Set-UID实验报告

57119101 王晨阳

2021年3月20日

TASK1

实验目的

熟悉有关环境变量的基本语句和操作。

实验步骤

- 使用 env 查看环境变量

```
seed@VM:~$ env
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:b22bfbba-56d1-45bc-8920-01135456caec
CLUTTER_IM_MODULE=xim
SESSION=ubuntu
GIO_LAUNCHED_DESKTOP_FILE_PID=2698
ANDROID_HOME=/home/seed/android/android-sdk-linux
GPG_AGENT_INFO=/home/seed/.gnupg/S.gpg-agent:0:1
TERM=xterm
XDG_MENU_PREFIX=gnome-
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_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/1000/1452
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:s
t=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:*.tz
o=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.l
rz=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=01;31:*.ea
r=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:*.tga=01;35:*.xbm=01;35:*.xpm=0
1;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;3
5:*.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:*.m
p3=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
```

```
LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/1920,unix/VM:/tmp/.ICE-unix/1920
GIO_LAUNCHED_DESKTOP_FILE=/usr/share/applications/terminator.desktop
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/usr/share/upstart/xdg:/etc/xdg
DESKTOP_SESSION=ubuntu
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/seed/android/android-ndk/android-ndk-r8d:/home/seed/.local/bin
QT_IM_MODULE=ibus
QT_QPA_PLATFORMTHEME=appmenu-qt5
XDG_SESSION_TYPE=x11
PWD=/home/seed
JOB=unity-settings-daemon
XMODIFIERS=@im=ibus
JAVA_HOME=/usr/lib/jvm/java-8-oracle
GNOME_KEYRING_PID=
LANG=en_US.UTF-8
GDM_LANG=en_US
MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path
COMPIZ_CONFIG_PROFILE=ubuntu-lowgfx
IM_CONFIG_PHASE=1
GDMSESSION=ubuntu
SESSIONTYPE=gnome-session
GTK2_MODULES=overlay-scrollbar
SHLVL=1
HOME=/home/seed
XDG_SEAT=seat0
LANGUAGE=en_US
LIBGL_ALWAYS_SOFTWARE=1
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
UPSTART_INSTANCE=
XDG_SESSION_DESKTOP=ubuntu
UPSTART_EVENTS=xsession started
LOGNAME=seed
COMPIZ_BIN_PATH=/usr/bin/
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-0xMMSd8cPf
J2SDKDIR=/usr/lib/jvm/java-8-oracle
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/share/gnome:/usr/local/share/:/usr/share/:/var/lib/snapd/desktop
QT4_IM_MODULE=xim
LESSOPEN=| /usr/bin/lesspipe %s
INSTANCE=
UPSTART_JOB=unity7
XDG_RUNTIME_DIR=/run/user/1000
DISPLAY=:0
XDG_CURRENT_DESKTOP=Unity
GTK_IM_MODULE=ibus
J2REDIR=/usr/lib/jvm/java-8-oracle/jre
LESSCLOSE=/usr/bin/lesspipe %s %s
XAUTHORITY=/home/seed/.Xauthority
```

```
COLORTERM=gnome-terminal
_=/usr/bin/env
```

然后查找 `PWD` 变量

```
seed@VM:~$ env | grep PWD
PWD=/home/seed
```

- 使用 `export` 创建环境变量

```
seed@VM:~$ export envvar=envvar1
seed@VM:~$ env | grep envvar
envvar=envvar1

```

然后使用 `unset` 删除刚刚创建的变量

```
seed@VM:~$ unset envvar
seed@VM:~$ env | grep envvar
```

TASK2

实验目的

探索 `fork()` 得到的 `child` 进程与 `parent` 进程的区别

实验步骤

- 编写程序 `2_child.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);
    }
}
```

编译并保存结果到`child`

```
seed@VM:~/Desktop$ gcc '2_child.c' -o '2_child.out'
seed@VM:~/Desktop$ '2_child.out' > 'child'
```

- 修改程序`2_parent.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);
    }
}
```

编译并保存结果到`parent`

```
seed@VM:~/Desktop$ gcc '2_parent.c' -o '2_parent.out'
seed@VM:~/Desktop$ '2_parent.out' > 'parent'
```

