

Chapter 1

1.姓名：唐翠霜

2.学号：57119111

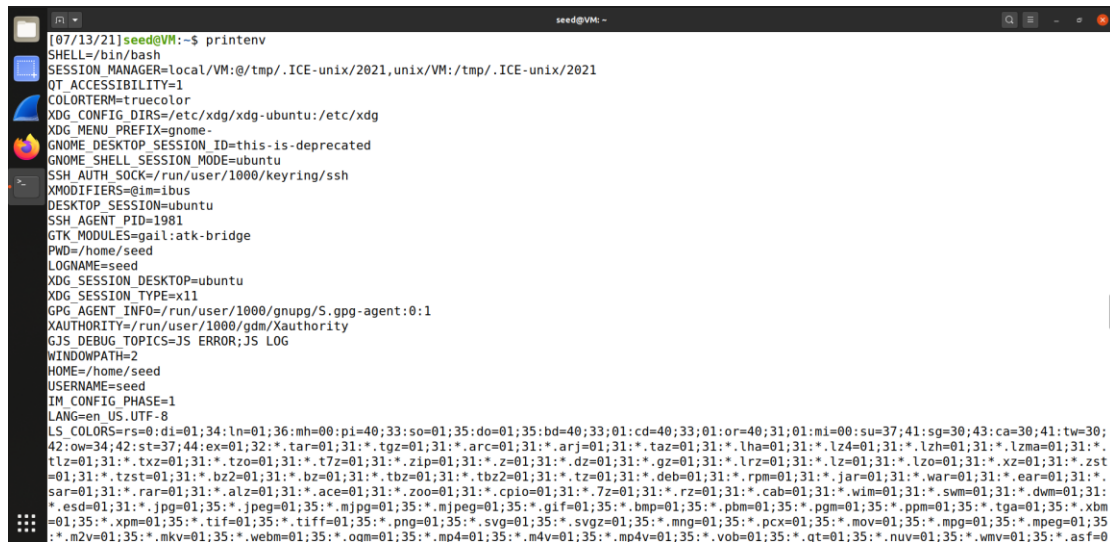
3.实验内容：环境变量与特权程序

4.实验时间：2021.7.13

5.实验过程：

Task1: Manipulating Environment Variables

1) Use `printenv` or `env` command to print out the environment variables.

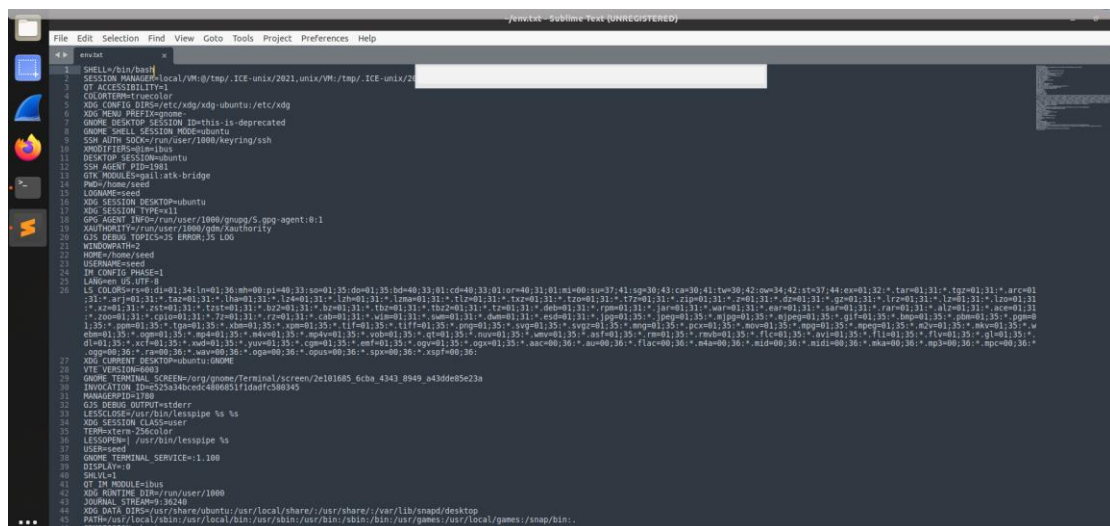


```
[07/13/21]seed@VM:~$ printenv
SHELL=/bin/bash
SESSION_MANAGER=local/VM:/tmp/.ICE-unix/2021,unix/VM:/tmp/.ICE-unix/2021
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
SSH_AGENT_PID=1981
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;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:*.lзма=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.d=01;31:*.gz=01;31:*.lrz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.t=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;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:*.png=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=0
```

```
[07/13/21]seed@VM:~$ env > env.txt
```

```
[07/13/21]seed@VM:~$ subl env.txt
```

