HACETTEPE UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BBM459 SECURE PROGRAMMING LAB - 2023 SPRING

Assignment 1

March 23, 2023

Student names:
Yiğit Emir İŞIKÇI
Abdullah Mert DİNÇER

Student Numbers:
2200356028
2200356016

1 Problem Definition

The aim of this assignment is that understand how environment variables affect program and system, how they work, how they are propagated from parent process to child, and how they affect system/program behaviors.

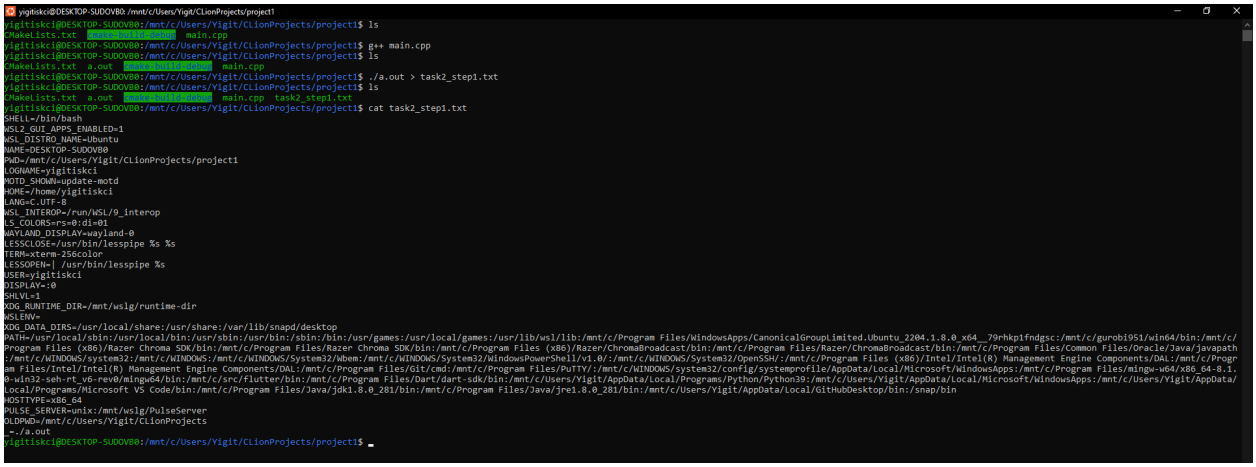
2 LAB TASKS

2.1 Task 1

In this task, we have learned the commands 'printenv', 'export' and 'unset'. we print the environment variables and PWD variable. Then we create a variable named "DENEME" with 'export' command. Then we deleted it with 'unset' command.

