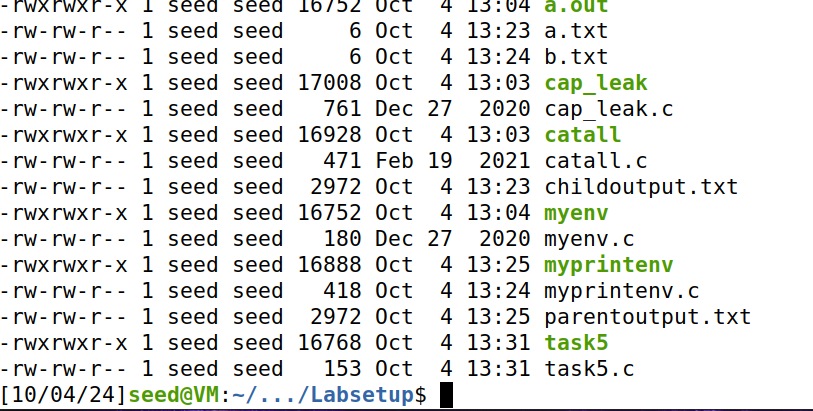
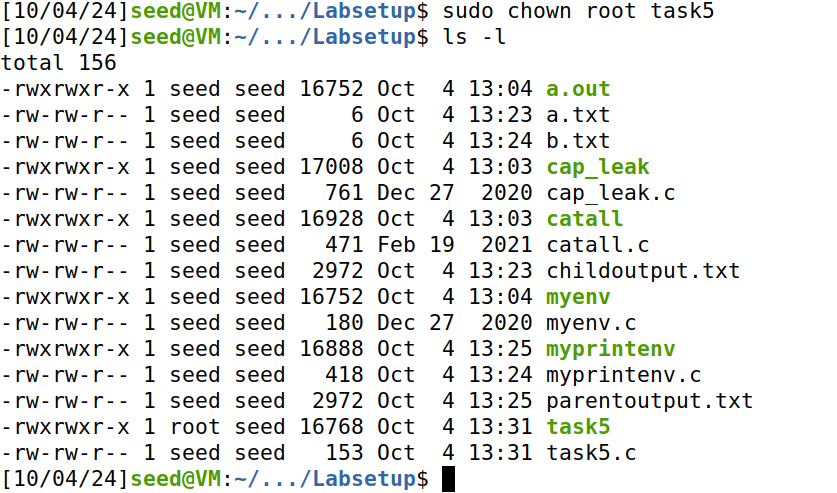
Name: Donghao Li

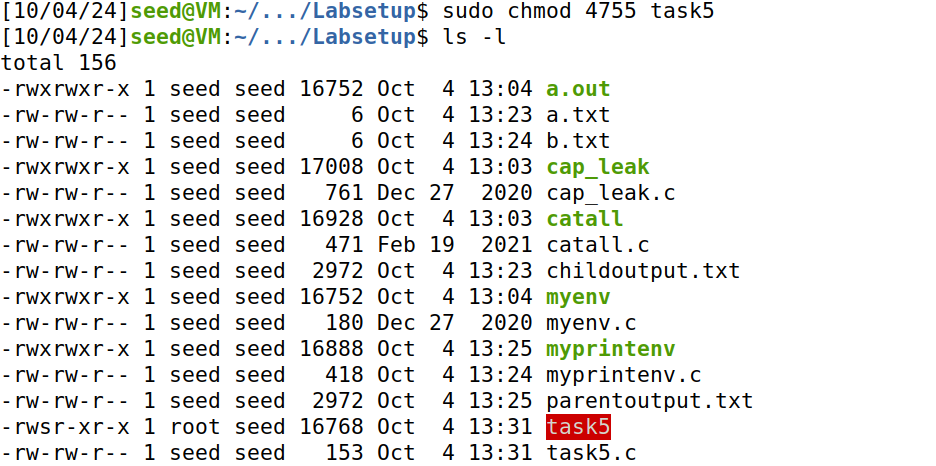
SUID: [dli106@syr.edu](mailto:dli106@syr.edu)

