

Wenn man mit "n" das linke Halbbyte und mit "k" das rechte Halbbyte jeder Stelle dieses Teiles der PRC - Zone bezeichnet, werden die Bedingungen nach folgendem Schema dargestellt:

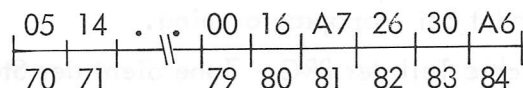
	n	k
Ein Programm "N" wurde durch einen STOP-Befehl angehalten.	Nummer des Programms	0
Ein Programm wurde durch den Schutz "k" geschlossen.	A	Nummer des Schutzes K
Ein Programm "N" wurde durch einen geschlossenen Schutz "k" angehalten	Nummer des Programms N	Nummer des Schutzes K

n = gepackte Dezimalziffer; k = hexadezimaler Zeichen

Die Zentraleinheit untersucht diese Zone durch Abtasten von rechts nach links. Die durch den gleichen Schutz "k" angehaltenen Programme werden in der Reihenfolge, in der sie blockiert wurden, wieder gestartet.

Diese Folge von Zeichen wird links durch eine Trennmarke 00 begrenzt. Die Grenze zwischen dem rechten und linken Teil dieser Zone ist jedoch fließend und zwar in Abhängigkeit von den plötzlich eintretenden Bedingungen des Zusammenspiels der Programme. Es kann sogar vorkommen, daß beide Teile dazu neigen, sich zu überschneiden, da die Gesamtlänge der Zone nicht ausreicht. In diesem Fall setzt der letzte Vorfall, der in dieser Zone ein neues Zeichen erzeugt, es an die Stelle der Trennmarke 00, und die Zentraleinheit zeigt Programmunterbrechung an. Das ist immer ein Zeichen für eine schlechte logische Konzeption in der Organisation der Simultaneität der Programme. Der Programmierer muss eingreifen, um die sich daraus ergebenden Änderungen vorzunehmen.

Beispiel:



- das Programm 5 ist in Arbeit,
- das Programm 4 ist gestartet,
- das Programm 3 wurde angehalten,
- der Schutz 6 und 7 ist geschlossen,
- die Programme 1 und 2 sind durch den Schutz 6 blockiert.

4.3.3. Die Befehle

4.3.3.1. Haltbefehle

STOP

STOP program execution

40

Der Befehl STOP, der in einem arbeitendem Programm mit der Nummer "N" steht, bewirkt den Halt des Programmablaufs und einen Programmwechsel.

Ein Zeichen "n0" wird an der ersten freien Stelle von rechts in der PRC-Zone eingesetzt (n - Nr. des Programms). Der linke Teil der PRC-Zone wird um eine Stelle nach links versetzt, wodurch die Nummer "n", die an Stelle 70 stand, verschwindet.

Anmerkungen:

1. Ein durch den Befehl STOP angehaltenes Programm kann nur durch einen Befehl START, der sich in einem anderen Programm befindet, das später in Aktion tritt, wiedergestartet werden (s.4.3.3.2).
2. Der Befehl STOP dient nicht dazu, ein Programm endgültig anzuhalten.
3. Er kann keine Kapazitätsüberschreitung der PRC-Zone hervorrufen.

4.3.3.2. Starten

START

reSTART a program halted by a stop.

41 N

N = Nummer des Programmes, das gestartet wird (01-05)

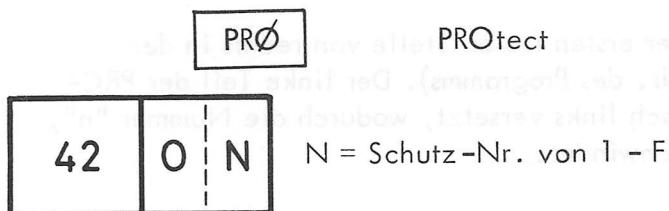
Der Befehl START, der in einem Programm, dessen Nummer ungleich "N" (im Befehl) ist, startet ein Programm "N", das vorher durch den Befehl STOP angehalten wurde.

Dazu wird der rechte Teil der PRC-Zone untersucht, ob das Zeichen "n0" eingespeichert wurde:

- a) wenn Ja : wird dieses Zeichen gelöscht, der rechte Teil der PRC-Zone wird zusammengeschoben und das Zeichen "1n" wird an der ersten freien Stelle von links in diese Zone eingesetzt. Das linke Halbbyte des vorhergehenden Bytes wird auf Null gesetzt.
- b) wenn Nein: ist der Befehl START ohne jede Wirkung.

In jedem Fall geht das laufende Programm zum nächsten Befehl über. Dieser Befehl kann keine Kapazitätsüberschreitung der PRC-Zone hervorrufen.

4.3.3.3. Schutz



Trifft ein Programm "N" auf diesen Befehl, wird der rechte Teil der PRC-Zone untersucht, ob ein dem Befehl entsprechendes Zeichen "Ak" eingespeichert wurde.

