网络空间安全实验基础实验报告

姓名：韩永欣　学号：57119107 报告日期：2021.7.6

1. 实验内容

SEED Labs – Environment Variable and Set-UID Program Lab

2. 实验目的

理解环境变量是如何影响程序和系统行为，环境变量是一组动态命名的变量，它们可以影响进程在计算机上的行为。

3. 主要数据结构及其说明

Task1

打印环境变量

```
[07/06/21]seed@VM:~$ printenv
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/1469,unix/VM:/tmp/.ICE-unix/14
69
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=1424
GTK_MODULES=gail:atk-bridge
PWD=/home/seed
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:ca=30;41:tw=30;4
2: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:*.tl
z=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=
```

打印环境变量 PWD

```
[07/06/21]seed@VM:~$ printenv PWD
/home/seed
```

```
[07/06/21]seed@VM:~$ 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/90f54d82_9
358_4ce8_8c4e_e526c51cbd45"
declare -x GNOME_TERMINAL_SERVICE=":1.86"
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="45472d2ceff64d479f020c69a8993cca"
declare -x JOURNAL_STREAM="9:32749"
declare -x LANG="en_US.UTF-8"
declare -x LESSCLOSE="/usr/bin/lesspipe %s %s"
declare -x LESSOPEN="| /usr/bin/lesspipe %s"
declare -x LOGNAME="seed"
declare -x 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=3
0;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:*.lzm
a=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:*.jar=01;31:*.war=01;31:*.ear=
```

设置环境变量 demo 为 H0me/bin
打印环境变量 demo

```
[07/06/21]seed@VM:~$ export demo="Home/bin"
[07/06/21]seed@VM:~$ echo $demo
Home/bin
[07/06/21]seed@VM:~$
```

取消设置环境变量 demo 并打印

```
[07/06/21]seed@VM:~$ unset demo
[07/06/21]seed@VM:~$ echo $demo
```

Task 2

源代码：

```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);
} }
```

编译运行

```
[03/21/21]seed@VM:~/.../Experiment1$ gcc -o task2 task2.c
[03/21/21]seed@VM:~/.../Experiment1$ ./task2
XDG_VTNR=7
ORBIT_SOCKETDIR=/tmp/orbit-seed
XDG_SESSION_ID=c1
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed
IBUS_DISABLE_SNOOPER=1
TERMINATOR_UUID=urn:uuid:8c2d0615-51f5-4e10-9fd6-9e0b4148ceee
CLUTTER_IM_MODULE=xim
SESSION=ubuntu
GIO_LAUNCHED_DESKTOP_FILE_PID=10354
ANDROID_HOME=/home/seed/android/android-sdk-linux
GPG_AGENT_INFO=/home/seed/.gnupg/S.gpg-agent:0:1
TERM=xterm
SHELL=/bin/bash
DERBY_HOME=/usr/lib/jvm/java-8-oracle/db
QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1
LD_PRELOAD=/home/seed/lib/boost/libboost_program_options.so.1.64.0:/home/seed/li
b/boost/libboost_filesystem.so.1.64.0:/home/seed/lib/boost/libboost_system.so.1.
64.0
WINDOWID=60817412
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1122
GNOME_KEYRING_CONTROL=
GTK_MODULES=gail:atk-bridge:unity-gtk-module
USER=seed
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;4
4: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:*.t7
z=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=0
```

结果 task2 保存在 child 中

将子进程的 printenv()注释，父进程的 printenv()取消注释，编译运行

结果保存在 task22 中

```
[03/21/21]seed@VM:~/.../Experiment1$ task2 > child
[03/21/21]seed@VM:~/.../Experiment1$ gcc -o task22 task2.c
[03/21/21]seed@VM:~/.../Experiment1$ ./task22
```

```
[03/21/21]seed@VM:~/.../Experiment1$ gcc -o task22 task2.c
[03/21/21]seed@VM:~/.../Experiment1$ ./task22
XDG_VTNR=7
ORBIT_SOCKETDIR=/tmp/orbit-seed
XDG_SESSION_ID=c1
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed
IBUS_DISABLE_SNOOPER=1
TERMINATOR_UUID=urn:uuid:8c2d0615-51f5-4e10-9fd6-9e0b4148ceee
CLUTTER_IM_MODULE=xim
SESSION=ubuntu
GIO_LAUNCHED_DESKTOP_FILE_PID=10354
ANDROID_HOME=/home/seed/android/android-sdk-linux
GPG_AGENT_INFO=/home/seed/.gnupg/S.gpg-agent:0:1
TERM=xterm
SHELL=/bin/bash
DERBY_HOME=/usr/lib/jvm/java-8-oracle/db
QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1
LD_PRELOAD=/home/seed/lib/boost/libboost_program_options.so.1.64.0:/home/seed/lib/boost/libb
oost_filesystem.so.1.64.0:/home/seed/lib/boost/libboost_system.so.1.64.0
WINDOWID=60817412
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1122
GNOME_KEYRING_CONTROL=
GTK_MODULES=gail:atk-bridge:unity-gtk-module
USER=seed
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
```

