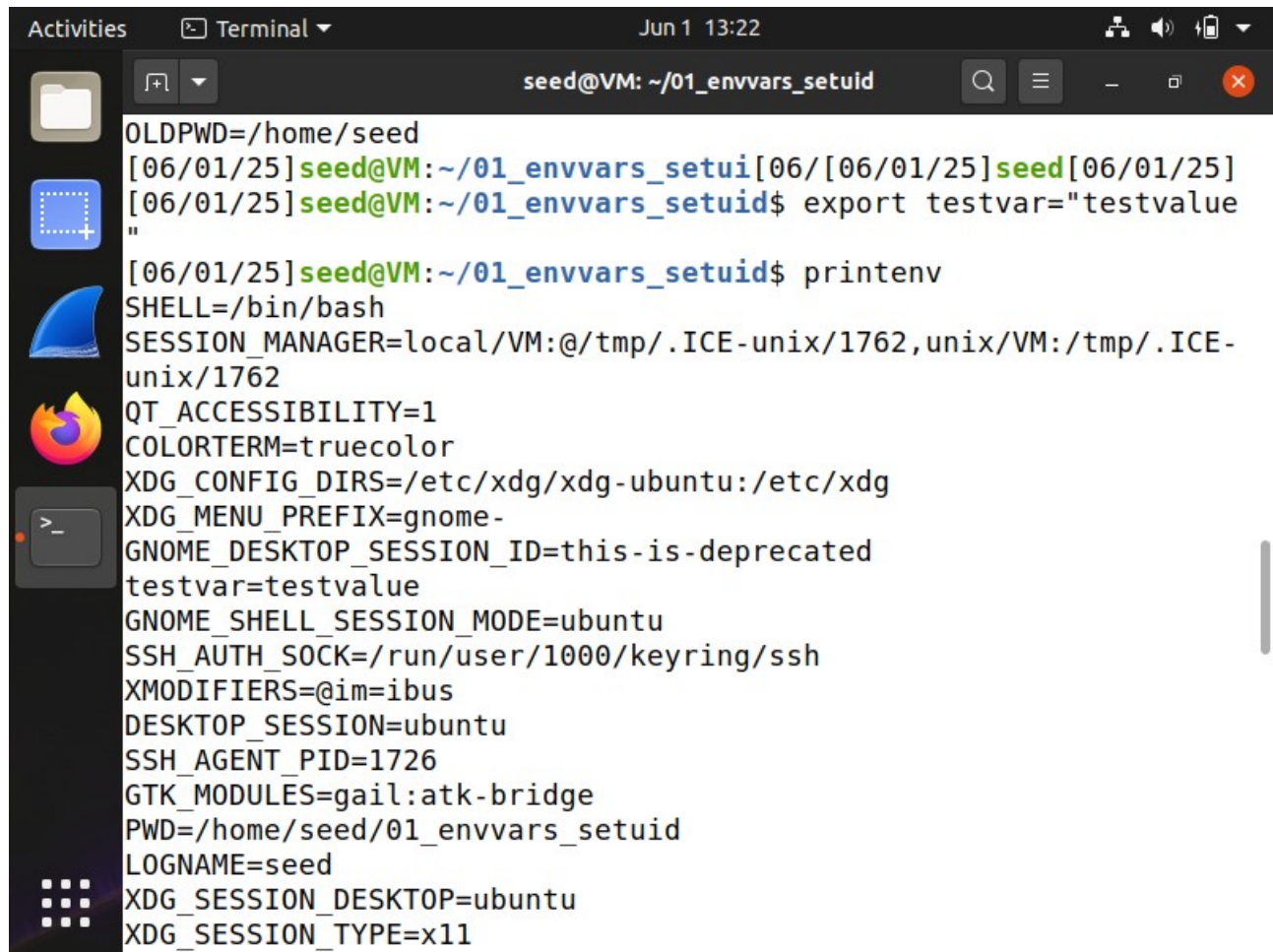Task 1: Manipulating Environment Variables

Task 1.1: Viewing Environment Variables

I used the printenv command to display all current environment variables.

