

Proseminar: Konzepte von Betriebssystemkomponenten

Bruno Kleinert
3.11.03

Prozesse:

Prozesskontrollblock,
-steuerung,
-umschaltung

Überblick

- Prozesskontrollblock
- Prozesszustände
- Prozessumschaltung

Was ist ein Prozess?

- Programm während der Ausführung
- Ein Programm kann während der Ausführung auch aus mehreren Prozessen bestehen

Prozesskontrollblock

Zentrales Element:

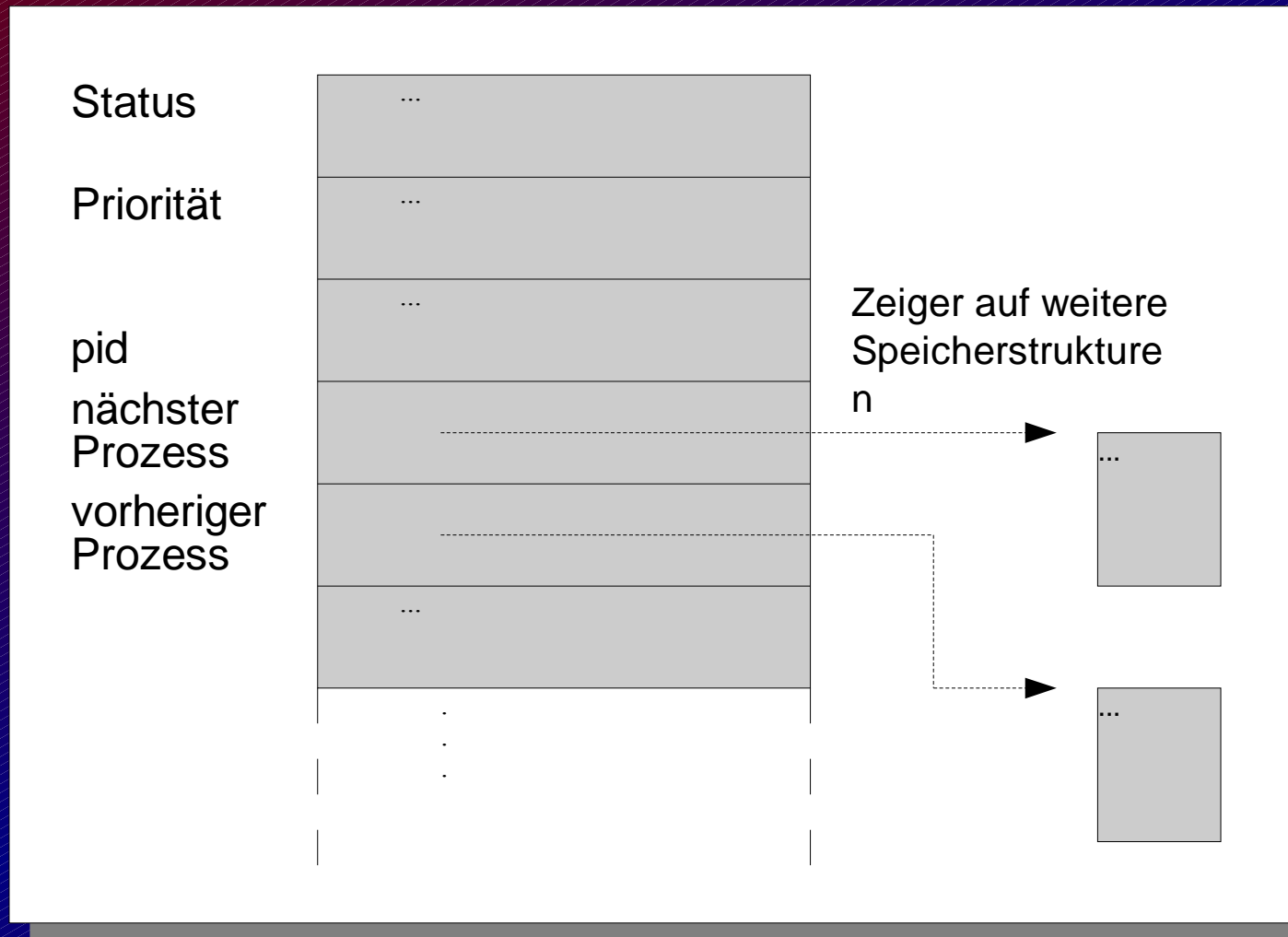
Der Process Descriptor

- Speicherstruktur
- enthält Informationen über genau einen Prozess
=> Je Prozess ein Process Descriptor

