

RECHENZENTRUM TH MÜNCHEN
ARBEITSGRUPPE BETRIEBSSYSTEME

INTERNSCHRIFT Nr. 49

THEMA Über die EA-Programmierung bei einem Zweiprozessorsystem

VERFASSER

Ramsperger

DATUM

15.01.71

FORM DER ABFASSUNG

Entwurf

☒ Ausarbeitung

Endform

SACHLICHE VERBINDLICHKEIT

☒ Allgemeine Information
Diskussionsgrundlage

Erarbeiteter Vorschlag

Verbindliche Mitteilung

Veraltet

ÄNDERUNGSZUSTAND

BEZUG AUF INTERNSCHRIFTEN 23, 36, 45

ANDERWEITIGE LITERATUR

Arbeitsunterlage, nicht zur Publikation bestimmt. Weitergabe
an Dritte nur im Einvernehmen mit der Arbeitsgruppe

Bemerkungen zu dieser Internschrift

Auch ein geneigter Leser wird stirnrunzelnd feststellen, daß er in der vorliegenden Arbeit zum Teil mit Zusammenfassungen von Dingen belästigt wird, die schon immer zum Allgemeingut zählten (Kapitel 1.2), über die schon vor langer Zeit geschrieben wurde (Kapitel 3.4), oder über die schon im größeren Rahmen diskutiert wurde (Kapitel 3.2). Der einfachste Grund hierfür ist, daß es sich bei dieser Internschrift um den Abzug eines Manuskriptes handelt, aus dem der Verfasser auf der Fachtagung "Rechnerstrukturen und Betriebsprogrammierung" (5. - 7.10.2970, Erlangen) vorgetragen hat.

Diese Arbeit gibt einen Teil von den Diskussionen wieder, die innerhalb der Arbeitsgruppe Kanalbewacher gehalten wurden, und deren Ergebnisse wesentlich in die Konstruktion des KB eingingen. Völlig ausgespart wurden Überlegungen zur Kopplung zweier Rechner.

Fast überflüssig erscheint die Bemerkung, daß die Menge der dargestellten Alternativen eine echte Untermenge der Menge der möglichen Alternativen ist.

Der KB kann gleichzeitig auf beiden RK laufen. Diese Tatsache geht ein bei der Benutzung von Daten, und das wird in der vorliegenden Arbeit mit dem Hinweis abgetan, daß an vielen Stellen stillschweigend vorausgesetzte Synchronisierungsmaßnahmen erforderlich seien. Zur näheren Erläuterung sei hier hinzugefügt, daß diese Synchronisierungen durch Elementarmutices realisiert werden (Sperrern, vor denen gekreist wird).

Die Existenz zweier RK kann aber auch wesentliche Voraussetzung für Algorithmen sein, und ist es auch beim KB genau für den im Kapitel 3.2 beschriebenen Regelmechanismus.

Über die EA-Programmierung bei einem Zweiprozessor-system

1. Ein Zweiprozessorsystem
 - 1.1 Vorbemerkungen
 - 1.2 Vorausgesetzte Hardware
2. Der Kanalbewacher (KB)
 - 2.1 Die Notwendigkeit eines KB
 - 2.2 Die Aufgaben des KB
 - 2.3 Zum Aufbau und zur Arbeitsweise des KB
3. Spezielle Probleme des KB
 - 3.1 Auftragswarteschlangen
 - 3.2 Eingriffs-Zuordnung
 - 3.3 Multiplex an Standardkanälen durch den KB
 - 3.4 Die Verwaltung der EA-Zone

1. Ein Zweiprozessorsystem

1.1 Vorbemerkungen

Es sollen einige Aspekte der EA-Programmierung bei einem Zweiprozessorsystem betrachtet werden. Dabei soll nicht verschwiegen werden, daß die Überlegungen dazu zunächst nur theoretischer Natur sind; Erfahrungen mit der aus ihnen abgeleiteten Software werden erst noch gesammelt. Um den Untersuchungen eine Grundlage zu geben, wird ein solches System kurz beschrieben (eine Realisierung ist die Telefunken-TR440-Anlage des Leibniz-Rechenzentrums in München). Dabei wird keine Vollständigkeit angestrebt; die aufgezählten Einzelheiten stellen vielmehr eine für den EA-Verkehr bedeutsame Auswahl dar.

1.2 Vorausgesetzte Hardware

Es sei folgende Hardware gegeben:

zwei Rechnerkerne (RK)
ein Kernspeicher als Arbeitsspeicher,
ein EA-Werk, das unabhängig von den RK und
parallel zu deren Arbeit den gesamten EA-Verkehr
mit den peripheren Geräten durchführt,
die Peripherie, bestehend aus rotierenden Hinter-
grundspeichern (Trommel und Platte), Magnetband-
geräten, langsamen EA-Geräten und einem Satelliten-
rechner (TR86S).

Für das Folgende sind gewisse Einzelheiten des EA-Werkes von Bedeutung. Nach einem Anstoß vom RKr mit dem Befehl "START (Kanal k)" arbeitet es einen EA-Vorgang autonom ab. Der RKr setzt seine Arbeit mit dem nächsten Befehl fort, ohne die Reaktion des EA-Werkes abzuwarten. Diese besteht (normalerweise) darin, daß es für den angesprochenen Kanal k ein EA-Programm in Zeitmultiplex mit EA-Programmen für andere schon aktive Kanäle abarbeitet.

