

Threads in C auf Linux

Die C Standard-Bibliothek beinhaltet kauf Funktionen, um mit dem Betriebssystem komplexere Sachen anzustellen.

Hier ein einfaches Beispiel, um paralleles Ausführen von Programmteilen zu ermöglichen:

[threads_example.c](#)

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

volatile int x=0;

void* timer_1s(void *vargp){
    for(int i=0; i<10; i++){
        sleep(1); // wait 1s
        printf("1s-Tick\n");
        x++;
    }
    return NULL;
}

void* timer_2s(void *vargp){
    for(int i=0; i<5; i++){
        sleep(2);
        printf("2s-Tick, t=%ds\n", x);
    }
    return NULL;
}

int main(){
    pthread_t thread_1 = 0;
    pthread_t thread_2 = 0;

    printf("Wir starten nun beide Timer:\n");
    pthread_create(&thread_1, NULL, timer_1s, NULL);
    pthread_create(&thread_2, NULL, timer_2s, NULL);

    pthread_join(thread_1, NULL);
    pthread_join(thread_2, NULL);

    printf("Wir sind nun fertig mit dem Programm, tschüss!\n");
    return 0;
}
```

Kompilieren und ausführen:

```
gcc thread.c -Wall -lpthread  
./a.out
```

Ausgabe

```
Wir starten nun beide Timer:  
1s-Tick  
2s-Tick, t=1s  
1s-Tick  
1s-Tick  
2s-Tick, t=3s  
1s-Tick  
1s-Tick  
2s-Tick, t=5s  
1s-Tick  
1s-Tick  
2s-Tick, t=7s  
1s-Tick  
1s-Tick  
2s-Tick, t=9s  
1s-Tick  
Wir sind nun fertig mit dem Programm, tschüss!
```

Zu bemerken ist, dass der Heap-Speicher von beiden Threads gleichermaßen benutzt werden kann (siehe Variable x). Ebenso die Standardausgabe mittels `printf()`.

Würde in einer HTL so einfachen Code lehren, wäre C sicherlich gleich um vieles attraktiver. Doch dort behandelt man max. die Standard C-Lib, mit der man nur müßig etwas umsetzen kann, und schnell an einfachen Anforderungen scheitert.

[linux](#), [c](#), [programming](#), [deutsch](#)

From:

<http://www.zeilhofer.co.at/wiki/> - **Verschiedenste Artikel von Karl Zeilhofer**

Permanent link:

http://www.zeilhofer.co.at/wiki/doku.php?id=threads_linux&rev=1614449018

Last update: **2021/02/27 19:03**

