

2-1 (Threads)

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

Wenn die Programmteile auf die selben Objekte zugreifen und daher ein geteilter Address-Bereich vorteilhaft ist.

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

Skalare: Vorteil: Optimierte Nutzung von freiem Speicher
Nachteil: Zusätzliche look-up table notwendig?

Arrays:

- Vorteil: Kurze Suchzeiten in der Struktur
- Nachteil: Beschränkte Anzahl an TCPs durch feste Array Länge

Listen:

- Vorteil: Dynamische Länge
- Nachteil: Es muss durch die gesamte Liste iteriert werden um auf einen TCB zuzugreifen

Bäume:

- Vorteil: Kurze Suchzeiten in der Struktur
- Nachteil: Kompliziertes Speicherlayout

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

Da der Kernel des Betriebssystems die Threads managed wird ein TCP im Betriebssystem-Kern gespeichert.

2-2 (Behandlung von Ausnahmen)

Git: <https://github.com/elenaf9/WS2022-betriebssysteme>
Branch: assignment-2