



Komponenten des OS/ES

**Autoren: Mitarbeiter im ehemaligen Bereich
OS/ES des RPD im Kombinat Robotron**

Fassung: 30.11.2007

Inhaltsverzeichnis

| | | |
|-------------|---|-----------|
| 1. | EC 1040 im OS/ES | 3 |
| 1.1. | Historisches..... | 3 |
| 1.2. | Charakteristik | 3 |
| 1.3. | Komponenten | 4 |
| 1.3.1. | Übersicht | 4 |
| 1.3.2. | Steuerprogramm | 5 |
| 1.3.3. | Serviceprogramme..... | 20 |
| 1.4. | Unterstützung der Datenfernverarbeitung für EC 1040 in den Betriebssystemen OS/ES 4.1 und 6.1..... | 28 |
| 1.5. | Compiler für Programmiersprachen..... | 30 |
| 2. | EC 1055, 1055M, EC 1056, EC 1057 im OS/ES | 33 |
| 2.1. | Historisches..... | 33 |
| 2.2. | Charakteristik | 34 |
| 2.3. | Komponenten | 35 |
| 2.3.1. | Übersicht | 35 |
| 2.3.2. | Zugriffsmethoden | 36 |
| 2.3.3. | Serviceprogramme..... | 39 |
| 2.4. | Unterstützung der Datenfernverarbeitung für EC1055, EC1055M, EC1056 und EC1057 in den Betriebssystemen OS/ES 6.1 und OS/ES 7.x | 40 |
| 2.5. | Programmiersysteme | 43 |
| 3. | SVM/ES..... | 45 |
| 3.1. | Historisches..... | 45 |
| 3.2. | Charakteristik | 45 |
| 3.3. | Komponenten | 45 |

Abbildungsverzeichnis

| | | |
|-------------|--|----|
| Abbildung 1 | Jobverwaltung bei den Steuerprogrammen für Multiprogrammbetrieb | 9 |
| Abbildung 2 | Hauptspeicherorganisation bei verschiedenen Steuerprogrammkonfigurationen | 12 |
| Abbildung 3 | Zugriffsmethoden..... | 17 |
| Abbildung 4 | Vom Quellmodul zum ausführbaren Programm | 22 |
| Abbildung 5 | Katalog-Hierarchie bei VSAM..... | 38 |

1. EC 1040 im OS/ES

1.1. Historisches

Die Entwicklung des OS/ES begann im Kombinat Robotron zeitlich später als die des DOS-1, da die zu Beginn der 1970er Jahre entwickelte und produzierte Anlage R21 von der Leistungsfähigkeit her mit diesem Betriebssystem überfordert gewesen wäre.

Mit der Gründung der Entwicklungsgemeinschaft der sozialistischen Länder für das Einheitssystem elektronischer Rechner - **ESER** und der damit verbundenen Festlegung der Entwicklungsaufgaben der einzelnen Staaten kam es in der DDR zur Entwicklung des Rechnersystems EC1040, was später als ESER Reihe I bezeichnet wurde. Sie entsprach in wesentlichen Parametern einer Maschine der IBM-Reihe 360.

Bereitgestellt wurde dann in den 1970er Jahren das Betriebssystem OS/ES 4.1 mit den Betriebssystemvarianten PCP, MFT und MVT, wobei das Betriebssystem PCP zwar aus dem gelieferten Generierungsmaterial (für die Anpassung an die Maschinenkonfiguration und Bedürfnisse der Nutzer) generierbar war, aber eine offizielle Beschreibung und Nutzbarkeitsempfehlung fehlten. D.h. das Betriebssystem PCP war nicht offizieller Bestandteil des OS/ES 4.1. Ähnlich verhielt es sich mit dem Betriebssystem MVT, welchem erst in der Variante 4.14 eine offizielle Beschreibung beigelegt wurde, die jedoch nicht alle möglichen Nutzungen enthielt.

Es folgte die Auslieferung des Betriebssystems OS/ES 6.1. Es enthielt für die Betriebssystemvariante MVT Funktionserweiterungen und Fehlerkorrekturen des OS/ES.

1.2. Charakteristik

| Betriebs-system | Prinzip der Hauptspeicherverwaltung | HS-sparende Möglichkeiten |
|-----------------|---|---|
| PCP | Steuerprogramm für eine Aufgabe - PRIMARY CONTROL PROGRAM | |
| | sequentielle Verarbeitung von Jobs der Stapelverarbeitung, parallele Arbeit von Reader bzw. Writer möglich. | Um HS zu sparen, besteht für Programme die Möglichkeit, diese in Überlagerungsstrukturen zu untergliedern und gewisse Programmstrukturteile nur dann in den HS zu laden, wenn sie gebraucht werden. |

| Betriebs-system | Prinzip der Hauptspeicherverwaltung | HS-sparende Möglichkeiten |
|-----------------|--|--|
| MFT | Multiprogrammbetrieb mit einer festen Anzahl von Aufgaben - MULTIPROGRAMMING WITH A FIXED NUMBER OF TASKS | |
| | Bei Systemstart Aufteilung des Hauptspeichers in Partitionen, in denen dann Stapelprogramme, Reader, Writer und Initiator arbeiten → Parallele Arbeit von Programmen = Multiprogramming | Einführung der Regionsstruktur innerhalb der Überlagerungsstruktur |

| Betriebs-system | Prinzip der Hauptspeicherverwaltung | HS-sparende Möglichkeiten |
|-----------------|---|--|
| MVT | Multiprogrammbetrieb mit einer variablen Anzahl von Aufgaben - MULTIPROGRAMMING WITH A VARIABLE NUMBER OF TASKS | |
| | Keine feste Aufteilung des Hauptspeichers in Partitionen mehr, dafür virtuelle Speicherbelegung (Aufteilung) entsprechend gefordertem Speicherplatz (REGION-Parameter) und zeitlich gegebenen Möglichkeiten | Auch hier Regionstruktur innerhalb der Überlagerungsstruktur möglich. Programmmodule können mit dem Parameter Rollin-Rollout versehen werden. Dadurch können diese Moduln bei HS-Knappheit auf Platte ausgelagert und später wieder zurückgeladen werden - Vorstufe zum virtuellen Speicher |

1.3. Komponenten

1.3.1. Übersicht

Das Betriebssystem OS/ES besteht aus einem Steuerprogramm und weiteren Komponenten wie Compilern und Serviceprogrammen.

| Steuerprogramm | |
|--|---|
| Anfangsprogrammloader | Dienen zum Starten des Betriebssystems |
| Programm zur Initialisierung des Steuerprogrammkerne | |
| Jobverwaltung | Überwachung der Auftragsabwicklung und -steuerung |
| Jobdisponent | Initiieren des nächsten Verarbeitungsprogramms und Beenden des Verarbeitungsprogramms nach der Abarbeitung |
| Nachrichtendisponent | Kommunikation des Betriebssystems mit dem Bediener |
| Aufgabenverwaltung (Supervisor) | Überwachung der gesamten Arbeit des Betriebssystems (z.B. Steuerung des Multiprogrammbetriebs, Unterbrechungsüberwachung ...) |
| Datenverwaltung | Überwachung der Dateien im Betriebssystem; Transport von Daten zwischen peripheren Geräten und dem Hauptspeicher |
| Programme der Fehlerverwaltung | Erfassen, Behandeln, Aufzeichnen und Ausdrucken von Fehlerzuständen |

| Serviceprogramme | |
|--|---|
| Dienstprogramme | Für häufig benötigte Funktionen der Dateiverarbeitung stehen Dienstprogramme zur Verfügung, z.B. zum Löschen, Umbenennen, Ausdrucken oder Verändern von Dateien |
| Programmverbinder | Einzel übersetzte Programme werden zu einem ausführbaren Programm zusammengefügt und in einer Datei gespeichert |
| Lader | Einzel übersetzte Programme werden zu einem ausführbaren Programm zusammengefügt und im Hauptspeicher zur Verfügung gestellt |
| Sortieren/Mischen | Eine Datei kann nach bestimmten Kriterien sortiert oder mehrere bereits sortierte Dateien können zu einer sortierten Datei zusammengefügt werden. |
| Testhilfe für in Assemblersprache geschriebene Programme | |
| Teilnehmerunterstützung TSO | TSO ermöglicht die Bereitstellung von Jobströmen vom Abonnentenpunkt (z.B. Bildschirm) aus und kann deshalb die veraltete Lochkartentechnik ablösen. |
| Prüfprogramm für externe Geräte | |

| Compiler für Programmiersprachen (siehe Abschnitt 1.5) | |
|---|--|
| Assembler | Maschinenorientiert – besonders gute Ausnutzung der Rechnereigenschaften |
| PL1 | Universell – man kann Algorithmen der numerischen und der nichtnumerischen Datenverarbeitung formulieren |
| RPG | Problemorientiert – stark ein- und ausgabeorientiert, basiert auf Formularen |
| FORTRAN IV | Problemorientiert – für numerische Berechnungen aus dem wissenschaftlich-technischen Bereich |
| COBOL | Problemorientiert – für ökonomische Probleme |
| ALGOL 60 | Problemorientiert – für mathematische Probleme |

1.3.2. Steuerprogramm

1.3.2.1. Initialisierung des Steuerprogrammkerens

Alle bei der Generierung des Betriebssystems ausgewählten Komponenten befinden sich auf Magnetplatten. Beim Starten des Betriebssystems werden Teile des Steuerprogramms permanent in den Hauptspeichers geladen. Der Rest des Hauptspeichers wird als sogenannter dynamischer Bereich für Compiler, Serviceprogramme und benutzereigene Programme vorbereitet.

1.3.2.2. Jobverwaltung

Die Jobverwaltung dient der Kommunikation der Programmierer und der Anlagenbediener mit dem Betriebssystem und besteht aus zwei Komponenten, dem Jobdisponenten und dem Nachrichtendisponenten. Der Jobdisponent steuert den Übergang von der Ausführung eines Jobs zur Ausführung des nächsten, d.h. er organisiert die Stapelverarbeitung. Der Nachrichtendisponent verarbeitet einerseits die Bedienerkommandos und andererseits Nachrichten von allen Komponenten des Betriebssystems an die Bediener.

Wirkungsweise

Ein Job ist ein abgeschlossener Auftrag, der von einem Kunden an eine Rechenanlage zur Ausführung übergeben wird. Bestandteile eines Jobs sind Steueranweisungen, zwischen denen sich Daten für die Verarbeitungsprogramme befinden können. Diese Steueranweisungen und Daten können zu so genannten Jobschritten zusammengefasst werden, wenn jeder Jobschritt die Ausführung eines Programms erfordert. Mehrere hintereinandergelegte Jobs bilden den Systemeingabestrom. Für die Verarbeitung des Systemeingabestroms ist der erste Teil des Jobdisponenten, die Jobannahme, verantwortlich.

Die Jobannahme arbeitet bei den einzelnen Steuerprogrammkonfigurationen unterschiedlich. Beim Steuerprogramm für eine Aufgabe wird am Ende eines Jobs oder Jobschritts die Arbeit der Jobannahme zunächst unterbrochen, und das nächste Teilprogramm des Jobdisponenten, der Programmlauforganisator, übernimmt die Vorbereitung der Ausführung des Verarbeitungsprogramms.

Bei Multiprogrammbetrieb arbeitet die Jobannahme ohne Unterbrechung bis das Ende des Jobeingabestroms erreicht ist. Die Steueranweisungen werden in ein internes Format umgewandelt und in einer Datei, der Eingabekette, auf einem Direktzugriffsgerät abgespeichert. In der Eingabekette können die Jobs in 15 verschiedene Eingabeklassen eingeordnet werden. Die Daten im Jobeingabestrom werden ebenfalls auf Direktzugriffsspeichern abgelegt.

Der Programmlauforganisator ist für den eigentlichen Ablauf der Stapelverarbeitung zuständig. Er entnimmt die Jobs den Jobstapeln, die beim Steuerprogramm für eine Aufgabe mit dem Systemeingabestrom identisch sind und bei Multiprogrammbetrieb mit den Eingabeklassen übereinstimmen. Der Programmlauforganisator bereitet Jobs oder Jobschritte für die Ausführung vor und beendet sie.

Bei der Vorbereitung werden folgende Funktionen durchgeführt:

- Prüfung, ob der Jobschritt ausgeführt werden soll, beispielsweise in Abhängigkeit der Ergebnisse vorangegangener Jobschritte,
- Zuordnung von E/A-Geräten zu den Dateien, die vom Job oder Jobschritt benutzt werden sollen,
- Prüfung, ob die angeforderten Datenträger aufgelegt worden sind,
- Zuweisung von Hauptspeicherplatz (nur bei Multiprogrammierung mit einer variablen Anzahl von Aufgaben),
- Aufruf des benötigten Verarbeitungsprogramms.

Bei der Beendigung von Jobs führt der Programmlauforganisator folgende Hauptfunktionen aus:

- Disposition der Dateien entsprechend der vom Anwender gemachten Vorgaben,
- Freigabe der belegten E/A-Geräte.

Neben den beiden Teilen Jobannahme und Programmlauforganisator des Jobdisponenten gibt es bei Multiprogrammbetrieb noch einen dritten Teil, das Systemausgabeprogramm. Die langsamen Ausgabegeräte wie Paralleldrucker und Lochkartenstanzer werden dabei den Jobs nicht direkt zugeordnet, sondern die Ausgabe der Anwenderprogramme erfolgt auf Direktzugriffsgeräte. Ist ein Job beendet, werden Informationen dieser Daten in eine Ausgabekette gespeichert. Diese Informationen ermöglichen dem Systemausgabeprogramm, die Daten auf die eigentlichen Ausgabegeräte zu übertragen. Bei Bedarf können die Daten wie beim Steuerprogramm für eine Aufgabe direkt auf Paralleldrucker oder Lochkartenstanzer ausgegeben werden.

Wie schon erwähnt, arbeiten beim Steuerprogramm für eine Aufgabe Jobdisponent und Verarbeitungsprogramm abwechselnd. Das Verarbeitungsprogramm überlagert den Jobdisponenten, sobald es gerufen wird, und der Jobdisponent überlagert das Verarbeitungsprogramm, um dieses zu beenden.

Beim Multiprogrammbetrieb arbeiten die drei Teile des Jobdisponenten parallel zueinander und zu den Anwenderprogrammen.

Steueranweisungen

Die Steuerung des Ablaufs eines Jobs erfolgt nach den Anweisungen der Jobsteuersprache. Der Programmierer benutzt dazu folgende Steueranweisungen:

- **Jobanweisung**
Sie bezeichnet den Jobanfang. Sofern der vorhergehende Job im Eingabestrom keine sogenannte Nullanweisung enthält, kann sie auch das Ende der Jobsteueranweisungen dieses vorhergehenden Jobs markieren. Die Jobanweisung enthält Informationen, die der Ausführung des gesamten Jobs dienen.
- **Jobschrittanweisung**
Sie markiert den Jobschrittanfang und das Ende des vorhergehenden Jobschritts. In der Jobschrittanweisung wird dem Jobdisponenten der Name des Verarbeitungsprogramms oder der Name einer katalogisierten Prozedur mitgeteilt. Die Jobschrittanweisung enthält Informationen zur Ausführung des Jobschritts, z.B. Prüfungen, ob dieser Jobschritt in Abhängigkeit von den Ergebnissen vorangegangener Jobschritte überhaupt ausgeführt werden soll.

Katalogisierte Prozeduren sind Folgen von häufig benutzten Jobsteueranweisungen. Sie sind in einer speziellen Prozedurbibliothek gespeichert. Sie werden von der Jobannahme in den Jobeingabestrom eingefügt und brauchen so nicht immer wieder neu codiert zu werden. Im Eingabestrom können die Anweisungen der katalogisierten Prozedur vom Programmierer mit Hilfe anderer Jobsteueranweisungen geändert werden.
- **Datendefinitionsanweisung**
Diese Anweisung beschreibt eine Datei, die während eines Jobschritts verarbeitet werden soll. Mit dieser Anweisung werden dem Jobdisponenten insbesondere Informationen über den Dateinamen und die Dateidisposition mitgeteilt sowie Angaben gemacht über Datenträgerarchivnummer und Typ des Datenträgers mit der zu verarbeitenden Datei. Mit den Angaben zur Dateidisposition wird beschrieben, ob eine Datei zu Beginn des Jobschritts schon besteht, ob sie neu angelegt werden soll und ob sie nach Beendigung des Jobschritts bestehen bleiben soll oder nicht.
- **Prozeduranweisung**
Sie ist Bestandteil einer katalogisierten Prozedur und dient dazu, symbolischen Parametern Standardwerte zuzuweisen.

- **Kommandoanweisung**
Einige Bedienerkommandos (siehe unten) können auch als Kommandoanweisung in den Jobeingabestrom eingefügt werden.
- **Begrenzungsanweisung**
Befinden sich Daten im Jobeingabestrom, werden sie durch die Begrenzungsanweisung von den Jobsteueranweisungen getrennt.
- **Nullanweisung** (Jobendeanweisung)
Diese Anweisung zeigt das Ende der zu einem Job gehörenden Daten und Jobsteueranweisungen an. Befinden sich im Eingabestrom mehrere Jobs, wird das Ende der Jobsteueranweisungen des vorhergehenden Jobs auch durch die Jobanweisung des nächsten Jobs markiert.

Bedienerkommandos

Bedienerkommandos werden im Allgemeinen vom Anlagenbediener über die Bedieneinheit eingegeben. Einige jedoch können auch mit Hilfe der Kommandoanweisung in den Jobeingabestrom eingefügt werden. Folgende Funktionen können mit Hilfe dieser Kommandos ausgeführt werden:

- Systemaufgaben, beispielsweise ein Jobannahmeprogramm, starten oder stoppen,
- den Lauf von Jobs steuern; zum Beispiel Jobs in der Eingabekette von der Verarbeitung zurückhalten oder freigeben oder ihre Ausführung abbrechen, sowie
- Anfordern von Informationen über den Systemzustand; zum Beispiel eine Liste, der in der Eingabekette enthaltenen Jobs, anzeigen.

