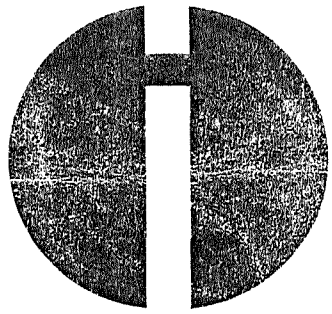
Olympia
System BOSS



PROLOGUE BETRIEBSSYSTEM
BENUTZERHANDBUCH

VERÖFFENTLICHUNG NR. B-1003 D

ÜBERARBEITETE AUSGABE A
DEZEMBER 1980

HINWEISE

BEMERKUNG: Wir behalten uns das Recht auf Änderung des gesamten Dokumentes oder von Teilen davon ohne Ankündigung vor.

PROLOGUE PLATTENBETRIEBSSYSTEM - BENUTZERHANDBUCH

INHALT

KAPITEL 3. BOSS BASIC UNTER PROLOGUE

KAPITEL 4. BAL UNTER PROLOGUE

KAPITEL 5. DATEISYSTEMAUFBAU

KAPITEL 1 - EINLEITUNG

1.1 ALLGEMEINE BESCHREIBUNG

PROLOGUE ist ein Betriebssystem für den Mikrocomputer BOSS, das einen allgemeinen Rahmen für Programmerstellung, Speicherung, Kontrolle und Durchführung bietet. Es verwendet Floppy- Disks, Hard-Disks und eine Kombination von beiden. PROLOGUE besitzt eine Dateiverwaltung, die dynamische Zuweisung von Speicherplatz gestattet. So erzeugt und verarbeitet der Benutzer die Daten anhand von Namen, während das System die Programmfunktionen für wirkungsvolle Platzverwaltung auf den Platten einsetzt.

1.2 LEISTUNGSUMFANG VON PROLOGUE

PROLOGUE bietet eine große Anzahl nützlicher Systemfunktionen, unterteilt in zwei Kategorien:

1. Benutzer-Funktionen - Die Betriebssystem-Funktionen, die normalerweise bei der Erstellung und Ausführung in höheren Programmiersprachen verwendet werden.

2. Systemfunktionen - Die wirksamen Systemfunktionen, die normalerweise nur von erfahrenen Systemprogrammierern unter Verwendung der Assemblersprache verwendet werden.

1.2.1 PROLOGUE-Systemprogramme

PROLOGUE ist aufgebaut auf einer Vielzahl von Standard-Dateiverwaltung und Betriebssystemroutinen, die mit allen PROLOGUE-Systemroutinen geliefert werden. Beim Einlesen von PROLOGUE werden diese Routinen in den unteren Teil des Speichers geladen.

Auf Wunsch des Benutzers wird die sequentielle und/oder index-sequentielle Dateiverwaltung als Teil des PROLOGUE-Systems eingesetzt. Hierfür wird zusätzlicher Platz im unteren Teil des Hauptspeichers benötigt, deshalb sollte die Dateiverwaltung je nach Bedarf eingesetzt werden.

Eine Anzahl Standard-Prologue-Dienstprogramme wird auf Ihrer PROLOGUE-Diskette als Objektmoduln geliefert, die nur bei Bedarf geladen und ausführt werden. Sie sind nicht im Hauptspeicher resident.

Diese Moduln sind

/	- Disketten-/Platten - Inhaltsverzeichnis
ED	- Text-Editor -- Erzeugt und ändert Quellprogramme
CP	- Kopierprogramm -- Erzeugt und verwaltet PROLOGUE-Datenträger und Dateien.
CPS	- Kopierprogramm -- kopiert jede Diskette/Platte spurenweise.

STATUS	- Zeigt Betriebszustand der magnetischen Peripheriegeräte an.
PATCH	- Gestattet die direkte Änderung der Daten in einem beliebigen Disketten-/Plattensektor.
DATE	- Initalisiert Zeit und Datum im System.
ASG	- Assign -- Bearbeitet Kommandos, die die automatische Ausführung einer Serie von PROLOGUE-Befehlen durch das System gestatten.
FM	- Format -- bringt eine Diskette in die von PROLOGUE geforderte Struktur.

Diese Standardprogramme sind in Kapitel 2 dieser Abhandlung genauer beschrieben.

1.2.2 In Option verfügbare Programme für höhere Sprachen

Die Programme für höhere Programmsprachen, die mit dem BOSS-Floppy-Disk-System in Option lieferbar sind, funktionieren unter Kontrolle des Betriebssystems von PROGOLUE. Sie enthalten:

BASIC	BASIC-Interpreter
BAL	Kaufmännisches BASIC-Compiler

Der Aufruf dieser Programme wird in Kapitel 2 und 3 dieses Dokuments beschrieben. Die Sprachen selbst sind in verschiedenen Programmier-Handbüchern beschrieben. Eine Liste der Dokumentation wird in Anhang B gegeben.

1.2.3 Prologue-Systementwicklungsprogramme

Von PROLOGUE ist ein Entwicklungssystem erhältlich. Es wird hauptsächlich von Systemprogrammierern benutzt, die in der Assemblersprachebene arbeiten. Es enthält alle Routinen der Benutzer-Fassung und folgende zusätzliche Moduln:

AZM	- Assembler -- verarbeitet Assembler - Quellprogramme für die Erstellung von Objektprogrammen.
EDL	- Linkage-Editor -- bindet assemblierte Objektmoduln und löst externe Referenzen auf.
MM	- Monitor -- Testhilfe für Assemblerprogramme, gewährt Zugriff zu Registern und Speichern der Zentraleinheit.
RELOC	- Wandelt ein absolutes Programm in ein verschiebbares um.

Die Benutzung dieser Programme wird ausführlicher beschrieben in : **PROLOGUE Operating System & File Management Programmer's Guide,** Veröffentlichung Nr. B-1004.

1.2.4 Schnittstellen zum PROLOGUE-System

Das PROLOGUE-System besteht aus verschiedenen Routinen und Funktionen, die in Assembler-Sprache geschrieben sind. Diese Routinen sind für den Programmierer in Assembler-Programmen und in anderen Programmen, die einen Assembler-Unterroutinen-Aufruf enthalten, nötig.
Diese Routinen sind im **PROLOGUE Operating System & File Management Programmer's Guide,** Veröffentlichung Nr. B-1004 ausführlich beschrieben.
Sie enthalten:

- Direkte Ein- und Ausgabe auf Peripheriegeräte
- Eingabe-Routine
- Bildschirm- und Drucker-Ausgaberoutinen
- Routinen zum Erstellen, Eröffnen, Lesen, Schreiben, Schließen und Löschen von Dateien
- Routine für die Analyse von Dateinamen
- Laderoutinen für Objektdateien
- Umwandlungsroutinen ASCII/BINÄR

1.3 SYSTEMORGANISATION

Für die möglichst wirkungsvolle Nutzung von PROLOGUE und den Sprachen und Routinen, die das System kontrolliert, ist eine allgemeine Kenntnis der Dateiorganisation und der Kommandostruktur des Systems Voraussetzung. Diese Organisation wird nachstehend ausführlich genug für das allgemeine Verständnis beschrieben. Komplette Einzelinformationen für den Systemprogrammierer finden sich im **PROLOGUE Operating System and File Management System Programier's Guide**, Veröffentlichung Nr. B-1004.

1.4 AUFBAU DER DATEIEN

PROLOGUE ist ein Disketten-/Plattenbetriebssystem, das die gesamte Verwaltung des Datenträgers übernimmt. Der Benutzer kann jede Datei durch einen Namen spezifizieren, ohne sich mit physikalischen Einzelheiten über die Speicherung und den Abruf der Dateien von der Diskette/Platte befassen zu müssen.

Um eine einwandfreie Verwaltung der Dateien mit dem PROLOGUE- System zu sichern, muß der Benutzer gewisse Aspekte des Dateiensystems durch Eingabe von Parametern bei der Erzeugung der Dateien definieren. Nachstehend werden einige der verwendeten Ausdrücke definiert.

Volume - Das PROLOGUE-System ist volume-weise aufgebaut, z.B. jede Minifloppy-Diskette stellt ein Volume dar, der seinen eigenen Namen und seine eigenen logischen Speichermerkmale besitzt. Jedes Volume enthält ein Inhaltsverzeichnis der in ihm enthaltenen Dateien.

Spur - Alle magnetischen Plattendatenträger sind aus konzentrischen magnetischen Spuren aufgebaut. Die im BOSS-System verwendeten Minifloppy-Disketten besitzen 32 Spuren pro Seite.

Sektor - Jede Diskette/Platte ist in eine Anzahl keilförmiger Abschnitte (Sektoren) unterteilt, so daß jede Spur aus der gleichen Anzahl von Abschnitten besteht - 16 für die Minifloppy-Disketten.

Granule - dies ist die kleinste logische Speichereinheit auf einem Volume des PROLOGUE-System. Wenn ein Volume angelegt wird, muß der Programmierer die Sektorenzahl pro Granule spezifizieren. In diesem Volume wird dann jedes Granule aus dieser Anzahl aufeinanderfolgender Sektoren der Diskette/Platte bestehen. Ein Granule besitzt mindestens 1 Sektor und höchstens 256. Der Vorgabewert beträgt 16.

Da eine Datei ganze Granules belegt, ist dieses Konzept der Granule-Größe wichtig.

Dieser Aufbau gestattet es dem Programmierer, die logischen Einheiten auf einem Datenträger in der für seinen Bedarf zweckmäßigsten Größe zu wählen.

Block - Ein Block besteht aus einem oder mehreren aufeinanderfolgenden Granules.

Datei – Alle Programme und Daten werden auf den Disketten/Platten als Dateien gespeichert. Jede Datei besitzt einen Dateinamen, der aus einem Namen, einer Typenbezeichnung (nicht unbedingt erforderlich) und einem Dateizugriffschlüssel (nicht unbedingt erforderlich) besteht (siehe Näheres hierzu in Abschnitt 1.5.2). Das System findet die Dateien mit Hilfe der Blocknummer und der Granules.

Einheit – Disketten sind als logische Volumes aufgebaut, die physisch in den Floppy-Disk-Antrieb eingelegt werden müssen. In der PROLOGUE-Kommandostruktur kann der Benutzer das Gerät wählen, in das die gewünschte Datei eingelegt ist. Wenn der Benutzer diese Spezifikation unterläßt, benutzt das System einen Vorgabe-Wert wie folgt:

System-Einheit -- dies ist das Gerät, das die **System**-Programme enthält, wie PROLOGUE selbst, das Inhaltsverzeichnis, die Kopierprogramme, BASIC, usw. Es wird beim Konfigurieren des Systems festgelegt und ist gewöhnlich Floppy-Einheit Ø.

Benutzer-Einheit -- Dies ist das Gerät, das die **Benutzer**-Dateien enthält. Es wird ebenfalls beim Konfigurieren des Systems festgelegt und ist meist Floppy-Einheit 1.

Wenn Sie in Ihrem System mehr als zwei Floppy-Laufwerke benutzen, könnte Floppy 3 oder Floppy 4 als Vorgabeeinheit benutzt werden. Der Benutzer kann diese Vorgaben durch Änderungen bestimmter Speicherparameter ändern, wie in Abschnitt 5.4 noch aufgeführt wird.

Wenn die Grundinformation gegeben ist, verwaltet PROLOGUE die Dateien automatisch und weist Disketten-/Plattenraum dynamisch in Granules zu. Daten werden hinzugefügt und gelöscht, und der verfügbare Raum wird optimal genutzt.

1.5 KOMMANDOSTRUKTUR

1.5.1 Schreibkonventionen

Der Bediener benutzt PROLOGUE mit genau festgelegten Kommandos. Nachstehend wird eine allgemeine Struktur dieser Kommandos beschrieben. Es werden folgende Schreibweisen verwendet:

{ } – gibt eine Auswahl von Eingaben an.

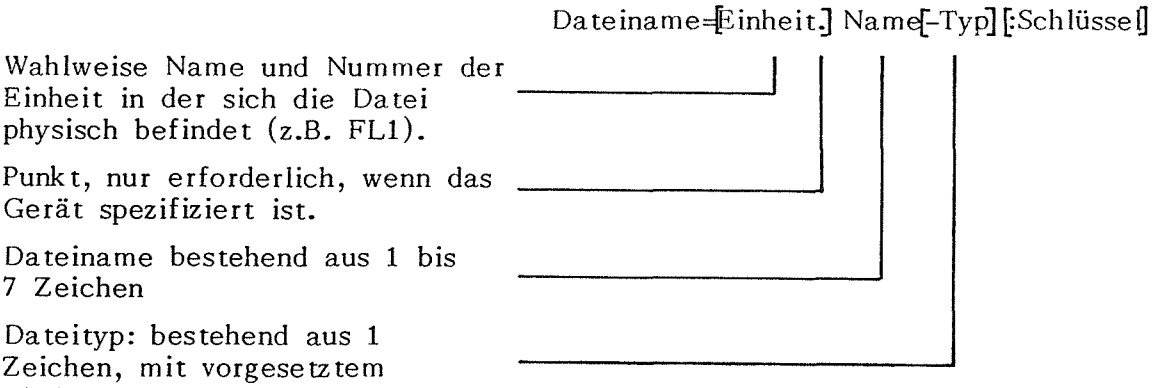
[] – gibt ein Optionsfeld in einem Kommando an. Wenn die Optionsinformation nicht eingegeben wird, wird eine Standardvorgabe angenommen (Achtung: die Klammern dürfen nicht getippt werden, wenn das Kommando in den Computer eingegeben werden soll).

␣ – zeigt eine erforderliche Leerstelle an.

cr oder (cr) – zeigt einen Wagenrücklauf an. Wird normalerweise geschrieben, um eine Kommandozeile abzuschließen (Freigabe).

1.5.2 **Dateireferenzen -- Dateiname**

PROLOGUE verwaltet ein System von Dateinamen, mit zahlreichen Dateien, die auf einer einzigen Diskette/Platte gespeichert, katalogisiert und verfügbar sind.Sämtliche Programme und Dateien im PROLOGUE-System sind mit einem Dateinamen wie folgt gekennzeichnet:



Hier bedeuten:

Gerät - Name und Nummer des Gerätes, in dem sich die Datei physisch befindet, wie:

FLØ - Floppy-Disk -Einheit Ø
FL1 - Floppy Disk -Einheit 1
FL2 - Floppy Disk -Einheit 2

Vorgabe-Wert: Wenn die Einheit nicht spezifiziert ist, wird das Benutzer-Laufwerk vorausgesetzt. Diese Einheit wird beim Konfigurieren des Systems gewählt (siehe Behandlung der Parameter in Anschnitt 5.4), meist Einheit 1 (Die Systemeinheit, von der PROLOGUE in den Speicher geladen wird, ist gewöhlich die Einheit Ø).

Name - Es ist ein Name mit 1-7 Zeichen erforderlich. Dafür kommen alphabetische oder numerische Zeichen oder die Sonderzeichen $ oder / in Frage. Andere Sonderzeichen sind zu vermeiden. Punkt, Bindestrich und Doppelpunkt dürfen **nicht** als Bestandteil des Namens verwendet werden.

Typ - Ein beliebiges einzelnes Zeichen, das zum Bezeichnen des Dateityps verwendet wird. Es kann jedes Zeichen (alphabetisch, numerisch oder $) verwendet werden, aber die folgende Liste enthält verschiedene von Systemprogrammen zugewiesene Zeichen und wird empfohlen:

O - Objekt-Programm, das Prologue einlesen und ausführen kann. Verschiebbares Programm,kann einer festen Speicheradresse zugewiesen werden. PROLOGUE-Systemprogramm und zusätzliche Compiler erscheinen auf Ihrem System-Inhaltsverzeichnis mit der Typenbezeichnung O. Wenn das Programm kontinuierlich ist (nur ein Block), wenn es keine externen Referenzen besitzt und

wenn es keine Prologue-Unterstützung erfordert, kann es direkt geladen werden (eingegeben in Antwort auf BOSS...).

S - **Quelle.** ASCII-Quellprogramm. Kann vom Editor als Eingabe für einen Compiler wie BAL oder BASIC oder Kommandodatei für das Verkettungskommando vorbereitet werden.

Quellprogrammen, die unter dem BASIC-Interpreter geschrieben und auf der Platte mit dem SAVE-Befehl gespeichert sind (mit Option A), wird Typ S als Vorgabe-Wert zugewiesen.

C - **Verdichtete Programme.** Empfohlener Typ für verdichtete Programme, die vom BASIC-Interpreter erzeugt werden. (Es sei darauf hingewiesen daß BASIC diesen Dateien Typ S als Vorgabe-Wert zuweist, aber Typ C dient der Unterscheidung zwischen verdichteten und erweiterten BASIC-Quellprogrammen . Näheres hierzu siehe in Kapitel 3.

T - **Übersetztes Zwischencodeprogramm.** Generiert vom BAL-Übersetzer. Kann nur vom BAL-EX (Ausführungs-) -Programm eingelesen und ausgeführt werden.

I - **Indexsequentielle Schlüsseldatei.** Wird vom indexsequentiellen Dateisystem generiert und verwendet.

D - **Datei.** Zur Schlüsseldatei korrespondierende Datendatei wird vom indexsequentiellen Dateisystem generiert und verwendet.

$ - **Temporäre Datei.** Wird von den PROLOGUE-Hilfsprogrammen generiert.

Vorgabe-Wert: wenn die Typenbezeichnung weggelassen wird, wird bei Dateien, die der Benutzer mit dem Kopierprogramm erzeugt (CP), eine ASCII-Leerstelle angenommen. Viele PROLOGUE-Dienstprogramme vergeben die passende Typenbezeichnung, wenn der Benutzer die Angabe unterläßt. Zum Beispiel wird der Linkage-Editor nur Dateien vom Typ O annehmen und wird Ausgabe-Dateien den Typ O zuordnen.

Schlüssel - Sicherheits-Zugriffschlüssel mit 1 bis 4 Zeichen wahlweise einsetzbar (Kennwort). Der Schlüssel wird nur angezeigt, wenn er erstmals eingegeben wird. Im Inhaltsverzeichnis der Dateien wird er nicht angezeigt.

Dieses Hilfsmittel ist nützlich für den Schutz bestimmter Programme gegen Änderung durch Unbefugte, da eine Datei nicht gelistet, geändert kopiert oder gelöscht werden kann, wenn nicht der richtige Schlüssel als Teil des Dateinamens spezifiziert wird.

Der Schlüssel hat das Format RRWW; und er kann folgendermaßen benutzt werden:

Kein Schlüssel eingegeben: ASCII-Leerstellen werden eingesetzt (Vorgabe-Wert),(␣ :Leerzeichen)

Schlüssel = RR – Das Kennwort wird als RR␣␣ gespeichert. Durch Eingabe von RR als Schlüssel für den Namen der Datei wird Lesen, Schreiben und Löschen ermöglicht.

Schlüssel = RRWW – wenn nur der RR-Teil des Kennwortes eingegeben wird, kann nur gelesen werden. Mit Eingabe von RRWW ist Lesen, Schreiben und Löschen möglich.

Schlüssel = ␣␣WW – die Datei ist gegen Schreiben geschützt. Lesen ist jederzeit möglich; aber des Kennwort-␣␣WW **muß** eingegeben werden, um das Schreiben zu ermöglichen.

Beispiel 1 – Nehmen wir das Anlegen der Datei NEW: 1234 an, die Eingabe von NEW:12 gestattet das Lesen der Datei, nicht aber das Schreiben oder Löschen. Mit Eingabe von NEW:1234 wird Lesen, Schreiben und Löschen möglich.

Beispiel 2 – Nehmen wir an, die Datei NEW:␣␣12 wird angelegt. Lesen ist möglich, wenn kein Kennwort eingegeben wird. Zum Schreiben und Löschen muß:␣␣12 eingegeben werden.

Hinweis: Verwenden Sie die Sicherheitskennworte nur, wenn Sie die Sicherheit wirklich brauchen, und **vergessen Sie dann das Kennwort nicht.**

Beispiele:

Folgende Dateinamen sind gültig:

SAMPLE
FLØ.A$B-S
FL1.SAMPLE-S
FL1.SAMPLE-S:1234 Hinweis : Beide Dateinamen, SAMPLE-S und SAMPLE-S:1234, sind gültig, aber sie können nicht gleichzeitig auf dem selben Volume erzeugt werden, da die Namen die gleichen sind und sich nur die Schlüssel unterscheiden.

Folgende Dateinamen sind ungültig:

SAMPLEFILE – Die Höchstlänge des Namens ist überschritten (max 7 Zeichen) .

