1. producer-consumer with queue

void sbuf\_insert(sbuf\_t \*sp, int item)

{

int next = sp->rear+1;

if(next == sp->n) next = 0;

while(next == sp->front) pthread\_yield();

sp->buf[sp->rear] = item;

sp->rear = next;

}

int sbuf\_remove(sbuf\_t \*sp)

{

while(sp->front == sp->rear) pthread\_yield();

int next = sp->front+1;

if(next == sp->n) next =0;

int val = sp->buf[sp->front];

sp->front = next;

return val;

}

2. second readers-writers problem

int readcount, writecount; //(initial value = 0)

semaphore rmutex, wmutex, readTry, resource; //(initial value = 1)

reader() {

readTry.P();

rmutex.P();

readcount++;

if (readcount == 1) resource.P();

rmutex.V();

readTry.V();

//reading

rmutex.P();

readcount--;

if (readcount == 0) resource.V();

rmutex.V();

}

writer() {

wmutex.P();

writecount++;

if (writecount == 1) readTry.P();

wmutex.V();

resource.P();

//writing

resource.V();

wmutex.P();

writecount--;

if (writecount == 0) readTry.V();

wmutex.V();

}