姓名：吴涵　　　　　　　　学号：202101000720　　　　　　　　成绩：

实验 31 共享内存通信

1、共享内存通信

源程序：

（1）shmmutexwrite.c：

```
#include <semaphore.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#include <sys/shm.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <string.h>
#include <unistd.h>
#define BUFFER_SIZE 10
#define sem_name "mysem"
int main()
{
   struct Stu
   {
      char name[10];
      int score;
   };
   int shmid;
   sem_t *sem;
   int score = 60, i = 1;
   char buff[BUFFER_SIZE];
   key_t shmkey;
   shmkey = ftok("shmmutexread.c", 0);
   sem = sem_open(sem_name, O_CREAT, 0644, 1);
   if (sem == SEM_FAILED)
   {
      printf("unable to creat semaphore!");
      sem_unlink(sem_name); // 删除有名信号量
      exit(-1);
   }
   shmid = shmget(shmkey, 1024, 0666 | IPC_CREAT);
   /*创建 IPC 键值为 shmkey 的共享内存，其大小为 1024 字节，允许读写*/
   if (shmid == -1)
      printf("creat shm is fail\n");
   struct Stu *addr;
   addr = (struct Stu *)shmat(shmid, 0, 0);
   if (addr == (struct Stu *)-1)
      printf("shm shmat is fail\n");
   addr->score = 0;
   printf("写进程映射的共享内存地址=%p\n", addr);
   do
   {
      sem_wait(sem);
      memset(buff, 0, BUFFER_SIZE);
      memset((addr + i)->name, 0, BUFFER_SIZE);
      printf("写进程:输入一些姓名（不超过 10 个字符）到共享内存(输入'quit' 退出):\n");
      if (fgets(buff, BUFFER_SIZE, stdin) == NULL)
      {
         sem_post(sem);
         break;
      }
      strncpy((addr + i)->name, buff, strlen(buff) - 1);
      (addr + i)->score = ++score;
```

```
        addr->score++;
        i++;
        sem_post(sem);
        sleep(1);
    } while (strncmp(buff, "quit", 4) != 0);
    if (shmdt(addr) == -1) /*将共享内存与当前进程断开*/
        printf("shmdt is fail\n");
    sem_close(sem);      // 关闭有名信号量
    sem_unlink(sem_name); // 删除有名信号量
}
```

（2）shmmutexread.c：

```
#include <semaphore.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#include <sys/shm.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <string.h>
#define sem_name "mysem"
int main()
{
    int shmid;
    sem_t *sem;
    int i = 1;
    key_t shmkey;
    shmkey = ftok("shmmutexread.c", 0);
    struct Stu
    {
        char name[10];
        int score;
    };
    sem = sem_open(sem_name, 0, 0644, 0);
    if (sem == SEM_FAILED)

    {
        printf("unable to open semaphore!");
        sem_close(sem);
        exit(-1);
    }
    shmid = shmget(shmkey, 0, 0666);
    if (shmid == -1)
    {
        printf("creat shm is fail\n");
        exit(0);
    }
    struct Stu *addr;
    addr = (struct Stu *)shmat(shmid, 0, 0);
    if (addr == (struct Stu *)-1)
    {
        printf("shm shmat is fail\n");
        exit(0);
    }
    printf("读进程映射的共享内存地址=%p\n", addr);
    do
    {
        sem_wait(sem);
        if (addr->score > 0)

        {
            printf("\n 读进程:绑定到共享内存 %p:姓名 %d   %s , 分值%d \n", addr, i, (addr +
i)->name, (addr + i)->score);
            addr->score--;
            if (strncmp((addr + i)->name, "quit", 4) == 0)
                break;
            i++;
        }
        sem_post(sem);
    } while (1);
    sem_close(sem);
    if (shmdt(addr) == -1)
        printf("shmdt is fail\n");
    if (shmctl(shmid, IPC_RMID, NULL) == -1)
        printf("shmctl delete error\n");
}
```

