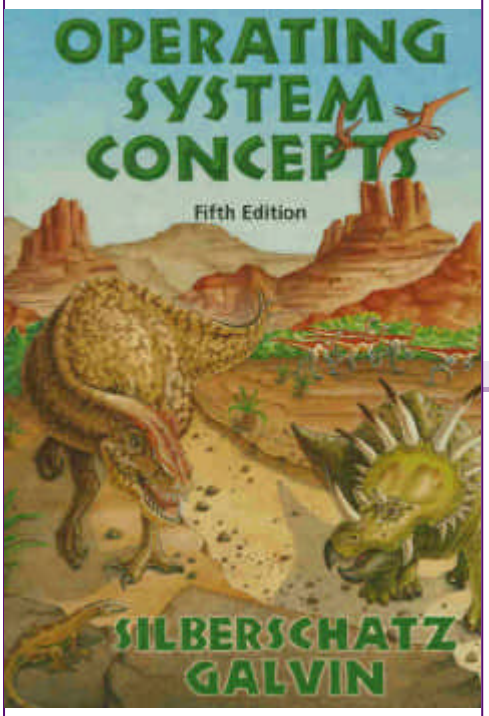
Disk Scheduling

Norbert Bartos

Plattenspeicher sind bekannterweise Massenspeicher mit relativ großer Zugriffs zeit. Um so eher muss da für ge sorgt werden, dass ihre Kooperation mit dem

Operating System Concepts, 5th Ed., Abraham Silberschatz, Peter Baer Galvin, Addison Wesley, 888 Seiten, ATS 673,-, ISBN 0-201-54262-5



Halb lei ter spei cher auf dem Pro zes sor pla ti ne mög lichst ef fi zi ent ab läuft. Der nachstehende Text ist aus folgendem Buch übernommen:

Das Buch ist ein ausgesprochen umfassendes Werk und kann durchaus als ein Stan dard werk an ge se hen wer den. Es wer den alle Aspek te mo der ner Be triebs sys teme behandelt. Am Ende jedes Kapitels be fin den sich vie le Fra gen zur Wie der ho lung und Ver tie fung des In hal tes. Re fe renzen auf rea le Be triebs sy ste me hal ten das Buch praxisnahe. Es ist besonders emp fehlenswert für Per so nen, wel che an Tiefen wis sen im Be reich der Be triebs sys teme in ter es siert sind.

Eine wich ti ge Auf ga be von Be triebs sys temen ist, die Hard ware ef fi zi ent zu nut zen. Im Falle von Disk-Drives bedeutet das eine geringe Zugriffszeit und eine hohe Bandbreite für die Übertragung. Die Zugriffszeit besteht im Wesentlichen aus zwei Komponenten. Die erste ist die Seek-Time, welche notwendig ist, den Arm mit den Lese-Schreib-Köpfen auf demjenigen Zylinder zu po si tio nie ren, der den gewünschten Sektor enthält. Die zweite Zeit ist die Rotational-Latency, wel che ver geht, bis der ge wünsch te Sektor un ter den Kopf ge langt. Die Band brei te der Disk ist die gesamte Anzahl der übertragenen Bytes, dividiert durch die ge sam te Zeit vom er sten I/O- Re quest bis zur Komplettierung des Datentransfers. Man kann nun die Zugriffszeit und die Band brei te durch eine ge eig ne te Wahl der Abarbeitungsreihenfolge der anstehenden Service-Requests beeinflussen. Im rea len Be trieb ist im mer eine be stimm te An zahl die ser Re quests in ei ner Queue anste hend. Der Sche du ler kann nun ver suchen, die Rei hen fol ge für die Aus füh rung zu op ti mie ren.

