

INTERNSCHRIFT Nr. 11

THEMA:

Bericht über das Compatible Time Sharing System des MIT
nach den CTSS Technical Notes von J.H. Saltzer MIT-MAC-TR16

VERFASSEN:
PAUL / STIEGLER

DATUM:
WS 68/69

FORM DER ABFASSUNG

ENTWURF
☒ AUSARBEITUNG
ENDFORM

SACHLICHE VERBINDLICHKEIT

☒ ALLGEMEINE INFORMATION
DISKUSSIONSGRUNDLAGE
ERARBEITETER VORSCHLAG
VERBINDLICHE MITTEILUNG
VERALTET

ÄNDERUNGSZUSTAND

BEZUG AUF BISHERIGE INTERNSCHRIFTEN

Vorkenntnisse aus:
Erweiterung von:
Ersatz für:

BEZUG AUF KÜNFTIGE INTERNSCHRIFTEN

Vorkenntnisse zu:
Erweiterung in:
Ersetzt durch:

ANDERWEITIGE LITERATUR

WS 1968/69

B. Bericht über das Compatible Time Sharing System
des MIT nach den CTSS Technical Notes von J. H.
Saltzer, MIT - MAC - TL 16.

(Paul / Stiegler)

I. Einleitung

Das CTSS wurde vom Massachusetts Institute of Technology im Rahmen des Forschungsprogramms "MAC" entwickelt. Eine schnell rechnende CPU soll mehreren Benutzern gleichzeitig und unabhängig voneinander zur Verfügung stehen.

Vorteile eines solchen Systems sind unter anderem:

- 1) Die CPU kann besser ausgelastet werden, da die Totzeiten der CPU, soweit sie vom Warten auf die Beendigung von I/O - Vorgängen herrühren, verkürzt werden.
- 2) Der Komfort einer großen Maschine wird durch Aufteilung der Rechenkapazität auch solchen Benutzern zugänglich gemacht, die alleine keine große Maschine rentabel betreiben können.

Jeder Benutzer bekam eine eigene Kernschreiberkonsole (es waren 30 bis 40 vorgesehen), über die er jederzeit einen bestimmten Anteil an der Maschinenkapazität erhalten kann. Er soll quasi on - line rechnen können und im Durchschnitt nicht länger auf die Erledigung seines Programms warten, als dies bei einer kleineren Maschine, die nur ihm zur Verfügung stünde, der Fall wäre.

Die Forderungen an die Konzeption lauteten damit:

Eingabe soll jederzeit möglich sein, Wartezeiten der CPU sollen nicht entstehen, solange noch rechenbereite Programme vorhanden sind, länger Programme werden zugunsten anderer unterbrochen, die Programme werden, zeitverschränkt mit anderen, sequentiell abgearbeitet.

Für die Organisation der Aufteilung und der Zeitverschränkung wurde ein Supervisor-Programm geschaffen, das sich die CPU

mit den Benutzern teilt. In der Zwischenzeit hat sich herausgestellt, daß CTSS die mit Vielfachzugriff - Systemen verbundenen Probleme nicht in jeder Hinsicht zufriedenstellend löst. Insbesondere ist das Verhältnis der produktiven zur unproduktiven CPU - Zeit bei intensivem Konsolbetrieb erheblich zu klein.

II Hardware-Voraussetzungen

Die verwendete IBM 7094 hat folgende speziellen Eigenschaften:

1) Sie besitzt einen Intervallwecker, der periodisch alle 0,2 sec den Prozess unterbricht und die CPU-Kontrolle dem Supervisor-Programm übergibt.

2) Es gibt einen hardwaremäßigen Speicherschutz. Speicherschutzalarm wird bei unzulässiger Speicherung gegeben, wodurch das laufende Programm unterbrochen wird.

3) Es gibt wenigstens 5 Datenkanäle:

A) Drucker, Stanzer, Kartenleser, Magnetkürder und Chronometer.

B) Bandgeräte.

C) Plattenspeicher und Trommelspeicher.

D) Direkte Datenein- und -ausgabe für spezielle Experimente.

E) Ein Kanal zum IBM 7750 Satellitenrechner als Multiplexsteuer-einrichtung für ca. 40 Fernschreiber.

