

Betriebssysteme (BS)

VL 9.2 – Fadenverwaltung – Scheduler

Volkmar Sieh / Daniel Lohmann

Lehrstuhl für Informatik 4
Verteilte Systeme und Betriebssysteme

Friedrich-Alexander-Universität
Erlangen Nürnberg

WS 21 – 20. Dezember 2021

https://www4.cs.fau.de/Lehre/WS21/V_BS



Agenda

Einordnung

Betriebssystemfäden

Motivation

Kooperativer Fadenwechsel

Präemptiver Fadenwechsel

Ablaufplanung

Grundbegriffe und Klassifizierung

unter Windows

unter Linux

Zusammenfassung



■ **Scheduler**

- trifft **strategische Entscheidungen** zur Ablaufplanung
- betrachtet wird immer eine Menge lauffähiger Fäden
 - die Fäden sind allgemein in einer CPU-Warteschlange aufgereiht
 - die Sortierung erfolgt entsprechend der **Scheduling-Strategie**
- laufende Faden ist immer von der Entscheidung mit betroffen
 - dazu muss der laufende Faden jederzeit „greifbar“ sein
 - vor der Umschaltung ist der laufende Faden zu vermerken
- ein ausgewählter neuer Faden wird dem *Dispatcher* übergeben

■ **Dispatcher**

- setzt die Entscheidungen durch und schaltet Fäden (mit Hilfe von *resume*) um



Ablaufplanung: Einteilung ...

- nach der **Betriebsmittelart**
der zeitweilig belegten Hardware-Ressourcen
- nach der **Betriebsart**
des zu bedienenden/steuernden Rechnersystems
- nach dem **Zeitpunkt**
der Erstellung des Ablaufplans
- nach der **Vorhersagbarkeit**
von Zeitpunkt und Dauer von Prozessabläufen
- nach dem **Kooperationsverhalten**
der (Benutzer/System-) Programme
- nach der **Rechnerarchitektur**
des Systems
- nach der **Ebene der Entscheidungsfindung**
bei der Betriebsmittelvergabe



... nach der Betriebsmittelart

■ ***CPU scheduling***

des Betriebsmittels "CPU"

- die Prozessanzahl zu einem Zeitpunkt ist höher als die Prozessoranzahl
- ein Prozessor ist zwischen mehreren Prozessen zu multiplexen
- Prozesse werden dem Prozessor über eine Warteschlange zugeteilt

■ ***I/O scheduling***

des Betriebsmittels "Gerät", speziell: "Platte"

- gerätespezifische Einplanung der von Prozessen abgesetzten E/A-Aufträge
- *disk scheduling*, z.B., berücksichtigt typischerweise drei Faktoren:
 - (1) Positionszeit, (2) Rotationszeit, (3) Transferzeit
- Geräteparameter und Gerätezustand bestimmen die nächste E/A-Aktion
- die getroffenen Entscheidungen sind ggf. nicht konform zum *CPU scheduling*



... nach der Betriebsart

■ ***batch scheduling***

interaktionsloser bzw. unabhängiger Programme

- nicht-verdrängende bzw. verdrängende Verfahren mit langen Zeitscheiben
- Minimierung der Kontextwechselanzahl

■ ***interactive scheduling***

interaktionsreicher bzw. abhängiger Programme

- ereignisgesteuerte, verdrängende Verfahren mit kurzen Zeitscheiben
- Antwortzeitminimierung durch Optimierung der Systemaufrufe

■ ***real-time scheduling***

zeitkritischer bzw. abhängiger Programme

- ereignis- oder zeitgesteuerte **deterministische** Verfahren
- Garantie der Einhaltung umgebungsbedingter Zeitvorgaben
- Rechtzeitigkeit ist entscheidend und nicht Geschwindigkeit



... nach dem Zeitpunkt

■ **online scheduling**

dynamisch, **während** der eigentlichen Ausführung

- interaktive und Stapelsysteme, aber auch weiche Echtzeitsysteme

■ **offline scheduling**

statisch, **vor** der eigentlichen Ausführung

- wenn die Komplexität eine Ablaufplanung im laufenden Betrieb verbietet
 - Einhaltung aller Zeitvorgaben garantieren: ein NP-vollständiges Problem
 - kritisch, wenn auf jede abfangbare katastrophale Situation zu reagieren ist
- Ergebnis der Vorberechnung ist ein vollständiger Ablaufplan (in Tabellenform)
 - (semi-) automatisch erstellt per Quelltextanalyse spezieller "Übersetzer"
 - oft zeitgesteuert abgearbeitet/ausgeführt als Teil der Prozessabfertigung
- die Verfahren sind zumeist beschränkt auf strikte Echtzeitsysteme