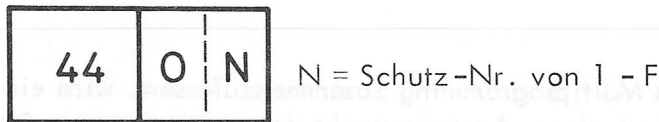
- a) bei Nein: ist dieser Schutz offen .
 - das Zeichen "Ak" wird an die erste freie Stelle rechts in der PRC-Zone eingesetzt (k = Nummer des im Befehl genannten Schutzes).
 - das Programm "N" geht zum nächsten Befehl über.
- b) bei Ja : ist dieser Schutz geschlossen .
 - das Zeichen "nk" wird an die erste freie Stelle von rechts in die PRC-Zone eingesetzt (n = Nummer des Programms an Stelle 70; k = Nummer des im Befehl gegebenen Schutzes).
 - das Programm "N" wird angehalten : der linke Teil der PRC-Zone wird um eine Stelle nach links versetzt, wodurch die Nummer "n" an Stelle 70 verschwindet und ein Wechsel des Programms erfolgt.

Anmerkungen:

1. Ein durch den Befehl PRØ angehaltenes Programm kann nur durch den Befehl FREE gestartet werden, der sich in einem anderen Programm befindet, das später zur Ausführung gelangt (s.4.3.3.4).
2. Der Befehl PRØ kann im Fall (a) eine Kapazitätsüberschreitung der PRC-Zone und damit eine Programmunterbrechung nach sich ziehen.

4.3.3.4. Freimachung

FREE FREE



Trifft ein Programm einen Befehl FREE, wird der rechte Teil der PRC-Zone untersucht, ob ein Zeichen "Ak" und ein oder mehrere Zeichen "nk", die dem angegebenen Schutz entsprechen, vorhanden sind. Drei Fälle können auftreten:

- a) Keines dieser Zeichen ist vorhanden: Der Schutz mit der Nummer "k" ist offen. Der Befehl FREE hat keine Auswirkung.
- b) Nur das Zeichen "Ak" befindet sich in der Zone : Der Schutz "K" ist geschlossen, aber es wird durch ihn kein Programm blockiert. Das Zeichen "Ak" wird gelöscht, wodurch der Schutz "K" geöffnet wird.
- c) Das Zeichen "Ak" und ein oder mehrere Zeichen "nk" befinden sich in der Zone: Der Schutz "K" ist geschlossen und blockiert ein oder mehrere Programme.
 - das erste Zeichen "nk" (k = im Befehl angegebene Nummer), das von rechts an angetroffen wird, wird gelöscht und das Zeichen "ln" (n = Nummer, die sich im gelöschten Zeichen "nk" befand) wird an die erste freie Stelle von links in die PRC-Zone eingesetzt (das linke Halbbyte der vorhergehenden Stelle wird auf Null gesetzt). Das drückt sich durch den Start des Programms "N" aus.
 - Das Zeichen "Ak" und die anderen evtl. Zeichen "nk" bleiben erhalten.

In allen Fällen geht das Programm zum nächsten Befehl über.

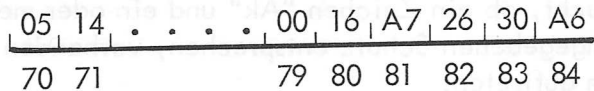
Anmerkung:

Die Fälle (b) und (c) ziehen ein Verschieben des rechten Teils der PRC-Zone nach sich. Der Befehl ruft keine Kapazitätsüberschreitung hervor.

Die beiden Befehle PRØ und FREE arbeiten analog dem Blocksystem der Eisenbahn. Der Befehl PRØ erzeugt Zeichen "Ak" oder "nk" (rotes Licht) bei jeder Ausführung und der Befehl FREE löscht jedesmal nur ein Zeichen (grünes Licht). Das bedeutet, daß jede Überschreitung eines PRØ, der einen Schutz "K" betrifft ein Übergang auf einen FREE entspricht, der den gleichen Schutz betrifft, wenn man nicht eine allgemeine Blockade des Programms riskieren will.

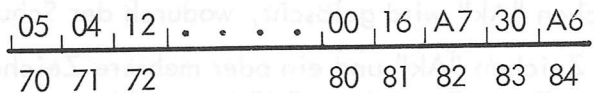
4.3.4. Beispiele

Um die Funktion des Multiprogramming zusammenzufassen, wird ein aus-sagefähiges Beispiel gegeben. Ausgangspunkt der angenommenen Situation ist das Ende des Abschnitts 4.3.2. Die PRC-Zone enthält:



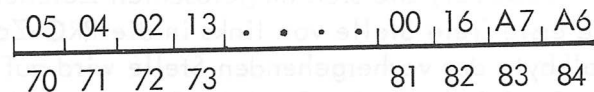
1. Das Programm 5 trifft den Befehl FREE 6

Das erste durch den Schutz 6 angehaltene Programm (Programm 2) wird gestartet.



2. Das Programm 5 trifft den Befehl START 3

Das Programm 3 wird gestartet.



3. Das Programm 5 trifft einen IØC-Befehl

Es erstarrt während der Durchführung dieses Befehls. Das Programm 4 wird zur Ausführung gebracht.