Zusätzlich besitzt der Projekt-LAC-Rechner die beiden folgenden Kanäle:

F) Zweiter Plattenspeicher und zweite Trommel

G) Hochgeschwindigkeitstrommel.

4) Es gibt zwei Kernspeichermodule zu 32 K 36-bit-Worten.

Der Satellitenrechner 7750 kann einen Datenkanaleingriff bewirken und die CPU Kontrolle dem Supervisor übergeben.

Im Übrigen wird die Fähigkeit des Satellitenrechners zu rechnen nicht ausgenutzt. Um den Supervisor zu vereinfachen, soll alles zentral von der 7094 erledigt werden.

Aus dem selben Grund ist auch der Kernspeicher unterteilt, in einen Teil A, der ausschließlich dem System zur Verfügung steht,

und einem Teil B für die Benutzer.

Im Teil A ist Speicherplatz zur Leitung von Registern, der dem Weckerprogramm untersteht, der Speicher ist so dimensioniert, daß dem Benutzer durch Files erlaubt sein soll, ein Programm auch noch nach Wochen wiederzufinden. Eine Verweistabelle (Master file directory) enthält die Namen aller potentiellen Benutzer und die Information, wo ein Benutzer seinen File stehen hat.

Jeder Benutzer, selbst der Supervisor, steht in der Verweistabelle. Auf Platten sind Modulpakete, die über Sprungvektoren zu erreichen sind, gespeichert. Wenn der Supervisor arbeitet, stehen die entsprechenden Moduln im Kernspeicher. Auf diese Weise ist es möglich, mit Supervisormoduln zu experimentieren, ohne daß der Ausgangszustand der Moduln verloren geht, weil letztere im Plattenspeicher unverändert erhalten bleiben.

III Aufruf der Supervisorprogramme

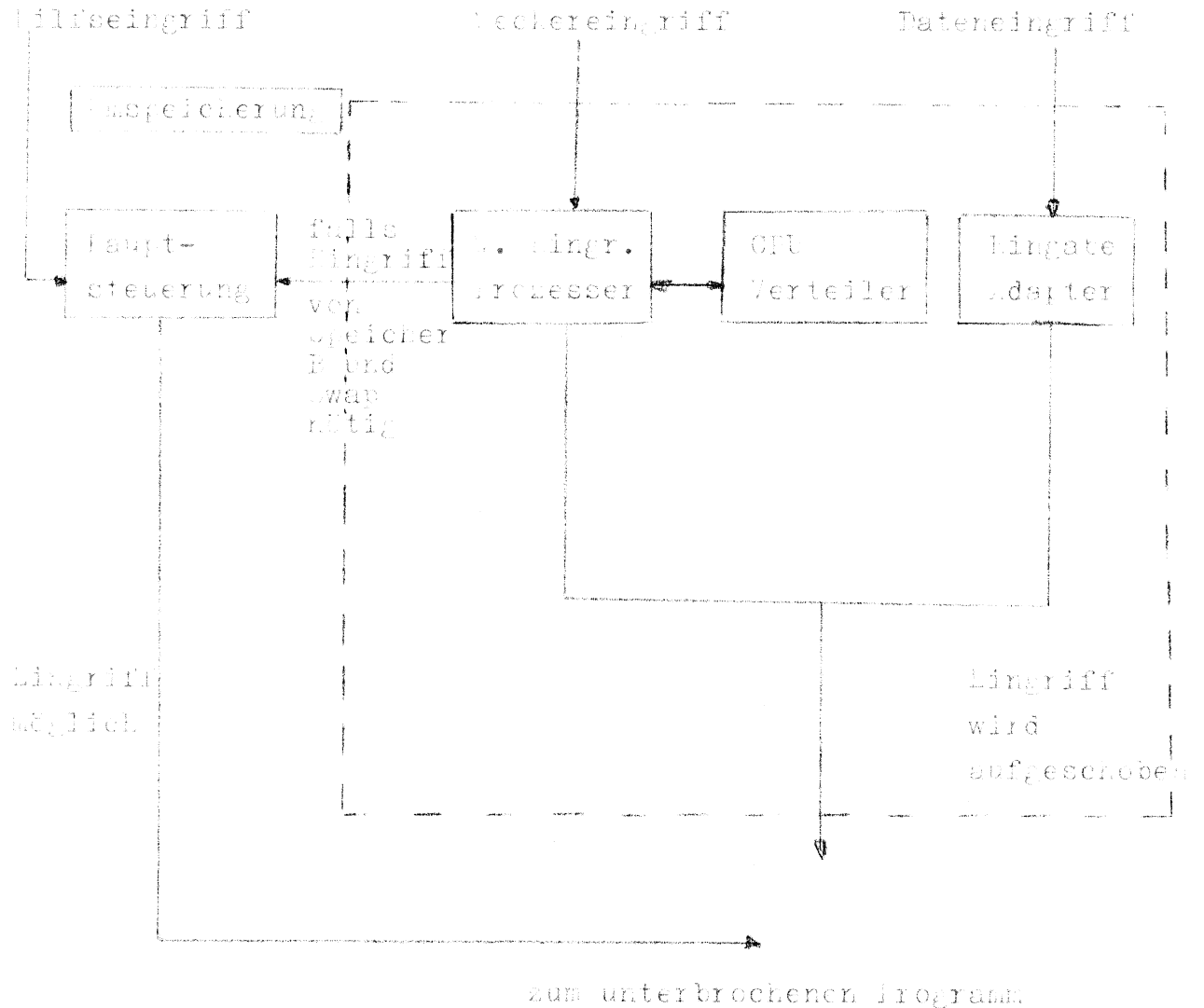
Drei Ereignisse können die CPU Kontrolle dem Supervisor übertragen, wobei die ersten beiden an beliebiger Stelle im laufenden Programm eintreten können:

- 1) Wenn irgendein Benutzer in einem Fernschreiben einen Buchstaben (6bits) schreibt, verursacht er einen Datenkanaleingriff. Das Zeichen wird übersetzt und im Eingabepuffer mit der Benutzernummer zusammen gespeichert, dann geht die Kontrolle der CPU an das unterbrochene Programm zurück.
- 2) Nach einer Weckerperiode, i.A. 0,2 sec, gibt es einen Weckereingriff. Der Supervisor prüft, ob das unterbrochene Programm noch weiterrechnen darf, und verursacht, falls ein anderes an die Reihe kommen soll, die nötigen Umspeicherungen, den sogenannten "swap".
- 3) Wenn das Programm ein Unterprogramm des Supervisors aufruft, etwa zur Datenausgabe.

Ein Prozess hat keine Möglichkeit einen Eingriff zu kontrollieren, außer ihn aufschieben zu lassen. Ist nämlich der laufende Prozess ein Supervisorprogramm, so kann ein Eingriff unzweckmäßig sein, wenn der Supervisor durch einen Wecker- oder

Dateneingriff, aufgerufen worden ist.

Eine Übersicht darüber gibt folgendes Fluiddiagramm der Eingriffswirkungen:



IV Funktionsmäßige Unterteilung der Benutzer

Um die Rechenzeitanteile an die Benutzer sinnvoll machen zu können, werden sie, je nach der Rolle, die sie gerade spielen, in 6 Zustandsklassen unterteilt.

Ein Benutzer ist in Zustand

- 6) tot, wenn er zwar als Benutzer eingetragen ist, aber von ihm ein Programm nicht mehr oder noch nicht vorhanden ist.

- 1) schlafend, wenn der Speichersubzug eines rechenfähigen Programms vorhanden ist, das der Benutzer momentan nicht rechnet läßt.
- 2) arbeitend, wenn das Programm in der Warteschlange auf Zuteilung von Rechenzeit wartet oder rechnet.
- 3) wartend auf Kommandoausführung, wenn der Benutzer auf Grund von eingegebenen Befehlen von "0)" oder von "1)" in "2)" übergehen will, oder wenn der Prozess von sich aus nach dem "nächsten Kommando" fragt.

Außerdem kann jedes Programm zwangsweise in diesen Zustand gesetzt werden.