将 task22 保存在 parent 中；用 diff 比较 child 和 parent 的区别

```
[03/21/21]seed@VM:~/.../Experiment1$ task22 > parent
[03/21/21]seed@VM:~/.../Experiment1$ diff child parent
75c75
< _=./task2
---
> _=./task22
[03/21/21]seed@VM:~/.../Experiment1$
```

结论：两次输出的环境变量完全相同，使用 fork()系统调用生成的子进程继承父进程的全部环境变量。

Task3

源代码：

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

编译运行，输出为空

```
[03/21/21]seed@VM:~/.../Experiment1$ ./task3
[03/21/21]seed@VM:~/.../Experiment1$
```

将 NULL 改为 environ，编译运行，如图：

```
[03/21/21]seed@VM:~/.../Experiment1$ ./task3_1
XDG_VTNR=7
ORBIT_SOCKETDIR=/tmp/orbit-seed
XDG_SESSION_ID=c1
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed
IBUS_DISABLE_SNOOPER=1
TERMINATOR_UUID=urn:uuid:8c2d0615-51f5-4e10-9fd6-9e0b4148ceee
CLUTTER_IM_MODULE=xim
SESSION=ubuntu
GIO_LAUNCHED_DESKTOP_FILE_PID=10354
ANDROID_HOME=/home/seed/android/android-sdk-linux
GPG_AGENT_INFO=/home/seed/.gnupg/S.gpg-agent:0:1
TERM=xterm
SHELL=/bin/bash
DERBY_HOME=/usr/lib/jvm/java-8-oracle/db
QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1
LD_PRELOAD=/home/seed/lib/boost/libboost_program_options.so.1.64.0:/home/seed/lib/boost/libb
oost_filesystem.so.1.64.0:/home/seed/lib/boost/libboost_system.so.1.64.0
WINDOWID=60817412
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1122
GNOME_KEYRING_CONTROL=
GTK_MODULES=gail:atk-bridge:unity-gtk-module
USER=seed
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
```

输出为当前进程的环境变量

结论：

int execve(const char *filename, char *const argv[], char *const envp[])

filename 指向子进程路径，argv 指向子进程参数，envp 指向子进程环境变量。

父进程通过 environ 传递参数。

进程在被初始化时获取环境变量的两种方式：fork()、execve()。

Task4

源代码：

```c
#include <stdio.h>
#include <stdlib.h>
int main()
{
system("/usr/bin/env");
return 0 ;
}
```

system()通过 execl()调用/bin/sh 执行，execl()调用 execve()并传递环境变量

编译运行，输出为当前进程的环境变量。

```
[03/21/21]seed@VM:~/.../Experiment1$ gcc -o task4 task4.c
[03/21/21]seed@VM:~/.../Experiment1$ ./task4
LESSOPEN=| /usr/bin/lesspipe %s
GNOME_KEYRING_PID=
USER=seed
LANGUAGE=en_US
UPSTART_INSTANCE=
J2SDKDIR=/usr/lib/jvm/java-8-oracle
XDG_SEAT=seat0
SESSION=ubuntu
XDG_SESSION_TYPE=x11
COMPIZ_CONFIG_PROFILE=ubuntu-lowgfx
ORBIT_SOCKETDIR=/tmp/orbit-seed
LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stag
e/lib:
SHLVL=1
LIBGL_ALWAYS_SOFTWARE=1
J2REDIR=/usr/lib/jvm/java-8-oracle/jre
HOME=/home/seed
QT4_IM_MODULE=xim
OLDPWD=/home/seed/Desktop
DESKTOP_SESSION=ubuntu
GIO_LAUNCHED_DESKTOP_FILE=/usr/share/applications/terminator.desktop
QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1
GTK_MODULES=gail:atk-bridge:unity-gtk-module
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
INSTANCE=
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-SsU2dmMk0a
GIO_LAUNCHED_DESKTOP_FILE_PID=10354
```

Task5

源代码：

```c
#include <stdio.h>
#include <stdlib.h>
extern char **environ;
void main()
{
int i = 0;
while (environ[i] != NULL) {
printf("%s\n", environ[i]);
i++;
} }
```

