

robotron

Systemhandbuch

SCP 3.0

Anleitung für den Bediener

robotron

Systemhandbuch SCP 3.0

Anleitung für den Bediener

Stand: 21. April 1987

VEB Robotron Büromaschinenwerk

„Ernst Thälmann“ Sömmerda

Inhaltsverzeichnis

	Seite
1. Einführung in das SCP 3.0	4
1.1 Starten des SCP 3.0	4
1.2. Kommandozeile	4
1.3. Die Notwendigkeit des Dateikopierens	6
1.4. Wie Disketten kopiert werden	6
2. Dateien, Disketten und Laufwerke	7
2.1. Was ist eine Datei?	7
2.2. Erstellen einer Datei	8
2.3. Bezeichnen von Dateien	8
2.4. Die Auswahl des richtigen Laufwerkes	9
2.5. Nutzerbereich (User)	10
2.6. Zugriff auf mehrere Dateien	11
2.7. Dateischutz	11
2.7.1. Dateiattribute	12
2.7.2. Datum- und Zeiteintragung	12
2.7.3. Paßwörter	13
2.8. Speichern von Dateien auf Diskette	13
2.9. Floppy Disk wechseln	13
2.10 Schutz eines Laufwerkes	14
2.11. RAM-Disk	14
2.12. Automatische Formaterkennung	15
3. Konsole und Drucker	16
3.1 Steuern der Konsolenausgabe	16
3.2. Steuern der Druckerausgabe	16
3.3. Zeileneditierung auf der Konsole	16
3.4. Umlenken von Ein- und Ausgaben	18
3.5. Zuordnung logischer Einheiten	19
4. SCP 3.0 Kommandokonzept	20
4.1 Zwei Arten von Kommandos	20
4.2. Interne Kommandos	20
4.3. Transiente Kommandos	21
4.4. Suchen von Programm- und Datendateien durch das SCP 3.0	22
4.4.1. Finden einer Datendatei	22
4.4.2. Finden einer Programmdatei	23
4.5. Ausführen von Mehrfachkommandos	24
4.6. Unterbrechen des Programmlaufes	25
4.7. Das HELP Programm	26
5. Zusammenstellung der Kommandos	26
5.1 Struktur des SCP-3.0-Kommandos	27
5.2. Darstellung der Kommandos	29

	Seite
6. Kommandos des SCP 3.0	31
COPYSYS	31
DATE	33
DEVICE	35
DIR	37
DUMP	42
ERASE	42
GENCOM	43
GET	46
HELP	47
INIT	50
INITDIR	51
LINK	52
MODCS	54
MODFD	54
MODIO	56
PIP	60
PUT	69
RENAME	71
SAVE	72
SET	73
SETDEF	78
SHOW	80
SUBMIT	83
TYPE	85
USER	86
Anhang Meldungen des SCP 3.0	88

1. Einführung in das SCP 3.0

In diesem Abschnitt wird beschrieben, wie Kommandozeilen einzugeben sind und wie Duplikate der Vertriebsdiskette angefertigt werden.

1.1. Starten des SCP 3.0

Starten oder Laden des SCP 3.0 bedeutet das Lesen einer Kopie des Betriebssystems von der SCP-3.0-Systemdiskette in den internen Speicher des Computers.

Zunächst muß überprüft werden, ob der Computer eingeschaltet wurde. Danach ist die SCP-3.0-Systemdiskette, z. B. in das Laufwerk A, einzulegen und das Laufwerk zu schließen. Automatisch wird nun das SCP 3.0 in den Speicher geladen. Dieser Prozeß wird als Kaltstart oder Laden des Systems bezeichnet.

Nachdem das System in den Speicher geladen wurde, wird auf dem Bildschirm eine Meldung ausgegeben, etwa in folgender Art:

```
SCP 3.0   V XXXX   TT/MM/JJ
```

Die mit XXXX bezeichnete Versionsnummer des Systems gibt Auskunft über den Stand des verwendeten SCP 3.0. Die Angabe TT/MM/JJ gibt Auskunft über das Erstellungsdatum der verwendeten SCP-3.0-Version. Nach dieser Anzeige wird das Bereitschaftszeichen des Systems ausgegeben:

```
A>
```

Mit dem Bereitschaftszeichen wird zum Ausdruck gebracht, daß das System bereit ist, Kommandos über die Tastatur zu empfangen. Das Bereitschaftszeichen bringt gleichzeitig noch zum Ausdruck, daß das Laufwerk A das Standardlaufwerk ist. Das bedeutet, daß vom System ohne zusätzliche Anweisungen Programm- und Daten-Dateien auf der Diskette im Laufwerk A gesucht werden. Weiterhin wird mit dem Bereitschaftszeichen angezeigt, daß der Nutzerbereich 0 zugewiesen ist, denn es wurde keine andere Nummer eines Nutzerbereiches angegeben.

Es ist auch möglich, von jedem anderen Laufwerk das SCP 3.0 zu starten. Das benutzte Laufwerk wird automatisch zum Standardlaufwerk.

1.2. Die Kommandozeile

SCP 3.0 steuert die Aktivitäten des Computers in Abhängigkeit von den über die Tastatur eingegebenen Kommandos. Eine SCP-3.0-Kommandozeile setzt sich zusammen aus einem Kommandoschlüsselwort und wahlweise anzugebenden Kommandozusätzen und wird mit einem Wagenrücklauf <cr> abgeschlossen. Das Kommandoschlüsselwort bezeichnet entweder ein residentes SCP Kommando (z. B. DIR) oder ein ausführbares Programm auf der Diskette. Der Zusatz beinhaltet zusätzliche Informationen zum Kommando wie Dateiname oder Parameter. Die Kommandozeile wird mit der Taste <ET> abgeschlossen. Nachfolgend ein Beispiel für eine Kommandozeile:

```
A>DIR MYFILE
```

Die vom Nutzer eingegebenen Zeichen werden im Fettdruck angegeben, um sie von den durch das System ausgegebenen Zeichen zu unterscheiden. Im oben angegebenen Beispiel ist DIR das Kommandoschlüsselwort und MYFILE der Zusatz. Die Taste <cr> erscheint nicht auf dem Bildschirm oder im Beispiel. Um die Kommandozeile an das System zu übergeben, muß aber immer die Taste <ET> (im Text auch mit <cr> bezeichnet) betätigt werden. In diesem Handbuch bezeichnen <RETURN> oder <ET> immer den Wagenrücklauf <cr>.

Wird ein Zeichen auf der Tastatur eingegeben, dann erscheint dieses Zeichen auch gleichzeitig auf dem Bildschirm. Der Cursor bewegt sich dabei um eine Stelle nach rechts. Ein Eingabefehler kann mit Hilfe der Tasten , CTRL-H, CTRL-G bzw. der Kursortasten korrigiert werden. CTRL ist dabei die Abkürzung für die Control-Taste. Um ein Controlzeichen einzugeben, muß die CTRL-Taste gedrückt gehalten und gleichzeitig die entsprechende andere Taste betätigt werden. Eine gebräuchliche Darstellung der Control-Taste auf dem Bildschirm ist ^ z. B. ^Z für CTRL-Z.

Alle Eingaben in der Kommandozeile können in einer beliebigen Kombination von Groß- und Kleinbuchstaben erfolgen. SCP interpretiert alle Buchstaben in der Kommandozeile als großgeschrieben.

Die Eingaben in der Kommandozeile erfolgen im allgemeinen unmittelbar hinter dem Bereitschaftszeichen. Das System verkraftet aber auch Leerzeichen zwischen dem Bereitschaftszeichen und dem Schlüsselwort.

SCP 3.0 unterscheidet zwei Kommandoarten: interne oder residente (built-in commands) und externe bzw. transiente Kommandos. Die residenten Kommandos veranlassen die Ausführung von Programmen, die im Speicher als Teil des SCP 3.0 vorhanden sind. Diese Kommandos können jederzeit ausgeführt werden, denn es sind keine Diskettendateien erforderlich. Die transienten Kommandos sind als Programmdateien auf Disketten gespeichert. Bevor sie ausgeführt werden können, müssen sie von der Diskette geladen werden. Die Programmdateien sind durch den Dateityp COM gekennzeichnet, so daß sie bei der Anzeige des Diskettenverzeichnisses sofort erkannt werden können.

Bei transienten Kommandos werden nur die Schlüsselwörter überprüft. Die Kommandozusätze werden durch SCP 3.0 nicht überprüft, sondern direkt für die Programme in einem Puffer abgelegt. Ein Kommandozusatz darf nicht mehr als 128 Zeichen enthalten.

Anhand des DIR-Kommandos soll einmal demonstriert werden, wie SCP 3.0 eine Kommandozeile interpretiert. Das DIR-Kommando, eine Abkürzung für ‚directory‘ (Verzeichnis), bewirkt die Anzeige eines Verzeichnisses von Diskettendateien auf dem Bildschirm. Das DIR-Kommando wird direkt hinter dem Bereitschaftszeichen ohne Zusatz eingegeben und mit <ET> abgeschlossen.

A>DIR

SCP 3.0 zeigt daraufhin die Namen aller auf der Diskette im Laufwerk A gespeicherten Dateien an. Liegt zum Beispiel die SCP-3.0-Systemdiskette im Laufwerk A, dann werden unter vielen anderen folgende Dateinamen angezeigt:

COPYSYS	COM	PIP	COM
SET	COM	HELP	COM

SCP 3.0 akzeptiert nur korrekt eingegebene Kommandoschlüsselwörter. Wird bei der Eingabe ein Fehler gemacht und dieser Fehler vor Betätigen von <ET> nicht korrigiert, dann gibt SCP 3.0 die Kommandozeile noch einmal aus und setzt ein Fragezeichen dahinter. Wird zum Beispiel das DIR Kommando verkehrt eingegeben, dann reagiert SCP 3.0 wie folgt:

A>DJR
DJR?

Damit wird dem Nutzer mitgeteilt, daß dieses Schlüsselwort nicht gefunden wurde. Ein Eingabefehler kann mit Hilfe der Tasten , CTRL-H, CTRL-G bzw. der Kursortasten

korrigiert werden. SCP 3.0 stellt noch eine Reihe weiterer Controlzeichen zur Verfügung, die ein effektives Editieren von Kommandozeilen ermöglichen. Eine Übersicht über die Controlzeichen ist dem Anhang zu entnehmen.

Dem Kommando DIR kann als Zusatz ein Dateiname angefügt werden. Mit dieser Form des DIR-Kommandos kann überprüft werden, ob sich eine bestimmte Datei auf der Diskette befindet. Um festzustellen, ob sich das transiente Programm COPYSYS.COM auf der Diskette befindet, ist einzugeben:

```
A>DIR COPYSYS.COM
```

SCP 3.0 führt die Aufgabe so aus, daß entweder der angegebene Dateiname oder die Meldung „No File“ angezeigt wird.

Zwischen dem Schlüsselwort und dem Zusatz muß mindestens ein Leerzeichen stehen. Wird das nicht beachtet, dann reagiert SCP 3.0 wie folgt:

```
A>DIRCOPYSYS.COM
DIRCOPYSYS.COM?
```

1.3. Die Notwendigkeit des Dateikopierens

„Humans have faults, and so do computers“, d. h., außer dem Menschen kann auch ein Computer einmal Fehler machen und Programme oder andere Dateien zerstören. Wird z. B. ein Kommando fehlerhaft eingegeben, dann könnte damit eine gerade erstellte oder innerhalb von mehreren Monaten erzeugte Datei gelöscht werden. Eine ähnliche Katastrophe kann durch den Ausfall einer elektronischen Komponente entstehen.

In der Datenverarbeitung schützt man sich vor einem solchen Verlust von Programmen oder anderen Dateien, indem man von den wichtigsten Dateien Kopien anfertigt. Es empfiehlt sich immer, von einem neu erworbenen Programm eine Arbeitskopie anzufertigen und das Original sicher zu verwahren. Wird die Arbeitskopie des Programmes versehentlich einmal gelöscht, dann kann leicht vom Original eine neue Kopie angefertigt werden.

Es empfiehlt sich ferner, von Dateien, die gerade erstellt werden, in verschiedenen Entwicklungsstadien Kopien anzufertigen, um den jeweiligen Arbeitsstand zu sichern. Die Häufigkeit, mit der Kopien angefertigt werden, ist bei jedem Programmierer anders.

Zunächst soll erläutert werden, wie die Kopie einer Diskette angefertigt werden kann.

1.4. Wie Disketten kopiert werden

Um Disketten zu kopieren, sind mindestens zwei Floppy-Disk-Laufwerke erforderlich. Die Zieldisketten müssen nicht neu sein. Sie müssen möglicherweise mit dem entsprechenden Programm neu initialisiert werden. Sind die Zieldisketten gebrauchte Disketten, dann muß man sich vorher davon überzeugen, daß auf diesen Disketten keine anderen Dateien mehr vorhanden sind.

Mit den Programmen COPYSYS und PIP kann eine Diskette kopiert werden. Mit PIP werden alle Programm- und Daten-Dateien kopiert und mit COPYSYS das Betriebssystem.

Im weiteren wird erläutert, wie ein Duplikat einer Vertriebsdiskette unter Verwendung von zwei Laufwerken hergestellt wird. Die Laufwerke werden mit A und B bezeichnet. Zunächst wird mit COPYSYS der Betriebssystemlader kopiert. Liegt die Vertriebsdiskette im Lauf-

werk A und die leere Diskette im Laufwerk B, dann ist folgendes Kommando nach dem Bereitschaftszeichen einzugeben:

```
A>COPYSYS A: B: [SYS]
```

SCP 3.0 lädt das Programm COPYSYS in den Speicher und startet es. COPYSYS gibt folgendes auf den Bildschirm aus:

```
SCP3  COPYSYS  R-BWS V 1/0
Reading system from disk!
Drive types are:
Drive A: MFS 1.6
Drive B: MFS 1.6
Drive C: MFS 1.6
Drive D: MFS 1.6
Writing system to disk!
End of copy system tracks.
Writing SCP3.SYS and CCP.COM to disk!
Function complete.
A>
```

Damit ist lediglich eine Kopie des Betriebssystems erzeugt. Sollen die Dateien kopiert werden, dann kann das mit folgendem PIP-Kommando veranlaßt werden:

```
A>PIP B: = A:*. *
```

Dieses PIP-Kommando kopiert alle im Verzeichnis vom Laufwerk A katalogisierten Dateien nach Laufwerk B. PIP gibt die Meldung COPYING aus, gefolgt vom Dateinamen der gerade übertragenen Datei. Wenn PIP das Kopieren abgeschlossen hat, dann wird das Bereitschaftszeichen des Systems ausgegeben.

Die Zieldiskette stellt dann eine exakte Kopie des Originals dar. Die Originaldiskette muß dem Laufwerk entnommen und sicher verwahrt werden. Bleibt das Original sicher aufbewahrt und unverändert, dann läßt sich leicht ein Programm kopieren, wenn etwas mit der Arbeitskopie passiert.

Die Kopie ist dem Laufwerk B zu entnehmen und in das Laufwerk A einzulegen. Diese Kopie wird ab jetzt als SCP-3.0-Systemdiskette verwendet.

2. Dateien, Disketten und Laufwerke

Die Verwaltung und der Zugriff auf Diskettendateien sind die wichtigsten Aufgaben des SCP 3.0. Mit dem SCP 3.0 können Diskettendateien erstellt, gelesen, geschrieben, kopiert und gelöscht werden. Im nachfolgenden Abschnitt wird erklärt, wie eine Datei zu erstellen und zu bezeichnen ist, wie auf eine Datei zugegriffen wird und wie eine Datei auf der Diskette abgespeichert wird. Ferner wird erklärt, wie Disketten und Laufwerke zu ändern sind.

2.1. Was ist eine Datei?

Eine Datei ist eine Menge zusammengehörender Informationen, die auf einer Diskette gespeichert wurden. Jede Datei muß einen eindeutigen Namen haben, denn SCP 3.0 greift über diesen Namen auf die Datei zu. Auf den Disketten wird ein Verzeichnis eingerichtet, in dem Namen und Platz der auf dieser Diskette gespeicherten Dateien notiert werden.

Es gibt zwei Arten von Dateien: die Programm- (oder Kommando-) Dateien und die Datendateien. In einer Programmdatei sind ausführbare Befehle abgespeichert, die vom Computer schrittweise abgearbeitet werden können. Eine Datendatei beinhaltet verschiedene Informationen, z. B. ein Namens- oder Adreßverzeichnis oder einen Text. Die Datendateien selbst können nicht ausgeführt werden, aber sie können von einer Programmdatei verarbeitet werden.

Eine Datendatei kann weiterhin auch den Quellcode für ein Programm beinhalten. Eine solche Quellprogrammdatei muß erst von einem Assembler oder Compiler verarbeitet werden, um daraus eine ausführbare Programmdatei zu erzeugen. In den meisten Fällen wird eine Datendatei von einem Programm verarbeitet. Es gibt aber auch Fälle, in denen ein Programm eine Programmdatei verarbeitet. Ein Beispiel dafür ist das Kopierprogramm PIP, mit dem Programmdateien kopiert werden können.

2.2. Erstellen einer Datei

Es gibt verschiedene Möglichkeiten eine Datei zu erstellen. Eine Möglichkeit ist z. B. die Anwendung eines Texteditors. Eine andere Möglichkeit ist das Kopieren einer bereits existierenden Datei auf einen anderen Platz, wobei die Datei gleichzeitig noch umbenannt werden kann. Das könnte z. B. mit dem PIP-Kommando erreicht werden. Schließlich gibt es auch Programme, die aus Ausgangsdateien neue Dateien erzeugen, wie das bei Assemblern oder Compilern der Fall ist.

2.3. Das Bezeichnen von Dateien

SCP 3.0 erkennt jede Datei anhand ihres eindeutigen Namens. Eine Dateibezeichnung besteht aus vier Teilen:

- der Laufwerksbezeichnung
- dem Dateinamen
- dem Dateityp
- dem Paßwort

Unbedingt erforderlich ist die Angabe des Dateinamens; alle anderen Bestandteile der Dateibezeichnung können, müssen aber nicht unbedingt angegeben werden.

Die Laufwerksbezeichnung ist ein einzelner Buchstabe (A-P), gefolgt von einem Doppelpunkt. Jedem Laufwerk ist ein Buchstabe zugeordnet. Wird in der Dateibezeichnung eine Laufwerksbezeichnung angegeben, dann sucht SCP 3.0 auf diesem Laufwerk nach der angegebenen Datei. Wird z. B. eingegeben:

B:MYFILE

dann wird auf dem Laufwerk B nach der Datei MYFILE gesucht.

Ein Dateiname kann bis zu acht Zeichen lang sein. Eine Dateibezeichnung kann im einfachsten Fall nur aus einem Dateinamen bestehen.

Der Dateiname sollte so gewählt werden, daß aus ihm Rückschlüsse auf den Inhalt der Datei möglich sind. Soll beispielsweise ein Kundenverzeichnis erstellt werden, dann kann als Name

KUNDEN

gewählt werden und man hat damit sofort einen Hinweis auf den Dateiinhalt.

Durch Anhängen eines Dateityps an den Dateinamen lassen sich Dateien mit ähnlicher Struktur und unterschiedlichen Dateinamen zusammenfassen. Diese Typbezeichnung kann bis zu drei Zeichen lang sein und wird vom eigentlichen Dateinamen mit einem Punkt "." abgetrennt. Auch bei der Wahl dieser Typbezeichnung sollte der Zusammenhang zur Art der Datei zum Ausdruck gebracht werden. Der Dateiname des Kundenverzeichnisses könnte wie folgt erweitert werden:

KUNDEN.NAM

Bei der Anzeige von Datei-Bezeichnungen durch ein SCP-Kommando werden kurze Dateinamen mit Leerzeichen auf acht Stellen aufgefüllt, so daß ein Vergleich der Dateitypen schnell erfolgen kann. Die Programmdateien, die SCP 3.0 in den Speicher lädt, haben unterschiedliche Dateinamen, aber alle den Dateityp COM.

Der Datei-Bezeichnung kann wahlweise ein Paßwort zugefügt werden. Das Paßwort kann bis zu acht Zeichen lang sein und wird bei der Eingabe mit einem Semikolon von den vorher eingegebenen Teilen der Datei-Bezeichnung abgetrennt.

Wird eine Datei mit einem Paßwort geschützt, dann muß für einen Zugriff auf diese Datei das Paßwort als Teil der Datei-Bezeichnung mit eingegeben werden.

Dateinamen, Dateitypen und Paßwörter sollten aus Buchstaben und Ziffern gebildet werden. Folgende Zeichen sind in Dateinamen, Dateitypen und Paßwörtern nicht erlaubt, denn sie besitzen für das SCP 3.0 eine spezielle Bedeutung:

< > = , ! | * ? & / \ [] () . : ; \ + -

Eine vollständige Datei-Bezeichnung, die alle möglichen Elemente enthält, besteht aus einer Laufwerksbezeichnung, einem Dateinamen, einem Dateityp und einem Paßwort, die alle durch ihre speziellen Trennzeichen voneinander separiert wurden.

Beispiel einer vollständigen Datei-Bezeichnung:

A>C:DOKUMENT.GES;OTTOKAR

2.4. Die Auswahl des richtigen Laufwerkes

Enthält ein Kommandozusatz eine Datei-Bezeichnung ohne Laufwerksangabe, dann sucht das Programm die Datei in dem durch das Bereitschaftszeichen angegebenen Laufwerk, dem sogenannten Standardlaufwerk. Wird z. B. folgendes Kommando eingegeben

A>DIR COPYSYS.COM

dann sucht DIR die Datei COPYSYS.COM im Verzeichnis der Diskette im Laufwerk A. Soll ein anderes Laufwerk, z. B. das Laufwerk B, angesprochen werden, dann muß das dem System mitgeteilt werden. Deshalb ermöglicht das System die Laufwerksangabe in der Datei-Bezeichnung. Das Kommando

A>DIR B:MYFILE.LIB

veranlaßt die Suche der Datei MYFILE.LIB im Verzeichnis der Diskette auf dem Laufwerk B. Bei der Kommandoeingabe muß immer beachtet werden, welche Datei im Standardlaufwerk liegt. Verschiedene Anwenderprogramme verlangen ihre Datendateien im Standardlaufwerk.

Auch einem Programmdateinamen kann eine Laufwerksangabe vorangestellt werden, selbst wenn der Programmdateiname als Kommandoschlüsselwort verwendet wird. Wird z. B. folgendes Kommando eingegeben

A>B:PIP

dann wird im Verzeichnis der Diskette im Laufwerk B nach der Datei PIP.COM gesucht. Findet SCP 3.0 die Datei PIP im Laufwerk B, dann wird PIP in den Speicher geladen und ausgeführt.

Soll auf mehrere Dateien im gleichen Laufwerk zugegriffen werden, dann ist es angebracht das Standardlaufwerk zu wechseln, um nicht immer wieder das Laufwerk angeben zu müssen. Für einen solchen Wechsel müssen nach dem Bereitschaftszeichen die Laufwerksangabe eingegeben und <ET> betätigt werden. SCP 3.0 verändert damit die Bereitschaftsanzeige auf das neue Standardlaufwerk.

A>B:

B>

Wird eine Diskette in ein anderes Laufwerk gelegt, dann ändert sich für eine Datei auf dieser Diskette die Laufwerksangabe.

2.5. Nutzerbereich (User)

Ein weiteres Erkennungsmerkmal der Dateien ist für das System SCP 3.0 die Zuordnung einer Nutzerbereichsnummer zu jeder Datei. Die Nutzernummer liegt im Bereich 0–15. Sie wird der Datei bei deren Erstellung zugewiesen. Mit den Nutzerbereichsnummern ist es möglich, Dateien in 16 verschiedene Bereiche einzuteilen.

Wird mit einem SCP-3.0-Programm eine Datei erstellt, dann wird ihr die Nummer des aktuellen Nutzerbereiches zugeordnet, außer wenn mit PIP diese Datei in einen anderen Nutzerbereich kopiert wird. Die aktuelle Nutzerbereichsnummer ist aus der Bereitschaftsmeldung ersichtlich.

4A> Nutzerbereichsnummer 4, Laufwerk A

A> Nutzerbereichsnummer 0, Laufwerk A

2B> Nutzerbereichsnummer 2, Laufwerk B

Die Nutzerbereichsnummer steht immer vor der Laufwerksbezeichnung. Nutzerbereich 0 ist Standardnutzerbereich und wird in der Bereitschaftsmeldung nicht angegeben.

Zum Ändern des aktuellen Nutzerbereiches kann der interne Befehl USER verwendet werden.

A>USER 3

3A>

Es können auch gleichzeitig Nutzerbereich und aktuelles Laufwerk geändert werden, indem eingegeben wird:

A>3B:

3B>

Die meisten Kommandos haben nur auf Dateien Zugriff, die dem aktuellen Nutzerbereich zugeordnet sind. Ist der aktuelle Nutzerbereich beispielsweise 7, dann werden mit einem DIR-Kommando ohne weitere Angaben nur die Dateien angezeigt, die unter der Nutzerbereichsnummer 7 erstellt wurden. Allerdings kann man aus allen Nutzerbereichen auf die Dateien zugreifen, die dem Nutzerbereich 0 zugeordnet und mit dem Dateiattribut SYS gekennzeichnet sind.

2.6. Zugriff auf mehrere Dateien

Bestimmte interne und transiente SCP-3.0-Kommandos können sich auf mehrere Dateien beziehen, wenn im Dateinamen oder Dateityp besondere Ersatzzeichen verwendet werden. Eine solche Dateibezeichnung wird auch mehrdeutige Dateibezeichnung genannt und kann sich auf mehrere Dateien beziehen, da sie nur eine Schablone für Dateibezeichnungen darstellt.

Die beiden Ersatzzeichen sind das Fragezeichen ?, was für einen beliebigen Buchstaben auf der gleichen Position steht, und der Stern *, der für beliebige Zeichen auf deren Positionen steht und beliebige andere Zeichen im Dateinamen oder Dateityp bleiben erhalten. Folgende Tabelle gibt die Regeln für die Verwendung von Ersatzzeichen an.

- Ein ? entspricht einem beliebigen Zeichen in einem Namen, wobei das Zeichen auch ein Leerzeichen sein kann.
- Ein * muß das letzte oder einzigste Zeichen in einem Dateinamen oder Dateityp sein. Intern ersetzt SCP 3.0 einen * durch Fragezeichen bis zum Ende des Dateinamens oder Dateityps.
- Ist der entsprechende Dateiname kürzer als acht Zeichen, dann nimmt SCP 3.0 Leerzeichen am Ende des Namens an.
- Ist der entsprechende Dateityp kürzer als drei Zeichen, dann nimmt SCP 3.0 Leerzeichen am Ende des Dateityps an.

Unter der Voraussetzung, daß auf einer Diskette folgende sechs Dateien enthalten sind:

A.COM, AA.COM, AAA.COM, B.COM, A.ASM und B.ASM

haben folgende Dateibezeichnungen die angegebene Wirkung:

.	wird behandelt wie ??????????.???
?????????.???	spricht alle sechs Namen an
*.COM	wird behandelt wie ??????????.COM
?????????.COM	spricht die ersten vier Namen an
?*.*	spricht A.COM und B.COM an
?.*	wird behandelt wie ?.???
?.*.*	spricht A.COM, B.COM, A.ASM und B.ASM an
A?*.COM	spricht A.COM und AA.COM an
A*.COM	wird behandelt wie A?????????.COM
A?????????.COM	spricht A.COM, AA.COM und AAA.COM an

SCP 3.0 verwendet derartige Schablonen mit Ersatzzeichen nur zum Durchsuchen eines Diskettenverzeichnisses. Deshalb sind Ersatzzeichen nur in Dateinamen und Dateitypen erlaubt. In einer Laufwerksbezeichnung darf kein Ersatzzeichen verwendet werden. Bei der Erstellung einer Datei darf ebenfalls im Dateinamen oder Dateityp kein Ersatzzeichen enthalten sein.

2.7. Dateischutz

Unter SCP 3.0 können Dateien in Gruppen organisiert werden, um sie vor einer zufälligen Veränderung oder unerlaubtem Zugriff zu schützen. Weiter kann spezifiziert werden, wie Dateien mit dem DIR-Kommando angezeigt werden und ob die Zeiteintragung der letzten Veränderung oder des letzten Zugriffs der Datei angezeigt wird. SCP 3.0 unterstützt diese Eigenschaften, indem den Dateien folgendes zugeordnet wird:

- Nutzerbereichsnummern
- Attribute
- Datum- und Zeiteintragungen
- Paßwörter

Alle diese Informationen zu jeder Datei werden im Diskettenverzeichnis gespeichert.

2.7.1. Dateiattribute

Dateiattribute steuern den Zugriff auf eine Datei. Wird eine Datei erstellt, dann erhält sie von SCP 3.0 zwei Attribute. Mit dem SET-Kommando können diese Attribute verändert werden.

Das erste Attribut kann entweder auf DIR (Directory) oder SYS (System) gesetzt werden. Dieses Attribut bewirkt die Anzeige der Dateinamen entweder mit dem DIR- oder dem DIR-SYS-Kommando. Wird eine Datei erstellt, dann erhält sie automatisch das DIR-Attribut. Der Dateiname einer mit dem DIR-Attribut gekennzeichneten Datei kann mit dem DIR-Kommando angezeigt werden. Erhält eine Datei das SYS-Attribut, dann kann der Dateiname mit dem DIRSYS-Kommando angezeigt werden. DIR- und DIRSYS-Kommandos zeigen ohne zusätzliche Angaben nur die Dateinamen an, die unter der aktuellen Nutzernummer erstellt wurden.

Eine unter dem Nutzerbereich 0 erstellte Datei mit dem SYS-Attribut besitzt einen speziellen Vorteil. Wird einer Datei mit der Nutzerbereichsnummer 0 das SYS-Attribut zugeordnet, dann kann diese Datei aus jedem anderen Nutzerbereich heraus gelesen und abgearbeitet werden. Diese Eigenschaft ermöglicht es auf einfache Art und Weise, die gebräuchlichsten Programme für alle Nutzerbereiche verfügbar zu machen. Es ist jedoch zu beachten, daß eine SYS-Datei im Nutzerbereich 0 nur mit einem DIRSYS-Kommando angezeigt wird, wenn 0 der aktuelle Nutzerbereich ist.

Das zweite Dateiattribut kann entweder R/W (Read-Write) oder R/O (Read-Only) sein. Wurde eine Datei mit R/O markiert, dann führt jeder Schreibversuch auf diese Datei zu einer Read-Only-Fehlermeldung. Demzufolge kann das Read-Only-Attribut zum Schutz wichtiger Dateien verwendet werden. Eine Datei mit dem R/W-Attribut kann jederzeit gelesen bzw. geschrieben werden oder gelöscht werden, wenn die Diskette nicht physisch schreibgeschützt ist.

2.7.2. Datum- und Zeiteintragung

Die Verwendung von Datum- und Zeiteintragungen ermöglichen ein schnelles Bestimmen der jüngsten Kopie einer Datei und ein Überprüfen, zu welchem Zeitpunkt die letzte Aktualisierung oder Veränderung stattgefunden hat. Es kann ausgewählt werden, ob das Erstellungsdatum oder der Zeitpunkt der letzten Lese- oder Schreiboperation auf die Datei gespeichert werden. Mit dem SET-Kommando werden Datum- und Zeiteintragungen eingerichtet und das DIR-Kommando mit der Option DATE zeigt die Zeiteintragungen der Dateien an.

Mit den folgenden Kommandos können Zeit- und Datumseintragungen für die Diskette ermöglicht werden, wobei aber zwischen ACCESS und CREATE gewählt werden muß. Wird ACCESS gewählt, dann enthält die Zeiteintragung den Zeitpunkt des letzten Zugriffs auf die Datei. Bei der Auswahl von CREATE enthält die Eintragung das Erstellungsdatum der Datei.

```
A>SET [ACCESS = ON]
A>SET [CREATE = ON]
A>SET [UPDATE = ON]
```

Dateien, die auf einer Diskette mit Datum- und Zeiteintragungen erstellt werden oder auf eine solche kopiert werden, erhalten automatisch die entsprechende Zeiteintragung. Das DATE Kommando ermöglicht die Anzeige und das Setzen des vom SCP 3.0 verwendeten Datums und der Zeit. Um die Beschreibung von Datum- und Zeiteintragungen zu vervollständigen, sind die Beschreibungen der Kommandos SET und INITDIR heranzuziehen.

2.7.3. Paßwörter

Paßwörter können verwendet werden, um den Zugriff auf bestimmte Dateien aus Sicherheitsgründen einzuschränken.

Mit SET wird sowohl die Einrichtung des Paßwortschutzes für eine Diskette durch die Zuordnung eines Paßwortes ermöglicht (so daß unberechtigte Nutzer den Paßwortschutz für ein Laufwerk nicht aufheben können), als auch die Zuordnung eines Paßwortes zu einer existierenden Datei. Paßwörter können allen Programm- und Datendateien zugeordnet werden. Daraus resultiert, daß eine Kommandozeile zur Ausführung die Angabe von zwei Paßwörtern erfordern kann: ein erstes Paßwort, um auf die Kommandodatei zuzugreifen und ein zweites Paßwort, um auf die im Zusatz angegebene Datei zuzugreifen.