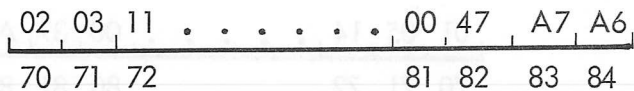
4. Das Programm 4 trifft einen Befehl PRØ 7

Der Schutz 7 ist geschlossen. Man geht zu Programm 2 über.



5. Das Programm 2 trifft einen Befehl FREE 6

Das durch den Schutz 6 blockierte Programm1 wird gestartet.



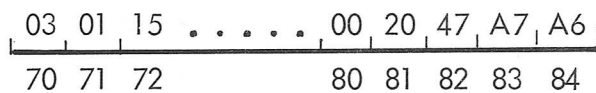
6. Der IØC-Befehl des Programms 5 ist beendet.

Das Programm 5 wird wiedergestartet.



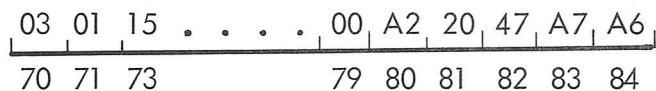
7. Das Programm 2 trifft einen Befehl STØP

Es wird angehalten und Programm 3 fährt fort.



8. Das Programm 3 trifft einen Befehl PRØ 2

Der Schutz 2 war "offen", es wird fortgefahren, nachdem er geschlossen wurde.

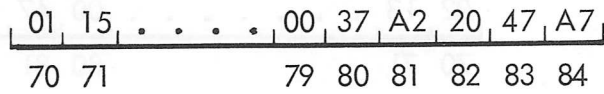


9. Das Programm 3 trifft einen Befehl FREE 6



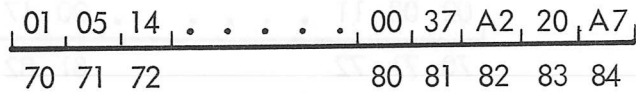
10. Das Programm 3 trifft einen Befehl PRØ 7

Es wird blockiert. Es wird mit Programm 1 fortgefahren.



11. Das Programm 1 trifft einen Befehl FREE 7

Das Programm 4 wird entblockiert und gestartet.



Diese Folge von Vorfällen in einem aussagefähigen Beispiel erläutert alle Fälle, die bei Anwendung des Multiprogramming beim GAMMA 55 auftreten können.

5. Die Bedienung des GE-55

5.1. Einführung

Um den GE-55 in Betrieb zu nehmen, ist eine Reihe von Programmen erforderlich, die in den verschiedenen Phasen zur Anwendung kommen. Diese Programme bilden "Systeme", die in den folgenden Abschnitten beschrieben werden sollen. Außerdem werden die Richtlinien zur Erstellung und Anwendung der Arbeitsprogramme gegeben.

5.2. Die Software des GE-55

Die Methoden zur Benutzung eines GE-55 richten sich nach den angeschlossenen Randeinheiten. Man unterscheidet darum zwei Methoden, je nachdem, ob der GE-55 mit oder ohne Hilfsspeicher (Magnettrommel) ausgestattet ist.

Außerdem unterscheidet man noch zwei Elemente, ohne die der Rechner nicht arbeiten kann:

- die Programme, die seine Arbeit steuern;
- die Dateien, die verarbeitet werden sollen.

5.2.1. Lochkartenanlage GE-55 (ohne Trommel)

Bei dieser Konfiguration werden die durchzuführenden Programme und zu verarbeitenden Dateien obligatorisch durch Karten in den Kernspeicher eingegeben. Daher rührt auch der Name dieser Art der Konfiguration.

Um eine Arbeit zu beginnen, müssen die Parameterkarten und anschließend die Datenkarten in den Kartenleser eingelegt werden. Das Programm wird nun geladen, gestartet und die Daten in der Reihenfolge verarbeitet, wie sie gelesen werden. Um mehrere Arbeiten aneinanderzureihen, gibt man in der gleichen Art die entsprechenden Programme und Dateien ein.

Während die Dateien, die Karte für Karte gelesen werden, einen unbegrenzten Umfang haben können, ist der Umfang eines Programmes, das in Zentralspeicher gespeichert wird, begrenzt durch die Kapazität dieses Speichers. Das gleiche gilt für bestimmte Daten, Tabellen etc. die am Anfang der Arbeit eingespeichert werden, und auf die man ständig Zugriff hat.

5.2.2. GE-55 mit Trommel

Die Magnettrommel bietet eine zusätzliche Speicherkapazität, die bei Anschluß von zwei Trommeln verdoppelt werden kann. Es ist durchaus möglich, Programme auf die Trommel zu übertragen, anstatt sie im Kernspeicher zu speichern und jeweils nur den Teil in den Zentralspeicher abzurufen, der gerade benötigt wird.

Man kann also ein Programm in mehrere "Segmente" unterteilen, die nacheinander in derselben Zone des Zentralspeichers stehen, und zwar in der erforderlichen Reihenfolge. Diesen Vorgang nennt man "Segmentierung".