- 使用`diff`比较`child`和`parent`

```
seed@VM:~/Desktop$ diff 'child' 'parent'
78c78
< _=./2_child.out
---
> _=./2_parent.out
```

结果分析

`child`相较于`parent`，结果的第76行发生了改变。可以认为，**子进程和父进程除了pid几乎完全相同。**

TASK3

实验目的

探究使用`execve()`执行程序时环境变量的变化。

实验步骤

- 编写程序`3_null.c`

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

编译并保存结果到`3_null`

```
seed@VM:~/Desktop$ gcc '3_null.c' -o '3_null.out'
seed@VM:~/Desktop$ '3_null.out' > '3_null'
```

- 修改程序`3_environ.c`

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

extern char **environ;

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

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

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

```
15
16      return 0 ;
17  }
```

编译并保存结果到`3_environ`

```
1   seed@VM:~/Desktop$ gcc '3_environ.c' -o '3_environ.out'
2   seed@VM:~/Desktop$ '3_environ.out' > '3_environ'
```

- 观察结果

`3_null`为空

```
1   seed@VM:~/Desktop$ cat 3_null
```

`3_environ`有内容

```
1   seed@VM:~/Desktop$ cat 3_environ
2   XDG_VTNR=7
3   ORBIT_SOCKETDIR=/tmp/orbit-seed
4   XDG_SESSION_ID=c1
5   XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed
6   IBUS_DISABLE_SNOOPER=1
7   TERMINATOR_UUID=urn:uuid:b22bfbba-56d1-45bc-8920-01135456caec
8   CLUTTER_IM_MODULE=xim
9   SESSION=ubuntu
10  GIO_LAUNCHED_DESKTOP_FILE_PID=2698
11  ANDROID_HOME=/home/seed/android/android-sdk-linux
12  GPG_AGENT_INFO=/home/seed/.gnupg/S.gpg-agent:0:1
13  TERM=xterm
14  XDG_MENU_PREFIX=gnome-
15  SHELL=/bin/bash
16  DERBY_HOME=/usr/lib/jvm/java-8-oracle/db
17  QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1
18  LD_PRELOAD=/home/seed/lib/boost/libboost_program_options.so.1.64.0:/home/seed
    /lib/boost/libboost_filesystem.so.1.64.0:/home/seed/lib/boost/libboost_system
    .so.1.64.0
19  WINDOWID=60817412
20  UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1452
21  GNOME_KEYRING_CONTROL=
22  GTK_MODULES=gail:atk-bridge:unity-gtk-module
23  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;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=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:*.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=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:
QT_ACCESSIBILITY=1
LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/1920,unix/VM:/tmp/.ICE-unix/1920
GIO_LAUNCHED_DESKTOP_FILE=/usr/share/applications/terminator.desktop
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/usr/share/upstart/xdg:/etc/xdg
DESKTOP_SESSION=ubuntu
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/seed/android/android-ndk/android-ndk-r8d:/home/seed/.local/bin
QT_IM_MODULE=ibus
QT_QPA_PLATFORMTHEME=appmenu-qt5
XDG_SESSION_TYPE=x11
PWD=/home/seed/Desktop
JOB=unity-settings-daemon
XMODIFIERS=@im=ibus
JAVA_HOME=/usr/lib/jvm/java-8-oracle
GNOME_KEYRING_PID=
LANG=en_US.UTF-8
GDM_LANG=en_US
MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path
COMPIZ_CONFIG_PROFILE=ubuntu-lowgfx
IM_CONFIG_PHASE=1
GDMSESSION=ubuntu
SESSIONTYPE=gnome-session
GTK2_MODULES=overlay-scrollbar
SHLVL=1
HOME=/home/seed
XDG_SEAT=seat0
LANGUAGE=en_US
LIBGL_ALWAYS_SOFTWARE=1
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
```

```
UPSTART_INSTANCE=
XDG_SESSION_DESKTOP=ubuntu
UPSTART_EVENTS=xsession started
LOGNAME=seed
COMPIZ_BIN_PATH=/usr/bin/
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-0xMMSd8cPf
J2SDKDIR=/usr/lib/jvm/java-8-oracle
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/share/gnome:/usr/local/share/:/usr/share/:/var/lib/snapd/desktop
QT4_IM_MODULE=xim
LESSOPEN=| /usr/bin/lesspipe %s
INSTANCE=
UPSTART_JOB=unity7
XDG_RUNTIME_DIR=/run/user/1000
DISPLAY=:0
XDG_CURRENT_DESKTOP=Unity
GTK_IM_MODULE=ibus
J2REDIR=/usr/lib/jvm/java-8-oracle/jre
LESSCLOSE=/usr/bin/lesspipe %s %s
XAUTHORITY=/home/seed/.Xauthority
COLORTERM=gnome-terminal
OLDPWD=/home/seed
_=./3_environ.out
```