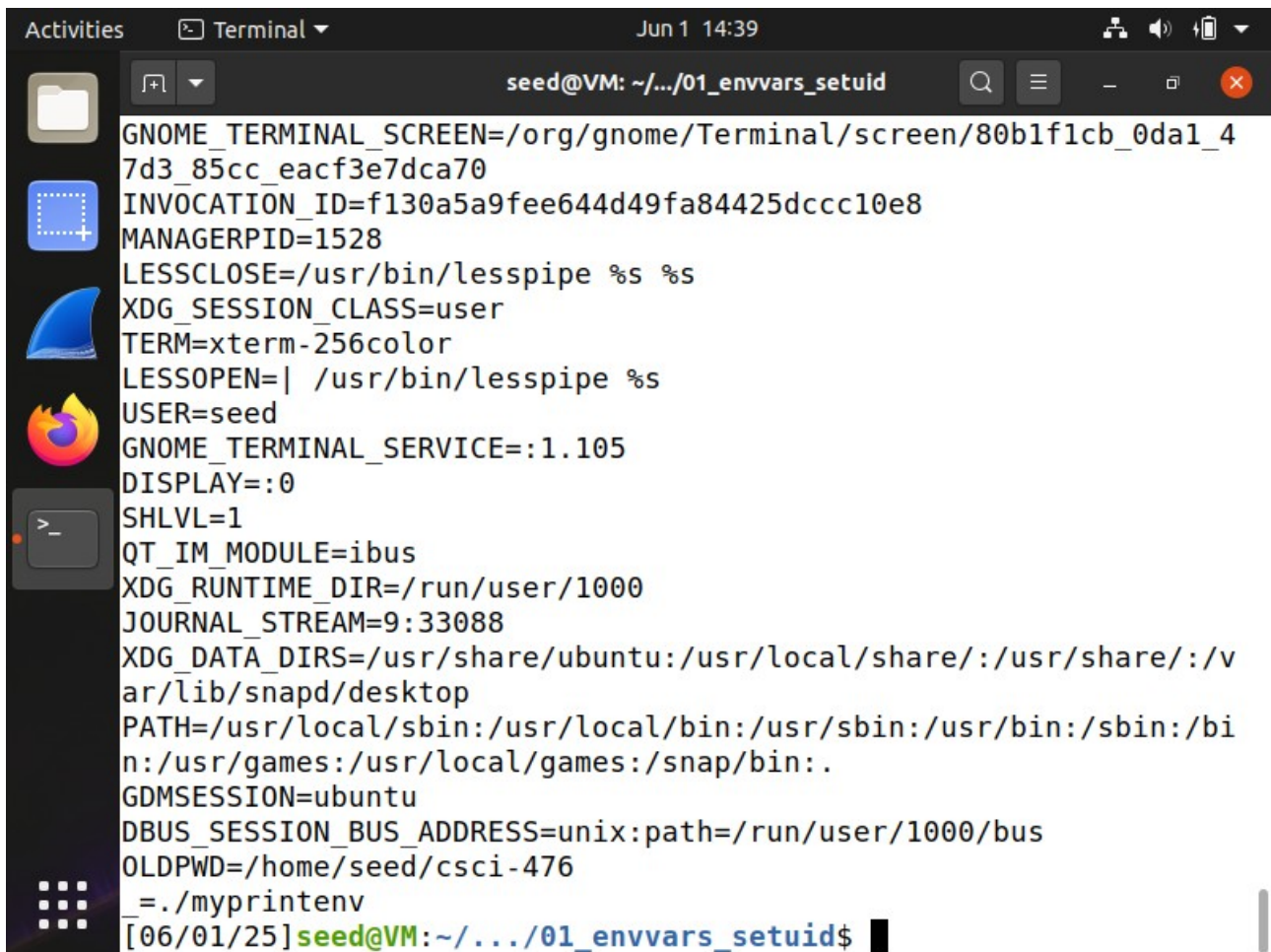Observations: The output lists key-value pairs representing the environment variables available in the current shell session.

Task 2: Passing Environment Variables (Parent → Child)