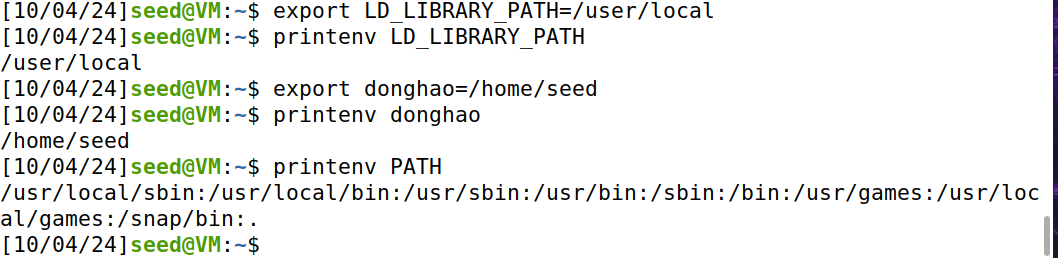
**Task 5: Environment Variable and Set-UID Programs**

Answer: In the figure one, the owner of file task5 is still seed, need to changes to root as figure two shows. And then is to use chmod command to change the task5. 4755 4: a set UID program. rwx(read, write, executable) 0111 is seven in binary so read, write, and executable are all true. r-x means 0101 which is five so rws r-x r-x is 755.