Einige SCP-3.0-Kommandos und die meisten auf SCP 3.0 lauffähigen Anwenderprogramme erlauben keine Paßwörter im Zusatz. Soll ein eigenes Programm geschützt werden und dieses weiterhin Anwendung finden, dann kann ein Standardpaßwort gesetzt werden. Zur Erläuterung dieser Verfahrensweise ist die Beschreibung des SET-Kommandos nachzulesen.

2.8. Speichern von Dateien auf Diskette

SCP 3.0 speichert den Dateinamen, den Dateityp, das Paßwort, die Nutzerbereichsnummer und die Attribute einer jeden Datei in einem speziellen Bereich auf der Diskette, dem Verzeichnis, ab. Im Verzeichnis wird ebenfalls abgespeichert, welcher Bereich auf der Diskette zu welcher Datei gehört.

SCP 3.0 weist einer Datei den Verzeichnisplatz und Speicherplatz auf die gleiche Weise zu, wie die zur Datei gehörenden Sätze. Wird eine Datei gelöscht, dann gibt SCP 3.0 den Speicherplatz auf der Diskette wie folgt frei: der Platz der Verzeichniseintragung wird für das Katalogisieren einer anderen Datei freigegeben und ebenfalls der Speicherplatz der Datei auf der Diskette. Diese dynamische Speicherzuordnung verkörpert einen wesentlichen Anteil an der Leistungsstärke des SCP 3.0. Dem System muß nicht mitgeteilt werden, wie groß die Datei wird, denn es wird der Datei automatisch mehr Speicher als erforderlich zugeordnet und der Speicherplatz wieder freigegeben, wenn die Datei gelöscht wird. Mit dem SHOW-Kommando kann der noch verfügbare freie Speicherplatz auf der Diskette ermittelt werden.

2.9. Floppy Disk wechseln

SCP 3.0 kann erst dann mit einer Datei arbeiten, wenn die Diskette mit dieser Datei im Laufwerk liegt und das Laufwerk bereit ist. Liegt eine Diskette im Laufwerk, dann kann SCP 3.0 auf deren Verzeichnis und Dateien zugreifen.

Manchmal ist es erforderlich eine Diskette dem Laufwerk zu entnehmen und eine andere Diskette mit anderen Dateien einzulegen. Eine Diskette kann immer entnommen werden, wenn die System-Bereitschaftsmeldung auf dem Bildschirm ausgegeben wird. Das ist ein sicheres Kennzeichen dafür, daß kein Programm auf dieses Laufwerk schreibt oder von ihm liest.

Ein Diskettenwechsel kann auch nach Aufforderung durch ein Anwenderprogramm erfolgen. Das kann z. B. dann der Fall sein, wenn nicht alle vom Programm benötigten Daten auf eine Diskette passen.

Beachte: Eine Diskette darf niemals entnommen werden, wenn ein Programm auf sie schreibt oder von ihr liest.

Ein Diskettenwechsel kann erfolgen, ohne daß es dem System SCP 3.0 mitgeteilt werden muß. Damit ist es möglich auf Anforderung vom Programm eine andere Diskette einzulegen und von dieser Dateien zu lesen oder Dateien auf der neuen Diskette zu erstellen.

2.10. Schutz eines Laufwerkes

In Analogie zu den Dateien können im SCP 3.0 auch Laufwerke als R/O gekennzeichnet werden. Der Standardzustand eines Laufwerkes ist R/W. Einem Laufwerk kann das R/O Attribut mit dem SET-Kommando zugewiesen werden. Um das Laufwerk auf R/W zurückzusetzen, ist das SET-Kommando zu benutzen oder CTRL-C bei angezeigter System-Bereitschaftsmeldung zu betätigen.

2.11. RAM-Disk

Bei einer Speicherausstattung des PC 1715W mit 256K RAM wird dem Nutzer der zusätzliche Speicher als sogenannter RAM-Disk zur Verfügung gestellt.

Ein RAM-Disk simuliert alle Eigenschaften eines Floppy-Disk. Es erfolgt jedoch ein wesentlich schnellerer Zugriff auf den RAM-Disk, da alle „Diskettenzugriffe“ Speicherzugriffe darstellen.

Der RAM-Disk wird bei PC 1715W als „Laufwerk“ E: bereitgestellt, d. h., alle System- und Anwenderprogramme, die mit einem FD-Laufwerk arbeiten, können ebenso mit dem RAM-Disk arbeiten.

Das Laufwerk E: besitzt im einzelnen folgende Eigenschaften:

1152	128-Byte Sätze (142K)
144	K-Byte Kapazität
64	Verzeichniseinträge
8	Sätze/Block (1K-Blöcke)
384	Sektoren/Spur (48K)
3	Spuren.
0	Systemspuren
1024	Byte/phys. Sektor

Mit dem erstmaligen Starten des SCP 3.0 nach dem Einschalten wird das Verzeichnis des RAM-Disk gelöscht und mit dem Diskettenkennsatz ‚R‘ versehen, um ein gültiges Verzeichnis zu kennzeichnen. Jeder weitere Neustart des Systems mittels RESET-Taste läßt den RAM-Disk unverändert, wenn der Diskettenkennsatz mit dem Buchstaben ‚R‘ beginnt, d. h., alle gespeicherten Dateien bleiben über RESET erhalten.

Hinweis: Der selbständig bei Systemstart vergebene Diskettenkennsatz ist mit SHOW [LABEL] nicht anzeigbar. Er kann jedoch mittels des Kommandos SET [NAME = kennsatz] geändert und damit auch anzeigbar gemacht werden. Der Kennsatz sollte jedoch mit dem Buchstaben ‚R‘ beginnen, um den RAM-Disk bei RESET nicht zu löschen.

Das Laufwerk E: wird standardmäßig nach Systemstart als Laufwerk für temporäre Dateien benutzt (siehe Kommando SETDEF). Damit ist es möglich, mit Schreibschutz versehene Systemdisketten zu verwenden, auch wenn die Datei PROFILE.SUB abgearbeitet wird.

2.12. Automatische Formaterkennung der FD-Laufwerke

Das SCP 3.0 unterstützt die automatische Formaterkennung für alle am PC 1715W anschließbaren FD-Laufwerktypen, d. h., in einem Laufwerk können mehrere sogenannte Standardformate verarbeitet werden.

Zu beachten ist lediglich, daß ein Diskettenwechsel nur in der Kommandoebene (A> . . . E>) oder nach Aufforderung durch ein Anwenderprogramm erfolgen darf. Die Eingabe von ^C (Rücksetzen Diskettensystem) ist nicht erforderlich, auch wenn auf die Diskette geschrieben werden soll.

Folgende Standardformate werden vom SCP 3.0 automatisch erkannt:

Laufwerktyp	Standardformat
MFS 1.6	256 * 16 * 80 SS Sys3 256 * 16 * 80 DS Sys2 1024 * 5 * 80 SS Sys3 1024 * 5 * 80 DS Sys2 256 * 16 * 40 SS Sys3 nur lesen! 1024 * 5 * 40 SS Sys3 nur lesen!
MFS 1.4	256 * 16 * 80 SS Sys3 1024 * 5 * 80 SS Sys3 256 * 16 * 40 SS Sys3 nur lesen! 1024 * 5 * 40 SS Sys3 nur lesen!
MFS 1.2	256 * 16 * 40 SS Sys3 1024 * 5 * 40 SS Sys3
MF 6400	128 * 26 * 74 SS Sys3 1024 * 4 * 77 SS Sys3 1024 * 8 * 77 SS Sys2

(Sektorlänge * Sektoranzahl/Spur * Spüranzahl)

(SS-einseitig, DS-doppelseitig, Sys n – Anzahl Systemspuren)

Alle Formate arbeiten ohne SKEW FACTOR. Das SCP 3.0 kann weiterhin Diskettenformate verarbeiten, die sich von den Standardformaten nur durch die Anzahl Systemspuren unterscheiden (siehe Kommando MODFD)

3. Konsole und Drucker

Dieser Abschnitt beschreibt die Kommunikation des Systems SCP 3.0 mit Konsole und Drucker. Es wird erläutert, wie Konsol- und Druckerausgaben gestartet und gestoppt werden, die Edit-Kommandos für die Konsole und das Umlenken von Konsol- und Drucker-Ein- und Ausgaben. Weiterhin wird das Konzept der logischen Geräte unter SCP 3.0 erläutert.

3.1. Steuern der Konsolenausgabe

Damit vom System SCP 3.0 angezeigte Informationen auch gelesen werden können, kann mit Betätigen von CTRL-S die Anzeige angehalten werden. Wurde die Information gelesen, dann kann mit CTRL-Q die Anzeige fortgesetzt werden.

DIR, TYPE und andere SCP-3.0-Routinen unterstützen automatisch die seitenorientierte Ausgabe. Werden vom Programm mehr Informationen ausgegeben als der Bildschirm gleichzeitig anzeigen kann, dann wird die Anzeige automatisch angehalten, wenn der Bildschirm gefüllt ist. Das System SCP 3.0 fordert dann zum Fortsetzen der Anzeige durch Betätigen der Taste <ET> (Press RETURN to continue) auf.

3.2. Steuern der Druckerausgabe

Mit einem CTRL-Zeichen kann die Konsolenausgabe auch gleichzeitig auf den Drucker geleitet werden (printer echo). Gestartet wird diese Ausgabe durch CTRL-P. Um sie zu beenden ist wieder CTRL-P einzugeben. Bei eingeschalteter Druckerausgabe wird jedes Zeichen, das auf dem Bildschirm ausgegeben wird, auch auf dem Drucker ausgegeben.

In Verbindung mit einem DIR-Kommando kann damit eine Liste der Dateien erstellt werden, die sich auf einer Diskette befinden. Es kann CTRL-P auch in Verbindung mit CTRL-S und CTRL-Q verwendet werden, um einen Teil einer Datei auszudrücken. Dazu wird mit dem TYPE-Kommando die Anzeige einer Datei gestartet. Erreicht die Bildschirmanzeige den zu druckenden Teil, dann muß mit CTRL-S die Anzeige gestoppt und mit CTRL-P der Drucker zugeschaltet werden. Dann wird mit CTRL-Q die Anzeige fortgesetzt und gleichzeitig gedruckt. Mit einer weiteren Folge von CTRL-S, CTRL-P und CTRL-Q kann die Druckerausgabe wieder beendet werden.

3.3. Zeileneditierung auf der Konsole

Eingabefehler können mit den Kursortasten sowie den Tasten INS, DEL, CTRL-G und CTRL-H korrigiert werden. SCP 3.0 stellt außerdem noch andere Editierfunktionen in Form von CTRL-Zeichen zur Verfügung. Mit diesen CTRL-Zeichen können Kommandozeilen oder Eingabezeilen anderer Programme editiert werden.

Das System ermöglicht die Editierung von Kommandozeilen ohne Löschen aller Zeichen. Die zur Editierung vorhandenen CTRL-Zeichen werden in der nächsten Tabelle angegeben. Damit kann der Cursor nach links oder rechts bewegt werden, um Zeichen innerhalb einer Kommandozeile einzufügen oder zu löschen. Es müssen nicht alle Zeichen rechts von der korrigierten Stelle neu eingegeben werden. Es kann <ET> betätigt werden, wenn der Cursor an einer beliebigen Stelle innerhalb der Kommandozeile steht. SCP 3.0 liest dann die gesamte Kommandozeile. Ein Kommando kann auch wieder aufgerufen werden, um es zu wiederholen oder es zu modifizieren.

Im folgenden Beispiel wurde das PIP-Kommando falsch eingegeben und damit eine Fehlermeldung erzeugt. Mit CTRL-W wird die fehlerhafte Kommandozeile wieder aufgerufen. Um den Fehler zu korrigieren, werden folgende CTRL-Zeichen verwendet:

```
A>POP A:=B:*.*_ (fehlerhafte Eingabe)
POP?
A>POP A:=B:*.*_ (mit CTRL-W wieder geholt)
A>POP A:=B:*.*_ (mit CTRL-B auf Zeilenanfang)
A>PQP A:=B:*.*_ (mit CTRL-S nach rechts)
A>PE A:=B:*.*_ (mit CTRL-G Zeichen gelöscht)
A>PIE A:=B:*.*_ (Korrektur)
```

Um die korrigierte Kommandozeile auszuführen, muß <ET> betätigt werden, selbst wenn der Cursor in der Mitte der Kommandozeile steht. Das Kommando wird sowohl ausgeführt als auch in einem Kommandopuffer abgespeichert, so daß es zum erneuten Editieren oder wiederholten Ausführen mit CTRL-W gerufen werden kann.

Wird innerhalb der Kommandozeile ein Zeichen eingefügt, dann werden die Zeichen rechts vom Cursor nach rechts verschoben. Wird die Zeile dadurch länger als der Bildschirm breit ist, dann verschwinden die Zeichen am rechten Rand des Bildschirms. Diese Zeichen gehen aber nicht verloren. Sie tauchen wieder auf, wenn Zeichen in der Zeile gelöscht werden oder CTRL-E eingegeben wird, wenn der Cursor in der Mitte der Zeile steht. CTRL-E verschiebt alle Zeichen rechts vom Cursor in die nächste Bildschirmzeile.

Control-Zeichen für die Zeileneditierung im System SCP 3.0

Zeichen Bedeutung

CTRL-A	Bewegt den Cursor um eine Stelle nach links.
CTRL-B	Bewegt den Cursor auf den Anfang der Kommandozeile, ohne dabei den Inhalt der Zeile zu beeinflussen. Steht der Cursor am Zeilenanfang, dann wird er mit CTRL-B auf das Zeilenende bewegt.
CTRL-E	Bewirkt einen Zeilenvorschub ohne das Kommando an das SCP zu übergeben. Bewegt den Cursor an den Anfang der nächsten Zeile, ohne die bisherigen Eingaben zu löschen.
CTRL-F	Bewegt den Cursor um ein Zeichen nach rechts.
CTRL-G	Löscht das Zeichen auf Cursorposition. Der Cursor wird dabei nicht bewegt. Die Zeichen rechts vom Cursor werden um eine Stelle nach links verschoben.
CTRL-H	Löscht das Zeichen links vom Cursor und bewegt den Cursor um eine Zeichenposition nach links. Die Zeichen rechts vom Cursor werden um eine Stelle nach links verschoben.
CTRL-I	Bewegt den Cursor auf die nächste TAB-Marke. Die TAB-Marken sind automatisch auf jede achte Stelle gesetzt. Besitzt die gleiche Funktion wie die TAB-Taste
CTRL-J	Übergibt die Kommandozeile an das SCP 3.0 und setzt den Cursor auf den linken Rand der nächsten Zeile. Hat die gleiche Funktion wie RETURN oder CTRL-M.

Zeichen	Bedeutung
CTRL-K	Löscht alle Zeichen rechts vom Cursor bis Zeilenende.
CTRL-M	Übergibt eine Kommandozeile an das SCP 3.0 und setzt den Cursor auf den linken Rand der nächsten Zeile. Hat die gleiche Funktion wie RETURN oder CTRL-J.
CTRL-R	Wiederholt die Befehlszeile bis zum Cursor in der nächsten Zeile. An der Cursorposition erscheint ein #-Zeichen und der Cursor wird um eine Zeile in derselben Spalte nach unten bewegt.
CTRL-U	Löscht alle Zeichen in der Kommandozeile, setzt ein #-Zeichen an die Cursorposition und bewegt den Cursor in die nächste Zeile.
CTRL-W	Holt die letzte ausgeführte Kommandozeile aus dem Kommandopuffer und zeigt sie an. Der Cursor muß dabei am Anfang der Kommandozeile stehen, sonst wird CTRL-W nicht ausgeführt. Wiederholt wird die letzte mit CTRL-J, CTRL-M, CTRL-U oder <ET> abgeschlossene Kommandozeile. Wenn sich in der Kommandozeile bereits Zeichen befinden, dann wird mit CTRL-W der Cursor lediglich an das Ende der Zeile bewegt. Mit <ET> wird die wiederholte Zeile an das SCP 3.0 übergeben.
CTRL-X	Löscht alle Zeichen links vom Cursor und bewegt den Cursor an den Anfang der Zeile. Alle Zeichen rechts vom Cursor werden ab Zeilenanfang kopiert.

Aus der Tabelle ist ersichtlich, daß einige CTRL-Zeichen die gleiche Bedeutung besitzen. Das trifft z. B. für CTRL-M und CTRL-J zu, die die gleiche Bedeutung besitzen wie RETURN. Bei der Bildschirmausgabe wird einem Control-Zeichen das Zeichen ^ vorangestellt.

3.4. Umlenken von Ein- und Ausgaben

Mit dem Kommando PUT kann die Ausgabe auf dem Bildschirm oder Drucker in eine Diskettendatei umgelenkt werden. Mit dem GET-Kommando wird die Konsoleneingabe für SCP 3.0 oder ein Programm anstelle von der Konsole aus einer Diskettendatei genommen. Die nachfolgenden Beispiele zeigen einige der Möglichkeiten, die GET und PUT bieten.

PUT wird verwendet, um die Ausgabe von der Konsole in eine Diskettendatei so umzuleiten, als ob es die Konsole wäre. Mit PUT kann eine Diskettendatei erstellt werden, die das Verzeichnis aller auf dieser Diskette gespeicherten Dateien enthält.

```
A>PUT CONSOLE OUTPUT TO FILE DIR.PRN
```

```
Putting console output to file: DIR.PRN
```

```
A>DIR
```

```
A:NAME      TEX : FRONT      TEX : FRONT      BAK : ONE      TEX
A:FOUR      TEX : ONE        BAK : LINEDIT    TEX : EXAMP1   TXT
A:TWO       TEX : THREE      BAK : EXAMP2     TXT
```

```
A>TYPE DIR.PRN
```

```
A:NAME      TEX : FRONT      TEX : FRONT      BAK : ONE      TEX
A:FOUR      TEX : ONE        BAK : LINEDIT    TEX : EXAMP1   TXT
A:TWO       TEX : THREE      BAK : EXAMP2     TXT
```

Mit einem ähnlichen PUT-Kommando kann die Druckerausgabe in eine Diskettendatei umgelenkt werden.

Das GET-Kommando ermöglicht die Verarbeitung von Daten, die normalerweise über die Tastatur eingegeben werden, aus einer Diskettendatei. Soll diese Datei von System SCP 3.0 verarbeitet werden, dann muß sie SCP-3.0-Standard-Kommandozeilen beinhalten. Wenn die Diskettendatei einem Programm als Eingabe dienen soll, muß sie sich aus den vom jeweiligen Programm lesbaren Kommandos und Daten zusammensetzen. Die Eingabedatei darf sowohl SCP-3.0-Kommandos als auch Programmeingaben beinhalten.

Mit der Option SYSTEM im GET-Kommando wird festgelegt, ob das System SCP 3.0 oder ein Dienstprogramm die Datei liest. Das folgende Beispiel soll dies verdeutlichen. Wird die SYSTEM-Option weggelassen, dann kann das Programm begonnen werden, welches die Eingaben von der angegebenen Datei nimmt. Ist die SYSTEM-Option enthalten, dann geht SCP 3.0 unmittelbar zur Eingabe aus der angegebenen Datei über.

```
3A>type pip.dat
```

```
b:=front.tex
```

```
b:=one.tex
```

```
b:=two.tex
```

```
3A>get console input from file pip.dat
```

```
Getting console input from file: PIP.DAT
```

```
3A>pip
```

```
SCP3 PIP R-BWS V 3/0
```

```
*b:=front.tex
```

```
*b:=one.tex
```

```
*b:=two.tex
```

```
*^C
```

```
3A>type ccp.dat
```

```
dir
```

```
show
```

```
dirsys
```

```
3A>get console input from file ccp.dat [system]
```

```
Getting console input from file: CCP.DAT
```

```
3A>dir
```

A: NAME	TEX : FRONT	TEX : FRONT	BAK : ONE	TEX
A: FOUR	TEX : ONE	BAK : LINEDIT	TEX : EXAMP1	TXT
A: TWO	TEX : THREE	BAK : EXAMP2	TXT : PIP	DAT
A: EXAMP3	: EXAMP4	: CCP	DAT	

```
3A>show
```

```
A: RW, Space: 299k
```

```
B: RW, Space: 311k
```

```
3A>dirsys
```

```
NON-SYSTEM FILE(S) EXIST
```

3.5. Zuordnung logischer Einheiten

Der Computer verfügt über Tastatur und Bildschirm. Neben einem Drucker können aber auch weitere Geräte angeschlossen werden, wie z. B. ein anderer Drucker oder ein Modern. Um auf diese physisch verschiedenen Geräte zugreifen zu können, ordnet SCP 3.0 ver-

schiedenen physischen Geräten logische Geräte zu. Die nachfolgende Tabelle enthält eine Übersicht über die im SCP 3.0 vorhandenen logischen Geräte. Weiterhin werden die in der Vertriebsversion zugeordneten physischen Geräte angegeben.

Logische Geräte des SCP 3.0

Logischer Gerätename	Gerätetyp	Zuordnung des physischen Gerätes
CONIN:	Konsoleneingabe	Tastatur
CONOUT:	Konsolenausgabe	Bildschirm
AUXIN:	Externe Eingabe	—
AUXOUT:	Externe Ausgabe	—
LST:	List-Ausgabe	Drucker

Diese Zuordnungen können mit dem DEVICE Kommando verändert werden.

4. SCP 3.0 Kommandokonzept

Wie bereits beschrieben, besteht eine Kommandozeile des SCP 3.0 aus einem Kommando-Schlüsselwort, einem Zusatz und einem Zeilenabschluß (Wagenrücklauf). In diesem Abschnitt werden zwei Arten von Kommandos beschrieben, die durch ein Kommando-Schlüsselwort angesprochen werden. Weiterhin wird beschrieben, wie SCP 3.0 eine Programmdatei auf einer Diskette sucht, wie Mehrfachkommandos ausgeführt werden und wie das Diskettensystem zurückgesetzt wird.

4.1. Zwei Arten von Kommandos

Ein Kommandoschlüsselwort bezeichnet ein Programm, welches sich entweder als Bestandteil des SCP 3.0 im Speicher oder als Programmdatei auf einer Diskette befindet. Interne Kommandos bezeichnen im Speicher vorhandene Programme, und externe oder transiente Kommandos bezeichnen Programmdateien auf Disketten.

Das System SCP 3.0 verfügt über sechs interne und mehr als zwanzig transiente Kommandos. Der Systemservice kann durch den Erwerb von SCP 3.0 kompatiblen Anwenderprogrammen weiter erhöht werden.

4.2. Interne Kommandos

Interne Kommandos sind Teil des SCP 3.0 und stets verfügbar, unabhängig davon, welche Diskette in welchem Laufwerk liegt. Interne Kommandos befinden sich im Speicher als Teil des SCP 3.0 und können deshalb auch schneller ausgeführt werden als transiente Routinen.

Einige interne Kommandos bieten Optionen, die eine Unterstützung durch ein zugehöriges transientes Programm erfordern. Das zugehörige transiente Programm hat den gleichen Namen wie das interne und den Dateityp COM. Dieser transiente Teil wird nur dann geladen, wenn die entsprechende Kommandozeile Optionen enthält, die vom internen Kommando nicht verarbeitet werden können.

Werden bestimmte Optionen im Zusatz des internen Kommandos angegeben, dann kann es passieren, daß SCP 3.0 die Anforderungsmeldung „COM Required“ ausgibt. Damit wird zum Ausdruck gebracht, daß die Optionen im Zusatz des Kommandos eine zu-

gehörige transiente Datei erfordern und SCP 3.0 kann diese Programmdatei nicht finden. Folgende Dateien müssen im Zugriff sein, um alle Funktionen der zugehörigen Kommandos zu realisieren: ERASE.COM, RENAME.COM, TYPE.COM und DIR.COM.

Interne Kommandos

Kommando	Funktion
DIR	Zeigt die Dateinamen aller im Verzeichnis enthaltenen Dateien an, außer den mit dem SYS-Attribut gekennzeichneten.
DIRSYS	Zeigt die Dateinamen der mit dem SYS-Attribut gekennzeichneten Dateien aus dem Verzeichnis an.
ERASE	Löscht einen Dateinamen aus dem Verzeichnis und gibt den durch die Datei belegten Speicherplatz wieder frei.
RENAME	Umbenennen einer Diskettendatei.
TYPE	Zeigt den Inhalt einer Textdatei auf dem Bildschirm an.
USER	Wechsel auf einen anderen Nutzerbereich.

SCP 3.0 ermöglicht die Abkürzung der internen Kommandos wie folgt:

DIRSYS	DIRS
ERASE	ERA
RENAME	REN
TYPE	TYP
USER	USE

4.3. Transiente Kommandos

Wird ein Kommando-Schlüsselwort eingegeben, welches ein transientes Kommando bezeichnet, dann lädt SCP 3.0 die Programmdatei von der Diskette und übergibt alle im Zusatz des Kommandos angegebenen Dateinamen, Daten oder Parameter an das Programm. Das nächste Kapitel liefert detaillierte Informationen über die in der nachfolgenden Tabelle aufgeführten transienten Kommandos.

Transiente Kommandos

Name	Funktion
COPYSYS	Erstellt eine neue Unladediskette.
DATE	Setzen oder Anzeigen von Datum und Uhrzeit.
DEVICE	Weist logische SCP-Geräte einem oder mehreren physischen Geräten zu, ändert Gerätetreiberprotokolle und Baudraten und definiert die Bildschirmgröße.
DUMP	Zeigt eine Datei im ASCII und hexadezimalen Format an.
GENCOM	Erstellen einer speziellen COM-Datei mit zugehörigen RSX-Dateien
GET	Vorübergehende Konsoleingaben aus einer Datei anstelle über die Tastatur.
HELP	Anzeige von Informationen über die Anwendungen der SCP-3.0-Kommandos.
INIT	Initialisieren Diskette
INITDIR	Initialisiert ein Diskettenverzeichnis für die Zeiteintragungen.

Name	Funktion
LINK	Programmverbinder
MODCS	Nachladen Zeichengenerator für Bildschirm
MODFD	Modifizieren der Parameter für Floppy Disk
MODIO	Modifizieren der Treiber für Zeichen-Ein-/Ausgabe
PIP	Kopiert und kombiniert Dateien.
PUT	Vorübergehende Umlenkung von Konsolen- oder Druckerausgaben auf Diskettendatei.
SET	Eintragen von Dateikennzeichen wie Diskettenkennsatz, Dateiattribute, Typ der Zeiteintragung und Paßwortschutz.
SETDEF	Setzen von System-Optionen einschließlich Suchreihenfolge der Laufwerke.
SHOW	Anzeige von Disketten- und LW-Parametern.
SUBMIT	Automatisches Ausführen von Mehrfachkommandos.

4.4. Suchen von Programm- und Datendateien durch das SCP 3.0

In diesem Abschnitt wird beschrieben, wie SCP-3.0-Programm- und Datendateien auf einer Diskette sucht. Wird die in einer Kommandozeile angegebene Programmdatei nicht gefunden, dann kann es auch daran liegen, daß SCP 3.0 nicht auf der Diskette sucht, auf der die Datei abgespeichert ist. Deshalb ist es erforderlich, die Vorgehensweise des SCP 3.0 bei der Suche einer Programm- oder Datendatei zu verstehen.

4.4.1. Finden einer Datendatei

Wird ein eingegebenes Kommando an das SCP 3.0 gesendet, dann wird dem im Kommando-Schlüsselwort angegebenen Programm der Zusatz übergeben. Enthält der Zusatz eine Dateibezeichnung, dann wird SCP 3.0 vom Programm dazu veranlaßt, nach dieser Datendatei zu suchen. Kann SCP 3.0 die Datendatei nicht finden, dann gibt das Programm auf der Konsole eine Fehlermeldung aus. Normalerweise heißt diese Fehlermeldung „File not found“ oder „No File“, aber sie ist abhängig von dem im Kommando-Schlüsselwort angegebenen Programm.

Enthält die im Zusatz vorhandene Dateibezeichnung keine Laufwerksangabe, dann durchsucht SCP 3.0 das Verzeichnis des aktuellen Nutzerbereiches auf dem Standardlaufwerk. Ist die Datei dort nicht enthalten, dann sucht SCP 3.0 nach dieser Datei mit dem SYS-Attribut im Nutzerbereich 0 des Standardlaufwerkes. Wird die Datei im Nutzerbereich 0 gefunden, dann ist ein Read-Only (Nur-Lese) -Zugriff möglich. Wird z. B. folgende Kommandozeile eingegeben

```
3A>TYPE MYFILE.TXT
```

dann durchsucht SCP 3.0 zuerst das Verzeichnis vom Nutzerbereich 3 im Laufwerk A. Wird die Datei MYFILE.TXT dort nicht gefunden, dann wird das Verzeichnis des Nutzerbereiches 0 auf Laufwerk A nach der Datei MYFILE.TXT mit dem Dateiattribut SYS durchsucht. Ist die Datei in keinem der Verzeichnisse, dann gibt SCP 3.0 die Steuerung zurück an TYPE und TYPE gibt die Meldung aus „No File“.

Einige der Routinen wie PIP und DIR beschränken ihre Dateisuche auf den aktuellen Nutzerbereich. Da das SCP 3.0 den Read-Write (Schreib-Lese) -Zugriff auf SYS-Dateien nicht ermöglicht, beschränken ERASE und RENAME ihre Suche ebenfalls auf den aktuellen Nutzerbereich.

Die Prozedur des Suchens ist auch bei Angabe eines Laufwerkes in der Dateibezeichnung grundlegend die gleiche. SCP 3.0 durchsucht zuerst das Verzeichnis des aktuellen Nutzerbereiches im angegebenen Laufwerk. Wenn die Datei nicht gefunden wurde, dann wird das Verzeichnis des Nutzerbereiches 0 auf dem angegebenen Laufwerk nach der Datei mit dem SYS Attribut durchsucht. Wird die Datendatei nach diesen Suchläufen von SCP 3.0 nicht gefunden, dann wird eine Fehlermeldung ausgegeben.

4.4.2. Finden einer Programmdatei

Die Suche nach einer Datendatei kann sich von der nach einer Programmdatei sehr stark unterscheiden. Das kann mit Hilfe des SETDEF-Kommandos erreicht werden, mit dem die Suchreihenfolge für das SCP 3.0 zur Suche nach einer Programmdatei festgelegt werden kann. Mit dem SETDEF-Kommando kann das SCP 3.0 dazu veranlaßt werden, sechzehn Suchläufe auszuführen, falls keine Laufwerksangabe vor dem Kommando-Schlüsselwort steht; allerdings ist das ein sehr seltener Fall! Zunächst soll beschrieben werden, wie SCP 3.0 eine Programmdatei sucht, ohne daß vorher ein SETDEF-Kommando gegeben wurde.

Bezeichnet ein Kommandoschlüsselwort ein transientes Programm, dann sucht SCP 3.0 die Programmdatei auf dem angegebenen oder dem Standardlaufwerk. Es wird im aktuellen Nutzerbereich gesucht und dann im Nutzerbereich 0 nach der Datei mit dem SYS-Attribut. SCP 3.0 bricht den Suchprozess sofort ab, wenn die Programmdatei gefunden wurde. Das Programm wird dann in den Speicher geladen und ausgeführt. Nach Beendigung des Programmes gibt SCP 3.0 die System-Bereitschaftsmeldung aus und erwartet das nächste Kommando. Findet SCP 3.0 die Programmdatei jedoch nicht, dann wird die Kommandozeile noch einmal ausgegeben mit einem Fragezeichen und das nächste Kommando erwartet.

Steht vor dem Kommando-Schlüsselwort aber eine Laufwerksangabe, dann wird damit dem SCP 3.0 genau vorgeschrieben, wo zu suchen ist. Demzufolge sucht SCP 3.0 nur an zwei Stellen: im Verzeichnis des aktuellen Nutzerbereiches im angegebenen Laufwerk und dann im Nutzerbereich 0 des angegebenen Laufwerkes. Erst dann kann es zur wiederholten Anzeige der Kommandozeile mit einem Fragezeichen kommen.

Wird z. B. eingegeben

```
4C>A:SHOW [SPACE]
```

dann sucht SCP 3.0 auf Laufwerk A zunächst im Nutzerbereich 4 und anschließend im Nutzerbereich 0.

Wird vor dem Kommando-Schlüsselwort keine Laufwerksbezeichnung angegeben, dann durchsucht SCP 3.0 die Verzeichnisse in einer bestimmten Reihenfolge, die auch als Laufwerkette bezeichnet wird. Ausgeliefert wird das SCP 3.0 mit nur einer Laufwerksangabe in dieser Kette, dem Standardlaufwerk. Ohne Veränderung dieser Laufwerkette mit dem SETDEF-Kommando sucht SCP 3.0 nur an zwei Stellen nach der gewünschten Programmdatei. Wird beispielweise eingegeben

```
7D>SHOW [SPACE]
```

dann sucht SCP 3.0 nach der Datei SHOW.COM an folgenden Stellen:

- 1) Laufwerk D, Nutzerbereich 7
- 2) Laufwerk D, Nutzerbereich 0

Es ist zu beachten, daß die Datei SHOW.COM im Nutzerbereich 0 mit dem SYS-Attribut gekennzeichnet sein muß, sonst wird sie vom SCP 3.0 nicht gefunden. Mit dem SET-Kommando können Dateien im Nutzerbereich 0 das SYS Attribut erhalten und es kann auf diese aus allen anderen Nutzerbereichen automatisch zugegriffen werden. Damit entfällt das Kopieren häufig benutzter Programme in alle Nutzerbereiche auf einer Diskette.

