

*** D I A L O G - B E T R I E B ***

Inhaltsverzeichnis

	Seite
1. Einführung	1-1
2. Terminals	2-1
2.1 Datenübertragung	2-1
2.2 MRX Terminal 2078	2-3
2.3 Graphik Terminals	2-5
2.4 SPERRY Terminal UTS20	2-6
2.5 DEVELNET Terminals	2-6
3. Arbeiten im Dialog-Betrieb	3-1
3.1 Dialog-Betrieb	3-1
3.1.1 Verbindung aufbauen	3-1
3.1.2 Session beginnen (Logon)	3-1
3.1.3 Commands eingeben	3-4
3.1.4 Session unterbrechen (disconnect) / fortsetzen (reconnect)	3-7
3.1.5 Session beenden (Logoff)	3-7
3.1.6 Verbindung abbauen	3-8
3.2 Unterbrechen, Fortsetzen, Abbrechen	3-8
3.2.1 Programme, CMS Commands, Prozeduren	3-9
3.2.2 Ein/Ausgaben	3-10
3.2.3 Normieren des Terminals	3-11
3.3 Informieren	3-12
3.3.1 Leistung und Format von Commands	3-12
3.3.2 Verbrauchte CPU-Zeit	3-12
3.3.3 Belegter Plattenspeicher	3-12
3.3.4 Files der eigenen Minidisk	3-12
3.3.5 Benutzer-Berechtigungen	3-13
3.3.6 Nachrichten, Mitteilungen	3-13
3.4 Ändern des CP Directory Eintrags	3-13
3.4.1 Logon Paßwort ändern	3-14
3.4.2 Minidisk Paßwörter und implizite Zugriffsart ändern	3-14
3.5 Drucken	3-15
3.5.1 Bildschirminhalt drucken	3-15
3.5.2 Protokollieren einer Session	3-15
3.5.3 File-Inhalt drucken	3-16
4. Arbeiten mit Files	4-1
4.1 Files auf Minidisks	4-1
4.2 Virtuelle Ein/Ausgabegeräte (Spooling)	4-3
4.2.1 Virtuelles Gerät Console	4-4
4.2.2 Virtuelles Gerät Printer	4-6

4.2.3	Virtuelles Gerät Punch	4-7
4.2.4	Virtuelles Gerät Reader	4-7
5.	Programmentwicklung in FORTRAN	5-1
5.1	Edieren (Quellprogramme)	5-2
5.2	Compilieren (Quellprogramme)	5-2
5.3	Binden (Objektprogramme)	5-3
5.4	File-Bearbeitung	5-4
5.5	Starten (ausführbares Programm)	5-4
6.	Beispiel für eine Session	6-1

1. Einführung

Dieses Kapitel führt in den Dialog-Betrieb unter VM/SP ein. Leistungen der Maschine (so nennt IBM einen Rechner) werden mit **Commands** angefordert. Im **Dialog-Betrieb** (SPERRY: Demand-Betrieb, DEC: Interaktiver Betrieb) wird i.a. jeweils ein Command (über die Tastatur des Terminals) eingegeben, von der Maschine ausgeführt und mit einer Antwort (auf dem Bildschirm des Terminals) quittiert; danach kann ein neues Command eingegeben werden. Im Batch-Betrieb dagegen werden die Commands eines Jobs als Ganzes bei der Initiierung übergeben (in einem File); die Antworten werden gesammelt und nach Bearbeitung des gesamten Jobs z.B. an einem Drucker ausgegeben (vgl. BATCH-BETRIEB).

Im Dialog-Betrieb können **Sessions** (SPERRY: Demand Runs, DEC: Interaktive Jobs) geführt werden; eine Session ist ein **Job** (SPERRY: Run, DEC: Job) im Dialog-Betrieb. Eine Session wird durch spezielle Commands begonnen und beendet. Beim Beginn stellt das Control Program (CP) der Session eine Umgebung bereit, die als **virtuelle Maschine** bezeichnet wird; die virtuelle Maschine wird standardmäßig von dem Betriebssystem **CMS** (Conversational Monitoring System) bedient.

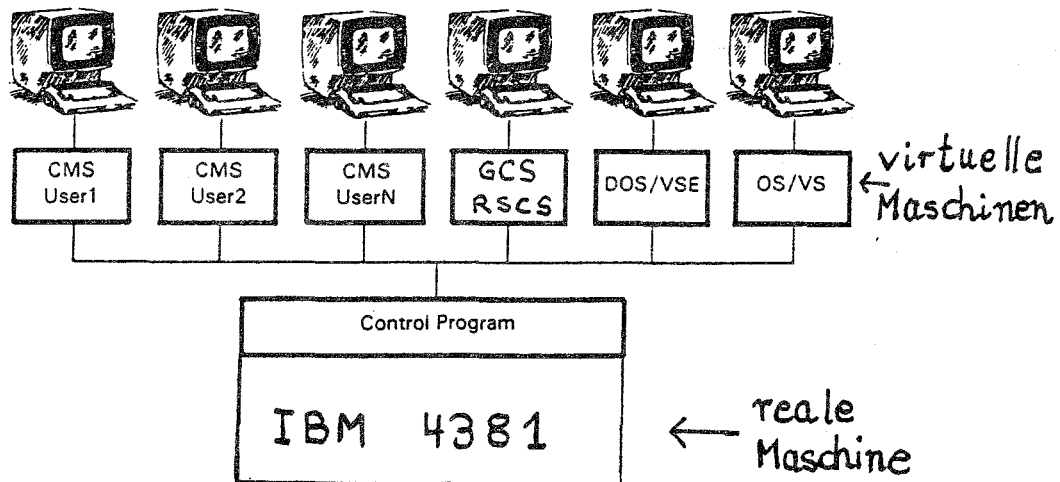


Abb. 1-1: Das Konzept der virtuellen Maschine vermittelt jedem Benutzer das einmalige PC-Gefühl: Diese Maschine gehört mir ganz allein.

Die Commands werden - abhängig davon, wer sie ausführt - unterteilt in CP Commands und CMS Commands. Mit CP Commands können z.B. Sessions begonnen und beendet werden, kann das Drucken von Files gesteuert werden, können temporäre Minidisks eingerichtet werden oder Informationen über verbrauchte Betriebsmittel ausgegeben werden. Die CP Commands sind in Classes eingeteilt; nur CP Commands der Classes Any und G (General User; sie wirken nur auf die eigene Session) sind allen Benutzern zugänglich, vgl. BENUTZER-BERECHTIGUNGEN. Mit Hilfe von CMS Commands können z.B. Programme ediert, compiliert, gebunden und gestartet werden.

Dokumentationen zu CP und CMS:

CMS Primer
CMS User's Guide
CMS Command Reference
CP Command Reference
Summary of Commands (im CMS Primer)
Quick Reference (CP, CMS, REXX, XEDIT)

Der CMS Primer ist als Einführungsliteratur sehr empfehlenswert; er beschreibt das Beginnen und Beenden von Sessions, das Edieren mit XEDIT und das Arbeiten mit Files; er enthält jedoch nichts zur Programmentwicklung. Der CMS User's Guide beschreibt ausführlich das Arbeiten mit CMS. Die CMS Command Reference dokumentiert Format und Leistung von CMS Commands mit Ausnahme von Compiler Aufrufen (zu den Compiler Aufrufen vgl. Notizen FORTRAN, COBOL, PASCAL,...). Die CP Command Reference beschreibt das Arbeiten mit CP sowie Leistung und Format der CP Commands aller Classes. Als Gedächtnisstütze bei der Arbeit am Terminal ist das Summary of Commands besonders empfehlenswert; es enthält das Wichtigste zum Arbeiten mit der IBM 4381 in knapper übersichtlicher Form (nach Funktionen geordnet); das Quick Reference enthält (alphabetisch geordnet) die Formate aller CP Commands, CMS Commands, REXX Instructions (für Prozeduren) und XEDIT Subcommands.

Gebrauchsanweisung: Dieses Kapitel sollte parallel zum CMS Primer durchgearbeitet werden; Benutzer mit sehr viel Erfahrung können das Kapitel ggf. ohne CMS Primer verstehen. Es wird empfohlen, zunächst die Abschnitte 2, 3 und 4 dieses Kapitels zu lesen und dann die Chapter 1 bis 4 des CMS Primers am Terminal durchzuarbeiten; bei Fragen kann auf die entsprechenden Abschnitte dieses Kapitels zurückgegriffen werden. Wer selber (FORTRAN) Programme entwickeln will, sollte anschließend Abschnitt 5 dieses Kapitels lesen und das Beispiel in Abschnitt 6 nachvollziehen.

Das Drücken einer *Taste* wird durch *<Taste>* dargestellt (z.B. *<ENTER>* für Drücken der ENTER Taste). Beispiele für Eingaben sind *kursiv und fett* gedruckt und durch *<ENTER>* abgeschlossen; Antworten der IBM 4381 sind i.a. originalgetreu angegeben, Abweichungen von der originalgetreuen Wiedergabe sind *kursiv* gedruckt. In den Erläuterungen sind wählbare Bezeichnungen (z.B. Filenames) und Stellvertreter (z.B. *fn*) *kursiv* gedruckt.

Alphanumerische Zeichen sind Buchstaben (ohne Unterscheidung groß/klein), Ziffern und die Sonderzeichen \$, +, -, _ (Underscore); das Sonderzeichen \$ darf nicht in Bezeichnungen verwendet werden, die vom Benutzer gewählt werden.

Übersicht: Abschnitt 2 beschreibt Eigenschaften von verschiedenen Terminals. Abschnitt 3 beschreibt die Funktionen für das Arbeiten mit der IBM 4381 im Dialog-Betrieb; mit diesen Funktionen hat das Arbeiten im Dialog-Betrieb folgende grundsätzliche Struktur:

ggf. Verbindung aufbauen (vgl. 3.1.1)

.
Session beginnen (Logon, vgl. 3.1.2)

.
Command ausführen

.
Prozedur ausführen (vgl. 3.1.3)

.
Programm ausführen

.
Session beenden (Logoff, vgl. 3.1.5)

.
ggf. Verbindung abbauen (vgl. 3.1.6)

Abschnitt 4 beschreibt das Arbeiten mit Files auf Minidisks sowie die Ein/Ausgabe von Files (z.B. das Drucken). Abschnitt 5 beschreibt das Compilieren, Binden und Starten von FORTRAN Programmen; Abschnitt 6 enthält ein Beispiel für eine Session zum Edieren, Compilieren, Binden und Starten eines FORTRAN Programms.

2. Terminals

Sessions können an **Terminals** vom Typ

- MRX 2078 Standard Terminal
- IBM 3279 } Graphik Terminals
- IBM 3192 }
- TEK CX41XX }
- SPERRY UTS20 }
- QUME VT201 } DEVELNET Terminals
- DEC VT220 }
- DEC VT320 }
- ...

geführt werden. Jedes Terminal trägt einen Aufkleber mit Typ (z.B. MRX 2078) und einer Geräte-Bezeichnung (z.B. OB5). Das Terminal MRX 2078 ist als Standard Terminal anzusehen; seine Eigenschaften werden im Abschnitt 2.2 erläutert; zu den abweichenden Eigenschaften der anderen Terminals vgl. 2.3 ff. Zu den TEK Terminals CX41XX gibt es **Bedienungsanleitungen** von TEKTRONIX; zu den SPERRY Terminals UTS20 sowie zu den DEVELNET Terminals gibt es Bedienungsanleitungen des HRZ (sie liegen neben den Terminals).