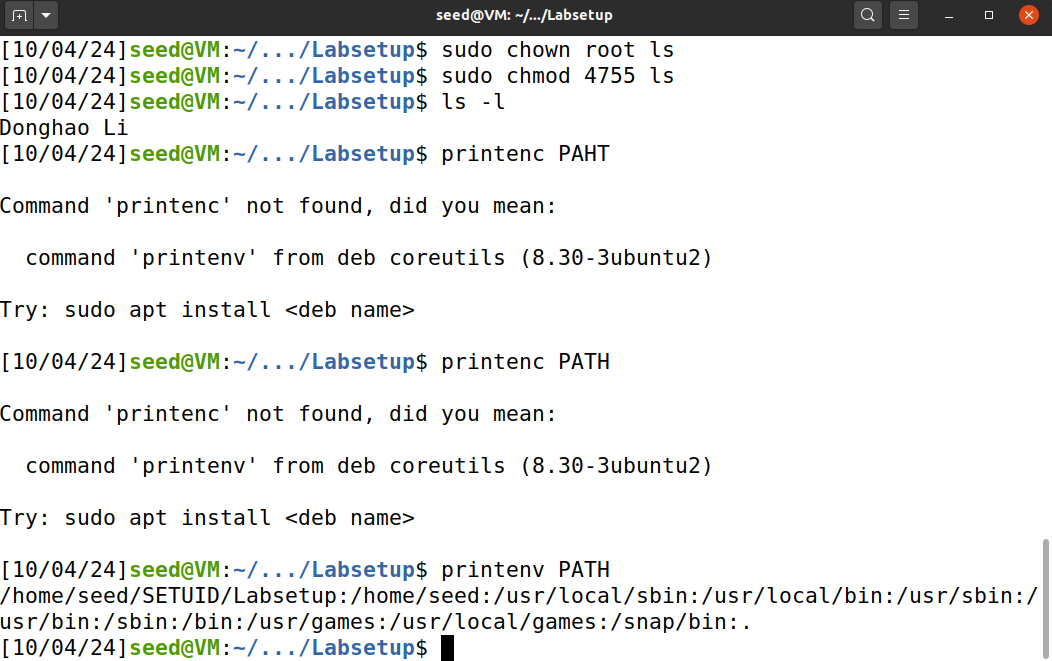


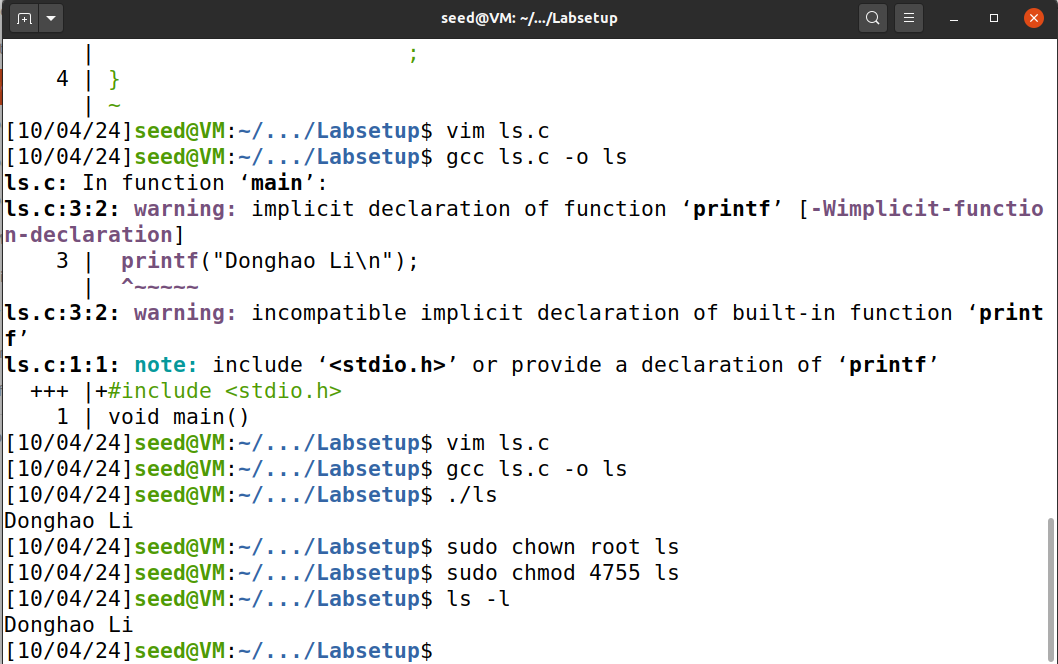




Here only donghao and PATH get displayed in the file task5.txt. There might be some mechanism that force the system to dismiss LD\_LIBRARY\_PATH so that it will not being run in the child process.

