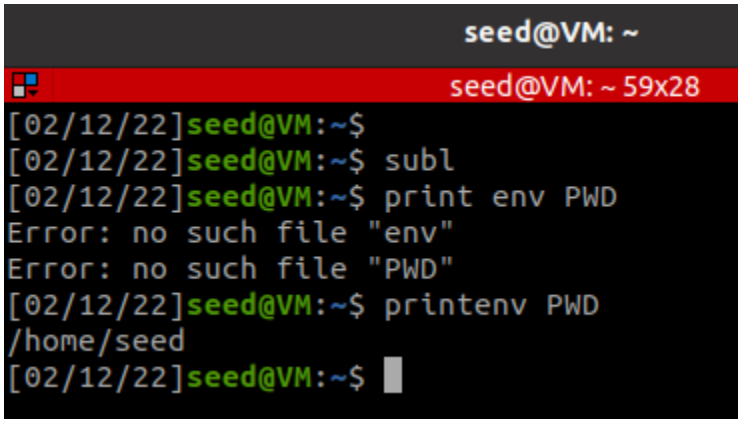
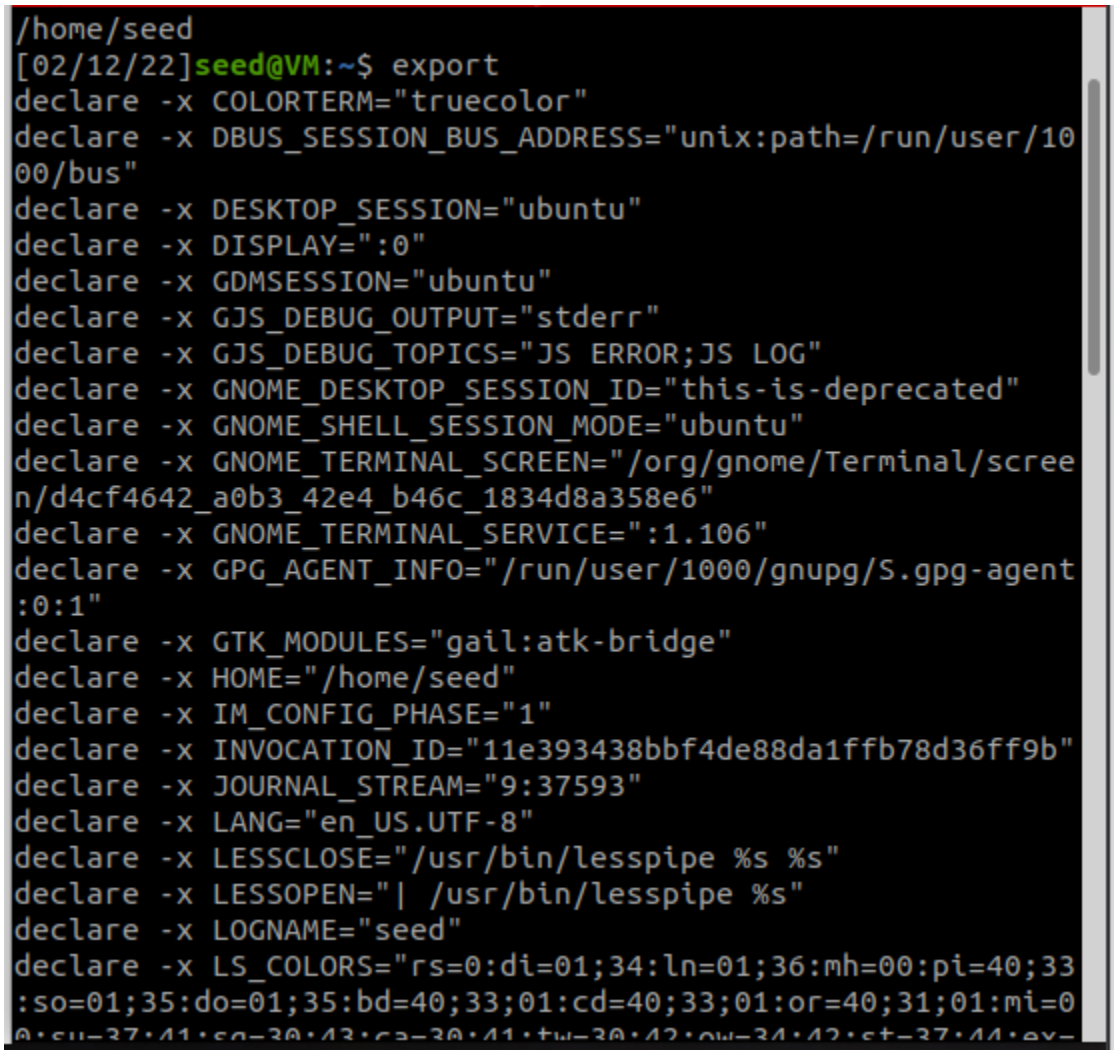
Val Robichaux
Homework 2
CSCE 465

Task 1:





