

WALTER WIMMER

EINFÜHRUNG IN DIE ALLEGRO-SYSTEMVERWALTUNG

Seminarpapier zum Seminar ALLEGRO: Grundlagen, Parametrisierung.

Bad Honnef, Mai 1999

1 VORWORT	5
2 ALLEGRO - KEIN PROGRAMM SONDERN EIN PROGRAMMPAKET	5
2.1 Das Programm PRESTO.EXE und seine „Abkömmlinge“	5
2.2 Das Programm APAC.EXE	6
2.3 Das Programm SRCH.EXE	7
2.4 Das Programm ASORT.EXE	7
2.5 Das Programm IMPORT.EXE	7
2.6 Das Programm UPDATE.EXE	7
2.7 Die Programme INDEX.EXE und QRIX.EXE	8
2.8 Die neuen Windows-Programme: A99.EXE und ALCARTA.EXE	8
3 DAS COCKPIT, DIE STEUERZENTRALE VON ALLEGRO	9
3.1 Programmstart und Konfiguration	9
3.2 Die wichtigsten Menüpunkte des Cockpits	10
3.2.1 Das Menü Routinen	10
3.2.2 Das Menü Funktionen	11
3.2.3 Das Menü Dateien	11
3.2.4 Das Menü Optionen	11
4 DIE WICHTIGSTEN DATEI-TYPEN IN EINER ALLEGRO-INSTALLATION	12
4.1 Wo liegen welche Dateien	12
4.1.1 Das Programmverzeichnis	12
4.1.2 Die Datenverzeichnisse	13
4.1.3 Das „Aufrufverzeichnis“	13
4.1.4 Auswertungsreihenfolge der Verzeichnisse	14
4.2 Aus welchen Dateien besteht eine ALLEGRO-Datenbank	14
4.3 Und was braucht man sonst noch an Dateien ?	16
4.3.1 Die Parameterdateien	16
4.3.2 Dateien im ALLEGRO-Grundformat	17
4.3.3 Das User Interface	17
4.3.4 Die Hilfedateien	18
4.4 Wichtig aber unscheinbar: Die Konfigurationsdatei (CFG-Datei)	18
4.4.1 Zur Orientierung: Eine kurze Übersicht über gängige Konfigurationsdateien	18
4.4.2 Die Definition der erlaubten Datenbankfelder (Kategorien)	19
4.4.3 Die Abfrageliste	20
4.4.4 Weitere Angaben in der Konfigurationsdatei	21

5 DIE INSTALLATION VON ALLEGRO	22
5.1 Installationsquellen und Installation der Programme	22
5.2 Vorbereitungen zur Installation	23
5.3 Ein Exkurs: Spezialitäten unter WINDOWS NT 4.0	24
5.4 Netzinstallation	25
5.4.1 Peer to Peer Netze	25
5.4.2 Netzwerke mit zentralem Netzserver	25
6 AUFRUFOPTIONEN VON ALLEGRO-PROGRAMMEN	26
6.1 Zur Syntax von ALLEGRO-Programmaufrufen	27
6.2 Aufrufoptionen von übergeordneter Bedeutung	28
6.3 Die Aufrufoptionen des Katalogisierungs- und des OPAC-Programms	30
6.4 Die Aufrufoptionen des Exportprogramms	31
6.5 Aufrufoptionen bei der Sortierung von Daten	32
6.6 Die Aufrufoptionen beim Import von Daten	32
6.6.1 Umwandlung von Daten mit IMPORT.EXE	32
6.6.2 Einspielen der Daten mit UPDATE.EXE	33
6.7 Aufrufoptionen bei der Indexierung von Datenbanken	33
7 BATCHDATEIEN - EINE KURZGEFASSTE EINFÜHRUNG	34
7.1 Womit schreibt man Batchdateien ?	35
7.2 Grundsätzliche Anmerkungen zu Batchdateien	35
7.3 Ein nützliches Beispiel - die Aller'sche Testschleife	35
7.4 Aus- und Eingabeumleitung	36
7.5 Variablen in Batchdateien	38
7.5.1 Übergabe von Variablen beim Aufruf der Batchdatei	38
7.5.2 Nutzung von Environment Variablen in Batchdateien	39
7.6 Vergleichsbefehle	41
7.7 Aufruf von Batchdateien aus Batchdateien heraus	42
7.8 Erstellung eigener Menüsysteme	44
8 DIE ALLEGRO EXPORTSPRACHE	44
8.1 Grundparameter, Zwischenteile, Umkodierung von Zeichen	45
8.1.1 Grund-Parameter (Grund-Konstanten)	45

8.1.2 Zwischenteile	49
8.1.3 Kodiertabellen und Ersetzungsbefehle	50
8.2 Die Kategorienliste - der Anweisungsteil der Parameterdatei	51
8.2.1 Aufbau und Interpretation einer Befehlszeile	52
8.2.1.1 Das Konzept des Arbeitstextes	52
8.2.1.2 Sprungbefehle	53
8.2.1.3 Prä- und Postfixe	55
8.2.1.4 Spezialfälle: Bedingtes Postfix und Mehrfachpräfixe	56
8.2.1.5 Sonderkategorien	57
8.3 Die wichtigsten Befehle der ALLEGRO-Exportsprache	58
8.3.1 Manipulationsbefehle	58
8.3.2 Vergleichsbefehle	60
8.4 Zwei einfache Beispiele für Parameterdateien	61
8.4.1 Die einfachste Parameterdatei - der Pauschalexport	61
8.4.2 Vorbereitung von Daten für eine Sortierung	61
9 FORTGESCHRITTENE(RE) PARAMETRISIERUNG	63
9.1 Die Verwendung von Unterprogrammen	63
9.2 Anwendervariablen	64
9.3 Satzverknüpfungen	68
10 ALLEGRO-DATENBANKEN IM INTERNET: DER AVANTI SERVER	69
11 LITERATURHINWEISE, INTERNETQUELLEN	70

1 Vorwort

Ziel des Kurses ist es, den Teilnehmern die Hintergründe des Programmpaketes ALLEGRO soweit nahezubringen, daß sie durch das Gelernte eigene kleinere Anpassungen an Parameterdateien durchführen und einfachere ALLEGRO-Batchroutinen organisieren, bzw. Ihre Anforderungen an EDV-Abteilungen u.ä. präziser formulieren können.

Man muß sich darüber im klaren sein, daß es nicht möglich die Allegro Parametrisierung bzw. die Einrichtung eines ALLEGRO-Systems mit umfangreicheren Funktionen in einem zweitägigen Kurs zu erlernen. Hier können nur Grundlagen gelegt werden, die dann bei Interesse oder unter dem Zwang der Umstände in Eigenarbeit vertieft und erweitert werden müssen.

Der Kurs basiert aus gegebenem Anlaß auf dem von vielen Bibliotheken in Nordrhein- Westfalen genutzten sog. M oder B-Kategorienschema (mehr dazu später). Der größte Teil der Ausführungen kann jedoch problemlos auf andere Kategorienschemata übertragen werden. In diesem Kurs sollen die technischen Hintergründe des Programms im Vordergrund stehen. Die Benutzung des Katalogisierungsprogramms PRESTO.EXE kann allenfalls am Rande ein Thema sein. Für das NRW-Kategorienschema sei hier auf das „Lernbuch zur Einführung von ALLEGRO in der Bibliothek der Friedrich-Ebert-Stiftung“ verwiesen, das über Fernleihe in einer Reihe von Bibliotheken erhältlich ist.

Im Anhang sind Hinweise auf weiterführende Literatur und Internetquellen zum Thema angegeben. Literaturhinweise beziehen sich jeweils auf diese Liste.

2 ALLEGRO - Kein Programm sondern ein Programmpaket

ALLEGRO ist ein Programmpaket, daß aus einer Reihe von einzelnen Programmen besteht, die jeweils bestimmte Funktionalitäten abdecken. Das eigentliche Programm ALLEGRO.EXE, das es auch gibt ist nur noch von historischem Interesse.

Ziel bei der Einrichtung einer ALLEGRO-Installation sollte es sein, die Funktionen der einzelnen Programme unter einer Benutzeroberfläche so einzurichten, daß der einzelne Benutzer die für ihn relevanten Funktionen starten kann, ohne sich Gedanken über die Hintergründe machen zu müssen. Im ALLEGRO-Programmpaket wird als eine Möglichkeit, dies zu realisieren das Programm Cockpit mitgeliefert. Andere Möglichkeiten werden im Abschnitt über Batch-Programmierung skizziert. Zunächst sollen jedoch die wichtigsten Programme des Allegro-Programmpaketes vorgestellt werden.

2.1 Das Programm PRESTO.EXE und seine „Abkömmlinge“

PRESTO.EXE ist das eigentliche **Katalogisierungsprogramm** des Programmpaketes und somit das Programm, mit dem Kolleginnen und Kollegen bei ihrer täglichen Arbeit in der Bibliothek am häufigsten in Berührung kommen. Es kann somit leicht zu einem Synonym für das gesamte Programmpaket werden. Es gibt jedoch zahlreiche

wichtige Funktionen, die von diesem Programm nicht geleistet werden. Hier seien die Produktion von gedruckten Listen aller Art, der Import bibliographischer Fremddaten, sowie der Aufbau und die Reorganisation von Datenbank-Indices erwähnt.

PRESTO existiert in Versionen für MS-DOS, sowie für verschiedene UNIX-Dialekte, unter anderem auch für LINUX. Die MS-DOS-Version wird in Zukunft wohl verstärkt durch das neue Programm A99.EXE, ein Windows Programm abgelöst werden.

Zu PRESTO existieren eine Reihe von Abwandlungen, die an dieser Stelle erwähnt werden sollen, um die oft verwirrende Vielfalt einzelner Programme etwas zu systematisieren.

1. **ORDER.EXE:** Für dieses Programm muß eine eigene Lizenz erworben werden. Es erweitert PRESTO um verschiedene Funktionen, durch die die Verwaltung von Monographienbestellungen, sowie die Erzeugung von Exemplarsätzen zu Titelaufnahmen erleichtert werden.
2. **ALF.EXE:** Hier muß ebenfalls eine eigene Lizenz erworben werden. Das Programm dient zur EDV-gestützten Ausleihverbuchung unter ALLEGRO. Zur sinnvollen Nutzung ist es ratsam, die zu verbuchenden Materialien mit Barcodes zu versehen. Außerdem muß für jede zu verbuchende physikalische Einheit ein Exemplarsatz existieren.
3. Für eine vereinfachte Ausleihverbuchung gibt es das Programm **ALFA.EXE**, das Teil des Grundpaketes ist. Hiermit ist eine Ausleihverbuchung auch ohne Exemplarsätze und Barcodes möglich. Es eignet sich jedoch nur für kleinere Anwendungen.
4. **MENUED.EXE:** Ein Version des Katalogisierungsprogramms, in dem die Kategorien, die sich im Hintergrundspeicher befinden über ein Menü ausgewählt werden können.
5. **PRESTOG.EXE:** Ein Version des Katalogisierungsprogramms, das es ermöglicht aus dem Programm heraus andere Programme, z.B. Bildbetrachtungs-Programme, aufzurufen. Dadurch wird es möglich Multimedia-Inhalte mit Titelaufnahmen zu verbinden. Hier bieten allerdings die neuen Windows-Programme wesentlich weitergehende und einfacher zu bedienende Möglichkeiten.

Alle diese Programme bieten neben ihren Zusatzfunktionen auch die volle Funktionalität des Programms PRESTO. Es handelt sich um Erweiterungen dieses Programms.

2.2 Das Programm APAC.EXE

Das Programm APAC ist das OPAC-Programm des Programmpaketes. Es hat keinerlei Schreibfunktion. Im Gegensatz zu den anderen ALLEGRO-Programmen ist es Freeware und kann somit zusammen mit den eigenen ALLEGRO-Datenbanken an andere weitergegeben werden. Dies wird auch bei einer Reihe von CD-ROM-Anwendungen genutzt. In Zukunft dürfte es durch das neue Windows-Programm ALCARTA.EXE abgelöst werden, das für die normalen Bibliotheks-Benutzer einfacher zu bedienen ist und ihnen insgesamt vertrauter sein dürfte. Hier existiert jedoch im Gegensatz zu PRESTO schon eine Menüführung.

2.3 Das Programm SRCH.EXE

Hierbei handelt es sich um das Such- und Exportprogramm von ALLEGRO. Es hat zwei Funktionen. Zum einen kann man damit eine ALLEGRO-Datenbank nach bestimmten Suchbegriffen durchsuchen und die gefundenen Titel aus der Datenbank exportieren.

Zum anderen wird es vielfach genutzt, um Datenbestände, die im ALLEGRO-Datenbankformat oder im ALLEGRO-Grundformat vorliegen (mehr dazu später) in eine andere Form umzuwandeln. Die Form der Umwandlung wird dabei durch Parameterdateien gesteuert. Welche Parameterdatei genutzt werden soll, wird dem Programm beim Programmaufruf mitgeteilt. Durch dieses Konzept erhält ALLEGRO seine vielgerühmte Flexibilität. Die Kehrseite ist, daß ein Aufruf des Programms SRCH bei jeder Produktion von Listen u.ä. erforderlich ist. Dies geschieht über Batchdateien, die sich auch hinter vielen Menüpunkten des ALLEGRO-Cockpits verbergen.

Erst die neuen Windows-Programme haben eine eigenständige Druckfunktion. Für kompliziertere Vorgänge, wie z.B. die Erstellung von Bibliographien wird das Programm SRCH aber zunächst unverzichtbar bleiben.

2.4 Das Programm ASORT.EXE

ASORT dient zur Sortierung von Datensätzen, die im ALLEGRO-Grundformat vorliegen. Es wird hauptsächlich bei der Erzeugung sortierter Listen benötigt. Im Gegensatz zu den anderen Programmen des ALLEGRO-Programmpaketes benötigt es für seine Tätigkeit keinerlei Informationen aus Parameterdateien.

2.5 Das Programm IMPORT.EXE

Das Programm IMPORT dient dazu, Daten, die in anderen Datenformaten vorliegen in das ALLEGRO-Format umzuwandeln. Als prominentestes Beispiel sei hier die Umwandlung von Daten im MAB-Format genannt. Es ist somit bei der Nutzung von Fremddaten von CD-ROMs, DB-Disketten usw. erforderlich. Die Umwandlung der Daten wird über eine eigene Import-Parametersprache gesteuert, die in diesem Kurs aus Zeitgründen nicht behandelt werden kann, aber im „ALLEGRO-Handbuch“ dokumentiert ist.

Insgesamt gesehen ist der Import von Daten aus anderen Datenformaten meist ein recht komplexer Vorgang, da es dazu erforderlich ist, die Datenstruktur der Quelle (also zum Beispiel die MAB-Spezifikation) gut zu kennen, bzw. selbstständig zu analysieren. Glücklicherweise existieren für eine Reihe von bibliothekarischen Austauschformaten, insbesondere für MAB bereits fertige Parameterdateien. Allerdings hat hier die NRW-Variante von ALLEGRO erhebliche Defizite. So gibt es für das NRW-Schema z.B. keine MARC-Schnittstellen.

2.6 Das Programm UPDATE.EXE

UPDATE dient dazu ALLEGRO-Grunddateien, die aus Datenimporten oder sonstigen Quellen stammen, in eine ALLEGRO-Datenbank einzuspielen und zwar so, daß dabei der normale Katalogisierungsbetrieb weitergeführt werden kann. Über einen sogenannten Primärschlüssel überprüft das Programm, ob der entsprechende Datensatz bereits in der Datenbank vorhanden ist. In der Regel wird es sich dabei um die Ident-

nummer eines Verbundes, eine DB-Identnummer o.ä. handeln. UPDATE kennt verschiedene Aufrufoptionen, durch die festgelegt werden kann, wie mit Datensätzen verfahren wird, die bereits in der Datenbank vorhanden sind. Beispielsweise können sie ersetzt oder der neu einzuspielende Datensatz verworfen werden. Es besteht aber auch die Möglichkeit, beide Datensätze zu mischen.

2.7 Die Programme INDEX.EXE und QRIX.EXE

Die beiden Programme bilden eine funktionale Einheit. Sie dienen dazu einen Index über eine Datenmenge zu erstellen, die im ALLEGRO-Grundformat vorliegt, bzw. den Index einer bereits vorhandenen ALLEGRO-Datenbank zu reorganisieren.

Der Datenbankindex wird während der Katalogisierungsarbeit durch die entsprechenden Programme ständig aktualisiert. Im Gegensatz dazu bauen INDEX und QRIX einen neuen Index in einem Zug auf. Eine Katalogisierungsarbeit oder eine Recherche in der Datenbank ist während dieser Zeit nicht möglich. Erforderlich wird ein Neuaufbau des Datenbankindex zum Beispiel dann, wenn Änderungen in der sogenannten „Indexparameterdatei“ durchgeführt wurden. Diese Änderungen sollen ja für alle, auch die bereits vor diesem Zeitpunkt erfaßten Datensätze wirksam werden.

Bei der Indexierung bildet das Programm **INDEX** zunächst aus den im ALLEGRO-Format vorliegenden Daten die Indexeinträge, die ihm durch die Anweisungen in der Indexparameterdatei vorgegeben werden. Diese Angaben werden in temporären Dateien in dem Verzeichnis, in dem die Datenbank abgelegt werden soll gesammelt. Das Programm **QRIX** mischt diese Dateien dann zur fertigen Indexdatei. Die temporären Dateien werden, wenn der Vorgang erfolgreich war, automatisch gelöscht.

Das Programm QRIX besitzt noch eine Reihe weiterer Funktionen, mit denen sich auch einzelne Teile des Datenbank-Index manipulieren lassen. Hierzu sei aber auf die entsprechenden Abschnitte im Systemhandbuch verwiesen.

Die Verwaltung der Datenbank-Indices sollte von Personen durchgeführt werden, die über etwas ALLEGRO-Hintergrundwissen verfügen. Gestartet werden diese Verwaltungsroutinen am sinnvollsten über das sogenannte „Cockpit“ (dazu später mehr).

2.8 Die neuen Windows-Programme: A99.EXE und ALCARTA.EXE

A99 ist das Windows-Katalogisierungsprogramm und ALCARTA das OPAC-Programm. Beide Programme gibt es in einsatzreifen Versionen seit dem Frühjahr 1999.

Sie laufen beide nicht unter Windows 3.1., da es sich um 32 Bit Anwendungen handelt. Voraussetzung ist also Windows 95 oder Windows NT. Beim Start entnehmen die Programme ihre Informationen aus „INI-Dateien“. Die Angabe von Aufruf-Parametern, die das Starten der DOS-Programme für den ALLEGRO-Einsteiger erschwert, entfällt also.

Bei den beiden Programmen wird die Anzeige der Titel auf dem Bildschirm, wie bei ALLEGRO üblich, ebenfalls durch Parameterdateien festgelegt. Für die Anzeige muß aber ein RTF-Format erzeugt werden. Es handelt sich dabei um ein Format, das von der Firma Microsoft für den Austausch von Texten zwischen verschiedenen Textverarbeitungen entwickelt wurde. Dieses Format erlaubt es, die Titelaufnahmen - ähnlich

wie im Internet - mit Hyperlinks zu erweitern, die beim Anklicken aktiviert werden. Dadurch können z.B. Bilder oder Internet-Dokumente sofort auf dem Bildschirm angezeigt werden. Allerdings sind dazu Programme erforderlich, die die aufgerufenen Dateiformate auch anzeigen können.

3 Das Cockpit, die Steuerzentrale von ALLEGRO

Das Programm Cockpit (etwas dröger ACP.EXE) ist die Standard-Menüoberfläche, die beim Kauf von ALLEGRO mitgeliefert wird. Jeder Neueinsteiger wird somit mit ihm konfrontiert. Andere Formen von Menüoberflächen kann sich der Anwender mit Hilfe von Batchdateien selber basteln. Das Cockpit bietet eine Fülle von Möglichkeiten, mit denen jedoch Neueinsteiger leicht überfordert sind.

Das sehr empfehlenswerte ALLEGRO-Lehrbuch von Heinrich Allers widmet diesem Thema 21 Seiten. Hier muß es leider wesentlich knapper behandelt werden.

3.1 Programmstart und Konfiguration

Gestartet wird das Programm über eine Batchdatei, die Datei **CP.BAT**. Einen Aufruf für diese Datei kann man ins Windows-Menü einbinden oder auf dem Desktop ablegen. Wichtig ist dabei die Wahl des Arbeitsverzeichnisses. Bei einer Einplatz-Version kann es das Allegro-Programmverzeichnis sein, im Netzwerk sollte es ein privates Verzeichnis des Anwenders sein (dazu später mehr).

Das Cockpit kann über eine eigene Konfigurationsdatei an die Bedürfnisse des Anwenders angepaßt werden. Diese Datei heißt standardmäßig **CP.OPT**. In der Batchdatei CP.BAT kann jedoch eine andere Datei als Konfigurationsdatei angegeben werden. Jeder Benutzer kann also eine eigene Konfigurationsdatei haben. Im Netzwerk wird diese sinnvollerweise im privaten Aufrufverzeichnis des Benutzers abgelegt. Die jeweilig angegebene Konfigurationsdatei des Cockpits kann über den Punkt *Vorgaben* im Menü **Optionen** zur Bearbeitung aufgerufen werden.

Die wichtigsten Angaben, die in der Konfigurationsdatei festgelegt werden müssen sind:

1. Der Kennbuchstabe der verwendeten Konfigurationsdatei (in NRW z.B. M oder B)
2. Das ALLEGRO-Programmverzeichnis. Standardmäßig geht ALLEGRO davon aus, daß das Programmverzeichnis C:\ALLEGRO ist. Dies stimmt z.B. bei einer Netzinstallation auf keinen Fall.
3. Das Verzeichnis, in dem die normalerweise verwendete Datenbank liegt. Diese Datenbank muß aber auch auf der angegebenen Datenbank-Konfigurationsdatei basieren.

Sinnvoll ist es auch einen anderen Editor, als das standardmäßig angegebene Programm X.EXE anzugeben (z.B. das Programm EDIT, das es auf jedem WINDOWS System gibt).

Häufig benutzte Funktionen sind im Cockpit bereits fest eingebaut. Zusätzliche Funktionen kann der Benutzer in der Cockpit Konfigurationsdatei definieren. Das Schema, nach dem dabei vorgegangen werden muß, ist in der Datei erläutert. Es werden jeweils Batchdateien aufgerufen, in denen dann erst die eigentlichen Aufrufe der benö-

tigten ALLEGRO-Programme stehen. Auf diese Art können auch andere Programme (*sogar Windows Programme mit dem DOS-Befehl* `start /W PROGRAMM ...`) aus dem Cockpit heraus aufgerufen werden.

Im Cockpit werden auch die fest eingebauten Standardfunktionen über eine Batchdatei realisiert. Diese Datei wird temporär erzeugt und ruft in ihrer letzten Befehlszeile wieder das Cockpit auf. Sie heißt **CCC.BAT**. Da sie jedoch nach getaner Arbeit wieder gelöscht wird, bekommt sie der Anwender normalerweise nicht zu Gesicht. Der DOS-Befehl `dir *.bat` im ALLEGRO-Programmverzeichnis bringt an den Tag, wieviele Batchdateien Bestandteil einer normalen ALLEGRO-Installation sind. Es lohnt sich also für den Betreiber eines ALLEGRO-Systems, sich etwas in diese im WINDOWS Zeitalter fast vergessene Thematik einzuarbeiten.

3.2 Die wichtigsten Menüpunkte des Cockpits

Zwischen den einzelnen Menüs kann man sich mit den Pfeiltasten links neben dem Nummernblock auf der Tastatur bewegen. Einzelne Menüpunkte in den Menüs können auch über den andersfarbig hervorgehobenen Hotkey aktiviert werden.