In Abbildung 1 wird die Jobverwaltung bei den Steuerprogrammen für Multiprogrammbetrieb dargestellt. Während beim Steuerprogramm für eine Aufgabe spätestens am Ende eines Jobs die Arbeit der Jobannahme unterbrochen wird und durch das nächste Teilprogramm des Jobdisponenten, den Programmlauforganisator, die Vorbereitung der Ausführung des Verarbeitungsprogramms durchgeführt wird, arbeitet die Jobannahme bei Multiprogrammbetrieb kontinuierlich, bis das Ende des Jobeingabestroms erreicht ist.

Der Eingabestrom wird aus den Daten gebildet, die sich auf den Medien Lochkartenleser, Magnetbandgerät oder Wechselplattenspeicher befinden. Über die Bedieneinheit besteht eine interaktive Kommunikation mit dem Nachrichtendisponenten. Die Jobannahme bedient sich ggf. zusätzlich katalogisierter Prozeduren mit Job-Steueranweisungen aus der Prozedurbibliothek. Nachdem die Eingabekette aufgebaut ist, läuft das Verarbeitungsprogramm ab und benutzt dabei Eingabedaten, die sich auf einem Wechselplattenspeicher befinden. Nach dem Ende des Verarbeitungsprogramms werden Ausgabedaten erzeugt sowie die Ausgabekette aufgebaut. Das Systemausgabeprogramm erzeugt Ausgabedaten, die im Allgemeinen sichtbar gedruckt werden oder auf Grund ihres Umfangs auf ein Magnetbandgerät ausgegeben werden. Weniger häufig ist die Ausgabe von so genannten Objektdecks auf Lochkarten im Lochkartenstanzer. Ausgaben auf Wechselplattenspeicher sind bei größerem Umfang wegen des begrenzten Speicherplatzes nicht üblich.

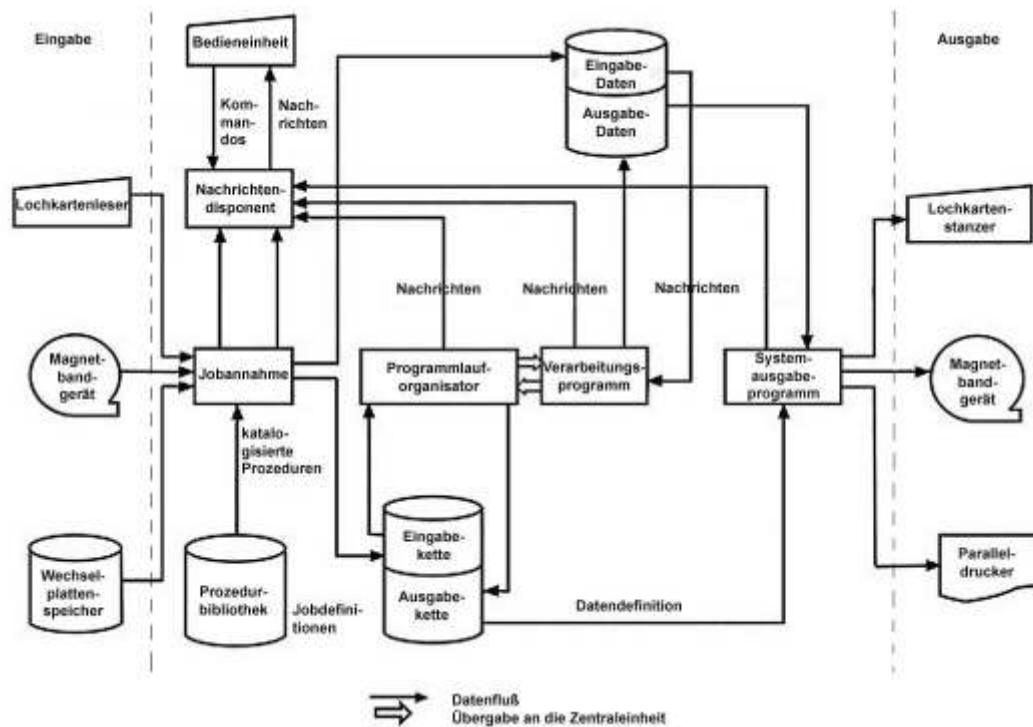


Abbildung 1 Jobverwaltung bei den Steuerprogrammen für Multiprogrammbetrieb

1.3.2.3. Aufgabenverwaltung

Unter Aufgabenverwaltung versteht man die Funktionen des Steuerprogramms OS/ES, die bei der unmittelbaren Abarbeitung von Verarbeitungsprogrammen benötigt werden. Diese Funktionen werden von der Komponente Supervisor des OS/ES ausgeführt.

Jedes Programm benötigt während seiner Abarbeitung gerätetechnische und programmtechnische Ressourcen. Die Abarbeitung eines Programms kann daher als Prozess betrachtet werden, in dessen Verlauf diese vorhandenen Ressourcen genutzt werden.

Die Charakterisierung und der Auftrag zur Ausführung dieses Prozesses bezeichnet man als Aufgabe für das Steuerprogramm des Betriebssystems.

Eine besondere Rolle spielt die Aufteilung der Ressourcen auf die Programme beim Multiprogrammbetrieb.

Die Funktionen des Steuerprogramms OS/ES, die die Übertragung von Daten zwischen externen Geräten und dem Hauptspeicher realisieren, gehören nicht zur Aufgabenverwaltung.

Supervisordienste

Wichtige Funktionen der Aufgabenverwaltung führt der Supervisor auf Anforderung durch das Benutzerprogramm aus. Diese Funktionen nennt man Supervisordienste. Verwendet der Benutzer die Assemblersprache, so werden die Supervisordienste als sogenannte SVC-Aufrufe in Form von Makros kodiert.

Zu den Supervisordiensten gehören:

- dynamische Speicherplatzverwaltung

Zur Bildung einer Aufgabe wird ihr vom Supervisor ein Hauptspeicherbereich zugewiesen und während der Abarbeitung der Aufgabe dynamisch verwaltet. Das Verarbeitungsprogramm belegt einen Teil dieses Bereichs selbst und kann während der Ausführung weiteren Hauptspeicherplatz anfordern. Diese angeforderten Bereiche werden vom Supervisor aus dem nicht belegten Teil des Hauptspeicherbereiches bereitgestellt, der der Aufgabe zugeordnet ist.

- dynamischer Programmaufruf

Die Verarbeitungsprogramme bestehen im Allgemeinen aus einzelnen Teilen, die auch Moduln genannt werden. Jeder Modul hat eine eigenständige Funktion. Die Moduln lassen sich an beliebige freie Stellen des Hauptspeichers laden. Sie befinden sich in ausführbarer Form in Bibliotheken auf externen Speichern mit Direktzugriff.

Das Verarbeitungsprogramm kann jeden beliebigen Modul aus einer solchen Bibliothek aufrufen, dessen Bereitstellung im Hauptspeicherbereich der Aufgabe anfordern und zu diesem Modul verzweigen. Der Supervisor lädt den Modul nur in den Hauptspeicherbereich der Aufgabe, wenn er sich noch nicht dort befindet bzw. wenn der im Hauptspeicher befindliche Modul nicht wiederverwendbar ist.

Nicht mehr benötigte Bereiche können freigegeben bzw. der Modul kann überspeichert werden, insbesondere dann, wenn nicht ausreichend Speicherplatz für den neu zu ladenden Modul vorhanden ist.

Der dynamische Programmaufruf besitzt verschiedene Vorteile:

- o Das Verarbeitungsprogramm passt sich auf Grund seiner Struktur an die Größe des Hauptspeicherbereichs an, der der Aufgabe zugeordnet ist.
- o Jedes Verarbeitungsprogramm kann jeden im Betriebssystem vorhandenen Modul aufrufen und z.B. als Unterprogramm benutzen.

- Zeitgeberdienste

Der Supervisor kann dem Verarbeitungsprogramm Datum und Uhrzeit mitteilen.

Außerdem kann das Verarbeitungsprogramm informiert werden, wenn Zeitintervalle abgelaufen sind, deren Größe das Verarbeitungsprogramm angefordert hat.

Prüfpunkt/Wiederanlauf

Treten bei der Abarbeitung eines Programms Geräte-, Daten-, Bedien- oder andere Fehler auf, so beseitigt das Betriebssystem in der Regel diese Fehler durch Wiederholen der ausgeführten Operationen. Der Bediener kann ebenfalls versuchen, durch Wiederholen von Kommandos den Fehler zu beheben.

Kann der Fehler nicht beseitigt werden, muss das Programm abgebrochen und wiederholt werden. Das kann bei Programmen mit langen Laufzeiten zu großen wirtschaftlichen Verlusten führen. Um das zu vermeiden, kann das Benutzerprogramm den Supervisor auffordern, an bestimmten Stellen im Programm, den Prüfpunkten, notwendige Daten für eine Wiederholung des Programms ab diesen Prüfpunkten zu sammeln.

Der Wiederanlauf kann nach der Beseitigung des Fehlers ohne erneute Eingabe des Jobs, also automatisch, oder verzögert nach erneutem Start des Jobs ab dem Prüfpunkt oder bei einem früheren Jobschritt erfolgen.

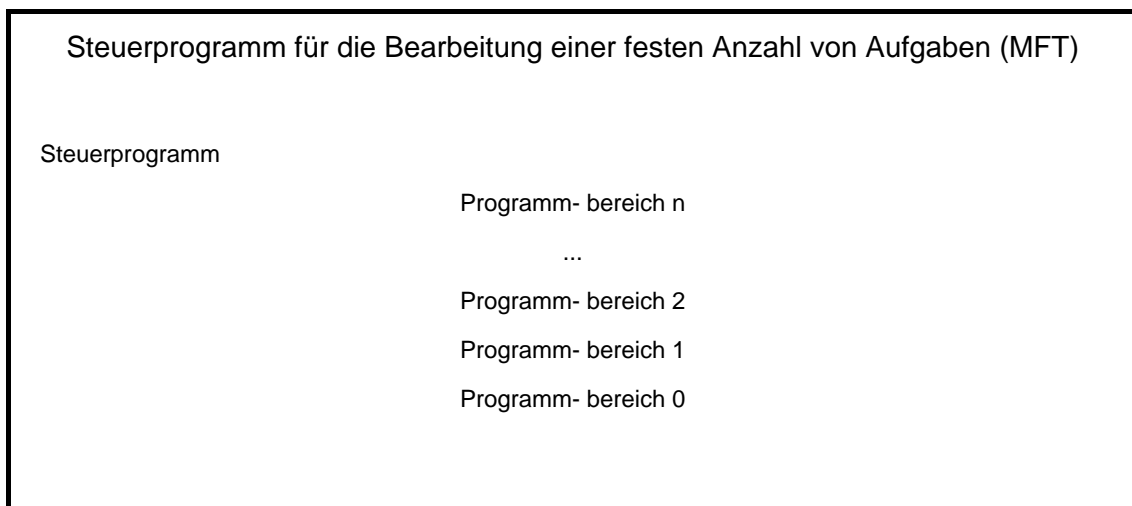
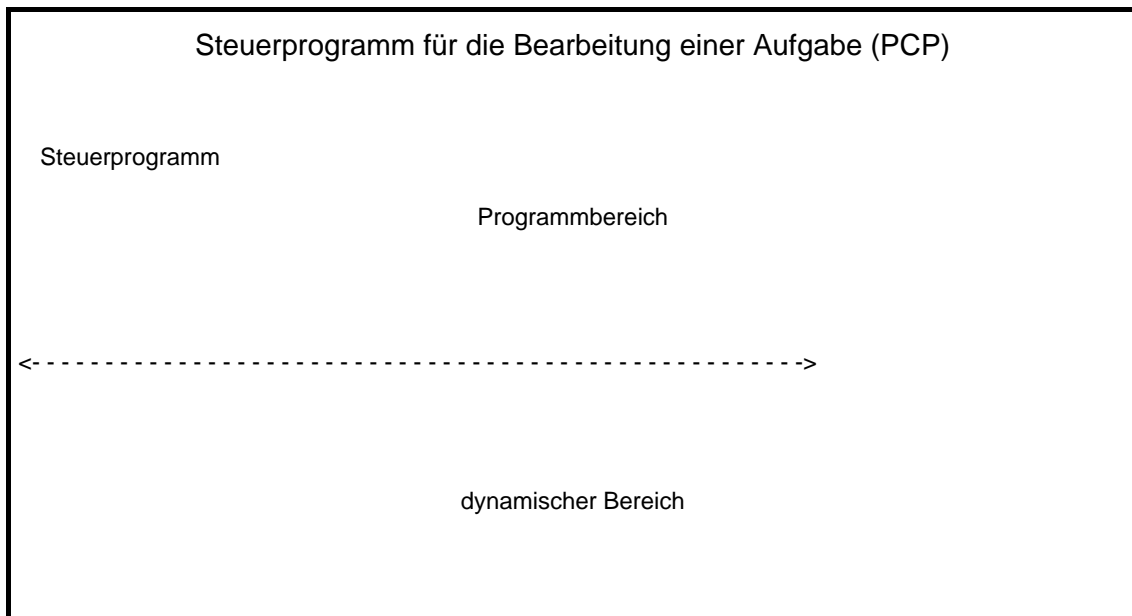
Der verzögerte Wiederanlauf dient auch dazu, einen Job neu zu starten, wenn dieser abgebrochen werden musste, um einem Job höherer Priorität den Vorrang zu geben.

Multiprogrammbetrieb

Die für die Abarbeitung eines Programms erforderlichen Ressourcen werden im Multiprogrammbetrieb auf alle parallel laufenden Programme aufgeteilt, um eine simultane Abarbeitung mehrerer Programme durch eine Zentraleinheit zu ermöglichen. Solche Ressourcen können z.B. E/A-Geräte, Hauptspeicher, Zentraleinheitszeit und auch Dateien und Programme sein.

Durch die simultane Nutzung der Ressourcen von mehreren Programmen wird eine wirtschaftliche Auslastung der gesamten Installation erreicht.

Für die Planung und Abarbeitung der einzelnen Programme und die Aufteilung der Ressourcen auf diese Programme ist der Jobdisponent zuständig. Der Jobdisponent bildet die zu bearbeitenden Aufgaben für den Supervisor.



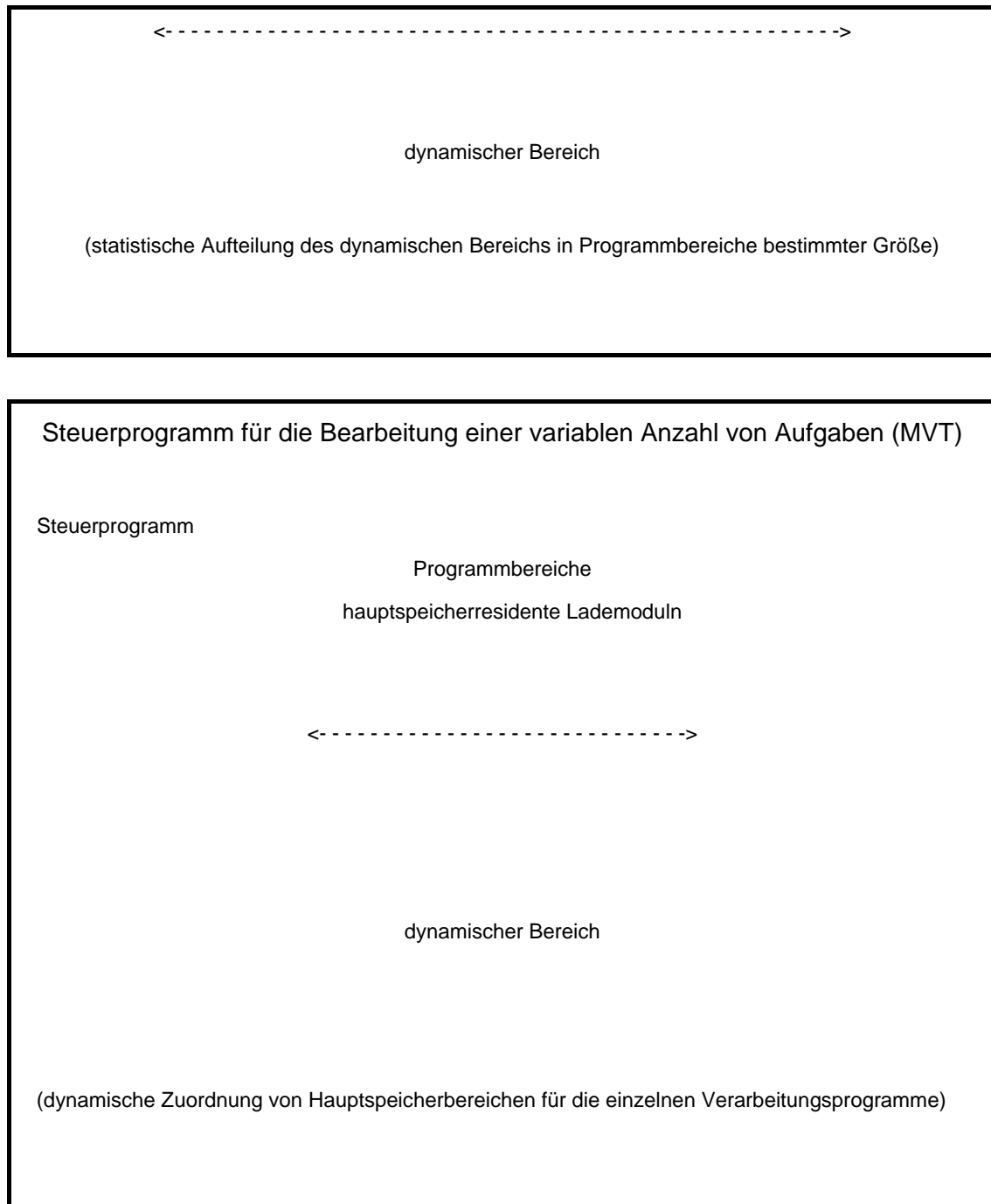


Abbildung 2 Hauptspeicherorganisation bei verschiedenen Steuerprogrammkonfigurationen

Die verschiedenen Arten des Multiprogrammbetriebs sind in Bild 2 dargestellt. Dadurch wird auch wesentlich die Struktur des Steuerprogramms bestimmt. Entsprechend der Art des Multiprogrammbetriebs unterscheidet man folgende Steuerprogrammkonfigurationen:

- a) Bearbeiten einer Aufgabe
- b) Bearbeiten einer festen Anzahl von Aufgaben (bis 15 benutzereigene)
- c) Bearbeiten einer variablen Anzahl von Aufgaben (bis 15 benutzereigene).