Das ist aber nicht der ganze Vorteil der Trommel. So kann man gewisse Hilfsoperationen übergehen und ihre Durchführung erleichtern, indem man sie am Anfang einer Arbeitsperiode, z.B. täglich, auf der Trommel speichert gemeinsam mit den durchzuführenden Programmen und den permanenten Dateien einschließlich der Dateien mit Direktzugriff, die sehr wichtig sein können. Man erreicht so eine Beschleunigung der Ladezeit für Programme und Daten, die man mehrmals innerhalb einer Arbeitsperiode benötigt.

Um diese Arbeiten durchzuführen, ist es nur noch erforderlich, die durchzuführenden Programme dem Rechner anzuzeigen und sie in den Zentralspeicher zu übertragen, wo sie dann zur Ausführung gelangen. Der Bedienungskraft obliegt es also nur noch, neue Werte einzugeben und die Resultate in Empfang zu nehmen.

Zusammenfassung:

Aufgrund dieser beiden schematisch dargestellten Anwendungsmethoden muß auch die Software den Besonderheiten einer jeden von ihnen Rechnung tragen, um die Forderungen des Benutzers zu erfüllen. Sie muß daher:

- sich nach den beiden Konfigurationen der Maschine richten;
- bei jeder Konfiguration eine Hilfe bieten, die es ermöglicht, jede Stufe erreichen zu können und dadurch den Programmierer zu entlasten.

5.2.2.1. Die verschiedenen Systeme

Den zwei oben beschriebenen Konfigurationen entsprechen auch zwei Arten der Software, das "Kartensystem" und das "Trommelsystem". Von der Funktion her gesehen unterteilen sich diese wiederum in zwei Kategorien:

- Die "Programmiersysteme" dienen zur Vereinfachung der Erstellung von spezifischen Programmen, und zwar vom Schreiben des Programms ab bis zur endgültigen Fertigstellung in Maschinensprache.
- Die "Steuerungssysteme" steuern entweder die Durchführung der Programme in Maschinensprache oder - im Rahmen der zusätzlichen Speicher - die Organisation und die Auswertung der Dateien nach den vorher festgelegten Methoden.

Diese Systeme werden außerdem noch durch eine Bibliothek von Hilfs- und Standardprogrammen ergänzt, die dem Programmierer das Schreiben von Hilfsprogrammen zur Steuerung der Programme, der Dateien und deren Auswertung abnehmen.

Die Karten- und Trommelsysteme haben gemeinsame Berührungspunkte, die im folgenden zusammen beschrieben werden. Abweichungen werden jeweils angezeigt.

5.2.2.2. Die Programmiersysteme

Die Programmiersysteme bieten dem Benutzer

1. die Programmiersprachen, und zwar die Grundsprachen und die Umwandlungssprache für die Trommel (siehe Kapitel 3). Diese beiden Sprachen sind leichter anzuwenden als die Maschinensprache, bieten eine größere Sicherheit und enthalten Makrobefehle.
2. ein Übersetz- und Kontrollsystem, daß
 - die in der Grundsprache abgefaßten Programme übersetzt,
 - Aufbau und Wahrscheinlichkeit der kodierten Befehle prüft,
 - die Programme in die Maschinensprache bringt und sie listet.

Beim GE-55 mit Trommel kommt zu diesem Übersetz- und Kontrollsystem noch ein Umwandlungssystem hinzu, das

- die zusätzliche Trommelsprache übersetzt,
- die verschiedenen Teile des Source-Programms (Sektionen, Unterprogramme usw.) umwandelt, indem es ihnen die effektiven Adressen im Speicher und auf der Trommel zuordnet und die notwendigen Verbindungen herstellt,
- die erhaltenen Programme in Segmente für die Verarbeitung aufteilt,
- die für die Programme erforderlichen Felder im Kernspeicher und auf der Trommel reserviert.

3. ein Listprogramm, das
 - beim Testen von Programmen Programmlisten erstellt, um Fehlersuche und -korrekturen zu erleichtern;
 - eine endgültige Programmliste nach dem Test, die zu den Programmunterlagen gehört, erstellt;
 - das Erstellen von Parameterkarten für das Programm "Verdichten" ermöglicht.
4. ein Programm "Verdichten", das die getesteten Programme "verdichtet", um die Anzahl von Programmkarten zu verringern und so die Ladezeiten jeder Arbeit zu optimieren.