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

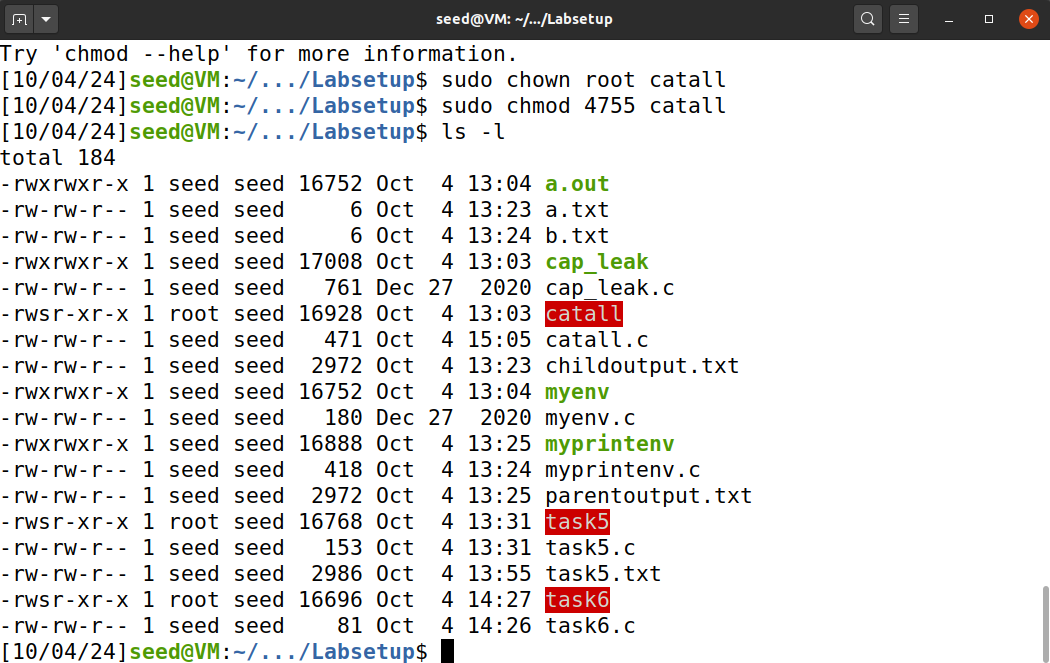


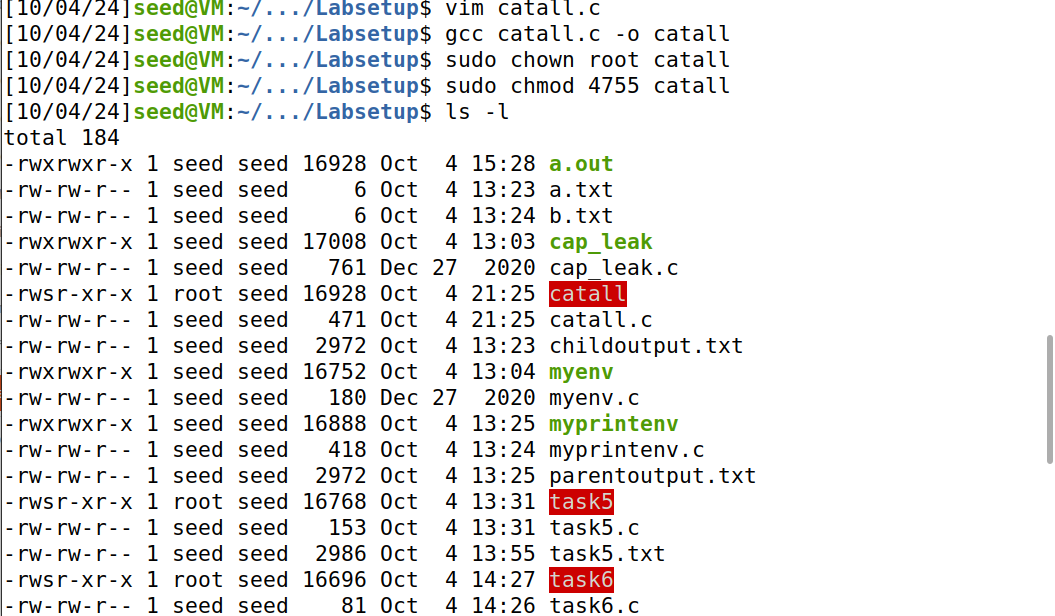


As in figure one shows i first change the privilege of my compiled ls file into root. Then I change the PATH into my folder first so that when it runs ls command it will first check my folder find my compiler ls file run it print Donghao Li then return as the second figure shows.

Meanwhile inorder to overcome the dash shell security policy machanism of dropping SET-UID program we link the /bin/sh to another shell by sudo ln -sf /bin/zsh /bin/sh