FL1,SAMPLE – Es muß heißen : FL1.SAMPLE. Der Punkt ist notwendig.

FLO.SAMPLE – Die Einheit muß als FLØ spezifiziert werden. Dafür ist die Ziffer Ø und nicht der Buchstabe O zu verwenden.

FL1A,B-B - Ein Komma als Teil des Dateinamens ist unzulässig.

1.5.3 Dateireferenzen -- Programmname

Dies ist eine Objektdatei, die von PROLOGUE eingelesen und ausgeführt werden kann. Sie ist aufgebaut wie der Dateiname, **muß** vom Typ O sein und darf keine unaufgelösten Referenzen enthalten. Sämtliche Systemdateien von PROLOGUE gehören zu dieser Kategorie.

1.5.4 Laden von PROLOGUE

Wenn Ihr BOSS-Mikrocomputersystem eingeschaltet ist, wird automatisch das Ladeprogramm in den Speicher eingelesen und das Wort BOSS... erscheint. Wenn dies nicht der Fall ist, oder wenn Sie das Programm neu einlesen müssen, ist der Knopf RESET auf der Rückseite des Computers zu drücken. Überprüfen Sie, ob Sie eine für Ihr System erstellte Diskette verwenden.

Zum Laden von PROLOGUE ist folgendermaßen vorzugehen:

1. Ihre Diskette in die richtige Einheit einlegen.
2. Als Antwort auf BOSS... eingeben:

   BOSS..B:U,S (cr)

   Hier bedeuten :B - Einlesen vom Floppy Disk
   U - Nummer der Einheit
   S - Sektorennummer im Hexadezimalkode -- 80H ist die geforderte Sektorennummer auf einer Standard-Diskette.

3. Das Betriebssystem wird von Sektor 80 der spezifizierten Einheit in den Speicher eingelesen, und auf dem Bildschirm erscheint folgende Meldung:

   Olympia International
   Prologue Version X.X -- DD/MM/YY
   Avail.User Space [XXXX/YYYY]
   Release ZZ V651/3
   →

Hier bedeuten:

X.X	- PROLOGUE-Version
DD/MM/YY	- Datum der Konfiguration im Format Tag/Monat/Jahr
XXXX/YYYY	- Zeigt die Anzahl der für Benutzerprogramme zur Verfügung stehenden Speicheradressen an, XXXX ist die Startadresse für Benutzerprogramme.
ZZ Release-Nummer und zuständige Abteilung	
→	- das PROLOGUE-Symbol, das angibt, daß das System jetzt durch ein beliebiges gültiges Kommando gesteuert werden kann, das über die Tastatur eingegeben wird.

Hinweise - Das ausführbare PROLOGUE-System-Programm führt gewöhnlich eine der Bezeichnung BOSS ,BOSS-1,BOSS2,BOSS-3.

(Es kommt dafür jeder gültige Dateiname mit Typkennung-O in Frage). Es befindet sich stets auf Spur 8 eines Standard-PROLOGUE-Volume und muß zusammenhängend sein.

Der System-Programmierer kann dieses Programm auf irgendeine andere Spur eines neuen Volume übertragen wenn dies gewünscht wird. Siehe **PROLOGUE Operating System & File Management System Programmer's Guide,** Veröffentlichung Nr. B-1004 für Einzelheiten.

4. Wenn PROLOGUE geladen ist, bleibt es im unteren Hauptspeicher resident. In einer PROLOGUE Routine besteht stets die Möglichkeit, durch Drücken der Taste BREAK-L oder der RESET-Taste zum BOSS-Symbol zurückzukehren. Ein Druck auf die BREAK-P-Taste bewirkt eine Rückkehr von BASIC oder BAL zu PROLOGUE.

1.5.5 Allgemeine Kommandosyntax von PROLOGUE

In Antwort auf das PROLOGUE-Symbol ist ein Kommando folgender Form einzugeben:

[&ADR␣]Programmname [,P1 ...,Pn] (cr)

Hier bedeuten:

&ADR	- Startadresse (in Option) für das Programm. - Und-Zeichen (&), auf das eine Hexadezimalzahl ohne Pre- und Suffix folgt. Beispiel : &3000. Wenn diese Adresse nicht spezifiziert ist, wird das Programm an die Systemvorgabe-Adresse in den Speicher geladen, die als erste Adresse des Benutzer-Speichers beim ersten Einlesen von PROLOGUE angegeben ist. (Im Speicher auf der Adresse 99H, 9AH spezifiziert, siehe Näheres hierzu in Abschnitt 5.4).
␣	- Eine Leerstelle oder ein Komma ist notwendig. (Die Leerstelle ist vorzuziehen).
Programmname	- Muß der Name eines Zielprogrammes sein (Format wie in Abschnitt 1.5.2 definiert) das **keine** unaufgelösten Referenzen enthalten darf. Das Programm muß vom Typ-O sein, aber die Typenbezeichnung braucht nicht eingegeben zu werden. Alle PROLOGUE System-Programme haben dieses Format und erscheinen im Inhaltsverzeichnis als Name-O.
P1..., Pn	- Liste der für das Programm erforderlichen Parameter. In der Beschreibung eines jeden PROLOGUE-Programms findet sich eine Liste der gültigen Parameter.

Dieser Befehl gestattet das Laden des spezifierten Objektes (Programm) in den Speicher durch PROLOGUE, an der gewünschten Adresse (oder an der im System vorgegebenen Adresse) und den Start des Programmes.

Es ist möglich, ein Programm ohne sofortige Ausführung einzulesen. Näheres hierzu auf der nächsten Seite.

Gewöhnlich wird das Zeichen "-" nach richtiger Ausführung eines PROLOGUE-Kommandos in der Kommandozeile angezeigt. Dann kehrt das PROLOGUE-Symbol zurück.

Beispiele:

Für diese Beispiele wird angenommen, daß Einheit Ø als Systemeinheit dient, das die Systemprogramme enthält, Einheit 1 ist das Benutzergerät das die vom Benutzer geschriebenen Programme enthält.

&4000, SAMBLE (cr) - ein Programm mit Namen SAMBLE-O wird auf Einheit Ø gesucht (Systemeinheit) und ab Adresse 4000 in den Speicher geladen. Dann wird das Programm angeführt (wenn dies vom Programm selbst zugelassen wird.) PROLOGUE setzt voraus daß das mit Programm-Namen spezifizierte Systemprogramm in die Systemeinheit eingelegt ist, falls der Benutzer nicht anders entscheidet.

CP,SF,TEST-S:2 (cr) - Das Kopierprogramm (mit dem richtigen Namen CP-O) wird von Einheit Ø eingelesen (Systemeinheit) und ausgeführt. Es wird seine SF-Funktion ausführen, Dateien löschen, durch Zuweisung einer Datei namens TEST-S:2 an Einheit 1 (Benutzereinheit) und Löschung. **Da die Geräte nicht als Teil der Dateibezeichnungen spezifiziert sind, werden die vorgegebenen Laufwerke angenommen.**

Hinweise zur Kommandosyntax

Für die Parameterliste sollten folgende allgemeine Regeln beachtet werden:

1. Alle Befehle in der Parameterliste (und der Kommandozeile) müssen durch Kommata getrennt werden. Die Kommandozeile einschließlich Parameterliste muß ohne Leerzeichen zwischen den einzelnen Kommandos geschrieben werden.

2. Numerische Kommandos werden gewöhnlich in Dezimalschreibweise ausgedrückt. Wenn Hexadezimalzahlen gefordert werden, wird Ihnen ein H nachgestellt (z.B.3000H, sofern nichts anderes angegeben ist.

3. Ein Programm kann ohne sofortige Durchführung geladen werden wie folgt:

   a. den Namen des Programmes schreiben. Darauf folgt ein cr (Freigabe) kein Parameter.

   b. Wenn das Programm von der Diskette geladen ist, erscheint auf dem Bildschirm ein Komma hinter dem Programmnamen (,) und das System wartet auf die Parametereingabe. Nun können Volumes nach Bedarf in das Laufwerk eingelegt werden, dann sind die benötigten Parameter einzugeben.

      Beispiel: eine Diskette von Laufwerk Ø auf Laufwerk 1 übertragen.

      a) Systemdiskette mit CP-Kopierprogramm in Laufwerk Ø einlegen; eine leere Diskette in Laufwerk 1 einlegen.

      b) CP tippen, dann CR (Freigabe)

      c) Das Kopierprogramm wird geladen, auf dem Bildschirm erscheint die Angabe: CP, mit blinkendem Cursor.

d) Die System-Diskette aus Laufwerk Ø nehmen und die zu kopierende Diskette einlegen.

e) Schreibe: DV, FLØ, FL1, cr (Freigabe)

f) Nun wird der Befehl ausgeführt und die Diskette kopiert.

4. Ein PROLOGUE-Systemprogramm, das eben ausgeführt wurde, kann ohne erneutes Einlesen nochmals ausgeführt werden. Wenn Sie den Programmnamen eingeben, wird das Programm, das sich bereits im Speicher befindet, erneut initialisiert und ausgeführt.

   Diese Möglichkeit kann in Assemblerprogrammen, die vom Benutzer geschrieben weden, durch Eingabe von NOP (Code ØØ) an die erste Stelle des Programmes gesetzt werden.

   Alle PROLOGUE-Systemprogramme, mit Ausnahme des Monitors beinhalten diese Möglichkeit.

5. Gewöhnlich zeigt das Symbol '-' die einwandfreie Durchführung des Programmes an.

6. Um ein Dienstprogramm zu verlassen, sind folgende Tasten zu drücken:

   ESC (ESCAPE) Taste oder
   CR (RETURN) (Freigabe)-Taste (Leer-Eingabe).

7. Zum Unterbrechen einer unendlichen Schleife in einer PROLOGUE-Routine ist die Taste BREAK-L zu drücken. dies führt zu einer Rückkehr zu BOSS. In einer unendlichen Schleife in einem BASIC- oder BAL-Programm bewirkt ein Druck auf BREAK-P die Rückkehr zu PROLOGUE. Wenn die Unterbrechungen nicht zu funktionieren scheinen, ist auf den RESET-Knopf an der Rückseite der Maschine zu drücken, was stets eine Rückkehr zu BOSS bewirkt.

1.6 FEHLERERKENNUNG

Wenn die PROLOGUE-Kommandos eingegeben sind, überprüft der Computer diese auf Syntax- und Systemfehler (wie Schreibfehler oder Spezifikation einer nicht vorhandenen Einheit). Während die verschiedenen PROLOGUE-Routinen ausgeführt werden, werden Fehler wie Kapazitätsüberschreitung festgestellt. Wenn ein Fehler gefunden wird, erfolgt eine Fehlerkode-Anzeige. Anhang A gibt eine Liste der Fehlerkodes mit Definitionen und Beseitigungsverfahren.

Fehlererkennung und Fehlerkodes für jede Programmsprache werden im zugehörigen Sprachhandbuch behandelt.

KAPITEL 2. STANDARD-PROLOGUE-KOMMANDOS/ROUTINEN

2.1 EINLEITUNG

Das PROLOGUE-Disketten/Platten-System enthält ein Paket von System- und Dienstprogrammroutinen. Diese werden als Bestandteil eines jeden PROLOGUE-Systems geliefert und werden in diesem Kapitel ausführlich beschrieben.

Andere Programme wie BOSS BASIC und BAL sind als Bestandteil des PROLOGUE-Systems in Option erhältlich. Die Arbeitsweise dieser Programme im PROLOGUE-System wird in den folgenden Kapiteln beschrieben. Für Programmiereinzelheiten wird der Benutzer auf die jeweiligen Programmier-Handbücher verwiesen.

Sämtliche Programme auf PROLOGUE-Grundlage werden durch Spezifizierung einer Kommandozeile in den Computer eingegeben. Diese enthält die Programmbezeichnung und eine Parameterfolge. Eine Beschreibung hierzu wurde in Abschnitt 1.5.5 gegeben .

Während die Kommandos eingegeben und die Programme ausgeführt werden, kann der Computer verschiedene Fehler erkennen und eine Fehlermeldung oder einen Fehlerkode anzeigen. Anhang A gibt eine vollständige Liste der Fehlerkodes mit Definitionen und Beseitigungsverfahren.

2.2 PROLOGUE-SYSTEMPROGRAMME

Die in diesem Kapitel beschriebenen Standardprogramme sind:

Objektprogramme	Beschreibung	Einzelheiten in Abschnitt
/-O	Inhaltsverzeichnis -- zeigt Inhalt (Dateien) einer Platte oder Diskette an.	2.3
ED-O	Text-Editor -- verwendet zum Eingeben und Ändern von Quellprogrammen.	2.4
CP-O	Kopierprogramm -- verwendet zum Anlegen eines Volume und zum Kopieren, Löschen, Anlegen und Umbenennen von Dateien.	2.5
CPS-O	Sektor-Kopie -- verwendet zum Kopieren von Disketten, ganz oder teilweise (gleichgültig ob von PROLOGUE generiert oder nicht).	2.6
STATUS-O	Zustand -- zeigt den gespeicherten Zustand der magnetischen Peripheriegeräte an.	2.7
PATCH-O	Direktkorrektur -- Zum Überprüfen und Ändern der auf einer Diskette aufgezeichneten Daten.	2.8

Objektprogrammname		Einzelheiten in Abschnitt
DATE-O	Datum -- bietet dem Benutzer die Möglichkeit einer Datums - und Zeiteingabe in das System.	2.9
ASG-O	Assign -- bearbeitet eine Verkettungsdatei, die hauptsächlich aus einer Serie von PROLOGUE-Kommandos besteht, die für die automatische Durchführung einer Reihe von Vorgängen ausgeführt werden können. PROLOGUE nimmt die Kommandos von einer Datei an, anstatt von der Tastatur.	2.10
FM -O	Format -- bringt eine Diskette auf die von PROLOGUE geforderte Struktur.	2.11

2.3 /-INHALTSVERZEICHNIS

Dieses Kommando sorgt für eine Auflistung der auf dem Volume enthaltenen Dateien in der spezifizierten Einheit. Das Inhaltsverzeichnis kann am Bildschirm angezeigt oder auf dem Drucker ausgegeben werden.

Die Auflistung eines Inhaltsverzeichnisses kann alle Dateien des Volumes, eine Gruppe von Dateien, die einer mehrdeutigen Bezugnahme entspricht, oder eine einzelne Datei enthalten.

2.3.1 Inhaltsverzeichnis eines Volumes

Diese Form des Inhaltsverzeichnis-Befehles bewirkt eine Auflistung aller Dateien eines Volumes. In Option können die Granule-Plätze der Datei auf der Diskette ausgegeben werden.

Syntax:
[Einheit 1]/[,Einheit 2][,OPTIONEN] (cr)

Hier bedeuten:

Einheit 1 - Spezifiert das Inhaltsverzeichnis-Programm und in Option die Einheit , in welcher es sich befindet. Vorgabewert ist die Systemeinheit.

Einheit 2 Einheit, die das Volume enthält, dessen Inhaltsverzeichnis aufgelistet werden soll. Wenn Einheit 2 nicht eingegeben wird, wird die Benutzereinheit eingesetzt.

Optionen (in beliebiger Reihenfolge):

LIS=LO - Das Inhaltsverzeichnis wird über den Drucker ausgegeben. Wenn LIS=LO nicht verwendet wird, erscheint das Inhaltsverzeichnis als Bildschirmanzeige.

GR - Spezifiziert eine Auflistung der physikalischen Lage auf der Diskette (granuleweise) für jede Datei, wie nachstehend im einzelnen beschrieben.

2.3.1.1 Anzeigeformat

Die erste Zeile einer jeden Inhaltsverzeichnis-Anzeige hat folgendes Format:

VOL:Volname File Nb=XX LG.GR:YY Used:N1/N2

Hier bedeuten:

VOL	- Volume-Name
File Nb=XX	- XX ist die größtmögliche Zahl der Dateien, die das Volume enthalten kann
LG.GR:YY	- Länge der Granule (Sektorenzahl)(für dieses Volume).
Used:N1/N2	- N1= Gesamtzahl verfügbarer Granules auf dem Volume. N2= Zahl der gegenwärtig verwendeten Granules (wenn diese Zahlen gleich sind, ist das Volume voll).

Die Dateien werden in verschiedenen Spalten und Zeilen in alphabetischer Reihenfolge nach Namen und Typ aufgelistet, wie nachstehend illustriert. Die Schlüsselworte erscheinen nicht im Inhaltsverzeichnis.

Anzeige ohne Option GR

VOL:PRO 12	File Nb=63	LG.GR:16	Used 40/31	
/-0	AZM -0	BASIC-0	COMPRES-C	CP-0
CPS-0	ED-0	EDL-0	EXTEND-S	MM-0
STATUS-0	SYST-0	TOT-S		

Anzeige mit Option GR

Wenn die Option GR gewählt wurde, entspricht das Anzeigeformat nachstehendem Bespiel:

VOL:Volname	File Nb=XX	LG.GR:YY	Used:N1/N2
Dateiname 1:	A:(B1,C1)	(B2,C2)...	
Dateiname 2:	A:(B1,C1)	(B2,C2)...	
Dateiname 3:	...		

Die Titelzeile der Anzeige erscheint wie oben. Der Name jeder Datei wird aufgelistet, darauf folgt das Verzeichnis der in jedem Block der Datei zugewiesenen Granules.

Granules. Hier bedeuten:

A	- Gesamtgröße der Datei in Granules
Bx	- Sequenznummer des ersten Granules in Block x.
Cx	- Sequenznummer des letzten Granules in Block x.

Damit erhält der Benutzer die Möglichkeit, die genaue physische Lage eines Programmes zu ermitteln, was für die Benutzung bestimmter Dienstprogramme notwendig ist.

Nehmen wir zu Beispiel an, eine Datei wird folgendermaßen aufgelistet:

```
 /-0      :    5:    (8,12)
CP-0      :    2:    (16,17)
TR-0      :    4:    ( 4,7)
SAM-S     :    4:    (14,15) (18,19)
```

So enthält /-0 5 Granules, die Nummern 8,9,10,11 und 12.

Beispiele für Inhaltsverzeichnis-Kommandos:

/(cr)	- Inhaltsverzeichnis der Disketten in der Benutzereinheit wird auf den Bildschirm übertragen. /-Programm wird von der Systemeinheit gelesen.
/,FLØ (cr)	- Inhalt der Diskette in Floppy-Einheit Ø wird am Bildschirm angezeigt. /-Programm von der Systemeinheit.
/,FLØ,GR (cr)	- Auflistung aller Dateien auf Einheit Ø mit ihren Kenndaten (Größe und Lage).
/,FLØ,LIS=LO,GR(cr)	- Wie oben, aber direkt auf dem Drucker ausgegeben.
FL1./,FLØ,LIS=LO (cr)	- /-Programm wird von Einheit 1 gelesen, Inhalt von Einheit Ø auf dem Drucker ausgegeben.

2.3.2 Inhaltsverzeichnis ausgewählter Dateien

Diese Form des Inhaltsverzeichnisses führt zu einer Auflistung einer oder mehrerer Dateien eines Volumes.

Syntax:

[Einheit 1.], [Einheit 2.] Dateiname [,Optionen] (cr)

Hier bedeuten:

Einheit 1	- Spezifiert das (Inhaltsverzeichnis) - Programm und in Option die Einheit, auf der es sich befindet. Vorgabewert ist die Systemeinheit.
Einheit 2	- Spezifiert die Einheit, in der sich das Volume mit dem spezifierten Dateinamen befindet. Vorgabewert ist die Benutzereinheit.
Dateiname	- Jeder gültige Dateiname im Standard-PROLOGUE-Format [Einheit.] Name [- Typ:Schlüssel].

Der Dateiname kann spezifisch oder mehrdeutig sein.

Spezifischer Dateiname -- Name[- Typ],wie TEST-O

Mehrdeutiger Dateiname -- Entweder am Anfang oder am Ende des Namens können ein oder mehrere Zeichen und/oder die Typenbezeichnung durch einen Stern ersetzen werden.

PROLOGUE wird nur die Dateien ausgegeben, in denen die vor oder nach dem Stern benannten Zeichen vorkommen.

Bespiele:

A*-O	- Spezifiziert alle Dateien vom Typ O, die mit dem Buchstaben A beginnen.
TO-	- Spezifiziert alle Dateien eines beliebigen Typs, deren Name mit TO endet.
*-S	- Spezifiziert alle Dateien vom Typ S.

Optionen - Wie in der Beschreibung des Volume-Inhaltsverzeichnis-Kommandos.

