

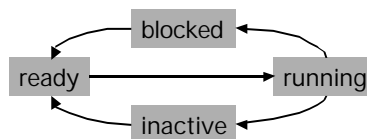


## 5 Betriebssystemunterstützung für Multimedia

### Prozessverwaltung



- Scheduler und Dispatcher
  - Logische Ressourcen (z.B. Prozesse) werden auf physikalische Ressourcen (z.B. CPU) abgebildet
- Ein Prozess kann folgende Zustände einnehmen
  - Running  
ein realer Prozessor ist dem Prozess zugeordnet
  - Ready  
der Prozess besitzt alle Betriebsmittel bis auf den Prozessor
  - Blocked  
der Prozess wartet auf das Eintreten eines Ereignisses
  - Inactive  
dem Prozess ist kein Programm zur Ausführung zugeordnet



## Scheduler und Dispatcher



- Scheduler
  - entscheidet, welche der wartenden Prozesse vom Zustand *Inactive* in den Zustand *Ready* übernommen werden
- Dispatcher
  - verwaltet den Übergang von *Ready* nach *Running*
- Scheduler und Dispatcher in herkömmlichen Betriebssystemen sind i.d.R. nicht echtzeitfähig



Folie 361

## Anforderungen für Multimedia



- Verarbeitung von kontinuierlichen Datenströmen
- Daten treten periodisch auf
- Operationen auf den Daten wiederholen sich
  - Erfassung/Ausgabe von Audio- und Videopaketen
  - Weiterleitung von Audio- und Videopaketen
  - Kompression/Dekompression von AV-Paketen
- Echtzeit-Anforderung
  - Verarbeitung muss bis zu einer Deadline abgeschlossen sein
  - Verarbeitung benötigt pro Periode in etwa dieselben Ressourcen



Folie 362

## Verfahren zur Reservierung von Betriebsmitteln



- Pessimistisch/deterministisch
  - berücksichtigt den ungünstigsten Fall
  - führt in der Regel zur Überreservierung
  - schlechte Ausnutzung der Ressourcen
- Optimistisch/stochastisch
  - Belegung richtet sich nach einem Erwartungswert/Mittelwert
  - gute Ausnutzung der Ressourcen
  - kann zu Überlastfällen führen
  - Monitor notwendig
    - bemerkt Überlast und veranlasst entsprechende Aktionen
    - z.B. Entzug der Ressource oder Benachrichtigung des Prozesses



Folie 363

## Echtzeit-Prozessverwaltung für MM



- Bisher gibt es i.d.R. in Betriebssystemen
  - entweder Scheduling für Time-Sharing
  - oder Scheduling für Echtzeitanwendungen
- für Multimedia muss ein Scheduler beides können
- Echtzeit heißt hier
  - Daten sind nur brauchbar, wenn sie vor der Deadline bereitstehen bzw. verarbeitet sind



Folie 364

## Unterschiede zu traditionellen Echtzeitsystemen



- Geringere Zuverlässigkeitsanforderungen
  - Daten traditioneller Echtzeitsysteme steuern/regeln Prozesse
  - Multimedia-Daten dienen meist der Präsentation für den Menschen
    - Tolerierbare Verletzung der Zeitschranken
    - Auswirkungen abhängig von der Kodierungstechnik
- Daten treten meist periodisch auf
  - wegen periodischer Abtastung
  - periodische Prozesse sind einfacher zu verwalten als allgemeine Realzeit-Prozesse
  - vorhersagbar



Folie 365

## Ziele der Prozessverwaltung



- Verwaltung der Ressourcen derart, dass alle Deadlines aller Prozesse befriedigt werden
  - bei hoher Auslastung der Betriebsmittel
  - bei schneller Berechnung der Zuteilung
- Zuteilung muss nicht optimal sein
  - dies leisten zu wollen, wäre NP-vollständig



Folie 366

## Anforderungen an den Scheduler



- Scheduler muss Dienstgütegarantien unterstützen
  - Berechnung von Scheduling-Entscheidungen muss Dienstgüteparameter einbeziehen
  - Alle gegebenen Zusagen müssen zur Laufzeit eingehalten werden
- Scheduler sollte die Eigenschaften kontinuierlicher Ströme einbeziehen
  - Periodische Anforderungen sollten gut unterstützt werden
  - Aperiodische Anforderungen (z.B. durch Kontrollprozesse) sollten nicht "verhungern"