3.2.1 Das Menü Routinen

Dies ist das für den Anwender wichtigste Menü. Der Leuchtbalken steht beim Start auf

- ⇒ **Benutzen:** Für die eingestellte Standard Datenbank wird hierdurch die Katalogisierung gestartet. Andere ALLEGRO-Datenbanken können über das Menü *Optionen* ausgewählt werden.
- ⇒ **Anlegen:** Eine neue oder die erste ALLEGRO-Datenbank wird angelegt.
- ⇒ **Sichern:** Hier kann die LOG-Datei der Datenbank, in der alle Schreibvorgänge in der Datenbank mit protokolliert werden auf Diskette gesichert werden. Zur Datensicherung sollte man sich jedoch besser ein Streamer-Laufwerk o.ä. besorgen und darauf die gesamte ALLEGRO-Datenbank inklusive Index regelmäßig sichern.
- ⇒ **Wiederherstellen:** Mischt die gesicherte LOG-Datei wieder in die Datenbank ein.
- ⇒ **Löschen:** Eine ALLEGRO-Datenbank kann damit gelöscht werden. Dies sollte man möglichst nicht auf die Katalogdatenbank anwenden.
- ⇒ **Organisieren:** Hier sind eine Reihe von Routinen zusammengefaßt, die sich alle mit der Reorganisation oder dem Neuaufbau des Datenbank-Index beschäftigen. Der wichtigste Punkt ist **Index wiederherstellen**. Dadurch wird ein komplett neuer Index erzeugt. Erforderlich wird dies - wie bereits erwähnt - wenn die Index-Parameterdatei verändert wurde oder wenn der Index einfach nicht mehr funktioniert. Die Funktion **Datenbank völlig neu aufbauen** verhält sich ähnlich. Mit ihr kann man jedoch auch einen ganz neuen Datenbank-Index aus ALLEGRO-Grunddateien und somit eine neue Datenbank erzeugen. Wichtig im Netzbetrieb sind die Funktionen **Datenbank sperren** und **Datenbank freigeben**, mit denen der Schreibzugriff auf die Datenbank gesperrt bzw. freigegeben wird. Durch Programmabstürze im Netzbetrieb kann es vorkommen, daß eine Datenbank gesperrt wird. Die

ALLEGRO-Datenbankadministration muß dann mittels dieser Routine eingreifen, da dieser Zustand anders nicht zu beheben ist.

3.2.2 Das Menü Funktionen

Hier können die wichtigsten Einzelprogramme des ALLEGRO-Systems aufgerufen werden. Im Unterschied zum Menü Routinen müssen hier aber alle Aufruf-Optionen der einzelnen Programme über Auswahlmenüs eingestellt werden. Der Benutzer muß also wissen, was diese Optionen für die Funktion des jeweiligen Programms bedeuten.

Das Menü Funktionen bietet dem Benutzer jedoch eine Benennung und Kommentierung der einzelnen Aufrufoptionen an und fragt nur die Optionen ab, die das jeweilige Programm wirklich benötigt. Es ist also - trotz des benötigten Hintergrundwissens- wesentlich einfacher als die „harte“ Variante, die Eingabe des Programmaufrufes am DOS-Prompt.

3.2.3 Das Menü Dateien

Im Menü Dateien kann man sich einen Überblick über die Vielzahl von Batch- und Parameterdateien verschaffen, mit deren Hilfe ALLEGRO-Anwender das System auf ihre Bedürfnisse anpassen können. Auf die einzelnen Datei-Typen werde ich später näher eingehen. Es werden jeweils nur diejenigen Parameterdateien gezeigt, die zur eingestellten Datenbank-Konfiguration gehören. Batchdateien dagegen sind konfigurationsunabhängig, nicht jedoch die jeweiligen ALLEGRO-Programmaufrufe in den Batchdateien.

Es handelt sich bei diesem Menü also um einen Art Datei-Browser, in dem die einzelnen Dateien allegro-spezifisch systematisiert werden. Jede dieser Dateien kann bearbeitet, angesehen und gedruckt werden.

Bei der Bearbeitung kommt der Editor zum Einsatz, der in der Cockpit Konfigurationsdatei eingetragen ist. Der Standard Editor X.EXE ist sehr gewöhnungsbedürftig. Man sollte also einen anderen Editor wählen. Falls das Verzeichnis, in dem dieser Editor liegt, nicht im DOS-Suchpfad angegeben ist, muß der Programmname mit der Pfadangabe eingetragen werden.

Der DOS-Suchpfad ist eine Zusammenstellung von Verzeichnissen, in denen nach Programmen und Batchdateien gesucht werden soll. Er wird normalerweise in der Datei AUTOEXEC.BAT definiert, die beim Start des PCs abgearbeitet wird.

Daneben können Dateien auch noch kopiert, umbenannt und gelöscht werden. Mit **neue Version erstellen** kann man auf der Basis ein bereits existierenden Parameterdatei eine neue erstellen und unter einem anderen Namen abspeichern.

3.2.4 Das Menü Optionen

Über diesen Menüpunkten werden die Einstellungen für die aktuelle Sitzung mit dem Cockpit verändert. Bei nächsten Aufruf startet das Programm aber wieder mit den in der Datei CP.OPT oder einer anders benannten Cockpit Konfigurationsdatei definierten Standard-Einstellungen. Die Konfigurationsdatei des Cockpits kann aber über

den Menüpunkt **Vorgaben** zur Bearbeitung aufgerufen werden. Änderungen hier wirken sich dann dauerhaft aus.

Welche Optionen der aktuellen Sitzung können verändert werden ?

⇒ **Konfiguration:** Hier kann eine andere Datenbank-Konfiguration gewählt werden. Dies ist nur für Anwender von Interesse, die Datenbanken mit unterschiedlichen Konfigurationen betreiben. Es können im weiteren nur Datenbanken ausgewählt werden, die mit der gewählten Konfiguration betrieben werden.

⇒ **Laufwerk:** Im Netzbetrieb ist es durchaus möglich, daß verschiedene ALLEGRO Datenbanken auf unterschiedlichen Netzlaufwerken liegen. Diese können hier ausgewählt werden, indem man einfach den entsprechenden Laufwerksbuchstaben (z.B. P) eingibt. Etwas irritierend ist, daß das Cockpit hier Netzlaufwerke nicht anzeigt. Grundsätzlich geht ALLEGRO davon aus, daß sich die Installation auf C:, also der lokalen Festplatte befindet.

⇒ **Datenverzeichnis:** Im angewählten Laufwerk kann man damit in einzelne Verzeichnisse schalten. Falls sich darin eine zur gewählten Konfiguration passende Datenbank findet, kann diese ausgewählt werden. In diesem Fall beziehen sich dann alle Funktionen des Menüs Routinen auf die gewählte Datenbank.

Mit **EXIT** oder **ALT X** wird das Programm Cockpit beendet.

4 Die wichtigsten Datei-Typen in einer ALLEGRO-Installation

Der ALLEGRO Neuling wird mit einer Vielfalt unterschiedlicher Dateitypen konfrontiert. Die wichtigsten Dateitypen sollen hier kurz erläutert werden. Zunächst stellt sich aber die Frage, wo diese Dateien denn eigentlich zu finden sind.

4.1 Wo liegen welche Dateien

Eine ALLEGRO-Installation verteilt sich auf folgende Verzeichnisse:

⇒ Das **Programmverzeichnis**

⇒ Das (oder die) **Datenverzeichnisse**.

⇒ Das Verzeichnis, aus dem heraus das Programm aufgerufen wird (**Aufrufverzeichnis**)

4.1.1 Das Programmverzeichnis

Hier liegt der Großteil einer ALLEGRO-Installation. Es enthält im Prinzip alle Dateien, die nicht nur von einer, sondern von allen ALLEGRO Datenbanken benötigt werden. Also die Programme, die Batchdateien, ein Großteil der Parameterdateien. Von den ALLEGRO-Programmen wird als Standard C:\ALLEGRO als Programmverzeichnis angenommen. Diese Vorgabe ist aber bei einem Netzbetrieb obsolet und stammt noch aus der Zeit der Einplatz-Anwendungen und der kleinen Festplatten.

Das Programmverzeichnis sollte auf jeden Fall in den Suchpfad aufgenommen werden. Auch im Netzbetrieb sollte dies der Fall sein. Normalerweise wird der Suchpfad mit dem Befehl

```
PATH=C:\DOS;C:\WINDOWS;D:\ALLEGRO ... (Beispiel)
```

in der Datei AUTOEXEC.BAT definiert. Bei Netzwerken erfolgt die Definition zusätzlicher Suchlaufwerke meist in den Startskripten des Netzwerk-Betriebssystems, die jeder PC beim Einloggen ins Netzwerk durchläuft. An der MS-DOS-Eingabeaufforderung kann man den Suchpfad überprüfen, indem man den Befehl **PATH** eingibt.

Ein Programmverzeichnis, das vom Standard (C:\ALLEGRO) abweicht, muß den ALLEGRO-Programmen bekannt gegeben werden. Beim Cockpit erfolgt dies - wie erwähnt - in der Konfigurationsdatei des Cockpits, bei den neuen Windows-Programmen in der INI-Datei des Programms. Beim direkten Aufruf eines Programms, z.B. in einer Batchdatei muß das Programmverzeichnis als Aufrufparameter bzw. als Environmentvariable (mehr dazu später) angegeben werden.

4.1.2 Die Datenverzeichnisse

Hier liegen die einzelnen Datenbanken, also die Ergebnisse der Arbeit mit ALLEGRO, sowie Dateien, die nur für diese eine Datenbank relevant sind. Dies können z.B. Hilfetexte sein, die sich auf eine bestimmte Datenbank beziehen.

Man sollte für jede ALLEGRO-Datenbank ein eigenes Verzeichnis anlegen. Dies ist übersichtlicher. Wo diese Verzeichnisse liegen ist im Prinzip egal. ALLEGRO-Programme gehen standardmäßig davon aus, daß sie unterhalb des Programmverzeichnisses angesiedelt werden.

Wenn die Datenverzeichnisse nicht unterhalb des ALLEGRO-Programmverzeichnisses liegen, ist bei Programmaufrufen mehr Schreibarbeit nötig, da dann der ganze Pfad des Datenverzeichnisses mit angegeben werden muß. Trotzdem wird dies in Netzumgebungen häufig der Fall sein, weil man hier vielfach zwischen Programm- und Datenbereichen unterscheidet. Dies bietet z.B. Vorteile bei der Organisation der Datensicherung.

4.1.3 Das „Aufrufverzeichnis“

Hierunter versteht man dasjenige Verzeichnis, aus dem heraus die ALLEGRO-Programme aufgerufen werden. Es ist mit dem Arbeitsverzeichnis, das man bei der Einrichtung eines Menüpunktes unter WINDOWS angeben muß, identisch. Das Aufrufverzeichnis muß keineswegs mit dem Verzeichnis identisch sein, in dem die ALLEGRO-Programme liegen.

Bei einer Einplatzinstallation ist dies zwar üblich und sinnvoll, nicht jedoch bei einer Netzinstallation, bei der unterschiedliche Nutzer mit denselben Datenbanken arbeiten. Das Aufrufverzeichnis ist dann sinnvollerweise ein privates Verzeichnis des jeweiligen Nutzers, vergleichbar z.B. einem Verzeichnis mit privaten Texten. Jeder Nutzer hat also ein anderes Aufrufverzeichnis.

In diesem Fall werden im Aufrufverzeichnis Dateien abgelegt, die Einstellungen enthalten, die speziell für den jeweiligen Benutzer gelten, z.B. die Konfigurationsdatei des Cockpits. Hinzu kommt, daß bei vielen Arbeitsvorgängen unter ALLEGRO temporäre Dateien entstehen. Auch diese müssen bei einer Netzinstallation in privaten Verzeichnissen der einzelnen Nutzer abgelegt werden, damit die sie sich nicht gegenseitig überschreiben.

Hier zeigt sich auch, wie wichtig die Aufnahme des ALLEGRO-Programmverzeichnisses in den Suchpfad ist. Die Programme könnten andernfalls nur gefunden werden, wenn man bei jedem Programmaufruf den kompletten Pfad, der zum Programm führt, angibt. Dies ist auf die Dauer etwas mühsam.

4.1.4 Auswertungsreihenfolge der Verzeichnisse

Die drei Verzeichnisse werden von den ALLEGRO-Programmen in folgender Reihenfolge ausgewertet:

1. Aufrufverzeichnis
2. Datenverzeichnis(se)
3. Programmverzeichnis

Eine Parameter- oder Hilfedatei, die im Aufrufverzeichnis liegt wird also auch dann von ALLEGRO benutzt, wenn im Daten- oder Programmverzeichnis eine andere Version dieser Datei liegt. Falls das Aufruf- und das Programmverzeichnis identisch sind, erhält natürlich gemäß dieser Logik die Version im Programmverzeichnis Vorrang vor derjenigen im Datenverzeichnis.

Man kann sich diesen Mechanismus zunutze machen, um Versionen von Parameterdateien zu entwickeln, die nur für einzelne Datenbanken oder einzelne Benutzer gelten. Somit könnte zum Beispiel jeder Nutzer in einem Netzwerk eine andere Form der Titelanzeige haben. Dies ist allerdings ein ziemlich theoretisches Beispiel, da es sehr arbeitsaufwendig für den ALLEGRO Systemverwalter wäre.

4.2 Aus welchen Dateien besteht eine ALLEGRO-Datenbank

Die Dateinamen der einzelnen Dateien, die eine ALLEGRO-Datenbank bilden, werden immer vom Namen der Indexparameterdatei, sowie vom Konfigurations-Kennbuchstaben (dem ersten Buchstaben der verwendeten Datenbank-Konfigurationsdatei) bestimmt.

Ein Beispiel:

⇒ Die Ebert-Stiftung verwendet eine Datenbank-Konfigurationsdatei B.CFG, der Konfigurationskennbuchstabe ist also B.

⇒ Dieser Kennbuchstabe bestimmt zunächst die Dateiendung der Indexparameterdatei, im weiteren aber auch die Dateiendungen aller Parameterdateien, die für Datenbanken mit dieser Konfiguration verwendet werden können.

⇒ Die Index-Parameterdatei im Beispiel heißt FES.BPI, der Konfigurationsbuchstabe taucht also als erster Buchstabe der Dateiendung wieder auf. Dies gilt für **alle** Parameterdateien. Mit einer Konfigurationsdatei M.CFG (wie sie in NRW überwiegend Verwendung findet) hieße die Index-Parameterdatei FES.MPI.

⇒ Der eigentliche Name der Indexparameterdatei (also FES) ist der Name der Datenbank. Alle Dateien, die zu dieser Datenbank gehören, beginnen mit FES...

Auf der Basis der skizzierten Vorgaben besteht unsere Datenbank aus folgenden Dateien.

B.CFG (aber auch BERTA.CFG)	Die Datenbank-Konfigurationsdatei. Sie ist die zentrale Datei und wird deshalb später noch ausführlicher besprochen werden. Sie legt fest, welche Datenfelder in der Datenbank überhaupt verwendet werden können und welchen Inhalt sie haben, also die Abfragemasken für die Katalogisierung. Daneben gibt es noch eine Reihe weiterer Konfigurationsmöglichkeiten.
FES.BPI	Die Index-Parameterdatei. Hier wird festgelegt, aus welchen Datenfeldern der erfaßten Daten Indexeinträge gebildet werden und wie sie aussehen sollen. Der Anwender hat hier völlige Gestaltungsfreiheit. Die Felder müssen lediglich in der Konfigurationsdatei als gültig erklärt sein.
STOP.BPT (fakultativ)	Die Stopwortliste. Hier sind Wörter definiert, die nicht als Stichwörter in den Index einsortiert werden sollen.
FES.TBL	Die Satztable. Hier ist eine interne Verwaltungsnummer jedes Datensatzes gespeichert.
FES.BDX	Der eigentliche Index. Hier sind in einer Baumstruktur alle gebildeten Indexeinträge gespeichert. Zu jedem Indexeintrag werden nur die Nummern der Datensätze in der Satztable gespeichert. Über diese Nummern werden die Datensätze selbst dann gefunden.
FES.STL (fakultativ)	Die Kurztitelliste. Hier werden zu jedem Datensatz die Kurzanzeigen im erweiterten Index gespeichert.
FES.RES (fakultativ)	Die Restriktionendatei. Eine Neuerung der ALLEGRO-Version 15. Ermöglicht zum Beispiel die Suche nach bestimmten Zeiträumen (alle Titel mit Erscheinungsjahr größer als 1970)
FES_1.BLD ... FES_255.BLD	Das sind die Dateien in denen die eigentlichen Katalogdaten abgespeichert werden. Es kann 255 dieser Dateien in einer Datenbank geben. Jede kann maximal 16 MB groß sein. Alle diese Dateien werden über den gemeinsamen Index erschlossen. Da man jedoch aus diesen Dateien jederzeit einen neuen Index erzeugen kann, stellen sie das eigentliche Arbeitsergebnis dar und müssen unbedingt regelmäßig gesichert werden. Die Daten in diesen Dateien liegen in einem relativ einfachen Format vor. Jeder erfaßte Datensatz beginnt mit dem ASCII Code 1. Dann folgt eine Sequenz von 4 Zeichen, in denen die interne Datensatznummer codiert ist. Dann beginnen die eigentlichen Daten. Jedes Datenfeld beginnt mit der Kategoriennummer und endet mit ASCII 0. Der gesamte Datensatz wird mit ASCII 13 10 abgeschlossen. Dann folgt der nächste Datensatz usw.

Einzig die Parameterdateien legen letztendlich fest, in welchen Formen diese erfaßten Daten auf dem Bildschirm dargestellt werden. Die ALLEGRO Katalogisierungsprogramme können sie ohne Parameterdateien lediglich in gefelderter Form darstellen.

Diese Inhalte werden dann von den entsprechenden ALLEGRO-Programmen so organisiert, daß ein funktionsfähiger und effizienter Index entsteht. Einen direkten Eingriff in die Dateien, aus denen der Index besteht, sollte man auf jeden Fall vermeiden. Der Aufbau und die laufende Aktualisierung der Index-Dateien ist Aufgabe der ALLEGRO-Programme. Der Anwender teilt den Programmen durch die Parameterdateien mit, was das Ziel ihrer Arbeit zu sein hat, aber nicht mit welchen Mitteln sie dieses Ziel erreichen. Diese sind Teil der eigentlichen Programmlogik.

4.3 Und was braucht man sonst noch an Dateien ?

Hier sollen nur die wichtigsten Datei-Typen angesprochen werden. In der Regel wird man sie im Programmverzeichnis finden.

4.3.1 Die Parameterdateien

Die Namensbildung der Parameterdateien wurde ja bereits angesprochen.

Grundsätzlich gilt dafür:

- ⇒ Die Namen müssen den alten DOS-Konventionen entsprechen (max. 8 Buchstaben eigentlicher Dateiname + max. 3 Buchstaben Dateiendung)
- ⇒ Der eigentliche Dateiname ist frei wählbar, auch wenn die ALLEGRO-Programme für bestimmte Aktionen (z.B. für die Anzeige auf dem Bildschirm) bestimmte Namen vermuten. Davon abweichende Namen von Parameterdateien müssen den Programmen durch Aufrufparameter oder bei den Windows Programmen in der INI Datei bekanntgegeben werden.
- ⇒ Der erste Buchstabe der Dateiendung muß immer dem Anfangsbuchstaben der verwendeten Datenbank-Konfigurationsdatei entsprechen (B.CFG = Dateiendung beginnt mit B, M.CFG = Dateiendung beginnt mit M usw.
- ⇒ Die restlichen beiden Buchstaben der Dateiendung bestimmen den Typ der Parameterdatei.

Man kann folgende Typen von Parameterdateien unterscheiden:

- | | |
|--|---|
| *.[Konfiguration]PI
(z.B. FES.BPI) | Eine Index-Parameterdatei. Sie ist unbedingt für die Indexierung eines Datenbestandes und somit für den Aufbau einer benutzbaren Datenbank erforderlich und regelt den Inhalt des Indexes. |
| *.[Konfiguration]PR
(z.B. D-OPAC.BPR) | Eine normale Parameterdatei. Sie kann allen möglichen Zwecken dienen, z.B. einer Bildschirmanzeige, der Erzeugung von Katalogkarten. |
| *.[Konfiguration]PT
(z.B. STOP.BPT) | Parameterdateien können modular aufgebaut werden. Man kann also einzelne Teile, die von verschiedenen Parameterdateien genutzt werden sollen, z.B. die Umkodierung von Zeichen, in „Hilfs-Parameterdateien“ dieses Typs auslagern. Diese werden dann von den eigentlichen Parameterdateien aufgerufen und auch nur so wirksam. Sie können einem ALLEGRO-Programm nicht direkt in Aufrufparametern an- |

gegeben werden.

*.[Konfiguration]IM Importparameter-Dateien. Eine Spezialform, die beim Import von Fremddaten aus anderen Datenformaten in das ALLEGRO-Format benötigt werden. Sie werden mit einer speziellen Programmiersprache, der Importsprache erstellt.

4.3.2 Dateien im ALLEGRO-Grundformat

Diese Dateien haben immer eine Dateiendung nach dem Schema *.[Konfiguration]LG, also z.B. FES.BLG. Sie sind zwar nicht Grundbestandteil einer ALLEGRO-Installation, entstehen aber bei der intensiveren Arbeit mit ALLEGRO so häufig, daß sie an dieser Stelle erwähnt werden müssen.

Sie entstehen in der Regel durch Exporte aus vorhandenen Datenbanken oder durch Import von Fremddaten. Beim Export aus einer Datenbank wird durch das Programm SRCH.EXE, das dafür zuständig ist, die interne Datensatz-Nummern entfernt, da die Daten ja kein Bestandteil einer Datenbank mehr sind. Ein Datensatz hat in ihnen also folgende Struktur:

```
[ASCII 1] Feld 1 [ASCII 0] Feld 2 [ASCII 0] Feld n [ASCII 0]
[ASCII 13] [ASCII 10]
```

Durch Indexierung können aus diesen Dateien wieder neue Datenbanken erzeugt werden, sie können aber auch wiederum durch das Programm SRCH.EXE aufgegriffen und mit Hilfe von Parameterdateien verändert werden. Nicht jedes gewünschte Endergebnis ist bei der Umwandlung dieser Dateien in einem Schritt erreichbar. Die Erstellung einer Bibliographie benötigt etwa 20 Aufrufe des Programms SRCH mit jeweils unterschiedlichen Parameterdateien. Auf der anderen Seite ermöglicht es dieses Konzept aus den ALLEGRO-Daten sehr flexibel verschiedenste Erzeugnisse herzustellen.

4.3.3 Das User Interface

Bei ALLEGRO ist auch das User Interface, also die Oberfläche und die Systemmeldungen der Programme in engen Grenzen veränderbar.

Die Grenzen werden durch eine vorgegebene maximale Länge für einzelne Systemmeldungen gesteckt. Abgespeichert sind diese Informationen in Dateien, die mit den drei Buchstaben UIF beginnen.

Der Name dieser Dateien folgt folgendem Schema:

⇒ Sie beginnen alle mit UIF

⇒ Es folgt eine Zahl oder ein Buchstabe, der festlegt für welches Programm, die Datei wirksam ist.

⇒ Die letzten drei Buchstaben sind ein Sprachkürzel (GER für Deutsch; ENG für Englisch).

UIF0GER ist also ein deutsches Interface. Es enthält grundlegende Meldungen, die alle Programme betreffen. UIF4GER enthält die Systemmeldungen für das Programm SRCH usw. Einen genauen Überblick bietet hier das ALLEGRO-Systemhandbuch.

In der Regel liegen diese Dateien im ALLEGRO-Programmverzeichnis. Wenn das Aufrufverzeichnis nicht mit dem Programmverzeichnis identisch ist, muß dieses den ALLEGRO-Programmen bekannt gemacht werden, weil die Programme sonst die UIF-Dateien nicht finden, die sie benötigen und als Folge nicht gestartet werden können.