Dieses Kommmando bewirkt Inhaltsverzeichnis-Auflistungen für die ausgewählten Dateien im gleichen Format wie das oben in 2.3.1 besprochene Volume-Inhaltverzeichnis.

Bespiele:

FL1./,FLØ.TEST-*,LIS=LO(cr)	- Programm vom Laufwerk 1 gelesen, alle Dateien mit Namen TEST von Laufwerk Ø werden auf dem Drucker ausgegeben.
/,FLØ.*-S,GR (cr)	- Das Programm wird von der Vorgabeeinheit eingelesen, alle Dateien vom Typ-S auf Einheit Ø werden zusammen mit ihren Deskriptoren (Größe,physische Anordnung) auf dem Bildschirm angezeigt.

2.4 ED-TEXT-EDITOR

2.4.1 Allgemeines

Der PROLOGUE Text-Editor wird zum Ändern bestehender Quelldateien (gewöhnlich vom Typ-S) und zum Eingeben neuer Quelldateien verwendet.

Der Text wird von einer Quelldatei in die neue Quelldatei auf einer sequentiellen zeilenweisen Basis vom Beginn des Textes bis zur Endmarke der Datei aufbereitet. Jede Zeile kann geändert, gelöscht oder unverändert übertragen werden. Es besteht die Möglichkeit, eine oder mehrere Einzelzeilen oder Textblöcke einzufügen.

Alle Textzeichen sind im Speicher als 8 Bit-ASCII-Kodes dargestellt. Eine Zeile wird durch cr (Freigabe) abgeschlossen (entspricht der Konfiguration ODH OAH). Das letzte Zeichen einer jeden Quelldatei muß das Kennzeichen für das Dateiende sein, CTRL-Z.

2.4.2 Syntax

Kommandosyntax:

[Einheit] ED, Dateiname 1[,Optionen] (cr)

Die Optionen sind hier: { DEST=Dateiname 2 / CF }

Beide spezifizierten Dateien müssen Quelldateien sein. Meist sind sie vom Typ-S, aber jede Typenbezeichnung ist erlaubt. Wenn die Typenbezeichnung(en) nicht angegeben wird (werden), werden Dateien vom Typ-S angenommen.

Dieses Kommando läßt drei Arten der Programmaufbereitung zu:

1. Wenn keine Optionen angegeben werden, wird das Programm Dateiname 1 zeilenweise auf einer temporären Datei gespeichert (mit TYP $ benannt). Die Bearbeitung kann jederzeit abgebrochen werden. Dann bleibt die Originaldatei erhalten (der Name der temporären Datei kann im Inhaltsverzeichnis erscheinen).

   Wenn Sie die Aufbereitung normal zu Ende führen und den Endbefehl E eingegeben, wird verbleibender Text aus Ihrer Originaldatei auf die temporäre Datei übertragen, die dann die Bezeichnung Dateiname 1 erhält, während Ihre frühere Datei gelöscht wird.

2. Wenn die Option DEST==Dateiname 2 zur Verfügung steht, wird das mit Dateiname 1 spezifizierte Programm in Dateiname 2 aufbereitet. Am Ende der Bearbeitung enthält Dateiname 2 das neue Programm, während Dateinname 1 unverändert bleibt und als Sicherheitsdatei dient.

3. Wenn die Option CF spezifiziert ist, wird Dateiname 1 angelegt (muß zu Beginn erzeugt werden.). In diesem Fall wird zum Abschluß der Bearbeitung entweder Befehl E oder Z gegeben.

Bespiele:

- ED, FILE, CF	- Datei File-S anlegen
- ED, FILE, DEST=NEWFIL	- NEWFIL - S von FILE - S anlegen
- ED, FILE	- FILE- S überarbeiten.

2.4.3 Benutzung des Editors

Zum Benutzen des Editors:

1. Die passende Kommandozeile zum Laden von ED schreiben.
2. Wenn der Editor geladen ist, erscheint die Anzeige:
   R2E EDMIC - Vn.n (n.n ist hier die Nummer der Fassung).

   :

   Der Doppelpunkt (gefolgt vom blinkenden Cursor) ist das Editorsymbol das einen der Befehle anfordert, die nachstehend in 2.4.4 beschrieben werden.

3. Nun müssen Sie einen passenden Befehl eingeben und mit der Textbearbeitung beginnen. Gewöhnlich werden Ihre Textbearbeitungsbefehle mit (cr) abgeschlossen.

   Durch Druck auf die Leertaste wird die Anzeige zur nächsten Befehlzeile weitergeschoben. Wenn die Bearbeitung einer Anweisung abgeschlossen (oder unverändert übernommen) ist, wird der Doppelpunkt wieder angezeigt, und ein neuer Befehl erwartet. Die Bearbeitung arbeitet nur in Vorwärts-Richtung. Es besteht nicht die Möglichkeit, auf eine bereits fertiggestellte Anweisung zurückzukommen.

4. Die Möglichkeit der Korrektur von Schreibfehlern ist durch folgende Tasten gegeben:

(linker Pfeil) ←	- Setzt den Cursor um ein Zeichen zurück, ohne das übergangene Zeichen zu löschen, so daß das vorangehende Zeichen neu getippt werden kann. Kann zusammen mit der REPEAT-Funktion verwendet werden, um rasch in die Mitte einer Zeile zurückzukehren, um Korrekturen vorzunehmen.
(rechter Pfeil) →	- Verschiebt den Cursor nach rechts, ohne ein übergangenes Zeichen zu löschen.
CTRL-C	- Bringt den Cursor an den Anfang einer Anweisung zurück und löscht alle eingegebenen Zeichen.
ESC	- Verschiebt den Cursor an das Ende der Anweisung, wobei alle übergangenen Zeichen erhalten bleiben und angezeigt werden, kann an beliebiger Stelle einer Zeile verwendet werden.

5. Die TAB-Taste kann für Änderungs- und Einfügungsbefehle verwendet werden. Wenn die TAB-Taste gedrückt wird, wird der Cursor bis zum nächsten TAB-Stopp geschoben. TAB-Stopps sind in 8er Gruppen von Zeichen über den Bildschirm verteilt.

6. Das Z-Kommando des Editors bedeutet das Ende der Datei (CTRL-Z) und muß getippt werden, um eine neue Datei richtig abzuschließen, wenn die CF-Option (create file) nicht spezifiziert wurde.

2.4.4. Editorkommandos

Wenn der Editor sein Symbol in Form eines Doppelpunktes anzeigt (:), ist mit einem der nachstehend beschriebenen Editorkommandos zu antworten.
Ein ungültiges Kommando wird vom System zurückgewiesen und erscheint nicht auf dem Bildschirm. Stattdessen ertönt ein Hupton.

Als erste Kommandos dürfen M,C,R oder D nicht eingegeben werden, weil diese Kommandos eine Anzeige der laufenden Zeile bewirken und eine solche noch nicht vorhanden ist.

2.4.4.1 Kommandos zum Bewegen des Cursors

Kommando	Funktion
Blank (Leertaste drücken)	Zu nächsten Anweisung übergehen.
A (Freigabe)	Rückt Anzeige zum Ende der Datei vor , mit Anzeige "END OF FILE". Nun kann das Kommando I verwendet werden, um zusätzlichen Text einzufügen, oder die Bearbeitung kann mit den Kommandos S,E oder Z abgeschlossen werden.

2.4.4.2 Einfügen, Ändern, Löschen

Kommando	Funktion
I (Einfügen) (Insert)	Zum Einfügen einer oder mehrerer Anweisungen nach der eben angezeigten Zeile. Dieses Kommando wird verwendet, um die Quelldatei mit neuen Zeilen zu ergänzen. Der Editor zeigt für jede neue Zeile ein aus 5 Punkten bestehendes Symbol an. Durch Drücken der Taste cr (Freigabe) vor Eingabe einer Anweisung wird die Einfügefunktion beendet. Dann wird die nächste Anweisung der Quelldatei angezeigt und der Doppelpunkt als Aufforderung zu einem Kommando erscheint. Die Leertaste drücken, wenn die nächste Anweisung angezeigt werden soll.
R (Korrektur)	Löscht die gerade angezeigte Anweisung und ersetzt sie durch eine oder mehrere neue, die genau wie im Kommando I (Einfügung) eingegeben werden.
J: Dateiname (cr)	Dieses Kommando fügt den Inhalt einer Quelldatei ein (Typ-S), der der angezeigten Anweisung folgt. Die einzufügende Datei muß mit CTRL-Z (Dateiendekennzeichen) abgeschlossen sein, das nicht kopiert wird. Wenn der Ursprungskode übertragen ist, wird erneut die Aufforderung mit einem Doppelpunkt angezeigt. Dann können Sie den hinzugefügten Text durchgehen und überarbeiten. Am Ende des neuen Textes erscheint die Anzeige END OF INSERTION, und Sie können mit der Bearbeitung Ihres Originaltextes fortfahren. Während dieses Befehls müssen Ursprungsdatei und Bestimmungsdatei zugänglich sein.
M (Änderung) (Modify)	Zum Ändern der zuletzt angezeigten Anweisung. Der Cursor wird an den Beginn der Anweisungszeile gesetzt und Sie können mit der Änderung beginnen. Durch Schreiben neuer Zeichen werden die in der darüberliegenden Zeile angezeigten Zeichen ersetzt. Ein Druck auf den Rechtspfeil bringt den Cursor um eine Stellung nach rechts, wobei das übergangene Zeichen erhalten bleibt und in der Zeile erscheint. (Mit dem Linkspfeil läßt sich der Cursor nach links verschieben). An jedem beliebigen Punkt bewirkt ein Druck auf ESC eine Verschiebung der Marke an das Ende der Anweisung, und die übergangenen Zeichen bleiben erhalten. Eine Freigabe an jedem beliebigen Punkt beendet die Anweisung, wobei alle rechts von der Freigabe (cr) liegenden Zeichen gelöscht werden. Auf Wunsch kann M erneut eingegeben werden, um die Anweisung weiter zu verändern.

C (Kommentar)
(Comment)

Wie M, jedoch wird der Cursor an das Ende der Anweisung gebracht, so kann das Ende abgeändert werden, oder Kommentare können hinzugefügt werden.

Dn (Löschen)

löscht n Zeilen Text der Quelldatei, (wobei n=1 bis 9), beginnend mit der eben angezeigten Zeile. In diesem Fall wird Zeile n-1 angezeigt, mit Doppelpunkt. DØ bewirkt lediglich die Löschung eines irrtümlich getippten D.

2.4.4.3 *Suchkommando*

,string (cr)
.string (cr)

Wird verwendet, um Zeichenfolgen im Text von der jeweiligen Zeile an beginnend, aufzufinden.

,string — Auffinden des ersten Auftretens der spezifierten Zeichenfolge an irgendeiner Stelle im Text.

.string — Auffinden des ersten Auftretens der spezifizierten Zeichenfolge, am Anfang einer Zeile.

* — Wiederholt vorangehenden Suchbefehl.

Der Editor sucht den ganzen in der Datei verbleibenden Text ab, (von der gegenwärtigen Zeile an), bis er die Zeichenfolge findet. Alle während der Suche übergangenen Zeilen werden unverändert in die Zieldatei übertragen, aber nicht angezeigt. Die Zeile die die spezifizierten Folge enthält, wird angezeigt. Wenn die Folge nicht gefunden wird, erscheint die Anzeige END OF FILE (Ende der Datei).

Eine Folge kann eine Länge von bis zu 15 Zeichen haben, darin vorkommende Leerstellen nicht mitgerechnet.

D,string (cr)

Wie die oben beschriebenden Zeichenfolgen, mit der Ausnahme, daß alle Zeilen nach dem Kommando bis zur gesuchten Folge **ausschließlich** gelöscht werden.

D,string - Löscht Zeilen bis zum ersten Auftreten der Zeichenfolge an irgendeiner Stelle im Text.

D.string - Löscht Zeilen bis zum ersten Auftreten der Zeichenfolge am Beginn einer Zeile.

* - Wiederholt ein früheres String-Kommando.

Hinweis : Diese Kommandos können verwendet werden, um Textelemente zu isolieren und zu löschen. Hierbei ist aber **Vorsicht angebracht.** Hier könnte ein Tippfehler eine Menge Text löschen, den Sie erhalten wollten.

2.4.4.4 Ende der Bearbeitung

Z (cr) (Ende der Datei)	Beendet ordnungsgemäß alle PROLOGUE-Quelldateien und **muß eingegeben werden, um ein neu geschaffendes Programm zu beenden, wenn die Option CF nicht in der Befehlszeile spezifiziert wurde.** Wenn dieses Kommando während der Ausgabe eines bestehenden Programmes erteilt wird, wird die neue Datei abgeschlossen und jeder in der Quelldatei verbleibende Text wird ignoriert.
E (cr) (Ende der Bearbeitung)	Dieser Befehl kann an jedem Punkt eines Programms verwendet werden, um die Bearbeitung zu beenden. Der verbleibende Text in der Quelldatei wird in die Zieldatei übertragen und das System kehrt dann zu PROLOGUE zurück. **Dieser Befehl kann nicht für den ordnungsgemäßen Abschluß einer neuen Quelldatei verwendet werden, außer wenn diese Datei mit der Option CF angelegt wurde.**
S (cr)	Bricht die Bearbeitung ab, löscht alle Veränderungen des Quellprogramms und kehrt zu PROLOGUE zurück.

2.4.4.5 Zeilenlängenkommando

L:nn (cr)	Legt die maximale Textzeilenlänge als nn Zeichen für Anwendungen fest, in denen die Zeilenlänge auf 72 Zeichen begrenzt ist (Lochkartenbild). Wenn versucht wird, diese Länge zu überschreiten, wird das Zeichen zurückgewiesen und ein Hupton ertönt. Die vorgegebene Zeilenlänge beträgt 255 Zeichen.

2.5 CP-Kopier-Dienstprogramm (Copy)

Das Kopierprogramm ist die Routine, die zum Anlegen eines PROLOGUE-Volumes und zum Erzeugen und Behandeln von Dateien verwendet wird.

2.5.1 Anlegen eines Volumes (Create Volume)

Alle im PROLOGUE-System verwendeten Platten und Disketten müssen im richtigen Format initialisiert werden und einen Identifizierungs-Volume-Namen erhalten. Dieses Kommando kann ein Volume schaffen oder ein bestehendes Volume neu benennen. Das in Abschnitt 2.11 beschriebene Dienstprogramm wird für die Formatierung der Disketten verwendet.

Syntax:

[Einheit.] CP,CV, Einheit, Volname [,Optionen] (cr)

Hier bedeuten:

Einheit. CP
- Spezifiziert das CP-Programm, und in Option die Einheit, auf der es sich befindet. Vorgabe-Wert ist die Systemeinheit.

CV
- Bedeutet Anlegen von Volumen (nicht Kopieren)

Einheit
- Spezifiziert die Einheit, auf der das Volume angelegt werden soll.

Volname
- Name des neu anzulegenden Volume, bis zu 8 Zeichen. Jede beliebige Kombination von Buchstaben, Ziffern und Spezialzeichen ist gestattet, mit Ausnahme des Kommas.

LGR = N
- N Spezifiziert die Anzahl der Sektoren pro Granule, und zwar 1,2,4,8,16,32,64,128,0 (mit 0 wird 256 spezifiziert). Der Vorgabe-Wert ist 16.

NBFIC = Y
- Y spezifiziert die Anzahl der Dateien, die im Inhaltsverzeichnis dieses Volumes eingetragen werden können. Die ersten Spuren der Platte/Diskette ist dem Inhaltsverzeichnis vorbehalten, mit folgenden Raumanforderungen:

  bis 31 Dateien - 1 Spur von 16 Sektoren
  bis 95 Dateien - 2 Spuren
  bis 159 Dateien - 4 Spuren
  bis 191 Dateien - 3 Spuren
  bis 255 Dateien - 5 Spuren

  Der Benutzer sollte kein größeres Inhaltsverzeichnis anlegen als notwendig (besonders auf einer Floppy-Disk), um Arbeitsraum auf der Platte zu erhalten. Vorgabe-Wert ist 63. (Dafür werden 2 Vorgabe-Granules von 16 Sektoren/Granule benötigt).

  Wenn die Platte nicht formatiert wurde, wird eine Fehlermeldung ausgegeben. Zum Formatieren der Diskette ist das FM-Dienstprogramm zu verwenden , siehe Abschnitt 2.11.

  Wenn das Volume bereits Dateien enthält, wird der vorhandene Volume-Name in der Form VOLNAME (Y/N) angezeigt: wenn Y (Yes) eingegeben wird, wird dem Volume der neue Volume-Name zugewiesen, mit den in der Befehlszeile spezifierten Merkmalen, und die Dateien im alten Volume **werden gelöscht.** Dies trifft auch für den Fall zu, daß Sie ein Volume mit seinem ersten Namen neu benennen. Für die

Neubenennung eines Volume ohne Löschung seines Inhaltes können Sie das Kommando Rename Volume (2.5.7) benutzen. Jede andere Antwort als Y auf die Prüfanzeige wird das Kommando löschen und Volume-Name und Dateien unverändert belassen.

CP,CV,FL1,USER1,LGR=8 NBFIC=159 (cr) - Der Benutzer gibt das Kommando ein, und das System liest das CP-Programm von der Systemeinheit.

Dann wird das Volume angelegt und mit dem Namen USER1 bezeichnet, es wird auf einen Inhalt von 8 Sektoren pro Granule festgelegt, mit Inhaltsverzeichnisraum für bis zu 159 Dateien.

CP,CV,FL1,USER1 (cr) - Ähnlich wie obiger Befehl, jedoch wird der Vorgabe-Wert von 16 Sektoren pro Granule angenommen.

Wenn die Diskette nicht formiert wurde, wird eine Fehlerbotschaft ausgegeben. Die Diskette mit dem in Abschnitt 2.11 beschriebenen FM-Dienstprogramm formatieren.

2.5.2 **Anlegen einer Datei** (Create a File)

Dieser Befehl legt eine Datei auf einem vom PROLOGUE-Dienstsystem generierten Volume an. Der Dateiname wird in das Inhaltsverzeichnis aufgenommen, aber es werden keine Granules zugewiesen. Wenn im Inhaltsverzeichnis kein Raum mehr frei ist, wird ein Fehlerkode ausgegeben.

Syntax:

[Einheit.] CP, CF, Dateiname (cr)

Hier bedeuten:

Einheit. CP - Spezifiziert das CP-Programm und in Option das zugehörige Laufwerk.

CF - Anlegen der Datei.

Dateiname - Jeder Dateiname darf nur einmal auf dem Volume vorkommen (Name,Typ,Schlüssel). wenn kein Laufwerk spezifiziert ist, wird die Datei auf dem Volume in der Systemeinheit angelegt.

-CP,CF,SAMBLE (cr) - Legt eine Datei mit Namen SAMBLE an, mit Leerstellen als Dateityp und Schlüssel.

-CP,CF,SAMBLE-S (cr) - Legt eine Datei mit Namen SAMBLE-S an, mit Leerstellen als Schlüssel. Die Datei wird in der Systemeinheit angelegt.

-CP,CF,FLØ.SAMBLE-1:1 (cr) - Legt eine Datei auf FLØ, mit Namen SAMBLE-S:1 an. Der Schlüssel (Kennwort), :1, erscheint nicht in der Anzeige.

Hinweis: Ein Fehlerkode (Siehe Anhang A) wird ausgegeben, wenn der spezifizierte Dateiname bereits auf dem spezifizierten Laufwerk vorhanden ist.

2.5.3 **Löschen einer Datei** (Suppress a File)

Dieses Kommando löscht die spezifizierte Datei auf einem vom Dienstsystem angelegten Volume. Die dieser Datei zugewiesenen Granules werden freigegeben und der Dateiname wird aus dem Inhaltsverzeichnis entfernt. Wenn die Datei im Inhaltsverzeichnis mit einem Dateityp vor dieser vorhanden ist, muß dieser Dateityp spezifiziert werden, sonst wird der Fehler-Kode "Datei nicht gefunden" (ERR.40) ausgegeben.
Wenn die Schlüssel für die spezifizierte Datei andere Zeichen als die ASCII-Leerstellen sind (Vorgabe-Wert) müssen sie richtig spezifiziert werden, sonst wird eine Fehleranzeige ausgegeben und es erfolgt keine Löschung.

Syntax

[Einheit.] CP, SF, Dateiname (cr)

2.5.4 **Umbenennen einer Datei** (Rename a File)

Dieses Kommando führt zur Umbenennung einer Datei ohne Änderung ihres Inhaltes.