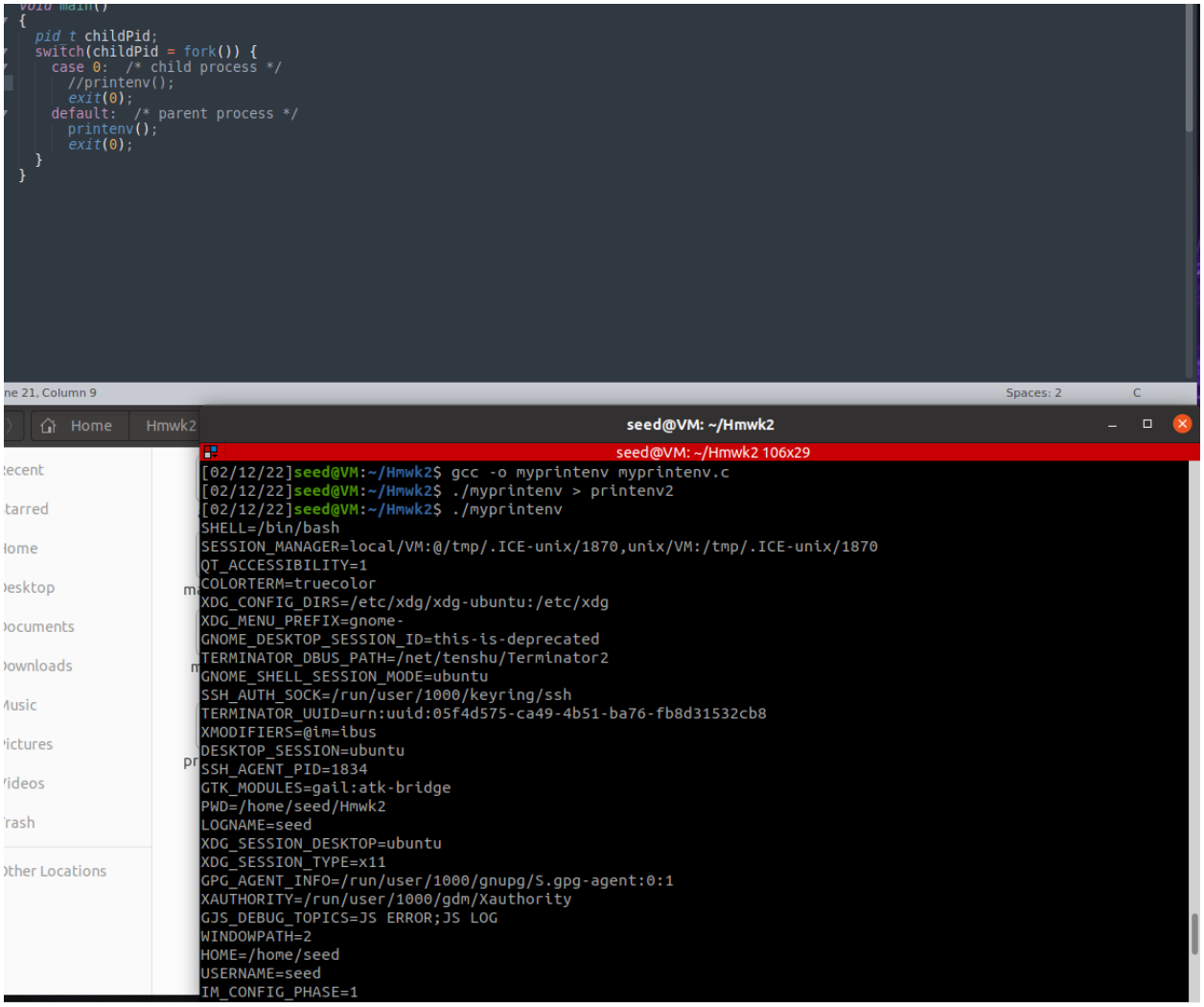
```
declare -x XDG_SESSION_TYPE="x11"
declare -x XMODIFIERS="@im=ibus"
[02/12/22]seed@VM:~$ unset
[02/12/22]seed@VM:~$
```

Task 2:

Step 1:

```
compilation terminated.
[02/12/22]seed@VM:~/Hmwk2$ gcc -o myprintenv myprintenv.c
[02/12/22]seed@VM:~/Hmwk2$ ./myprintenv
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/1870,unix/VM:/tmp/.ICE-unix/1870
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
TERMINATOR_DBUS_PATH=/net/tenshu/Terminator2
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
TERMINATOR_UUID=urn:uuid:05f4d575-ca49-4b51-ba76-fb8d31532cb8
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
SSH_AGENT_PID=1834
GTK_MODULES=gail:atk-bridge
PWD=/home/seed/Hmwk2
LOGNAME=seed
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=x11
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
XAUTHORITY=/run/user/1000/gdm/Xauthority
GJS_DEBUG_TOPICS=JS ERROR;JS LOG
WINDOWPATH=2
HOME=/home/seed
USERNAME=seed
IM_CONFIG_PHASE=1
LANG=en_US.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=0
0: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:*.ja
r=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:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=0
1;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:*.m
ov=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:
```

- Prints out my environment variables!



- It still prints out my environment variables even in the parent process.

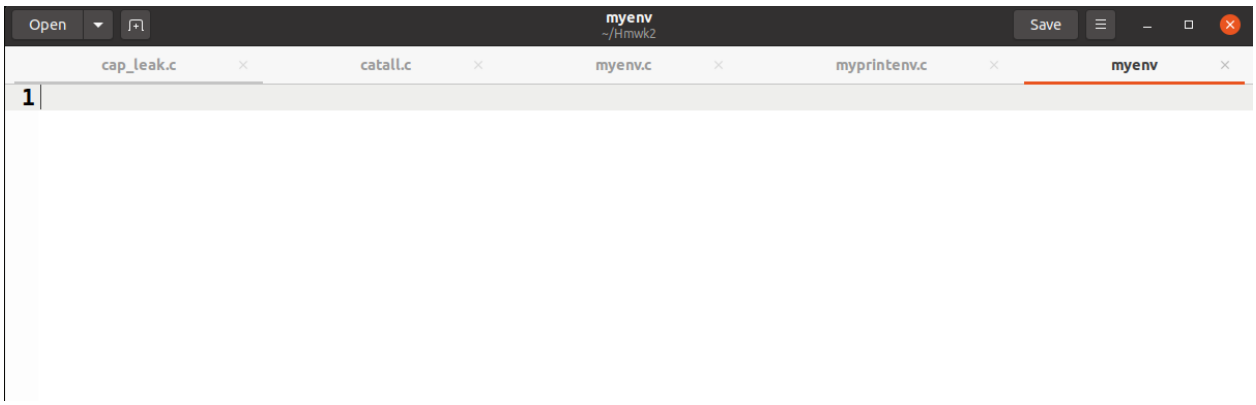
```c
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

extern char **environ;

void printenv()
{
  int i = 0;
  while (environ[i] != NULL) {
     printf("%s\n", environ[i]);
     i++;
  }
}

void main()
{
  pid_t childPid;
  switch(childPid = fork()) {
    case 0:  /* child process */
      //printenv();
      exit(0);
    default:  /* parent process */
      printenv();
      exit(0);
  }
}
```

- Here I ran the diff of these two files after compiling the two different sets and it seems as if they are identical. This seems odd as I would expect that the child process and parent process would have some differences. I have re-ran and debugged my process in this specific task as you can see, but I have come to no different conclusion other than that the files are the same.