结果分析

`execve()`函数的格式为

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

在第一个程序中，我们没有向`envp[]`传入参数，故没有结果；

在第二个程序中，传入了环境变量，故能够打印出环境变量。

可见，`execve()`产生的新进程是被独立赋予环境变量的，相当于它是在已有进程上开启了新的进程。

TASK4

实验目的

探究使用`system()`执行程序时环境变量的变化。

实验步骤

- 编写程序`4.c`

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

int main()
{
    system("/usr/bin/env");

    return 0 ;
}
```

编译并保存结果到 `4`

```
seed@VM:~/Desktop$ gcc '4.c' -o '4.out'
seed@VM:~/Desktop$ '4.out' > '4'
```

得到结果

```
seed@VM:~/Desktop$ cat 4
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/stage/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_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-0xMMSd8cPf
GIO_LAUNCHED_DESKTOP_FILE_PID=2698
COLORTERM=gnome-terminal
GNOME_KEYRING_CONTROL=
QT_QPA_PLATFORMTHEME=appmenu-qt5
MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path
IM_CONFIG_PHASE=1
SESSIONTYPE=gnome-session
UPSTART_JOB=unity7
LOGNAME=seed
GTK_IM_MODULE=ibus
WINDOWID=60817412
_=./4.out
DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path
XDG_SESSION_ID=c1
```

```
TERM=xterm
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GTK2_MODULES=overlay-scrollbar
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/seed/android/android-ndk/android-ndk-r8d:/home/seed/.local/bin
DERBY_HOME=/usr/lib/jvm/java-8-oracle/db
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/1920,unix/VM:/tmp/.ICE-unix/1920
GDM_LANG=en_US
XDG_MENU_PREFIX=gnome-
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_RUNTIME_DIR=/run/user/1000
COMPIZ_BIN_PATH=/usr/bin/
DISPLAY=:0
IBUS_DISABLE_SNOOPER=1
LD_PRELOAD=/home/seed/lib/boost/libboost_program_options.so.1.64.0:/home/seed/lib/boost/libboost_filesystem.so.1.64.0:/home/seed/lib/boost/libboost_system.so.1.64.0
LANG=en_US.UTF-8
XDG_CURRENT_DESKTOP=Unity
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:*.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=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:*.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=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:
XMODIFIERS=@im=ibus
XDG_SESSION_DESKTOP=ubuntu
XAUTHORITY=/home/seed/.Xauthority
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
TERMINATOR_UUID=urn:uuid:b22bfbba-56d1-45bc-8920-01135456caec
SHELL=/bin/bash
QT_ACCESSIBILITY=1
GDMSESSION=ubuntu
LESSCLOSE=/usr/bin/lesspipe %s %s
UPSTART_EVENTS=xsession started
GPG_AGENT_INFO=/home/seed/.gnupg/S.gpg-agent:0:1
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1452
XDG_VTNR=7
QT_IM_MODULE=ibus
PWD=/home/seed/Desktop
```

```
JAVA_HOME=/usr/lib/jvm/java-8-oracle
CLUTTER_IM_MODULE=xim
ANDROID_HOME=/home/seed/android/android-sdk-linux
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/usr/share/upstart/xdg:/etc/xdg
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/share/gnome:/usr/local/share/:/usr/share/:/var/lib/snapd/desktop
JOB=unity-settings-daemon
```

结果分析

可以看到，程序输出了环境变量。

TASK5

实验目的

使用`set-uid`获取环境变量

实验步骤

- 编写程序`5.c`

```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++;
    }
}
```

- 编译程序到`5.out`