Folie 367

## Preemptive vs. Non-preemptive Scheduling

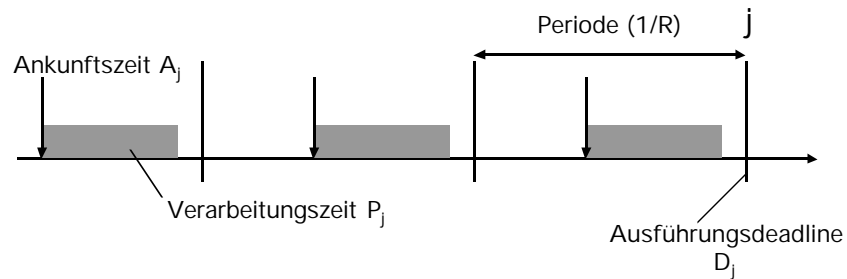


- Preemptive scheduling
  - stellt ein Prozess mit höherer Priorität eine Anforderung, wird dem gerade aktiven Prozess die Ressource entzogen
  - erhöhter Aufwand für die Prozessverwaltung
  - mehr Prozesswechsel
- Non-preemptive scheduling
  - aktive Prozesse werden nicht unterbrochen
  - u.U. durch äußere Zwänge vorgegeben
    - z.B. für einen Prozess, der vom Netz liest
  - weniger Aufwand für Prozesswechsel
- Insbesondere aus Sicht Multimedia:
  - Non-preemptive scheduling ist i.d.R. vorzuziehen, wenn die Ausführungszeiten kurz sind



Folie 368

## Modell eines periodischen Datenstroms



- die Parameter  $A$ ,  $D$ ,  $P$  und  $R$  steuern den Scheduling-Algorithmus

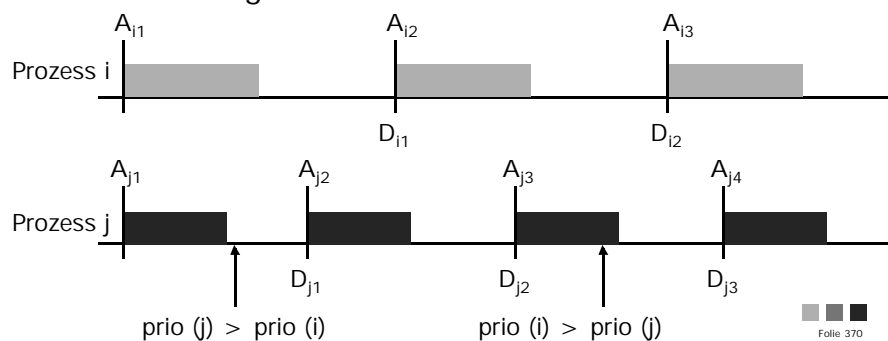


Folie 369

## Earliest Deadline First (EDF)



- Der Prozess mit der frühesten Deadline erhält jeweils die höchste Priorität
- Die Prioritäten ändern sich über der Zeit
- EDF bestimmt stets einen möglichen Schedule, wenn es überhaupt einen gibt
- Auslastung der Ressourcen bis zu 100%



Folie 370

## EDF Scheduling



- in den meisten Fällen des periodischen Schedulings gilt:  
Deadline = Periodenende
  - weil beispielsweise der Datenpuffer wieder gebraucht wird



Folie 371

## EDF Scheduling



- QoS-Berechnung
  - Preemptive scheduling (Liu / Layland, 1973)
    - maximal möglicher Durchsatz

$$\sum_i R_i P_i \leq 1$$

- Paketverzögerung  $\leq 1 / R_i$
  - Non-preemptive scheduling (Nagarajan / Vogt, 92)
    - gleicher Durchsatz wie oben
    - Paketverzögerung  $\leq 1 / R_i + P$ ,  $P$  = Verarbeitungszeit