Task 3:

```c
#include <unistd.h>

extern char **environ;

int main()
{
```

```
  char *argv[2];

  argv[0] = "/usr/bin/env";
  argv[1] = NULL;

  execve("/usr/bin/env", argv, NULL);

  return 0 ;
}
```



- This block of code seems to not inherit the environment variables.

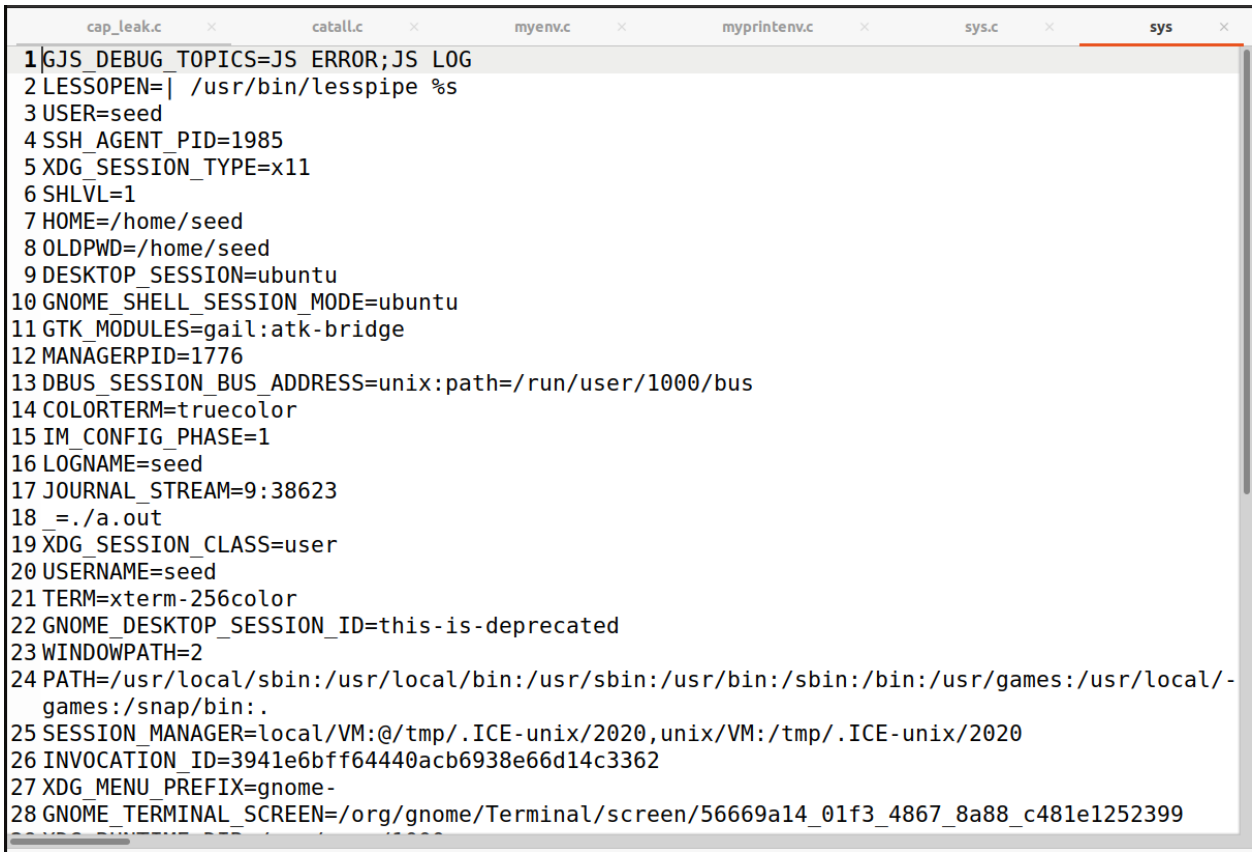
```
#include <unistd.h>

extern char **environ;

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

```
  argv[0] = "/usr/bin/env";
  argv[1] = NULL;

  execve("/usr/bin/env", argv, environ);

  return 0 ;
}
```

```
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/2020,unix/VM:/tmp/.ICE-unix/2020
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
SSH_AGENT_PID=1985
GTK_MODULES=gail:atk-bridge
PWD=/home/seed/Hmwk2
LOGNAME=seed
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=x11
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
XAUTHORITY=/run/user/1000/gdm/Xauthority
GJS_DEBUG_TOPICS=JS ERROR;JS LOG
WINDOWPATH=2
HOME=/home/seed
USERNAME=seed
"~/Hmwk2/myenv2" 49L, 2950C                                   1,1
```

- Whilst this block of code correctly inherits the environment variables
- The execve() command executes a program referred to by the pathname.
- The first argument is the PATH
- The second argument is an array of pointers to strings that passes to the program as command-line arguments
- The third argument is an array of pointer to strings, that are passed as the environment of the new program.
- This last point is extremely important because it shows us that firstly the environment and all of its inherited variables are NULL
- In the second example it is easy to see that we are passing our local environment to the function and thus getting our local environment variables returned to us.
- It is confirmed that environment variables are **NOT** automatically inherited

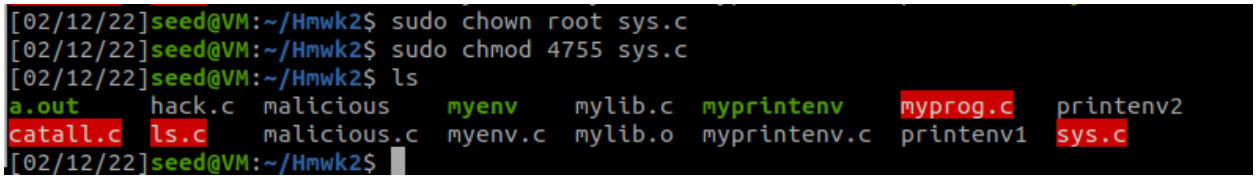
Task 4:



```
cap_leak.c  catall.c  myenv.c  myprintenv.c  sys.c  sys

1 GJS_DEBUG_TOPICS=JS ERROR;JS LOG
2 LESSOPEN=| /usr/bin/lesspipe %s
3 USER=seed
4 SSH_AGENT_PID=1985
5 XDG_SESSION_TYPE=x11
6 SHLVL=1
7 HOME=/home/seed
8 OLDPWD=/home/seed
9 DESKTOP_SESSION=ubuntu
10 GNOME_SHELL_SESSION_MODE=ubuntu
11 GTK_MODULES=gail:atk-bridge
12 MANAGERPID=1776
13 DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
14 COLORTERM=truecolor
15 IM_CONFIG_PHASE=1
16 LOGNAME=seed
17 JOURNAL_STREAM=9:38623
18 _=./a.out
19 XDG_SESSION_CLASS=user
20 USERNAME=seed
21 TERM=xterm-256color
22 GNOME_DESKTOP_SESSION_ID=this-is-deprecated
23 WINDOWPATH=2
24 PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/-
   games:/snap/bin:.
25 SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/2020,unix/VM:/tmp/.ICE-unix/2020
26 INVOCATION_ID=3941e6bff64440acb6938e66d14c3362
27 XDG_MENU_PREFIX=gnome-
28 GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/56669a14_01f3_4867_8a88_c481e1252399
```

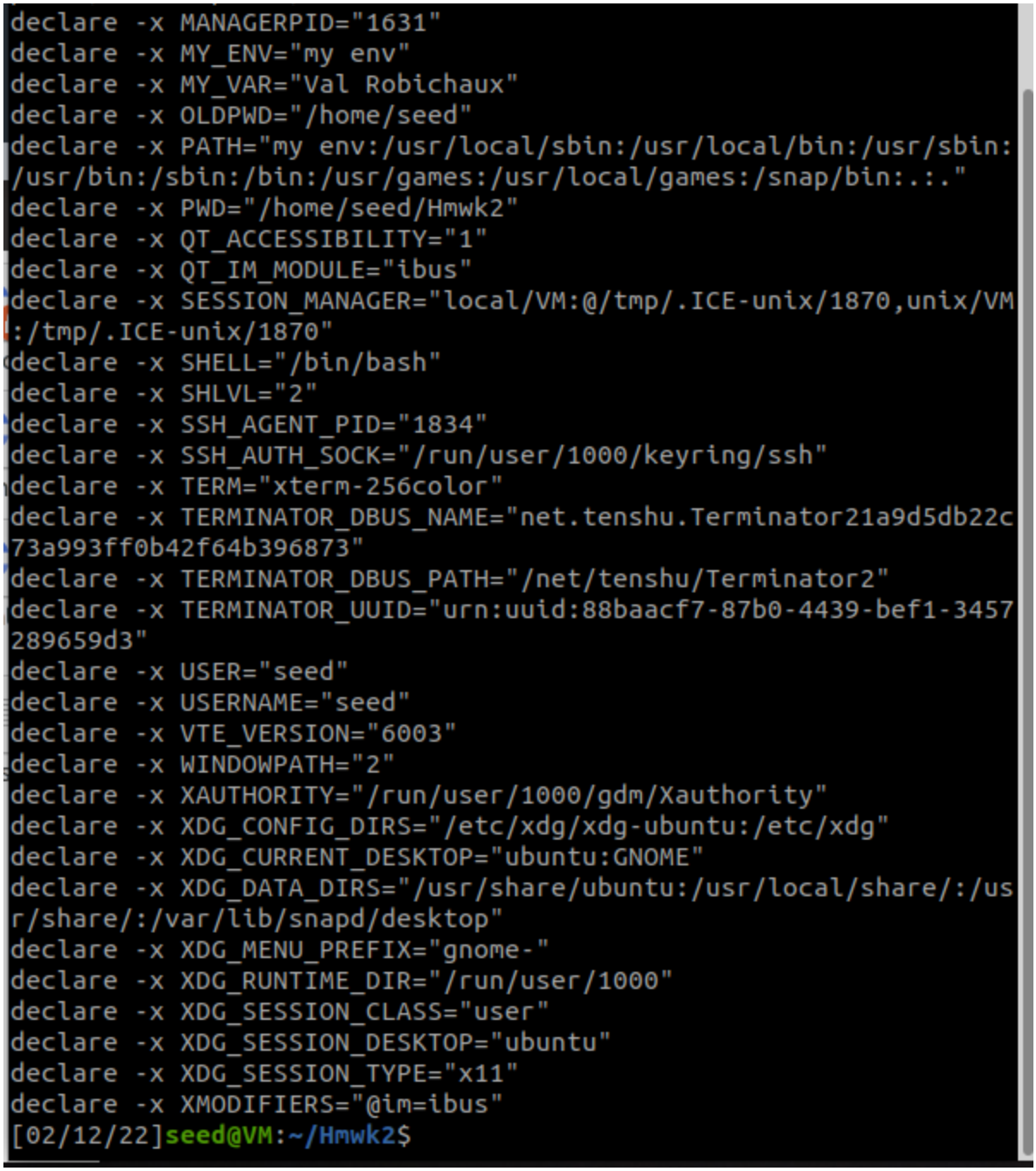
- Verified

Task 5:



```
[02/12/22]seed@VM:~/Hmwk2$ sudo chown root sys.c
[02/12/22]seed@VM:~/Hmwk2$ sudo chmod 4755 sys.c
[02/12/22]seed@VM:~/Hmwk2$ ls
a.out     hack.c  malicious    myenv    mylib.c  myprintenv    myprog.c   printenv2
catall.c  ls.c    malicious.c  myenv.c  mylib.o  myprintenv.c  printenv1  sys.c
[02/12/22]seed@VM:~/Hmwk2$
```

```
[02/12/22]seed@VM:~/Hmwk2$ export MY_ENV="my env"
[02/12/22]seed@VM:~/Hmwk2$ export PATH="my env":$PATH
[02/12/22]seed@VM:~/Hmwk2$ echo LD_LIBRARY_PATH
LD_LIBRARY_PATH
[02/12/22]seed@VM:~/Hmwk2$ echo $LD_LIBRARY_PATH