4.3.4 Die Hilfedateien

Bei einer ALLEGRO-Installation finden sich eine Vielzahl von Dateien, die mit H beginnen. Diese enthalten Hilfetexte, die dann kontextabhängig mit F1 abgerufen werden können. Sie sind im Gegensatz zu den UIF-Dateien für das Funktionieren des Systems nicht erforderlich, sondern ein Service-Angebot an die Benutzer des Systems. In ihnen können sogar Texte definiert werden, die bei der Erfassung einzelner Kategorien aufgerufen werden können und dann Erfassungshilfe geben. Für das Standard-Schema von ALLEGRO, das A-Kategorienschema ist diese Arbeit auch weitestgehend geleistet, leider nicht für nordrhein-westfälische Variante.

Es stellt sich jedoch die Frage, ob es sich noch lohnt, dies nachzuholen, da die neuen WINDOWS Programme auch bei den Hilfetexten neue Wege gehen. Hier werden mittels einer Textverarbeitung RTF-Dateien erzeugt, die dann durch Links miteinander verknüpft werden.

4.4 Wichtig aber unscheinbar: Die Konfigurationsdatei (CFG-Datei)

Ohne eine Datenbank-Konfigurationsdatei läuft mit den ALLEGRO-Programmen nichts. Konfigurationsdateien sind folgerichtig Bestandteil jeder ALLEGRO-Installation und werden bereits mitgeliefert. Ihren Platz finden sie am besten im ALLEGRO-Programmverzeichnis, da eine Konfigurationsdatei ja in der Regel für unterschiedliche Datenbanken benutzt wird.

Sinnvoller ist es schon Konfigurationsdateien für unterschiedliche Benutzer zu differenzieren. Die Konfigurationsdatei enthält auch die Abfragemaske, die der Benutzer bei der Katalogisierung zu sehen bekommt. Hier können ja die Bedürfnisse einzelner Kolleg(inn)en durchaus voneinander abweichen.

ALLEGRO ermöglicht dies über folgenden Mechanismus: Der eigentliche Kennbuchstabe der Konfiguration, der die Dateiendung der Datenbank- und Parameterdateien bestimmt, ist der erste Buchstabe des Namens der Konfigurationsdatei.

Die Dateien B.CFG, BMUELLER.CFG, BMEIER.CFG, BSCHMITZ.CFG repräsentieren also alle das gleiche Kategorienschema und erlauben die Benutzung der gleichen Datenbanken und Parameterdateien.

Wichtig ist hier jedoch, den ALLEGRO-Programmen beim Programmaufruf oder in der Konfigurationsdatei des Cockpits den vollständigen Namen der Konfigurationsdatei mitzuteilen. Der erste Buchstabe reicht hier nicht, da in diesem Fall von Frau Meier die Datei B.CFG benutzt würde, nicht ihre BMEIER.CFG.

4.4.1 Zur Orientierung: Eine kurze Übersicht über gängige Konfigurationsdateien

A.CFG Die Standard-Konfigurationsdatei von ALLEGRO. Sie orientiert sich

- (auch \$A.CFG) am Schema des alten niedersächsischen Verbundes und wird von der ALLEGRO-Entwicklungsabteilung am besten mit Parameterdateien unterstützt.
- Die ALLEGRO-Programme gehen von ihr als Standard-Konfigurationsdatei aus. Eine andere Konfiguration muß den Programmen als Aufrufparameter oder über das DOS-Environment übergeben werden.
- B.CFG (auch M.CFG) Die beiden Versionen des in Nordrhein-Westfalen verbreiteten Konfigurationsschemas, das sich an MAB orientiert. Wieso man sich nicht auf einen Konfigurationsbuchstaben geeinigt hat, bleibt irgendwie im Dunkeln. Die beiden Konfigurationen sind weitestgehend identisch. Leider erzwingt ein anderer Konfigurationsbuchstabe aber die Umbenennung aller Parameterdateien.
- P.CFG Eine Konfigurationsdatei, die es erlaubt Daten im PICA-Format zu erfassen und zu verarbeiten.
- HANS.CFG Eine von Thomas Berger, Bonn entwickelte Konfiguration, die sich auf die Bedürfnisse von Archiven hin orientiert. Sie ist nicht Teil der offiziellen ALLEGRO Dateien. Dies gilt aber auch für das Schema von NRW.

Im Prinzip kann sich jeder Anwender eine Konfigurationsdatei speziell für seine Bedürfnisse zurechtschneiden, wenn er nur die formalen Regeln befolgt. Eine Konfigurationsdatei, die bestimmte Kategorieinhalte definiert, muß aber auch durch Parameterdateien unterstützt werden, die diese Kategorieinhalte auswerten und darstellen.

4.4.2 Die Definition der erlaubten Datenbankfelder (Kategorien)

Hier gibt es leider historische Brüche, die durch die Weiterentwicklung der ALLEGRO-Programme bedingt sind. Neuere ALLEGRO-Versionen erlauben eine wesentlich präzisere Definition der erlaubten Kategorien und der in einer Kategorie erlaubten Aktionen

Bei der nordrhein-westfälischen Version ist diese Entwicklung nicht mit vollzogen worden. Dies ist an sich nicht weiter tragisch, weil die neueren Programme abwärtskompatibel sind, also mit Konfigurationsdateien des alten Typs einwandfrei funktionieren. Da man sich aber beim Datenbankentwurf um größtmögliche Stringenz bemühen sollte, ist dieses Verfahren natürlich unsauberer.

Die ALLEGRO-Programme erkennen an einem Befehl $t[2|3|4|5|6]$, der am Anfang der Konfigurationsdatei steht, ob es sich um den alten oder neuen Typ handelt. Die Zahl gibt dabei an wie viele Stellen die Kategorienkennung hat. Die B Konfiguration hätte also den Befehl $t3$, da die Kategoriennummern dreistellig sind.

Um eine Kategorie gültig zu machen, reicht es, sie aufzuführen. Hierbei legt die Reihenfolge, in der Kategorien aufgeführt werden, fest, in welcher Reihenfolge sie in der Datenbank intern abgelegt werden. Sie müssen aber nicht in den Parameterdateien in dieser Reihenfolge abgerufen werden.

#200
#320
#425

Diese Zeilen reichen bereits, um die aufgeführten Kategorien gültig zu machen. Alle anderen Angaben sind fakultativ. Es gibt hier eine Fülle möglicher Präzisierungen, die jeweils direkt im Anschluß an die Kategoriennummer angegeben werden. Wichtig sind in diesem Zusammenhang besonders evtl. zu definierende Prüfroutinen. So wird man auf jeden Fall festlegen wollen, daß z.B. beim Hauptsachtitel überprüft wird, ob das erste Wort ein Ordnungswort ist oder nicht. Dies geschieht im dreistelligen NRW-Kategorienschema wie folgt: #320^z c. Das Zeichen ^z hat den ASCII Code 30 und wird in einem MS-DOS-Editor mit STRG und Bindestrich eingegeben. Man wird bei der Definition der erlaubten Kategorien nicht um das Studium der entsprechenden Seiten im Handbuch herumkommen. Glücklicherweise ist es so, daß der Anwender die Kategoriedefinitionen bestehender Konfigurationsdateien in der Regel nicht modifizieren muß.

4.4.3 Die Abfrageliste

Dies gilt nicht für die Abfrageliste. Sie legt fest, welche Kategorien bei der Titelaufnahme abgefragt werden und in welcher Reihenfolge diese geschieht. Diese Reihenfolge kann völlig unabhängig von der internen Speicherung sein, die ja durch die Abfolge der Kategoriedefinitionen festgelegt wird.

In der Regel beginnt die Abfrageliste mit einer Typabfrage, in der verschiedene Dokumenttypen abgefragt werden.

```
Typ " 1=Monographien 2=Zeitschriften 3=Aufsätze x=Befehlsmodus"
```

Dadurch wird zu typspezifischen Folgen abgefragter Kategorien verzweigt. Bei jeder aufgeführten Kategorie sollte eine Benennung des Inhalts angegeben werden. Außerdem können bestimmte Inhalte vorgegeben werden und es können Pflichtkategorien definiert werden. Weiterhin sind Sprungbefehle möglich. So könnte etwa der Beginn der Abfragen zu Typ 1 des Beispiels aussehen.

```
-1  
574 "Datentyp : "Monographie<           Vorgegebener Text  
-A  
200 "Verfasser : "+A                     Sprungbefehl  
320 "! Sachtitel : "                     Pflichtkategorie  
335 "Titelzusatz : "  
....  
....  
....  
-x                                       Sprung zum Ende
```

Anmerkungen:

⇒ Jede abgefragte Kategorie beginnt auf einer Zeile. Eingerückte Zeilen gelten als Kommentare. Dies gilt auch für Parameterdateien.

⇒ Am Ende jedes abgefragten Typs muß der Sprung nach -x (zum Befehlsmodus) stehen

- ⇒ Das Anführungszeichen hinter der Kategoriennummer muß an der Position stehen, an der die eigentlichen Kategorieinhalte beginnen sollen. Diese Position wird in der Konfigurationsdatei durch den Befehl `k[4|5|6]` angegeben. Für ein dreistelliges Schema, wie in Nordrhein-Westfalen gilt in der Regel `k5`.
- ⇒ Der gezeigte Sprung löst eine Schleife aus, durch die beim Vorliegen eines Verfassers weitere Verfasserkategorien abgefragt werden. Bei den Folgekategorie wird jeweils eine 1 zur Zahl addiert, also 201, 202 usw.
- ⇒ Die abschließende spitze Klammer beim vorgegebenen Text bewirkt, daß sofort die nächste Kategorie abgefragt wird und der Benutzer in der Eingabemaske keine Möglichkeit zur Korrektur mehr hat.
- ⇒ Durch das Ausrufezeichen vor der Kategoriebeschreibung wird die Kategorie zur Pflichtkategorie, ein Fragezeichen macht die Kategorie zur Hinweiskategorie.

4.4.4 Weitere Angaben in der Konfigurationsdatei

In der Konfigurationsdatei werden noch eine Reihe weiterer Definitionen für die Datenbank festgelegt. Hier die wichtigsten davon.

- ⇒ Die **Artikelliste** definiert welche Wörter nicht als Ordnungswörter am Anfang von Sachtiteln u.ä. betrachtet werden. Entsprechende Angaben in der Kategoriendefinition vorausgesetzt, werden diese Wörter bei der Eingabe automatisch mit Nichtsortierzeichen gekennzeichnet. Die entsprechenden Zeilen beginnen mit **d**, zwischen zwei Wörtern steht jeweils ein Leerzeichen. In den Konfigurationsdateien sind Artikellisten enthalten (oder evtl. in die Datei ARTIKEL.CFG ausgelagert). Handlungsbedarf besteht hier nur, wenn neue Artikel aus exotischen Sprachen entdeckt werden.
- ⇒ **Erfassungs- und Änderungsdatum.** Der Befehl **cn** legt fest, in welche Kategorie automatisch das Erfassungsdatum eingetragen wird, **ce** die Kategorie für den Eintrag des Änderungsdatums. Im NRW-Kategorienschema lauten die entsprechenden Einträge `cn068` und `ce069`.
- ⇒ **Automatische Nummernvergabe.** ALLEGRO bietet die Möglichkeit automatisch laufende Nummern bei der Eingabe eines Datensatzes zu erzeugen. Diese Nummern werden als eindeutiger Primärschlüssel für den Datensatz verwendet. Es muß dann aber ein Index existieren, in dem dieser Primärschlüssel indexiert wird. Auf der Basis dieses Index erfolgt dann die Nummernerzeugung. In der Konfigurationsdatei läßt sich die Form der Nummer und die Kategorie, in der sie abgespeichert wird festlegen: Die Angaben `cg000` und `ci99.?5` legen z.B. fest, daß die Nummer in Kategorie 000 gespeichert wird und eine Form hat, bei der die Konstante 99. durch eine laufende Nummer ergänzt wird, die rechtsbündig auf fünf Stellen mit Nullen aufgefüllt wird.
- ⇒ **Nichtsortierzeichen und Nichtsortiermodus.** Hier kann festgelegt werden, welches Zeichen als Nichtsortierzeichen genutzt wird. Das Standardschema nutzt z.B. ASCII Code 170 (`N170`) als Nichtsortierzeichen, das NRW-Kategorienschema dagegen ASCII Code 95 (`N95`). Der Nichtsortiermodus legt fest, ob das Nichtsortierzeichen nur vor dem gekennzeichneten Wort steht (Sortiermodus `n0` wie im Stan-

dard-Schema) steht, oder ob dieses von zwei Nichtsortierzeichen eingeschlossen wird (Sortiermodus n1 wie im NRW-Schema).

- ⇒ Optionen zur **Konfiguration von Arbeitsspeicher-Einstellungen**. Diese Einstellungen ermöglichen es den Speicherbedarf von ALLEGRO zu verringern. Die Codes für diese Optionen beginnen alle mit **m**. Die wichtigste ist die maximale Größe der Ergebnismenge (**mr**). Eine Ergebnismenge kann in den MS-DOS-Programmen maximal 16.000 Titel umfassen. Bei Speicherproblemen sollte man zunächst diese Konstante reduzieren. Bei den MS-DOS ALLEGRO-Programmen kann es durchaus zu Problemen mit freiem Arbeitsspeicher kommen, da MS-DOS maximal 1 MB Arbeitsspeicher nutzen kann. Dies gilt auch wenn der Rechner an sich über viel mehr Arbeitsspeicher verfügt. Diesen können die DOS-Programme nicht nutzen..
- ⇒ Farbeinstellungen sind eine Wissenschaft für sich, für deren Studium die Lektüre des Handbuchs empfohlen sei. Die entsprechenden Konfigurationszeilen beginnen alle mit dem Buchstaben **w**. Ganz nett sieht folgende Kombination aus, aber das ist Geschmackssache.

wd007
wb007
wm134
ws134
wh134

5 Die Installation von ALLEGRO

5.1 Installationsquellen und Installation der Programme

Mittlerweile werden die ALLEGRO-Programme auf **CD-ROM** ausgeliefert. Die CD-ROM enthält zusätzlich noch eine Reihe Beispieldatenbanken u.a. den kompletten Katalog der UB Braunschweig.

Die Installation selbst erfolgt in zwei Schritten. Zunächst werden von der CD-ROM aus **Installationsdisketten** erzeugt. Dies geschieht unter **WINDOWS 9x und WINDOWS NT** mit dem Programm **AUTORUN32.EXE**, unter **Windows 3.1** mit dem Programm **START16.EXE**. Zur Erstellung der Installationsdisketten wird eine **Schlüsselzahl** benötigt, die jeder registrierte Anwender erhält. Über die erstellten Installationsdisketten erfolgt dann durch Aufruf der Datei **install.bat** auf der Diskette 1 die Installation. Update-Installationen können über den Menüpunkt "Neue Version installieren" des ALLEGRO-Cockpits ausgeführt werden.

Falls ein Internet-Zugang existiert, ist es möglich sich jeweils die aktuellsten Programmversionen vom **FTP-Server** in Braunschweig zu besorgen. Die IP-Adresse des FTP-Servers lautet: **134.169.20.1**. Die Nutzerkennung und das Paßwort bekommt jeder registrierte ALLEGRO-Nutzer mitgeteilt. Für die Nutzung des FTP-Servers ist es sinnvoll, sich im Internet einen speziellen FTP-Client zu besorgen. Empfehlenswert und für wissenschaftliche Einrichtungen kostenlos ist z.B. das Programm WS_FTP.

Falls der Internet-Zugang über einen **Firewall** läuft, muß man darauf achten, daß dieser so konfiguriert ist, daß er **aktive FTP-Verbindungen** ermöglicht. Viele Firewall-

Konfigurationen ermöglichen nur passive FTP-Verbindungen, die aber auf dem Braunschweiger FTP-Server nicht funktionieren.

Die Programme auf dem FTP-Server liegen als **gepackte Archive** vor. Traditionell wird für ALLEGRO-Programme das Programm LHARC als Packer verwendet. Die so gepackten Dateien sind an der Dateiendung LZH zu erkennen. Die Programme und sonstigen Dateien müssen entpackt und an die richtigen Stellen kopiert werden.

5.2 Vorbereitungen zur Installation

Vor der eigentlichen Installation müssen auf jedem Rechner, auf dem ALLEGRO genutzt werden soll, verschiedene Vorbereitungen getroffen werden. Der wichtigste Punkt ist für die Nutzung der DOS-Programme die Einbindung des Treibers **ANSI.SYS**. Dies geschieht in der Datei CONFIG.SYS. Es muß folgende Zeile in diese Datei eingefügt werden. Voraussetzung ist natürlich, daß DOS bzw. WINDOWS wirklich auf Laufwerk C: installiert sind.

WINDOWS 3.11	DEVICE=C:\DOS\ANSI.SYS
WINDOWS 95/98	DEVICE=C:\WINDOWS\COMMAND\ANSI.SYS
WINDOWS NT 4.0.	DEVICE=%SYSTEMROOT%\SYSTEM32\ANSI.SYS

*(Der Eintrag erfolgt in
%SYSTEMROOT%\SYSTEM32\CONFIG.NT)*

Die neuen WINDOWS-Programme benötigen den Treiber ANSI.SYS nicht mehr. Er sollte jedoch trotzdem installiert werden, da ja für eine Reihe von Funktionalitäten noch die alten DOS-Programme benötigt werden.

Wenn intensiver Batch-Dateien mit Environment-Varablen genutzt werden sollen, ist ein weiterer Eintrag in CONFIG.SYS sinnvoll, durch den der dafür nutzbare Speicherbereich vergrößert wird. In diesem Fall auf 2048 Byte oder 2 Kilobyte.

```
SHELL=C:\COMMAND.COM /E:2048 /P
```

Bei Windows 3.11 ist zusätzlich noch folgender Eintrag in der Datei SYSTEM.INI im WINDOWS-Verzeichnis erforderlich. INI-Dateien sind in Sektionen eingeteilt, die jeweils eine Überschrift in eckigen Klammern haben. Der Eintrag erfolgt in der Sektion **[NonWindowsApp]** und lautet **CommandEnvSize=2048**. Er regelt die Größe des Environments, das Programmen, die in DOS-Boxen laufen, zur Verfügung steht. Unter WINDOWS 95/98 wird diese Größe automatisch angepaßt.

ALCARTA und A99, die Windows-Programme, benötigen (zur Zeit jedenfalls) spezielle Fonts, die über die entsprechende Funktion der Systemsteuerung installiert werden müssen. Wenn diese Fonts, die im Archiv FONTS.LZH gepackt vorliegen, nicht installiert sind, ist keine befriedigende Bildschirmdarstellung möglich. Außerdem muß unbedingt eine Bildschirmauflösung von mindestens 800x600 Pixel eingestellt werden. Es ist also mindestens ein 15 Zoll Bildschirm erforderlich. Ein größerer Bildschirm wäre aber unbedingt von Vorteil. Außerdem funktionieren die WINDOWS-Programme nicht unter WINDOWS 3.11, da es sich um 32 Bit Anwendungen handelt.

5.3 Ein Exkurs: Spezialitäten unter WINDOWS NT 4.0

Unter Windows NT ist vieles anders. Vor den folgenden Ausführungen möchte ich mich bei Thomas Berger für die freundliche Unterstützung in diesem Punkt bedanken.

Bei einem Windows-NT Rechner gibt es eine Reihe von Environment-Variablen, die die Programmumgebung beschreiben. Den Inhalt dieser Variablen kann man sich mit dem Befehl **set** in einer DOS-Box anzeigen lassen. Wichtig zum Auffinden der Konfigurationsdateien ist der Inhalt der Variablen SYSTEMDRIVE (normalerweise C:) und SYSTEMROOT (normalerweise C:\WINNT). Für die Auswertung in Batchdateien sind weiterhin die Variablen USERNAME und COMPUTERTNAME von Interesse.

Die Konfigurationsdateien, die beim Start von MS-DOS-Programmen ausgewertet werden, heißen CONFIG.NT und AUTOEXEC.NT und liegen im Verzeichnis %SYSTEMROOT%\SYSTEM32, also normalerweise im Verzeichnis C:\WINNT\SYSTEM32.

Der Treiber ANSI.SYS wird in der Datei CONFIG.NT geladen. Hier nochmals der entsprechende Eintrag in dieser Datei:

DEVICE=%SYSTEMROOT%\SYSTEM32\ANSI.SYS

⇒ **Zum Verständnis der nächsten Abschnitte:** Der Kommandoprozessor ist ein Programm, das für die Ausführung von MS-DOS-Programmaufrufen zuständig ist. Das Standard-Programm heißt COMMAND.COM. Unter WINDOWS 3.11 und WINDOWS 95/98 wird dieser Kommandoprozessor immer implizit beim Systemstart geladen. Nicht so unter WINDOWS NT. Um das Ganze noch etwas komplizierter zu machen existieren hier zwei verschiedene Kommandoprozessoren, COMMAND.COM und CMD.EXE, die sich teilweise noch unterschiedlich verhalten. Beide liegen im Verzeichnis %SYSTEMROOT%\SYSTEM32, also normalerweise in C:\WINNT\SYSTEM32.

Bei jedem Programmaufruf, der in das WINDOWS Menü eingebunden wird, können Eigenschaften eingestellt werden. MS-DOS-Programme haben andere Eigenschaften als WINDOWS Programme. Unter WINDOWS 3.11 und WINDOWS 95/98 wird automatisch erkannt, daß es sich um ein DOS-Programm handelt und es werden die entsprechenden Eigenschaftseinstellungen angeboten. Diese werden dann in sogenannten PIF-Dateien gespeichert.

Damit eine Anwendung unter WINDOWS NT als DOS-Anwendung erkannt wird und eine PIF-Datei entsteht, muß sie indirekt über den Start eines Kommandoprozessors aufgerufen werden. Dem Kommandoprozessor wird dann der eigentlich gewünschte Programmaufruf als Option mitgegeben. Aufgerufen werden soll die Datei CP.BAT, die das Cockpit startet. Der Aufruf unter WINDOWS NT sieht dann so aus.

C:\WINNT\SYSTEM32\COMMAND.COM /C CP.BAT

Nun kann man im Eigenschaften Menü dieses Programmaufrufs über die Registerkarte "Speicher", für diese Anwendung mehr Speicher für Environment Variablen zur Verfügung stellen u.a.m. Die PIF-Datei entsteht nur, wenn als Kommandoprozessor COMMAND.COM aufgerufen wurde. Sobald sie aber einmal existiert, ist es möglich, den Programmaufruf so zu ändern, daß CMD.EXE als Kommandoprozessor verwendet wird. Hier sollte man allerdings darauf achten, daß die sogenannten NT-

Extensions aktiviert werden, falls dies nicht schon standardmäßig der Fall ist. Dies geschieht mit der Option /X, die im folgenden geänderten Beispiel gesetzt wird.

```
C:\WINNT\SYSTEM32\CMD.EXE /X /C CP.BAT
```

Wie man sieht ist die Benutzung von MS-DOS-Programmen unter WINDOWS NT ein wesentlich komplexeres Thema, als unter den anderen WINDOWS-Versionen. Was die Zukunft, sprich WINDOWS 2000 bringt, wird man sehen müssen.

5.4 Netzinstallation

In der Regel werden Netzinstallationen von EDV-Abteilungen u.ä. durchgeführt. Die gehören aber nicht zur Zielgruppe dieses Seminars. Im folgenden werden also nur einige Punkte aufgeführt, auf die man achten sollte. Ziel eines Netzwerks ist es, Ressourcen, insbesondere Dateien und Drucker, von mehreren Rechnern aus gemeinsam zu nutzen. Man kann zwischen Peer to Peer Netzen und Netzen mit zentralem Netzserver unterscheiden.