直接 `env` 输出的环境变量太多了一屏放不下，可用以上方式放在一个 `sublime` 文本里面看



```
File Edit Selection Find View Goto Tools Project Preferences Help
1 SHELL=/bin/bash
2 SESSION_MANAGER=local/VM:/tmp/.ICE-unix/2021,unix/VM:/tmp/.ICE-unix/2021
3 QT_ACCESSIBILITY=1
4 COLORTERM=truecolor
5 XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
6 XDG_MENU_PREFIX=gnome-
7 GNOME_DESKTOP_SESSION_ID=this-is-deprecated
8 GNOME_SHELL_SESSION_MODE=ubuntu
9 SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
10 XMODIFIERS=@im=ibus
11 DESKTOP_SESSION=ubuntu
12 SSH_AGENT_PID=1981
13 GTK_MODULES=gail:atk-bridge
14 PWD=/home/seed
15 LOGNAME=seed
16 XDG_SESSION_DESKTOP=ubuntu
17 XDG_SESSION_TYPE=x11
18 GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
19 XAUTHORITY=/run/user/1000/gdm/Xauthority
20 GJS_DEBUG_TOPICS=JS ERROR;JS LOG
21 WINDOWPATH=2
22 HOME=/home/seed
23 USERNAME=seed
24 IM_CONFIG_PHASE=1
25 LANG=en_US.UTF-8
26 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:*.lзма=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.d=01;31:*.gz=01;31:*.lrz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.t=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;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:*.png=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=0
27 XDG_CURRENT_DESKTOP=ubuntu:GNOME
28 VTE_VERSION=6603
29 GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/2e101685_6c8a_4343_8949_a33dd85e23a
30 INVOCATION_ID=523a34bcde4806051fddfc580345
31 MANAGERPID=789
32 GJS_DEBUG_OUTPUT=stderr
33 LESSCLOSE=/usr/bin/lesspipe %s %s
34 XDG_SESSION_CLASS=user
35 TERM=xterm-256color
36 LESSOPEN=/usr/bin/lesspipe %s
37 USER=seed
38 GNOME_TERMINAL_SERVICE=:1.100
39 DISPLAY=:0
40 SNL=1
41 QT_IM_MODULE=ibus
42 XDG_RUNTIME_DIR=/run/user/1000
43 XDG_STREAM=936240
44 XDG_DATA_DIRS=/usr/share/ubuntu:/usr/local/share:/usr/share:/usr/lib/napst/desktop
45 PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:
46 COMPOSE_SESSION=ubuntu
```

2) Use “`printenv PWD`” or “`env | grep PWD`” to print out some particular environment variables.

```
[07/13/21]seed@VM:~$ printenv PWD
/home/seed
[07/13/21]seed@VM:~$ env|grep PWD
PWD=/home/seed
[07/13/21]seed@VM:~$
```

3) Use `export` and `unset` to set or unset environment variables.

```
[07/13/21]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 GNOMESESSION="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/2e101685_6cba_4343_8949_a43dde85e23a"
declare -x GNOME_TERMINAL_SERVICE=":1.100"
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="e525a34bcdcd4806851f1dadfc580345"
declare -x JOURNAL_STREAM="9:36240"
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"
```

```
[07/13/21]seed@VM:~$ printenv PWD
/home/seed
[07/13/21]seed@VM:~$ unset PWD
[07/13/21]seed@VM:~$ printenv PWD
[07/13/21]seed@VM:~$
```