Wird mit dem SETDEF-Kommando eine spezielle Laufwerkette definiert, dann sollte das Standardlaufwerk enthalten sein und auch das Laufwerk, welches die Diskette mit den am meisten verwendeten Routinen enthält. Als Beispiel soll eine Laufwerkette mit * (aktuelles Laufwerk) und Laufwerk A definiert werden. Wird dann folgendes Kommando eingegeben

```
2D>SHOW [SPACE]
```

dann sucht SCP 3.0 die Datei SHOW.COM in folgender Reihenfolge:

- 1) Laufwerk D, Nutzerbereich 2
- 2) Laufwerk D, Nutzerbereich 0
- 3) Laufwerk A, Nutzerbereich 2
- 4) Laufwerk A, Nutzerbereich 0

Mit einer Option in einem SETDEF-Kommando kann das Standardlaufwerk in eine Laufwerkette aufgenommen werden. Jede mit dem SETDEF-Kommando definierte Laufwerkette bleibt bis zum Neustart des Systems gültig.

Das SETDEF-Kommando kann ferner auch dazu genutzt werden, automatisch nach einer Submit-Datei in der Laufwerkette zu suchen. Nähere Ausführungen dazu sind im nachfolgenden Abschnitt enthalten.

4.5. Ausführen von Mehrfachkommandos

In den bisherigen Beispielen hat SCP 3.0 zu einer bestimmten Zeit immer nur ein Kommando ausgeführt. SCP 3.0 kann aber auch eine Folge von Kommandos ausführen. Eine Folge von Kommandos kann nach dem System-Bereitschaftszeichen eingegeben oder in eine Diskettendatei abgelegt werden. Ist die Folge einmal in einer Diskettendatei abgespeichert, dann kann sie wann immer es gewünscht wird mit dem SUBMIT-Kommando abgearbeitet werden.

Um nach dem System-Bereitschaftszeichen Mehrfachkommandos einzugeben, ist jedes Kommando-Schlüsselwort und der zugehörige Zusatz vom nächsten Schlüsselwort durch ein Ausrufungszeichen abzutrennen. Ist die Folge komplett, dann wird <ET> eingegeben. SCP 3.0 führt die Kommandos der Reihe nach aus:

```
3A>dirsys!dir examp*.*!show [space]
```

```
NON-SYSTEM FILE(S) EXIST
```

```
3A>dir examp*.*
```

```
A:EXAMP7      : EXAMP1      TXT : EXAMP3      : EXAMP2      TXT
```

```
A:EXAMP5      : EXAMP6
```

```
3A>show [space]
```

```
A: RW, Space: 2,221 k
```

Wird eine bestimmte Kommandofolge häufig benötigt, dann sollte die Folge in eine Diskettendatei abgespeichert werden. Eine solche Datei kann mit einem Texteditor erstellt werden. Die Datei muß den Dateityp `.SUB` besitzen. Jedes Kommando in der Datei muß auf einer neuen Zeile beginnen. Z. B. könnte die Datei `UPDATE.SUB` wie folgt aufgebaut sein

```
DIR A:*.COM  
ERA B:*.COM  
PIP B:=A:*.COM
```

Um diese Reihe von Kommandos auszuführen, ist folgendes Kommando einzugeben:

```
A>SUBMIT UPDATE
```

Das `SUBMIT`-Programm übergibt jedes Kommando an das `SCP 3.0`, damit diese nacheinander ausgeführt werden können. Während `SUBMIT` arbeitet werden die Kommandos auf dem Bildschirm noch einmal zusätzlich zu den Bildschirmausgaben der einzelnen Kommandos ausgegeben. Ist ein Kommando beendet, dann wird die Systembereitschaftsmeldung mit dem nächsten Kommando aus der `.SUB`-Datei ausgegeben. Wenn alle Kommandozeilen der `.SUB`-Datei abgearbeitet wurden, dann wartet das System auf die nächste Kommandozeile über die Tastatur.

Die Datei `PROFILE.SUB` ist eine spezielle Submit-Datei. Existiert eine solche Datei, dann wird sie automatisch vom `SCP 3.0` ausgeführt. Das ist insbesondere dann vorteilhaft, wenn eine feststehende Menge von Kommandos jeweils vor Beginn der eigentlichen Arbeit mit dem Computer auszuführen ist; z. B. `SETDEF` und `DATE SET`. Existiert auf der Vertriebsdiskette keine solche Datei `PROFILE.SUB`, dann kann sie mit dem Texteditor erstellt werden.

Weitere Informationen über das Erstellen einer Submit-Datei und die Anwendung der `SUBMIT`-Parameter sind der Beschreibung von `SUBMIT` zu entnehmen.

Mit dem Kommando `SETDEF` kann ein automatisches Submit eingestellt werden. Dazu veranlaßt `SETDEF` das System nach `SUB`-Dateien so zu suchen, wie nach `COM`-Dateien in der vorgegebenen Suchreihenfolge gesucht wird. Damit wird die Anzahl der Suchläufe verdoppelt bevor eine Fehlermeldung ausgegeben wird. Der Vorteil dieses automatischen Submit liegt darin, daß der Name einer `SUB`-Datei als Kommando-Schlüsselwort in einer Kommandozeile angegeben werden kann. Findet `SCP 3.0` die `SUB`-Datei, dann werden die enthaltenen Kommandos automatisch ausgeführt.

4.6. Unterbrechen des Programmablaufes

Mit `CTRL-C` kann die Ausführung eines Programmes unterbrochen oder das Rücksetzen der Diskettenlaufwerke erreicht werden.

Eine Reihe von transienten Programmen kann mit `CTRL-C` direkt abgebrochen werden. Wenn ein Programmablauf während der Datenausgabe auf den Bildschirm abgebrochen werden soll, muß zuerst die Ausgabe mit `CTRL-S` angehalten und dann `CTRL-C` eingegeben werden.

Mit `CTRL-C` können auch die Diskettenlaufwerke rückgesetzt werden. Wird `CTRL-C` nach Anzeige des System-Bereitschaftszeichens eingegeben, dann werden alle aktiven Laufwerke, deselektiert. Die aktiven Laufwerke sind alle die Laufwerke, auf die nach dem letzten Kalt- oder Warmstart zugegriffen wurde. Das Kommando `SHOW [SPACE]` zeigt den verbleibenden Platz auf allen aktiven Laufwerken an. Im folgenden Beispiel ermittelt das Kommando `SHOW [SPACE]` drei aktive Laufwerke. Wird jedoch unmittelbar nach dieser

Anzeige CTRL-C und erneuert das Kommando SHOW [SPACE] eingegeben, dann wird nur der Platz auf dem aktuellen Laufwerk A angezeigt.

```
A>SHOW [SPACE]
A: RW, Space: 420k
B: RO, Space: 142k
C: RO, Space: 500k
A>^C
A>SHOW [SPACE]
A: RW, Space: 420k
```

4.7. Das HELP-Programm

Mit dem transienten Kommando HELP kann man aus einer Hilfsdatei Informationen über die in diesem Handbuch beschriebenen Kommandos, deren Eingaben und deren Bedienung auf dem Bildschirm anzeigen. Um das Programm aufzurufen ist ganz einfach einzugeben:

```
A>HELP
```

Damit zeigt HELP eine Liste von Hauptthemen an, für die weitere Informationen verfügbar sind. Nach der Anzeige der vorhandenen Hauptthemen wird das Bereitschaftszeichen des HELP-Programms ausgegeben

```
HELP>
```

Danach kann eines der in der Liste vorhandenen Hauptthemen eingegeben werden, z. B.

```
HELP>SHOW
```

Es wird eine Zusammenfassung des SHOW-Kommandos ausgegeben und anschließend zeigt HELP eine Liste von Teilthemen an, die verschiedene Aspekte des SHOW-Kommandos näher beschrieben. Um Informationen über ein Teilthema anzuzeigen, nachdem die Informationen des Hauptthemas gelesen wurden, ist der Name des Teilthemas mit vorangestelltem Punkt einzugeben.

```
HELP>.OPTIONEN
```

Im voranstehenden Beispiel zeigt HELP die für das SHOW-Kommando verfügbaren Optionen an. Sind Informationen nur für ein spezielles Teilthema erwünscht, dann kann auch unmittelbar nach dem System-Bereitschaftszeichen mit folgendem Kommando darauf zugegriffen werden:

```
A>HELP SHOW OPTIONEN
```

Mit HELP kann die Handhabung der SCP-3.0-Kommandos schnell erlernt werden. Die Beschreibungen der einzelnen Kommandos in diesem Handbuch werden dann nur noch zur Klärung von Fragen herangezogen, die aus den Zusammenfassungen des HELP-Kommandos heraus nicht geklärt werden können. Die Informationen des HELP-Kommandos können auch erweitert, modifiziert oder aber auch gekürzt werden. Genaueres darüber enthält die Beschreibung des HELP-Kommandos selbst.

5. Zusammenstellung der Kommandos

In diesem Abschnitt werden die mit dem Betriebssystem SCP 3.0 gelieferten Kommandos und Programme beschrieben. Die Kommandos werden in alphabetischer Reihenfolge beschrieben. Jedem Kommando folgen eine kurze Beschreibung seiner Arbeitsweise und Beispiele.

5.1. Struktur der SCP-3.0-Kommandos

Dieser Abschnitt beschreibt die Teile einer Dateibezeichnung in einer Kommandozeile. Eine Dateibezeichnung besteht aus vier Teilen, die zur Unterscheidung jeweils eine spezielle formale Bezeichnung haben

- Laufwerksbezeichner – wahlweise Angabe des Disketten-Laufwerkes A, B, C, . . . , P, auf dem die Datei oder die Gruppe von Dateien, auf die Bezug genommen werden soll, katalogisiert ist. Wird in einer Kommandozeile ein Laufwerksbezeichner angegeben, dann muß hinter diesem ein Doppelpunkt gesetzt werden.
- Dateiname – ein ein bis acht Zeichen langer Name einer Datei oder einer Gruppe von Dateien.
- Dateityp – die wahlweise Angabe einer ein bis drei Zeichen langen Bezeichnung der Kategorie einer Datei oder einer Gruppe von Dateien. Wird ein Dateityp angegeben, dann muß dieser durch einen Punkt vom Dateinamen getrennt werden.
- Paßwort – die wahlweise Angabe eines ein bis acht Zeichen langen Paßwortes, welches den Schutz von Dateien ermöglicht.

Wird kein Laufwerksbezeichner angegeben, dann arbeitet SCP 3.0 automatisch mit dem Standardlaufwerk. Werden Punkt und Dateityp weggelassen, dann fügt SCP 3.0 automatisch als Dateityp drei Leerzeichen an.

Diese allgemeine Form wird Dateibezeichnung genannt. Eine Dateibezeichnung bezeichnet eine bestimmte Datei oder Gruppe von Dateien im Verzeichnis der durch den Laufwerksbezeichner zugewiesenen Diskette.

Zum Beispiel

B:MYFILE.DAT

ist eine Dateibezeichnung, die Laufwerk B:, den Dateinamen MYFILE, und den Dateityp DAT angibt. Die Dateibezeichnung wird in der Syntaxzeile der Kommandos mit

`dateibez`

abgekürzt.

Einige SCP-3.0-Kommandos erlauben im Dateinamen und Dateityp die Verwendung von Ersatzzeichen als Teile des Arguments. Z. B. ist

B:MY*.A??

eine Dateibezeichnung mit dem Laufwerksbezeichner B:, dem Dateinamen MY* und dem Dateityp A??. Diese Schablone einer Dateibezeichnung kann auf verschiedene Dateien im Verzeichnis passen.

Die Teile einer Dateibezeichnung lassen sich wie folgt darstellen:

`d:dateiname.typ;passwort`

Dabei entspricht d: dem Laufwerksbezeichner, dateiname ist der bis zu acht Zeichen lange Name einer Datei und typ ist der bis zu drei Zeichen lange Dateityp. In diesem Kapitel verwenden die Syntaxbeschreibungen den Ausdruck `dateibez`, um eine gültige Kombination der in einer Dateibezeichnung möglichen Elemente darzustellen. In der folgenden Liste sind gültige SCP-3.0-Dateibezeichnungen angegeben.

dateiname
 dateiname.typ
 dateiname;passswort
 dateiname.typ;passwort
 d:dateiname
 d:dateiname.typ
 d:dateiname;passwort
 d:dateiname.typ;passwort

Die Zeichen in der nachfolgenden Tabelle besitzen im SCP 3.0 eine spezielle Bedeutung, die bei Angabe dieser Zeichen beachtet werden muß.

Reservierte Zeichen

Zeichen	Bedeutung
< = , ! > []	Begrenzer für Dateibezeichnung
TAB Leertaste	
<cr>	
:	Laufwerkbegrenzer in Dateibezeichnung
.	Dateitypbegrenzer in Dateibezeichnung
;/	Paßwortbegrenzer in Dateibezeichnung
* ?	Ersatzzeichen in mehrdeutigen Dateibezeichnungen
< > & ! \ + -	Begrenzer für Listenelemente
[]	Begrenzer in Optionenliste für globale und lokale Optionen
()	Begrenzer für Mehrfachmodifizierer innerhalb eckiger Klammern für Optionen mit Modifizierern
/ □	Optionenbegrenzer in einer Kommandozeile
;	Kommentarbegrenzer am Anfang einer Kommentarzeile

Innerhalb des SCP 3.0 existieren bereits verschiedene Gruppen von Dateien. Die nachfolgende Tabelle enthält einige dieser Dateitypen mit einer kurzen Beschreibung der Kategorie. Im Anhang ist eine vollständige Liste enthalten.

SCP-3.0-Dateitypen

Dateityp	Bedeutung
MAC	Assembler-Quelldatei
BAS	BASIC-Quellprogramm
COM	ausführbares Programm in Maschinensprache
HLP	Datei mit Informationen für HELP
SUB	Liste von Kommandos, die durch SUBMIT ausgeführt werden
□ □ □	temporaere Datei (Zwischendatei)

In einigen Kommandos werden für die Darstellung der Dateibezeichnungen zusätzlich Bezeichner verwendet, mit denen nähere Informationen über die Dateibezeichnungen dargestellt werden. Zum Beispiel wird mit `quell-dateibez` die Dateibezeichnung der Quelldatei angegeben.

5.2. Darstellung der Kommandos

Die SCP-3.0-Kommandos werden in alphabetischer Reihenfolge beschrieben. Jede Kommandobeschreibung besitzt einen bestimmten Aufbau.

- Die Darstellung beginnt mit dem Kommando-Schlüsselwort in Großbuchstaben.
- Der Syntax-Abschnitt enthält eine oder mehrere allgemeine Formen als Vorschrift für die Bildung einer Kommandozeile.
- Der Abschnitt Erklärung beschreibt die allgemeine Anwendung des Kommandos und hebt Ausnahmen und Spezialfälle hervor. Dieser Abschnitt kann Tabellen mit den für das Kommando verfügbaren Optionen enthalten.
- Der Abschnitt Beispiele stellt eine Anzahl gültiger Kommandozeilen mit dem beschriebenen Kommando-Schlüsselwort dar. Um die Dialogarbeit mit dem System auszudrücken, werden alle Eingaben vom Bediener fettgedruckt dargestellt. Die Anzeigen des Betriebssystems als Reaktion auf eine Bedieneraktion werden im normalen Druck dargestellt.

Die Angaben in der Syntaxzeile beschreiben die allgemeine Form des Kommandos unter Beachtung folgender Regeln:

- Wörter in Großbuchstaben müssen exakt wie angegeben geschrieben werden, aber in einer beliebigen Kombination von Groß- und Kleinbuchstaben.
- Ein Wort in Kursivschrift in der Kommandozeile besitzt eine allgemeine Bedeutung, die im Text definiert wird.
- Die symbolischen Angaben *d:*, *dateiname*, *:typ*, *;paßwort* und *dateibez* besitzen die im vorangegangenen Abschnitt beschriebene Bedeutung.
- Wenn nicht anders angegeben, können an einer Stelle, an der ein Leerzeichen steht, ein oder mehrere Leerzeichen eingegeben werden. Zum Beispiel müssen die Optionen von PIP nicht durch Leerzeichen getrennt werden.

Nachfolgende Tabelle definiert die in der Syntaxzeile angegebenen speziellen Symbole und Abkürzungen.

Syntax Bezeichnungen

Symbol	Bedeutung
DIR	Verzeichnis-Attribut
<i>n</i>	steht für eine Zahl
<i>o</i>	steht für eine Option oder Optionsliste
RO	Read-Only
RW	Read-Write
<i>s</i>	steht für eine Zeichenkette
SYS	System-Attribut
{ }	Die Ausdrücke innerhalb der geschweiften Klammern sind optional. Das Kommando kann auch ohne einen optionalen Ausdruck eingegeben werden. Die optionalen Ausdrücke erweitern die Wirkung des Kommandos.
[]	Ausdrücke in eckigen Klammern stellen Optionen oder eine Optionsliste dar. Wird eine innerhalb der eckigen Klammern angegebene Option verwendet, dann müssen die eckigen Klammern mit eingegeben werden.

()	Ausdrücke in runden Klammern kennzeichnen einen Bereich von Optionen. Wird ein Bereich aus einer Optionsliste verwendet, dann muß dieser Bereich in runde Klammern eingeschlossen werden.
...	Der davorstehende Ausdruck kann beliebig oft wiederholt werden.
	Das Oder-Zeichen trennt alternative Ausdrücke in einer Kommandozeile. Es können beliebige oder alle alternativen Ausdrücke ausgewählt werden. Sich gegenseitig ausschließende Optionen werden in einer zusätzlichen Syntaxzeile gekennzeichnet oder im Text besonders hervorgehoben.
CTRL	Stellt die CTRL-Taste auf der Tastatur dar. (CTRL-Zeichen werden als ^ auf dem Bildschirm angezeigt)
<cr>	bezeichnet den Wagenrücklauf (Carriage Return)
*	Ersatzzeichen – ersetzt alles oder einen Teil des Dateinamens und/oder Dateityps.
?	Ersatzzeichen – ersetzt ein einzelnes Zeichen auf der gleichen Position in einem Dateinamen oder Dateityp.

Nachfolgend noch einige Beispiele der Syntax-Darstellungen. Das SCP-3.0-DIR-Kommando zeigt die Namen aller im Diskettenverzeichnis katalogisierten Dateien an und wahlweise andere Informationen über die Dateien.

Die Syntax des DIR-Kommandos mit Optionen zeigt, wie die Syntax-Darstellung der Kommandozeile zu interpretieren ist:

```
Syntax:      DIR {d:} | {dateibez} {[optionen]}
```

| | |

optional optional optional

Damit wird deutlich, daß der auf das Kommando-Schlüsselwort folgende Zusatz wahlweise angegeben werden kann. DIR ohne weitere Angaben ist schon ein gültiges Kommando, es können aber auch eine Dateibezeichnung oder die Optionen in der Kommandozeile angegeben werden. Deshalb sind

```
DIR
DIR dateibez
DIR d:
DIR [RO]
```

gültige Kommandos. Ferner können hinter der Laufwerks- oder Dateibezeichnung andere optionale Angaben aus der nachstehenden Liste der DIR-Optionen folgen:

```
RO
RW
DIR
SYS
```

Demzufolge ist

```
DIR d:dateibez [RO]
```

ein gültiges Kommando.

Das DIR-Kommando erlaubt auch die Angabe von Ersatzzeichen in der Dateibezeichnung. Mit dieser Syntaxbeschreibung können verschiedene gültige Kommandozeilen aufgebaut werden:

```
DIR
DIR X.PAS
DIR X.PAS [RO]
DIR X.PAS [SYS]
DIR *.PAS
DIR *.* [RW]
DIR X.* [DIR]
```

Das SCP-3.0-Kommando PIP ist das Dateikopierprogramm. PIP kann Informationen aus einer Datei auch auf einen Drucker oder Bildschirm kopieren. PIP kann weiter zwei oder mehrere Dateien zu einer größeren verbinden. Ferner kann PIP Dateien nach dem Kopieren auch umbenennen. Anhand von PIP soll ein weiteres Beispiel für die Interpretation der Kommandozeile in der Syntaxbeschreibung gegeben werden. PIP kopiert natürlich auch von Diskette zu Diskette.

Syntax: PIP ziel-dateibez=quell-dateibez[,dateibez . . .]

Es ist ziel-dateibez eine Angabe für eine Ziel-Dateibezeichnung oder ein peripheres Gerät (z. B. Drucker), welches Daten empfängt. Analog dazu ist quell-dateibez ein Name für die Quell-Dateibezeichnung oder ein peripheres Gerät (z. B. Tastatur), welches Daten sendet. PIP erlaubt die Angabe von Ersatzzeichen im Dateinamen und Dateityp. Aus dieser Syntax lassen sich eine ganze Reihe gültiger Kommandozeilen ableiten.

Einige Beispiele dazu sind:

```
PIP NEWFILE.DAT=OLDFILE.DAT
PIP B:=A:THISFILE.DAT
PIP B:X.BAS=Y.BAS, Z.BAS
PIP X.BAS=A.BAS, B.BAS, C.BAS
PIP B:=A:*.BAK
PIP B:=A:*.*
```

An diese allgemeinen Erläuterungen zum Lesen der Syntaxbeschreibungen schließen sich die Beschreibungen der einzelnen Kommandos an. Um den Umgang mit diesen Beschreibungen zu erleichtern, wurden sie alphabetisch nach den Kommandos geordnet.

6. Kommandos des SCP 3.0

In diesem Abschnitt werden die im Betriebssystem enthaltenen Kommandos und Programme beschrieben. Die Beschreibung erfolgt in alphabetischer Reihenfolge der Kommandos. Jedem Kommando sind eine kurze Erläuterung und Beispiele angefügt.

Das COPYSYS-Kommando

Syntax:

```
COPYSYS ds: dd: [optionen]
```

Erklärung:

COPYSYS kopiert das Betriebssystem SCP 3.0 von einer Systemdiskette im mit ds bezeichneten Laufwerk auf eine andere Diskette im mit dd bezeichneten Laufwerk.

Mit dem Kommando COPYSYS ohne Optionen werden nur die Systemspuren auf die neue Diskette kopiert. Um die neue Diskette als eine SCP-3.0-Systemdiskette verwenden zu können, müssen zusätzlich die Systemdateien SCP3.SYS und CCP.COM kopiert werden. Durch die Angabe der Option SYS wird das zusätzlich zum Kopieren der Systemspuren ausgeführt. Andere Dateien sind mit dem PIP-Kommando zu kopieren.

Die im Kommando möglichen Optionen erlauben die Änderung der Laufwerktypen für die Laufwerke A, B, C und D.

COPYSYS Optionen

Option	Funktion
SYS	Zusätzliches Kopieren der Dateien SCP3.SYS und CCP.COM
d=lwtyp	Zuordnen eines Laufwerktyps zum angegebenen Laufwerk für die Kopie des Systems. Es gilt: d ist die Bezeichnung eines der Laufwerke A, B, C oder D. Dabei wird kein Doppelpunkt hinter die Laufwerksbezeichnung gesetzt. lwtyp ist eine der gültigen Bezeichnungen für die Laufwerktypen. Die Bezeichnung muß ohne Leerstellen angegeben werden. Folgende Bezeichnungen können derzeit verwendet werden: MFS1.6 MFS1.4 MFS1.2 MF3200 MF6400

Beispiele:

```
A>COPYSYS b: d: [c=mf6400, d=mf6400,sys]
```

```
SCP3 COPYSYS R-BWS V 1/0
```

```
Reading system from disk!
```

```
Drive types are:
```

```
Drive A: MFS1.6
```

```
Drive B: MFS1.6
```

```
Drive C: MF6400
```

```
Drive D: MF6400
```

```
Writing system to disk!
```

```
End of copy system tracks.
```

```
Writing SCP3.SYS and CCP.COM to disk!
```

```
Function complete.
```

Das voranstehende Beispiel kopiert zunächst die Systemspuren von der Diskette im Laufwerk B auf die Diskette im Laufwerk D. Anschließend werden die Dateien SCP3.SYS und CCP.COM von der Diskette im Laufwerk B auf die Diskette im Laufwerk D kopiert.

```
A>COPYSYS C:
```

```
SCP3 COPYSYS R-BWS V 1/0
```

```
Reading system from disk!
```

Drive types are:
 Drive A: MFS 1.6
 Drive B: MFS 1.6
 Drive C: MF 3200
 Drive D: MF 3200

Mit dem Kommando in diesem Beispiel ist es möglich, sich über die eingestellten Laufwerktypen des auf der Diskette im Laufwerk C vorhandenen Systems zu informieren. Es werden weder Systemspuren noch die Systemdateien SCP3.SYS und CCP.COM kopiert.

```
A>COPYSYS A: B:
SCP3 COPYSYS R-BWS V 1/0
Reading system from disk!
Drive types are:
Drive A: MFS 1.6
Drive B: MFS 1.6
Drive C: MF 6400
Drive D: MF 6400
Writing system to disk!
End of copy system tracks.
```

Mit diesem Kommando werden lediglich die Systemspuren von der Diskette im Laufwerk A auf die Systemspuren der Diskette im Laufwerk B kopiert.

Wird nach dem Bereitschaftszeichen A> nur COPYSYS eingegeben, dann werden lediglich die Syntaxbeschreibung für dieses Kommando sowie ein Beispiel ausgegeben.

Das DATE-Kommando

Syntax:
 DATE {CONTINUOUS}
 DATE {zeitangabe}
 DATE SET

Erklärung:

DATE ist ein transientes Kommando zur Anzeige und zum Setzen von Datum und Uhrzeit. Beim Starten des Betriebssystems wird das Erstellungsdatum des Betriebssystems eingestellt. Mit DATE können dann das Datum und die Uhrzeit aktualisiert werden.

Anzeige der eingestellten Werte von Datum und Uhrzeit

Syntax:
 DATE {CONTINUOUS}

Erklärung:

Diese Form des DATE-Kommandos zeigt die eingestellten Werte von Datum und Uhrzeit an. Die Option CONTINUOUS erlaubt eine ständige Aktualisierung der Anzeige von Datum und Uhrzeit. Die Option CONTINUOUS kann auch abgekürzt mit C angegeben werden.

Beispiele:

```
A>DATE
A>DATE C
```

Im ersten Fall werden die momentanen Werte für Datum und Uhrzeit ausgegeben. Die Anzeige könnte zum Beispiel lauten:

Tue 02/11/86 09:35:16

Im zweiten Beispiel werden Datum und Uhrzeit bis zum Betätigen einer Taste fortlaufend angezeigt.

Setzen von Datum und Uhrzeit

Syntax:

DATE {zeitangabe}

DATE SET

Erklärung:

Die erste Form ermöglicht dem Nutzer sowohl das Tagesdatum als auch die Uhrzeit im Kommando direkt anzugeben. Datum und Uhrzeit müssen in folgender Form angegeben werden:

MM/DD/YY HH:MM:SS

wobei

MM	den Monat,
DD	den Tag,
YY	die letzten beiden Ziffern des Jahres
HH	die Stunden,
MM	die Minuten und
SS	die Sekunden angeben.

Die eingegebenen Werte für Datum und Uhrzeit werden vom System auf Gültigkeit geprüft und der zum Datum gehörende Wochentag bestimmt.

Die zweite Form fordert die Eingabe von Datum und Uhrzeit einzeln ab. Um das aktuelle Datum oder die aktuelle Uhrzeit beizubehalten, ist die Eingabe mit <cr> zu übergehen.

Beispiele:

A>DATE 03/09/86 10:30:0

Das System meldet sich darauf mit:

Press any key to set time

Es ist eine beliebige Taste zu betätigen, wenn die angegebene Zeit erreicht wurde. DATE initialisiert in diesem Moment die Zeit und zeigt Datum und Uhrzeit an:

Sun 03/09/86 10:30:00

A>DATE SET

Das System meldet sich mit:

Enter today's date (MM/DD/YY):

Es ist das Tagesdatum einzugeben oder die Eingabe mit <ET> zu übergehen. Das System meldet dann:

Enter the time (HH:MM:SS):

Auch hier kann die Eingabe der Uhrzeit mit <ET> übergangen werden. Vom System kommt dann die Ausschrift:

Press any key to set time

Damit hat der Nutzer die Möglichkeit, die Uhr exakt einzustellen.

Das DEVICE-Kommando**Syntax:**

```

DEVICE {NAMES | VALUES | phys. Gerät | log. Gerät }
DEVICE log. Gerät=phys.Gerät {option}
      { ,phys.Gerät {option} , . . . }
DEVICE log.Gerät=NULL
DEVICE phys.Gerät {option}
DEVICE CONSOLE [PAGE | COLUMNS=Spalten | LINES=Zeilen]

```

Erklärung:

Das transiente Kommando DEVICE zeigt die aktuelle Zuordnung der logischen und die Namen der physischen Geräte an. Mit DEVICE können die logischen Geräte des Systems den an den Computer gekoppelten physischen Geräten zugewiesen werden. Das DEVICE-Kommando definiert die Übertragungsgeschwindigkeit und ermöglicht die Anzeige oder das Definieren der aktuellen Bildschirmgröße.

Das System unterstützt folgende fünf logischen Geräte

```

CONIN:
CONOUT:
AUXIN:
AUXOUT:
LST:

```

Bekannt sind diese logischen Geräte auch durch folgende Bezeichnungen:

```

CON:           (für CONIN: und CONOUT:)
CONSOLE:       (für CONIN: und CONOUT:)
KEYBOARD       (für CONIN:)
AUX:           (für AUXIN: und AUXOUT:)
AUXILIARY:     (für AUXIN: und AUXOUT:)
PRINTER        (für LST:)

```

Anzeige der Geräteeigenschaften und Zuordnungen**Syntax:**

```

DEVICE {NAMES | VALUES | phys.Gerät | log.Gerät}

```

Erklärung:

Diese Form des DEVICE-Kommandos dient zur Anzeige der Namen, Werte und Eigenschaften der physischen Geräte sowie der aktuellen Zuordnung der logischen Geräte im System.

Beispiele:

```

A>DEVICE

```

Mit diesem Kommando werden die physischen Geräte und die Zuordnung der logischen Geräte im System angezeigt. Eine mögliche Anzeige wäre die folgende:

Physical Devices:

```

I=Input, O=Output, S=Serial, X=Xon-Xoff
KEY NONEI DISPLAY NONE O PRN 9600 OS
V24 9600 OS CHAN_A 6900 OSX CHAN_B 9600 OSX
RESERV 1200 IOS

```

Current Assignments:
 CONIN: = KEY
 CONOUT: = DISPLY
 AUXIN: = V24
 AUXOUT: = V24
 LST: = Null Device

Enter new assignment or hit RETURN

Das System wartet auf die Eingabe einer gültigen Gerätezuordnung (wie in einem späteren Abschnitt beschrieben). Soll keine der Gerätezuordnungen geändert werden, dann ist die RETURN-Taste zu drücken.

A>DEVICE NAMES

Dieses Kommando zeigt die Geräte mit ihren Eigenschaften an.

A>DEVICE VALUES

Damit wird die aktuelle logische Gerätezuordnung angezeigt.

A>DEVICE DISPLY

Das Kommando zeigt die Eigenschaften des physischen Gerätes DISPLY an.

A>DEVICE CON

Mit diesem Kommando wird die Zuordnung des logischen Gerätes CON: angezeigt.

Zuordnen eines logischen Gerätes

Syntax:

DEVICE log.Gerät=phys.Gerät {option}
 {,phys.Gerät {option}, . . .}

DEVICE log.Gerät=NULL

Erklärung:

Die erste Form des Kommandos ordnet einem logischen Gerät ein oder mehrere physische Geräte zu. Die zweite Form trennt das logische Gerät vom physischen.

Folgende Optionen sind möglich:

Option	Bedeutung
XON	Die Datenübertragung des Gerätes kann mit ^S unterbrochen und mit ^Q wieder gestartet werden.
NOXON	Die Tasten ^S und ^Q sind wirkungslos
baudrate	ist die Geschwindigkeit des Gerätes. Das System erlaubt folgende Baudraten:
	50 75 110 150
	300 600 1200 2400
	4800 7200 9600

Beispiele:

A>DEVICE COUNOUT:=PRN,DISPLY

A>DEVICE AUXIN:=V24 [XON,9600]

A>DEVICE LST:=NULL

Das erste Beispiel weist der Konsolenausgabe den Drucker PRN und den Bildschirm DISPLY zu.

Das zweite Beispiel weist dem logischen Gerät AUXIN: das physische Gerät V24 unter Verwendung von XON/XOFF zu und setzt die Übertragungsgeschwindigkeit für das Gerät auf 9600. Das dritte Beispiel hängt das logische Ausgabegerät LST: ab.

Eintragen von Eigenschaften eines physischen Gerätes

Syntax: DEVICE phys.Gerät {option}

Erklärung:

Diese Form des DEVICE-Kommandos legt die Eigenschaften eines physischen Gerätes entsprechend der im Kommando angegebenen Werte fest.

Beispiel:

A>DEVICE PRN[XON,9600]

Mit diesem Kommando werden für das physische Gerät PRN die Benutzung von XON/XOFF und eine Übertragungsgeschwindigkeit von 9600 Baud vereinbart.

