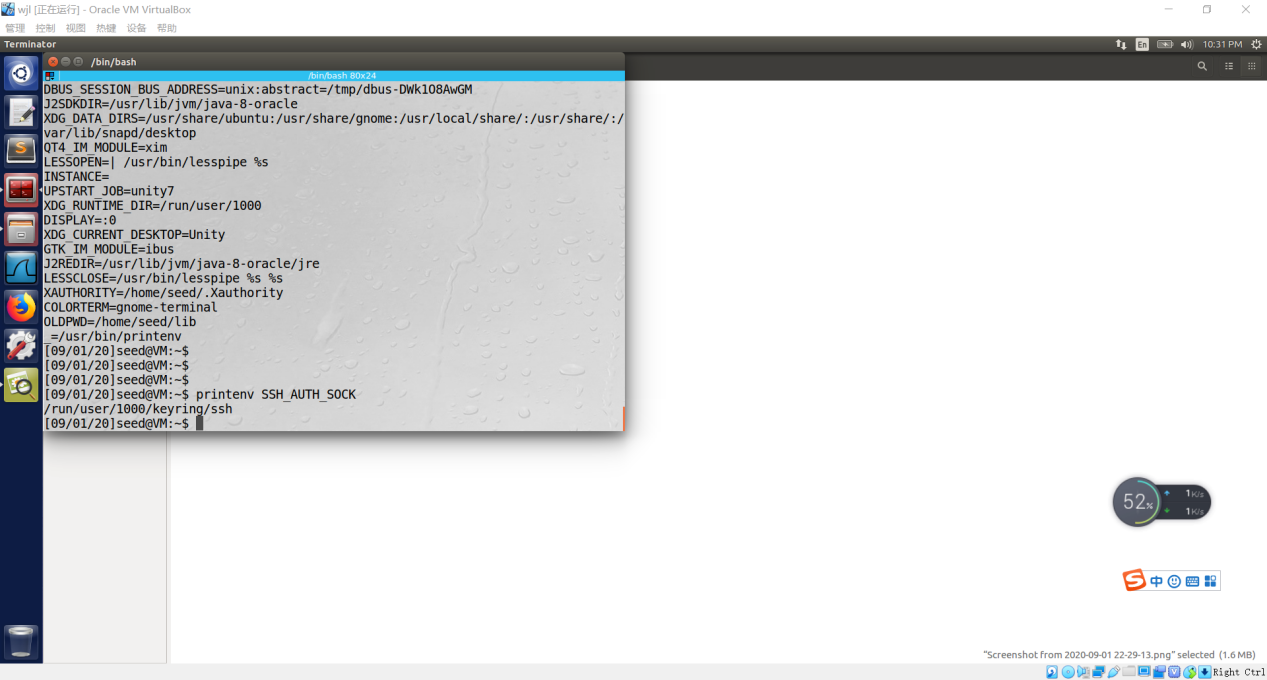