[02/12/22]seed@VM:~/Hmwk2$ export LD_LIBRARY_PATH
[02/12/22]seed@VM:~/Hmwk2$ export MY_VAR="Val Robichaux"
[02/12/22]seed@VM:~/Hmwk2$ export
declare -x COLORTERM="truecolor"
declare -x DBUS_SESSION_BUS_ADDRESS="unix:path=/run/user/1000/bus
"
declare -x DESKTOP_SESSION="ubuntu"
declare -x DISPLAY=":0"
declare -x GDMSESSION="ubuntu"
declare -x GJS_DEBUG_OUTPUT="stderr"
declare -x GJS_DEBUG_TOPICS="JS ERROR;JS LOG"
declare -x GNOME_DESKTOP_SESSION_ID="this-is-deprecated"
declare -x GNOME_SHELL_SESSION_MODE="ubuntu"
declare -x GNOME_TERMINAL_SCREEN="/org/gnome/Terminal/screen/d4cf
4642_a0b3_42e4_b46c_1834d8a358e6"
declare -x GNOME_TERMINAL_SERVICE=":1.106"
declare -x GPG_AGENT_INFO="/run/user/1000/gnupg/S.gpg-agent:0:1"
declare -x GTK_MODULES="gail:atk-bridge"
declare -x HOME="/home/seed"
declare -x IM_CONFIG_PHASE="1"
declare -x INVOCATION_ID="11e393438bbf4de88da1ffb78d36ff9b"
declare -x JOURNAL_STREAM="9:37593"
declare -x LANG="en_US.UTF-8"
```



- It worked! As you can see I export my MY_VAR variable to my own name and it shows up in the environment variables on my set-UID program. That's really neat.
- My PATH variable seemed really surprising because it seems to point to multiple different locations including some that I am surprised with such as bin:/usr/games

Task 6

```c
#include <stdio.h>
#include <stdlib.h>

extern char** environ;

int main()
{
    printf("Running my code\n");
    system("/bin/zsh");
    return 0;
}
```

```
[02/12/22]seed@VM:~/Hmwk2$ ./ls
a.out     ls.c          myenv.c     myprintenv.c  sys
catall.c  malicious     mylib.c     myprog.c      sys.c
hack.c    malicious.c   mylib.o     printenv1
ls        myenv         myprintenv  printenv2
[02/12/22]seed@VM:~/Hmwk2$ ls -l /bin/zsh
-rwxr-xr-x 1 root root 878288 Feb 23  2020 /bin/zsh
[02/12/22]seed@VM:~/Hmwk2$ ls -l /bin/sh
lrwxrwxrwx 1 root root 8 Feb 12 15:17 /bin/sh -> /bin/zsh
[02/12/22]seed@VM:~/Hmwk2$ export PATH=.:$PATH
[02/12/22]seed@VM:~/Hmwk2$ echo $PATH
.:my env:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:
/bin:/usr/games:/usr/local/games:/snap/bin:.:.
[02/12/22]seed@VM:~/Hmwk2$ gcc -o ls ls.c
[02/12/22]seed@VM:~/Hmwk2$ ./ls
[02/12/22]seed@VM:~/Hmwk2$ gcc -o malicious malicious.c
[02/12/22]seed@VM:~/Hmwk2$ ./malicious
Running my code
VM% exit
[02/12/22]seed@VM:~/Hmwk2$ ./ls
[02/12/22]seed@VM:~/Hmwk2$ gcc -o malicious malicious.c
[02/12/22]seed@VM:~/Hmwk2$ ./ls
[02/12/22]seed@VM:~/Hmwk2$ ./ls
id
[02/12/22]seed@VM:~/Hmwk2$ id
uid=1000(seed) gid=1000(seed) groups=1000(seed),4(adm),24(cdrom),
27(sudo),30(dip),46(plugdev),120(lpadmin),131(lxd),132(sambashare
),136(docker)
[02/12/22]seed@VM:~/Hmwk2$
```

- I am having a hard time running my own code, but I can still make the original code run something else! I changed the path so that I could possibly get it to run by the absolute path instead of the relative path but could not navigate to run my malicious code
- It also seems as if the malicious code that I am running does not have root privileges as you can see from the ID call I made.

Task 7

malicious | cap_leak.c | malicious.c | mylib.c | myprog.c | file

