31-14-16:~/Documents/lab4$ ./parent > pout.txt
seed@ip-172-31-14-16:~/Documents/lab4$ diff cout.txt pout.txt
33c33
< _=./child
---
> _=./parent
seed@ip-172-31-14-16:~/Documents/lab4$ 
```

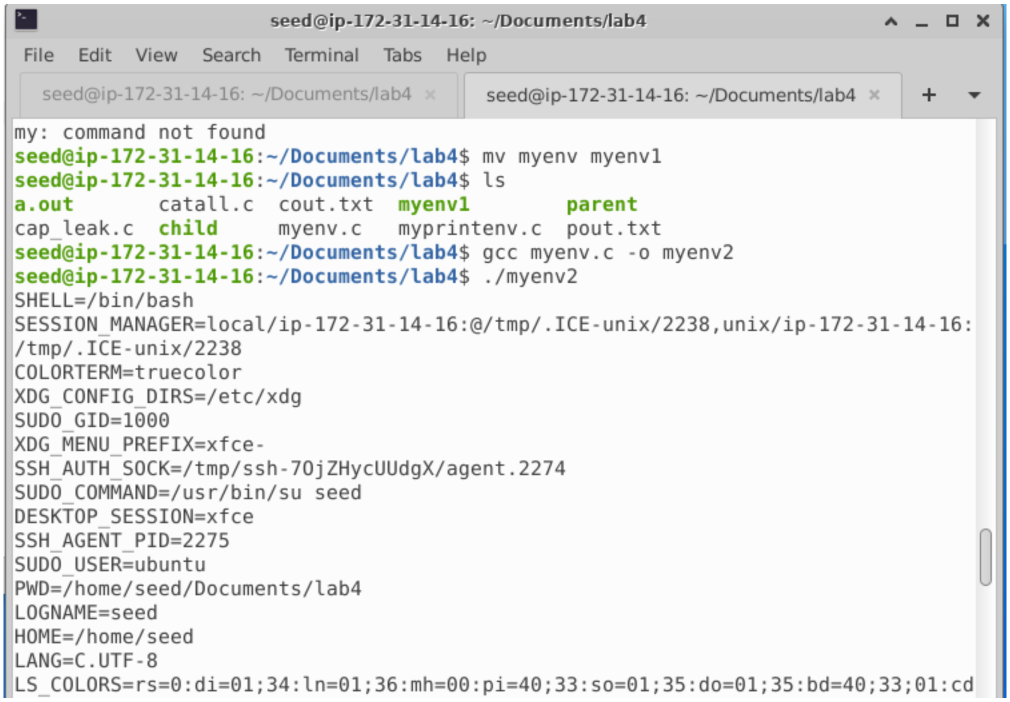
This shows that the _ environment variable takes on the value of the last command executed, here the command of program execution. It is considered a special shell variable and contains different values depending on the scenario.
This shows that the _ environment variable changed depending on the compiled program being run but other than that there is no change in the environment variables. If both the programs were compiled into a file with the same name, there would not be any difference between the output of the parent and child process.

TASK 3:

Here, as seen, the Task 3 program is compiled and executed into respective output files and the output is stored in before its output (with NULL as the argument) and after its output (with environ as the argument).

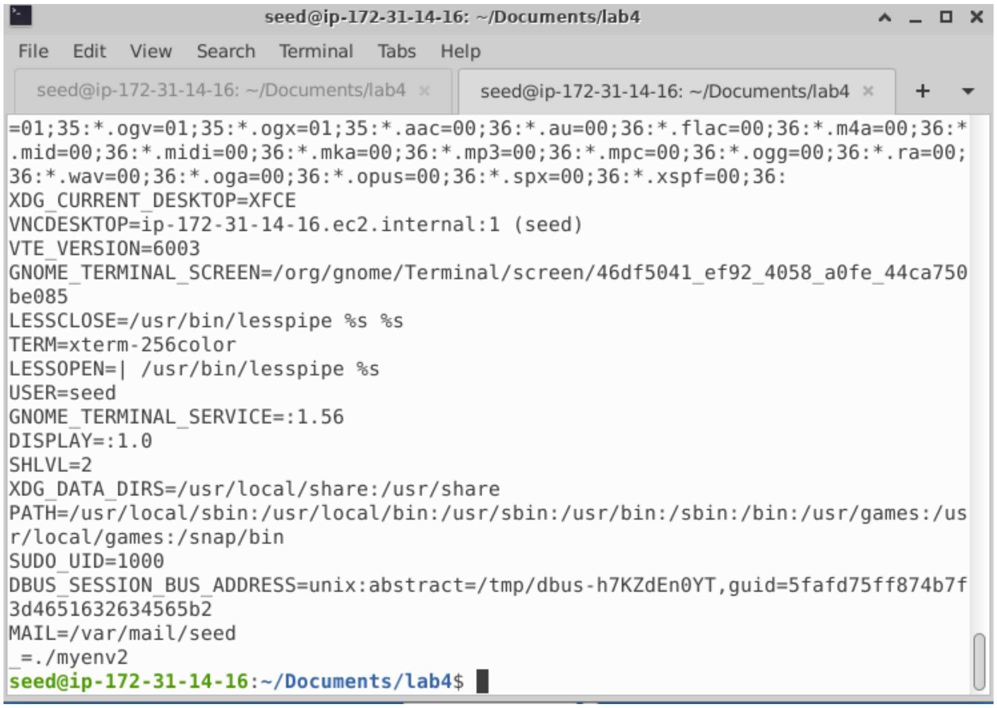


```
#include <unistd.h>

extern char **environ;

int main()
{
  char *argv[2];

  argv[0] = "/usr/bin/env";
  argv[1] = NULL;
//Task 1
 // execve("/usr/bin/env", argv, NULL);
//task 2
  execve("/usr/bin/env", argv, environ);

  return 0 ;
}
```


seed@ip-172-31-14-16: ~/Documents/lab4
File Edit View Search Terminal Tabs Help
seed@ip-172-31-14-16: ~/Documents/lab4
seed@ip-172-31-14-16: ~/Documents/lab4