- 4) Auf Eingabe wartend, wenn ein rechnendes Programm auf Eingabe in den Hauptspeicher von Benutzerkonsolen wartet. (Nicht, wenn es vom Plattenspeicher liest, da dann der Informationsfluß so groß ist, daß ein Programmaustausch nicht lohnt).
- 5) Auf Ausgabe wartend, wenn ein rechnendes Programm eine Ausgabe veranlaßt und der Ausgabebuffer voll ist.

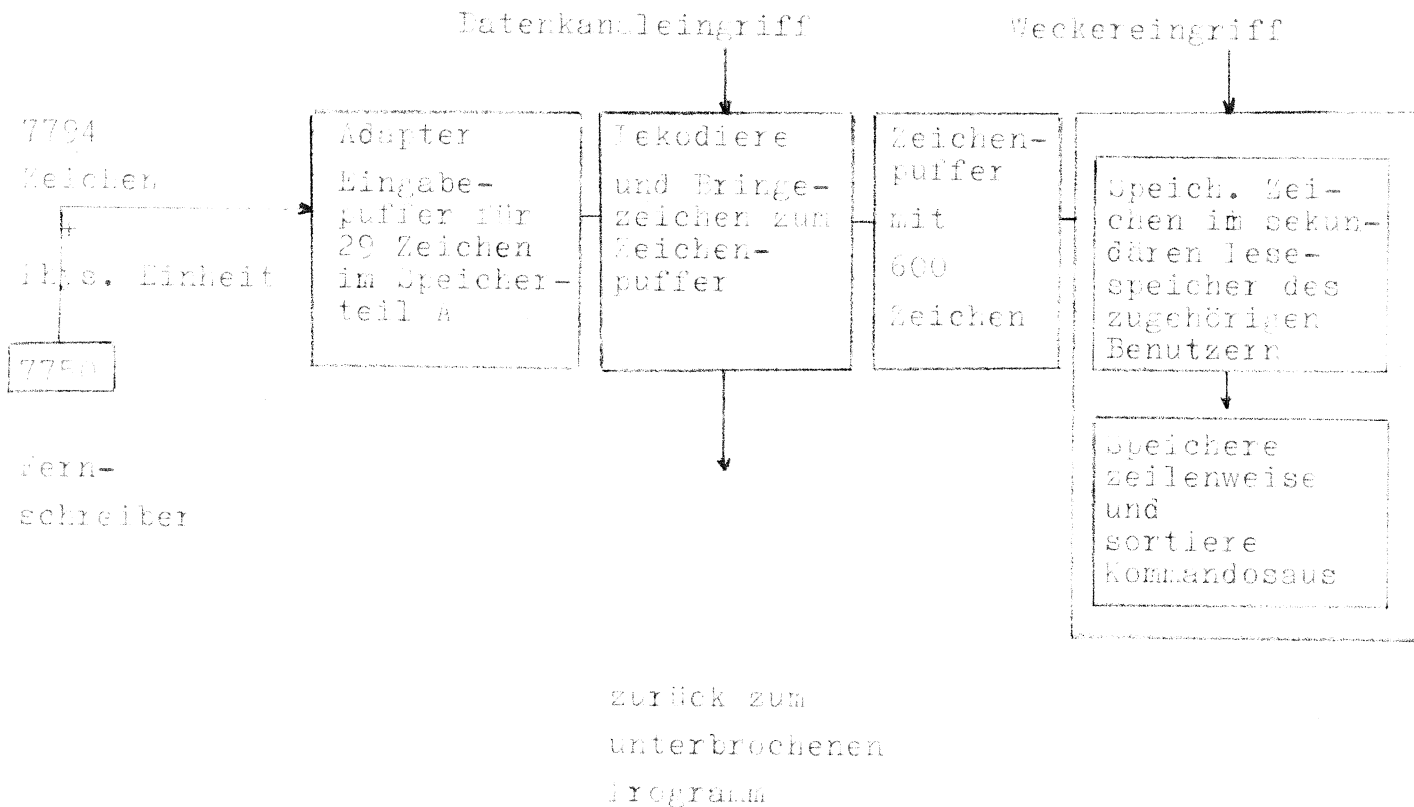
Zustandswechsel können also von der Konsole, vom Programm und vom System herrühren.