5.2.2.3. Die Betriebssysteme

Dem Benutzer stehen zur Verfügung:

Zum Fahren der Programme

Bei einer GE-55 Kartenanlage

- ein Ladeprogramm Kernspeicher, das
 - . die Verbindungszonen des verdrahteten Supervisors initialisiert (diese Zonen befinden sich am Anfang des Speichers) und den Kernspeicher evtl. ab Byte 161 löscht,
 - . das Programm "Unterbrechung" am Ende des Kernspeichers lädt,
 - . entweder das zu testende Source-Programm oder das auszuführende Teilprogramm in verdichteter Form in den Kernspeicher lädt,,
 - . bei einem Source-Programm die symbolischen Adressen in absolute Adressen umwandelt (siehe 3.2.),
 - . das Source-Programm oder Teilprogramm startet an der Anfangsadresse, die in einer Befehlskarte angegeben ist.

bei einem GE-55 mit Trommel

- ein Monitor-Programm, das von der Trommel abgerufen wird und
 - . das Programm "Supervisor TB" in den Kernspeicher lädt,
 - . die Verbindungszone (erweitert) des verdrahteten Supervisors und des Supervisors TB initialisiert,
 - . entweder das zu testende Programm oder das auszuführende Teilprogramm in den Kernspeicher lädt aufgrund des Kernspeicherladeprogramms, das zu einem Programm-Modul des Monitor-Programms wird.
- das Programm startet.

Anmerkung:

Das Kernspeicher-Ladeprogramm und der Monitor sorgen für die Betriebsfunktionen zwischen zwei Arbeiten, um sie aufeinander folgen zu lassen. Zu diesem Zweck werden sie in den Speicher geladen, sobald eine Arbeit beendet ist, und verschwinden, wenn die nächste Arbeit fertig zur Ausführung ist.

Bei einer GE-55 Kartenanlage

- ein "Unterbrechungsprogramm" P.G.I., das dem Bediener Standardverfahren zur Behebung von Störungen oder zum Informationsaustausch mit dem ablaufenden Programm liefert.

Dieses Programm wird vom Bediener wie unter 2.1.6. beschrieben in Betrieb gebracht. Es kann aufgrund angegebener Anweisungen folgende Funktionen ausführen:

- . das Lochen von Informationen wiederholen, die sich noch in der Stanzzone befinden,
- . eine Karte noch einmal lesen,
- . über die numerische Tastatur eine verschlüsselte Nachricht für das laufende Programm eingeben,
- . ein spezifisches Wiederanlaufprogramm zu starten, das evtl. unter mehreren solcher Programme ausgewählt wird,
- . vor allem das laufende Programm dort zum Wiederanlauf bringen, wo es unterbrochen worden ist.

Das P.G.I. kann ebenfalls auf der Sichtanzeige eine verschlüsselte Nachricht vom laufenden Programm anzeigen.

GE-55 mit Trommel

- ein Programm "Supervisor TB" (TB-Trommel)
 - . verarbeitet die Eingabe und Ausgabe der Trommel physikalisch, d.h. steuert (kontrolliert) die Lese/Schreiboperationen,
 - . verarbeitet Störungen auf den Randeinheiten und Programmunterbrechungen mittels eines P.G.I.-Moduls,
 - . verarbeitet die Anrufe von Segmenten eines gleichen Programms: Suche auf der Trommel, Laden in den Kernspeicher.
 - . Ruft schließlich das Monitor-Programm in den Kernspeicher, wenn eine Arbeit beendet ist, um die nächste anschließen zu können.

Anmerkung:

Das P.G.I. und der Subvisor TB befinden sich während des Arbeitsablaufes im Kernspeicher, um den Arbeitsablauf zu steuern.

GE-55 mit Trommel

- ein Ladeprogramm Trommel, das das oder die auszuführenden Programme in verdichteter Form auf die Trommel lädt und evtl. Trommelzonen für die Dateien reserviert. Diese Zonen werden anschließend vom Supervisor TB verarbeitet.

Zur Datenverarbeitung auf der Trommel

1. ein Programm "Dateienspeicherung", das die Dateien auf die Trommel speichert. Es gibt zwei Speicherungsverfahren:
 - sequentiell verkettet, d.h. jeder Satz auf der Trommel enthält die Adresse des nächstfolgenden Satzes,
 - indiziert sequentiell, d.h. die Sätze werden der Reihe nach nach ihren Kennzeichen gespeichert und können anhand einer Tabelle gesucht werden.
2. ein Programm "Dateifortschreibung", das es ermöglicht, in einer auf der Trommel gespeicherten Datei Sätze zu löschen, zu ersetzen oder hinzuzufügen. Die Datei kann wie oben angegeben gespeichert sein.
3. ein Programm "Eingabe und Ausgabe ausgewählter Informationen" das es ermöglicht, Teile einer Datei oder Sätze, die sich auf der Trommel befinden, in Karten zu lochen oder auszudrucken, um sie extern zu bearbeiten oder aus Sicherheitsgründen später wieder einzugeben.