Mikrocomputer/PC's mit 3270 Emulation können wie das Standard Terminal MRX 2078 über eine 3270 Steuereinheit auf die IBM 4381 zugreifen, vgl. Einzelschrift 3270 EMULATION (MIKROCOMPUTER). Mikrocomputer/PC's mit VT100 Emulation können wie DEVELNET Terminals auf die IBM 4381 zugreifen, vgl. Einzelschrift KERMIT (MIKROCOMPUTER).

2.1 Datenübertragung

Der Bildschirm ist in der Regel in schreibfreie und schreibgeschützte Bereiche unterteilt; die schreibfreien Bereiche werden im folgenden auch als **Eingabebereiche** bezeichnet; durch

<ENTER>

werden die in den schreibfreien Bereichen des Bildschirms stehenden Zeichen übertragen; die Datenübertragung vom Terminal zur IBM 4381 ist also bildschirmorientiert. Das Arbeiten mit der IBM 4381, insbesondere die Eingabe von Commands, erfolgt jedoch i.a. **zeilenorientiert** (syn. Line Mode); es gibt aber spezielle Commands, die **bildschirmorientiert** arbeiten (syn. Full Screen Mode), z.B. XEDIT (Editor) oder FILELIST. (Das Arbeiten mit CMS im Full Screen Mode ist auch möglich, wird aber hier nicht beschrieben).

Zeilenorientierte Arbeitsweise: In der rechten unteren Ecke des Bildschirms befindet sich ein Status Feld, in der der **Status** des Terminals sowie die Zeichenfolge DMRHRZ11 (EARN Nodeid der IBM 4381, vgl. Notiz EARN) ausgegeben wird. Mögliche Status innerhalb einer Session sind:

CP READ CP wartet auf die Eingabe eines Commands;
begin <ENTER> liefert VM READ.
VM READ Die virtuelle Maschine erwartet eine Eingabe;
... <ENTER> liefert RUNNING, vgl. Abb. 3-2.
RUNNING Die virtuelle Maschine arbeitet (Ausführung eines
Commands oder warten auf Eingabe eines Commands).
<ENTER> liefert VM READ, vgl. Abb. 3-2.
MORE... Der Bildschirm ist voll; weitere Ausgaben stehen
bereit. <ENTER> liefert HOLDING. <PA2> oder 1 Minu-
te warten liefert weitere Ausgaben (max. 1 Bild-
schirm, vgl. Abb. 3-5).
HOLDING Der Bildschirm ist voll; weitere Ausgaben stehen
bereit. <PA2> liefert weitere Ausgaben (max. 1
Bildschirm, vgl. Abb. 3-5).
NOT ACCEPTED Eingabe wurde von CP nicht angenommen.

Der Eingabebereich besteht nur aus den beiden letzten Zeilen abzüglich des Status Feldes. Durch <ENTER> werden die im Eingabebereich stehenden Zeichen übertragen und gelöscht; die Eingabe wird im oberen schreibgeschützten Bereich des Bildschirms protokolliert. Eine Eingabe kann nur mit Hilfe der RETRIEVE Taste (i.a. <PF6>, vgl. 3.1.3) mehrmals übertragen werden; durch Drücken der RETRIEVE Taste wird die letzte von CP gepufferte Eingabe in den Eingabebereich zurückgeschrieben. Sie kann dort korrigiert und erneut übertragen werden.

Bildschirmorientierte Arbeitsweise: Schreibfreie und schreibgeschützte Bereiche werden vom bildschirmorientiert arbeitenden Command (z.B. XEDIT, FILELIST) festgelegt. Die Bearbeitung der Eingaben ist Command-spezifisch.

2.2 MRX Terminal 2078

Das MRX 2078 wurde vom HRZ als Standard Terminal der IBM 4381 festgelegt; es entspricht dem IBM Terminal 3278. Neben dem hier beschriebenen Standard Terminal mit EBCDIC/APL Tastatur gibt es am HRZ auch noch einige MRX 2078 mit EBCDIC Tastatur (APL Tasten fehlen, andere Anordnung der PF Tasten). Alle MRX 2078 sind direkt an die IBM 4381 angeschlossen (entweder über eine lokale 3270 Steuereinheit oder über eine entfernte 3270 Steuereinheit und den Vorrechner, die DFV-Steuereinheit 3720, vgl. HARDWARE).

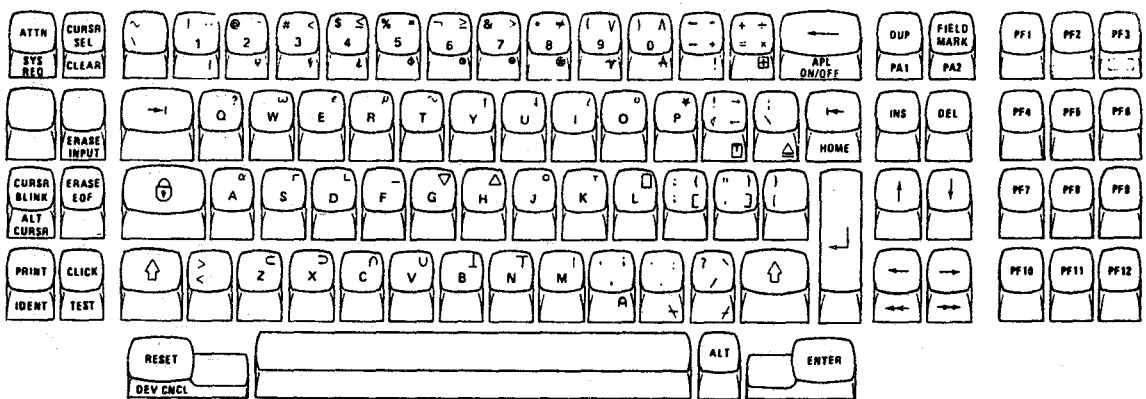
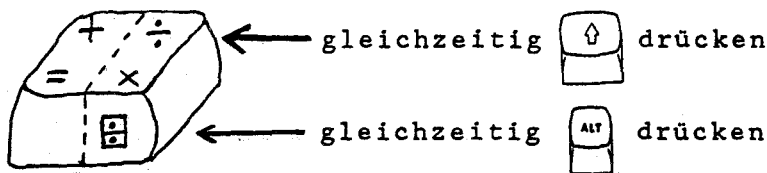


Abb. 2-1: EBCDIC/APL Tastatur

Die EBCDIC/APL Tastatur besteht aus Zeichentasten (weiß) und Funktionstasten (grau). Zum Umschalten von EBCDIC auf APL und umgekehrt ist <APL ON/OFF> (zusammen mit <ALT>) zu verwenden; zum Umschalten innerhalb von EBCDIC oder APL sind die Tasten \uparrow und ALT zu verwenden, z.B.:

EBCDIC APL



Zeichenvorrat: Von und zum Terminal werden 8-bit Zeichen gesendet; diese können als EBCDIC oder als APL Zeichen interpretiert werden (umschaltbar mit <APL ON/OFF> und dem CP Command TERMINAL APL ON bzw. OFF). APL Zeichen können nur innerhalb von APL verarbeitet werden.

Eingabe: Alle EBCDIC Schriftzeichen (vgl. CODES im Teil ALLGEMEINES) sind über die Tastatur eingebbar, die übrigen (8-bit) Zeichen sind nur mit Hilfe des Editors eingebbar (nach Ausführung des XEDIT Subcommands SET HEX ON ersetzt der Editor in allen nachfolgenden Eingaben z.B. die Zeichenfolge X'AD' durch das 8-bit Zeichen mit hexadezimalen Wert AD). Bei zeilenorientierter Arbeitsweise werden per Default die folgenden Zeichen von CP für die Datenübertragung ausgewertet (vgl. 3.1.3):

```
# neue logische Zeile beginnen
ç logische Zeile löschen
@ vorstehendes Zeichen löschen
" nachfolgendes Zeichen nicht auswerten
! wird auf das Zeichen mit hexadezimalen Wert 05 abgebildet
```

Ausgabe: Auf dem Bildschirm darstellbar sind neben den EBCDIC Schriftzeichen noch andere Zeichen (Schriftzeichen in EBCDIC Lücken, APL Zeichen), deren Bedeutung anwendungsspezifisch ist.

Info-Zeile: So heißt die Zeile unter der durchgezogenen Linie; sie zeigt neben der Cursor Position ggf. einen besonderen Zustand des Terminals an (z.B. ^ nach <INS>, Männchen falls Tastatur wegen Schreibschutz gesperrt, APL falls Tastatur auf APL geschaltet); das Terminal kann mit <RESET> normiert werden.

Taste	Funktion
< >	umschalten (oben)
<ALT>	umschalten (vorne)
<APL ON/OFF>	umschalten EBCDIC/APL
<ENTER>	übertragen von Eingaben, vgl. 2.1
<PA1>	unterbrechen der virtuellen Maschine, der Status geht über in CP READ, vgl. Abb. 3-3
<PA2>	löschen des Bildschirms; ggf. Ausgabe fortsetzen, vgl. Abb. 3-5
<RESET>	Terminal normieren
<CLICK>	Klicken bei Cursor Bewegungen ein/ausschalten
<PF1> bis <PF24>	sind mit dem CP Command SET PFnn programmierbar, vgl. 3.1.3; können auch von Programmen (z.B. Editor) programmiert werden
<←><↑><↓><→>	Cursor positionieren
<→ > < ←>	Cursor an den Anfang des nächsten schreibfreien Bereichs positionieren
<←>	Cursor an den Anfang der nächsten schreibfreien Zeile positionieren
	Zeichen auf Cursor Position löschen
<INS>	INSert Mode einschalten; ermöglicht das Einfügen von Zeichen auf Cursor Position (im XEDIT vorher <PA2> erforderlich); ausschalten mit <RESET>

Tabelle 2-1: wichtige Funktionen des MRX 2078

2.3 Graphik Terminals

Das IBM Terminal 3279 ist (über Steuereinheiten) direkt an die IBM 4381 angeschlossen. Es verhält sich alphanumerisch im wesentlichen wie das Standard Terminal MRX 2078, lediglich einige Tasten sind anders beschriftet (die Zuordnung ergibt sich durch Vergleich mit Abb. 2-1).

Das IBM Terminal 3192 ist (über Steuereinheiten) direkt an die IBM 4381 angeschlossen. Es verhält sich alphanumerisch im wesentlichen wie das Standard Terminal MRX 2078; es hat einige zusätzliche Tasten (z.B. numerischer Block zur Eingabe von Ziffern, PF13 bis PF24), die Tasten sind anders angeordnet, einige Tasten sind anders beschriftet (â statt INS, ~~⌘~~ statt DEL).

Das TEK Terminal CX41XX hat zwei Anschlüsse (Coaxial und asynchron); mit dem Coaxial-Anschluß ist es (über eine Steuereinheit) direkt an die IBM 4381, mit dem asynchronen Anschluß an einen der beiden Rechner DEC VAX 11/750 angeschlossen (ggf. über DEVELNET); die Wahl des Anschlusses erfolgt im Setup-Mode des Terminals mit

```
<Setup>  
*dalines 32 <<←>  
*host coax <<←>  
<Setup>
```

vgl. Bedienungsanleitung von TEKTRONIX. Das CX41XX verhält sich an der IBM 4381 alphanumerisch im wesentlichen wie das Standard Terminal MRX 2078 (mit EBCDIC Tastatur), lediglich einige Tasten sind anders beschriftet (die Zuordnung ergibt sich durch Vergleich mit Abb. 2-1).

Alle Terminals haben grafikfähige Farbbildschirme; die Dialogfarben sind mit dem CP Command SCREEN einstellbar. Graphische Ausgaben sind i.a. nur auf den IBM Terminals möglich; das TEK CX41XX wird nur von SAS/GRAPH unterstützt. (vgl. Notiz ANW SAS/GRAPH).