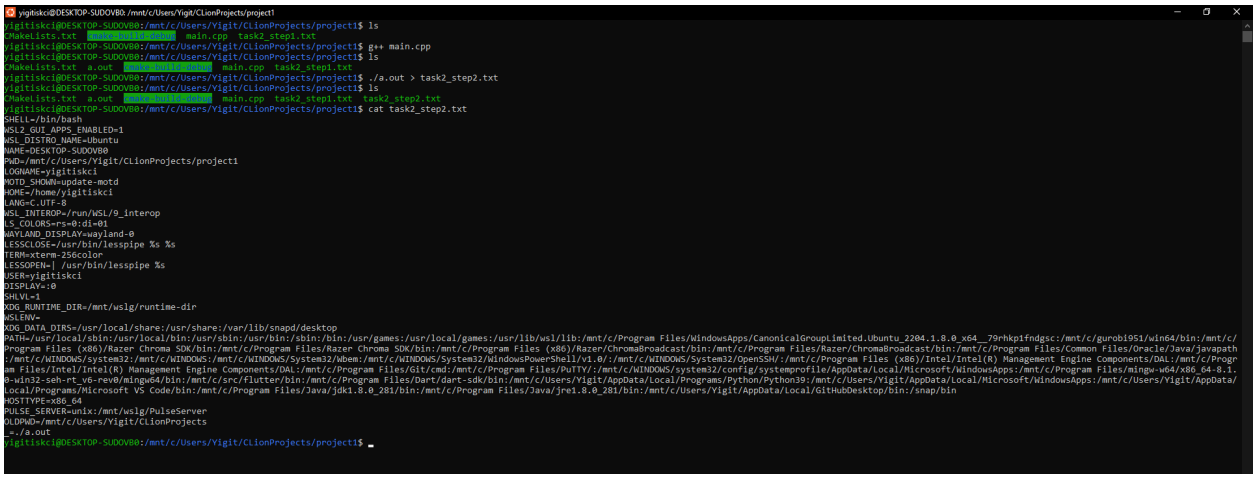
2.2 Task 2

2.2.1 Step 1



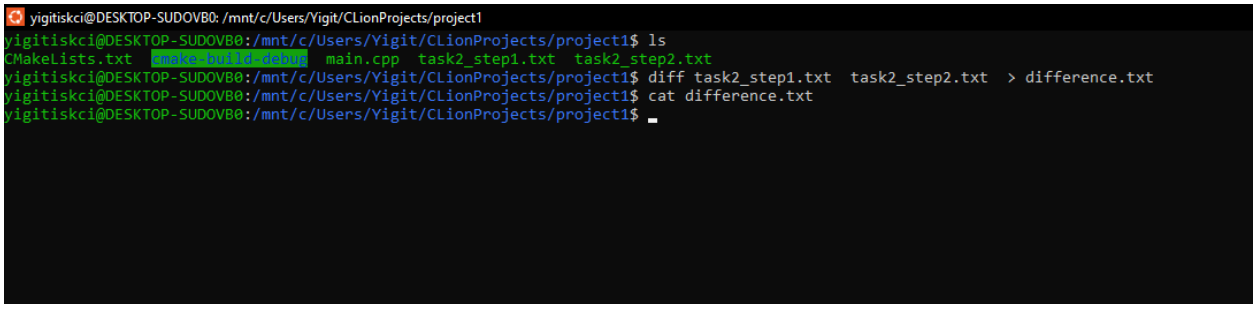
In this task, we simply printed all of the environment variables with given c++ code. It did the same with 'printenv' command. We are in the child process and it is still printing variables. So in this step, our guess is child process inherits env variables from its parent.

2.2.2 Step 2



This output looks like the same with the step 1 which is the child process output. Let see whether they are equal or not at step 3.

2.2.3 Step 3



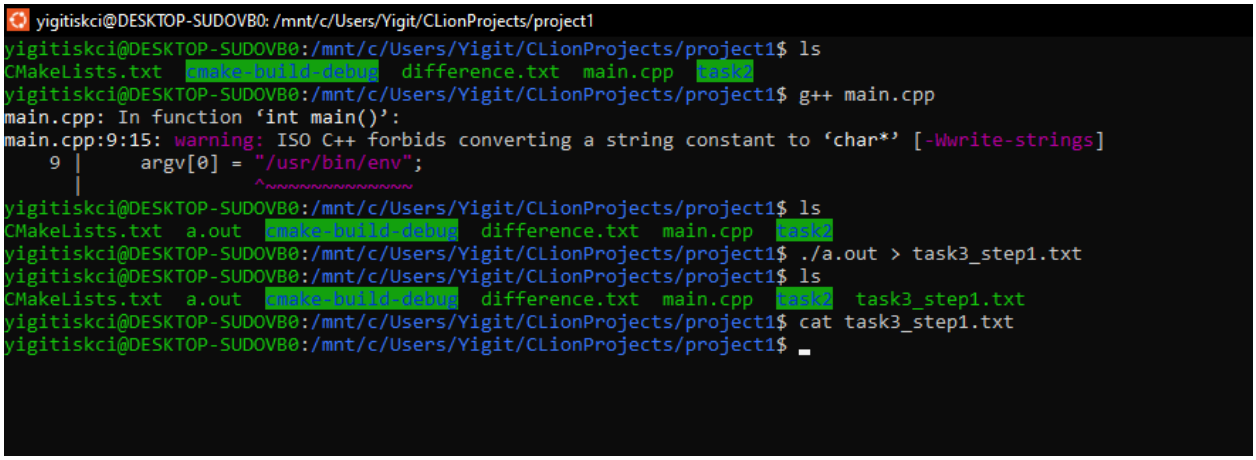
We can see that two outputs are equal. Therefore, parent's environment variables are inherited by the child process.

2.3 Task 3

At this task, we have learned how environment variables are affected when a new program executed with the function "execve()".

