Rong Jun Leong 30152623
CSCI 157 Assignment 2

Section 1

1. Alice runs a Set-UID program that is owned by Bob. The program tries to read from /tmp/x, which is readable to Alice, but not to anybody else. Can this program successfully read from the file?
No. The Set-UID program owned by Bob will try to read /tmp/x as Bob, and not Alice. Since only Alice can access that file, the program cannot read the file successfully

2. Both system() and execve() can be used to execute external programs. Why is system() unsafe while execve() is safe?
If the program to be executed is determined by user input, and system() uses /bin/sh internally, then a user could input a dummy program, followed by a semicolon and a malicious command. This would cause the malicious command to be executed as well. However, with execve(), the entire input string is treated as a single program, so the malicious command would not be executed.

3. The superuser wants to give Alice a permission to view all the files in the system using the
more command. He does the following:
$ cp /bin/more /tmp/mymore
$ sudo chown root /tmp/mymore
$ sudo chmod 4700 /tmp/mymore

Basically, the above commands turns /tmp/mymore into a Set-UID program. Right now, because the permission is set to 4700, other users cannot execute the program. The superuser uses another command (now shown) to grant the execution permission only to
Alice. We are not assuming that Alice is completely trusted. It is OK if Alice can only read other people's files, but it is not OK if Alice can gain any privilege beyond that, such as writing to other people's files. Read the manual of the "more" program and find out what Alice can do to gain more privilege.

Using the v command in more when it is running as root can give you unlimited access to all files on the system, so use it with caution.

4. Assume that you have a file that you would allow other users to read, only if a user's ID is smaller than 1000. Describe how you can actually achieve this.
We can use "groupadd -r under1000" to create a new group with less than a 1000 users and use "cat /etc/passwd | cut -d : -f1,3" to list out all the user with their unique ID

5. What is the difference between environment variables and shell variables?
The difference is environment variable is globally available in the program and its child programs but a shell variable is only available in the current shell in which they were defined.

6. The followings are two different ways to print out environment variables. Describe their differences:
$ /usr/bin/env
$ /usr/bin/strings /proc/$$/environ
The $ /usr/bin/env is a way to print out the environment variable of the current process but the $ /usr/bin/strings /proc/$$/environ will print out all the environment variables whether they had been modified or inherited

7. In Linux, many environment variables are ignored by the dynamic linker if the program to be executed is a Set-UID program. Two such examples are LD PRELOAD and LD LIBRARY PATH. Read the manual of ld-linux (https://linux.die.net/man/8/ld-linux) and explain why the following environment variables are also ignored:
• LD AUDIT
• LD DEBUG OUTPUT

These two variables are ignored because LD_AUDIT allow dynamic linkers auditing which can intercept and change the program behaviour and LD DEBUG OUTPUT can cause information leaking which is a threat to system security.

Section2

Task1

1. Step 1

After running gcc myprintenv.c, I saved the child process as cout.txt and parent process as pout.txt

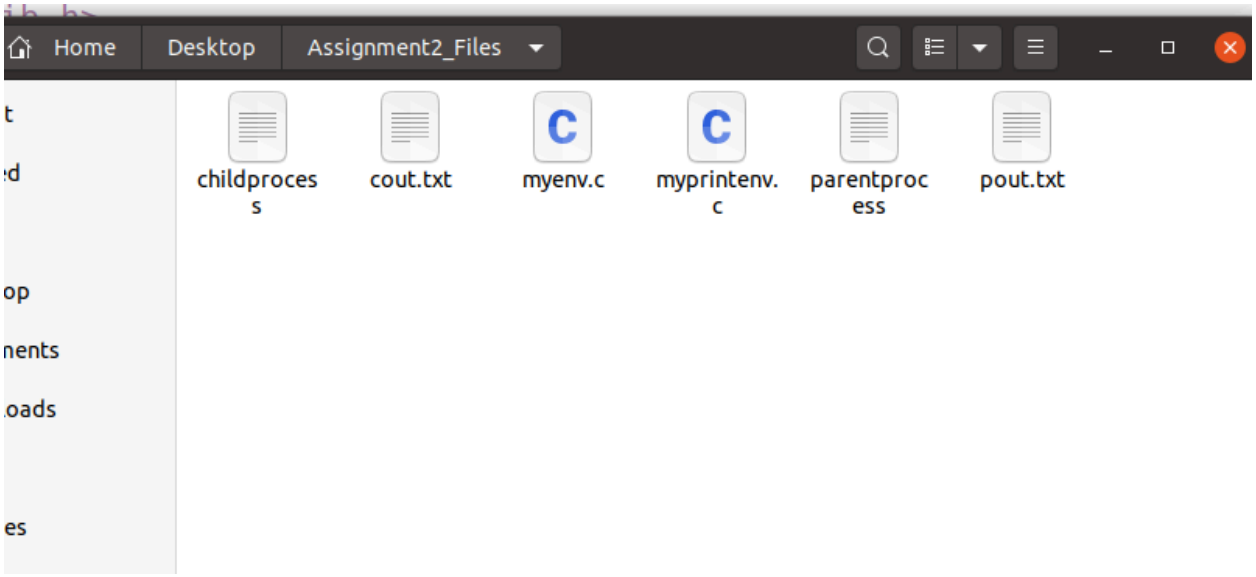
myprintenv.c

```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);
  }
}
```


