

Thread detach

Se noi abbiamo un `pthread_t`, e quindi abbiamo un identificatore di un thread che eventualmente è stato lanciato, possiamo chiamare questa API per andare ad indicare il `detach`, ossia identifichiamo il thread di cui passiamo il codice numerico in input come thread “`detached`”, ossia che va ad implicare che quando quel thread completa la sua esecuzione, qualsiasi informazione che caratterizzava quell’esecuzione viene ad essere rilasciata: il che implica dire che se quello è un thread `detached` altri thread non sono nemmeno interessati ad aspettare il suo completamento e nemmeno a prelevare i codici di terminazione. Non ci saranno problemi di inter-relazione con altri thread che eventualmente vorranno interagire con lui.

“La funzione `pthread_detach()` viene utilizzata per indicare all'implementazione che l'archiviazione per il thread `thread` può essere recuperata quando il thread termina. Se il thread non è terminato, `pthread_detach()` non lo farà terminare. L'effetto di più chiamate `pthread_detach()` sullo stesso thread di destinazione non è specificato.”

-Manuale ufficiale - 4 aprile 1996

```
NAME
pthread_detach - detach a thread

SYNOPSIS
#include <pthread.h>

int
pthread_detach(pthread_t thread);
```

```
#include <pthread.h>

int pthread_detach(pthread_t thread);

Compile and link with -pthread.
```

DESCRIPTION

[top](#)

The `pthread_detach()` function marks the thread identified by `thread` as detached. When a detached thread terminates, its resources are automatically released back to the system without the need for another thread to join with the terminated thread.

Attempting to detach an already detached thread results in unspecified behavior.

RETURN VALUE

[top](#)

On success, `pthread_detach()` returns 0; on error, it returns an error number.