```
my: command not found
seed@ip-172-31-14-16:~/Documents/lab4$ mv myenv myenv1
seed@ip-172-31-14-16:~/Documents/lab4$ ls
a.out        catall.c  cout.txt  myenv1        parent
cap_leak.c   child     myenv.c   myprintenv.c  pout.txt
seed@ip-172-31-14-16:~/Documents/lab4$ gcc myenv.c -o myenv2
seed@ip-172-31-14-16:~/Documents/lab4$ ./myenv2
SHELL=/bin/bash
SESSION_MANAGER=local/ip-172-31-14-16:@/tmp/.ICE-unix/2238,unix/ip-172-31-14-16:
/tmp/.ICE-unix/2238
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg
SUDO_GID=1000
XDG_MENU_PREFIX=xfce-
SSH_AUTH_SOCK=/tmp/ssh-70jZHycUUdgX/agent.2274
SUDO_COMMAND=/usr/bin/su seed
DESKTOP_SESSION=xfce
SSH_AGENT_PID=2275
SUDO_USER=ubuntu
PWD=/home/seed/Documents/lab4
LOGNAME=seed
HOME=/home/seed
LANG=C.UTF-8
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
```

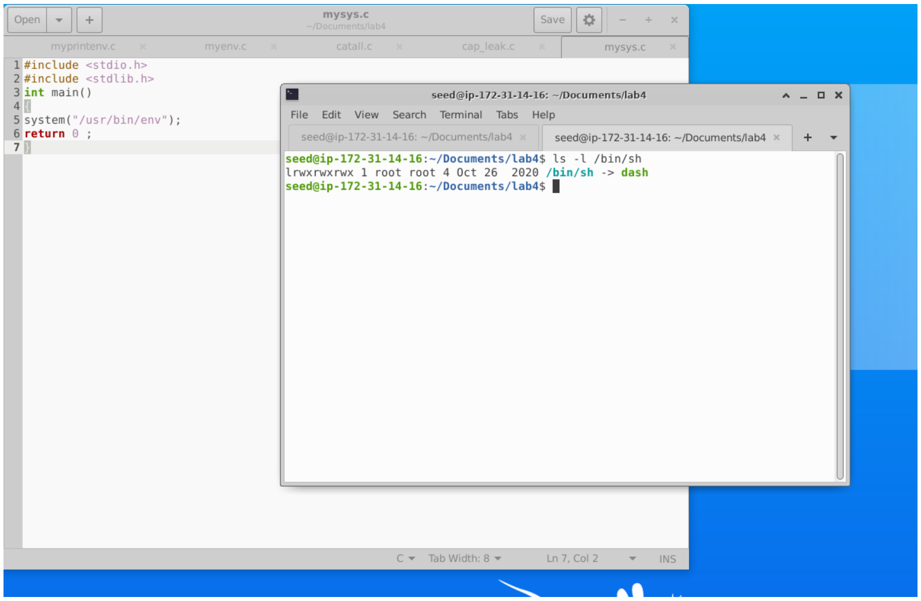
The explanation for this is that even though the global environ variable was specified in the program, the beforeedit program contained NULL as the third argument of the execve and the afteredit program contained environ variable as the third argument of the execve. This change affected the output of the program because the third argument to execve() function specifies the environment variable of the current process. Since the environ variable was not passed in the initial program and hence no environment variables were associated with this new process, the output was null. But after editing the program, we passed the environ variable as the third argument to execve, which contained all the environment variables of the current process, the output of the program had all the environment variables, as expected. In conclusion, the third argument of the execve() command gets the program its environment variables.


seed@ip-172-31-14-16: ~/Documents/lab4
File Edit View Search Terminal Tabs Help
seed@ip-172-31-14-16: ~/Documents/lab4
seed@ip-172-31-14-16: ~/Documents/lab4

```
=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:
XDG_CURRENT_DESKTOP=XFCE
VNCDESKTOP=ip-172-31-14-16.ec2.internal:1 (seed)
VTE_VERSION=6003
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/46df5041_ef92_4058_a0fe_44ca750
be085
LESSCLOSE=/usr/bin/lesspipe %s %s
TERM=xterm-256color
LESSOPEN=| /usr/bin/lesspipe %s
USER=seed
GNOME_TERMINAL_SERVICE=:1.56
DISPLAY=:1.0
SHLVL=2
XDG_DATA_DIRS=/usr/local/share:/usr/share
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/us
r/local/games:/snap/bin
SUDO_UID=1000
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-h7KZdEn0YT,guid=5fafd75ff874b7f
3d4651632634565b2
MAIL=/var/mail/seed
_=./myenv2
seed@ip-172-31-14-16:~/Documents/lab4$
```

TASK 4:

The program is compiled and executed and as seen, even though we don't explicitly send any environment variables in the program, the output shows the environment variable of the current process. This happens because the system function implicitly passes the environment variables to the called function /bin/sh.


Open
Save
mysys.c
~/Documents/lab4
myprintenv.c
myenv.c
catall.c
cap_leak.c
mysys.c
seed@ip-172-31-14-16: ~/Documents/lab4
File Edit View Search Terminal Tabs Help
seed@ip-172-31-14-16: ~/Documents/lab4
seed@ip-172-31-14-16: ~/Documents/lab4
C
Tab Width: 8
Ln 7, Col 2
INS

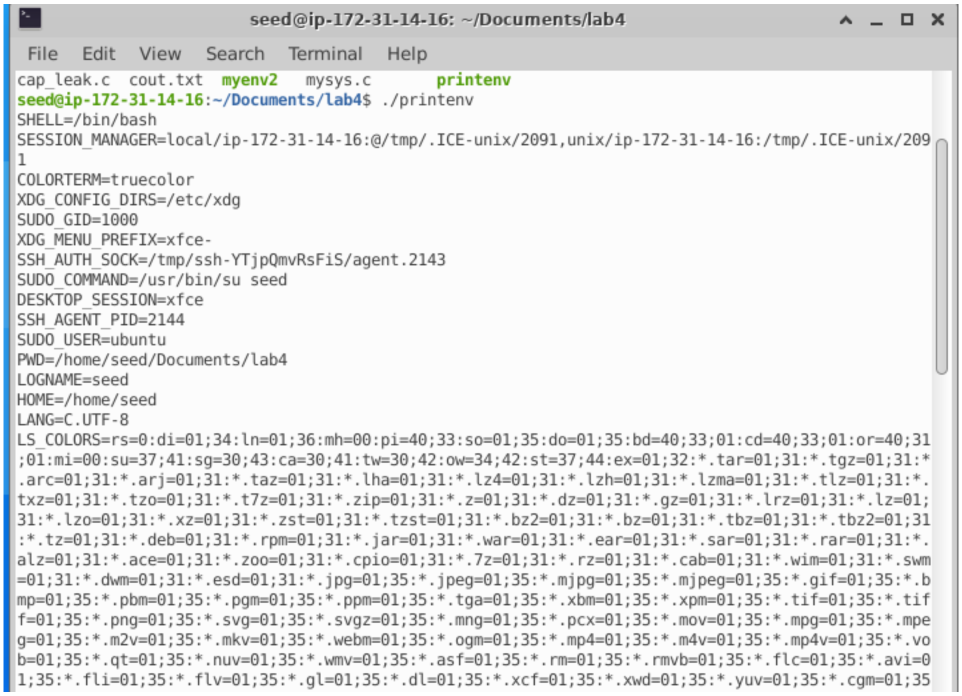
```
#include <stdio.h>
#include <stdlib.h>
int main()
{
system("/usr/bin/env");
return 0 ;
}
```

