

Fortsetzung von Ausgabe 3/93, Seite 126

von Peter Aurich

Ausgabeumleitung Übertragung aller von einem (System-) Programm zum Standardausgabegerät (dem Shell-Fenster) geschickten Daten in eine Datei oder zu einem -> DOS-Gerät. Die Umleitung gilt nur während die Anweisung ausgeführt wird, in der sie angegeben wurde. Das Übertragungsziel kann sich, mit einem »>« oder »>>« am Anfang markiert, an beliebiger Stelle der Shell-Anweisung befinden:

```
dir >prt: df0:
dir df0: >ram:verzeichnis
```

DIR gibt das Inhaltsverzeichnis statt auf den Bildschirm erst auf den Drucker, dann in eine Datei aus. Bei mit »>>« gekennzeichneten Umleitungen in Dateien werden diese vor der Ausgabe nicht geleert; die Shell fügt die auszugebenden Daten dem bisherigen Inhalt hinzu.

AUTOPOINT Shell-Befehl (-> Commodities) (Extras3.0/Tools/Commodities); aktiviert das Fenster, in dem sich der Mauszeiger befindet. **Befehlsformat:** AUTOPOINT [CX_PRIORITY=<n>]

<Ctrl E> oder -> BREAK beendet das von der Shell gestartete Programm.

aux-handler -> DOS-Treiber für das -> DOS-Gerät -> AUX.

AUX 1) -> DOS-Gerät für den Datentransfer von der und zur seriellen Schnittstelle.

2) Datei im Systemverzeichnis DOSDrivers mit den zur Einbindung des gleichnamigen -> DOS-Geräts notwendigen Daten. (-> MOUNT)

AVAIL Shell-Befehl; gibt die vorhandene bzw. verfügbare RAM-Kapazität aus. **Befehlsformat:** AVAIL [CHIP/FAST/TOTAL] [FLUSH]

Bei Angabe von CHIP gibt AVAIL nur den verfügbaren Grafikspeicher, bei FAST verfügbaren Nichtgrafikspeicher (Fast-RAM) und bei TOTAL deren Summe aus. AVAIL ohne Argumente zeigt diese Informationen plus die maximal vorhandenen Kontingente aller drei Gruppen, die davon in Gebrauch befindlichen Teile sowie deren größter zusammenhängender Block. Beispiel:

```
avail fast
```

Bei Angabe der Option FLUSH entfernt das Systemprogramm zunächst alle nicht mehr genutzten Bibliotheks- und Gerätemodulmodule aus dem Speicher.

B

Bedingungskennzeichen -> Rückgabewert eines Programms bzw. Befehls. (-> IF, -> RC)

Befehlsdatei -> Kommandofolge

Befehlsformat Formale Beschreibung der Syntax einer Anweisung. Hier in der Referenz Shell 2.1/3.0 verwenden wir dasselbe Befehlsformat wie Commodore in der Systemdokumentation zum OS 2.0. Danach schreiben wir -> Shell-Befehle in Großbuchstaben. In spitze Klammern setzen wir unbedingt anzugebende -> Argumente variablen Inhalts, also Zahlenwerte oder z.B. »Datei« für eine beliebige Datei (einschl. Pfad). Argumente oder -> Schlüsselwörter in eckigen Klammern sind wahlweise und solche in geschweiften Klammern dürfen in einer Anweisung mehrmals angegeben werden (z.B. mehrere Dateinamen bei DELETE). Der senkrechte Strich bedeutet »oder« und trennt Alternativen, von denen eine auszuwählen ist. Eine weitere Methode für die Angabe der Anweisungssyntax sind -> Befehlsschablonen.

Befehlsschablone (argument template) vereinfachte Formatbeschreibung der Shell-Befehle bzw. -programme (Befehlszeilenmuster). Die Befehlsschablone enthält die Namen der -> Argumente, in der Regel zusammen mit einem oder mehreren Kennzeichen, die den Typ spezifizieren:

- ⊃ Argument/A: Das Argument muß immer angegeben werden.
- ⊃ Argument/F: Argument besteht aus einer nicht näher bestimmten Anzahl, evtl. durch Leerzeichen getrennten Zeichenfolgen. (-> /F)
- ⊃ Option/K: Das Argument muß mit dem angegebenen Schlüsselwort eingeleitet werden. (-> /K)
- ⊃ Argument/M: Angabe mehrerer Zeichenfolgen für dieses Argument möglich.
- ⊃ Wert/N: numerisches Argument.
- ⊃ Option/S: Das Argument funktioniert wie ein Schalter; die Angabe des Namens genügt, um die Funktion zu aktivieren.

