

Systemadministration

Teil 6a

Prof. Dr.-Ing. Jörn Schneider

WIEDERHOLUNG

UNIX u. Windows: Prozesserzeugung

- Prozesserzeugung =
 1. register (Prozess im System bekannt machen bzw. anlegen)
 2. activate (Prozess bereit machen zur Ausführung → Prozess nimmt am Scheduling teil)
- System Calls zur Prozesserzeugung
 - UNIX: fork
 - Windows: CreateProcess

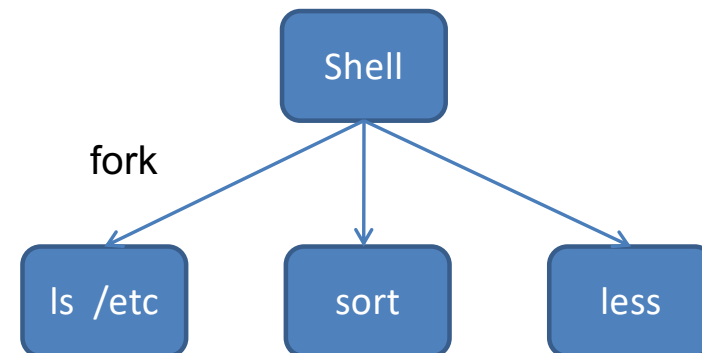
UNIX: fork & exec

fork

- Erzeugt Kindprozess (Child) mit gleichem Umfeld:
 - Programmcode
 - Speicherimage (Kopie)
 - Umgebungsvariablen (Kopie)
 - Offene Dateien

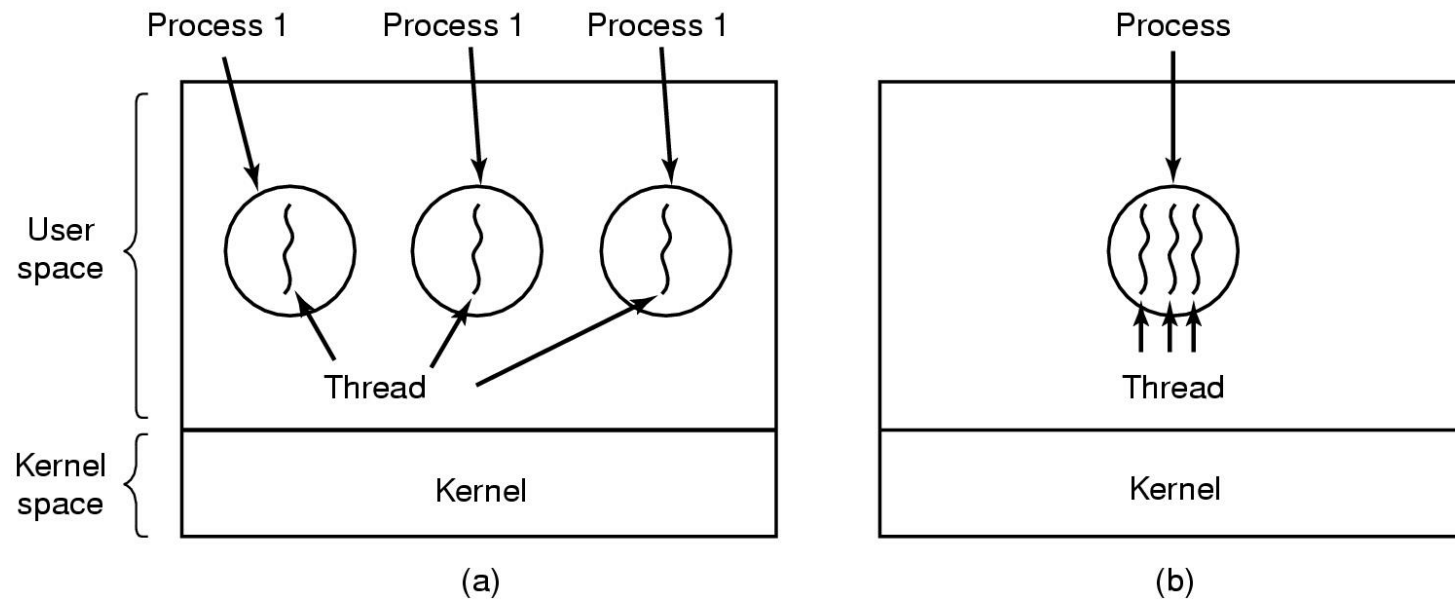
exec

- Führt neues Programm anstelle des bisherigen aus



Threads

The Thread Model (1)