Syntax

[Einheit.] CP, RF, Dateiname 1, Dateiname 2 (cr)

Die mit Dateiname 1 spezifizierte Datei erhält den Namen Dateiname 2. Dateiname 1 muß genau, wie er im Inhaltsverzeichnis erscheint, spezifiziert werden, einschließlich Dateityp und Schlüssel (Kennwort). Die vollständige Bezeichnung oder ein Teil davon Name, Typ, und Schlüssel, kann geändert werden. Die spezifizierte Einheit (falls sie spezifiziert wurde) muß für beide Dateinamen die gleiche sein.

2.5.5 **Kopieren einer Datei** (Duplicate a File)

Dieses Kommando ermöglicht das Kopieren einer Datei wie folgt:

1. Kopie von einer Einheit in die andere, unter Zuweisung des gleichen oder eines anderen Namens für die Zieldatei.

2. Kopie einer Datei auf der gleichen Einheit unter Zuweisung eines anderen Dateinamens.

3. Kopie einer Datei auf den Bildschirm oder einen Drucker.

Syntax

[Einheit.] CP, DF, Dateiname 1 ,Dateiname 2 oder Einheit oder LO

a b

Die Daten von Dateiname 1 werden in Dateiname 2 übertragen. Wenn Dateiname 2 auf der spezifizierten Einheit nicht existiert, wird er generiert.

Wenn er bereits vorhanden ist, werden alle Daten, die er enthält,gelöscht. Nach der Kopie sind beide Dateien identisch.

Es sei darauf hingewiesen, daß Einheiten,Dateityp und Schlüssel für dieses Kommando spezifiziert werden müssen. Dateiname 1 und Dateiname 2 müssen verschieden sein. Zum Beispiel: FLO.TEST-S und FL1.TEST-S.

Andere Optionen für Parameter b sind:

1. Einheit – Wenn nur die Einheit spezifiziert ist, wird der Zieldatei der gleiche Name wie der Quelldatei zugewiesen.

   Beispiel: CP, DF, FLØ.FILE-S, FL1 ist gleichwertig mit CP, DF, FLØ.FILE-S, FL1. FILE-S

2. LO – Drucker

(Konsolenausgabe (Bildschirm) ist Vorgabeoption, wenn Parameter b nicht eingegeben wird.

2.5.6 Kopieren eines Volumes (Duplicate a Volume)

Dieses Kommando bewirkt die Übertragung der Dateien eines Volumes auf ein zweites Volume.

Syntax

[Einheit.]CP, DV, Vol 1, Vol 2[,RZ](cr)

Ohne Option RZ

Dies führt zur Übertragung aller Dateien aus Vol 1 auf Vol 2, das vorher vom System angelegt werden muß. Jede Datei auf Vol 2 mit dem gleichen Namen wie eine zu übertragende Datei wird gelöscht. Alle anderen Dateien auf Vol 2 bleiben unverändert. Bevor die Dateien übertragen werden, wird ihr Name auf dem Bildschirm angezeigt. Wenn bei diesem Vorgang das Inhaltsverzeichnis voll wird, wird das Kopieren abgebrochen und ein Fehler wird angezeigt.

Mit Option RZ

Wenn die Option RZ im Befehl spezifiziert wird, werden alle in Vol 2 vorhandenen Dateien gelöscht, bevor eine Datei von Vol 1 übertragen wird. Dieses Kommando wird wie folgt bestätigt:

1. Es wird folgende Meldung angezeigt:

   Volume-Name (Y/N): Hier ist der Volume-Name der Name des Zielvolumes.

2. Sie müssen antworten:

   Y (cr) (Ja) Alle in Vol 2 vorhandenen Dateien werden gelöscht.

   N (cr) (Nein) Das Kommando wird rückgängig gemacht, im neuen Volume erfolgt keine Änderung.

Dieser Vorgang bewahrt die Merkmale des neuen Volumes:

* Volumename
* Länge der Granules
* Maximale Anzahl der Dateien im Inhaltsverzeichnis.

2.5.7 **Umbenennen eines Volumes** (Rename a Volume)

Dieses Kommando benennt ein Volume um, ohne irgendwelche der auf diesem Volume aufgezeichneten Daten zu ändern. In Option kann die Granule-Zahl aus dem Volumen geändert werden.

Syntax:

[Einheit.] CP, RV, Einheit, NeuVolName (cr)

Hier bedeuten:

Einheit. CP	- Kopierprogramm
RV	- Spezifiziert Umbenennungsfunktion Volume
Einheit	- Spezifiziert die Einheit, auf der sich das Zielvolume befindet.
NeuVolName	- Dem Volume zugewiesener neuer Name. Dafür kommt eine beliebige Kombination von bis zu 8 Buchstaben (außer Komma) in Frage.

Folgende Bedingungen können oder müssen gegeben sein, wenn der Befehl ausgeführt wird:

1. Wenn der Befehl ausgeführt wird, wird folgende Meldung angezeigt:

VolName (Y/N)	- Hier ist VolName der laufende Name des Zielvolumes.

Sie müssen antworten:

Y (cr) (Ja)	- Das Volume wird neu benannt (unter Beachtung der in 2 und 3 beschriebenen Bedingungen).
N (cr) (Nein)	- Das Kommando wird rückgängig gemacht, ohne Änderung des Zielvolumes.

2. Wenn PROLOGUE eine Anzahl von Spuren pro Diskette findet, die kleiner ist als die im Inhaltsverzeichnis dieser Diskette genannte Spurenzahl, wird die nachstehende Fehlermeldung ausgegeben und das Volume bleibt unverändert:

   INCOMPATIBILITY BETWEEN NUMBER OF GRANULES OF THE VOLUME AND INFORMATION RETURNED BY DISK CONTROLLER.

Wenn dieser Fall vorkommt, überprüfen Sie, ob eine richtig für Ihre Hardware aufgebaute PROLOGUE-Version verwendet wird.

3. Wenn PROLOGUE eine größere Spurenanzahl als die im Disketten-Inhaltsverzeichnis genannte angibt, wird folgende Meldung angezeigt:

   READJUSTMENT OF NUMBER GRANULES ON VOLUME, (Y/N):

Sie müssen antworten:

Y (cr) (Ja)	- Das Volume wird umbenannt und die Spurenzahl wird berichtigt. In diesem Fall muß die Diskette vorher für die neue Spurenzahl formatiert werden.
N (cr) (Nein)	- Das Volume wird umbenannt, aber es erfolgt keine Berichtigung.

Dies wird in zwei allgemeinen Fällen zutreffen:

a. Wenn das CPS (Sektorenkopier-) - Dienstprogramm eine einseitige Quell-Diskette auf eine doppelseitige neue Diskette übertragen hat. In diesem Fall wird nur die Hälfte des zur Verfügung stehenden Platzes verwendet und die hier beschriebene Neuanpassung muß vorgenommen werden, um sämtliche auf der doppelseitigen Diskette verfügbaren Spuren zu verwenden.

b. Wenn eine Diskette mit N Spuren in einem System eingesetzt wird, dessen Antrieb mehr als N Spuren gestattet.

2.6. CPS - Sektorenkopie

Dies ist eine physische Kopierroutine, die eine Diskette/Platte oder einen Teil einer Diskette/Platte kopiert, ohne die Daten zu ändern. Sie dient zum Kopieren von Disketten oder Programmen, die in einem anderen Format als PROLOGUE-Volumes arbeitet.

Es sind zwei Kopierarten möglich:

1. Gesamtkopie - vollständige Kopie von einer Diskette, ohne Änderung der Struktur oder der Position der Dateien. Der gesamte Inhalt der Zieldiskette wird überschrieben. Nur die Spurenzahl entsprechend dem speicherresidenten PROLOGUE wird kopiert. Vergewissern Sie sich, daß Sie eine für Ihre Hardware passende PROLOGUE-Fassung benutzen.

2. Freie Kopierart - Kopie der vom Benutzer nach Spur und Sektor oder nach absoluter Sektoradresse ausgewählten Daten. Dies kann nützlich sein, um im Fehlerfall Daten von einer Backup-Diskette zu übertragen. Es kann auch für den Programmierer nützlich sein, um binäre Dateien verschiedenen Stellen in einem Volume zuzuweisen.
(Backup = Sicherheitskopie).

2.6.1 Gesamtkopie

2.6.1.1 Beschreibung

In dieser Betriebsart wird eine ganze Diskette kopiert, und zwar spurweise, anders als in der langsameren freien Kopierart, wo ein beliebiger Abschnitt einer Diskette sektorweise kopiert wird.

Syntax:

[Einheit.] CPS, Einheit-S, Einheit-D (cr)

Hier bedeuten:

Einheit-S - Quelleinheit wie z.B. FLØ.
Einheit-D - Zieleinheit wie FL1.

Wenn die Zieldiskette ein PROLOGUE-Volume ist, wird die Volume-Bezeichnung für die Bestätigung des Vorganges wie folgt angezeigt: NAME (Y/N). Zum Durchführen des Vorganges ist Y (Ja) einzugeben.

Beachten Sie folgende Hinweise:

1. Die Disketten müssen die gleichen Merkmale aufweisen - Anzahl Spuren pro Seite; Anzahl Sektoren pro Spur.

2. Das CPS-Programm behält die Merkmale einer einseitigen Diskette auf der Zieldiskette bei; d.h. es gestattet nur die Speicherkapazität einer einseitigen Diskette, auch auf einer doppelseitigen Zieldiskette. Dies kann anschließend mit der Volumenumbenennung, RV, Kommando des CP-Dienstprogramms, geändert werden.

3. Zu Beginn des Kopiervorganges wird die Anzahl der noch zu kopierenden Spuren angezeigt. Diese Zahl geht mit jeder kopierten Spur zurück.

4. Wenn alle Spuren richtig kopiert sind, bleibt als Anzeige der noch zu kopierenden Spur Ø, darauf folgt ein Bindestrich "-".
   Es erfolgt eine Rückkehr zu PROLOGUE und das Symbol → wird angezeigt.

2.6.1.2 Fehleranzeige

Zusätzlich zu den normalen PROLOGUE - System - Fehlern werden folgende Fehler erfaßt und angezeigt:

Fehleranzeige	Beschreibung
SUPPORT UNITS DIFFERENT	Die Merkmale der Quell- und Zieleinheiten unterscheiden sich.
NON-SECTORED DEVICE	Versuch, eine Systemeinheit zu wählen, die keinen logischen Sektorenaufbau besitzt (z.B. Kassette).
INCORRECT OPTION	Versuch, von einer einseitigen auf eine doppelseitige Diskette zu kopieren, wenn das System nur ein einseitiges Laufwerk besitzt.
UNSER MEMORY OVERFLOW	Der verfügbare Benutzerspeicher ist nicht groß genug, um spurweises Kopieren zu gestatten. Versuchen Sie, sektorweise in freier Kopierart zu übertragen.

2.6.2. Freie Kopierart

Syntax siehe Abschnitt 2.6.2.2 (nächste Seite)

2.6.2.1 Beschreibung

Diese Kopierart gestattet es, Teile einer Diskette wie folgt zu kopieren:

Quelle	Ziel	Durch
1. beliebige Spur	beliebige Spur	Ganze Spur, mit Angabe der Anzahl aufeinanderfolgender Spuren.

2. Beliebige Spur, mit Sektorbeginn auf dieser Spur (wenn kein Sektor bestimmt wird, Beginn bei Sektor Ø).	Beliebige Spur mit Sektorbeginn auf dieser Spur (wenn kein Sektor bestimmt wird, Beginn bei Sektor Ø)	Anzahl aufeinanderfolgender Spuren oder Anzahl aufeinanderfolgender Sektoren
3. Beliebige absolute Sektoradresse (z.B. Ø bis 559 für einseitige Disketten)	Beliebige Sektoradresse	Beliebige Sektorenanzahl

das Kopieren erfolgt sektorweise, deshalb würde für die Kopie einer kompletten Diskette sehr viel Zeit benötigt (z.B. 560 Zugriffe für eine einseitige Diskette).

2.6.2.2 Verfahren

Das Verfahren für die Kopie wird nachstehend beschrieben. Die erste Spalte enthält schrittweise Anweisungen, die zweite gibt die Anzeigen an, die auf dem Bildschirm erscheinen. Die Antworten des Benutzers auf die Computerforderungen sind unterstrichen.

Anweisungen	**Anzeige**
1. Als Antwort auf das PROLOGUE-Symbol CPS (cr) schreiben.	- CPS (cr)
2. PROLOGUE lädt das CPS-Dienstprogramm und gibt ein Komma aus. Darauf mit Freigabe antworten.	- CPS (cr), (cr)
3. Das System zeigt ORIGIN an. Den Namen der Quelleinheit angeben, wie FLØ, FL1, usw.	- ORIGIN : FLØ (cr)
4. Das System benötigt die Spurnummer a. beim Kopieren mit Spur und Sektor irgendeine gültige Spurnummer eingeben b. für das Kopieren mit absoluter Sektoradresse (cr) eingeben.	TRACK : **T (cr)** TRACK : **(cr)** - Anzeige wird mit SECTOR überschrieben.
5. Das System fordert die Sektornummer der Quelle an. a. beim Kopieren mit Spur und Sektor die Anfangssektornummer eingeben (siehe Hinweis 1) b. beim Kopieren mit absoluter Sektoradresse die Anfangssektornummer eingeben. Wenn statt einer Sektornummer (cr) geschrieben wird, wird auf ORIGIN zurückgegangen.	TRACT : T SECTOR : S **(cr)** SECTOR : S **(cr)**
6. System zeigt DESTINATION an Name der Zieleinheit eingeben.	DESTINATION : **FL1 (cr)**
7. System fordert Spurnummer der Zieldiskette an (**muß** formatiert sein) a. für spurweises Kopieren gültige Spurnummer eingeben.	TRACK : **N (cr)**

b. für Kopieren mit absoluter Sektoradresse (cr) eingeben — TRACK : **(cr)** - Anzeige wird mit SECTOR überschrieben.

8. System fordert die Sektornummer der Zieldiskette an.

   a. wenn spur- und sektorweise kopiert wird, Anfangssektornummer eingeben (sh.Hinweis 1). — TRACK : T SECTOR : **S (cr)**

   b. wenn mit absoluten Sektoren gearbeitet wird, Anfangssektornummer eingeben. Wenn anstatt einer Sektornummer (cr) eingegeben wird, kehrt ORIGIN zurück. — SECTOR : **S (cr)**

9. Das System fordert die Spuranzahl an.

   a. eine beliebige gültige Spuranzahl eingeben — NO OG TRACKS : **N (cr)**

   b. wenn keine vollen Spuren kopiert werden sollen, (cr) eingeben. Das System fordert die Sektoranzahl an. Eine beliebige gültige Zahl eingeben. — NO OF SECTORS : **S (cr)**

10. Wenn die Anzahl der Spuren oder Sektoren eingegeben ist, zeigt das System (Y/N) an. Zur Bestätigung Ihrer Wahl Y tippen. Dadurch wird der Vorgang durchgeführt. Wenn N eingegeben wird, wird zu ORIGIN zurückgekehrt. — NO OF TRACKS : N (Y/N) **Y (cr)**

11. Wenn der Vorgang richtig durchgeführt wurde, wird in der Kommandozeile "-" angezeigt und die Aufforderung ORIGIN wird erneut angezeigt.

    Sie können nun eine weitere Kopie vornehmen oder (cr) eingeben, um zum Betriebssystem zurückzukehren.

Bitte beachten Sie:

1. Beim Kopieren mit Angabe von Spur und Sektor bezieht sich die spezifizierte Sektornummer auf Sektor 0 der ausgewählten Spur. Sie können zum Beispiel Spur 5, Sektor 100 wählen, und die Referenz bezieht sich dann auf den 100. Sektor nach Sektor 0 der Spur 5.

2. Zahlen und Adressen können in Dezimal- oder Hexadezimalkode (mit Suffix H, Beispiel 1AH) spezifiziert werden.

2.6.2.3 Fehleranzeige

Zusätzlich zu den normalen PROLOGUE-Systemfehlern können folgende Fehler festgestellt und angezeigt werden:

Fehlermeldung	Beschreibung
INCORRECT DEVICE NAME	Es wurde eine nicht vorhandene Einheit spezifiziert.
NON-SECTORED DEVICE	Versuch, eine Systemeinheit zu wählen,die keine logische Sektoranordnung besitzt. (z.B. Kassette)
INCORRECT PARAMETER	Spur oder Sektornummer unzutreffend (nur gültige Dezimal - oder Hexadezimal - Zahlen kommen in Frage).
PARAMETER TOO LARGE	Numerische Parameter (entweder direkt eingegeben oder durch Spur/Sektoreingabe berechnet) übersteigen 65,535.

Nach einem dieser Fehler wird ORIGIN angezeigt, und sie können einen neuen Versuch starten.

2.7 STATUS

Die Daten zum Betriebszustand einer jeden Diskette werden in Zählern gesammelt, die volumenunabhängig sind. Diese Tabellen werden initialisiert, sooft das System eingeschaltet oder neu geladen wird und sie zeichnen alle Lese- und Schreibprobleme auf, die im Lauf der Benutzung der Datenträger auftreten.

Diese Routine kontrolliert und zeigt den Zustand einer spezifizierten Diskette an. Sie ermöglicht die regelmäßige Kontrolle des Betriebszustandes einer Einheit während ihrer Benutzung. Die meisten Positionierungsfehler treten auf, wenn die höheren Spuren einer Diskette erreicht werden. Das Betriebssystem wird einen Zugriff bis zu 10 mal wiederholen, bevor ein Fehlerkode ausgegeben wird. Jeder Fehler wird in den Zustandszählern aufgezeichnet. Wenn die Anzahl der Fehler ziemlich groß wird, ist es wahrscheinlich Zeit für eine Justage der Laufwerke.

Syntax

[Einheit.]STATUS[,Optionen] (cr)

Optionen:

Einheit	- die ausgewählte Einheit, wie FLØ, FL1,usw. Vorgegeben ist die Benutzereinheit.
LIS = LO	- Ausgabe des Status auf den Drucker.

Der Status wird in folgendem Format angezeigt:

DEVICE FL.1 LAST ERR-XX TRACK NO.N/ SECT.NO.S

R/W EXCH	RD ERR	WRT ERR	POSIT.ERR	INV SECT
17	3	0	0	0

Hier bedeuten:

DEVICE	**- Überprüfte Einheit**
LAST ERR-XX	**- Angabe des letzten Lese- oder Schreib-oder** Positionierungsfehler, wobei XX der Fehlerkode gemäß Anhang A ist.

TRACK NO.N	- N ist die Spur, in der der letzte Fehler auftrat.
SECT.NO.S	- S ist der Sektor auf Spur N, auf dem der letzte Fehler auftrat.
R/W EXCH	- Gesamtzahl der Lese- und Schreibvorgänge dieser Einheit seit dem letzten Laden des Systems.
RD ERR	- Gesamtzahl der Lesefehler
WRT ERR	- Gesamtzahl der Schreibfehler
POSIT. ERR	- Gesamtzahl der Spurpositionierungsfehler.
INV. SECT.	- Nicht gültig für Floppy-Disk-Systeme.

Hinweis: Fehler 04 wird angezeigt, wenn der Zustand der betroffenen Einheit nicht generiert wurde.

2.8 PATCH-DIREKTKORREKTUR

2.8.1 Beschreibung

Mit dieser Routine können Sie die Daten in einem Volumensektor auf der spezifizierten Einheit prüfen und abändern. Diese Korrektur kann auf zwei Arten erfolgen:
1) durch Angabe von Dateiname und Sektor innerhalb der Datei, oder 2) durch Spezifizierung der absoluten Spur- und Sektorzahl.

Syntax

1. Mit Dateinamen: [Einheit.]PATCH, Dateiname[,NSEC=no](cr)

Hier bedeuten:

Dateiname	-	Jeder gültige Dateiname. Der Dateityp muß spezifiziert werden.
NSEC=no	-	Nummer des zu berichtigenden Sektors innerhalb der Datei, Sektor Ø ist der erste Sektor der Datei. Der Sektor kann in Dezimal- oder Hexadezimalkode angegeben werden. (Suffix H für Hexadezimalkode z.B.10H) (Vorgabewert: Ø)

2. Mit Spur/Sektor:

[Einheit.] PATCH, Einheit[,NPIS==TRACK, NSEC==Sector](cr)

Optionen:

Einheit	-	Ausgewählte Einheit (Vorgabe: Benutzereinheit)
NPIS=Track	-	Absolute Spurnummer/Vorgabe= Ø
NSEC=Sector	-	Nummer des Sektors innerhalb der spezifizierten Spur (ab Ø); spezifiziert in Dezimal- oder Hexadezimalkode (mit Suffix H). (Vorgabe= Ø).