2.4 SPERRY Terminal UTS20

Ein UTS20 am DCP/40 der SPERRY 1100/60 kann wie das Standard Terminal MRX 2078 an der IBM 4381 benutzt werden; der 3270 Inverted Handler im DCP/40 (Komponente der DCP/40 Software TELCON) verbindet das Terminal mit der IBM 4381. Der 3270 Inverted Handler emuliert MRX 2078 Funktionen mit Hilfe von UTS20 Funktionen, vgl. MRX 2078 Emulation in der Bedienungsanleitung des Terminals.

2.5 DEVELNET Terminals

Ein Terminal QVT201, VT220, VT320 am DEVELNET kann wie das Standard Terminal MRX 2078 an der IBM 4381 benutzt werden. Das DEVELNET verbindet das Terminal über eine 7171 ASCII-Steuereinheit mit der IBM 4381. Terminals am DEVELNET müssen sich gegenüber dieser Steuereinheit wie ein DEC Terminal VT100 verhalten; sie emuliert MRX 2078 Funktionen mit Hilfe von VT100 Funktionen, vgl. MRX 2078 Emulation in der Bedienungsanleitung des Terminals.

3. Arbeiten im Dialog-Betrieb

Die Benutzung der IBM 4381 muß beantragt werden, vgl. BENUTZUNG im Teil ALLGEMEINES. Das HRZ teilt dem Benutzer eine Userid mit Logon Paßwort KEINES zu; dieses Paßwort muß schnellstmöglich geändert werden, vgl. 3.4.1 und BENUTZERBERECHTIGUNGEN. Unter einer Userid kann zu einem Zeitpunkt nur eine Session geführt werden.

Zum Kennenlernen können potentielle Benutzer folgende Userids verwenden:

1STEIN	}	mit Logon Paßwort KEINES
2STEIN		
.		
.		
8STEIN		

Freie Userids werden beim Beginnen einer Session mit Userid STEIN und Logon Paßwort KEINES mitgeteilt, vgl. 6. Jede dieser Userids hat eine permanente Minidisk mit Lese-Paßwort ALBERT und Zugriffsart WR (vgl. 4.1 und FILES); der CP Directory Eintrag (insbesondere das Logon Paßwort) kann nicht geändert werden.

3.1 Dialog-Betrieb

Die grundsätzliche Struktur für das Arbeiten im Dialog-Betrieb finden Sie in Abschnitt 1.

3.1.1 Verbindung aufbauen

Das explizite Aufbauen einer Verbindung zur IBM 4381 ist nur an den Terminals nötig, die an den Datenübertragungsrechner DCP/40 oder an das DEVELNET angeschlossen sind, d.h. an UTS20 und DEVELNET Terminals. Der Verbindungsaufbau wird in der Bedienungsanleitung dieser Terminals beschrieben. Am CX41XX muß ggf. der Coaxial-Anschluß gewählt werden (vgl. 2.3).

3.1.2 Session beginnen (Logon)

Im Dialog-Betrieb können Sessions geführt werden; das Beginnen einer Session wird Logon genannt.

Eine Session kann begonnen werden, wenn das VM/SP Logo (vgl. Abb. 3-1) auf dem Bildschirm steht. Es erscheint beim Einschalten eines direkt angeschlossenen Terminals bzw. nach dem erfolgreichen Aufbau einer Verbindung, falls die IBM 4381 betriebsbereit ist (Status RUNNING).

VIRTUAL MACHINE/SYSTEM PRODUCT

```

VV      VV  MM      MM
VV      VV  MMM     MMM
VV      VV  MMMM    MMMM
VV  VV  MM MM MM MM  SSSSSSSS  PPPPPPPP
VV VV  MM  MMM  MM  SS      SS  PP      PP
VVV    MM  M   MM  SS      SS  PP      PP
V      MM      MM  SSSSSSS  PPPPPPPP
                                     SS  PP
                                     SS  SS  PP
                                     SSSSSSS  PP

```

HRZ UNI MARBURG - ZENTRALER RECHNER IBM 4381

Fill in your USERID and PASSWORD and press ENTER
(Your password will not appear when you type it)

USERID ==>

PASSWORD ==>

COMMAND ==>

Abb. 3-1: VM/SP Logo

Eine Session wird mit dem CP Command LOGON begonnen; dieses Command wird implizit ausgeführt, wenn Userid und Logon Paßwort in die entsprechenden Felder des VM/SP Logos eingegeben werden (das COMMAND Feld bleibt leer):

```

USERID  ---> lstein <-->
PASSWORD ---> keines <ENTER>

```

Beim Eingeben des Logon Paßworts wird nur der Cursor bewegt, die eingegebenen Zeichen erscheinen nicht auf dem Bildschirm. Nach bestandener Prüfung von Userid und Logon Paßwort werden nun die Nachrichten des HRZ (Log Message) ausgegeben. Der Status geht über in VM READ. Mit

<ENTER>

wird nun (bei allen Userid's) die

System Logon Prozedur

ausgeführt, die u.a. den Zugriff auf gewisse System- und Anwender-Software ermöglicht (LINK und ACCESS Commands, vgl. 4.1) und die RETRIEVE Taste definiert (vgl. 3.1.3). Danach wird die (zu dieser Userid gehörende)

User Logon Prozedur

im File PROFILE EXEC A ausgeführt, falls dieser File existiert. Das Command GETPROF liefert ein Beispiel für eine User Logon Prozedur, die weitere System- und Anwender-Software bereitstellt:

getprof <ENTER>

kopiert das Beispiel in den File PROFILE EXEC A; dieser File kann mit Hilfe des Editors XEDIT geändert werden (zum Arbeiten mit Prozeduren vgl. KOMMANDOSPRACHEN und Chapter 8 im CMS Primer). Falls mit dem Inhalt dieser neuen User Logon Prozedur innerhalb der Session weitergearbeitet werden soll, muß sie noch ausgeführt werden mit

profile <ENTER>

Falls die Ausgaben der User Logon Prozedur nicht auf den Bildschirm passen (Status MORE...), ist mit <PA2> bis zur Ready Message (s.u.) weiterzublättern (Status RUNNING).

Mögliche Fehlermeldungen beim Logon:

- DMKCFM288E LOGON from the initial screen was unsuccessful:
Fehlerhafte Eingabe von Userid und Logon Paßwort.
- DMKLOG053E *userid* not in CP directory:
Userid ist nicht im CP Directory eingetragen oder gesperrt.
- DMKLOG054E Already logged on ...:
Unter dieser Userid wird bereits eine Session geführt (mehrere Sessions mit der gleichen Userid sind nicht möglich).
- DMKLOG050E LOGON unsuccessful - incorrect password:
Falsches Logon Paßwort.

Status ist CP READ; es muß ein LOGON Command explizit eingegeben werden; z.B.:

logon 2stein <ENTER>

ENTER PASSWORD (IT WILL NOT APPEAR WHEN TYPED):

keines <ENTER>

Groß/Kleinschreibung: Kleine Buchstaben werden wie große interpretiert.

3.1.3 Commands eingeben

Nach dem Logon können CMS Commands oder CP Commands eingegeben werden. Eine Liste aller CP bzw. CMS Commands erhalten Sie mit HELP, vgl. 3.3.1. Die Eingabe von Commands erfolgt per Default im Line Mode (d.h. zeilenorientiert). Es ist auch möglich, mit CMS im Full Screen Mode zu arbeiten; davon wird jedoch abgeraten: Der Full Screen Mode bietet keinen wesentlich größeren Komfort als der Line Mode, verbraucht aber deutlich mehr Betriebsmittel. Im folgenden wird nur der Line Mode beschrieben; dieser bietet die Möglichkeiten:

- Commands einzeln einzugeben
- mehrere Commands auf einmal einzugeben
- neue Commands einzugeben, bevor die alten abgearbeitet sind

Von den letzten beiden Möglichkeiten sollte nur in Ausnahmefällen Gebrauch gemacht werden. Vielmehr wird die in Abb. 3-2 dargestellte Arbeitsweise empfohlen: ein Command eingeben - warten bis Ausführung beendet - nächstes Command eingeben. Die Ausführung einer Prozedur oder eines Programms wird von CMS wie die Ausführung eines Commands behandelt. CMS gibt nach der Ausführung eines Commands entweder eine Ready Message aus in der Form

Ready; T=virtuelle CPU-Zeit/totale CPU-Zeit Uhrzeit

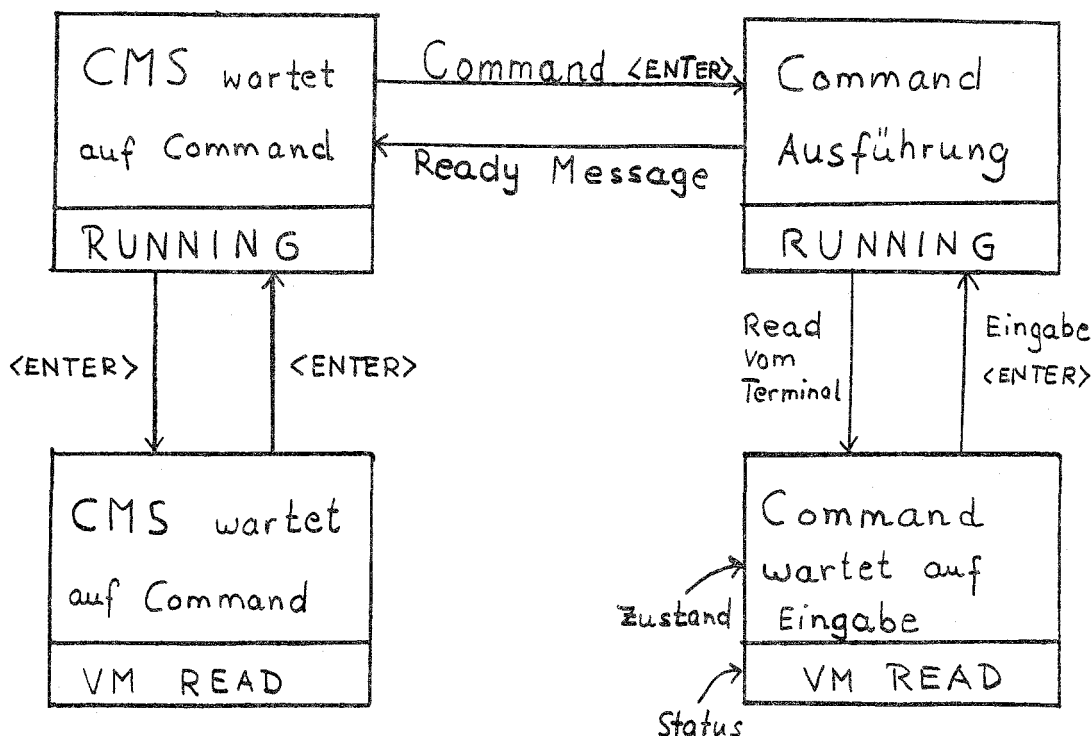


Abb. 3-2: Command - Ready Message - Command

(virtuelle CPU-Zeit = von der virtuellen Maschine verbrauchte CPU-Zeit, totale CPU-Zeit = virtuelle CPU-Zeit + von CP verbrauchte CPU-Zeit in Sekunden) oder - falls wegen falschem Command Name kein Command ausgeführt wurde (Tippfehler) - die Message UNKNOWN CP/CMS COMMAND.

Korrigieren im Eingabebereich: Vor dem Übertragen (<ENTER>) kann im Eingabebereich mit Hilfe der in 2.2 beschriebenen Terminal Funktionen (z.B. ,<INS>) beliebig korrigiert werden; der Cursor muß vor dem <ENTER> nicht ans Ende der Eingabe positioniert werden.