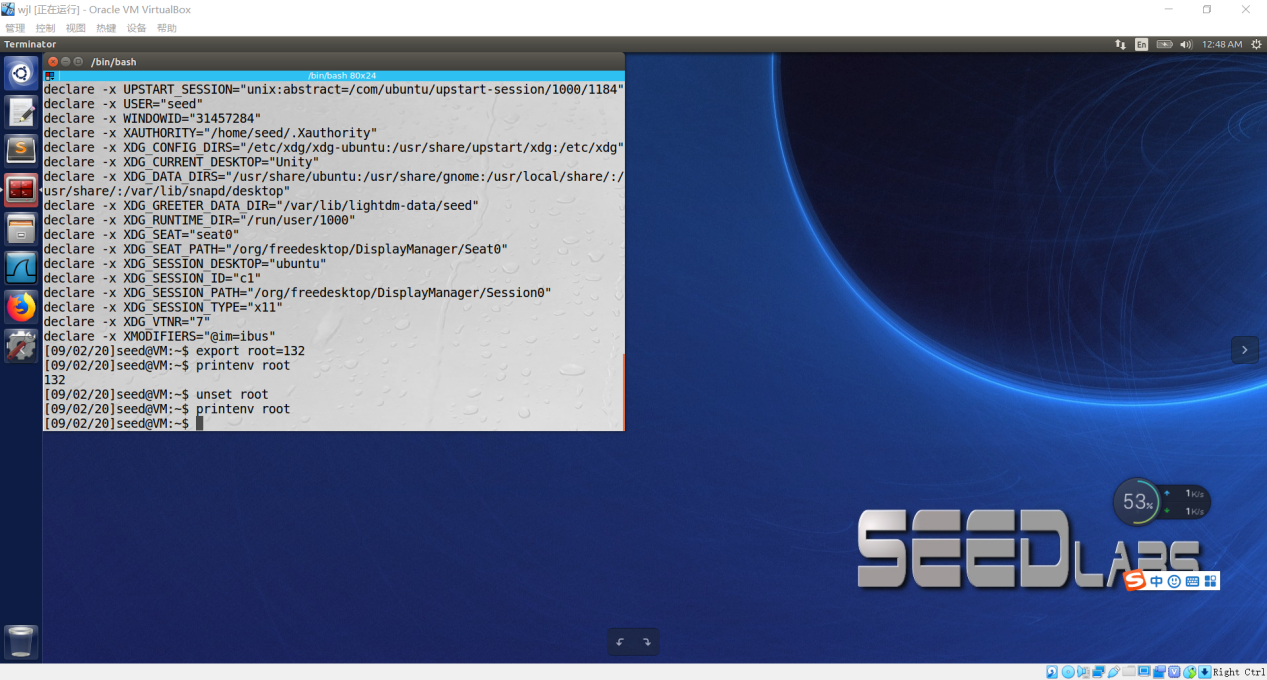
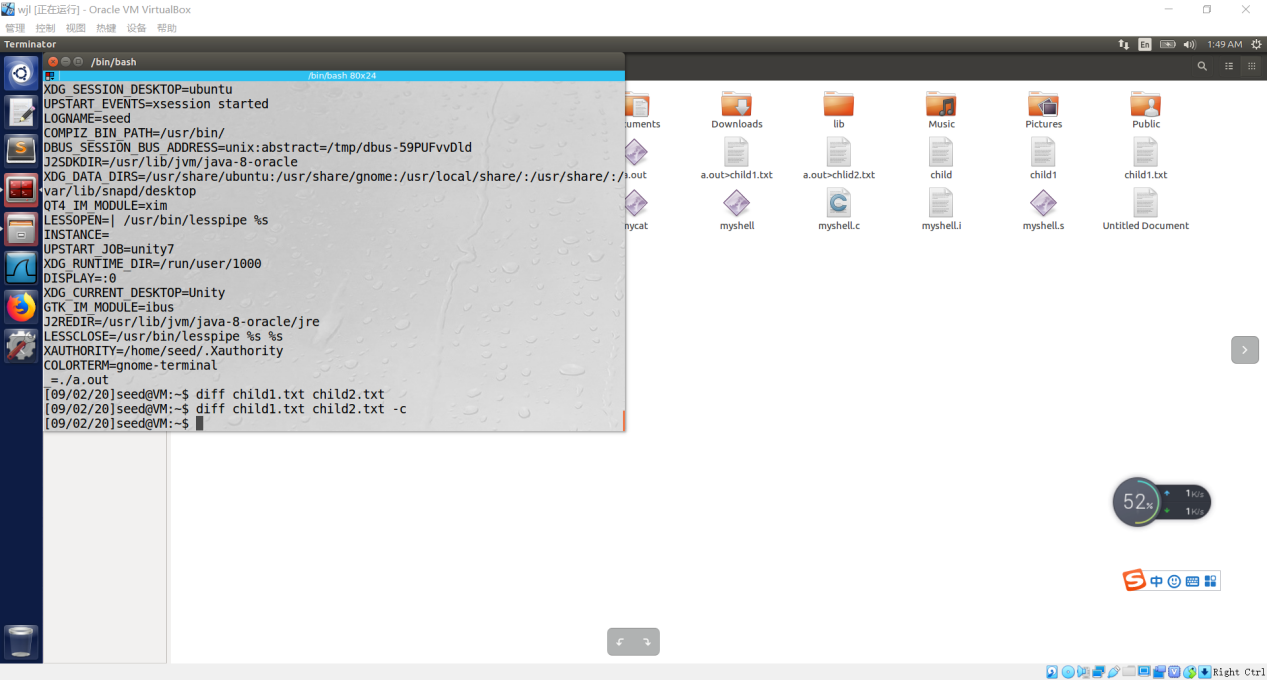