Wenn NPIS nicht eingegeben wird, spezifiziert NSEC die absolute Sektornummer, beginnend mit Sektor Ø.

Optionen können in beliebiger Reihenfolge spezifiziert werden, sind aber durch Kommata voneinander zu trennen.

2.8.2. Korrektur eines Sektors

Wenn der Korrekturbefehl eingegeben ist, wird das System den spezifizierten Sektor aufsuchen und auf dem Bildschirm als Zeilen von 32 hexadezimalen und 16 ASCII-Zeichen ausgeben. Der Cursor ist über dem ersten Zeichen des Sektors positioniert.

Sie können eingeben:

1. **Cursorsteuerzeichen** — Pfeile nach rechts, links, oben, unten (Zeilenvorschub) für die Positionierung des Cursors auf ein beliebiges Zeichen.

2. **cr** — Verschiebt den Cursor an den Beginn der nächsten Zeile der Hexadezimalzeichen.

3. **0-9,A- F** — Das unter dem Cursor stehende Zeichen wird durch das eingegebene Zeichen ersetzt. Das neue Zeichen erscheint auf dem Bildschirm.

4. Wenn alle Korrekturen durchgeführt sind, kann eines der folgenden Kommandos eingegeben werden. (das Kommando kann unabhängig von der Position des Cursors eingegeben werden, es wird nicht angezeigt).

   a. V (cr) - Wiederholt die Anzeige mit dem neuen ASCII-Zeichen der geänderten hexadezimalen Werte. Die Änderungen sind **noch nicht** auf der Diskette aufgezeichnet.

   b. I (cr) - Drucken — Druckt den geänderten Sektor und seine neuen ASCII-Zeichen auf dem Drucker aus, und zeigt diese Daten an. Die Änderungen sind **noch nicht** auf der Diskette aufgezeichnet.

   c. M (cr) - Änderung - Schreibt den korrigierten Abschnitt auf die Diskette, und generiert dann Befehl V für die Anzeige des geänderten Sektors.

   d. ␣ (cr) - (Leerstelle) - Zeigt den nächsten Sektor an. Wenn ein Dateiname spezifiziert wurde, wird nicht über den letzten Sektor der entsprechenden Datei hinausgegriffen.

   Vorsicht!! Immer wenn Sie ein Leerzeichen eingeben, wird der nächste Sektor angezeigt und gerade vorgenommene Änderungen am gegenwärtigen Sektor werden aufgehoben. Benutzen Sie für die Bewegung des Cursors die Cursortasten, **nicht** die Leertaste.

   e. ESC - Rückkehr zum PROLOGUE-System.

Es sei darauf hingewiesen, daß PATCH nicht für eine absolute binäre Datei auf einer Diskette verwendet werden sollte. Das Auswechseln von Werten wird dazu führen, daß sich das generierte CRC-Kontrollzeichen vom mit der Datei aufgezeichneten Zeichen unterscheidet - und die Datei kann nicht mehr eingelesen werden.

2.9 DATE-DATUM

PROLOGUE führt während der Arbeit eine Datums- und Uhrzeitenangabe. Wenn PROLOGUE eingelesen wird, wird das Datum als 1.Januar 1980 0 Uhr 0 Minuten initialisiert. Das Datumsdienstprogramm wird zum Eingeben der jeweiligen Zeit und des jeweiligen Datums verwendet.

Syntax:

1. [Einheit.] DATE,DD/MM/YY/HH/MM(cr)
2. [Einheit.] DATE(cr)DD/MM/YY/HH/MM (Display)

Hier bedeuten:

DD	== Tag	(1-31)
MM	== Monat	(1-12)
YY	== Jahr	(80-)
HH	== Stunde	(0-23)
MM	== Minute	(0-59)

In Option 1 sind Tag, Monat, Jahr, Stunde, Minute jeweils als zweistellige Angaben in die Kommandozeile zu schreiben, getrennt durch Schrägstriche, ohne Leerstellen.

In Option 2 zeigt PROLOGUE DD/MM/YY/HH/MM an. Dann müssen Sie die ensprechenden Zeichen eingegeben, auf die ein (cr) folgt.
Führende Nullen dürfen entfallen.

Beispiel:
- **DATE (cr)**
- DD/MM/YY/HH/MM

  10/9/80/14/05(cr)

Antwort von PROLOGUE

Bei richtiger Eingabe wird am Ende der Befehlszeile "-" angezeigt und das PROLOGUE-Symbol → wird ausgegeben.

Wenn die Eingabe nicht stimmt (z.B.Tag 30 für Februar), wird INCORRECT am Ende der Befehlszeile angezeigt und zu PROLOGUE zurückgekehrt.

Hinweise:

1. Der eingegebene Tag muß für den entsprechenden Monat gelten, der 29. Februar ist jedes vierte Jahr gültig.

2. Dieses Programm inkrementiert nicht automatisch den Tag.

3. Über Speicheranweisungen wie PEEK sind die Zeit- und Datenzähler für die Benutzerprogramme zugänglich. Das Datum ist gespeichert, wie in der nächsten Seite beschrieben.

Hexadezimal-Adresse	Datum
009B	Jahr
009D	Monat
009E	Tag
009F	Tag des Jahres (0-365)
00A1	Stunde

00A2	Minute
00A3	Sekunde
00A4	1/10 Sekunde
00A5	1/100 Sekunde

2.10 ASG - VERKETTUNG (Assign)

Dieses Dienstprogramm wird dafür verwendet, eine Datei als Ersatz für die Tastatur zuzuweisen. Es wird eine Kommandodatei angelegt und dann als Ersatz für die Tastatur zugewiesen. So kann eine Reihe von Vorgängen automatisch durchgeführt werden, wofür jedes Kommando systemgerecht von der Kommandodatei geliefert wird (Kommandoverkettung).

2.10.1 Aufruf der Kommandodatei

2.10.1.1 Kommandoformat

Syntax :

[Einheit.] ASG, CI, Dateiname [,List] (cr)

Hier bedeuten:

ASG	- Verkettungsdienstprogramm.
CI	- Spezifiert, daß die Eingabe von einer Datei erwartet wird, und nicht von der Tastatur.
Dateiname	- Muß ein Dateiname vom Typ-S sein. Die Datei selbst muß eine Quelldatei sein, wie unten beschrieben.
List	- In Option. Liste der Parameter, die in die Quelldatei eingegeben werden müssen. Die Parameter müssen durch Kommata getrennt werden.

Wenn dieses Kommando ausgeführt wird, weist es die spezifizierte Kommandodatei auf der Diskette als Eingabequelle zu, und die Datei ersetzt dann die Tastatur des Systems. Die Tastatur kann nicht verwendet werden, solange nicht alle Anweisungen in der Kommandodatei ausgeführt sind, oder die Kommandodatei aufgrund eines Systemfehlers ausgeschaltet wird.

2.10.1.2 Format einer Kommandodatei

Die Kommandodatei ist eine Reihe von PROLOGUE-Kommandos, genau als ob sie durch die Tastatur eingegeben würden. Sie wird unter Verwendung des Editors angelegt, muß Quelldateiformat aufweisen und muß dem Typ-S zugewiesen sein.

Hinweis: Achten Sie darauf, die erforderlichen Kommandobestätigungen einzugeben. Zum Beispiel ist für die Funktionen Anlegen eines Volume und Kopieren eines Volume im CP-Kommando eine Y/N (Ja/Nein) -Bestätigung erforderlich, wenn das Zielvolume Dateien enthält. Dieses Bestätigungszeichen muß in einer eigenen Zeile erscheinen.

Explizite Kommandodatei:

Die Kommandos müssen explizit sein, wenn die Parameterliste weggelassen wird.

Bespiel: Nehmen wir an, Sie entwickeln ein Programm und möchten in regelmäßigen Anständen eine Sicherheitskopie anfertigen, ohne alle Dateien auf Ihrer Entwicklungsdiskette zu kopieren. Anstatt eine Reihe von Kommandos einzugeben, um jedesmal 5 Dateien zu kopieren, wenn Sie eine Sicherheitskopie anfertigen, können Sie eine Kommandodatei schreiben (Name AUTODUP), wie folgt:

```
CP, CV, FL1, VOL1, LGR = 16, NBFIC 31 (cr)  - Volume anlegen (neu
                                              formierte Diskette).

CP,DF,FILE1,FL1(cr)          5 Arbeitsdateien kopieren, auf Floppy 1.
CP,DF,FILE2,FL1(cr)          Vorgabewerte sind die gleichen Namen
CF,DF,FILE3,FL1(cr)          wie in den Quelldateien.
CF,DF,FILE4,FL1(cr)
CF,DF,FILE5,FL1(cr)

/,FL1,LIS=LO(cr)             Inhaltsverzeichnis der neuen Diskette
                             ausdrucken.
```

Nehmen wir an, Floppyeinheit Ø wird als Systemeinheit verwendet, dann würde die Kommandodatei durch Eingabe von

```
ASG,CI,AUTODUP(cr)
```

ausgeführt.

Implizite Kommandodatei

Wenn die Parameterliste in einem Verkettungskommando verwendet wird, muß die Kommandodatei implizite Parameter der Form $X enthalten, in denen X eine Ziffer von 1 bis 9 ist.

Ein solches Kommando würde als CP,CV,$1,$2,LGR=16,NBFIC=63,P geschrieben. $1 und $2 bedeuten hier Parameter, die im Verkettungsbefehl selbst enthalten sind.

Wenn die Verkettung mit einer gegeben Parameterliste ausgeführt wird, tritt jeder Parameter an die entsprechende Stelle in der Kommandodatei. Das heißt, der erste Parameter in der Liste wird $1 ersetzen, der zweite $2, usw.

Kommando-Verkettung (Assign)

```
ASG,CI,Dateiname,PAR1,PAR2,PAR3,PAR4,usw.
                 $1   $2   $3    $4....$9
                 ---------------------
                      Parameterliste
```

Beispiel

Die Kommandodatei in unserem obigen Beispiel könnte folgendermaßen geschrieben werden:

```
CP,CV,$1,$2.LGR=16,NBFIC=63
CP,DF,$3,$1
CP,DF,$4,$1
CP,DF,$5,$1
CP,DF,$7,$1
CP,DF,$9,$1
/,$1,LIS=LO
```

Diese Datei würde durch folgendes Kommando ausgeführt:

ASG,CI,AUTODUP,FL1,VOL1, FILE1,FILE2,FILE3,FILE4,FILE5,(cr)

Die Wirkung dieses Kommandos wäre die gleiche wie bei unserem vorangehenden Beispiel.

2.10.1.3 Erläuterungen

1. Ein impliziter Parameter gilt als Leerstelle, wenn im ASG-Kommando kein entsprechender Parameter gestellt wird.

   Beispiel: ASG,CI,Dateiname,PAR1,,PAR3,PAR4(cr)

   $1 = PAR 1
   $2 = Leerstelle, hier **müssen** zwei Kommata eingegeben werden, um Position X2 anzuzeigen.
   $3 = PAR3
   $4 = PAR4

2. Die Wirkung des ASG-Kommandos wird beendet und die Tastatur wird erneut als Eingabevorrichtung angewiesen, wenn folgende Bedingungen erfüllt sind:

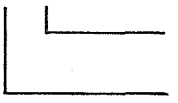
   a. Das Datei-Ende-Zeichen in der Befehlsdatei ist erreicht (normales Ende).

   b. Ein Systemfehler wird festgestellt. In den meisten Fällen wird der laufende Vorgang abgebrochen und die verbleibenden Zeilen in der Kommandodatei werden nicht berücksichtigt.

3. In der Parameterliste können bis zu 9 Parameter angegeben werden.

4. Wenn ein Dollarzeichen als Teil eines Namens in einer Kommandodatei geschrieben werden soll, muß es zweimal eingegeben werden.

   Beispiel: FLO.$$ $3

   Impliziter Parameter

   Kennzeichnet $ als Teil des Dateinamens

   Wenn das ASG-Kommando mit Parameter 3 = TEST ausgeführt wird, ergibt sich die Funktion FLO.$TEST.

5. ASSIGN-Kommandos können selbst verkettet werden, d.h. die letzte Eingabe in der Kommandodatei kann ein neues ASG-Kommando sein. ASG-Kommandos dürfen jedoch nicht geschachtelt werden.

2.11 FM - FORMATIEREN EINES VOLUMES

Das FM-Dienstprogramm ist die Routine, die eine Diskette formatiert (initialisiert), durch Schreiben der von PROLOGUE geforderten Bit-Muster. Dies muß vor der ersten Benutzung der Medien erfolgen.

Erneutes Formatieren kann erforderlich werden, wenn das Format durch Fehler gestört wurde.

Syntax

[Einheit.] FM,FLX(cr)

Hier bedeuten:

Einheit. FM	- Spezifiert das FM-Programm und in Option die Einheit, in der es sich befindet. Vorgegeben ist die Systemeinheit.
FLX	- Spezifiziert das Laufwerk, auf dem sich die zu formatierende Diskette befindet, mit einer Ziffer von Ø bis 3 für X.

Wenn dieses Kommando eingegeben ist, wird FORMAT (Y/N) angezeigt. Sie müssen durch Eingabe von Y (Ja) bestätigen. Dann wird das System 15 bis 20 Sekunden benötigen, um die Diskette zu formatieren. Wenn der Vorgang richtig ausgeführt ist, erscheint "-" und es erfolgt eine Rückkehr zu PROLOGUE.

KAPITAL 3. BOSS BASIC UNTER PROLOGUE

3.1 Allgemeines

BASIC ist eine höhere Sprache, die unter PROLOGUE-Kontolle abläuft. BASIC ist ein Interpreter -- eine interaktive Sprache. Das Programm kann unmittelbar nach der Eingabe ausgeführt werden, Zwischenschritte sind nicht notwendig.
BASIC beinhaltet Programme zum Anlegen, Speichern und Verarbeiten sequentieller Dateien. Ferner können Sie die sequentiellen und index-sequentiellen Zugriffsmethoden des File Management System verwenden, über den Aufruf der entsprechenden PROLOGUE-Assemblerroutinen.

Dieses Kapitel gibt Aufschluß über das Laden von BASIC und die Benutzung von BASIC-Kommandos, die über die Tastatur eingegeben werden. Die BASIC-Anweisungen und Funktionen sind im Einzelnen im BOSS BASIC-Handbuch, Veröffentlichung Nr.B-1005 D erläutert.

3.2 BENUTZUNG DES BASIC INTERPRETERS UNTER PROLOGUE

Wie oben erläutert, ist das Programm namens BASIC ein interaktiver Interpreter der Ihre Programmzeilen analysiert, wie sie eingegeben werden und auftretende Programm-Fehler anzeigt. Diese Fehler müssen dann sofort berichtigt werden. Ein Programm kann direkt eingegeben werden und die Anweisungen werden dann sofort ausgeführt (wie mit dem Rechner); oder indirekt, wobei das Programm im Speicher aufbewahrt und erst ausgeführt wird, wenn alle Zeilen geschrieben sind. Ein indirekt eingegebenes Programm kann für späteren Gebrauch auf der Diskette gespeichert werden.

3.2.1 Aufruf von BASIC

Nachdem PROLOGUE geladen wurde, wird BASIC wie folgt aufgerufen:

[Einheit.] BASIC [,Progname][,Optionen] (cr)

Hier bedeuten:

Einheit BASIC	- Spezifiziert den BASIC Interpreter.
Progname	- Optionales, auf der Diskette gespeichertes Anwenderprogramm. Der Progname muß im Standardformat [Einheit.] Name [-Typ] [Schlüssel] formuliert sein.

Optionen:

S=XXX	- Spezifiziert die Datensatzlänge für durch BASIC angelegte Dateien. Vorgabe-Wert ist 256 Bytes. Die Wahl von 256 Bytes beansprucht einen vollen Sektor der Diskette.

M:Addr – Spezifiziert die höchste von BASIC benutzbare Speicheradresse, im Dezimal- oder im BASIC-Hexadezimalformat &HXXXX. Vorgabe-Wert ist die von PROLOGUE spezifizierte höchste Anwender-Speicher-Adresse.

F:numbfiles – Spezifiziert die maximale Anzahl der gleichzeitig geöffneten Dateien. Vorgabe-Wert ist 3, Höchstwert ist 15.

Die Optionen können in beliebiger Reihenfolge spezifiziert werden, sie sind durch Kommata zu trennen. Wenn eine oder mehrere Optionen spezifiziert sind, aber kein Progname, müssen Kommata die Position von Progname in der Kommandozeile anzeigen. Beispiel: BASIC,,S:256(cr).

Wenn das Kommando durchgeführt wird, wird BASIC geladen und folgende Meldung angezeigt:

```
BASIC REV.5.2
R2E BASIC
Copyright 1977, 78, 79, 80 (C) by Microsoft
Created DY-MO-YR
XXXXX Bytes Free
OK
```

XXXXX Bytes Free gibt den für ein BASIC-Anwenderprogramm freien Speicherplatz an.

Das OK-Symbol zeigt an, daß BASIC in der Kommandoebene ist und Kommandos oder Programme eingegeben werden können. Wenn Progname spezifiziert wurde, wird dieses Programm in den Speicher geladen und ausführt.

Beispiele:

BASIC(cr) – Lädt den BASIC Interpreter. Spezifiziert implizit, daß drei Dateien gleichzeitig geöffnet sein können.

BASIC,TREK-S – Wie oben, nur wird das Programm TREK-S eingelesen und ausgeführt.

BASIC,TREK-S,F:5 – Wie oben, nur wird hier erklärt, daß bis zu 5 Dateien gleichzeitig geöffnet sein können.

3.2.2 Betriebsarten

BASIC kann in zwei Betriebsarten eingesetzt werden:

Direkt – Anweisungen (Kommandos und Befehle) werden ohne vorherige Zeilennummern eingegeben, sie werden sofort ausgeführt und nach der einwandfreien Ausführung wird OK angezeigt. Die Ergebnisse arithmetischer und logischer Schritte werden sofort angezeigt und für späteren Gebrauch gespeichert, aber die Anweisungen selbst sind nach der Ausführung verloren.
Diese Betriebsart eignet sich zum Austesten und für rasche Rechenvorgänge, die kein komplettes Programm erfordern.

Indirekt – Diese Betriebsart wird für die Programmeingabe verwendet. Jeder Programmzeile ist eine Zeilennummer

vorangestellt,und das Programm wird gespeichert. Das gespeicherte Programm wird durch Eingabe des RUN-Kommandos ausgeführt. Das Programm kann durch die Kommandos SAVE, LOAD und RUN auf der Diskette gespeichert und wieder geladen werden. Für eine Beschreibung dieser Kommandos siehe Abschnitt 3.2.4.

3.2.3 Zeilennummern

Zeilennummern geben die Reihenfolge an, in der die Zeilen gespeichert und ausgeführt werden, und werden auch als Referenzen für Verzweigungen und zum Bearbeiten benutzt. Zeilennummern müssen im Bereich von 0 bis 65529 liegen. Die Zeilen müssen nicht in numerischer Reihenfolge eingegeben werden, aber sie werden in richtiger numerischer Reihenfolge im Programm erscheinen. Eine Zeile kann durch Neueingabe der Nummer und durch Eingabe der neuen Informationen oder durch Verwendung des BASIC-EDIT-Kommandos geändert werden. (Näheres hierzu finden Sie im BASIC-Handbuch).

Es empfiehlt sich, Zeilen in Zehnerabständen zu numerieren, so daß zum Einfügen zusätzlicher Zeilen Platz bleibt.

3.2.4. BASIC Programm Datei-Kommandos

BASIC arbeitet auf einer Kommandoebene und einer Programmebene. In der Kommandoebene werden die Kommandos von BASIC von der Tastatur erwartet. Die Struktur dieser Kommandos ist nachstehend beschrieben. Weil die Kommandos mit dem PROLOGUE-Betriebssystem kompatibel sind, unterscheiden sich die dateibestimmenden Kommandos in der Syntax geringfügig von anderen BASICS.

Die verwendeten Dateinamen müssen das Standardformat von PROLOGUE haben: [Einheit.] Name [-Typ] [:Schlüssel].

Wenn der Typ nicht angegeben ist, wird Typ -S angenommen.

3.2.4.1 Kommando RUN

Syntax, Format 1:

RUN [Zeilennummer]

Dieser Befehl wird nach Eingabe oder Laden eines BASIC Programms eingegeben. Die Ausführung des Programms beginnt mit der niedrigsten Zeilennummer oder mit der im RUN-Kommando genannten Zeilennummer. Wenn RUN ausgeführt ist, kehrt BASIC auf die Kommandoebene zurück.