5.2.3. Bibliothek der Service- und Standardverarbeitungsprogramme

Die Service-Programme, die die Programmier- und Betriebssysteme vervollständigen, sind in den vorangehenden Punkten beschrieben worden. Folgende Programme können noch dazugezählt werden:

- Erstellen Kartendateien
- Standardprogramm doppeln, Listen, Lochen
- Listen
- Bearbeitung von Kartendateien

Bei den Standard-Bearbeitungsprogrammen handelt es sich im wesentlichen um spezifische Unterprogramme:

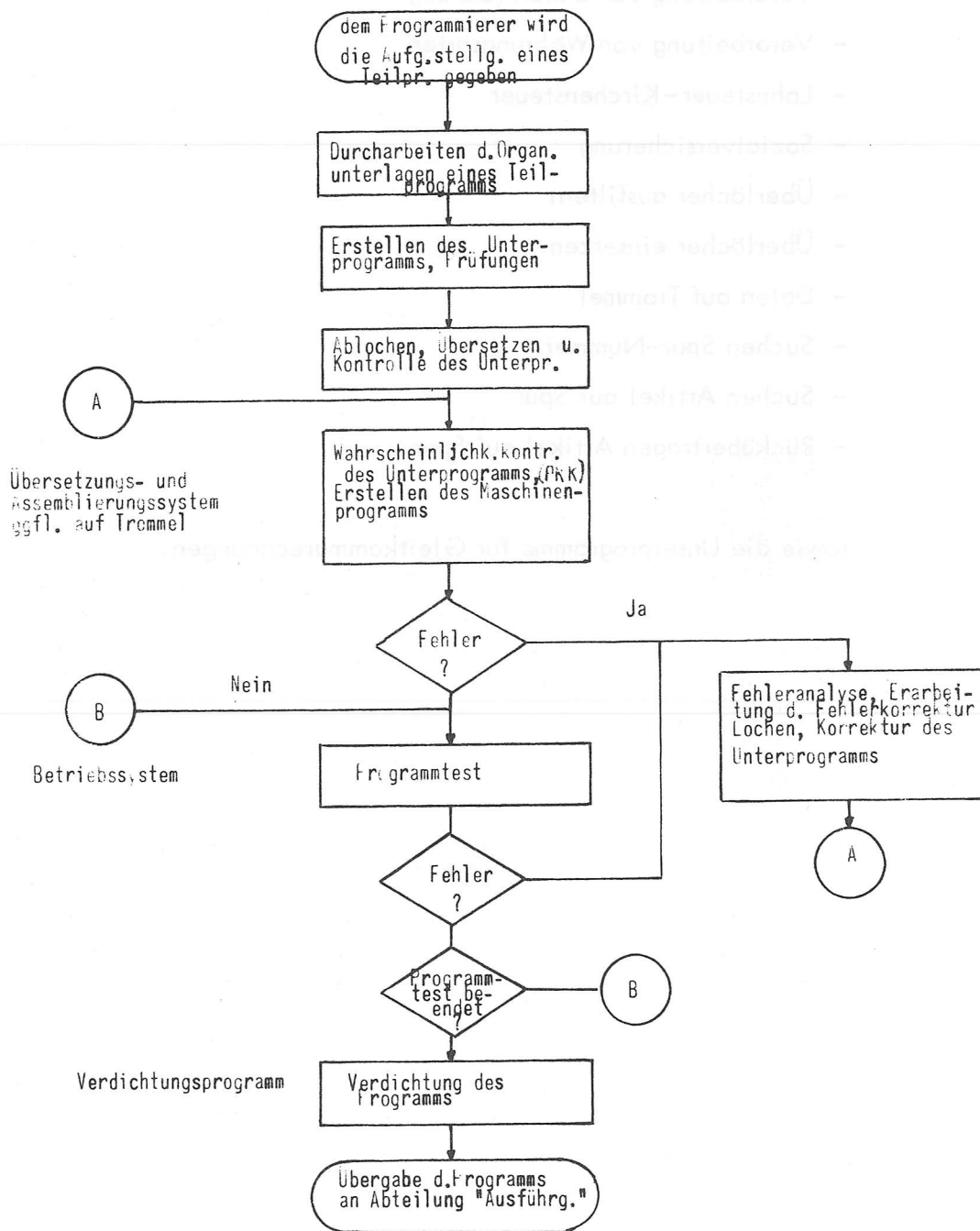
- Division
- Kennzeichenkontrolle
- Verarbeitung von Überweisungen (franz. System)
- Verarbeitung von Daten (Datum)
- Verarbeitung von Währungsarten
- Lohnsteuer-Kirchensteuer
- Sozialversicherung
- Überlöcher ausfiltern
- Überlöcher einsetzen
- Daten auf Trommel
- Suchen Spur-Nummer
- Suchen Artikel auf Spur
- Rückübertragen Artikel auf Spur

sowie die Unterprogramme für Gleitkommarechnungen.

5.3. Erstellen eines Programms

5.3.1. Verfahren

Die verschiedenen Operationen eines funktionsfähigen Programms gehen folgendermaßen ineinander über:



5.3.2. Erstellen der Unterprogramme

5.3.2.1. Untersuchung und Organisation eines Teilprogramms

Vor dem Erstellen eines Programms untersucht der Programmierer die Spezifikationen, die in der Aufgabenstellung des jeweiligen Teilprogramms enthalten sind. Diese Spezifikationen beziehen sich auf die zu verarbeitenden Daten, die erwarteten Ergebnisse und die auszuführenden Verarbeitungen. Die Spezifikationen stellen den ersten Teil der Programmunterlagen dar.