```
seed@ip-172-31-14-16:~/Documents/lab4$ ls -l /bin/sh
lrwxrwxrwx 1 root root 4 Oct 26  2020 /bin/sh -> dash
seed@ip-172-31-14-16:~/Documents/lab4$
```

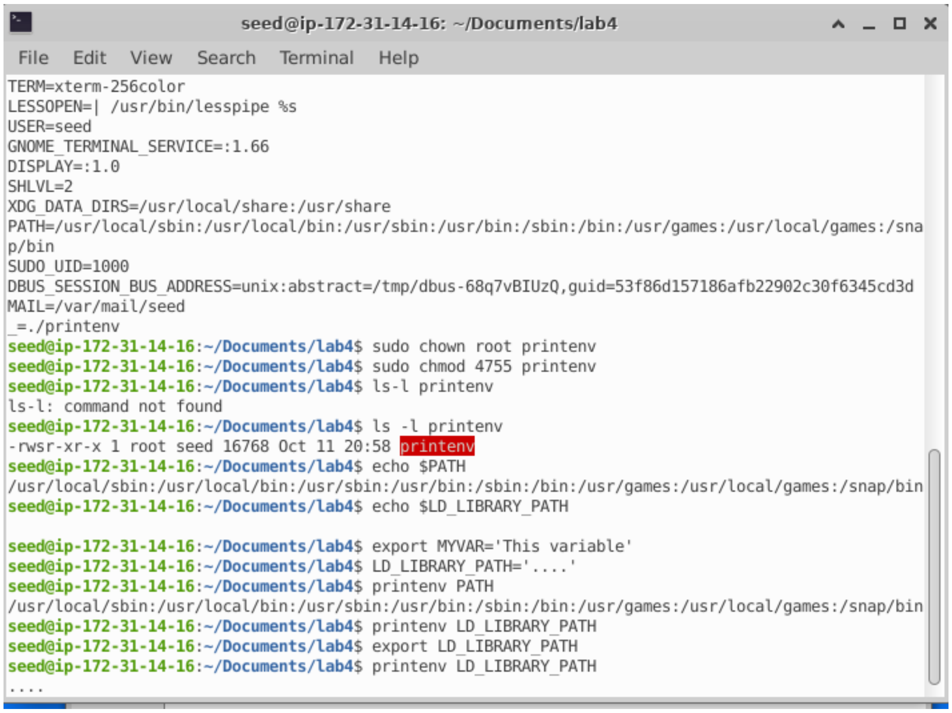
TASK 5:

After compiling the given program, we change the ownership and permission of the file using the following commands:
sudo chown root filename (making the root as the owner of filename)
sudo chmod 4755 filename (making the program a SET-UID program by setting set-uid bit)

This makes the program a SET-UID root program. Then on looking for the environment variables, since PATH and LD_LIBRARY_PATH are already present, I only initialize a new variable with name and value /home/seed using export command and allow the other environment values to be the same. The following screenshot shows the performed steps:

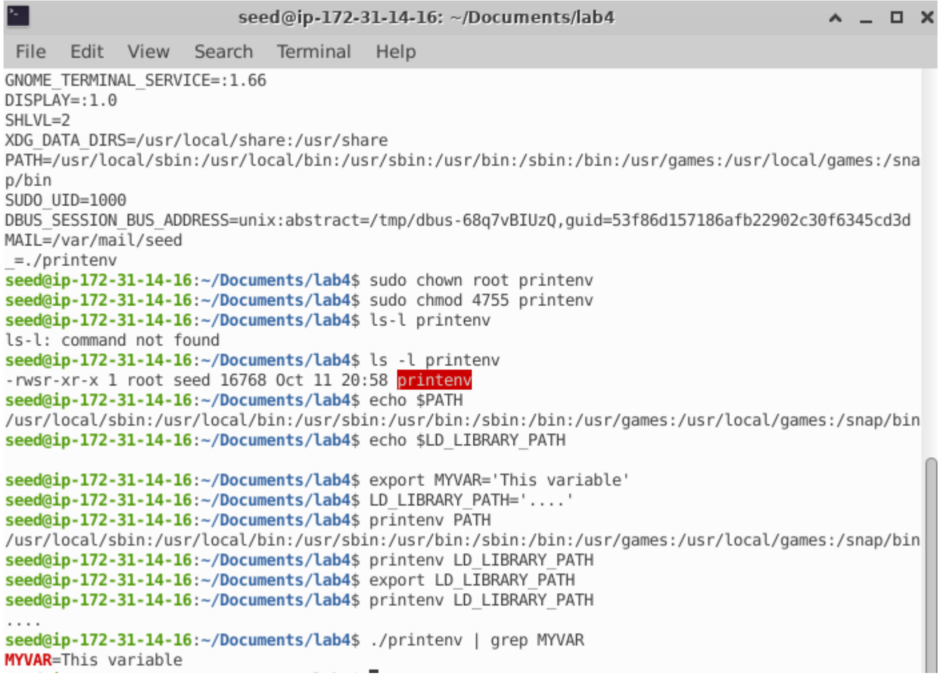

seed@ip-172-31-14-16: ~/Documents/lab4
File Edit View Search Terminal Help

```
cap_leak.c  cout.txt  myenv2   mysys.c       printenv
seed@ip-172-31-14-16:~/Documents/lab4$ ./printenv
SHELL=/bin/bash
SESSION_MANAGER=local/ip-172-31-14-16:@/tmp/.ICE-unix/2091,unix/ip-172-31-14-16:/tmp/.ICE-unix/209
1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg
SUDO_GID=1000
XDG_MENU_PREFIX=xfce-
SSH_AUTH_SOCK=/tmp/ssh-YTjpQmvRsFiS/agent.2143
SUDO_COMMAND=/usr/bin/su seed
DESKTOP_SESSION=xfce
SSH_AGENT_PID=2144
SUDO_USER=ubuntu
PWD=/home/seed/Documents/lab4
LOGNAME=seed
HOME=/home/seed
LANG=C.UTF-8
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:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;
31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=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:*.wim=01;31:*.swm
=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.b
mp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tif
f=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpe
g=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vo
b=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=0
1;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
```


seed@ip-172-31-14-16: ~/Documents/lab4
File Edit View Search Terminal Help

```
TERM=xterm-256color
LESSOPEN=| /usr/bin/lesspipe %s
USER=seed
GNOME_TERMINAL_SERVICE=:1.66
DISPLAY=:1.0
SHLVL=2
XDG_DATA_DIRS=/usr/local/share:/usr/share
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/sna
p/bin
SUDO_UID=1000
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-68q7vBIUzQ,guid=53f86d157186afb22902c30f6345cd3d
MAIL=/var/mail/seed
_=./printenv
seed@ip-172-31-14-16:~/Documents/lab4$ sudo chown root printenv
seed@ip-172-31-14-16:~/Documents/lab4$ sudo chmod 4755 printenv
seed@ip-172-31-14-16:~/Documents/lab4$ ls-l printenv
ls-l: command not found
seed@ip-172-31-14-16:~/Documents/lab4$ ls -l printenv
-rwsr-xr-x 1 root seed 16768 Oct 11 20:58 printenv
seed@ip-172-31-14-16:~/Documents/lab4$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
seed@ip-172-31-14-16:~/Documents/lab4$ echo $LD_LIBRARY_PATH

seed@ip-172-31-14-16:~/Documents/lab4$ export MYVAR='This variable'
seed@ip-172-31-14-16:~/Documents/lab4$ LD_LIBRARY_PATH='....'
seed@ip-172-31-14-16:~/Documents/lab4$ printenv PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
seed@ip-172-31-14-16:~/Documents/lab4$ printenv LD_LIBRARY_PATH
seed@ip-172-31-14-16:~/Documents/lab4$ export LD_LIBRARY_PATH
seed@ip-172-31-14-16:~/Documents/lab4$ printenv LD_LIBRARY_PATH
....
```

This shows that the SET-UID program’s child process may not inherit all the environment variables of the parent process, LD_LIBRARY_PATH being one of them over here. This is a security mechanism implemented by the dynamic linker. The LD_LIBRARY_PATH is ignored here because the real user id and effective user id is different. That is why only the other two environment variables are seen in the output.

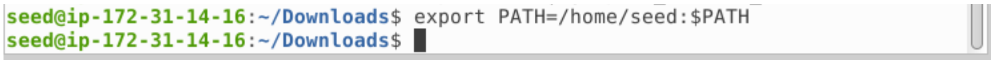


seed@ip-172-31-14-16: ~/Documents/lab4
File Edit View Search Terminal Help

