```
// creahilos.c

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>

/*
Ejemplo simple de procesos livianos (threads) hilos
compilar como:
cc  creahilos.c -lpthread

ejecutar con argumento de entrada
por ejemplo

./a.out 5

*/

void *hilo();

int main(int argc,char **argv)
{
	int canthilos = atoi(argv[1]) ;
	int i ;

	pthread_t hilos[300];
	for ( i = 0 ; i < canthilos ; i++)
	  {
	    pthread_create(&hilos[i],NULL,hilo,NULL);
	    printf("main(): lanzo hilo i=%d \n",i);
	  }
	for ( i = 0 ; i < canthilos ; i++)
	  {
	   pthread_join(hilos[i],NULL);
	       printf("Hilo i=%d fin\n", i);
	  }
	   printf("main(): fin\n");
	   return 0;
}

void *hilo()
{
	char letra = 'a';
	   while(letra <= 'z')
	    {
	   printf(" %c escrita por el hilo %d\n",letra,pthread_self());
	      letra=letra+1;
	      sleep(1);
	    }
```

```
        pthread_exit(0);
}
```

```
// sincro.c

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>

void *funhilo1();
void *funhilo2();

pthread_mutex_t m1 = PTHREAD_MUTEX_INITIALIZER;
pthread_mutex_t m2 = PTHREAD_MUTEX_INITIALIZER;

int main(int argc,char *argv[])
{
     pthread_t hilo1;
     pthread_t hilo2;

     pthread_mutex_lock(&m2);

     pthread_create(&hilo1,NULL,funhilo1,NULL);
     printf(" hilo 1 creado \n");
     pthread_create(&hilo2,NULL,funhilo2,NULL);
     printf(" hilo 2 creado \n");

     pthread_join(hilo1,NULL);
     printf(" hilo 1 terminado \n");
     pthread_join(hilo2,NULL);
     printf(" hilo 2 terminado \n");

     printf(" Creador pid=%d termina \n",getpid());
     exit(0);
}

void * funhilo1()
{
  char letra = 'a';
  while(letra <= 'y')
   {
      pthread_mutex_lock(&m1);
      printf(" hilo 1 %d escibe %c \n",pthread_self(),letra);
      letra = letra + 2;
      pthread_mutex_unlock(&m2);
   }
   pthread_exit(0);
}
```

```
void * funhilo2()
{
  char letra = 'B';
  while(letra <= 'Z')
   {
      pthread_mutex_lock(&m2);
      printf("        hilo 2 %d escibe %c \n",pthread_self(),letra);
      letra = letra + 2;
      pthread_mutex_unlock(&m1);
   }
   pthread_exit(0);
}
```

```
// sincro1.c

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>


void *funhilo1();
void *funhilo2();

pthread_mutex_t m1 = PTHREAD_MUTEX_INITIALIZER ;
pthread_mutex_t m2 = PTHREAD_MUTEX_INITIALIZER ;

char letra = 'A';

int main(int argc,char **argv)
{

        pthread_t hilo1;
        pthread_t hilo2;

        pthread_mutex_lock(&m2);

        pthread_create(&hilo1,NULL,funhilo1,NULL);
        printf("main(): lanzo hilo1 \n");
        pthread_create(&hilo2,NULL,funhilo2,NULL);
        printf("main(): lanzo hilo2 \n");

        pthread_join(hilo1,NULL);
        printf("Hilo 1 : fin\n");
        pthread_join(hilo2,NULL);
        printf("Hilo 2 : fin\n");
        return 0;
}
```

```
void *funhilo1()
{
      while(letra <= 'y')
           {
             pthread_mutex_lock(&m1);
             letra = letra + 32;
          printf("Hilo 1 : %c\n",letra);
             letra = letra + 1;
             sleep(1);
             pthread_mutex_unlock(&m2);
           }
          pthread_exit(0);
}

void *funhilo2()
{
          do
           {
             pthread_mutex_lock(&m2);
             letra = letra - 32;
             printf("    Hilo 2 : %c\n",letra);
             letra = letra + 1;
             sleep(1);
             pthread_mutex_unlock(&m1);
           } while(letra != 'Z'+1);
          pthread_exit(0);
}
```