Task2: Passing Environment Variables from Parent Process

to Child Process 通过 `fork()`来探究父进程的环境变量是否被子进程所继承

1) compile and run the following program, and describe your observation.

```
1 #include<unistd.h>
2 #include<stdio.h>
3 #include<stdlib.h>
4
5 extern char **environ;
6
7 void printenv()
8 {
9     int i=0;
10    while(environ[i]!=NULL)
11    {
12        printf("%s\n",environ[i]);
13        i++;
14    }
15 }
16
17 void main()
18 {
19     pid_t childPid;
20     switch(childPid=fork())
21     {
22         case 0:/*child process*/
23             printenv();
24             exit(0);
25         default:/*parent process*/
26             //printenv;
27             exit(0);
28     }
29 }
```

```
[07/13/21]seed@VM:~$ cd Desktop/lab1
[07/13/21]seed@VM:~/.../Lab1$ gcc -o task2 task2.c
[07/13/21]seed@VM:~/.../Lab1$ ./task2
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/2021,unix/VM:/tmp/.ICE-unix/2021
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=1981
GTK_MODULES=gail:atk-bridge
PWD=/home/seed/Desktop/lab1
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:cd=40;33: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:*tzst=01;31:*b2=01;31:*bz=01;31:*tbz=01;31:*tbz2=01;31:*tzt=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;31:*cab=01;31:*wim=01;31:*swm=01;31:*dwm=01;31:*esd=01;31:*pq=01;31:*ipq=01;35:*mjq=01;35:*mjeq=01;35:*qif=01;35:*bmp=01;35:*pbm=01;35:*pgm=01;35:*ppm=01;35:*tga=01;35:*xbm=01;35:*
```

```

1#include<unistd.h>
2#include<stdio.h>
3#include<stdlib.h>
4
5extern char **environ;
6
7void printenv()
8{
9    int i=0;
10    while(environ[i]!=NULL)
11    {
12        printf("%s\n",environ[i]);
13        i++;
14    }
15}
16
17void main()
18{
19    pid_t childPid;
20    switch(childPid=fork())
21    {
22        case 0:/*child process*/
23            //printenv();
24            exit(0);
25        default:/*parent process*/
26            printenv;
27            exit(0);
28    }
29}

```

```
[07/13/21]seed@VM:-.../Lab1$ gcc -o task22 task2.c  
[07/13/21]seed@VM:-.../Lab1$ ./task22  
  
SHELL=/bin/bash  
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/2021,unix/VM:/tmp/.ICE-unix/2021  
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=1981  
GTK_MODULES=gail:atk-bridge  
PWD=/home/seed/Desktop/Lab1  
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=s0;di=01;34;ln=01;36;mh=00;pi=40;33;so=01;35;do=01;35;bd=40;33;oi=cd=40;33;or=40;31;oi=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*;tz=01;31*;arc=01;31*;arj=01;31*;az=01;31*;lha=01;31*;lz4=01;31*;lzh=01;31*;lzma=01;31*;  
t=01;31*;txz=01;31*;tp=01;31*;t7z=01;31*;zip=01;31*;z=01;31*;dz=01;31*;gz=01;31*;lrz=01;31*;lzo=01;31*;xz=01;31*;xzt  
=01;31*;tzt=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*;car=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*;  
pjsd=01;31*;jpg=01;35*;jpeg=01;35*;mjpeg=01;35*;mpeg=01;35*;gif=01;35*;bmp=01;35*;pbm=01;35*;pgm=01;35*;ppm=01;35*;tga=01;35*;xbm  
=01;35*;ps=01;35*;pdf=01;35*;cvs=01;35*;cvs=01;35*;nc=01;35*;nc=01;35*;nc=01;35*;nc=01;35*;nc=01;35*;nc=01;35*;nc=01;35*;
```

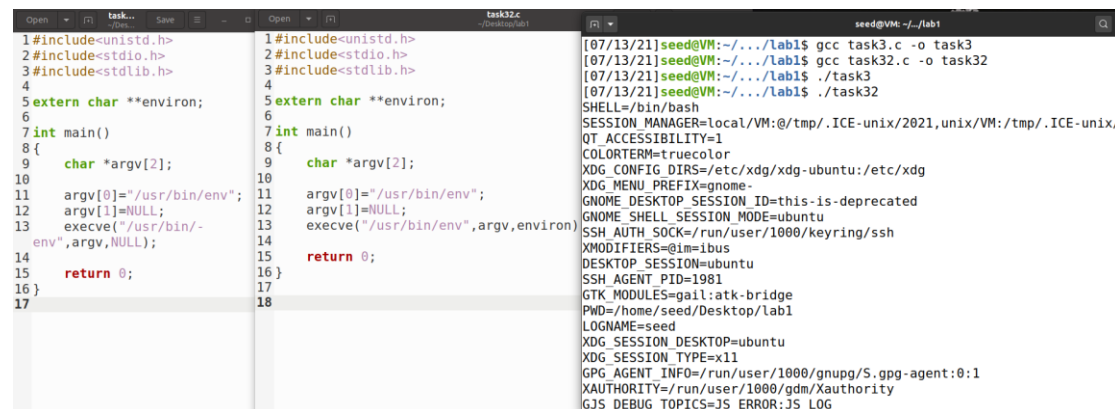
2) Compare the difference of these two files using the diff command.

```
[07/13/21]seed@VM:~/../lab1$ diff task2 task22
[07/13/21]seed@VM:~/../lab1$
```

