



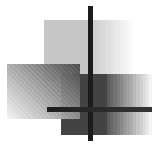
Betriebssysteme

Threads



Überblick

- Prozesse versus Threads
- Beispiele für Multithreading
- Vorteile von Multithreading
- Anwender- und Kernel-Threads
- Multithreading-Modelle
- Implementierungsaspekte
- Solaris-2-Multithreading



Prozesse und Threads (I)

▪ Prozess

- Ein in
- Benötigt Ressourcen:
- Bislang betrachtet: arbeitende Prozesse
()

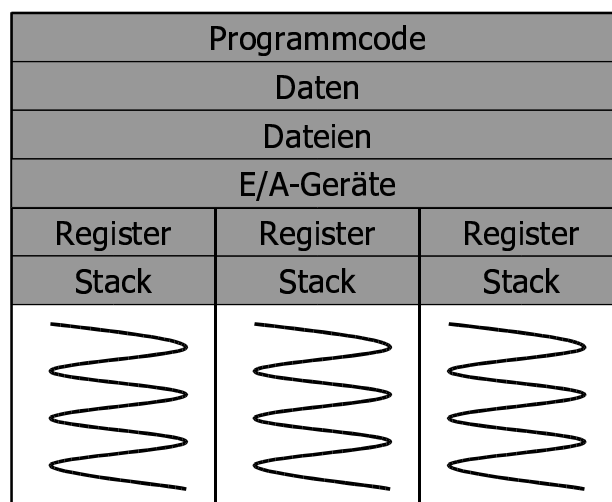
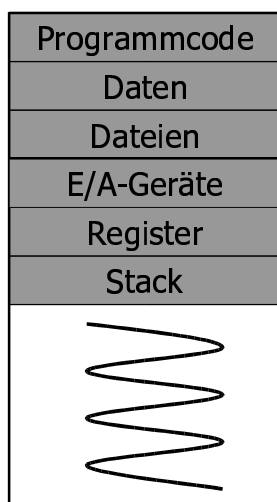
▪ Thread

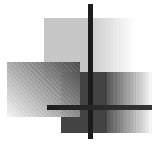
- Ein innerhalb eines Prozesses
- Benötigt:
- Nutzt:

- Threads innerhalb eines Prozesses möglich



Prozesse und Threads (II)





Beispiele für Multithreading

- Anwendungen mit , z.B. Textverarbeitung:
 - Texteingabe
 - Rechtschreibprüfung
 - Ausdruck
- , z.B. Webserver, DB-Server:
 - Administration
 - Simultane Bearbeitung vieler Anfragen



Vorteile von Multithreading

- Bei interaktiven Anwendungen kann auch auf Benutzereingaben reagiert werden, während andere, langandauernde Aufgaben durchgeführt werden.
- Auf gemeinsamen Speicher sowie gemeinsame Dateien und E/A-Geräte kann ohne weiteren Aufwand zugegriffen werden.
- Die Erzeugung eines neuen Threads und der Wechsel zwischen zwei Threads eines Prozesses verursacht erheblich weniger Aufwand (im Vergleich zur Prozesserzeugung/zum Prozesswechsel).
- Auch ein einziger multithreading Prozess kann gleichzeitig mehrere Prozessoren nutzen.



Anwender- und Kernel-Threads

- **Anwender-Threads**

Erzeugung, Scheduling und Verwaltung der Threads erfolgt über _____ auf Ebene des Anwendungsprogramms. Für den Kernel besteht das Programm aus einem _____

Vorteil: _____ (Kernel muss nicht eingreifen)

Nachteil: _____

- **Kernel-Threads**

Erzeugung, Scheduling und Verwaltung der Threads werden _____

Vorteile: _____ möglich;
ein wartender Thread _____ die anderen Threads nicht.