(a) Three processes each with one thread

(b) One process with three threads

The Thread Model (2)

Per process items	Per thread items
Address space Global variables Open files Child processes Pending alarms Signals and signal handlers Accounting information	Program counter Registers Stack State

- Items shared by all threads in a process
- Items private to each thread

WIEDERHOLUNG - ENDE

Teil 6

- Was ist ein Rechnersystem?
- Was ist ein Betriebssystem?
- Aufgaben eines Systemadministrators
- Rechneraufbau
- Betriebssystemkonzepte
- Benutzer
- **Prozesse und Threads (2)**
- Bootvorgang

Beispiel UNIX

- Anzeigen von Prozessen mit dem Kommando **ps**
 - Anzeigen der Prozesse des Benutzers notroot:
 - `ps -u notroot`

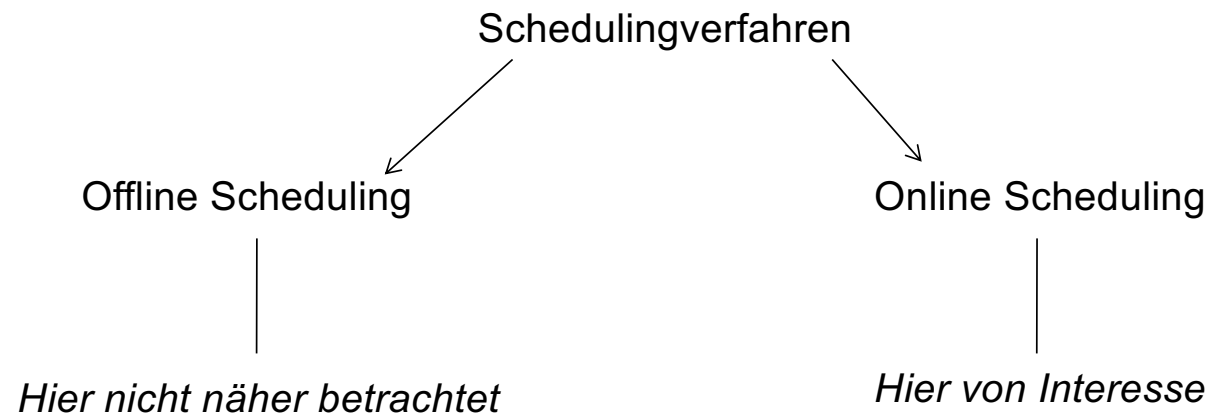
SCHEDULING

Klassifizierung von Schedulingverfahren (1)

- Offline Scheduling
 - Ablauf der einzelnen Programme zur Entwicklungszeit festgelegt
 - z.B.:
 - Cyclic Executive
 - Fabrikfließband
- Online Scheduling
 - Genauer Ablauf wird zur Laufzeit entschieden
 - Regeln und Einschränkungen dafür werden zur Entwicklungszeit festgelegt
 - z.B.:
 - Warteschlange bei Kfz-Anmeldung (Nummern ziehen)
 - Wartezimmer beim Arzt
 - Linux, Windows, OSEK OS, AUTOSAR OS

Hinweis

- Im Folgenden werden soweit nichts anderes erwähnt wird, lediglich Online Scheduling Verfahren betrachtet.