... nach der Vorhersagbarkeit

■ ***deterministic scheduling***

bekannter, exakt vorberechneter Prozesse

- Prozesslaufzeiten/-termine sind bekannt, sie wurden ggf. "offline" berechnet
- die genaue Vorhersage der CPU-Auslastung ist möglich
- das System garantiert die Einhaltung der Prozesslaufzeiten/-termine
- die Zeitgarantien gelten unabhängig von der jeweiligen Systemlast

■ ***probabilistic scheduling***

unbekannter Prozesse

- Prozesslaufzeiten/-termine bleiben unbestimmt
- die (wahrscheinliche) CPU-Auslastung kann lediglich abgeschätzt werden
- das System kann Zeitgarantien nicht geben und auch nicht einhalten
- Zeitgarantien sind durch Anwendungsmaßnahmen bedingt erreichbar



... nach dem Kooperationsverhalten

■ ***cooperative scheduling***

voneinander abhängiger Prozesse

- Prozesse müssen die CPU freiwillig abgeben, zugunsten anderer Prozesse
- die Programmausführung muss (direkt/indirekt) **Systemaufrufe** bewirken
- die Systemaufrufe müssen (direkt/indirekt) den *Scheduler* aktivieren

■ ***preemptive scheduling***

voneinander unabhängiger Prozesse

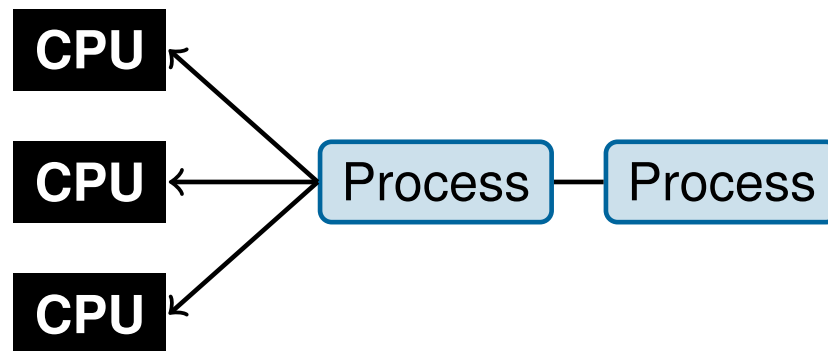
- Prozessen wird die CPU entzogen, zugunsten anderer Prozesse
- **Ereignisse** können die Verdrängung des laufenden Prozesses bewirken
- die Ereignisverarbeitung aktiviert (direkt/indirekt) den *Scheduler*



- ***uni-processor scheduling***
in Mehr{programm,prozess}systemen
 - die Verarbeitung von Prozessen kann nur pseudo-parallel erfolgen
- ***multi-processor scheduling***
in Systemen mit gemeinsamen Speicher
 - die parallele Verarbeitung von Prozessen wird ermöglicht



- ***uni-processor scheduling***
in Mehr{programm,prozess}systemen
 - die Verarbeitung von Prozessen kann nur pseudo-parallel erfolgen
- ***multi-processor scheduling***
in Systemen mit gemeinsamen Speicher
 - die parallele Verarbeitung von Prozessen wird ermöglicht
 - alle Prozessoren arbeiten eine globale Warteschlange ab



■ ***uni-processor scheduling***

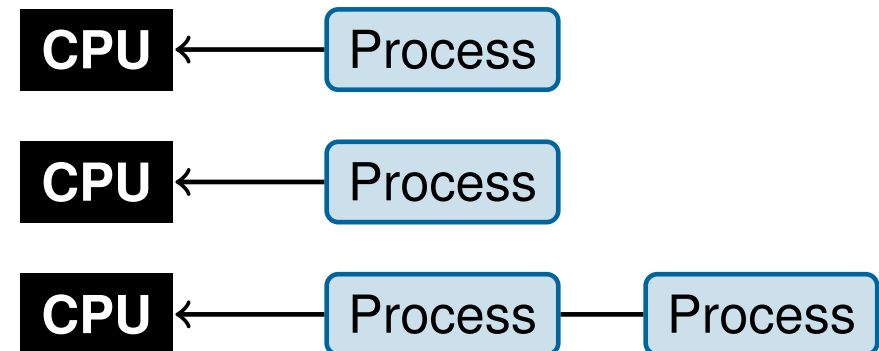
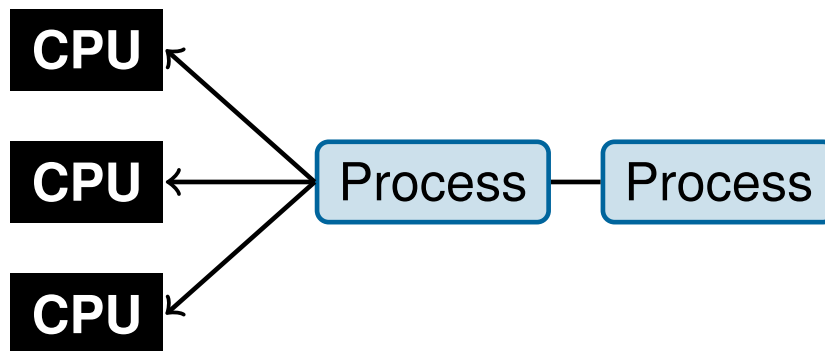
in Mehr{programm,prozess}systemen