Ein EA-Programm ist eine durch Verweise verkettete Folge von EA-Befehlen. Das sind Worte, die neben einem speziellen dem EA-Werk verständlichen Operationscode Kernspeicheradressen und Längen der zu übertragenden Information enthalten. Das EA-Werk erwartet EA-Programme nur in einem bestimmten EA-Zone genannten Speicherbereich ; speziell das erste Wort eines EA-Programms erwartet es in einer kanalspezifischen Zelle, der Kanalbefehlszelle. Mit EA-Vorgang werde die Abarbeitung eines EA-Programms durch das EA-Werk bezeichnet.

Es gibt zwei Arten von Kanälen: Schnellkanäle, an die nur ein EA-Gerät angeschlossen werden kann und Standardkanäle, die an vier Unterkanälen den Anschluß je eines EA-Gerätes erlauben.

Die von bestimmten EA-Befehlen übertragene Information wertet das EA-Werk als Steuerinformation aus, die bei Standardkanälen auch die Wahl des Unterkanals bestimmt. Auf diese Weise ist es möglich, innerhalb eines EA-Programms mehrere Male den Unterkanal zu wechseln. Es ist jedoch nicht möglich, an einem Standardkanal gleichzeitig mehrere EA-Vorgänge zu betreiben.

Die reguläre Beendigung oder der wegen Fehlers vorzeitige Abbruch eines EA-Vorgangs wird im EA-Werk als Eingriffswunsch (EG-Wunsch) wegen Stop ohne/mit Fehler vermerkt. Weitere Ursachen für das Entstehen von EG-Wünschen sind Anrufe der einzelnen Geräte und bei entsprechender Steuerinformation das Erreichen bestimmter Stellen der Abarbeitung eines EA-Programmes. Ein Anruf sendendes Gerät ist zur Übertragung von Information bereit. Meistens wird ein Anruf durch das Drücken einer Anruftaste im Bedienfeld des Gerätes verursacht.

Das EA-Werk speichert die EG-Wünsche. Ein EG-Wunsch führt zu einem seiner Ursache entsprechenden EG. Einer dynamisch veränderlichen EG-Zuordnungsliste entnimmt das EA-Werk, welcher RK sich für welche EG-Art interessiert. Nach der Abspeicherung eines EG-Wunsches im EA-Werk werden die betroffenen RK zur Abnahme eines entsprechenden EG aufgefordert. Die Abnahme erfolgt dann, wenn der angesprochene RK keine EG-Sperre gesetzt hat. Sie besteht darin, daß der RK die Ausführung des laufenden Programms abbricht und unter Setzen der EG-Sperre und Retten gewisser Zustände an hardwareseitig vorgegebenen Adressen die Ausführung eines anderen Programms aufnimmt, das die Unterbrechung behandeln soll. Dazu ist ihm noch vom EA-Werk ein den EG beschreibendes EG-Wort an eine wohldefinierte von r abhängige Stelle abgelegt worden.

Die Tatsache zweier RK erfordert beim Arbeiten unmittelbar auf der Hardware an vielen Stellen Synchronisierungsmaßnahmen. Im Folgenden werden diese meistens stillschweigend vorausgesetzt.

2. Der Kanalbewacher (KB)

2.1 Die Notwendigkeit eines KB

Es soll die Situation betrachtet werden, daß beim Lauf eines Programms P auf dem RKr ein EA-Programm für ein Gerät G am Kanal $k = k(G)$ aufbereitet und in die EA-Zone abgelegt wird. Zum Start des EA-Programms muß dessen erster Befehl in die entsprechende Kanalbefehlszelle eingetragen werden und anschließend der Befehl "START (Kanal k)" gegeben werden. Der Start kann jedoch nur dann durchgeführt werden, wenn folgende Voraussetzungen erfüllt sind:

- (a) auf RKr' ($r' + r$) wird nicht gleichzeitig ein Startbefehl für denselben Kanal k gegeben,
- (b) über Kanal k läuft kein EA-Vorgang ab.

Die Voraussetzung (a) läßt sich dadurch erfüllen, daß man die Startphase für einen Kanal k zu einer kritischen Sektion macht, in der sich zu jedem Zeitpunkt höchstens ein RK befinden darf. Über die Richtigkeit von (b) kann nur eine Buchführung zum Belegungszustand eines Kanals Auskunft geben: P kann den Startbefehl geben, wenn (b) erfüllt ist; wie soll sich P aber verhalten, wenn (b) falsch ist?

P muß vor einem Start die Bedingung (b) prüfen; wenn sie nicht erfüllt ist, kann P dies entweder ununterbrochen tun (eine sichere, aber in höchstem Maße ineffiziente Methode) oder immer wieder (ein sehr unsicheres Verfahren - Lattenzauneffekt). Beide Antworten sind unbefriedigend. Aber auch die Frage ist schlecht gestellt worden, weil sie impliziert, daß P den Startbefehl geben soll.

Eine Lösung dieses Problems ist, daß im Betriebssystem ein spezieller Modul existiert, der allein die Startbefehle für EA-Vorgänge gibt. Die Aufgabe, einen EA-Vor-

gang zu starten, übernimmt dieser Modul in Form eines EA-Auftrages. Dieser Modul soll Kanalbewacher (KB) genannt werden.

2.2 Die Aufgaben des KB

Nach der aufgezeigten Lösung erscheint der KB als Empfänger von EA-Aufträgen. Einen übernommenen EA-Auftrag ordnet der KB nach bestimmten Gesichtspunkten in die entsprechende kanalspezifische Warteschlange auf der Position p ein (dabei seien mindestens die Positionen $1 \dots p-1$ schon besetzt). Im Fall $p=1$ läuft noch kein EA-Vorgang über den angesprochenen Kanal; der KB startet dann das zum Auftrag gehörige EA-Programm. Im Fall $p \neq 1$ müssen mindestens $p-1$ EA-Vorgänge erledigt worden sein, bevor der neue EA-Auftrag gestartet werden kann.