Die Auswahl der passenden Steuerprogrammkonfiguration muss auf der Grundlage des Spektrums der abzuarbeitenden Programme und der vorhandenen gerätetechnischen Ressourcen erfolgen. Bei der Steuerprogrammkonfiguration a) kann keine gleichzeitige Benutzung der Ressourcen erfolgen. Der Wirkungsgrad dieser Konfiguration ist deshalb wesentlich geringer als bei b) und c). Allerdings benötigt a) auch wenig gerätetechnische Ressourcen.

Multiaufgabenbetrieb

Die beschriebene Art des Multiprogrammbetriebs ist nur möglich, wenn diese simultan arbeitenden Programme die Ressource bzw. den Datenbestand, den sie bearbeiten, nicht verändern. Wird dieser Datenbestand jedoch verändert, so kann er im Sinne des Multiprogrammbetriebs nur von einem Programm benutzt werden.

Hat jedoch der Benutzer die Möglichkeit, selbst Aufgaben im Benutzerprogramm zu definieren und die Bearbeitung dieser Aufgaben zu steuern, so kann auch eine simultane Benutzung eines Datenbestandes erfolgen, der bei der Bearbeitung der Aufgaben geändert wird.

Eine solche Arbeitsweise wird Multiaufgabenbetrieb genannt. Bei den Steuerprogrammkonfigurationen b) und c) kann der Benutzer selbst neue Aufgaben, sogenannte Unteraufgaben, definieren. Es können neue Ressourcen definiert und gesteuert werden.

Auf diese Weise wird beim Multiaufgabenbetrieb die Möglichkeit der simultanen Ausnutzung von Ressourcen erhöht und damit die wirtschaftliche Auslastung der Anlage verbessert.

Simultane Bearbeitung von Aufgaben

Bei der gleichzeitigen Bearbeitung von Aufgaben sind die Zuordnung von Ressourcen und die Priorität, unter der die Bearbeitung der Aufgabe erfolgt, von Bedeutung. Die Priorität für jede Aufgabe wird vom Supervisor aus den Angaben des Benutzers und aus den bereits vorliegenden Aufgaben ermittelt. Aufgaben höherer Priorität erhalten vorrangig Zentraleinheitszeit und Hauptspeicherplatz. Um ein Blockieren von Aufgaben mit niedriger Priorität zu vermeiden, ist es möglich, allen Aufgaben so genannte Zeitscheiben bestimmter Größe zuzuteilen.

1.3.2.4. Datenverwaltung

Die Routinen der Datenverwaltung haben die Aufgabe, den Anwender bei der Verarbeitung von Daten zu unterstützen, die in Form von Dateien auf Datenträgern aufgebaut werden. Die Datenverwaltung übernimmt für den Anwender die Steuerung der Ein-/Ausgabeoperationen, die Aufbereitung der Daten und die Überlappung und Synchronisation der Arbeit der Zentraleinheit mit den Ein-/Ausgabeoperationen sowie die Datensicherung und Katalogisierung der Dateien.

Die Datenverwaltung erleichtert dem Anwender die Speicherung von Daten, den Zugriff zu ihnen und die Wartung von Dateien, z.B. ist eine bereits bestehende Datei allein durch Angabe einer Kennzeichnung, des sogenannten Dateinamens, eindeutig und vollständig beschrieben. Außerdem sind die Programme weitgehend unabhängig von Geräten und vom Dateiaufbau.

Datenträger und Dateien

Entsprechend dem zu lösenden Problem und seiner Zielstellung wählt der Anwender für seine zu bearbeitenden Datenbestände eine Organisationsform und die Art der Datenträger und damit die zu verwendenden Geräte aus.

Die Eingabe größerer Datenmengen zum Anlegen einer Datei kann zum Beispiel über Lochkarten erfolgen. Die zu verwaltende Datei dagegen wird je nach Häufigkeit der Verwendung und Notwendigkeit einer Aktualisierung entweder auf Lochband oder Lochkarte (bei geringem Anwendungsgrad), auf Magnetband (große Bestände für häufigen Zugriff) oder auf Magnetplatte (Bestände mit hoher Aktualisierungsrate) aufgebaut. Eine für den Anwender lesbare Form von Datenbeständen kann über Drucker oder Mikrofilmausgabegerät erzeugt werden. Die Dateien unterscheiden sich durch die Art der Speicherung der Daten auf einem Datenträger.

Begriffserläuterungen:

| | |
|----------------------------|---|
| Satz: | Grundeinheit der vom Problemprogramm verwendeten Information |
| Satzformat: | der für eine Datei typische Satzaufbau; es gibt Dateien mit Sätzen fester, variabler oder undefinierter Länge. Dabei können mehrere Sätze in einem Block zusammengefasst sein. |
| Feste Satzlänge: | Alle Sätze sind gleich lang |
| Variable Satzlänge: | Die Sätze haben ein zusätzliches Feld, in dem die Satzlänge eingetragen ist. Sind die zu verarbeitenden Sätze länger als die maximal mögliche Blocklänge, so besteht die Möglichkeit der Satzsegmentierung. Der Teil des logischen Satzes, der in einem Block enthalten ist, wird als Segment bezeichnet. |
| Sätze undefinierter Länge: | Bei Sätzen undefinierter Länge liegt nur die maximal zu erwartende Satzlänge, aber nicht die Länge jedes einzelnen Satzes fest. |
| Block: | Zusammenfassen mehrerer Sätze vor dem Ausschreiben auf einen Datenträger. Die Blöcke sind durch Aufzeichnungslücken getrennt. Der Block ist die physische Dateneinheit, die zwischen Hauptspeicher und Gerät übertragen wird. |
| Datei: | besteht aus einer Folge von Sätzen, die den gleichen Aufbau besitzen. |
| Kennsätze: | Maschinenlesbare standardisierte physische Blöcke, die auf den Datenträger aufgezeichnet und von den Datenverwaltungsroutinen ausgewertet werden. Entsprechend ihrer Funktion werden Datenträgerkennsätze und Dateikennsätze unterschieden. (siehe Abschnitt „Kennsätze“) |
| Datenträger: | Lochkarte, Lochband, Magnetband, Magnetplatte |

Dateiorganisation

Der Aufbau einer Datei wird außer durch das Satzformat vor allem durch die Speicherungsform bestimmt, die die Struktur der gesamten Datei definiert. Nach ihrer Speicherungsform unterscheidet man im OS/ES

- sequentielle Dateien,
- indexsequentielle Dateien,
- gestreute Dateien (direkt organisiert) und
- untergliederte Dateien.

Die Auswahl einer bestimmten Speicherungsform für eine Datei hängt vom Verwendungszweck und von den Verarbeitungswünschen ab. Die Verarbeitung kann entweder sequentiell oder wahlfrei erfolgen.

Bei einer **sequentiellen Verarbeitung** werden alle Sätze einer Datei in die Verarbeitung einbezogen und zwar in der Reihenfolge, in der sie den Datenträger belegen. Es werden keine logischen Zusammenhänge berücksichtigt.

Für die Aufnahme einer sequentiellen Datei steht jeder Datenträger zur Auswahl.

Bei **wahlfreier Verarbeitung** wird die Reihenfolge der Verarbeitung durch die Reihenfolge bestimmt, in der die Sätze der Datei angefordert werden, und nur die angeforderten Sätze der Datei werden in die Verarbeitung einbezogen.

Es werden E/A-Geräte mit direktem Zugriff, wie beispielsweise Plattenspeicher, verlangt.

Indexsequentielle Dateiorganisation

Eine indexsequentielle Datei kann sowohl sequentiell als auch wahlfrei verarbeitet werden. Sätze können fortgeschrieben, gelöscht und hinzugefügt werden. Bei einer indexsequentiellen Datei werden die Sätze in aufsteigender Sortierfolge in aufeinanderfolgende Speicherplätze gespeichert. Jeder Satz erhält einen Ordnungsbegriff (Schlüssel) vorangestellt. Eine indexsequentielle Datei besteht aus einem Hauptdatenbereich für die Datensätze, einem Indexbereich, der vom Betriebssystem zum effektiven Auffinden der Sätze benutzt wird, und einem Überlaufbereich, der zum Auslagern von Sätzen benutzt wird, wenn neue Sätze in die Satzfolge eingefügt werden.

Direkte Dateiorganisation

Bei einer direkt organisierten Datei erfolgt die Speicherung der Sätze gestreut. Ausgangspunkt für die Adressierung ist ein Ordnungsbegriff, den jeder Satz der Datei enthalten muss. Aus dem Ordnungsbegriff wird die Satzadresse abgeleitet. Es gibt 2 Möglichkeiten der Adressermittlung:

Direkte Adressierung – aus dem Ordnungsbegriff kann die Adresse des Satzes abgeleitet werden. Diese Adressierung wird verwendet, wenn die Ordnungsbegriffe einer Datei nur gering streuen.

Indirekte Adressierung – aus dem Ordnungsbegriff wird eine Spuradresse ermittelt, ab der sequentiell nach dem Satz zu suchen ist. Der Anwender muss selbst die konkrete Adresse durch ein Umrechnungsverfahren bereitstellen.

Untergliederte Dateiorganisation

Die dem Benutzer des Betriebssystems OS/ES gegebene Möglichkeit der Speicherung einer untergliederten Datei stellt eine wichtige Erweiterung gegenüber bisherigen Betriebssystemen dar. Diese Speicherungsform dient vor allem der Aufbewahrung von Programmen. Untergliederte Dateien entsprechen deshalb in ihrem Aufbau Programmbibliotheken, wie sie in anderen Betriebssystemen auch

existieren, dort aber ausschließlich vom Betriebssystem verwaltet werden. Deshalb werden solche Dateien, wenn sie Programme enthalten, auch im OS/ES häufig als Bibliotheken bezeichnet.

Eine untergliederte Datei besteht aus einem Verzeichnis und einer Folge von sequentiell organisierten Beständen. Das Verzeichnis belegt den Anfang der Datei und enthält den Namen und die Adresse jedes Bestandes relativ zum Dateianfang. Alle Bestände müssen gleiche Eigenschaften wie Satzformat, Blocklänge usw. besitzen.

Die Bestände werden in der Reihenfolge ihrer Aufnahme in die Datei gespeichert, während die Eintragungen im Verzeichnis nach den Bestandsnamen sortiert sind.

Der Benutzer kann eine untergliederte Datei dem Betriebssystem als Bibliothek deklarieren, damit es die von ihm angeforderten Programme (Bestände) daraus holt; Er kann aber auch direkt mit einem Bestand einer solchen Datei arbeiten. In beiden Fällen genügt die Angabe des Bestandsnamens, um das Programm geladen zu bekommen bzw. Zugriff zu dem gewünschten Bestand zu erhalten. Der Benutzer kann einen Bestand zu einer Datei hinzufügen, einen Bestand streichen und, wie bereits erwähnt, zu einem Bestand zugreifen.

Eine besonders für Bibliotheken sehr nützliche Funktion der Datenverwaltung ist die Verkettung von Eingabedateien. Die Dateikette wird durch Jobsteueranweisungen festgelegt. Es können jeweils bis zu 255 sequentielle oder bis zu 16 untergliederte Dateien miteinander verkettet werden. Eine solche Dateikette wird vom Betriebssystem für die Zeit der Gültigkeit der Jobsteueranweisungen logisch als eine Datei behandelt, und die einzelnen Dateien werden in der in den Jobsteueranweisungen aufgeführten Reihenfolge benutzt. Beispielsweise können die in einem Assemblerlauf benötigten Makros auf 16 verschiedenen Bibliotheken verteilt sein, die dann beim Aufruf eines Makros in der in der Kette festgelegten Reihenfolge nach der entsprechenden Makrodefinition durchsucht werden.

1.3.2.5. Die Zugriffsmethoden des OS/ES

Zugriffstechnik

Die Datenübertragung zwischen Hauptspeicher und externem Gerät erfolgt blockweise. Durch die Datenverwaltung können die Daten dem Benutzerprogramm block- oder satzweise angeboten werden. Diese unterschiedliche Form der Datenbereitstellung heißt Zugriffstechnik.

Einfache Zugriffstechnik:

Sie unterstützt die blockweise Verarbeitung. Die Beendigung der Datenübertragung sowie das Blocken und Entblocken muss das Benutzerprogramm übernehmen.

Erweiterte Zugriffstechnik:

Sie unterstützt die satzweise Verarbeitung und ist nur für sequentielle Verarbeitung geeignet (nicht für Direktzugriffs- und untergliederte Dateien). Sie übernehmen das Blocken und Entblocken der Sätze.

Die Kombinationen zwischen Zugriffstechniken und Dateiorganisationen werden als **Zugriffsmethoden** bezeichnet.

Unter Berücksichtigung dieser Funktionen ergibt sich, dass auf Grund der Datenstruktur die erweiterte Zugriffstechnik auf Direktzugriffs-Dateien sowie auf untergliederte Dateien einschließlich der Verarbeitung des Verzeichnisses dieser Dateien nicht möglich ist. Die anderen Kombinationen zwischen Dateiorganisation und Zugriffstechnik ergeben die in folgendem Bild dargestellten Zugriffsmethoden.

| Datei- organi- sation \ Zugriffs- technik | Zugriffstechnik | |
|--|---|--|
| | erweitert | einfach |
| physisch sequentiell | Erweiterte sequentielle Zugriffsmethode | Einfache sequentielle Zugriffsmethode |
| unter- gliedert | im vollem Umfang nicht möglich | Einfache Zugriffsmethode für untergliederte Dateien |
| Index- sequentiell | Erweiterte indexsequentielle Zugriffsmethode | Einfache indexsequentielle Zugriffsmethode |
| direkt | nicht möglich | Einfache Direktzugriffs- methode |

Abbildung 3 Zugriffsmethoden

Zusätzlich stehen noch Zugriffsmethoden für grafische Geräte und Datenfernverarbeitungsgeräte zur Verfügung, die ähnliche Funktionen beinhalten. Mit der erweiterten sequentiellen Zugriffsmethode ist die Verarbeitung geblockter und ungeblockter Sätze aller Satzformate (fest, variabel und undefiniert) möglich. Die einfache sequentielle und die einfache Direktzugriffsmethode verarbeiten ungeblockte Sätze aller Formate. Die erweiterte bzw. einfache indexsequentielle Zugriffsmethode verarbeitet geblockte und ungeblockte bzw. nur ungeblockte Sätze von festem und variablem Satzformat. Mit der einfachen Zugriffsmethode für untergliederte Dateien ist außer der ungeblockten Verarbeitung einzelner Bestände auch ein Zugriff zum Verzeichnis der untergliederten Datei (Lesen und Hinzufügen von Bestands-einträgen) möglich. Bei Verarbeitung eines einzigen Bestandes durch ein Programm ist es jedoch möglich und meist auch vorteilhafter, die sequentiellen Zugriffsmethoden zu verwenden.

Die Pufferung der Daten der Eingabe/Ausgabe

Der Anwender kann mit der Zugriffsmethode auch die Art der Zwischenspeicherung der Ein-/Ausgabedaten in Pufferbereichen auswählen. Damit ist eine Parallelarbeit zwischen Verarbeitungseinheit und den Operationen der Ein-/Ausgabe möglich.

Puffer: Unter einem Puffer im Sinne des Betriebssystems ist ein Bereich des Hauptspeichers zu verstehen, der über einen bestimmten Zeitraum ausschließlich zur Aufnahme von E/A-Daten vorgesehen ist. Ein Puffer dient im Normalfall zur Aufnahme eines Datenblockes. Ein Puffer ist in Segmente unterteilt, die je einen Satz aufnehmen können.

Pufferkomplex: Mehrere zusammenhängende Puffer bilden einen Pufferkomplex. Der Anwender kann explizit die Anzahl der Puffer anfordern (maximal 256 Puffer für eine Datei). Fehlt diese Angabe, so werden bei erweiterter Zugriffstechnik standardmäßig zwei Puffer zugewiesen, bei einfacher Zugriffstechnik erfolgt dagegen keine Pufferzuweisung, d.h. bei fehlender expliziter Pufferanforderung muss im Programm ein E/A-Bereich vorgesehen sein. Ein Pufferkomplex wird im Normalfall automatisch beim Eröffnen der Datei zugewiesen.

Dynamische Pufferung: Ein Spezialfall ist die sogenannte dynamische Pufferung. Hierbei wird ein Puffer aus dem Komplex erst unmittelbar vor der Eingabe der Daten zugewiesen und direkt nach der Ausgabe wieder freigegeben, wodurch eine bessere Ausnutzung des Pufferkomplexes ermöglicht wird. Dynamische Pufferung ist bei Verwendung der einfachen indexsequentiellen Zugriffsmethode und der einfachen Direktzugriffsmethode möglich.

Austauschpufferung: Bei Verwendung der sogenannten Austauschpufferung des Systems OS/ES erfolgt bei der Verarbeitung von E/A-Daten kein Datentransport mehr. Durch Austausch der Bedeutung der Segmente (das gleiche Segment ist abwechselnd Teil eines Eingabepuffers, Arbeitsbereich und Teil eines Ausgabepuffers) liegen dann die Segmente eines Puffers, die in ihrer Gesamtheit einen physischen Block aufnehmen, nicht mehr zusammenhängend im Hauptspeicher. Die Eingabe und Ausgabe der Datenblöcke muss deshalb bei Austauschpufferung unter Verwendung der Möglichkeiten des Datenkettens erfolgen.

Als Ergänzung ist zu erwähnen, dass auch die Einrichtung des Kommandokettens vom System verwendet wird, um die Eingabe bzw. Ausgabe mehrerer Datenblöcke durch eine einzige E/A-Operation zu ermöglichen. Diese Unterstützung kann wahlweise als Dienst der Datenverwaltung angefordert werden.

Die Katalogverwaltung

Durch den Systemkatalog werden viele anfallende Arbeiten, die mit der Archivierung von Daten verbunden sind, vom System übernommen.

Der Anwender kann davon ausgehen, dass seine Datei nur durch Angabe ihres Namens eindeutig für die Verarbeitung durch das System beschrieben ist. Andere wichtige Informationen werden dann durch die Katalogverwaltung den entsprechenden Systemkomponenten zur Verfügung gestellt, wie beispielsweise die Archivnummer des Datenträgers, der Typ des Gerätes usw.