a) FCFS-Scheduling

Die einfachste Form ist die First-Come-First-Served-Strategie (FCFS, auch FIFO bzw. First-In-First-Out-Strategie genannt). Der Algorithmus ist fair, aber liefert kein zei top ti ma les Sche du ling. Als Bei spiel sei die Reihenfolge der einlangenden Requests für Datenblöcke durch die Zylinder num mern 98, 183, 37, 122, 14, 124, 65, 67 de fi niert (die zu läs si gen Zy lin der nummern sind von 0 bis 199 gewählt). Falls sich der Kopf auf Po si ti on 53 be fin det, so ist die Be we gungs fol ge dann 53, 98, 183, 37, 122, 14, 124, 65, 67. Der Kopf be wegt sich so mit über ins ge samt 640 Zy lin der.

b) SSTF-Scheduling

Eine weitere Möglichkeit ist das Shortest-Seek-Time-First-Verfahren (SSTF). Dabei wird als nächstes derjenige Zylinder aus der Requestliste als Ziel selektiert, welcher der momentanen Position am nächsten ist. In unserem Beispiel ergibt sich dann die Fol ge 53, 65, 67, 37, 14, 98, 122, 124, 183. Da bei wer den nur 236 Zylinder überstrichen.

c) SCAN-Scheduling

Die ses Ver fah ren wird auch Fahr stuhl verfah ren ge nannt, da zu nächst ein mal alle Requests in aufsteigender Folge erfüllt wer den und nach der Um kehr der Kopf be we gungs rich tung die jenigen in absteigender Folge und so weiter. Im bereits be kann ten Bei spiel und ei ner mo men ta nen Kopf be we gung in Rich tung von Zy lin der 0 wür de sich die Fol ge 53, 37, 14, 0, 65, 67, 98, 122, 124, 183 ergeben. Da mit überstreicht man eben falls 236 Zy lin der. Eine Variante ist das C-SCAN-Scheduling, bei dem während der Rückkehr in Richtung von Zy lin der 0 kei ne Re quests an ge nommen wer den. Die Re questli ste wird so mit als zir ku läre Li ste auf ge fasst. Da mit er gibt sich die Folge 53, 65, 67, 98, 122, 124, 183, 199, 0, 14, 37. Somit überstreicht man jetzt (ohne den schnel len Rück lauf) nur mehr 183 Zy lin der.

d) LOOK-Scheduling

So wohl SCAN als auch C-SCAN be we gen den Kopf je weils bis zum er sten (0.) bzw. letzten (199.) Sektor. Das ist eigentlich nicht not wen dig, d.h. die tat säch lich implementierten Verfahren besuchen nur den am wei te sten ent fern ten Zy lin der. Die entsprechenden Verfahren heißen dann LOOK und C-LOOK. Bei LOOK-Verfahren er hal ten wir die Fol ge 53, 37, 14, 65, 67, 98, 122, 124, 183 und über strei chen da mit 208 Zy lin der. Beim C-LOOK-Ver fah ren je doch 53, 65, 67, 98, 122, 124, 183, 14, 37 und so mit wer den 153 Zy lin der überstrichen.

e) Auswahl des geeignetsten Algorithmus

SSTF ist aus dem Bereich des Multiprocessings allgemein bekannt und wird gerne verwendet, SCAN/C-SCAN sind gut für starke I/O-Belastung geeignet. Außer den oben erwähnten Algorithmen existieren noch weitere Verfahren, welche eher von geringerer Bedeutung für die Praxis sind. Natürlich kann für jede Requestliste ein optimales Scheduling durchgeführt werden, aber der Gewinn bei der Zugriffszeit, wird u.U. durch den Aufwand bei der Berechnung wieder zunichte gemacht. Bei neueren Disk-Systemen ist aber auch die Rotational-Latency zu berücksichtigen, da sie in der selben Größenordnung wie die Seek-Time liegt. Das macht aber zusätzliche Probleme, da die physikalische Anordnung der Sektoren eine andere ist, als die logische Anordnung. Damit müsste der Scheduling-Algorithmus auf der Controllerkarte beheimatet sein, was einzelne Kartenhersteller auch schon realisiert haben. Zusammenfassend kann gesagt werden, dass die für ein System optimale Strategie am besten durch Experimente zu ermitteln ist. Jeder Versuch eines absoluten Vergleichs ist von bestimmten Randbedingungen abhängig und damit nicht zu verallgemeinern.