Der KB muß die Erledigung eines EA-Vorgangs erkennen können, um ihn aus der Warteschlange herausnehmen zu können, und um zugleich dem Absender des entsprechenden Auftrages davon Kenntnis geben zu können. Die Erledigung eines EA-Vorganges kann nur durch einen Stop-EG oder durch einen EG erkannt werden, der das Erreichen einer bestimmten Stelle eines EA-Programmes anzeigt. Diese Eingriffe hat also der KB zu behandeln. Da ist es nur vernünftig, ihm auch noch die Behandlung der Anrufs-EG aufzubürden, zumal diese häufig in bestimmten Zusammenhängen zu EA-Vorgängen stehen.

Die vollständige Behandlung eines EG läßt sich in mehrere Phasen gliedern:

- (a) hardwaremäßige Behandlung
- (b) EG-Vorbehandlung (unter EG-Sperre laufend)
 - (b.1) Konservierung der Unterbrechungsstelle für eine spätere Fortsetzung
 - (b.2) Konservierung der Information über den EG für eine Behandlung des EG zu einem späteren genehmeren Zeitpunkt

- (c) eigentliche EG-Behandlung
(Korrektur der Listen, Rückmeldungen, neue Starts)
- (d) Fortsetzung des unterbrochenen Programms.

Die Durchführung von (c) ist wesentlich Aufgabe des KB. Dagegen ist die Durchführung von (d) i.a. nicht Sache des KB, sondern die einer Rechnerkernvergabe, die die eventuell durch den EG veränderte Situation berücksichtigt und den Unterbrochenen gegebenenfalls erst zu einem späteren Zeitpunkt fortsetzt. (b) kann von einem weiteren speziellen Modul durchgeführt werden oder auch vom KB (dann kann (b.2) entfallen).

Die kanalspezifischen Auftragswarteschlangen sind natürlich begrenzt. Daher kann es geschehen, daß der KB weitere Aufträge zunächst ablehnen muß. Die EA-Programme sind nach den bislang gemachten Annahmen bereits in der EA-Zone abgelegt. Dies stellt sich nun als unzweckmäßig heraus: EA-Programme, die das EA-Werk in nächster Zukunft noch nicht zur Abarbeitung bekommt, blockieren unnötigerweise über zu lange Zeit hinweg den von ihnen belegten Platz in der EA-Zone; ihretwegen kann die EA-Zone zu einem Engpaß bei der Aufbereitung von EA-Programmen werden. Um das zu verhindern, sollten EA-Programme nur so kurz wie möglich in der EA-Zone abgelegt sein. Natürlich müssen sie mindestens für die Zeit dort abgelegt sein, zu der sich die zugehörigen Aufträge in der Warteschlange befinden.

Dies legt folgende Vorgehensweise nahe: mit dem EA-Auftrag wird dem KB lediglich ein vorgefertigtes EA-Programm übergeben, das erst der KB als fertiges EA-Programm in die EA-Zone einträgt.

Damit schälen sich insgesamt folgende Aufgaben für den KB heraus:

- Verwaltung der EA-Zone,
- Bedienung der kanalspezifischen Warteschlangen,
- Behandlung der EG.

2.3 Zum Aufbau und zur Arbeitsweise des KB

Der KB wird aus zwei Richtungen angestoßen:

- "von oben" durch den Auftraggeber bei der Übergabe eines Auftrages
- und "von unten" von der EG-Vorbehandlung, um die eigentliche EG-Behandlung durchzuführen.

Dementsprechend besteht der KB aus zwei Programmteilen: dem KB-Auftragsteil (KBA) und dem KB-Eingriffsteil (KBE).

Der KBA übernimmt die EA-Aufträge, trägt die zugehörigen EA-Programme in die EA-Zone ein und startet sie, wenn der entsprechende Kanal frei ist.

Der KBE untersucht die ihm von der EG-Vorbehandlung gemeldeten EG (dabei kann er im Fehlerfall eine Hardware-orientierte Fehlerbehandlung durchführen), übermittelt gegebenenfalls entsprechende Rückmeldungen, startet das nächste EA-Programm aus der Warteschlange, wenn ein EA-Vorgang mit Stop-EG beendet wurde, und gibt belegte EA-Plätze in der EA-Zone wieder frei.

Der Sinn der EG-Vorbehandlung kann nur darin liegen, daß sie es ermöglicht, den verursachenden EG zu einem späteren geeigneteren Zeitpunkt zu behandeln. Wann kann ein späterer Zeitpunkt geeigneter sein? Wenn im Moment ein noch wichtigeres Programmstück durchgeführt werden sollte als die ohnehin schon sehr wichtige EG-Behandlung. Wenn etwa die Behandlung eines EG von der Trommel durch einen EG von einem langsameren EA-Gerät unterbrochen wird, so ist es sicherlich nicht unvernünftig, die unterbrochene Behandlung des Trommel-EG fortzusetzen und im Anschluß daran den anderen EG zu behandeln. Würde jedoch umgekehrt die Behandlung des EG von dem langsamen EA-Gerät durch den EG von der Trommel unterbrochen werden, so könnte es nützlich sein, in die unterbrochene EG-Behandlung die Behandlung des Trommel-EG einzuschieben.

Will man diese Möglichkeit der Überholung von EG-Behandlungen zulassen, so käme man zu folgenden Regelungen:

Der KB erscheint programmlogisch als eine Menge von Moduln $KBE_0, KBE_1 \dots KBE_n$:

KBE_0 stehe für KBA,

KBE_j behandle die EG vom Gerät G_j ($j \geq 1$).