Der Programmierer untersucht dann die Organisation dieser Verarbeitungen auf der Maschine und erstellt dabei folgende Programmunterlagen:

- Kernspeicherbelegung
- In diesem Schema sind die zu reservierenden Speicherzonen (Ein-/Ausgabe; Arbeitsfelder usw.), die Zuordnung der numerischen Register, d.h. also die Kernspeicherbelegung der Daten und Ergebnisse und die verbleibende Kapazität für das Programm angegeben.
- Ein detailliertes Flußdiagramm.

Dieses Flußdiagramm ermöglicht:

- Das Programm evtl. in mehrere Teilprogramme zu unterteilen, die in Multiprogramming laufen, und diese Programme in Sektionen aufzuteilen, die mit symbolischen Kennzeichen versehen sind. Diese Symbole werden vom Programmierer sofort festgelegt.
- Die Sektionen zu definieren, die als Unterprogramme aufgebaut werden können.
- Im Rahmen der Trommelprogrammierung diese Sektionen herauszulösen, die unabhängige Segmente darstellen können (z.B. die zu Beginn des Programms benutzten Sektionen können nach ihrer Ausführung durch Wiederholungssektionen im Kernspeicher überlagert werden), oder solche Sektionen, die über zwei Segmente gelegt werden können, die im Kernspeicher überlagert werden können.

Vor der Erstellung dieser Unterlagen hat der Programmierer die Normen und Einsatzbedingungen des betreffenden Betriebssystems zur Kenntnis genommen, um:

- die erforderliche Speicherkapazität für dieses System zu berücksichtigen (PGI. Supervisor, TB),
- die Standard-Zuordnung für Ein-/Ausgabefelder und Register zu berücksichtigen,
- zu berücksichtigen, welche Initialisationen beim Laden des Programms in den Kernspeicher vorgenommen werden, die nicht programmiert werden (Initialisation von Registern und Ein-/Ausgabefeldern), und diejenigen, die er vorsehen muß.
- Die Standard-Wiederanlaufverfahren nach Fehlern, um vor allem die spezifischen Wiederanlaufverfahren zu verkürzen, die aufgrund der Art der Verarbeitung und der Dauer programmiert werden müssen.
- Das Ladeverfahren von Segmenten, die von der Trommel kommen, um die bestmögliche Aufteilung festzulegen unter Berücksichtigung des Verhältnisses verfügbare Kernspeicherkapazität / voraussichtliche Ladezeit.
- Die Aufbaunormen dieser Segmente.

Der Programmierer muß außerdem die Standard-Unterprogramme kennen, um sie sinnvoll einzusetzen, und deren Einsatzbedingungen, die reservierten Symbole usw. kennen.

Symbolische Kennzeichnung der Sektionen

In der Basis-Programmiersprache stehen dem Programmierer 240 verschiedene Kennzeichnungen zur Verfügung, von denen er ggfl. die für die benutzten Software-Programme reservierten Bezeichnungen abziehen muß.

Was die Anwendung der Kennzeichnung betrifft, ist es gemäß den besonderen Kriterien in jedem Programm vorzuziehen, einheitliche Normen für alle Programme festzulegen, und die symbolischen Kennbezeichnungen anzusehen als ein praktisches Mittel zur Untersuchung und zur Verbindung zwischen dem Flußdiagramm und dem zugehörigen Programm. Zu diesem Zweck ordnet man die Kennzeichnung aufsteigend in der gleichen Ordnung wie im Flußdiagramm. Die Kennzeichnung wird aus zwei Zeichen gebildet: Das erste kann zum Beispiel die Seitennummer des Flußdiagramms sein und das zweite die laufende Nummer der Sektion auf der Seite. Auf diese Weise kann man 15 Seiten und 16 Sektionen pro Programm unterscheiden.

5.3.2.2. Ausarbeitung des Programms

Ist das Flußdiagramm detailliert aufgestellt, kann der Programmierer die Ausarbeitung des Programms (oder der Programme) in Angriff nehmen. Dazu benutzt er Programmformulare, die sowohl für die Basissprache als auch für die Assembler-Sprache in der Trommel gültig sind. Die ausgefüllten Programmformulare dienen als schriftliche Grundlage für die Ablochung der Programme. Jede Zeile entspricht einer Karte von 80 Spalten.

Das Programmformular

In 8.4. wird ein Beispiel für ein Programmformular gezeigt. Bei der Programmierung ist dem Aufbau dieses Formulars genau zu entsprechen. Die einzelnen Angaben sind nur in die zugehörigen Spalten einzusetzen.

Die Rubriken

Der Programmierer hat die Möglichkeit, in die einzelnen Zeilen neben den Befehlen zusätzliche Bemerkungen einzusetzen, um den Verlauf des Programms zu verdeutlichen.

Rubrik Kartenart (Spalte 2 und 3).

Diese Kartenart gibt an, wie der in der Zeile enthaltene Befehl verarbeitet werden muß.

Das erste Zeichen (Spalte 2) zeigt an, wo der Befehl gespeichert werden muß.