Die Sonderzeichen # ¢ @ " ; haben in Eingaben besondere Funktionen (Logical Line Editing). CP wertet (per Default) diese Zeichen aus, bevor es eingegebene Zeichen ggf. an die virtuelle Maschine weiterreicht. Eine Eingabe, die mit #CP beginnt, z.B.

#cp query time <ENTER>

wird nicht weitergereicht, sondern sofort von CP ausgeführt. Eine Eingabe kann in (logische) Eingabezeilen eingeteilt sein, Trennzeichen ist #; jede Eingabezeile kann ein Command enthalten, z.B.

fortvs programm#load programm#start <ENTER>

zum Compilieren, Binden und Starten eines FORTRAN Programms. Eingabezeilen, die nicht sofort bearbeitet werden können, werden im Terminal Eingabepuffer zwischengespeichert. Falls # nicht Trennzeichen sein soll, ist "# statt # zu verwenden (" unterdrückt die Auswertung des nachfolgenden Zeichens), z.B.

set pf05 "#cp query time <ENTER>

belegt die Taste PF5 mit der Zeichenfolge #cp query time. Die Funktionen von ¢, @ und ; werden an Bildschirm Terminals nicht benötigt; ¢ löscht die logische Zeile, in der es steht, @ löscht das vorstehende Zeichen; ; wird von CP auf das Zeichen mit hexadezimalen Wert 05 abgebildet. Die System Logon Prozedur schaltet die Auswertung von ¢, @ und ; ab; auch die Auswertung von # und " kann abgeschaltet werden; statt # @ ¢ " können auch andere Zeichen gewählt werden (CP Command TERMINAL).

PF Tasten belegen mit Commands: Mit Hilfe des CP Commands **SET PFnn** können die PF Tasten mit beliebigen Zeichenfolgen belegt werden. Nach Ausführung des obigen SET PF05 Commands wird mit <PF5> das Command #cp query time in den Eingabebereich kopiert; es muß mit <ENTER> übertragen werden. PF Tasten können auch so belegt werden, daß das Command sofort übertragen wird (IMMED), z.B.

```
set pf03 immed filelist * fortran <ENTER>
```

Danach kann mit <PF3> das CMS Command FILELIST * FORTRAN ausgeführt werden.

```
set pf03 <ENTER>
```

löscht die Belegung der Taste PF3. Auflisten aller PF Tastenbelegungen erfolgt mit

```
query pf <ENTER>
```

Es wird empfohlen, die entsprechenden SET und QUERY Commands in die User Logon Prozedur zu schreiben (z.B. mit XEDIT PROFILE EXEC, vgl. KOMMANDOSPRACHEN)

Commands zurückrufen: Eine der PF Tasten kann als RETRIEVE Taste definiert werden, z.B.

```
set pf06 retrieve <ENTER>
```

Danach werden von CP die jeweils letzten Eingaben gepuffert (Puffer fester Länge, ringförmig organisiert); <PF6> kopiert die letzte Eingabe in den Eingabebereich (sie kann ggf. korrigiert und mit <ENTER> übertragen werden); mehrmaliges <PF6> liefert ältere Eingaben. Die System Logon Prozedur definiert PF6 als RETRIEVE Taste.

Groß/Kleinschreibung: kleine Buchstaben werden wie große interpretiert.

Abkürzungen: Zulässige Abkürzungen in Commands sind in den Dokumentationen mit großen Buchstaben dargestellt; Abkürzungen für Command Names können auch vom Benutzer mit Hilfe des CMS Commands SYNONYM festgelegt werden; ohne Operanden listet dieses Command die möglichen Abkürzungen von Command Names am Terminal auf:

```
synonym <ENTER>
```

Fortsetzung von Commands ist nur in Prozeduren nicht aber bei Terminal Eingaben möglich.

3.1.4 Session unterbrechen (disconnect) / fortsetzen (reconnect)

Mit dem CP Command DISCONN kann die virtuelle Maschine vom Terminal disconnected, d.h. getrennt werden. Im Zustand "CMS wartet auf Command" (vgl. Abb. 3-2) unterbricht

```
disconn <ENTER>  
DISCONNECT AT ...
```

die Session. <ENTER> liefert das VM/SP Logo (vgl. Abb. 3-1). Die Session kann (an einem beliebigen anderen Terminal) durch Logon mit der gleichen Userid (vgl. 3.1.2) fortgesetzt werden. Die virtuelle Maschine wird dabei mit diesem Terminal **reconnected** (Status CP READ):

```
RECONNECTED AT ...  
begin <ENTER>
```

Hinweis: Während der Ausführung eines Programms kann die virtuelle Maschine vom Terminal disconnected werden mit

```
#cp disconn <ENTER>
```

Die Ausführung des Programms wird fortgesetzt; versucht das Programm vom Terminal zu lesen (CONSWAIT), wird die Session abgebrochen, falls die virtuelle Maschine nicht innerhalb von 15 Minuten reconnected wird.

3.1.5 Session beenden (Logoff)

Das Beenden einer Session wird **Logoff** genannt; der Logoff erfolgt mit dem CP Command LOGOFF. Logoff Prozeduren (analog zu PROFILE EXEC) gibt es nicht.

```
logoff <ENTER>
```

wird von CP mit folgender Ausgabe quittiert (zur Interpretation der CPU-Zeiten vgl. 3.1.3):

```
CONNECT=hh:mm:ss VIRTCPU=hh:mm:ss TOTCPU=hh:mm:ss  
LOGOFF AT hh:mm:ss CET weekday yy:mm:dd
```

Press enter or clear key to continue

Falls Sie das Terminal nicht ausschalten, drücken Sie bitte noch auf ENTER. Auf dem Bildschirm erscheint das VM/SP Logo, vgl. Abb. 3-1. Es zeigt dem nachfolgenden Benutzer, daß das Terminal frei ist.

3.1.6 Verbindung abbauen

Das explizite Abbauen der Verbindung zur IBM 4381 ist nur an den Terminals möglich, die an den Datenübertragungsrechner DCP/40 oder an das DEVELNET angeschlossen sind, also an UTS20 und DEVELNET Terminals, und wird in der Bedienungsanleitung dieser Terminals beschrieben. Für DEVELNET Terminals wird beim Logoff implizit die Verbindung abgebaut, am UTS20 muß die Verbindung explizit abgebaut werden, wenn nicht sofort eine neue Session begonnen wird.

Beim Abbau der Verbindung (UTS20, DEVELNET Terminal) innerhalb einer Session wird die Session unterbrochen (disconnect); sie kann innerhalb von 15 Minuten fortgesetzt werden (reconnect, 3.1.4).

3.2 Unterbrechen, Fortsetzen, Abbrechen

Das Arbeiten mit diesen Funktionen ist zweistufig; sie können entweder auf Ein/Ausgaben oder auf die Programme angewendet werden, die die Ein/Ausgaben erzeugen:

Programm beginnen

·
·
· Programm unterbrechen
· Programm fortsetzen/abbrechen

·
·
· Eingabe beginnen

·
·
· Eingabe abbrechen/beenden

·
·
· Ausgabe beginnen

·
·
· Ausgabe unterbrechen
· Ausgabe fortsetzen/abbrechen

·
·
· Ausgabe abbrechen/beenden

·
·
· Programm abbrechen/beenden

Alle Aussagen des Abschnitts 3.2 gelten nicht nur für Programme, sondern ganz analog auch für CMS Commands und Prozeduren.

3.2.1 Programme, CMS Commands, Prozeduren

Ein Programm (bzw. ein CMS Command oder eine Prozedur) kann im Status RUNNING oder VM READ durch <PA1> oder durch #CP<ENTER> unterbrochen werden. Der Status geht über in CP READ; anschließend können CP Commands eingegeben werden, z.B. QUERY TIME (vgl. 3.3.2); die Commands werden von CP bearbeitet; es gibt keine Ready Message. Mit dem CP Command BEGIN wird das unterbrochene Programm fortgesetzt, vgl. Abb. 3-3.

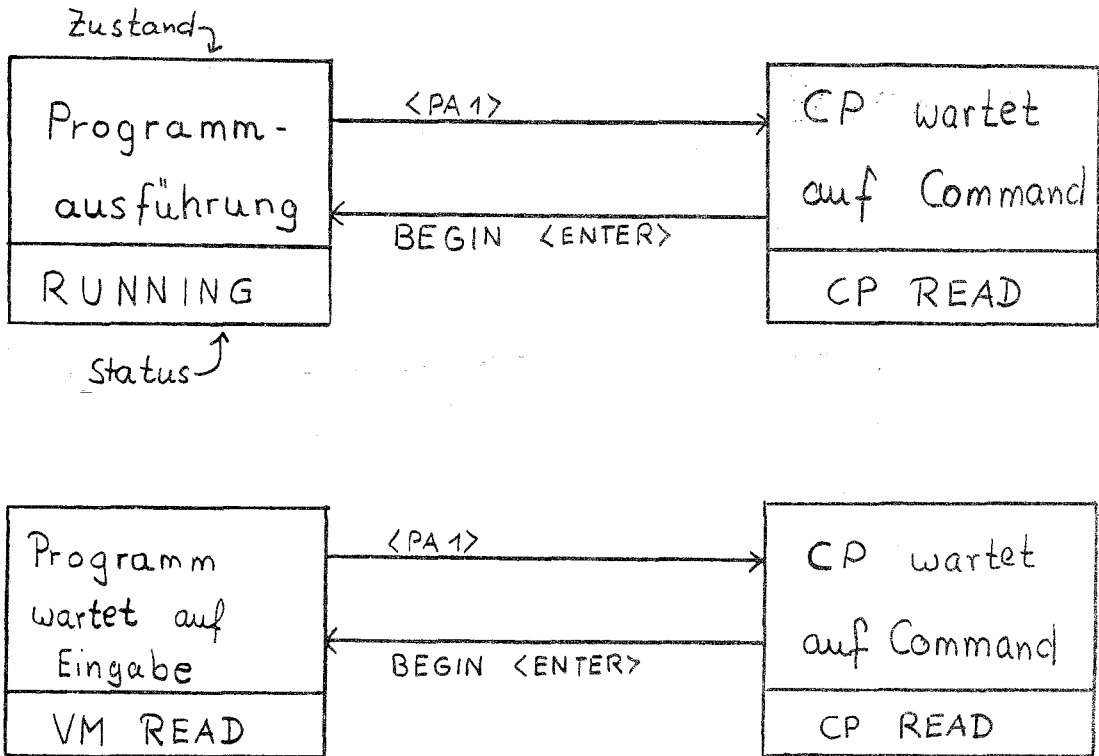


Abb. 3-3: Programm unterbrechen/fortsetzen

Ein Programm kann im Status RUNNING **abgebrochen** werden mit dem CMS Immediate Command HX. Wenn das Programm auf eine Terminal Eingabe wartet (Status VM READ), wird "HX" vom Programm gelesen und nicht von CMS ausgeführt; in diesem Fall muß zuerst das Programm unterbrochen werden mit <PA1>; danach kann die virtuelle Maschine mit dem CP Command EXTERNAL in den DEBUG Mode versetzt werden, sie erwartet dann DEBUG Subcommands; das DEBUG Subcommand HX bricht das unterbrochene Programm ab, vgl. Abb. 3-4.

Vorsicht: HX löscht den Program Stack, den Terminal Eingabepuffer sowie alle FILEDEFS (auch die permanenten).