Diese Informationen werden in Verbindung mit dem entsprechenden Dateinamen in einer speziellen Systemdatei aufbewahrt, die als Systemkatalog bezeichnet wird. Diese Katalogdatei kann sich über mehrere Wechsell Plattenspeicher erstrecken. Zur Unterstützung einer logisch sinnvollen Namensgebung für Dateien bietet das System die Möglichkeit, einen Namen zu wählen, der sich aus einzelnen, bis zu 8 Zeichen langen und durch Punkte voneinander getrennten Unterbegriffen zusammensetzt. Dadurch wird die Gefahr der Auswahl doppelter Namen wesentlich herabgesetzt. Derartige Namen werden als qualifizierte Dateinamen bezeichnet. Der Aufbau der Katalogdatei in Form sogenannter Indexstufen ist diesem Unterteilungsprinzip angepasst.

Der Anwender nimmt die Dienste der Katalogverwaltung dadurch in Anspruch, dass er in einer Anweisung der Jobsteuersprache nur den Namen der Datei angibt und bewusst auf die Angaben von Geräte- und Datenträgerinformationen verzichtet: Die Programme der Katalogverwaltung durchsuchen dann den Systemkatalog nach der entsprechenden Eintragung und stellen die fehlenden Angaben den entsprechenden Systemkomponenten zur Verfügung. Der Anwender muss seine Dateien „katalogisieren“, d.h. eine Eintragung in den Systemkatalog veranlassen, bevor er die beschriebenen Dienste in Anspruch nehmen kann.

Es muss an dieser Stelle darauf verwiesen werden, dass der Begriff des Katalogisierens im Betriebssystem OS/ES damit eine völlig andere Bedeutung als im Plattenbetriebssystem DOS/ES erhält.

Kennsätze

Zur Kennzeichnung von Datenträgern und Dateien befinden sich auf dem Datenträger zusätzlich zu den Benutzerdaten noch sogenannte Kennsätze. Je nach ihrer Beschreibungsfunktion unterscheidet man

- Datenträgerkennsätze und
- Dateikennsätze.

Ausgehend vom Aufbau ist eine Unterteilung möglich in

- Standardisierte Kennsätze und
- Nicht-Standardkennsätze.

Standardisierte Kennsätze haben eine vorgeschriebene Länge und beginnen mit einer definierten Zeichenfolge. Sie werden wiederum unterteilt in System-Standardkennsätze, meist kurz als Standardkennsätze bezeichnet und Benutzer-Standardkennsätze.

Nicht-Standardkennsätze haben wählbare Länge und Inhalt. Ihre Verwendung ist nur bei Magnetband möglich.

Wechselplattenspeicher und die auf ihnen angelegten Dateien müssen Standardkennsätze enthalten. Magnetbänder können Standardkennsätze, Nicht-Standardkennsätze oder keine Kennsätze enthalten.

Wie bereits erwähnt wurde, übernehmen die Standardkennsätze im Betriebssystem OS/ES eine wichtige Funktion bei der Realisierung des Konzepts der Datenorientierung und der Flexibilität von Programmen. Standardkennsätze enthalten alle wichtigen Informationen zur Beschreibung der Datei (Blockgröße, Satzlänge, Satzformat u.a.) Weitere Funktionen der Kennsätze dienen der Identifizierung sowie dem Schutz der Dateien.

Dateischutz

Die Datenverwaltung des Betriebssystems OS/ES bietet Möglichkeiten des Schutzes der Dateien gegen unbeabsichtigtes Zerstören bzw. unbefugten Zugriff zu Dateien auf Wechselplattenspeichern und Magnetbändern.

Verfallsschutz: Vorbeugende Maßnahmen gegen unbeabsichtigtes Zerstören bietet der Verfallsschutz. Beim Aufbau der Datei kann der Benutzer über die Jobsteuersprache ein Verfallsdatum angeben. Wird diese Datei zu einem späteren Zeitpunkt für eine Ausgabe eröffnet oder soll sie überschrieben werden, so wird bei noch nicht erreichtem Verfallsdatum der Bediener vom System informiert. Dieser kann die Erweiterung bzw. das Überschreiben der Datei erlauben oder andererseits veranlassen, dass der laufende Jobschritt abgebrochen wird.

Kennwortschutz: Ein Schutz gegen unbefugte Verwendung von geheimzuhaltenden Daten wird durch das Kennwortprinzip verwirklicht. Es wird hierfür eine spezielle Kennwortdatei angelegt, in der für jede geheimzuhaltende Datei eine Eintragung vorgenommen wird, die den bis zu 44 Zeichen langen Namen der Datei mit einem bis zu 8 Zeichen langen Kennwort verbindet. Bevor der Zugriff zu den Daten ermöglicht wird, erhält der Bediener die Aufforderung, über die Bedieneinheit dieses Kennwort einzugeben. Bleibt ein zweimaliger Versuch erfolglos, so wird dem Benutzer die Verwendung der Daten durch Abbruch des laufenden Jobschrittes verwehrt. Zur Vervollkommenung dieses Prinzips kann die Kennwortdatei selbst ebenfalls durch ein Kennwort geschützt werden.

Speicherplatzverwaltung beim Wechsellattenspeicher

Zur effektiven Nutzung eines Wechsellattenspeichers ist eine sorgfältige Planung und Aufteilung seines Speichers erforderlich. Die Speicherplatzverwaltung für Direktzugriffsspeichergeräte des Systems OS/ES erleichtert diese Arbeiten wesentlich, indem sie dem Benutzer die Möglichkeit bietet, über Parameter der Jobsteuersprache Speicherplatz für den Aufbau von Dateien auf diesen Datenträgern anzufordern. Der Anwender hat dabei im Normalfall keinen Einfluss darauf, wo sich die ihm zugeordneten Bereiche auf dem Datenträger befinden. Dies ist jedoch auch nicht erforderlich, da bei der Zuordnung des angeforderten Speicherplatzes Kriterien der effektiven Belegung des Datenträgers durch die Systemkomponenten in ausreichendem Maße berücksichtigt werden. Für spezielle Anwendungsfälle kann der Bereich jedoch auch beginnend an einer bestimmten Adresse auf dem Datenträger angefordert und zugeordnet werden.

Die Anforderung dieses externen Speicherplatzes kann in Form von Spuren, Zylindern oder Vielfachem von Datenblocklängen erfolgen. Eine sogenannte Primärmenge, die bei jedem Anlegen einer Datei auf einem Wechsellattenspeicher angegeben werden muss, wird während der Vorbereitungsarbeiten zur Ausführung des Programms zugeordnet. Durch die wahlweise Angabe einer Zusatzmenge erhält die Speicherplatzverwaltung bezüglich der Planung des erforderlichen Speicherbedarfs im Hinblick auf Erweiterungen von bestehenden Dateien eine große Variationsmöglichkeit. Diese Zusatzmenge wird beim Aufbau bzw. der Erweiterung der Datei immer dann zugeordnet, wenn das Ende eines vorher angelegten Bereichs erreicht wird. Sie kann bis zu 15mal auf einem Datenträger zugeordnet werden.

Die Flexibilität der Speicherplatzverwaltung wird durch die Möglichkeit der Anforderung spezieller Dienste (Freigabe von nicht belegtem Speicherbereich, Anforderung eines zusammenhängenden Bereichs, Anforderung von Maximalbereichen u.a.) noch erhöht.

1.3.3. Serviceprogramme

1.3.3.1. Sortieren/Mischen

Um Sätze in Dateien schneller verarbeiten bzw. aufsuchen zu können, ist es oft sinnvoll, dass sie in einer bestimmten Reihenfolge angeordnet sind. Bekannte Beispiele sind Wörterbücher oder Lexika.

Mit dem Programm Sortieren/Mischen ist es möglich, Dateien nach bestimmten Sortierbegriffen zu sortieren und bereits sortierte Dateien zu mischen (zu vereinigen). Die Dateien können folgende Formate haben:

- feste Satzlänge
- variable Satzlänge

Die Organisation der Datei muss so sein, dass sie sich mit den üblichen Zugriffsmethoden lesen lässt (SAM, ISAM, BSAM). Das Einlesen ist also sowohl von Platte als auch von Band möglich. Das Einlesen der Sätze von einem Kartenleser ist möglich, hat aber eher eine theoretische Bedeutung.

Die Sortierung kann oft nicht im Hauptspeicher erfolgen, weil dieser für die meisten Dateien zu klein ist. Außerdem steht im Normalfall nur ein kleiner Teil des Hauptspeichers für das Sortieren zur Verfügung.

Daher erfolgt erst einmal ein Vorsortieren der Sätze mit Hilfe eines sogenannten Sortierbaums. Das Ergebnis sind Folgen von sortierten Sätzen, die auf einem Zwischenspeicher (Platte oder Band) abgelegt werden. Dabei ist das Ablegen auf Band im Allgemeinen wesentlich ungünstiger als das Ablegen auf Platte. Allerdings verhindert der knappe Plattenspeicherplatz manchmal ein Plattensortierverfahren. Zur Optimierung der Eingabe/Ausgabe für die Zwischenspeicher wird mit eigenen Kanalprogrammen gearbeitet. Der Zugriff auf Bänder wird dadurch optimiert, dass es möglich ist, die Dateien sowohl vorwärts als auch rückwärts zu lesen. Das spart Zeit zum Positionieren.

Es gibt verschiedene Sortierverfahren. Für den Anwender ist es allerdings schwer, das richtige Verfahren auszusuchen. Am Besten ist es, die Entscheidung dem Sortierprogramm selbst zu überlassen.

Für die Sortierreihenfolge gibt es eine Reihe von Kriterien:

- Position der Sortierfelder im Satz
- Länge der Sortierfelder
- Format der Sortierfelder
- Aufsteigend oder absteigend

Die Eingabe und Ausgabe der Sätze erfolgt normalerweise von bzw. in Dateien, gesteuert über eine DD-Anweisung.

- Eingabe : SORTIN
- Ausgabe: SORTOUT

Es ist aber auch möglich, einen Teil oder alle Sätze über spezielle Benutzerroutinen ein- bzw. auszugeben.

Beim Mischen müssen die einzelnen Eingabedateien bereits nach den Kriterien sortiert sein, nach denen dann die Gesamtdatei sortiert sein soll. Ist das an irgendeiner Stelle nicht der Fall, so erfolgt ein Abbruch des Laufs. Die zu mischenden Dateien werden durch die DD-Namen SORTINxx (xx = 01-16) ausgewählt.

Sortieren wird als Haupt- und Unterprogramm benutzt. Die Anteile sind grob 1:1.

1.3.3.2. Programmverbinder und Lader

Ausführbare Programme bestehen aus Befehlen im Maschinencode und aus Konstanten. Aus Gründen der Übersichtlichkeit werden die Programme aber im Normalfall in einem Quellcode formuliert. Der Quellcode kann in Assemblersprache, ALGOL, COBOL, FORTRAN, PL1 oder RPG vorliegen.

Mit Hilfe eines Sprachübersetzers wird der Quellmodul in einen sogenannten Objektmodul überführt. Dieser hat Sätze von fester Länge (80 Byte). Objektmodule enthalten zwar schon Befehle im Maschinencode, können aber nicht einfach vom Steuerprogramm ausgeführt werden.

Zur Überführung in die ausführbare Form dienen der Programmverbinder bzw. der Lader. Der Hauptunterschied zwischen diesen beiden besteht darin, dass das Ergebnis eines Programmverbinderlaufs als sogenannter Lademodul in einer Bibliothek abgelegt wird. Erst dieser Lademodul kann dann ausgeführt werden. Beim Lader kann das Ergebnis sofort im Hauptspeicher ausgeführt werden. Da aber das Ergebnis des Laderlaufs nicht aufgehoben wird, ist für eine erneute Ausführung auch ein neuer Laderlauf erforderlich.

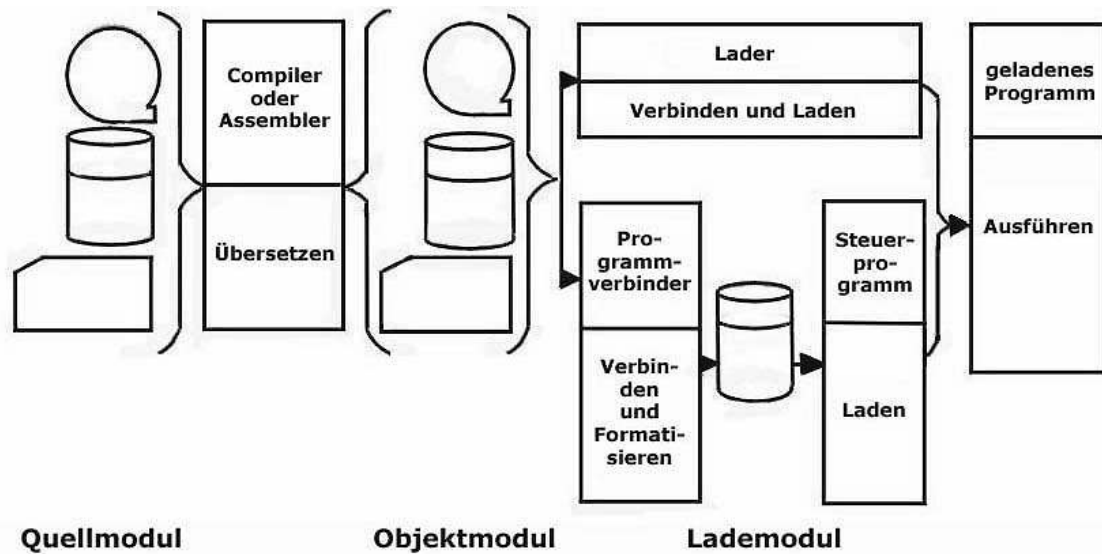


Abbildung 4 Vom Quellmodul zum ausführbaren Programm

Aus dem Bild 4 geht hervor, dass sich sowohl der Quellmodul als auch der Objektmodul auf Lochkarte, Band oder Platte befinden kann.

Der Lademodul befindet sich stets auf einer Platte.

Da der Lademodul sogenannte Verschiebeinformationen enthält, ist es möglich, den Modul an beliebige Stellen im Hauptspeicher zu laden (Mindestforderung allerdings Doppelwortgrenze) und dort auszuführen. Auch der Lader hat das Programm so aufbereitet, dass es an der Stelle ausgeführt werden kann, die ihm der Lader zugewiesen hat.

Neben dem/den beschriebenen Objektmodul(en) können als Eingabe auch bereits aufbereitete Lademodule dienen. Dadurch ist es möglich, Lademodule zu neuen Paketen zusammenzufassen bzw. einzelne Teile (Programmabschnitte) auszutauschen.

Es ist möglich Objektmodule aus Assembler- und anderen Compilerläufen zusammenzufassen. Ein Verbinderlauf besteht grob aus den folgenden Schritten:

- Überführung von Programmabschnitten aus den Eingabedateien in die Ausgabedatei
- Aktualisieren von Adresskonstanten
- Lösen externer Bezugnahmen
- Aufbau einer Eintragung im Bibliotheksverzeichnis (beispielsweise Name und Eintrittspunkt des Lademoduls).

Bei einem Programmverbinderlauf werden alle Programmabschnitte aus den Eingabedateien in den Ausgabemodul übernommen. Durch spezielle Steueranweisungen können allerdings einzelne Abschnitte davon ausgenommen werden. Auch bei Namensgleichheiten werden einzelne Abschnitte ausgeschlossen.

Im Normalfall liegen alle Programmabschnitte im Ausgabemodul so, dass sie keinen Bereich des Hauptspeichers gemeinsam belegen, sondern sind nacheinander angeordnet. Eine Ausnahme ist die sogenannte Überlagerungsstruktur, bei der Teile des Moduls für denselben Hauptspeicherplatz vorgesehen sind wie andere. Das bedeutet, dass sich solche Teile niemals gemeinsam im Hauptspeicher befinden können. Das muss vor allem bei der Logik des erzeugten Programms berücksichtigt werden.

Als Eingaben für den Programmverbinder dienen einmal die Primäreingaben und zum anderen die Sekundäreingaben.

Die primären Eingabedaten werden stets zuerst verarbeitet.

Sekundäre Eingabedaten unterteilen sich in sekundäre Einschlusseingabe und Aufrufeingabe. Dabei werden die sekundären Einschlussdaten gezielt auf der Grundlage von Programmverbinder-Steueranweisungen durchsucht. Erst, wenn dabei alle Möglichkeiten ausgeschöpft sind, werden alle bis dahin noch ungelösten externen Referenzen mit Hilfe der Einschlusseingabedateien zu lösen versucht.

In der Ausgabebibliothek wird für jeden Lademodul zumindest ein Verzeichniseintrag mit dem Namen des Moduls angelegt. Außerdem kann der Programmverbinder für den Modul bis zu 16 Aliasnamen vergeben. Unter all diesen Namen kann der Modul später aufgerufen werden. Der Eintrittspunkt beim Start des Programms kann sich außerdem von dem der einzelnen Aliasnamen unterscheiden. Es kann also mit den Aliasnamen auch eine geänderte Funktionalität des Programms (Moduls) erreicht werden.