```
seed@VM:~/Desktop$ gcc '5.c' -o '5.out'
```

修改权限，然后使其成为Set-UID程序

```
seed@VM:~/Desktop$ sudo chown root 5.out
seed@VM:~/Desktop$ sudo chmod 4755 5.out
```

- 检查`PATH`和`LD_LIBRARY_PATH`环境变量是否存在

```
seed@VM:~/Desktop$ env | grep PATH
LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path
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/seed/android/android-ndk/android-ndk-r8d:/home/seed/.local/bin
MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path
COMPIZ_BIN_PATH=/usr/bin/
```

```
新建`ANY_NAME`环境变量

```bash
seed@VM:~/Desktop$ export ANY_NAME=ANYNAME
seed@VM:~/Desktop$ env | grep ANY_NAME
ANY_NAME=ANYNAME
```

使用刚刚的程序打印这三个环境变量

```
seed@VM:~/Desktop$ '5.out' | grep PATH
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path
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/seed/android/android-ndk/android-ndk-r8d:/home/seed/.local/bin
MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path
COMPIZ_BIN_PATH=/usr/bin/
[03/20/21]seed@VM:~/Desktop$ '5.out' | grep ANY_NAME
ANY_NAME=ANYNAME
```

结果分析

尽管 `5.out` 设置为root所有，但因为其设置了SUID权限，故可以通过它产生拥有特殊权限地子进程，打印环境变量。

TASK6

实验目的

通过SUID访问 `PATH` 环境变量。
```

实验步骤

- 编写程序 `6.c`

```
#include<stdlib.h>

int main()
{
    system("ls");
    return 0;
}
```

编译程序到 `6.out`

```
seed@VM:~/Desktop$ gcc '6.c' -o '6.out'
```

修改权限，然后使其成为Set-UID程序

```
seed@VM:~/Desktop$ sudo chown root 6.out
seed@VM:~/Desktop$ sudo chmod 4755 6.out
```

使用 `6.out` 实现 `ls` 的功能

```
seed@VM:~/Desktop$ '6.out' /
2_child.c    2_parent.c    3_environ    3_environ.out  3_null.c    4    4.out
 5.out  6.out  parent
2_child.out  2_parent.out  3_environ.c  3_null         3_null.out  4.c  5.c
 6.c    child
```

结果分析

SUID程序成功执行了 `ls` 指令。

TASK7

实验目的

探究 `LD_PRELOAD` 环境变量和SUID程序关系。

实验步骤

- 编写程序 `mylib.c`

```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");
}
```

编译程序

```
seed@VM:~/Desktop$ gcc -fPIC -g -c mylib.c
seed@VM:~/Desktop$ gcc -shared -o libmylib.so.1.0.1 mylib.o -lc
```

设置 `LD_PRELOAD` 环境变量

```
seed@VM:~/Desktop$ export LD_PRELOAD=./libmylib.so.1.0.1
```

编写程序 `myprog.c`

```c
/* myprog.c */
#include<unistd.h>

int main()
{
    sleep(1);
    return 0;
}
```

编译程序到 `myprog`

```
seed@VM:~/Desktop$ gcc 'myprog.c' -o 'myprog'
```

- 以普通用户身份执行 `myprog`

```
seed@VM:~/Desktop$ myprog
I am not sleeping!
```

将 `myprog` 设置为Set-UID root程序，并以普通用户身份执行

```
seed@VM:~/Desktop$ sudo chown root myprog
seed@VM:~/Desktop$ sudo chmod 4755 myprog
seed@VM:~/Desktop$ myprog
```

将 `myprog` 设置为Set-UID root程序，在root账户下再次设置 `LD_PRELOAD` 环境变量并运行

```
seed@VM:~/Desktop$ su
Password:
root@VM:/home/seed/Desktop# gcc 'myprog.c' -o 'myprog'
root@VM:/home/seed/Desktop# sudo chown root myprog
root@VM:/home/seed/Desktop# sudo chmod 4755 myprog
root@VM:/home/seed/Desktop# export LD_PRELOAD=./libmylib.so.1.0.1
root@VM:/home/seed/Desktop# myprog
I am not sleeping!
```

在非root的另一个账户中设置 `LD_PRELOAD` 环境变量并运行

```
 root@VM:/home/seed/Desktop# useradd seed2
 root@VM:/home/seed/Desktop# passwd seed2
```