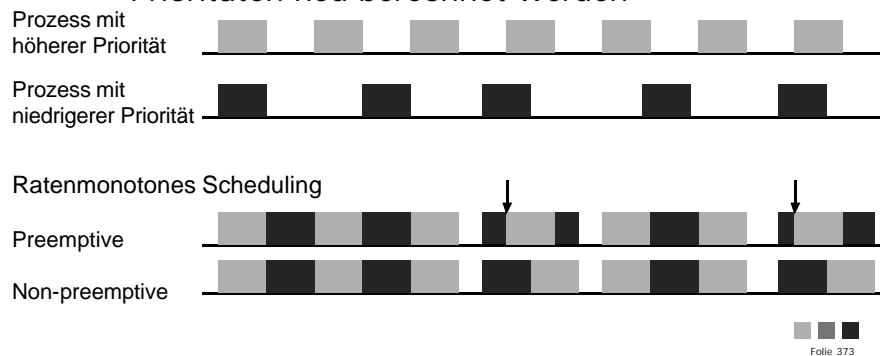


Folie 372

## Ratenmonotones Scheduling



- Der Prozess mit der höchsten Paketrate erhält die höchste Priorität
- Die Priorität ist konstant
  - nur zu Beginn und Ende eines Stromes müssen Prioritäten neu berechnet werden



## Ratenmonotones Scheduling



- Sei Deadline = Periodenende
- QoS-Berechnung
  - Preemptive scheduling (Liu / Layland, 1973)
    - Maximal möglicher Durchsatz:

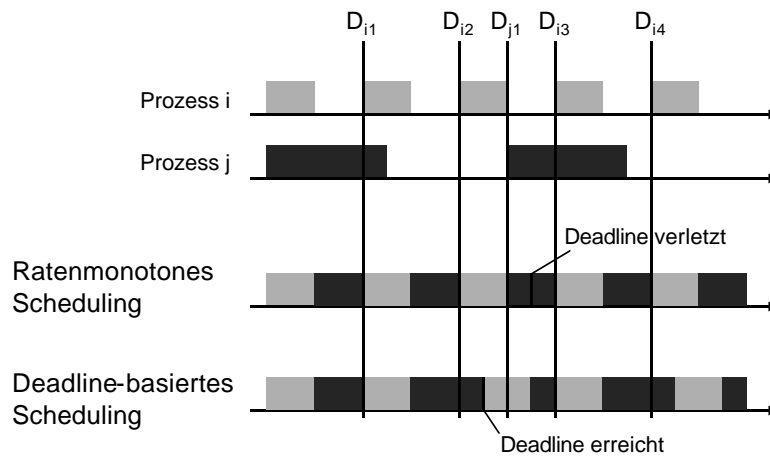
$$\sum R_i P_i \leq \ln 2$$

- Paketverzögerung  $\leq 1 / R_i$
- Non-preemptive scheduling (Nagarajan / Vogt, 92)
  - sehr komplizierte Berechnung
  - garantierter Durchsatz deutlich niedriger



Folie 374

## Beispiel



Folie 375

## Scheduling mit festen Prioritäten



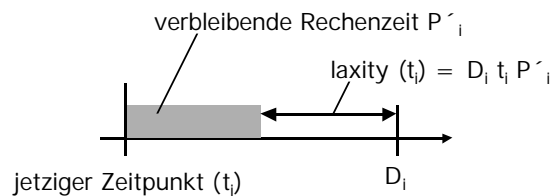
- Jeder Strom erhält eine feste Priorität
  - Ratenmonotones Scheduling ist ein Spezialfall davon
- Die Scheduling-Berechnungen erfolgen auf der Basis von „worst-case“-Annahmen für alle Ströme höherer Priorität

Folie 376

## Scheduling mit festen Prioritäten



- Scheduling nach „laxity“
  - „Laxity“ = maximale Wartezeit bis zum Beginn der Verarbeitung
  - Der Strom mit der kürzesten „Laxity“ erhält die höchste Priorität
  - Die Prioritäten müssen ständig neu berechnet werden



Folie 377

## Implementierungsprobleme



- die tatsächliche Verarbeitungszeit pro Paket auf der aktuellen CPU muss bekannt sein
- eine analytische Berechnung ist kaum möglich
- Lösungsansatz:  
Messung und Normierung



Folie 378