RECHENZENTRUM TH MÜNCHEN
ARBEITSGRUPPE BETRIEBSSYSTEME

INTERNSCHRIFT Nr. 56

THEMA Wartesysteme in Betriebssystemen
bzw. Betriebssystemmodellen

VERFASSER Wolf DATUM 15.3.1971

FORM DER ABFASSUNG

Entwurf
Ausarbeitung
X Endform

SACHLICHE VERBINDLICHKEIT

X Allgemeine Information
Diskussionsgrundlage
Erarbeiteter Vorschlag
Verbindliche Mitteilung
Veraltet

ÄNDERUNGSZUSTAND

BEZUG AUF INTERNSCHRIFTEN Nr. 14, 4o

ANDERWEITIGE LITERATUR

siehe Literaturhinweise am Ende dieser Schrift.

Wartesysteme in Betriebssystemen bzw. Betriebssystemmodellen

Vorbemerkung:

Vor etwa einem Jahr habe ich (für das Institut für Nachrichtenvermittlung und Datenverarbeitung der Universität Stuttgart) kurz dargelegt, welche Arten von Wartesystemen in Betriebssystemen oder Betriebssystemmodellen von Interesse sein könnten. Diese Schrift liegt hier in überarbeiteter Form mit einigen Ergänzungen vor.

1. Allgemeine Gesichtspunkte:

Ein Betriebssystem soll einerseits einen möglichst hohen Programmdurchsatz erzielen und damit die Rechenanlage möglichst gut auslasten, andererseits muß es aber gewisse Zeitbedingungen beachten, insbesondere für Konsolbenutzer hinreichend kurze Antwortzeichen garantieren. Dabei taucht das Problem auf, Komponenten der Anlage, die als sehr "wertvoll" angesehen werden, möglichst gut auszulasten oder den Durchsatz durch Komponenten, die einen "Engpaß" darstellen, geeignet zu steuern. Man wird beim Aufstellen von Modellen versuchen, solche Komponenten als "Schalter" oder "Bedienungsstation" eines Wartesystems zu betrachten, die Dienstleistungen dieser Komponenten als Abfertigung von "Kunden" oder Abarbeitung von "Aufträgen" anzusehen und das Zusammenspiel mit anderen Komponenten in den anderen Charakteristika des Modells (z.B. im Ankunftsprozeß, im Wartespeicher usw.) zu berücksichtigen. Was als Schalter angesehen wird, hängt stark vom gestellten Problem ab; die meisten Untersuchungen betrachten als Bedienungstation den Rechnerkern.

./. 2

Grundsätzlich bestehen jedoch zwei große Probleme bei der Aufstellung und Verwendung von Wartesystemmodellen für die Arbeit an Betriebssystemen:

a) Was an Informationen, speziell an quantitativen Aussagen über ein geplantes (und oft auch über ein bereits laufendes!) Betriebssystem zur Verfügung steht, ist im allgemeinen ziemlich wenig. Außerdem sind Schlüsse von existierenden Systemen auf geplante nur mit großer Vorsicht zu ziehen. Es kann etwa durchaus vorkommen, daß man eine Komponente als Engpaß ansieht, von der sich später herausstellt, daß sie nur zu einem geringen Prozentsatz ausgelastet ist.

b) Es besteht die Gefahr, daß man im Modell die Verknüpfungen mit anderen Komponenten nicht ausreichend berücksichtigt; man erreicht optimales Verhalten für den betrachteten Ausschnitt u.U. um den Preis schwerwiegender Verschlechterungen an anderen Stellen (nur "lokale Optimierung").

Um diesen Problemen abzuhelfen, muß man von vornherein quantitative Untersuchungen einplanen, um eine Grundlage für das Aufstellen von Modellen zu haben, und durch weitere Messungen überprüfen, ob das Modell die Wirklichkeit hinreichend gut annähert.

2. Modell für den Parallelablauf von "Rechnen" und "Ein-/Ausgabe", entwickelt am Beispiel des TR4-Verteilers

Ein erster Schritt zu realistischeren Modellen dürfte darin bestehen, daß man in die Betrachtung neben der Benutzung des Rechnerkerns (RK) auch den Ablauf von E/A-Vorgängen einbezieht Ich möchte mich im folgenden (unter starker Vereinfachung) an das Beispiel des TR4 halten; jedoch lassen sich eine Reihe von Aussagen auch auf andere Systeme übertragen.