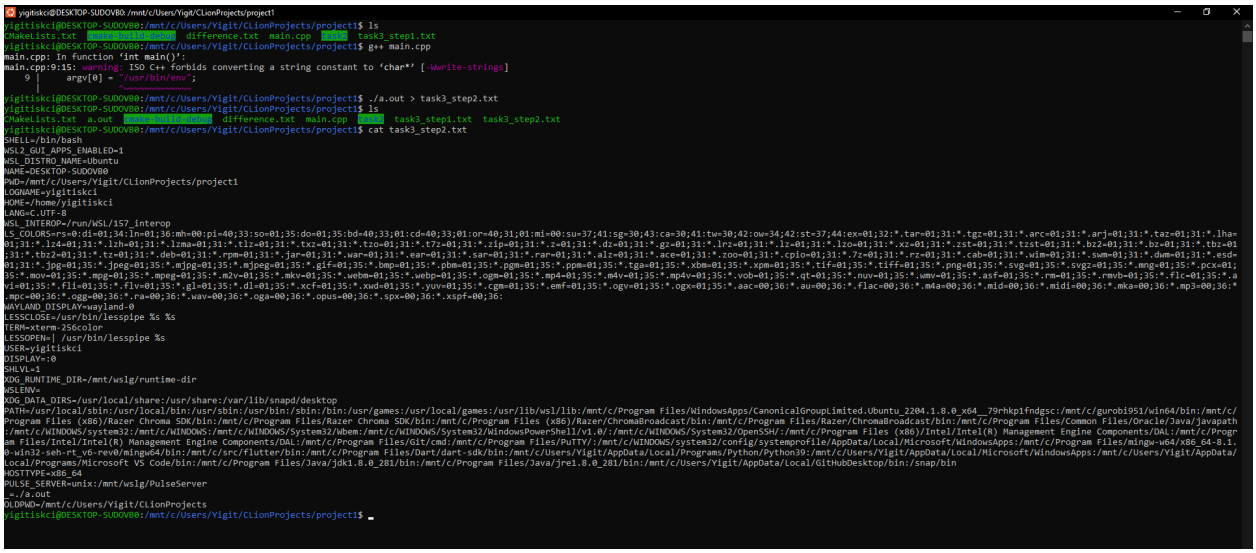
2.3.1 Step 1

When a NULL value is passed as the third argument to the function during the first step, the newly executed program inherits a copy of the environment variables from the calling program.



2.3.2 Step 2

When the "environ" variable is passed to the function as a third parameter, new program used the passed environment variables.

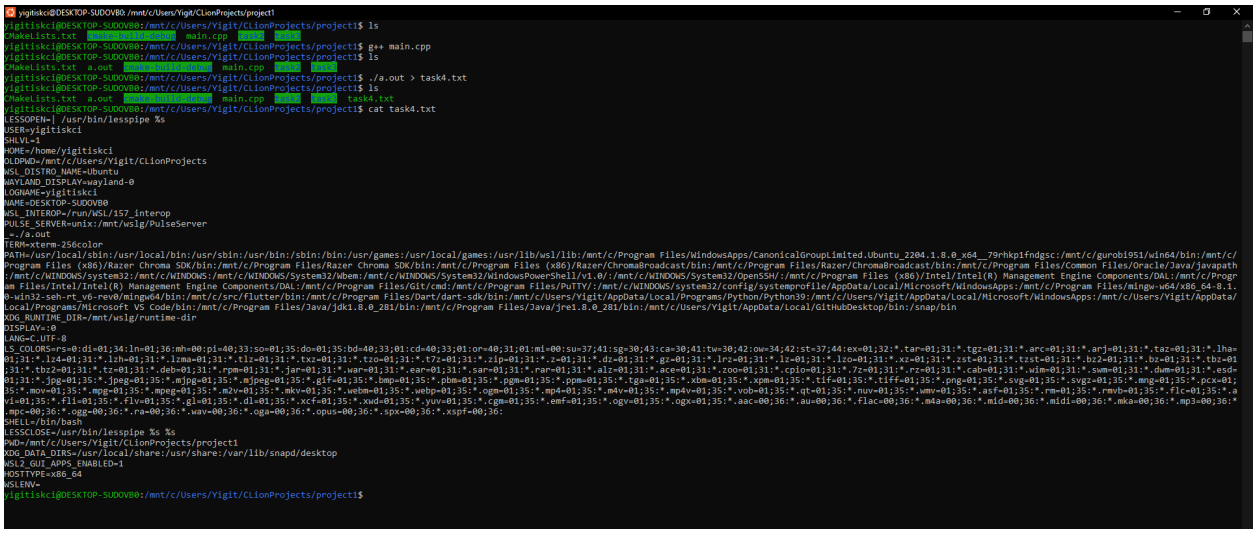


2.3.3 Step 3

In the second code, the environment variables are passed to "execve()" as an argument, whereas the first code depends on the default behavior of inheriting the calling program's environment variables.