```
GNOME_TERMINAL_SERVICE=:1.66
DISPLAY=:1.0
SHLVL=2
XDG_DATA_DIRS=/usr/local/share:/usr/share
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/sna
p/bin
SUDO_UID=1000
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-68q7vBIUzQ,guid=53f86d157186afb22902c30f6345cd3d
MAIL=/var/mail/seed
_=./printenv
seed@ip-172-31-14-16:~/Documents/lab4$ sudo chown root printenv
seed@ip-172-31-14-16:~/Documents/lab4$ sudo chmod 4755 printenv
seed@ip-172-31-14-16:~/Documents/lab4$ ls-l printenv
ls-l: command not found
seed@ip-172-31-14-16:~/Documents/lab4$ ls -l printenv
-rwsr-xr-x 1 root seed 16768 Oct 11 20:58 printenv
seed@ip-172-31-14-16:~/Documents/lab4$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
seed@ip-172-31-14-16:~/Documents/lab4$ echo $LD_LIBRARY_PATH

seed@ip-172-31-14-16:~/Documents/lab4$ export MYVAR='This variable'
seed@ip-172-31-14-16:~/Documents/lab4$ LD_LIBRARY_PATH='....'
seed@ip-172-31-14-16:~/Documents/lab4$ printenv PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
seed@ip-172-31-14-16:~/Documents/lab4$ printenv LD_LIBRARY_PATH
seed@ip-172-31-14-16:~/Documents/lab4$ export LD_LIBRARY_PATH
seed@ip-172-31-14-16:~/Documents/lab4$ printenv LD_LIBRARY_PATH
....
seed@ip-172-31-14-16:~/Documents/lab4$ ./printenv | grep MYVAR
MYVAR=This variable
```

TASK 6:

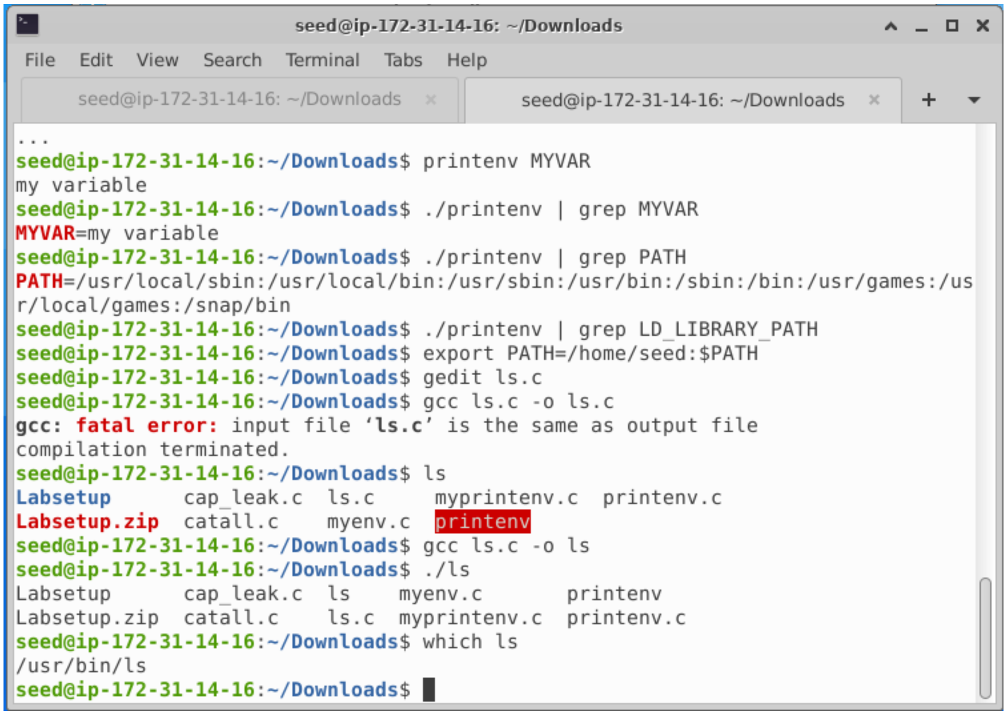
Export Path as HOME/SEED:



```
seed@ip-172-31-14-16:~/Downloads$ export PATH=/home/seed:$PATH
seed@ip-172-31-14-16:~/Downloads$ 
```

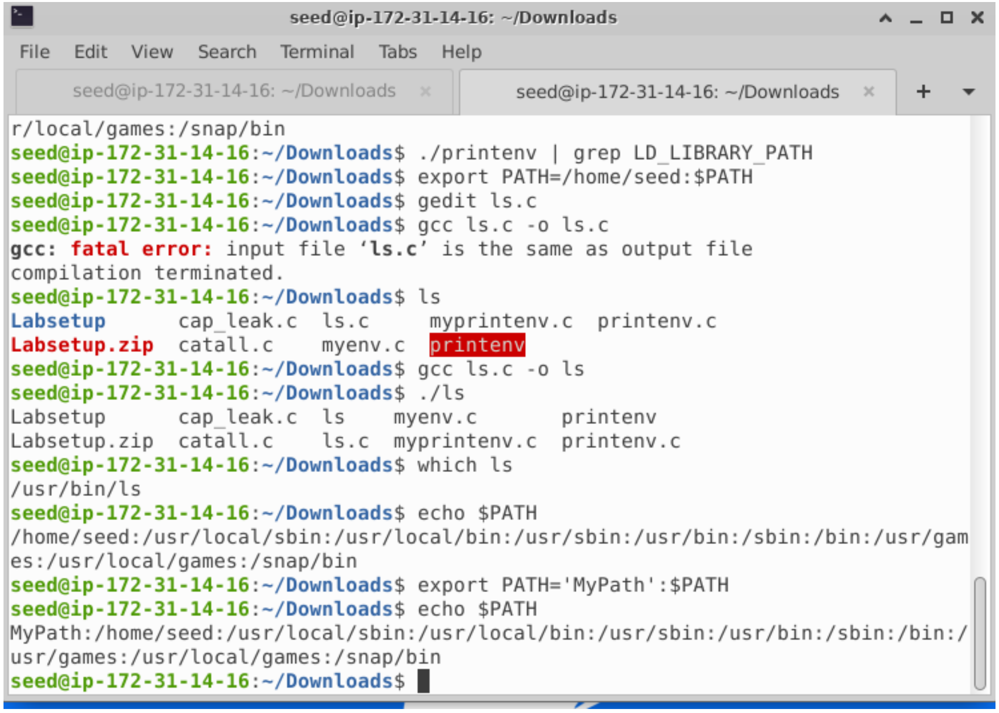
We change its owner to root and make it a Set-UID program.
Get the Set-UID program to run your own malicious code, instead of /bin/ls and finally make it run with root priviliges.

First we check using which ls where it is running and then :


seed@ip-172-31-14-16: ~/Downloads
File Edit View Search Terminal Tabs Help
seed@ip-172-31-14-16: ~/Downloads
seed@ip-172-31-14-16: ~/Downloads

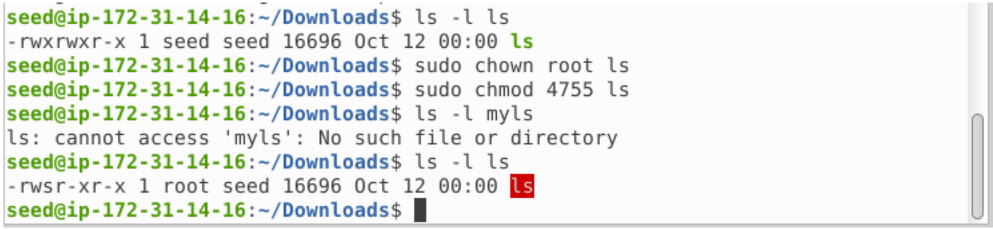
```
...
seed@ip-172-31-14-16:~/Downloads$ printenv MYVAR
my variable
seed@ip-172-31-14-16:~/Downloads$ ./printenv | grep MYVAR
MYVAR=my variable
seed@ip-172-31-14-16:~/Downloads$ ./printenv | grep PATH
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/us
r/local/games:/snap/bin
seed@ip-172-31-14-16:~/Downloads$ ./printenv | grep LD_LIBRARY_PATH
seed@ip-172-31-14-16:~/Downloads$ export PATH=/home/seed:$PATH
seed@ip-172-31-14-16:~/Downloads$ gedit ls.c
seed@ip-172-31-14-16:~/Downloads$ gcc ls.c -o ls.c
gcc: fatal error: input file ‘ls.c’ is the same as output file
compilation terminated.
seed@ip-172-31-14-16:~/Downloads$ ls
Labsetup      cap_leak.c  ls.c     myprintenv.c  printenv.c
Labsetup.zip  catall.c    myenv.c  printenv
seed@ip-172-31-14-16:~/Downloads$ gcc ls.c -o ls
seed@ip-172-31-14-16:~/Downloads$ ./ls
Labsetup      cap_leak.c  ls    myenv.c       printenv
Labsetup.zip  catall.c    ls.c  myprintenv.c  printenv.c
seed@ip-172-31-14-16:~/Downloads$ which ls
/usr/bin/ls
seed@ip-172-31-14-16:~/Downloads$ 
```

We change its owner to root and make it a Set-UID program.
Get the Set-UID program to run your own malicious code, instead of /bin/ls and finally make it run with root priviliges.


seed@ip-172-31-14-16: ~/Downloads
File Edit View Search Terminal Tabs Help
seed@ip-172-31-14-16: ~/Downloads
seed@ip-172-31-14-16: ~/Downloads