./. 3

Daß E/A-Vorgänge, verglichen mit der Abarbeitung von Befehlen im RK, langsam ablaufen, versucht man dadurch zu kompensieren, daß die E/A-Geräte parallel zur Arbeit des RK's betrieben werden können. Da die Ausführung und Koordinierung der Hardware-Starts mit einer Reihe von Problemen verbunden ist, sind diese Startbefehle für ein "normales" Programm, einen Operator, verboten und in ein spezielles (verdrahtetes) Programm, den Verteiler, verlegt. Ein Operator kann lediglich Startwünsche beim Verteiler absetzen und sich über die Beendigung eines E/A-Vorgangs informieren bzw. informieren lassen.

Im zeitlichen Ablauf eines Operators ergibt sich folgendes Bild: Zunächst "rechnet" der Operator, d.h. er belegt den RK; zu einem späteren Zeitpunkt veranlaßt er einen E/A-Vorgang, d.h. er setzt einen Startwunsch beim Verteiler ab. U.U. kann er noch einige Zeit weiterrechnen, ehe er darauf angewiesen ist, daß der E/A-Vorgang abgeschlossen ist. Ist der Operator an dem Punkt angelangt, an dem die Weiterarbeit den Abschluß des E/A-Vorganges voraussetzt (z.B. weil die durch einen Eingabevorgang in den Kernspeicher transportierte Information verarbeitet werden soll), so fragt er beim Verteiler nach einer "Endemeldung" für diesen Transport. Ist eine solche bereits eingetroffen, so kann der Operator weiterrechnen; der E/A-Vorgang wurde zeitlich völlig von der Bearbeitung im RK überlappt. Liegt keine solche Endemeldung vor, so kann der Operator keinen weiteren sinnvollen Gebrauch vom RK machen. Er könnte eine Schleife durchlaufen und periodisch abfragen, ob die Meldung eingetroffen ist. Zweckmäßiger erscheint es, wenn er den RK (sofern dieser anderweitig sinnvoll verwendet werden kann) abgibt und sich vom Verteiler wieder "aufwecken" läßt, wenn die gewünschte Endemeldung kommt. Der Ablauf eines Operators kann (schematisch) angesehen werden als eine Folge von Zeitintervallen, wobei sich

./. 4

der Operator jeweils abwechselnd in den Zuständen "rechnend" (R-Intervall) und "sichtbar auf Abschluß eines E/A-Vorgangs wartend" (E/A-Intervall) befindet. Vereinfachend sei angenommen, daß ein Operatorlauf stets die Form

R-Intervall, E/A-Intervall, R-, E/A-, ..., R-, E/A-Intervall

habe. Mit dieser Schematisierung kann man einen Operator als eine Gruppe von Aufträgen (Kunden) betrachten, die nacheinander zwei Schalter S_R und S_{EA} passieren, wobei der nächste Kunde der Gruppe erst dann in S_R bedient wird, wenn sein Vorgänger S_{EA} verlassen hat. Die Länge des R- bzw. E/A-Intervalls ist die Servicezeit in S_R bzw. S_{EA}. Zur "Servicezeit" in S_{EA} ist anzumerken, daß nur derjenige Teil der E/A-Vorgänge in diesem Modell sichtbar wird, der nicht überlappt werden kann; die Servicezeit in S_{EA} ist also zusammengesetzt aus Wartezeit bis zum Start + Dauer des E/A-Vorgangs, vermindert um die durch Rechnen überlappte Zeit (vorausgesetzt, die sich ergebende Größe ist positiv).

Daß der RK während der nicht überlappten E/A-Zeit eines Operators nicht sinnvoll genutzt wird, ist unbefriedigend. Daher ist im TR4-Verteiler vorgesehen (wenn auch im Betriebssystem nur teilweise genutzt), daß maximal 8 Operatoren gestartet sein können, um (nicht überlappte) E/A-Intervalle eines Operators durch R-Intervalle eines anderen Operators überdecken zu können. Es gibt 8 "Prioritäten", wobei in jeder Priorität höchstens 1 Operator laufen kann, in jeder Prioritätsebene also maximal 1 Gruppe von Kunden warten kann. Gibt ein Operator den RK frei, d.h. verläßt ein Kunde der entsprechenden Gruppe den Schalter S_R, so kann ein Kunde aus der nächsthöheren besetzten Priorität in S_R bedient werden. Kollisionen zwischen den E/A-Vorgängen, die von verschiedenen Prioritäten gestartet werden, seien aus der Betrachtung ausgeklammert, d.h. für jede Priorität i sei ein Schalter S_{EAi}, i = 1,2,...8 vorhanden.