编译链接命令：

gcc shmmutexwrite.c -o shmmutexwrite -lpthread

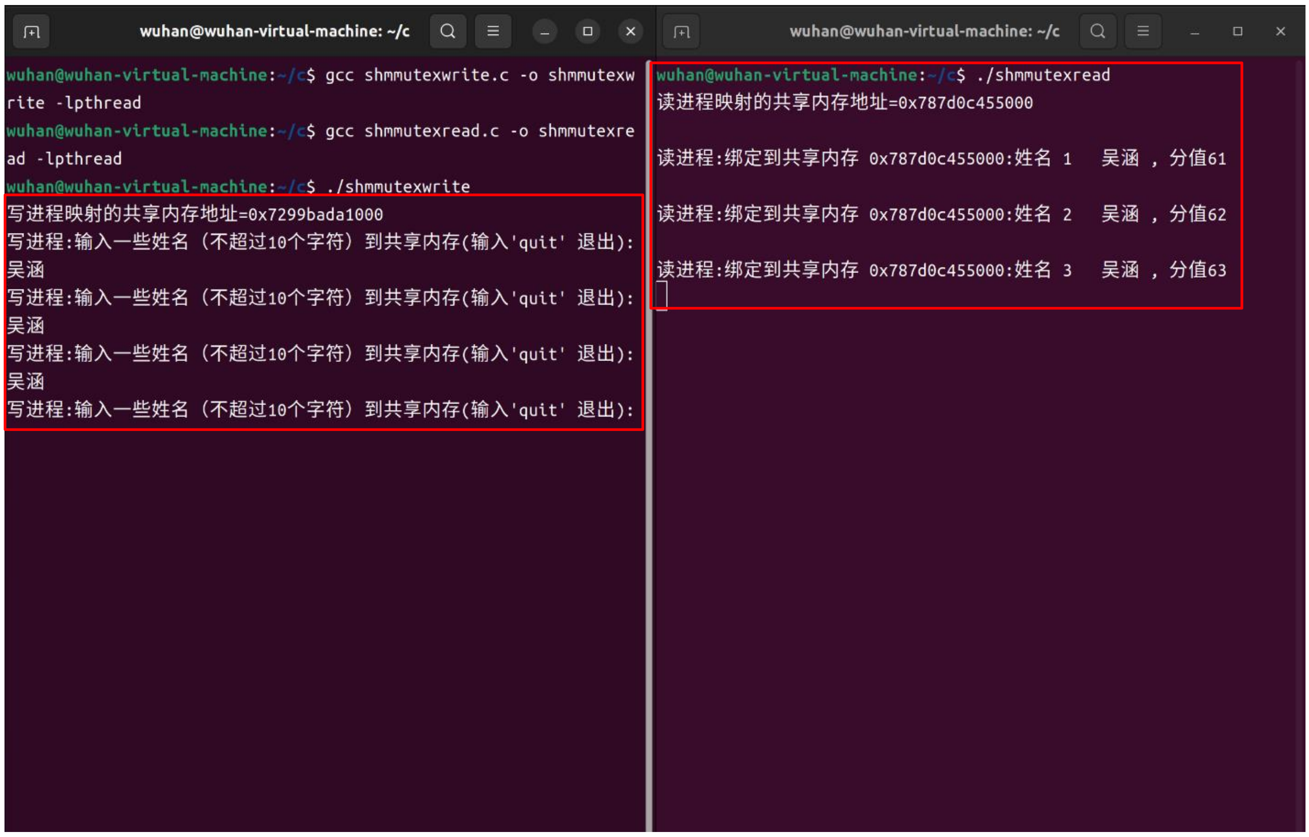
gcc shmmutexread.c -o shmmutexread –lpthread

运行命令：

./shmmutexwrite

./shmmutexread

交互与结果：



```
wuhan@wuhan-virtual-machine:~/c$ gcc shmmutexwrite.c -o shmmutexw
rite -lpthread
wuhan@wuhan-virtual-machine:~/c$ gcc shmmutexread.c -o shmmutexre
ad -lpthread
wuhan@wuhan-virtual-machine:~/c$ ./shmmutexwrite
写进程映射的共享内存地址=0x7299bada1000
写进程:输入一些姓名（不超过10个字符）到共享内存(输入'quit' 退出):
吴涵
写进程:输入一些姓名（不超过10个字符）到共享内存(输入'quit' 退出):
吴涵
写进程:输入一些姓名（不超过10个字符）到共享内存(输入'quit' 退出):
吴涵
写进程:输入一些姓名（不超过10个字符）到共享内存(输入'quit' 退出):
```

```
wuhan@wuhan-virtual-machine:~/c$ ./shmmutexread
读进程映射的共享内存地址=0x787d0c455000

读进程:绑定到共享内存 0x787d0c455000:姓名 1   吴涵 ，分值61

读进程:绑定到共享内存 0x787d0c455000:姓名 2   吴涵 ，分值62

读进程:绑定到共享内存 0x787d0c455000:姓名 3   吴涵 ，分值63
```