Im Process Descriptor wird z.B. gespeichert:

- Zugeordneter Speicherbereich
- Priorität
- Dateien auf die der Prozess Zugriff hat
- Zeiger auf weitere Speicherstrukturen

Der Process Descriptor



Zustände von Prozessen

- **TASK_RUNNING**

- wird ausgeführt

- **TASK_INTERRUPTIBLE**

- angehalten
- wird durch Signal oder frei gewordenen Hardware Gerät wieder aufgeweckt

- **TASK_UNINTERRUPTIBLE**

- angehalten
- kann NUR von frei gewordenem Hardware Gerät wieder aufgeweckt werden
- Anwendung z.B. bei Gerätetreibern

Zustände von Prozessen

- **TASK_STOPPED**

- gestoppt
- Ausführung kann wieder aufgenommen werden
- Anwendung z.B. bei Debuggern

- **TASK_ZOMBIE**

- beendet
- ein anderer Prozess benötigt eventuell Daten über toten Prozess
=> Kernel kann Ressourcen noch nicht freigeben

In welchem Zustand sich ein Prozess befindet wird im Process Descriptor im Status Feld gespeichert!

Die Identifikation von Prozessen im System

verschiedene Möglichkeiten:

- **Über den Process Descriptor**

- Zeiger auf Speicheradresse zu Beginn des Process Descriptors
- hauptsächlich Kernel nutzt diesen Weg