Task 2.1: Child Process Environment Variables

I compiled and ran the provided myprintenv.c program, which prints environment variables from the child process. The output was redirected to a file named myenv1.

```
remote: Total 582 (delta 115), reused 524 (delta 76), pack-reuse
d 0 (from 0)
Receiving objects: 100% (582/582), 12.13 MiB | 1.12 MiB/s, done.
Resolving deltas: 100% (115/115), done.
[06/01/25]seed@VM:~$ ls
01_envvars_setuid   Documents   Music       Templates
csci-476            Downloads   Pictures    Videos
Desktop             lab0        Public
[06/01/25]seed@VM:~$ cd csci-476/
[06/01/25]seed@VM:~/csci-476$ ls
00_intro             04_sqli          08_ske        README.md
01_envvars_setuid    05_xss           09_hashing    sniff_spoof
02_shellshock        06_tcp_attacks   apple_arm
03_buffer_overflow   07_dns_attacks   arm_build
[06/01/25]seed@VM:~/csci-476$ cd 01_envvars_setuid/
[06/01/25]seed@VM:~/.../01_envvars_setuid$ ls
cal.c         ls_vuln.c        myenv_execve.c   myprintenv.c
cap_leak.c    Makefile         myenv_pwd.c      passenv.c
catall.c      myenv_environ.c  myenv_system.c   README.md
hello.c       myenv_envp.c     mylib.c          sleep_prog.c
[06/01/25]seed@VM:~/.../01_envvars_setuid$ gcc myprintenv.c -o m
yprintenv
[06/01/25]seed@VM:~/.../01_envvars_setuid$ ./myprintenv > myenv1
[06/01/25]seed@VM:~/.../01_envvars_setuid$ 
```

Observations: The child process inherits environment variables from the parent process, as evidenced by the environment variables listed in myenv1.

Task 2.2: Parent Process Environment Variables

I modified myprintenv.c to comment out the printenv() call in the child process and uncomment it in the parent process. After recompiling and running the program, I redirected the output to out2.

Observations: The parent process's environment variables are consistent with those of the child process, indicating inheritance.

Task 2.3: Comparing Parent and Child Environment Variables

I used the diff myenv1 out2 command to compare the environment variables of the parent and child processes.
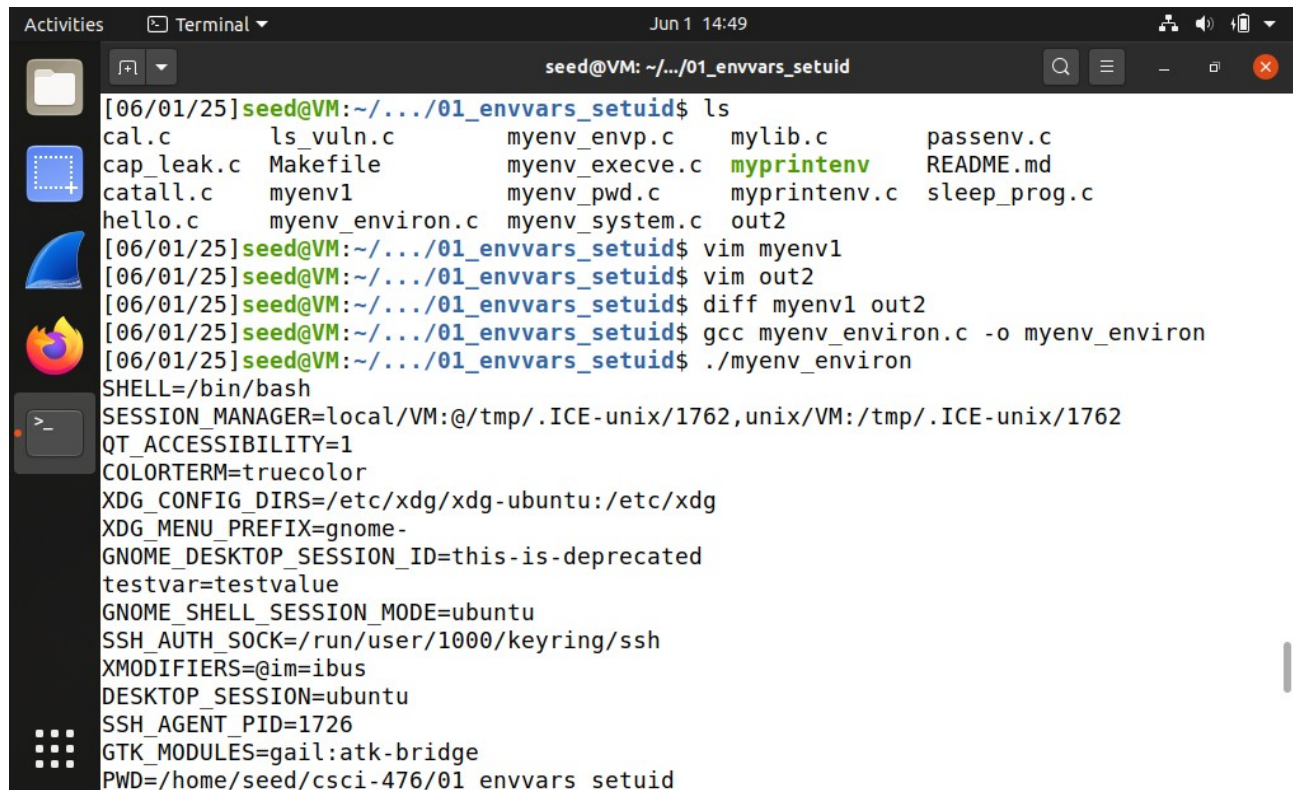
Observations: The absence of differences confirms that environment variables are inherited from parent to child processes.

