



Ejercicio

Realizar un programa que cree dos threads.

El thread 1 debe imprimir en pantalla los números impares desde el 1 al 9 con el mensaje th1: número

El thread 2 debe imprimir en pantalla los números pares desde el 2 al 10 con el mensaje th2: número

La ejecución de los threads se debe producir de forma alterna de forma que los mensajes mostrados tengan el formato

```
th1: 1
th2: 2
th1: 3
th2: 4
th1: 5
th2: 6
th1: 7
th2: 8
th1: 9
th2: 10
```

La sincronización se debe realizar utilizando mutex. No se podrá utilizar sleep ni ningún otro mecanismo de sincronización.

Solución

```
/* fichero alterno.c
THREADS
Crea 2 threads que ejecutan de forma alterna.
compilar con gcc -lpthread alterno.c
José Manuel Pérez Lobato
*/

#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

pthread_attr_t attr;
pthread_t idth[10];
pthread_mutex_t mtx;
pthread_cond_t varcond;
int turno=1;

void *hilol(void *num) {
    int cont=1;

    while (cont <=10){
        pthread_mutex_lock (&mtx);
        while (turno!=1) pthread_cond_wait(&varcond, &mtx);
        printf ("th1:%d\n", cont);
        cont=cont+2;
    }
}
```

Tema: 7

Este material se comparte con licencia Creative Commons Reconocimiento no Comercial-Compartida-Igual 3.0. España.



```
        turno = 2;
        pthread_cond_signal(&varcond);
        pthread_mutex_unlock (&mtx);
    }
    pthread_exit(0);
}
void *hilo2(void *num) {
    int cont=2;

    while (cont <=10){
        pthread_mutex_lock (&mtx);
        while (turno!=2) pthread_cond_wait(&varcond, &mtx);
        printf ("th2:%d\n", cont);
        cont=cont+2;
        turno = 1;
        pthread_cond_signal(&varcond);
        pthread_mutex_unlock (&mtx);
    }
    pthread_exit(0);
}

int main(){
    int i;

    pthread_mutex_init (&mtx, NULL);
    pthread_attr_init(&attr);

    pthread_create(&idth[0], &attr, hilo1, &i);
    pthread_create(&idth[1], &attr, hilo2, &i);
    // Espero la finalizaci3n del thread
    for (i=0; i<2; i++)
        pthread_join(idth[i], NULL);
    return(0);
}
```