```c
/* Ejemplo de variables de condicion */

#define _POSIX_C_SOURCE 199506L
#define _REENTRANT

#include <stdio.h>
#include <pthread.h>
#include <time.h>

#define NUM_THREADS  4
#define TCOUNT 10        /* Numero de incrementos realizados por inc_cuenta */
#define COUNT_THRES 12   /* Nivel que dispara la condicion */

  /* Variable compartida */

  int cuenta = 0;

  /* mutex y variable de condicion */

  pthread_mutex_t lock =     PTHREAD_MUTEX_INITIALIZER;
  pthread_cond_t cuenta_ok = PTHREAD_COND_INITIALIZER;

  /* Identificadores de cada thread */

  int thread_ids[4] = {0,1,2,3};

  /* Opciones, que no se tocan mientras funcionan los 4 threads */

  int usa_timed;
  int usa_broad;

/* Hilo que incrementa la cuenta */

void *inc_cuenta(void *idp)
{
  int i=0;
  int *my_id = idp;
  struct timespec tim = {0, 1000000L};

  printf("inc_cuenta: thread %d\n", *my_id);

  for (i=0; i<TCOUNT; i++)
  {
    pthread_mutex_lock(&lock);
    cuenta++;
    printf("inc_cuenta: thread %d, cuenta = %d\n",
         *my_id, cuenta);

    if(cuenta == COUNT_THRES)
    {
      printf("inc_cuenta: Thread %d, cuenta %d; alcanzado nivel.\n",
      *my_id, cuenta);

      if(usa_broad) pthread_cond_broadcast(&cuenta_ok);
      else          pthread_cond_signal(&cuenta_ok);
    }
    pthread_mutex_unlock(&lock);

    nanosleep(&tim, NULL);
  }

  return(NULL);
}

/* Espera pacientemente que se alcance la cuenta */

void *mira_cuenta(void *idp)
{
  int i=0;
  int *my_id = idp;
  struct timespec timout;
  struct timespec tim = {0, 1000000L};
  int res;
  int cuenta_leida;

  printf("mira_cuenta: thread %d\n", *my_id);

  pthread_mutex_lock(&lock);

  while (cuenta < COUNT_THRES)
  {
    printf("thread %d esperando.\n", *my_id);

    if(usa_timed)
    {
      /* Timeout de 1 segundos */

      clock_gettime(CLOCK_REALTIME, &timout);
      timout.tv_sec += 1;
      res = pthread_cond_timedwait(&cuenta_ok, &lock, &timout);
      if(res)
      {
        printf("%s\n", strerror(res));
        printf("Soy %d; ya me he hartado de esperar.\n", *my_id);

        /* Si sale sin liberar el mutex, ya no puede entrar nadie mas */

        pthread_mutex_unlock(&lock);
        pthread_exit((void *)res);
      }
    }
    else res = pthread_cond_wait(&cuenta_ok, &lock);
  }

  /* Ya tiene la condicion garantizada, y el mutex */

  cuenta_leida = cuenta;
  pthread_mutex_unlock(&lock);

  printf("mira_cuenta: thread %d, cuenta leida %d\n", *my_id, cuenta_leida);

  return(NULL);
}

/* Hilo main */

void prueba(void);

int void main(void)
{
  printf("\nPrueba sin broadcast y sin temporizador\n");
  usa_broad = 0;
  usa_timed = 0;
  prueba();

  printf("\nPrueba con broadcast y sin temporizador\n");
  usa_broad = 1;
  usa_timed = 0;
  prueba();

  printf("\nPrueba sin broadcast y con temporizador\n");
  usa_broad = 0;
  usa_timed = 1;
  prueba();

  printf("\nPrueba con broadcast y con temporizador\n");
  usa_broad = 1;
  usa_timed = 1;
  prueba();
}
```

```
/*  Realizacion de una de las pruebas */

void prueba(void)
{
  int i;
  pthread_t threads[4];  /* Ids de los hilos */

  cuenta = 0;

  /* Se activan primero los que esperan, para que se queden bloqueados
     con mas seguridad */

  /* Cada thread recibe como argumento un puntero a un identificador  */

  pthread_create(&threads[2], NULL, mira_cuenta, (void *)&thread_ids[2]);
  pthread_create(&threads[3], NULL, mira_cuenta, (void *)&thread_ids[3]);
  pthread_create(&threads[0], NULL, inc_cuenta, (void *)&thread_ids[0]);
  pthread_create(&threads[1], NULL, inc_cuenta, (void *)&thread_ids[1]);

  /* Espera un rato y luego cancela los hilos */

  sleep(2);

  printf("prueba: esperando terminacion de los hilos\n");

  for (i = 0; i < NUM_THREADS; i++)
  {
    pthread_cond_signal(&cuenta_ok); /* Por si alguno se queda esperando */
    pthread_join(threads[i], NULL);
  }

  printf("prueba: han terminado todos\n");
}
```

```
Prueba sin broadcast y sin temporizador
mira_cuenta: thread 2
thread 2 esperando.
mira_cuenta: thread 3
thread 3 esperando.
inc_cuenta: thread 0
inc_cuenta: thread 0, cuenta = 1
inc_cuenta: thread 1
inc_cuenta: thread 1, cuenta = 2
inc_cuenta: thread 0, cuenta = 3
inc_cuenta: thread 1, cuenta = 4
inc_cuenta: thread 1, cuenta = 5
inc_cuenta: thread 0, cuenta = 6
inc_cuenta: thread 0, cuenta = 7
inc_cuenta: thread 1, cuenta = 8
inc_cuenta: thread 1, cuenta = 9
inc_cuenta: thread 0, cuenta = 10
inc_cuenta: thread 0, cuenta = 11
inc_cuenta: thread 1, cuenta = 12
inc_cuenta: Thread 1, cuenta 12; alcanzado nivel.
