```c
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <pthread.h>
#include "semaphore.h"

Semaphore *make_semaphore (int value)
{
  Semaphore *semaphore = malloc (sizeof(Semaphore));
  semaphore->value = value;
  semaphore->wakeups = 0;
  pthread_mutex_init (semaphore->pmutex, NULL);
  semaphore->cond = make_cond ();
  return semaphore;
}

void semaphore_wait (Semaphore *semaphore)
{
  pthread_mutex_lock (semaphore->pmutex);
  semaphore->value--;

  if (semaphore->value < 0) {
    do {
      cond_wait (semaphore->cond, semaphore->pmutex);
    } while (semaphore->wakeups < 1);
    semaphore->wakeups--;
  }
  pthread_mutex_unlock (semaphore->pmutex);
}

void semaphore_signal (Semaphore *semaphore)
{
  pthread_mutex_lock (semaphore->pmutex);
  semaphore->value++;

  if (semaphore->value <= 0) {
    semaphore->wakeups++;
    cond_signal (semaphore->cond);
  }
  pthread_mutex_unlock (semaphore->pmutex);
}
```