

# CSCI 191T – Computer Security

**Assignment 1: SetUID and Env.Variables      Due:9/18/2022 11:59 PM**

**Name: Saishnu Ramesh Kumar (300758706)**

## **Instructions:**

There are two sections in this assignment. For Section 1, answer all questions completely. For Section 2, execute the code (using files from Assignment1\_Files.zip) and answer all questions.

**The final submission is a single report (PDF file).** You need to submit a detailed report, with:

Section 1 - Detailed answers and observations.

Section 2 - Screenshots, to describe what you have done and what you have observed. You also need to provide explanation to the observations that are interesting or surprising. Also list the important code snippets followed by explanation. **Simply attaching code or screenshot without any explanation will not receive credits.**

## **Section 1**

**10 points each.**

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?

In this case, the program will not be able to read from the file. Since the Set-UID program was originally owned by Bob and is being attempted to run on Alice's machine, the program will not have the same privileges as being on Bob's machine. Therefore, it is unable to read from /tmp/x on Alice's machine as it does not have access to it.

2. Both system() and execve() can be used to execute external programs. Why is system() unsafe while execve() is safe?

The reason why system() is unsafe is because the user has the ability to execute programs with user inputs as it runs through /bin/sh. Whereas in comparison to execve(), this is the safer option as the entire program is treated as a executable and will fail to execute when attempted.

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.

If Alice views a file using more, she would be able to view the file through the v command to view the file in an editor. Alternatively, if more is run through the root, the editor will be opened using the root as well which then allows Alice access to all the files located within the system.

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 take the root to be the user. So in this case, the user will create a group with all other users that have an id that is less than 1000. The /etc/passwd will be able to list all users and their user ID and you would be able to check which users have an ID that is smaller than 1000, you would then be able to give these specific users access to the file and only give them the permission to read the file. They will not be able to edit or modify the file in any way.

5. What is the difference between environment variables and shell variables?

The difference between environment variables and shell variables is that environment variables are valid throughout the entire system. Whereas shell variables are only valid in the current shell that is active and will be terminated once the shell is closed.

6. The followings are two different ways to print out environment variables. Describe their differences:

\$ /usr/bin/env → This command displays all the environment variables in the directory.

\$ /usr/bin/strings /proc/\$\$/environ → This command gets the printable characters in the file and finds the process number with proc/\$\$ where \$\$ is used for the current shell number and the environ helps print the environment variable of the shell name environ.

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 AUDIT gets ignored because these objects can be used to audit the dynamic linker operation. The linker ignores this as the list may cause certain unwanted programs to be running and could possibly modify the operation of the linker. It can result to vulnerability in the data.

- LD DEBUG OUTPUT

- LD DEBUG OUTPUT is ignored as it can result in data integrity problems if the program is not checked in the normal environment, it could cause issues within the system that is being run on.

## Section 2

### 15 Points each

1. In Unix, fork() creates a new process by duplicating the calling process. The new process, referred to as the child, is an exact duplicate of the calling process, referred to as the parent; however, several things are not inherited by the child (see the manual of fork() by typing the following command: man fork). In this task, we would like to know whether the parent's environment variables are inherited by the child process or not.

Step 1. Compile and run the following program, and describe your observation.

The program "myprintenv.c", can be found in the "Assignment1\_Files.zip"; it can be compiled using "gcc myprintenv.c", which will generate a binary called a.out. Run it and save the output into a file using "a.out > file".