Wird ein KBE_i durch einen EG von Gerät G_j unterbrochen, so wird von der EG-Vorbehandlung in eine spezielle KB-eigene RK-Vergabe gesprungen. Diese entscheidet nach einem eventuell veränderlichen Prioritätenschema, ob der unterbrochene KBE_i -Lauf gleich fortgesetzt werden soll, oder ob ein neuer Lauf von KBE_j begonnen werden soll. Diese spezielle RK-Vergabe muß natürlich nach einem KBE_i -Lauf wieder angesprungen werden, um nach fortzusetzenden oder neu zu beginnenden KBE_j -Läufen zu suchen. Dabei ist es denkbar, daß ein auf RKr unterbrochener Lauf nun auf RKr' fortgesetzt wird. Erst dann, wenn es keinen solchen mehr gibt, hat der KB seine Tätigkeit beendet.

Diese Vorgehensweise erfordert manchen verwaltungstechnischen Aufwand. Dieser ist nur dann zu rechtfertigen, wenn die Wahrscheinlichkeit für die Unterbrechung einer EG-Behandlung durch einen wichtigeren EG genügend hoch ist (dabei ist zu beachten, daß dieser Aufwand die Wahrscheinlichkeit erhöht). Für das in 1.1 genannte System ergibt eine pessimistische Schätzung unter der Voraussetzung einer gleichmäßigen Verteilung der EG auf die beiden RK eine Wahrscheinlichkeit der Unterbrechung einer EG-Behandlung durch einen EG von maximal etwa 7.5 %. Die Wahrscheinlichkeit, daß dieser unterbrechende EG der wichtigere ist, ist natürlich noch geringer. Für ein solches System verliert die EG-Vorbehandlung ihren Sinn. Statt dessen soll gleich der unter EG-Sperre laufende KBE angesprungen werden.

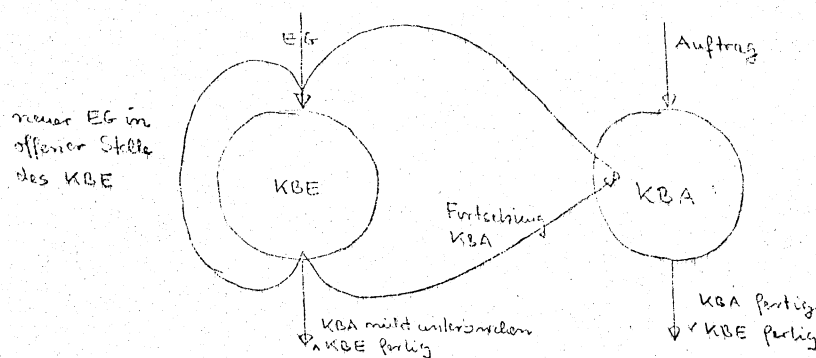
Es ist nützlich, nach erfolgter EG-Behandlung im KBE die EG-Sperre kurz aufzuheben. Steht bereits ein EG-Wunsch an, so kommt er an dieser Stelle durch: es wird ein neuer KBE-Durchlauf begonnen. Für diese wohldefinierte

Unterbrechungsstelle spart man sich das Retten von Fortsetzungsinformationen.

Zur Auftragsübergabe rufe ein Auftraggeber A den KBA wie ein Unterprogramm auf. Wird A durch einen EG unterbrochen, so ist es durchaus möglich, daß durch diesen EG eine neue Situation entstanden ist, auf Grund deren A nach erfolgter EG-Behandlung nicht sofort von einer RK-Vergabe fortgesetzt wird, sondern erst dann, nachdem andere wieder aufgenommenen Programme ihre Tätigkeit beendet haben. Muß mit dieser Möglichkeit auch dann gerechnet werden, wenn A im KBA unterbrochen wurde, so kann sich bei einer rekursiven Wiederholung dieses Ablaufs ein unerwünschter Nebeneffekt einstellen: von mehreren Auftraggebern sind fast fertige EA-Programme in die EA-Zone abgelegt worden. Im Extremfall könnte die EA-Zone mit solchen unfertigen EA-Programmen fast ganz überdeckt sein, so daß ein weiteres EA-Programm nicht mehr hineinpaßt. Bei der Ablage dieses neuen EA-Programmes würde der KBA dann auf Freigabe von EA-Bereichen in der EA-Zone warten, jedoch vergeblich, weil die von den anderen nicht fertigen EA-Programmen belegten Plätze erst nach erfolgtem EA-Verkehr freigegeben werden können.

Diese Situation läßt sich am einfachsten dadurch vermeiden, daß der KBA unter EG-Sperre läuft. Nun sollte die EG-Sperre aber nicht ohne zwingenden Grund gesetzt werden (dies kann zu einer Verlangsamung des EA-Verkehrs führen). Der KBE aber kann für den KBA eine EG-Sperre dadurch simulieren, daß er einen durch EG unterbrochenen KBA nach der EG-Behandlung fortsetzt.

Damit ergibt sich folgender RK-Fluß durch den KB:



3. Spezielle Probleme des KB

Es werden nun einige spezielle Probleme des KB betrachtet. Der KB soll, um es noch einmal zu betonen, gleichzeitig auf beiden RK laufen können; er soll aber auch auf einem RK laufen können. Diese letzte Forderung ist nicht notwendig dadurch erfüllt, daß die erste erfüllt ist.

