

Aufgabe K18: Prozessverwaltung (Nachlese)

Die Zuteilung von Rechenzeit durch den Aufgreifer (engl.: *scheduler*) ist eng verzahnt mit der Verwaltung von Prozessen in der Bereitmenge. Der Aufgreifer muss schnell bestimmen, welcher Prozess als nächstes Rechenzeit erhalten soll. Dazu ordnet er die bereitstehenden Prozesse nach Prioritäten. Eine Tabelle in der englischsprachigen Wikipedia nennt die Algorithmen, die in verschiedenen Betriebssystemen zum Einsatz kommen:

[https://en.wikipedia.org/wiki/Scheduling_\(computing\)#Summary](https://en.wikipedia.org/wiki/Scheduling_(computing)#Summary)

Besonders häufig ist die *multilevel feedback queue* (MLFQ) aufgeführt. Lesen Sie nach, was es damit auf sich hat und beantworten Sie die folgenden Fragen.

Für den Detailgrad dieser Aufgabe genügt die in der Tabelle verlinkte Seite:

https://en.wikipedia.org/wiki/Multilevel_feedback_queue

- a) Welche Auszeichnung erhielt der Erfinder der MLFQ?

Fernando J. Corbató erhielt 1990 den Turing-Award.

https://amturing.acm.org/award_winners/corbato_1009471.cfm

- b) In welcher Größenordnung liegt die Anzahl der *Level*?

Die in der Aufgabenstellung verlinkte Wikipedia-Seite nennt vor der Tabelle einige Beispiele. Konkrete Zahlen liegen im Wertebereich eines Byte, zum Beispiel 32 in Windows oder 256 in OpenBSD. Es geht also um dutzende bis hunderte Level.

- c) Woher kommt das *Feedback*? Wie wirkt es sich aus?

Das Feedback beruht auf dem Verhalten jedes einzelnen Prozessen. Prozesse, die ihre Zeitscheiben aufbrauchen, wandern in Level mit niedrigerer Priorität. Prozesse, die vor Ablauf ihrer Zeitscheibe blockieren, wandern in Level mit höherer Priorität. Das soll Prozesse bevorzugen, die nur kurz auf Ereignisse (I/O) reagieren und schnell wieder blockieren. Bei diesen ist die Reaktionszeit wichtig. Bei Prozessen die lange rechnen spielen Reaktionszeiten eine geringere Rolle.

- d) Vergleichen Sie die MLFQ mit der Datenstruktur aus Aufgabe V2 „Bereitmenge und Aufgreifstrategie“.

Aufgabe V2 verwendet eine *multilevel queue* mit drei Leveln, ohne Feedback.

- e) Eignet sich die MLFQ auch für andere Situationen, in denen Prozesse verwaltet werden? Zum Beispiel bei blockierten Prozessen in einer Wartemenge?

Nein. Das Sortieren von Prozessen nach Prioritäten ist auch in anderen Situationen sinnvoll. Aber das *Feedback* gibt es nur beim Aufgreifer. Die Datenstruktur ist anderswo keine MLFQ mehr. Ob separate Queues für dutzende bis hunderte Prioritätsstufen ohne Feedback noch sinnvoll sind, hängt von der Anzahl der Prozesse und der Häufigkeit des Einfügens ab. Eher nicht.