通过对比，发现两者除了文件名外完全相同，说明子进程环境变量会继承父的环境变量。查找发现，子进程自父进程继承到进程的资格、环境、堆栈、内存等，但子进程所独有的是不同的父进程号、自己的文件描述符和目录流的拷贝、不继承异步输入和输出等。

Task3: Environment Variables and `execve()`

通过 `execve()` 来探究新进程是否会继承环境变量



```
task3.c
1#include<unistd.h>
2#include<stdio.h>
3#include<stdlib.h>
4
5extern char **environ;
6
7int main()
8{
9    char *argv[2];
10
11    argv[0]="/usr/bin/env";
12    argv[1]=NULL;
13    execve("/usr/bin/env",argv,NULL);
14
15    return 0;
16}
17

task32.c
1#include<unistd.h>
2#include<stdio.h>
3#include<stdlib.h>
4
5extern char **environ;
6
7int main()
8{
9    char *argv[2];
10
11    argv[0]="/usr/bin/env";
12    argv[1]=NULL;
13    execve("/usr/bin/env",argv,environ);
14
15    return 0;
16}
17

seed@VM: ~/../lab1
[07/13/21]seed@VM:~/../lab1$ gcc task3.c -o task3
[07/13/21]seed@VM:~/../lab1$ gcc task32.c -o task32
[07/13/21]seed@VM:~/../lab1$ ./task3
[07/13/21]seed@VM:~/../lab1$ ./task32
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/2021,unix/VM:/tmp/.ICE-unix/
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=1981
GTK_MODULES=gail:atk-bridge
PWD=/home/seed/Desktop/lab1
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
```

可以看到 `execve()` 的第三个参数是 `NULL` 时，进程不会传递任何环境变量，如果想把它自己的环境变量传给新程序，就将 `environ` 传给 `execve()` 函数即可。

补充: `execve()` 函数的使用方法:

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

`execve()` 用来执行参数 `filename` 字符串所代表的文件路径, `filename` 必须是一个二进制的可执行文件, 或者是一个脚本以 `#!` 格式开头的解释器参数。如果是后者, 这个解释器必须是一个可执行的有效路径名。第二个参数系利用数组指针来传递给执行文件, `argv` 是要调用的程序执行的参数序列, 也就是我们要调用的程序需要传入的参数。 `envp` 则为传递给执行文件的新环境变量数组, 同样也为参数序列。

Task4: Environment Variables and `system()`

了解使用 `system()` 执行一个新程序时, 环境变量是如何变化的



```
task4.c
~/Desktop/lab1
1#include<stdio.h>
2#include<stdlib.h>
3
4int main()
5{
6    system("/usr/bin/env");
7    return 0;
8}
9
```

编译并运行:

```
[07/13/21]seed@VM:~/.../lab1$ gcc task4.c -o task4
[07/13/21]seed@VM:~/.../lab1$ ./task4
GJS_DEBUG_TOPICS=JS ERROR;JS LOG
LESSOPEN=| /usr/bin/lesspipe %s
USER=seed
SSH_AGENT_PID=1981
XDG_SESSION_TYPE=x11
SHLVL=1
HOME=/home/seed
OLDPWD=/home/seed
DESKTOP_SESSION=ubuntu
GNOME_SHELL_SESSION_MODE=ubuntu
GTK_MODULES=gail:atk-bridge
MANAGERPID=1780
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
COLORTERM=truecolor
IM_CONFIG_PHASE=1
LOGNAME=seed
JOURNAL_STREAM=9:36240
_=./task4
XDG_SESSION_CLASS=user
USERNAME=seed
TERM=xterm-256color
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
WINDOWPATH=2
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:.
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/2021,unix/VM:/tmp/.ICE-unix/2021
INVOCATION_ID=e525a34bcdcd4806851f1dadfc580345
XDG_MENU_PREFIX=gnome-
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/e6992156_6784_4ed5_b82a_653aa626cd66
XDG_RUNTIME_DIR=/run/user/1000
DISPLAY=:0
LANG=en_US.UTF-8
```