Home
Desktop
Assignment2_Files
childproces
s
cout.txt
myenv.c
myprintenv.
c
parentproc
ess
pout.txt

Step 2

```
This file "/home/seed/Desktop/Assignment2_Files/myprintenv.c" is already open in another window.
Do you want to edit it anyway?
```

```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);
  }
}
```

Screenshot below is the output of child process

myprintenv.c | cout.txt | pout.txt

```
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/2459,unix/VM:/tmp/.ICE-unix/2459
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=2423
GTK_MODULES=gail:atk-bridge
PWD=/home/seed/Desktop/Assignment2_Files
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=00:su=37;41:sg=30;43:c
  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=0
XDG_CURRENT_DESKTOP=ubuntu:GNOME
VTE_VERSION=6003
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/78f1574d_ee1c_4041_9ae2_736ed1ea1c22
INVOCATION_ID=1bcd9a90143e4127bb5fea1522b3a9c6
MANAGERPID=2194
GJS_DEBUG_OUTPUT=stderr
```

Screenshot below is the output of parent process

myprintenv.c | cout.txt | pout.txt

```
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/2459,unix/VM:/tmp/.ICE-unix/2459
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=2423
GTK_MODULES=gail:atk-bridge
PWD=/home/seed/Desktop/Assignment2_Files
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=00:su=37;41:sg=30;4
  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:*.mjpe
XDG_CURRENT_DESKTOP=ubuntu:GNOME
VTE_VERSION=6003
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/78f1574d_ee1c_4041_9ae2_736ed1ea1c22
INVOCATION_ID=1bcd9a90143e4127bb5fea1522b3a9c6
MANAGERPID=2194
GJS_DEBUG_OUTPUT=stderr
```

Based on my observation, both processes have the same shell, path ,user information etc but the one thing that is different is the invocation ID.

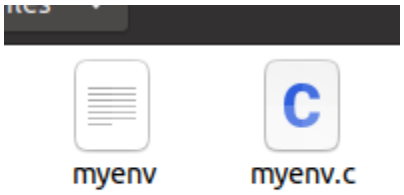
Step 3

```
[03/11/24]seed@VM:~/.../Assignment2_Files$ diff cout.txt pout.txt
48c48
< _=./childprocess
---
> _=./parentprocess
[03/11/24]seed@VM:~/.../Assignment2_Files$
```

In conclusion, we can say that the programs are different but operate in the same environment and the environment variables are passed from the parent process to the child process which are identical.

Task 2

Step 1.



After compiling the program, there's an output file named myenv but after i saved the output file to another txt file, it's empty inside.

Step 2

```c
#include <unistd.h>

extern char **environ;

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

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

  //Step 1:
  //execve("/usr/bin/env", argv, NULL);

  //Step 2:
  execve("/usr/bin/env", argv, environ);

  return 0 ;
}

```

```
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/2459,unix/VM:/tmp/.ICE-unix/2459
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=2423
GTK_MODULES=gail:atk-bridge
PWD=/home/seed/Desktop/Assignment2_Files
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=00:
  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:*.m
XDG_CURRENT_DESKTOP=ubuntu:GNOME
VTE_VERSION=6003
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/78f1574d_ee1c_4041_9ae2_736ed1ea1c22
INVOCATION_ID=1bcd9a90143e4127bb5fea1522b3a9c6
MANAGERPID=2194
GJS_DEBUG_OUTPUT=stderr
```

After compiling, the whole process was displayed after I saved the output file into 2.txt.

Step 3

In conclusion, the environment variable is inherit by the new program automatically, the environ array is passed to the execve() as the third argument which new program can access it like the original program