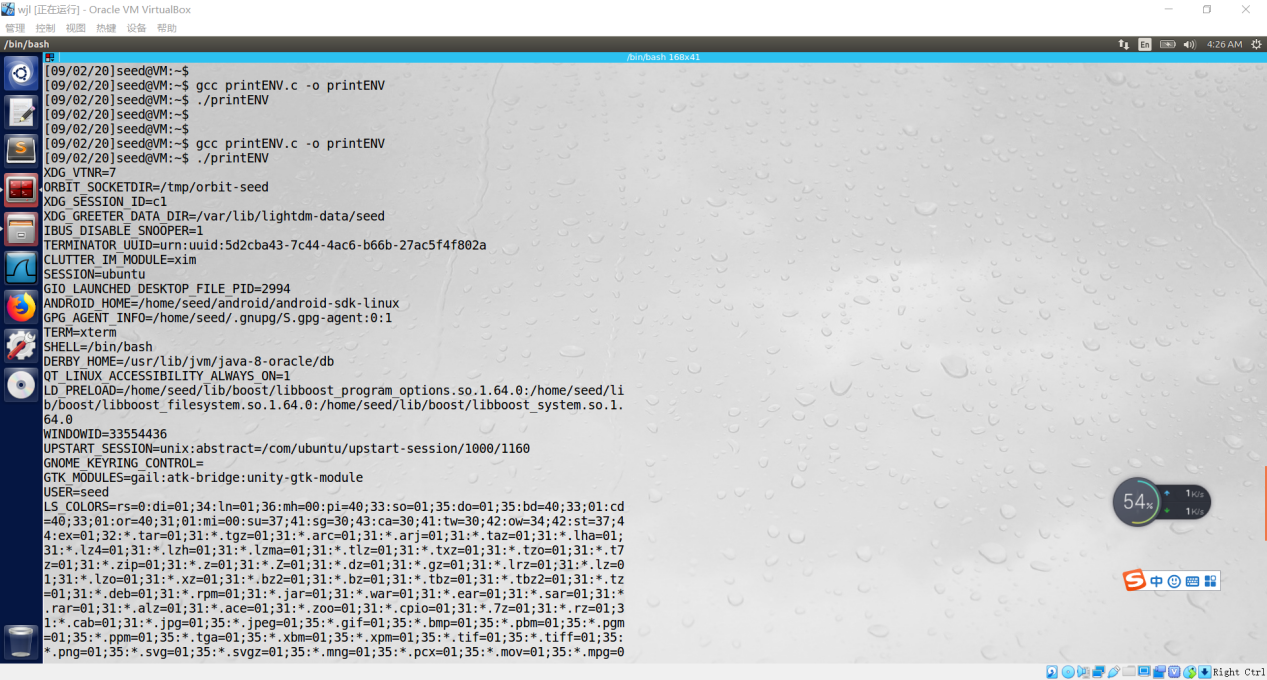
57118130 王嘉麟