3.1 Auftragswarteschlangen

Eine der Aufgaben des KB besteht in der Verwaltung der kanalspezifischen Auftragswarteschlangen. Das bedeutet insbesondere, daß ein neuer EA-Auftrag in eine solche Warteschlange W eingeordnet wird. Die Dauer der Abarbeitung von W ist abhängig von der Stelle in W , an die der neue EA-Auftrag gesetzt wird. Dies führt auf den Begriff der durchsatzmäßig optimalen Einordnung. Diese liegt dann vor, wenn gilt:

$$(E) \forall \pi: t_0^{\pi(W)} \geq t_0^W$$

mit t_0^W : Dauer der weiteren Abarbeitung von W
zum Zeitpunkt t_0 ,

$\pi(W)$: eine notwendige Reihenfolgen nicht störende Umordnung von W .

Für den EA-Verkehr ist es von Vorteil, wenn der KB die Relation (E) für alle Kanäle zu jedem Zeitpunkt erfüllen kann. Es soll nun untersucht werden, wie (E) erfüllt

werden kann, um damit eventuell Aussagen über die Rückwirkungen auf das übrige System zu gewinnen. Dazu werde die Warteschlange W mit den EA-Programmen EAP_i betrachtet. EAP_i bestehe aus der Befehlskette $B_i^1, \dots, B_i^{n_i}, \text{STOP}$. Dabei ist B_i^1 ein spezieller EA-Befehl, der Steuerinformation für das EA-Werk, in der bei Standardkanälen insbesondere der Unterkanal ausgewählt wird, und die PE überträgt.

Beim Übergang von EAP_j nach EAP_{j+1} erfolgt wegen des STOP-Befehls mit einem Stop-EG eine Unterbrechung des EA-Verkehrs. Bei der Behandlung dieses Stop-EG startet der KB den nächsten Auftrag EAP_{j+1} mit dem Befehl START (Kanal k). Die verlorene Kanalzeit zwischen dem Stop-EG und dem neuen Startbefehl läßt sich dadurch vermeiden, daß beim Eintrag der Befehlskette von EAP_{j+1} der Verweis in $B_j^{n_j}$ auf den STOP-Befehl ersetzt wird durch einen Verweis auf B_{j+1}^1 . Das EA-Werk durchläuft dann beim Übergang von EAP_j nach EAP_{j+1} eine normale Stopphase und danach sofort und ohne Anstoß durch den Startbefehl eine neue Startphase, in der erforderlichenfalls der Unterkanal gewechselt wird.

Die dadurch eingesparte tote Kanalzeit ist von der Größenordnung einer EG-Behandlung. Diese Zeitspanne wird für das Betriebssystem der in 1.1 genannten Anlage maximal etwa 0.3ms betragen und ist damit klein gegenüber der Zeit, die bei bestimmten Geräten durch Stop- und Anlaufzeit verloren gehen. Diese Zeiten betragen bei einem Magnetbandspeicher je etwa 4ms. Wenden sich die Aufträge EAP_j und EAP_{j+1} an verschiedene Magnetbänder, so ergeben sich 4ms tote Kanalzeit, jedoch 8ms, wenn sie sich an dasselbe Magnetband wenden. Stop- und Anlaufzeiten kann man vermeiden, wenn EAP_{j+1} dieselbe Steuerinformation verlangt wie EAP_j . Dann kann man nämlich den Verweis in $B_j^{n_j}$ auf den STOP-Befehl durch einen Verweis auf B_{j+1}^2 ersetzen und dadurch gewissermaßen EAP_j um EAP_{j+1} verlängern.

Die beiden geschilderten Anhängungsarten - Verweis auf den ersten oder zweiten Befehl des Folgeauftrages - führen auf folgende spezielle Umordnung \mathbb{W}_0 der Warteschlange:

Ein im EA-Verkehr befindliches Gerät bleibt im EA-Verkehr, solange weitere Aufträge für dieses Gerät an bereits vorhandene angehängt werden können. Die Anhängung geschieht unter Aussparung überflüssiger Übertragungen von Steuerinformation.

Nach einem Stop-EG wird von den vorhandenen Aufträgen eines Kanals der erste von denen gestartet, die sich an das Gerät höchster Priorität wenden.

Hierbei sind stillschweigend Geräteprioritäten vorausgesetzt worden, die sich natürlich auf Unterkanalprioritäten zurückführen lassen. Diese - man kann sie sich auch veränderlich vorstellen - sollen lediglich dazu dienen, nach einem Stop das nächste EA-Programm zu starten. Von der Möglichkeit des selbständigen Unterkanalwechsel ist dabei kein Gebrauch gemacht worden, weil der dadurch erkaufte Vorteil den dafür erforderlichen Verwaltungsaufwand nicht zu rechtfertigen scheint.

Die Regelung \mathbb{W}_0 läßt folgende nachteilige Situation zu: für ein unwichtiges im EA-Verkehr befindliches Gerät werden laufend Aufträge nachgeliefert, so daß Aufträge für ein wichtiges Gerät nicht zum Zuge kommen. Doch wann ist ein Gerät wichtig? Ist dies eine Geräteeigenschaft (dann würde man den geschilderten Nachteil dadurch vermeiden können, daß man an einen Kanal nur Geräte etwa gleichen "Wichtigkeitsgrades" installiert), oder ist dies eine von wechselnden Auftragsgebern vererbte Eigenschaft (das würde dann zu Auftragsprioritäten führen, die vom KB entsprechend zu verarbeiten wären)?

Gleichwohl stellt \mathbb{W}_0 eine durchsatzmäßig optimale Einordnung dar. Zur weiteren Beurteilung sollten erst Erfahrungen gesammelt werden.

π_0 gestattet auch folgende Sicht: die kanalspezifische Warteschlange W zerfällt in unterkanalspezifische Warteschlangen W_u . Die Bearbeitung von W_u wird beendet, wenn sie entweder leer geworden ist oder wenn eine Anhängung an einen im EA-Verkehr befindlichen Auftrag zu spät durchgeführt wurde. Nach Beendigung einer Bearbeitung einer Warteschlange wird die Bearbeitung der Warteschlange des Unterkanals höchster Priorität aufgenommen.