Syntax, Format 2:

RUN "Filenname" [,R]

Lädt das spezifizierte Programm von der Diskette und führt es aus. Vor dem Laden werden durch RUN alle Dateien geschlossen und der Speicherinhalt gelöscht. Jedoch bleiben die Dateien offen, wenn die Option R eingesetzt wird.

Bespiel: RUN "FLØ.PROG",R – Liest PROG von Floppyeinheit Ø ein, nachdem der Speicherinhalt gelöscht wurde.
Alle Dateien bleiben geöffnet, das Programm PROG wird ausgeführt.

3.2.4.2 Kommando SAVE

Syntax:
SAVE "Dateiname" [,A] oder SAVE "Dateiname" [,P]

Diese Anweisung schreibt das im Speicher befindliche BASIC-Programm auf die spezifizierte Datei der Diskette. Wenn keine Einheit als Teil des Dateinamens angegeben wird, wird die Benutzereinheit vorgegeben. Wenn der Typ nicht angegeben wird, wird Typ-S (Quelle) angenommen. Wenn bereits eine Datei mit dem spezifizierten Dateinamen vorhanden ist, wird sie überschrieben.

Wenn die Option A spezifiziert ist, wird die Datei in ASCII-Format gespeichert und kann von PROLOGUE mit der Edit (ED)-Routine bearbeitet werden. Wenn A nicht spezifiziert ist, wird die Datei im verdichteten Binärformat gespeichert, das weniger Platz benötigt, aber es ist für den PROLOGUE-Editor unverständlich.
Achtung: Wenn eine Datei von einem BASIC MERGE-Kommando geladen werden soll, muß sie in ASCII-Format vorliegen. Es ist vorteilhaft, für ASCII BASIC Dateien den Typ-S zu verwenden, und für komprimierte BASIC Dateien den Typ-C.

Das Kommando LOAD kann beide Dateiarten verarbeiten.

Verwenden Sie die Option P zum Schutz der Datei duch Speichern in einem kodierten binären Format. Wenn später eine geschützte Datei mit dem Kommando RUN (oder LOAD) geladen wird, wird jeder Versuch einer Auflistung oder Bearbeitung von BASIC abgelehnt.

Bespiel: SAVE"FL1.NEW-S",A - Überträgt das BASIC-Programm aus dem Speicher im ASCII-Format auf die Diskette in Einheit 1, als Datei mit dem Namen NEW-S.

3.2.4.3 Kommando LOAD

Syntax:
LOAD "Dateiname" [,R]

DieseAnweisung lädt das spezifizierte Programm von der Diskette in den Speicher. Wenn der Typ nicht spezifiziert ist, wird vom Programm eine Datei vom Typ-S gesucht.

LOAD schließt alle offenen Dateien und löscht das laufende Programm im Speicher bevor das spezifizierte Programm geladen wird. Wenn das Programm geladen ist, kehrt BASIC in die Kommandoebene zurück. Wenn die Option R spezifiziert ist, wird das Programm gestartet, nachdem es geladen wurde, und alle offenen Dateien bleiben geöffnet.

3.2.4.4. Kommando MERGE

Syntax:
MERGE "Dateiname"

Diese Anweisung mischt das spezifizierte Programm von einer Diskette mit dem im Speicher befindlichen Programm ein. Sofern das neue Programm Zeilen mit der gleichen Zeilennummer wie das im Speicher befindliche aufweist, werden die entsprechenden Zeilen im Speicher durch die neuen ersetzt.

Das spezifizierte Programm muß vorher in ASCII-Format gespeichert worden sein, sonst kommt zu einem "Bad File Mode"-Fehler. Wenn der Dateityp nicht spezifiziert ist, wird Typ-S angenommen.

3.2.5 Kommando KILL

Syntax:

KILL "Dateiname"

Kill löscht die spezifizierte Datei auf der Diskette/Platte. Wenn versucht wird, eine offene Datei zu löschen, tritt ein "File Already Open"-Fehler auf. Kill kann jede beliebige Datei löschen, ob sie unter BASIC angelegt wurde oder nicht.

3.2.4.6 Kommando NAME

Syntax:

NAME Altdateiname AS Neudateiname

Diese Anweisung benennt die alte Datei um (anstelle des Altdateinamens tritt der Neudateiname). Altdateiname muß auf der spezifizierten Einheit vorhanden sein, Neudateiname darf nicht vorhanden sein; andernfalls kommt es zu einer Fehleranzeige. Diese Anweisung ändert lediglich den Namen der Datei. Ihr Inhalt bleibt unverändert.

3.2.4.7 Kommando FILES

Syntax:

FILES["Einheit.XXX"]

Diese Anweisung arbeitet wie folgt:

FILES (cr)	- Zeigt den Inhalt der Benutzereinheit an.
FILES "Einheit"	- Zeigt den Inhalt der spezifizierten Einheit an.
FILES "Einheit.XX"	- Zeigt die Dateien auf der spezifizierten Einheit an, die die Bedingungen XX erfüllen. XX kann aus einem oder mehreren Buchstaben des Dateinamens bestehen oder ein Dateityp sein.

Beispiele:

FILES"FL1.S"	- Zeigt alle Dateien auf Einheit 1 an die mit dem Buchstaben S beginnen.
FILES"FL1.-B"	- Zeigt alle Dateien auf Einheit 1 mit Suffix B an.

3.2.4.8 Sonstige Kommandos

NEW löscht das im Speicher vorhandene Programm und alle Variablen.

LIST[line no][- line no] -- diese Anweisung listet das im Speicher befindliche

Programm von der optionalen Startnummer (oder der niedrigsten Zeilennummer) an bis zur optionalen Endnummer (oder der höchsten Zeilennummer).

LLIST - Wie List, jedoch Ausgabe des Programmes auf dem Drucker.

SYSTEM - Die Eingabe dieses Kommandos bewirkt eine Rückkehr zu PROLOGUE.

3.2.5. Befehle, die Dateien betreffen

Verschiedene BASIC-Befehle betreffen die Dateiverwaltung. Dazu gehören OPEN, PRINT USING , INPUT, LINE INPUT , WRITE , CLOSE, FIELD, LST/RSET, EOF, LOC. Wenn diese Befehle Dateinamen spezifizieren, müssen diese Dateinamen Standard-PROLOGUE-Format aufweisen wie:

[Einheit.] Name [-Typ] [:Schlüssel]

Siehe **BOSS BASIC Handbuch**, Veröffentlichung B-1005D, wo Sie eine komplette Beschreibung dieser Anweisungen finden.

KAPITEL 4. BAL UNTER PROLOGUE

4.1 ALLGEMEINES

BAL ist eine kaufmännisch orientierte BASIC-Sprache -- eine leistungsfähige Übermenge von ANSI BASIC. Es bietet verbesserte Befehle für Ein- und Ausgabe kaufmännisch orientierter Information und umfaßt Befehle für das optionale Dateiverwaltungsystem (Dateiverwaltung mit freiem Zugriff, sequentiell und indexsequentiell).

BAL ist eine Compilersprache, die als Eingabe für ein Übersetzermodul eine BAL-Quelldatei erfordert. Der Übersetzer (TR genannt) übersetzt diese Quelldatei und erzeugt eine Zwischendatei. Diese Datei kann unter Kontrolle des BAL-Executors (EX) geladen und ausgeführt werden.

Dieses Quellprogramm kann einen beliebigen Dateinamen führen und wird im BAL-Format unter Verwendung des PROLOGUE Editors (ED) (in Abschnitt 2.4 beschrieben) geschrieben. Die Quelldatei ist implizit vom Typ-S, wenn der Typ nicht spezifiziert wird.

Komplette Erläuterungen zum BAL-Befehlssatz und Format eines BAL-Programms finden Sie in **BOSS BAL Reference Manual,** Veröffentlichung B-1006.

4.2. BAL-Übersetzer - TR (Translator)

Wenn ein BAL-Programm im geeigneten Format vorbereitet wurde, unter Benutzung des PROLOGUE Editors, muß es von der BAL Translator Routine übersetzt werden. Diese Routine ist auf einer BAL-Diskette als TR-O gespeichert. Das Übersetzungskommando ist wie folgt zu verwenden:

Syntax:

[Einheit.] TR, Dateiname [,Optionen]

Hier bedeuten:

Einheit. TR	- Übersetzungsprogramm
Dateiname	- Eine BAL-Quelldatei. Der Dateiname muß richtiges PROLOGUE-Format aufweisen wie: [Einheit.] Name [-Typ] [:Schlüssel]. Wenn der Typ weggelassen wird, wird Typ-S angenommen.
Optionen	- Verschiedene Kompilierungsoptionen können spezifiziert werden, in beliebiger Reihenfolge, durch Kommata getrennt . Dies sind:

Optionen	Definition	Vorgabewert
NL	kein Listing	Listing auf Bildschirm, wenn keine Listoption angegeben ist.
LIS = LO	Liste auf Zeilendrucker	
ND	Debugadressen werden im Programmlisting nicht ausgegeben.	Debugadressen werden geliefert.
TP	Teilübersetzung	Vollständige Übersetzung
DEST = Dateiname	Spezifiziert den Namen der sich ergebenden Zwischencodedatei (Typ-T ist Vorgabe-Wert)	Zwischendateien erhalten den gleichen Namen wie die Quelldatei mit Typ-T.

Dieses Kommando führt zur Übersetzung der spezifizierten Quelldatei mit den angegebenen Optionen. Beachten Sie:

1. Wenn ND weggelassen wird, wird das Listing mit den Debugadressen erstellt, die berechnet und links von jeder Programmzeile gelistet werden. Dies sind die Maschinenadressen des Kodes für diese Anweisung. Diese Adressen sind notwendig für die Verwendung der Debug-Option für den Test des Programmes. Näheres hierzu in **BOSS BAL Reference Manual,** B-1006.

2. Wenn TP, Teilübersetzung, spezifiziert ist, können Sie bestimmte Teile Ihres Programmes für die Übersetzung auswählen. Das System druckt : **...SEGMENT NUMBER:**. Sie müssen nun eine Segmentnummer und (cr) eingeben, darauf wird die nächste Segmentnummer angefordert. Wenn Sie statt einer Nummer (cr) eingeben, werden die ausgewählten Segmente übersetzt.

   Diese Option ist nützlich, wenn Teile Ihres Programmes einwandfrei arbeiten und wenn Sie nur ein oder mehrere Segmente umwandeln wollen, in denen Fehler aufgetreten sind. Achtung: wenn Segment Ø geändert wird, muß das ganze Programm umgewandelt werden.

3. Wenn DEST = Filename eingesetzt ist, können Sie Ihrer Zwischencodedatei einen beliebigen gültigen Namen zuweisen.

Wenn bei der Übersetzung Programmfehler entdeckt werden, werden sie folgendermaßen behandelt:

1. Wenn das Listing auf dem Bildschirm angezeigt wird, wird eine Fehlermeldung angezeigt und ein Stop ausgegeben, so daß Sie Zeit haben, den Fehler festzustellen. Das falsche Zeichen wird zwischen Klammern gesetzt und eine Fehlermeldung erscheint (siehe BAL Reference Manual).

   Nach Drücken von ESC wird die Übersetzung fortgesetzt.

2. Wenn die Liste ausgedruckt wird, wird eine Fehlermeldung unter die falschen Befehle gedruckt und die Ausgabe fortgesetzt.

Beispiel: DCL A, B, CD
(D) DECLARATION ERROR 61 DBUG ADDRESS 0007.

Wenn ein Übersetzungsfehler schwerwiegend ist, wird die Programmlänge Ø am Ende der Übersetzung für das Segment erscheinen, in dem der Fehler aufgetreten ist.

Beispiele:

1. TR,FLØ.NEW,LIS = LO — Das Übersetzungsprogramm wird von der Systemeinheit geladen. BAL-Quelldatei NEW-S (S als Vorgabe angenommen) wird von Einheit Ø einlesen und übersetzt. Die Zwischencodedatei wird auf der gleichen Einheit generiert und automatisch NEW-T genannt; die Liste wird auf dem Drucker ausgegeben.

2. FL1.TR,FLØ.DEMO,ND DEST = DEMO2-T — Das Programm TR wird von FL1 geladen und übersetzt die Datei DEMO-S von Einheit Ø. Listing ohne Debugadressen auf Bildschirm. Die Zwischencodedatei heißt DEMO2-T.

4.3 BAL-EXECUTOR - EX

Der BAL-Executor (auf der Diskette als EX-O katalogisiert) ist das Programm für die Ausführung übersetzter BAL-Programme. Das Ausführungs-Kommando ist wie folgt zu verwenden:

Syntax:
[Einheit.] EX, IntDateiname [,DB]

Hier bedeuten:

Einheit. EX	- BAL-Executor
IntDateiname	- BAL-Zwischencodedatei, erzeugt durch die TR-Routine. Hierfür kommt jeder gültige Dateiname in Frage. Typ-T wird vorgegeben.
DB	- Ausführung des Programms unter Kontrolle des Debuggers. Eine komplette Beschreibung der Benutzung des Debuggers befindet sich im BAL Reference Manual.

Wenn dieses Kommando ausgeführt wird, wird das BAL-Ausführungsprogramm in den Speicher geladen (12 K Bytes sind erforderlich), dann wird das spezifizierte BAL-Programm geladen und ausgeführt. Wenn Ausführungsfehler auftreten, erscheint eine Fehlermeldung (siehe Fehlermeldungsverzeichnis im BOSS BAL Reference Manual), das BAL-Programm wird abgebrochen und das PROLOGUE-Symbol kehrt zurück.

Eine Fehlermeldung hat folgendes Format:

ERROR N IN SEGMENT X AT ADDRESS YY.

Hier bedeuten:

N Die Fehlernummer
X das Programmsegment
YY die Debugadresse innerhalb des Segmentes

KAPITEL 5. DATEISYSTEMAUFBAU

5.1 ALLGEMEINES

Dieses Kapitel gibt dem Anwender allgemeine Information zu den Merkmalen von PROLOGUE. Besonders nützlich ist der Abschnitt 5.4, der verschiedene Systemparameter beschreibt. So können Sie zum Beispiel durch Prüfen des Inhaltes der Speicherstellen 99H und 9AH Ladeadressen eines jeden von PROLOGUE geladenen Programmes ermitteln.

Für komplette Einzelheiten zum Interface mit PROLOGUE einschließlich Angaben zum Format der Dateien und den Assemblerroutinen, aus denen PROLOGUE aufgebaut ist, siehe **PROLOGUE Operating System and File Management System Programmer's Guide,** Veröffentlichung Nr.B-1004.

5.2 KERN DES DATEIVERWALTUNGSSYSTEMS

Der Kern oder das Herz des Dateiverwaltungssystems ist ein Satz von Grundfunktionen, die die Dateien auf den Systemeinheiten verwalten. Diese Funktionen liefern den Beginn einer dynamischen Dateistruktur und steuern die Zuweisung des auf der Platte oder Diskette verfügbaren Platzes, und das Lesen und/oder Schreiben von Dateien.

5.3 ORGANISATION DER DISKETTE

Jede Platte und jede Diskette besteht aus einer Reihe konzentrischer Spuren, von denen jede in Sektoren zu je 256 Zeichen unterteilt ist, wie aus Abb. 5-1 ersichtlich.

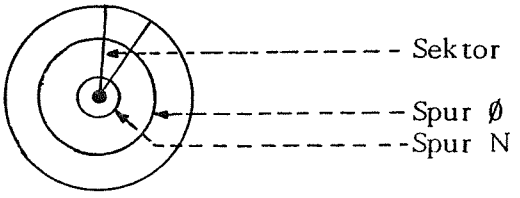


Abb.5-1 Disketten-/Plattenaufbau aus Spuren und Sektoren

Logisch gesehen ist die Diskette eine Folge aufeinanderfolgender Sektoren von Spur Ø, Sektor Ø bis Spur Ø, Sektor 16, und weiter zu Spur 1, Sektor Ø usw. bis Spur n, Sektor 16.

5.3.1 DAS GRANULE - die logische Speichereinheit

Die kleinste **logische** Speichereinheit im PROLOGUE-System ist das **Granule**, das aus 2 bis 256 Sektoren besteht. die Anzahl der Sektoren pro Granule wird für jedes Volume bei der Anlage des Volumes durch den Programmierer bestimmt.

Jedes Programm oder jede Datei belegt ein oder mehrere Granules. Auch wenn ein Programm nur wenige Bytes lang ist, muß ihm ein **komplettes** Granule zugewiesen werden. So kann der Programmierer die logischen Einheiten des Speichers in einer Größe wählen, die seinem Bedarf möglichst weit entgegenkommt. Durch Festlegung kleiner Granules wird der für eine Reihe kurzer Programme erforderliche Platz optimiert. Dagegen benötigt das System mehr Zeit, um eine bestimmte Datei zu suchen und zu laden. Die Anlage von Dateien mit großen Granules hilft bei der Auffindung von Dateien Zeit sparen, kann jedoch zu mehr Platzverschwendung führen.

5.3.2. Tabelle der Disketten-/Plattenspeicherzuweisung

Der Platz auf jeder Diskette wird durch eine Tabelle in Spur Ø, Sektor Ø zugewiesen, in der jedes Bit von 254 8-Bit-Zeichen ein Granule im Volume darstellt. Diese Tabelle, die gleichzeitig mit der Beschreibung einer jeden Datei generiert wird, ermöglich es dem Dateisystem, die Entwicklung einer jeden Datei des Volumes zu verfolgen und zu kontrollieren. Einzelheiten hierzu finden sich im
PROLOGUE Operating System & File Management System Programmer's Guide.

5.3.3 Wichtige Betrachtungen zum Platzbedarf

1. Wenn die Sektorenzahl pro Granule bei der Anlage der Diskette nicht spezifiziert wird, wird 16 als Vorgabe angenommen.

2. Es ist möglich, Dateien von einer Diskette/Platte mit einer Granulegröße auf Disketten/Platten einer anderen Granulegröße zu übertragen. Dies kann zu Platzverschwendung bei der Benutzung des Kopierpro - grammes (CP) führen.

Beispiel 1:

Kopie von		Kopie auf
Volume mit 4 Sektoren pro Granule	→	Volume mit 16 Sektoren pro Granule
Ein kleines Programm, das nur einen Granule und nur einen Sektor dieses Granules verwendet = 3 verschwendete Sektoren.		Das gleiche Programm, das nur einen Granule (und nur einen Sektor) verwendet = 15 verschwendete Sektoren

Beispiel 2 :

Kopie von		Kopie auf
Volume mit 16 Sektoren pro Granule	→	Volume mit 4 Sektoren pro Granule

Ein kleines Ein-Sektor-Programm, das ein Granule verwendet = 15 leere Sektoren

Das gleiche kleine Programm, das nur **vier** Granules mit je vier Sektoren verwendet = 15 leere Sektoren.

Hier wird kein zusätzlicher Platz benötigt, aber auch kein Platz eingespart, da die Routine CP die gleiche Sektorenzahl kopiert.

Wenn Sie ein Quellprogramm kopieren möchten, können Sie in einer solchen Situation mit dem Editor Platz sparen, da der Editor des Zeichen Ende der Datei, 1AH, erkennt (nicht aber CP), und eine Quelldatei in die kleinst - mögliche Granulezahl kopiert.

5.4 SYSTEMPARAMETER

Einer bestimmten Anzahl von Parametern werden festen Speicheradressen zugewiesen und beim Konfigurieren des Systems festgelegt. Diese Speicheradressen, Parameter (und in bestimmten Fällen ihr üblicher Wert) werden nachstehend definiert.

Adresse	**Parameter**	**Normaler Wert**
80H 81H	Höchste für den Anwender verfügbare Speicheradresse (invertierte Schreibweise)	Systemabhängig
82H	Zeilenzahl auf dem Bildschirm	"
83H	Zeichenzahl pro Bildschirmzeile	"
84H	Zeilenzahl pro Druckseite	"
85H	Zeichenzahl pro Druckzeile	"

Tastatur und Anzeigezeichenzuordnung

			ASCII-Zeichen
86H	Escape	Umschaltzeichen	ESC 18H
87H	Right Arrow	Rechtspfeil	→ 12H
88H	Left Arrow	Linkspfeil	← 11H
89H	Up Arrow	Aufwärtspfeil	↑ 13H
8AH	Down Arrow	Abwärtspfeil	Zeilenvorschub ↓ ØAH
8BH	CLEAR (screen)	Löschen (Bildschirm)	ØCH
8CH	HOME (Reset)	Rückstellung	↖ 14H

Adresse		Parameter	Normaler Wert
8DH	TAB	Tabulator	() 09H
8EH	CANCEL LINE	Zeilenlöschung	CTRL-C 03H

Definitionen zur Systemeinheit

95H	Systemeinheit Laufwerk (zum Laden von PROLOGUE)	10 = Floppy
96H	Nummer der Einheit	0 - Einheit 0 1 - Einheit 1
97H	Benutzereinheit (zum Laden von Benutzerprogrammen)	10 - Floppy
98H	Nummer der Einheit	0 = Einheit 0

Ladeadresse

99H 9AH	Vorgegebene Ladeadresse für verschiebbare Objektprogramme, die vom System verwendet werden (invertierte Schreibweise)	Systemabhängig

Optionen

A8H	Sprachoption für Meldungen	0 = Französisch 1 = Englisch
A9H	Option Dezimalpunkt	2C = Komma 2E = Punkt

Peripheriezuweisungstabelle

Für die Beschreibung eines jeden Peripheriegerätes sind 6 Byte notwendig.