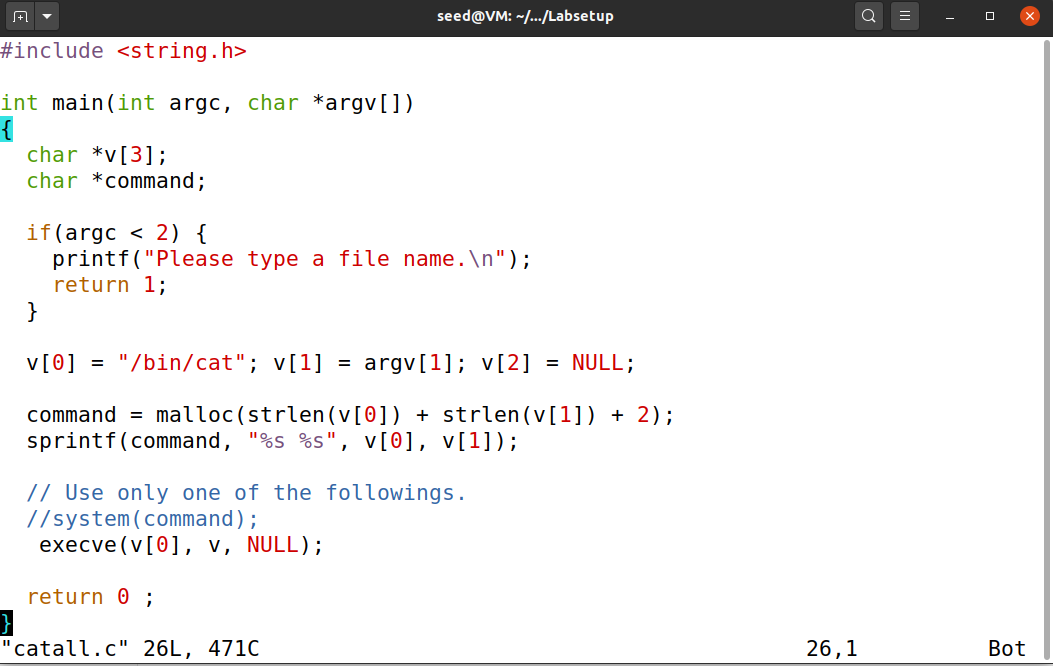
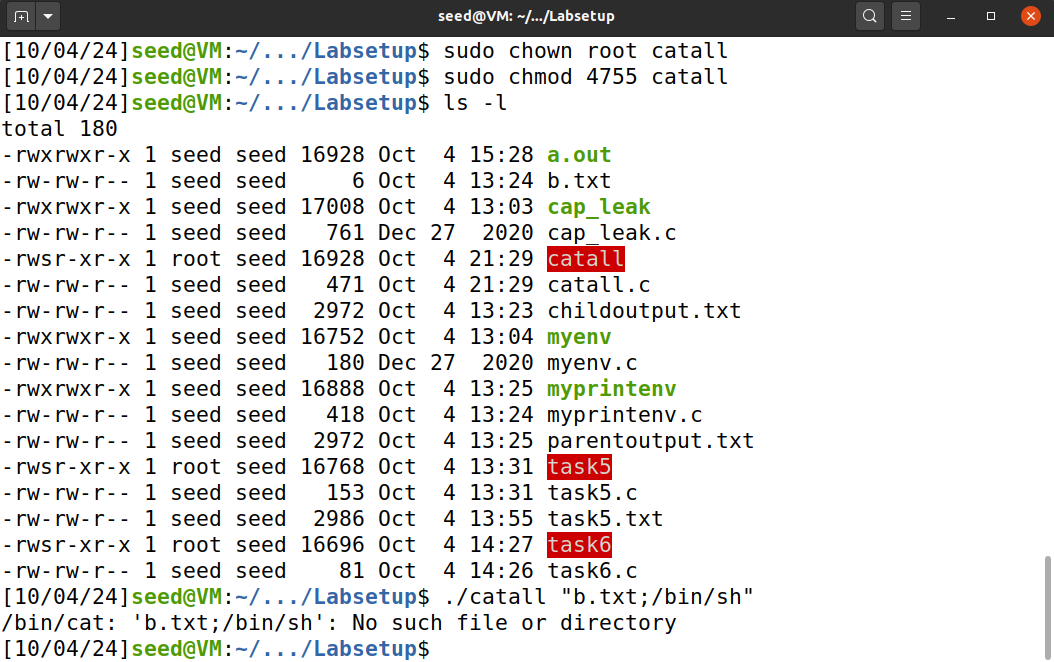
**Task 8: Invoking External Programs Using system() versus execve()**