Anzeigen und Definieren der aktuellen Bildschirmgröße

Syntax:

DEVICE CONSOLE [PAGE | COLUMNS=Spalten |
LINES=Zeilen]

Erklärung:

Mit dieser Form des DEVICE-Kommandos wird die aktuelle Bildschirmgröße angezeigt oder definiert. (Nur logisch für die BDOS-Funktionen und Dienstprogramme! Es erfolgt keine physische Änderung der Betriebssystem-Treiber.)

Beispiele:

A>DEVICE CONSOLE [PAGE]

A>DEVICE CONSOLE [COLUMNS=40,LINES=16]

Im ersten Beispiel wird die Bildschirmgröße in Anzahl Spalten und Anzahl Zeilen ausgegeben.

Mit dem zweiten Beispiel wird die Bildschirmgröße auf 40 Spalten und 16 Zeilen festgelegt.

Das DIR-Kommando

Syntax:

DIR {d:}

DIR {dateibez}

DIRSYS {d:}

DIRSYS {dateibez}

DIR {d:} {option}

DIR {dateibez} {dateibez} . . . {option}

Erklärung:

Das Kommando DIR zeigt die Namen aller Dateien und die zugehörigen Eigenschaften an. DIR und DIRSYS sind residente Routinen; DIR mit Optionen ist ein transientes Kommando.

Verzeichnis anzeigen

Syntax:

DIR {d:}

DIR {dateibez}

DIRSYS {d:}
DIRSYS {dateibez}

Erklärung:

Die Kommandos DIR und DIRSYS zeigen die Namen aller im Verzeichnis der zugewiesenen Diskette katalogisierten Dateien an. Das Kommando DIR listet alle Dateinamen des aktuellen Nutzerbereiches auf, die das Verzeichnis – (DIR) – Attribut besitzen. Ersatzzeichen (* und ?) in der Dateibezeichnung sind erlaubt. Das Kommando DIRSYS kann mit DIRS abgekürzt werden.

Das Kommando DIRSYS zeigt die Namen aller Dateien im aktuellen Nutzerbereich an, die das System-Attribut (SYS) besitzen. Obwohl die im Nutzerbereich 0 abgespeicherten System-Dateien von jedem anderen Nutzerbereich des gleichen Laufwerkes gelesen werden können, zeigt DIRSYS nur die Dateien des Nutzerbereiches 0 an, wenn der aktuelle Nutzerbereich auch 0 ist. Ersatzzeichen (* und ?) sind im DIRSYS-Kommando ebenfalls erlaubt. Werden Laufwerk und Dateibezeichnung weggelassen, dann zeigt das DIR-Kommando nur die Dateinamen mit dem DIR-Attribut auf dem aktuellen Laufwerk des aktuellen Nutzerbereiches an. Entsprechend zeigt das DIRSYS-Kommando alle Dateien mit dem SYS-Attribut an.

Wird nur das Laufwerk angegeben, dann zeigt das DIR-Kommando alle Dateinamen mit dem DIR-Attribut im aktuellen Nutzerbereich der Diskette im angegebenen Laufwerk an. DIRSYS zeigt entsprechend die SYS-Dateien an.

Enthält die Dateibezeichnung ein Ersatzzeichen, dann werden auf dem Bildschirm alle Dateinamen angezeigt, die dieser Dateibezeichnung genügen.

Gibt es keine Dateinamen, die der angegebenen Dateibezeichnung genügen oder gibt es im Verzeichnis der Diskette des angegebenen Laufwerkes keine katalogisierten Dateien, dann geben die Kommandos DIR bzw. DIRSYS folgende Meldung aus:

No File

Gibt es System-Dateien (SYS), die der angegebenen Dateibezeichnung genügen, dann meldet das DIR-Kommando:

SYSTEM FILE(S) EXIST

Genügen Nicht-System-Dateien (DIR) der Dateibezeichnung, dann meldet DIRSYS:

NON-SYSTEM FILE(S) EXIST

Wenn der Bildschirm mit Angaben gefüllt wurde, unterbricht das DIR-Kommando seine Anzeige bis zum Betätigen einer Taste.

Beachte: Mit dem DEVICE-Kommando kann die Anzahl der mit DIR oder DIRSYS angezeigten Spalten verändert werden.

Beispiele:

A>DIR

Alle in Nutzerbereich 0 des Standardlaufwerkes A katalogisierten DIR-Dateien werden angezeigt.

A>DIR B:

Alle DIR-Dateien im Nutzerbereich 0 auf Laufwerk B werden angezeigt.

A>DIR B:X.BAS

Zeigt den Dateinamen X.BAS an, wenn die Datei X.BAS im Laufwerk B existiert.

4A>DIR *.BAS

Es werden alle DIR-Dateien mit dem Dateityp BAS im Nutzerbereich 4 des Laufwerkes A angezeigt.

B>DIR A:X*.C?D

Angezeigt werden alle DIR-Dateien im Nutzerbereich 0 auf Laufwerk A, deren Dateiname mit dem Buchstaben X beginnt und deren 3 Zeichen Dateityp als ersten Buchstaben C und als letzten D enthalten.

A>DIRSYS

Alle Dateien mit dem Attribut SYS im Nutzerbereich 0 auf Laufwerk A werden angezeigt.

3A>DIRS *.COM

Mit dieser abgekürzten Form des DIRSYS-Kommandos werden alle SYS-Dateien mit dem Dateityp COM im Nutzerbereich 3 des Standardlaufwerkes A angezeigt.

Verzeichnisanzeige mit Optionen

Syntax:

DIR {d:} {option}

DIR {dateibez} {dateibez} . . . {option}

Erklärung:

Das DIR-Kommando mit Optionen ist eine erweiterte Variante des DIR-Kommandos. Das DIR-Kommando zeigt SCP-3.0-Dateien in verschiedenen Variationen an. DIR kann Dateien auf einem oder allen Laufwerken in einem oder allen Nutzerbereichen suchen.

Die Optionen können hinter dem Kommando an beliebiger Stelle angegeben werden. Sie modifizieren die gesamte Kommandozeile. Es darf nur eine Liste von Optionen angegeben werden.

Die Optionen müssen in eckige Klammern eingeschlossen werden. Sie können einzeln angegeben werden oder zusammen, getrennt voneinander durch Komma oder Leerzeichen. Die Optionen können mit nur einem oder zwei Buchstaben abgekürzt werden, wenn diese Abkürzung die Optionen noch eindeutig identifiziert.

Überschreitet eine Verzeichnisliste die Bildschirmgröße, dann wird nach Füllen des Bildschirms die Anzeige unterbrochen. Mit Betätigen einer beliebigen Taste wird die Anzeige fortgesetzt.

Option	Funktion
ATT	zeigt die vom Nutzer definierbaren Dateiattribute F1, F2, F3 und F4 an.
DATE	zeigt Dateien mit Tag- und Uhrzeit-Eintragungen an. Sind Datum und Uhrzeit nicht vorhanden, dann gibt DIR folgende Meldung aus: Date and Time Stamping Inactive
DIR	zeigt nur Dateien mit DIR-Attribut an.
DRIVE=ALL	zeigt Dateien auf allen zugewiesenen Laufwerken an. Für DRIVE kann in allen Optionen auch DISK geschrieben werden.
DRIVE=(A,B,C,...P)	zeigt die Dateien der angegebenen Laufwerke an.

Option	Funktion
DRIVE=d	zeigt die Dateien auf dem durch d spezifizierten Laufwerk an.
EXCLUDE	zeigt die Dateien des Standardlaufwerkes und Nutzerbereiches an, die nicht der in der Kommandozeile angegebenen Datei- bezeichnung genügen.
FF	sendet zu Beginn ein Form-Feed an den Drucker aus, wenn der Drucker vorher durch CTRL-P aktiviert wurde. Wurde zusätzlich LENGTH=n angegeben, dann wird vom DIR-Kommando nach jeweils n Zeilen ein Form-Feed ausgegeben. Anderenfalls setzt die FF-Option die standardmäßige seitenweise Bildschirmanzeige außer Kraft.
FULL	zeigt den Namen und die Größe der Datei an. Die Größe wird als Speicherplatz in Kilobyte und als Anzahl zugeordneter 128-Byte Sätze angezeigt. FULL zeigt ebenfalls die Dateiattribute an (s. SET-Kommando für die Erklärung der Dateiattribute). Gibt es auf dem Laufwerk einen Verzeichniskennsatz, dann zeigt DIR den Paßwort-Schutz und die Zeiteintragen an. Die Anzeige ist alphabetisch geordnet. Werden bei DIR Optionen angegeben, dann ist FULL Standard für die Bildschirmanzeige.
LENGTH=n	Es werden n Zeilen ausgegeben, bevor eine Tabellenüberschrift eingefügt wird. Dabei muß n im Bereich zwischen 5 und 65536 liegen. Die Standardlänge ist ein mit Informationen voll beschriebener Bildschirm.
MESSAGE	zeigt die Namen der spezifizierten Laufwerke und die Nummern der Nutzerbereiche an, die gegenwärtig im Zugriff sind. Sind unter dem angegebenen Platz keine Dateien vorhanden, dann gibt DIR die Meldung aus, daß keine Dateien gefunden wurden (No File).
NOPAGE	Die Informationen werden ununterbrochen über den Bildschirm gerollt. Es wird nicht auf das Betätigen einer Taste gewartet, um die Bildschirmanzeige fortzusetzen.
NOSORT	zeigt die Dateien in der Reihenfolge an, in der sie auf der Diskette katalogisiert sind. Ist diese Option nicht angegeben, dann werden die Dateien in alphabetischer Reihenfolge angezeigt.
RO	zeigt nur die Dateien mit READ-ONLY-Attribut an.
RW	zeigt nur die Dateien mit dem READ-WRITE-Attribut an.
SIZE	zeigt den Dateinamen und die Größe der Datei in Kilobyte an.
SYS	zeigt nur Dateien mit dem SYS-Attribut an.
USER=ALL	zeigt alle Dateien aller Nutzerbereiche des Standardlaufwerkes an.
USER=n	zeigt die Dateien des durch n angegebenen Nutzerbereiches an.
USER=(0,1,...15)	zeigt die Dateien der angegebenen Nutzerbereiche an.

Beispiele:

A>DIR B: [FULL]

A>DIR B: [SIZE]

Durch die Angabe der Option [FULL] wird mit dem ersten Beispiel folgende mögliche Anzeige erzeugt:

Directory For Drive B: User 0

Name	Bytes	Recs	Attributes	Prot	Update	Access
DATE COM	4k	22	SYS RW	NONE	01/06/86 9:46	02/08/86 15:30
DEVICE COM	8k	58	SYS RW	NONE	12/18/85 10:09	01/10/86 8:15
DUMP COM	2k	8	SYS RW	NONE	12/18/85 10:20	02/15/86 11:33
PROFILE SUB	2k	1	SYS RW	NONE	12/18/85 11:03	02/09/86 12:15

Total Bytes = 16k Total Records = 89 Files Found = 4

Total 1k Blocks = 13 Used/Max Dir Entries for Drive B: 6/128

Eine mögliche Anzeige für das zweite Beispiel unter Verwendung der Option [SIZE] ist folgende:

Directory For Drive B: User 0

B: DATE COM 4k : DEVICE COM 8k DUMP COM 2k
 B: PROFILE SUB 2k

Total Bytes = 16k Total Records = 89 Files Found = 4

Total 1k Blocks = 13 Used/Max Dir Entries for Drive B: 6/128

Sowohl der ersten als auch der zweiten Form der Anzeige folgen zwei Zeilen zusätzlicher Informationen. In der ersten Zeile stehen die Gesamtanzahl Bytes in Kilobyte, die Anzahl Datensätze insgesamt und die Anzahl aller Dateien des Nutzerbereiches auf diesem Laufwerk. In der zweiten Zeile wird die zur Speicherung der aufgeführten Dateien benötigte Anzahl 1K-Blöcke angegeben. Die Anzahl von 1K-Blöcken ist der Speicherbereich, der benötigt wird, um die Dateien auf einer Diskette einfacher Dichte oder auf einer Diskette abzuspeichern, die eine Blockgröße von 1k Byte besitzt. Die zweite Zeile gibt Auskunft über die Anzahl benutzter und vorhandener Verzeichniseinträge. Diese Angaben werden unterdrückt, wenn nur eine Datei gefunden wurde.

A>DIR [DRIVE=C,FF]

Bevor die Anzeige der Dateien vom Laufwerk C vorgenommen wird, sendet DIR ein Form-Feed an den Drucker aus.

A>DIR D: [RW,SYS]

Mit diesem Kommando werden alle Dateien mit den Attributen RW und SYS des Laufwerkes D angezeigt.

A>DIR C: [USER = ALL]

Zeigt alle Dateien aller Nutzerbereiche (0-15) im Laufwerk C an.

A>DIR [USER=2]

Zeigt alle Dateien des Nutzerbereiches 2 auf dem Standardlaufwerk an.

A>DIR C: [USER=(3,4,10)]

Dieses Kommando zeigt alle Dateien der Nutzerbereiche 3, 4 und 10 auf Laufwerk C an.

A>DIR [DRIVE=ALL]

Zeigt alle Dateien im Nutzerbereich 0 aller Laufwerke in der zum Durchsuchen der Laufwerke festgelegten Reihenfolge (vgl. dazu SETDEF-Kommando) an.

4A>DIR [DRIVE=C]

Zeigt alle Dateien des Nutzerbereiches 4 auf Laufwerk C an.

A>DIR [DRIVE=(B,D)]

Zeigt alle Dateien des Nutzerbereiches 0 auf den Laufwerken B und D an.

A>DIR [EXCLUDE] *.COM

Mit diesem Kommando werden alle Dateien vom Nutzerbereich 0 des Standardlaufwerkes angezeigt, die nicht den Dateityp COM besitzen.

A>DIR [user=all, drive=all, sys] *.PLI *.COM *.ASM

Mit diesem Kommando werden alle System-Dateien mit dem Dateityp PLI, COM und ASM aller Nutzerbereiche aller Laufwerke angezeigt.

A>DIR X.SUB [MESSAGE, USER=ALL, DRIVE=ALL]

Auf allen Laufwerken wird in allen Nutzerbereichen die Datei X.SUB gesucht. Während dieses Suchlaufes zeigt DIR die Laufwerke und Nutzerbereiche an.

A>DIR [drive=all, user=all] TESTFILE.BOB

Das Kommando zeigt den Dateinamen TESTFILE.BOB an, wenn er in einem Nutzerbereich der zugewiesenen Laufwerke vorkommt.

A>DIR [size,rw] D:

Es werden alle READ/WRITE-Dateien vom Laufwerk D mit ihrer Größe in Kilobyte angezeigt. Beachte, daß D: gleichbedeutend ist mit der Angabe D: *.*.

Das DUMP-Kommando

Syntax:

DUMP Dateibezeichnung

Erklärung:

DUMP zeigt den Inhalt einer Datei im hexadezimalen und ASCII-Format an.

Beispiel:

A>DUMP ABC.TEX

Die Anzeige könnte wie folgt aussehen:

DUMP-Version X.X

```
0000: 41 42 43 0D 0A 44 45 46 0D 0A 47 48 49 0D 0A 1A   ABC..DEF..GHI...
0010: 1A   .....
```

DAS ERASE-Kommando

Syntax:

ERASE {dateibez} {[CONFIRM]}

Erklärung:

Das ERASE-Kommando entfernt eine oder mehrere Dateien aus dem Verzeichnis einer Diskette im aktuellen Nutzerbereich. In der Dateibezeichnung können Ersatzzeichen ent-

halten sein. Verzeichnisplatz und Datenbereich werden zur späteren Verwendung durch eine andere Datei automatisch freigegeben. Das Kommando ERASE kann auf ERA abgekürzt werden.

Das ERASE-Kommando ist mit Vorsicht anzuwenden, denn es werden alle die Dateien des aktuellen Nutzerbereiches aus dem Verzeichnis entfernt, die der angegebenen Dateibezeichnung genügen.

Kommandozeilen in der Form

ERASE {d:} ersatzzeichen-dateibezeichnung

verlangen eine Bestätigung, denn es wird damit nicht nur eine Datei, sondern eine ganze Gruppe von Dateien gelöscht. Das System reagiert mit der Meldung:

ERASE {d:} ersatzzeichen-dateibezeichnung (Y/N)?

Es ist mit Y zu antworten, wenn alle entsprechenden Dateien entfernt werden sollen und mit N, wenn das Löschen der Dateien vermieden werden soll.

Gibt es keine Dateien, die der Dateibezeichnung genügen, wird folgende Meldung ausgegeben:

No File

Die Option CONFIRM bewirkt eine Abfrage durch das System, bevor eine Datei gelöscht wird, die der angegebenen Dateibezeichnung genügt. CONFIRM kann mit C abgekürzt werden.

Wird die Option CONFIRM in Verbindung mit einer Ersatzzeichen-Dateibezeichnung verwendet, dann verlangt ERASE für jede Datei eine Bestätigung. Damit können wahlweise die Dateien gelöscht werden durch Y als Antwort auf die Bestätigungsmeldung oder erhalten bleiben mit N als Antwort.

Beispiele:

A>ERASE X.PAS

Entfernt die Datei X.PAS von der Diskette in Laufwerk A.

A>ERA *.PRN

Das System fragt nach der Bestätigung:

ERASE *.PRN (Y/N)?Y

Alle Dateien mit dem Dateityp .PRN werden von der Diskette im Laufwerk A entfernt.

ERA A:MY*. * [CONFIRM]

Jede Datei vom Laufwerk A, die mit MY beginnt, wird mit einem Fragezeichen zur Bestätigung angezeigt. Es ist Y einzugeben, um die angezeigte Datei zu löschen und N um sie zu erhalten.

A>ERA B:*. *

ERASE B:*. * (Y/N)?Y

Auf Laufwerk B werden alle Dateien von der Diskette entfernt.

DAS GENCOM-Kommando

Syntax:

GENCOM {COM-Dateibez} {RSX-Dateibez} . . .
 {[LOADER | NULL | SCB=(offset,value)]}

Erklärung:

Das Kommando GENCOM erzeugt eine spezielle COM-Datei mit angehängten RSX-Dateien. RSX-Dateien werden als residente Systemerweiterungen benutzt und im „System-

handbuch SCP 3.0 Anleitung für den Programmierer“ detailliert beschrieben. GENCOM legt einen speziellen Header am Anfang der generierten Datei ab, um dem System anzuzeigen, daß ein RSX-Laden erforderlich ist. Es kann auch ein Flag setzen, um den Programmlader aktiv zu halten.

Das GENCOM Kommando kann weiterhin eine zuvor mit GENCOM erstellte Datei in die ursprüngliche COM-Datei zurückführen ohne Header und RSX. GENCOM besitzt drei Optionen, die das Zufügen von RSX-Dateien unterstützen:

- * Die LOADER-Option setzt ein Flag, um den Programmlader aktiv zu halten. Diese Option wird nur dann benutzt, wenn keine RSX-Dateien an die COM-Datei angefügt wurden.
- * Die NULL-Option zeigt an, daß nur RSX-Dateien spezifiziert wurden. GENCOM legt eine leere COM-Datei für die RSX-Dateien an. Die erzeugte COM-Datei erhält als Dateinamen den Namen aus der ersten RSX-Dateibezeichnung.
- * Die Option SCB=(offset,value) erstellt den System-Control-Block von dem Programm unter Verwendung der angegebenen hexadezimalen Werte von (offset,value). Vollständige Informationen über die SCB-Option sind in der Dokumentation „Systemhandbuch SCP 3.0 Anleitung für den Programmierer“ nachzulesen.

Anhängen von RSX-Dateien an eine COM-Datei

Syntax:

```
GENCOM COM-dateibez RSX-dateibez . . .
      {[LOADER | SCB=(offset,value)]}
```

Erklärung:

Die oben angegebene Form des GENCOM-Kommandos erzeugt eine COM-Datei mit angehängenen RSX-Dateien. Es können maximal 15 RSX-Dateien angehängen werden. GENCOM erwartet als erste Dateibezeichnung die einer COM-Datei und die nachfolgenden Dateibezeichnungen als die von RSX-Dateien. Zu beachten ist, daß die ursprüngliche COM-Datei durch die neu erstellte COM-Datei ersetzt wird.

Beispiel:

```
A>GENCOM MYPROG PROG1 PROG2
```

Generiert die neue COM-Datei MYPROG.COM unter Einbeziehung der RSX-Module PROG1 und PROG2.

Generieren einer COM-Datei unter ausschließlicher Verwendung von RSX-Dateien

Syntax:

```
GENCOM RSX-dateibez {RSX-dateibez} . . .
      [NULL {SCB=(offset,value)}]
```

Erklärung:

Diese Form des GENCOM-Kommandos hängt RSX-Dateien an eine leere COM-Datei an. GENCOM erstellt eine COM-Datei mit dem Dateinamen der ersten RSX-Dateibezeichnung. Diese Form ermöglicht dem System RSX-Dateien direkt zu laden.

Beispiel:

```
A>GENCOM PROG1 PROG2 [NULL]
```

Voranstehendes Kommando erstellt die COM-Datei PROG1.COM mit den residenten Systemerweiterungen PROG1.RSX und PROG2.RSX.

Wiederherstellen der Original-COM-Datei aus einer Datei mit angehangenen RSX-Dateien

Syntax:

GENCOM dateiname

Erklärung:

GENCOM nimmt eine Datei, die schon einmal mit GENCOM bearbeitet wurde und überführt sie in ihr ursprüngliches COM-Dateiformat. Diese Form des Kommandos setzt den Dateityp COM voraus.

Beispiel:

A>GENCOM MYPROG

GENCOM nimmt MYPROG.COM, entfernt den Header und löscht alle zugehörigen RSX-Module, um die Datei in ihr Original-COM-Format zu überführen.

Aktualisieren (Zufügen oder Ersetzen) von RSX-Dateien

Syntax:

GENCOM COM-dateibez RSX-dateibez . . .
 {[LOADER | SCB=(offset,value)]}

Erklärung:

Diese Form des GENCOM-Kommandos fügt RSX-Dateien hinzu und/oder ersetzt RSX-Dateien in einer bereits mit GENCOM bearbeiteten Datei.

GENCOM untersucht die Liste der RSX-Dateien. Sind diese neu, dann werden sie zu der bereits mit GENCOM erzeugten Datei hinzugefügt. Existieren sie schon, dann ersetzt GENCOM die existierenden RSX-Dateien durch die neuen.

Beispiel:

A>GENCOM MYPROG PROG1 PROG2

GENCOM überprüft die bereits mit GENCOM bearbeitete Datei MYPROG.COM daraufhin, ob PROG1.RSX und PROG2.RSX bereits einbezogen wurden in den Modul. Wurde einer bereits einbezogen, dann ersetzt GENCOM ihn durch den neuen RSX-Modul. Anderenfalls hängt GENCOM die angegebenen RSX-Dateien an die COM-Datei an.

Zufügen eines Header-Satzes

Syntax:

GENCOM dateiname [SCB=(offset,value), . . . | loader]

Erklärung:

Durch die oben angegebene Kommandozeile wird ein GENCOM-Header-Satz mit gesetztem SCB oder Lader-Flag an eine Datei vom Typ COM angehangen, die keine RSX-Dateien enthält.

Beispiele:

A>GENCOM FILETWO [LOADER]

Es wird ein 256 Byte-Header-Satz an der Datei FILETWO.COM angefügt und das Lader-Flag im Header-Satz gesetzt.

A>GENCOM FILEFOUR [SCB=(1,1)]

Im obigen Beispiel wird der Programmloader veranlaßt, das Byte 1 des SCB auf 1 zu setzen, wenn FILEFOUR.COM geladen wird.

Das GET-Kommando

Syntax:

```
GET {CONSOLE INPUT FROM} FILE dateibez
      {{{ECHO | NO ECHO} | SYSTEM}}
GET {CONSOLE INPUT FROM} CONSOLE
```

Erklärung:

GET legt die Konsoleneingabe auf eine Datei um. Die Datei kann SCP-3.0-Systemkommandos und/oder Eingaben für ein Nutzerprogramm enthalten. Wird die Option SYSTEM angegeben, dann nimmt GET das nächste Systemkommando von der Datei.

Die Konsoleneingabe wird solange von einer Datei genommen, bis das Programm beendet ist. Ist die Datei schon leer, bevor die Programmeingaben beendet sind, dann erwartet das Programm die weiteren Eingaben von der Konsole. Ist das Programm beendet, bevor alle seine Eingaben erschöpft sind, dann wird das System auf die Konsole für Konsoleneingaben zurückgesetzt.

Mit der Option SYSTEM geht das System unmittelbar zur Konsoleneingabe von der angegebenen Datei über. Wird die Option SYSTEM weggelassen, dann kann ein Systemkommando eingegeben werden, mit dem ein Nutzerprogramm aufgerufen wird, das seine Konsoleneingaben von der im GET-Kommando angegebenen Datei liest. Das System kehrt für weitere Eingaben zur Konsole zurück, wenn das Dateiende erreicht ist. Das System kann auch auf die Konsoleneingabe über Konsole mit dem Kommando GET CONSOLE INPUT FROM CONSOLE als eine Kommandozeile in der Eingabedatei zurückgesetzt werden.

Konsoleneingabe von einer Datei

Syntax:

```
GET {CONSOLE INPUT FROM} FILE dateibez {[option]}
```

Erklärung:

Diese Form des GET-Kommandos teilt dem System mit, daß nachfolgende Konsoleneingaben aus einer Datei erfolgen. Die nachfolgende Tabelle gibt die GET-Optionen an, die im folgenden Format genutzt werden:

```
[ {ECHO | NO ECHO } | SYSTEM]
```

Option	Bedeutung
ECHO	gibt an, daß die Eingabe auf der Console wiederholt wird. Diese Option ist Standard.
NO ECHO	gibt an, daß die Dateieingabe nicht auf dem Bildschirm wiederholt wird. Die Programmausgaben und Systemanforderungen werden durch diese Option nicht beeinflußt und weiterhin auf dem Bildschirm wiederholt.
SYSTEM	gibt an, daß alle Systemeingaben unmittelbar von der in der Kommandozeile angegebenen Diskettendatei genommen werden. GET nimmt die System- und Programmeingaben solange von der Datei, bis diese erschöpft ist oder bis GET ein GET-Konsole-Kommando von der Datei liest.

Beispiele:

```
A>GET FILE XINPUT
```

```
A>MYPROG
```

Das System aktiviert die GET-Routine. Da die Option SYSTEM nicht angegeben wurde, liest das System die nächste Eingabezeile von der Console und führt MYPROG aus. Verlangt das Programm MYPROG Konsoleneingaben, dann werden diese von der Datei XINPUT genommen. Ist das Programm MYPROG beendet, dann wird das System auf Konsoleneingaben über Konsole zurückgesetzt.

A>GET FILE XIN2 [SYSTEM]

Das System nimmt alle Konsoleneingaben von der Datei XIN2, da die Option SYSTEM angegeben wurde. Das System erwartet wieder Eingaben über die Konsole, wenn es das Ende der Datei XIN2 erreicht hat. XIN2 kann das System auf die Konsole zurücksetzen, wenn es ein GET-CONSOLE-Kommando enthält.

Beenden der Konsoleneingaben aus einer Datei

Syntax:

GET {CONSOLE INPUT FROM} CONSOLE

Erklärung:

Dem System wird mitgeteilt, daß die Konsoleneingabe über die Konsole erfolgt.

Beispiel:

A>GET CONSOLE

Teilt dem System mit, daß die Konsoleneingabe über Konsole erfolgt. Dieses Kommando kann in einer Datei verwendet werden (wie weiter oben angegeben in einem GET FILE Kommando), die bereits durch das System als Konsoleneingabe gelesen wird. Es wird dazu genutzt, die Konsoleneingaben wieder auf die Konsole zu legen noch bevor das Dateiende erreicht wird.

Das HELP-Kommando

Syntax:

HELP {topic} {subtopic1 . . . subtopic8} {[NOPAGE]}

HELP {EXTRACT}

HELP {CREATE}

Erklärung:

Das HELP Kommando ist ein transientes Programm, welches zusammengefaßte Informationen für alle in diesem Handbuch beschriebenen SCP-3.0-Kommandos liefert. In der Vertriebsversion des SCP-3.0-Systems liefert HELP allgemeine Informationen über ein Kommando als Begriff (topic) und detaillierte Informationen über ein Kommando als Unterbegriff (subtopic). HELP ohne weitere Angaben zeigt eine Liste verfügbarer Begriffe an. Die Angabe HELP topic zeigt die Informationen zu diesem Begriff und alle verfügbaren Unterbegriffe an. HELP mit topic und subtopic zeigt die Informationen über diesen speziellen Unterbegriff an.

Nachdem HELP die Informationen über den angegebenen Begriff angezeigt hat, wird das spezielle Bereitschaftszeichen HELP> auf dem Bildschirm ausgegeben. Auf die Unterbegriffe kann zugegriffen werden, wenn dem subtopic ein Punkt vorangestellt wird. Der Punkt bewirkt, daß nach dem subtopic auf dem untersten bekannten Niveau gesucht wird. Es kann damit fortgefahren werden, für zusätzliche Information Begriffe anzugeben oder durch einfaches Betätigen der RETURN Taste zum SCP-3.0-System zurückzukehren.

Die Namen von topic und subtopic können abgekürzt werden. Normalerweise reichen ein oder zwei Buchstaben aus, um die topics zu identifizieren.

Informationsanzeige

Syntax:

HELP topic {suptopic1 . . . subtopic8} {[NOPAGE | LIST]}

HELP .subtopic

Erklärung:

Die oben dargestellten Formen des HELP-Kommandos zeigen Informationen über die angegebenen Begriffe und Unterbegriffe an. Folgende zwei Optionen können mit dieser Form des HELP-Kommandos genutzt werden:

- * Die Option NOPAGE schaltet die seitenweise Ausgabe von n Zeilen ab, wobei n die durch den Nutzer oder das System festgelegte Anzahl Zeilen pro Seite darstellt. Die Anzeige kann mit Betätigen von CTRL-S angehalten und mit CTRL-Q fortgesetzt werden.
- * Die Option LIST wirkt wie NOPAGE, nur daß sie zusätzliche Zeilen zwischen den Überschriften aussondert. Diese Option ist in Verbindung mit CTRL-P zu benutzen, um die HELP-Informationen auf den Drucker auszugeben.

Beispiele:

A>HELP

Zeigt eine Liste aller Begriffe an, für die Informationen verfügbar sind.

A>HELP DATE

Es werden die allgemeinen Informationen über das DATE-Kommando sowie die verfügbaren Unterbegriffe angezeigt.

A>HELP DIR OPTIONS [N]

Dieses Kommando enthält den Unterbegriff Optionen. Es werden die Informationen zu den Optionen des DIR-Kommandos angezeigt. Die Anzeige erfolgt nicht seitenweise.

A>HELP PIP

Es werden die allgemeinen Informationen zum PIP-Programm angezeigt.

A>HELP PIP OPTIONEN

Mit dieser Form von HELP werden die Informationen zu den OPTIONEN von PIP angezeigt. Das obige Beispiel kann auch wie folgt eingegeben werden

A>HELP PIP

HELP>.OPTIONEN

Das Anfügen eigener Beschreibung an die Datei HELP.HLP

Syntax:

HELP [EXTRACT]

HELP [CREATE]

Erklärung:

SCP 3.0 wird mit zwei zusammengehörenden HELP-Dateien vertrieben: HELP.COM und HELP.HLP. Die Datei HELP.COM ist die Kommandodatei, die mit dem Text von HELP.HLP arbeitet und diesen auf dem Bildschirm anzeigt. Die Datei HELP.HLP ist eine Textdatei, an die auch entsprechend aufbereitete Informationen angefügt werden

können, die aber nicht direkt editiert werden kann. Mit der Datei HELP.COM muß HELP.HLP in die Datei HELP.DAT konvertiert werden, bevor editiert werden kann bzw. eigene Texte zugefügt werden können.

Diese Form des HELP-Kommandos hat folgende Optionen:

- * Die Option EXTRACT erzeugt aus der Datei HELP.HLP auf dem Standardlaufwerk eine Datei mit dem Namen HELP.DAT. Mit einem Textverarbeitungsprogramm kann nun editiert oder Text zur Datei HELP.DAT hinzugefügt werden. EXTRACT kann auf E abgekürzt werden.
- * Die Option CREATE erstellt aus der editierten Datei HELP.DAT wieder die Datei HELP.HLP auf dem Standardlaufwerk. CREATE kann auf C abgekürzt werden.

Topics und Subtopics müssen in einem speziellen Format an die Datei HELP.DAT angefügt werden. Eine Kopfzeile für ein topic in der Datei HELP.DAT muß folgende Form haben:

```
///nTopicname<cr>
```

Die drei Schrägstriche sind die Begrenzungszeichen für den Begriff und müssen in Spalte 1 beginnen. Im dargestellten Format ist n eine Zahl im Bereich zwischen 1 und 9, die das Niveau des Begriffes kennzeichnet. Ein Hauptbegriff hat immer die Zahl 1. Der erste Unterbegriff hat die Zahl 2. Der nächste Unterbegriff hat die Niveaunzahl 3 und so weiter bis zu einem Maximum vom 9. Topicname ist der Name des Begriffes und für diesen Namen sind bis zu 12 Zeichen möglich. Die vollständige Zeile wird mit einem Wagenrücklauf abgeschlossen:

Folgende Hinweise sind beim Editieren und Einfügen von Text in die Datei HELP.DAT zu beachten.