1.3.3.3. Dienstprogramme

Das Betriebssystem OS/ES enthält eine Reihe von Programmen, durch deren Anwendung der Codieraufwand für häufig auszuführende Arbeitsgänge stark herabgesetzt wird. Diese Programme werden Dienstprogramme genannt. Je nach Funktion werden folgende Gruppen unterschieden:

Systemunabhängige Dienstprogramme

Systemunabhängige Dienstprogramme arbeiten ohne Betriebssystem. Beim Laden der Programme vom Magnetband in den Hauptspeicher werden alle zur Ausführung der Programme benötigten Betriebssystemfunktionen mit geladen.

| | |
|----------|---|
| IBCDASDI | Dient zum Initialisieren bzw. Formatieren von Datenträgern für direkten Zugriff |
| IBCDMPRS | Dient zum Ziehen oder Rückspeichern einer Reservekopie. |

Spezielle System-Dienstprogramme

Für Systemprogrammierer und das Wartungspersonal stehen einige spezielle Dienstprogramme zur Verfügung.

| | |
|----------|---|
| IEHATLAS | Auf einem Datenträger für direkten Zugriff werden Daten, die durch Datenfehler oder fehlende Adressmarken nicht lesbar sind, wieder zugänglich gemacht. |
| IEHCHCAT | Ändert den Gerätetypcode in den Eintragungen des Kataloges. |
| IEHIOSUP | Erstellt die Adressliste bestimmter Module der SVCLIB. |
| IEHSTATR | Erzeugt eine Fehlerstatistik für Magnetband |

Allgemeine System-Dienstprogramme

Die allgemeinen System-Dienstprogramme behandeln Datenmengen und Systemsteuerdaten.

| | |
|----------|--|
| IEHDASDR | Ermöglicht die Wartung von Datenträgern für direkten Zugriff. Diese Datenträger können initialisiert oder formatiert werden. Es können Kennsätze geändert, Ersatzspuren zugeordnet und eine Reservekopie gezogen oder rückgespeichert werden |
| IEHINITT | Initialisiert Magnetbänder. |
| IEHLIST | Druckt Dateisteuerblöcke eines VTOC, Katalogeintragungen von Dateien oder Verzeichnisse untergliederter Dateien. |
| IEHMOVE | Eine Datei, eine Gruppe katalogisierter Dateien, ein Katalog oder der Inhalt eines Datenträgers wird von dem Quell- auf einen Zieldatenträger kopiert oder umgespeichert. Beim Umspeichern wird die Quelldatei gelöscht. |
| IEHPROGM | Katalogisiert oder entkatalogisiert Dateien, löscht Dateien oder Bestände, ordnet Dateien oder Beständen einen neuen Namen zu, vergibt und löscht Kennworte für Dateien und führt verschiedene Katalogfunktionen aus. |

Datei-Dienstprogramme

Die Dateidienstprogramme bearbeiten Dateien, indem sie die Daten umorganisieren, austauschen oder vergleichen.

| | |
|----------|---|
| IEBCOMPR | Vergleicht sequentielle oder untergliederte Dateien miteinander. Bei Verwendung einer Benutzeroutine können auch Satzteile von sequentiellen oder untergliederten Dateien verglichen werden. |
| IEBCOPY | Kopiert, lädt oder entlädt untergliederte Dateien. Dabei können einzelne Bestände ersetzt, ausgewählt oder ausgeschlossen werden sowie der Speicherplatz vergrößert oder verkleinert und umgeblockt werden. |
| | Untergliederte Dateien können am Ort verdichtet werden, so dass sie danach keinen unbenutzten Platz zwischen ihren Beständen enthalten. |
| IEBDG | Erzeugt sequentielle, untergliederte oder indexsequentielle Dateien. |
| EDIT | Erstellt einen Jobeingabestrom |
| IEBGENER | Ermöglicht das Kopieren und/oder Aufbereiten von Sätzen einer sequentiellen Datei oder eines Bestandes einer untergliederten Datei. |
| | Dient zum Umwandeln einer sequentiellen Datei in einen Bestand einer untergliederten Datei. |
| IEBISAM | Ermöglicht das Kopieren, Drucken und Umorganisieren (Laden oder Entladen) von indexsequentiellen Dateien. |
| IEBPTPCH | Druckt oder stanzt Sätze einer sequentiellen oder untergliederten Datei. |

| | |
|----------|---|
| IEBUPDTE | Kann einer untergliederten Datei Bestände hinzufügen, ersetzen, kopieren oder ihnen einen Aliasnamen zuordnen. |
| | Weiterhin können logische Sätze einer sequentiellen Datei oder einem Bestand hinzugefügt, ersetzt, gelöscht oder neu nummeriert werden. |
| | Das Programm kann einen Bestand einer untergliederten Datei in eine sequentielle Datei überführen und umgekehrt. |
| IEBMFCNV | Wandelt sequentielle Dateien in Mikrofilmdateien um. |
| IEBMFTRF | Setzt magnetbandgespeicherte Dateien auf ein Mikrofilmausgabegerät um. |
| IEBPRINT | Druckt Sätze einer bzw. zweier Dateien auf zwei Druckbahnen (speziell für Drucker 7031) |

1.3.3.4. Das Dialogsystem „Teilnehmerunterstützung TSO“ ab OS/ES 6.1

TSO ermöglicht die Bereitstellung von Jobströmen vom Abonnentenpunkt aus und kann deshalb die veraltete Lochkartentechnik ablösen. Die Eingabe der Anweisungen und deren Korrektur erfolgt dabei von einem Bildschirm oder einer Schreibmaschine aus. Der Ablauf und die Ergebnisse der erstellten Programme im Rechner kann über TSO gestartet und weitgehend kontrolliert werden. Eine Testung dieser Programme kann durch die vorhandene Testhilfe (→ TESTRAN) vereinfacht durchgeführt werden.

Die Nutzung des viel komfortableren TSO (gemessen an CRJE) setzt die Erzeugung eines TCAM-MCPs (zentrales Steuerprogramm) voraus. Dieses ist von Systembetreuern auf der Basis von Generierungs-Makros zu erstellen und vor dem TSO-Start zu starten. Die Erstellung des MCP's ist relativ aufwendig und durch sehr umfangreiche Parametrierung in den Generierungs-Makros recht unübersichtlich. Erst die Bereitstellung gewisser vorbereiteter Standardgenerierungen dieser MCP's vereinfacht dieses Problem.

Mit der umfassenden Bereitstellung des Bildschirmsystems EC 7920 wird das TSO das meist genutzte Dialogsystem im OS/ES.

Als E-/A-Geräte für die TSO-Nutzung können alle Abonnentenpunkte (Bildschirmgeräte-, Schreibmaschinen-AP und Bürocomputer mit entsprechender Emulation) verwendet werden, soweit sie durch die Zugriffsmethode TCAM unterstützt sind (siehe Abschnitt Fernverarbeitung). Am häufigsten genutzt ist die nahaufgestellten Variante des Bildschirmsystems EC 7920 – EC 7922/EC 7927.

Das TSO steht zum TCAM-MCP in einem ähnlichen Verhältnis wie beliebige Anwenderprogramme, die dieses MCP direkt benutzen; Diese können bei entsprechender MCP-Generierung auch parallel zu TSO arbeiten.

Die Datenströme vom AP fließen über Warteschlangen zum MCP und von diesem wieder über andere Warteschlangen zum TSO bzw. dem entsprechenden Anwenderprogramm und umgekehrt. Genauso erfolgen der Transport und die Steuerung der Nachrichten, die ausgetauscht werden müssen, um den TSO-Betrieb zu ermöglichen.

TSO ist eine zusätzliche Funktion des Betriebssystems (später wurde sie fest mit dem System ausgeliefert), die die gleichzeitige und von einander unabhängige Nutzung der Möglichkeiten des Bildschirms von Datenstationen aus gestattet. Dabei werden die Ressourcen der Zentraleinheit (CPU) in Zeitscheiben den einzelnen Nutzern zur Verfügung gestellt. Eine Zeitscheibe muss dabei so bemessen sein, dass ein sinnvoller Teil der Arbeiten durchgeführt werden kann, andererseits darf sie nicht zu groß sein, um eine angemessene Antwortzeit für alle Teilnehmer zu ermöglichen.

TSO arbeitet mit dem Betriebssystem in 2 Regionen zusammen. In der TSO-Steuerregion übernimmt die TSO-Steueraufgabe die Initialisierungsprozesse, die Verbindung zum Nachrichtensteuerprogramm und ruft die Regionssteueraufgabe auf, die die Entaktivierung und Aktivierung der TSO-Jobs vor und nach dem Austausch durchführt.

Jedem aktivem Teilnehmer wird während seiner Sitzung ein TSO-Job zugeordnet, der in einer bestimmten TSO-Region abläuft. In der TSO-Region ist der Teilnehmer-disponent (Verarbeitung des LOGON-Kommandos) aktiv, der das Monitorprogramm aufruft, das wiederum die Kommandoverarbeiter bzw. Anwenderprogramme aktiviert. Kommt es in diesen zu Fehlersituationen, übernimmt wieder das Monitorprogramm die Steuerung. Der TSO-Job wird nicht wie bei der Stapelverarbeitung abgebrochen, sondern der Teilnehmer erhält die Möglichkeit der Ursachenprüfung. Daneben wird auch eine Endlosschleife im Programm nach einer im MCP vereinbarten Zeit (time out) unterbrochen und der Teilnehmer kann an der Unterbrechungsstelle sein Programm untersuchen.

Da im TSO-Betrieb eine Vielzahl von Teilnehmern gleichzeitig arbeitet, bestehen erhöhte Anforderungen an den Daten- und Programmschutz. Teilnehmer ohne Benutzungserlaubnis (Teilnehmerattributdatei mit Sitzungsanforderungen, wie z.B. Kennwort, Abrechnungsnummer) werden abgewiesen, Programme und Dateien eines Teilnehmers müssen vor einer zufälligen Zerstörung durch andere Teilnehmer geschützt (alle anderen Teilnehmer sind in einer Seitendatei ausgelagert), vertrauliche Daten gegen unerlaubten Zugriff gesichert werden (Lese-/Schreibschutz über Kennworte).

Durch Eingabe eines TSO-Kommandos von seiner Datenstation fordert der Teilnehmer eine bestimmte Funktion an. Fehlerhafte Eingaben können nach einer Aufforderungsnachricht korrigiert werden. Kommandos dienen zum

- Eröffnen, Überwachen und Beenden einer Sitzung,
- Bearbeiten von Dateien mit Aufbau und Korrektur von Texten, Daten, Jobsteueranweisungen,
- Übersetzen, Verbinden, Testen und Ausführen von Programmen sowie
- Übergeben von Jobs an die Stapelverarbeitung von der Datenstation aus, wobei die Ergebnisse zur Auswertung an die Datenstation oder an den Drucker weitergeleitet werden.

Um häufig benötigte Folgen von Kommandos zu unterstützen, bietet TSO die Nutzung einer Kommandoprozedursprache. Der Aufruf von Dienst- und Systemhilfsprogrammen kann so vereinfacht werden.

Auch die Kommunikation zwischen den Teilnehmern und zwischen Teilnehmer und Operator wird durch Kommandos unterstützt.

Gegenüber der Stapelverarbeitung führt TSO bei der Programmentwicklung an der Datenstation zu einer erheblichen Effektivitätserhöhung. Neben der Eingabe der Quelltexte und deren Korrektur durch den Programmierer selbst (die Locherin wird arbeitslos) kann der Teilnehmer die Übersetzung durch den Assembler und alle im Betriebssystem implementierten Compiler während der Sitzung ausführen lassen oder an die Stapelverarbeitung übergeben. Übersetzungsfehler werden schnell erkannt und können korrigiert werden. Die für die Ausführung von Programmen notwendigen Dateien können dynamisch zugeordnet werden und das Programm getestet werden. Hierfür stellen die Compiler entsprechende Dialogtester bereit, für Assemblerprogramme das TEST-Kommando von TSO.

Diese Testmöglichkeiten bieten das

- Festlegen von Unterbrechungspunkten im Voraus,
- Automatisches Ausführen von Funktionen an Unterbrechungspunkten,
- Anzeigen des Inhaltes von Registern, Steuerblöcken und von Hauptspeicherbereichen,
- Modifizieren von Registern und Hauptspeicherbereichen
- Fortsetzen der Arbeit des Programms an einem gewünschten Befehl

Damit werden die Fehlerbeseitigung und die Änderung auch bestehender Programme wesentlich erleichtert.

Die Bereitstellung der Teilnehmerunterstützung führt zu einem Quantensprung in der Entwicklung und Wartung von Programmen.

1.3.3.5. System zur Entwicklung und Ausführung von Programmen im Dialog (DSPF)

Die zeilenweise Eingabe der TSO-Kommandos mit den Operanden an den entsprechenden Positionen wurde mit der Zeit eintönig und schwerfällig, so dass an einem **System zur Entwicklung und Ausführung von Programmen im Dialog (DSPF)** gearbeitet wurde, das Routineabläufe vereinfachte. Wesentlicher Bestandteil von DSPF ist neben der Dialogverwaltung die Programmierunterstützung PDF. DSPF arbeitet als Subsystem unter der Teilnehmerunterstützung TSO in den Betriebssystemen SVS und MVS.

Die zeilenweise Eingabe wird ersetzt durch sogenannte Kurzkommandos auf den verschiedenen Funktionsebenen bzw. durch Nutzung des gesamten Bildschirms. Die Verständigung des Nutzers mit PDF erfolgt über Paneele. Sie bieten

- die Texteingabe über den gesamten Bildschirm,
- das Editieren und Anzeigen (BROWSE) von Dateien und Beständen,
- das Verschieben des Bildschirmbildes über Quelldaten oder Ergebnisprotokolle,
- die Teilung des Bildschirms in zwei logisch getrennte Arbeitsbereiche,
- die Nutzung von Dienstprogrammen zur Datei- und Bibliotheksbearbeitung,
- den Aufbau eigener Dialoge und deren Test,
- den versionsbezogenen Aufbau von Bibliotheken,
- den Aufruf von Compilern durch TSO-Vermittler,
- die Nutzung der Lernhilfe über das Subsystem.

Die Dateneingabepaneele bieten den Nutzern die Möglichkeit, in gekennzeichneten Feldern die notwendigen Informationen zu übergeben, wie z.B. Bibliothek, Typ, Bestand oder Name der sequentiellen Datei. Andererseits kann der Dateiname auch aus einer angezeigten Liste mit dem entsprechenden Kurzkommando ausgewählt werden. Ebenso können Bestände untergliederter Dateien aus einer Liste mit Kurzkommandos ausgewählt werden. Daneben können aber auch die TSO-Kommandos in ihrer ausführlichen Form genutzt werden.

Beim Editieren oder Anzeigen von Texten können diese auch in hexadezimaler Form dargestellt bzw. auf entsprechende Zeichenketten durchsucht werden. Besondere Unterstützung erfuhr das Editieren, so z.B. Finden und Ersetzen von Zeichenketten, Kopieren oder Übertragen von Textteilen, Übergeben an die Stapelverarbeitung als Job, maskenweise Bearbeitung von Zeilen.

Weiterhin können die Programmfunktionstasten der Datenstation mit PDF-Kommandos belegt werden, um solche Funktionen wie Teilen des Bildschirmes, Verschieben des Bildschirmbildes, usw. direkt auszuführen.

PDF bietet auch die Möglichkeit der Erzeugung eines Sitzungsprotokolls.

Durch die neuen Möglichkeiten der Programmentwicklung wird eine erhebliche Zeitersparnis erreicht, das Testen von Programmen erleichtert und die Effektivität erhöht. Viele der im TSO erprobten Möglichkeiten fanden später Eingang in das Betriebssystem.

1.3.3.6. OLTEP (Prüfprogramm für externe Geräte)

OLTEP ermöglicht den Test von externen Geräten im Multiprogrammbetrieb. Die Komponente besteht aus voneinander abhängigen Modulen und Gerätesektionen. Die Module ermöglichen die Kommunikation mit dem Wartungsingenieur, die Auswahl von Testvarianten und die Ausgabe der Testergebnisse. Die Gerätesektionen führen die gewünschten Tests der E/A-Geräte aus und übergeben die Ergebnisse an die Module.

1.3.3.7. TESTRAN

TESTRAN ist eine Testhilfe für Assemblerprogramme. Spezielle Systemmakros analysieren den Programmablauf.

1.4. Unterstützung der Datenfernverarbeitung für EC 1040 in den Betriebssystemen OS/ES 4.1 und 6.1

Von der Gerätetechnik her muss unterschieden werden, dass es Geräte (Bildschirm- und Schreibmaschinengeräte) gab, die direkt über den Multiplexkanal (1) oder über einen Multiplexor / später DFV-Prozessor (2) angesteuert wurden. Für den Fall (1) waren nur kurze Übertragungswege möglich. Alle DFV-Zugriffsmethoden und damit auch die Anwendersysteme wie TSO und CRJE, die diese Zugriffsmethoden benutzten, können aber beide Gerätearten bedienen.

In den Betriebssystemen des OS/ES 4.1 gibt es zunächst nur eine Unterstützung durch die graphische Zugriffsmethode GAM für das einzige zur Verfügung stehende Bildschirmgerätesystem BSS-ESER, später EC 7920, dessen Steuergerät nur am Multiplexkanal anschließbar ist.

Vorhandene Hardware:

- Fernschreibtechnik (T34)
- Bildschirmsysteme auf Basis von Start/Stop-Prozeduren EC 8562/EC 8564 analog IBM 2260
- IBM-kompatibles Bildschirmsystem EC 7910/EC 7920 analog der IBM 3240 (bis zu 32 BS-Arbeitsplätze) mit Steuereinheiten für den Nahbetrieb (Anschluss an Multiplex- oder Selektorkanal), Fernbetrieb über Modems unter Benutzung der Synchronprozedur BSCIII/CHX3 (Stand Software- und hardwaremäßig ab Erscheinen von OS/ES 6.1 zur Verfügung)

MPD4 / EC8404:

In der Folge der Entwicklung des Kleinststeuerrechners KRS4200 wird auf dessen Basis der Multiplexor EC8404 bereitgestellt. Am Multiplexkanal der Rechenanlage angeschlossen, werden bis zu 12(16) Fernleitungen bzw. Direktleitungen, abhängig von Übertragungsprozedur und Übertragungsgeschwindigkeit, bedient (gesteuert).

Der Multiplexor unterstützte folgende Übertragungsprozeduren:

Start/Stop-Prozeduren:

1. entsprechend EC8562/EC8564,
2. Fernschreibtechnik,
3. Schreibmaschinenprozeduren analog EC8501

Basis-Synchronprozeduren:

1. für Mehrpunktgeräte BSCIII/CHX3
2. für Punkt-zu-Punkt-Geräte BSCI/CHX1 - Diese Prozedur wird zwar im Betriebssystem OS/ES 4.1. unterstützt, eine Nutzung erfolgt lange wegen der nicht vorhandenen Gerätetechnik nur bedingt. (Dazu muss Gerätetechnik, die Dateiübertragung zulässt, vorhanden sein. Schreibmaschinen oder Bildschirme können das nur bedingt für kleinste Datenmengen)

TAM 601:

Von der VR Ungarn wurde etwa zur gleichen Zeit das Modem TAM 601 zur Verfügung gestellt, welches eine maximale Übertragungsgeschwindigkeit bis zu 1200 Bit/s bei Benutzung der Start-Stop- oder einer Synchron-Übertragungsprozedur ermöglichte.