```
r/local/games:/snap/bin
seed@ip-172-31-14-16:~/Downloads$ ./printenv | grep LD_LIBRARY_PATH
seed@ip-172-31-14-16:~/Downloads$ export PATH=/home/seed:$PATH
seed@ip-172-31-14-16:~/Downloads$ gedit ls.c
seed@ip-172-31-14-16:~/Downloads$ gcc ls.c -o ls.c
gcc: fatal error: input file ‘ls.c’ is the same as output file
compilation terminated.
seed@ip-172-31-14-16:~/Downloads$ ls
Labsetup      cap_leak.c  ls.c     myprintenv.c  printenv.c
Labsetup.zip  catall.c    myenv.c  printenv
seed@ip-172-31-14-16:~/Downloads$ gcc ls.c -o ls
seed@ip-172-31-14-16:~/Downloads$ ./ls
Labsetup      cap_leak.c  ls    myenv.c       printenv
Labsetup.zip  catall.c    ls.c  myprintenv.c  printenv.c
seed@ip-172-31-14-16:~/Downloads$ which ls
/usr/bin/ls
seed@ip-172-31-14-16:~/Downloads$ echo $PATH
/home/seed:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/gam
es:/usr/local/games:/snap/bin
seed@ip-172-31-14-16:~/Downloads$ export PATH='MyPath':$PATH
seed@ip-172-31-14-16:~/Downloads$ echo $PATH
MyPath:/home/seed:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/
usr/games:/usr/local/games:/snap/bin
seed@ip-172-31-14-16:~/Downloads$ 
```

We see it is running from root

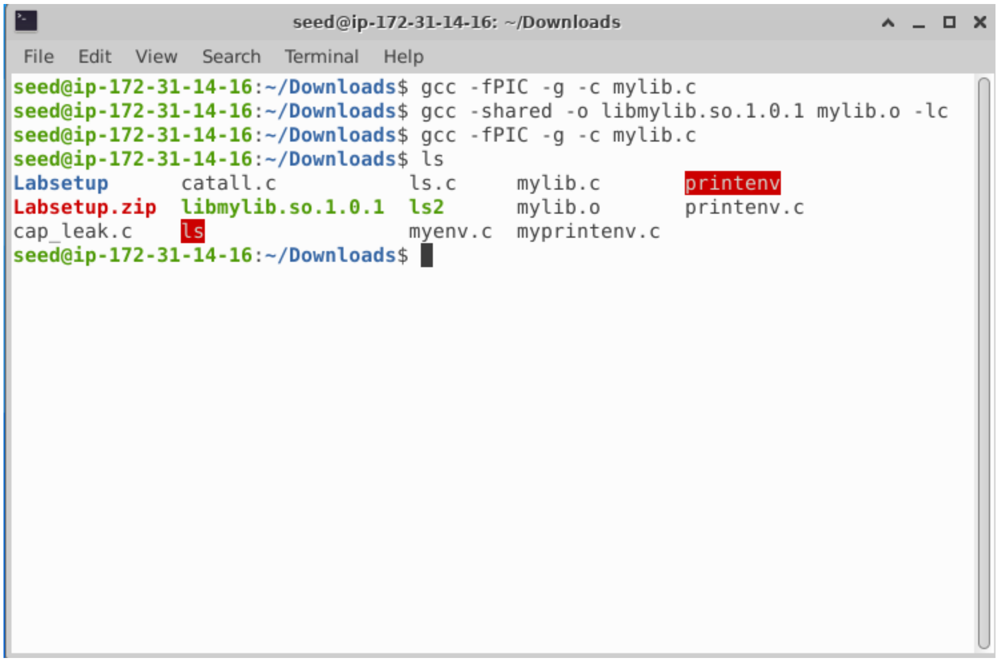


```
seed@ip-172-31-14-16:~/Downloads$ ls -l ls
-rwxrwxr-x 1 seed seed 16696 Oct 12 00:00 ls
seed@ip-172-31-14-16:~/Downloads$ sudo chown root ls
seed@ip-172-31-14-16:~/Downloads$ sudo chmod 4755 ls
seed@ip-172-31-14-16:~/Downloads$ ls -l myls
ls: cannot access 'myls': No such file or directory
seed@ip-172-31-14-16:~/Downloads$ ls -l ls
-rwsr-xr-x 1 root seed 16696 Oct 12 00:00 ls
seed@ip-172-31-14-16:~/Downloads$ 
```

TASK 7:

First, I create a program named mylib.c that has the sleep function overriding the system's sleep function as given in the assignment. This function is just printing a statement on the standard output. After this, we compile the program using the following command:

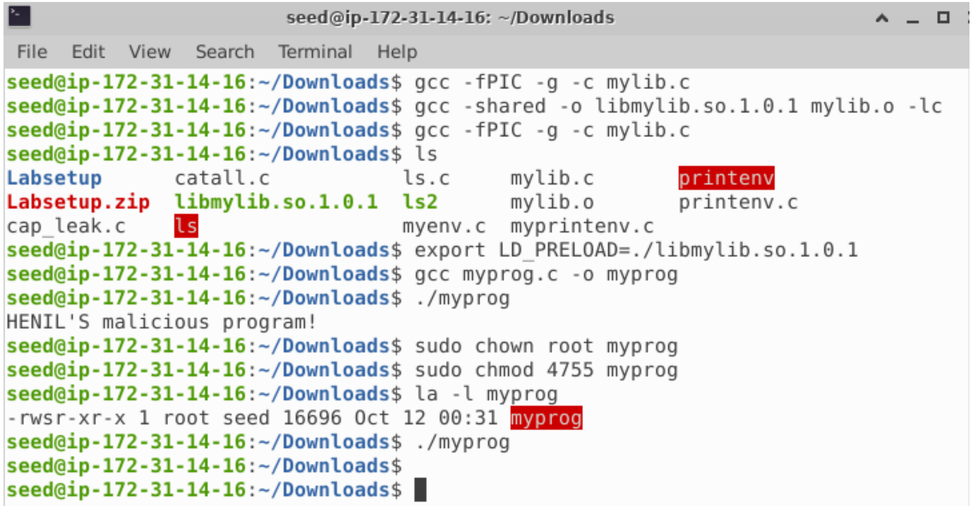
gcc -fPIC -g -c mylib.c (where -fPIC means that emit position-independent code, suitable for dynamic linking



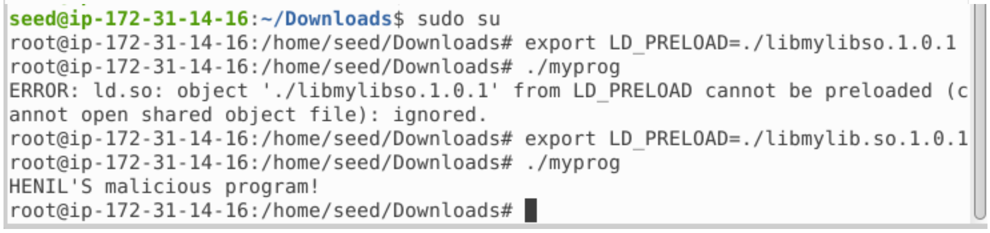
seed@ip-172-31-14-16: ~/Downloads
File Edit View Search Terminal Help