5.4.1 Peer to Peer Netze

Peer to Peer Netze sind nur für kleinere Anwendungen sinnvoll. Hier kann jeder beteiligte Rechner für andere Rechner Ressourcen freigeben. Die freigegebenen Ressourcen werden in der Regel über ein Paßwort geschützt. Als Betriebssystem der Rechner können sämtliche gängigen WINDOWS-Versionen zum Einsatz kommen.

Im Prinzip können in einem Peer to Peer Netz auf der Festplatte jedes beteiligten PCs ALLEGRO-Datenbanken und Installationen liegen. Es ist aber sinnvoll die ALLEGRO-Installation und die Datenbanken auf einem Rechner zu konzentrieren.

Wenn ein Rechner freigegebene Verzeichnisse importiert, werden sie mit bestimmten Laufwerken verknüpft. Auf dem freigebenden Rechner liegt ALLEGRO beispielsweise auf C:\ALLEGRO. Der Programmpfad ist also C:\ALLEGRO. Dieses Verzeichnis C:\ALLEGRO wird nun bei den anderen Rechnern mit dem Laufwerk F: verknüpft. Auf diesen Rechnern ist der Programmpfad also F:\. Für die Datenverzeichnisse gilt das selbe. Dem muß man bei der Gestaltung von Batchdateien bzw. durch die Bereitstellung unterschiedlicher Konfigurationsdateien für das Cockpit oder die neuen WINDOWS Programme Rechnung tragen. Diese Konfigurationsdateien sollten dann auf allen beteiligten Rechnern in ein spezielles Arbeitsverzeichnis abgelegt werden, von dem aus das Cockpit gestartet wird. Außerdem muß das ALLEGRO-Programmverzeichnis bei allen beteiligten Rechnern korrekt im Suchpfad eingetragen sein.

5.4.2 Netzwerke mit zentralem Netzserver

Hier existiert ein Zentralrechner, an dem sich alle Rechner im Netz anmelden und auf dem die ALLEGRO-Programme und Datenverzeichnisse abgelegt werden. Als Netzwerk-Betriebssystem kommt vielfach NOVELL NETWARE oder WINDOWS NT zum Einsatz. Es ist aber auch möglich einen LINUX-Rechner als Server zu verwenden, der mittels der Software SAMBA einen WINDOWS NT Server emuliert. Folgende Punkte sollte man bei einer Installation von ALLEGRO im Netz beachten.

- ⇒ In einem WINDOWS NT basierten Netzwerk muß auf jedem am Netzwerk angeschlossenen PC beim Systemstart das Programm SHARE.EXE geladen werden. Bei einem Netzwerk unter NOVELL ist dies nicht erforderlich. Statt dessen sollten die Dateien in den ALLEGRO-Datenverzeichnissen mit dem Befehl FLAG.EXE auf "shareable" gesetzt werden.
- ⇒ Das ALLEGRO Programmverzeichnis sollte in den Suchpfad integriert werden. Bei NOVELL NETWARE geschieht dies beispielsweise durch Anlegen eines sogenannten Suchlaufwerkes.
- ⇒ Pfadnamen in Netzwerken können oft sehr lang werden. Da es bei ALLEGRO Programmaufrufen, wie wir noch sehen werden, vielfach nötig ist, Pfadangaben mit anzugeben, können daraus sehr lange Befehlszeilen resultieren. Die erlaubte Länge einer Befehlszeile ist jedoch begrenzt. Wenn diese Grenze überschritten wird, passieren seltsame Sachen. Im ungünstigsten Fall kann die Datenbank beschädigt werden. So liegt die Grenze beim Aufruf von DOS-Programmen auf einem WINDOWS NT Rechner beispielsweise bei 124 Zeichen. WINDOWS 95/98 ist hier allerdings großzügiger. Aus diesem Grund ist es empfehlenswert, auf die Verzeichnisse, in denen das ALLEGRO Programmverzeichnis und die Datenverzeichnisse liegen ein Root-Mapping durchzuführen. Der Pfad zu diesen Verzeichnissen wird dabei durch einen Laufwerksbuchstaben ersetzt. Bei einem Root Mapping mit Laufwerk P: auf G:\DATEN\BIBLIO\ALLEGRO ist das Verzeichnis G:\DATEN\BIBLIO\ALLEGRO\KATDATEN dann auch als P:\KATDATEN auffindbar. Man sieht den Unterschied in der Länge.
- ⇒ Es sollte für jeden Benutzer im Netz ein privates Arbeitsverzeichnis existieren, in dem er alle Rechte hat. Aus diesem Verzeichnis heraus sollten die ALLEGRO-Programme aufgerufen werden. Bei der Benutzung des Cockpits und bei der Nutzung der neuen WINDOWS ALLEGRO-Programme ist dies sogar zwingend erforderlich, da sich die einzelnen Benutzer sonst zwangsläufig gegenseitig temporäre Dateien überschreiben. Aber auch beim Aufruf anderer ALLEGRO-Programme ist dies sinnvoll, da so sehr einfach sichergestellt werden kann, daß alle temporären Dateien, benutzerspezifisch in diesem Verzeichnis landen. Bei der Sortierung einer größeren Datenmenge können beispielsweise Dutzende temporäre Dateien entstehen.
- ⇒ Die Zugriffsrechte müssen richtig gesetzt werden. Im Programmverzeichnis benötigen alle Benutzer Leserechte. Personen, die Parameterdateien pflegen sollen, müssen in diesem Verzeichnis so agieren können, als wenn es sich auf ihrer lokalen Festplatte befände. In den Datenverzeichnissen brauchen alle Benutzer, die katalogisieren, neben dem Leserecht noch das Schreibrecht und das Recht neue Dateien anzulegen. OPAC-Nutzer brauchen nur dann ein Schreibrecht auf die Index-Dateien einer Datenbank, wenn sie die Möglichkeit haben sollen, Listen oder sonstige Datenbankexporte zu erstellen. SRCH.EXE benötigt das Schreibrecht, um Datensatzverknüpfungen über den Index aufzulösen.

6 Aufrufoptionen von ALLEGRO-Programmen

Wie bereits erwähnt wurde, sind die wichtigsten Standardroutinen bereits im ALLEGRO-Cockpit fest eingebaut. Außerdem bietet das Cockpit im Menü „**Funktio-**

nen“ eine Oberfläche über die sich die Aufrufoptionen der einzelnen ALLEGRO-Programme auswählen lassen. Es ist aber trotzdem erforderlich, einiges über Aufrufoptionen zu wissen, wenn man tiefer in die Arbeit mit ALLEGRO einsteigen will. Dies gilt besonders dann, wenn man eigene Batchdateien schreiben will oder muß, die bestimmte immer wiederkehrende Arbeitsabläufe automatisieren sollen.

Zunächst aber: Was sind Aufrufoptionen? Für seine Tätigkeit benötigt ein Programm Informationen über bestimmte Sachverhalte. Es muß z.B. wissen, wo die Datenbank ist, mit der gearbeitet werden soll, welche Dateien durchsucht werden sollen usw.

In der normalen Arbeit mit Programmen, besonders unter WINDOWS ist man es gewohnt, diese Informationen über ein Menü auszuwählen. Die DOS Programme von ALLEGRO haben eine solche Menüoberfläche nicht. Man muß ihnen also alle Informationen, die sie benötigen beim Programmstart mitteilen. Dies geschieht mit den Aufrufoptionen. Eine Eigenheit von ALLEGRO, die Systemadministratoren, die meist WINDOWS-Programme gewohnt sind, viel „Freude“ bereitet. Die beim Start getroffenen Einstellungen können während der Laufzeit des Programms nicht mehr verändert werden.

Bei den neuen WINDOWS Programmen ist dies anders. Diese benötigen nur eine einzige Aufrufoption, den Namen und Dateipfad einer INI-Datei, aus der die Programme weitere Informationen entnehmen. Für eine Reihe von ALLEGRO-Programmen gibt es aber noch kein WINDOWS Äquivalent, wenngleich hier die Tendenz eindeutig hin zu einer Integration von immer mehr Funktionalitäten, wie z.B. Sortierfunktionen, in das Kernprogramm geht. Bei den WINDOWS Programmen gibt es auch während der Laufzeit des Programms die Möglichkeit, die Tätigkeit des Programms durch die Auswahl bestimmter Optionen über Menüs zu beeinflussen.

Die einzelnen ALLEGRO-Programme benötigen unterschiedliche Aufrufoptionen. Im Kapitel 12 des Systemhandbuchs sind sie in toto beschrieben. Ich werde mich hier aber auf die wichtigsten beschränken. Viele eher „exotische“ Optionen werden ausgespart. Sehr schön ist zum Beispiel die Option **-h0**, die bei manchen ALLEGRO-Programmen verhindert, daß nach getaner Arbeit eine Melodie von Haydn gespielt wird.

6.1 Zur Syntax von ALLEGRO-Programmaufrufen

Die Syntax der ALLEGRO Programmaufrufe orientiert sich an den UNIX-Konventionen, nicht an der bei MS-DOS gebräuchlichen Art der Parameterübergabe. Es gelten folgende Grundregeln:

- ⇒ Vor jeder Aufrufoption steht ein Bindestrich, die einzelnen Optionen werden durch Leerzeichen getrennt.
- ⇒ Zwischen dem Bindestrich und dem Kennbuchstaben der Option darf kein Leerzeichen stehen.
- ⇒ Zwischen dem Kennbuchstaben der Option und dem Wert der dazugehört (dem Argument) kann ein Leerzeichen stehen, muß aber nicht.
- ⇒ Die Reihenfolge der einzelnen Optionen ist beliebig. Falls eine benötigte Option fehlt, fragt das ALLEGRO-Programm entweder den Wert ab oder es nimmt einen

Standardwert an. Dies führt regelmäßig dann zu Problemen, wenn der Wert für diese Option vom Standard abweicht.

⇒ Bestimmte Optionen können unter MS-DOS auch als Environment Variablen (mehr dazu im Abschnitt über Batchdateien) gesetzt werden. Es sind dies Optionen, die global für alle Programme von Bedeutung sind. Die Environment Variable heißt dann so, wie die Option, also z.B. -P für den Programmpfad.

6.2 Aufrufoptionen von übergeordneter Bedeutung

Folgende Optionen kommen bei allen Programmen vor und müssen angegeben oder über Environment Variablen gesetzt werden, falls sie vom Standardwert abweichen. Die Option **-d** für die benutzte Datenquelle (Datenbankverzeichnis oder ALLEGRO-Grunddatei) muß immer angegeben werden.

-d ⇒ Datenbankverzeichnis oder ALLEGRO-Grunddatei, die als Datenquelle für die auszuführende Aktion benutzt wird. Die Quelle muß korrekt mit dem Dateipfad angegeben werden. Dieser kann entweder absolut oder relativ zur Position im Dateisystem zum Zeitpunkt des Programmaufrufes angegeben werden.

⇒ Beim Katalogisierungs- und OPAC-Programm kann diese Option bis zu dreimal angegeben werden. Zwischen den drei angegebenen Datenbanken kann dann mit der Tastenkombination **ALT a** umgeschaltet werden.

⇒ Die Option kann als Environmentvariable gesetzt werden.

Beispiele:

**presto -dg:\daten\biblio\allegro\katdaten -
dg:\daten\biblio\allegro\import ...
srch -dk:\import.blg ...**

-k ⇒ Benutzte Datenbank-Konfiguration (erster Buchstabe der benutzten CFG-Datei).

⇒ Wenn diese Angabe fehlt wird die Standard-Konfiguration A angenommen. Bei anderen Kategorien-Schemata, wie etwa in NRW, muß diese Option also **immer** angegeben werden.

⇒ Die Option kann dadurch ersetzt werden, daß man die Angabe in eine Environment Variable ablegt. Dies ist nützlich, wenn man eine Batchdatei hat, die sehr viele ALLEGRO-Programmaufrufe enthält.

Beispiele:

**presto -kb -dg:\daten\biblio\allegro\katdaten ...
apac -km -dc:\allegro\katdaten ...**

-P ⇒ Programmpfad, also das Verzeichnis, in dem die Allegro-Programme und die zugehörigen Parameterdateien und insbesondere auch die UIF-Dateien installiert sind.

⇒ Als Standard wird C:\ALLEGRO angenommen

⇒ Falls die Programme - z.B. im Netz - irgendwo anders installiert sind, muß diese Option **immer** angegeben werden.

⇒ Es empfiehlt sich diese Option in Batchdateien immer sofort am Anfang der Batch Datei als Environment Variable zu setzen.

Beispiele:

set -P=g:\programm\bibsys\allegro (Als Environment Variable)

presto -P=g:\programm\bibsys\allegro -kb ... (Beim Programmaufruf)

-l ⇒ Die Sprache der Programm Meldungen. Die Programm Meldungen sind in den UIF-Dateien abgespeichert. Zur Zeit werden GER (Deutsch) und ENG (Englisch) standardmäßig unterstützt.

⇒ Als Standard wird GER angenommen.

⇒ Falls in Ausnahmefällen eine andere Sprache gewählt wird empfiehlt es sich, die Option sofort am Anfang von Batchdateien als Environment Variable zu setzen.

Beispiel:

set -l=ENG (als Environment Variable)

presto -Pg:\programm\bibsys\allegro -lGER -kb (beim Programmaufruf)

-f Die Funktionskennung des Programmes. Jedes ALLEGRO-Programm hat eine oder mehrere unterschiedliche Funktionskennungen, die dem Programm beim Aufruf angegeben werden muß.

Die wichtigsten Funktionskennungen

⇒ **1** = Katalogisierung- oder OPAC-Programm (**PRESTO.EXE, APAC.EXE**)

⇒ **4** = Export und Suchprogramm (**SRCH.EXE**)

⇒ **5** = Importprogramm (**IMPORT.EXE**)

⇒ Im Index-Programm (INDEX.EXE) ist die Funktionskennung immer zweistellig, um genau zu definieren, welche Funktion des Index-Programms gewünscht ist.

Die wichtigsten Kennungen sind

70 = Datenbank neu aufbauen

71 = Daten an eine bestehende Datenbank anhängen

i0 = Index erneuern.

⇒ Ähnliches gilt für das Einmischen von Daten in Datenbanken durch

UPDATE.EXE. Die wichtigste Funktionskennung ist

m = Mischen. Diese Kennung wird noch durch zwei Zahlen ergänzt, die genau definieren, in welcher Form die Daten eingemischt werden.

Beispiel: update -fm41 -dg:\daten\biblio\allegro\import

Neben diesen Aufrufoptionen mit mehr globaler Bedeutung, gibt es verschiedene Optionen, die bei den einzelnen ALLEGRO-Programmen wichtig sind. Die wichtigsten

sollen hier gegliedert nach häufig vorkommenden Arbeitsvorgängen aufgeführt werden.

6.3 Die Aufrufoptionen des Katalogisierungs- und des OPAC-Programms

-a ⇒ Die Zugangsberechtigung legt fest, welche Rechte ein Benutzer in der Datenbank hat.

Gültige Werte sind: **-a0**, **-a1**, **-a2**, **-a3**. Dabei bedeutet **-a0** nur Leseberechtigung, **-a3** volle Berechtigung für alle denkbaren Aktionen mit dem Programm, inklusive Löschen der gesamten Datenbank auf einen Zug.

⇒ Wenn mehrere Datenbanken parallel aufgerufen werden, kann für jede Datenbank eine andere Zugangsberechtigung vergeben werden. Das sieht dann so aus **-a300**. In der ersten Datenbank gibt es volle Zugangsberechtigung, in den anderen nur Leseberechtigung.

-n In welcher ALLEGRO-Datenbankdatei sollen neue Titel abgespeichert werden. Zur Erinnerung: In einer ALLEGRO-Datenbank kann es bis zu 255 Datendateien geben, die durch einen gemeinsamen Index erschlossen werden.

Beispiel: presto -f1 -kb -n15 -IGER -Pg:\programm\bibsys\allegro

-b ⇒ Name der Indexparameterdatei der Datenbank. Dieser wird nur benötigt, wenn in einem Datenverzeichnis, das ja mit **-d** spezifiziert wird, mehrere Datenbanken liegen.

⇒ Bei den ALLEGRO-Programmen unter UNIX wird die Angabe immer benötigt.

Beispiel: presto -f1 -a3 -kb -n1 -dg:\daten\biblio\allegro\katdaten -bfes

-p ⇒ Name der Parameterdatei, die für die Bildschirmanzeige benutzt werden soll. Als Standard wird der Dateiname d-1 angenommen.

⇒ Die Angabe erfolgt ohne Dateiendung, da diese ja durch die verwendete Konfiguration, sowie durch den Typ der Parameterdatei für das Programm bereits klar ist.

Beispiel: presto -f1 -a3 -kb -n1 -popac

-e ⇒ Name der Parameterdatei, die für den Export von Daten aus der Datenbank benutzt wird, sowie der Name und ggf. der Dateipfad einer Datei, in die diese Daten abgelegt werden.

⇒ Die Syntax lautet wie folgt: [Name der Parameterdatei]/[Pfad und Name der Exportdatei].

Beispiel: presto -f1 -a3 -kb -popac -eexport/c:\windows\tmp\export.blg

-i ⇒ Ein Sonderfall der nur wichtig ist, wenn man mit dem Katalogisierungsprogramm eine neue Datenbank anlegen will. Die Option enthält dann den Namen der Index-Parameterdatei, die dabei verwendet werden soll.

⇒ Das Verzeichnis, in dem die Datenbank angelegt werden soll, wird durch

die Option **-d** angeben.

Beispiel: presto -f1 -a3 -kb -n1 -dg:\daten\biblio\allegro\neu-ifes ...

-S ⇒ Diese Option benötigt keinen Wert. Sie muß benutzt werden, wenn das Programm auf einem Einplatzsystem benutzt wird und das DOS-Programm SHARE.EXE nicht geladen ist.

⇒ Standardmäßig arbeiten ALLEGRO-Programme im Mehrplatzbetrieb. Hier darf diese Option auf keinen Fall gesetzt werden.

Beispiel: presto -f1 -a3 -kb -s0 -dkatdaten -popac -S

6.4 Die Aufrufoptionen des Exportprogramms

-e ⇒ Name der Parameterdatei, mit deren Hilfe die Daten aus einer Datenbank oder einer ALLEGRO-Grunddatei weiterverarbeitet werden sollen, sowie der Name und ggf. der Dateipfad der Datei, in die das Ergebnis abgelegt wird.

⇒ Die Syntax lautet wie folgt: [Name der Parameterdatei]/[Pfad und Name der Exportdatei].

Beispiel:

**srch -f4 -kb -dc:\windows\tmp\export.blg -ee-
blg/c:\windows\temp\export1.blg ...**

-s ⇒ Suchbegriff. Diese Angabe ist dann wichtig, wenn man SRCH.EXE zur Freitextsuche benutzt.

⇒ Wenn man es dagegen als Exportprogramm zur Weiterverarbeitung von Daten benutzt und alle in der Datenquelle enthaltenen Datensätze verarbeiten will, reicht die Angabe **-s0**. Alternativ ist es möglich die Funktionsbezeichnung für das Programm auf **-f6** zu ändern. Dadurch wird der Exportvorgang etwas beschleunigt. Beide Beispiele sind aber im Prinzip gleichwertig.

Beispiele

**srch -f4 -s0 -kb -dc:\windows\tmp\export.blg -ee-
blg/c:\windows\temp\export1.blg**

**srch -f6 -kb -dc:\windows\tmp\export.blg -ee-
blg/c:\windows\temp\export1.blg ..**

-b ⇒ Hier wird angegeben, welche Datenbank benutzt werden soll, um Nachladebefehle in Parameterdateien auszuwerten. Die Datenbank muß genau mit Pfad und Namen der Index-Parameterdatei angegeben werden.

⇒ Die Angabe kann entfallen, wenn genau die Datenbank durchsucht wird, die auch zum Auswerten der Nachladebefehle genutzt wird.

⇒ Erforderlich ist die Option, wenn ALLEGRO-Grunddateien weiterverarbeitet werden, die bereits in einem früheren Schritt aus einer Datenbank exportiert wurden.

⇒ Bei längeren Batchdateien, in denen exportierte Daten in verschiedenen

Schritten weiterverarbeitet werden, ist es sinnvoll, diese Option als Environment Variable zu setzen, da man sich so viel Schreibarbeit spart.

Beispiel:

srch -f6 -kb -dexport.blg -ee-blg/export1.blg -
bg:\daten\biblio\allegro\katdaten\fes ...

-m ⇒ Manueller Eingriff erlaubt oder nicht: **-m1** erlaubt ihn, **-m0** nicht.

⇒ In Batchdateien sollte man immer **-m0** verwenden.

6.5 Aufrufoptionen bei der Sortierung von Daten

Das Sortierprogramm ASORT.EXE kennt nur zwei Optionen, die es unbedingt benötigt.

⇒ Den Namen und ggf. den Dateipfad der Datei, die sortiert werden soll.

⇒ Den Namen und ggf. den Dateipfad der Datei, in die die sortierten Daten geschrieben werden sollen. Die beiden Dateien dürfen nicht identisch sein.

⇒ Die Option **-u** (ohne irgendein Argument) veranlaßt, das evtl. vorhandene Sortierköpfe in den sortierten Daten entfernt werden. Sortierköpfe sind Angaben, die durch Parameterdateien in speziellen Kategorien gebildet werden, die jeweils am Anfang eines Datensatzes stehen. Dadurch werden sehr komplexe Sortierungen möglich.

Beispiel: asort c:\windows\temp\unsort.blg c:\windows\temp\sorted.blg -u

6.6 Die Aufrufoptionen beim Import von Daten

In der Regel erfordert dieser Vorgang zwei Schritte. Das Umwandeln der Daten aus einem Fremdformat, z.B. MAB und das Einspielen in eine Datenbank.

6.6.1 Umwandlung von Daten mit IMPORT.EXE

-i ⇒ Name der benutzten Importparameterdatei. Die Angabe der Dateierdung ist nicht erforderlich.

⇒ Beim Datenimport arbeiten immer eine Importparameterdatei und eine Exportparameterdatei zusammen. Nur in ihrer Kombination entsteht das gewünschte Ergebnis. Welche Arbeitsschritte dabei in welcher Datei erledigt werden, hängt von der Art der importierten Daten ab.

Beispiel:

import -f5 -kb -s0 -dc:\windows\tmp\db.mab -ii-mab -ei-mab/allegro.blg -m0 ...

-e ⇒ Name der benutzten Exportparameterdatei, sowie Dateipfad und Dateiname der Datei, in der das Ergebnis der Umwandlung gespeichert wird.

⇒ Die Syntax lautet wie folgt: [Name der Parameterdatei]/[Pfad und Name der Exportdatei].

Beispiel:

import -f5 -kb -s0 -dc:\windows\tmp\db.mab -ii-mab -ei-mab/allegro.blg -

m0 ...

- s ⇒ In den zu importierenden Daten kann analog zum Exportprogramm SRCH.EXE nach bestimmten Begriffen gesucht werden.
⇒ Wenn alle Daten importiert werden sollen, verwendet man ebenfalls analog zum Exportprogramm **-s0**.
- m ⇒ Manueller Eingriff erlaubt oder nicht: **-m1** erlaubt ihn, **-m0** nicht.
⇒ In Batchdateien sollte man immer **-m0** verwenden

6.6.2 Einspielen der Daten mit UPDATE.EXE

- u Name und ggf. Dateipfad der Datei, in der sich die Daten befinden, die in die Datenbank eingespielt werden sollen.

Das Verzeichnis der Datenbank, in die die Daten eingespielt werden sollen, wird durch die Option **-d** angegeben !!

Beispiel:

update -fm41 -kb -uc:\windows\tmp\import.blg -dg:\daten\biblio\allegro\import ..

- b ⇒ Falls in einem mit der Option **-d** spezifizierten Datenverzeichnis mehrere Datenbanken liegen, wird mit dieser Option die Datenbank durch die Angabe des Namens der Indexparameterdatei genau bestimmt.