Es war Duplex- und Halbduplexbetrieb möglich. Infolge des Engpasses von Übertragungsleitungen in der DDR kam fast immer nur der Halb-Duplexbetrieb (2-Drahtbetrieb) in Betracht. Nachteilig wirkte sich aus, dass die Umschaltung von Senden auf Empfangen und umgekehrt durch Relais bewirkt wurde (lange Umschaltzeit).

Weiterhin standen einige Schreibmaschinen-Abonnentenpunkte (z.B. EC 8570, EC 8501 mit Start/Stop-Prozedur) und Bildschirm-Abonnentenpunkte EC 8564 (EC 8562) (VT 620) zur Verfügung

EC 8564 /EC 8562

Diese in der VR Ungarn gebauten Start/Stop-Abonnentenpunkte, die in der DDR in größerer Menge eingesetzt wurden, wurden entsprechend der IBM 2260 unterstützt und bestanden aus einer Steuereinheit, bis zu 6 BS-Geräten und einem Drucker.

Es stehen dann zunächst im OS/ES folgende Nutzungsmöglichkeiten für diese Geräte zur Verfügung:

Zugriff über die Einfache Zugriffsmethode für Datenfernübertragung BTAM (basic access method)

Die einfache Zugriffsmethode für Datenfernverarbeitung BTAM unterstützt die technischen Mittel für System-Datenfernverarbeitung des ESER und lokale Bildschirmsysteme. Zur System-Datenfernverarbeitung gehören Multiplexsteuergeräte und Datenstationen, die sowohl an Standleitungen als auch an Wählleitungen angeschlossen sein können. Datenstationen können entweder mit dem Start-Stop-Übertragungsverfahren oder mit dem synchronen Übertragungsverfahren arbeiten, bei dem keine Netzprotokolle benutzt werden. BTAM unterstützt die Steuerung des Datenempfangs bzw. der Datenübertragung zwischen Zentraleinheit und den Datenstationen und führt Dienste für die Geräteeinrichtungen der DFV-Leitung aus. Programme, die auf der Basis von BTAM arbeiten, werden mit Ausnahme von CRJE nicht zur Verfügung gestellt. Die Programmierung ist besonders aufwendig, weil alle gewünschten geräteabhängigen Funktionen, wie z.B. Zeilenpositionen auf dem Bildschirm, im Anwenderprogramm programmiert werden müssen.

Das Jobferneingabeprogramm CRJE (control remote job entry) auf der Basis der Zugriffsmethode BTAM

Dieses Programm ermöglicht es, von Abonnentenpunkten (Terminals) aus Jobs einzugeben, abarbeiten zu lassen und das Ergebnis zu kontrollieren und auszugeben. Nachteilig ist vor allem die Verarbeitung der Jobs über eine einzige zentrale Datei, die CRJE.ACTIVE, die sich zusammenhängend auf einer Platteneinheit befinden muss. Beim damaligen Stand der Plattentechnik und bei Ausschöpfung der relativ hohen Anzahl gleichzeitig angeschlossener Nutzer führte das zu übermäßiger Belastung der entsprechenden Platteneinheit!

Bildschirmsystem EC7922/EC7927

Am Multiplexkanal (Nahbetrieb) über TSO/TCAM ab OS/ES 6.1 → häufigste Verwendung

Zugriff über Erweiterte Zugriffsmethode für Datenfernübertragung TCAM (telecommunication access method) ab OS/ES 6.1

Kern der Zugriffsmethode TCAM ist das zentrale Nachrichtensteuerprogramm **MCP** (message control program), welches sowohl die Steuerung der Datenendgeräte wie auch die Weiterleitung der eingegeben Daten/Programme zur Verarbeitung an Zentralrechner sowie die dort fertiggestellten Ergebnisse an die Datenstationen bzw. die Drucker und andere Medien übernimmt. Das geschieht jeweils über Warteschlangen zwischen dem MCP und den Datenendgeräten sowie zwischen MCP und Zentraleinheit.

Die Erweiterte Zugriffsmethode für Datenfernverarbeitung mit erweiterter Netzfunktion TCAM unterstützt die technischen Mittel der Systemdatenfernverarbeitung sowie lokale Bildschirmsysteme. Die Mittel des TCAM steuern den Nachrichtenfluss zwischen den Datenstationen und den Anwenderprogrammen in einem DFV-Netz.

1.5. Compiler für Programmiersprachen

Algol 60

Algol 60 wurde mit dem Ziel entwickelt, für (rein-)mathematische Problemstellungen eine problemorientierte Ausdrucksmöglichkeit (Sprache) zu haben. Anliegen der Entwickler war, die allgemein bekannte Symbolik der Mathematik dafür zu Grunde zu legen und darüber hinaus Ausdrucksmittel zur eindeutigen Beschreibung eines Algorithmus (*algorithmic language*) zu geben. Die Sprache sollte eine einfache Struktur haben und maschinelle Umsetzung in Programme für Rechenanlagen erlauben.

Auf Grund der eingangs erwähnten Zielsetzung (orientiert auf mathematische Probleme) wurde zunächst geringes Augenmerk auf die Formulierung kommerzieller Anwendungen (die sich auszeichnen durch Massendaten-Verarbeitung sowie hohen Anteil an Eingabe- und Ausgabe-Operationen) gelegt.

Die Implementierung von Algol 60 für das OS/ES hatte diesbezügliche Erweiterungen, konnte sich aber gegenüber den im OS/ES außerdem angebotenen, konkurrierenden Programmiersprachen auf Dauer nicht durchsetzen und wurde in den späteren, nachfolgenden Ausgaben des OS/ES aus dem Angebot genommen.

Fortran IV

FORTTRAN IV (formula translating system) wurde noch vor Algol 60 in den Jahren 1954 bis 1957 in den USA für den IBM-Rechner 704 entwickelt und in den Folgejahren mehrfach verbessert, erweitert und standardisiert (USAS X 3.10 -1966) und hat danach seinen Siegeszug mit FORTRAN IV angetreten.

Die Programmiersprache FORTRAN IV wird für die Programmierung von Problemen in den Naturwissenschaften und Technik mit Erfolg eingesetzt, kann aber auch in der kommerziellen Datenverarbeitung verwendet werden.

FORTTRAN-Programme sind im Allgemeinen portabel, können in der Quellform meist problemlos auch auf Rechnern, für die sie ursprünglich nicht entwickelt wurden, zum Laufen gebracht werden.

COBOL

Die vorgenannten Programmiersprachen Algol 60 und FORTRAN IV wurden zur Lösung vorwiegend wissenschaftlicher Problemstellungen geschaffen. Typisch für solche Probleme sind in der Regel umfangreiche Berechnungen auf der Grundlage weniger Eingabedaten. Für den Einsatz von EDVA in Wirtschaftsbereichen besteht die Herausforderung darin, umfangreiche Datenbestände zu verarbeiten (Sortieren, Datensätze auswählen, Datensätze neu gruppieren, Summieren, ...) wobei der Anteil an Berechnungen in der Regel vergleichsweise gering ist.

COBOL (Common Business Oriented Language) ist diesen Problemstellungen angepasst. Die erste Veröffentlichung von COBOL erfolgte im April 1960; im Jahre 1974 erfolgte die Herausgabe des Standards für COBOL ANS-74.

COBOL ist seither weit verbreitet als *die* problemorientierte Programmiersprache für Aufgabenstellungen auf kommerziellem Gebiet.

COBOL-Programme zeichnen sich durch eine strenge, rein äußerlich erkennbare Struktur aus. Sie bestehen aus Hauptteilen (DIVISION), jeder Hauptteil ist gegliedert in Kapitel (SECTION), Kapitel sind unterteilt in Paragraphen.

RPG/AUTOREPORT

Um lauffähige Programme für RPG/AUTOREPORT zu entwickeln bedarf es keiner besonderen Programmiersprache mit Syntax und Semantik (wie dies beispielsweise Algol 60, COBOL, FORTRAN oder PL1 auszeichnet) die damit eine weitestgehend „freie“, nicht formatgebundene Formulierung des Verarbeitungsalgorithmus und aller darüber hinaus notwendigen Beschreibungen erlauben. Dem gegenüber basiert RPG/AUTOREPORT auf einem Satz von RPG-typischen Formularen, die in der Art und Weise wie Fragebögen auszufüllen sind, und mit den erforderlichen Steuerungsinformationen für den Generator zu versehen sind. Die Formularköpfe enthalten Beschreibungen und unterstützen damit die Programmierung.

Im Ergebnis erhält man eine Menge von Datensätzen (jeder Datensatz eine Formularzeile, diese hat 80 Zeichen – ursprünglich Sätze auf 80-spaltigen Lochkarten!), die nur noch in die richtige Reihenfolge zu bringen sind, damit der Programm-Generator RPG daraus ein fehlerfreies Programm erzeugen kann.

RPG ist zugeschnitten auf Massendaten-Verarbeitung; Eingabe/Ausgabe auf Datenträger (Magnetplatte, Magnetband) Auswahl von Daten, Durchsuchen von Dateien, Erzeugen von Drucklisten, das unterstützt in geeigneter Weise die Zusatz-Komponente AUTOREPORT. Problemlos ist der Anschluss des Systemprogramms „Sortieren/Mischen“.

PL1

PL1 (Programming Language 1) wurde in den Jahren 1963 bis 1966 von einer Arbeitsgemeinschaft von EDVA-Nutzern und IBM-Vertretern entworfen. Zielstellung war die Schaffung einer neuen Programmiersprache, welche Algol 60, COBOL und FORTRAN IV ersetzen sollte und darüber hinaus auch Sprachelemente enthalten sollte, die die zukünftige Entwicklung der Rechentechnik und Anwenderforderungen berücksichtigen.

Die Aufgabenstellung wurde erfüllt auch hinsichtlich einer weitestgehenden Maschinenunabhängigkeit der Sprachelemente; es entstand eine „große, mächtige“ Sprache, die alle Anwendungsgebiete der o.g. „alten“ Sprachen abdeckt.

Folgende besonderen Eigenschaften zeichnen die Implementierung des PL1-Normalcompilers gegenüber den anderen im OS/ES verfügbaren Compilern aus:

- bequeme, problemgerechte Behandlung der Eingabe und Ausgabe von Daten,
- Mannigfaltigkeit der im Datenverarbeitungsprozess bearbeitbaren Daten und Datengruppierungen,
- Vorhandensein einer Vielzahl von Standardfunktionen,
- Universalität bei der Speicherplatzverwaltung in Verbindung mit einem Blockkonzept ermöglicht es, sich allen Problemstellungen bestmöglich anzupassen,
- Unterstützung des Mehraufgabenbetriebs,
- Implementierung einer PL1-Makrosprache sowie
- gute Unterstützung bei der Fehlersuche (Syntaxfehler, Programmtest).

2. EC 1055, 1055M, EC 1056, EC 1057 im OS/ES

2.1. Historisches

Mit der Auslieferung der ersten Maschinen der ESER Reihe II (bei Robotron der EC 1055) erfolgte die Auslieferung des Betriebssystems OS/ES 6.1. Es enthielt die Betriebssystemvarianten MVT, SVS und BPS sowie eine etwas eingeschränkte Variante des SVM, wovon MVT vor allem als Funktionserweiterung und Fehlerkorrektur des OS/ES für die weiter vorhandene ESER Reihe I gedacht war, aber auch auf ESER Reihe II-Maschinen lief.

Mit der Auslieferung weiterer Modelle von Maschinen der ESER Reihe II (bei Robotron der EC 1056) erfolgte die Auslieferung des Betriebssystems OS/ES 7.1. Es enthielt die Betriebssystemvarianten SVS, SVM und BPS sowie später MVS.

Bei den bis zu diesem Zeitpunkt ausgelieferten Systemen wurden Magnetbänder immer noch mit dem gesamten DLIB-Material sowie ein STARTER-Betriebssystem mitgeliefert. Daraus konnte sich der Kunde, nach seinen Wünschen konfiguriert, ein MVT- oder SVS-Betriebssystem herstellen. Die große Variationsbreite an Generierungsmöglichkeiten führte aber zu Unübersichtlichkeiten bei der Herstellung des Generierungsjobstroms so wie auch zu Generierungsergebnissen, die eine teilweise fehlerhafte Arbeitsweise des Betriebssystems erzeugen konnten.

Ab der Auslieferung des Betriebssystems OS/ES 7.1 entschloss man sich deshalb für die Bereitstellung vorkonfigurierter Systeme. Damit wurde eine wesentliche Vereinfachung der vorher aufwändigen Generierung erreicht und eine Senkung der Fehlerquote bei der folgenden Nutzung erreicht. Dies geschah natürlich auch durch Nutzungseinengung auf das in der Dokumentation beschriebene Funktionsspektrum.

Mit diesem System wurden dann neben dem SVS noch die beiden Betriebssysteme SVM und BPS ausgeliefert. Die Betriebssysteme MFT und MVT standen nicht mehr zur Verfügung.

Mit der Bereitstellung des OS/ES 6.1 begann auch die Auslieferung der Teilnehmerunterstützung TSO, mit deren Hilfe dann die Datenerfassung/-verwaltung und die Aufgabenverwaltung durch den Nutzer über Bildschirmtechnik gesteuert werden konnte und der Einsatz der Lochkartentechnik nach und nach abgelöst werden konnte. Damit war das TSO dann, da das Betriebssystem MVT weiterhin zur Verfügung stand, auch für die Rechner der ESER Reihe I nutzbar.

Mit der Bereitstellung des OS/ES 7.1 standen dann auch wesentliche Verbesserungen der Teilnehmerunterstützung TSO zur Verfügung.

Letzte Stufe der Betriebssystementwicklung war die Entwicklung des MVS (System mit mehreren virtuellen Adressräumen), das sich Ende 1989 in der Testphase befand, aber wegen der Auflösung des Kombines Robotron nicht mehr ausgeliefert wurde.

2.2. Charakteristik

Betriebssysteme für ESER Reihe II → EC1055, EC1055M, EC1056

Betriebssysteme für ESER Reihe III → EC1057

SVS, SVM, BPS waren für die Arbeit von Maschinen der ESER Reihe II/III konzipiert.

| Betriebs-system | benutzte Maschinenart und charakteristische Eigenschaften | Prinzip der Hauptspeicherverwaltung |
|-----------------|--|---|
| SVS | System mit einem virtuellen Adressraum SINGLE VIRTUAL STORAGE | |
| | SVS arbeitet auf realen Maschinen der ESER Reihe II (Voraussetzung ist die Erweiterung des PSWs der ESER-Maschinen). | Der HS wird scheinbar auf die max. adressierbare Größe von 16MB erweitert. Dabei werden die nicht gerade gebrauchten Programmteile auf externem Speicher zeitweilig ausgelagert. Der Austausch erfolgt in einer sogenannten Seitengröße von 4KB. Dieses Prinzip löst auf effektivere Art das aus dem MVT bekannte Hauptspeicher –Rollin- / Rolloutverfahren zur Hauptspeicherauslagerung ab. Die Programmabarbeitung mittels Überlagerungsstrukturen wird ebenfalls unnötig (alte Programme bleiben lauffähig). |

| Betriebs-system | benutzte Maschinenart und charakteristische Eigenschaften | Prinzip der Hauptspeicherverwaltung |
|-----------------|--|--|
| BPS | Stapelverarbeitungssystem für virtuelle Maschinen BATCH PROCESSING SYSTEM | |
| | Das Stapelverarbeitungssystem BPS ist für die Arbeit auf virtuellen Maschinen vorgesehen. Es stellt eine direkte Weiterentwicklung des OS/ES 6.1 dar, wobei verschiedene Funktionen vom Steuerprogramm des SVM übernommen und Leistungserhöhungen erreicht werden. Bedingt durch die Arbeit auf virtuellen Maschinen können gleichzeitig mehrere BPS auf einer EDVA betrieben werden. Die prinzipielle Arbeitsweise von BPS ist mit der eines SVS auf einer virtuellen Maschine vergleichbar, wobei jedoch eine engere Verflechtung mit dem SVM vorhanden ist. | BPS unterstützt einen adressierbaren virtuellen Speicher von maximal 16 MBytes, realisiert jedoch keine eigene Virtualisierung des Hauptspeichers. Durch die Möglichkeit der parallelen Arbeit von mehreren BPS auf einer EDVA kann der für den Nutzer verfügbare virtuelle Speicher entsprechend der Leistungsfähigkeit der EDVA vergrößert werden. |

| Betriebs-system | benutzte Maschinenart und charakteristische Eigenschaften | Prinzip der Hauptspeicherverwaltung |
|-----------------|---|---|
| MVS | System mit mehreren virtuellen Adressräumen MULTIPLE VIRTUAL STORAGE | |
| | MVS arbeitet auf realen Maschinen der ESER Reihe III (Voraussetzung ist die Erweiterung des PSWs der ESER-Maschinen). Es nutzt die Möglichkeiten des virtuellen Speichers bis zu einer Größe von 16 MByte (24-Bit-Adressen). Dabei wird im Unterschied zum SVS jedem Nutzer ein solcher Adressraum zur Verfügung gestellt. Bei SVS steht dieser Adressraum allen Nutzern zusammen nur einmal zur Verfügung. Charakteristisch für das MVS ist die starke Einbindung des TSO in das Steuerprogramm. Prinzipien des TSO-Dispatchers werden für das Gesamtsystem übernommen. Damit werden TSO-Teilnehmer und Stapeljobs gleichberechtigt behandelt. | Die Adressierung der einzelnen Adressräume geschieht über die so genannte Segmenttabelle. Diese Technik isoliert die einzelnen Adressräume und damit die Nutzer voneinander. Damit ist ein hohes Maß an Datensicherheit garantiert. Jedem Nutzer steht ein eigener virtueller Adressraum von 16 MB zur Verfügung. Er sieht den Speicher der anderen Anwendungen nicht und hat das Gefühl, alleiniger Nutzer der Anlage zu sein. |