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

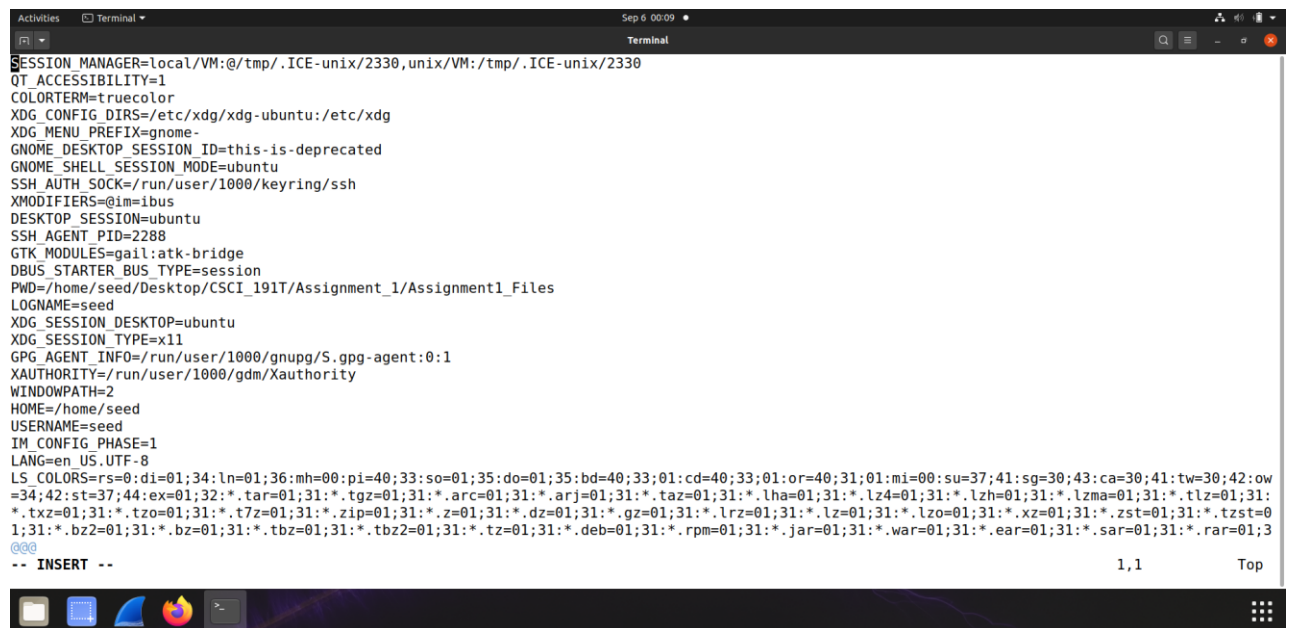
extern char **environ;
void printenv()
{
    int i = 0;
    while (environ[i] != NULL) {
        printf("%s\n", environ[i]);
        i++;
    }
}

void main()
{
    pid_t childPid;
    switch(childPid = fork()) {
        case 0: /* child process */
            printenv();           ①
            exit(0);
        default: /* parent process */
            //printenv();         ②
            exit(0);
    }
}
```

Step 2. Now comment out the `printenv()` statement in the child process case (Line ①), and uncomment the `printenv()` statement in the parent process case (Line ②). Compile and run the code again, and describe your observation. Save the output in another file.

Step 3. Compare the difference of these two files using the *diff* command. Describe your conclusion

Step 1 Observation: After getting the `a.out` file I was confused on how to run it but manage to run it using `./a.out` and I got the output of a lot of information relating to the machine and the files and folders where the `a.out` file is stored.

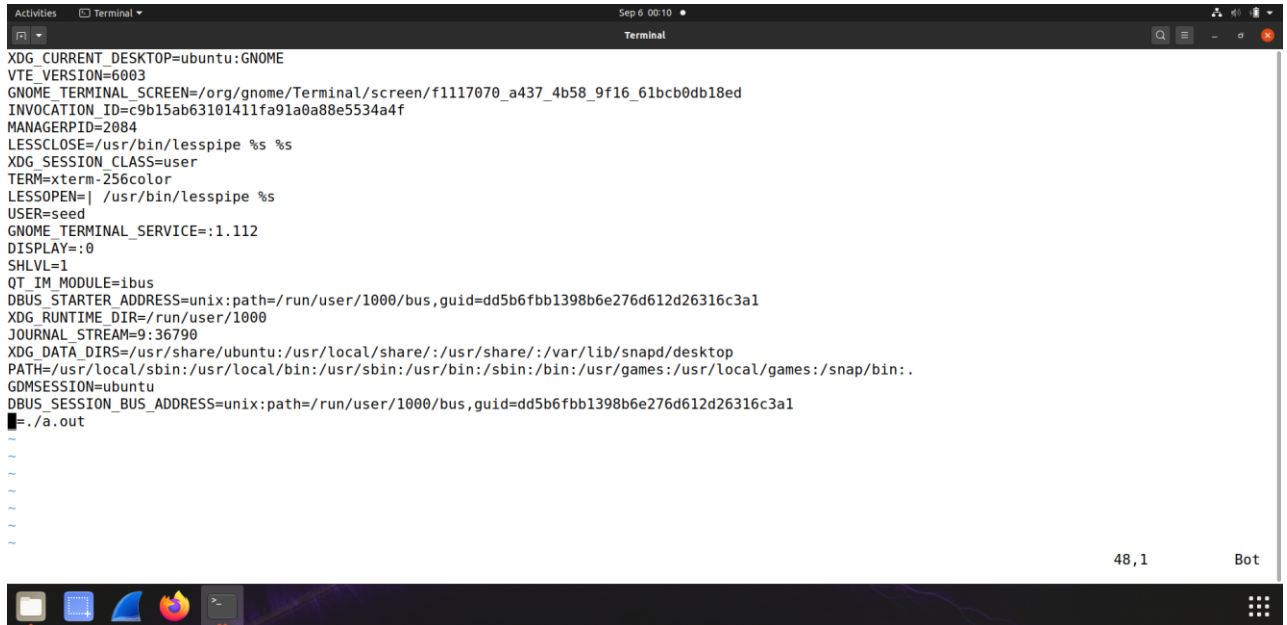


