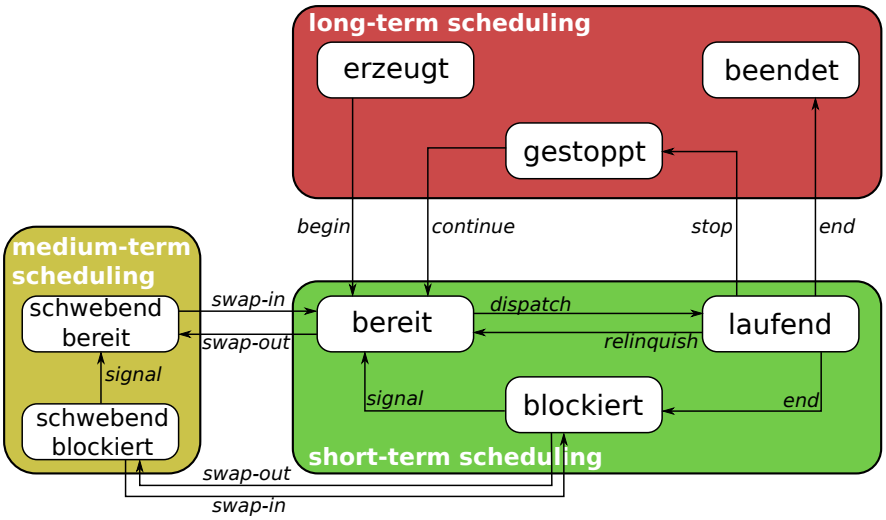
1 Allgemeines

1.1 Dauerhaftigkeit



long-term scheduling *s-min*
Zustände der Prozesse: *erzeugt, gestoppt, beendet*
Die langfristige Einplanung ist Teil der Lastkontrolle. Es wird also geprüft ob ein Prozess laufen kann, oder ob das System bereits überlastet ist.

medium-term scheduling *ms-s*
Zustände der Prozesse: *schwebend bereit, schwebend blockiert*
Die Einlastung der sich hier befindenden Prozesse ist ausgesetzt, da der Adressraum dieses Prozesses auf den Hintergrundspeicher **ausgelagert** ist.

short-term scheduling *µs-ms*
Zustände der Prozesse: *bereit, laufend, blockiert*
Reihenfolge der einzulastenden Prozesse festlegen.

1.1.1 Serialisierung

Um mehr 'gleichzeitig' laufende Fäden als Prozessoren haben zu können müssen Fäden serialisiert werden.
Die mittlere Fadenverzögerung (1 CPU) beträgt bei n Prozessen, die jeweils die Bearbeitungsdauer t_b haben: $\frac{1}{n}\sum_{i=1}^{n}(i-1)\cdot t_b = \frac{n-1}{2}\cdot t_b$

2 Kriterien

Je nach Anforderung muss das Schedulingverfahren gewählt werden. Man unterscheidet zwei Arten der Kriterien.

benutzerorientiert	**systemorientiert**
Antwortzeit	Durchsatz
Durchlaufzeit	Prozessorauslastung
Termineinhaltung	Gerechtigkeit
Vorhersagbarkeit	Dringlichkeit
	Lastausgleich

2.1 Zuordnung zu Betriebsarten

Je nach Betriebsart eines Systems sind unterschiedliche Kriterien wichtig.
Stapelbetrieb: Durchsatz, Durchlaufzeit, Prozessorauslastung
Dialogbetrieb: Antwortzeit
Echtzeitbetrieb: Termineinhaltung, Vorhersagbarkeit, Dringlichkeit

3 Klassifikation von Schedulingverfahren

kooperativ	*vs.*	**preemptiv**
Prozesse geben die CPU freiwillig ab kein CPU-Schutz vorhanden		Prozessen wird die CPU entzogen CPU-Schutz durch verdrängen

deterministisch	*vs.*	**probabilistisch**
CPU-Stoßlängen, Termine sind bekannt		CPU-Stoßlänge, Termine sind unbekannt

offline	*vs.*	**online**
statisch vor der Programmausführung		dynamisch während der Programmausführung

asymmetrisch	*vs.*	**symmetrisch**
seperate Bereitlisten (pro CPU) i.A. ungleichmässige Auslastung		globale Bereitliste (für alle CPUs) i.A. gleichmässige Auslastung