2.3. Komponenten

2.3.1. Übersicht

| Steuerprogramm | |
|--|---|
| Anfangsprogrammlader | Dienen zum Starten des Betriebssystems |
| Programm zur Initialisierung des Steuerprogrammkerne | |
| Jobverwaltung | Überwachung der Auftragsabwicklung und Auftragssteuerung |
| Jobdisponent | Initiieren des nächsten Verarbeitungsprogramms und Beenden des Verarbeitungsprogramms nach der Abarbeitung |
| Nachrichtendisponent | Kommunikation des Betriebssystems mit dem Bediener |
| Aufgabenverwaltung (Supervisor) | Überwachung der gesamten Arbeit des Betriebssystems (beispielsweise Steuerung des Multiprogrammbetriebs, Unterbrechungsüberwachung ...) |
| Datenverwaltung | Überwachung der Dateien im Betriebssystem; Transport von Daten zwischen peripheren Geräten und dem Hauptspeicher |
| Programme der Fehlerverwaltung | Erfassen, aufzeichnen und ausdrucken von Fehlerzuständen |

| Serviceprogramme | |
|---------------------------------|---|
| Dienstprogramme | Für häufig benötigte Funktionen der Dateiverarbeitung stehen Dienstprogramme zur Verfügung, z.B. zum Löschen, Umbenennen, Ausdrucken oder Verändern von Dateien |
| Programmverbinder | Einzel übersetzte Programme werden zu einem ausführbaren Programm zusammengefügt und in einer Datei gespeichert |
| Lader | Einzel übersetzte Programme werden zu einem ausführbaren Programm zusammengefügt und im Hauptspeicher zur Verfügung gestellt |
| Sortieren/Mischen 2 | Eine Datei kann nach bestimmten Kriterien sortiert oder mehrere bereits sortierte Dateien können zu einer sortierten Datei gemischt werden. |
| Teilnehmerunterstützung TSO | TSO ermöglicht die Bereitstellung von Jobströmen vom Abonnentenpunkt aus und kann deshalb die veraltete Lochkartentechnik ablösen. |
| DSPF | System zur Entwicklung und Ausführung von Programmen im Dialog |
| Graphisches Kernsystem GKS | Definiert eine einheitliche Schnittstelle zwischen Anwenderprogrammen und graphischer Gerätetechnik. Enthält grundlegende Funktionen zur Erzeugung zweidimensionaler graphischer Darstellungen. |
| DAVM | Programmsystem zum effektiven Warten von Direktzugriffsdatenträgern |
| Prüfprogramm für externe Geräte | |

| Programmiersysteme (siehe Abschnitt 2.5) | |
|--|---|
| Assembler 2 | Maschinenorientiert – besonders gute Ausnutzung der Rechnereigenschaften; höhere Effektivität als der bisherige Assembler |
| COBOL | Problemorientierte Programmiersprache, vorwiegend für Aufgaben der kommerziellen Datenverarbeitung |
| FORTTRAN / FORTRAN 77 | Problemorientierte Programmiersprachen, vorwiegend für numerische Berechnungen in wissenschaftlich-technischen Bereichen |
| PASCAL | Problemorientierte Programmiersprache (Lehr- und Lernsprache) |
| PL1 PL1-Optimierungscompiler PL1-Testcompiler | PL1 ist eine universelle, problemorientierte Programmiersprache zur Formulierung von Algorithmen der numerischen und der nichtnumerischen Datenverarbeitung |
| RPG 2 / AUTOREPORT | Programm-Generator, vorwiegend für die Programmierung von Listen (Geschäftsberichten); Verarbeitung großer Datenmengen wobei die Eingabe/Ausgabe dominiert, Sortieren/Mischen eingeschlossen. |

2.3.2. Zugriffsmethoden

Es gibt neben den bereits beschriebenen eine neue Zugriffsmethode VSAM.

2.3.2.1. Zugriffsmethode VSAM

VSAM ist eine verallgemeinerte Zugriffsmethode für Direktzugriffsgeräte im Betriebssystem OS/ES. Die Abkürzung ist aus dem englischen Begriff Virtual Storage Access Method entstanden. VSAM ersetzt alle bisher in Gebrauch befindlichen Datenverwaltungssysteme. Diese Zugriffsmethode basiert auf der Idee, Dateiinhalte im virtuellen Hauptspeicher adressieren zu können. Das wird mit Hilfe einer so genannten Relative Byte Address (RBA) unterstützt. Dadurch wurde es erstmalig möglich, auf bautechnische Eigenschaften der Speichermedien keine Rücksicht mehr nehmen zu müssen. VSAM-Dateien werden im Allgemeinen als Cluster (Haufen, Gruppe) bezeichnet. VSAM-Dateien werden mit Hilfe des Dienstprogramms IDCAMS angelegt und verwaltet.

Struktur der Dateien

Während konventionelle Zugriffsmethoden nur einen einzigen Dateityp unterstützen (Non-VSAM-Dateien), gibt es folgende Formen von VSAM-Clustern:

- **KSDS (Key Sequential Data Set):**
Der Dateizugriff erfolgt über einen Index, der in einer eigenen Datei gespeichert wird. Eine KSDS besteht aus zwei logischen Komponenten, einem Daten- und einem Indexteil. Der Indexteil enthält Sätze, die eine Beziehung zwischen den logischen Schlüsseln und den physischen Sätzen herstellen. KSDS-Dateien ähneln ISAM-Dateien.
- **ESDS (Entry Sequential Data Set):**
Der Zugriff erfolgt sequentiell auf den Datenbestand der Datei. Eine ESDS besitzt nur den Datenteil, keinen Indexteil. Nachträglich kann mit Hilfe von Alternativindexen ein Direktzugriff ermöglicht werden. ESDS-Dateien ähneln QSAM- oder BSAM-Dateien.

- **RRDS (Relative Record Data Set):**

Der Zugriff erfolgt mit Hilfe von logischen Satznummern. Eine RRDS besitzt nur den Datenteil, keinen Indexteil.

RRDS-Dateien ähneln BDAM-Dateien.

Bei Verwendung der Zugriffsmethode VSAM wird nicht jede einzelne Datei durch eine Eintragung im Dateiverzeichnis (VTOC) beschrieben, sondern nur jeder VSAM zugeordnete Speicherbereich. Die Beschreibung und Verwaltung der einzelnen VSAM-Dateien innerhalb eines solchen VSAM-Bereiches erfolgt durch spezielle Kataloge, so genannte VSAM-Kataloge. Weiterhin gibt es eine Steuerdatei, die VVDS (VSAM Volume Data Set), die alle Statistikeinträge für alle VSAM-Dateien enthält, die sich auf dem gleichen Direktzugriffsspeicher befinden. Es handelt sich um eine ESDS-Datei.

Bei einer VSAM-Datei werden Steuerintervalle (CI – Control Interval) und Steuerbereiche (CA – Control Area) unterschieden. Ein Steuerintervall ist ein physischer Satz einer VSAM-Datei. Er enthält die logischen Sätze und Steuerinformationen. Ein Steuerbereich ist die Zusammenfassung mehrerer Steuerintervalle. Beim Einfügen von Sätzen in eine KSDS-Datei kann ein sogenannter CA-Split auftreten. Dabei werden Steuerbereiche (CAs) geteilt. Diese Teilung ist ein sehr E/A-intensiver Vorgang.

Ein Steuerintervall ist die Basiseinheit für die Datenübertragung zwischen Hauptspeicher und Direktzugriffsgerät. Alle CIs einer Datei haben die gleiche Größe. Die Größe kann vom Benutzer oder von VSAM festgelegt werden und ist geräteunabhängig. Ein CI kann größer als eine Spur auf einem Direktzugriffsspeicher sein.

Ein Steuerbereich enthält eine feste Anzahl von CIs für eine VSAM-Datei und belegt immer einen zusammenhängenden Bereich auf einem Direktzugriffsspeicher.

VSAM-Kataloge

Beim Erzeugen einer Nicht-VSAM-Datei hatte der Benutzer die Wahl, diese Datei in einen Katalog einzutragen oder nicht. VSAM-Dateien müssen in einem Katalog eingetragen sein. Dieser Katalog ist selbst ein KSDS-Cluster. In diesen Katalog können auch Einträge für Nicht-VSAM-Dateien gemacht werden. Es muss mindestens ein VSAM-Katalog vorhanden sein. Existieren mehrere Datenträger mit VSAM-Dateien, so ist es sinnvoll, auf jedem Datenträger einen VSAM-Katalog anzulegen, der die auf dem Datenträger vorhandenen VSAM-Dateien beschreibt.

Jedes System, das VSAM benutzt, hat einen und nur einen Hauptkatalog. Dieser wird während der Systemgenerierung erzeugt und befindet sich in der Regel auf der Systemresidenz. Er enthält Verweise zu den anderen VSAM-Katalogen, den sogenannten Benutzerkatalogen. Die Benutzer des Betriebssystems dürfen nur Dateien in diesen Benutzerkatalogen eintragen. Der Systemkatalog enthält seinerseits einen Verweis zum VSAM-Hauptkatalog.

Das Durchsuchen der Kataloge erfolgt für VSAM- und Nicht-VSAM-Dateien in folgender Reihenfolge: Benutzerkatalog, Hauptkatalog, Systemkatalog. Deshalb ist es sinnvoll, in einen VSAM-Benutzerkatalog nicht nur die VSAM-Dateien, sondern auch alle anderen Dateien des betreffenden Datenträgers einzutragen. Auf diese Weise werden Suchzeiten verringert.

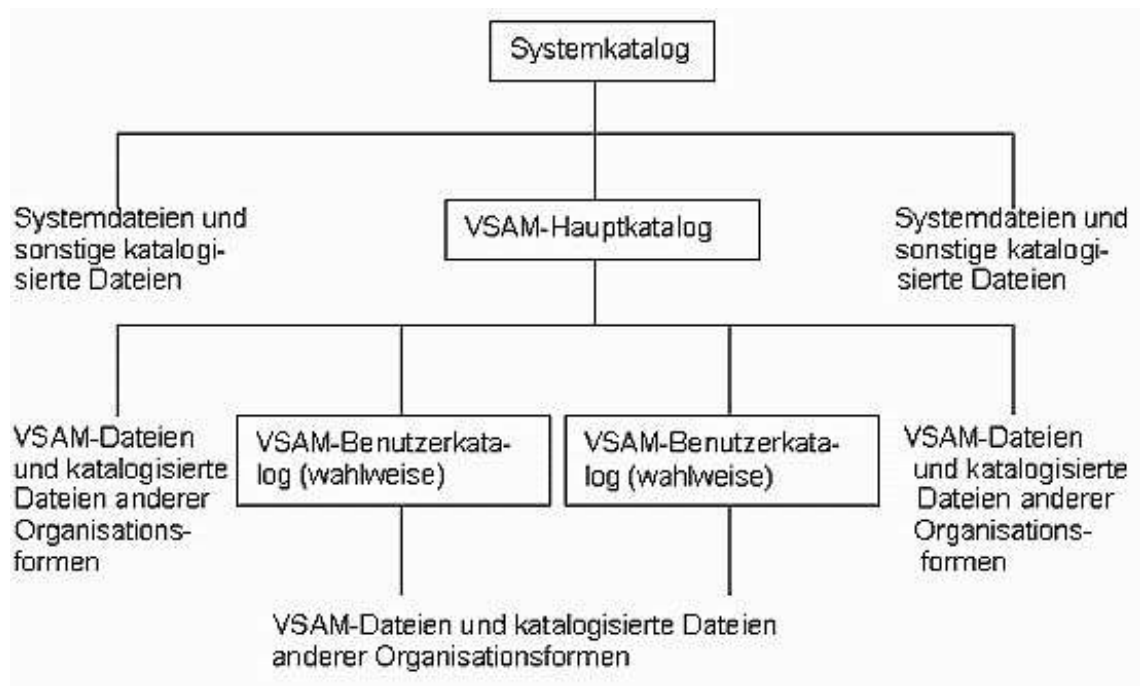


Abbildung 5 Katalog-Hierarchie bei VSAM

VSAM-Datenräume

Bevor VSAM-Dateien auf einem Datenträger definiert werden, müssen ein oder mehrere Datenräume (VSAM Data Space) erzeugt werden. Ein Datenraum ist ein Bereich auf einem Direktzugriffsspeicher, der ausschließlich für die Benutzung durch VSAM vorgesehen ist. Dieser Bereich wird in das Datenträgerinhaltsverzeichnis (VTOC) eingetragen.

Die Namen der Dateien für Kataloge und Datenräume, die im VTOC eingetragen sind, werden von VSAM generiert. Es handelt sich um qualifizierte Namen. Für Kataloge (Haupt- und Benutzer) heißt die höchste Stufe des Namens Z9999992, für Datenräume Z9999994.

Kennwortschutz

VSAM-Dateien und VSAM-Kataloge können durch Kennworte geschützt werden, die beim Anlegen einer Datei oder eines Katalogs festgelegt werden. Es sind vier Stufen des Kennwortschutzes vorgesehen. Ist das Hauptkennwort bekannt, so können Dateien geändert und gelöscht werden sowie Katalogeintragungen und Kennworte geändert werden. Bei Kenntnis des Steuerintervallkennwortes können Steuerintervalle geschrieben und gelesen werden. Das Schreibkennwort ermöglicht das Schreiben, Lesen, Ändern, Löschen und Einfügen von Dateisätzen, während Katalogsätze nur begrenzt geändert werden können. Ist nur das Lesekennwort bekannt, so können auch nur Sätze einer Datei oder eines Katalogs gelesen werden.

Alle VSAM-Dateien und VSAM-Kataloge werden mit dem Dienstprogramm für die verallgemeinerte Zugriffsmethode IDCAMS, kurz AMS (Access Method Services), angelegt. Auch Nicht-VSAM-Dateien können mit diesem Dienstprogramm in einen VSAM-Katalog eingetragen werden.

2.3.3. Serviceprogramme

2.3.3.1. Dienstprogramme

Siehe EC 1040

2.3.3.2. IDCAMS (Dienstprogramm für die verallgemeinerte Direktzugriffsmethode)

Das Dienstprogramm IDCAMS (kurz AMS) wird zusammen mit der Zugriffsmethode VSAM bereitgestellt. Es dient dazu, VSAM-Datenbereiche, -Kataloge und -Dateien anzulegen oder zu löschen. Weiterhin bietet AMS eine Reihe von Servicefunktionen an, z.B.

- Ändern von Katalogeintragungen
- Aufbau eines Alternativindex
- Aus- und Eingliedern von Katalogen, Verbänden und Alternativindizes
- Wiederherstellen eines defekten Kataloges
- Kopieren von Verbänden, Alternativindexen und sequentiellen Dateien
- Umwandlung von sequentiellen und indexsequentiellen Dateien in VSAM-Dateien und umgekehrt
- Drucken von Katalogen, Katalogeintragungen, Alternativindexen sowie sequentieller und indexsequentieller Dateien

2.3.3.3. Das Dialogsystem „Teilnehmerunterstützung TSO“

Die Nutzung des viel komfortableren TSO (gemessen an CRJE) setzt die Erstellung eines TCAM-MCPs (zentrales Steuerprogramm) voraus. Dieses ist von Systembetreuern auf der Basis von Generierungs-Makros zu erstellen und vor dem TSO-Start zu starten. Die Erstellung des MCP ist relativ aufwendig und durch sehr umfangreiche Parametrierung in den Generierungs-Makros recht unübersichtlich. Erst die Bereitstellung gewisser vorbereiteter Standardgenerierungen vereinfacht dieses Problem. Mit der umfassenden Bereitstellung des Bildschirmsystems EC7920 ist das TSO das meist genutzte Dialogsystem im OS/ES.

Die Programm-Entwicklung im Dialogsystem „Teilnehmerunterstützung TSO“ wird durch Programmiersysteme (PS COBOL, PS FORTRAN, PS PASCAL, PS PL1) unterstützt.

Die im SVS integrierte Teilnehmerunterstützung TSO gewährleistet für SVS einen flexiblen Teilnehmerbetrieb mit einfacher Anwendung und großem Funktionsumfang. Über lokal und entfernt aufgestellte Datenstationen hatte der Teilnehmer die Möglichkeit der direkten und unmittelbaren Kontrolle über die Abarbeitung seiner Aufträge, SVS unterstützt bei der Arbeit mit TSO einen Hintergrund-Stapelbetrieb, der die freien Ressourcen des Teilnehmerbetriebs nutzt. Der TSO-Teilnehmer hatte die Möglichkeit, Aufträge in diesen Stapelbetrieb einzugliedern und die Ergebnisse über seine Datenstationen auszugeben bzw. zentral über die Systemausgabe abzulegen.

Durch eine spezielle Verbindung zwischen SVS und TSO bezüglich der Nutzung des virtuellen Speichers sowie des zeitgeteilten Zugriffs zu den technischen Mitteln wurde die Zahl der möglichen parallelen Auftragsbearbeitungen wesentlich erhöht. Die einzelnen Teilnehmer werden dabei in der Regel mit gleicher Priorität bedient.

2.4. Unterstützung der Datenfernverarbeitung für EC1055, EC1055M, EC1056 und EC1057 in den Betriebssystemen OS/ES 6.1 und OS/ES 7.x

Die gesamte DFV-Gerätetechnik der Peripherie des EC 1040 wurde weiter im selben Umfang unterstützt wie bisher. In diese Zeit fällt die Produktionseinstellung des EC7907M (BSS/ESER). Die im Abschnitt „Unterstützung der Datenfernverarbeitung für EC1040 in den Betriebssystemen OS/ES 4.1 und OS/ES 6.1“ erwähnten Geräte wurden weiter angeboten und durch eine Reihe anderer Geräte, die im Folgenden geschildert sind, ergänzt.

Einen wesentlichen Schub in der Entwicklung der Fernverarbeitung bei Robotron war die Bereitstellung der 8-Bit-Rechentechnik (K 1520). Auf dieser Basis wurden das Multiplexsteuergerät EC 8404M (K 8563) und die Bürocomputer der Reihe A 51x0 gebaut.

EC 8404M

Der EC 8404M war ein Multiplexor der auf der Basis der K1520-Technik basierte. Er konnte bis zu 8*16 Leitungen steuern. Wie der EC 8404 besaß er ein bei Betriebsbeginn zu ladendes, vorher entsprechend zu konfigurierendes Steuerprogramm, welches die gleichen Übertragungsprotokolle wie am EC8404 unterstützte.