- * Topics müssen in alphabetischer Reihenfolge angeordnet werden.
- * Subtopics müssen ebenfalls alphabetisch geordnet werden.
- * Die Niveaus müssen durch die Ziffern 1–9 gekennzeichnet werden.

Einige Beispiele von Topic- und Subtopic-Zeilen:

```
///1NEUER SERVICE<cr>
```

```
///2Kommandos<cr>
```

```
///3PARAMETER<cr>
```

```
///2Beispiele<cr>
```

Das erste Beispiel zeigt das Format für eine Zeile zu einem Hauptbegriff. Das zweite Beispiel zeigt die Numerierung des ersten Unterbegriffes zum Hauptbegriff. Das dritte Beispiel zeigt, wie die nächste Ebene für den Unterbegriff nach dem Niveau 2 zu numerieren ist. Jeder Topicname mit der Ebenen-Nummer 1 ist ein Hauptbegriff. Jeder Topicname mit einer Ebenen-Nummer größer oder gleich 2 ist ein Unterbegriff innerhalb seines Hauptbegriffes.

Bei der Arbeit mit HELP.COM müssen immer nur so viele Buchstaben vom Begriff eingegeben werden, wie zu seiner eindeutigen Identifizierung nötig sind. Wird auf einen Unterbegriff Bezug genommen, dann müssen der Topicname und der Subtopicname eingegeben werden, anderenfalls kann das HELP-Programm nicht entscheiden, auf welchen Hauptbegriff Bezug genommen wird. Topic und Subtopic können auch wie folgt nach dem programminternen Bereitschaftszeichen HELP> eingegeben werden.

```
HELP>PIP OPTIONEN
```

Es werden die OPTIONEN von PIP angezeigt.

Das INIT-Kommando

Syntax:

INIT lw: [format{,R}]

Erklärung:

Dieses Kommando ermöglicht es, Disketten zu initialisieren. Die Wahl des Laufwerkes und des Formats sind beliebig. Die möglichen Formate sind laufwerkabhängig.

Bei zweiseitigen Laufwerken erfolgt bei der Wahl eines einseitigen Diskettenformates stets ein Löschen der Rückseite der Diskette. Soll die Rückseite unverändert bleiben, muß INIT mit Dialogbetrieb ausgeführt werden.

lw = Laufwerkadresse A,B,C,D

Wird Laufwerk A als ausführendes Laufwerk gewählt, erfolgt unabhängig vom aktuellen Laufwerk eine Abfrage, ob initialisiert werden soll; bei allen anderen Laufwerken wird die Diskette im entsprechenden Laufwerk sofort initialisiert und anschließend ein Warmstart ausgeführt.

format = 0,1,2,...

Die Anzahl und die Zuordnung der Nummern zu den entsprechenden Formaten wird durch den Laufwerktyp vorgegeben.

,R = der Initialisiervorgang wird mit demselben Laufwerk und demselben Format wiederholt. Es erfolgt nach der Anzeige des Formats eine Tastaturabfrage zum Diskettenwechseln bzw. zum Beenden.

Warnung

Die Diskette mit dem Kommando INIT wird ohne Warnung initialisiert, wenn sie in ein von A verschiedenes Laufwerk eingelegt und dieses als auszuführendes Laufwerk gewählt wird.

Fehlerbehandlung

Wurde das Kommando fehlerhaft eingegeben, dann wird INIT mit Warmstart beendet und die fehlerhafte Kommandozeile wird mit einem '^' markiert.

Wurde ein falsches Laufwerk eingegeben, wird das Kommando bei der Laufwerkanzeige mit entsprechender Fehlermeldung abgebrochen.

Wurde ein falsches Format eingegeben, wird das Kommando bei der Formatanzeige mit entsprechender Fehlermeldung abgebrochen.

Treten beim Initialisieren Lese- oder Schreib-Fehler auf, wird eine Fehlermeldung gebracht und das Initialisieren fortgesetzt. Alle anderen Fehler führen mit entsprechender Fehleranzeige zum Abbruch des Kommandos.

Beispiele:

A>B:INIT B: [0]

Mit dieser Kommandozeile wird im Laufwerk B das Kommando INIT gelesen und auf Laufwerk B ausgeführt. Das bedeutet, daß die Diskette in Laufwerk B ohne Warnung initialisiert wird.

A>INIT B:[0]

Mit dieser Kommandozeile wird im Laufwerk A das Kommando INIT gelesen und die Diskette im Laufwerk B mit dem Format 0 initialisiert. Nach dem Kommando befindet man sich wieder im Grundzustand (z.B.: A>).

B>A:INIT A:[1]

Mit dieser Kommandozeile wird im Laufwerk A das Kommando INIT gelesen und nach der LW- und der Format-Anzeige abgefragt, ob initialisiert werden soll. Jetzt kann die Diskette im Laufwerk A gewechselt werden. Unmittelbar nach der Eingabe von Y erfolgt die Initialisierung der Diskette im Laufwerk A mit dem Format 1. Nach dem Kommando befindet man sich wieder im Grundzustand (z. B.: A>).

B>A:INIT B: [1,R]

Mit dieser Kommandozeile wird im Laufwerk A das Kommando INIT gelesen und die Diskette im Laufwerk B mit dem Format 1 initialisiert. Nach Ausführung des Kommandos wird erneut LW- und Format angezeigt und abgefragt, ob initialisiert werden soll. Jetzt kann die Diskette im Laufwerk B gewechselt werden. Unmittelbar nach der Eingabe von Y erfolgt eine Wiederholung der Initialisierung. Mit X oder ^C wird das Kommando verlassen.

Dialogbetrieb**Syntax:**

INIT

Erklärung:

Dieses Kommando ermöglicht das Initialisieren von Disketten. Die Wahl des Laufwerkes und des Formats sind beliebig. Die möglichen Formate sind laufwerkabhängig. Die Abarbeitung erfolgt dialoggesteuert. Dabei werden alle für das entsprechend adressierte Laufwerk möglichen Formate angezeigt.

Bei zweiseitigen Laufwerken besteht die Möglichkeit, bei Wahl eines einseitigen Formats, die Rückseite der Diskette zu löschen bzw. die Rückseite so zu belassen, wie sie war. Nach dem Initialisieren kann man das Kommando mit X verlassen, mit R den Vorgang wiederholen und ET das Kommando bei der Laufwerkauswahl erneut starten.

Fehlerbehandlung

Werden falsche Werte angegeben (Laufwerk, Format, ...) muß die Eingabe wiederholt werden.

Treten beim Initialisieren Lese- oder Schreib-Fehler auf, wird eine Fehlermeldung gebracht und das Initialisieren fortgesetzt. Alle anderen Fehler führen mit entsprechender Fehleranzeige zum Abbruch des Initialisiervorganges aber nicht zum Verlassen des Kommandos.

Beispiele:

B>B:INIT

Mit dieser Kommandozeile wird im Laufwerk B das Kommando INIT gelesen. Es wird Laufwerk A als auszuführendes Laufwerk angezeigt. Nun kann das Laufwerk geändert werden, ein Format ausgewählt werden und das Initialisieren gestartet werden.

Das INITDIR-Kommando**Syntax:**

INITDIR d:

Erklärung:

Das Kommando INITDIR bereitet ein Diskettenverzeichnis für die Aufnahme von Datum- und Uhrzeiteintragungen vor. INITDIR kann auch den Platz für Datum und Zeit im Verzeichnis wieder freigeben.

INITDIR ist auf jede Diskette anzuwenden, für die die Eintragung von Datum und Uhrzeit der Dateien vorgesehen ist. Ist die Diskette leer, dann initialisiert INITDIR das Verzeichnis für Datum und Uhrzeit. Existieren bereits Dateien auf der Diskette, dann überprüft INITDIR den für die Eintragung von Datum und Uhrzeit verfügbaren Platz im Verzeichnis. Falls für Datum und Uhrzeit nicht ausreichend Platz vorhanden ist, dann initialisiert INITDIR das Verzeichnis nicht und kehrt mit einer Fehlermeldung zurück.

Nach dem Initialisieren des Verzeichnisses muß mit dem SET-Kommando die Option für Datum und Uhrzeit auf der Diskette festgelegt werden.

Beispiel:

A>INITDIR C:-

Das System verlangt die Bestätigung:

INITDIR WILL ACTIVATE TIME-STAMPS FOR SPECIFIED DRIVE.

Do you want to re-format the directory on C: (Y/N)?Y

Wurde das Verzeichnis bereits für Datum und Uhrzeit initialisiert, dann gibt INITDIR folgende Meldung aus:

Directory already re-formatted

Do you wish to recover date/time directory space (Y/N)?

Durch die Eingabe Y werden Datum- und Zeiteinträge aus dem Verzeichnis entfernt. Durch die Eingabe N bleiben Datums- und Zeiteinträge auf der Diskette aktiv und INITDIR gibt folgende Meldung aus:

DO you want the existing date/time stamps cleared (Y/N)?

Mit Y werden die existierenden Einträge gelöscht. Mit N bleiben die Datums- und Zeiteinträge erhalten.

Das LINK-Kommando

Syntax: LINK d: {dateibez, {[optionen]} = } dateibez
 {[optionen]} {, ...}

Erklärung: Mit LINK werden verschiebliche Objektmodule, wie sie von ASM oder BASIC erzeugt werden, zu einer ausführbaren .COM Datei gebunden. Verschiebliche Dateien können externe Bezüge und Eintrittspunkte (Public-Variablen) enthalten. Verschiebliche Dateien können auch externe Bezüge zu anderen Modulen in der Bibliotheksdatei enthalten. LINK durchsucht die Bibliotheks-Dateien und schließt die erforderlichen Module in die Ausgabendatei ein. Eine vollständige Beschreibung von LINK ist dem Programmierer-Handbuch zu entnehmen.

Mit der Verwendung von LINK-Optionen können Ausführungsparameter gesteuert werden. Diese LINK-Optionen werden in eckigen Klammern hinter der Dateibezeichnung angegeben. Mehrere Angaben werden durch Kommas getrennt.

Option	Bedeutung
A	Zusätzlicher Speicher; reduziert den Pufferbereich und schreibt temporäre Daten auf die Diskette
B	Linkt BIOS für gebanktes System SCP 3.0. Das Datensegment wird auf Seitengrenze ausgerichtet; die Länge des Codesegments wird im Header abgelegt; der erzeugte Standarddateityp ist SPR.

Option	Bedeutung
Dhhhh	Adresse Datensegment; setzt die Speicheradresse für den Common- und Datenbereich
Gn	Go; setzt die Startadresse auf die Marke n
Lhhhh	Load; ändert die Ladeadresse des Moduls auf hhhh. Standard ist 0100H
NL	Keine Ausgabe der Symboltabelle auf Console.
NR	Keine Datei der Symboltabelle
OC	Ausgabe ist Kommandodatei .COM (Standard).
OP	Ausgabe einer .PRL Datei (page relocatable)
OR	Ausgabe einer .RSP Datei (resident system process)
OS	Ausgabe einer .SPR Datei (system page relocatable)
Phhhh	Programmadresse; ändert die Standardprogrammadresse auf hhhh. Standard ist 0100H.
Q	Gibt die Symbole mit einem vorangestellten Fragezeichen aus.
S	Behandelt die angegebene Datei als eine Bibliothek.
αCd	Ziel der Console-Meldungen d kann sein: X (Console), Y (Drucker), oder Z (ohne Ausgabe). Standard ist X.
αld	Laufwerk der Arbeitsdateien. d ist ein Laufwerk A-P. Standard ist das aktuelle Laufwerk.
αLd	Quelle der Bibliotheks-Dateien; d ist ein Laufwerk A-P. Standard ist das aktuelle Laufwerk.
αOd	Ziel einer Objekdatei; d kann Z oder ein Diskettenlaufwerk A-P sein. Standard ist die Ausgabe auf das gleiche Laufwerk, auf dem sich auch die erste Datei im LINK-80-Kommando befindet.
αSd	Ziel der Symboldatei; d kann Y oder Z oder ein Diskettenlaufwerk A-P sein. Standard ist die Ausgabe auf das gleiche Laufwerk, auf dem sich auch die erste Datei im LINK-80 Kommando befindet.

Beispiele:

A>LINK b:MYFILE[NR]

LINK-80 auf Laufwerk A verwendet als Eingabe die Datei MYFILE.REL auf Laufwerk B und erzeugt die ausführbare Maschinencode-Datei MYFILE.COM auf Laufwerk B. Die Option [NR] unterdrückt die Ausgabe einer Symboltabelle.

A>LINK m=m1,m2,m3

LINK-80 bindet die getrennt übersetzten Dateien m1, m2, und m3, löst ihre externen Bezüge auf und erzeugt die ausführbare Maschinencode-Datei m.COM.

A>LINK MYFILE, FILE5[s]

Die Option [s] teilt LINK-80 mit, daß FILE5 als eine Bibliothek zu behandeln ist. LINK-80 bindet die Datei MYFILE.REL mit den in FILE5 enthaltenen externen Routinen auf dem Standardlaufwerk A und generiert die Datei MYFILE.COM auf Laufwerk A.

DAS MODCS-Kommando

Syntax:

MODCS dateibezeichnung [zeichensatz] . . .

Erklärung:

Dieses Kommando ermöglicht das Laden einer Datei mit dem Namen dateibezeichnung von Diskette in den Zeichengeneratorspeicher des Bildschirms. Dabei kann zwischen den beiden möglichen Zeichensatzspeichern zeichensatz (1 oder 2) gewählt werden. Eine Prüfung des Inhaltes der Zeichengeneratordatei, die eine Länge von 2048 Byte haben muß, erfolgt nicht. Mit Kaltstart des SCP 3.0 ist im Zeichengenerator 1 der Zeichensatz zur internationalen Datenerfassung geladen. Der Zeichengeneratorspeicher 2 ist undefiniert.

Ein automatisches Laden eines gewünschten Zeichensatzes kann mit Hilfe des Kommandos MODCS im Kaltstart-SUBMIT-File PROFILE.SUB erfolgen.

Ausgabe des Hilfstextes

Syntax:

MODCS

Erklärung:

Das oben angegebene Kommando bringt den Hilfstext zur Anzeige. Aus dem Hilfstext sind die möglichen Werte für die Parameter sowie ein Kommandobeispiel ersichtlich.

Fehlerbehandlung

Wurde ein fehlerhaftes Kommando eingegeben, dann wird MODCS mit Warmstart beendet und die fehlerhafte Kommandozeile wird mit einem '^' markiert.

A:ZG1.DIS[3]

ERROR: Invalid parameter

Beispiele:

A>MODCS B:LATKYR.DIS[2]

Die Zeichengeneratordatei LATKYR.DIS wird von Laufwerk B: in den Zeichengenerator 2 geladen.

A>MODCS LATEIN.DIS[1] LATKYR.DIS[2]

Die Datei LATEIN.DIS wird vom aktuellen Laufwerk in den Zeichengenerator 1 geladen und die Datei LATKYR.DIS in den Zeichengenerator 2.

Das MODFD-Kommando

Syntax:

MODFD {lw: {[parameter]} {lw: {[parameter]} . . .

Erklärung:

Mit Hilfe des MODFD-Kommandos ist es möglich, bestimmte Floppy-Disk-Parameter laufwerksspezifisch zu modifizieren.

Es können folgende Parameter geändert werden:

- der Laufwerktyp (DEVICE=)
- die Formaterkennung ein/aus (LOGIN=)
- die Anzahl Systemspuren (OFFSET=)

Die Reihenfolge der Parameter muß eingehalten werden (DEVICE=,LOGIN=,OFFSET), wobei einzelne Parameter fehlen können.

Modifizierte Parameter bleiben bis zu einer erneuten Änderung erhalten.

Ausgabe des Hilfstextes

Syntax:

MODFD

Erklärung:

Das oben angegebene Kommando bringt den Hilfstext zur Anzeige. Aus dem Hilfstext sind die möglichen Werte für die Parameter sowie ein Kommandobeispiel ersichtlich.

Anzeige der momentanen Parameter

Syntax: MODFD lw: {lw:} {lw:} . . .

Erklärung:

Diese Form des MODFD-Kommandos zeigt die momentanen Werte der FD-Parameter für die angegebenen Laufwerke lw: an.

lw: [DEVICE=typ,LOGIN=on/off,OFFSET=offset]

Ändern des Laufwerktyps

Syntax:

MODFD lw:[DEVICE=typ]

Erklärung:

Dieses Kommando ordnet dem Laufwerk lw: den Laufwerktyp typ zu.

Damit ist es möglich den Laufwerktyp auch nach der Systemgenerierung (siehe COPY-SYS) zu ändern. Die installierbaren Laufwerktypen sind aus dem Hilfstext ersichtlich.

Abschalten/Zuschalten der automatischen Formaterkennung

Syntax:

MODFD lw:[LOGIN=OFF|ON]

Erklärung:

Mit diesem Kommando kann die automatische Erkennung der Standardformate des SCP 3.0 abgeschaltet [LOGIN=OFF] oder zugeschaltet [LOGIN=ON] werden.

LOGIN=OFF schaltet die Formaterkennung für die weiteren Zugriffe auf das genannte Laufwerk aus, führt jedoch selbst noch eine Formaterkennung durch, um das momentane Format der eingelegten Diskette zu erkennen.

In diesem Laufwerk kann dann nur noch mit dem erkannten Diskettenformat gearbeitet werden!

Ist keine Diskette eingelegt oder kann das Format der eingelegten Diskette nicht erkannt werden, dann meldet sich MODFD mit

SCP2 Error on lw: INVALID DRIVE

und das Programm wird verlassen.

LOGIN=ON schaltet die Formaterkennung für das spezifizierte Laufwerk wieder zu. Eine Diskette im Laufwerk ist nicht erforderlich, da die Formaterkennung mit dem nächsten Zugriff auf die Diskette automatisch erfolgt.

Dieser Parameter ist für alle Laufwerke nach Systemstart eingestellt.

Ändern der Anzahl Systemspuren

Syntax:

```
MODFD lw:[LOGIN=OFF,OFFSET=n]
```

Erklärung:

Dieses Kommando legt die Anzahl Systemspuren n für das angegebene Laufwerk lw: fest. Die Anzahl kann von 0 bis 4 variieren.

Damit können im SCP 3.0 auch solche Disketten verarbeitet werden, die sich von den Standardformaten nur in der Anzahl Systemspuren unterscheiden.

HINWEIS

Dieses Kommando setzt eine Diskette im angegebenen Laufwerk voraus.

Der Parameter LOGIN=OFF muß angegeben werden um die Formaterkennung abzuschalten.

Fehlerbehandlung

Wurde ein fehlerhaftes Kommando eingegeben, dann wird MODFD mit Warmstart beendet und die fehlerhafte Kommandozeile wird mit einem '^' markiert.

```
lw:[OFFSET=10]
```

ERROR: Invalid parameter

Beispiele:

```
A>MODFD a: b: c: d:
```

Ausgabe der installierten Parameter der Laufwerke A: B: C: D:

```
A:Device=MFS1.6,Login=ON,Offset=2]
```

```
B:Device=MFS1.6,Login=ON,Offset=2]
```

```
C:Device=MFS1.2,Login=ON,Offset=1]
```

```
D:Device=MFS1.2,Login=ON,Offset=1]
```

```
B>MODFD c:[device=mf6400] d:[device=mfs6400]
```

Die Laufwerke C: und D: sind vom Typ MF6400.

```
B>MODFD b:[login=off,offset=0]
```

Die Diskette in Laufwerk B: soll mit 0 Systemspuren verarbeitet werden.

```
B>modfd d:[device=mfs16]
```

```
D:[DEVICE=MFS16]
```

ERROR: Invalid parameter

Der Laufwerktyp MFS16 existiert nicht (es muß MFS1.6 heißen).

Das MODIO-Kommando**Syntax:**

```
MODIO {phys.I/O-Gerät | {[Optionen]}} | {[ALL]}
```

```
MODIO {device [par1,par2, . . . parn]}
```

```
MODIO {[ALL]}
```

Erklärung:

Das transiente Kommando MODIO ermöglicht die Änderung der Werte für die Parameter der physischen I/O-Geräte und bietet die Möglichkeit, die aktuellen Parameter anzuzeigen. In der Beschreibung für das Kommando MODIO bezieht sich der Begriff I/O-Geräte auf den Bildschirm und auf die seriellen I/O-Anschlüsse, aber nicht auf die Tastatur oder vom Anwender definierte physische I/O-Geräte.

Die Reihenfolge der Parameter in der Kommandozeile muß eingehalten werden. Sie ist aus dem Hilfstext zu entnehmen.

Ausgabe des Hilfstextes

Syntax:
MODIO

Erklärung:

Das oben angegebene Kommando bringt einen Hilfstext zur Anzeige, aus dem die Kommandoformen, die Namen der physischen Geräte, die Parameternamen und die Parameterwerte zu entnehmen sind. Mehrere Beispiele für Kommandoaufrufe sollen dem Bediener Hinweise für die Anwendung des MODIO-Kommandos geben.

Inhalt eines Hilfstextes:

SCP 3 MODIO R-BWS V 0/1

MODIO device[par1,par2 . . . ,parn] – change parameter[s]

device = DISPLY – change display-parameter[s]

par1: SCREEN FORMAT = 24*80, 16*64

par2: CURSOR FORMAT = BR, BU, NR, NU

par3: CHARACTER SET = CHS1, CHS2

device = PRN, V24, CHAN_A, CHAN_B

par1: BIT-RATE = 9600,4800,2400,1200,600,300,150,110,75,50

device = V24, CHAN_A, CHAN_B

par2: MODE = OS, IOS

par3: PROCEDURE = DTR, DC1/DC3

par4: BITS/CHAR = 8, 7, 6, 5

par5: PARITY = NONE, ODD, EVEN

par6: STOP-BITS/CHAR = 1, 1.5, 2

MODIO device – look at parameters of device

MODIO [ALL] – overview of the IO-Drivers

examples:

MODIO DISPLY[24*80,NR,CHS1] – change display-parameters

MODIO PRN[1200] – change parameter for PRN

MODIO V24[4800,,DC1/DC3,7,ODD] – change parameters of V24

MODIO V24 – look at parameters of V24

Änderung der Parameter für ein I/O-Gerät

Syntax:
MODIO device [par1, par2, . . . ,parn]

Erklärung:

Das Kommando MODIO bewirkt in diesem Falle eine Änderung der Parameter des ausgewählten I/O-Gerätes.

Für „device“ ist der Name des gewünschten I/O-Gerätes anzugeben. Die möglichen Namen sind im Hilfstext enthalten.

Nachfolgend Beispiele für Gerätenamen:

device	Bedeutung
DISPLY	Bildschirm
PRN	Anschluß für Drucker (PRINTER)
V24	Anschluß mit V.24-Interface (V24)
CHAN_A	Anschluß Kanal A auf Zusatz-STE
CHAN_B	Anschluß Kanal B auf Zusatz-STE

Für die Kürzel „par1“ bis „parn“ in der Kommandozeile sind die Parameterwerte einzusetzen. Die Parameter bzw. ihre Namen stehen nicht in der Kommandozeile.

Die Parameterwerte sind durch Komma zu trennen. Die Reihenfolge der Werte in der Kommandozeile ist im Hilfstext angegeben. Entsprechend dieser Reihenfolge werden die Parameterwerte den Parametern zugeordnet und auf Gültigkeit geprüft.

Für jeden Parameterwert, der nicht geändert werden soll, kann ein Komma stehen, wenn noch zu ändernde Werte folgen. Die Kommandozeile kann nach dem letzten zu ändernden Parameterwert abgeschlossen werden.

Parameterübersicht für „DISPLY“:

Parameter	Parameterwert	Bedeutung
SCREEN FORMAT		Bildschirmformat
	24*80	24 Zeilen, 80 Spalten
	16*64	16 Zeilen, 64 Spalten
CURSOR FORMAT		Cursorformat
	BR	blinkend invers (blinking reverse)
	BU	blinkend unterstreichend (blinking underline)
	NR	nicht blinkend invers (noblinking reverse)
	NU	nicht blinkend unterstreichend (noblinking underline)
CHARACTER SET		Zeichensatz
	CHS1	Zeichensatz 1 (character set 1)
	CHS2	Zeichensatz 2 (character set 2)

Parameterübersicht für „PRN“:

Parameter	Parameterwert	Bedeutung
BIT-RATE	9600, 4800,, 75, 50	Übertragungsgeschwindigkeit in Baud

Parameterübersicht für „V24“, „CHAN_A“ und „CHAN_B“:

Parameter	Parameterwert	Bedeutung
BIT-RATE	9600, 4800,, 75, 50	Übertragungsgeschwindigkeit in Baud

Parameter	Parameterwert	Bedeutung
MODE	OS	serielle Ausgabe (output serial)
	IOS	serielle Ein- und Ausgabe (*1 (input/output serial)
PROCEDURE	DTR	DTR-Protokoll (Hardwareprotokoll)
	DC1/DC3	DC1/DC3 – Procedur
BITS/CHAR	8, 7, 6, 5	Anzahl der Bits pro Zeichen
PARITY	NONE	ohne Parität
	ODD	ungerade Parität
	EVEN	gerade Parität
STOP-BITS/CHAR	1, 1.5, 2	Anzahl der Stopbits pro Zeichen

(*1

Im Mode „IOS“ kann immer nur ein physisches Gerät arbeiten.

Wird diese Arbeitsweise einem weiteren physischen Gerät zugewiesen, so erfolgt gleichzeitig die Änderung des vorher mit „IOS“ eingestellten Gerätes auf den Mode „OS“, so daß die oben angegebene Bedingung erfüllt bleibt.

Diese automatische Änderung birgt die Gefahr eines durch sie hervorgerufenen Fehlers in der logischen Zuordnung in sich.

Die Meldung:

„Check Logical Assignment“
weist auf diese Fehlermöglichkeit hin.

Der Bediener sollte nach dieser Meldung die logische Zuordnung der Geräte überprüfen und im Fehlerfalle ändern.

Eine Änderung der Parameterwerte schließt einen Kaltstart mit den neuen Werten ein, so daß diese sofort wirksam werden. Der Kaltstart für den Bildschirm beinhaltet ein Löschen des Bildes.

Beispiel:

```
MODIO DISPLY [24*80,NR,CHS1]
SCP 3 MODIO R-BWS V 0/1
DEVICE: DISPLY
SCREEN FORMAT: 24*80 CURSOR FORMAT: NR
CHARACTER SET: CHS1
```

Beispiel:

```
MODIO PRN[1200]
SCP3 MODIO R-BWS V 0/1
DEVICE: PRN
BIT-RATE: 9600 MODE: OS PROCEDURE: DTR
BITS/CHAR:8 PARITY: NONE STOP-BITS/CHAR:1
```

```

DEVICE: PRN
BIT-RATE: 1200 MODE: OS PROCEDURE: DTR
BITS/CHAR:8 PARITY: NONE STOP-BITS/CHAR:1

```

Beispiel:

```
MODIO V24[4800,,DC1/DC3,7,ODD]
```

```
SCP3 MODIO R-BWS V 0/1
```

```
DEVICE: V24
```

```
BIT-RATE: 9600 MODE: OS PROCEDURE: DTR
BITS/CHAR:8 PARITY: NONE STOP-BITS/CHAR:1
```

```
DEVICE: V24
```

```
BIT-RATE: 4800 MODE: OS PROCEDURE: DC1/DC3
BITS/CHAR:7 PARITY: ODD STOP-BITS/CHAR:1
```

Anzeige der Parameter für ein I/O-Gerät

Syntax:

```

MODIO device
MODIO DISPLY
MODIO PRN
MODIO V24

```

Erklärung:

Mit diesem Kommando werden für das angegebene Gerät die aktuellen Parameterwerte angezeigt. Für „device“ ist der Name des gewünschten I/O-Gerätes anzugeben.

Beispiel:

```
MODIO V24
```

```
SCP3 MODIO R-BWS V 0/1
```

```
DEVICE: V24
```

```
BIT-RATE: 9600 MODE: OS PROCEDURE: DTR
BITS/CHAR:8 PARITY: NONE STOP-BITS/CHAR:1
```

Anzeige der Parameter für alle I/O-Geräte

Syntax:

```
MODIO [ALL]
```

Erklärung:

Diese Form des Kommandos gibt eine Übersicht über alle im Betriebssystem generierten I/O-Geräte.

Das PIP-Kommando

Syntax:

```
PIP ziel-dateibez|d: {[Gn]} = quell-dateibez {[o]}
{, ...} | d: {[o]}
```

Erklärung:

Das transiente Programm PIP kopiert eine oder mehrere Dateien von einer Diskette und/oder Nutzerbereich auf eine andere. Weiterhin kann mit PIP eine Datei nach dem Kopieren umbenannt werden. Mit PIP kann weiterhin eine Textdatei von Diskette zum Drucker oder

einem anderen logischen Ausgabegerät übertragen werden. Mit PIP kann eine Datei auf Diskette auch über die Eingabe von Konsole oder über ein logisches Eingabegerät erzeugt werden. PIP kann Daten von einem logischen Eingabegerät zu einem logischen Ausgabegerät übertragen, woraus sich der Programmname PIP (Peripheral Interchange Program) ableitet.

PIP kopiert die Dateiattribute der Datei mit. Dazu gehören Read-Write oder Read-Only und SYS oder DIR sowie die vom Nutzer definierbaren Attribute F1 bis F4. Besitzt eine Datei einen Paßwortschutz, dann muß das Paßwort in der Kommandozeile hinter dem Dateinamen und/oder Dateityp der zugehörigen Datei angegeben werden. Wenn das Paßwort fehlt, dann wird die Datei übergangen und das Fehlen angemerkt.

Wird zu einer Zieldatei ein Paßwort angegeben, dann ordnet PIP der Zieldatei das Paßwort zu und setzt automatisch den Paßwortschutz-Mode auf READ. Das bedeutet, daß zum Lesen der Datei das Paßwort erforderlich ist. Wird eine Zieldatei ohne Paßwort angegeben, dann ordnet PIP der Zieldatei auch kein Paßwort zu. Wird lediglich das Ziellaufwerk angegeben, dann erhält die Zieldatei den gleichen Paßwortschutz-Mode und das gleiche Paßwort, wie es in der Quelldatei angegeben wurde. Wird eine Zeildatei mit Paßwort angegeben, dann setzt PIP automatisch den Paßwortschutz-Mode auf READ. Das bedeutet, daß zum Lesen der Datei das Paßwort erforderlich ist.

Einfaches Dateikopieren

Syntax:

```
PIP d: {[Gn]} = quell-dateibez {[optionen]}
PIP ziel-dateibez {[Gn]} = d: {[optionen]}
PIP ziel-dateibez {[Gn]} = quell-dateibez[o]}
```

Erklärung:

Die erste Form zeigt die einfachste Art des Dateikopierens. PIP sucht die durch quell-dateibez benannte Datei auf dem Standardlaufwerk oder dem angegebenen Laufwerk. Die Datei wird mit PIP auf das durch d: festgelegte Laufwerk kopiert und erhält den Namen aus der quell-dateibez. Wenn gewünscht, kann mit der Option [Gn] der Nutzerbereich n für die Zieldatei festgelegt werden. Die einzige von der Zieldatei angenommene Option ist [Gn]. Verschiedene Optionen können kombiniert und mit der Quelldateibezeichnung angegeben werden.

Die zweite Form ist eine Variante der ersten. PIP sucht die durch ziel-dateibez bezeichnete Datei auf dem durch d: angegebenen Laufwerk, kopiert sie auf das Standardlaufwerk oder angegebene Laufwerk und gibt ihr den durch ziel-dateibez festgelegten Namen.

Die dritte Form zeigt das Umbenennen der Datei nach dem Kopieren. Die Datei kann sowohl auf das gleiche Laufwerk und den gleichen Nutzerbereich als auch auf ein anderes Laufwerk und/oder Nutzerbereich kopiert werden. Die Festlegungen für die Optionen sind die gleichen wie oben. PIP sucht die durch quell-dateibez benannte Datei, kopiert diese auf den durch ziel-dateibez festgelegten Platz und gibt ihr den Namen aus der ziel-dateibez.

Wird die Option [GN] nicht angegeben, dann bezieht sich PIP immer auf den aktuellen Nutzerbereich.

Vor der Anwendung von PIP muß sichergestellt sein, daß auf der Zieldiskette genügend freier Platz in Kilobyte vorhanden ist, um alle zu kopierenden Dateien aufzunehmen. Soll eine alte Kopie auf der Zieldiskette durch eine neue ersetzt werden, dann benötigt PIP

noch genügend freien Raum auf der Diskette für die neue Kopie, bevor die alte gelöscht wird. Mit dem DIR-Kommando können die Dateigrößen und damit dem SHOW-Kommando der Platz auf der Diskette bestimmt werden. Ist nicht mehr genügend Platz auf der Diskette, dann kann mit ERASE zuerst die alte Kopie gelöscht werden.

Die Daten werden zunächst in eine temporäre Datei kopiert, um sicherzustellen, daß die gesamten Daten der Datei in dem verfügbaren Raum auf der Diskette Platz finden. Die temporäre Datei erhält den Dateinamen der Zieldatei und als Dateityp $\alpha \alpha \alpha$. Nach erfolgreichem Abschluß des Kopiervorganges wird der Dateityp der temporären Datei in den der Zieldatei umbenannt.