Die meisten Systemprogramme geben die Befehlsschablone aus, wenn statt eines Arguments ein Fragezeichen (-> ?) angegeben wird.

Befehlszeilenmuster -> Befehlsschablone

Befehlszeilenspeicher (auch Befehlsvorgeschichte) Ein zwei KByte großer Speicherbereich der Shell für die zuletzt ausgeführten Anweisungen. Ist er voll, wird die am weitesten zurückliegende Anweisung entfernt (FIFO = First In First Out).

<Cursor_oben> holt die vorherige Anweisung in die -> Befehlszeile, die so oder korrigiert nach einem <Return> erneut ausgeführt werden kann. Jedes weitere <Cursor_oben> vor dem <Return> bringt die unmittelbar vorherige Anweisung in die Befehlszeile. Mit <Cursor_unten> kommt man wieder zurück an den Eingang des Speichers, an die am wenigsten zurückliegenden Shell-Anweisungen.

Werden die Cursor-Tasten zusammen mit <Shift> gedrückt, sucht die Shell die erste Anweisung in der jeweiligen Richtung, die genauso beginnt, wie die augenblicklich in der Befehlszeile befindliche.

Befehlszeile Historische Bezeichnung einer Stelle am Bildschirm, wo -> Anweisungen an den Computer eingegeben werden. Beim Amiga ist das weder ein fester Platz, noch nur eine Zeile. Alle Angaben hinter der -> Eingabeaufforderung (Prompt) bis zum (unsichtbaren) Abschluß mit <Return> sind die »Befehlszeile«.

Befehl In der EDV die Spezifikation einer vom Computer auszuführenden Tätigkeit. Ein Befehl beim Amiga besteht in der Regel aus dem Namen eines Programms. Die Befehlsausführung beinhaltet damit das Laden der Software, die Übergabe eventueller -> Argumente und den Programmablauf. Obwohl -> Applikationen wie Textverarbeitungen, Zeichen- oder DTP-Programme genauso von der Shell aufgerufen werden, bezeichnet man nur die -> Systemprogramme als (Shell- oder System-)Befehle.

Begrenzungszeichen (delimiter) (Sonder-)Zeichen, das zwei Zeichenfolgen trennt bzw. den Anfang und/oder das Ende einer Zeichenfolge markiert. In Shell-Anweisungen trennen Leerzeichen und Tabulator die -> Argumente. Besteht ein Argument aus mehreren Wörtern bzw. Begriffen (und nur dann), ist es in Anführungszeichen zu setzen.

Benutzerschnittstelle, textorientiert -> Shell, CLI

beschreibbar (writable) -> Attribut eines Speichermediums (meist Massenspeicher wie Diskette/Festplatte, aber auch Halbleiterspeicher), auf das (neue) Daten geschrieben und damit gespeichert werden können.

Betriebssystem (operating system) Software, die benötigt wird, um ein oder mehrere Anwenderprogramm(e) (-> Applikationen) auf dem Computer zu betreiben einschl. Organisation dafür erforderlicher bzw. dabei erzeugter -> Daten.

BINDDRIVERS -> Shell-Befehl; bindet die im Systemverzeichnis Expansion vorhandenen Treiber für Erweiterungshardware ein. **Befehlsformat:** BINDDRIVERS

BLANKER Shell-Befehl (-> Commodities) (Extras3.0/Tools/Commodities); schaltet den Bildschirm dunkel, wenn während einer bestimmten Zeit keine Eingabe per Maus oder Tastatur erfolgt. **Befehlsformat:** BLANKER [CX_PRIORITY=<n>] [CX_POPKEY=<Taste(n)>] [CX_POPUP=<yes/no>] [SECONDS=<n>] [CYCLOCOLORS=<yes/no>] [ANIMATION=<yes/no>]