```
seed@ip-172-31-14-16:~/Downloads$ gcc -fPIC -g -c mylib.c
seed@ip-172-31-14-16:~/Downloads$ gcc -shared -o libmylib.so.1.0.1 mylib.o -lc
seed@ip-172-31-14-16:~/Downloads$ gcc -fPIC -g -c mylib.c
seed@ip-172-31-14-16:~/Downloads$ ls
Labsetup      catall.c           ls.c     mylib.c       printenv
Labsetup.zip  libmylib.so.1.0.1  ls2      mylib.o       printenv.c
cap_leak.c    ls                 myenv.c  myprintenv.c
seed@ip-172-31-14-16:~/Downloads$ 
```

Now, set the LD PRELOAD environment variable and Make my prog a regular program, and run it as a normal user.

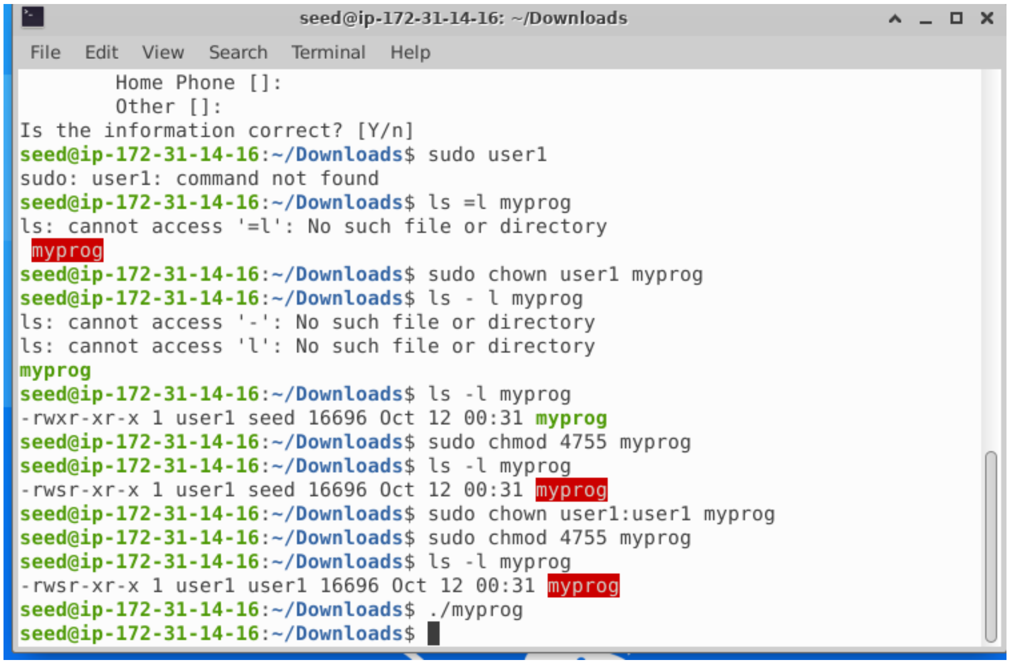

seed@ip-172-31-14-16: ~/Downloads
File Edit View Search Terminal Help

```
seed@ip-172-31-14-16:~/Downloads$ gcc -fPIC -g -c mylib.c
seed@ip-172-31-14-16:~/Downloads$ gcc -shared -o libmylib.so.1.0.1 mylib.o -lc
seed@ip-172-31-14-16:~/Downloads$ gcc -fPIC -g -c mylib.c
seed@ip-172-31-14-16:~/Downloads$ ls
Labsetup      catall.c           ls.c     mylib.c       printenv
Labsetup.zip  libmylib.so.1.0.1  ls2      mylib.o       printenv.c
cap_leak.c    ls                 myenv.c  myprintenv.c
seed@ip-172-31-14-16:~/Downloads$ export LD_PRELOAD=./libmylib.so.1.0.1
seed@ip-172-31-14-16:~/Downloads$ gcc myprog.c -o myprog
seed@ip-172-31-14-16:~/Downloads$ ./myprog
HENIL'S malicious program!
seed@ip-172-31-14-16:~/Downloads$ sudo chown root myprog
seed@ip-172-31-14-16:~/Downloads$ sudo chmod 4755 myprog
seed@ip-172-31-14-16:~/Downloads$ la -l myprog
-rwsr-xr-x 1 root seed 16696 Oct 12 00:31 myprog
seed@ip-172-31-14-16:~/Downloads$ ./myprog
seed@ip-172-31-14-16:~/Downloads$
seed@ip-172-31-14-16:~/Downloads$ 
```