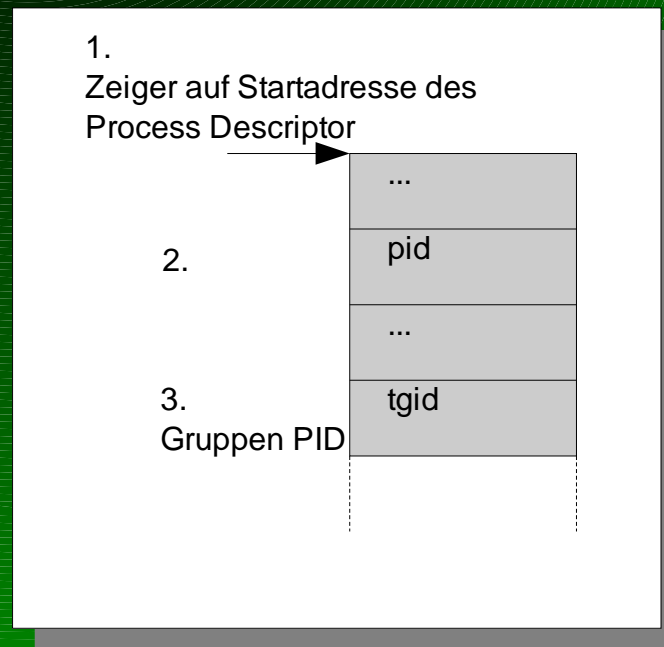
- **Über Process IDs**

- Fortlaufende Nummerierung der Prozesse
- für Benutzer einfache Variante

Zustände von Prozessen

• Gruppen

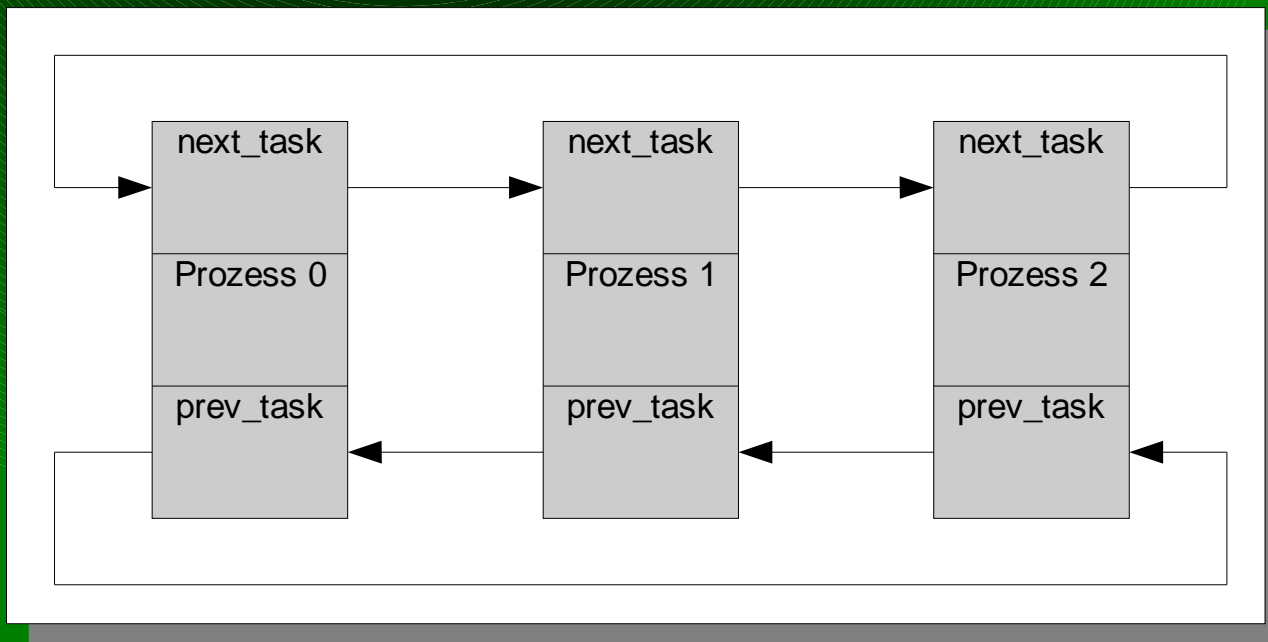
- Prozesse einer Gruppen werden zu Listen zusammengefasst
- können parallel über gleiche Gruppen PID erreicht werden – die PID stammt vom ersten Prozess der Gruppe
- aber: haben dennoch alle unterschiedliche eigene PIDs!



Zustände von Prozessen

Verwaltung der Prozessstrukturen

- Listen für verschiedene Zustände
- Prozesse werden in entsprechende Listen eingegliedert
- Verkettung über Zeiger auf Process Descriptoren
- gespeichert in den Feldern `next_task` und `prev_task`



Zustände von Prozessen

Prozesslisten als Warteschlangen

Durch Systemaufruf, kann ein Prozess in eine Warteschlange gelangen.

- gleiche Verkettung zu einer Prozessliste

Prozesse, die auf die gleiche Ressource warten

- exklusiv → nur ein Prozess wird aufgeweckt
- nicht exklusiv → alle Prozesse werden aufgeweckt

Hintergrund der Umschaltung:

- in Mehrprozess-Betriebssystem sind zwar mehrere Prozesse gleichzeitig möglich, aber...
- auf CPU kann nur ein Prozess bearbeitet werden

=> Prozesse müssen abwechselnd CPU-Zeit bekommen

Prozessumschaltung

Problemstellung bei der Umschaltung:

Prozesse müssen sich alle CPU Register teilen!

Das bedeutet:

- Jeder Prozess besitzt anderes Zustandsmuster der Register
- Zustände der Register vor einem Prozesswechsel speichern
- Zustände in die CPU Laden, bevor ein anderer Prozess seine Arbeit wieder aufnimmt

Prozesswechsel unter Linux

Bei Interarchitektur: in Hardware bereits Implementierung zur Vereinfachung der Umschaltung, aber...

Linux verlässt sich nicht auf die Hardware und übernimmt die Umschaltung nur durch Software!

Das bietet zwei wesentliche Vorteile...

Prozessumschaltung

1. Software Umschaltung **flexibel** und **modifizierbar**
bessere Anpassungsmöglichkeit
Spielraum um Optimierungen der Umschaltung vorzunehmen
2. Software Umschaltung setzt sich aus einer Reihe
Assemblerbefehlen zusammen
=> der Vorgang der Umschaltung wird **transparent**
bessere **Kontrolle** der zu ladenden Registerdaten auf
Fehler

Der Ablauf der Umschaltung

Wann eine Umschaltung stattfindet bestimmt der Scheduler.