Beispiel:

update -fm41 -kb -uc:\tmp\import.blg -dg:\daten\biblio\allegro\import .-bfes ...

- m ⇒ Manueller Eingriff erlaubt oder nicht: **-m1** erlaubt ihn, **-m0** nicht.
⇒ In Batchdateien sollte man immer **-m0** verwenden.

Beispiel:

update -fm41 -kb -uc:\tmp\import.blg -dg:\daten\biblio\allegro\import .-bfes -m0

- S ⇒ Diese Option benötigt keinen Wert. Sie muß benutzt werden, wenn das Programm auf einem Einplatzsystem benutzt wird und das DOS-Programm SHARE.EXE nicht geladen ist.
⇒ Standardmäßig arbeiten ALLEGRO-Programme im Mehrplatzbetrieb. Hier darf diese Option auf keinen Fall gesetzt werden.

Beispiel: update -fm41 -kb -uimport.blg -dkatdaten -S

6.7 Aufrufoptionen bei der Indexierung von Datenbanken

Hier werden nur wichtige Aufrufoptionen behandelt, die dem Programm INDEX.EXE übergeben werden. Dieses ruft das Programm QRIX.EXE dann selbständig auf. QRIX.EXE kann auch eigenständig aufgerufen werden. Hierzu sei aber auf das entsprechende Kapitel im ALLEGRO-Handbuch verwiesen.

ACHTUNG: Falls die ALLEGRO-Programme nicht im Standard-Programmverzeichnis C:\ALLEGRO installiert sind, muß das Programmverzeichnis als Option oder in einer Environment Variable angegeben werden, damit der automatische Aufruf von QRIX.EXE klappt.

-f ⇒ INDEX.EXE kennt eine Reihe unterschiedlicher Funktionskennungen. Die wichtigsten sind - wie erwähnt - **-f70** (Datenbankneuaufbau) und **-f71** (Erweiterung einer bestehenden Datenbank).

⇒ Für weitere Funktionskennungen sei auf das entsprechende Kapitel im ALLEGRO Handbuch verwiesen.

-n ⇒ Nummer der ersten Datendatei, die bei der Indexierung entstehen soll. Sobald diese Datei voll ist, wird zu dieser Nummer 1 addiert und die nächste Datei angelegt.

Beispiel: index -f70 -kb -n1 -dneudat.blg -Pg:\programm\bibsys\allegro

-z ⇒ Gewünschte maximale Dateigröße einer Datendatei in Byte.

⇒ Wenn diese Angabe nicht spezifiziert wird, wird nach 16 MB eine neue Datendatei begonnen.

Beispiel: index -z500000 (Die Daten-Dateien werden maximal 5 MB groß)

-e ⇒ Angabe der Index-Parameterdatei, die zur Indexierung benutzt werden soll, sowie des Verzeichnisses, in dem die ALLEGRO-Datenbank liegt oder entstehen soll.

⇒ Syntax: [Name der Indexparameterdatei]/[Verzeichnis ALLEGRO-Datenbank]

Beispiel:

set -P=g:\programm\bibsys\allegro (Environment Variable setzen)

index -f70 -kb -n1 -dc:\windows\tmp\neudat.blg -

efes/g\daten\biblio\allegro\neu

-m ⇒ Manueller Eingriff erlaubt oder nicht: **-m1** erlaubt ihn, **-m0** nicht.

⇒ In Batchdateien sollte man immer **-m0** verwenden.

7 Batchdateien - eine kurzgefaßte Einführung

MS DOS Batchdateien dienen - wie bereits erwähnt - der Automatisierung von Arbeitsvorgängen, die regelmäßig anfallen. In ihnen werden sämtliche ALLEGRO Programmaufrufe sowie sonstige DOS-Befehle (z.B. Kopierbefehle) gesammelt, die zur Erledigung eines bestimmten Vorgangs erforderlich sind. Als DOS-Befehl gilt dabei jeder Befehl, den man auch am DOS-Prompt (oder der „MS-DOS Eingabeaufforderung“ im WINDOWS Jargon) eingeben kann.

Zur Automatisierung können auch andere Skriptsprachen, wie zum Beispiel WINBATCH oder PERL, verwendet werden. Batchdateien dürften allerdings für Kolleg(inn)en, die keine EDV-Freaks sind, noch am leichtesten zu handhaben sein. Zu diesem Thema sei unbedingt auch auf das entsprechende Kapitel im ALLEGRO-Lehrbuch von Heinrich Allers verwiesen.

7.1 Womit schreibt man Batchdateien ?

Batchdateien müssen unbedingt im sogenannten ASCII-Format abgespeichert werden. Diese Dateien enthalten nur den reinen Text und keinerlei Textformatierungen. Dies gilt im Übrigen auch für die Parameterdateien, die später behandelt werden. Im Prinzip ist es möglich ASCII Dateien mit einer Textverarbeitung, z.B. WINWORD zu schreiben. Dabei muß aber unbedingt darauf geachtet werden, daß die Dateien als „**Nur Text + Zeilenwechsel**“ oder so ähnlich gespeichert werden, da sie sonst unbrauchbar sind.

Besser ist es jedoch einen reinen ASCII Editor zu verwenden, mit dessen Benutzung man sich gut vertraut machen sollte. Kostenlos, da auf jedem DOS- und WINDOWS System vorhanden, ist das Programm EDIT. Mit ALLEGRO wird der ASCII-Editor X.EXE ausgeliefert, der zwar mehr kann als EDIT, aber schwieriger zu bedienen ist. In Frage kommt auch das Programm NOTEPAD unter Windows. NOTEPAD hat - allerdings als einzigen - den Vorteil, daß es für Anwender, die überwiegend unter WINDOWS arbeiten vertrauter ist.

7.2 Grundsätzliche Anmerkungen zu Batchdateien

- ⇒ Batchdateien haben immer die Dateiendung **BAT**. Der eigentliche Dateiname ist beliebig, darf allerdings auch unter WINDOWS 9x nur maximal acht Buchstaben lang sein.
- ⇒ Beim Aufruf einer Batchdatei werden sämtliche darin gesammelten Programmaufrufe und MS-DOS-Befehle in der Reihenfolge ausgeführt, in der sie in der Batchdatei stehen.
- ⇒ Batchdateien gelten als ausführbare Dateien und werden vom Betriebssystem ähnlich wie Programme behandelt. Sie können also z.B. in WINDOWS-Menüs eingebunden werden.
- ⇒ Jeder Programmaufruf oder DOS-Befehl steht in der Batchdatei auf einer eigenen Zeile. Er darf sich nicht über mehrere Zeilen erstrecken.
- ⇒ Zeilen die mit einem Doppelpunkt beginnen sind sogenannte Sprungmarken. Sprungmarken schreibt man aus Gründen der Übersichtlichkeit meist in Großbuchstaben.
- ⇒ Alle Zeilen die mit der Zeichenfolge **rem und Leerzeichen** beginnen sind Kommentare. Alle anderen Zeilen müssen entweder gültige MS-DOS-Befehle oder Programmaufrufe oder Sprungmarken enthalten.

7.3 Ein nützliches Beispiel - die Aller'sche Testschleife

Diese Batchdatei wurde von Herrn Allers, dem Verfasser des ALLEGRO Lehrbuches entwickelt, um die Wirkung von Änderungen in Parameterdateien sofort zu testen. Bei eigenen Experimenten mit Parameterdateien kann man sie gut einsetzen.

Achtung: Die Zeilennummern am Anfang gehören nicht zur Batchdatei. Sie dienen nur der Zuordnung der Kommentare.

Zeile 1 rem Die Aller'sche Testschleife: allers.bat

Zeile 2 @echo off
Zeile 3 :LOOP
Zeile 4 presto -f1 -a3 -kb -n1 -dc:\allegro\katdaten -pd-opac -ee-
 blg/c:\tmp\export.blg -S
Zeile 5 pause
Zeile 6 edit c:\allegro\fes.bpi
Zeile 7 pause
Zeile 8 goto LOOP

Kommentare zur Batchdatei:

- Zeile 1** Ein Kommentar kenntlich an der Zeichenfolge **rem** am Zeilenanfang.
- Zeile 2** Durch diese Angabe wird verhindert, daß jede Befehlszeile vor der Ausführung erst einmal am Bildschirm angezeigt wird.
- Zeile 3** Eine Sprungmarke
- Zeile 4** Der Aufruf des ALLEGRO-Programms PRESTO. Beim Aufruf von Programmen und Batchdateien muß die Dateierweiterung nicht mit angegeben werden.

 Es wird hier davon ausgegangen, daß sich das ALLEGRO-Programmverzeichnis im DOS-Suchpfad befindet oder das aktuelle Verzeichnis ist. Das aktuelle Verzeichnis ist das Arbeitsverzeichnis, in dem man sich beim Programmaufruf befindet. Sonst müßte man den Dateipfad des Programms mit angeben.
- Zeile 5** An dieser Stelle kann man die Batchdatei durch die Tastenkombination STRG C unterbrechen. Wenn man weitermachen will, muß man erst eine beliebige andere Taste drücken, sonst pausiert die Batchdatei ewig.

 Erforderlich ist der Befehl, weil die Testschleife eine Endlosschleife ist und man ja auch irgendwann Feierabend machen will.
- Zeile 6** Eine Indexparameterdatei wird mit dem Programm EDIT zur Bearbeitung aufgerufen. EDIT liegt immer im DOS-Suchpfad.
- Zeile 7** siehe oben
- Zeile 8** Die Sprungmarke LOOP wird angesprungen. Sprünge können in der Batchdatei auch nach oben gehen nicht nur nach unten. Dadurch entstehen Schleifen, die bestimmte Vorgänge mehrmals wiederholen.

7.4 Aus- und Eingabeumleitung

Normalerweise geben Programme ihre Meldungen auf den Bildschirm aus und nehmen Befehle über die Tastatur entgegen. Diese Standardaus- und -eingabe kann man aber auf andere Geräte, wie z.B. den Drucker oder auf eine Datei umlenken.

Geräte haben unter MS-DOS symbolische Namen. **LPT1** ist der Drucker an der ersten Druckerschnittstelle, **LPT2** der Drucker an der zweiten Schnittstelle usw. Ein ganz besonderes Gerät ist das Gerät **NUL**. Es handelt sich hierbei um ein Daten-Nirwana, in das die Ausgaben eines Programms verschwinden. Dies ist in Batchdateien oftmals sehr nützlich, da man den Benutzer ja nicht mit den Systemmeldungen der aufgerufenen Programme verwirren möchte.

Die Standardausgabe wird mit dem **Zeichen** `>` umgelenkt. Falls die Umlenkung in eine Datei erfolgt, wird diese neu angelegt. Eine vorhandene alte Datei gleichen Namens wird überschrieben. Wenn als Umlenkungszeichen **zwei spitze Klammern** `>>` angegeben werden, dann wird die Ausgabe an eine vorhandene Datei angehängt.

Die Standardeingabe wird mit dem **Zeichen** `<` umgelenkt. Eine Umlenkung der Standardeingabe ist nur auf eine Datei sinnvoll. Benutzt wird die Umlenkung der Standardeingabe in Batchdateien, um Programmabfragen durch den Inhalt einer Datei zu beantworten.

Ein Programm, das in einer Batchdatei aufgerufen wird, könnte zum Beispiel erwarten, daß man die Taste J drückt, bevor es startet. Dies müßte nun jeder Benutzer dieser Batchdatei tun, damit die Batchdatei weiterläuft. Hier kommt die Eingabeumleitung ins Spiel. Das Zeichen J steht in einer Datei und wird aus dieser entnommen. Das Programm merkt den Unterschied nicht.

```
Zeile 1 rem Eine Batchdatei mit Aus- und Eingabeumleitung
Zeile 2 @echo off
Zeile 3 cls
Zeile 4 echo Lieber Benutzer, ich arbeite für Sie. Ihr Computer.
Zeile 5 srch -f6 -kb -dexport.blg -ee-blg/result.blg -m0 > NUL
Zeile 6 copy result.blg fertig.blg < c:\allegro\ja.key > NUL
Zeile 7 echo Hat alles gut geklappt > ok.dat
Zeile 8 echo Ihr Computer >> ok.dat
```

Erläuterungen:

Zeile 3 Der Befehl **cls** sorgt für einen leeren Bildschirm, damit die Meldung an den Benutzer gut sichtbar wird.

Zeile 4 Durch den Befehl **echo** wird eine Meldung für den Benutzer auf den Bildschirm geschickt. Dies ist das Einzige, was der Benutzer von der Arbeit der Batchdatei zu sehen bekommt.

Wenn die Meldung auf mehrere Zeilen verteilt werden soll, muß jede Zeile mit einem eigenen echo Befehl ausgegeben werden.

Zeile 5 Die Meldungen, die das Programm SRCH produziert werden weggelassen und somit für den Benutzer unsichtbar.

Zeile 6 Der Copy Befehl erwartet evtl. ein J um weiter zu arbeiten, falls die Zieldatei bereits existiert. Dieses J kommt aus der Datei ja.key. Die Meldung des

Programms, die normalerweise auf dem Bildschirm erscheint, wird ins Nichts geschickt.

Wenn gleichzeitig eine Eingabe- und eine Ausgabeumleitung in einer Zeile vorkommen, wird die Eingabeumleitung immer zuerst angegeben.

Zeile 7 Eine Meldung wird in eine Datei umgeleitet. Wenn die Datei schon existiert, wird sie gelöscht.

Zeile 8 Eine Meldung wird als neue Zeile an die Datei, die in Zeile 7 angelegt wurde, angehängt.

7.5 Variablen in Batchdateien

Variablen machen eine Batchdatei flexibler. Vielfach könnte eine Batchdatei für verschiedenste Aufgaben eingesetzt werden, wenn nur eine Kleinigkeit nicht anders wäre. Hier helfen Variablen weiter.

7.5.1 Übergabe von Variablen beim Aufruf der Batchdatei

Beim Aufruf können einer Batchdatei Informationen übergeben werden, auf die dann in der Batchdatei über Variablen zugegriffen werden kann. Die Übergabe dieser Informationen erfolgt dadurch, daß die einzelnen übergebenen Werte jeweils getrennt durch ein Leerzeichen hinter dem Namen der Batchdatei angegeben werden. Etwa so:

urlaub.bat sommer frankreich grönland

In der Batchdatei sind diese einzelnen übergebenen Informationen (Parameter) dann durch die Platzhalter **%1**, **%2**, **%3** usw. auswertbar. **%1** ist der erste übergebene Parameter, **%2** der zweite usw. Maximal können neun Parameter übergeben werden. Dies kann man durch Tricks umgehen, die aber in den Bereich der fortgeschrittenen Batch-Programmierung gehören.

Zeile 1 rem Auswertung von Parametern in einer Batchdatei

Zeile 2 @echo off

Zeile 3 cls

Zeile 4 echo Ich fahre im **%1** in Urlaub

Zeile 5 echo Und zwar nach **%2** nicht nach **%3**

Soviel zu meinen Urlaubsplänen. Nun ein etwas nützlicheres Beispiel, eine Modifizierung der Aller'schen Testschleife. Es könnte ja sein, daß man unterschiedliche Parameterdateien bearbeiten und testen will. Die Batchdatei, die benötigt wird, ist für alle Parameterdateien fast völlig identisch. Lediglich der Dateiname der Parameterdatei differiert. Was tun? Für jede Parameterdatei eine eigene Batchdatei schreiben? Natürlich wird das variable Element, der Dateiname als Parameter an die Batchdatei übergeben. Der Aufruf sähe dann beispielsweise so aus: **allers.bat fes.bpi** oder aber **allers.bat display.bpr**. In der Batchdatei sieht das so aus.

Zeile 1 rem Die Aller'sche Testschleife: allers.bat etwas flexibler gestaltet

Zeile 2 @echo off

Zeile 3 :LOOP
Zeile 4 presto -f1 -a3 -kb -n1 -dc:\allegro\katdaten -pd-opac -ee-
 blg/c:\tmp\export.blg -S
Zeile 5 pause
Zeile 6 edit c:\allegro\%1
Zeile 7 pause
Zeile 8 goto LOOP

Erläuterungen:

Zeile 6 Das %1 steht für den ersten und in diesem Fall einzigen übergebenen Parameter, den Dateinamen und wird bei der Abarbeitung der Batchdatei durch diesen ersetzt.

Falls man für seine Batchdatei mehrere Parameter vorsieht, z.B. 4, muß man dafür sorgen, daß die Batchdatei beim Aufruf auch alle vorgesehenen Parameter erhält.

7.5.2 Nutzung von Environment Variablen in Batchdateien

Environment Variablen sind Informationen, die in einem reservierten Speicherbereich des Rechners, eben dem Environment vorgehalten werden und einzelnen Programmen und auch Batchdateien zugänglich sind. Mit dem Befehl **set** kann man sich am DOS-Prompt die aktuell existierenden Environment Variablen anzeigen lassen. Viele Environment Variablen werden bereits beim Start des Rechners in der Datei AUTOEXEC.BAT oder beim Einloggen in ein Netzwerk belegt. Es ist aber möglich sich bei Bedarf neue Variablen zu definieren, um damit Batchdateien flexibler zu gestalten oder sich Schreibarbeit zu sparen. Auch einige ALLEGRO Aufrufoptionen können - wie wir bereits wissen - in Environment Variablen untergebracht werden. Zur Erzeugung einer Environment Variable wird ebenfalls der Befehl **set** benutzt.

In einem Netzwerk gibt es diverse Bibliotheksmitarbeiter, die sich alle mit ihrem Namen einloggen.. Wir gehen davon aus, daß dieser Name in einer Environment Variable namens LOGIN gespeichert ist. (Eine solche Variable ist in Netzwerken wirklich sehr nützlich. Fragen Sie Ihren Netzwerk Administrator danach.) Unterschiedliche Mitarbeiter sollen jetzt unterschiedliche Einstellungen beim Start Ihres Katalogisierungsprogramms bekommen.

Zeile 1 rem Nutzung von Environment Variablen in einer Batchdatei
Zeile 2 @echo off
Zeile 3 goto %LOGIN%
Zeile 4 :MEIER
Zeile 5 set PROG=presto
Zeile 6 set RECHT=-a3
Zeile 7 set DISPLAY=-pkatalog

Zeile 8 goto START
Zeile 9 :SCHMITZ
Zeile 10 :MUELLER
Zeile 11 set PROG=apac
Zeile 12 set RECHT=-a0
Zeile 13 set DISPLAY=-popac
Zeile 14 goto START
Zeile 15 :START
Zeile 16 %PROG% %RECHT% %DISPLAY% -dc:\allegro\katdaten -f1 -n1 -kb
Zeile 17 set PROG=
Zeile 18 set RECHT=
Zeile 19 set DISPLAY=

Erläuterungen:

- Zeile 3** Der Inhalt einer Environmentvariable kann auch als Ziel einer Sprungmarke benutzt werden. Wenn auf den Inhalt einer Environmentvariable zugegriffen werden soll, wird der Name der Environment-Variable in % Zeichen eingeschlossen.
- Zeile 4** Das Ziel der Sprungmarke ist ein Name, der in der Environment Variable LOGIN steht. Diese Variable enthält bei jedem angemeldeten Benutzer einen anderen Inhalt.
- Zeile 5-7** Für den Benutzer MEIER werden ALLEGRO Aufrufoptionen, die speziell für ihn gelten, in Environment Variablen untergebracht.
Die Environment Variablen werden mit **set** erzeugt und bekommen als Inhalt den Text hinter dem = Zeichen zugewiesen.
Bei der Erzeugung und beim Löschen dürfen die Namen der Environmentvariablen nicht in % Zeichen eingeschlossen werden.
- Zeile 9-10** Es können mehrere Sprungmarken unmittelbar aufeinander folgen. Dadurch wird es möglich aus den einzelnen Benutzernamen Gruppen zu bilden.
- Zeile 11-13** Die speziellen ALLEGRO Aufrufoptionen für MUELLER und SCHMITZ werden gesetzt.
- Zeile 14** Wenn die benötigten Environmentvariablen belegt sind, wird jeweils zu START gesprungen.
- Zeile 16** Der eigentliche ALLEGRO Programmaufruf. Ein Teil der benötigten Angaben steht in den Environmentvariablen. Da auf den Inhalt der Variablen zugegriffen wird, werden die Namen der Variablen in % Zeichen eingeschlossen.
- Zeile** Die Environment-Variablen, die nur für die Nutzung in dieser einen

17 - 19 Batchdatei definiert wurden, werden wieder gelöscht. Ihr Inhalt wird auf leer gesetzt. Dies sollte man nicht vergessen, um das Environment wieder zu entlasten.

Nicht gelöscht wird dagegen die Variable LOGIN, weil sie später auch wieder in anderen Batchdateien benötigt werden könnte.

7.6 Vergleichsbefehle

Vergleichsbefehle sind erforderlich, damit man in einer Batchdatei auf unterschiedliche Situationen angemessen reagieren kann. Es kann dabei der Inhalt einer Variablen oder die Existenz einer Datei geprüft werden. Es handelt sich hierbei um sogenannte Kontrollstrukturen. Im Vergleich zu anderen Skriptsprachen stehen in MS-DOS Batchdateien kaum Kontrollstrukturen zur Verfügung. Trotzdem kann man schon einiges damit machen. In der Variable LOGIN steht wieder der Benutzername im Netzwerk.

Zeile 1 rem Vergleichsbefehle in Batchdateien

Zeile 2 @echo off

Zeile 3 if exist nullbock.txt goto ENDE

Zeile 4 if %LOGIN%/==MEIER/ echo Guten Morgen Frau Meier

Zeile 5 if %LOGIN%/==MUELLER/ echo Sie schon wieder !

Zeile 6 :LOOP

Zeile 7 if %LOGIN%/==SCHMITZ/ echo Hilfe, muß das sein !!!

Zeile 8 if not %LOGIN%/==SCHMITZ/ goto ENDE

Zeile 9 goto LOOP

Zeile 10 :ENDE

Erläuterungen:

Zeile 3 Mit der Konstruktion **if exist** wird die Existenz einer Datei überprüft. Das Gegenteil ist **if not exist**. Damit wird überprüft, daß eine Datei nicht existiert.

Die angegebene Aktion wird durchgeführt, wenn die Bedingung zutrifft. In diesem Fall wird sofort ans Ende der Batchdatei gesprungen.

Zeile 4-5 Der Inhalt der Variable LOGIN wird überprüft. Die einzelnen Benutzer werden unterschiedlich freundlich begrüßt.

Zeile 7 Da auf den Inhalt der Variable zugegriffen wird, ist ihr Name in Prozentzeichen eingeschlossen.

Der eigentliche Vergleich erfolgt durch zwei Gleichheitszeichen. Das **if** am Zeilenanfang macht klar, daß auf Gleichheit geprüft werden soll. Falls diese Bedingung zutrifft, wird die Aktion, also die Ausgabe der Meldung durchgeführt.

Auf beiden Seiten, die verglichen werden, werden die zu vergleichenden Angaben jeweils durch ein / ergänzt. Dies ist erforderlich, weil die Variab-

le ja leer sein könnte. Das hieße, daß auf einer Seite des Vergleichs nichts stünde. Dies darf aber in Batchdateien nicht sein. Damit der Vergleich funktionieren kann, muß die Ergänzung auf beiden Seiten durchgeführt werden.

Zeile 8 Hier wird auf Ungleichheit der beiden zu vergleichenden Angaben geprüft. Dies ist am Kontrollwort **if not** zu erkennen. Nur wenn beide Seiten nicht gleich sind wird die Aktion, also der Sprung an das Ende der Batchdatei ausgeführt.

In der Konsequenz bedeutet dies, daß Herr Schmitz sein ganzes weiteres Berufsleben hindurch von dieser Batchdatei gemobbt wird.