Beispiele:

Der Schlüssel 1 zeigt an, daß die Befehle der Reihe nach von der Anfangsadresse an gespeichert werden. Die Adresse wird durch den Pseudo-Befehl BEGIN definiert (Elementarbefehle der Basissprache etc.).

Der Schlüssel 3 zeigt an, daß der Befehl an einer absoluten Adresse zu speichern ist. Diese wird vom Programmierer angegeben.

Das zweite Zeichen (Spalte 3) gibt Art und Größe des Befehls an.

Beispiele:

Der Schlüssel 1 gibt an, daß der Befehl in entpackter Form anzuordnen ist, d.h. daß jedes Zeichen ein ganzes Byte belegt (alphanumerische Konstante).

Der Schlüssel 2 gibt an, daß der Befehl in gepackter Form gespeichert wird, d.h. daß jedes Zeichen ein Halbbyte belegt (Befehlsparameter, hexadezimale Konstanten).

Rubrik Zahl der Bytes (Spalte 4 und 5)

Der Programmierer gibt in dieser Rubrik die Anzahl der Bytes an, die durch die Parameter der Befehle ab Spalte 17 belegt werden (Befehle, Konstanten).

Eventuelle Kommentare sind in dieser Anzahl nicht enthalten.

Diese Zahl begründet einen der Parameter der Pseudo-Befehle der Konstantenerstellung (siehe 3.9.2.).

Rubrik Ordnungsnummer (Spalte 6 bis 9)

Diese Nummer setzt sich aus 4 Dezimalstellen zusammen und legt die Reihenfolge der Programmkarten fest.

Während der Anfangsausarbeitung eines Programms ist es empfehlenswert, die äußerste rechte Ziffer nicht zu benutzen und ihr systematisch den Wert 0 zu geben. Dieses erleichtert die späteren Modifikationen des Programms und erlaubt das Einsetzen von 1 bis 9 Zeilen zwischen zwei ursprünglich auf dem Formular vorhandenen Zeilen. Es ist nicht erforderlich, die bereits vorhandenen Nummern zu ändern; die neuen Linien werden dann beziffert von xxx1 bis xxx9.

Bemerkung

Der Programmierer beziffert nicht die Pseudo-Befehle der Konstantenerstellung, die sich in einer beliebigen Reihenfolge befinden können. Er vermerkt vielmehr in den Spalten 6 bis 9 einen der Parameter dieser Pseudo-Befehle, nämlich die Adresse der durch die spezifischen Konstanten belegten Zone in der betreffenden Linie.

Der Befehl

Rubrik Symbolischer Operationstyp. (Spalte 10 bis 16)

Die Buchstaben und Ziffern, die zum symbolischen Operationstyp zusammengefügt sind, werden von Spalte 10 an ohne Zwischenraum geschrieben. Die rechts nicht gebrauchten Spalten bleiben frei.

Rubrik Befehl (Spalte 17 bis 44)

Die Parameter werden von Spalte 17 an nacheinander ohne Zwischenraum geschrieben auf der Grundlage eines alphanumerischen oder hexadezimalen Zeichens pro Spalte. Die rechts unbenutzten Spalten bleiben frei. Beim Schreiben der Parameter sind die Formvorschriften und Kodierungsregeln, die in den Kapiteln 3 und 4 angegeben sind, genau einzuhalten.

Was indessen die Grundbefehle betrifft, sind die Spalten 17 und 18 reserviert. Der Programmierer kann hier dem hexadezimalen Operationstyp angeben.

Die Kommentare (Spalten 34 oder 46 bis 80)

Die Kommentare sind Informationen in Klartext, um die Sektionen, Sequenzen oder Befehle zu erläutern.

Das Programmiersystem führt keine Bearbeitung des Kommentartextes durch. Die Kommentare werden nur auf den Programmlisten wiedergegeben.

Der Programmierer kann sowohl Befehlslinien durch Kommentare vervollständigen als auch zwischen den Befehlslinien besondere Kommentarzeilen einfügen. Im ersten Falle können die Kommentare in Spalte 46, im zweiten Falle in Spalte 34 begonnen werden.

Der Programmierer kann zu diesem Zweck alle druckbaren Zeichen gebrauchen und so viele Abstände hinzufügen, wie er benötigt. Die Kommentare können eine beliebige Stelle innerhalb der zugeordneten Zonen belegen.

Bemerkungen:

Wird ein Kommentar in eine Zeile gesetzt, in der auch ein Befehl enthalten ist, muß die Spalte 45 frei bleiben. In einer Zeile, die nur einen Kommentar enthält, müssen die Spalten 10 bis 33 frei bleiben.

Kommentare können nach den absoluten spezifischen Werten eines Pseudo-Befehls der Konstantenerstellung geschrieben werden. Es ist indessen dringend erforderlich, sie durch Abstände zu trennen.

Kennzeichen (Spalte x bis 80)

Es ist empfehlenswert, die letzten Spalten jeder Linie zu reservieren, um hier das Kennzeichen des Programms zu vermerken (Nummer, Code, etc.). Dieses Kennzeichen wird in alle Karten eines bestimmten Programms wiederholt.