V Die Lin und Ausgabe

a) Vorbemerkung

Zwei Dinge sind hier verwirklicht worden: Eingabe ist jederzeit möglich, die Ausgabe behindert nicht den CPU-Betrieb.

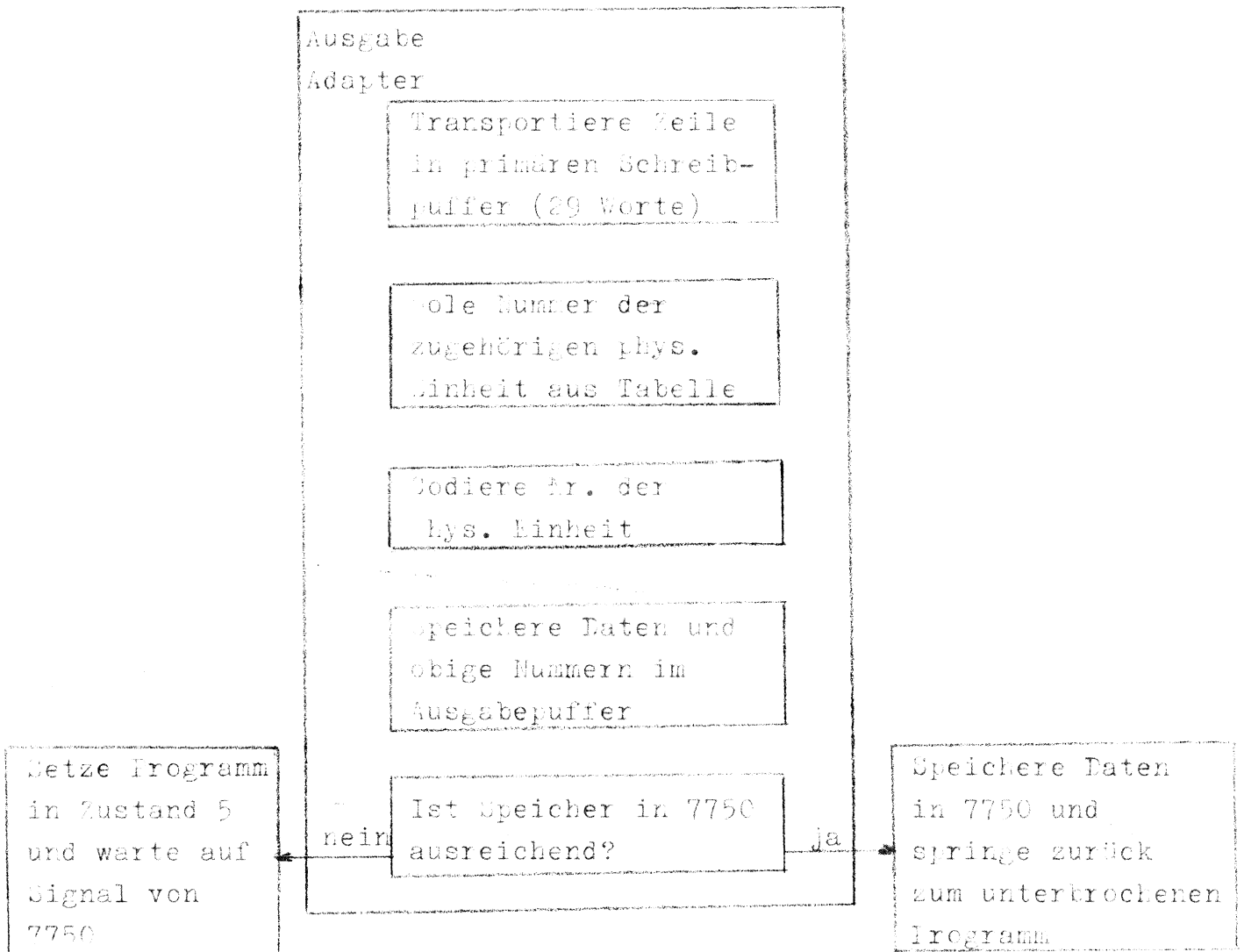
Das erste wurde durch gestaffelte und genügend dimensionierte Puffer erreicht, das zweite dadurch, daß Programme, die ihre Ausgabe nicht direkt der 7750 weitergeben können, von Zustand 2 in Zustand 5 wechseln und den Rechenbetrieb so nicht behindern. Allerdings müssen sie von Zustand 5 erst wieder in 2 zurück, um die Ausgabe erneut versuchen zu können.