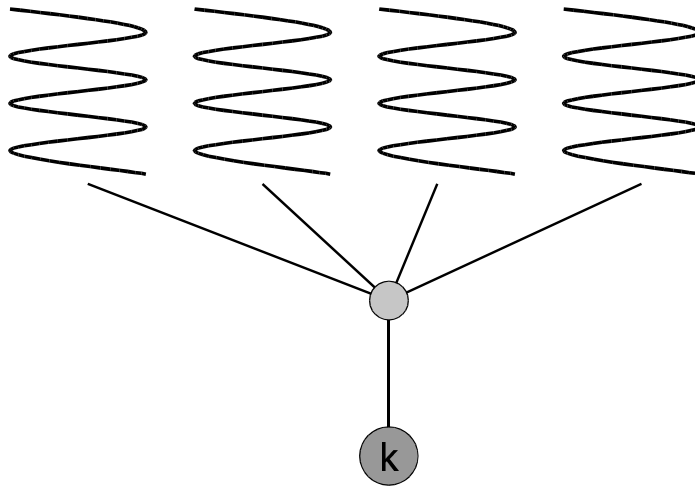
Nachteil: _____ als Anwender-Threads.



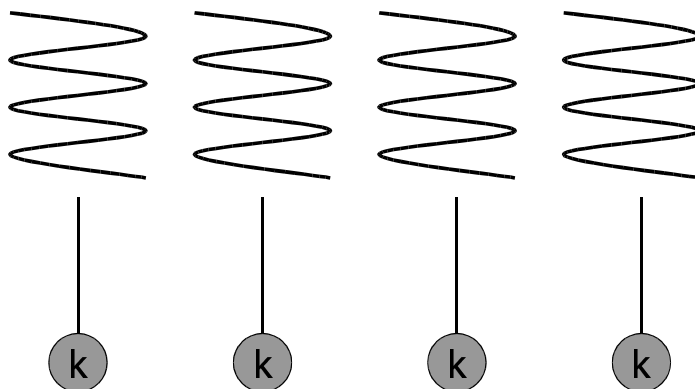
Multithreading-Modelle

- Mehrere Anwender-Threads werden auf einen Kernel-Thread abgebildet.
- Beispiele: Green-Thread-Library bei Solaris 2, POSIX Pthread-Library, Betriebssysteme ohne Thread-Unterstützung
- Jeder Thread eines Anwendungsprogramms wird auf genau einen Kernel-Thread abgebildet
- Beispiele:
- Die Threads der Anwendungsprogramme werden auf eine Anzahl von Kernel-Threads gemultiplext.
- Beispiele:

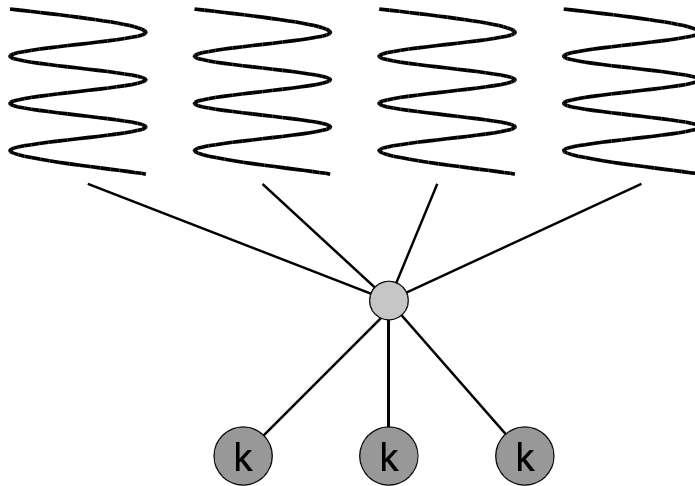
Multithreading-Modelle: Many-to-One



Multithreading-Modelle: One-to-One



Multithreading-Modelle: Many-to-Many



Implementierungsaspekte (I)

- :
 - Alle Threads des Prozesses werden dupliziert
 - Nur der Thread, der fork aufgerufen hat, wird dupliziert
- :
 - :
 - Sofortige Beendigung des Threads
 - : Der Thread überprüft regelmäßig, ob er sich beenden soll. Ist dies der Fall, führt er noch alle notwendigen Abschluss-arbeiten aus und beendet sich dann.

Implementierungsaspekte (II)

- Das Signal wird übermittelt (z.B. „division by zero“)
 - Das Signal wird übermittelt (z.B. „<control><c>“)
 - Das Signal wird übermittelt (z.B. „window resize“)
 - Das Signal wird übermittelt (z.B. „window resize“)
-
- Erzeugen einer von Threads
 - nicht mehr benötigter Threads als neu erzeugen.)
(Recycling ist)

Solaris-2-Threads