That means the environment variables of child process is exactly the same with the parent process.



Environment variables of Current process is NULL, therefore the first time prints nothing.

The second time prints the extern environment variables,which don’t belongs to the process.

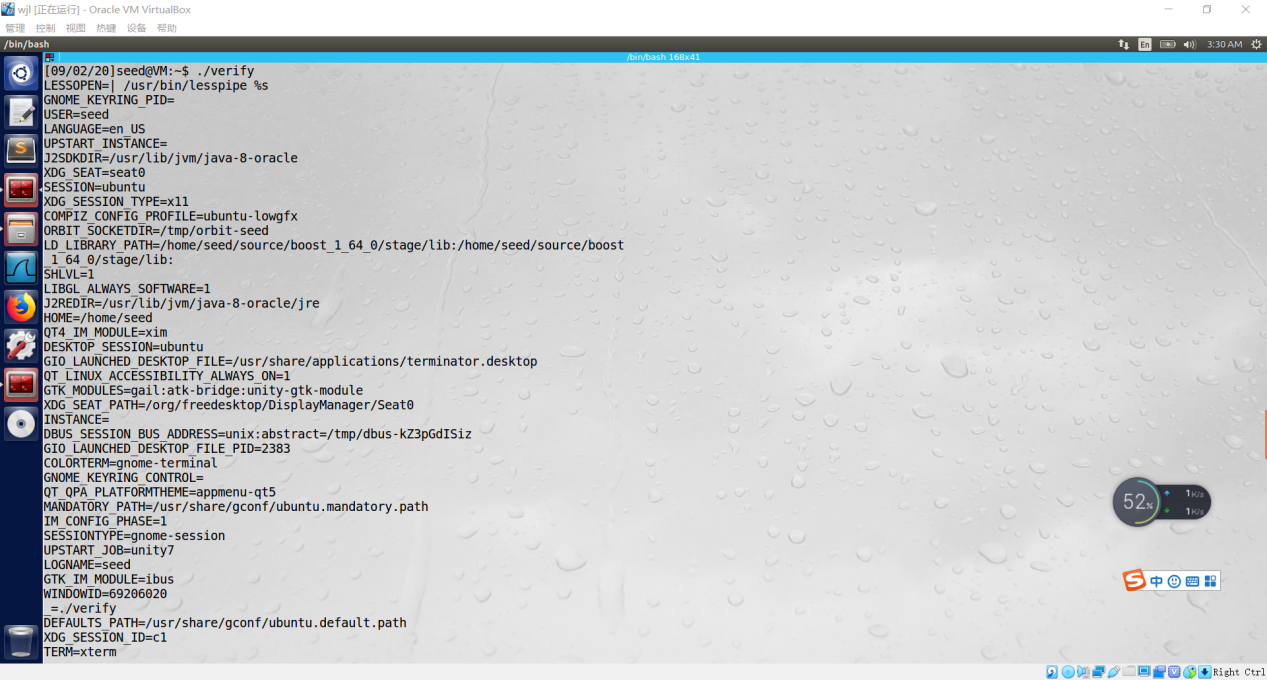
The first parameter of execve() is command, it can’t pass environment variables .

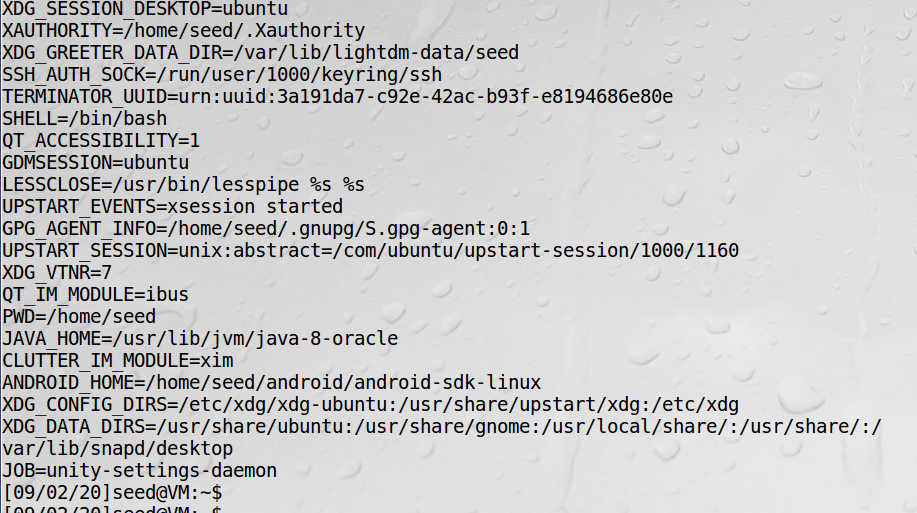
But when running “/usr/bin/env” in shell or use it like this: system(“/usr/bin/env”). It will output many environment variables.

The second parameter of execve() is filename,it can’t be used to pass environment variables either.

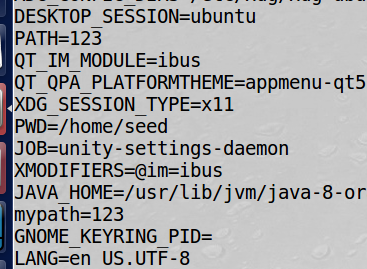
The third parameter of execve() is used to pass environment variables obviously.

The program equals to running “bin/sh -c /usr/bin/env” in shell



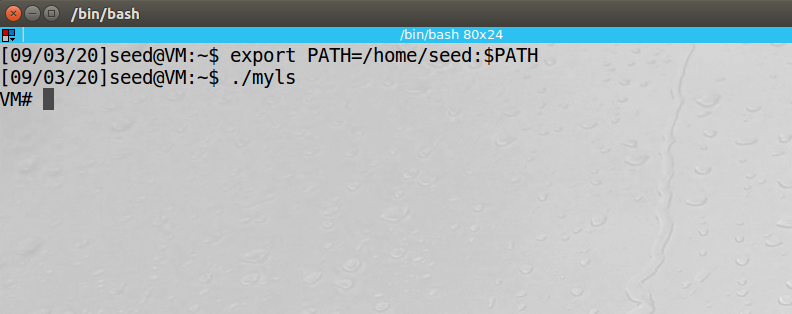


最下方显示JOB环境变量是daemon方式，说明程序交给了一个特权进程执行



I didn’t find the LD\_LIBRARY\_PATH in the output,but PATH and mypath are modfied by me and passed the SET-UID child process.