```
1 I am not sleeping!
```

Plain Text | Tab Width: 8 | Ln 1, Col 19 | INS

```
ored.
Please workERROR: ld.so: object './libmyl
 LD_PRELOAD cannot be preloaded (cannot c
 file): ignored.
[02/10/22]seed@VM:~/Documents$ ./a.out >
ERROR: ld.so: object './libmylib.so.1.0.1
 cannot be preloaded (cannot open shared
ored.
ERROR: ld.so: object './libmylib.so.1.0.1
 cannot be preloaded (cannot open shared
ored.
[02/10/22]seed@VM:~/Documents$ cd ..
ERROR: ld.so: object './libmylib.so.1.0.1
 cannot be preloaded (cannot open shared
ored.
[02/10/22]seed@VM:~$ cd Hmwk2
[02/10/22]seed@VM:~/Hmwk2$ ls
a.out       hack.c             ls.c
myprog.c
cap_leak.c  libmylib.so.1.0.1  malicious.
new
catall.c    ls                 myenv.c
sys.c
[02/10/22]seed@VM:~/Hmwk2$ gcc myprog.c
[02/10/22]seed@VM:~/Hmwk2$ a.out > file
[02/10/22]seed@VM:~/Hmwk2$
```

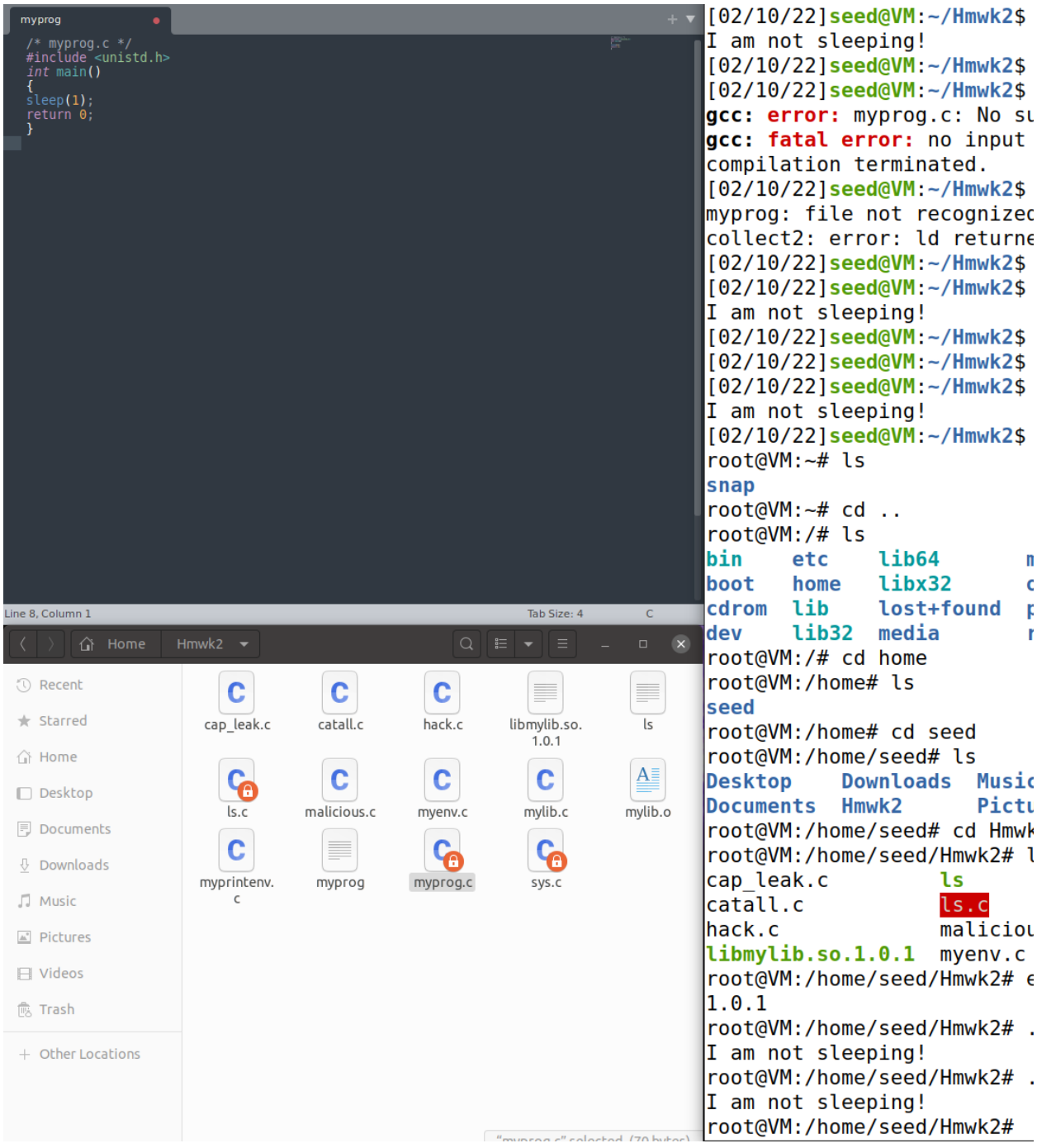
- My prog running as a normal file

```
1 I am not sleeping!
```

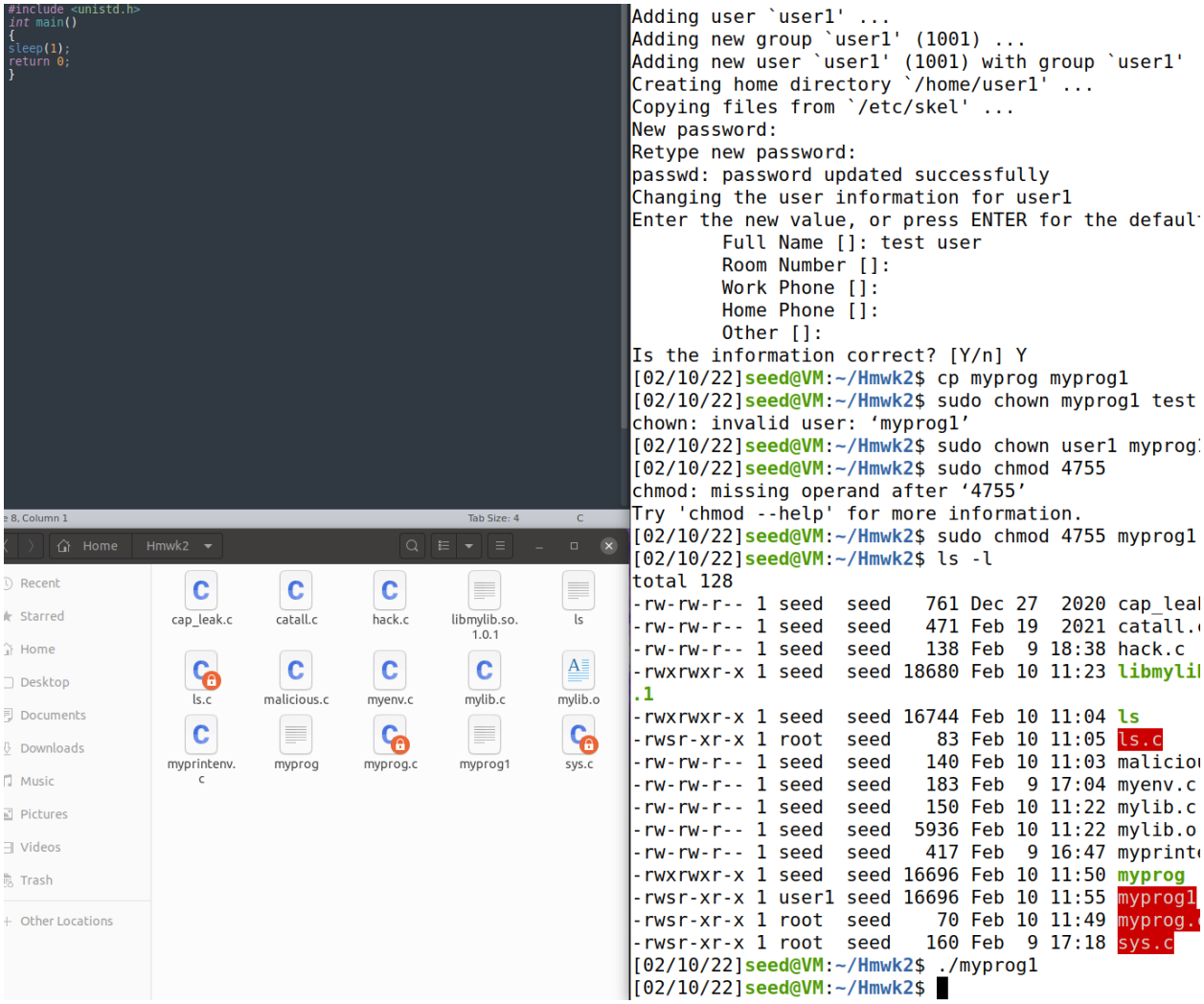
Plain Text | Tab Width: 8 | Ln 1, Col 19 | INS

