

Betriebssysteme

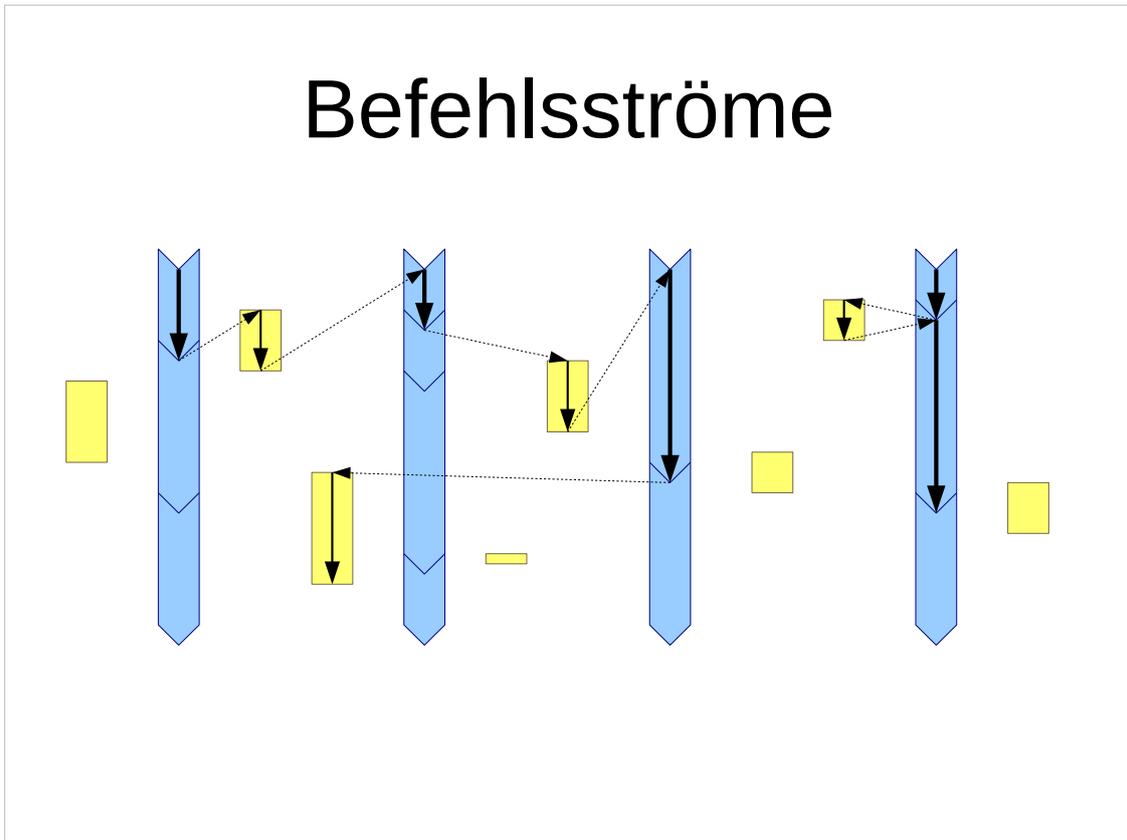
Umschalten in Kernaktionen

DHBW Stuttgart

Roland Weber

Kapitel "Umschalten" im Begleitbuch

Befehlsströme



Langlebige Befehlsströme (blaue Streifen) werden stückweise abgearbeitet, immer wieder unterbrochen, zurückgestellt oder blockiert.