2.4 Task 4

In this task, we explore how environment variables are propagated when a new program is run using the "system()" function.
The given program uses the "system()" function to execute the command "/usr/bin/env", which displays the environment variables of the current process. Since "system()" uses "/bin/sh" to execute the command, the environment variables of the calling process passed to "/bin/sh" displayed by the "env" command.



2.5 Task 5

Set-UID is an important security mechanism in Unix operating systems. When a Set-UID program runs, it assumes the owner's privileges. We have learned how Set_UID programs are affected.

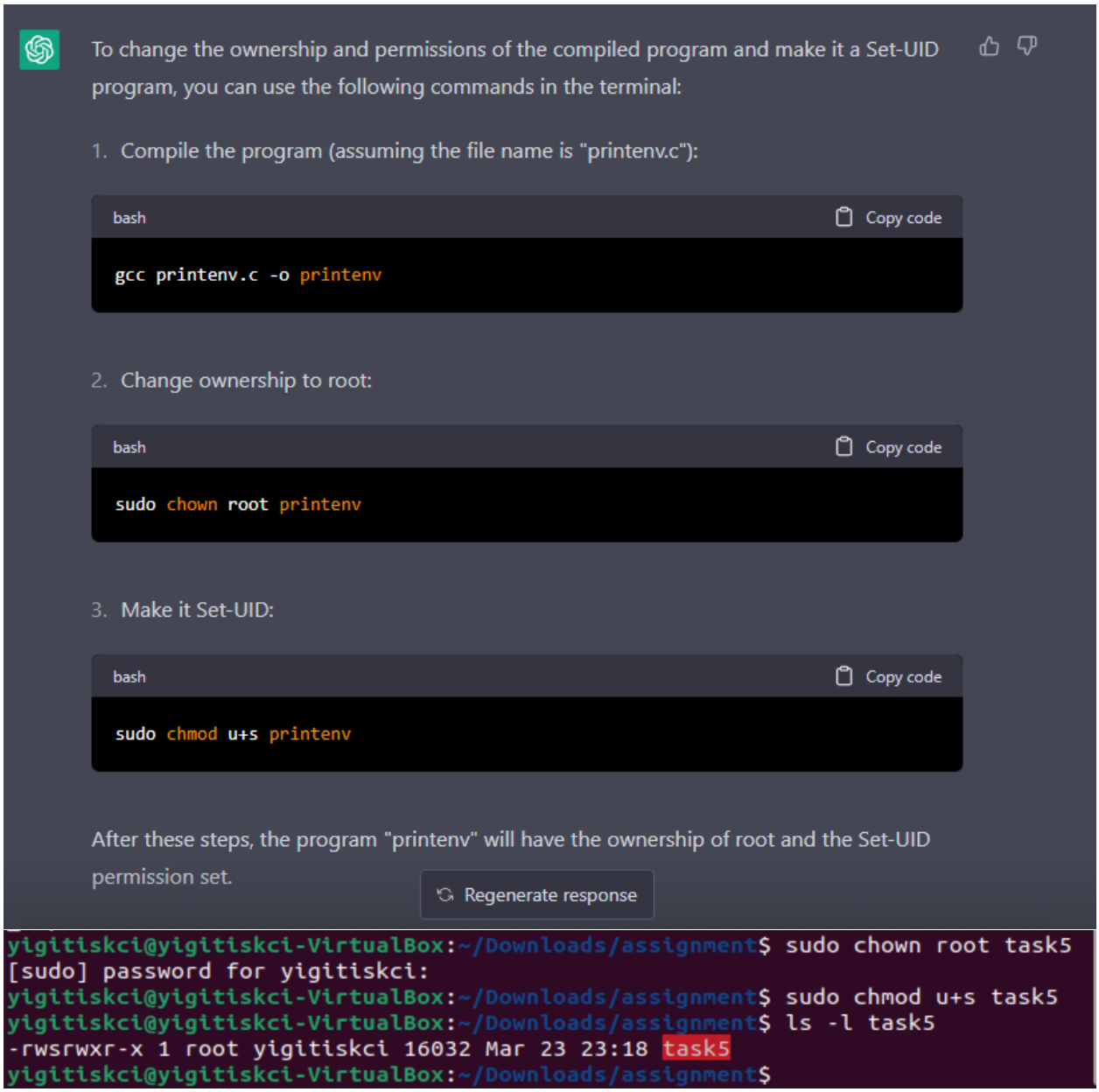
2.5.1 Step 1

```
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ ls
libmylib.so.1.0.1  mylib.c  mylib.o  myprog.c  task5.c
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ gcc task5.c -o task5
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ ls
libmylib.so.1.0.1  mylib.c  mylib.o  myprog.c  task5  task5.c
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ ./task5
SHELL=/bin/bash
SESSION_MANAGER=local/yigitiskci-VirtualBox:@/tmp/.ICE-unix/1932,unix/yigitiskc
i-VirtualBox:/tmp/.ICE-unix/1932
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
SSH_AGENT_LAUNCHER=gnome-keyring
