

Kap. 2: Prozesse und Threads

2.3 Scheduling

Stand: WS 08/09 (19.11.08)

Prof. Dr. Wolfgang Kuchlin

Dipl.-Inform., Dr. sc. techn. (ETH)

Arbeitsbereich Symbolisches Rechnen
Wilhelm-Schickard-Institut für Informatik
Fakultät für Informations- und Kognitionswissenschaften

Universität Tübingen

Steinbeis Transferzentrum
Objekt- und Internet-Technologien (OIT)

Wolfgang.Kuechlin@uni-tuebingen.de
<http://www.sr.informatik.uni-tuebingen.de>



Kap 2.3 Prozess-Ablaufplanung (Scheduling)

➤ Inhalt

- Ziele einer Prozess-Ablaufplanung
- Karussell Laufplanung (round robin scheduling)
- Laufplanung mit Prioritäten (priority scheduling)
- Mehrfach-Warteschlangen
- Kürzester Auftrag zuerst
- 2 Ebenen Ablaufplanung



Wolfgang Kuchlin, WSI und STZ OIT, Uni Tübingen 08.06.2009 2



Ziele des Scheduling

- Fairness für die Prozesse
- Maximale HW-Auslastung
- Minimaler Scheduling-Aufwand
- Tolerierbare Antwortzeiten
- Bevorzugung von Prozessen mit erwünschtem Verhalten
- Sinnvolles Verhalten bei Überlast
- Behandeln von Prozessen gemäß ihrer Wichtigkeit
- Konsistentes und voraussagbares Schedulingverhalten



Wolfgang Kuchlin, WSI und STZ OIT, Uni Tübingen 08.06.2009 3



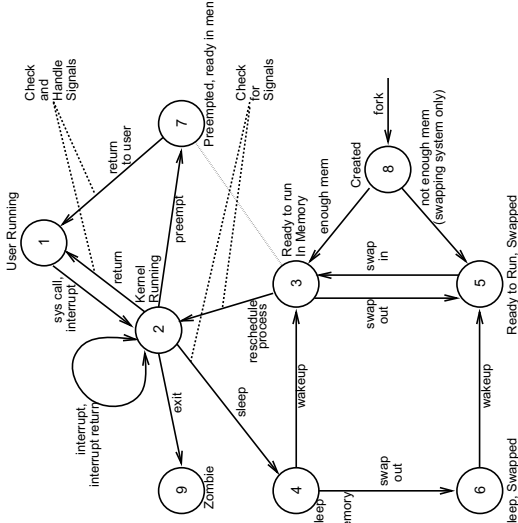
Scheduling

- Multiprogramming
 - CPU Auslastung maximieren
- Time-Sharing
 - Wechsel der Prozesse so schnell, dass interaktives Arbeiten möglich ist.
- Realzeit-Systeme
 - Reaktionszeiten garantieren (Echtzeit → **Rechtzeitig**)
- Queues
 - Job Queue
 - Ready Queue
 - Device Queue
- Prozesse migrieren zwischen den Queues.



Wolfgang Kuchlin, WSI und STZ OIT, Uni Tübingen 08.06.2009 4





Scheduling-Warteschlange Lineare Liste von Prozessen (PCBs)

- Kurzfristiger Scheduler
 - verwaltet CPU
 - wählt Prozesse aus und teilt ihnen CPU zu (dispatching)
- Mittelfristiger Scheduler (Swapper)
 - entscheidet, welche Prozesse im Hauptspeicher bleiben und welche ausgelagert werden. Unterstützt Speicherverwaltung und gleicht Systemlast aus.
- Langfristiger Scheduler (Job-Scheduler in Batch-Systemen)
 - entscheidet, welche Programme als nächstes gestartet (Prozesse zum System zulassen und in Hauptspeicher laden)



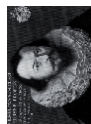
Scheduling-Warteschlange Lineare Liste von Prozessen (PCBs)

- Dispatcher
 - übernimmt Aufsetzen des Prozesses nach Auswahl durch Scheduler
 - Context Switching/Umschaltung in User Mode / Übergeben der Kontrolle an den Prozess



Non-Preemptive Scheduling

- Laufender Prozess kann nicht unterbrochen werden, bleibt auf CPU bis er entweder
 - blockiert oder
 - terminiert
 - (Systemaufruf tätigt und Prozess höherer Prio vorhanden ist)
- Max. Laufzeit eines Prozesses vorhersagbar (keine unfreiwillige Unterbrechung unbestimmter Dauer)
- Einsatz in Realzeitsystemen
 - Systeme mit deterministischem Verhalten: Vollständiges Ablaufdesign.



Preemptive Scheduling

BS 1, WS 2008/09

- Preemption (=Verdrängung)
 - Allgemein in BS: Entzug von Betriebsmitteln
 - Beim Scheduling: Entzug des Prozessors
- Laufender Prozess wird vom Prozessor genommen bei
 - Termination
 - Blockierung nach system call (Wechsel *running* → *blocked*)
 - Preemption
 - für Scheduling relevantes Ereignis tritt ein
 - Interrupts, insbesondere Timer-Interrupts
 - System call
 - Neuer Prozess wird lauffähig und/oder Prioritäten ändern sich
 - Scheduler bestimmt Prioritäten neu
 - Scheduler findet anderen Prozess höchster Priorität
 - Scheduler verdrängt bisherigen Prozess (*running* → *ready*)
 - Scheduler veranlasst dispatching des neuen Prozesses höchster Priorität