./. 5

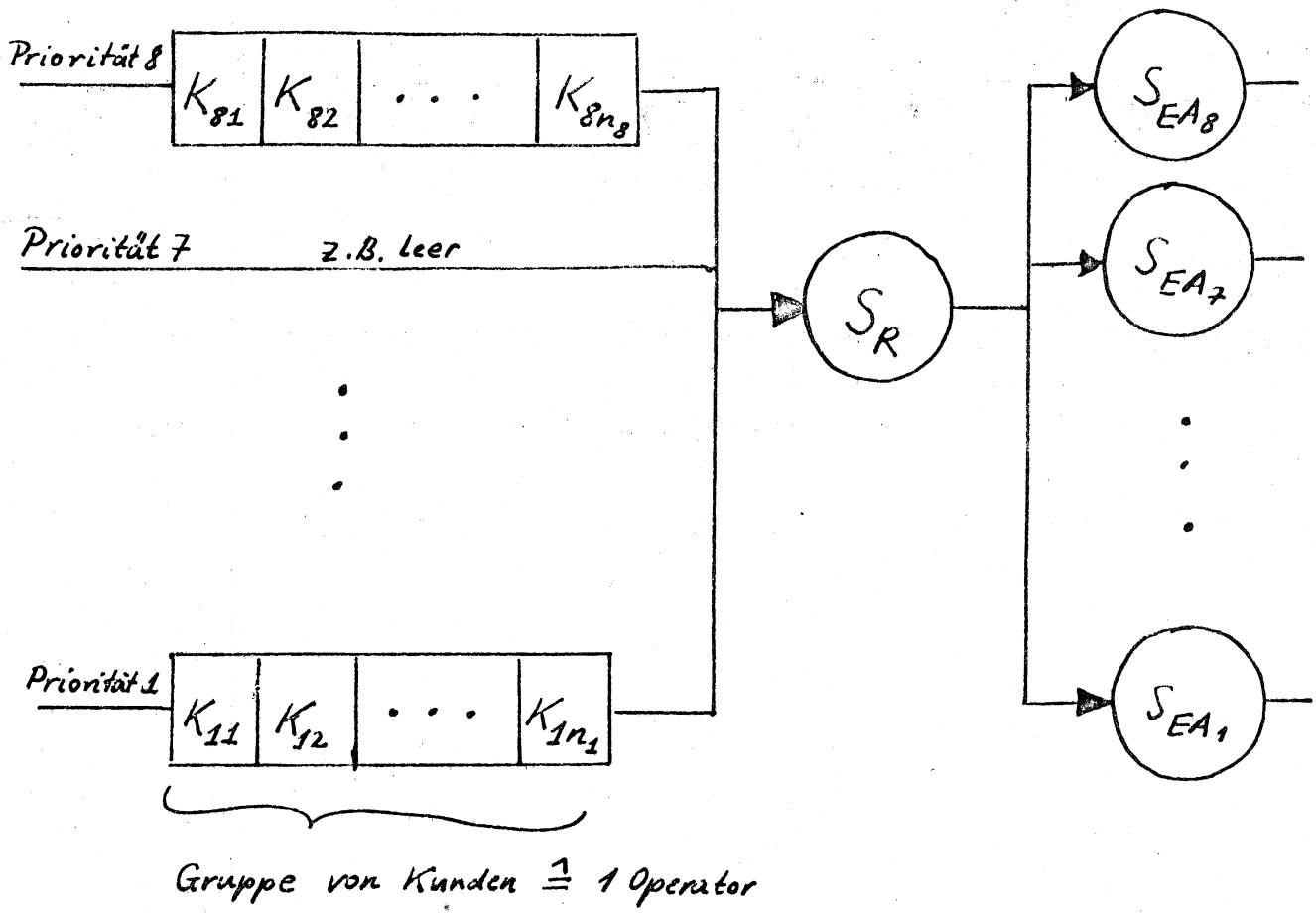

Fig. 1

(Die durch Kollisionen bewirkten Verzögerungen könnten gegebenenfalls in den Servicezeiten in S_{EAi} berücksichtigt werden). Verläßt ein Kunde den Schalter S_{EAi} und ist noch ein weiterer Kunde in Schicht i vorhanden (d.h. der Operatorlauf noch nicht beendet), so wird ein evtl. in S_R befindlicher Kunde niederer Priorität unterbrochen und zurückgestellt und der weitere Kunde aus Schicht i bedient. Damit ergibt sich das in Fig. 1 skizzierte Modell. Für dieses Modell seien die Voraussetzungen noch einmal kurz zusammengestellt:

./. 6

Kunden kommen in jeder Priorität i in Gruppen an (batch arrival), die Größe n_i einer Gruppe ist eine Zufallsveränderliche. Als zusätzliche Einschränkung gilt: pro Prioritätsebene zu jedem Zeitpunkt höchstens 1 Gruppe. Ein Kunde K_{im} aus der Prioritätsebene i hat eine Servicezeit T_{Ri} in S_R und eine Servicezeit T_{EAi} in S_{EAi} (beides Zufallsveränderliche). Beim Übertritt eines Kunden von S_R nach S_{EAi} kommt ein Kunde aus der nächstniederen besetzten Priorität j ($j < i$), sofern eine solche besetzte Priorität k vorhanden ist, nach S_R. Sobald jedoch ein Kunde mit Priorität $k > j$ den Schalter S_{EAk} verläßt und ein weiterer Kunde mit Priorität k vorhanden ist, wird der Kunde mit Priorität j unterbrochen und zurückgestellt (preemptive - resume priority).

Das hier skizzierte Modell beruht auf recht drastischen Vereinfachungen und ist noch unvollständig: Der für die Aufnahme von "Kundengruppen" (also Operatoren) vorgesehene Speicher ist begrenzt, was im Wartespeicher zu berücksichtigen wäre. (Praxis des TR4-Systems: Operatoren laufen meist nur in 1 Priorität, der sogenannten Hauptstufe). Weiter sind über die Verteilungen der Zufallsveränderlichen keine Angaben gemacht. Als generelle Strategie für dieses Modell kann man angeben: Man lege "E/A-intensive" Programme in hohe, "rechenintensive" in niedrigere Priorität.

./. 7

3. Bemerkungen über Modelle TR440 - BS3

Inzwischen existiert auch ein Modell für die Bearbeitung von "Aufgaben" im BS3 auf dem TR440, das auch durch Messungen unterstützt und kontrolliert wird. (vgl. [1]). Hier sollen nur kurz die dabei verwendeten Definitionen skizziert werden:

Eine "Aufgabe" existiert für den Rechner dann, wenn im Hintergrundspeicher (Trommel, Platte) ein bearbeitbares Programm liegt, das Rechenzeit und Arbeitsspeicherplatz fordert. Eine Aufgabe ist beendet, wenn die Daten auf Trommel und Platte nicht ausreichen, um eine Aufgabe weiter zu rechnen, oder das Programm keine Rechenzeit mehr fordert. Eine Aufgabe besteht aus Teilaufgaben. Die Rechenzeit einer "Teilaufgabe" beginnt nach einem Ladevorgang von Trommel oder Platte und endet durch freiwillige Aufgabe des Rechnerkerns wegen erforderlichen Hintergrundtransports dorthin. Dieser Transport ist durch die entsprechende Transportzeit gekennzeichnet.

Mit diesen Definitionen wird ein Modell für das BS3 aufgebaut, das allerdings den KSP-Bedarf der Aufgaben nur in Form einer festen Anzahl von Plätzen berücksichtigt.

4. Modell für den Parallelablauf von "Rechnen" und Seitentransport

Um den RK optimal auszulasten, benötigt man in vielen Fällen mehr rechenwillige Programme als man gleichzeitig ganz im KSP halten kann. Man hat daher im wesentlichen zwei Möglichkeiten: Entweder man hält stets nur 1 Programm im KSP und muß bei jedem Regiewechsel einen Programmtausch (KSP ⟷ Trommel bzw. schnelle Platte) durchführen (swapping) oder man versucht, von jedem Programm nur gerade benötigte Teile im KSP zu halten; dazu kann man den KSP in "Kacheln"

./. 8

gleicher Länge, entsprechend die Programme in "Seiten" aufteilen und versuchen, die Kacheln möglichst nur mit gerade "aktuellen" Seiten zu belegen (paging). Im folgenden soll nur die zweite Möglichkeit betrachtet werden.

In diesem Fall kommen zu den bereits oben beschriebenen E/A-Vorgängen die Seitentransporte als eine Klasse sehr häufiger und relativ kurz dauernder E/A-Vorgänge hinzu. Da ein Modell, das alle E/A-Vorgänge einschließlich der Seitentransporte beschreibt, ziemlich kompliziert werden dürfte, erscheint es sinnvoll, sich auf ein hinreichend kurzes Zeitintervall zu beschränken und nur Seitentransporte zu betrachten. Ein relativ einfaches Modell dieser Art existiert für das MTS (Michigan Terminal System) (siehe [2]); in der zitierten Arbeit wird dieses Modell analytisch behandelt und es werden auch die Ergebnisse von Simulationen mitgeteilt. In dem Modell werden der Durchlauf von Programmen durch den RK und die Seitentransporte zwischen Trommel und KSP schematisch dargestellt unter folgenden Voraussetzungen:

Eine feste Anzahl von Programmen befindet sich im System. Ein Programm behält den RK solange, bis es versucht auf eine nicht im KSP stehende Seite zuzugreifen. Danach erhält das nächste Programm den RK, während das erste wartet, bis die verlangte Seite von der Trommel in den KSP gebracht wurde; anschließend wird das Programm als letztes in die Warteschlange vor dem RK eingereiht. Um die analytische Behandlung zu vereinfachen, wird eine (gemäß FIFO abgearbeitete) Warteschlange nur vor dem RK (Schalter S_R) angenommen; die Servicezeit in S_R wird als negativ-exponentialverteilt angenommen. Wartezeiten vor der Trommel werden in die Servicezeit einbezogen. Die Abfertigung an der Trommel (Schalter S_T) wird durch eine geeignete Erlang-Verteilung approximiert

./. 9

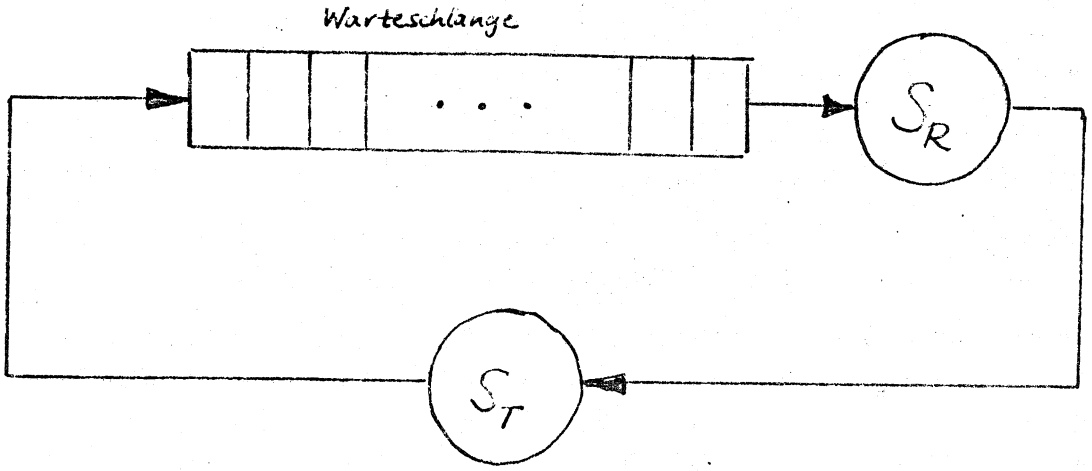

Fig. 2

(und diese Approximation für die gegebene Trommelorganisation durch Berechnungen gerechtfertigt). In diesem Modell ist nur die Verteilung der Zeit zwischen zwei Zugriffen auf fehlende Seiten und das Verhalten der paging-Trommel berücksichtigt; unter den gegebenen Annahmen läßt es sich relativ leicht auch analytisch behandeln. Für weitergehende Untersuchungen sind jedoch komplizierte Modelle notwendig, die nur mittels Simulation zu behandeln sind und deren Aufstellung und Verbesserung durch Messungen am realen System unterstützt werden muß.

./. 1o

5. Abschließende Bemerkungen

Die beschriebenen Modelle leiden alle noch darunter, daß sie die Wirklichkeit nur grob annähern und nur sehr begrenzte Aussagen erlauben. Generell wird in solchen Modellen zu wenig (oder gar nicht) die Größe des KSP berücksichtigt. Es erscheint daher sinnvoll, sich einerseits Modellen zuzuwenden, die im wesentlichen die Ausnützung des KSP betrachten und den RK nur am Rande berücksichtigen (bis hinunter zur Simulation des paging-Algorithmus), andererseits in solchen Modellen, die sich vorrangig mit der Auslastung des RK befassen, doch wenigstens die KSP-Größe und Aussagen über den KSP-Bedarf der Programme (etwa Umfang der "working set" $\approx$ Anzahl der Seiten, die das Programm zu einer "vernünftigen" Arbeit braucht) zu berücksichtigen.

Literaturhinweise

[1] Forster, Mersmann:
Untersuchungen am BS3/2 im Hinblick
auf die Rechnerkern - und Kernspeicherplanung
der Kontrollfunktion

[2] Pinkerton, T.B.:
Program Behavior and Control in Virtual
Storage Computer Systems
University of Michigan, CONCOMP Report 4, April 1968

Wegen weiterer Literaturhinweise siehe auch IS 14 und IS 4o.