Ist der Kopiervorgang erfolgt und es existiert bereits eine Datei mit dem gleichen Namen wie die Zieldatei, dann wird zunächst die alte Datei mit dem gleichen Namen gelöscht und anschließend die temporäre Datei umbenannt.

Die Dateiattribute (DIR, SYS, RO, RW) werden mit den Dateien übertragen.

Hat die existierende Datei das Dateiattribut Read-Only, dann fragt PIP ab, ob sie gelöscht werden soll. Diese Abfrage ist mit Y oder N zu beantworten. Mit der Option [W] können Read-Only-Dateien überschrieben werden.

Für die Quelldateien können verschiedene Optionen angegeben werden. Für die Zieldateibezeichnung gibt es nur eine gültige Option ([Gn] – go to user number n). Optionen werden in eckige Klammern eingeschlossen. Für die Quelldateien können verschiedene Optionen angegeben werden. Sie können zusammen oder durch Leerzeichen getrennt angegeben werden. Optionen können prüfen, ob eine Datei korrekt kopiert wurde, das Programm PIP in die Lage versetzen, Dateien mit dem Attribut SYS zu lesen, Read-Only-Dateien zu überschreiben, Dateien aus oder in einem angegebenen Nutzerbereich zu kopieren, Umwandlung von Klein- und Großbuchstaben auszuführen und vieles mehr.

Beispiele:

```
A>PIP B:=A:oldfile.dat
```

```
A>PIP B:=oldfile.dat=A:
```

In beiden Formen dieses Kommandos liest PIP die Datei oldfile.dat von Laufwerk A und legt eine Kopie der Datei auf Laufwerk B ab. Diese Kommandozeilen werden als Kurzform von PIP bezeichnet, da die Quelle oder das Ziel nur ein Laufwerk bezeichnen und kein Dateiname enthalten ist. Bei Verwendung dieser Form kann keine Datei von einem Laufwerk und Nutzerbereich auf das gleiche Laufwerk und Nutzerbereich kopiert werden. Die Zieldatei muß in einem anderen Laufwerk oder einem anderen Nutzerbereich abgelegt werden. Die zweite Kurzform erzeugt das gleiche Ergebnis wie die erste. PIP sucht die Datei oldfile.dat auf dem als Quelle definierten Laufwerk A.

```
A>PIP B:newfile.dat=A:oldfile.dat
```

Dieses Kommando kopiert die Datei oldfile.dat von Laufwerk A nach Laufwerk B und benennt sie dabei in newfile. dat um. Die Quelldatei bleibt als oldfile.dat auf Laufwerk A erhalten. Dies ist die lange Form des PIP-Kommandos, da auf beiden Seiten der Kommandozeile die Dateinamen angegeben werden.

```
A>PIP newfile.dat=oldfile.dat
```

Mit dieser langen Form von PIP kann eine Datei von einem Laufwerk und Nutzerbereich (das ist u. a. der Nutzerbereich 0, da SCP 3.0 automatisch mit Nutzerbereich 0 beginnt –

Standardnutzerebereich) auf das gleiche Laufwerk und Nutzerebereich kopiert werden. Damit erhält man zwei Kopien einer Datei mit unterschiedlichen Namen auf einem Laufwerk und Nutzerebereich.

A>PIP B:PROGRAM.BAK=A:PROGRAM.DAT[G1]

Das oben angegebene Kommando kopiert die Datei PROGRAM.DAT vom Nutzerebereich 1 auf Laufwerk A in den zur Zeit ausgewählten Nutzerebereich auf Laufwerk B und benennt dabei den Dateityp auf Laufwerk B in BAK um.

B>PIP program2.dat=A:program1.dat[E V G3]

Mit diesem Kommando kopiert PIP die Datei program1.dat von Laufwerk A und gibt die Übertragung auch auf der Console aus [E], überprüft [V], daß die beiden Kopien völlig identisch sind und holt die Datei program1.dat aus dem Nutzerebereich 3 von Laufwerk A [G3]. Da für die Zieldatei kein Laufwerk angegeben wurde, kopiert PIP die Datei automatisch auf Standardnutzerebereich und -laufwerk, in diesem Fall sind das der Nutzerebereich 0 und Laufwerk B.

Mehrfaches Dateikopieren

Syntax:

PIP d: {[Gn]} = {d:} ersatzzeichen-dateibez {[optionen]}

Erklärung:

Werden Ersatzzeichen in der Quellbezeichnung verwendet, dann kopiert PIP alle der Datei-bezeichnung genügenden Dateien nacheinander auf das Ziellaufwerk, wobei die ursprünglichen Namen jeder Datei übernommen werden. PIP zeigt die Meldung COPYING an, der jeweils der Name der gerade kopierten Datei folgt. PIP gibt eine Fehlermeldung aus und bricht den Kopiervorgang ab, wenn Laufwerk und Nutzerebereich für Quelle und Ziel gleich sind.

Beispiele:

A>PIP B:=A:*.COM

PIP kopiert alle Dateien, die den Dateityp COM besitzen, von Laufwerk A auf Laufwerk B.

A>PIP B:=A:*. *

Mit diesem Kommando kopiert PIP alle Dateien von Laufwerk A nach Laufwerk B. Damit könnte z. B. eine Kopie der Vertriebsdiskette angefertigt werden. Allerdings muß dabei berücksichtigt werden, daß dieses Kommando nicht die Systemspuren kopiert. Für das Kopieren der Systemspuren ist COPYSYS zuständig.

A>PIP B:=A:PROG????.*

Damit werden alle Dateien, deren Dateiname mit PROG beginnt, von Laufwerk A nach Laufwerk B kopiert.

A>PIP B:[G1]=A:*.BAS

PIP kopiert alle Dateien mit dem Dateityp BAS vom Standardnutzerebereich (Nutzerebereich 0) des Laufwerkes A in den Nutzerebereich 1 auf Laufwerkes A in den Nutzerebereich 1 auf Laufwerk B. Im Gegensatz dazu greifen DIR, TYPE, ERASE und andere Kommandos nur auf Dateien in den Nutzerebereich zu, von dem sie aufgerufen wurden.

Dateien kombinieren

Syntax:

PIP ziel-dateibez {[Gn]} = quell-dateibez {[o]},
 quell-dateibez {[o]} { ,... }

Erklärung:

In dieser Form des PIP-Kommandos können zwei oder mehr Dateien als Quelle angegeben werden. PIP kopiert die angegebenen Dateien und kombiniert sie in eine Datei mit dem in der Zieldateibezeichnung angegebenen Namen. Dieses Verfahren wird als Dateiverkettung bezeichnet. Hinter der Zieldateibezeichnung kann die Option [Gn] angegeben werden, um die Datei in den Nutzerbereich n abzulegen. Für die Quelldateien können eine oder mehrere Optionen angegeben werden.

Einige Optionen veranlassen PIP zu einem zeichenweisen Dateikopieren. In diesem Fall sucht PIP das Zeichen CTRL-Z, um das Dateiende festzulegen. Alle PIP-Optionen bewirken eine Datenübertragung außer den folgenden:

A, C, Gn, K, O, R, V und W.

Das Kopieren von Daten zu oder von einem logischen Gerät bewirkt auch eine zeichenweise Übertragung.

Die Arbeit von PIP kann durch die Eingabe von CTRL-C abgebrochen werden.

Bei der Verkettung von Dateien untersucht PIP nur den letzten Satz der Datei nach dem EOF-Zeichen CTRL-Z. PIP unterbricht jedoch sofort die zeichenweise Übertragung, wenn es ein CTRL-Z Zeichen findet.

Sollen Dateien im Maschinencode verkettet werden, dann ist die Option [O] anzugeben. Die Option [O] bewirkt, daß eingebettete CTRL-Z (EOF) Zeichen ignoriert werden. Die EOF-Zeichen stellen in Textdateien das Ende der Datei dar, sind aber gültige Daten in einer Objektcode-Datei.

Beispiele:

A>PIP NEWFILE=FILE1,FILE2,FILE3

Die drei Dateien FILE1, FILE2 und FILE3 werden miteinander verbunden und in die Datei NEWFILE. α α α kopiert. Nach erfolgreichem Abschluß des Kopiervorganges wird die Datei NEWFILE. α α α umbenannt in NEWFILE. Alle Quell- und Zieldateien befinden sich auf der Diskette im Standardlaufwerk A.

A>PIP B:X.BAS=Y.BAS,B:Z.BAS

Die Datei Y.BAS auf Laufwerk A wird mit Z.BAS von Laufwerk B verbunden und in der temporären Datei X. α α α auf Laufwerk B abgelegt. Die Datei X. α α α wird umbenannt in X.BAS auf Laufwerk B, wenn PIP die Übertragung erfolgreich abgeschlossen hat.

Dateikopieren von und nach logischen Geräten**Syntax:**

```
PIP ziel-dateibez {[Gn]} =quell-dateibez {[o]}
      AUX:           AUX: {[O]}
      CON:           CON: {[O]}
      PRN:           NUL:
      LST:           EOF:
```

Erklärung:

Diese Form ist ein spezieller Fall der PIP-Kommandozeile, mit der eine Datei von einer Diskette auf ein Gerät, von einem Gerät auf eine Diskette oder von einem Gerät auf ein anderes kopiert werden kann. Die Dateien müssen druckbare Zeichen enthalten. Jedes periphere Gerät ist einer logischen Einheit zugeordnet, die eine Quelleneinheit zum Senden

von Daten oder eine Zieleinheit zum Datenempfang darstellen kann. (Vergl. dazu DEVICE-Kommando). Der an jeden logischen Gerätenamen angefügte Doppelpunkt dient der Unterscheidung von einer Dateibezeichnung. Ein Kopiervorgang, der ein logisches Gerät als Ziel oder Quelle benutzt, kann mit der Eingabe von CTRL-C abgebrochen werden.

Logische Gerätenamen:

- * CON: Konsolen-Ein- oder -Ausgabegerät. Als Quelle wird im allgemeinen die Tastatur verwendet; als Ziel der Bildschirm.
- * AUX: AUXIN- oder AUXOUT-Gerät.
- * LST: Als Ausgabegerät wird normalerweise der Drucker zugeordnet.

Die folgenden drei Gerätenamen haben eine besondere Bedeutung:

- * NUL: Diese Quelleinheit liefert 40 hexadezimale Nullen.
- * EOF: Diese Quelleinheit liefert ein einfaches CTRL-Z, das SCP-3.0-End-Of-File-Zeichen.
- * PRN: Drucker mit Tabulatorsprüngen auf jede achte Spalte, Zeilennummern und Seitenvorschub nach jeder 60. Zeile.

Beispiele:

```
B>PIP PRN:=CON:,MYDAT.DAT
```

Die Zeichen werden vom Konsolen-Eingabegerät, gewöhnlich ist das eine Tastatur, gelesen und direkt zum Drucker gesendet. Um PIP mitzuteilen, daß die Tastatureingabe abgeschlossen ist, muß ein CTRL-Z eingegeben werden. PIP liest dann sofort die Daten von der Datei MYDAT.DAT auf Laufwerk B. Da PRN: das Zielgerät ist werden Tabulatorsprünge ausgeführt, Zeilennummern hinzugefügt und Seitenwechsel nach jeder 60. Zeile ausgeführt.

Es ist zu beachten, daß bei Verwendung des logischen Gerätes CON: als Quelle sowohl der Wagenrücklauf (RETURN) als auch der Zeilenvorschub (LF) für eine neue Zeile eingegeben werden müssen.

```
A>PIP B:FUNFILE.SUE=CON:
```

Alles, was über die Console eingegeben wird, wird in die Datei FUNFILE.SUE auf Laufwerk B geschrieben. Die Tastatureingabe wird mit CTRL-Z abgeschlossen.

```
A>PIP LST:=CON:
```

Was über die Tastatur eingegeben wird, wird auf das List-Gerät ausgegeben, im allgemeinen ist das ein Drucker. Mit CTRL-Z wird die Eingabe abgeschlossen.

```
A>PIP LST:=B:DRAFT.TXT[T8]
```

Die Datei DRAFT.TXT auf Laufwerk B wird auf den Drucker ausgegeben. Jeder Tabulatorsprung wird zur nächsten Spalte, die ein Vielfaches von acht darstellt, ausgeführt.

```
A>P PRN:B:DRAFT.TXT
```

Mit diesem Kommando schreibt PIP die Datei auf das List-Gerät. Es werden automatisch Tabulatorsprünge ausgeführt, Zeilennummern zugefügt und Seitenvorschübe nach 60 Zeilen vorgenommen.

Mehrfachkommando Arbeitsweise

Syntax:

PIP

Erklärung:

Diese Form des PIP-Kommandos startet das Programm PIP und ermöglicht die Eingabe mehrerer Kommandozeilen, wobei PIP im Nutzerspeicher bleibt.

PIP gibt als Bereitschaftszeichen einen Stern auf dem Bildschirm aus.

Nach dem Bereitschaftszeichen kann jede gültige Kommandozeile, wie sie in den vorangegangenen Abschnitten beschrieben wurde, angegeben werden.

Die Arbeit mit PIP kann beendet werden, wenn nach dem Stern nur die RETURN-Taste betätigt wird. Die leere Kommandozeile ist für PIP das Kennzeichen für das Beenden der Arbeit und die Rückkehr zum SCP-3.0-Bereitschaftszeichen.

Beispiele:

A>PIP

SCP 3.0 PIP VERSION 1/0

*NEWFILE=FILE1, FILE2, FILE3

*APROG.COM=BPOG.COM

*A:=B:X.BAS

B:=.*

*^M

A>

Dieses Kommando lädt das PIP-Programm. Das Bereitschaftszeichen Stern zeigt an, daß PIP zur Kommandoeingabe bereit ist. Die Wirkung dieser Kommandoreihenfolge ist die gleiche wie in den vorangegangenen Beispielen, in denen die Kommandozeile bereits im Kommandoaufruf selbst enthalten ist. PIP wird nicht für jedes Kommando in den Speicher geladen. Um diese Arbeitsweise von PIP abzuschließen, ist die RETURN-Taste zu betätigen, oder eines der äquivalenten Control-Zeichen CTRL-J oder CTRL-M einzugeben.

Die Verwendung von Optionen

Erklärung:

Mit Hilfe der Optionen kann eine Quelldatei auf verschiedenen Wegen vervielfältigt werden. Es können TAB-Zeichen angefügt werden, eine Umwandlung von Groß- und Kleinbuchstaben erfolgen, Textteile herausgezogen werden, eine Kontrolle auf Überprüfung der Korrektheit der Kopie erfolgen und vieles mehr.

Die PIP-Optionen sind in der nachfolgenden Tabelle angegeben, wobei n eine Zahl und s eine Zeichenkette mit Abschluß CTRL-Z darstellen. Eine Option muß unmittelbar nach dem durch sie beeinflussten Gerät oder Datei stehen. Die Option muß in eckige Klammern eingeschlossen werden. Bei den Optionen, die einen numerischen Wert verlangen, darf zwischen dem Buchstaben und dem Wert kein Leerzeichen stehen.

Die Option [Gn] kann nach einer Zieldatei-Bezeichnung angegeben werden. Nach Dateien oder logischen Geräten als Quelle kann eine Liste von Optionen angegeben werden. Eine Liste von Optionen ist eine Folge von Buchstaben und numerischen Werten, die durch Leerzeichen getrennt werden können und in eckige Klammern [] eingeschlossen werden.

PIP-Optionen

Option Funktion

A	Archive. Kopiert nur Dateien, die nach dem letzten Kopieren verändert wurden.
C	Confirm. PIP verlangt eine Bestätigung vor dem Kopieren jeder Datei.
Dn	Löscht alle Zeichen nach Spalte n. Z. B. zum Begrenzen der Zeilenlänge bei der Druckerausgabe auf die druckbare Zeilenlänge.
E	Echo. Alle Zeichen werden während der Übertragung auch auf der Konsole angezeigt.
F	Filter (Form-Feeds). Alle Seitenvorschubzeichen werden beim Kopieren entfernt.
Gn	Kopieren in oder aus dem Nutzerbereich n.
H	Überprüfen der Dateien auf korrektes INTEL-Hex-Format bei der Übertragung von Hex-Dateien. Fehler werden auf der Console angezeigt.
I	Ignorieren von :00 Hex Datensätze bei der Übertragung von Dateien im INTEL-Hex-Format. Mit dieser Option wird automatisch auch die Option H aktiviert.
L	Umwandeln von Groß- in Kleinbuchstaben. Alle anderen Zeichen bleiben unverändert.
N	Numerieren der Ausgabezeilen. Die Nummern beginnen mit 1 und werden von Zeile zu Zeile um 1 erhöht. Der Zeilennummer folgt ein Doppelpunkt. Wird N2 angegeben, dann fügt PIP führende Nullen zur Zeilennummer hinzu und fügt TAB-Zeichen nach der Nummer ein. Ist die Option T ebenfalls angegeben, dann führt PIP Tabulatorsprünge aus.
O	Kopieren von Objektdateien. Das Dateiendezeichen CTRL-Z wird wie ein normales Zeichen übertragen aber beendet die Übertragung nicht.
Pn	Setzen Seitenlänge. Dabei stellt n die Anzahl Zeilen pro Seite dar. Wird diese Option bei der Quelldatei angegeben, dann fügt PIP einen Seitenvorschub an den Beginn und nach jeweils n Zeilen ein. Ist n=1 oder nicht angegeben, dann fügt PIP nach jeweils 60 Zeilen einen Seitenvorschub ein. Wird zusätzlich die Option F angegeben, dann ignoriert PIP die Form-Feeds in der Quelldatei und fügt neue Form-Feeds nach der durch n angegebenen Seitenlänge in die Zieldatei ein.
Qs^Z	Abbrechen des Kopiervorganges von der Quelle bei Zeichenkette s. Wird zusätzlich die Option S angegeben, dann kann damit ein Auszug aus der Quelldatei angefertigt werden. Die Zeichenkette muß mit CTRL-Z abgeschlossen sein.
R	Liest Dateien mit dem Attribut SYS. Normalerweise ignoriert PIP Dateien, die im Verzeichnis mit dem Systemattribut markiert wurden. Mit dieser Option aber kopiert PIP Systemdateien einschließlich dieses Attributes.
Ss^Z	Starten des Kopiervorganges von der Quelle bei Zeichenkette s. Die Zeichenkette muß mit CTRL-Z abgeschlossen sein. Wird zusätzlich die Option Q verwendet, dann kann ein Auszug aus einer Datei hergestellt werden. Sowohl die Beginn- als auch die Ende-Zeichenkette werden in die Zieldatei übernommen.

- Tn Diese Option nach einem Quelldateinamen bewirkt das Einfügen von TAB-Zeichen in die Zieldatei. PIP ersetzt jedes CTRL-I durch so viele Leerzeichen, daß das nächste Zeichen auf eine durch n dividierbare Spalte gesetzt wird.
- U Umwandeln von Klein- in Großbuchstaben. Alle anderen Zeichen bleiben unverändert.
- V Kontrollieren, daß die Daten korrekt kopiert wurden. Das ist nur bei Disketten-dateien möglich.
- W Überschreiben von Read-Only-Dateien ohne Consoleabfrage. Enthält die Kommandozeile mehrere Quelldateien, dann reicht es aus, wenn die Option nach der letzten Datei angegeben wird.
- Z Löschen des Paritätsbits. Mit dieser Option setzt PIP das Paritätsbit eines jeden Datenbytes in der Zieldatei auf Null. Die Quelldaten müssen Zeichen sein.

Beispiele:

A>PIP NEWPROG.BAS=CODE.BAS[L],DATE.BAS[U]

Dieses Kommando erzeugt die Datei NEWPROG.BAS auf Laufwerk A durch Verkettung der beiden Dateien CODE.BAS und DATA.BAS von Laufwerk A. Während des Kopierens wird CODE.BAS in Klein- und DATA.BAS in Großbuchstaben konvertiert.

A>PIP CON:=WIDEFIELD.BAS[D80]

Mit diesem Kommando wird die Textdatei WIDEFIELD.BAS von Laufwerk A auf das Consolegerät kopiert, löscht aber alle Zeichen nach der 80. Spalte.

A>PIP B:=LETTER.TXT[E]

Die Datei LETTER.TXT von Laufwerk A wird als LETTER.TXT auf Laufwerk B kopiert. Die Datei LETTER.TXT wird während des Kopierens auch auf dem Bildschirm ausgegeben.

A>PIP LST:=B:LONGPAGE.TXT[FP65]

Dieses Kommando gibt die Datei LONGPAGE.TXT auf den Drucker aus. Bei der Ausgabe der Datei werden die vorhandenen Form-Feed-Zeichen entfernt und zu Beginn und nach jeder 65. Zeile neue Form-Feed-Zeichen eingefügt.

B>PIP LST:=PROGRAM.BAS[NT8U]

Das Kommando gibt die Datei PROGRAM.BAS von Laufwerk B auf den Drucker aus. Durch die Angabe der Option N werden die Zeilen numeriert. Die Option T8 legt die Tabulatorsprünge auf jede achte Spalte fest. Wenn die Datei gedruckt wird, werden mit der Option U Klein- in Großbuchstaben umgewandelt.

A>PIP PORTION.TXT=LETTER.TXT[SDear Sir^ZQSincerely^Z]

Dieses Kommando zieht aus der Datei LETTER.TXT von Laufwerk A einen Teil heraus, indem es vor dem Kopierbeginn die Zeichenkette „Dear Sir“ sucht. Ist diese Zeichenkette gefunden, dann werden solange Zeichen in die Datei PORTION.TXT auf Laufwerk A kopiert, bis in der Quelldatei die Zeichenkette „Sincerely“ gefunden wird.

B>PIP B:=A:*.COM[VWR]

Dieses Kommando kopiert alle Dateien mit dem Dateityp COM von Laufwerk A nach B. Die Option V bewirkt, daß die Zieldateien gelesen werden, um eine exakte Datenübertragung sicherzustellen. Die Option W bewirkt das Überschreiben aller Zieldateien, die mit dem Attribut RO (Read Only) gekennzeichnet sind. Mit der Option R ist es PIP möglich, Dateien von Laufwerk A zu lesen, die mit dem SYS (System) Attribut markiert sind.

Das PUT-Kommando

Syntax:

```
PUT CONSOLE {OUTPUT TO} FILE dateibez {option}
PUT PRINTER {OUTPUT TO} FILE dateibez {option}
PUT CONSOLE {OUTPUT TO} CONSOLE
PUT PRINTER {OUTPUT TO} PRINTER
```

Erklärung:

Mit dem PUT-Kommando wird die Konsole- oder Druckausgabe in eine Diskettendatei umgeleitet. PUT legt die Konsolen- oder Druckerausgabe für das nächste auf Konsole eingegebene Kommando oder Nutzerprogramm auf eine Datei. PUT mit der Option SYSTEM lenkt alle nachfolgenden Konsolen- oder Druckausgaben in eine Datei.

Die Konsolenausgabe läuft solange in eine Datei, bis das Programm zu Ende ist. Dann erfolgt die Konsolenausgabe wieder auf Konsole.

PUT mit der Option SYSTEM lenkt alle nachfolgenden Konsolen-/Druckausgaben auf die angegebene Datei. Diese Option wird beendet, wenn das Kommando PUT CONSOLE oder PUT PRINTER eingegeben wird.

Die Syntax für die Liste von Optionen ist wie folgt:

```
[ {ECHO | NO ECHO} {FILTER | NO FILTER} | {SYSTEM} ]
```

PUT-Optionen

Option	Bedeutung
ECHO	gibt an, daß das Echo der Ausgabe auf die Konsole erfolgt. Diese Option ist Standard, wenn die Konsolenausgabe auf eine Datei gelegt wird.
NO ECHO	bedeutet, daß kein Echo auf der Konsole ausgeführt wird, wenn die Ausgabe in eine Datei erfolgt. NO ECHO ist Standard für das Kommando PUT PRINTER.
FILTER	bewirkt das Herausfiltern von Control-Zeichen, d. h., die Control-Zeichen werden in druckbare Zeichen übersetzt. Z. B. wird ein ESC-Zeichen in ^[übersetzt.
NO FILTER	bedeutet, daß PUT keine Control-Zeichen übersetzt. Diese Option ist Standard.
SYSTEM	bewirkt, daß sowohl die Ausgaben vom System als auch die vom Programm in die durch dateibez angegebene Datei geschrieben werden. Die Ausgaben werden solange in die Datei geschrieben, bis ein nachfolgendes Kommando PUT CONSOLE die Konsolenausgaben wieder auf die Console legt.

Umlenken der Konsolenausgabe in eine Datei

Syntax:

```
PUT CONSOLE {OUTPUT} TO FILE dateibez {[o]}
```

Erklärung:

Dieses Kommando veranlaßt das System, die nachfolgenden Konsolenausgaben in eine Datei zu leiten.

Beispiel:

```
A>PUT CONSOLE OUTPUT TO FILE XOUT [ECHO]
```

- * Lenkt die Konsolenausgaben in die Datei XOUT, wobei ein Echo der Ausgaben auf der Konsole erfolgt.

Umlenken der Druckausgabe in eine Datei

Syntax:

PUT PRINTER {OUTPUT TO} FILE dateibez {[o]}

Erklärung:

Diese Form des PUT-Kommandos lenkt die Druckausgabe in eine Datei.

Die Optionen sind die gleichen wie im PUT-CONSOLE-Kommando, außer das die Option NO ECHO für das Kommando PUT PRINTER Standard ist. Es ist zu beachten, daß bei Angabe von ECHO ein Echo der Druckausgabe auf den Drucker erfolgt.

Beispiele:

```
A>PUT PRINTER OUTPUT TO FILE XOUT
A>MYPROG
```

Damit werden die Druckausgaben des Programms MYPROG in die Datei XOUT gelenkt. Ein Echo der Ausgaben auf den Drucker erfolgt nicht.

```
A>PUT PRINTER OUTPUT TO FILE XOUT2 [ECHO, SYSTEM]
```

Damit erfolgen alle Druckausgaben sowohl in die Datei XOUT2 als auch auf den Drucker (mit ECHO-Option) und PUT bleibt wirksam bis das Kommando PUT PRINTER OUTPUT TO PRINTER eingegeben wird.

Die Druckausgabe kann auf eine oder mehrere Dateien umgelenkt werden. Die Ausgabe in diese Dateien wird abgebrochen, wenn mit folgendem Kommando die Druckerausgabe zurück auf den Drucker gelegt wird:

```
PUT PRINTER OUTPUT TO PRINTER
```

Beenden der Konsolenausgabe in eine Datei

Syntax:

```
PUT CONSOLE {OUTPUT TO} CONSOLE
```

Erklärung:

Die Konsolenausgabe wird auf die Konsole gelenkt.

Beispiel:

```
A>PUT CONSOLE OUTPUT TO CONSOLE
```

Die Konsolenausgaben erfolgen damit wieder auf die Konsole.

Beenden der Druckausgabe in eine Datei

Syntax:

```
PUT PRINTER {OUTPUT TO} PRINTER
```

Erklärung:

Das PUT-Kommando lenkt die Druckausgabe auf den Drucker.

Beispiel:

```
A>PUT PRINTER OUTPUT TO PRINTER
```

Die Druckausgaben erfolgen auf den Drucker.

Das RENAME-Kommando

Syntax:

RENAME {neue-dateibez=alte-dateibez}

Erklärung:

Mit RENAME kann der Name einer Datei im Verzeichnis einer Diskette geändert werden. Um mehrere Dateinamen in einem Kommando zu ändern, sind die Zeichen * und ? in den Dateibezeichnungen zu verwenden. Das Kommando RENAME kann mit REN abgekürzt werden.

Die neue Dateibezeichnung darf nicht der Name einer auf der Diskette existierenden Datei sein. Der alte Name bezeichnet eine existierende oder mehrere existierende Dateien auf der Diskette.

Das Kommando RENAME ändert den Namen einer Datei von dem in alte-dateibez angegebenen in den Namen neue-dateibez.

RENAME erstellt keine Kopie der Datei sondern ändert lediglich den Namen der Datei.

Wird die Laufwerksangabe weggelassen, dann erwartet RENAME die Datei, die umbenannt werden soll, auf dem Standardlaufwerk. Eine Laufwerksbezeichnung kann als ein Teil des neuen Namens angegeben werden. Wird in beiden Dateibezeichnungen ein Laufwerk angegeben, dann muß es das gleiche Laufwerk sein.

Existiert die durch den alten Namen angegebene Datei nicht, dann gibt RENAME folgende Meldung auf dem Bildschirm aus:

No File

Ist die durch den neuen Dateinamen angegebene Datei bereits im Verzeichnis, dann gibt RENAME folgende Meldung auf dem Bildschirm aus:

Not renamed: filename.typ file already exists, delete (Y/N)?

Soll die alte Datei gelöscht werden, dann ist ein Y einzugeben. Anderenfalls ist ein N einzugeben, und die alte Datei bleibt erhalten und die neue Datei wird nicht umbenannt.

Werden in den Dateibezeichnungen Ersatzzeichen (* und ?) verwendet, dann müssen die Ersatzzeichen in der neuen Dateibezeichnung denen in der alten Dateibezeichnung exakt entsprechen. In den folgenden zwei Kommandos ist die Verwendung der Ersatzzeichen korrekt:

A>REN *.TX1=*.TEX

A>REN A*.T*=S*.T*

Im folgenden Beispiel sind die Ersatzzeichen falsch angewendet und das System gibt eine Fehlermeldung aus.

A>REN A*.TEX=A.T*

Beispiele:

A>RENAME NEWFILE.BAS=OLDFILE.BAS

Die Datei OLDFILE.BAS wird umbenannt in NEWFILE.BAS auf Laufwerk A.

A>RENAME

Das System verlangt die Dateibezeichnungen:

Enter New Name:X.PRN

Enter Old Name:Y.PRN

Y .PRN=X .PRN

A>

Die Datei Y.PRN auf Laufwerk A wird umbenannt in X.PRN.

B>REN A:X.PAS=Y.PLI

Die Datei Y.PLI wird umbenannt in X.PAS auf Laufwerk A.

A>RENAME S*.TEX=A*.TEX

Das obige Kommando benennt alle Dateien um, die der Dateibezeichnung A*.TEX genügen in Dateien mit Dateinamen, die der Dateibezeichnung S*.TEX genügen.

A>REN B:NEWLIST=B:OLDLIST

Die Datei OLDLIST wird umbenannt in NEWLIST auf Laufwerk B. Da die zweite Laufwerksangabe B: durch die erste eingeschlossen wird, ist sie in diesem Falle nicht erforderlich.

Die oben angegebene Kommandozeile hat die gleiche Wirkung wie die folgenden:

A>REN B:NEWLIST=OLDLIST

oder

A>REN NEWLIST=B:OLDLIST

Das SAVE-Kommando

Syntax:

SAVE

Erklärung:

SAVE kopiert den Speicherinhalt in eine Datei. Zur Anwendung von SAVE ist zuerst das Kommando SAVE einzugeben und anschließend das Programm zu starten, das eine Datei in den Speicher liest. Bei Verlassen des Programms mit Warmstart wird zur SAVE-Routine verzweigt und diese verlangt sowohl die Eingabe einer Dateibezeichnung, in der der Speicherinhalt abgelegt werden soll, als auch die Anfangs- und Endadresse des zu sichernden Speicherbereiches.

Beispiel:

A>SAVE

Die SAVE-Routine wird aktiviert. Dann ist der Name des Programms einzugeben, der eine Datei in den Speicher liest.

A>DU dump.com

Als nächstes ist das Programm auszuführen.

#g0

Wird das Programm verlassen, dann versperrt SAVE die Rückkehr zum System und fordert vom Nutzer die Eingabe der Dateibezeichnung und der Grenzen für den zu sichernden Speicherbereich.

SCP3 SAVE R-BWS V1/0

Enter file (type RETURN to exit):dump2.com

Delete dump2.com? Y

Beginning hex address:100

Ending hex address: 400

Der Speicherinhalt von 100H bis 400H wird in die Datei DUMP2.COM kopiert.

Das SET-Kommando

Syntax:

SET [optionen]

SET d: [optionen]

SET dateibez [optionen]

Erklärung:

SET richtet den Paßwortschutz und die Zeiteintragungen für die Dateien ein. Auch die Dateiattribute Read-Write, Read-Only, DIR und SYS können mit SET gesetzt werden. Mit diesem Kommando kann die Diskette mit einem Kennsatz versehen und zu diesem Kennsatz ein Paßwortschutz eingerichtet werden.

Das SET-Kommando umfaßt Optionen, die das Diskettenverzeichnis, das Laufwerk oder eine Datei bzw. eine Menge von Dateien beeinflussen. Die Beschreibung des SET-Kommandos legt im einzelnen dar, welche der Kategorien beeinflusst wird.

Um die Zeiteintragung für die Dateien zu ermöglichen, ist das Diskettenverzeichnis zuvor mit INITDIR zu formatieren.

Datei-Attribute setzen

Syntax:

SET dateibez [optionen]

Erklärung:

Mit diesem Kommando werden die angegebenen Attribute für eine Datei oder eine Gruppe von Dateien gesetzt.