不同于 `execve()`，`system()` 执行一个程序时，实际上通过 `/bin/sh -c command` 执行新程序，该函数执行 `/bin/sh` 创建一个 `shell`，并通过 `shell` 来执行新程序

先看一下 `system()` 函数的简单介绍。`system` 函数定义为 `int system (const char * string)`，该函数调用 `/bin/sh` 来执行参数指定的命令。`/bin/sh` 一般是一个软连接，指向某个具体的 `shell`，比如 `bash -c` 选项是告诉 `shell` 从字符串 `command` 中读取命令；在该 `command` 执行期间，`SIGCHLD` 信号会被暂时搁置，`SIGINT` 和 `SIGQUIT` 则会被忽略，意思是进程收到这两个信号后没有任何动作。`system()` 函数的函数返回值有些复杂。为了更好地理解 `system()` 函数的返回值，需要了解其执行过程，实际上 `system()` 函数执行了三步操作：

1 fork 一个子进程；

2 在子进程中调用 `exec` 函数去执行 `command`；

3 在父进程中调用 `wait` 去等待子进程结束。

若 `fork` 失败，`system()` 函数返回 -1。如果 `exec` 执行成功，也即 `command` 顺利执行完毕，则返回 `command` 通过 `exit` 或 `return` 返回的值。（注意，`command` 顺利执行不代表执行成功，例如 `command: "rm debuglog.txt"`，不管文件存不存在该 `command` 都顺利执行了）如果 `exec` 执行失败，也即 `command` 没有顺利执行，比如信号被中断，或者 `command` 命令根本不存在，`system()` 函数返回 127，如果 `command` 为 `NULL`，则 `system()` 函数返回值非 0，一般为 1。

Task5: Environment Variable and Set-UID Programs

指出哪些环境变量是否由用户进程的 `Set-UID` 程序的进程继承

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 extern char** environ;
5
6 int main()
7 {
8     int i=0;
9     while(environ[i]!=NULL)
10     {
11         printf("%s\n",environ[i]);
12         i++;
13     }
14 }
```

1) 先写一个能够输出所有环境变量的程序，编译并运行程序


```
[07/13/21]seed@VM:~/.../lab1$ gcc task5.c -o task5
[07/13/21]seed@VM:~/.../lab1$ ./task5
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/2021,unix/VM:/tmp/.ICE-unix/2021
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=1981
GTK_MODULES=gail:atk-bridge
PWD=/home/seed/Desktop/lab1
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:cd=40;33: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:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.t2=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;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:*.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=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/a7df623a_877b_4ec6_80ec_a191db9af5c3
INVOCATION_ID=e525a34bcdcd4806851f1dadfc580345
MANAGERPID=1780
GJS_DEBUG_OUTPUT=stderr
LESSCLOSE=/usr/bin/lesspipe %s %s
XDG_SESSION_CLASS=user
TERM=xterm-256color
LESSOPEN=| /usr/bin/lesspipe %s
USER=seed
GNOME_TERMINAL_SERVICE=:1.225
DISPLAY=:0
SHLVL=1
QT_IM_MODULE=ibus
XDG_RUNTIME_DIR=/run/user/1000
JOURNAL_STREAM=9:36240
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:./Desktop/Lab1/Task5
GDMSESSION=ubuntu
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
OLDPWD=/home/seed
```

2) 再将程序的权限修改为 root,使其成为一个 Set-UID 程序

```
[07/13/21]seed@VM:~/.../lab1$ sudo chown root:root task5.c
```

```
[07/13/21]seed@VM:~/.../lab1$ sudo chmod 4755 task5.c
```

3) 使用一般用户登录终端,使用 export 命令设置如下环境变量:

PATH , LD_LIBRARY_PATH , ANY_NAME