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
LC_ADDRESS=tr_TR.UTF-8
GNOME_SHELL_SESSION_MODE=ubuntu
LC_NAME=tr_TR.UTF-8
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
LC_MONETARY=tr_TR.UTF-8
GTK_MODULES=gail:atk-bridge
DBUS_STARTER_BUS_TYPE=session
PWD=/home/yigitiskci/Downloads/assignment
LOGNAME=yigitiskci
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=wayland
SYSTEMD_EXEC_PID=1932
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/18658b26_5f8a_4286_98b3_1ce4b1
029567
GNOME_SETUP_DISPLAY=:1
LESSCLOSE=/usr/bin/lesspipe %s %s
XDG_SESSION_CLASS=user
TERM=xterm-256color
LC_IDENTIFICATION=tr_TR.UTF-8
LESSOPEN=| /usr/bin/lesspipe %s
USER=yigitiskci
GNOME_TERMINAL_SERVICE=:1.142
DISPLAY=:0
SHLVL=1
LC_TELEPHONE=tr_TR.UTF-8
QT_IM_MODULE=ibus
LC_MEASUREMENT=tr_TR.UTF-8
DBUS_STARTER_ADDRESS=unix:path=/run/user/1000/bus,guid=54860d9c9a96bdeae6e6f8e4
641c5331
XDG_RUNTIME_DIR=/run/user/1000
LC_TIME=tr_TR.UTF-8
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/local/share/:/usr/share/:/var/lib/snapd/de
sktop
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/u
sr/local/games:/snap/bin:/snap/bin
GDMSESSION=ubuntu
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus,guid=54860d9c9a96bdeae6e6
f8e4641c5331
LC_NUMERIC=tr_TR.UTF-8
_=./task5
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$
```