编译运行，输出所有环境变量：

```
[03/21/21]seed@VM:~/.../Experiment1$ gcc -o task5 task5.c
[03/21/21]seed@VM:~/.../Experiment1$ ./task5
XDG_VTNR=7
ORBIT_SOCKETDIR=/tmp/orbit-seed
XDG_SESSION_ID=c1
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed
IBUS_DISABLE_SNOOPER=1
TERMINATOR_UUID=urn:uuid:8c2d0615-51f5-4e10-9fd6-9e0b4148ceee
CLUTTER_IM_MODULE=xim
SESSION=ubuntu
GIO_LAUNCHED_DESKTOP_FILE_PID=10354
ANDROID_HOME=/home/seed/android/android-sdk-linux
GPG_AGENT_INFO=/home/seed/.gnupg/S.gpg-agent:0:1
TERM=xterm
SHELL=/bin/bash
DERBY_HOME=/usr/lib/jvm/java-8-oracle/db
QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1
LD_PRELOAD=/home/seed/lib/boost/libboost_program_options.so.1.64.0:/home/seed/lib/boost/libb
oost_filesystem.so.1.64.0:/home/seed/lib/boost/libboost_system.so.1.64.0
WINDOWID=60817412
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1122
GNOME_KEYRING_CONTROL=
GTK_MODULES=gail:atk-bridge:unity-gtk-module
USER=seed
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
```

编译，更改拥有者为root，并将它变为Set-UID程序，将环境变量ANY_NAME设置为hepburn

```
[03/21/21]seed@VM:~/.../Experiment1$ sudo chown root task5
[03/21/21]seed@VM:~/.../Experiment1$ sudo chmod 4755 task5
[03/21/21]seed@VM:~/.../Experiment1$ printenv PATH
/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/
local/games:.:/snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db/bin:/us
r/lib/jvm/java-8-oracle/jre/bin:/home/seed/android/android-sdk-linux/tools:/home/seed/androi
d/android-sdk-linux/platform-tools:/home/seed/android/android-ndk/android-ndk-r8d:/home/seed
/.local/bin
[03/21/21]seed@VM:~/.../Experiment1$ printenv LD_LIBRARY_PATH
/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:
[03/21/21]seed@VM:~/.../Experiment1$ printenv ANY_NAME
[03/21/21]seed@VM:~/.../Experiment1$ export ANY_NAME=hepburn
```

运行，发现输出的环境变量中出现了设置的环境变量 ANY_NAME，如图

```
LOGNAME=seed
ANY_NAME=hepburn
COMPIZ_BIN_PATH=/usr/bin/
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-SsU2dmMk0a
```

结论：

shell 程序执行时会为每个环境变量创建名称和值都相同的 shell 变量。而 Bash 中，用户自定义且导出（export）的 shell 变量会被传递给子进程。

Bash 中，两种类型的 shell 变量会被传递给子进程：从环境变量复制得到的 shell 变量、用户导出（export）的 shell 变量。

Task6

源代码：

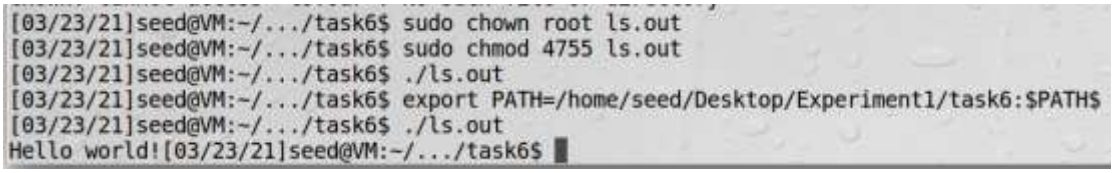
```c
#include <stdio.h>
void sleep (int s)
{ /* If this is invoked by a privileged program,
you can do damages here! */
printf("I am not sleeping!\n");
}
```

```c
/* myprog.c */
int main()
{
sleep(1);
return 0;
}
```

编译运行，取消 Ubuntu 16.04 保护机制

```
[03/23/21]seed@VM:~/.../task6$ gcc -o ls.out ls.c
[03/23/21]seed@VM:~/.../task6$ ./ls.out
Hello world![03/23/21]seed@VM:~/.../task6$ sudo rm /bin/sh
[03/23/21]seed@VM:~/.../task6$ sudo ln -s /bin/zsh/ /bin/sh
```

运行，输出为执行 ls 的结果



结论：

shell 程序执行命令时，如果没有提供命令的具体位置，shell 程序将使用 PATH 环境变量搜索命令。system()将环境变量传递给子进程，通过修改 PATH 环境变量，程序会先搜索当前目录，调用预先准备好的 ls 程序。

4. 实验体会

本次实验是网络空间安全实验基础的第一次实验，所以进行过程困难较多，但在同学和老师的帮助下顺利完成，并理解了环境变量影响程序和系统行为的过程，收获颇多。