```
[07/13/21]seed@VM:~/.../lab1$ export PATH=$PATH:/Desktop/Lab1/Task5
```

```
[07/13/21]seed@VM:~/.../lab1$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/Desktop/lab1/Task5
```

```
[07/13/21]seed@VM:~/.../lab1$ export TCS=Desktop/lab1/Task5
```

4) 运行程序 task5,发现多了导入的新环境变量 TCS, PATH 后面也多了一部分设置的环境变量,但是 LD_LIBRARY_PATH 却没有在子进程的环境变量列表中

```
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/2021,unix/VM:/tmp/.ICE-unix/2021
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
TCS=Desktop/lab1/Task5
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
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:./Desktop/Lab1/Task5
GDMSESSION=ubuntu
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
```

Task6: The PATH Environment Variable and Set-UID Programs

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 int main()
5 {
6     system("ls");
7     return 0;
8 }
```

1) 编译以上程序并将其设置为特权程序

```
[07/13/21] seed@VM:~$ cd Desktop/lab1
[07/13/21] seed@VM:~/.../lab1$ gcc -o task6 task6.c
[07/13/21] seed@VM:~/.../lab1$ sudo chomn root task6
sudo: chomn: command not found
[07/13/21] seed@VM:~/.../lab1$ sudo chown root task6
[07/13/21] seed@VM:~/.../lab1$ sudo chmod 4755 task6
[07/13/21] seed@VM:~/.../lab1$ ls -l task6
-rwsr-xr-x 1 root seed 16696 Jul 13 11:42 task6
[07/13/21] seed@VM:~/.../lab1$
```

2) 为了防止 Set-UID 程序自动放弃特权，先去掉 Ubuntu 的保护机制，然后将/bin/sh 复制到当前文件夹中，然后设置环境变量 PATH=.:\$PATH，运行 task6，发现该程序没有按照 system("ls")执行 bin/ls，而是执行了刚刚设置的 bin/sh (记得用 sudo 权限)

```
[07/13/21] seed@VM:~/.../lab1$ sudo rm /bin/sh
[07/13/21] seed@VM:~/.../lab1$ ln -s /bin/zsh /bin/sh
ln: failed to create symbolic link '/bin/sh': Permission denied
[07/13/21] seed@VM:~/.../lab1$ cp /bin/sh ./ls
cp: cannot stat '/bin/sh': No such file or directory
[07/13/21] seed@VM:~/.../lab1$ sudo rm /bin/sh
rm: cannot remove '/bin/sh': No such file or directory
[07/13/21] seed@VM:~/.../lab1$ sudo ln -s /bin/zsh /bin/sh
[07/13/21] seed@VM:~/.../lab1$ cp /bin/sh ./ls
[07/13/21] seed@VM:~/.../lab1$ export PATH=.:$PATH
[07/13/21] seed@VM:~/.../lab1$ task6
VM#
```

Task8: Invoking External Programs Using `system()` versus `execve()`

```

1#include<string.h>
2#include<stdio.h>
3#include<stdlib.h>
4
5int main(int argc,char *argv[])
6{
7    char *v[3];
8    char *command;
9
10    if(argc<2)
11    {
12        printf("Please type a file name .\n");
13        return 1;
14    }
15
16    v[0]="bin/cat";
17    v[1]=argv[1];
18    v[2]=NULL;
19    command=malloc(strlen(v[0]) + strlen(v[1]) + 2);
20    sprintf(command,"%s %s",v[0],v[1]);
21
22    //Use only one of the followings.
23    system(command);
24    //execve(v[0],v,NULL);
25
26    return 0;|
27 }
28

```

1) 编译以上程序，然后在 root 状态下，更改其所有者为 root 并将其设置为 SetUID 程序

```

[07/13/21]seed@VM:~$ cd Desktop/lab1
[07/13/21]seed@VM:~/.../lab1$ gcc task8.c -o task8
[07/13/21]seed@VM:~/.../lab1$ ls
ls      task22  task3   task32.c  task4     task5     task6     task8
task2   task2.c task32   task3.c   task4.c   task5.c   task6.c   task8.c
[07/13/21]seed@VM:~/.../lab1$ ls -l task8
-rwxrwxr-x 1 seed seed 16928 Jul 13 12:04 task8
[07/13/21]seed@VM:~/.../lab1$ sudo chown root task8
[07/13/21]seed@VM:~/.../lab1$ sudo chmod 4755 task8
[07/13/21]seed@VM:~/.../lab1$ ls -l task8
-rwsr-xr-x 1 root seed 16928 Jul 13 12:04 task8
[07/13/21]seed@VM:~/.../lab1$ █