Option	Bedeutung
RO	setzt das Dateiattribut auf Read-Only.
RW	setzt das Dateiattribut auf Read-Write.
SYS	setzt das Dateiattribut auf SYS.
DIR	setzt das Dateiattribut auf DIR.
ARCHIVE=OFF	setzt das Archive-Attribut auf OFF. Das heißt, daß es kein Duplikat von der Datei gibt. PIP mit der Option [A] kann Dateien kopieren, deren Archive-Attribut auf OFF gesetzt wurde. PIP mit dieser Option verlangt eine mehrdeutige Dateibezeichnung und kopiert nur die Dateien, die seit dem letzten Kopieren mit PIP [A] erstellt oder verändert wurden. PIP setzt dann für jede erfolgreich kopierte Datei das Archive-Attribut auf ON.
ARCHIVE=ON	setzt das Archive-Attribut auf ON. Das heißt, daß die Datei archiviert wurde. Das Archive-Attribut kann durch SET oder durch PIP auf ON gesetzt werden, wenn eine Gruppe von Dateien mittels PIP mit Option [A] kopiert wird. SHOW und DIR zeigen das Archive-Attribut an.
F1=ON OFF	setzt oder löscht das vom Nutzer definierbare Dateiattribut F1.
F2=ON OFF	setzt oder löscht das vom Nutzer definierbare Dateiattribut F2.
F3=ON OFF	setzt oder löscht das vom Nutzer definierbare Dateiattribut F3.
F4=ON OFF	setzt oder löscht das vom Nutzer definierbare Dateiattribut F4.

Beispiele:

A>SET MYFILE.TEX [RO SYS]

Für die Datei MYFILE.TEX werden die Dateiattribute RO und SYS gesetzt.

A>SET MYFILE.TEX [RW DIR]

Mit diesem Kommando werden für die Datei MYFILE.TEX die Attribute RW und DIR gesetzt.

Laufwerks-Attribut bestimmen

Syntax:

SET {d:} {RO}

SET {d:} {RW}

Erklärung:

Dem angegebenen Laufwerk wird mit dem SET-Kommando entweder das Read-Only-(RO) oder das Read-Write-(RW) Attribut zugeordnet.

Ein Laufwerk, dem das RO-Attribut zugeordnet ist, kann nur zum Lesen von Dateien verwendet werden. Auf Disketten in diesem Laufwerk kann das PIP-Kommando keine Dateien kopieren, mit ERASE können keine Dateien gelöscht und mit RENAME keine Dateien umbenannt werden. Beim Eingeben von CTRL-C wird allen Laufwerken das RW-Attribut zugeordnet (Laufwerk Reset).

Beispiel:

A>SET B: [RO]

Das Laufwerk B wird als Read-Only-Laufwerk markiert.

Disketten-Label festlegen

Syntax:

SET {d:} [NAME=LABELNAM.TYP]

Erklärung:

Zuordnen eines Namens (Label) zu einer Diskette im angegebenen Laufwerk oder Standardlaufwerk.

SCP 3.0 bietet die Möglichkeit, für jede Diskette einen Verzeichniskennsatz anzulegen. Für die Zusammensetzung des Kennsatzes gelten die gleichen Regeln wie für den Namen und die Typbezeichnung einer Datei. Kennsätze erleichtern ein Katalogisieren und Auffinden von Disketten. Der Standard-Kennsatzname ist LABEL.

Beispiel:

A>SET [Name=DISK100]

Mit diesem Kommando wird der Verzeichniskennsatz DISK100 für die Diskette im Standardlaufwerk festgelegt.

Diskettenkennsatz ein Paßwort zuordnen

Syntax:

SET [PASSWORD=password]

SET [PASSWORD=<cr>

Erklärung:

Die erste Form des SET-Kommandos weist dem Diskettenkennsatz ein Paßwort zu. Die zweite Form des Kommandos nimmt den Paßwortschutz für den Kennsatz zurück.

Einer Diskette kann ein Paßwort zugeordnet werden. Solange die Diskette kein Paßwort besitzt, kann jeder Nutzer mit dem SET-Programm der Diskette andere Attribute zuweisen, und

der Eigentümer hat schließlich keinen Zugriff mehr. Wird dem Kennsatz jedoch ein Paßwort zugewiesen, dann muß dieses für jede andere den Kennsatz betreffende Funktion angegeben werden. SET verlangt jedesmal das Paßwort, wenn ein Paßwortschutz vorhanden ist.

Beispiele:

```
A>SET [PASSWORD=SECRET]
```

```
A>SET [PASSWORD=<cr>
```

Das erste Kommando weist dem Diskettenkennsatz das Paßwort SECRET zu. Das zweite Beispiel löscht das existierende Paßwort.

Beachte: Wird der Paßwortschutz für eine Diskette verwendet, dann ist das Paßwort sicher zu bewahren. Wird das Paßwort vergessen, dann geht der Zugriff auf die Diskette bzw. die Dateien verloren.

Paßwortschutz für Diskettendateien setzen oder rücksetzen

Syntax:

```
SET [PROTECT=ON]
```

```
SET [PROTECT=OFF]
```

Erklärung:

Die erste Form des SET-Kommandos setzt den Paßwortschutz für alle Dateien auf der Diskette. Es muß zuerst der Paßwortschutz gesetzt werden, bevor bestimmten Dateien ein Paßwort zugewiesen werden kann.

Das zweite Kommando setzt den Paßwortschutz für die Diskettendateien zurück.

Wurde dem Diskettenkennsatz ein Paßwort zugewiesen und die Option PROTECT gesetzt, dann können den einzelnen Dateien Paßwörter zugewiesen werden.

Mit dem SHOW-Kommando kann jederzeit überprüft werden, ob eine Diskette mit einem Paßwort geschützt ist oder nicht.

Zuordnen Paßwortschutz zu Diskettendateien

Syntax:

```
SET dateibez [PASSWORD=passwort]
```

Erklärung:

Diese Art des SET-Kommandos ordnet den mit dateibez definierten Dateien einen Paßwortschutz zu. Das Paßwort kann bis zu acht Zeichen lang sein. Kleinbuchstaben werden in Großbuchstaben umgewandelt.

In der Dateibezeichnung können auch die Ersatzzeichen * und ? verwendet werden. SET ordnet dann das Paßwort allen Dateien zu, die dieser Dateibezeichnung genügen.

Beachte: Die festgelegten Paßwörter für die Dateien sind zu notieren. Ohne Angabe des Paßwortes ist es nicht möglich, auf die geschützten Dateien zuzugreifen. Wird das Paßwort für den Diskettenkennsatz vergessen, dann kann auch der Paßwortschutz für die Diskette nicht wieder aufgehoben werden.

Beispiel:

```
A>SET MYFILE.TEX [PASSWORD=MYFIL]
```

Der Datei MYFILE.TEX wird das Paßwort MYFIL zugeordnet.

Art des Paßwortschutzes für geschützte Dateien setzen

Syntax:

```
SET dateibez [PROTECT=READ]
```

```
SET dateibez [PROTECT=WRITE]
```

SET dateibez [PROTECT=DELETE]

SET dateibez [PROTECT=NONE]

Erklärung:

Der Paßwortschutz von Dateien kann in vier verschiedenen Stufen durchgeführt werden. Die Schutzarten sind READ, WRITE, DELETE und NONE und werden in nachfolgender Tabelle beschrieben.

Arten des Paßwortschutzes

Option	Schutz
READ	Das Paßwort ist erforderlich zum Lesen, Kopieren, Schreiben, Löschen oder Umbenennen der Datei.
WRITE	Das Paßwort muß zum Schreiben, Löschen oder Umbenennen der Datei angegeben werden. Das Paßwort ist zum Lesen der Datei nicht erforderlich.
DELETE	Das Paßwort ist nur zum Löschen oder Umbenennen der Datei erforderlich. Zum Lesen oder Modifizieren der Datei ist ein Paßwort nicht erforderlich.
NONE	Der Datei wurde kein Paßwort zugeordnet. Existiert ein Paßwort, dann kann es mit dieser Angabe gelöscht werden.

Beispiel:

B>SET *.TEX [PASSWORD=SECRET,PROTECT=WRITE]

Allen Dateien vom Typ TEX auf Laufwerk B wird das Paßwort SECRET zugeordnet. Mit der Schutzart WRITE ist jede dieser Dateien gegen unberechtigtes Schreiben und Verändern geschützt.

Standard-Paßwort bestimmen

Syntax:

SET [DEFAULT=passwort]

Erklärung:

Mit dem SET-Kommando für das Standard-Paßwort wird für die Einschaltdauer des Computers das vorgegebene Paßwort als Standard-Paßwort verwendet. Auf alle Dateien, die mit demselben Paßwort wie das Standard-Paßwort geschützt sind, kann zugegriffen werden. Wird bei der Dateierstellung kein Paßwort festgelegt und die PROTECT-Option ist auf ON gesetzt, so wird diesen Dateien das Standard-Paßwort als ihr Paßwort zugewiesen.

Beispiel:

A>SET [DEFAULT=dd]

Das System verwendet dd als Passwort für die paßwortgeschützten Dateien, wenn kein anderes Paßwort für die Dateien eingegeben wird.

Art der Zeit- und Datumskennung

Syntax:

SET [CREATE=ON]

SET [ACCESS=ON]

SET [UPDATE=ON]

Erklärung:

Mit diesen Kommandos ist es möglich, Zeit und Datum des Erstellens und Aktualisierens oder des letzten Zugriffs und Aktualisierens der Datei abzuspeichern.

- [CREATE=ON] setzt die CREATE-Zeitkennzeichnung für die Diskette im angegebenen oder Standardlaufwerk. Um die Erstellungszeit einer Datei aufzuzeichnen, muß die Option CREATE vor dem Erstellen der Datei eingegeben werden.
- [ACCESS=ON] setzt die ACCESS-Zeitkennzeichnung für die Diskette im angegebenen oder Standardlaufwerk. Die Optionen ACCESS und CREATE schließen sich gegenseitig aus; nur eine von beiden kann aktiv sein. Wird die ACCESS-Zeitkennzeichnung für eine Diskette gesetzt, die zuvor die CREATE-Zeitkennzeichnung hatte, dann wird die CREATE-Zeitkennzeichnung automatisch außer Kraft gesetzt.
- [UPDATE=ON] setzt die UPDATE-Zeitkennzeichnung für die Diskette im angegebenen oder Standardlaufwerk. Mit UPDATE wird die Zeit eingetragen, zu der die Datei letztmalig bearbeitet wurde.

Um die Zeiteintragungen zu ermöglichen, muß zuerst mit dem Kommando INITDIR das Diskettenverzeichnis eingerichtet werden.

Obwohl es drei Arten der Zeit/Datumeintragung gibt, können nur zwei davon einer gegebenen Datei zugeordnet werden. Es kann zwischen CREATE und ACCESS für Dateien auf einer bestimmten Diskette ausgewählt werden.

Werden sowohl UPDATE als auch CREATE gesetzt, dann werden auch beide Eintragungen geändert. Das resultiert daraus, daß beim Editieren einer Datei nicht die Originaldatei aktualisiert, sondern eine neue Version mit dem Namen der Originaldatei erstellt wird.

Beispiel:

```
A>SET [ACCESS=ON]
```

Das DIR-Kommando mit der Option [FULL] zeigt dann folgendes an:

```
B>DIR [FULL]
```

Directory for Drive B:

Name	Bytes	Recs	Attributes	Prot	Update	Access
DREI .TEX	10k	76	Dir RW	None		02/11/86 14:28
EINS .TEX	9k	71	Dir RW	None		09/03/86 12:16
ZWEI .TEX	12k	95	Dir RW	None		09/03/86 09:36

Die Zugriffszeiteintragung gibt Auskunft darüber, wann die Datei letztmalig angezeigt oder bearbeitet wurde. Die Anzeige eines Dateinamens in der Verzeichnisliste wird nicht als Zugriff auf die Datei gewertet.

```
A>SET [CREATE=ON, UPDATE=ON]
```

Mit DIR wird folgende Ausgabe erreicht:

```
B>DIR [FULL]
```

Directory for Drive B:

Name	Bytes	Recs	Attributes	Prot	Update	Create
EWALD .DAT	109k	873	Dir RW	None	01/31/86 13:25	02/11/86 11:28
FRANZ .DAT	59k	475	Dir RW	None	03/22/86 07:36	03/09/86 10:16
GUSTAV .DAT	76k	608	Dir RW	None	03/22/86 08:12	03/09/86 09:36

Weitere Beispiele für das SET-Kommando

Beispiele:

```
A>SET*.COM [SYS,RO, PASS=123,PROT=READ]
```

Mit diesem Kommando wird der größtmögliche Schutz für alle COM-Dateien auf Laufwerk A erreicht. Da als Art des Paßwortschutzes READ gewählt wurde, kann auf keine COM-Datei ohne Angabe des Paßwortes 123 zugegriffen werden, außer wenn als Standard-Paßwort 123 gewählt wurde. Auch wenn das richtige Paßwort eingegeben wurde, kann in die Datei nicht geschrieben werden, da sie als Read Only markiert wurde.

```
A>SET *.COM [RW,PROTECT=NONE,DIR]
```

Dieses Kommando hebt den Schutz und die Zugriffsattribute für die vom vorhergehenden Beispiel angesprochenen COM-Dateien auf. Nach Ausführung dieses Kommandos gibt es keinen Paßwortschutz mehr, die Dateien vom Typ COM können gelesen und geschrieben werden und werden als DIR-Dateien markiert.

Das SETDEF-Kommando

Syntax:

```
SETDEF {d: {,d: {,d: {,d:}}}} [[ TEMPORARY = d: ] |  
[ ORDER = (typ {,typ}) ] ]
```

```
SETDEF [DISPLAY | NO DISPLAY]
```

```
SETDEF [PAGE | NOPAGE]
```

Erklärung:

SETDEF ermöglicht dem Nutzer das Anzeigen oder Definieren von bis zu vier Laufwerken für die Programmsuch-Reihenfolge, des Laufwerkes für Arbeitsdateien und der Suchreihenfolge für den Dateityp. Die SETDEF-Definitionen beeinflussen nur das Laden von Programmen und/oder die Ausführung von SUBMIT-(SUB)-Dateien. Mit SETDEF können auch DISPLAY- und PAGE-Modi ein- und ausgeschaltet werden. Ist der DISPLAY-Modus an, dann zeigt das System den Herkunftsort und Namen der geladenen Programme oder der ausgeführten SUB-Dateien an. Beim eingeschalteten PAGE-Modus wird die Bildschirmanzeige nach jeder voll ausgefüllten Bildschirmseite angehalten, damit diese gelesen werden kann. Die Anzeige wird mit Betätigen einer beliebigen Taste fortgesetzt.

Gewöhnlich sucht das System nur auf dem aktuellen Laufwerk nach Dateien, wenn kein anderes Laufwerk angegeben wird. Mit dem SETDEF-Kommando kann diese Suche auch auf andere Laufwerke erweitert werden, allerdings nur zum Ausführen von COM- und SUB-Dateien.

Anzeige der Suchreihenfolge zum Programmladen

Syntax:

```
SETDEF
```

Erklärung:

Mit dieser Form des SETDEF-Kommandos werden die Diskettensuchreihenfolge, das Laufwerk für die Arbeitsdateien und die Dateityp-Suchreihenfolge angezeigt.

Zuordnung des Laufwerkes für Arbeitsdateien

Syntax:

```
SETDEF [TEMPORARY=D:]
```

Erklärung:

Alle Arbeitsdateien werden nicht auf dem aktuellen, sondern auf dem im SETDEF-Kommando angegebenen Laufwerk abgelegt.

Beispiel:

```
A>SETDEF [TEMPORARY=C:]
```

Laufwerk C wird als Laufwerk für die Arbeitsdateien benutzt.

Definieren der Reihenfolge zum Durchsuchen der Laufwerke**Syntax:**

```
SETDEF {d: {,d: {,d: {d:}}}}
```

Erklärung:

Diese Form des SETDEF-Kommandos legt die Laufwerke fest, die das System zum Laden und Ausführen von Programm- und/oder Submit-Dateien durchsucht. Standardmäßig sucht SCP 3.0 nur auf dem Standardlaufwerk.

Beachte: Für d: kann ein * angegeben werden, um das Standardlaufwerk in die Suchreihenfolge aufzunehmen.

Beispiel:

```
A>SETDEF C:,*
```

Dem System wird mitgeteilt, daß ein Programm auf Laufwerk C zu suchen ist, und wenn es dort nicht gefunden wird, dann auf dem Standardlaufwerk.

Bestimmen der Reihenfolge für das Suchen von Dateitypen**Syntax:**

```
SETDEF [ORDER=(typ {,typ})]
```

wobei typ = COM oder SUB

Dieses Kommando legt die Suchreihenfolge für den Dateityp fest, die vom System zum Laden von Programmen benutzt wird. Der Dateityp darf nur COM oder SUB sein. SCP 3.0 sucht standardmäßig nur nach COM-Dateien.

Beispiel:

```
A>SETDEF [ORDER=(SUB,COM)]
```

Damit wird das System angewiesen, zunächst nach einer SUB-Datei zu suchen, um diese auszuführen. Wird keine SUB-Datei gefunden, dann wird nach einer COM-Datei gesucht.

System-Anzeigemodus setzen**Syntax:**

```
SETDEF [DISPLAY | NO DISPLAY]
```

Erklärung:

Mit diesem Kommando wird der System-Anzeigemodus gesetzt oder rückgesetzt. Standard ist, daß der System-Anzeigemodus nicht gesetzt ist. Wurde der System-Anzeigemodus gesetzt, dann gibt SCP 3.0 folgende Informationen über eine Programmdatei aus, bevor diese Datei geladen wird: Laufwerk, Dateiname, Dateityp (sofern vorhanden) und die Nummer des Nutzerbereiches (wenn es nicht der Standard-Nutzerbereich ist).

Beispiel:

```
A>SETDEF [DISPLAY]
```

Schaltet den System-Anzeigemodus ein. Danach zeigt das System den Namen und den Herkunftsort der geladenen Programme oder der ausgeführten SUBMIT-Dateien an. Wird zum Beispiel nach dem Einschalten des System-Anzeigemodus das PIP-Programm geladen, dann gibt das System folgendes aus:

```
A>PIP
A:PIP      COM
SCP3 PIP R-BWS V1/0
*
```

Die Datei PIP wurde vom Laufwerk A aus dem aktuellen Nutzerbereich geladen. Ist 0 nicht die Nummer des aktuellen Nutzerbereiches und PIP nicht im aktuellen Nutzerbereich vorhanden, dann zeigt das System den Herkunftsort wie folgt an:

```
4A>PIP
A:PIP      COM      (USER 0)
SCP3 PIP R-BWS V1/0
*
```

PIP.COM wurde aus dem Nutzerbereich 0 von Laufwerk A geladen. Dieser Modus bleibt erhalten bis zur Eingabe von:

```
SETDEF [NO DISPLAY]
```

mit der der System-Anzeigemodus DISPLAY abgeschaltet wird.

Nach dem SCP-3.0-Start ist der DISPLAY-Modus ausgeschaltet.

System-PAGE-Modus ein- oder ausschalten

Syntax:

```
SETDEF [PAGE | NO PAGE ]
```

Erklärung:

Mit diesem Kommando wird der System-PAGE-Modus ein- oder ausgeschaltet. Ist der System-PAGE-Modus an, dann wird vom System die Bildschirmanzeige nach jeder voll ausgefüllten Bildschirmseite angehalten. Die Bildschirmanzeige wird nach dem Betätigen einer beliebigen Taste fortgesetzt. Standard ist der eingeschaltete System-PAGE-Modus.

Beispiel:

```
A>SETDEF [NO PAGE]
```

Die Bildschirmausgabe erfolgt ohne Unterbrechung, auch wenn der angezeigte Text größer als die Bildschirmgröße ist.

Das SHOW-Kommando

Syntax:

```
SHOW {d:} {[SPACE | Label | USERS | DIR | DRIVE]}
```

Erklärung:

Das SHOW-Kommando zeigt folgende Laufwerkinformationen an:

- Zugriffsart und freien Platz auf der Diskette
- Diskettenkennsatz
- aktuelle Nutzerbereichsnummer
- Anzahl der Dateien für jeden Nutzerbereich auf der Diskette
- Anzahl der freien Verzeichniseinträge für die Diskette
- Laufwerkparameter

Anzeige des Zugriffsstatus und des freien Platzes auf der Diskette

Syntax:

SHOW {d:} {[SPACE]}

Erklärung:

Angezeigt wird die Laufwerksbezeichnung, der Zugriffsstatus und der freie Platz auf der Diskette in Kilobyte für die angegebenen Laufwerke. Das Kommando SHOW ohne weitere Angaben zeigt die Informationen für alle im System zugewiesenen Laufwerke an.

Beispiele:

A>SHOW B:

B:RW, Space: 444k

A>SHOW

A:RW, Space: 6k

B:RW, Space: 444k

Das erste Beispiel zeigt für Laufwerk B den Zugriffsstatus Read-Write sowie einen freien Platz von 444k Byte an. Das zweite Beispiel zeigt die Informationen für die Laufwerke A und B an.

Anzeige Diskettenkennsatz

Syntax:

SHOW {d:} [LABEL]

Erklärung:

Es werden alle Informationen über den Diskettenkennsatz und seine Attribute angezeigt.

Beispiel:

A>SHOW B:[LABEL]

Das Kommando zeigt für das Laufwerk B folgendes an:

Label for drive B:

Directory	Passwds	Stamp	Stamp	Label	Created	Label	Updated
Label	Reqd	Create	Update				

UWESDISK.	on	on	on	04/07/86	11:25	04/09/86	15:06
-----------	----	----	----	----------	-------	----------	-------

In der ersten Spalte wird der Diskettenkennsatz angezeigt und in der zweiten Spalte wird angezeigt, daß der Paßwortschutz für das Laufwerk gesetzt ist.

Wie im SET-Kommando beschrieben, kann jede Datei bis zu zwei Zeiteintragungen besitzen. Die erste dieser Zeitmarkierungen ist entweder Datum und Zeit der Dateierstellung oder Datum und Zeit des letzten Zugriffs auf die Datei. Zugriff bedeutet Lesen oder Schreiben der Datei. Die dritte Spalte der Anzeige enthält den Typ der Zeitmarkierung und ob diese gesetzt ist. Im Beispiel erhalten neue Dateien die Eintragung der Erstellungszeit, wie das aus der Spalte Stamp Create ersichtlich ist (on).

Die vierte Spalte zeigt den Status der zweiten Zeiteintragung an, der Update-Zeitmarkierung. Die Update-Zeiteintragung gibt Datum und Uhrzeit der letzten Aktualisierung der Datei an, d. h., den Zeitpunkt, als letztmalig auf die Datei geschrieben wurde. Im Beispiel ist die Update-Zeitmarkierung eingeschaltet (on).

Neben der Anzeige von Paßwortschutz und der aktiven Zeitmarkierung auf dem Laufwerk werden ferner Datum und Uhrzeit der Erstellung und letzten Änderung des Kennsatzes angezeigt.

Anzeige der Nutzerbereiche auf der Diskette

Syntax:

SHOW {d:} [USERS]

Erklärung:

Es werden die Nummern des aktuellen Nutzerbereiches und aller vorhandenen Nutzerbereiche auf der Diskette angezeigt, dazu die Anzahl der Dateien in den einzelnen Nutzerbereichen.

Beispiel:

A>SHOW [USERS]

```
Active User:      1
Active Files:    0    2    3    4
A: # of files:   95   40    1   26
A: Number of time/date directory entries: 32
A: Number of free directory entries:    350
```

A>

Anzeige der freien Einträge im Diskettenverzeichnis

Syntax:

SHOW {d:} [DIR]

Erklärung:

Mit diesem Kommando wird die Anzahl freier Verzeichniseinträge auf dem angegebenen Laufwerk ausgegeben.

Beispiel:

A>SHOW C:[DIR]

C: Number of free directory entries: 22

A>

Dieses Kommando gibt an, daß auf Laufwerk C nur 22 freie Verzeichniseinträge vorhanden sind.

Anzeige der Disketteneigenschaften

Syntax:

SHOW {d:} [DRIVE]

Erklärung:

Diese Form des SHOW-Kommandos zeigt die Eigenschaften des angegebenen Laufwerkes an.

Beispiel:

A>SHOW [DRIVE]

Damit werden die Eigenschaften von Laufwerk A angezeigt.

```
A : Drive Characteristics
3,600 : 128 Byte Record Capacity
450 : Kilobyte Drive Capacity
96 : 32 Byte Directory Entries
96 : Checked Directory Entries
128 : Records/Directory Entry
```

16 : Records/Block
 48 : Sectors/Track
 512 : Bytes/Physical Record

Das SUBMIT-Kommando

Syntax:

SUBMIT {dateibez} {argument} . . . {argument}

Erklärung:

Mit SUBMIT ist es möglich, eine Reihe (batch) von Kommandos aus einer SUBMIT-Datei (eine Datei mit dem Dateityp SUB) abzuarbeiten.

Normalerweise werden die Kommandos zeilenweise eingegeben. Wird eine Folge von Kommandos zu verschiedenen Zeiten benötigt, dann ist es angebracht, diese Kommandos durch ein SUBMIT-Kommando zusammenzufassen. Dazu ist eine Datei zu eröffnen und die Kommandos in diese Datei zu erfassen. Die Datei erhält einen wählbaren Dateinamen und den vorgeschriebenen Dateityp SUB. Wird das SUBMIT-Kommando angegeben, dann liest SUBMIT die durch dateibez bezeichnete Datei und bereitet sie für eine Interpretation durch das SCP 3.0 vor. Die Kommandos werden genauso ausgeführt, als würden sie über die Tastatur eingegeben.

Die SUB-Datei wird mit einem Texteditor erfaßt. Sie kann SCP-3.0-Kommandos, geschachtelte SUBMIT-Kommandos und Eingabedaten für ein SCP-3.0-Kommando oder Programm enthalten.

Der SUB-Datei können zur Ausführung Parameter übergeben werden. Jeder eingegebene Parameter wird einer Variablen in der SUB-Datei zugeordnet. Der erste Parameter ersetzt die erste Variable $\alpha 1$ in der Datei, der zweite Parameter ersetzt die zweite Variable $\alpha 2$ usw. bis Variable $\alpha 9$. Enthält z. B. die Datei START.SUB folgende Kommandos:

```
ERA  $\alpha 1$ .BAK
DIR  $\alpha 1$ 
PIP  $\alpha 1$ =A: $\alpha 2$ .COM
```

und wird SUBMIT mit folgender Eingabe in der Kommandozeile aufgerufen:

```
A>SUBMIT START SAM TEX
```

dann wird jedesmal anstelle $\alpha 1$ der Parameter SAM und anstelle $\alpha 2$ TEX eingesetzt. Dazu erstellt SUBMIT eine $\alpha \alpha \alpha$.SUB-Datei, in der die korrekten Parameter eingesetzt sind:

```
ERA SAM.BAK
DIR SAM
PIP SAM=A:TEX.COM
```

Werden weniger Parameter in der SUBMIT-Kommandozeile angegeben als in der SUB-Datei Variablen existieren, dann werden die überzähligen Variablen nicht in die Kommandos aufgenommen.

Werden mehr Parameter in das SUBMIT-Kommando eingegeben als Variablen in die SUB-Datei, dann werden die überzähligen Parameter ignoriert.

Um das Zeichen α echt in die SUB-Datei aufzunehmen, ist es zweimal anzugeben $\alpha \alpha$. SUBMIT ersetzt dann diese beiden Zeichen bei der Parametersubstitution durch ein einfaches Zeichen α . Enthält z. B. die Datei AA.SUB die Zeile

```
MAC  $\alpha 1$   $\alpha \alpha$   $\alpha 2$ 
```

und das SUBMIT-Kommando wird wie folgt angegeben

```
A>SUBMIT AA ZZ SZ
```

dann enthält die überarbeitete Datei folgende Zeile

```
MAC ZZ αSZ
```

Programmeingaben in einer SUB-Datei

Eine SUB-Datei kann Zeilen zur Programmeingabe enthalten. Eine Programmeingabe wird gekennzeichnet durch ein vorangestelltes Zeichen kleiner als „<“, wie das im folgenden Beispiel dargestellt wird:

```
PIP
<B:=*.ASM
<CON:=DUMP.ASM
<
DIR
```

Die drei Zeilen nach dem PIP-Aufruf sind Eingaben für das PIP-Kommando. Die dritte Zeile enthält nur das Zeichen <, was einen Wagenrücklauf <cr> bedeutet. Damit wird PIP beendet und es wird zum System SCP 3.0 zurückgekehrt und das abschließende Kommando DIR ausgeführt.

Wird das Programm beendet bevor alle Eingaben verwendet wurden, dann ignoriert SUBMIT die restlichen Eingabezeilen und gibt folgende Meldung aus:

```
Warning: Program input ignored
```

Verlangt das Programm mehr Eingaben als in der SUB-Datei vorhanden, dann werden diese Eingaben von der Tastatur erwartet.

In eine SUB-Datei können auch unter Beachtung der üblichen Regeln Steuerzeichen eingegeben werden, d. h., es muß das Zeichen ^ angegeben werden gefolgt von einem Buchstaben, der in das Steuerzeichen konvertiert wird. Um aber das Zeichen ^ selbst in die Datei zu erfassen, muß es doppelt eingegeben werden ^^ . Diese Zeichenkette wird wie α α auch in das einzelne Zeichen ^ bzw. α überführt.

Die SUB-Datei

Die SUB-Datei kann folgende Zeilentypen enthalten:

- * Ein gültiges SCP-3.0-Kommando
- * Ein gültiges SCP-3.0-Kommando mit SUBMIT-Parametern
- * Eine Dateneingabezeile
- * Eine Programmeingabezeile mit Parametern (α0 bis α9)

Die Zeilen dürfen 135 Zeichen nicht überschreiten.

Beispiel:

Die folgenden Zeilen verdeutlichen die Vielfalt an Zeilen, die in eine SUB-Datei eingegeben werden können:

```
DIR
DIR *.BAK
MAC α1 α α α4
PIP LST:=α1.PRN[Tα2 α3 α5]
DIR *.ASM
PIP
```

```
<B:=*.ASM
<CON:=DUMP.ASM
<
```

DIR B:

Ausführung eines SUBMIT-Kommandos

Syntax:

SUBMIT

SUBMIT dateibez

SUBMIT dateibez argument . . . argument

Wird nur SUBMIT ohne weitere Angaben eingegeben, dann fordert das System die restlichen Eingaben ab. Es können dann noch Dateibezeichnung und Parameter eingegeben werden.

Beispiele:

A>SUBMIT

Das System gibt eine Anforderung aus. Darauf sind die Dateibezeichnung und die Parameter wie folgt einzugeben:

Enter File to Submit: START B TEX

Ein anderes Beispiel wäre:

A>SUBMIT SUBA

Ein weiteres Beispiel mit Angabe der Parameter ist

A>SUBMIT AA ZZ SZ

wobei AA die Datei AA.SUB darstellt; ZZ und SZ sind die Parameter, die überall für α_1 und α_2 in der Datei AA.SUB eingesetzt werden.

Die Kaltstartdatei PROFILE.SUB

Nach jedem Einschalten oder RESET des Computers sucht SCP 3.0 eine spezielle SUB-Datei mit dem Namen PROFILE.SUB, um diese auszuführen. Existiert eine solche Datei nicht, dann geht SCP 3.0 zur normalen Arbeit über. Ist die Datei PROFILE.SUB vorhanden, dann werden ihre Kommandos vom System ausgeführt. Eine solche Datei erweist sich als sehr nützlich, wenn vor jeder weiteren Arbeit am Computer eine Reihe von Kommandos auszuführen sind, um die benötigte Systemeinstellung herzustellen. Soll z.B. sichergestellt werden, daß vor der Eingabe eines Kommandos das aktuelle Datum und die aktuelle Uhrzeit im Computer gespeichert werden, dann kann mit einem Texteditor die Datei PROFILE.SUB erstellt werden, die das DATE-Kommando wie folgt enthält:

DATE SET

Damit wird bei jedem Systemstart das DATE Kommando ausgeführt, d. h., Datum und Uhrzeit sind einzugeben. Mit dieser Möglichkeit wird garantiert, daß vor der eigentlichen Arbeit eine abgeschlossene Folge von Kommandos abgearbeitet wird.

Das TYPE-Kommando

Syntax:

TYPE {dateibez {[PAGE | NOPAGE]}}

Erklärung:

Das Kommando TYPE zeigt den Inhalt einer ASCII-Zeichen-Datei auf dem Bildschirm an. Die Option PAGE gibt die Informationen im Page-Mode aus, d. h., die Bildschirmausgabe

wird jedesmal nach Darstellung von n Zeilen angehalten. Die Zeilenzahl n entspricht der Systemvoreinstellung auf 24 Zeilen und kann mit dem DEVICE-Kommando geändert werden. Mit Betätigen einer beliebigen Taste wird mit der Ausgabe der nächsten n Zeilen fortgefahren. Die Eingabe von CTRL-C bewirkt den Rücksprung zum System. PAGE ist die Standardeinstellung.

Wird NO PAGE angegeben, dann erfolgt die Ausgabe ohne Unterbrechung.

Wird nur TYPE ohne Dateibezeichnung angegeben, dann erscheint vom System folgende Aufforderung:

Enter filename:

Daraufhin kann der Dateiname eingegeben werden.