7.7 Aufruf von Batchdateien aus Batchdateien heraus

Es ist möglich und oft sinnvoll, aus Batchdateien heraus andere Batchdateien aufzurufen. Dies geschieht mit dem Befehl **call** [Name der aufgerufenen Batchdatei]. Dadurch wird es möglich bestimmte Befehlsfolgen oder das Setzen bestimmter Environment Variablen, die in verschiedensten Batchdateien gebraucht werden in einer Datei zusammen zu fassen. Diese wird dann in den anderen Batchdateien an den Stellen aufgerufen, an denen die definierten Aktionen benötigt werden.

Nach der Abarbeitung der mit **call** aufgerufenen Batchdatei, wird die Batchdatei, in der der Aufruf erfolgte weiter ab gearbeitet.

Im folgenden Beispiel werden Environment-Variablen in einer zentralen Batchdatei verwaltet. Außerdem wird in dieser jeweils in ein privates Verzeichnis des Benutzers geschaltet. Dieses Verzeichnis ist nach dem Namen des Benutzers benannt, der jeweils aus der schon bekannten Variablen LOGIN entnommen wird.

Zeile 1 rem Eine Batchdatei ruft eine weitere Batchdatei auf.

Zeile 2 @echo off

Zeile 3 call al_sets.bat START

Zeile 4 %-P%\presto -f1 -a3 -d%AL_DD% -pkatalog -f1 -a3 -ee-
blg/%AL_UD%\01.blg

Zeile 5 call al_sets.bat STOP

Erläuterungen:

Zeile 3 Die Batchdatei al_sets.bat wird aufgerufen. Dabei wird ein Parameter übergeben, der in al_sets.bat mit %1 genutzt werden kann.

Zeile 4 Der Programmaufruf für das Katalogisierungsprogramm wird durchgeführt. Dabei werden die Variablen benutzt, die in al_sets.bat belegt wurden.

Zeile 5 Die Batchdatei al_sets.bat wird nochmals aufgerufen. Dabei wird ein Parameter übergeben, der in al_sets.bat mit %1 genutzt werden kann

Zeile 1 rem Die aufgerufene Batchdatei al_sets.bat

Zeile 2 @echo off
Zeile 3 goto %1
Zeile 4 :START
Zeile 5 set -P=g:\programm\bibsys\allegro
Zeile 6 set -k=b
Zeile 7 set AL_DD=g:\daten\biblio\allegro\katdaten
Zeile 8 set AL_DU=h:\daten\user\%LOGIN%
Zeile 9 h:
Zeile 10 cd \%AL_UD%
Zeile 11 goto ENDE
Zeile 12 :STOP
Zeile 13 set -P=
Zeile 14 set -k=
Zeile 15 set AL_DD=
Zeile 16 set AL_DU=
Zeile 17 :ENDE

Erläuterungen:

- Zeile 3** Die aufgerufene Batchdatei besteht aus zwei Abschnitten START und STOP. Welcher aktuell benutzt werden soll, wird durch den beim Aufruf angegebenen Parameter definiert, der mit %1 aufgegriffen werden kann.
- Zeile 4** Hier beginnt der Abschnitt START
- Zeile 5-8** Es werden verschiedene Environment Variablen mit Inhalten belegt. Diese werden dann später in der aufrufenden Batchdatei verwendet.
 In Zeile 8 wird der Inhalt einer Environment-Variable als Teil einer Zuweisung an eine andere Variable benutzt.
- Zeile 9-10** Es wird in das jeweilige private Nutzerverzeichnis umgeschaltet. Dadurch erfolgt der Aufruf des ALLEGRO-Programms von dort aus.
- Zeile 11** Wenn der Abschnitt START ausgeführt wird, muß der Abschnitt STOP übersprungen werden.
- Zeile 12** Der Abschnitt STOP beginnt hier. Wenn der Parameter STOP beim Aufruf übergeben wird, wird direkt von Zeile 3 hierhin gesprungen.
- Zeile 13-16** Die im Abschnitt START belegten Variablen werden wieder gelöscht.
- Zeile 17** Das ENDE der Batchdatei. Hierhin wird vom Ende des START Abschnittes (Zeile 11) gesprungen.

7.8 Erstellung eigener Menüsysteme

Aus Batchdateien lassen sich eigene Menüs erzeugen, in denen man - in einem Netzwerk sogar benutzerspezifisch - alle Routinen der täglichen Arbeit unterbringen kann. Die einzelnen Menüpunkte rufen dabei jeweils die benötigte Batchdatei auf.

Ziel eines solchen Menü-Systems sollte es sein, dem Benutzer nur das anzubieten, was er auch wirklich benötigt. Wo bindet man nun die mühsam erstellten Batchdateien ein? Eine Möglichkeit ist das Cockpit. Hier können im Menü „Routinen“ große Mengen eigener Batchdateien untergebracht und aufgerufen werden. Durch benutzerspezifische Cockpit Konfigurationsdateien können diese Aufrufe differenziert werden.

Ein weitere, gern genutzte Möglichkeit ist das kostenlose Programm **DOUGMENU**, das auf dem ALLEGRO-FTP-Server in Braunschweig zu finden ist. Dies ist ein spezielles Programm zur Erzeugung von (MS-DOS) Menüs. Es bietet die Möglichkeit Menüs beliebig zu verschachteln und es bietet den Vorzug einer variablen Parameter-Übergabe an Batchdateien und zwar in Abhängigkeit vom Ergebnis von Benutzerabfragen.

Dadurch wird ein großes Problem gelöst. Es gibt in DOS-Batchdateien keine vernünftige Möglichkeit, Benutzerwünsche abzufragen.

Wenn man nun beispielsweise dem Benutzer die Erstellung von Listen anbieten und ihm dabei die Wahl zwischen verschiedenen Sortierungen lassen will, benötigt man eine Methode, diesen Wunsch des Benutzers abzufragen und das Ergebnis der Abfrage an die Batchdatei zu übergeben. Die Alternative wäre nur, für jede denkbare Sortierung einen eigenen Menüpunkt einzurichten, bei dem dann jeweils die passenden Parameter an die Batchdatei übergeben werden.

Eine durchaus zeitgemäße Möglichkeit wäre auch, die einzelnen Batchdateien in das WINDOWS Menü einzubinden. Batchdateien können ja, wie Programme, als WINDOWS Menüpunkte eingebunden werden. Auch eine Übergabe von Parametern ist dabei möglich. Anders sieht es aus, wenn die Parameterübergabe an die Batchdateien variabel gestaltet werden soll. Hier sind weitere Hilfsmittel erforderlich, die zunächst einzelne Optionen vom Benutzer abfragen und diese dann an die Batchdatei weiterleiten, wie etwa das bereits erwähnte DOUGMENU.

Ein weiteres Problem ergibt sich aus der Tatsache, daß WINDOWS ein Multitasking-System ist. Nichts hindert den Benutzer daran, einen Import- oder Exportvorgang oder einen Katalog mehrmals zu starten. Dies führt dazu, daß einzelne Benutzer sich ihre temporären Dateien selbst überschreiben. Es ist durchaus möglich, auch dieses Problem mit Hilfe zu Batchdateien zu lösen. Leider ist die Lösung relativ komplex, so daß eine nähere Behandlung den Rahmen dieses Kurses bei weitem sprengen würde.

8 Die ALLEGRO Exportsprache

Die sogenannten Parameterdateien steuern das Verhalten der eigentlichen ALLEGRO-Programme. Wie wir bereits wissen, sind Parameter Informationen, die an ein Programm übergeben werden. Bei ALLEGRO können diese Informationen sehr komplex sein und werden deshalb in Dateien gesammelt, die von den Programmen ausgewertet werden können, da ihnen beim Programmaufruf mitgeteilt wird, welche Parameterdateien sie nutzen sollen.

Programmiert werden Export-Parameterdateien mit einer speziellen Programmiersprache, der ALLEGRO Exportsprache. Daneben gibt es noch eine ALLEGRO Importsprache, die in dieser Einführung aber nicht behandelt wird.

Die Exportsprache unterscheidet sich in ihrer Syntax sehr stark von anderen bekannten Programmiersprachen. So sind beispielsweise alle Befehle auf einen Buchstaben reduziert, was natürlich Speicherplatz spart. Die folgenden Ausführungen folgen weitgehend der von Herrn Allers in seinem ALLEGRO-Lehrbuch verwendeten Gliederung.

8.1 Grundparameter, Zwischenteile, Umkodierung von Zeichen

Zunächst noch einige grundsätzliche Anmerkungen zum Schreiben von Parameterdateien.

- ⇒ Parameterdateien müssen, wie auch Batchdateien, als reine Textdateien abgespeichert werden. Alles, was dort zur Verwendung von Editoren gesagt wurde gilt auch für Parameterdateien.
- ⇒ Nur diejenigen Eintragungen in einer Parameterdatei sind gültig, die am Zeilenanfang beginnen. Eine **Einrückung** kennzeichnet die Zeile als **Kommentar**. Es gibt also in der Exportsprache keine spezielle Kennzeichnung für Kommentare. Es ist aber trotzdem sinnvoll, für sich ein Zeichen zu definieren, das man Kommentaren vorausschickt, da dies die Übersichtlichkeit erhöht (beispielsweise das % Zeichen, wie in TEX).
- ⇒ Wenn mehrere Befehle der Exportsprache hintereinander in einer Zeile stehen, werden sie jeweils durch ein Leerzeichen getrennt. Sobald in einer solchen Befehlszeile **mehrere Leerzeichen** hintereinander stehen, gilt der Rest der Zeile als **Kommentar**.

8.1.1 Grund-Parameter (Grund-Konstanten)

Die Position der einzelnen Teile in einer Parameterdatei ist frei wählbar. Trotzdem ist es sinnvoll, die Grundparameter an den Anfang der Datei zu stellen. Grundparameter regeln Grundeinstellungen des gewünschten Exports. Die folgende Klassifizierung folgt dem ALLEGRO-Handbuch.

1. **Aufnahme-Parameter**. Diese regeln Einstellungen, die den gesamten Datensatz betreffen. Hierzu gehört beispielsweise die Definition von Zeichen, die vor und hinter einem exportierten Datensatz ausgegeben werden sollen. ⇒ Handbuch Kap. 10.2.1.
Einen Spezialfall bilden hier die sogenannten „Kopfbefehle“.
2. **Index-Parameter**. Diese Grundparameter sind speziell für den Einsatz in Indexparameterdateien geschaffen worden. ⇒ Handbuch Kap. 10.2.1.3
3. **Formular- und Seitenparameter** betreffen Einstellungen. ⇒ Handbuch Kap.

gen, die die Aufteilung der exportierten Datensätze in Listen oder Formulareinteilungen bei Katalogkarten-Produktion u.ä. regeln. 10.2.2.

4. **Zeilen-Parameter** regeln beispielsweise Zeilenlängen oder die Zeilenzahl pro Druckseite. In diesen Bereich gehören auch Maßnahmen, die in gewissem Umfang das Entstehen von „Hurenkindern“ und „Schusterjungen“ verhindern können ⇒ Handbuch Kap. 10.2.3.

5. **Drucker und Codierungsparameter.** Damit können Initialisierungsfrequenzen an den Drucker geschickt werden. Unter MS-DOS hat ALLEGRO keine Druckertreiber im herkömmlichen Sinn. Es handelt sich bei den „ALLEGRO Druckertreibern“ um eine Kombination aus Parameterdateien und Initialisierungsfrequenzen. ⇒ Handbuch Kap. 10.2.4.

6. **Kategorie-Parameter** definieren, wie Kategorien generell beim Exportvorgang behandelt werden sollen, als beispielsweise welche Zeichenfolge hinter jede Kategorie geschrieben werden soll. ⇒ Handbuch Kap. 10.2.5.

Im folgenden werden jeweils nur die wichtigsten Grundparameter jeder Gruppe näher erläutert. Ansonsten sei auf das ALLEGRO-Handbuch verwiesen.

Einführende Erläuterung: Viele Grundparameter erwarten als Wert eine **Zeichenkette**. Eine Zeichenkette kann in folgenden Formen auftreten.

⇒ Als Aneinanderreihung einzelner Zeichen, die in Anführungsstriche oder Hochkommata eingeschlossen werden.

Beispiel: *ae="Neue Aufnahme"*

⇒ Als eine Ansammlung einzelner Zahlen, die dem ASCII-Wert des Zeichens entsprechen. Über den ASCII-Wert einzelner Zeichen kann man sich in MS-DOS Handbüchern u.ä. informieren. Die einzelnen Zahlen sind dabei jeweils durch ein Leerzeichen getrennt und stehen nicht in Anführungszeichen.

Beispiel: *ae=78 101 117 101 32 65 117 102 110 97 104 109 101*

⇒ Als Mischung aus Buchstaben und ASCII-Codes. Die Buchstaben sind in Anführungszeichen oder Hochkommata eingeschlossen, die ASCII-Codes nicht.

Beispiel: *ae="Neu" 101 32 „Aufn“ 97 „hm“ 101*

⇒ Außerdem können noch sogenannte **Strukturbefehle** in Zeichenketten integriert werden. Die wichtigsten sind **C** (neue Zeile) und **N** (neue Seite). Sie werden nicht in Anführungszeichen eingeschlossen, sondern wie ASCII-Codes behandelt.

Beispiel: *ae=N (gibt jeden Datensatz auf einer neuen Seite aus).*

Aufnahme-Grundparameter:

ab=<Zeichenkette> Die definierte Zeichenkette wird beim Export vor jeden neuen Datensatz geschrieben. Als Standard wird nichts ausgegeben.

ae=<Zeichenkette> Die definierte Zeichenkette wird beim Export hinter jeden

ausgegeben Datensatz geschrieben. Als Standard wird nichts ausgegeben.

ak=Kategorie
["Trennmuster"]+
Sprungmarke

Der **wichtigste** Ausgabeparameter. Durch Kopfbefehle können aus einem Datensatz mehrere Exportsätze erzeugt werden. Ob diese Exportsätze gebildet werden, hängt davon ab, ob die angegebene Kategorie im Datensatz vorkommt.

Dies ist z.B. bei der **Indexparametrierung** wichtig. Aus einem Datensatz werden ja in der Regel viele Indexeinträge gebildet, von denen jeder logisch einen eigenen Exportsatz darstellt.

Für jede Kategorie können **Trennmuster** definiert werden, an denen der Inhalt der Kategorie in einzelne Teile aufgespalten wird. Für jeden dieser Teile wird nun ein Exportsatz gebildet.

Die **Sprungmarke** gibt an, an welcher Stelle der Parameterdatei die Bearbeitung des Datensatzes für den durch diesen Kopfbefehl angestoßenen Exportvorgang beginnt.

Der Kopfbefehl **ak=zz+[Sprungmarke]** bewirkt, daß der Exportsatz, der daraus resultiert, auf jeden Fall gebildet wird, unabhängig von den jeweils vorliegenden Kategorien.

Beispiel: ak=320[".'-"]+A

Hier wird der Inhalt der Kategorie 320 (Sachtitel) bei jedem der im Trennmuster definierten Zeichen getrennt. Es entstehen also Stichwörter. Für jedes Stichwort wird ein Exportdatensatz erzeugt. Die Bearbeitung des Exportdatensatzes beginnt bei Sprungmarke A.

ad=[0|1]
Standard = 1

Legt fest, ob eine Aufnahme ausgegeben werden soll, wenn im Datensatz die in den Kopfbefehlen angegebenen Kategorien nicht vorkommen. In der Standardeinstellung wird in diesem Fall ein Exportsatz erzeugt.

Wenn es keine Kopfbefehle gibt, wird immer genau ein Exportsatz pro Datensatz erzeugt.

Index-Parameter:

il=[Zahl]
Maximum = 249
Standard=60

Gibt die maximale Länge an, die ein einzelner Schlüssel im Index haben kann. Bei einer Länge von 72 wird die Bildschirmbreite genau ausgenutzt. Wenn man allerdings z.B. ganze Körperschaftsnamen als Indexeintrag haben will, sollte man einen wesentlich höheren Wert wählen.

Achtung: Im Handbuch wird als maximale Länge 250 angegeben. Damit funktionieren aber die neuen WINDOWS-Programme nicht.

Dieser Parameter sollte in **jeder** Exportparameterdatei, die in irgendeiner Form auf eine Datenbank zugreift korrekt, also mit

dem in der Indexparameterdatei verwendeten Wert besetzt sein.

i0=[Zahl]
Maximum:
72 bis Version 15e
255 ab Version 16

Gibt die Länge der Kurzanzeige einzelner Titel im Index an. Nur wenn hier ein Wert, der größer als Null ist steht, wird überhaupt eine solche Kurzanzeige erzeugt.

Der zugehörige Kopfbefehl lautet **ak=zz+0**. Die **Sprungmarke 0** ist in Indexparameterdateien für diese Zwecke reserviert.

I KÜRZEL
Registerkennung
["Registername"]

Die neuen Windows-Programme sowie der AVANTI-Server, mit dem ALLEGRO-Datenbanken im Internet angeboten werden können, benötigen die Definition von symbolischen Namen für einzelne Register, die mit den neu geschaffenen **find** Befehlen durchsucht werden sollen.

Beispiele: I PER 1 „Personen“ oder I STW 3 „Stichwörter“

ic=1
Standard ist 0

Wenn dieser Parameter auf **1** gesetzt ist, werden die Anfragen, die der Benutzer in die Suchzeile tippt, umkodiert, bevor sie bearbeitet werden.

Wie sie umkodiert werden wird in der Indexparameterdatei unterhalb der **Sprungmarken 1 bis 9** geregelt. Die Zahlen korrespondieren dabei mit der Zählung der Register.

Formular- und Seitenparameter:

fl=[Zahl]

Die Formularlänge. Hauptsächlich ist sie für die Produktion von Katalogkarten nötig. Hier muß man dann in der Regel mit den einzelnen Formularparametern experimentieren, bis alles stimmt.

Wichtig ist: Falls man eine Liste und keine Karten erzeugen will, muß man **fl=0** setzen.

fm=[0 | 1 | 2]

Der Formularmodus. Ebenfalls für Kartendruck interessant. Legt fest, ob für jede - nach alter Väter Sitte hierarchisch erfaßte - Bandaufführung eine eigene Katalogkarte erzeugt wird. Bei **Listendruck** gilt **fm=0**.

sz=[Zahl]

Legt die erste Seitenzahl beim Listendruck fest.

Zeilen-Parameter:

zl=[Zahl]
Maximum: 150
Standard: 56

Nach maximal der angegebenen Zahl von Zeichen wird ein Zeilenumbruch durchgeführt. Dies ist ein Wert, der evtl. verändert werden muß, wenn Zeilen nicht ganz auf dem verwendeten Drucker ausgegeben werden.

Eine Ausgabe ohne Zeilenumbruch erreicht man mit **zl=0**. Dies ist sinnvoll, wenn das Exportergebnis in einem Textverarbeitungs-Programm weiterverarbeitet werden soll.

ze=<Zeichenkette>
Standard: ze=13 10

Die definierte Zeichenfolge wird hinter jede ausgegebene Zeile gesetzt. Dies geschieht nur bei der Druckausgabe nicht bei einer Ausgabe auf dem Bildschirm.

ner Ausgabe auf dem Bildschirm.

Kategorie-Parameter:

ke=<Zeichenkette> Die definierte Zeichenfolge wird **hinter** jeder Kategorie ausgegeben, wenn nicht im Hauptteil der Parameterdatei für eine Kategorie eine andere Zeichenfolge als sogenanntes **Postfix** (also etwas, das hinter eine Kategorie geschrieben wird) definiert wird.

Standard ist leer

Wichtig ist die Angabe **ke=0**, wenn man eine Exportdatei im ALLEGRO-Grundformat erzeugen will. Hier muß ja jede Kategorie mit dem ASCII-Code 0 abgeschlossen werden. Dies erreicht man am einfachsten mit dieser Angabe.

Unterdrückt werden kann die Ausgabe der in ke definierten Zeichenfolge, in dem man im Hauptteil der Parameterdatei am Ende einer Befehlszeile, die eine Kategorie bearbeitet den Befehl **#zz 0** anfügt. Mehr dazu später.

ks=[4 | 5 | 6]
Standard: Definition in der CFG-Datei.

Der angegebene Wert legt fest, ab welcher Position der Kategorieinhalt bei der Bearbeitung der Kategorie durch die Befehlszeilen im Hauptteil der Parameterdatei verwendet wird.

Die Kategoriennummern und das in der Regel folgende Leerzeichen sind ja Bestandteil des Kategorieinhaltes. Sie belegen innerhalb einer Kategorie die Positionen 1-3 oder 1-4 (je nach Kategorienschema). Diese Kategoriennummern wird man bei den meisten Exporten nicht mit ausgeben wollen. Hier kann man die Standardeinstellung beibehalten.

Anders sieht dies bei Exporten in das **ALLEGRO-Grundformat** zur späteren Weiterbearbeitung aus. Hier muß man die Kategoriennummern mit ausgeben, da sonst die Inhalte der Kategorien nicht mehr zu zuordnen und somit wertlos sind. In diesem Fall muß man **ks=1** setzen.

8.1.2 Zwischenteile

Zwischenteile sind Zeichenketten, die an einer zentralen Stelle der Parameterdatei definiert werden. Die Definition von Zwischenteilen kann auch in eine Hilfsparameterdatei ausgelagert werden. Diese kann dann von verschiedenen Parameterdateien gemeinsam genutzt werden. Bei der Definition von Zwischenteilen gelten die beschriebenen Regeln für Zeichenketten. Es können insgesamt 99 Zwischenteile definiert werden, die jeweils einer Zahl zwischen 1 und 99 zugewiesen werden.

Beispiele: 1=13 10 "In: "
2="Fussnote: " usw.

Im Hauptteil der Parameterdatei, in der die eigentliche Verarbeitung der Datensätze stattfindet, können diese Zwischenteile dann als **Präfixe** oder **Postfixe** aufgerufen werden. Was versteht man nun unter diesen Begriffen?

Präfixe und Postfixe sind Zeichenketten, die bei der Verarbeitung vor oder hinter einer verarbeiteten Kategorie ausgegeben werden. Unmittelbar deutlich wird dies, wenn man an eine RAK-Titelaufnahme denkt. Die Zeichenketten, die die einzelnen Blöcke der Titelaufnahme trennen werden ja nicht mit eingegeben, sondern müssen durch die Parameterdatei erzeugt werden. Dies kann entweder durch eine Angabe der Zeichenkette an Ort und Stelle oder durch den Aufruf eines vor definierten Zwischenteils geschehen. Letzteres spart Schreiarbeit, macht aber die Parameterdatei unübersichtlicher, weil man sich zunächst über den Inhalt des aufgerufenen Zwischenteils Klarheit verschaffen muß.

8.1.3 Kodiertabellen und Ersetzungsbefehle

Während der Verarbeitung von Datensätzen in Parameterdateien können einzelne Zeichen oder auch Zeichenfolgen durch andere Zeichen oder Zeichenfolgen ersetzt werden. Dies geschieht bei einzelnen Zeichen an Hand von Befehlen, die in den sogenannten Kodiertabellen definiert werden. Die Kodiertabellen werden sinnvollerweise in Hilfsparameterdateien ausgelagert und können so von mehreren Parameterdateien gemeinsam genutzt werden.

Es gibt zwei Tabellen, die **Normaltabelle** und die **Alternativtabelle**. Hier können jeweils völlig unterschiedliche oder auch gar keine Ersetzungen definiert werden. In der Alternativtabelle werden normalerweise die Ersetzungen definiert, die vom gewünschten Standard abweichen. Bei jeder Kategorie, die im Hauptteil der Parameterdatei verarbeitet wird muß angegeben werden, welche der beiden Tabellen verwendet werden soll.

Der **Kodierbefehl** für die Normaltabelle ist **p**, der für die Alternativtabelle ist **q**. Dahinter folgt durch Leerzeichen getrennt das zu ersetzende Zeichen und nach einem weiteren Leerzeichen das Zeichen oder die Zeichenfolge die das Ergebnis der Ersetzung bildet.