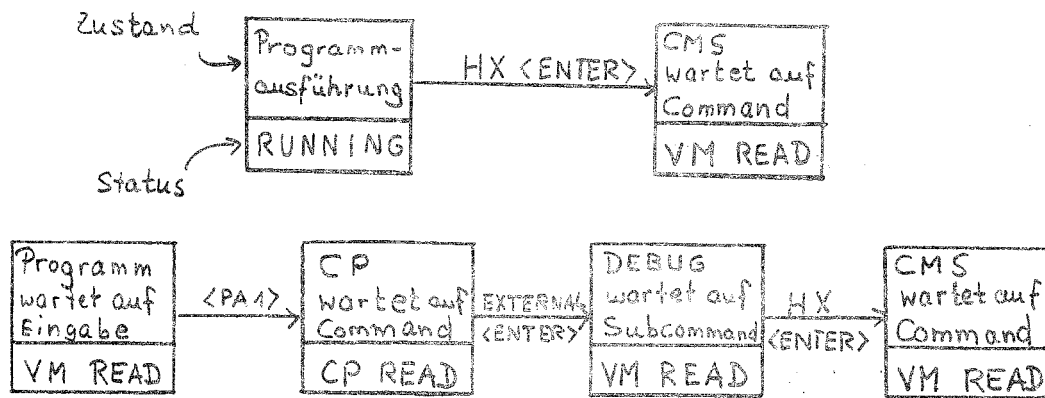


Abb. 3-4: Programm abbrechen

3.2.2 Ein/Ausgaben

Ausgaben: Ausgaben auf das Terminal können zeilenorientiert (Normalfall) oder bildschirmorientiert (XEDIT, FILELIST) erfolgen.

Bei der zeilenorientierten Ausgabe werden so lange Zeilen ausgegeben, bis der Bildschirm voll ist; dann wird die Ausgabe von CP unterbrochen; der Status geht in der Regel über in MORE..., unter speziellen Bedingungen in HOLDING. Mit

<PA2>

kann der Benutzer die Ausgabe fortsetzen; im Status MORE... wird die Ausgabe nach einer Wartezeit von 1 Minute ggf. von CP fortgesetzt, vgl. Abb. 3-5. Im Status MORE... oder HOLDING kann der Benutzer die Ausgabe abbrechen mit

ht<ENTER>

Das CMS Immediate Command HT (Halt Typing) unterdrückt alle weiteren Ausgaben des Programms.

Bei der bildschirmorientierten Ausgabe wird immer genau ein Bildschirm ausgegeben; weitere Ausgaben müssen explizit angefordert werden (bei XEDIT z.B. mit dem Subcommand FORWARD bzw. <PF8>). Die Funktionen Unterbrechen, Fortsetzen, Abbrechen werden somit bei der bildschirmorientierten Ausgabe nicht benötigt.

Eingaben: Die Eingabe von Zeilen (z.B. nach INPUT beim Editieren mit XEDIT) wird i.a. durch eine leere Eingabe (nur <ENTER>) abgebrochen. Eine leere Eingabe liefert auch die END-Bedingung beim Lesen vom Terminal im FORTRAN Programm (z.B. READ(5,END=99,...) ...).

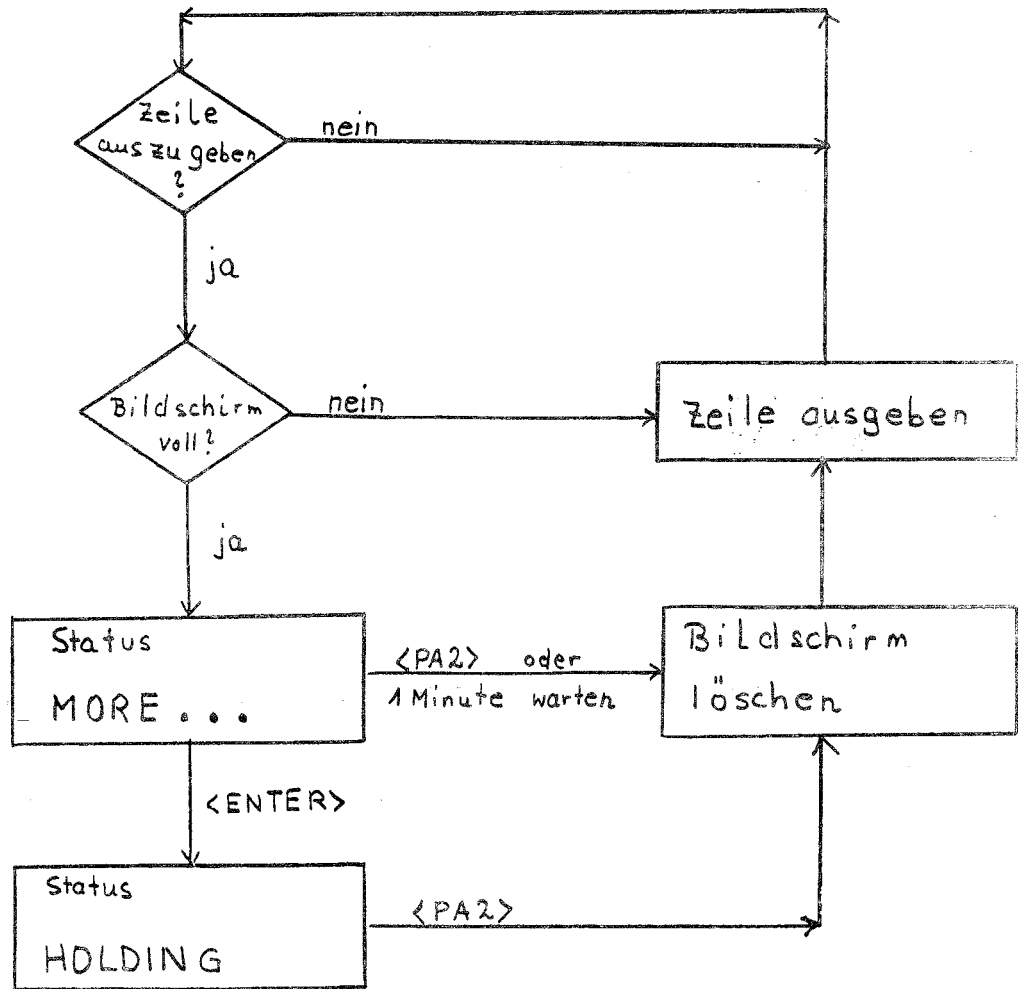


Abb. 3-5: Zeilenorientierte Ausgabe

3.2.3 Normieren des Terminals

Falls die Tastatur gesperrt ist (Männchen in der Info-Zelle eines MRX 2078) kann das Terminal normiert werden mit

<RESET>

Mit <PA1> kann ggf. ein Command/Programm unterbrochen werden (Status CP READ) und die Session mit LOGOFF beendet werden; wenn danach durch <ENTER> das VM/SP Logo ausgegeben wird, ist das Terminal i.a. normiert.

Die Normierung von UTS20 bzw. DEVELNET Terminals wird in der Bedienungsanleitung beschrieben.

3.3 Informieren

3.3.1 Leistung und Format von Commands

Das CMS Command **HELP** gibt Informationen am Terminal aus über Leistung und Format von

- CP Commands
- CMS Commands
- DEBUG Subcommands (vgl. 3.2.1)
- REXX Instructions (vgl. KOMMANDOSPRACHEN)
- XEDIT Subcommands (vgl. 5.1)

Zum Arbeiten mit **HELP** vgl. KOMMANDOSPRACHEN. **HELP** ohne Operanden gibt ein Menü zur Auswahl der gewünschten Information aus:

help <ENTER>

3.3.2 Verbrauchte CPU-Zeit

Das CP Command **QUERY TIME** gibt u.a. die vom Job verbrauchte virtuelle und totale CPU-Zeit in *Stunden:Minuten: Sekunden* am Terminal aus. Das Command kann in der Form

#cp query time <ENTER>

auch während der Ausführung eines Programms eingegeben werden; es wird dann sofort von CP ausgeführt (vgl. 3.1.3).

3.3.3 Belegter Plattenspeicher

Das CMS Command **QUERY DISK** informiert am Terminal über die Belegung aller Minidisks, auf die die Session einen Access gesetzt hat (vgl. 4.1 und FILES):

query disk <ENTER>

3.3.4 Files der eigenen Minidisk

Das CMS Command **FILELIST** listet ohne Operanden alle Files der Minidisk A am Terminal auf (vgl. 4.1 und Chapter 3 im CMS Primer):

filelist <ENTER>

3.3.5 Benutzer-Berechtigungen

Das CMS Command DIRMAINT REVIEW gibt alle Benutzer-Berechtigungen zur Userid der Session in einen Spool File der Reader Queue (Filename: *userid*, Filetype: DIRECT) aus; dieser Spool File kann mit Hilfe des CMS Commands RDRLIST am Terminal ausgegeben werden, vgl. 4.2.4.

```
dirmaint review <ENTER>  
...Auf Anforderung Logon Paßwort eingeben...  
rdrlist <ENTER>  
...Cursor vor entsprechenden Filename positionieren...  
<PF11>
```

Die Minidisk Paßwörter der eigenen permanenten Minidisks werden nicht mit ausgegeben. Sie können jedoch am Terminal mit dem CMS Command DIRMAINT MDPW aufgelistet werden:

```
dirmaint mdpw <ENTER>  
...Auf Anforderung Logon Paßwort eingeben...
```

3.3.6 Nachrichten, Mitteilungen

Die Nachrichten des HRZ werden beim Logon am Terminal ausgegeben (Log Message); dies kann innerhalb einer Session wiederholt werden mit dem CP Command

```
query logmsg <ENTER>
```

Die Mitteilungen des HRZ können mit Hilfe der Prozeduren NEU und INFO am Terminal ausgegeben werden (vgl. Notiz NACHRICHTEN/MITTEILUNGEN); erforderliche Angaben werden mit den entsprechenden Erläuterungen angefordert:

```
neu <ENTER>  
info <ENTER>
```

3.4 Ändern des CP Directory Eintrags

Das CP Directory enthält zu jeder Userid einen Eintrag, der z.B. Kenndaten des Benutzers, Defaults und Schranken zu Betriebsmitteln sowie Größe, Paßwörter und Zugriffsart der eigenen permanenten Minidisks enthält; einige dieser Angaben können vom Benutzer mit Hilfe des CMS Commands DIRMAINT geändert werden, vgl. BENUTZER-BERECHTIGUNGEN. Der erste Operand des DIRMAINT Commands legt fest, welche Angabe geändert werden soll; alle weiteren Informationen werden angefordert. Beim Eingeben von Paßwörtern erscheinen die eingegebenen Zeichen nicht auf dem Bildschirm.

3.4.1 Logon Paßwort ändern

Das Logon Paßwort wird mit Hilfe des CMS Commands **DIRMAINT PW** geändert:

```
dirmaint pw <ENTER>
```

```
DVHDIR004R ENTER NEW CP LOGON PASSWORD:  
geheim <ENTER>
```

```
DVHDIR005R ENTER CURRENT CP PASSWORD TO VALIDATE COMMAND OR  
A NULL TO EXIT:  
keines <ENTER>
```

Ready; T- ...

```
DVHMGB075I COMMAND DIRM PW: SOURCE UPDATED AND CHANGE ONLINE
```

Die eingegebenen Paßwörter erscheinen nicht auf dem Bildschirm. Das Logon Paßwort muß aus 6 bis 8 alphanumerischen Zeichen bestehen; es muß mindestens alle 90 Tage geändert werden, andernfalls wird die Userid gesperrt, vgl. BENUTZERBERECHTIGUNGEN. Falls Sie nicht berechtigt sind, Ihren Eintrag im CP Directory zu ändern, erhalten Sie statt der letzten Meldung eine Fehlermeldung:

```
DVHDMA033E COMMAND DIRM PW REJECTED; LOCKED SINCE ...  
CONTACT OPERATIONS STAFF.
```

3.4.2 Minidisk Paßwörter und implizite Zugriffsart ändern

Minidisk Paßwörter und implizite Zugriffsart für die eigenen permanenten Minidisks (vgl. 4.1 und FILES) werden mit Hilfe des CMS Commands **DIRMAINT MDISK** geändert, z.B.

```
dirmaint mdisk <ENTER>
```

```
...  
Minidisk Adress ----> 191<-->|>
```

```
...  
Access Mode ----> wr<-->|>
```

```
...  
Read password ----> rp17s345<ENTER>
```

...
setzt für die eigene permanente Minidisk 191 die implizite Zugriffsart auf **WR** und das Lese-Paßwort auf **rp17s345** (vgl. 4.1 und FILES). Vor Ausführung des Commands wird (wie in 3.4.1) das Logon Paßwort angefordert und überprüft.

