

Wiederholung

Ein Prozess ist ein Programm in Ausführung

Ein Prozess ist eindeutig durch seine Prozess-ID (PID) identifiziert

Multitasking: Ein Rechner mit einer CPU kann mehrere Aufgaben gleichzeitig erledigen, d.h. mehrere Prozesse gleichzeitig bearbeiten. Es sind mehrere Prozesse aktiv.

Motivation Scheduling:

Das Scheduling (Zeitplanung) legt fest zu welcher Zeit ein Prozess die CPU des Rechners zugeteilt bekommt.

Der Scheduler im Betriebssystem regelt die zeitliche Ausführung der Prozessen in einem Multitasking System

Präemptives Scheduling (Unterbrechende Zeitplanung)

Unterbrechende Scheduler teilen die CPU von vornherein nur für eine bestimmte Zeitspanne zu und entziehen dem Prozess diese daraufhin wieder
Standard in LINUX/UNIX oder Windows Systemen

Der Prozess kann nicht entscheiden, dass der Scheduler dem Prozess die CPU nicht wegnimmt.

Wird in allen Multitasking Betriebssystemen wie Windows, LINUX/UNIX, Android, iOS verwendet. Der Entwickler hat keine Möglichkeit zu entscheiden wie lange die CPU den Prozess bearbeitet

Non-Präemptives Scheduling (Nicht-Unterbrechende Zeitplanung).

Nicht unterbrechende Scheduler lassen einen Prozess, nachdem ihm die CPU einmal zugeteilt wurde, solange laufen, bis dieser diese von sich aus wieder freigibt oder bis er blockiert.

Wichtig bei Scheduling

Beim Scheduling von Prozessen müssen Informationen der einzelnen Prozesse gespeichert werden, um diesen zu einem späteren Zeitpunkt wieder fortsetzen zu können

Die Prozess Tabelle

Die Prozess-Tabelle beinhaltet pro laufenden Prozess einen Eintrag, den sog. Prozess-Kontroll-Block (Process-Control-Block) oder auch Prozess-Kontext (Process-Context)

Der Prozess-Wechsel wird auch als Context-Switch bezeichnet

Der Prozess-Kontroll-Block beinhaltet

- Die eindeutige Prozess ID (PID)
- Die Register-Werte des Prozesses, der Programm-Zähler des Prozesses, und der Stack-Pointer
- Den Address-Raum des Prozesses
- Den Status des Prozesses
- Die Priorität
- Statistische Prozess-Informationen (z.B. Dauer der verbrauchten CPU-Zeit)
- I/O Informationen wie die Liste der geöffneten Dateien, etc.

Scheduling Verfahren (1)

RR Round Robin (Warteschlangen-Verfahren) Motto „Jeder darf kurz, und der Reihe nach später wieder“

Verbreitetstes Verfahren, fair und einfach zu implementieren

Jeder Prozess erhält für eine konstante Zeit (Quantum t) die CPU

Nach Ablauf des Quantums bekommt der nächste Prozess aus der Warteschlange die CPU. Der zuvor abgearbeitete Prozess wird wieder am Anfang der Liste eingefügt

Hauptproblem: Festlegung einer sinnvollen Zeitspanne für das Quantum

Typische Zeispanne für den Kontext-Switch ist 5 ms

Typische Zeitspanne für ein Quantum ist 100 ms

Scheduling Verfahren (2)

Prioritäten-Basierende Verfahren, Motto: „Der Wichtigste darf zuerst“

Jedem Prozess wird eine Priorität zugewiesen

Ausgeführt wird vom Scheduler der Prozess im Zustand ready mit der höchsten Priorität,

Problem Aushungern: Prozesse mit niedriger Priorität kommen nie zum Zuge. Abhilfe:
Priorität steigt mit steigendem Alter des Prozesses (Aging)

Problem Prioritätenumkehr: Ein hochpriorisierter Prozess wartet auf ein Betriebsmittel, welches ein niedrig priorisierter Prozess reserviert hat. Konsequenz: z.B. Prozesse mit häufigen Leseoperationen von der Festplatte dürfen keine niedrige Priorität besitzen

Scheduling Verfahren (3)

Mischform aus Round Robin und Prioritäten-Basierenden Verfahren bei Dialog-Betriebssystemen

Ein Prozess mit grafischer Benutzeroberfläche (z.B. Excel) befindet sich im Wartezustand (sleeping) auf eine Benutzer-Eingabe

Durch die Benutzer-Eingabe wird der Prozess in den Zustand bereit (ready) gesetzt
Der Prozess bekommt eine sehr hohe Priorität, so dass er vom Scheduler schnell ausgewählt wird

Der Benutzer hat so den Eindruck, dass das Programm sehr schnell auf die Benutzereingabe reagiert