Die letzten drei -> Kennzeichenparameter bestimmen, wieviele Sekunden nach der letzten Eingabe abgedunkelt wird (1 bis 9999, Vorgabe 60), und ob statt eines dunklen Bildschirms eine Animation, mit oder ohne Farbwechsel, angezeigt werden soll. Beispiel:

```
blanker cx_popup=no seconds=1
20 animation=yes
```

boot disk -> Startdiskette

Boot -> Neustart

BREAK Systembefehl; setzt alle oder die angegebene -> Unterbrechungskennung eines Shell-Prozesses. Das entspricht der Eingabe entsprechender Ctrl-Kombination im Shell-Fenster. **Befehlsformat:** BREAK <Prozeßnummer> [ALL/CID/IE/F]

```
break 5 d
```

hat dieselbe Wirkung wie <Ctrl D> eingegeben im Fenster von Shell-Prozeß 5 - eine dort ablaufende -> Kommandofolge wird abgebrochen. <Ctrl C> bricht darauf vorbereitete Programme ab.

C

Catalogs Systemverzeichnis (Locale:); enthält die Dateien mit Programmtextrn verschiedener Landessprachen in jeweils einem Unterverzeichnis - »deutsch« heißt das für uns. Die Unterverzeichnisse enthalten je ein weiteres Verzeichnis z.B. die (Fehler-)Meldungen der Shell-Befehle. bzw. für das Betriebssystem (sys) mit den Katalogdateien.

catalog Suffix für Dateien (in Locale:Catalogs) mit den Meldungen bzw. Texten eines Programms in bestimmter (Landes-) Sprache. In der Datei »dos.catalog« befinden sich z.B. die (Fehler-)Meldungen der Shell-Befehle.

CD Interner Shell-Befehl; macht das angegebene Verzeichnis zum -> aktuellen Verzeichnis (current directory) bzw. zeigt bei Aufruf ohne Parameter dessen Namen an. *Befehlsformat: CD [-Verzeichnis\Namensmuster-]*

Befindet sich das neue aktuelle Verzeichnis auf einem anderen Datenträger als das alte, ändert CD auch das -> aktuelle Gerät. Beispiele:

```
cd df0:fonts
cd sys:f#?
```

Das zweite Beispiel macht das Verzeichnis fonts (falls vorhanden) des -> logischen Datenträgers -> sys zum aktuellen. Der Einsatz von -> Namensmustern funktioniert allerdings nur, wenn das Muster auf ein einziges Verzeichnis zutrifft. Weil die bloße Eingabe eines Verzeichnisses für die Einstellung genügt, kann die Angabe von CD entfallen. Beispiel:

```
sys:f#
```

CHANGETASKPRI -> Shell-Befehl; weist einem Shell-Prozeß eine Priorität von -128 bis 127 zu. Je höher der Wert, desto mehr -> Prozessorzeit bekommt der -> Prozeß zugeteilt, und damit auch alle darüber gestarteten Programme. *Befehlsformat: CHANGETASKPRI <Priorität> [-Prozeß-]*

Fehlt die Angabe von <Prozeß>, stellt der Systembefehl die Priorität der Shell ein, von der er aufgerufen wurde.

Classes Systemverzeichnis; enthält die erforderlichen Datenstrukturen für die Anzeige und Verwaltung spezieller Dialogschalter (Unterverzeichnis Gadgets) sowie »Importtreiber« für Dateiformate wie ILBM-Grafik, 8SVX-Sound, ASCII-Text (-> DataTypes)

CLICKTOFRONT Shell-Befehl (Extras3.0:Tools/Commodities);

bringt angeklickte Fenster in den Vordergrund. *Befehlsformat: CLICKTOFRONT [CX_PRIORITY = <n>] [QUALIFIER = <Taste>]*

Über QUALIFIER kann die beim Klick zusätzlich zu drückende Taste (LAlt, RAlt, Control, None) bestimmt werden. <Ctrl E> oder -> BREAK beenden das von der Shell gestartete Programm.

CLI 1) (Command Line Interface, auch Command Line Interpreter) textorientierte Benutzerschnittstelle des Amiga; mittlerweile durch die -> Shell ersetzt. -> NEWCLI startet eine Shell und hat damit dieselbe Funktion wie NEWSHELL. Die -> Kommando-folge »cli-startup« wird nicht mehr unterstützt (-> shell-startup).