3.5 Drucken

Eine Liste aller System-Drucker (mit Name und Standort) gibt die Prozedur INFO am Terminal aus:

info drucker <ENTER>

3.5.1 Bildschirminhalt drucken

MRX 2078, IBM 3279: im Prinzip möglich, jedoch am HRZ nicht vorgesehen. Es wird empfohlen, die Session zu protokollieren, vgl. 4.2.1.

IBM 3192/Hardcopy: Der Bildschirminhalt kann nur an dem Hardcopygerät, das an das Terminal angeschlossen ist, ausgedruckt werden. Hardcopygerät einschalten, ggf. mit <Alt/Print> aktivieren (in der Info-Zeile muß das Symbol stehen), dann <Print>.

TEK CX41XX/Hardcopy: Der Bildschirminhalt kann nur an dem Hardcopygerät, das an das Terminal angeschlossen ist, ausgegeben werden. Es sollte jedoch nur zur Ausgabe einer Graphik verwendet werden. Hardcopygerät einschalten, Dialogbereich mit <Dialog> ausblenden, dann <S Copy>.

SPERRY UTS20/Matrixdrucker: im Prinzip möglich, jedoch fehlerträchtig (Terminalparameter müssen umgestellt werden); es wird empfohlen, die Session zu protokollieren, vgl. 4.2.1.

DEVELNET Terminal/Matrixdrucker: Der Bildschirminhalt kann nur an dem Matrixdrucker, der an das Terminal angeschlossen ist, ausgegeben werden. Drucker ein- und auf ON LINE schalten, dann <PRINT> bzw. äquivalente Taste drücken, vgl. VT100 Beschreibung in Bedienungsanleitung.

3.5.2 Protokollieren einer Session

Das Protokollieren einer Session erfolgt auf der IBM 4381 mit Hilfe eines **Console Spool Files**. Das Beginnen und Beenden des Protokollierens sowie das Drucken des Console Spool Files wird unter 4.2.1 beschrieben. Die gleichzeitige Ausgabe aller Terminal Ausgaben auf einen an das Terminal angeschlossenen Matrixdrucker ist nicht möglich.

3.5.3 File-Inhalt drucken

Das Drucken eines File-Inhalts erfolgt auf der IBM 4381 mit Hilfe des CMS Commands PRINT. Welche Files mit PRINT gedruckt werden können sowie die Beschreibung des PRINT Commands steht in Abschnitt 4.2.2; die möglichen Drucker liefert die Prozedur INFO (s.o.). Darüber hinaus kann mit Hilfe des FILETRANSFERS auf weitere Drucker des HRZ zugegriffen werden, vgl. FILETRANSFER im Teil ALLGEMEINES.

4. Arbeiten mit Files

4.1 Files auf Minidisks

Programme und Daten werden in Files auf **Minidisks** gehalten; wenn Sie Programmentwicklung an der IBM 4381 machen wollen, müssen Sie einiges über Files und Minidisks wissen. Das Wichtigste ist im folgenden beschrieben (vgl. auch Chapter 3 im CMS Primer), weitergehende Fragen behandelt das Kapitel FILES.

Jedem Benutzer (Userid) gehört (i.a. mindestens) eine **permanente Minidisk** (vgl. Plattenspeicher-Kontingent in BENUTZERBERECHTIGUNGEN); auf dieser Minidisk hält er seine permanenten Files. Für die Dauer der Session kann er darüber hinaus eine **temporäre Minidisk** für temporäre Files einrichten (CP Command DEFINE T3380).

File Identification: In CMS Commands hat eine File Identification zur Bezeichnung eines Files folgendes Format

fn ft fm

(z.B. QUELLE FORTRAN A), wobei *fn ft* auf der Minidisk eindeutig sein muß.

Der **Filename** *fn* ist frei wählbar (max. 8 alphanumerische Zeichen).

Der **Filetype** *ft* (max. 8 alphanumerische Zeichen) ist in der Regel vorgeschrieben, z.B. FORTRAN für einen File, der FORTRAN Quellprogramme enthält. Es gibt CMS Commands, in denen muß *ft* angegeben werden (z.B. XEDIT), in CMS Commands mit vorgeschriebenem Filetype darf *ft* nicht angegeben werden (z.B. FORTVS, LOAD).

Der **Filemode** *fm* besteht aus einem Buchstaben - dem Filemode Letter *fml* - ggf. gefolgt von einer Ziffer - der Filemode Number. Der **Filemode Letter** *fml* bezeichnet die Minidisk; die eigene permanente Minidisk wird i.a. mit dem Filemode Letter A bezeichnet; die Filemode Letters werden beim Access setzen (s.u.) festgelegt. Default für den Filemode Letter ist in der Regel A beim Schreiben und * beim Lesen (z.B. XEDIT); * bedeutet, daß alle Minidisks, auf die der Job einen Access gesetzt hat, nach dem File durchsucht werden (in der Reihenfolge der Filemode Letters). Die **Filemode Number** *fmn* ($0 \leq fmn \leq 6$) gibt gewisse Eigenschaften des Files an (vgl. FILES); sie wird beim Informieren mit ausgegeben; sie ist zur Bezeichnung eines Files i.a. ohne Bedeutung und wird im folgenden stets weggelassen.

Arbeiten mit Minidisks: Vor dem Arbeiten mit Files auf einer Minidisk muß in der Session ein Link und ein Access auf die Minidisk gesetzt werden; für die eigenen permanenten Minidisks sowie für einige Minidisks mit System- und Anwender-Software werden Link und Access implizit beim Logon bzw. explizit von der System Logon Prozedur gesetzt; nur wer auf andere permanente Minidisks zugreifen will, muß Link und Access selbst explizit setzen, vgl. FILES. Grundsätzlich gilt folgende Struktur:

```
Link setzen (auf Minidisk; CP Command LINK)
  Access setzen (auf Minidisk; CMS Command ACCESS)
    Open (File)
      Read/Write (Satz)
    Close (File)
  Access lösen (von Minidisk, CMS Command RELEASE)
Link lösen (von Minidisk, CP Command DETACH)
```

Beim Logoff werden implizit alle Files geschlossen sowie alle Accesses und Links gelöst.

Per Default kann nur der Eigentümer auf seine permanente Minidisk zugreifen. Es wird dringend empfohlen, der Minidisk ein Lese-Paßwort zu geben, damit das HRZ diese Minidisk in die CMS-File-Sicherung einbeziehen kann; andere Benutzer können dann bei Kenntnis dieses Paßworts lesend auf die Minidisk zugreifen; gleichzeitig sollte die implizite Zugriffsart, d.h. die Zugriffsart für den impliziten Link beim Logon, auf WR gesetzt werden, Default ist MR (DIRMAINT MDISK, vgl. 3.4.2).

Arbeiten mit Files auf Minidisks: Das CMS Command FILELIST informiert (ohne Operanden) über alle Files der Minidisk A (dies ist per Default die eigene permanente Minidisk).

filelist <ENTER>

In der Liste kann mit <PF8> und <PF7> vorwärts bzw. rückwärts geblättert werden. Mit <PF3> wird FILELIST beendet.

In jede Zeile der Liste kann - beginnend in der Spalte Cmd - ein Command zur Bearbeitung des entsprechenden Files eingetragen werden (Spalten Filename, Filetype, ... ggf. überschreiben); durch <ENTER> werden diese Commands ausgeführt. Folgende Commands können verwendet werden:

```
XEDIT           : edieren
PRINT           : drucken (vgl. 4.2.2)
RENAME / fn ft fm : umbenennen in fn ft fm
COPYFILE / fn ft fm : kopieren nach fn ft fm
COMPARE / fn ft fm : vergleichen mit fn ft fm
DISCARD         : löschen
```

Groß/Kleinschreibung: Kleine Buchstaben werden wie große interpretiert.

Zur File-Bearbeitung vgl. 5.4 (FILEDEF).

Folgende Besonderheiten sind im **File Handling** des CMS zu beachten:

- **Kreation:** Files werden i.a. implizit durch einen Schreibzugriff kreiert.
- **Lebensdauer:** hängt von der Lebensdauer der Minidisk ab (s.o.)
- **Zugriffsschutz:** erfolgt über die Paßwörter der Minidisk beim Link setzen (s.o.)
- **Zugriffskoordination:** erfolgt über die Zugriffsart beim Link setzen auf die Minidisk.
- **File-Organisation:** CMS unterscheidet Files mit Sätzen fester (F) Länge und Files mit Sätzen variabler (V) Länge. Files mit Sätzen fester Länge können von FORTRAN Programmen sequentiell (ACCESS='SEQUENTIAL' in der OPEN Anweisung) und direkt (ACCESS='DIRECT') bearbeitet werden. Files mit Sätzen variabler Länge können nur sequentiell bearbeitet werden. Der Editor XEDIT kreiert i.a. Files mit Sätzen fester Länge.

4.2 Virtuelle Ein/Ausgabegeräte (Spooling)

Jeder Session stehen neben den Minidisks folgende **virtuellen Geräte** für die Ein/Ausgabe von Files zur Verfügung:

- Console(Ein/Ausgabe)
- Printer(Ausgabe)
- Punch (Ausgabe)
- Reader (Eingabe)

Console wird im wesentlichen durch das Terminal realisiert; für Printer, Punch und Reader verwaltet CP **Queues** mit **Spool Files** auf systemeigenen Minidisks, vgl. Abb. 4-1. Eine **Printer Queue** oder **Punch Queue** wird durch eine **Spool Class** (A-Z, 0-9) bezeichnet; sie enthält i.a. Spool Files, die verschiedenen Benutzern gehören. Jeder Benutzer (Userid) hat eine eigene **Reader Queue**. Jeder Spool File wird durch seinen Eigentümer (Userid) und eine **Spoolid** (4 Ziffern) identifiziert; er kann zusätzlich einen Filename und einen Filetype haben.

4.2.1 Virtuelles Gerät Console

Console Ausgabe: Zeilenorientierte Ausgaben können zusätzlich in einen Console Spool File geschrieben werden. Das CP Command SPOOL CONSOLE steuert dieses Protokollieren der Session.

***spool console start* <ENTER>**

startet das Protokollieren; es wird ein Console Spool File kreiert, der beim Schließen (CLOSE s.u.) in eine von CP verwaltete Queue eingereiht wird; Default ist die Printer Queue mit Spool Class T; die Spool Files dieser Queue werden vom Operator gelöscht. Vor dem Schließen muß daher festgelegt werden, daß Console Spool Files in eine andere Queue einzureihen sind, z.B. in die Printer Queue mit Spool Class A (sie werden dann auf dem Schnelldrucker SHRZ1 im Rechnerraum gedruckt):

***spool console class a* <ENTER>**

oder in die eigene Reader Queue (sie können dann z.B. mit RECEIVE auf Minidisk kopiert werden, vgl. 4.2.4):

***spool console to ** <ENTER>**

oder in die Printer Queue eines remote Druckers, z.B. MHRZ3 (vgl. 4.2.2):

***spool console to rscs* <ENTER>**

***tag dev console mhrz3* <ENTER>**

Ein Console Spool File wird (explizit) geschlossen mit

***spool console close* <ENTER>**

Beim Schließen wird - falls das Protokollieren nicht gestoppt wurde - ein neuer Console Spool File kreiert, in dem das Protokollieren fortgesetzt wird. (Beim Logoff werden alle nicht explizit geschlossenen Spool Files implizit geschlossen).