Die zulässige maximale Länge von W_u wird man so groß wie nötig wählen (um Aufträge anhängen zu können) und so klein wie möglich (um Engpaßsituationen bei Belegungen in der EA-Zone zu vermeiden). Davon hängt natürlich auch die zulässige Länge eines EA-Programms ab. Je kleiner das Verhältnis von Aufbereitungsdauer eines Auftrags zur Übertragungsdauer ist, umso kleiner kann die zulässige Länge von W_u gewählt werden. Eine Schätzung dieses Verhältnisses für das Betriebssystem ^{der} in 1.1 genannten Anlage erlaubt die minimal zulässige Länge von 2.

An die Hardware ergibt sich folgende Forderung: Der Übergang von EAP_j zum angehängten EAP_{j+1} stellt das Erreichen einer wohldefinierten Stelle bei der Abarbeitung eines EA-Programmes dar und sollte eindeutig durch einen speziellen EG ohne unerwünschte Nebenwirkungen genau an dieser Stelle gemeldet werden können.

3.2 EG-Zuordnung

Nach der in 1.2 vorausgesetzten Hardware lassen sich Anforderungen folgender Art realisieren:

"Von jetzt an sind alle EG des Typs t vom Kanal k dem RKr zuzuordnen!"

Hierbei darf nicht die Möglichkeit übersehen werden, daß die Behandlung eines EG vom Typ t_1 auf dem RKr später beendet sein kann als die Behandlung eines späteren EG von demselben Kanal vom Typ t_2 auf RKr'. Es ist mindestens

ein erheblicher Verwaltungsaufwand erforderlich, um dann noch Inkonsistenzen zu vermeiden. Deshalb soll die obige Aufforderung beschränkt werden auf folgende dynamische Veränderung der EG-Zuordnung:

"Von jetzt an sind alle EG vom Kanal k dem RKr zuzuordnen!"

(dafür werde auch geschrieben: $EG(k) \rightarrow RKr$)

Es sollen einige darauf basierende wesentlich verschiedene EG-Zuordnungs-Strategien betrachtet werden:

- (a) Die EG sind alle ein und demselben RK zugeordnet.
- (b) Die Kanäle werden in zwei Gruppen KG_i eingeteilt. Die EG von KG_i sind RK_i zugeordnet.
 - (b.1) Die Einteilung ist starr.
 - (b.2) Die Einteilung ist dynamisch veränderlich.
- (c) Der KBE ordnet zu Beginn seines Laufes alle EG dem RK zu, auf dem er nicht läuft.

Zur Beurteilung dieser Möglichkeiten muß insbesondere die Forderung nach der Effizienz des EA-Verkehrs betrachtet werden, die durch den EG-Auflauf gemindert wird. Der EG-Auflauf ist nämlich verbunden mit einem Abbruch des EA-Verkehr über einen Kanal und ergibt sich dann, wenn auf einem Kanal während des Anstehens eines bestimmten EG, d.h. wenn der EG-Wunsch noch nicht abgenommen wurde, derselbe EG-Wunsch noch einmal entsteht. Die Wahrscheinlichkeit des EG-Auflaufs wird i.a. sehr gering sein. Sie wächst trivialerweise mit der Dauer, über die Unterbrechungssperren gesetzt werden. Sie wächst ferner dann, wenn Geräte, die in einer den EG-Auflauf leicht ermöglichenden Weise betrieben werden, an Kanäle niedriger Hardware-Priorität angeschlossen werden (entsteht nämlich ein EG-Wunsch auf dem Kanal k_1 , während schon ein EG vom Kanal k_2 ansteht, so führt der zuerst zum EG, der auf dem Kanal höherer Hardware-Priorität ansteht).

Die Methode (a) erzeugt eine Asymmetrie zwischen den beiden bis auf den Vorrang beim gleichzeitigen Speicherzu-

griff gleichberechtigten RK. Diese könnte von der RK-Ver-
gabe des Systems dadurch ausgenutzt werden, daß sie wich-
tige Aufgaben prinzipiell dem RK überträgt, dem die EG
nicht zugeordnet sind. Der Umstand, daß der KBE nur auf
einem bestimmten RK läuft, darf nicht zu dem Schluß ver-
führen, daß deshalb keine Synchronisierungsmaßnahmen be-
nötigt werden. Die weiterhin notwendigen Synchronisierungs-
maßnahmen zwischen KBE und KBA sind im Prinzip dieselben,
die zwischen KBE und KBE notwendig wären. Ein Nachteil
dieser Methode ist die gegenüber einer Verteilung der EG
auf die beiden RK erhöhte Wahrscheinlichkeit des EG-Auf-
laufs.

Der Fall (b.1) ist ein Sonderfall von (b.2). Er wäre
geeignet, die wichtigsten Kanäle zu der kleineren der
beiden Gruppen zusammenzufassen, um so den EG von diesen
Kanälen eine möglichst schnelle Behandlung zu sichern.
So könnte etwa der Kanal, über den der Satellitenrech-
ner angeschlossen ist, zu dieser Gruppe gehören.

Beim Fall (b.2) können die beiden Gruppen KG_i vom KBE
dynamisch mit dem Ziel verändert werden, die EG möglichst
gleichmäßig auf die beiden RK zu verteilen. Die Symmetrie
der RK bezüglich der Unterbrechung durch EG ergibt sich
dann aber nur bei Betrachtung eines längeren Zeitraumes.
Temporär können durchaus asymmetrische Verhältnisse ein-
treten, wenn Ballungen von EG für einen RK entstehen,
wobei die Gefahr des EG-Auflaufs gegeben ist.