Flussdiagramm der Eingabe

Der Adapter-Eingabepuffer hat die Zeichen nur solange aufzubewahren, bis der Datenkanaleingriff wirksam wird, um die Zeichen zu dekodieren und in einem größeren Zeichenpuffer zu transportieren. Dieser wird alle 0,2 msec im Rahmen des Weckerprogramms entleert und den Benutzern, deren Programme auf Band liegen können, zugeteilt.

Flußdiagramm der Ausgabe

Programm ruft Unter-
programm des Supervisors



(10000Zeichen)

Die Zeichenübergabe geschieht jeweils zeilenweise, wobei unvollständige Zeilen mit "Leerstellen" aufgefüllt werden. Der Überlauf des 7750 Speichers, der durch keine Dimensionierung abgefangen werden kann, wird durch den Zustandswechsel gelöst.

VI Rechenzeituteilunga) Anfangseinstufung in Prioritätsebenen

Es gibt eine feste Zahl von Prioritätsebenen in die jeder Benutzer, der sich im Zustand 2 oder 3 befindet eingeordnet wird. Anfänglich wird dabei ein Benutzer in Abhängigkeit von der statischen Länge des Programms so eingestuft, daß das Verhältnis des Zeitquantums zur Wartezeit zunächst für alle Benutzer ungefähr gleich ist. Die Anfangseinstufung ist leicht zu ändern und dadurch wird eine flexible Prioritätspolitik ermöglicht. Ein Programm wird dann am Ende seiner Prioritätsebene eingereiht.

b) Der Zeituteilungsalgorithmus

Jedem Benutzer wird ein Zeitquantum zugeteilt, das linear ansteigt mit der Zahl der Prioritätsebene, in dem sein Programm sich befindet. Benutzer, die lange warten müssen, sollen dafür längere Zeiten rechnen dürfen. Gerechnet wird in der Reihenfolge, die durch die Prioritätsebenen gegeben wird. Sind die ersten n leer, so rechnet das erste in der $(n + 1)$ -ten Warteschlange.

Es wird außer durch Datenkanaleingriffe erst durch das Weckerprogramm unterbrochen, das alle 0,2sec eine Umordnung der Warteschlangen bewirkt:

- 1) Ein Programm der i -ten Ebene, das nach der ihm zugeteilten Zeit nicht beendet ist, wird in die $(i + 1)$ -te Ebene ans Ende der Warteschlange gestellt.
- 2) Alle Programme, die mehr als 60 Sekunden auf der Ebene j gewartet haben, werden am Ende der Warteschlange der Ebene $j-1$ eingereiht.

Auf der ersten Ebene werden Umstufungen in eine höhere Ebene, auf der letzten in eine tiefere unterdrückt.

Die Warteschlangensituation kann sich unabhängig vom Weckerprogramm auf Grund neu in den Zustand 2 oder 3, Übergangener Programme geändert haben.

Die CPU-Kontrolle wird dann vom Zeituteilungsalgorithmus

den Programm zugesprochen, das bei Abschluß des Weckerprogramms die höchste Priorität hat.

c) Stapelbetrieb und gemischter Betrieb

Es besteht die Möglichkeit in gewöhnlichem Stapelbetrieb zu rechnen. Dazu ist ein Systemparameter einzustellen, der bewirkt, daß ein vorgegebener Prozentsatz für Stapelbetrieb im Verhältnis zum Gesamtbetrieb eingehalten wird.

VII Speicherverteilung

Die Speicherzuteilung der rechnenden Programme ist so einfach wie möglich gehalten. Es wird immer auf Null absolut adressiert. Soll ein Segment B auf den Speicherplatz des Segments A geladen werden und ist B kürzer als A, so bleibt das letzte Stück von A im Kernspeicher erhalten. Wenn A später wieder benötigt wird, so spart man dadurch Zeit, daß der noch erhalten gebliebene Teil von A nicht erneut geladen wird. Die Seitenlänge beträgt 2048 Worte.