As in the first figure compiled catall file has been changed to root privilege with executable permission to other users

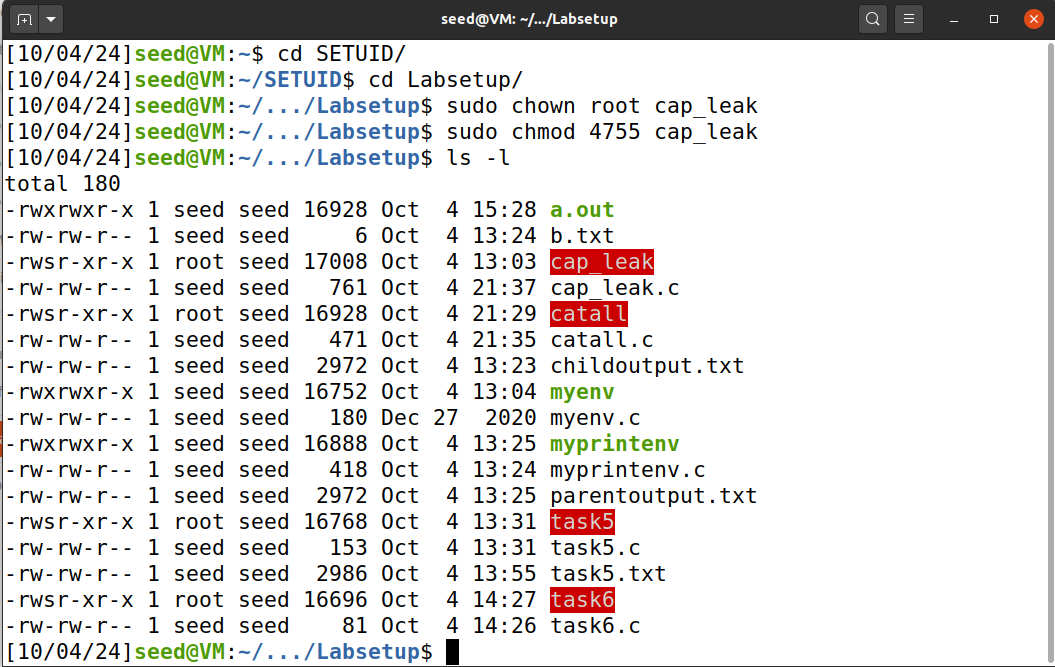
Here when i run catall and use semiculumn to input “a.txt;/bin/sh”, the attacker will be able to first print out the a.txt and then get into the root privilege, then the attacker will be able to run rm command to remove a.txt and exit.