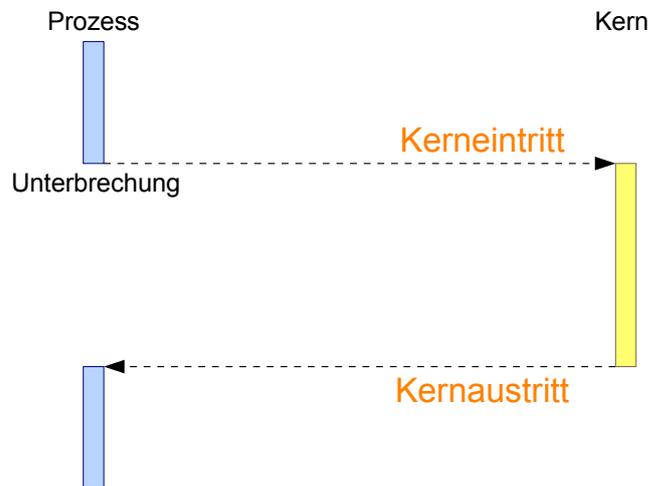
Dafür sind Prozesse und der Aufgreifer zuständig. So bekommen Instanzen Rechenzeit.

Kurze Befehlsströme (gelbe Streifen) starten plötzlich und laufen schnell zu Ende. Sie können und dürfen weder blockieren noch zurückgestellt werden.

Das sind Unterbrechungsbehandlungen. So kann sich der Kern Rechenzeit nehmen.

Vereinfachung: Im Folgenden tun wir so, als arbeite der Kern ausschließlich in kurzen Befehlsströmen.

Kernaktion ohne Umschalten



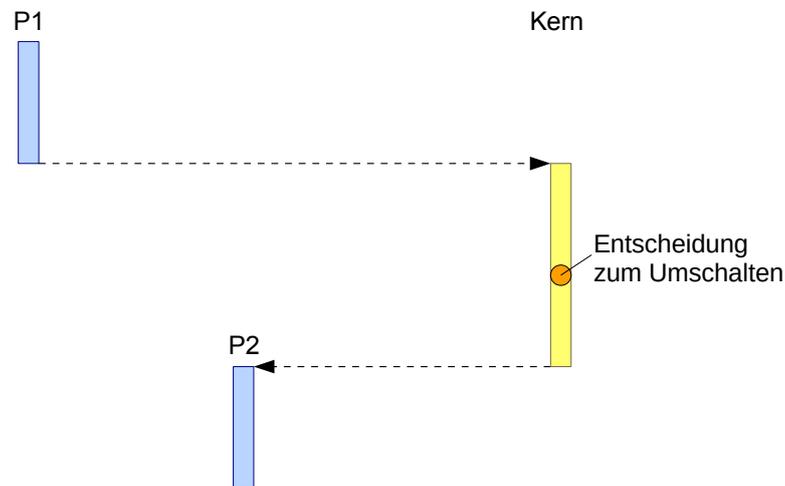
Die Zeitachse dieser und der folgenden Darstellungen verläuft von oben nach unten.

Ein Prozessor (CPU) arbeitet zunächst an einem Prozess, einem langlebigen Befehlsstrom. Denn gibt es eine Unterbrechung, und damit einen Kerneintritt. Der langlebige Befehlsstrom ruht, während der Prozessor eine Kernaktion als kurzen Befehlsstrom ausführt.

Mit dem Ende des kurzen Befehlsstroms springt der Prozessor zurück zum Prozess. Das ist der Kernaustritt.

Der Prozess bleibt die ganze Zeit über im Zustand Rechnend. Der kurze Befehlsstrom läuft so schnell zu Ende, dass ein Zustandswechsel keinen Sinn ergibt.