```
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/2330,unix/VM:/tmp/.ICE-unix/2330
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
SSH_AGENT_PID=2288
GTK_MODULES=gail:atk-bridge
DBUS_STARTER_BUS_TYPE=session
PWD=/home/seed/Desktop/CSCI_191T/Assignment_1/Assignment1_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
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: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:*.tztst=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;3
-- INSERT --
```

```
Activities Terminal Sep 6 00:10
XDG_CURRENT_DESKTOP=ubuntu:GNOME
VTE_VERSION=6003
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/f1117070_a437_4b58_9f16_61bcb0db18ed
INVOCATION_ID=c9b15ab63101411fa91a0a88e5534a4f
MANAGERPID=2084
LESSCLOSE=/usr/bin/lesspipe %s %s
XDG_SESSION_CLASS=user
TERM=xterm-256color
LESSOPEN=| /usr/bin/lesspipe %s
USER=seed
GNOME_TERMINAL_SERVICE=:1.112
DISPLAY=:0
SHLVL=1
QT_IM_MODULE=ibus
DBUS_STARTER_ADDRESS=unix:path=/run/user/1000/bus,guid=dd5b6fbb1398b6e276d612d26316c3a1
XDG_RUNTIME_DIR=/run/user/1000
JOURNAL_STREAM=9:36790
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/local/games:/snap/bin:
GDMSESSION=ubuntu
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus,guid=dd5b6fbb1398b6e276d612d26316c3a1
_=/a.out
-- INSERT --
40,1 Bot
```

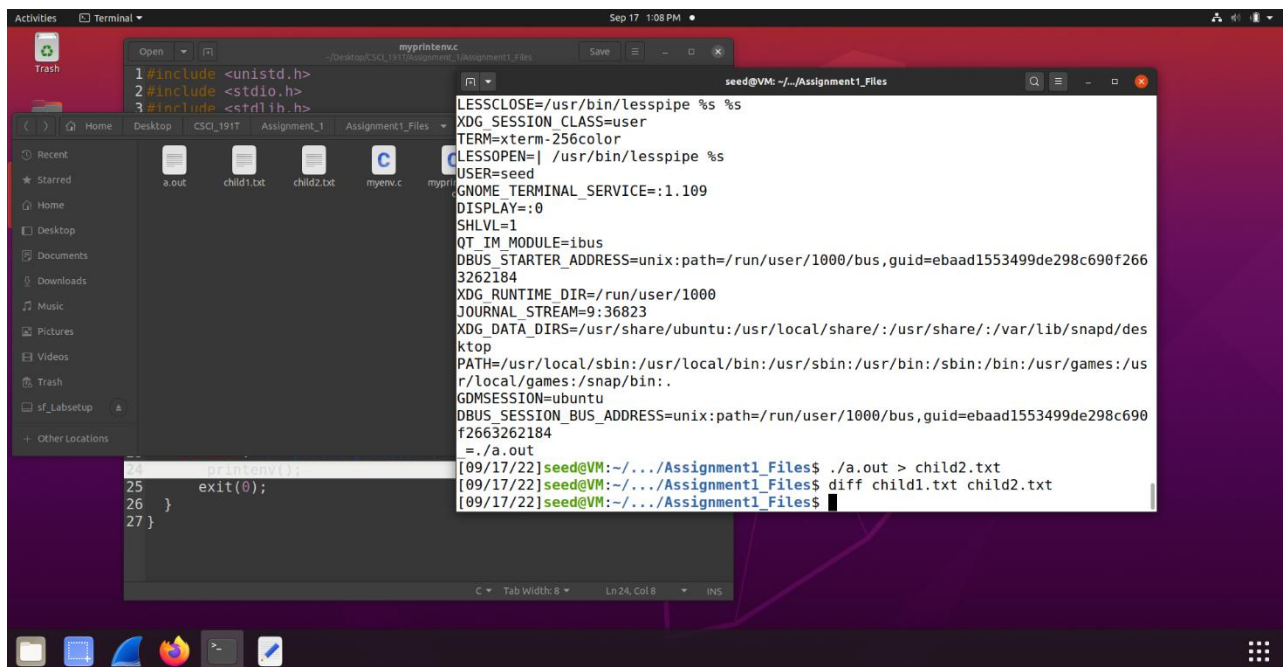
Step 2 Observation: Much of the information by simply looking through the information obtained looks very similar to the result in Step 1.