mira_cuenta: thread 2, cuenta leida 12
inc_cuenta: thread 1, cuenta = 13
inc_cuenta: thread 0, cuenta = 14
inc_cuenta: thread 0, cuenta = 15
inc_cuenta: thread 1, cuenta = 16
inc_cuenta: thread 1, cuenta = 17
inc_cuenta: thread 0, cuenta = 18
inc_cuenta: thread 0, cuenta = 19
inc_cuenta: thread 1, cuenta = 20
prueba: esperando terminacion de los hilos
mira_cuenta: thread 3, cuenta leida 20
prueba: han terminado todos

Prueba con broadcast y sin temporizador
mira_cuenta: thread 2
thread 2 esperando.
mira_cuenta: thread 3
thread 3 esperando.
inc_cuenta: thread 0
inc_cuenta: thread 0, cuenta = 1
inc_cuenta: thread 1
inc_cuenta: thread 1, cuenta = 2
inc_cuenta: thread 1, cuenta = 3
inc_cuenta: thread 0, cuenta = 4
inc_cuenta: thread 0, cuenta = 5
inc_cuenta: thread 1, cuenta = 6
inc_cuenta: thread 1, cuenta = 7
inc_cuenta: thread 0, cuenta = 8
inc_cuenta: thread 0, cuenta = 9
inc_cuenta: thread 1, cuenta = 10
inc_cuenta: thread 1, cuenta = 11
inc_cuenta: thread 0, cuenta = 12
inc_cuenta: Thread 0, cuenta 12; alcanzado nivel.
mira_cuenta: thread 2, cuenta leida 12
mira_cuenta: thread 3, cuenta leida 12
inc_cuenta: thread 0, cuenta = 13
inc_cuenta: thread 1, cuenta = 14
inc_cuenta: thread 1, cuenta = 15
inc_cuenta: thread 0, cuenta = 16
inc_cuenta: thread 0, cuenta = 17
inc_cuenta: thread 1, cuenta = 18
inc_cuenta: thread 1, cuenta = 19
inc_cuenta: thread 0, cuenta = 20
prueba: esperando terminacion de los hilos
prueba: han terminado todos

Prueba sin broadcast y con temporizador
mira_cuenta: thread 2
thread 2 esperando.
mira_cuenta: thread 3
thread 3 esperando.
inc_cuenta: thread 0
inc_cuenta: thread 0, cuenta = 1
inc_cuenta: thread 1
inc_cuenta: thread 1, cuenta = 2
inc_cuenta: thread 1, cuenta = 3
inc_cuenta: thread 0, cuenta = 4
inc_cuenta: thread 0, cuenta = 5
inc_cuenta: thread 1, cuenta = 6
inc_cuenta: thread 1, cuenta = 7
inc_cuenta: thread 0, cuenta = 8
inc_cuenta: thread 0, cuenta = 9
inc_cuenta: thread 1, cuenta = 10
inc_cuenta: thread 1, cuenta = 11
inc_cuenta: thread 0, cuenta = 12
inc_cuenta: Thread 0, cuenta 12; alcanzado nivel.
mira_cuenta: thread 2, cuenta leida 12
inc_cuenta: thread 0, cuenta = 13
inc_cuenta: thread 1, cuenta = 14
inc_cuenta: thread 1, cuenta = 15
inc_cuenta: thread 0, cuenta = 16
inc_cuenta: thread 0, cuenta = 17
inc_cuenta: thread 1, cuenta = 18
inc_cuenta: thread 1, cuenta = 19
inc_cuenta: thread 0, cuenta = 20
Connection timed out
Soy 3; ya me he hartado de esperar.
prueba: esperando terminacion de los hilos
prueba: han terminado todos

Prueba con broadcast y con temporizador
mira_cuenta: thread 2
thread 2 esperando.
mira_cuenta: thread 3
thread 3 esperando.
inc_cuenta: thread 0
inc_cuenta: thread 0, cuenta = 1
inc_cuenta: thread 1
inc_cuenta: thread 1, cuenta = 2
inc_cuenta: thread 1, cuenta = 3
inc_cuenta: thread 0, cuenta = 4
inc_cuenta: thread 0, cuenta = 5
inc_cuenta: thread 1, cuenta = 6
inc_cuenta: thread 1, cuenta = 7
inc_cuenta: thread 0, cuenta = 8
inc_cuenta: thread 0, cuenta = 9
inc_cuenta: thread 1, cuenta = 10
inc_cuenta: thread 1, cuenta = 11
inc_cuenta: thread 0, cuenta = 12
inc_cuenta: Thread 0, cuenta 12; alcanzado nivel.
mira_cuenta: thread 2, cuenta leida 12
mira_cuenta: thread 3, cuenta leida 12
inc_cuenta: thread 0, cuenta = 13
inc_cuenta: thread 1, cuenta = 14
inc_cuenta: thread 1, cuenta = 15
inc_cuenta: thread 0, cuenta = 16
inc_cuenta: thread 0, cuenta = 17
inc_cuenta: thread 1, cuenta = 18
inc_cuenta: thread 1, cuenta = 19
inc_cuenta: thread 0, cuenta = 20
prueba: esperando terminacion de los hilos
prueba: han terminado todos
```