code/conc/sbuf.c

```
#include "csapp.h"
#include "sbuf.h"

/* Create an empty, bounded, shared FIFO buffer with n slots */
void sbuf_init(sbuf_t *sp, int n)
{
    sp->buf = Calloc(n, sizeof(int));
    sp->n = n;                         /* Buffer holds max of n items */
    sp->front = sp->rear = 0;          /* Empty buffer iff front == rear */
    Sem_init(&sp->mutex, 0, 1);        /* Binary semaphore for locking */
    Sem_init(&sp->slots, 0, n);        /* Initially, buf has n empty slots */
    Sem_init(&sp->items, 0, 0);        /* Initially, buf has zero data items */
}

/* Clean up buffer sp */
void sbuf_deinit(sbuf_t *sp)
{
    Free(sp->buf);
}

/* Insert item onto the rear of shared buffer sp */
void sbuf_insert(sbuf_t *sp, int item)
{
    P(&sp->slots);                          /* Wait for available slot */
    P(&sp->mutex);                          /* Lock the buffer */
    sp->buf[(++sp->rear)%(sp->n)] = item;   /* Insert the item */
    V(&sp->mutex);                          /* Unlock the buffer */
    V(&sp->items);                          /* Announce available item */
}

/* Remove and return the first item from buffer sp */
int sbuf_remove(sbuf_t *sp)
{
    int item;
    P(&sp->items);                          /* Wait for available item */
    P(&sp->mutex);                          /* Lock the buffer */
    item = sp->buf[(++sp->front)%(sp->n)];  /* Remove the item */
    V(&sp->mutex);                          /* Unlock the buffer */
    V(&sp->slots);                          /* Announce available slot */
    return item;
}
```

code/conc/sbuf.c

图 12-25　SBUF：同步对有限缓冲区并发访问的包

练习题 12.9　设 p 表示生产者数量，c 表示消费者数量，而 n 表示以项目单元为单位的缓冲区大小。对于下面的每个场景，指出 sbuf_insert 和 sbuf_remove 中的互斥锁信号量是否是必需的。

A. $p=1$，$c=1$，$n>1$

B. $p=1$，$c=1$，$n=1$

C. $p>1$，$c>1$，$n=1$

2. 读者-写者问题

读者-写者问题是互斥问题的一个概括。一组并发的线程要访问一个共享对象，例如