Nach der Methode (c) werden die EG abwechselnd auf die
beiden RK verteilt, und damit bleibt jedenfalls bezüg-
lich der Unterbrechung durch EG die Symmetrie der beiden
RK erhalten. Die alternierende Verteilung geschieht je-
doch nicht notwendig in der Reihenfolge der Entstehung
der entsprechenden EG-Wünsche (wegen den oben erwähnten
Hardware-Prioritäten der Kanäle). Auf diese Weise wird

eine zufällige Ballung von EG am schnellsten abgearbeitet und damit zugleich die Wahrscheinlichkeit des EG-Auflaufs am geringsten gehalten.

Diese beiden Gründe - Erhaltung der RK-Symmetrie und schnellste Reaktion auch auf mehrere EG-Wünsche - lassen uns der Methode (c) den Vorzug geben. Diese soll daher noch etwas näher betrachtet werden.

Vor Beginn der eigentlichen Behandlung eines EG von Kanal k führt der unter EG-Sperre auf RKr laufende KBE folgende Tätigkeiten durch:

$$\forall k: EG(k) \rightarrow RKr'.$$

Es ist nun möglich, daß die Behandlung des EG eine Verzögerung erfährt und über einen nun von RKr' angenommenen weiteren EG von Kanal k hinausreicht. EG von einem Gerät werden also nicht notwendig in der Reihenfolge abgehandelt, in der sie auftreten: Situationen, in denen sich der KB nicht mehr zurechtfinden kann (wenn etwa nach der Behandlung des Stop-EG des ersten Auftrages ein EG gemeldet wird, der eine bestimmte Stelle der Abarbeitung des zum ersten Auftrag gehörigen EA-Programms anzeigt). Gegen solche Situationen kann sich der KB durch eine einfache Synchronisierung schützen, etwa von der Art, daß mit der Behandlung eines EG von Kanal k auf RKr erst dann begonnen wird, wenn eine auf RKr' eventuell laufende Behandlung eines EG von demselben Kanal beendet ist. Am einfachsten und ohne Leerlauf läßt sich eine solche Synchronisierung dann durchführen, wenn die Hardware die Möglichkeit vorsieht, die EG einem nicht vorhandenen RK, der mit RK_{∞} bezeichnet werde, zuordnen zu können. Dann führt ein auf RKr laufender KBE außer der eigentlichen Behandlung eines EG vom Kanal k' folgende Tätigkeiten durch:

$$EG(k') \rightarrow RK_{\infty};$$

$$(U) \quad \forall k: k \neq k': EG(k) \rightarrow RKr';$$

eigentliche EG-Behandlung;

$$EG(k') \rightarrow RKr_{\text{aktuell}};$$

dabei bezeichne r_{aktuell} den RK, dem die EG aktuell zuge -

ordnet sind. Die Umordnung $EG(k') \rightarrow RK_{\infty}$ werde kanalselektive EG-Sperre genannt.

Nun ist es denkbar, daß auf $RK_{r'}$ bereits eine EG-Behandlung läuft, deren Umordnung (U) zu dem gerade betrachteten EG auf RK_r führte. Dabei wurde bereits eine kanalselektive EG-Sperre gesetzt, die bei der neuerlichen Umordnung zu berücksichtigen ist. An Stelle von (U) muß daher im allgemeinen Fall die Umordnung

$$(U') \quad \forall k: k \neq k' \wedge \neg EG(k) \rightarrow RK_{\infty} : EG(k) \rightarrow RK_{r'}$$

durchgeführt werden.

Diese EG-Zuordnung zeigt, daß die Forderung, das System solle auch auf nur einem RK laufen können, nicht trivial ist. Um sie zu erfüllen, müssen bei nur einem laufenden RK nach erfolgter EG-Behandlung alle EG wieder dem laufenden RK zugeordnet werden.

3.3 Multiplex an Standardkanälen durch den KB

Die in 1.2 vorausgesetzte Hardware gestattet keinen Multiplex der Übertragung für die an den Unterkanälen eines Kanals angeschlossenen EA-Geräte. Spezielle EA-Geräte, nämlich bestimmte Schnelldrucker, können jedoch in einer Weise betrieben werden, die als softwaremäßiger Multiplex bezeichnet werden könnte. Die dafür wesentlichen Eigenschaften des Schnelldruckers sollen nun kurz erläutert werden:

Der Schnelldrucker kennt eine Druckphase: diese beginnt mit dem Ausdrucken einer Zeile, wenn deren Übertragung in den Druckpuffer des Schnelldruckers beendet ist, und endet, wenn die Zeile fertig ausgedruckt ist. Die Übertragung einer Zeile dauert etwa $1/50$ der Druckphase (nämlich $d \approx 1\text{ms}$). Das Ende einer Druckphase meldet der Drucker dem EA-Werk, das darauf entweder die Übertragung einer weiteren Zeile durchführt, wenn im EA-Programm weitere

Befehle dafür vorhanden sind, oder einen Anruf meldet, wenn nach Übertragung der Zeile das EA-Programm mit einem STOP-Befehl endete. Im ersten Fall bleibt der Kanal während der Druckphase belegt, im zweiten Fall wird der Kanal mit Beginn der Druckphase frei. (Ein Kanal heiße genau dann belegt, wenn der Befehl `START(Kanal k)` abgelehnt würde, und frei, wenn er nicht belegt ist.)