2) Systemprogramm (sys:System); öffnet eine neue Shell (das Piktogramm dazu hat den Namen Shell)

CLOCK Systemprogramm (in Workbench3.0:Utilities); legt eine Digital- oder Analoguhr - auf Wunsch mit Sekunden- und Datumsanzeige - auf die Workbench. *Befehlsformat: CLOCK [DIGITAL] [[LEFT] <n>] [[TOP] <n>] [[WIDTH] <n>] [[HEIGHT] <n>] [24 HOUR] [SECONDS] [DATE]*

Vier Argumente bestimmen den Abstand der Uhr vom Rand des Bildschirms und damit deren Position sowie (nur bei der Analoguhr) Breite und Höhe. Bei Angabe ohne Kennzeichenparameter (-> Parameter) ist die Reihenfolge der Zahlen einzuhalten. Sollen nur Höhe und Breite eingestellt werden, sind die Kennzeichenparameter einzugeben. Eine mit 24HOUR gestartete Digitaluhr zeigt die Stunden von 1 bis 24 Uhr statt zweimal von 1 bis 12.

CMD Shell-Befehl (Extras2.0:Tools); leitet die an PAR, SER oder PRT ausgegebenen Druckdaten in eine bestimmte Datei um. *Befehlsformat: CMD <Gerätename> <Datei> [OPT s/m/n]*

Als Gerätenamen sind »serial«, »parallel« oder »printer« einzusetzen. Die Optionen:

⊕ s: CMD übergeht kurze Druckausgaben am Anfang.

⊕ m: CMD leitet nicht nur eine, sondern alle Druckausgaben bis -> BREAK oder <Ctrl C> um.

⊕ n: CMD gibt Statusmeldungen ins Fenster der aufrufenden Shell aus.

Beispiel:

```
cmd parallel ram:druckdatei
```

Command Line Interface ->

CLI

Command Line Interpreter ->

CLI

Commodities Systemverzeichnis (Extras3.0:Tools) mit im Hintergrund arbeitenden Systemprogrammen (-> Commodity).

Commodity im Hintergrund arbeitendes Systemprogramm, das Funktionen realisiert wie Maus aus-, Bildschirm dunkel- oder Fenster in den Vordergrund schalten. Neben funktionsbedingten Argumenten besitzen Commodities bis zu drei Standardargumente: Über den -> Kennzeichenparameter CX_PRIORITY (Vorgabe 0) kann die -> Priorität des Commodity gegenüber anderen festgelegt werden. Auf CX_POPKEY folgt die Bezeichnung einer oder mehrerer Tasten, nach dessen Betätigen das funktionsbedingte Einstellungsfenster des Systemprogramms erscheint, und über CX_POPUP wird festgelegt, ob dieses Fenster sofort nach dem Aufruf angezeigt werden soll. (-> MOUSEBLANKER, -> BLANKER, -> FKEY, -> KLIKTOFRONT, -> CROSSDOS, -> AUTOPOINT, -> NOCAPSLOCK)

CONCLIP Shell-Befehl; realisiert eine Kopier-/Einfüge-Funktion (Copy/Paste) in Console- und damit auch in Shell-Fenstern (-> CON) oder schaltet sie ab. *Befehlsformat: CONCLIP [Unit <n>] [OFF]*

CONCLIP läuft im Hintergrund und sorgt dafür, daß Textteile in Console-Fenstern mit der Maus markiert, mit <Amiga_rechts C> in einen Zwischenspeicher (Clipboard) kopiert, und mit <Amiga_rechts V> an der aktuellen Cursor-Position wieder eingefügt werden können (benötigt ifparse.library und clipboard.device). Über das Argument Unit kann die Einheitsnummer des Clipboard-Device (0 bis 255) spezifiziert werden.

Console Line Interpreter -> CLI
Control Sequence Introducer (CSI) Zeichenfolge, die -> Steuersequenzen des DOS-Geräts -> CON einleitet. Es besteht aus den Zeichen Escape (27₁₀) und »[« (91₁₀). Escape kann durch »E« ersetzt werden (-> *).

