

```
/* Prueba de la cola de espera */

#define _POSIX_C_SOURCE 199506L
#define _REENTRANT

#include <unistd.h>
#include <pthread.h>
#include <stdio.h>
#include <errno.h>
#include <string.h>

#include "cola.h" /* Prototipos de las funciones de acceso a la cola */

#define NLEC 2
#define NESC 2

void *escriptor(void *);
void *lector(void *);

int main(void)
{
    pthread_t lectores[NLEC];
    pthread_t escritores[NESC];
    int i;
    int res;

    /* Activando lectores y escritores */

    printf("activando lectores\n");

    for(i=0; i< NLEC; i++)
        pthread_create(lectores + i, NULL, lector, (void *)i);

    printf("activando escritores\n");

    for(i=0; i< NESC; i++)
        res = pthread_create(escritores + i, NULL, escritor, (void *)i);

    /* Esperando a que los escritores acaben */

    for(i=0; i< NESC; i++)
        pthread_join(escritores[i], NULL);

    /* Espera para que los lectores terminen de leer lo que hay y
       se queden bloqueados ante una cola vacia */

    sleep(2);
}

/* Escritor */

#define NDATO 15

void *escriptor(void *pn)
{
    int id = (int)pn;
    int dato;
    int i;
    struct timespec retraso = {0, 200000000L};

    printf("escritor %d activado\n", id);

    dato = 100*id;
    for(i=0; i<NDATO; i++, dato++)
```

```
    {
        printf("Escritor %d escribiendo %d\n", id, dato);
        pon_dato(dato);
        nanosleep(&retraso, NULL);
    }
}

/* Lector */

void *lector(void *pn)
{
    int id = (int)pn;
    int dato;
    struct timespec retraso = {0, 400000000L};

    printf("Lector %d activado\n", id);

    while(1)
    {
        printf("Lector %d va a leer\n", id);
        extrae_dato(&dato);
        printf("Lector %d ha leído %d\n", id, dato);
        nanosleep(&retraso, NULL);
    }
}

/* Cabecera para una cola basada en variables compartidas */

void pon_dato(int dato);
void extrae_dato(int *dato);
```