Given code printed environment variables.

2.5.2 Step 2

In this task, we got help from ChatGPT and made our program a Set-UID root program.



2.5.3 Step 3

Firstly, we used 'export' command to change "PATH" variable and "LD_LIBRARY_PATH" variable. Also created a variable named "ANY_NAME". As a result, program printed new variables.

```
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ export PATH=path_for_t
ask5
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ export LD_LIBRARY_PATH
=lib_path_for_task5
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ export PWD=pwd_for_tas
k5
yigitiskci@yigitiskci-VirtualBox:pwd_for_task5$ ./task5 > new_output.txt
yigitiskci@yigitiskci-VirtualBox:pwd_for_task5$
```

Here is the differences;

```
yigitiskci@yigitiskci-VirtualBox:pwd_for_task5$ diff old_output.txt new_output.
txt
48d47
< LD_LIBRARY_PATH=lib_path_for_task5
52c51
< PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:
/usr/local/games:/snap/bin:/snap/bin
---
> PATH=path_for_task5
yigitiskci@yigitiskci-VirtualBox:pwd_for_task5$
```

2.6 Task 6

2.6.1 Step 1

In this step. Created a program named "mylib.c" which overrides "sleep()" function.Then compiled it according to given codes.
After that, we set the 'LD_PRELOAD' variable with export command.
Finally, we compiled another program named "myprog.c". It gave us a warning but it compiled anyways.

```
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ ls
mylib.c  myprog.c
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ gcc -fPIC -g -c mylib.
c
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ gcc -shared -o libmyli
b.so.1.0.1 mylib.o -lc
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ ls
libmylib.so.1.0.1  mylib.c  mylib.o  myprog.c
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ export LD_PRELOAD=./li
bmylib.so.1.0.1
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ gcc myprog.c -o myprog
myprog.c: In function ‘main’:
myprog.c:4:5: warning: implicit declaration of function ‘sleep’ [-Wimplicit-fun
ction-declaration]
    4 |     sleep(1);
      |     ^~~~~
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ ls
libmylib.so.1.0.1  mylib.c  mylib.o  myprog  myprog.c
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$
```

2.6.2 Step 2

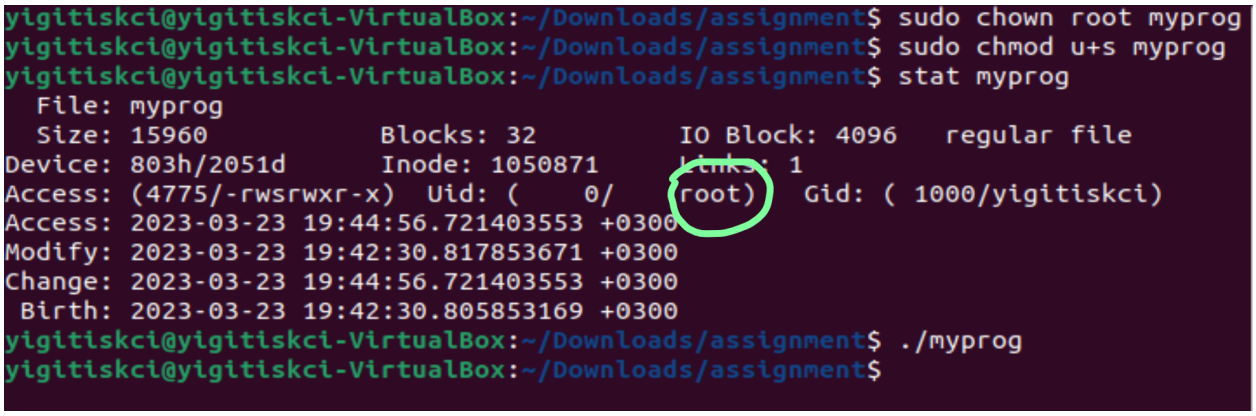
In this this step, we study how programs deal with environment variables (LD_PRELOAD in this task, which effect shared libs) in different cases.

At the first case, we run the "myprog" program as a normal user under the environment variable "LD_PRELOAD".

```
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ ./myprog
I am not sleeping!
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$
```

It printed "Im not sleeping!". Because it used the environment variable and it invoked the sleep() function in mylib.c.

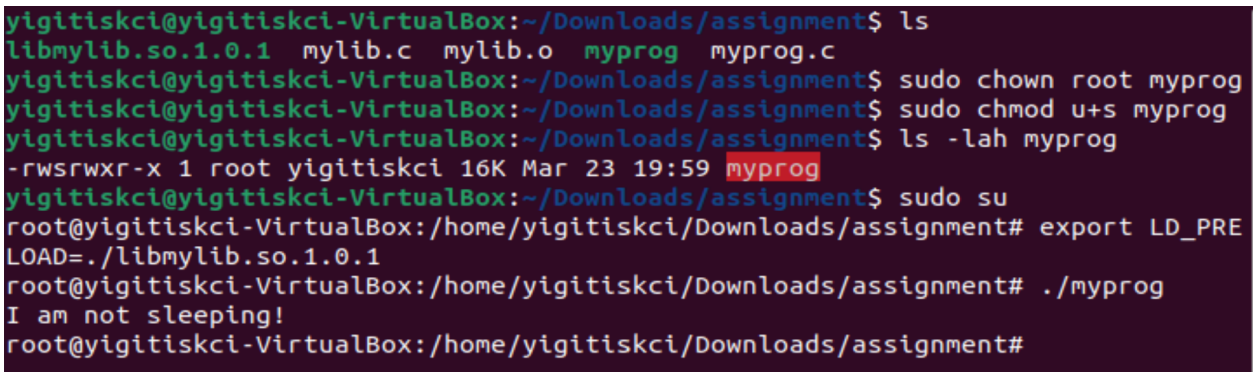
At the second case, we changed the owner of the myprog as root with the command "sudo chown root myprog". Also, we set the Set-UID bit of myprog with the command "sudo chmod u+s myprog". We checked the changes that we made with "ls -l" command. As we can see, owner is root and it has set-Uid bit which is 's' at "-rwsrwxr-x" in the permissions part of myprog file.