2 Bytes	- Mnemonischer Name für den Gerätetyp
1 Byte	- Anzahl der Einheiten
1 Byte	- Betriebsmittelnummer
2 Bytes	- Abrufadresse (invertierte Schreibweise)

CØH	- Beginn der Tabelle, erstes Peripheriegerät
C6H	- Zweites Peripheriegerät
usw.	

Tabelle mnemonischer Namen

FL	5 1/4 Floppy Laufwerk
LO	Drucker
CO	Bildschirmgerät
CI	Tastatur
MD	5 1/4" Festplatte

ANHANG A. PROLOGUE-FEHLERCODES

A.1 ALLGEMEINES

Immer wenn in einer der Routinen unter PROLOGUE ein Fehler auftritt, wird eine Fehlermeldung ausgegeben, die einen Fehlercode enthält. Die Fehlerkodes von PROLOGUE sind in diesem Anhang nach Peripheriefehlern, Interpreter-Befehlen- und Dateiverwaltungsfehlern geordnet zusammengestellt.

Zusätzlich zu den Systemfehlern von PROLOGUE können einzelne Fehler bei der Arbeit mit BASIC und BAL auftreten. Für Einzelheiten zur Fehlererfassung und Anzeige zu diesen Programmiersprachen siehe die zugehörigen Handbücher.

In den meisten Fällen wird bei Auftreten eines Fehlers der laufende Vorgang abgebrochen und das PROLOGUE - Symbol kehrt zurück.

Für jeden nachstehend aufgeführten Fehler finden Sie eine Fehlernummer, eine Bescheibung, und eine mögliche ursachenbedingte Abhilfemaßnahme. Wenn die Abhilfe offensichtlich ist, wie Prüfen der aufgelisteten Ursachen und Korrektur wird sie nicht angeführt.

A.2 PERIPHERIEFEHLERKODES

Folgende Fehlerkodes werden auf dem Bildschirm angezeigt, wenn das System einen Fehler im Zusammenhang mit einer Peripherieeinheit festgestellt hat. Diese Fehler sind von der verwendeten Sprache unabhängig und betreffen Hardware und Driver-Module.

Fehlerkode (dezimal)	Beschreibung	Abhilfevorschlag
01	Einheit ist nicht bereit	1. Laufwerkstür nicht richtig geschlossen 2. Floppy -Disk falsch eingelegt (seitenverkehrt) 3. Einheit nicht angeschlossen 4. Signalkabel nicht angeschlossen 5. Laufwerksriemen gerissen oder abgerutscht 6. Versuch, eine einseitige Diskette in einem Doppellaufwerk zu benutzen
02	Schreib/Lesefehler	Gewöhnlich tritt ein Lese- nach Schreibfehler auf, wenn das CRC-Zeichen (zyklischen Blockprüfung)

Fehlerkode (dezimal)	Beschreibung	Mögliche Ursache/Abhilfe
		nicht stimmt. Das System hat den Vorgang erneut versucht, und 10 mal versagt. Kann ein Problem mit dem Datenträger sein. 1. Versuchen Sie die Backupdiskette. 2. Lesen Sie sofort das Statusprogramm ein (siehe 2.7), um Spur und Sektor festzustellen, in denen das Problem auftritt. Übertragen Sie anhand des CPS-Programmes (2.6) **nur** diesen Sektor neu von einer Backupdiskette. Wenn das Problem so nicht gelöst wird, kann ein dauernder unbehebbarer Datenträgerfehler vorliegen. Lassen Sie dieses Programm unverändert, unbenutzt und ungelöscht.
03	Fehler bei der Spurpositionierung	Tritt meist auf, wenn das Laufwerk mechanisch leicht abgenutzt ist oder leicht dejustiert ist. Wenn dieser Fehler häufiger auftritt, müssen die Laufwerke neu eingestellt werden.
04	1.adressierter Sektor außerhalb des Laufwerkbereichs	Mit einem Dienstprogrammkommando haben Sie einen Vorgang mit einer auf der ausgewählten Einheit nicht verfügbaren Sektornummer gewählt, z.B. Sektor 650 auf einer einseitigen Floppy-Disk, die nur 559 Sektoren enthält. Dies kann auch vorkommen, wenn von einer doppelseitigen auf eine einseitige Diskette kopiert wird.
	2. Parameteraufruffehler	Die Parameter wurden falsch in den Kontrollblock gegeben, wenn das Programm PROLOGUE-Assemblerroutinen benutzt. Siehe PROLOGUE-Operating System Programmer's Guide.
05	Ausgewählter Träger steht unter Schreibschutz	Versuch, auf einer Floppy-Disk mit Schreibschutz zu schreiben. Schutz lösen oder ungeschützten Träger verwenden.

Hinweis: Folgende Fehler, Nr.06 bis 12, sind Floppy-Disk-Controller-Fehler (FDC). Wenn sie auftreten, RESET-Knopf

Fehlerkode (dezimal)	Beschreibung	Mögliche Ursache/Abhilfe
		an der Rückseite des Gerätes drücken, um den Computer neu einzuschalten, dann das Betriebssystem neu laden. Wenn die Fehler systematisch auftreten, wenden Sie sich an den Kundendienst.
06	Floppy-Disk-Kontrollfehler	Reset drücken und neu laden
07	Sektorenadressfehler	FDC versuchte, einen Sektor nach dem letzten Sektor einer Spur anzusteuern
08	FDC Betriebsfehler	FDC wurde vom Hauptsystem während der Datenübertagung länger nicht bedient. Reset und neu laden.
09	Datenadreßmarke nicht erkannt	FDC war nicht fähig, die Datenadreß-Marke zu erkennen. Versuchen Sie die Backupdiskette oder formatieren Sie die Diskette neu.
10	ID-Adreßmarke nicht erkannt	FDC war nicht fähig, die ID Adreßmarke zu erkennen; versuchen Sie die Backupdiskette oder formatieren Sie die Diskette neu.
12	FDC inkohärent	Reset und neu laden.
13	Spur 0 nicht auffindbar	Schwerwiegender Fehler. Reset drücken.
14	CRC-Fehler auf ID	Schwerwiegender Fehler. Reset drücken.

A.3 KOMMANDOINTERPRETER-FEHLERCODES

PROLOGUE erwartet die Kommandos in einem bestimmten Format und analysiert dann die Kommandozeile, um die erforderliche Maßnahme zu bestimmen. Folgende Fehler treten auf, wenn ein Kommando nicht einwandfrei eingegeben wird und nicht verarbeitet werden kann. Als Abhilfe wird empfohlen die geforderte Syntax nachzuprüfen und den Befehl zu berichtigen.

Fehlerkode (dezimal)	Beschreibung	Mögliche Ursache/Abhilfe
30	Syntax der Einheit falsch	1. Name falsch geschrieben- FLØ (Buchstabe O statt Ziffer Ø) 2. Zeichensetzung falsch - unzulässige Leerstellen oder fehlende, oder falsche Kommata.
32	Nummer der Peripherieeinheit falsch	Vielleicht wurde FL3 in einem System

Fehlercode (dezimal)	Beschreibung	Mögliche Ursache/Abhilfe
		spezifiziert, das keine Einheit Nr.3 enthält.
31	Peripherietyp unbekannt	Wahrscheinlich Schreibfehler zum Beispiel statt FLØ wurde FDØ spezifiziert.
33	Falscher Dateityp	Für das Kommando wird ein bestimmter Dateityp gefordert; die spezifizierte Datei ist vom falschen Typ. z.B. nimmt der Linkage-Editor nur Typ-O (Objekt-)Datei an.
34	Syntax der Ladeadresse falsch	1. Der spezifizierten Ladeadresse ist kein Und-Zeichen vorangestellt (&). 2. Die spezifizierte Adresse ist nicht hexadezimal.
35	Externe Referenz in Objekt-datei nicht aufgelöst	Versuch eine Objektdatei zu laden, die nicht aufgelöste Referenzen zu einer anderen Objektdatei besitzt. Siehe Assembler-und Linkage-Ladebe-schreibung in PROLOGUE Operating System Programmer's Guide.
36	Objektdatei falsch aufgezeichnet	Objektdatei nicht in Standardformat, siehe Linkage-Editor oder Assembler-beschreibung. Neu assemblieren oder neu kopieren.
37	Versuch über das Speicherende hinaus einzulesen	Das Programm ist zu groß für den verfügbaren Anwender-Speicherbereich Größe verringern (wenn möglich), im Speicher weiter unten laden.

A.4 DATEIVERWALTUNGSFEHLER

Folgende Fehler können bei der Verwaltung und Verarbeitung von Dateien durch das PROLOGUE-System festgestellt werden.

Fehlerkode (dezimal)	Beschreibung	Mögliche Ursache/Abhilfe
40	Datei nicht gefunden	1. Vielleicht ist die Datei nicht auf der Diskette /Platte. 2. Vielleicht muß der Dateityp an-gegeben werden. So fordert die Dateilöschung eine Typenangabe, nicht aber viele andere Kommandos. 3. Vielleicht gibt das Kommando das falsche Laufwerk an.

Fehlercode (dezimal)	Beschreibung	Mögliche Ursache/Abhilfe
41	Datei besteht bereits	Datei mit spezifiziertem Namen besteht bereits auf der spezifizierten Einheit.
42	Datei geschlossen	Versuch, beim Aufruf von Assemblerroutinen die Datei zu lesen, ohne sie zu öffnen.
43	Datei bereits geöffnet	Wie oben, Versuch, eine bereits geöffnete Datei zu öffnen.
44	Nicht verwendet	
45	Datei enthält zu viele Erweitungsblöcke	Versuch, eine Datei über 18 nicht zusammenhängende Erweiterungsblöcke hinaus auszudehnen. Die Datei auf eine leere Diskette übertragen, um die Anzahl nicht zusammenhängender Granules zu reduzieren.
46	Volume überfüllt	alle Spuren auf einem Volume sind voll. Anzahl oder Größe der Dateien reduzieren oder ein anders Volume benutzen.
47	Zugriffsschlüssel falsch	Versuch, eine geschützte Datei zu ändern, zu kopieren oder zu löschen, ohne das richtige Zugriffskennwort zu benutzen.
48	Dateiende	Anormales Dateiende festgestellt. Bedeutet meist, daß eine Datei von Ø Granules spezifiziert wurde.
49	Inhaltsverzeichnis voll	Die maximale Dateizahl ist im Inhalt verzeichnet (das Volume muß nicht voll sein, Dateien löschen oder übertragen, oder den gesamten Inhalt auf ein Volume mit mehr Speicherraum übertragen. Vorgegeben sind 63 Einträge; bis 255 können angelegt werden. Siehe PROLOGUE Operating System Programmer's Guide, wo Sie Näheres zum Konfigurieren finden.

Fehlercode (dezimal)	Beschreibung	Mögliche Ursache/Abhilfe
50	Logische Nummer falsch	Beim Aufruf von Assemblerroutinen wurden falsche logische Nummern verwendet. Gilt auch für BAL-Dateibefehle. die logischen Nummern 1 bis 15 sind gültig.
51	Tabelle der logischen Nummern voll	Tritt in ähnlicher Situation wie obiger Fehler auf. Verwendete Zahl reduzieren.
52	Nicht verwendet	
53	Diskettenadresse der Datei nicht bekannt	Ein Programm wie eine Dienstprogrammroutine hat eine Spur oder einen Sektor zugewiesen, der nicht im Bereich der spezifizierten Einheit liegt.
54	Das Volume wurde nicht vom PROLOGUE -Dateisystem angelegt	Versuch, ein Volume zu verwenden das nicht vom PROLOGUE-System angelegt wurde. Das kann nicht funktionieren.
55	Unbekannte Dateifunktion	
56	Unbekannte Funktion in Ihrem System	Versuch, eine zulässige PROLOGUE Funktion aufzurufen, die aber in Ihrem System nicht konfiguriert wurde (z.B. ISAM oder ASG).

Bemerkung:
Weitere Fehlercodes werden vom Dateiverwaltungssystem benutzt. Diese werden in den ersten 3 umseitig aufgeführten Programmier-Handbüchern behandelt.

ANHANG B. LITERATURANGABEN

Zusätzliche Angaben zu den in diesem Handbuch beschriebenen Themen finden Sie in folgenden Werken:

Titel	Veröffentlichung Nr.
BOSS PROLOGUE Operating System & File Management System Programmer's Guide	B - 1004
BOSS BASIC Reference Manual	B - 1005
BOSS BAL Reference Manual	B - 1006

ANHANG C. VERZEICHNIS DER PROLOGUE-KOMMANDOS

PROLOGUE LADEN

B : U, S S normal = 80 H

VERKETTUNG

[Einheit.] ASG, CI, Dateiname [,Parameterliste]

BAL COMPILER

Übersetzer.
[Einheit.] TR, Dateiname [,NL oder LIS = LO,ND,TR,DEST = Dateiname]

NL	keine Auflistung
LIS = LO	Auflistung auf Drucker
ND	Ohne Debug
TP	Teilübersetzung
DEST = Dateiname	Bestimmung Dateiname

Ausführung
[Einheit.] EX, Intdateiname [,DB]

DB	Option Debug

BASIC INTERPRETER

Einheit. BASIC[,Progname][,S : XXX, M : Addr, F : Dateizahl]

S : XXX	Datensatzlänge
M : Addr	Höchste BASIC-Speicheradresse
F : Dateizahl	Zahl gleichzeitig geöffneter Dateien

KOPIERDIENSTPROGRAMM (CP)

Volume anlegen [Einheit.] CP, CV, Einheit2, Volname
[,LGR = n, NBFIC = Y]

Einheit2	Einheit des Volumes
LRG = n	Anzahl der Sektoren pro Granule
NBFIC = Y	Anzahl der Dateien

Datei anlegen	[Einheit.] CP, CF, Dateiname
Datei löschen	[Einheit.] CP, SF, Dateiname
Datei kopieren	[Einheit.] CP, DF, Dateiname1 ,Dateiname2 oder [Einheit oder LIS = LO]
Datei umbenennen	[Einheit.] CP, RF Dateiname1, Dateiname2
Volume kopieren	[Einheit.] CP, DV, Vol1, Vol2, RZ

RZ	Dateien auf Zeilvolume löschen

Volume umbenennen [Einheit.] CP, RV, Einheit, NeuVolName

SEKTORENKOPIE-DIENSTPROGRAMM (CPS)

Gesamtkopie [Einheit.] CPS, Einheit-S, Einheit-D

Einheit-S	zu kopierende Einheit
Einheit-D	Einheit auf die kopiert wird

Freie Kopierart [Einheit.] CPS (cr), (cr)

Beantwortet Fragen nach Ursprung, Ziel, Spur und Sektor.

DIENSTPROGRAMM INHALTSVERZEICHNIS (/)

Inhaltsverzeichnis des Volumes [Einheit]/[, Einheit2] [,LIS=LO,GR]

LIS = LO	Ausgabe auf Drucker
GR	Einschließlich Daten zur Anordnung der Datei auf der Diskette

Inhaltsverzeichnis von Dateien [Einheit.] /[,Einheit2.] Dateiname[,LIS = LO]

Dateiname	expliziter oder mehrdeutiger Dateiname
LIS = LO	Ausgabe auf Drucker
GR	Einschließlich Daten zur Anordnung der Datei in der Diskette.

DATUMSDIENSTPROGRAMM

[Einheit.] DATE (cr) oder [Einheit.] DATE, DD, MM, YY, HH, MM

DD/MM/YY/HH/MM	Tag/Monat/Jahr/Stunde/Minute

TEXTEDITOR-DIENSTPROGRAMM

Aufruf [Einheit.] ED, Dateiname1 [CF, DEST = Dateiname2]

CF	Datei anlegen

DEST = Dateiname2	Ziel-Dateiname, wenn von Dateiname1 verschieden
Editor-Kommandos	
Leertaste	Vorrücken zum nächsten Befehl Zeile
A	Vorrücken zum Ende der Datei
I	Text einfügen
R	Zeile löschen und anschließend Text einfügen
J : Dateiname	Quellkode von der Quelldatei einfügen
M	Ändern
C	Ergänzen
Dn	Zeilen löschen
,String	Aufsuchen einer Zeichenfolge im Text
.String	Aufsuchen einer Zeichenfolge am Beginn einer Zeile
*	Wiederholung eines Such-kommandos
D,String	Zeilen im Text löschen bis zur Zeichenfolge
D.String	Zeilen löschen bis zum Auftreten der Zeichenfolge am Beginn einer Zeile
Z	Ende der Datei
E	Ende der Bearbeitung
S	Abbrechen der Bearbeitung
L : nn	Festlegen der Zeilenlänge des Textes

FORMATIERDIENSTPROGRAMM (FM)

[Einheit.] FM, FLX

FLX	Laufwerk, in das die Diskette eingelegt ist

PATCH-DIREKTKORREKTUR-DIENSTPROGRAMM

Dateinname [Einheit.] PATCH, Dateiname [,NSEC = no] spezifizieren

NSEC = no	Sektornummer in der Datei

Spur/Sektor [Einheit.] PATCH [,Einheit. NPIS = Track no] NSEC = Sektor no spezifizieren

Einheit	Spezifizierte Einheit
NPIS = Track no	Start-Spur
NSEC = Sektor no	Start-Sektor

STATUS (Einheit)

[Einheit.] STATUS[,Einheit ,LIS = LO]

Einheit	Zu kontrollierende Einheit
LIS = LO	Ausgabe auf Drucker

ANHANG D. VERZEICHNIS DER VERWENDETEN AUSDRÜCKE

Abbruch (BREAK) - Ein Schlüssel, der zusammen mit einem anderen Schlüssel für den Abbruch des Ausführungsprogrammes und die Rückkehr in die PROLOGUE-Kommandoebene (BREAK-P), die Start-Kommandoebene (BREAK-L) oder die Kontroll-Kommandoebene (BREAK-M) verwendet wird.

Adresse - Die einer jeden Speicherstelle zugewiesene Nummer. Im BOSS, von Ø bis 65535 (oder Ø bis FFFF bei Verwendung von Hexadezimalzahlen).

Alphanumerisch - ein alphanumerisches Zeichen ist ein Buchstabe, eine Ziffer, oder ein Spezialzeichen.

Anschlußnummer (Port) - die physische Verbindungsstelle zwischen dem zentralen Prozessor und einem Peripheriegerät. Jede Anschlußnummer besitzt eine numerische Adresse, die der Prozessor für die Kommunikation verwendet. 8080/Z-80 Typ-Mikroprozessoren können bis 256 Input und 256 Output-Anschlußnummern adressieren.

ASCII - Abkürzung für American Standard Code for Information Interchange. Eine Einheit von Standardkodes, die jeweils einen Wert von Ø bis 127 einem jeden von 128 Zahlen, Buchstaben, Spezialzeichen und Kontrollzeichen zuweisen. Diese Kodeeinheit gestattet bequem Informationstransfer von einem Computer oder Peripheriegerät zu einem anderen.

Assembler - ein Programm, das die Hilfsnamen und Symbole der Assembler-Sprache in die binäre Maschinensprache des Computers überträgt.

Assembler-Sprache - eine Sprache, die abgekürzte englische Kommandos (Kurzwörter und Symbole) verwendet, die mit der Maschinensprache Ähnlichkeiten aufweist, aber leichter merkbar und verwendbar ist. Programmieren in Assembler-Sprache ist rasch und kurz, aber komplizierter als höhere Sprache wie BASIC und BAL.

Aufbereitung (Editing) - Durchführen von Korrekturen oder Veränderungen in einem Programm oder in Dateien.

Aufruf (Call) - gewöhnlich ein Befehl, der eine Subroutine aufruft.

Ausführen (Execute) - Ausführen eines Kommandos oder eines Programms oder eines Programmteils.