```
 cannot be preloaded (cannot open shared object fil
ored.
[02/10/22]seed@VM:~/Documents$ cd ..
ERROR: ld.so: object './libmylib.so.1.0.1' from LD_
 cannot be preloaded (cannot open shared object fil
ored.
[02/10/22]seed@VM:~$ cd Hmwk2
[02/10/22]seed@VM:~/Hmwk2$ ls
a.out       hack.c             ls.c          mylib.c
myprog.c
cap_leak.c  libmylib.so.1.0.1  malicious.c   mylib.o
new
catall.c    ls                 myenv.c       myprint
sys.c
[02/10/22]seed@VM:~/Hmwk2$ gcc myprog.c
[02/10/22]seed@VM:~/Hmwk2$ a.out > file
[02/10/22]seed@VM:~/Hmwk2$ sudo chown root myprog.c
[02/10/22]seed@VM:~/Hmwk2$ sudo chmod 4755 myprog.c
[02/10/22]seed@VM:~/Hmwk2$ gcc myprog.c
[02/10/22]seed@VM:~/Hmwk2$ a.out file1
I am not sleeping!
[02/10/22]seed@VM:~/Hmwk2$ a.out
I am not sleeping!
[02/10/22]seed@VM:~/Hmwk2$ a.out > file2
[02/10/22]seed@VM:~/Hmwk2$
```

- My prog running as a SET-UID program



- Ran as a root use and changed LD_PRELOAD



- Ran as user1

Conclusion

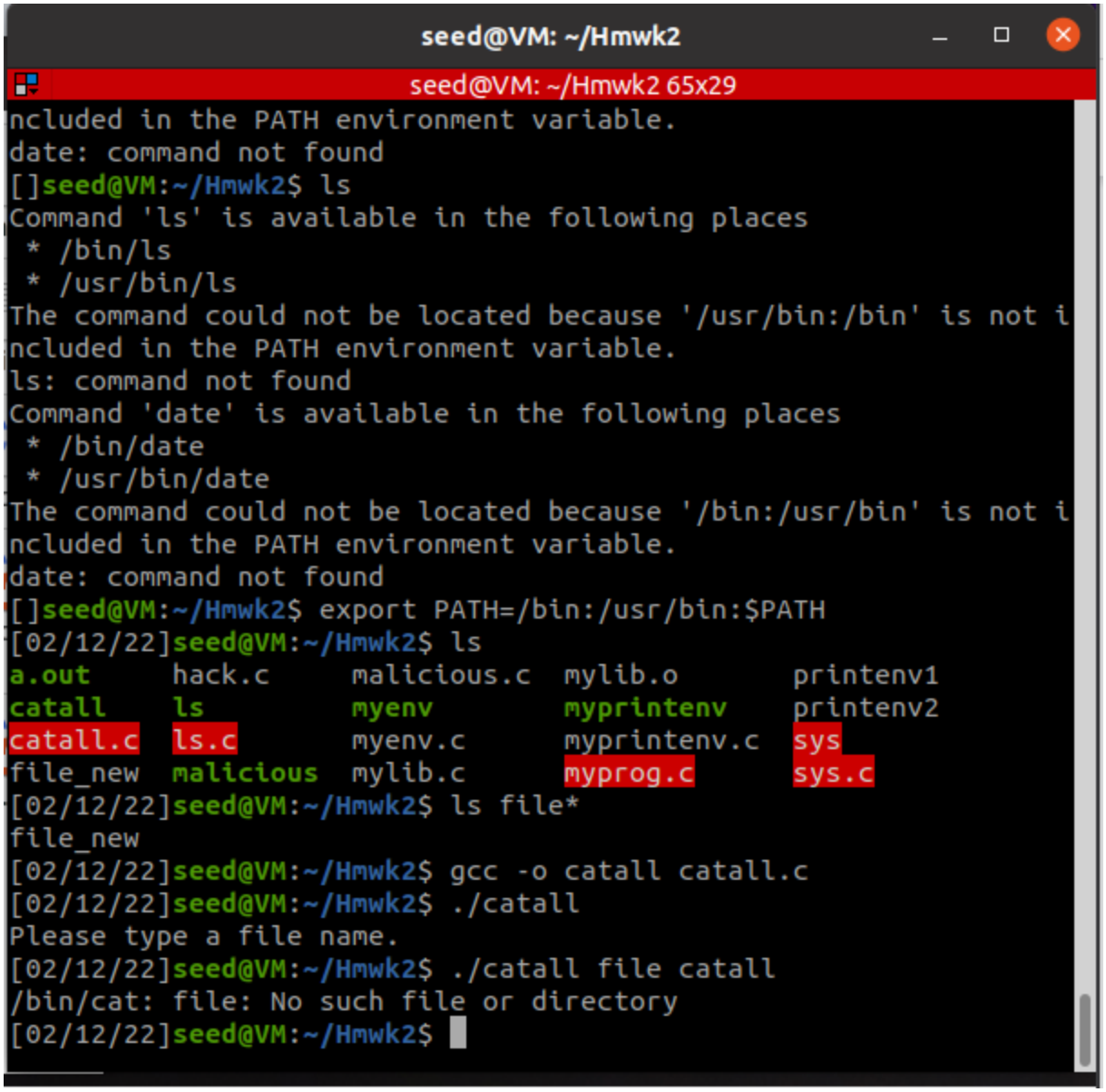
When the program is run either as a normal program of a SET-UID program, by exporting LD_PRELOAD, the program invokes the sleep function in **libmylib**. However, this is not the case when the function is run by user1. It will always call the sleep function in the default libraries. The LD_preload environment variables are not inherited.

Task 8