Task 3: Environment Variables and execve()

Task 3.1: Executing /usr/bin/env with NULL Environment

Task 3.2:

```
0
INVOCATION_ID=f130a5a9fee644d49fa84425dccc10e8
MANAGERPID=1528
LESSCLOSE=/usr/bin/lesspipe %s %s
XDG_SESSION_CLASS=user
TERM=xterm-256color
LESSOPEN=| /usr/bin/lesspipe %s
USER=seed
GNOME_TERMINAL_SERVICE=:1.105
DISPLAY=:0
SHLVL=1
QT_IM_MODULE=ibus
XDG_RUNTIME_DIR=/run/user/1000
JOURNAL_STREAM=9:33088
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/local/share/:/usr/share/:/var/lib/snapd/desktop
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/lo
cal/games:/snap/bin:.
GDMSESSION=ubuntu
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
OLDPWD=/home/seed/csci-476
_=./myenv_environ
[06/01/25]seed@VM:~/.../01_envvars_setuid$ sudo chown root myenv_environ
[06/01/25]seed@VM:~/.../01_envvars_setuid$ sudo chmod 4755 myenv_environ
[06/01/25]seed@VM:~/.../01_envvars_setuid$ █
```
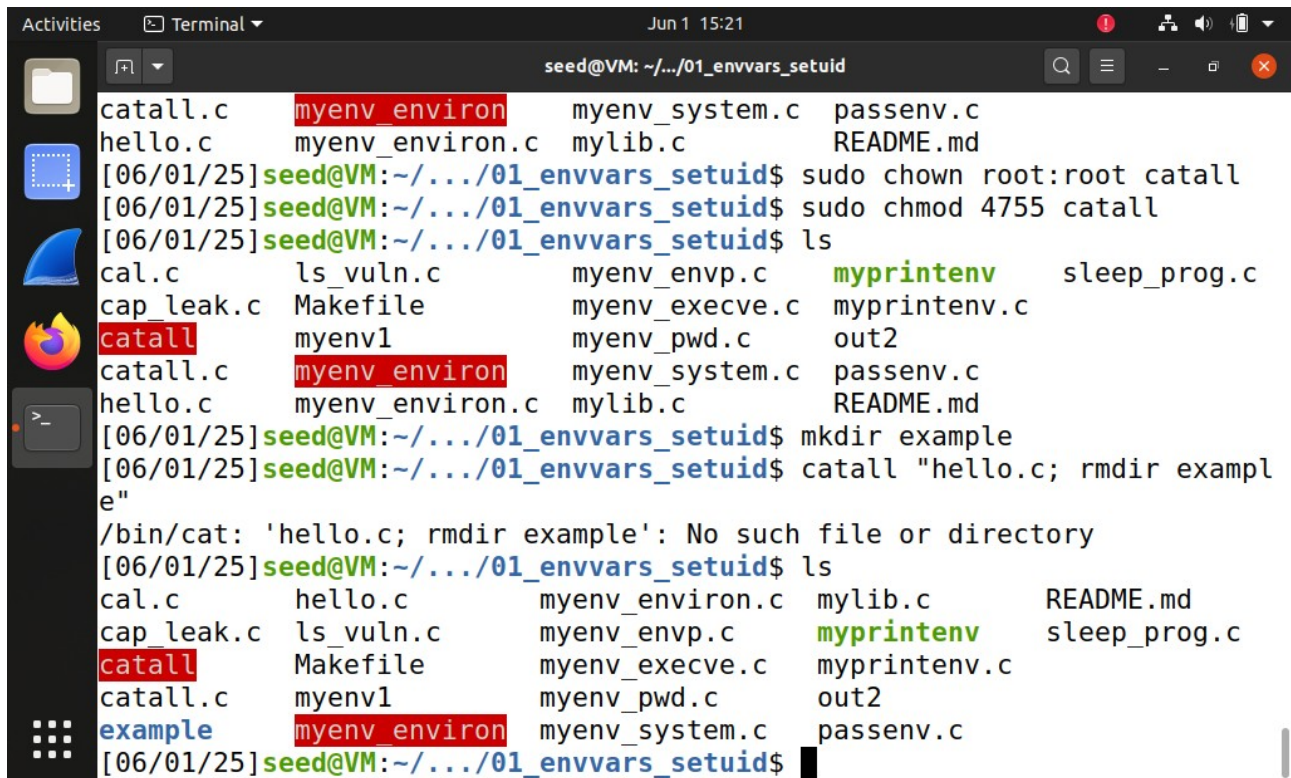
```
snapd/desktop
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/
games:/usr/local/games:/snap/bin:.
GDMSESSION=ubuntu
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
_=/usr/bin/env
OLDPWD=/home/seed/csci-476
[06/01/25]seed@VM:~/.../01_envvars_setuid$ export PATH=.:$PATH
[06/01/25]seed@VM:~/.../01_envvars_setuid$
[06/01/25]seed@VM:~/.../01_envvars_setuid$ export LD_LIBRARY_PATH=.:$LD
_LIBRARY_PATH
[06/01/25]seed@VM:~/.../01_envvars_setuid$
[06/01/25]seed@VM:~/.../01_envvars_setuid$ export TASK5="completed"
[06/01/25]seed@VM:~/.../01_envvars_setuid$ export TASK5="uuhhh"
[06/01/25]seed@VM:~/.../01_envvars_setuid$ ./myenv_environ
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/1762,unix/VM:/tmp/.ICE-unix/17
62
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
```

```
INVOCATION_ID=f130a5a9fee644d49fa84425dccc10e8
MANAGERPID=1528
LESSCLOSE=/usr/bin/lesspipe %s %s
XDG_SESSION_CLASS=user
TERM=xterm-256color
LESSOPEN=| /usr/bin/lesspipe %s
USER=seed
GNOME_TERMINAL_SERVICE=:1.105
TASK5=uuhhh
DISPLAY=:0
SHLVL=1
QT_IM_MODULE=ibus
XDG_RUNTIME_DIR=/run/user/1000
JOURNAL_STREAM=9:33088
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/local/share/:/usr/share/:/var/lib/
snapd/desktop
PATH=.:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/us
r/games:/usr/local/games:/snap/bin:.
GDMSESSION=ubuntu
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
OLDPWD=/home/seed/csci-476
```