***spool console stop* <ENTER>**

stoppt das Protokollieren (ohne den Console Spool File zu schließen).

Console Eingabe: CMS verwaltet einen Console Stack, in dem Eingabezeilen für das virtuelle Geräte Console gepuffert werden können. Er besteht aus dem Terminal Eingabepuffer und dem Program Stack; der Program Stack wird vor dem Terminal Eingabepuffer abgearbeitet. Die virtuelle Maschine liest i.a. nur dann vom Terminal, wenn der Console Stack leer ist. Im Terminal Eingabepuffer werden Terminal Eingaben zwischengespeichert, die nicht sofort bearbeitet werden können (z.B. Eingabe mehrerer Commands mit # getrennt, Eingabe während Ausführung eines Commands, vgl. 3.1.3). Im Program Stack können mit Hilfe des CMS Commands EXECIO Eingabezeilen gepuffert werden, z.B.

`execio * diskr tue date1 <ENTER>`

kopiert alle (*) Zeilen des Minidisk Files (DISKR) tue date1 A in den Program Stack. Die Zeilen werden wie Eingabezeilen von Terminal Eingaben abgearbeitet (SPERRY: ADD, DEC: DEFINE SYS\$INPUT). Innerhalb von REXX Prozeduren können Eingabezeilen mit Hilfe der Instruction QUEUE im Program Stack gepuffert werden, vgl. BATCH-BETRIEB und KOMMANDOSPRACHEN.

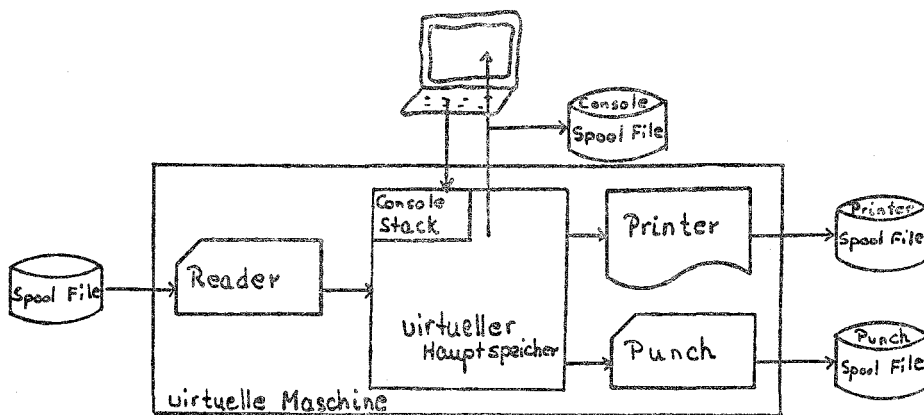


Abb. 4-1: Virtuelle Geräte

4.2.2 Virtuelles Gerät Printer

Das virtuelle Gerät **Printer** wird in der Regel zum Drucken von File-Inhalten verwendet (Programme können aber auch direkt auf Printer ausgeben). Gedruckt werden kann der Inhalt von **Text Files** mit höchstens 132 Zeichen pro Zeile (LRECL ≤ 132). **Text Files** sind Files, die mit dem Editor bearbeitbar sind; neben den vom Editor erzeugten Files sind dies z.B. von CMS Commands erzeugte Files mit Filetype **MAP** oder **LISTING** oder von FORTRAN Programmen mit sequentiellen formatierten **WRITES** erzeugte Files.

Das Drucken erfolgt mit dem **CMS Command PRINT**. **PRINT** erzeugt einen **Printer Spool File** und reiht ihn in eine von CP verwaltete Queue ein, z.B.

print eingabe daten <ENTER>

kopiert den Inhalt von File **eingabe daten A** in einen **Printer Spool File** (**eingabe daten A** bleibt unverändert). Files mit FORTRAN Vorschubzeichen in Position 1 jeder Zeile sind mit **CC Option** zu drucken (Default beim Filetype **LISTING**; **LRECL 133** ist zulässig), z.B.

print druck ausgabe (cc) <ENTER>

kopiert den Inhalt von File **druck ausgabe A** in einen **Printer Spool File**, wobei das erste Zeichen jeder Zeile in die entsprechende Vorschubinformatung umgewandelt wird.

Printer Spool Files werden per Default in die **Printer Queue** mit **Spool Class A** eingereiht. Die **Spool Files** dieser Queue werden von CP am **Schnelldrucker SHRZ1** im Rechnerraum ausgegeben. Alle anderen System-Drucker der IBM 4381 werden als **remote Drucker** betrieben; zum Drucken auf remote Druckern muß der Benutzer vor dem **PRINT** mit den **CP Commands SPOOL PRINTER** und **TAG DEV PRINTER** festlegen, wo gedruckt werden soll, z.B. auf dem **Matrixdrucker** mit dem Namen **MHRZ3**:

spool printer to rscs <ENTER>

tag dev printer mhrz3 <ENTER>

Die remote Drucker werden von der virtuellen Maschine **RSCS** bedient; der **Spool File** wird von CP in die **Reader Queue** von **RSCS** eingereiht; das **CP Command TAG DEV PRINTER** legt den Namen des remote Druckers fest, vgl. Notiz **DRUCKEN**. Die Prozedur **INFO** listet Name und Standort aller System-Drucker am Terminal auf:

info drucker <ENTER>

Das Zurücksetzen auf die Default Print Queue erfolgt mit

spool printer to system <ENTER>

tag dev printer <ENTER>

4.2.3 Virtuelles Gerät Punch

Das virtuelle Gerät **Punch** wird zum **Filetransfer** zwischen virtuellen Maschinen verwendet. Das **CMS Command SENDFILE** sendet einen **Punch Spool File** in die Reader Queue der angegebenen Userid, z.B.

sendfile prima programm to lstein <ENTER>

sendet eine Kopie des Files *prima programm A* in die Reader Queue von *lstein*. Mit **SENDFILE** wird auch der Filetransfer im Rechnernetz **EARN** (vgl. gleichnamige Notiz) realisiert, z.B.

sendfile quelle fortran to earnie at d00demo <ENTER>

sendet eine Kopie des Files *quelle fortran A* an den Empfänger *earnie* (Userid) am **EARN-Knoten d00demo** (Nodeid).

4.2.4 Virtuelles Gerät Reader

CP verwaltet für jede Userid eine Reader Queue, in die z.B. über **EARN** angekommene Files eingereicht werden. Die Spool Files der eigenen Reader Queue können mit Hilfe des virtuellen Geräts **Reader** "eingelese" werden, z.B.

receive next brief post a <ENTER>

kopiert den ersten Spool File der Reader Queue in den File *brief post a* und löscht den Spool File in der Reader Queue (Programme können auch direkt vom Reader lesen.) Das **CMS Command RDRLIST**, z.B.

rdrlist <ENTER>

listet Filenames und Filetypes aller Spool Files der Reader Queue auf; none none bedeutet, daß der Spool File keinen Filenamen und Filetype hat. In dieser Liste kann mit **<PF8>** und **<PF7>** vorwärts und rückwärts geblättert werden; mit **<PF3>** wird **RDRLIST** beendet.

In jede Zeile der Liste kann - beginnend in der Spalte **Cmd** - ein Command zur Bearbeitung des entsprechenden Files eingetragen werden (Spalten **Filename**, **Filetype**, ... ggf. überschreiben); **<ENTER>** führt die Commands aus. Folgende Commands können verwendet werden:

DISCARD : löschen
PEEK : am Terminal ausgeben
RECEIVE / fn ft fm : in File *fn ft fm* kopieren

Weitere Informationen: **CMS Primer, Chapter 5**

5. Programmentwicklung in FORTRAN

Zur Programmentwicklung stehen mehrere höhere Programmiersprachen zur Verfügung (z.B. FORTRAN, PASCAL, COBOL, PL/I, C). Dieser Abschnitt stellt exemplarisch die Programmumgebung für die Programmiersprache FORTRAN 77 vor; er ist lediglich als Einstieg konzipiert; weitergehende Informationen finden Sie in der Notiz FORTRAN. Zu den anderen Programmiersprachen gibt es entsprechende Notizen. Erläuterungen zur Abb. 5-1 finden Sie in den nachfolgenden Abschnitten.

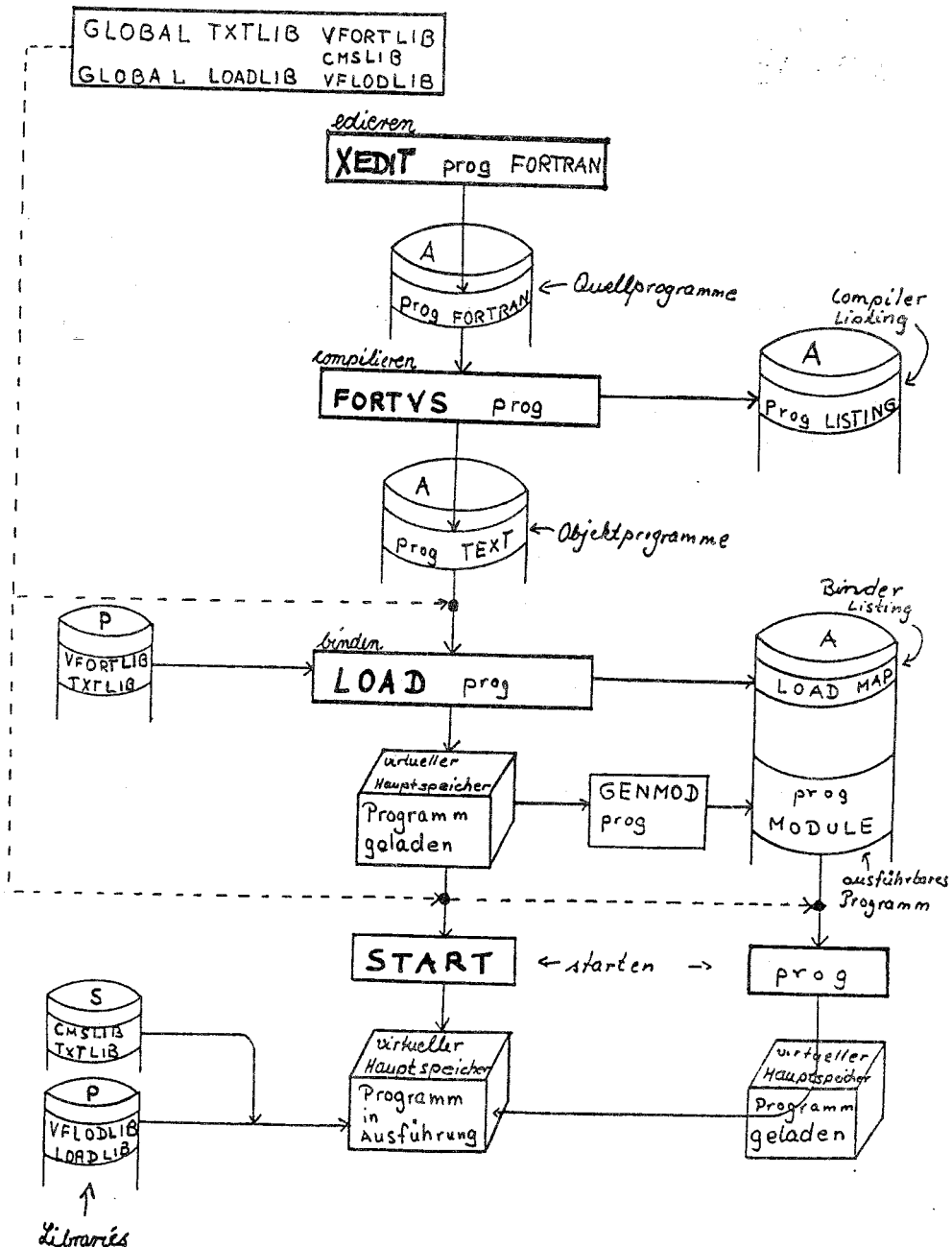


Abb. 5-1: Programmentwicklung in FORTRAN

5.1 Edieren (Quellprogramme)

Zum Edieren steht der System Product Editor, ein komfortabler **bildschirmorientierter Editor**, zur Verfügung. Zum Einarbeiten sollten Sie Chapter 2 im CMS Primer durcharbeiten. Wer viel Erfahrung mit Editoren hat, kommt eventuell auch mit Chapter 2 des Summary of Commands aus.