```
seed@ip-172-31-14-16:~/Downloads$ sudo su
root@ip-172-31-14-16:/home/seed/Downloads# export LD_PRELOAD=./libmylibso.1.0.1
root@ip-172-31-14-16:/home/seed/Downloads# ./myprog
ERROR: ld.so: object './libmylibso.1.0.1' from LD_PRELOAD cannot be preloaded (c
annot open shared object file): ignored.
root@ip-172-31-14-16:/home/seed/Downloads# export LD_PRELOAD=./libmylib.so.1.0.1
root@ip-172-31-14-16:/home/seed/Downloads# ./myprog
HENIL'S malicious program!
root@ip-172-31-14-16:/home/seed/Downloads# 
```

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


seed@ip-172-31-14-16: ~/Downloads
File Edit View Search Terminal Help

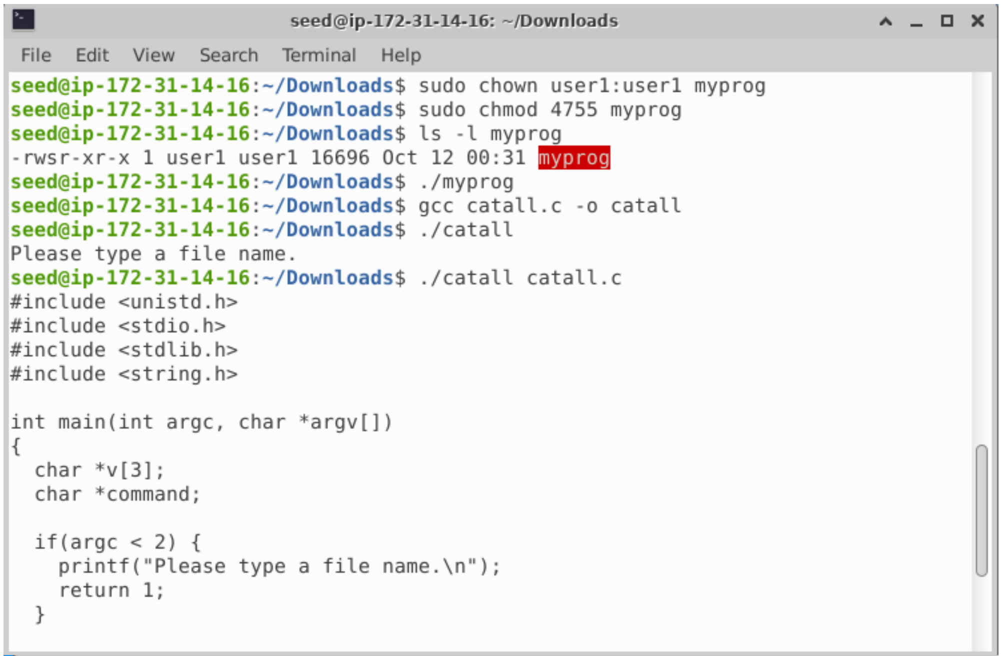
```
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
seed@ip-172-31-14-16:~/Downloads$ sudo user1
sudo: user1: command not found
seed@ip-172-31-14-16:~/Downloads$ ls =l myprog
ls: cannot access '=l': No such file or directory
 myprog
seed@ip-172-31-14-16:~/Downloads$ sudo chown user1 myprog
seed@ip-172-31-14-16:~/Downloads$ ls - l myprog
ls: cannot access '-': No such file or directory
ls: cannot access 'l': No such file or directory
myprog
seed@ip-172-31-14-16:~/Downloads$ ls -l myprog
-rwxr-xr-x 1 user1 seed 16696 Oct 12 00:31 myprog
seed@ip-172-31-14-16:~/Downloads$ sudo chmod 4755 myprog
seed@ip-172-31-14-16:~/Downloads$ ls -l myprog
-rwsr-xr-x 1 user1 seed 16696 Oct 12 00:31 myprog
seed@ip-172-31-14-16:~/Downloads$ sudo chown user1:user1 myprog
seed@ip-172-31-14-16:~/Downloads$ sudo chmod 4755 myprog
seed@ip-172-31-14-16:~/Downloads$ ls -l myprog
-rwsr-xr-x 1 user1 user1 16696 Oct 12 00:31 myprog
seed@ip-172-31-14-16:~/Downloads$ ./myprog
seed@ip-172-31-14-16:~/Downloads$ 
```

Based on these various scenarios we conclude that we can say:

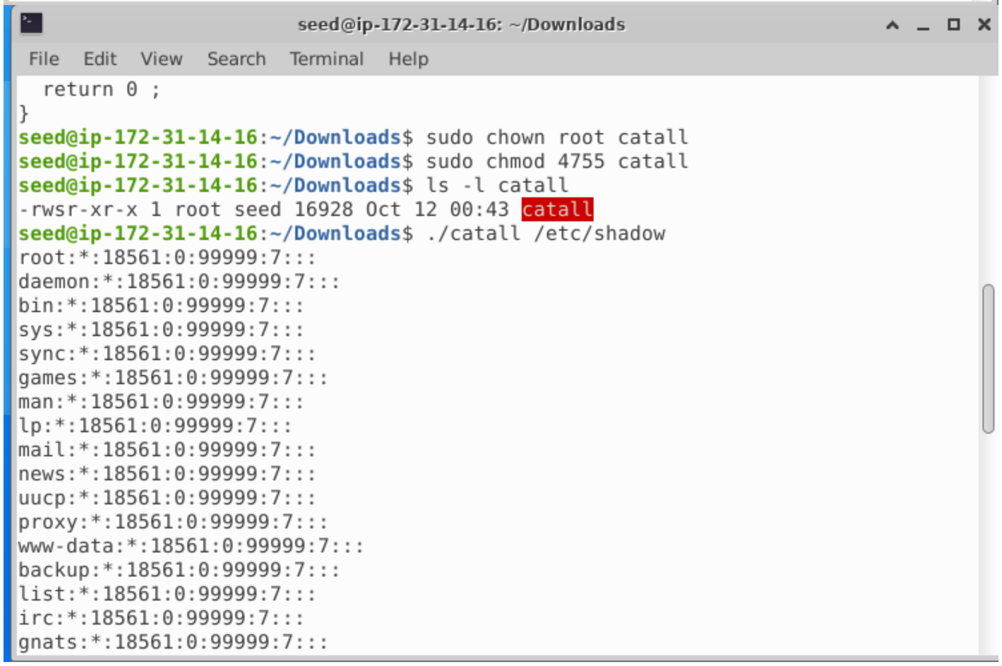
This behavior indicates that the LD_PRELOAD variable is present if the effective and real ID are the same and is dropped if they are different. This is due to the SET-UID program's security mechanism. In the first, third and fourth case, since the owner and the account executing the file were the same, the LD_PRELOAD variable was present everytime and user-defined library was preloaded. Whereas, in the second case, the effective ID was of root and real ID was of seed, the LD_PRELOAD variable was dropped, and system-defined sleep function was called instead.

TASK 8:

Here, first I compile the program provided into a file. Next, this file is converted into a root-owned SET-UID program with executable permission to other users:


seed@ip-172-31-14-16: ~/Downloads
File Edit View Search Terminal Help

```
seed@ip-172-31-14-16:~/Downloads$ sudo chown user1:user1 myprog
seed@ip-172-31-14-16:~/Downloads$ sudo chmod 4755 myprog
seed@ip-172-31-14-16:~/Downloads$ ls -l myprog
-rwsr-xr-x 1 user1 user1 16696 Oct 12 00:31 myprog
seed@ip-172-31-14-16:~/Downloads$ ./myprog
seed@ip-172-31-14-16:~/Downloads$ gcc catall.c -o catall
seed@ip-172-31-14-16:~/Downloads$ ./catall
Please type a file name.
seed@ip-172-31-14-16:~/Downloads$ ./catall catall.c
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(int argc, char *argv[])
{
  char *v[3];
  char *command;