- die Verarbeitung von Prozessen kann nur pseudo-parallel erfolgen

■ ***multi-processor scheduling***

in Systemen mit gemeinsamen Speicher

- die parallele Verarbeitung von Prozessen wird ermöglicht
 - alle Prozessoren arbeiten eine globale Warteschlange ab
 - jeder Prozessor arbeitet seine lokale Warteschlange ab



... nach der Ebene

- ***long-term scheduling*** [s – min]
kontrolliert den Grad an Mehrprogrammbetrieb
 - Benutzer Systemzugang gewähren, Programme zur Ausführung zulassen
 - Prozesse dem *medium-* bzw. *short-term scheduling* zuführen
- ***medium-term scheduling*** [ms – s]
als Teil der Ein-/Auslagerungsfunktion
 - Programme zwischen Vorder- und Hintergrundspeicher hin- und herbewegen
 - *swapping*: auslagern (*swap-out*), einlagern (*swap-in*)
- ***short-term scheduling*** [μ s – ms]
regelt die Prozessorzuteilung an die Prozesse
 - ereignisgesteuerte Ablaufplanung: Unterbrechungen, Systemaufrufe, Signale
 - Blockierung bzw. Verdrängung des laufenden Prozesses



Scheduling-Kriterien

- **Antwortzeit** Minimierung der Zeitdauer von der Auslösung einer Systemanforderung bis zur Entgegennahme der Rückantwort, bei gleichzeitiger Maximierung der Anzahl interaktiver Prozesse.
- **Durchlaufzeit** Minimierung der Zeitdauer vom Starten eines Prozesses bis zu seiner Beendigung, d.h., der effektiven Prozesslaufzeit und aller Prozesswartzeiten.
- **Termineinhaltung** Starten und/oder Beendigung eines Prozesses zu einem fest vorgegebenen Zeitpunkt.
- **Vorhersagbarkeit** Deterministische Ausführung des Prozesses unabhängig von der jeweils vorliegenden Systemlast.
- **Durchsatz** Maximierung der Anzahl vollendeter Prozesse pro vorgegebener Zeiteinheit. Liefert ein Maß für die geleistete Arbeit im System.
- **Prozessorauslastung** Maximierung des Prozentanteils der Zeit, während der die CPU Prozesse ausführt, d.h., "sinnvolle" Arbeit leistet.
- **Gerechtigkeit** Gleichbehandlung der auszuführenden Prozesse und Zusicherung, den Prozessen innerhalb gewisser Zeiträume die CPU zuzuteilen.
- **Dringlichkeiten** Bevorzugte Verarbeitung des Prozesses mit der höchsten (statisch/dynamisch zugeordneten) Priorität.
- **Lastausgleich** Gleichmäßige Betriebsmittelauslastung bzw. bevorzugte Verarbeitung der Prozesse, die stark belastete Betriebsmittel eher selten belegen.

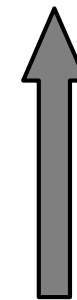


Scheduling-Kriterien

- Antwortzeit
- Durchlaufzeit
- Termineinhaltung
- Vorhersagbarkeit
- Durchsatz
- Prozessorauslastung
- Gerechtigkeit
- Dringlichkeiten
- Lastausgleich

Benutzerorientierte Kriterien

- wahrgenommenes Systemverhalten
- bestimmen die Akzeptanz durch Benutzer



beeinflussen

Systemorientierte Kriterien

- effiziente Nutzung der Betriebsmittel
- bestimmen die Kosten des Rechnerbetriebs



Kriterien bei typischen Betriebsarten

- allgemein (unabhängig von der Betriebsart)
 - Gerechtigkeit
 - Lastausgleich
- **Stapelsysteme**
 - Durchsatz
 - Durchlaufzeit
 - Prozessorauslastung
- **interaktive Systeme**
 - Antwortzeit
(Proportionalität – Bearbeitungsdauer entspricht Erwartung)
- **Echtzeitsysteme**
 - Dringlichkeit
 - Termineinhaltung
 - Vorhersagbarkeit

