

INFO SHEET

Processes / Threads

Prozesse:

- ✦ Ein Prozess ist eine Instanz eines laufenden Programms.
- ✦ Prozesse sind unabhängige Einheiten.
- ✦ Jeder Prozess hat seinen eigenen Adressraum.
- ✦ Prozesse sind voneinander isoliert.
- ✦ Jeder Prozess hat mindestens einen Thread, der als Hauptthread bezeichnet wird.

Threads:

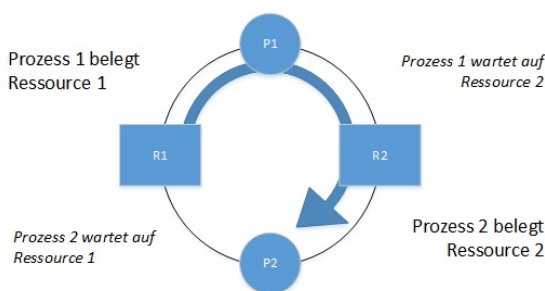
- ✦ Ein Thread ist ein ausführbarer Teil eines Prozesses.
- ✦ Threads innerhalb desselben Prozesses teilen sich den gleichen Adressraum.
- ✦ Threads teilen sich Ressourcen.
- ✦ Threads können Aufgaben unabhängig voneinander ausführen.

Mutual Exclusion

Mutual Exclusion (gegenseitiger Ausschluss) ist ein grundlegendes Konzept im Bereich der Synchronisation von Prozessen oder Threads. Es bezieht sich darauf, dass nur ein Prozess oder Thread zu einem bestimmten Zeitpunkt auf eine gemeinsam genutzte Ressource zugreifen kann.

- Semaphore
- Mutex-Lock
- Monitore
- Condition Variables

Dead-Lock

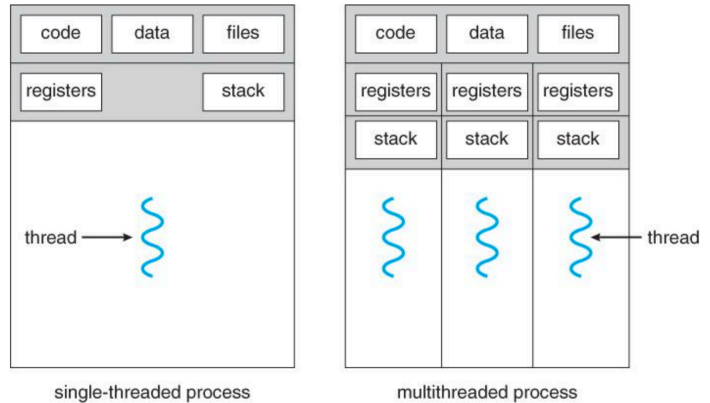


Unter einem **Deadlock** versteht man eine Prozesssituation, bei der zwei oder mehr Prozesse auf das Freiwerden von Betriebsmitteln warten.

Die Ressource 1 wird vom Prozess 1 (exklusiv) belegt, Prozess 1 wartet auf die Freigabe von Ressource 2.

Die Ressource 2 wird jedoch von Prozess 2 belegt, der wiederum auf die Freigabe der Ressource 1 wartet und die Ressource 2 exklusiv bindet. Es entsteht eine Situation der Verklemmung.

Single-Thread Multi-Thread



Single Thread:

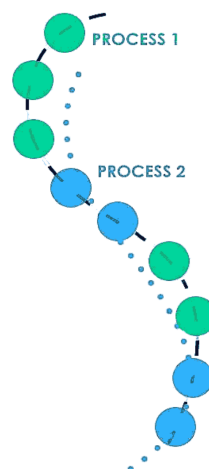
- Ein Single-Thread-Programm besteht aus einem einzigen Ausführungsstrang.
- Es kann immer nur eine Aufgabe gleichzeitig erledigen.
- Keine Synchronisierung oder Koordination zwischen Threads erforderlich ist.

Multi Thread:

- Ein Multi-Thread-Programm besteht aus mehreren Threads, die parallel oder nebenläufig arbeiten können.
- Jeder Thread kann eine separate Aufgabe oder Funktion ausführen.
- Multi-Thread-Programme nutzen x Prozessorkerne, um Aufgaben gleichzeitig auszuführen.

Nebenläufigkeit

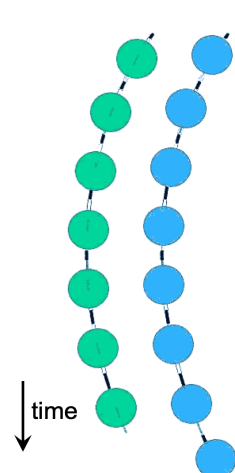
CONCURRENCY



Nebenläufigkeit bezieht sich darauf mehrere Aufgaben scheinbar gleichzeitig auszuführen. Hierbei können mehrere Aufgaben nacheinander auf einem einzelnen Prozessorkern ausgeführt werden, indem die Ausführung einer Aufgabe unterbrochen wird.

Parallelität

PARALLELISM



Parallelität bezieht sich auf die gleichzeitige Ausführung x Aufgaben, wobei jede Aufgabe auf einem separaten Prozessor oder Kern innerhalb eines Prozessors ausgeführt wird. Dies ermöglicht eine tatsächliche gleichzeitige Verarbeitung mehrerer Anweisungen.