EC 8371 (IBM 327x)

Im ESER wurden unter der Nummer EC 8371 2 Fernverarbeitungsprozessoren analog IBM 327x entwickelt und gebaut (VR Polen + VR Bulgarien). Diese unterstützen neben den klassischen Übertragungsprozeduren das IBM SNA-Konzept und das bit-serielle Übertragungsprotokoll SDLC. Das zu ladende Steuerprogramm NCP (Network Control Programm) wurde am Zentralrechner konfiguriert und dann in den Prozessor geladen. Alle klassischen Prozeduren (Protokolle) wurden wie beim EC8404M weiter unterstützt.

Da die Entwicklung einer EC 7920 Steuereinheit für bit-serielle Übertragung nicht abgeschlossen wurde, erfolgte die Nutzung dieser Prozedur nur in Ausnahmefällen bzw. mittels anderer Technik.

Bürocomputer entsprechend EC 8561- EC 8565 (A 5110, A 5120, A 5130)

Diese auf der Basis des U 880-Prozessors gebauten Computer konnten mit einer V.24-Leiterplatte ausgerüstet werden. Als Betriebssystem stand das SIOS bzw. BROS zur Verfügung. Mittels von Diskette ladbarer Emulationsprogramme für verschiedene Übertragungsprotokolle konnten diese BC zu Abonnentenpunkten umfunktioniert werden. In diesem Betriebsmodus war keine andere BC(PC)-Arbeit möglich.

Für folgende Protokolle gab es offizielle Emulatoren:

- Start/Stop EC 8564
- BSCIII entsprechend EC 7925
- BSCI
- T34
- AP 70 (EC 8570)

Gleichstrom-Daten-Nahübertragungsgerät – GDN (DNÜ EC 79xx):

Dieses Gerät ermöglichte Übertragungsgeschwindigkeiten bis zu 9600 Bit/s für die gleichen Übertragungsprozeduren wie bei TAM 601, allerdings mit gravierenden Einschnitten bei der Übertragungsentfernung.

Maximale Übertragungsentfernung bei geringster Übertragungsgeschwindigkeit waren 30 km

Die 9600 Bit/s waren nur bis zu einer Entfernung (Leitungslänge) von ~8 km möglich.

Das Gerät arbeitete sehr zuverlässig.

Es ermöglichte bei entsprechender Schaltung (Wickelbrücken) die Arbeit einer sekundären GDN als Master-GDN um bis zu 5 weitere GDN mit je einem Terminal zu steuern.

Entwicklung eines KIF-Adapters für die BC- bzw. PC-Technik auf U880-Basis

Da abzusehen war, dass die Bildschirmgeräte vom Typ EC 7927 abgelöst werden mussten (zu schwer, zu teuer und technisch veraltet) wurde im Nachgang zur Entwicklung der DDT von RPD/E2 (Betriebsteil Karl-Marx-Stadt) ein Adapter zur Emulation des „Kleinen Interface“ hergestellt. Die Interfaces zwischen den Bildschirmen EC 7920 und den Gerätesteuereinheiten EC 7921 (Fernaufstellung), EC 7922 (Lokalaufstellung) und der noch zu entwickelnden SDLC-Gerätesteuereinheit) waren identisch, deshalb konnte man sich mit der Entwicklung eines einzigen KIF-Adapters begnügen. Die BC's und PC's konnten dann, direkt angeschlossen an eine EC 792x-Gerätestation, wie ein Bildschirm EC 7920 arbeiten. Während der Zeit, da das Emulationsprogramm geladen war, war keine andere PC-Arbeit möglich. In einer späteren Version des Adapters sollte dann noch Dateitransfer zum und vom PC möglich sein.

PC 1715, PC 1715W, A7100, A7150

Für die Nachfolgetechnik der Bürocomputer wurden analoge Emulatoren zu diesen entwickelt und eingesetzt.

Rechnerkopplungen:

In dieser Zeit wurde intensiv an der Kopplung mit Rechnern auf der Basis der Übertragungsprozedur BSCI gearbeitet. Realisiert wurde sie vor allem für die Kopplung mit der K 1600-Technik

Es standen dann zunächst im OS/ES folgende Nutzungsmöglichkeiten für diese Geräte zur Verfügung:

Zugriff über die Einfache Zugriffsmethode für Datenfernübertragung BTAM (basic telecommunication access method)

Die einfache Zugriffsmethode für Datenfernverarbeitung BTAM unterstützt die technischen Mittel für System-Datenfernverarbeitung des ESER und lokale Bildschirmsysteme. Zur System-Datenfernverarbeitung gehören Multiplexsteuergeräte und Datenstationen, die sowohl an Standleitungen als auch an Wählleitungen angeschlossen sein können. Datenstationen können entweder mit dem Start-Stop-Übertragungsverfahren oder mit dem synchronen Übertragungsverfahren arbeiten, bei dem keine Netzprotokolle benutzt werden. BTAM unterstützt die Steuerung des Datenempfangs bzw. der Datenübertragung zwischen Zentraleinheit und den Datenstationen und führt Dienste für die Geräteeinrichtungen der DFV-Leitung aus.

Zugriff über Erweiterte Zugriffsmethode für Datenfernübertragung TCAM (telecommunication access method)

Die Zugriffsmethode für Datenfernverarbeitung mit erweiterter Netzfunktion TCAM/ANF unterstützt die technischen Mittel der System- und Netzdatenfernverarbeitung sowie lokale Bildschirmsysteme. Elemente der Netz-DFV sind DFV-Prozessoren und Datenstationen, die die Netzprotokolle der Netzarchitektur des ESER realisieren. Die Mittel des TCAM/ANF unterstützen die Steuerung des Nachrichtenflusses zwischen den Datenstationen und den Anwenderprogrammen in einem DFV-Netz. TCAM/ANF ermöglicht die Einbeziehung mehrerer EDVA in ein DFV-Netz bei weitgehender Rekonfigurierbarkeit des Netzes im Fall von Störungen oder notwendigen Veränderungen des Nachrichtenverkehrs.

Arbeitet SVS unter Steuerung eines SVM, besteht die Möglichkeit, TCAM-Datenstationen zwischenzeitlich zur Steuerung von virtuellen Maschinen des SVM zu benutzen und TCAM-Programme unter dem Programmier- und Testsystem PTS des SVM anzuwenden.

Daneben unterstützte die Komponente Dateifernübertragung (NFT) einen Austausch von Dateien mit fernaufgestellten stapelorientierten Abonnentenpunkten.

ITS (intelligent terminal support) (Zusatzkomponente)

Um die Effektivität der Arbeit der Bürocomputer zu erhöhen und die Leitungskapazitäten besser auslasten zu können wurde für die Programmierung von Anwendungen auf der Basis der Zugriffsmethode TCAM das Makrosystem ITS entwickelt. Es bedingt, dass im Bürocomputer ein spezieller ITS-Emulator, der auf der Basis der BSCIII-Prozedur oder des AP62/64-Start-Stopp-Protokolls arbeitete, geladen wird.

Grundgedanke ist folgender:

Bildschirmmasken müssen nicht vom Zentralrechner jedesmal auf den AP übertragen werden. Es sollte genügen, diese auf einer Diskette auf dem BC zu speichern und diese nur durch Senden einer Kennzeichnung (Nr.) vom Zentralrechner zum gewünschten Zeitpunkt abzurufen und auf dem Bildschirm erscheinen zu lassen. Danach kann der Dialog unter den Bedingungen dieser Maske vom AP geführt werden bis eine andere Maske, die wiederum nur durch Senden einer Kennung von der ZE von Diskette abgerufen wird, erforderlich wird. Das führt natürlich zu erheblich besseren Übertragungszeiten. Dabei ist es nötig, dass auf der ZE ein Abbild der jeweiligen Maske existiert.

Das auf Zentralrechner-Seite bereitgestellte Makrosystem vereinfacht die Programmierung der Anwendungsprogramme. Für die Generierung eines TCAM-MCP stehen, ähnlich wie bei TSO, spezielle, die Generierung vereinfachende, Makrosysteme zur Verfügung.

Unterstützte Übertragungsprotokolle:

Start/Stop-Prozeduren:

1. entsprechend der IBM 2260,
2. Fernschreibtechnik,
3. Schreibmaschinenprozeduren analog IBM 1050

Basis-Synchronprozeduren:

1. für Mehrpunktgeräte BSCIII/CHX3
2. für Punkt-zu-Punkt-Geräte BSCI/CHX1 - diese wurde vor allem für Dateübertragungen genutzt.

Das Bit-serielle Übertragungsprotokoll SDLC für die SNA-Netzarchitektur stand betriebssystemmäßig ebenso wie das entsprechende NCP-Steuerprogramm für die EC837x-Prozessoren zur Verfügung. Eine entsprechende End-Gerätetechnik wurde nicht mehr bereitgestellt.

2.5. Programmiersysteme

Programmiersystem COBOL

Der im PS COBOL implementierte Sprachumfang basiert auf dem COBOL-Standard ANSI X 3.23-1974 Stufe 1 und enthält außerdem Elemente der Stufe 2.

Das PS COBOL umfasst den COBOL-Compiler, den COBOL-Dialogtester, Vermittler für beide COBOL-Komponenten zum Dialogsystem „Teilnehmerunterstützung TSO“ und zum „Dialogfähigen Programmier- und Testsystem PTS“.

Besonderheiten der COBOL-Implementierungen im PS COBOL sind:

- Verarbeitung von Zeichenketten,
- dynamischer Aufruf/dynamische Freigabe von Unterprogrammen,
- Verarbeitung externer Dezimalzahlen mit getrenntem Vorzeichen,
- Festlegung spezieller Sortierfolgen sowie
- Unterstützung der Datenfernverarbeitung.

Programmiersystem FORTRAN

Das PS FORTRAN umfasst die Compiler FORTRAN CC, FORTRAN SE, und das FORTRAN-Konvertierungsprogramm. Für diese drei Komponenten existieren Vermittler zum Dialogsystem „Teilnehmerunterstützung TSO“ und zum „Dialogfähigen Programmier- und Testsystem PTS“.

FORTRAN OE (= Optimierungscompiler) ist für die Arbeit im Stapelbetrieb und im PTS vorgesehen. Für ein mit dem PS FORTRAN entwickeltes und getestetes Programm liefert FORTRAN OE einen effektiven Code hinsichtlich der Ausführungszeit und des Speicherplatzbedarfs.

Der im PS FORTRAN und im FORTRAN OE implementierte Sprachumfang entspricht dem FORTRAN-Standard ANSI X 3.9 – 1966.

Programmiersystem PASCAL

Das PS PASCAL umfasst den PASCAL-Compiler und den PASCAL-Dialogtester und darüber hinaus Vermittler zum Dialogsystem „Teilnehmerunterstützung TSO“ und „Dialogfähigen Programmier- und Testsystem PTS“.

Der im PASCAL-Compiler implementierte Sprachumfang basiert auf dem PASCAL-Standard ISO 7185-83 Stufe 0 und ist zur Implementierung des PS PASCAL 1600 kompatibel.

Über den Standard hinausgehende Erweiterungen betreffen

- den Basistyp von Mengen; maximal 256 Elemente sind möglich,
- getrennte Übersetzung von PASCAL-Routinen,
- Aufruf externer Routinen (auch in anderen Programmiersprachen geschriebene sind möglich).

Programmiersystem PL1

Der PL1-Testcompiler (PL1-TC) und der PL1-Optimierungscompiler (PL1-OC) ergänzen sich wechselseitig und bilden zusammen mit dem Vermittler zum Dialogsystem „Teilnehmerunterstützung TSO“ und einem Interface zum „Dialogfähigen Programmier- und Testsystem PTS“ das Programmiersystem PL1 (PS PL1).

Beide Compiler basieren auf einem einheitlichen Sprachumfang; gegenüber dem PL1-Normalcompiler (PL1-NC) gibt es zahlreiche Erweiterungen des Sprachumfangs, Aufwärtskompatibilität ist bis auf ganz unbedeutende, unwesentliche Besonderheiten der Implementierung im PL1-NC, gewährleistet.

Der PL1-TC gestattet die rationelle Programmentwicklung sowie einen effektiven Programmtest. Der PL1-OC ermöglicht die Erzeugung eines guten Objektcodes hinsichtlich der Ausführung des fertigen Programms bei der Nutzrechnung.

RPG2 mit AUTOREPORT

Der Listen-Programmgenerator RPG2 ist eine Weiterentwicklung von RPG und dient der effektiven Programm-Entwicklung zur Verarbeitung großer Datenbestände.

RPG2 ist nur für die Arbeit im Stapelbetrieb vorgesehen.

AUTOREPORT erweitert die Anwendungsbreite von RPG2 beispielsweise hinsichtlich der Auswahl von Eingabesätzen, des Sortierens von Datenbeständen und zur automatischen Druckbild-Erzeugung.

3. SVM/ES

3.1. Historisches

Zur Erweiterung der Einsatzbereitschaft des Modells EC 1055M wurde 1982 ein neues Betriebssystem, das System virtueller Maschinen (SVM/ES), bereitgestellt. SVM ist auf allen Modellen des ESER II/III einsetzbar.

3.2. Charakteristik

| Betriebssystem | benutzte Maschinenart und charakteristische Eigenschaften | Prinzip der Hauptspeicherverwaltung |
|----------------|---|---|
| SVM | SVM arbeitet auf realen Maschinen des ESER. Es bietet mehreren Nutzern gleichzeitig die Möglichkeit, voneinander unabhängig über Datenstationen mit einer EDVA zu arbeiten. SVM organisiert die Arbeit der EDVA so, dass jeder Nutzer den Eindruck erhält, es stehe ihm eine vollständige EDVA zur Verfügung. Diese scheinbare EDVA wird als virtuelle Maschine bezeichnet. Der Nutzer hat die Möglichkeit, auf einer solchen virtuellen Maschine ein beliebiges Betriebssystem des ESER zu betreiben, systemunabhängige Programme arbeiten zu lassen oder spezielle Dienste des SVM zu nutzen. Die virtuelle ZE wird durch zeitgeteiltes Bereitstellen der realen ZE realisiert. Voraussetzungen: Virtuelle ZE, virtueller HS, virtuelle Bedieneinheit und virtuelle periphere Geräte. | Jeder virtuellen Maschine wird nach dem Prinzip der dynamischen Adressumsetzung ein virtueller Speicher von max. 16MB zur Verfügung gestellt. Das Betriebssystem, das auf der virtuellen Maschine läuft, betrachtet den virtuellen Speicher als Hauptspeicher einer realen Anlage. Der Seitenaustausch erfolgte in einer sogenannten Seitengröße von 4KB. |

SVM/ES gestattet durch die Einrichtung und Verwaltung mehrerer virtueller Maschinen die parallele und voneinander unabhängige Arbeit mehrerer Nutzer auf einer einzigen Datenverarbeitungsanlage (DV-Anlage).

Das Konzept der virtuellen Maschine erweitert das Konzept des virtuellen Speichers in dem Sinne, dass alle Elemente einer DV-Anlage virtuell realisiert werden. Mehrere Nutzer können mit ihren virtuellen Maschinen, die sich auf einer realen DV-Anlage befinden, so arbeiten, als würden sie allein die Anlage benutzen. Insbesondere muss jeder Nutzer zur Steuerung der virtuellen Maschine ein Betriebssystem des ESER (OS/ES, DOS/ES aber auch SVM/ES selbst) oder ein systemunabhängiges Dienstprogramm starten.

3.3. Komponenten

- **Steuerprogramm (CP)**

Das Steuerprogramm des SVM verwaltet die Ressourcen der realen DV-Anlage und koordiniert die Anforderungen der einzelnen virtuellen Maschinen. Jede virtuelle Maschine muss vor ihrer Benutzung durch eine Eintragung im Nutzerverzeichnis des SVM definiert werden. Der Nutzer aktiviert eine virtuelle Maschine durch ein LOGON-Kommando von einer Datenstation, die lokal oder fern aufgestellt sein kann und die als virtuelle Bedieneinheit fungiert. Über die virtuelle Bedieneinheit erfolgt sowohl die Steuerung des Betriebssystems in der virtuellen Maschine als auch die Kommunikation mit dem CP. Das CP verfügt über eine eigene Kommandosprache, mit deren Hilfe die Funktionen des Steuerprogramms aktiviert werden. Diese Kommandos dienen der Steuerung des SVM/ES, der

Gerätezuordnung, der Gerätesteuerung, der Informationsbeschaffung, dem Programmtest und der Dateipufferung.

Entsprechend dem Konzept der virtuellen Maschinen werden durch das CP eine virtuelle Zentraleinheit (ZE), ein virtueller Speicher, eine virtuelle Bedieneinheit und virtuelle E/A-Geräte simuliert.

Das CP ordnet jeder virtuellen Maschine Zeitscheiben zu, in denen ihr die ZE zur Verfügung steht.

Der reale Hauptspeicher wird durch das CP für die virtuellen Maschinen nach dem Prinzip des virtuellen Speichers verwaltet, wobei jede Maschine einen eigenen Adressraum erhält. Der einer virtuellen Maschine zugeordnete virtuelle Speicher erfüllt für das Nutzerbetriebssystem die Funktion des Hauptspeichers. Die Umsetzung der virtuellen Adressen in reale Adressen erfolgt durch die Nutzung der dynamischen Adressumsetzung.

Virtuelle E/A-Geräte werden realisiert, indem einem virtuellen Gerät ein reales Gerät fest zugeordnet wird oder mehrere virtuelle Geräte auf einem realen Gerät gebildet werden oder virtuelle Geräte durch das CP vollständig simuliert werden.

- **Dialogfähiges Programmier- und Testsystem** (PTS Programming and Testing Support)

Das PTS bietet die Möglichkeit, im Dialog Programme oder andere Dateien aufzubauen und zu verändern, Programme zu übersetzen, zu testen und auszuführen.

- **Dateifernübertragungssystem** (RFTS Remote File Transfer Subsystem)

Das RFTS dient zur Übertragung von Dateien zwischen fernaufgestellten stapelorientierten Datenstationen und der DV-Anlage sowie zwischen Datenstationen. In Verbindung mit der Dateipufferung des CP kann damit der Datenaustausch zwischen virtuellen Maschinen und fernaufgestellten Datenstationen erfolgen und u.a. die Jobfernverarbeitung realisiert werden.

- **Fehleranalysesystem** (PDAS Problem and Dump Analysis Subsystem)

PDAS ist eine Erweiterung des PTS und dient zum Erfassen und Auswerten von Fehlerabläufen im SVM/ES.