CON -> DOS-Gerät für die Datenübertragung von der Tastatur und zum Bildschirm (Console). Das zuständige -> DOS-Treiberprogramm öffnet ein Fenster auf der Workbench. Dessen Position, Maße und Titel sind hinter den Gerätenamen zu schreiben:

```
con:50/50/400/200/Fenster
```

Bestimmte, nach CON übertragene Zeichenfolgen werden nicht ausgegeben, sondern steuern

Ausgabefunktionen wie Fenster löschen oder Cursor positionieren. Das Gerät CON arbeitet damit praktisch wie ein (ANSI-)Terminal, über das Meldungen ausgegeben sowie Eingaben zusammengestellt werden. Zusammenstellen heißt in diesem Fall, daß das Gerät einen Editierbereich zur Verfügung stellt, der sich auch über das gesamte Fenster ausdehnen kann, eingegebene Zeichen dort anzeigt, Korrekturen (z.B. mit Backspace) ausführt, Hilfsfunktionen (wie den Befehlszeilenspeicher) organisiert, und die Eingabe nach Abschluß mit <Return> an das anfordernde Programm übergibt.

Shell-Fenster sind ebenfalls CON-Fenster und deshalb wirken die Steuerzeichen auch bei der Ausgabe darin (z.B. mit ECHO). Beispiele:

```
*E[0m      Grafikmodi auf Vorgabe
*E[1m      Fettdruck ein
*E[3m      Kursiv ein
*E[4m      Unterstreichung ein
*E[7m      Invers ein
*E[8m      Text in Hintergrundfarbe
*E[22m     Fettdruck aus
*E[23m     Kursiv aus
*E[24m     Unterstreichen aus
*E[27m     Invers aus
*E[34m     Text in Farbe # (4-8)
*E[39m     Text in Vorgabefarbe 1
*E[44m     Text vor Hintergrundfarbe #
*E[49m     Text vor Hintergrund der Vorgabefarbe 0
*E[#u      Maximale Zeilenlänge im Fenster: #
*E[#t      Anzahl Zeilen im Fenster: #
*E[#x      Text beginnt # Pixel vom oberen Fenster-rand entfernt
*E[#y      Text beginnt # Pixel vom linken Fenster-rand entfernt
*E[xx;yyH  Cursor in Spalte xx, Zeile yy
*E[J       Löschen des Fensters ab Cursor
*E[@       Leerzeichen einfügen, Rest der Zeile nach rechts verschieben
*E[#A      Cursor nach oben
*E[#B      Cursor nach unten
*E[#C      Cursor nach rechts
*E[#D      Cursor nach links
*E[#E      Cursor Zeile tiefer
*E[#F      Cursor Zeile höher
*E[#K      zum Zeilenende löschen
*E[#L      Zeile einfügen
*E[#M      Zeile löschen
*E[#P      Zeichen löschen
*E[#S      nach oben (sc)rollen
*E[#T      nach unten (sc)rollen
```

Beispiel:

```
echo **E[1mText in Fett
echo **E[1;1H*E[JFenster ist gelöscht*
```

COPY (Von der Standard-Startup-Sequence resident gemacht) -> Systemprogramm; kopiert eine oder mehrere Dateien in eine andere Datei bzw. den Inhalt eines Verzeichnisses in ein anderes Verzeichnis. **Befehlsformat:** COPY [FROM] <Name/Nameismuster> [TO] <Name/Nameismuster> [ALL] [QUIET] [BUFBUFFER = <n>] [CLONE] [DATES] [COM] [NOPRO] [NOREQ]

COPY überschreibt vorhandene TO-Dateien bzw. legt nicht vorhandene TO-Verzeichnisse an. ALL sorgt dafür, daß COPY nicht nur die Dateien angegebener Verzeichnisse kopiert, sondern auch alle derer Unterverzeichnisse. Bei Angabe von QUIET werden die Namen der gerade kopierten Dateien nicht ausgegeben. Hinter BUF erwartet das Systemprogramm die Anzahl anzulegender Kopierpuffer (à 512 Byte). Alle weiteren Optionen bestimmen, welche Attribute (-> Schutzbits, Datum, Kommentare) der Datei(en) übernommen werden: Anlagedatum (DATES), Kommentare (COM), keine Übernahme der Schutzbits (NOPRO) und alle Attribute übernehmen (CLONE).

Normalerweise meldet COPY über System-Requester nicht angemeldete Geräte, auf die der Befehl zugreifen soll, oder sonstige Fehlerzustände (Diskette schreibgeschützt oder voll). NOREQ verhindert das – und damit bricht COPY beim Auftreten solcher Zustände ab. Beispiel:

```
copy datei df0:verzeichnis/Dat
eikopie
copy datei dh0:verzeichnis
copy -(#?.info) work:progs
copy verzeichnis1 verzeichnis2
all quiet
```

(-> "", -> *, -> Namensmuster).