Läßt man n Zeilen mit einem EA-Programm ausdrucken, so ist der Kanal etwa für die Dauer von $n \cdot (50+1) \cdot d$ belegt, davon jedoch nur $n \cdot d$, also etwa 2%, durch die Übertragung. Man kann den Kanal durch folgende Vorgehensweise besser ausnutzen: Von dem vorgefertigten EA-Programm eines Druckauftrages übernimmt der KB jeweils nur die Befehle, die für den Ausdruck der nächsten Zeile notwendig sind. Diese legt er als EA-Befehle, ergänzt um einen STOP-Befehl, in die EA-Zone ab. Überträgt das so entstandene EA-Programm die erste Zeile des Druckauftrages, so schleust der KB es wie ein normales EA-Programm in den EA-Verkehr ein, überträgt es jedoch eine andere als die erste Zeile, so wird es zur Einschleusung erst dann freigegeben, nachdem der vom KB intern verarbeitete Anruf von dem entsprechenden Drucker erfolgt ist, der die Druckphase der vorherigen Zeile abschließt. Zur Behandlung des Stop-EG zu Beginn dieser Druckphase gehört der Aufbau des EA-Programms für die nächste Zeile.

Der zu vernachlässigende Nachteil dieser Methode besteht darin, daß die dabei erreichbare Druckergeschwindigkeit gegenüber der maximalen pro Zeile um die Dauer einer Anrufsbehandlung verringert wird (und das macht bei pessemistischer Schätzung etwa 0.4% aus). Der Vorteil liegt auf der Hand: an einem Kanal können 4 Schnelldrucker simultan betrieben werden, wobei die Belegung in der EA-Zone minimal ist.

Aber auch dann ist der Kanal nur zu etwa 8% ausgelastet. Es kann andere Kombinationen von Geräten mit einem oder mehreren Schnelldruckern an einem Kanal geben, die eine bessere Auslastung ergeben. Günstig wäre etwa die Kombination eines Schnelldruckers mit solchen Geräten, für die die EA-Aufträge den Kanal für etwas weniger als die Zeit einer Druckphase belegen, wobei der Schnelldrucker die höchste Priorität am Kanal haben sollte.

Es wäre wünschenswert, wenn noch andere EA-Geräte Eigenschaften hätten, die einen softwaremäßigen Multiplex gestatten. Für die langsameren Ausgabegeräte wie Kartenstanzer, Streifenstanzer könnte man sich derartiges vorstellen.

3.4 Die Verwaltung der EA-Zone

Nach den Überlegungen zu den Aufgaben des KB ist die Verwaltung der EA-Zone eine der wesentlichen Tätigkeiten des KB. Diese besteht aus der Ablage von EA-Programmen auf freie Plätze der EA-Zone und aus der Freigabe von belegten Plätzen, wenn ein EA-Vorgang beendet ist. Für die folgenden Betrachtungen werde davon ausgegangen, daß der KB für jedes Gerät G_j höchstens zwei EA-Aufträge in Bearbeitung habe (vergl. 3.1).

Die Methode, die EA-Zone in gerätespezifische Bereiche B_j aufzuteilen, die $l(B_j)$ EA-Befehle aufnehmen können, und festzusetzen, daß jedes EA-Programm für G_j nicht mehr als $l(B_j)/2$ EA-Befehle umfassen darf, kann letzten Endes nur aus der Erfahrung heraus beurteilt werden. Es ist möglich, daß diese Methode die EA-Zone nicht genügend ausnutzt, nämlich dann nicht, wenn die Längen der EA-Programme innerhalb der zulässigen Grenzen schwanken. Der dann von Aufträgen nicht belegte Platz in der EA-Zone könnte temporär anderen Aufträgen zugute kommen.

Hierbei darf nicht vergessen werden, daß die Übertragung einer bestimmten Informationsmenge umso weniger Verwaltungsaufwand erfordert, je kleiner die Anzahl der dafür erforderlichen EA-Aufträge ist, d.h. je längere EA-Programme zugelassen sind.

Eine andere flexiblere Methode basiert auf dem Zweiadreßmechanismus der vorausgesetzten EA-Befehle, der die Befehle eines EA-Programms zu einer Kette K verknüpft. Auf dieselbe Weise sollen die nicht belegten Plätze der EA-Zone zu einer Kette K' zusammengefaßt werden. Nach Beendigung eines EA-Vorgangs wird die zugehörige Kette K freigegeben durch Anhängen an die Kette K' :

$K' := K'$ verknüpft mit K .

Eine Kette K wird sukzessiv aufgebaut durch Ablage des nächsten EA-Befehls an den Platz des ersten Gliedes von K' :

$K' := K'$ verkürzt um erstes Glied K'_{alt}

$K := K$ verlängert um weiteren in K'_{alt} abgelegten Befehl.

Veränderungen an der Kette K' sind natürlich Tätigkeiten, die nicht gleichzeitig auf beiden RK durchgeführt werden dürfen.

Eine solche Verwaltung der EA-Zone gestattet folgende Regelung:

Ein EA-Programm für das Gerät G_j darf nicht mehr als s_j EA-Befehle umfassen. Dabei sei zugelassen, daß $2 \leq s_j$ die Länge der EA-Zone um einen bestimmten Prozentsatz p überschreite.

Es wird also bewußt in Kauf genommen, daß die EA-Zone dann nicht ausreicht, wenn für alle Geräte Aufträge mit EA-Programmen maximaler Länge aufbereitet werden sollen. Dann muß eben auf das Ende eines EA-Vorganges und die damit verbundene Freigabe einer Kette K gewartet werden. Aus der Erfahrung heraus können p und damit s_j so gewählt werden, daß diese extreme Situation eine Ausnahme

bildet. Dann hat diese Methode gegenüber der ersteren den Vorteil einer unabhängig von den Längen der EA-Programme vollständigen Ausnutzung der verschnitttfrei belegten EA-Zone.