

Lisa Obermaier
Simon Thum

Detached/Joinable

Unterschiede von *pthread*-Typen

Detached threads geben dem Elternprozess keine ID zurück, weswegen man im aufrufenden Prozess nicht auf ihre Beendigung warten kann, und entsprechend keine Rückgabewerte über ein `pthread_exit` einsehen kann.

Vorteil von *detached threads* ist jedoch, nach Beendigung des Threads gehaltene Ressourcen sofort wieder freigegeben werden können.

Im Gegensatz dazu können Ressourcen von *joinable threads* erst nach Rückkehr ins Main-Thread (dem namensgebenden *join*) freigegeben werden. Dafür ist es möglich, die Beendigung der generierten Threads programmatisch abzuwarten und ihre Rückgabewerte zu übernehmen.

Faustregel: Soll auf den Thread gewartet werden, ist *joinable* zu wählen, ansonsten kann ohne Weiteres *detached* gewählt werden.

Ausarbeitung

Threads und Prozesse

Zur Realisierung der Aufgabe werden zwei Threads in einer for-Schleife mit Hilfe der Funktion "threadHello" erzeugt. In der Funktion "threadHello" wird der Thread schlafen gelegt und die Adresse über `pthread_exit` zurückgegeben.

Mit einer zweiten for-Schleife werden die zurückgekehrten Threads mit `pthread_join` wieder zusammengeführt. Anschließend werden die Adressen verglichen, die bei Erzeugung und Rückkehr ausgegeben werden, ob es dieselben sind.