

Übungsblatt 02

André Reissig

Benedikt Peterson

20. November 2025

Aufgabe 1

Wann ist es sinnvoll, nebenläufige Programmteile mit Hilfe von Threads anstatt Prozessen (heavyweight processes) zu implementieren?

Es ist wichtig zu wissen, ob Data geteilt werden muss mit anderen threads, dann ist es sinnvoll. Außerdem sind Threads simpler gestrickt als Prozesse und somit kosteneffizienter [1].

Welche Vorteile bieten User-Level-Threads gegenüber den Kernel-Level-Threads? Gibt es auch Nachteile?

	User-Level-Threads	Kernel-Level-Threads
Vorteil	Quick/ easy creation, highly portable (across many OS), no kernel privileges needed (context switching)	parallel execution in multithreading, other threads can run while one is blocked, access to system resources
Nachteil	Limited use of multiprocessing, blocking can halt entire process	no easy creation and managing (overhead), kernel mode for switching

[2]

Welche Vorteile und Nachteile gibt es, wenn man Thread-Kontrollblöcke (TCB) als Skalare, in Arrays, Listen, Bäumen oder invertierten Tabellen speichert?

Datentyp	Vorteil	Nachteil
Skalar	individuell erstellbar, einfaches Löschen	keine Ordnung/ Reihenfolge
Array	einfaches Löschen, Gruppierung möglich	keine Unterteilung
Listen	klare Reihenfolge	kein einfaches löschen (verschachtelt)
Bäume	Möglichkeit für Prios	kein einfaches löschen (Baumstruktur kann zerstört werden)
Invertierte Tabellen	einfache Gruppierung nach Attributen	Reihenfolge schwierig untereinander und zwischen Tabellen

In welchem Adressraum (Prozess-Eigner, Dienste-Prozess, BS-Kern) wird ein TCB gespeichert?

Im Kernel Memory [4]

Literatur

- [1] <https://medium.com/reverse-engineering/process-vs-thread-which-one-can-handle-tasks-better-264fc1188fb6>
- [2] <https://www.geeksforgeeks.org/difference-between-user-level-thread-and-kernel-level-thread> (as of Nov 09 2024)
- [3] B. Linnert, Vorlesungsfolien : Betriebssysteme WS 24/25
- [4] https://en.wikipedia.org/wiki/Thread_control_block (as of Nov 09 2024)

Aufgabe 2

Die Programme sind zu finden unter dem folgenden link

https://github.com/benedikt-code/bs_ws_25-26

Die Dateien sind zu finden in dem Ordner **Uebung2**. Die Programme sind im Unterordner **Core/src**.

Als verwendete Hilfsmittel ist ChatGPT zu nennen. ChatGPT wurde benutzt, um eine erste, grobe Idee zu bekommen, wie eine Implementierung aussehen könnte.