When we run the program, it used the original sleep() function in libc.so and waited 1 second. Because, we set the owner of the program as root and it used root's environment variables. Our local "LD_PRELOAD" variable did not effect our program.

In the third case, we should have do Set_UID root our program. While we was doing it, we noticed something. In the cases where we changed the permissions of the program as "chmod u+s myprog" and as "chmod ugo+s myprog" it behaves different. Normally, according to our research, both commands are same. But it behaved different.

In the following 2 image, we did the same things except "chmod" command. We made myprog a Set_UID root program, then we changed our user to root and exported the variable LD_PRELOAD again.



```
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ ls
libmylib.so.1.0.1  mylib.c  mylib.o  myprog  myprog.c
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ sudo chown root myprog
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ sudo chmod ugo+s mypro
g
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ ls -lah myprog
-rwsrwsr-x 1 root yigitiskci 16K Mar 23 20:01 myprog
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ sudo su
root@yigitiskci-VirtualBox:/home/yigitiskci/Downloads/assignment# export LD_PRE
LOAD=./libmylib.so.1.0.1
root@yigitiskci-VirtualBox:/home/yigitiskci/Downloads/assignment# ./myprog
root@yigitiskci-VirtualBox:/home/yigitiskci/Downloads/assignment#
```

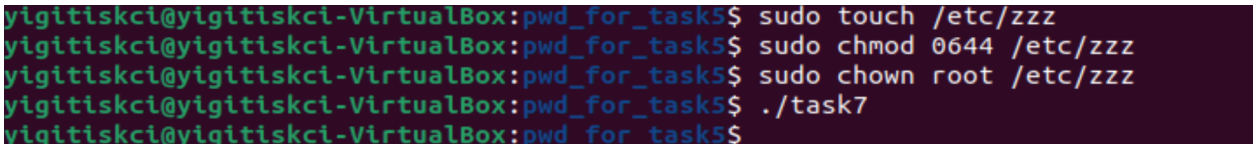
In first image, where we change the mod with the command "chmod u+s myprog", it override the sleep() function. But in second, it used the original one in libc.so. For us, it should behave as first one. Because we are setting the variable in root. So it should override the function.

In the 4th case, we created new user named "user1" firstly. Then, we made the myprog a Set-UID user1 program. Then we switch to the user1's account and we set the environment variable "LD_PRELOAD" with export command. Then we exit from user1's account and run the program in our real account.
It didnt override the sleep() function, because we didnt used the environment variables in user1.

```
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ ls
libmylib.so.1.0.1  mylib.c  mylib.o  myprog  myprog.c
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ sudo chown user1 mypro
g
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ sudo chmod u+s myprog
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ ls -lah myprog
-rwsrwxr-x 1 user1 yigitiskci 16K Mar 23 20:04 myprog
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ su user1
Password:
user1@yigitiskci-VirtualBox:/home/yigitiskci/Downloads/assignment$ export LD_PR
ELOAD=./libmylib.so.1.0.1
user1@yigitiskci-VirtualBox:/home/yigitiskci/Downloads/assignment$ exit
exit
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$ ./myprog
yigitiskci@yigitiskci-VirtualBox:~/Downloads/assignment$
```

3 Task 7

These commands create an empty file named "/etc/zzz" with permission 0644 and ownership set to root.



The program runned with root privileges, and it opened the file "/etc/zzz" and perform some tasks. After that, it relinquished its root privileges permanently using the "setuid()" function, and it forked a child process. The child process tried to write some malicious data to the file "/etc/zzz", but it failed to do so because the program has already relinquished its root privileges. As a result, /etc/zzz file has not been modified.

4 Referances

https://openai.com/