Klassifizierung Online Schedulingverfahren (1)

- **Prioritätsgesteuert**
 - Schedulingobjekten werden Prioritäten zugeordnet
 - z.B.:
 - Ticket System des r/ft
 - Persönlicher Zeitplaner
- **Nicht prioritätsgesteuert**
 - Es werden keine expliziten Prioritäten vergeben
 - z.B.:
 - Warteschlange beim Bäcker
 - Druckerwarteschlange

Klassifizierung Online Schedulingverfahren (2)

- Unterbrechbar/Preemptable
 - Bereits gestartete Vorgänge/Programme können unterbrochen werden
- Nicht Unterbrechbar/Non preemptable
 - Einmal gestartete Vorgänge/Programme können nicht unterbrochen werden

Klassifizierung Prioritätsgesteuerte Schedulingverfahren

- Dynamische Prioritäten/Dynamic Priority Scheduling
 - Prioritäten werden zur Laufzeit vergeben
- Feste Prioritäten/Fixed Priority Scheduling
 - Prioritäten werden zur Entwicklungszeit vergeben

First Come First Serve (FCFS) Scheduling

- Scheduling nach dem FIFO-Prinzip

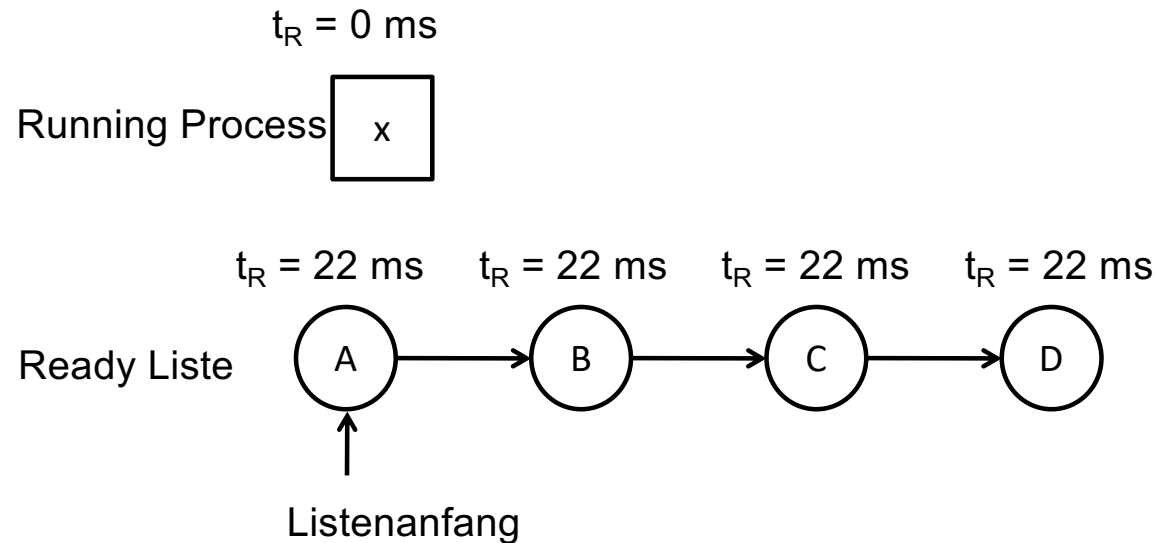
Round-Robin Scheduling

- Jeder Task der zur Ausführung gelangt erhält ein Quantum an zugestanderener Rechenzeit
- Hat er das Quantum aufgebraucht, wird er unterbrochen und ans Ende der Ready-Liste gehängt, dann wird der nächste Task in der Liste ausgeführt
- Beendet sich der Task vor Aufbrauchen des Quantums wird der Scheduler aufgerufen, der den nächsten Task auswählt

Fingerübung: Round-Robin

Ausgangssituation:

Quantum: 20 ms



Aufgabe:

Zeichne jeweils den Zustand zu den Zeitpunkten a) **$t=1 \text{ ms}$** und b) **$t=105 \text{ ms}$** , wenn bei **$t=0 \text{ ms}$** **Prozess x terminiert** und bei **$t=30 \text{ ms}$** Prozess **E ($t_R=22 \text{ ms}$)** in die **Ready-Liste** aufgenommen wird

$t_R = \underline{\hspace{1cm}} \text{ ms}$
Running Process 