Innerhalb der Scheduler-Funktion müssen zwei Variablen zur Verfügung stehen:

- prev -> zu ersetzender Prozess
- next -> zu aktivierender Prozess

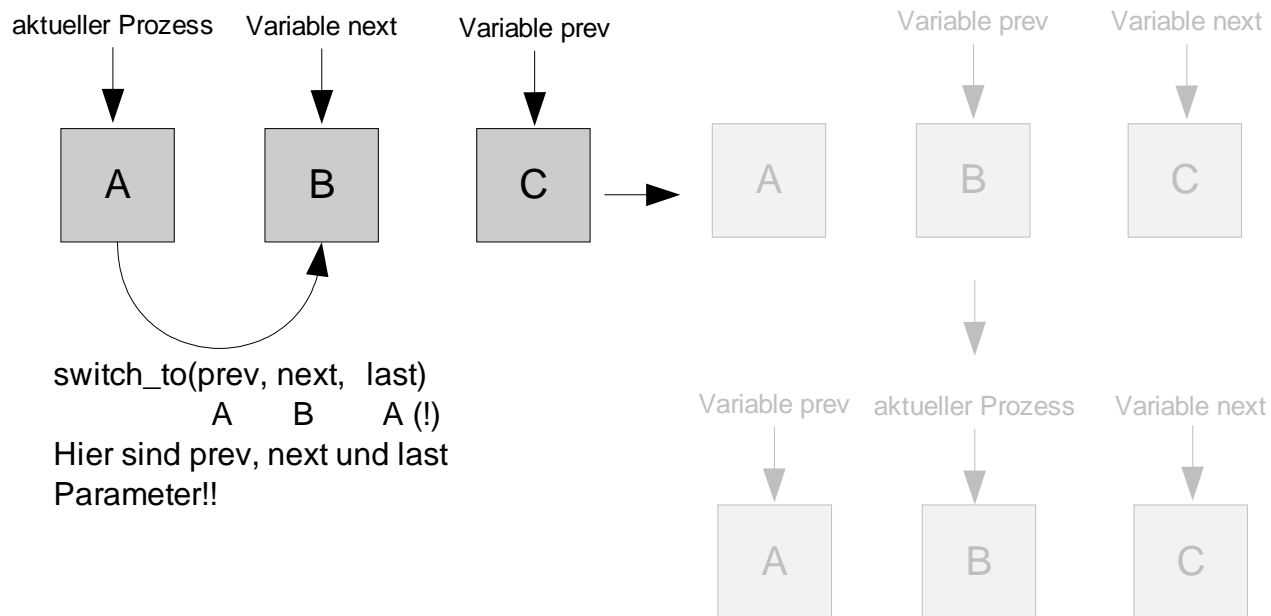
Prozessumschaltung

Zur Ausführung der Umschaltung selbst, muss ein Makro aufgerufen werden, das als Parameter drei Prozesse benötigt.

- prev -> zu ersetzender Prozess
- next -> zu aktivierender Prozess
- last -> zu ersetzender Prozess