```

2) 创建一个 test.c 文件，输入信息"hello world!"，然后在 root 状态下将其所有者和权限全部改为 000，此时用户状态下无法对 test.c 文件进行读写操作

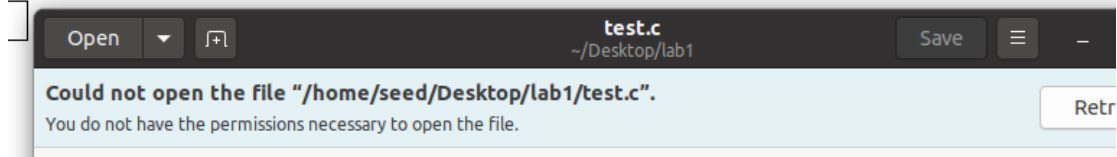
```

Open  test.c
~/Desktop/lab1
1#include<string.h>
2#include<stdio.h>
3#include<stdlib.h>
4
5int main()
6{
7    printf("hello world");
8    return 0;
9}
10

```



```
[07/13/21]seed@VM:~/.../lab1$ sudo chown root:root test.c
[07/13/21]seed@VM:~/.../lab1$ sudo chomd 000 test.c
sudo: chomd: command not found
[07/13/21]seed@VM:~/.../lab1$ sudo chmod 000 test.c
[07/13/21]seed@VM:~/.../lab1$ gedit test.c
```



```
[07/13/21]seed@VM:~/.../lab1$ ls -l test.c
----- 1 root root 114 Jul 13 12:21 test.c
```

3) 理论上运行 **task8**, 此时 **task8** 在进行了 **SetUID** 操作后已经具有了 **root** 权限, 并且可以对 **test.c** 文件进行读写操作, 改变 **task8** 中的 **"/bin/cat"** 为 **"rm"** 还能实现删除操作, 删除前如果是这样, 删除后 **test.c/** 就不在了

```
[07/13/21]seed@VM:~/.../lab1$ ls -l
total 1052
-rwxr-xr-x 1 seed seed 878288 Jul 13 11:53 ls
-rwxrwxr-x 1 seed seed 16888 Jul 13 09:41 task2
-rwxrwxr-x 1 seed seed 16888 Jul 13 09:49 task22
-rw-rw-r-- 1 seed seed 390 Jul 13 10:16 task2.c
-rwxrwxr-x 1 seed seed 16752 Jul 13 10:24 task3
-rwxrwxr-x 1 seed seed 16824 Jul 13 10:24 task32
-rw-rw-r-- 1 seed seed 228 Jul 13 10:23 task32.c
-rw-rw-r-- 1 seed seed 224 Jul 13 10:23 task3.c
-rwxrwxr-x 1 seed seed 16696 Jul 13 10:44 task4
-rw-rw-r-- 1 seed seed 96 Jul 13 10:44 task4.c
-rwxrwxr-x 1 seed seed 16768 Jul 13 10:58 task5
-rwsr-xr-x 1 root root 180 Jul 13 10:57 task5.c
-rwsr-xr-x 1 root seed 16696 Jul 13 11:42 task6
-rw-rw-r-- 1 seed seed 86 Jul 13 11:41 task6.c
-rwsr-xr-x 1 root seed 16928 Jul 13 12:04 task8
-rw-rw-r-- 1 seed seed 487 Jul 13 12:20 task8.c
----- 1 root root 114 Jul 13 12:21 test.c
```

4) 但可能因为 **Ubuntu** 的保护机制, 这个问题我一直没能有效解决, 这个 **task8** 的结尾没能很好的做出来……

6.实验小结:

本次实验熟悉了 **Linux** 环境, 通过 7 个 **task**, 也加深了对于环境变量的了解, 以及 **system()**函数和 **execve()**函数等的独到之处, 对 **SetUID** 特权程序, 普通用户和 **root** 用户的权限等等也有了更深的理解。