Beispiel: Edieren eines FORTRAN Programms (Filetype muß FORTRAN sein).

```
xedit prog fortran <ENTER>
```

Das CMS Command XEDIT ruft den System Product Editor auf. Ediert wird in der Workarea des Editors; die Workarea liegt im virtuellen Hauptspeicher (= Hauptspeicher der virtuellen Maschine).

```
---> input <ENTER>  
    ... Quellprogramme eingeben ...  
    ... (2 mal <ENTER> beendet INPUT)...  
---> file <ENTER>
```

Mit dem XEDIT Subcommand FILE wird das Edieren beendet; die edierten **Quellprogramme** werden in den File prog FORTRAN A kopiert, vgl. 6.

Groß/Kleinschreibung: Per Default werden kleine Buchstaben in große umgewandelt. Im FORTRAN Programm sind kleine Buchstaben nur in Kommentaren und in CHARACTER Konstanten erlaubt. Das XEDIT Subcommand SET CASE MIXED ermöglicht die Eingabe kleiner Buchstaben.

5.2 Compilieren (Quellprogramme)

Problem: Compiliert werden sollen die Quellprogramme (Hauptprogramm, ggf. mit Unterprogrammen) im File

```
prog FORTRAN A
```

Lösung: Das CMS Command FORTVS ruft den FORTRAN Compiler auf.

```
fortvs prog <ENTER>
```

compiliert die **Quellprogramme** im File prog FORTRAN A und schreibt die erzeugten **Objektprogramme** in den File prog TEXT A und das **Compiler Listing** in den File prog LISTING A. Das Compiler Listing kann mit dem Editor gelesen (vgl. 5.1) oder mit PRINT gedruckt werden (vgl. 4.2.2).

5.3 Binden (Objektprogramme)

Problem: Gebunden werden sollen die Objektprogramme (Hauptprogramm, ggf. mit Unterprogrammen) im File

```
prog TEXT A
```

Das erzeugte ausführbare Programm soll geschrieben werden in den File

```
prog MODULE A
```

Lösung: Die CMS Commands GLOBAL TXTLIB und GLOBAL LOADLIB stellen die Libraries bereit, die beim Binden und zur Laufzeit benötigt werden:

```
global txtlib vfortlib cmslib <ENTER>  
global loadlib vflodlib <ENTER>
```

Die angegebenen Libraries werden immer benötigt; es können weitere Libraries angegeben werden, z.B. die Libraries NAG1DLIB und NAG2DLIB, falls Unterprogramme der NAG Library aufgerufen werden:

```
GLOBAL TXTLIB NAG1DLIB NAG2DLIB VFORTLIB CMSLIB
```

Ein GLOBAL TXTLIB bzw. GLOBAL LOADLIB bleibt bis zum nächsten GLOBAL TXTLIB ... bzw. GLOBAL LOADLIB ... in Kraft; es wird daher empfohlen, die GLOBAL Commands ggf. mit den zugehörigen LINK und ACCESS Commands in die User Logon Prozedur einzutragen (z.B. mit XEDIT PROFILE EXEC). Die CMS Commands QUERY TXTLIB bzw. QUERY LOADLIB listen die GLOBAL TXTLIBs bzw. GLOBAL LOADLIBs am Terminal auf.

Das CMS Command LOAD ruft den Binder auf:

```
load prog <ENTER>
```

bindet die Objektprogramme aus prog TEXT A mit Objektprogrammen aus den angegebenen Libraries (s.o.) zu einem ausführbaren Programm zusammen. Das ausführbare Programm steht im virtuellen Hauptspeicher. Das Binder Listing wird in den File LOAD MAP A geschrieben.

Das CMS Command GENMOD speichert das ausführbare Programm ab:

```
genmod prog <ENTER>
```

schreibt das ausführbare Programm vom virtuellen Hauptspeicher in den File prog MODULE A.

5.4 File-Bearbeitung

Es wird empfohlen, jeden File im FORTRAN Programm explizit zu öffnen und der Unit Number einen Data Definition Name zuzuordnen, z.B.

```
OPEN(UNIT=17,FILE='DDNAME',...)
```

Dem Data Definition Name (im Beispiel DDNAME) ist vor dem Start des Programms mit dem CMS Command FILEDEF der File zuzuordnen:

```
FILEDEF ddname DISK fn ft fm [(options)]
```

Beispiele: File *data seq a* mit Data Definition Name *ddname1* soll sequentiell gelesen werden (ACCESS='SEQUENTIAL'):

```
filedef ddname1 disk data seq a <ENTER>
```

File *neu seq a* mit Data Definition Name *ddname2* soll als sequentieller formatierter File kreiert werden (ACCESS='SEQUENTIAL', FORM='FORMATTED', STATUS='NEW'); die Satzlänge LRECL ist in Zeichen anzugeben (im Beispiel 96; Default ist 80):

```
filedef ddname2 disk neu seq a (recfm f lrecl 96 blksize 96)  
<ENTER>
```

File *data dir a* mit Data Definition Name *ddname3* soll direkt bearbeitet werden (ACCESS='DIRECT'); die Anzahl der Sätze XTENT ist anzugeben (im Beispiel 4711; Default ist 50):

```
filedef ddname3 disk data dir a (xtent 4711) <ENTER>
```

OPEN Anweisung und CMS Command FILEDEF sind nicht erforderlich für

```
UNIT=5 : Terminal Eingabe (LRECL 80)  
UNIT=6 : Terminal Ausgabe (LRECL 133)  
UNIT=7 : Punch Ausgabe (LRECL 80)
```

5.5 Starten (ausführbares Programm)

Auch zur Laufzeit müssen die beiden in 5.3 angegebenen GLOBALs in Kraft sein. Nach LOAD (und ggf. GENMOD und FILEDEFS) kann das ausführbare Programm im virtuellen Hauptspeicher gestartet werden mit dem CMS Command START:

```
start <ENTER>
```

Das ausführbare Programm im File *prog* MODULE A wird gestartet mit

```
prog <ENTER>
```

6. Beispiel für eine Session

Das Beispiel setzt voraus, daß die Session an einem MRX 2078 geführt wird; nach dem Einschalten erscheint das VM/SP Logo (vgl. Abb. 3-1), es kann eine Session begonnen werden. Um unter den Userid's 1STEIN, 2STEIN, ... eine freie zu finden, ist zuerst eine Session mit der Userid STEIN zu beginnen (sie wird implizit beendet):

```
USERID   ---> stein <—>|>  
PASSWORD ---> keines <ENTER>
```

```
...  
FREIE USERIDS:  
2STEIN  
3STEIN  
7STEIN  
...  
LOGOFF
```

Falls statt der freien Userid's die Meldung DMKLOG054E Already logged on ... erscheint, versuchen Sie es bitte nach kurzer Wartezeit erneut. Falls

KEINE FREIEN USERID'S!

erscheint, sollten Sie eine Kaffee-Pause einlegen oder die Zeit nutzen, um eine eigene Userid zu beantragen. Falls z.B. Userid 2STEIN frei ist, kann mit

```
<ENTER>  
... VM/SP Logo ...
```

```
USERID   ---> 2stein <—>|>  
PASSWORD ---> keines <ENTER>
```

eine Session begonnen werden. Bei der Eingabe des Logon Paßworts werden die eingegebenen Zeichen nicht auf dem Bildschirm angezeigt. Nach

```
<ENTER>
```

wird die System Logon Prozedur und danach ggf. die User Logon Prozedur von 2STEIN ausgeführt; im Status MORE ... fordert

```
<PA2>
```

weitere Ausgaben an; der Status geht über in RUNNING; es können CMS oder CP Commands eingegeben werden.

```
spool console class a start <ENTER>
```

startet das Protokollieren der Session und legt fest, daß das Protokoll am Schnelldrucker SHRZ1 im Rechnerraum (CLASS A) gedruckt werden soll.

xedit wurzel fortran <ENTER>

ruft den Editor auf; das edierte Quellprogramm soll in File *wurzel FORTRAN A* geschrieben werden. Falls der File nicht existiert, steht in der zweiten Bildschirmzeile

CREATING NEW FILE:

Hinter *----*> können XEDIT Subcommands eingegeben werden, z.B. *INPUT* zum Einschalten des *INPUT-MODEs*:

----> *input* <ENTER>

Zwischen der jetzigen Cursor Position und der Zeile

* * * INPUT ZONE * * *

können Zeilen eingetragen werden:

```
10      program wurzel      <—>|>
      read (5,*) r          <—>|>
      if (r.lt.0.) stop     <—>|>
      write (6,*) ' sqrt(' ,r,') = ',sqrt(r) <—>|>
      goto 10               <—>|>
      end                   <—>|>
```

<ENTER>

<ENTER>

2 mal <ENTER> schaltet den *INPUT-MODE* aus. Um hinter der ersten Zeile eine Zeile einzufügen, wird die Zeile

----- PROGRAM WURZEL

geändert zu

---a1 PROGRAM WURZEL <ENTER>

(*a1*: add 1 line). In die entstandene freie Zeile wird eingetragen

----- 10 write (6,*) 'bitte real-zahl eingeben:' <—>|>

und in der nächsten Zeile die Anweisungsnummer 10 gelöscht (Leertaste):

----- READ (5,*) R <ENTER>

--- *file* <ENTER>

kreiert File *wurzel* FORTRAN A und schreibt das eben erstellte Quellprogramm hinein.

fortvs *wurzel* <ENTER>

compiliert das Quellprogramm.

global *txtlib vfortlib cmslib* <ENTER>

global *loadlib vflodlib* <ENTER>

stellt die benötigten Libraries bereit.

load *wurzel* <ENTER>

bindet das vom Compiler erzeugte Objektprogramm mit Objektprogrammen aus der FORTRAN Library und der CMS Library zu einem ausführbaren Programm; es steht im Hauptspeicher; es kann entweder mit

start <ENTER>

gestartet werden oder mit

genmod *wurzel* <ENTER>

in File *wurzel* MODULE A geschrieben werden und dann gestartet werden mit

wurzel <ENTER>

BITTE REAL-ZAHL EINGEBEN:

?

17. <ENTER>

SQRT(17.0000000) = 4.12310600

BITTE REAL-ZAHL EINGEBEN:

?

-10. <ENTER>

Ready; T= ...

Beendet wird die Session mit

logoff <ENTER>

Das erzeugte Protokoll wird auf dem Schnelldrucker SHRZ1 im Rechnerraum gedruckt.