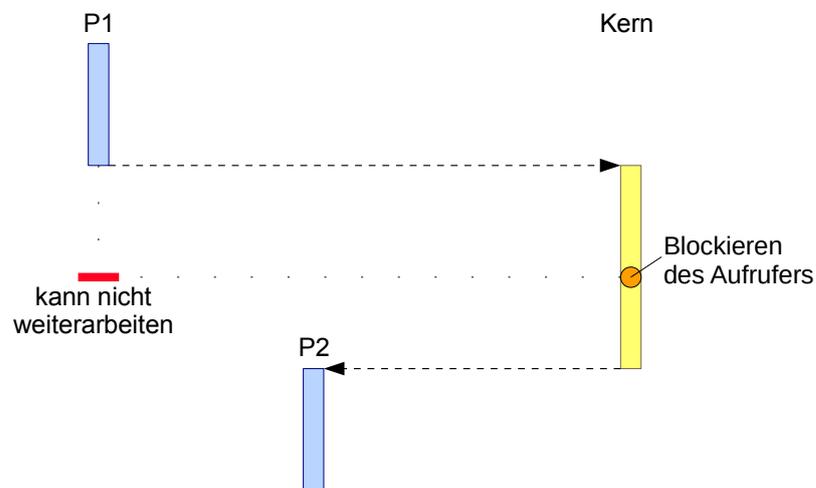
Kernaktion mit Umschalten



Entscheidungen über Zustandswechsel von Prozessen fallen mitten in einer Kernaktion. Die Kernaktion läuft noch zu Ende, erst danach springt der Prozessor zu einem anderen Prozess.

Beispiel: Der Ablauf einer Zeitscheibe führt zu einer Unterbrechung. In der Kernaktion entscheidet der Aufgreifer, welcher andere Prozess die nächste Zeitscheibe erhält. P1 wird in die Bereitmenge zurückgestellt, P2 dort aufgegriffen.

Kernaktion mit Blockieren

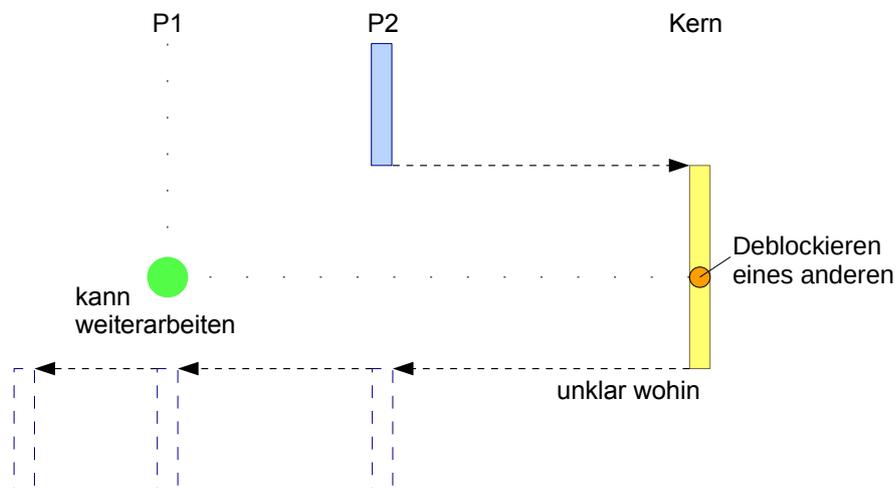


Eine Kernaktion kann den laufenden, d.h. gerade unterbrochenen Prozess blockieren. Auch dann wählt der Aufgreifer einen anderen Prozess aus, der als nächstes Rechenzeit bekommt.

Der Unterschied zur vorhergehenden Abbildung ist, dass P1 hier nicht in die Bereitmenge zurückgestellt, sondern blockiert und in eine Wartemenge eingetragen wird.

Beispiel: P1 ruft den Kern, um an einem Kanal eine Nachricht synchron zu empfangen. Aber es liegt gerade keine Nachricht im Kanal.

Kernaktion mit Deblockieren



Eine Kernaktion kann einen blockierten Prozess deblockieren, oder auch mehrere. Dann steht der Aufgreifer vor der Wahl, welcher Prozess auf dem Prozessor als nächstes Rechenzeit erhält.

Die wahrscheinlichsten Kandidaten sind hier P2, der vor dem Kerneintritt rechnete, und P1, der in der Kernaktion deblockiert wird. Aber der Aufgreifer könnte auch einen dritten Prozess auswählen.

Auf den Verlauf der Kernaktion hat die Entscheidung keine Auswirkungen. Erst wenn die Kernaktion endet zeigt sich, woran der Prozessor als nächstes arbeitet. Das passiert nicht nur beim Deblockieren, prinzipiell könnte jede Kernaktion umschalten. In den Abbildungen zuvor habe ich gezielt einfache Fälle herausgegriffen.