HILOS EN C

Héctor Menéndez

AIDA Research Group
Computer Science Department
Universidad Autónoma de Madrid

29 de enero de 2013

Crear un hilo POSIX

```
#include <pthread.h>
int pthread_create (pthread_t *thread_id,
 const pthread_attr_t *attributes,
 void *(*thread_function)(void *), void *arguments);
```

- Esta función crea un nuevo hilo.
- pthread_t es un tipo que actúa como un manejador para el hilo nuevo.
- attributes es otro tipo que permite añadir parámetros, para usar los parámetros por defecto poner NULL.
- thread_function es la función del nuevo hilo que se está ejecutando. El hilo terminará cuando esta función finalice salvo que se le mate explícitamente.
- arguments es un puntero void que se pasa como único argumento a thread_function.

Salir de un hilo POSIX

```
#include <pthread.h>
int pthread_exit (void *status);
```

- Termina la llamada a un hilo explícitamente (dentro del propio hilo).
- status es el valor de retorno del hilo.

Esperar un hilo POSIX

```
#include <pthread.h>
int pthread_join (pthread_t thread, void **status_ptr);
```

- Hace que un hilo espere hasta la terminación de otro para terminar.
- status es el valor de retorno del hilo.

Ejemplo Hilos

```c
#include <stdio.h>
#include <pthread.h>
main()  {
   pthread_t f2_thread, f1_thread;
   void *f2(), *f1();
   int i1,i2;
   i1 = 1;
   i2 = 2;
   pthread_create(&f1_thread,NULL,f1,&i1);
   pthread_create(&f2_thread,NULL,f2,&i2);
   pthread_join(f1_thread,NULL);
   pthread_join(f2_thread,NULL);
}
void *f1(int *x){
   int i;
   i = *x;
   sleep(1);
   printf("f1: %d",i);
   pthread_exit(0);
}
void *f2(int *x){
   int i;
   i = *x;
   sleep(1);
   printf("f2: %d",i);
   pthread_exit(0);
}
```

Esperar un hilo POSIX

```
#include <pthread.h>
int pthread_mutex_init (pthread_mutex_t *mut,
 const pthread_mutexattr_t *attr);
```

- Se le pasa un mutex a la función que se inicializa.
- Los atributos por defecto se ponen a NULL.

Esperar un hilo POSIX

```
#include <pthread.h>
int pthread_mutex_lock (pthread_mutex_t *mut);
```

- Bloquea el mutex.
- Si el mutex está bloqueda, queda en espera.

Esperar un hilo POSIX

```
#include <pthread.h>
int pthread_mutex_unlock (pthread_mutex_t *mut);
```

- Desbloquea el mutex.

Esperar un hilo POSIX

```
#include <pthread.h>
int pthread_mutex_trylock (pthread_mutex_t *mut);
```

- Bloquea el mutex si puede, si no devuelve EBUSY.

Esperar un hilo POSIX

```
#include <pthread.h>
int pthread_mutex_destroy (pthread_mutex_t *mut);
```

- Libera el mutex y limpia toda la memoria o recurso asociado a él.

Ejemplo mutex

THREAD 1	THREAD 2
`pthread_mutex_lock (&mut);`	
	`pthread_mutex_lock (&mut);`
`a = data;`	`/* blocked */`
`a++;`	`/* blocked */`
`data = a;`	`/* blocked */`
`pthread_mutex_unlock (&mut);`	`/* blocked */`
	`b = data;`
	`b--;`
	`data = b;`
	`pthread_mutex_unlock (&mut);`

- ¿Tienes dudas sobre el funcionamiento de una función?
- Usa `man` y el nombre de la función en tu terminal.