

# Betriebssysteme, SoSe 2020

## Praktische Übung 4: Threads

### Ziele des Labortermens

- POSIX Threads verwenden
- Probleme parallelisieren

### Aufgabe 4.1: Verständnisfragen

- a) Was ist der Unterschied zwischen Prozessen und Threads?
- b) Kann ein Thread zu mehreren Prozessen gehören?
- c) Nennen Sie jeweils Elemente die im Prozess gespeichert und die in einem seiner Threads gespeichert werden.
- d) Nennen Sie je einen Vorteil für Kernel Level- und User Level-Threads.
- e) Beschreiben Sie das Fork-Join-Modell.

### Aufgabe 4.2: POSIX Threads Grundlagen (praktisch)

- a) Schreiben Sie ein Programm, dass drei Threads startet. Die Threads sollen ihre Thread ID ausgeben. *Hinweis: `pthread_self()` gibt die Thread ID zurück.*
- b) Ändern Sie ihr Programm entsprechend, dass es auf die Fertigstellung der drei Threads wartet.

- c) Übergeben Sie nun jedem der drei Threads die Zahlen 1, 2 und 3. Geben Sie dieses Argument aus.
- d) Starten Sie in ihrem Programm nach dem join einen neuen Thread und entkoppeln Sie ihn (detach). Der neue Thread soll 20 Sekunden schlafen. Was beobachten Sie bei der Ausführung?

### Aufgabe 4.3: Array-Operation parallelisieren (praktisch)

Folgende Schleife soll parallelisiert werden:

```
int A[1000];  
for (int i = 0; i < 1000; i++) {  
    A[i] = i * 2;  
}
```

Verteilen Sie die Berechnung sinnvoll auf vier Threads.