- | | |
|--------------------|--|
| p a A | Ersetzungsbefehl für die Normaltabelle . a -> A
⇒ Einzelzeichen als Ziel der Ersetzung stehen nicht in Anführungszeichen. |
| p ä "ae" | Ersetzungsbefehl für die Normaltabelle ä -> ae
⇒ Zeichenketten als Ziel der Ersetzung stehen in Anführungszeichen. |
| p a 65 | Ersetzungsbefehl Normaltabelle a -> A (Ascii Code 65)
⇒ Einzelne ASCII-Codes als Ziel der Ersetzung stehen nicht in Anführungszeichen. |
| q ä .97 101 | Ersetzungsbefehl für die Alternativtabelle ä -> ae (Ascii Code 97 und 101)
⇒ Eine Folge von ASCII-Codes als Ziel der Ersetzung beginnt mit einem Punkt vor der Codesequenz. |
| q _ 1 | Ersetzungsbefehl für die Alternativtabelle . Der Unterstrich wird ent- |

fernt.

⇒ Durch Angabe des **ASCII-Code 1** als Ziel der Ersetzung wird ein Zeichen aus dem Exportergebnis entfernt.

q * 255 Ersetzungsbefehl für die **Alternativtabelle**.

⇒ Eine Ersetzung eines Zeichens durch den **ASCII-Code 255** führt dazu, daß in der Ausgabe an dieser Stelle ein Zeilenumbruch durchgeführt wird.

p !. 1 Ersetzungsbefehl für die **Normaltabelle**.

⇒ Hier werden **mehrere einzelne Zeichen**, die aufeinander folgende ASCII-Codes haben durch einen Befehl aus der Ausgabe entfernt.

Wozu dienen die Kodierbefehle und warum gibt es zwei Tabellen? Durch die Kodierbefehle kann die erstellte Ausgabe flexibel auf andere Zeichensätze angepaßt werden. So können die WINDOWS und die DOS Programme von ALLEGRO eine gemeinsame Datenbank benutzen, obwohl WINDOWS einen anderen Zeichensatz benutzt als DOS. Ein anderes Beispiel ist die Auflösung von Umlauten im Datenbank-Index oder wenn Datensätze sortiert werden sollen.

Zwei Tabellen existieren, um die Umkodierung in Parameterdateien flexibler zu machen. So müssen in einer Parameterdatei, die Daten für eine Sortierung aufbereitet, nur bestimmte Kategorien sortierfähig aufbereitet werden. Beim Großteil der exportierten Kategorien ist dies nicht erwünscht.

Neben den Befehlen zur Umkodierung von Zeichen existieren noch sogenannte **Ersetzungsbefehle**. Sie werden benutzt um eine Zeichenfolge durch eine andere zu ersetzen. Im Gegensatz zu den Umkodierbefehlen, die erst dann wirksam werden, wenn im Hauptteil der Parameterdatei Kategorien zur Bearbeitung aufgerufen werden, werden die Ersetzungsbefehle ausgeführt, bevor die Parameterdatei interpretiert wird.

Ersetzungsbefehle werden wie folgt definiert. Jeder Ersetzungsbefehl steht dabei auf einer eigenen Zeile.

[Ersetzte Zeichenkette][Ergebnis der Ersetzung]_

In einem Ersetzungsbefehl kommen also **immer drei Unterstriche** vor. Neben dieser sogenannten globalen Ersetzung (global weil sie auf den gesamten Datensatz wirkt) kann der selbe Befehl auch im Rahmen einer Befehlszeile auf eine Kategorie angewandt werden. Man spricht dann von einer lokalen Ersetzung.

8.2 Die Kategorienliste - der Anweisungsteil der Parameterdatei

Die Kategorienliste bildet den Hauptteil der Parameterdatei. Hier stehen die Anweisungen, durch die aus dem zu bearbeitenden Datensatz ein greifbares Ergebnis erzeugt wird. Die anderen Teile der Parameterdateien regeln die Rahmenbedingungen für die Befehle des Hauptteils.

8.2.1 Aufbau und Interpretation einer Befehlszeile

- ⇒ Die Anweisungszeilen werden **sequentiell** also von oben nach unten abgearbeitet. Wenn man diese Reihenfolge verändern will, muß man Sprungbefehle verwenden.
- ⇒ **Startpunkt** ist die erste Zeile der Kategorienliste. Falls **Kopfbefehle** existieren, definiert die dort angegebene Sprungmarke den Startpunkt für den Exportvorgang, der durch diesen Kopfbefehl ausgelöst wird. Die beim Kopfbefehl angegebene Sprungmarke muß existieren, da sonst keine Verarbeitung stattfindet.
- ⇒ An der ersten Position jeder Befehlszeile steht entweder das Zeichen # oder das Zeichen !. Durch diese Zeichen wird festgelegt, welche Kodiertabelle in dieser Befehlszeile verwendet wird. Das **Lattenkreuz** führt zur Benutzung der **Normaltabelle**, das **Ausrufezeichen** zur Benutzung der **Alternativtabelle**.
- ⇒ Unmittelbar nach diesem führenden Zeichen folgt die Angabe, was in dieser Zeile verarbeitet werden soll. In der Regel handelt es sich hier um die Angabe einer **Kategoriennummer**. Alternativ können auch **Sonderkategorien** oder **Anwendervariablen** verarbeitet werden.
- ⇒ Die Befehlszeile wird nur ausgeführt, wenn die angegebene Kategorie oder Anwendervariable existiert.
- ⇒ Innerhalb einer Parameterdatei kann eine Kategorie beliebig oft aufgerufen werden.
- ⇒ Nach der Kategoriennummer kann ein **Sprungbefehl** folgen. Dieser wird nur dann ausgeführt, wenn die gesamte Befehlszeile erfolgreich bearbeitet werden konnte. Dadurch wird es möglich, Kontrollstrukturen in einer Parameterdatei nach zu bilden.
- ⇒ Nach einem eventuellen Sprungbefehl folgen dann die eigentlichen **Manipulationsbefehle**. Wie der Name bereits sagt verändern sie den Inhalt der Kategorie. Dies allerdings nicht im in der Ausgangsdatei vorliegenden Original, sondern in einer Kopie im Arbeitsspeicher, dem sogenannten **Arbeitstext**.

8.2.1.1 Das Konzept des Arbeitstextes

In einer Befehlszeile können eine ganze Reihe von Manipulationsbefehlen stehen. Sie werden jeweils durch genau ein Leerzeichen voneinander getrennt. Auch der Sprungbefehl wird von der angegebenen Kategoriennummer durch genau ein Leerzeichen getrennt. Mehrere Leerzeichen führen dazu, daß der Rest der Zeile als Kommentar betrachtet wird.

Die **Befehlszeile** wird nun von **links nach rechts** abgearbeitet. Auf den unveränderten Kategorieninhalt wird der erste Manipulationsbefehl angewendet. Das Ergebnis dieses Befehls wird nun zum Arbeitstext. Auf diesen so entstandenen Arbeitstext wird der zweite Manipulationsbefehl angewendet. Als Ergebnis entsteht ein neuer Arbeitstext, auf den dann der nächste Manipulationsbefehl angewendet wird, usw. usw.

Präfixe und **Postfixe** stehen in der Regel am Ende der Befehlszeile. Es handelt sich hier um Zeichenketten, die vor bzw. hinter dem Arbeitstext ausgegeben werden, der nach dem letzten Manipulationsbefehl entstanden ist.

Ein Beispiel:

Der **Inhalt von #320 (Hauptsachtitel)** sei
„ALLEGRO lernen in drei Stunden - kein Problem ?“

Dieser Sachtitel wird nun mit folgender Befehlszeile bearbeitet:

#320 b6 b" " b" - " b1 e"?" p{ "ALLEGRO lernen" }

Befehl	Erklärung	Aktueller Arbeitstext
b6	Beginne nach dem 6. Zeichen	<i>O lernen in drei Stunden - kein Problem ?</i>
b" "	Beginne nach dem nächsten Leerzeichen	<i>lernen in drei Stunden - kein Problem ?</i>
b" - "	Beginne nach der Zeichenkombination -	<i>kein Problem ?</i>
b1	Beginne nach dem ersten Zeichen des aktuellen Arbeitstextes	<i>ein Problem ?</i>
e"?"	Ende vor Fragezeichen	<i>ein Problem</i>
p{ "ALLEGRO lernen" }	Vor das Ergebnis wird die definierte Zeichenkette gesetzt.	<i>ALLEGRO lernen ein Problem</i>

Dieses Ergebnis wird als Resultat der Befehlszeile ausgegeben. Der ursprüngliche Inhalt wurde also erheblich modifiziert. Nun wird die nächste Befehlszeile aufgerufen. Der Inhalt der dort angegebenen Kategorie wird zum Arbeitstext.

8.2.1.2 Sprungbefehle

Sprungbefehle dienen dazu, die normale Reihenfolge der Bearbeitung in der Kategorienliste zu verändern. Man kann dadurch Kontrollstrukturen nachbilden, die man aus anderen Programmiersprachen kennt. Diese Sprungbefehle sind die einzige Möglichkeit, die die ALLEGRO Exportsprache dafür bietet. Sprungbefehle folgen **immer als erste Angabe** auf die Kategoriennummer.

Es gibt zwei Formen von Sprungbefehlen. Die erste Form hat als Ziel eine Sprungmarke, die zweite Form eine Kategoriennummer. Die Sprungmarke wird immer vom Anfang der Kategorienliste aus gesucht, die anzuspringende Kategorie nur von der aktuellen Position aus nach unten.

Mit Sprungmarken sind also auch Rücksprünge möglich. Dies ist sehr nützlich, um zum Beispiel Schleifen zu parametrisieren. Eine Sprungmarke kann nur aus einem Zeichen bestehen, also ist ihre Zahl endlich. Man sollte deshalb, soweit möglich, die Form der auf Kategoriennummern zielenden Sprünge wählen. Dadurch wird außerdem die Interpretation der Parameterdatei durch die Programme beschleunigt.

⇒ Das **Ziel** des Sprungbefehls **muß existieren**, sonst ist in der Bearbeitung der Parameterdatei an dieser Stelle Schluß. Dies macht man sich zu nutze, indem man ein Zeichen definiert, das nicht als Sprungmarke existiert. Wenn man nun diesen

Sprungbefehl angibt, wird die Bearbeitung der Parameterdatei an dieser Stelle beendet. In der Regel wird dafür das Lattenkreuz (#) verwendet.

Ein Beispiel:

In **#412** steht ein Verleger, in **#416** steht ein Drucker, in **#425** das Erscheinungsjahr. Der Drucker soll nun nur ausgegeben werden, wenn es keinen Verleger gibt. Die nächste auszugebende Kategorie ist dann das Erscheinungsjahr.

- #412 +A** ⇒ Kategorie #412 wird aufgerufen, wenn sie existiert. Nur dann wird die Befehlszeile ausgeführt.
 - ⇒ Eine Bearbeitung mit Manipulationsbefehlen findet nicht statt.
 - ⇒ Nach der Ausgabe der Kategorie wird zur Sprungmarke #-A gesprungen
- #416** ⇒ Kategorie #416 wird aufgerufen, wenn sie existiert, und ohne Veränderung durch Manipulationsbefehle ausgegeben.
- #-A** ⇒ Sprungmarke #-A wird bei der Existenz von #412 angesprungen. Die Ausgabe von #416 wird also übersprungen.
- #425 +# p", "** ⇒ Das Erscheinungsjahr wird ausgegeben. Als Präfix wird ein Komma davor gesetzt.
 - ⇒ Wenn #425 existiert wird die Bearbeitung der Parameterdatei beendet. (Ausprungmarke #)

Mit einer Kategoriennummer als Ziel des Sprungbefehls sieht das Beispiel so aus:

```
#412 +#425  
#416  
#425 +# p", "
```

Bei diesen Sprungbefehlen handelt es sich um sogenannte „**bedingte Sprungbefehle**“. Bedingt sind sie, weil sie nur ausgeführt werden können, wenn die am Anfang der Befehlszeile angegebene Kategorie oder Anwendervariable existiert. Daneben gibt es noch „**unbedingte Sprungbefehle**“. Diese werden immer ausgeführt, wenn die entsprechende Stelle der Parameterdatei erreicht wird.

Ein Beispiel für einen unbedingten Sprungbefehl.

```
#410  
#+A  
#412 e10  
...  
...  
 #-A
```

Durch den Sprungbefehl in der zweiten Zeile wird immer zur Sprungmarke **#-A** gesprungen. Unbedingte Sprungbefehle können als Ziel nur eine Sprungmarke haben

und keine Kategoriennummer. Der Sprungbefehl **##425** ist also falsch und würde nicht funktionieren.

8.2.1.3 Prä- und Postfixe

Präfixe und **Postfixe** sind allgemein gesagt Zeichenketten, die **vor** bzw. **hinter** eine fertig aufbereitete Kategorie gesetzt werden. Diese Zeichenketten werden nur dann ausgegeben, wenn die entsprechende Kategorie auch existiert und die gesamte Befehlszeile abgearbeitet werden konnte. Der Befehl für das Präfix ist **p**, der für das Postfix **P**. Bei den Prä- und Postfixen unterscheidet man zwischen

⇒ **direkten** Prä- und Postfixen und

⇒ **indirekten** Prä- und Postfixen.

Direkte Prä- und Postfixe sind der einfachere Fall. Diese werden in der Kette von Befehlen einer Befehlszeile genau an der Stelle abgearbeitet, an der sie stehen. Mit folgender Konstruktion ist es z.B. möglich in jeden Datensatz einer ALLEGRO-Grunddatei eine Kategorie mit einem festen Inhalt einzufügen.

⇒ *Zum Verständnis: Die Kategorie **#dt** ist eine sogenannte Sonderkategorie, die man immer nutzen kann. Sie enthält das aktuelle Tagesdatum.*

#dt b4 p“Kategorie mit festem Inhalt“ e“*“ p“513 ”

Zur Erläuterung:

#dt Aufruf der Sonderkategorie **#dt** = aktuelles Tagesdatum.

b4 Bei einer Ausgabe in das ALLEGRO-Grundformat wird der **Grundparameter ks** generell auf 1 gesetzt. Die Kategoriennummer wird also mitgenommen.

Diese Angabe wirkt auch auf Sonderkategorien und Anwendervariablen. Durch den Befehl wird hinter der Kategoriennummer beim eigentlichen Kategorieinhalt begonnen.

p“Kategorie ...*“ Ein direktes Präfix. Der Inhalt des Präfixes wird also im Arbeitstext vor den eigentlichen Kategorieinhalt gesetzt.

e“*“ Ende vor dem Zeichen *. Der Arbeitstext enthält jetzt also nur noch den Inhalt des direkten Präfixes von eben.

p“513 ” Vor den so erzeugten Kategorieinhalt wird nun mittels eines weiteren direkten Präfixes eine Kategoriennummer gesetzt.

Grundregeln für direkte Prä- und Postfixe:

⇒ Ihr gewünschter Inhalt folgt in Anführungszeichen eingeschlossen direkt auf den **p** (Präfix) bzw. **P** (Postfix) Befehl.

⇒ Inhalt kann nur eine Kette von Zeichen sein, keine Mischung aus Zeichen und ASCII Codes

⇒ Sie unterliegen genau wie der eigentliche Inhalt der bearbeiteten Kategorie der Umkodierung durch evtl. definierte Kodierbefehle.

⇒ Sie werden an der Stelle in der Befehlskette, an der sie definiert werden, vor oder hinter den jeweils aktuellen Arbeitstext gesetzt.

Für die **indirekten Prä- und Postfixe** gelten andere Regeln. Identisch sind die Befehle durch die sie eingefügt werden (p bzw. P).

⇒ Indirekte Prä- und Postfixe werden vor bzw. hinter das **fertige Ergebnis** einer Befehlszeile gesetzt. Es werden also erst alle anderen Manipulationsbefehle inklusive der direkten Prä- und Postfixe ausgeführt.

⇒ Sie werden **nicht** über etwaig definierte Umkodierbefehle **umkodiert**. Wenn in einem Datenbankindex hinter einer Zeile, bei der der Kategorieninhalt in Kleinbuchstaben umkodiert wird, ein Nachspann in Großbuchstaben stehen soll, muß dieser als indirektes Postfix definiert werden.

⇒ Sie können als Inhalt genau wie Zwischenteile auch **gemischte Angaben** aus Zeichenketten, ASCII-Codes und Steuerbefehlen enthalten.

⇒ Bei der Definition des Inhalts kann auf vor definierte **Zwischenteile** zurück gegriffen werden. In diesem Fall wird hinter dem p oder P Befehl nur die Nummer des Zwischenteils angegeben.

⇒ Der Inhalt kann aber auch an der Position in der Parameterdatei definiert werden, an der er benötigt wird. In diesem Fall wird der Inhalt in **{ }** eingeschlossen und hinter dem führenden Befehl angegeben. Dies macht mehr Schreiarbeit, fördert aber die Übersichtlichkeit der Parameterdatei.

Beispiele für indirekte Prä- und Postfixe:

p1	Indirektes Präfix. Der Inhalt von Zwischenteil 1 wird ausgegeben.
P13	Indirektes Postfix. Der Inhalt von Zwischenteil 13 wird ausgegeben.
P{ t1 t13 }	Mehrere vorher definierte Zwischenteile werden zu einem Postfix zusammengefaßt. Die Definition geschieht in { } . Vor jedem benutzten Zwischenteil steht der Befehl t .
p{ "Ein Text" }	Ein ganz normaler Text wird als indirektes und nicht als direktes Präfix definiert. Den Unterschied machen die geschweiften Klammern.
P{ 13 10 "Aufsatz" 13 10 }	Ein indirektes Postfix, bei dem Angaben aus Zeichen und ASCII-Codes gemischt werden. Es gelten die Regeln, die bereits für Zwischenteile beschrieben wurden.

8.2.1.4 Spezialfälle: Bedingtes Postfix und Mehrfachpräfixe

Bedingte Postfixe sind Konstruktion, bei denen in Abhängigkeit von der nächsten im Datensatz vorkommenden Kategorie festgelegt wird, welches indirekte Postfix (in der Regel ein definiertes Zwischenteil) ausgegeben werden soll. Im Prinzip kann man diese Konstruktion durch entsprechende Präfixe vor den folgenden Kategorien erset-

zen, was die Übersichtlichkeit und Verständlichkeit einer Parameterdatei erheblich erhöht. Da aber in vielen Parameterdateien Konstruktionen wie

#410 e10 #425 5 #433 14 #505 15 #zz 23

stehen, sollte man in etwa wissen, was es damit auf sich hat. Die Zeile bedeutet in folgendes:

*Nimm Kategorie 410 und ende nach 10 Zeichen. Falls die nächste Kategoriennummer zwischen **411 und 425** liegt, gib Zwischenteil **5** aus, falls sie zwischen **426 und 433** liegt gib Zwischenteil **14** aus, von **434 bis 505** gib Zwischenteil **15** aus und **in allen anderen Fällen** Zwischenteil **23**.*

Ein **wichtiger Spezialfall eines bedingten Postfixes** ist folgender.

#410 e20 #zz 0

⇒ Durch die Angabe **#zz 0** hinter einer Befehlszeile wird verhindert, daß ein Kategorie-Endezeichen, das im Grundparameter **ke** definiert wurde ausgegeben wird.

Normalerweise gibt es kein definiertes Kategorie-Endezeichen. Bei einer Ausgabe ins ALLEGRO-Grundformat wird man in der Regel aber in **ke** den ASCII-Code 0 als Endezeichen definieren, weil ja jede ausgegebene Kategorie mit diesem Zeichen enden muß. Wenn nun zum Beispiel mehrere Kategorien zu einer zusammen gefaßt werden sollen, darf die Ausgabe des Kategorie-Endezeichens nur bei der letzten dieser Kategorien erfolgen und muß bei den anderen unterdrückt werden.

Mehrfach-Präfixe werden mit dem Befehl **m** definiert. In ALLEGRO-Parameterdateien gibt es die Möglichkeit ganze Kategoriengruppen, die mit den gleichen Zahlen beginnen (also z.B. 201, 202, 203, 204 usw.), mit einer Befehlszeile auszugeben. Dazu dient der Befehl **++**, der diese Wiederholung veranlaßt.

#20. ++

Diese Befehlszeile gibt alle im Datensatz vorkommenden Kategorien aus, die mit 20 beginnen. Der Punkt ist ein Platzhalter für das dritte Zeichen der Kategoriebenennung. Wenn nun in dieser Befehlszeile ein **Präfix** mit dem Befehl **p** definiert wird, wird es **nur vor der ersten Kategorie** ausgegeben. Die **Präfixe für alle anderen Kategorien** müssen als Mehrfach-Präfix mit dem Befehl **m** definiert werden. Das Postfix dagegen wird bei allen Kategorien ausgegeben Etwa so:

#20. p{ "Autor: " } m{ "Autor: " } P{ 13 10 }

8.2.1.5 Sonderkategorien

Sonderkategorien sind Kategorien, auf die bei der Erstellung einer Parameterdatei zurückgegriffen werden kann, obwohl sie im bearbeiteten Datensatz nicht existieren. Sie können genau wie richtige Kategorien mit Manipulationsbefehlen bearbeitet und ausgegeben werden. Im ALLEGRO-Handbuch sind sie in Kapitel 10.2.6.2 in toto beschrieben. Hier nur die wichtigsten:

#cc ⇒ Enthält den unveränderten Inhalt der zuletzt aufgerufenen Kategorie.

#u1 ⇒ Das jeweils aktuell verarbeitete Resultat eines Kopfbefehls. Bei-

spielsweise enthält die Kategorie bei einem Kopfbefehl, bei dem ein Sachtitel in Stichwörter aufgeteilt wird, das einzelne Stichwort. Dadurch kann man bei der Ausgabe gezielt auf diesen Teil einer Kategorie zugreifen.

⇒ Falls keine Kopfbefehle in einer Parameterdatei existieren, kann man diese Kategorie wie eine normale Kategorie exportieren. Sie wird dann im entstandenen Datensatz vor allen anderen Kategorien einsortiert. In der Praxis benutzt man dies um mit Hilfe von Parameterdateien einen Sortierkopf zu bilden.

#ch ⇒ Enthält den gesamten Inhalt einer in einem Kopfbefehl benutzten Kategorie inklusive der Kategorienummer. #u1 enthält dagegen nur den jeweiligen Teil der durch eine Trennvorschrift im Kopfbefehl entstanden ist.

#dt ⇒ Das aktuelle Tagesdatum in deutscher Form

#dts ⇒ Das aktuelle Tagesdatum in Sortierform (z.B. 20000101/00:00:01)

#nr ⇒ Die **interne Datensatznummer** des Datensatzes in der ALLEGRO-Datenbank, aus der exportiert wird.

⇒ Wenn eine **ALLEGRO-Grunddatei** weiter bearbeitet wird, werden hier alle Datensätze in einer **laufenden Nummer** durchnummeriert, die in der Regel mit 1 oder aber der Zahl, die im **Grundparameter an** angegeben wurde beginnt.

8.3 Die wichtigsten Befehle der ALLEGRO-Exportsprache

In der ALLEGRO-Exportsprache kann man zwischen Manipulations- und Vergleichsbefehlen unterscheiden. Manipulationsbefehle verändern den jeweiligen Arbeitstext, auf den sie angewendet werden. Vergleichsbefehle dagegen prüfen im jeweils aktuellen Arbeitstext bestimmte Sachverhalte. Je nach Ergebnis der Prüfung wird dann zu unterschiedlichen Sprungmarken verzweigt.