TAB-Zeichen in der durch die Dateibezeichnung angegebenen Datei werden auf die jeweils achte Zeile auf dem Bildschirm ausgerichtet.

Zu einem beliebigen Zeitpunkt kann die Bildschirmausgabe mit CTRL-S unterbrochen und mit CTRL-Q fortgesetzt werden.

Die Rückkehr zum System kann mit CTRL-C erfolgen.

Es ist dafür Sorge zu tragen, daß die Dateibezeichnung den Namen einer Textdatei beinhaltet.

Ist die angegebene Datei auf dem Laufwerk nicht vorhanden, dann gibt TYPE folgende Meldung aus:

No File

Um eine Datei sowohl auf dem Bildschirm als auch auf den Drucker auszugeben, ist vor der Eingabe der TYPE-Kommandozeile CTRL-P einzugeben. Die Ausgabe auf den Drucker wird beendet, wenn ein zweites CTRL-P eingegeben wird. Das TYPE-Kommando zeigt den Inhalt einer Datei auf dem Bildschirm an, bis dieser gefüllt ist. Dann wird angehalten und die Anzeige nach Betätigen einer beliebigen Taste fortgesetzt.

Beispiele:

A>TYPE MYPROG.PLI

Es wird der Inhalt der Datei MYPROG.PLI auf dem Bildschirm angezeigt, und zwar immer 24 Zeilen.

A>TYPE B:THISFILE [NO PAGE]

Es wird der Inhalt der Datei THISFILE von Laufwerk B auf dem Bildschirm angezeigt.

Das USER-Kommando

Syntax:

USER {nummer}

Erklärung:

Das USER-Kommando stellt die aktuelle Nutzernummer ein. Nach dem Start des SCP 3.0 ist der Nutzerbereich 0 eingestellt. Mit dem USER-Kommando kann einer der Nutzerbereiche 0 bis 15 als aktueller Nutzerbereich eingestellt werden.

SCP 3.0 ordnet jeder Datei die zugehörige Nutzerbereichsnummer zu. Gewöhnlich kann man nur auf Dateien zugreifen, die im aktuellen Nutzerbereich liegen. Eine Ausnahme bilden Dateien im Nutzerbereich 0, denen das Datei-SYS-Attribut zugeordnet wurde. Auf diese Dateien kann auch aus jedem anderen Nutzerbereich zugegriffen werden.

Beispiele:

A>USER

Das System fordert die Nutzerbereichsnummer wie folgt ab:

Enter User#:5

5A>

Der aktuelle Nutzerbereich auf Laufwerk A ist jetzt 5.

A>USER 3

3A>

Dieses Kommando ändert die aktuelle Nutzerbereichsnummer auf 3.

Anhang

SCP-3.0-Meldungen

Die Meldungen können verschiedenen Ursprung haben. Das Betriebssystem SCP 3.0 kann Fehlermeldungen anzeigen, wenn das BDOS einen Fehlercode zurückgibt. Ferner kann SCP 3.0 auch Meldungen ausgeben, wenn in einer Kommandozeile Fehler aufgetreten sind. Jede mit SCP 3.0 ausgelieferte Routine besitzt ihre eigenen Meldungen. Die nachfolgende Tabelle liefert eine Übersicht über die SCP-3.0-Meldungen und die Meldungen der Routinen.

Wird aber ein Anwenderprogramm abgearbeitet, dann können andere als die hier angegebenen Meldungen auftreten. In diesem Fall müssen die Erklärungen für die Meldungen den Beschreibungen des Anwenderprogramms entnommen werden.

Den Meldungen in der nachfolgenden Tabelle kann der Begriff ERROR: vorangehen. Einige andere werden durch die Dateibezeichnung der gerade bearbeiteten Datei ergänzt. In einigen Fällen wird der Fehler in einer Eingabezeile direkt markiert. Einigen Meldungen folgt eine zusätzliche Zeile mit INPUT:, Option: oder DRIVE:, gefolgt von der speziellen Fehlermeldung.

Meldung	Bedeutung
Assign a password to the file.	SET. Für diese Datei wurde ein Paßwort-Modus gewählt, aber kein Paßwort zugewiesen.
Auxiliary device redirection not implemented.	GET und PUT. AUXIN und AUXOUT können nicht auf eine Datei umgelenkt werden.
BAD close.	SAVE. Beim Versuch die Datei zu schließen ist ein Fehler aufgetreten. Möglicherweise deshalb, weil die Datei schreibgeschützt ist.
Bad Logical Device Assignment.	DEVICE. Nur die folgenden logischen Geräte sind gültig: CONIN:, CONOUT:, AUXIN:, AUXOUT: und LST:.
BAD PARAMETER	PIP. Das PIP-Kommando enthält einen ungültigen Parameter. Die Eingabe muß wiederholt werden.
Bad password.	RENAME. Das vom Nutzer angegebene Paßwort ist falsch.
Baud rate cannot be set for this device.	DEVICE. Nur bei Geräten mit dem SOFT-BAUD. Attribut kann deren Baudrate geändert werden. Um die Attribute des physischen Gerätes zu ermitteln ist das Kommando DEVICE phys. Gerät einzugeben.

Meldung	Bedeutung
CANNOT CLOSE: Cannot close file. CANNOT CLOSE FILE. CANNOT CLOSE DESTINATION FILE – dateibez	LIB, LINK, PIP, SUBMIT. Eine Ausgabedatei kann nicht angeschlossen werden. Das kann auftreten, wenn die Diskette vor Beenden des Programms aus dem Laufwerk genommen wird.
Cannot delete file.	GENCOM. SCP 3.0 kann eine Datei nicht löschen. Es ist zu überprüfen, ob die COM-Datei schreibgeschützt (Read-Only) oder paßwortgeschützt ist.
Cannot have both create and access time stamps.	SET. SCP 3.0 unterstützt die Eintragung entweder bei Erstellen oder bei Zugriff auf die Datei, aber nicht beide gleichzeitig.
Cannot label a drive with a file referenced.	SET. SET erlaubt keine Vermischung von Dateien und Laufwerken.
Cannot redirect from BIOS.	GET, PUT. Diese Meldung wird nur dann als eine Warnung angezeigt, wenn das System ein ungültiges BIOS besitzt.
Cannot set both RO and RW	SET. Eine Datei kann nicht gleichzeitig Read-Only und Read-Write sein.
Cannot set both SYS and DIR.	SET. Eine Datei kann nur eines der beiden Attribute SYS und DIR besitzen.
CAN'T DELETE TEMP FILE	PIP. Es existiert bereits eine Arbeitsdatei α α α mit dem Read-Only-Attribut. Mit dem SET-Kommando muß zuerst diese Datei das Read-Write-Attribut erhalten, dann ist sie zu löschen.
Close error.	XREF. Vor dieser Meldung steht der Dateiname XRF. Die Diskette könnte vor Beenden des Programms entnommen worden sein.
CHECKSUM ERROR. Checksum error.	PIP. Ein Prüfsummen-Fehler ist aufgetreten. Der Satz, der diesen Fehler erzeugt hat, muß korrigiert werden. Die Datei muß möglicherweise neu erzeugt werden.
Close operation failed.	COPYSYS. Es gab Schwierigkeiten die Datei am Ende des Kopierens zu schließen.

Meldung	Bedeutung
Closing file HELP.DAT Closing file HELP.HLP	HELP. Während der Bearbeitung von HELP.DAT oder HELP.HLP mit dem Kommando HELP ist ein Fehler aufgetreten.
COM file found and NULL option.	GENCOM. Die Option NULL bedeutet, daß nur die RSX-Module geladen werden müssen und keine COM-Datei.
.COM file required	DIR, ERA, REN, TYP. Die Optionen in der Kommandozeile erfordern die Unterstützung durch eine transiente COM-Datei und diese kann durch SCP 3.0 auf der Diskette nicht gefunden werden.
COMMON ERROR:	LINK. Ein nicht definierter Common-Block wurde selektiert.
CORRECT ERROR, TYPE RETURN OR CTRL-Z	PIP. Ein Prüfsummen-Fehler ist bei der Übertragung einer Hex-Datei aufgetreten. Die Hex-Datei mit dem Prüfsummen-Fehler muß korrigiert werden, möglicherweise durch Neuerstellen der Hex-Datei.
SCPLDR error: failed to open SCP3.SYS	SCPLDR. Die Systemdatei SCP3.SYS fehlt.
SCPLDR error: failed to read SCP3.SYS	SCPLDR. Ein Fehler ist beim Lesen der Datei SCP3.SYS aufgetreten.
SCP3 Error on d: DISK I/O BDOS Function = xx File = dateibez	SCP 3.0 gibt die vorstehende Meldung aus, wenn die Diskette defekt oder falsch formatiert ist.
SCP3 Error on d: Invalid Drive BDOS Function = xx File = dateibez	Diese Meldung wird von SCP 3.0 ausgegeben, wenn keine Diskette im Laufwerk liegt, das Laufwerk nicht geschlossen ist oder die Netzspannung fehlt. Diese Meldung wird auch ausgegeben, wenn das angegebene Laufwerk im System nicht vorhanden ist.
SCP3 Error on d: Read/Only Disk BDOS Function = xx File = dateibez	SCP 3.0 erlaubt kein Löschen, Umbenennen, Aktualisieren oder Attribut-Setzen für eine Datei, die sich in einem Laufwerk mit dem Attribut Read-Only befindet. Mit dem SET-Kommando ist das Laufwerk-Attribut auf Read-Write zu setzen.

Meldung	Bedeutung
SCP3 Error on d: Read/Only File BDOS Function = xx File = dateibez	SCP 3.0 erlaubt nicht eine Datei mit dem Read-Only-Attribut zu löschen, umzubenennen, zu aktualisieren oder ein Attribut zuzuweisen. Mit dem SET-Kommando ist das Dateiattribut auf Read-Write zu setzen.
Date and Time Stamping Inactive.	DIR. Die Option DATE wurde angegeben, aber das Diskettenverzeichnis ist nicht für die Eintragung von Datum und Zeit initialisiert.
DESTINATION IS R/O, DELETE (Y/N)	PIP. Die im PIP-Kommando angegebene Zielfeile existiert und besitzt das Attribut Read-Only. Wird ein Y eingegeben, dann wird die Zielfeile vor dem Kopieren gelöscht. Wird ein N eingegeben, dann gibt PIP die Meldung **NOT DELETED** aus und bricht das Kopieren ab.
Device Reassignment Not Supported. Enter new assignment or hit RETURN.	DEVICE. Die Gerätezuordnung ist ungültig.
Directory already re-formatted.	INITDIR. Das Verzeichnis beinhaltet schon Eintragungen für Datum und Zeit.
Directory full DIRECTORY FULL	SUBMIT. Auf dem Laufwerk für Arbeitsdateien ist im Verzeichnis kein Platz mehr frei, um die für die Ausführung von SUBMIT-Dateien notwendige Arbeitsdatei auf diesem Laufwerk abzulegen. Mit dem Kommando SETDEF kann das Laufwerk für Arbeitsdateien neu festgelegt werden. Mit dem Kommando ERASE können nicht mehr benötigte Dateien gelöscht werden. LIB, LINK. Es gibt keinen Platz mehr im Verzeichnis, um erzeugte Dateien oder Arbeitsdateien abzulegen. Mit dem Kommando ERASE sind nicht mehr benötigte Dateien zu entfernen.
Directory needs to be re-formatted for date/time stamps.	SET. Es wurde eine Datum/Zeit-Option angegeben, aber das Verzeichnis ist nicht für die Eintragung von Datum und Zeit initialisiert. Mit dem Kommando INITDIR ist das Verzeichnis zu initialisieren.
DISK READ DISK READ ERROR: Disk read error: dateibez DISK READ ERROR – dateibez	LIB, LINK, PIP. Die angegebene Diskettendatei kann nicht gelesen werden.

Meldung	Bedeutung
DISK WRITE. Disk Write Error DISK WRITE ERROR: DISK WRITE ERROR – dateibez	LIB, LINK, PIP, SUBMIT. Eine Schreiboperation auf die Diskette kann nicht erfolgreich ausgeführt werden, weil möglicherweise die Diskette voll ist. Mit dem Kommando ERASE können nicht mehr benötigte Dateien von der Diskette entfernt werden.
Do you want another file? (Y/N)	PUT. Es ist ein Y einzugeben, wenn die Ausgabe auf eine weitere Datei umgelenkt werden soll. Anderenfalls ist N einzugeben.
Drive defined twice in search path	SETDEF. Ein Laufwerk kann nur einmal in der Suchreihenfolge angegeben werden.
Drive Read Only	ERASE, RENAME. Die angegebene Datei ist eine READ-Only-Datei und kann deshalb nicht gelöscht oder umbenannt werden.
Duplicate RSX in header. Replacing old by new. This file was not used.	GENCOM. Die angegebene RSX wurde bereits einer COM-Datei zugeordnet. Die alte Datei wurde verworfen.
Duplicate input RSX.	GENCOM. Zwei oder mehr RSX mit dem gleichen Namen wurden angegeben. GENCOM verwendet nur einen der RSX.
END OF FILE, ^Z, ?	PIP erkennt ein unerwartetes Dateiende-Zeichen (end-of-file) bei der Übertragung einer Hex-Datei.
End of line expected.	DEVICE, GET, PUT, SETDEF. Das eingegebene Kommando hat keine weiteren Parameter. Ein Zeilenende wurde erwartet. Alle weiteren Zeichen in der Kommandozeile werden ignoriert.
Error at end of line:	DEVICE, GET, PUT, SETDEF. Der festgestellte Fehler ist am Ende der Eingabezeile aufgetreten.

Meldung	Bedeutung
Error on line nnnnn:	SUBMIT. Das SUBMIT-Kommando gibt seine Meldungen in der oben dargestellten Form aus, wobei nnnnn die Zeile in der SUBMIT-Datei bezeichnet. Die darauffolgende Zeile erklärt den aufgetretenen Fehler.
FILE IS READ/ONLY	PUT. Die für die Ausgabedaten angegebene Datei besitzt das Read-Only-Attribut.
File already exists; delete it? (Y/N) file already exists, delete (Y/N)?	PUT. Durch die Eingabe Y wird die Datei gelöscht. Anderenfalls bricht das Programm ab. RENAME. Vor der oben angegebenen Meldung steht die Dateibezeichnung. SCP 3.0 wurde veranlaßt eine Datei zu erstellen oder umzubenennen, wobei die verwendete Dateibezeichnung bereits für eine andere Datei verwendet wurde. Es ist entweder die bereits existierende Datei zu löschen oder eine andere Dateibezeichnung zu wählen.
FILE NAME ERROR:	LIB. Die Form des Quell-Dateinamen ist nicht gültig.
File not found. FILE NOT FOUND – dateibez	DUMP, GENCOM, GET, PIP, SET. Die angegebene Ausgangsdatei existiert nicht. Es ist zu überprüfen, ob die Laufwerksbezeichnung richtig angegeben wurde oder die richtige Diskette im Laufwerk liegt.
First submitted file must be a COM file.	GENCOM. Als erste Datei im Kommando-Zusatz wird eine COM-Datei erwartet. Die einzige Ausnahme, in der GENCOM keine COM-Datei als erste Datei im Kommando-Zusatz erwartet, liegt bei Angabe der Option NULL vor.
FIRST COMMON NOT LARGEST:	LINK. Eine nachfolgende COMMON-Vereinbarung ist größer als die erste COMMON-Vereinbarung für den bezeichneten Block. Es ist zu überprüfen, ob die Dateien in der richtigen Reihenfolge gebunden wurden oder daß die Modulen in der Bibliothek in der richtigen Reihenfolge angeordnet sind.
HELP.DAT not on current drive.	HELP. Das HELP-Kommando kann die Datei HELP.DAT nicht finden und deshalb keine Informationen anzeigen.
Illegal command tail.	DIR. Die Kommandozeile hat ein ungültiges Format oder beinhaltet eine ungültige Option.

Meldung	Bedeutung
Illegal Format Value.	DIR. Für die Anzeigeformate dürfen nur die Optionen SIZE und FULL verwendet werden.
Illegal Global/Local Drive Spec Mixing.	DIR. In einer Kommandozeile treten sowohl eine Dateibezeichnung mit Laufwerksangabe als auch die Option DRIVE auf.
Illegal filename.	SAVE. Die Dateibezeichnung in der Kommandozeile ist fehlerhaft.
Illegal Option or Modifier.	DIR. Es wurde eine ungültige Option oder Abkürzung benutzt.
Illegal date/time specification.	DATE. Das Format von Datum bzw. Zeit ist ungültig.
Incorrect file specification.	RENAME. Das Format der Dateibezeichnung ist ungültig.
INDEX ERROR:	LINK. Der Index eines IRL enthält ungültige Informationen.
Insufficient Memory INSUFFICIENT MEMORY:	GET, LINK, PUT, SUBMIT. Für die Zuweisung von Pufferbereichen ist nicht genügend Speicher vorhanden oder die Schachtelungstiefe von SUBMIT ist zu groß.
Invalid ASCII character	SUBMIT. Die SUBMIT-Datei enthält ein ungültiges Zeichen (OFFH).
Invalid command.	GET und PUT. Die in die Kommandozeile eingegebene Zeichenkette oder Teilzeichenkette wurde in dem verwendeten Zusammenhang als ungültig erkannt.
Invalid delimiter.	DEVICE, GET, PUT, SETDEF. Die Begrenzer [] = oder Leerzeichen waren an den verwendeten Stellen falsch. Es wurde beispielsweise das Zeichen [anstelle von = verwendet.
INVALID DESTINATION:	PIP. Es wurde eine ungültige Laufwerks- oder Gerätebezeichnung angegeben.
INVALID DIGIT – dateibez	PIP. Es wurde ein ungültiger hexadezimaler Wert beim Lesen einer Hex-Datei gefungen. Die Hex-Datei mit dem ungültigen Wert ist zu korrigieren, eventuell durch Neuerstellen.

Meldung	Bedeutung
Invalid drive.	SETDEF. Das angegebene Laufwerk ist nicht gültig. Als Laufwerksbezeichnungen werden von SETDEF nur * (für Standardlaufwerk) und A bis P akzeptiert. TYPE. Gültige Laufwerke sind A bis P.
Invalid File. INVALID FILENAME Invalid file name. Invalid Filename. Invalid file specification.	ERASE, GENCOM, GET, PIP, PUT, SET, SUBMIT, TYPE. Der eingegebene Dateiname entspricht nicht den Bedingungen für einen Dateinamen im SCP 3.0.
INVALID FORMAT	PIP. Das Format des PIP-Kommandos ist falsch. In der Beschreibung ist nachzulesen, wie ein PIP-Kommando anzugeben ist.
Invalid number.	DEVICE. Eine Zahl wurde erwartet, aber nicht gefunden oder die Zahl liegt außerhalb des Wertebereiches. Die Zahl muß im Bereich von 0 bis 255 liegen.
Invalid option.	DEVICE und GET. Es wurde eine Option erwartet und die gefundene Zeichenkette ist keine gültige Option oder in dem Zusammenhang nicht gültig. SETDEF. Die in der Kommandozeile eingegebene Option ist nicht gültig. Gültige Optionen sind DISPLAY, NO DISPLAY, NO PAGE, ORDER, PAGE, TEMPORARY.
Invalid option or modifier.	DIR, GET, PUT. Die eingegebene Option ist nicht gültig.
INVALID PASSWORD Invalid password or passwords not allowed.	PIP. Das angegebene Paßwort ist falsch oder ein Paßwort wurde angegeben und die Datei ist nicht paßwortgeschützt.
Invalid physical device.	DEVICE. Es wurde der Name eines physischen Gerätes erwartet. Der gefundene Name in der Kommando-Zeichenkette entspricht nicht dem Namen eines physischen Gerätes im System.
INVALID REL FILE:	LINK. Die bezeichnete Datei beinhaltet ein ungültiges Bitmuster. Es ist sicherzustellen, daß ein REL- oder IRL-Datei angegeben wurde.

Meldung	Bedeutung
Invalid RSX type.	GENCOM. Der Dateityp muß RSX sein.
Invalid SCB offset.	GENCOM. Der angegebene SCB-Offset liegt außerhalb des Bereiches. Der Bereich für den SCB-Offset ist OOH bis 64H.
INVALID SEPARATOR	PIP. Als Trennzeichen zwischen zwei Eingangsdateinamen wurde ein ungültiges Zeichen verwendet.
INVALID SOURCE	PIP. Ein ungültiges Laufwerk oder Gerät wurde angegeben. Nur AUX und CON sind gültige Geräte.
Invalid type for ORDER option.	SETDEF. Der in der Kommandozeile angegebene Dateityp ist nicht COM oder SUB.
Invalid SYM file format	REF. Die Datei dateinam.SYM als Ausgangsdatei für REF ist falsch.
INVALID USER NUMBER	PIP. Es wurde eine Nutzer-Nummer größer als 15 angegeben. Die Nutzer-Nummern liegen zwischen 0 und 15.
Invalid wildcard.	RENAME. Die Dateibezeichnung enthält ein ungültiges Ersatzzeichen.
Invalid wild card in the FCB name or type field.	GENCOM. GENCOM erlaubt nicht die Angabe von Ersatzzeichen in Dateibezeichnungen.
MAIN MODULE ERROR:	LINK. Es wurde ein zweites Hauptprogramm ermittelt.
Make error	REF. Auf dem angegebenen Laufwerk ist kein Verzeichnisplatz mehr frei.
MEMORY OVERFLOW:	LINK. Es steht kein genügend großer Speicher zur Verfügung, um den Binde- lauf abzuschließen.
Missing Delimiter or Unrecognized Option.	ERASE. Das Format der ERASE-Kommandozeile ist ungültig.

Meldung	Bedeutung
Missing left parenthesis.	GENCOM. Die Angaben der SCB-Option müssen in runden Klammern stehen; die linke fehlt.
Missing right parenthesis.	GENCOM. Die Angaben der SCB-Option müssen in runden Klammern stehen; die rechte fehlt.
Missing SCB value.	GENCOM. Die Option SCB verlangt die Angabe eines Wertes.
More than four drives specified.	SETDEF. Es wurden mehr als vier Laufwerke für die Laufwerk-Suchkette angegeben.
MULTIPLE DEFINITION:	LINK. Das angegebene Symbol ist in mehr als einem der zu bindenden Module definiert.
n?	USER. Es wurde eine Zahl größer als 15 für einen Nutzerbereich angegeben. Wird zum Beispiel USER 18 eingegeben, dann wird auf dem Bildschirm 18? ausgegeben.
No directory label exists.	SHOW. Die Option LABEL wurde angegeben, aber die Diskette besitzt keinen Kennsatz.
No directory space NO DIRECTORY SPACE – dateibez	COPYSYS, GENCOM, PIP, SAVE. Für die Ausgabe ist kein Verzeichnisplatz frei. Mit dem ERASE-Kommando können nicht mehr benötigte Dateien gelöscht und das wegen Fehler abgebrochene Kommando wiederholt werden.
No disk space.	SAVE. Auf der Diskette ist für die Ausgabedatei nicht genügend Platz. Mit dem SHOW-Kommando kann der noch verbliebene Platz auf der Diskette ermittelt werden. Nun ist mit ERASE durch Löschen nicht mehr benötigter Dateien Platz zu schaffen oder eine andere Diskette mit mehr freiem Platz zu verwenden.
No file NO FILE: NO FILE – dateibez	DIR, ERASE, LIB, LINK, PIP, RENAME, TYPE. Die Datei kann in den angegebenen Laufwerken nicht gefunden werden.

Meldung	Bedeutung
No HELP.HLP file on the default drive.	HELP. Die Datei HELP.HLP muß sich auf dem Standardlaufwerk befinden.
NO INPUT FILE PRESENT ON DISK	DUMP. Die verlangte Datei existiert nicht.
No memory	Zum Laden des angegebenen Programms ist nicht genügend Speicher verfügbar.
No modifier for this option.	GENCOM. Ein nicht erforderlicher Parameter wurde angegeben.
NO MODULE:	LIB. Der bezeichnete Modul kann nicht gefunden werden.
No options specified.	SET. Es ist eine Option anzugeben.
No more space in the header for RSXs or SCB initialization.	GENCOM. Der Header hat nur für 15 Eintrittspunkte Platz oder die Kombination von RSX-Modulen und SCB-Blöcken überschreitet das Maximum.
No PRN file.	REF. Die Datei dateinam.PRN gibt es auf dem angegebenen Laufwerk nicht.
No Records Exist	DUMP. Von der Datei existiert lediglich eine Verzeichniseintragung.
No source file on disk	COPYSYS. Die angegebene Datei existiert nicht.
NO SPACE	SAVE. Im Verzeichnis gibt es keinen Platz für die Eintragung der Datei.
No 'SUB' file found.	SUBMIT. Die in der Kommandozeile angegebene SUB-Datei kann beim Durchsuchen der Laufwerke nicht gefunden werden.
No such file to rename.	RENAME. Die Datei, die umbenannt werden soll, existiert auf dem angegebenen Laufwerk nicht.
No SYM file	REF. Die Datei dateinam.SYM gibt es auf dem angegebenen Laufwerk nicht.

Meldung	Bedeutung
NON-SYSTEM FILE(S) EXIST	DIRS. Diese Meldung wird von DIRS ausgegeben, wenn auf dem angegebenen Laufwerk Dateien mit dem Attribut DIR existieren.
Not enough available memory. Not Enough Memory Not Enough Memory for Sort.	DIR, INITDIR. Der Speicherplatz reicht für Daten- oder Sortiertpuffer nicht aus.
Not enough room in directory.	INITDIR. Der im Verzeichnis noch freie Platz reicht nicht aus, um das Verzeichnis für die Eintragung von Datum und Zeit zu erweitern.
NOT FOUND	PIP. PIP kann die angegebene Datei nicht finden.
Not renamed, filespec read only.	RENAME. Die angegebene Datei kann nicht umbenannt werden, denn die Datei ist Read-Only.
OPEN FILE NONRECOVERABLE	PIP. Eine Diskette hat das falsche Format oder einen defekten Sektor.
Option only for drives.	SET. Die angegebene Option ist für Dateien nicht gültig.
Option requires a file reference.	SET. Die angegebene Option verlangt eine Dateibezeichnung.
Options not grouped together.	DIR. Optionen müssen in eckigen Klammern eingeschlossen sein.
Output File Exists, Erase it.	Die angegebene Ausgabedatei darf nicht schon existieren.
OVERLAPPING SEGMENTS	LINK. Link versucht ein Segment in einen Speicherbereich zu schreiben, der bereits von einem anderen Segment benutzt wird.
Page and nopage option selected. No page in effect.	SET. Die verwendeten Optionen schließen sich gegenseitig aus.
Parameter Error	SUBMIT. Gültiger Parameter innerhalb einer SUBMIT-Datei mit dem Dateityp SUB sind α1 bis α9.

Meldung	Bedeutung
Paßword Error.	DUMP, ERASE, GENCOM, TYPE. Das Paßwort ist falsch.
Physical Device Does Not Exist.	DEVICE. Das angegebene physische Gerät ist im System nicht definiert.
PROGRAM INPUT IGNORED.	SUBMIT. Vor dieser Meldung steht „WARNING“. Die SUBMIT-Datei beinhaltet eine Zeile mit < und das Programm verlangt keine weiteren Eingaben.
PUT>	PUT. Dieses Bereitschaftszeichen wird ausgegeben, wenn ein Programm Eingaben während der Abarbeitung eines PUT-FILE [NO ECHO]-Kommandos verlangt.
PUT ERROR: FILE ERASED.	PUT. Die Ausgabedatei von PUT wurde gelöscht und konnte nicht geschlossen werden.
QUIT NOT FOUND	PIP. Die zum Parameter Q angegebene Zeichenkette konnte in der Eingabedatei nicht gefunden werden.
Random Read	SUBMIT. Beim Lesen der vom SUBMIT-Kommando verwendeten Arbeitsdatei ist ein Fehler aufgetreten.
Read only	GENCOM, SET. Das Laufwerk oder die angegebene Datei sind schreibgeschützt.
Read error	TYPE. Beim Lesen der im TYPE-Kommando angegebenen Datei ist ein Fehler aufgetreten. Die Diskette ist zu prüfen und das Kommando zu wiederholen.
Reading file HELP.HLP Reading HELP.HLP index	HELP. Ein Fehler ist beim Lesen der Datei HELP.HLP aufgetreten. Es muß die Datei HELP.HLP von der Systemdiskette kopiert werden.
RECORD TOO LONG	PIP. Beim Kopieren einer Datei mit der Option [H] wurde festgestellt, daß ein HEX-Satz mehr als 80 Zeichen lang ist.

Meldung	Bedeutung
Requires SCP 3.0 or higher	DATE, DEVICE, DIR, ERASE, GENCOM, HELP, PIP, INITDIR, SET, SETDEF, SHOW, RENAME, TYPE. Diese Version des Kommandos läuft nur auf dem Betriebssystem SCP 3.0 oder höheren Versionen.
R/O DISK	PIP. Das Ziellaufwerk ist auf Read-Only gesetzt und PIP kann deshalb nicht darauf schreiben.
R/O FILE	PIP. Die Zieldatei ist auf Read-Only gesetzt und PIP kann nicht auf sie schreiben.
Sort Stack Overflow	DIR. Für den Sortier-Stack ist nicht genügend Speicher verfügbar.
START NOT FOUND	PIP. Die Zeichenkette zum Parameter S kann in der Quelldatei nicht gefunden werden.
Symbol Table overflow	REF. Für eine versuchte Symbolzuordnung ist kein Speicher verfügbar.
Symbol Table reference overflow	REF. Für eine versuchte Symbolreferenz ist kein Speicher verfügbar.
SYNTAX ERROR:	LIB. Das LIB-Kommando ist nicht richtig aufgebaut.
Too many entries in Index Table. Not enough memory	HELP. Es ist nicht genügend Speicher verfügbar, um beim Erstellen der Datei HELP.HLP die Tabelle mit den Begriffen im Speicher zu halten.
Topic: xxxxxx Not found.	HELP. Der verlagerte Begriff existiert nicht in der Datei HELP.HLP. HELP zeigt die verfügbaren Begriffe an.
Total file size exceeds 64K.	GENCOM. Die Ausgabedatei überschreitet die maximale Größe einer Datei.
Try „PAGE“ or „NO PAGE“	TYPE. Nur die Optionen PAGE und NO PAGE sind für dieses Kommando gültig.

Meldung	Bedeutung
Unable to close HELP.DAT Unable to close HELP.HLP	HELP. Beim Schließen der Datei HELP.HLP oder HELP.DAT ist ein Fehler aufgetreten. Möglicherweise ist auf der Diskette oder im Verzeichnis der Diskette nicht genügend Platz.
Unable to find HELP.HLP	HELP. Das HELP-Kommando benötigt für seine Abarbeitung die Datei HELP.HLP. Diese Datei ist von der Systemdiskette auf die verwendete Diskette zu kopieren.
Unable to Make HELP.DAT. Unable to Make HELP.HLP.	HELP. Auf der Diskette ist nicht genügend freier Platz für die Dateien HELP.HLP und HELP.DAT oder die Dateien sind Read-Only.
UNDEFINED START SYMBOL:	LINK. Das mit dem Schalter G angegebene Symbol ist in keinem der zu bindenden Module definiert.
UNDEFINED SYMBOLS:	LINK. Die nach der Meldung angegebenen Symbole wurden verwendet, aber in keinem der zu bindenden Module definiert.
UNEXPECTED END OF HEX FILE – dateibez	PIP. Ein EOF wurde festgestellt vor dem Abschluß-Satz der Hex-Datei. Die Hex-Datei ohne Abschluß-Satz muß möglicherweise durch Neuerstellen korrigiert werden.
Unrecognized drive.	SHOW. Das angegebene Laufwerk ist nicht gültig. Gültige Laufwerke sind A bis P.
UNRECOGNIZED ITEM:	LINK. Ein unbekanntes Bitmuster wurde gefunden und von LINK ignoriert.
Unrecognized input.	SHOW. Die Kommandozeile ist falsch aufgebaut.
Unrecognized option.	GENCOM und SHOW. Eine in die Kommandozeile eingegebene Option ist nicht gültig.
USER ABORTED	PIP. Die Ausführung von PIP wurde durch Betätigen von CTRL-C unterbrochen.

Meldung	Bedeutung
VERIFY ERROR: – dateibez	PIP. Es wurde mit der Option V kopiert und PIP hat eine Differenz zwischen den gerade geschriebenen Daten und den in seinem Puffer vorhandenen Daten beim Vergleich festgestellt.
Write error	REF. Vor dieser Meldung steht dateinam.XRF. Damit wird ausgesagt, daß auf der Diskette im angegebenen Laufwerk im Datenbereich oder im Verzeichnis kein freier Platz verfügbar ist.
Writing file: dateibez	HELP. Beim Schreiben der durch die angegebene Dateibezeichnung festgelegten Datei ist ein Fehler aufgetreten.
Wrong Paßword.	SET. Das angegebene Paßwort ist fehlerhaft oder ungültig.

robotron

**VEB Robotron Büromaschinenwerk
„Ernst Thälmann“ Sömmerda**

Weißenseer Straße 52
Sömmerda
DDR-5230

robotron

Robotron Export-Import

Volkseigener
Außenhandelsbetrieb der
Deutschen Demokratischen
Republik
Allee der Kosmonauten 24
Berlin
DDR-1140

1849 III/14/8 RIG 6/186/87 6