```
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@VM:/home/seed/Desktop# exit
exit
seed@VM:~/Desktop$ gcc 'myprog.c' -o 'myprog'
seed@VM:~/Desktop$ sudo chown root myprog
seed@VM:~/Desktop$ sudo chmod 4755 myprog
seed@VM:~/Desktop$ su seed2
Password:
seed2@VM:/home/seed/Desktop$ export LD_PRELOAD=./libmylib.so.1.0.1
seed2@VM:/home/seed/Desktop$ myprog
```

结果分析

程序是seed用户的SUID程序，所以放弃已有的`LD_PRELOAD`环境变量路径，在seed全局中寻找链接库，所以不会被覆盖。

TASK8

实验目的

使用`system()`和`execve()`调用外部程序。

实验步骤

- 新建文件`tmp`

```
seed@VM:~/Desktop$ touch tmp
seed@VM:~/Desktop$ gedit tmp
```

编辑内容为

```
tmp file
```

编写程序`8_system.c`

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

int main(int argc, char *argv[])
{
    char *v[3];
    char *command;

    if(argc < 2) {
        printf("Please type a file name.\n");
        return 1;
    }
```

```
14
15      v[0] = "/bin/cat"; v[1] = argv[1]; v[2] = NULL;
16      command = malloc(strlen(v[0]) + strlen(v[1]) + 2);
17      sprintf(command, "%s %s", v[0], v[1]);
18
19      // Use only one of the followings.
20      system(command);
21      // execve(v[0], v, NULL);
22
23      return 0 ;
24  }
```

编译程序到 `8_system`

```
1   seed@VM:~/Desktop$ gcc '8_system.c' -o '8_system'
```

将 `8_system` 设置为Set-UID root程序并运行

```
1   seed@VM:~/Desktop$ sudo chown root 8_system
2   seed@VM:~/Desktop$ sudo chmod 4755 8_system
3   seed@VM:~/Desktop$ 8_system tmp
4   tmp file
```

- 编写程序 `8_execve.c`

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

编译程序到 `8_execve`

```
1   seed@VM:~/Desktop$ gcc '8_execve.c' -o '8_execve'
```

将 `8_execve` 设置为Set-UID root程序并运行

```
seed@VM:~/Desktop$ sudo chown root 8_execve
seed@VM:~/Desktop$ sudo chmod 4755 8_execve
seed@VM:~/Desktop$ 8_execve tmp
/bin/cat: tmp: No such file or directory
```

结果分析

`system()`可以成功攻击，而`execve()`不能

TASK9

实验目的

探究Capbility泄露

实验步骤

- 查看`etc/zzz`为空

  编写程序`9.c`

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

void main()
{ int fd;

    /* Assume that /etc/zzz is an important system file,
    * and it is owned by root with permission 0644.
    * Before running this program, you should creat
    * the file /etc/zzz first. */
    fd = open("/etc/zzz", O_RDWR | O_APPEND);
    if (fd == -1) {
        printf("Cannot open /etc/zzz\n");
        exit(0);
    }

    /* Simulate the tasks conducted by the program */
    sleep(1);

    /* After the task, the root privileges are no longer needed,
    it's time to relinquish the root privileges permanently. */
    setuid(getuid()); /* getuid() returns the real uid */

    if (fork()) { /* In the parent process */
        close (fd);
        exit(0);
```

```
30      } else { /* in the child process */
31          /* Now, assume that the child process is compromised, malicious
32          attackers have injected the following statements
33          into this process */
34
35          write (fd, "Malicious Data\n", 15);
36          close (fd);
37      }
38  }
```

编译程序到 `9`

```
seed@VM:~/Desktop$ gcc '9.c' -o '9'
```

将 `9` 设置为Set-UID root程序并运行

```
seed@VM:~/Desktop$ sudo chown root 9
seed@VM:~/Desktop$ sudo chmod 4755 9
seed@VM:~/Desktop$ 9
```

查看 `etc/zzz`，发现已经有内容

```
Malicious Data
```

结果分析

利用Capbility泄露，成功获取了对文件的修改权限。

实验体会

通过本次实验，掌握了set-uid的基本原理，学习了如何通过特权程序实现攻击，加深了对操作系统概念的理解。提高了动手能力，解决问题的能力得到强化。