Wolfgang Küchlin, WSI und STZ OIT, Uni Tübingen 08.06.2009

9



Einsatz des Preemptive Scheduling

BS 1, WS 2008/09

- Priorität von wichtigen Prozessen im Bedarfsfall
 - Bei reaktiven Systemen (Realzeit) u.U. sofortige Reaktion auf externe Ereignisse
 - > laufender Prozeß wird verdrängt.
- Akzeptable Antwortzeiten (Multitasking)
 - Bei interaktiven Systemen (Timesharing) soll ein CPU-lastiger Prozeß (z.B. Simulationsberechnung) regelmäßig unterbrochen werden von interaktiven Prozessen
 - z.B. interaktive Dateneingabe durch Benutzer



Wolfgang Küchlin, WSI und STZ OIT, Uni Tübingen 08.06.2009

10



Beeinflussung des Scheduling

BS 1, WS 2008/09

- Folgende Prozesseigenschaften können (abhängig vom eingesetzten Verfahren) das Scheduling beeinflussen
 - Laufzeitverhalten
 - I/O-lastiger Prozeß blockiert nach kurzer Laufzeit
 - CPU-lastiger Prozeß wird meistens preempted
 - Wichtigkeit des Prozesses
 - Ressourcenbedarf des Prozesses
 - Bisher verbrauchte Prozessorzeit
 - Bisherige Verweildauer im System
 - Speicherverhalten, Page Fault Rate
 - Bedarf an Ressourcen (neben Speicher/CPU)
 - Noch nötige Prozessorzeit bis zur Termination



Wolfgang Küchlin, WSI und STZ OIT, Uni Tübingen 08.06.2009

11



FIFO-Scheduling

BS 1, WS 2008/09

- Einfache First-In First-Out Queue
- Neue Prozesse werden an die Queue angefügt
- Non-Preemptive
 - Prozess läuft bis zur Termination oder bis zum Blockieren
- Große durchschnittliche Antwortzeit
 - > ungeeignet für interaktive Systeme

Ungebräuchlich, enthalten in anderen Scheduling-Algorithmen



Wolfgang Küchlin, WSI und STZ OIT, Uni Tübingen 08.06.2009

12



Round Robin Scheduling

- FIFO-Scheduling mit Time-Slice
 - Preemptive
 - Limitierte Antwortzeiten
 - > geeignet für interaktive Systeme
- Bestimmung der Time-Slice
 - kurze Zeitscheibe ==> relativ großer Overhead durch Context Switch
 - kurze Antwortzeiten
 - lange Zeitscheibe ==> kleiner Overhead, lange Antwortzeiten
 - typischer Bereich: 10-100 ms



Round Robin Scheduling

Timeslice/Zeitscheibe/Zeitquantum

Prozessor hat spezielles Register, das beim Laden des Prozesses entsprechend Zeitscheibe gesetzt wird.
Bei jedem Ablauf der HW-Zeitscheibe wird Register dekrementiert (parallel zur Bearbeitung des Prozesses), bei 0 wird HW-Interrupt ausgelöst.



Priority Scheduling

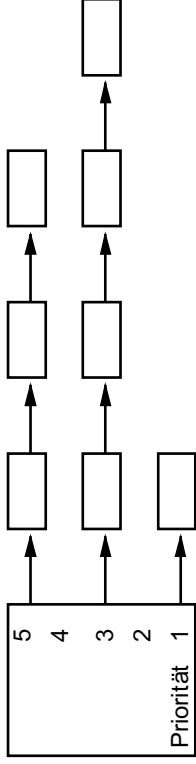
- Unterteilung der Prozesse in Prioritätsklassen
- Separate Round Robin Queue für jede Priorität
- Höhere Priorität hat absoluten Bearbeitungs-Vorrang vor allen anderen.
 - Problem
 - > Prozesse mit tieferer Priorität können verhungern
 - Lösungen
 - (gewisse) Zeit für tiefere Prioritäten reservieren
 - Periodische Erhöhung der Prioritäten von nicht bearbeiteten Prozessen (=> Multi-Level Feedback Queue Scheduling)



Multi-Level Feedback Queue Scheduling

- Zusätzlich
 - Veränderung der Prioritäten aufgrund des Prozessverhaltens
 - blockiert: Priorität wird erhöht
 - preempted: Priorität wird gesenkt





- Ziel
 - Gleichzeitige Bearbeitung von Realtime- und Timesharing-Applikationen
- Neue Scheduling Klassen-Einteilung
 - Realtime
 - System
 - User
- Preemptive
- Prioritätsgesteuerte Multilevel (32) Queue
 - Immer der lauffähigste Prozeß mit der höchsten Proirität wird ausgewählt



- 0-15: Timesharing
 - Round Robin mit Timeslice innerhalb Prioritätsklasse
 - Prozessverhalten verändert Priorität (Feedback, Priorität verbleibt innerhalb 0-15)
 - Unterbrechung bei Ende der Timeslice oder falls Prozess mit höherer Priorität lauffähig wird
- 16-31: Realtime
 - FIFO innerhalb Prioritätsklasse
 - Feste Prioritäten
 - Unterbrechung nur falls Prozess mit höherer Priorität lauffähig wird.



- Ziel
 - Timesharing, Bevorzugung von interaktiven Prozessen, nicht-interaktive Prozesse sollen aber nicht verhungern.
- Multilevel Feedback
- Preemptive (Timeslices)
- Anforderungen an Realtime-Systeme erfüllt
- Zuteilung von CPU-Zeit aufgrund der Prioritäten



- Unterbrechung eines Prozesses
 - Ablauf der Zeitscheibe
 - Blockieren (aber nicht, weil ein anderer Prozess mit höherer Priorität deblockiert wird)
- Feedback
 - Bisherige Laufzeit wird gewichtet akkumuliert, d.h. wenn der Prozess läuft, wird erhöht, wenn er nicht läuft, wird gesenkt.