3.1 Beispiele für Schedulingverfahren

3.1.1 FCFS - first come first served

kooperativ
Die Einlastung der Prozesse erfolgt nach ihrer Ankunftszeit. Da es sich um ein **nichtverdrängend**es Verfahren handelt sind kooperative Prozesse Voraussetzung.
Nachteil: **Konvoieffekt** (durch Mix langer und kurzer CPU-Stöße)

3.1.2 RR - round robin

verdrängend
Jeder Prozess erhält eine Zeitscheibe ($\rightarrow$ obere Schranke der CPU-Stoßlänge) und wird nach Ablauf ebendieser verdrängt. Ist der CPU-Stoßvorzeitig beendet wird die CPU dem nächsten Prozess in der Queue zugeteilt.
Nachteil: Prozesse mit **hoher I/O-Aktivität** werden benachteiligt.

3.1.3 VRR - virtual round robin

verdrängend
Ähnlich RR, allerdings mit Vorzugswarteschlangen für E/A intensive Prozesse.
Bei VRR wird ein Prozess, der blockiert (auf E/A wartet) nach Beendigung seiner Blockade in eine Vorzugswarteschlange eingereiht und bekommt die CPU für die Restdauer seiner zuvor nicht ausgenutzten Zeitschreibe zugeteilt.
Prozesse die Verdrängt wurden werden in die 'normale' Warteschlange eingereiht.

3.1.4 SPN - shortest process next

kooperativ probabilistisch
Prozesse werden, der Länge ihrer CPU-Stöße nach aufsteigend, abgearbeitet (ohne Unterbrechung). Hierzu musst eine Abschätzung erfolgen. Es findet **kein Verdrängen** der Prozesse statt.
Zur Abschätzung der CPU-Stoßlänge können folgende Formeln verwendet werden:
Gleiche Gewichtung aller Stöße: $S_{n+1} = \frac{1}{n} \cdot T_n + \frac{n-1}{n} \cdot S_n$
Dämpfung alter Stöße: $S_{n+1} = \alpha T_n + (1 - \alpha) \cdot S_n$
Nachteil: **Verhungern** lang laufender Prozesse möglich.

3.1.5 SRTF - shortest remaining time first

verdrängend probabilistisch
Verdrängendes SPN: Die Umplanung der Prozesse erfolgt ereignisbedingt, z.B. bei Aufhebung der Wartebedingung für einen Prozess.
Wird ein neuer Prozess gestartet, dessen erwartete CPU-Stoßzeit geringer ist als die verbleibende des aktuell Laufenden Prozesses, wird der laufende Prozess verdrängt und der neue Prozess eingelastet.

3.1.6 HRRN - highest response ratio next

verdrängend probabilistisch
Ähnlich SRTF, allerdings wird hier die Wartezeit mit eingerechnet und die Umplanung erfolgt periodisch.
Eingelastet wird der Prozess mit dem größten Verhältniss $R = \frac{w+s}{s}$.

3.1.7 FB - feedback

verdrängend
Hierarchie von Queues mit unterschiedlicher Zeitscheibenlänge. Die Queues nutzen FIFO, mit Ausnahme der tiefsten, welche RR nutzt.
Ein 'neuer' Prozess wird in die 'oberste' Queue mit kürzester Zeitscheibe, aber höchster Priorität, eingereiht. Nutzt ein Prozess die ihm zugeteilte Zeitscheibe komplett aus wird er in die Queue eine Ebene tiefer eingeordnet (längere Zeitscheibe, niedrigere Priorität).

3.1.8 MLQ - multilevel queue

Mehrstufiges Verfahren bei dem zwischen *lokaler Einplanung* und *globaler Einplanung* unterschieden wird. Für jede Art von Prozessen gibt es eine lokale Einplanungsstrategie (mit zugehöriger Queue). Für diese Queues gibt es eine globale Einplanungsstrategie (d.h. 'aus welcher Queue wird der nächste Prozess gewählt').
Die Eigenschaften dieses Verfahrens sind abhängig von den jeweils eingesetzten Verfahren der einzelnen Queues.