```
[02/12/22]seed@VM:~/Hmwk2$ ls -al file catall
[02/12/22]seed@VM:~/Hmwk2$ ls -al file catall
[02/12/22]seed@VM:~/Hmwk2$ a.out > file
[02/12/22]seed@VM:~/Hmwk2$ ls -al file catall
[02/12/22]seed@VM:~/Hmwk2$ ./catall "file;mv file file_new"
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/1870,unix/VM:/tmp/.ICE-u
nix/1870
QT_ACCESSIBILITY=1
COLORTERM=truecolor
```

```
[]seed@VM:~/Hmwk2$ export PATH=/bin:/usr/bin:$PATH
[02/12/22]seed@VM:~/Hmwk2$ ls
a.out     hack.c     malicious.c  mylib.o       printenv1
catall    ls         myenv        myprintenv    printenv2
catall.c  ls.c       myenv.c      myprintenv.c  sys
file_new  malicious  mylib.c      myprog.c      sys.c
[02/12/22]seed@VM:~/Hmwk2$ ls file*
file_new
[02/12/22]seed@VM:~/Hmwk2$
```

- Bob can remove, move, write to any file.
- We can change to the root user and compromise it by invoking a new shell and modifying the file.

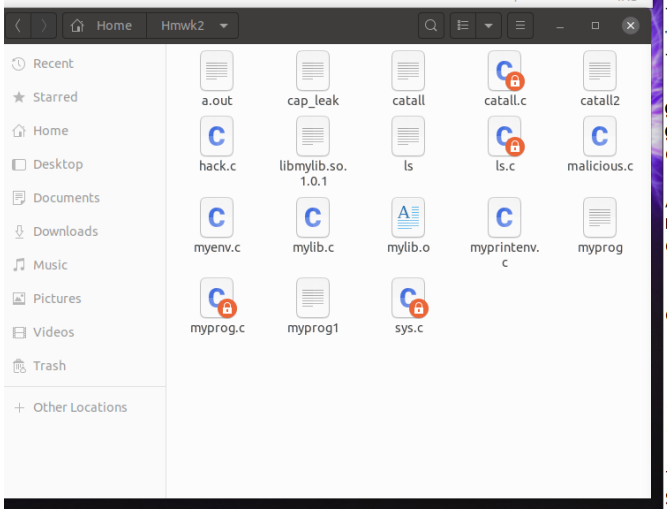


- The attack that I used will not work because the system() call /bin/sh links zsh. Ater running the program with root privileges I can write and move files freely. When running with execve() it will see my attacks as a folder name, so the system will have no prompt or usage of the file.
- It can’t even find my file in this case

Task 9

```
 9   char *v[2];
10
11   /* Assume that /etc/zzz is an important system
   file,
12    * and it is owned by root with permission 0644.
13    * Before running this program, you should create
14    * the file /etc/zzz first. */
15   fd = open("/etc/zzz", O_RDWR | O_APPEND);
16   if (fd == -1) {
17      printf("Cannot open /etc/zzz\n");
18      exit(0);
19   }
20
21   // Print out the file descriptor value
22   printf("fd is %d\n", fd);
23
24   // Permanently disable the privilege by making the
25   // effective uid the same as the real uid
26   setuid(getuid());
27
28   // Execute /bin/sh
29   v[0] = "/bin/sh"; v[1] = 0;
30   execve(v[0], v, 0);
31 }
```

C ▾ Tab Width: 8 ▾ Ln 19, Col 4 ▾ INS




```
initramfs-tools                 ubuntu-advantage
inputrc                         ucf.conf
insserv.conf.d                  udev
iproute2                        udisks2
issue                           ufw
issue.net                       update-manager
kernel                          update-motd.d
kernel-img.conf                 update-notifier
kerneloops.conf                 UPower
ldap                            usb_modeswitch.conf
ld.so.cache                     usb_modeswitch.d
ld.so.conf                      vim
ld.so.conf.d                    vmware-tools
legal                           vsftpd.conf
libao.conf                      vtrgb
libaudit.conf                   vulkan
libblockdev                     wgetrc
libnl-3                         whoopsie
libpaper.d                      wireshark
libreoffice                     wpa_supplicant
locale.alias                    X11
locale.gen                      xattr.conf
localtime                       xdg
logcheck                        xml
login.defs                      zsh
logrotate.conf                  zsh_command_not_found
logrotate.d                     zzz
lsb-release
[02/10/22]seed@VM:/etc$ gcc -o capleak capleak.c
gcc: error: capleak.c: No such file or directory
gcc: fatal error: no input files
compilation terminated.
[02/10/22]seed@VM:/etc$ gcc -o cap_leak cap_leak.c
/usr/bin/ld: cannot open output file cap_leak: Permi
n denied
collect2: error: ld returned 1 exit status
[02/10/22]seed@VM:/etc$ sudo gcc -o cap_leak cap_lea
[02/10/22]seed@VM:/etc$ chmod u+s cap_leak
chmod: changing permissions of 'cap_leak': Operatio
 permitted
[02/10/22]seed@VM:/etc$ sudo chmod u+s cap_leak
[02/10/22]seed@VM:/etc$ ls -al zzz cap_leak
-rwsr-xr-x 1 root root 17008 Feb 10 12:30 cap_leak
-rw-r--r-- 1 root root     0 Feb 10 12:22 zzz
[02/10/22]seed@VM:/etc$ ./cap_leak
fd is 3
$ ca
```

The contents of the file have been modified and the reason is that file zzz is opened before the UID is set. It could be fixed if we simply moved setuid(getuid()) in front of the open() function.