

THREAD POSIX

TPSIT

Libreria <pthread.h>

- ▶ Lo standard ANSI C non prevede l'utilizzo dei Thread.
- ▶ Lo standard POSIX definisce una nuova libreria contenente delle nuove funzioni per la gestione dei thread.

Che cos'è un thread?

- ▶ Un thread non è altro che una funzione C che viene eseguita in maniera concorrente ad altre funzioni nell'ambito di uno stesso processo.
- ▶ Tutti i thread che fanno parte di uno stesso processo ne condividono lo spazio di indirizzamento.
- ▶ Le variabili che vengono dichiarate localmente all'interno della funzione che implementa il codice di ciascun thread sono private per quello specifico thread, e pertanto non accessibili da parte degli altri thread del processo.
- ▶ La memoria condivisa sarà pertanto rappresentata dalle variabili globali del processo.
- ▶ Gli identificativi dei thread sono gestiti mediante il tipo `pthread_t`, che è la ridefinizione di un `Unsigned long int`

Abilitazione dei thread

- ▶ **`int pthread_setconcurrency (int nThread);`**
- ▶ Comunica al sistema il numero di thread che si intende creare.
- ▶ Restituisce 0 in caso di successo, qualsiasi cosa diversa da 0 in caso di insuccesso.
- ▶ `nThread` rappresenta il numero di thread da creare

Creazione di un nuovo thread

- ▶ **int pthread_create (pthread_t * tID, const pthread_attr *attr, void *(*foo)(void*), void *arg);**
- ▶ Restituisce 0 in caso di successo, qualsiasi cosa diversa da 0 in caso di insuccesso.
- ▶ **tID** è l'indirizzo dell'oggetto pthread_t destinato a contenere l'id che il sistema assegna al thread
- ▶ **attr** sono gli attributi che l'utente vuole assegnare al thread. (se vale NULL vengono usati quelli di default)
- ▶ **foo** è il nome della funzione C che verrà eseguita all'interno del nuovo thread.
- ▶ **arg** è il puntatore ai parametri da passare alla funzione foo. Deve assolutamente essere un void *

Attendere un thread

- ▶ **`int pthread_join (pthread_t tID, void ** risPt);`**
- ▶ Sospende il thread principale (main) in attesa che il thread identificato da tID giunga al termine
- ▶ risPt rappresenta l'indirizzo del puntatore al valore restituito dalla funzione foo eseguita nel thread
- ▶ Ritorna 0 in caso di successo, diverso da 0 altrimenti.
- ▶ Non è prevista invece, nello standard POSIX, una funzione di attesa di un thread generico.

Funzioni utili

- ▶ **int pthread_kill (pthread_t tid, int signo);**
 - ▶ Invia un segnale di terminazione al thread specificato dal 1° par Ritorna 0 in caso di successo, diverso da 0 altrimenti.
 - ▶ **signo** è l'identificatore del segnale che si vuole inviare al thread (normalmente SIGKILL definito in "signal.h").
- ▶ **int pthread_self ();**
 - ▶ restituisce il tID del thread corrente.

Istruzioni per la compilazione

- ▶ Per compilare un programma multithread occorre specificare il flag **-l pthread** sulla linea di comando
- ▶ Es:
- ▶ `gcc main.c -o main -lpthread`