It seems the program originally won’t cout LD\_LIBRARY\_PATH.



将myls.c编译为myls

#include <stdlib.h>

int main()

{

system("ls");

return 0;

}

而后改变system的调用路径PATH，将其改为seed目录，

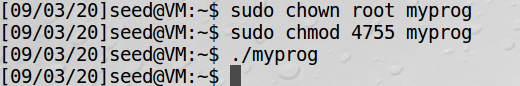
因为bash和dash被调用会降权，将zsh复制到seed目录并改名为ls，

最后，将myls程序owner改为root，并将其设为set-uid程序，可以实现在shell中运行

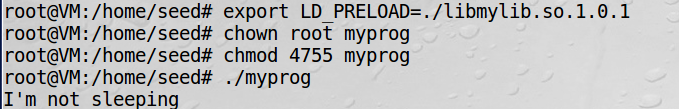
$ /home/seed/ls 的效果，如上图

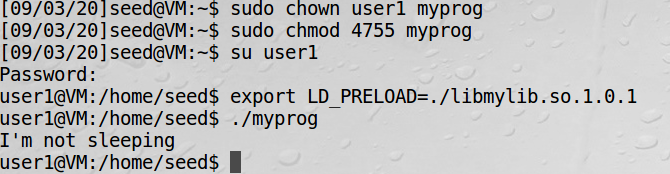


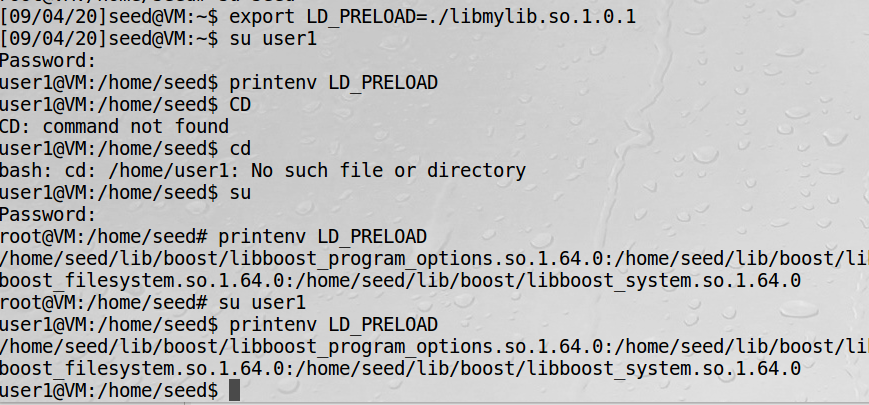
Dynamic linker turn to libmylib.so.1.0.1 to search the program,consistent with situation thought to be.



Nothing happened





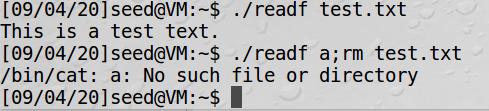


I changed the LD\_PRELOAD in seed account, by found it didn’t change in other two accounts.

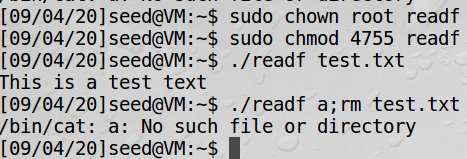
My thought:

Each account have different environment variables LD\_PRELOAD, and when running the program,

Linux will search its owner’s LD\_PRELOAD.

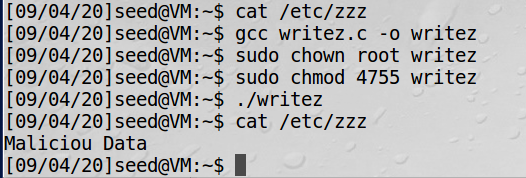


Test.txt被删除



Test.txt也被成功删除？

可能是太懒将test.txt放在seed目录中导致



Zzz文件本来没有东西，子进程成功修改/etc/zzz文件

由此，设计这个程序时，close（fd）要在更改uid之前，更改完uid后马上终止进程，否则会被其他程序利用privilege。