Task 4: Environment Variables and system()

I compiled and ran a program that uses the system() function to execute rmdir

```
[06/01/25]seed@VM:~/.../01_envvars_setuid$ $ ./catall "hello.c; rmdir -f
 example"
[06/01/25]seed@VM:~/.../01_envvars_setuid$ catall "; hello.c; rmdir -f e
xample"
[06/01/25]seed@VM:~/.../01_envvars_setuid$ $ catall "hello.c; rmdir - ex
ample"
[06/01/25]seed@VM:~/.../01_envvars_setuid$ catall "hello.c; rmdir exampl
e"
#include <stdio.h>
int main(void)
{
    printf("hello world\n");
    return 0;
}
[06/01/25]seed@VM:~/.../01_envvars_setuid$ ls
cal.c        ls_vuln.c       myenv_envp.c    myprintenv     sleep_prog.c
cap_leak.c   Makefile        myenv_execve.c  myprintenv.c
catall       myenv1          myenv_pwd.c     out2
catall.c     myenv_environ   myenv_system.c  passenv.c
hello.c      myenv_environ.c mylib.c         README.md
[06/01/25]seed@VM:~/.../01_envvars_setuid$
```

Observations: The system() function invokes a shell, which inherits the environment variables, allowing the executed command to access them.

only system() seems to make a new terminal, the other option seems to try the enclosed as a full command

Task 5: Environment Variables and Set-UID Programs

Task 5.1: Creating a Set-UID Program

I wrote a program that prints environment variables, compiled it, changed its ownership to root, and set the Set-UID bit.

Observations: The program runs with root privileges due to the Set-UID bit, but the environment variables available to it may differ from those of the invoking user.

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

Observations: Modifying PATH can cause a Set-UID program to execute unintended commands, highlighting a security vulnerability.