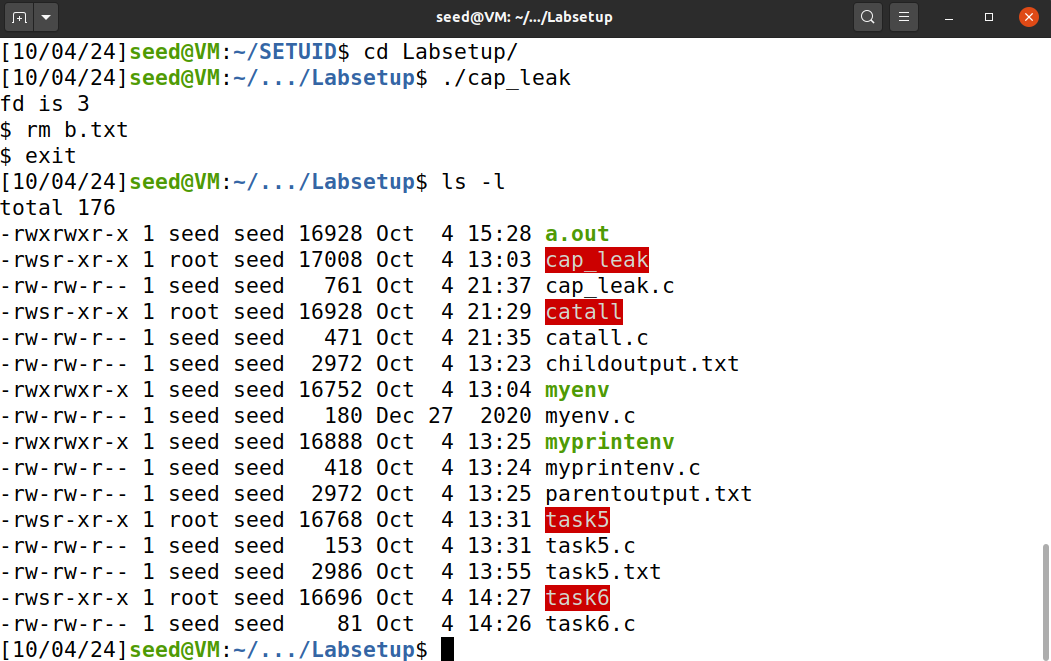


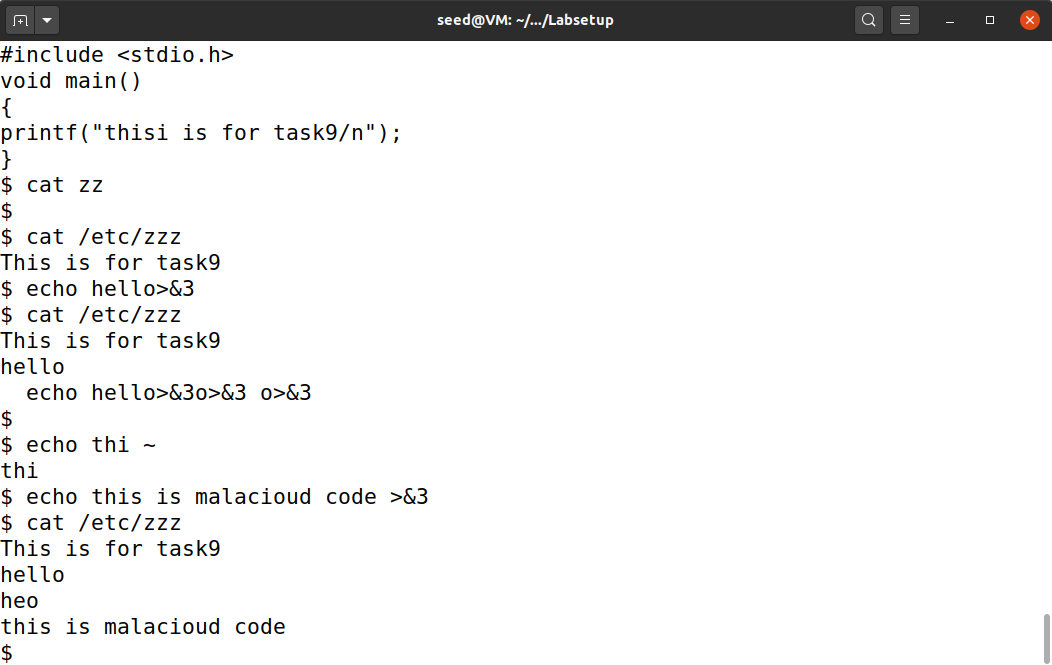
When i use execve() the system call require constructing the command using the input, it will avoid system function in the program and instead use execve function which treat anything inputted from the user as input string and does not allow

it to be run as a command, therefore as the figure shows, the command separated by ; will not be run and system reply can not find such file.

**Task 9: Capability Leaking**

****

****

****

**First as the figure one shows, i compile the cap\_leak.c and set it to root. Then by running cap\_leak i compiler i ll be able to receive the root privilege as a normal user, then i use echo this is malicious code>&3 to input things in a read only file zzz as the figure shows.**