  if(argc < 2) {
    printf("Please type a file name.\n");
    return 1;
  }
```


seed@ip-172-31-14-16: ~/Downloads
File Edit View Search Terminal Help

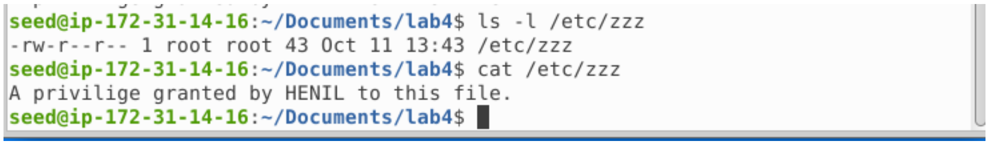
```
  return 0 ;
}
seed@ip-172-31-14-16:~/Downloads$ sudo chown root catall
seed@ip-172-31-14-16:~/Downloads$ sudo chmod 4755 catall
seed@ip-172-31-14-16:~/Downloads$ ls -l catall
-rwsr-xr-x 1 root seed 16928 Oct 12 00:43 catall
seed@ip-172-31-14-16:~/Downloads$ ./catall /etc/shadow
root:*:18561:0:99999:7:::
daemon:*:18561:0:99999:7:::
bin:*:18561:0:99999:7:::
sys:*:18561:0:99999:7:::
sync:*:18561:0:99999:7:::
games:*:18561:0:99999:7:::
man:*:18561:0:99999:7:::
lp:*:18561:0:99999:7:::
mail:*:18561:0:99999:7:::
news:*:18561:0:99999:7:::
uucp:*:18561:0:99999:7:::
proxy:*:18561:0:99999:7:::
www-data:*:18561:0:99999:7:::
backup:*:18561:0:99999:7:::
list:*:18561:0:99999:7:::
irc:*:18561:0:99999:7:::
gnats:*:18561:0:99999:7:::
```

Here, as we can see the program runs normally when we just provide the file to be read. But, if we provide a malicious input such as “document;/bin/sh”, here the program will first read the contents of the document and then run /bin/sh as a command (according to the program.) The /bin/sh allows Bob to run the shell program which has root privileges and bob then runs the rm command to remove a file on which it did not have the write permission. The root terminal is indicated by the #. This shows that even though Bob did not have any permission to write, it could remove a file easily by assuming the privileges of the root user.

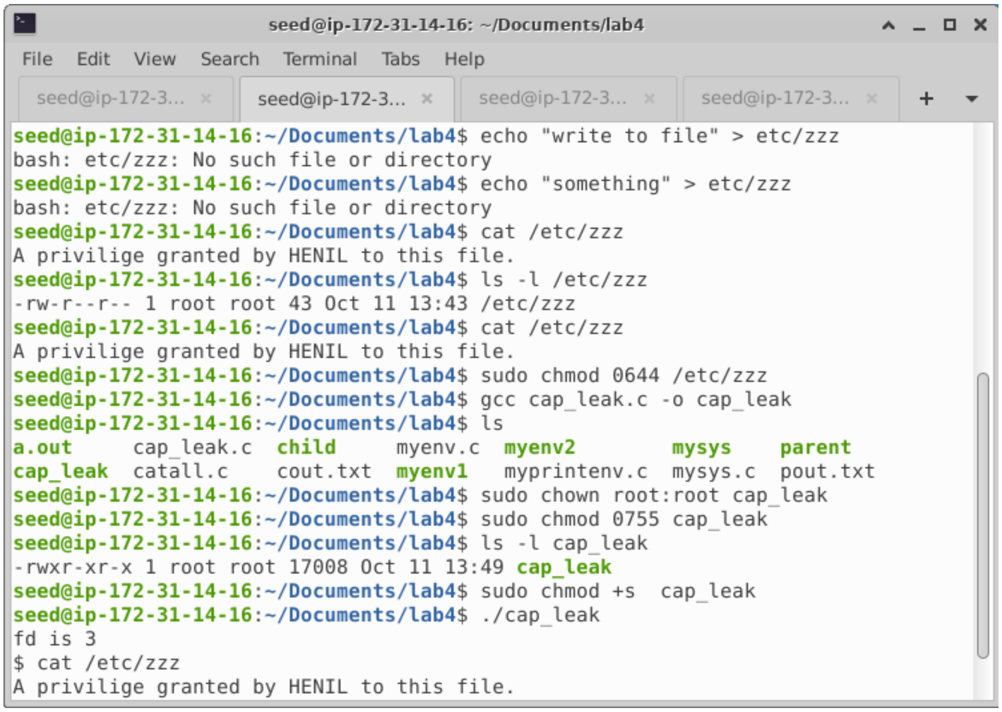
The problem here is the system call inside the program which does not separate the command and user input. The user input is eventually treated as a command instead of data/document name. This can be avoided by segregating the user input and command in the program. Since the system call requires constructing the command using the input, we should avoid using system function in the program and instead use execve function which treats anything inputted from the user as input string and does not allow it to be run as a command. For this, we edit our program and compile it again, making it a root-owned SET-UID program:

This happens because, as seen in the program, the command in system is constructed using strings inputted while executing. In terminal, we can enter multiple commands using ';' and hence the second part after ';' in the input is directly considered as a command rather than a part of the file name. There is no input validation while using system (), but there is some when we use execve. When we use execve, the input is directly entered as the second parameter to the function which in fact is considered as the entire file name and is not appended into a string to construct the command, as before. This avoids this kind of attack.

TASK 9:

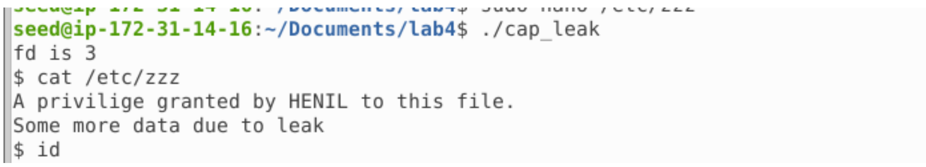


```
seed@ip-172-31-14-16:~/Documents/lab4$ ls -l /etc/zzz
-rw-r--r-- 1 root root 43 Oct 11 13:43 /etc/zzz
seed@ip-172-31-14-16:~/Documents/lab4$ cat /etc/zzz
A privilige granted by HENIL to this file.
seed@ip-172-31-14-16:~/Documents/lab4$ 
```


seed@ip-172-31-14-16: ~/Documents/lab4
File Edit View Search Terminal Tabs Help
seed@ip-172-3...
seed@ip-172-3...
seed@ip-172-3...
seed@ip-172-3...

```
seed@ip-172-31-14-16:~/Documents/lab4$ echo "write to file" > etc/zzz
bash: etc/zzz: No such file or directory
seed@ip-172-31-14-16:~/Documents/lab4$ echo "something" > etc/zzz
bash: etc/zzz: No such file or directory
seed@ip-172-31-14-16:~/Documents/lab4$ cat /etc/zzz
A privilige granted by HENIL to this file.
seed@ip-172-31-14-16:~/Documents/lab4$ ls -l /etc/zzz
-rw-r--r-- 1 root root 43 Oct 11 13:43 /etc/zzz
seed@ip-172-31-14-16:~/Documents/lab4$ cat /etc/zzz
A privilige granted by HENIL to this file.
seed@ip-172-31-14-16:~/Documents/lab4$ sudo chmod 0644 /etc/zzz
seed@ip-172-31-14-16:~/Documents/lab4$ gcc cap_leak.c -o cap_leak
seed@ip-172-31-14-16:~/Documents/lab4$ ls
a.out     cap_leak.c  child     myenv.c  myenv2        mysys    parent
cap_leak  catall.c    cout.txt  myenv1   myprintenv.c  mysys.c  pout.txt
seed@ip-172-31-14-16:~/Documents/lab4$ sudo chown root:root cap_leak
seed@ip-172-31-14-16:~/Documents/lab4$ sudo chmod 0755 cap_leak
seed@ip-172-31-14-16:~/Documents/lab4$ ls -l cap_leak
-rwxr-xr-x 1 root root 17008 Oct 11 13:49 cap_leak
seed@ip-172-31-14-16:~/Documents/lab4$ sudo chmod +s  cap_leak
seed@ip-172-31-14-16:~/Documents/lab4$ ./cap_leak
fd is 3
$ cat /etc/zzz
A privilige granted by HENIL to this file.
```

Next, we run the program and again see the content of the zzz file, and we see that the file content is modified. This happens because even though in the program, we dropped the privileges, we did not close the file at the right time and hence the file was still running with privileged permissions that allowed the data in the file to be modified, even without the right permissions. Here, after calling fork, the control is passed to the child process and hence the malicious user is successful in modifying the content of a privileged file. This shows that it is important to close the file descriptor after dropping privileges, in order for it to have the appropriate permissions.



```
seed@ip-172-31-14-16:~/Documents/lab4$ sudo nano /etc/zzz
seed@ip-172-31-14-16:~/Documents/lab4$ ./cap_leak
fd is 3
$ cat /etc/zzz
A privilige granted by HENIL to this file.
Some more data due to leak
$ id
```