```
Activities Terminal Sep 6 00:10
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/2330,unix/VM:/tmp/.ICE-unix/2330
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=2288
GTK_MODULES=gail:atk-bridge
DBUS_STARTER_BUS_TYPE=session
PWD=/home/seed/Desktop/CSCI_191T/Assignment_1/Assignment1_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
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: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:*.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:*.tztst=0
@@@
1,1 Top
```

A terminal window titled 'Terminal' showing a list of environment variables. The variables include XDG\_CURRENT\_DESKTOP, VTE\_VERSION, GNOME\_TERMINAL\_SCREEN, INVOCATION\_ID, MANAGERPID, LESSCLOSE, XDG\_SESSION\_CLASS, TERM, LESSOPEN, USER, GNOME\_TERMINAL\_SERVICE, DISPLAY, SHLVL, QT\_IM\_MODULE, DBUS\_STARTER\_ADDRESS, XDG\_RUNTIME\_DIR, JOURNAL\_STREAM, XDG\_DATA\_DIRS, PATH, GDMSESSION, and DBUS\_SESSION\_BUS\_ADDRESS. The prompt is './a.out'.

```
XDG_CURRENT_DESKTOP=ubuntu:GNOME
VTE_VERSION=6003
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/f1117070_a437_4b58_9f16_61bcb0db18ed
INVOCATION_ID=c9b15ab63101411fa91a0a88e5534a4f
MANAGERPID=2084
LESSCLOSE=/usr/bin/lesspipe %s %s
XDG_SESSION_CLASS=user
TERM=xterm-256color
LESSOPEN=| /usr/bin/lesspipe %s
USER=seed
GNOME_TERMINAL_SERVICE=:1.112
DISPLAY=:0
SHLVL=1
QT_IM_MODULE=ibus
DBUS_STARTER_ADDRESS=unix:path=/run/user/1000/bus,guid=dd5b6fbb1398b6e276d612d26316c3a1
XDG_RUNTIME_DIR=/run/user/1000
JOURNAL_STREAM=9:36790
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/local/games:/snap/bin:.
GDMSESSION=ubuntu
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus,guid=dd5b6fbb1398b6e276d612d26316c3a1
./a.out
```

Step 3 Observation: When attempting to use the diff command, I was unable to see if there is a difference between the two files because I did not seem to get any result from the terminal, as seen in the screenshot below.

A terminal window titled 'seed@VM: ~/Assignment1\_Files' showing the execution of a program and the use of the diff command. The program 'myprintm.c' is compiled and run, creating 'child1.txt' and 'child2.txt'. The diff command is used to compare these two files, but it returns no output, indicating they are identical.

```
1 #include <unistd.h>
2 #include <stdio.h>
3 #include <stdlib.h>

...