Countries Systemverzeichnis (Locale); enthält die Dateien mit landestypischen Merkmalen wie Währungssymbol, Zahlenformat oder Maßeinheiten.

CPU -> Systemprogramm; aktiviert/deaktiviert CPU-Caches für schnelle Befehls- und Datenzugriffe oder zeigt deren augenblickliche Einstellung an, kopiert ROM-Bereiche in schnellen 32-Bit-Speicher oder aktiviert eine Funktion, die unzulässige Speicherzugriffe abfängt. **Befehlsformat:** CPU [CACHE] [BURST] [NOBURST] [DATACACHE] [DATABURST] [NODATACACHE] [NODATABURST] [INSTCACHE] [INSTBURST] [NOINSTCACHE] [NOINSTBURST] [FASTROM]

[NOFASTROM] [COPYBACK] [NOCOPYBACK] [EXTERNALCACHE] [NOEXTERNALCACHE] [TRAP] [NOTRAP] [NOMMUTEST] [CHECK 68010/68020/68030/68881/68882/68851/IMMU/FPU]

Instruktions- (ab 68020) und Daten-Cache (ab 68030) können gemeinsam über die Option CACHE, oder getrennt ein- oder ausgeschaltet werden. Das gilt auch für den Burst-Modus, der die Datenübertragungs-Rate in die Caches erhöht (nur 68030 mit SCRAM). Bei 68040ern kann auch ein externer Cache eingerichtet werden.

Ist TRAP aktiviert, stellt das System bei einem Zugriff auf die ersten 256 Speicherstellen (Seite Null) oder solche oberhalb 16 MByte entsprechende Informationen an der seriellen Schnittstelle zur Verfügung.

CPU FASTROM kopiert das Kickstart-ROM ins schnelle 32-Bit-RAM und sorgt dafür, das ROM-Zugriffe in diesen RAM-Bereich umgelenkt werden. Dafür müssen Einstellungen der MMU (memory managing unit) geändert werden. Fehlt die Option NOMMUTEST, überprüft CPU, ob schon ein anderes Programm die MMU nutzt, und berücksichtigt dessen Einstellungen.

68040-Prozessoren schreiben im Copyback-Modus auch in die internen Cache-Zwischenspeicher, und speichern die Daten erst später im Hauptspeicher. Manche Programme vertragen das allerdings nicht (NOCOPYBACK angeben).

CrossDOSFileSystem -> DOS-Treiber; übernimmt den Datentransfer von und zu MS-DOS-Disketten. (-> Dateisystem)

CROSSDOS -> Commodity (Extras3.0:Tools/Commodities) zur Steuerung zweier Konvertierungsfunktionen beim Lesen und Schreiben von MS-DOS-Disketten. **Befehlsformat:** CROSSDOS [CX_PRIORITY=<n>] [CX_POPKEY=<Taste(n)>] [CX_POPUP=<yes/no>]

Der Textfilter ergänzt Amiga-Textdateien mit einem Wagenrücklauf (carriage return) vor jedem Zeilenvorschub (line feed) und am Ende ein »End of File« bzw. filtert diese Zeichen bei MS-DOS-Dateien heraus. Bei der Textkonvertierung ASCII-7 werden alle Zeichencodes von 128 bis 255 in solche von 0 bis 127 umgewandelt (das 8. Bit gelöscht), bei INTL versucht CrossDOS, nationale Zeichen wie Umlaute zu erhalten.

CrossDOS 1) Ein von »Consultron« hergestelltes Programmpaket für die Daten-Ein- und -Ausgabe von und zu Datenträgern, die im Atari- bzw. MS-DOS-Format vorliegen, also von den entsprechenden Computern gelesen bzw. beschrieben werden können. Ab OS 2.1 ist der Teil zur Verwaltung von MS-DOS-Disketten Bestandteil des Betriebssystems. Dazu gehört das »mfm.device« (sys:devs), der -> DOS-Treiber -> CrossDOSFileSystem und die Konvertierungstabellen im Verzeichnis sys:/FileSystem_Trans.