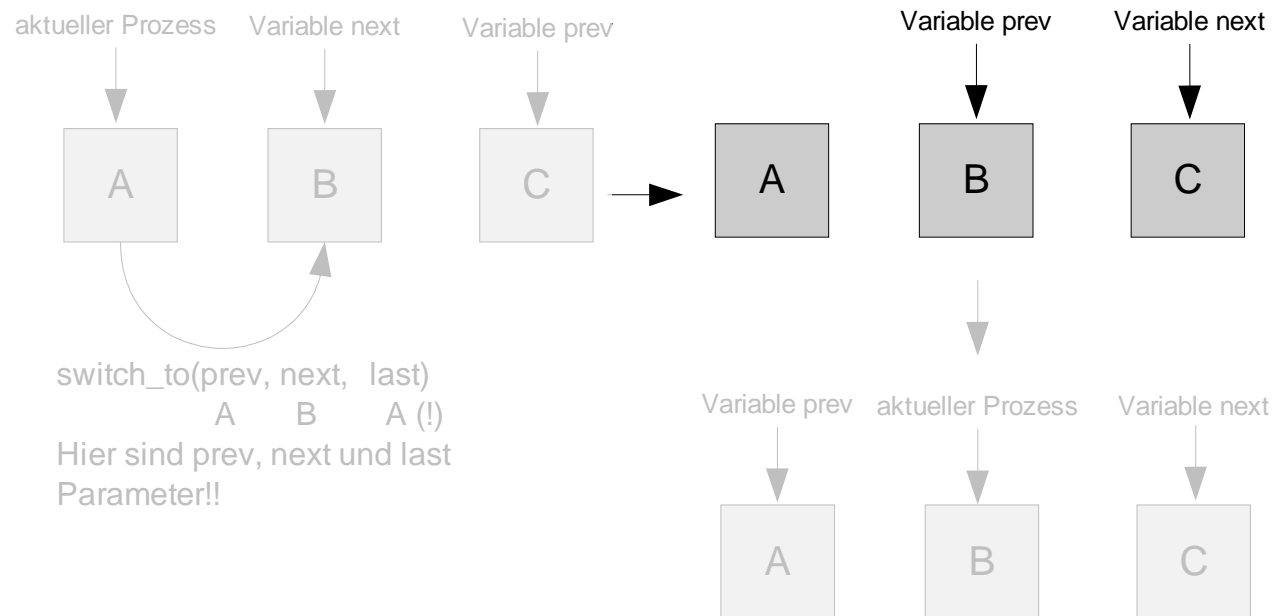
Prozessumschaltung

Von Prozess A soll zu B umgeschaltet werden.



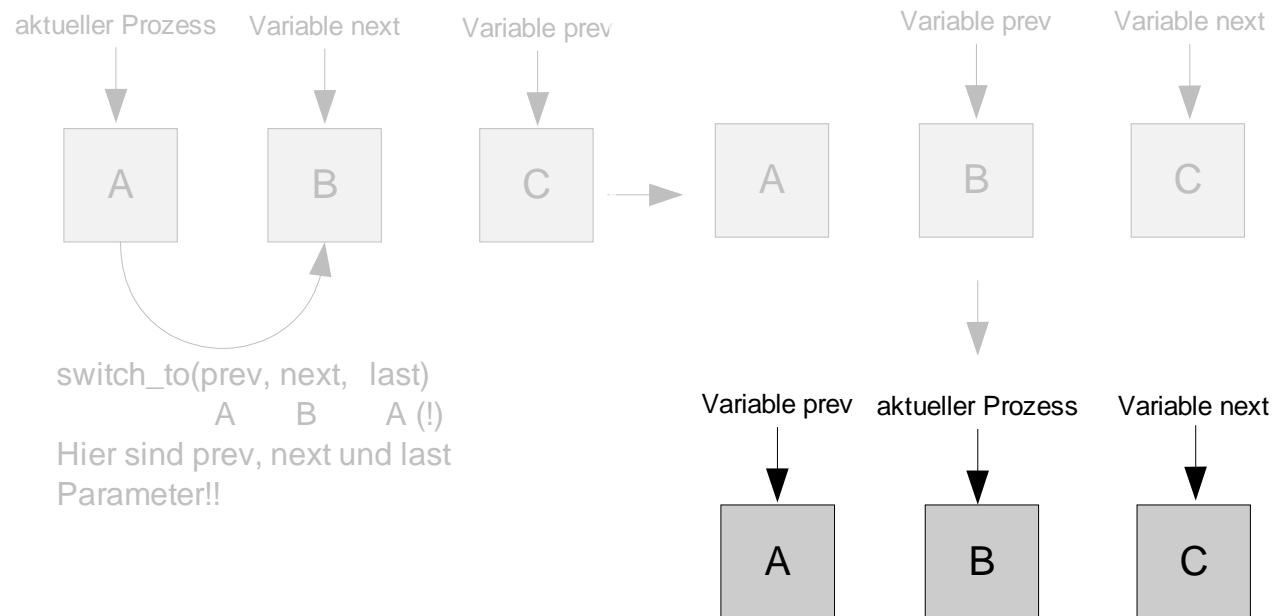
Prozessumschaltung

Von Prozess A soll zu B umgeschaltet werden.



Prozessumschaltung

Von Prozess A soll zu B umgeschaltet werden.



Zusammenfassung

- Prozesskontrollblock
 - Process Descriptor speichert Prozessinformationen
- Prozesszustände
 - gespeichert im Process Descriptor
 - PID ist nicht der einzige Weg Prozesse zu identifizieren um Zustände zuzuweisen
- Prozessumschaltung
 - Linux nutzt nicht Implementierung der Hardware