BAL - Business Application Language, eine höhere Sprache mit leicht verwendbar formatierten Ein- und Ausgabeanweisungen und einem leistungsfähigen Dateiverwaltungssystem in Option.

BASIC - Abkürzung von "Beginners All-purpose Symbolic Instruction Code". BASIC ist eine äußert populäre höhere Programmsprache, die in den meisten kleineren Computern verwendet wird. Sie wurde 1963 im Dartmouth College von Kemeny und Kurtz erfunden.

Baud - Die Bit-Zahl einer Information, die pro Sekunde vom Computer übertragen oder empfangen wird (meist in Bezug auf einen I/O-Kanal verwendet).

Befehl (Instruction) - Der kleinste Teil eines Programms, den ein Computer ausführen kann, Gewöhnlich handelt es sich um einen Befehl der den Computer anweist, einen Arbeitsgang durchzuführen.

Befehl (Statement) - Eine Anweisung in einer höheren Programmiersprache. Oft definiert als Einzelzeile, die eine einzige Programmanweisung enthält.

Bildschirm (Display) - eine Videovorrichtung für die Ausgabe.

Binär - ein Zweiziffernsystem, in dem Ø und 1 verwendet werden. Jede Stelle in einer binären Zahl steht für eine Potenz von zwei, ein binäres Signal wird ausgedrückt durch Vorhandensein oder Fehlen einer Bedingung wie einem elektrischen Potential oder einem magnetischen Feld.

Bit - eine binäre Ziffer (Ø oder 1). Ein Bit kann als Darstellung einer einfachen Wahl zwischen Ja und Nein, richtig oder falsch, an oder aus usw. verwendet werden.

Block - eine Einheit von Speicherplatz, bestehend aus einem oder mehreren aufeinanderfolgenden Granules (logische Speichereinheiten). Eine Datei wird auf einer Platte (oder einer Diskette) als ein oder mehrere Blöcke gespeichert, die physisch über die Platte (Diskette) verstreut sein können (siehe Granule).

Bug - Ein Fehler in der Software eines Computers.

Byte - Eine Grundmaßeinheit für Computer-Speicher, bestehend aus 8 Bit (Binary digits). Kann also einen Wert von 0 bis 255 (im Hexadezimalkode 0 bis FF) annehmen. Alle BOSS Speicherplätze speichern eine Datenmenge von 1 Byte; die Adressen dieser Plätze werden durch eine Zwei-Byte-Speicheradresse spezifiziert.

Chip - Winzige Silikonscheibe, die für die Herstellung verschiedener Computer-Bausteine verwendet wird.

Compiler - gewöhnlich ein Programm, das vom Benutzer geschaffene höhere Programmsprachen in einen Maschinenkode umwandelt, den der Computer direkt ausführen kann. Der BAL-Compiler erzeugt eine Zwischendatei, die dann ein Laufzeitprogramm verwendet, um die endgültige Übersetzung des Anwenderprogrammes in den ausführbaren Binärkode vorzunehmen.

Concatenation (Verkettung) - Das Verbinden zweier Zeichenfolgen zu einer längeren Zeichenfolge.

Konfiguration - ein Programm, das benutzt wird, um ein PROLOGUE-Betriebssystem zu erzeugen, das auf eine bestimmte Hardware-Konfiguration zugeschnitten ist.

Console - Gewöhnlich die Tastatur

CRT - Abkürzung für "Cathode Ray Tube" (Kathodenstrahlröhre). Damit wird oft der Bildschirm bezeichnet.

Cursor - ein Symbol auf dem Bildschirm, das die Stelle anzeigt, an der das nächste geschriebene Zeichen erscheinen wird.

Cursor-Kontrolle - Möglichkeit, den Cursor in eine beliebige Position auf dem Bildschirm zu verschieben, entweder über die Tastatur oder über die Programmsteuerung.

Datei (File) - ein Programm oder eine Gruppe von zusammengehörigen Datenelementen, gespeichert als eine logische Einheit und identifiziert durch einen Dateinamen. Das PROLOGUE-System verwaltet die Dateien in Blöcken und Granules, die nicht notwendigerweise auf der Diskette unmittelbar aneinanderliegen.

Dateiname (Filename) - die einzige Identifizierung für eine PROLOGUE-Datei bestehend aus einem Namen von 1-7 Zeichen, einer Dateityp-Bezeichnung aus einem Zeichen und einem Zugangsschlüssel von Ø bis 4 Zeichen.

Daten - Information irgendwelcher Art. Oft numerische Information.

Debugging - Auffinden und Berichtigen von Fehlern (bugs).

Digital Computer - rechnet mit diskreten numerischen Informationen (bits) im Gegensatz zu Analog-Computern.

Diskette - Eine flexible Kunststoffplatte mit magnetischer Beschichtung in einem quadratischen Schutzumschlag, gewöhnlich mit 130 oder 200 mm Durchmesser. die Diskette speichert digitale Informationen in 35 oder mehr kreisförmigen Spuren auf ihrer Oberfläche.

Driver - Ein Software-Driver ist eine Reihe von Befehlen, die vom Computer verwendet werden, um Daten für den Transfer von und zu einer bestimmten Peripherie-Einheit zu formatieren. Der Software-Driver trägt den elektrischen und mechanischen Anforderungen einer Einheit Rechnung, um das Datenformat zwischen Einheit und Computer zu standardisieren.

Drucker (Printer) - Eine Computer-Output-Einheit, die Hard-Copy-Daten erzeugt. Ein Zeilendrucker druckt komplette Textzeilen auf einmal. Ein Zeichendrucker druckt jeweils ein Zeichen.

Drucklauf (Run) - Ablauf der Befehlssequenz eines Programmes.

Editor - ein PROLOGUE-Dienstprogramm, das zum Aufbereiten von ASCII-Quellprogrammen verwendet wird.

Einsprungspunkt (Entry point) - Die von einer Maschinensprache-Subroutine verwendete Stelle, die die erste ausführbare Anweisung in dieser Subroutine enthält.

Executor - BAL-Einheit, verwendet für die Ausführung von Zwischenkodes, die vom BAL-Translator erzeugt werden.

Floppy disk - Siehe Diskette.

Formatieren - Im BOSS-System eine Diskette in das richtige Bit-Muster für den Gebrauch mit dem PROLOGUE-Betriebssystem bringen.

Ganzzahl (Integer - Eine ganze Zahl, die positiv oder negativ sein kann, oder Null.

Gedruckte Schaltung (Printed circuit) - Elektrische Verbindung aus Lötzinn zwischen Bauteilen auf einer Epoxyd-Karte.

Granule - Die kleinste logische Speichereinheit eines Einzelvolume (Platte Diskette) des PROLOGUE-Systems, bestehend aus 2 bis 256 Sektoren. Die Anzahl der aufeinanderfolgenden Sektoren pro Granule wird für jedes einzelne Volume bei seiner Anlage spezifiziert.

Graphic - Ein Element von einer bestimmten erkennbaren Form oder Farbe.

Graphics - Ein Hardware/Software-System für die Anzeige graphischer Elemente.

Hard copy - Auf Papier gedruckte Information, im Gegensatz zu dem vorübergehend auf dem Bildschirm angezeigten Bild.

Hard-Sector - Floppy-Disks sind in Teile geteilt, die Sektoren heißen, Hard-Sektor-Disketten sind gelocht. Die Löcher zeigen die verschiedenen Sektoren an. Dieser Diskettentyp ist im BOSS-System nicht verwendbar, da dieses mit dem Soft-Sectoring-System, wo die Position der Daten auf der Diskette durch Software-Berechnungen bestimmt wird, und nicht durch physische Kennzeichnung der Diskette.

Hardware - Die Schaltung und elektronischen Bauteile in einem Computer im Gegensatz zur Software, den Computerprogrammen.

Hexadezimal - Ein Nummernsystem, das mit 16er-Potenzen arbeitet, und sechzehn Zeichen besitzt, 0 bis 9 und A bis F. Die Dezimalzahl 178 wird im Hexadezimalkode B2 geschrieben. Ein 8 Bit-Byte kann im Hexadezimalkode mit zwei Stellen ausgedruckt werden.

Höhere Programmsprache (High-Level-Language) - Eine Computer-Sprache, die nach gewissen Regeln und mit gewissen Einschränkungen in menschlicher Sprache geschriebene Befehle annimmt. Eine höhere Programmsprache muß für die Verwendung durch den Computer in Maschinensprache umgesetzt werden.

Index-sequentielles Dateisystem - eine PROLOGUE-Option, die Dateien anlegt, in denen jede Aufzeichnung durch einen einzigen Schlüssel und Index identifiziert wird. Die Schlüssel werden in alphabetischer Reihenfolge als Zeiger für die zugeordneten Daten verwendet, und sorgen für sehr raschen Zugriff zu großen Dateien.

Initialisieren - 1. Startbedingungen für ein Programm herstellen. 2. Eine Diskette im richtigen Bit-Muster vorbereiten (formatieren), so daß der Computer später Daten darauf speichern kann.

Input - In einer Einheit ankommende Information. Oft wird die gleiche Input-Information aus einer Diskette ausgegeben, Input-Daten werden oft durch eine Input-Einheit, wie z.B. eine Tastatur, generiert.

Input/Putput (I/O) - Bezieht sich meist auf die Software oder Hardware, welche Informationen mit der Außenwelt austauscht.

Interface - die Elektronik, die Format und Fluß von Daten steuert, so daß zwei verschiedene Einheiten (z.B. Computer und Peripheriegeräte) miteinander in Verbindung stehen.

Interpreter - Ein Programm, das dem Computer die direkte Ausführung von in einer höheren Sprache wie BASIC geschriebenen Kommandos und Befehlen ermöglicht.

K - Kurz für Kilo oder Tausend. Für den Computergebrauch bedeutet K meist 1Ø24 (4ØØ im Hexadezimalkode).

Kernprogramme (Nucleus programs) - Die Datenverwaltungsroutinen, die als Standardteil des PROLOGUE-Betriebssystem geliefert werden.

Kode - siehe Programm.

Kommando (Command) - eine Aufforderung an den Computer, die unmittelbar nach Erhalt ausgeführt wird. Unterscheidet sich von "Befehl" (instruction), der sich gewöhnlich auf Programmanschnitte bezieht, die in einer bestimmten Reihenfolge ausgeführt werden.

Kontroll-Zeichen (CRTL) (Control (CRTL) characters) - ASCII-Zeichen oder Befehle, die man durch Niederdrücken der CTRL (Kontroll)- Taste erhält, bei gleichzeitigem Niederdrücken einer anderen Taste auf der Tastatur. Die Kontrollzeichen haben gewöhnlich keine graphische Darstellung, sie werden für die Kontrolle verschiedener Funktionen verwendet.

Linkage Editor - Ein Dienstprogramm das durch Auflösen externer Referenzen zwei oder mehr Objektprogramme miteinander verbindet und ein einziges Objektprogramm daraus macht, das direkt ausführbar sein kann.

Maschinensprache (Machine Language) - Die binäre Sprache des Computers. Befehle in der Maschinensprache sind einfache Byte-Operationsschlüssel, auf die verschiedene Operanden folgen.

Megabyte - Eine Million Bytes.

Merkname (Mnemonic) - Eine Abkürzung (oder ein anderes Symbol), das verwendet wird um Gedächnisstützen für schwieriger zu behaltende Ausdrücke zu bieten. Beispiel: Der Merkname NOP ersetzt einen numerischen Operationsschlüssel, wie ØØ.

Mikrocomputer - Ein auf Mikroprozessor-Grundlage aufgebauter Computer. Geringe Größe, aber große Leistung.

Mikroprozessor - Die zentrale Prozeßeinheit eines Computers, als einfache integrierte Schaltung geschaltet.

Modem - Abkürzung für die Wörter "MOdulator-DEModulator". Ermöglicht die Kommuikation eines Computers über Telefonleitungen durch Umsetzen digitaler Informationen in Frequenzen und umgekehrt.

Monitor - 1. Ein PROLOGUE-Dienstprogramm, das die Benutung des Computers und seines Speichers in der Assemblerebene gestattet. 2. CRT-Monitor-Bildschirm, ein Fernsehempfänger in geschlossenem Kreis.

Objektdatei (Object file) - Datei die von PROLOGUE geladen und ausgeführt werden kann. Sie wird von einem Assembler, Compiler oder Link Editor ausgegeben.

Output - Auf eine Einheit gelangende Information. Output des Computers wird oft an eine Output-Einheit gegeben, wie Bildschirm oder Drucker.

Parallel - Eine Art Interface, in dem alle Datenbits in einem gegebenen Byte gleichzeitig übertragen werden, wobei für jedes Bit eine getrennte Datenzeile verwendet wird.

Parameter - Gruppe von Elementen, deren Werte die Merkmale eines Systems bestimmen (dieses kann eine Software-Routine sein).

Password - Siehe Zugriffsschlüssel.

Peripherie (Peripheral) - Ein Zubehör, das an einem Computer angeschlossen werden kann, um seine Möglichkeit zu vergrößern (ein Floppy-Disk-Laufwerk, ein Drucker, usw.)

Platte - Eine kreisrunde Scheibe mit magnetischer Beschichtung, die für die magnetische Speicherung digitaler Informationen verwendet werden kann, wie musikalische Information auf einem Tonband gespeichert wird. Der Ausdruck Disk wird oft auch für das Laufwerk gebraucht. Siehe Diskette.

Platteneinheit (disk drive) - Elektromechanische Speichereinheit.

Programm - die Liste der Befehle, die die Arbeitsanweisungen für den Computer darstellen.

PROLOGUE - Disketten-Betriebssystem für die BOSS-Computer, die die allgemeinen Grundlagen für die Erzeugung von Programmen, Speicherung, Kontrolle und Ausführung bieten.

PROM - Merkwort für Programmable Read-Only-Memory. Dies ist ein Computer-Speicher, der seine Daten aufbewahrt, wenn der Strom abgeschaltet wird. PROM kann gelöscht (oft mit einer UV-Quelle) und neu programmiert werden.

Quelldatei - eine ASCII-Datei von Programmanweisungen, verwendet als Input an einen Assembler oder Compiler.

RAM - Merkwort für Random Access Memory - dies ist der Hauptspeicher des Systems, der Information und Programme speichert. Der RAM-Speicher ist flüchtig, d.h. er verliert die Daten, wenn der Strom ausgeschaltet wird.

Reserviertes Wort (Reserved Word) - siehe Schlüsselwort.

ROM - Merkwort für Read-Only Memory. Dieser Speichertyp ist ständig programmiert, wenn gearbeitet wird. Er kann nur gelesen werden, ein Verlust oder ein Löschen der Daten ist nicht möglich (außer bei Beschädigung).

RS232 - Ein Industriestandard-Protokoll für sequentielle Kommunikation.

Rückfrage (Prompt) - ein Symbol, das auf dem Bildschirm erscheint, und anzeigt, daß der Computer einen Befehl oder Input-Daten erwartet.

Schlüssel (Key) - siehe Zugriffschlüssel.

Schlüsselwort (Keyword) - Ein (gewöhnlich reserviertes) Wort, das eine Bedeutung in einer Computersprache besitzt. Beispiel : GOTO.

Scroll - Verschiebung des ganzen Textes auf dem Bildschirm (gewöhnlich nach oben), um Platz für weiteren Text zu machen (gewöhnlich unten).

Seite (Page) - Eine Menge Speicheradressen, adressierbar mit einem Byte 256 (FF) Stellen.

Sektor Siehe Hard Sektor.

Sequentielles Dateisystem - Ein Dateiverwaltungssystem, in dem Aufzeichnungen sequentiell erzeugt und gelesen werden.

Serial - Eine Interface-Art, in der alle Bits von Daten in einem gegebenen Byte sequentiell aufgereiht werden (gewöhnlich mit Start, Stop und Paritätsfehler-Kontroll-Bits am Anfang und am Ende des Byte), um auf einer einzelnen Datenzeile zu kommunizieren.

Software - In der Computer-Hardware ablaufende Programme.

Speicheradresse (Memory Address) - Ein Zwei Byte-Wert, der eine Speicherstelle auswählt. Jede Speicherstelle speichert ein Byte Daten.

Sprache (Language) - Ein sowohl vom Programmierer als auch vom Computer leicht verständliches System von Programmanweisungen. Eine Programmiersprache hat Syntaxregeln, die beim Schreiben von Befehlen für den Computer zu befolgen sind. Diese Befehle werden in Maschinensprache übersetzt, welche aus Sequenzen binärer Zahlen bestehen, die auf den internen Schaltungen des Mikroprozessors beruhen, und von Computer verstanden und ausgeführt werden können.

Spur (Track) - eine der konzentrischen magnetischen Spuren auf einer Diskette oder Platte.

Startroutine (Bootstrap) - Ein Programm, das in einen Lesespeicher geladen wird, aus dem es automatisch beim Einschalten oder bei Reset Betätigung gelesen wird. Damit wird die Maschine in Betriebsbereitschaft versetzt.

Subroutine - Ein Teil eines Programmes, der durch einen einzigen Subroutinen-Aufruf, wie GOSUB, ausgeführt werden kann. Subroutinen werden dafür verwendet, die gleiche Befehlsfolge an verschiedenen Stellen eines Programmes auszuführen.

Syntax - Die Regeln, die genau spezifizieren, wie ein Befehl in einer besonderen Sprache zu schreiben ist.

Text - Im Computer-Gebrauch gewöhnlich Gruppen aus Buchstaben und Zahlen in natürlicher Sprache gewöhnlich ASCII) und nicht Buchstaben und Zahlen in Computersprache.

Text Editor - siehe Editor.

Trägereinheiten (Support devices) - Vorgabeeinheit, in PROLOGUE als Systemeinheit definiert, enthält gewöhnlich Systemprogramme (BASIC,BAL), und Dienstprogramme; und Benutzereinheit, enthält Benutzerprogramme.

Typ (Type) - Ein einstelliger Dateitypenkode, der zu einem Dateinamen gehört. Er definiert den Typ der Datei als Quelle, binäre Datei, Objektdatei, Zwischendatei, Textdatei usw.

Übersetzer (Translator) - BAL-Modul, das ASCII-Quellkode übersetzt, um einen Zwischenkode zu erzeugen, der mit dem BAL-Executer ausgeführt werden kann.

Variable - der Name für eine Menge, der zu verschiedenen Zeiten verschiedene Werte zugewiesen werden können.

Verzweigung (Branch) - Normalerweise führt der Computer Programmbefehle der Reihe nach durch. Verzweigungsbefehle, wie GOTO in BASIC und BAL ermöglichen die direkte Ausführung eines anderen Punktes im Programm.

Verdichtete Quelle (Compressed source) - ein Typ einer BASIC-Quelldatei in der der ASCII-Kode in einer verdichteten Form gespeichert wird, um Speicherplatz zu sparen.

Virtueller Speicher (Virtual memory) - Eine Methode, in der der Speicherplatz auf der Diskette für die Erweiterung der Kapazität des Hauptspeichers verwendet wird. Variable werden auf der Diskette definiert und erhalten genau die gleiche Referenz, als ob sie sich im Hauptspeicher befänden. Der einzige sichtbare Unterschied ist die vergrößerte Zugriffszeit.

Volume - Eine logische Speichereinheit des PROLOGUE-Systems, in dem jede Diskette als ein Volume aufgebaut ist, mit eigenem Namen und logischen Speichermerkmalen (Granule/Sektor, Größe des Inhaltsverzeichnisses, usw.).

Zeichen (Character) - Eines der graphischen Symbole, die im System verwendet werden (aus der Tastatur und /oder am Bildschirm).

Zeichenfolge (String) - Eine Gruppe von Dateielementen (gewöhnlich ASCII-Zeichen), die in aufeinanderfolgenden Speicherplätzen gespeichert und als eine Einheit für I/O-Vorgänge, Textaufbereitung und andere Programmbearbeitungen behandelt werden.

Zeile (Line) - Eine einzelne horizontale Zeichenfolge auf einem Bildschirm die von einer Seite des Bildschirms bis zur anderen reicht. In einem Programm ist eine Befehlszeile eine Sequenz von bis zu 255 Zeichen, die durch das Kontrollzeichen RETURN abgeschlossen wird.

Zeilendrucker (Line printer) - siehe Drucker.

Zugriffschlüssel (Access key) - Kodewort, mit dem der Zugriff zu einer Datei nur den Personen ermöglicht wird, die den Schlüssel kennen.

Zwischendatei (Intermediate file) - ein von einem Compiler, wie BAL, erzeugtes Programm, das unter Kontrolle eines Laufzeit-Programms arbeitet.

INDEX