25 exit(0);
26 }
27 }

[09/17/22]seed@VM:~/Assignment1_Files$ ./a.out > child2.txt
[09/17/22]seed@VM:~/Assignment1_Files$ diff child1.txt child2.txt
[09/17/22]seed@VM:~/Assignment1_Files$
```

2. In this task, we study how environment variables are affected when a new program is executed via `execve()`. The function `execve()` calls a system call to load a new command and execute it; this function never returns. No new process is created; instead, the calling process's text, data, bss, and stack are overwritten by that of the program loaded. Essentially, `execve()` runs the new program inside the calling process. We are interested in

what happens to the environment variables; are they automatically inherited by the new program?

Step 1. Compile and run the following program, and describe your observation.

The program “myenv.c” can be found in the “Assignment1\_Files.zip”; it can be compiled using "gcc myenv.c", This program simply executes a program called /usr/bin/env, which prints out the environment variables of the current process.

```
#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 ;
}
```

Step 2. Change the invocation of execve() in Line ① to the following and describe your observation.

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

Step 3. Describe your conclusion regarding how the new program gets its environment variable.

Step 1 Observation: I tried to compile and run the program and when I do, I get a Permission denied message in the terminal.

Step 2 Observation: The information obtained is very similar to the information as before for the previous section. The terminal/program was able to also detect the environment this was being run on which I found interesting.



```
Activities Terminal Sep 6 00:38
seed@VM: ~/Assignment1_Files
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus,guid=dd5b6fbb1398b6e276d612d26316c3a1
./a.out
[09/06/22]seed@VM:~/Assignment1_Files$ gcc myenv.c
[09/06/22]seed@VM:~/Assignment1_Files$ ./a.out
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/2330,unix/VM:/tmp/.ICE-unix/2330
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=2288
GTK_MODULES=gail:atk-bridge
DBUS_STARTER_BUS_TYPE=session
PWD=/home/seed/Desktop/CSCI_191T/Assignment_1/Assignment1_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
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=04;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:*.lzh=01;31:*.lzm=01;31:*.tlz=01;31:
1:*.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=01;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:*.mov=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;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.a
ac=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=ubuntu:GNOME
VTE_VERSION=6003
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/bd809287_1460_414d_9424_599b03aa945d
INVOCATION_ID=c9b15ab63101411fa91a0a88e5534a4f
MANAGERPID=2084
LESSCLOSE=/usr/bin/lesspipe %s %s
XDG_SESSION_CLASS=user
TERM=xterm-256color
LESSOPEN=| /usr/bin/lesspipe %s
USER=seed
GNOME_TERMINAL_SERVICE=:1.140
DISPLAY=:0
SHLVL=1
QT_IM_MODULE=ibus
DBUS_STARTER_ADDRESS=unix:path=/run/user/1000/bus,guid=dd5b6fbb1398b6e276d612d26316c3a1
XDG_RUNTIME_DIR=/run/user/1000
JOURNAL_STREAM=9:36790
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/local/games:/snap/bin:
GDMSESSION=ubuntu
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus,guid=dd5b6fbb1398b6e276d612d26316c3a1
./a.out
[09/06/22]seed@VM:~/Assignment1_Files$
```

Step 3 Observation: The new program was able to get the environment variable from the environ keyword that was used. Since the environment variable is basically a variable with a name and value, it will scroll through all the numbers and names it can find to get the right output. When the program was initially set to NULL, the program did not find anything because setting NULL at the end/the beginning will result in nothing.

```
Activities Terminal Sep 6 00:38
seed@VM: ~/Assignment1_Files
1:*.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=01;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:*.mov=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;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.a
ac=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=ubuntu:GNOME
VTE_VERSION=6003
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/bd809287_1460_414d_9424_599b03aa945d
INVOCATION_ID=c9b15ab63101411fa91a0a88e5534a4f
MANAGERPID=2084
LESSCLOSE=/usr/bin/lesspipe %s %s
XDG_SESSION_CLASS=user
TERM=xterm-256color
LESSOPEN=| /usr/bin/lesspipe %s
USER=seed
GNOME_TERMINAL_SERVICE=:1.140
DISPLAY=:0
SHLVL=1
QT_IM_MODULE=ibus
DBUS_STARTER_ADDRESS=unix:path=/run/user/1000/bus,guid=dd5b6fbb1398b6e276d612d26316c3a1
XDG_RUNTIME_DIR=/run/user/1000
JOURNAL_STREAM=9:36790
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/local/games:/snap/bin:
GDMSESSION=ubuntu
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus,guid=dd5b6fbb1398b6e276d612d26316c3a1
./a.out
[09/06/22]seed@VM:~/Assignment1_Files$
```