Die -> DOS-Geräte PC0 und PC1 werden durch Anklicken der entsprechenden Piktogramme im Verzeichnis DOSDrivers angemeldet. Danach können MS-DOS-Disketten im internen und ersten externen Laufwerk gelesen und beschrieben werden.

C 1) -> Systemverzeichnis (Workbench3.0:) mit den meisten -> Systemprogrammen (commands).

2) -> logischer Datenträger, der vor Ablauf der Startup-Sequence dem Systemverzeichnis Workbench3.0:C zugeordnet wird. Die Shell sucht u.a. darin ohne -> Pfad angegebene Programme. (-> Suchpfad)

D

dansk.crossdoss Textkonvertierungstabelle (in sys:/FileSystem_Trans) für die Umsetzung von MS-DOS-Texten ins Amiga-Format unter Beibehaltung dänischer Sonderzeichen. (-> CROSSDOS)

DataTypes Systemverzeichnis (Workbench3.0:Classes, Workbench3.0:Devs); enthält Datentyp-Dateien (Suffix datatype) mit Programmcode und Informationen zum Laden und Anzeigen einer bestimmten Datenform (Grafik, Sound, Text, Hilfsdateien). Man könnte die Datentyp-Dateien auch als Importtreiber bezeichnen, die von Programmen wie -> Multiview benutzt werden.

Dateiattribut Zur Datei gehörende, zusammen mit deren Inhalt auf Massenspeichern abgelegte Informationen. Dazu gehören

- ◊ Art und Zustand der Datei (-> Schutzbits),
- ◊ Anlagedatum und Datum der letzten Änderung,
- ◊ vom Anwender mit -> FILENOTE ergänzte Kommentare.

Dateisystem (file system) Routinen des Betriebssystems (-> DOS-Treiber), die Daten auf Massenspeichern organisieren.

Die Routinen der DOS-Funktionsbibliothek fordern Dateien an, liefern sie oder befehlen die Anlage von Verzeichnissen. Das Dateisystem führt die Anweisungen aus. Weil die Struktur der Massenspeicher (Dateien in Verzeichnissen) der bei anderen Computersystemen (MS-DOS, Atari, Apple) ähnelt, braucht man nur das Dateisystem zu ändern, um mit denselben Befehlen der DOS-Funktionsbibliothek Disketten anderer Systeme zu verwalten (-> CROSSDOS).

Seit OS 1.3 wird das neue FastFileSystem neben dem heute OldFileSystem genannten Dateisystem eingesetzt. Aus Kompatibilitätsgründen initialisieren aber auch die neuen FORMAT-Befehle Disketten standardmäßig für das alte System.

Datenträger, logischer Verzeichnis, dem mit dem Systembefehl -> ASSIGN ein Name zugeordnet wurde, über den wie auf eine Diskette gleichen Namens zugegriffen werden kann. Wird ein richtiger Datenträger mit dem Namen eines logischen eingelegt (angemeldet), greift das System bis zur Abmeldung darauf zu.

Die Systementwickler bezeichnen logische Datenträger als logische Geräte (logical devices). Wir haben uns für den anderen Begriff entschieden, weil er das Verständnis der Zusammenhänge erleichtert.

Datenträger (volume) Diskette oder Partition einer Festplatte. Jeder Datenträger hat einen Namen. Dessen Angabe im -> Pfad, durch einen Doppelpunkt dahinter gekennzeichnet, ist gleichbedeutend mit dem -> Wurzelverzeichnis des Datenträgers.

DATE -> Systemprogramm; stellt die -> Systemzeit (Datum und Uhrzeit) ein bzw. gibt sie auf dem Bildschirm oder in eine Datei aus. **Befehlsformat:** DATE [<Tag>] [<Datum>] [<Uhrzeit>] [TO=VER <Dateiname>]

Bei Angabe nur eines Wochentages stellt DATE das Datum auf das des entsprechenden Tags nach dem aktuellen Datum ein (auch möglich: tomorrow/yesterday bzw. morgen/gestern). Beispiele:

```
date freitag 5-jan-93 11:45
date 5-1-93
date 11:45
date mittwoch
```

(-> SETCLOCK)

d -> Tastaturliste (keymap) im deutschen Sprachraum.

Fortsetzung in der nächsten Ausgabe