8.3.1 Manipulationsbefehle

Befehl	Beispiel	Erläuterung
b Zahl	b10	Beginne den Arbeitstext hinter der angegebenen Position in der Kategorie. Wenn die Kategorie kürzer ist, ist der Arbeitstext leer und es erfolgt keine Ausgabe
b"Zeichen(kette)"	b"ALLEGRO"	Beginne den Arbeitstext hinter der angegebenen Zeichen(kette). Wenn die Zeichen(kette) nicht vorkommt wird die Befehlszeile beendet und es wird nichts ausgegeben.
B"Zeichen(kette)"	B"ALLEGRO"	Beginne den Arbeitstext hinter der angegebenen Zeichen(kette). Wenn die Zeichen(kette) nicht

		vorkommt bleibt der Arbeitstext unverändert und es wird der nächste Befehl ausgeführt.
e Zahl	e32	Beende den aktuellen Arbeitstext nach der durch die Zahl festgelegten Anzahl von Zeichen.
E Zahl	E32	Beende den aktuellen Arbeitstext nach der durch die Zahl festgelegten Anzahl von Zeichen. Das letzte Wort wird dabei weggelassen, wenn es zerschnitten würde.
e"Zeichen(kette)"	e"ALLEGRO"	Beende den Arbeitstext vor der angegebenen Zeichenkette. Wenn die Zeichenkette nicht vorkommt, bleibt der Arbeitstext unverändert.
E"Zeichen(kette)"	E"ALLEGRO"	Beende den Arbeitstext hinter der angegebenen Zeichenkette. Wenn die Zeichenkette nicht vorkommt, bleibt der Arbeitstext unverändert.
f Zahl	f32	Entferne das Zeichen mit dem durch die Zahl angegebenen ASCII-Code am Anfang des Arbeitstextes. Dies geschieht auch wenn das Zeichen mehrmals vorkommt.
F Zahl	F32	Entferne das Zeichen mit dem durch die Zahl angegebenen ASCII-Code am Ende des Arbeitstextes. Dies geschieht auch wenn das Zeichen mehrmals vorkommt.
f"Zeichen"	f" "	Entferne das angegebene Zeichen am Anfang des Arbeitstextes. Dies geschieht auch wenn das Zeichen mehrmals vorkommt.
F"Zeichen"	F" "	Entferne das angegebene Zeichen am Ende des Arbeitstextes. Dies geschieht auch wenn das Zeichen mehrmals vorkommt.
R	R	Der ausgegebene Arbeitstext wird rechtsbündig auf der Zeile positioniert.
r Zahl,Zeichen	r6,0	Die Zahl gibt an auf wie viele Stellen der Arbeitstext rechtsbündig aufgefüllt werden soll. Dies geschieht mit dem angegebenen Zeichen.
s Zahl,Zeichen	s6,0	Die Zahl gibt an auf wie viele Stellen der Arbeitstext linksbündig aufgefüllt werden soll. Dies geschieht mit dem angegebenen Zeichen.
T Zahl	T13	Die letzten Zeichen des aktuellen Arbeitstextes werden zum neuen Arbeitstext. Die Zahl gibt an wie viele Zeichen für den neuen Arbeitstext benutzt werden.
T"Zeichen(kette)"	T"ALLEGRO"	Das Ende des aktuellen Arbeitstextes ab dem Beginn der angegebenen Zeichenkette wird

		Beginn der angegebenen Zeichenkette wird zum neuen Arbeitstext. Die angegebene Zeichenkette gehört also mit zum neuen Arbeitstext.
t Zahl	t23	Vom Ende des Arbeitstextes werden so viele Zeichen entfernt, wie in der Zahl festgelegt wurde.
t"Zeichen(kette)"	t"ALLEGRO"	Vom Ende des Arbeitstextes wird der Teil entfernt, der mit der angegebenen Zeichenkette beginnt.
u[Zeichen Zeichen]	u[\$\$]	Alles was zwischen den zwei Zeichen steht, wird aus dem Arbeitstext entfernt. Wenn die Zeichen nicht angegeben werden, wird alles was in Nichtsortierzeichen steht entfernt.
U[Zeichen Zeichen]	U[\$\$]	Alles was zwischen den zwei Zeichen steht, wird aus dem Arbeitstext entfernt. Wenn die Zeichen nicht angegeben werden, wird alles was in Nichtsortierzeichen steht entfernt. Das erste Wort des übriggebliebenen Arbeitstextes wird groß geschrieben.
y0	y0	Die Umkodierung des Arbeitstextes wird verhindert.
y1	y1	Der Arbeitstext wird sofort mit den durch p definierten Umkodierbefehlen umkodiert. Danach kann er weiter bearbeitet werden.
y2	y2	Der Arbeitstext wird sofort mit den durch q definierten Umkodierbefehlen umkodiert. Danach kann er weiter bearbeitet werden.

8.3.2 Vergleichsbefehle

Befehl	Beispiel	Erläuterung
i Zahl,Zeichen	i5,R	Es wird überprüft, ob an der durch die Zahl festgelegten Position des Arbeitstextes das angegebene Zeichen vorkommt. Nur wenn das der Fall ist, wird die Befehlszeile weiter bearbeitet. Wenn nicht, wird die Befehlszeile beendet und es wird nichts ausgegeben.
I Zahl,Zeichen	I5,R	Es wird überprüft, ob an der durch die Zahl festgelegten Position des Arbeitstextes das angegebene Zeichen nicht vorkommt. Nur wenn das der Fall wird die Befehlszeile weiter bearbeitet. Wenn es vorkommt wird die Befehlszeile beendet und es

wird nichts ausgegeben.

c"Zeichen(kette)" c"ALLEGRO" Es wird überprüft, ob die angegebene Zeichenkette im Arbeitstext vorkommt. Wenn sie nicht vorkommt wird die Befehlszeile nicht mehr weiter bearbeitet und es wird nichts ausgegeben.

8.4 Zwei einfache Beispiele für Parameterdateien

8.4.1 Die einfachste Parameterdatei - der Pauschalexport

Beim folgenden Beispiel handelt es sich um die denkbar einfachste Parameterdatei, einen Pauschalexport aller Kategorien eines Datensatzes in das ALLEGRO-Grundformat.

ae=13 10 Grundparameter für das Aufnahmeende wird auf ASCII-Codes 13 und 10 gesetzt. Dies ist die Zeichenfolge, die einen ALLEGRO-Datensatz beendet.

ke=0 Grundparameter für das Kategorieende auf ASCII-Code 0 setzen.

as=h0 Eine Zeichenfolge wird definiert, die vor jeder Aufnahme ausgegeben wird. **h0** ist der sogenannte **Hierarchieseperator**. Bei normalen Datensätzen entspricht er dem ASCII-Code 1. Einen anderen Wert nimmt er nur bei Bandsätzen an, die hierarchisch (also mit der Zwischenspeicherung mit F9) erfaßt werden.

zl=0 Zeilenlänge wird auf unendlich gesetzt. Es findet kein Zeilenumbruch statt.

ks=1 Die Ausgabe der Kategorien beginnt mit der Position 1. Die Kategoriennummern werden also mit ausgegeben.

Damit sind die Grundparameter definiert.

Pauschalexport-Befehl. Alle Kategorien des Datensatzes werden exportiert.

/415 Aber die Kategorie 415 wird dabei weggelassen. Diese Form, um Kategorien vom Export aus auszuschließen, funktioniert nur beim Pauschalexport.

/50. Ebenso alle Kategorien, die mit 50 beginnen. Der Punkt ist ein Platzhalter für das dritte Zeichen der Kategoriennummer.

#+# Sprung zur Sprungmarke #, durch die Parameterdatei verlassen wird. Wäre allerdings am Ende der Parameterdatei nicht unbedingt nötig, schadet aber auch nicht.

8.4.2 Vorbereitung von Daten für eine Sortierung

In diesen Parameterdateien muß für jeden Datensatz ein Sortierkopf gebildet und am Anfang des Datensatzes ausgegeben werden. In der Regel werden dann alle Kategorien des Datensatzes pauschal exportiert, da die sortierten Daten in einem weiteren Schritt zum Endprodukt verarbeitet werden.

zl=0	Kein Zeilenumbruch, da eine unbegrenzte Zeilenlänge eingestellt wird.
ks=1	Die Daten der einzelne Kategorien werden ab der ersten Position ausgegeben. Dadurch wird die Kategorienummer mit exportiert. Dies ist bei Exporten, die Daten im ALLEGRO-Grundformat erzeugen sollen, immer notwendig.
ae=13 10	Am Ende jedes Datensatzes werden die ASCII-Codes 13 und 10 ausgegeben.
as=h0	Eine Zeichenfolge wird definiert, die vor jeder Aufnahme ausgegeben wird. h0 ist der sogenannte Hierarchieseperator . Bei normalen Datensätzen entspricht er dem ASCII-Code 1. Einen anderen Wert nimmt er nur bei Bandsätzen an, die hierarchisch (also mit der Zwischenspeicherung mit F9) erfaßt werden
ke=0	Grundparameter für das Kategorieende auf ASCII-Code 0 setzen.
ts.bpt	Teile von Parameterdateien können in Hilfsparameterdateien ausgelagert werden. Dies ist sehr häufig bei Zeichen-Umkodierungen der Fall. Mit dem Befehl t (Achtung: ohne Lattenkreuz vor dem t) werden diese Hilfsparameterdateien in eine Parameterdatei eingebunden.
!200 b4 e" [" p"u1 " #zz 0	Die Umkodierung der Zeichen erfolgt über die Alternativtabelle. Diese wird in Parameterdateien vielfach für Umkodierungen zu Sortierzwecken benutzt. (! an der 1. Position) Es werden mehrere Kategorien zu einer zusammengefaßt. In diesem Fall zur Sonderkategorie #u1, in die bei Sortierungen der Sortierkopf ausgegeben wird. (p"u1 ") Da der Kategorie vom ersten Zeichen an ausgegeben wird muß die Kategorienummer übergangen werden. (b4) Der ASCII-Code 0, der als Kategorieende definiert wurde, darf nicht ausgegeben werden, da ja noch weitere Kategorien an den Sortierkopf angehängt werden. (#zz 0)
!320 b4 u #zz 0	Eine weitere Kategorie wird angehängt, das Kategorieende wird unterdrückt. Durch den Befehl u werden Angaben in Nichtsortierzeichen, also Artikel am Titelanfang entfernt.
!425 b4 #zz 0	Eine weitere Kategorie wird an die Sortierkategorie angehängt. Das Kategorieende Zeichen wird wieder unter-

drückt.

#t{ 0 }

Das benötigte Kategorieende Zeichen wird nun eingefügt. Dazu wird wieder der sogenannte **Transfer-Befehl t** benutzt. Mit diesem Befehl können alle Zeichenketten ausgegeben werden, die auch als indirektes Postfix benutzt werden könnten. Auch die Syntax bei der Definition dieser Zeichenketten ist identisch.

Hier wird der Transfer-Befehl im Unterschied zum Aufruf von Hilfsparameterdateien aber mit führendem Latzenkreuz benutzt.

##

Pauschalexport aller Kategorien

#+#

Ende der Parameterdatei

Diese kurzen Beispiele sollen an dieser Stelle genügen. Im ALLEGRO-Lehrbuch von Heinrich Allers gibt es eine systematische Übersicht über verschiedene Typen von Export-Parameterdateien. Insbesondere sind hier die verschiedenen Grundparameter interessant, die für die einzelnen Typen von Parameterdateien erforderlich sind. Diese Grundparameter sind weitgehend unabhängig vom Konfigurationsschema.

9 Fortgeschrittene(re) Parametrisierung

9.1 Die Verwendung von Unterprogrammen

Mit Hilfe von Unterprogrammen können längere Parameterdateien übersichtlicher gestaltet werden. So kann es vorkommen, daß an verschiedenen Stellen einer Parameterdatei weitgehend identische Aktionen durchgeführt werden sollen. Diese Aktionen werden in Unterprogrammen beschrieben. An den benötigten Stellen der Parameterdatei werden dann die Unterprogramme aufgerufen. Unterprogramme können **unterhalb** der Stelle, von der aus sie aufgerufen werden, an einer beliebigen Stelle der Parameterdatei definiert werden.

Definition eines Unterprogramms:

#(A

Beginn des Unterprogramms. Gekennzeichnet wird dies durch die öffnende runde Klammer.

Unterprogramme werden mit einem einzelnen Buchstaben oder einer einzelnen Ziffer benannt. Über diesen "Namen" können sie aufgerufen werden.

#.....

Nun folgen beliebige Befehlszeilen. Ein Unterprogramm kann beliebig lang sein. Aus einem Unterprogramm heraus können wiederum weitere Unterprogramme aufgerufen werden.

#.....

#.....

#)A

Das Ende des Unterprogramms wird durch die schließende runde Klammer gekennzeichnet. Die Bezeichnung des Unterprogramms wird wiederholt.

Der Aufruf von Unterprogrammen:

Man kann hier zwischen zwei Formen unterscheiden. Dem bedingten und dem unbedingten Aufruf eines Unterprogramms. Beim **bedingten Aufruf** wird der Unterprogrammaufruf als Manipulationsbefehl in einer Befehlszeile benutzt. Diese Befehlszeile selbst erzeugt dann keine Ausgabe. Statt dessen wird das Unterprogramm ausgeführt. Es ist aber möglich im Unterprogramm selbst auf den unveränderten Inhalt der Kategorie, die am Anfang der Befehlszeile steht über die **Sonderkategorie #cc** zu zugreifen. Dadurch wird ein Unterprogramm flexibler in der Anwendung.

Zur Verdeutlichung ein Beispiel. Ein Unterprogramm soll bei verschiedenen Kategorien, die Personennamen enthalten, jeweils eine Invertierung der bibliothekarischen Schreibweise vornehmen.

#200 p>A Bedingter Aufruf eines Unterprogramms A aus drei verschiedenen Kategorien heraus. Der **Befehl** für den Aufruf des Unterprogramms ist **p>**.

#201 p>A

#202 p>A

Die drei Zeilen bewirken keine Ausgabe. Statt dessen wird jeweils das Unterprogramm ausgeführt.

Der unveränderte Inhalt der Kategorie, von der aus der Aufruf erfolgt, steht in der Sonderkategorie #cc.

#(A

Beginn des Unterprogramms

#cc b", "

In der Sonderkategorie #cc steht der jeweils zu verarbeitende Inhalt.

Es wird nach dem ersten Komma, also beim Vornamen begonnen. Wenn es kein Komma gibt, passiert nichts.

#cc e", " p" "

Nun wird der Familienname, also der Kategorieninhalt bis zum Komma ausgegeben. Davor wird ein Leerzeichen als Präfix gesetzt.

#)A

Ende des Unterprogramms

⇒ **Achtung:** Wenn aus einem Unterprogramm heraus ein weiteres Unterprogramm aufgerufen wird, dann enthält die Sonderkategorie #cc den Inhalt der Kategorie, von der aus das erste Unterprogramm aufgerufen wurde. Wenn Kategorieninhalte an das folgende Unterprogramm weitergegeben werden sollen, muß man mit Anwendervariablen arbeiten.

Unbedingte Aufrufe von Unterprogrammen hängen nicht von der Existenz bestimmter Kategorien ab. Sie sind fest in den Ablauf der Parameterdatei eingebaut und werden auf jeden Fall ausgeführt, wenn die entsprechende Stelle der Parameterdatei bei der Abarbeitung erreicht wird. Das Unterprogramm wird mit dem Befehl **#>[Name des Unterprogramms]** aufgerufen.

9.2 Anwendervariablen

In jeder Programmiersprache gibt es Variablen. In der ALLEGRO Exportsprache heißen sie **Anwendervariablen**. Sie dienen zur Zwischenspeicherung von Arbeitstexten,

die bei der Verarbeitung von Kategorien entstehen. Diese Anwendervariablen können innerhalb der Parameterdatei wie normale Kategorien aufgerufen, verändert oder ausgegeben werden. Sie können also in Befehlszeilen anstelle einer Kategorienummer eingesetzt werden. Besonders wichtig sind sie in Fällen, in denen Inhalte aus einem Datensatz mit Inhalten eines darauf folgenden Datensatzes verglichen werden sollen.

Die **Namen von Anwendervariablen** können in der Regel unter Beachtung der vorgegebenen Syntax frei gewählt werden. Es gibt jedoch bestimmte Anwendervariablen, deren Name meist mit **x** beginnt die bereits von den ALLEGRO-Programmen automatisch mit Inhalten belegt werden.

Die Namen von Anwendervariablen bestehen bei der ALLEGRO Exportsprache aus einer Kombination von zwei Zeichen. Das **erste** der beiden Zeichen muß dabei ein **Kleinbuchstabe** sein. **aX, ab, aC, d2, dX** sind also gültige Namen für Anwendervariablen, **D1, Ae, EE** dagegen nicht.

Beim Aufruf einer Anwendervariable anstelle einer Kategorienummer wird vor diesen aus zwei Zeichen bestehenden Namen immer ein **u** gestellt. Die oben benannten Anwendervariablen werden also so aufgerufen: **#uaX, #uab, #uaC, #ud2, #udX**. Der aktuelle Inhalt dieser Variablen kann dann wie der Inhalt einer normalen Kategorie bearbeitet werden.

Der Inhalt von Anwendervariablen bleibt solange bestehen, bis die Anwendervariable gelöscht oder mit einem anderen Inhalt belegt wird oder bis der Exportvorgang, für den die Parameterdatei benutzt wird, beendet ist.

Der Inhalt von Anwendervariablen kann mit folgenden Befehlen verändert werden. Als Beispiel wird jeweils eine Anwendervariable mit der Benennung w2 benutzt.

dw2 Der Inhalt der Anwendervariable wird gelöscht. Dieser Löschbefehl für Anwendervariablen kann in jede beliebige Anweisungszeile eingebaut werden. Er hat keine Auswirkung auf die weitere Bearbeitung der Anweisungszeile.

Anwendervariablen sollten sofort gelöscht werden, sobald sie nicht mehr benötigt werden, da man sonst leicht die Übersicht über ihren Inhalt verliert. Dies ist z.B. auch in einer Befehlszeile möglich, in der die Variable selbst verarbeitet wird.

#uw2 dw2 b4 e10

Hier kann die Anwendervariable noch verarbeitet werden, obwohl sie in der Befehlszeile gelöscht wird.

aw2 Vor den aktuellen Inhalt der Anwendervariable wird der in einer Befehlszeile entstandene Arbeitstext gesetzt. Wenn die Anwendervariable bisher nicht existierte, wird sie angelegt. Der entstandene Arbeitstext wird nicht ausgegeben, sondern in der Anwendervariable gespeichert.

#320 b4 u e10 aw2

Aw2 An den aktuellen Inhalt der Anwendervariable wird der in einer Befehlszeile entstandene Arbeitstext angefügt. Wenn die Anwendervariable

Befehlszeile entstandene Arbeitstext angefügt. Wenn die Anwendervariable bisher nicht existierte, wird sie angelegt. Der entstandene Arbeitstext wird nicht ausgegeben, sondern in der Anwendervariable gespeichert.

#320 b4 u e10 Aw2

=w2

Der aktuelle Inhalt der Anwendervariable wird durch den in einer Befehlszeile entstandenen Arbeitstext ersetzt. Wenn die Anwendervariable bisher nicht existierte, wird sie angelegt. Der entstandene Arbeitstext wird nicht ausgegeben, sondern in der Anwendervariable gespeichert.

#320 b4 u e10 =w2

=w2+M

Inhalte von Anwendervariablen dienen sehr oft Vergleichszwecken. Der **Vergleichsbefehl** ist mit dem Zuweisungsbefehl im Prinzip identisch. Er wird jedoch durch eine Sprungmarke bzw. die Angabe der Nummer eines Zwischenteils ergänzt.

#320 b4 u e10 =w2+M

Der aktuelle Arbeitstext wird mit dem Inhalt der Anwendervariable verglichen. Wenn er mit dem Inhalt der Anwendervariable **identisch** ist, wird zur Sprungmarke #-M gesprungen. Der entstandene Arbeitstext wird nicht ausgegeben und ersetzt auch nicht den Inhalt der Anwendervariable.

Wenn der Arbeitstext und der Inhalt der Anwendervariable **nicht identisch** sind, wird der entstandene Arbeitstext ausgegeben **und** der bisherige Inhalt der Variable wird durch den Arbeitstext, mit dem sie verglichen wurde ersetzt.

Ein Beispiel soll dies verdeutlichen: In einer ALLEGRO-Grunddatei könnten Datensätze doppelt vorkommen. Sobald die Datei sortiert wurde, folgen diese doppelten Datensätze unmittelbar aufeinander. Durch den Vergleich über eine Anwendervariable, in der jeweils die Identnummer des Datensatzes gespeichert wird, können diese doppelten Datensätze eliminiert werden.

#000 =w2+# e0 #zz 0 Der Inhalt der Kategorie #000, also die Identnummer des Datensatzes, wird jeweils mit dem aktuellen Inhalt der Variable, also mit der Identnummer des vorhergehenden Satzes, verglichen. Wenn die beiden Inhalte gleich sind wird die Parameterdatei beendet.

Die Angabe **e0** dient dazu, eine Ausgabe der Identnummer zu verhindern, falls die verglichenen Nummern ungleich sind.

##

Nur wenn die Identnummern beim Vergleich unterschiedlich waren werden die Kategorien des Datensatzes ausgegeben.

#+#

Ende der Parameterdatei

Ein weiteres Beispiel zur Nutzung von Anwendervariablen. In einer Kategorie stehen durch Leerzeichen * Leerzeichen getrennt beliebig viele gleichartige Inhalte, z.B. Schlagworte. Jedes Schlagwort soll nun auf einer neuen Zeile ausgegeben werden. Mit Hilfe von Anwendervariablen läßt sich eine Schleife konstruieren, die dieses leistet.

- #dt dw1 dw2 e0** Die benötigten Anwendervariablen werden erst einmal vorsichtshalber gelöscht. Da die Sonderkategorie #dt (Datum) immer existiert, kann man einen Aufruf dieser Kategorie dazu benutzen. Durch die Angabe e0 wird verhindert, daß das Datum ausgegeben wird.
- #114 =w1** Der Inhalt der Kategorie wird in eine Anwendervariable kopiert.
- #-A** Das Sprungziel der Schleife.
- #uw1 e" * " =w2** Der erste Begriff der in die Anwendervariablen w1 kopierten Kategorie wird einer weiteren Anwendervariablen w2 zugewiesen
- #uw2 p{ 13 10 }** Nun wird der Inhalt der Anwendervariablen w2 ausgegeben. Als indirektes Präfix wird ein Zeilenumbruch vorangestellt.
- #uw1 +A b" * " =w1** Nun wird von der Anwendervariable w1 der jeweils erste Begriff abgeschnitten. Dies funktioniert solange, wie im Inhalt der Anwendervariable die Zeichenfolge Leertaste * Leertaste vorkommt.
- Der so verkürzte Inhalt in die Anwendervariable w1 kopiert. Wenn alle Aktionen erfolgreich durchgeführt werden konnten, wird zurück zum Sprungziel der Schleife gesprungen und der Vorgang beginnt von neuem.

Dies ist eine Aufgabenstellung, die ohne die Nutzung von Anwendervariablen nicht zufriedenstellend gelöst werden könnte. Abschließend noch ein Beispiel, bei dem mehrere Kategorien in einer Anwendervariable zusammen gefaßt werden. Der gesamte Inhalt der Anwendervariable wird dann in eine neue Kategorie exportiert.

- #dt dw2 e0 #zz 0** Inhalt der Anwendervariablen löschen. Es wird wieder die Sonderkategorie #dt verwendet.
- Durch die Angabe **e0** wird die Ausgabe des Datums verhindert, durch den Befehl **#zz 0** wird zusätzlich verhindert, daß ein im Grundparameter ke definiertes Kategorieendezeichen ausgegeben wird.
- #320 b4 =w2 #zz 0** Der Inhalt von Kategorie 320 wird ohne Kategorienummer (**b4**) in die Anwendervariable w2 kopiert.
- #335 p" : " Aw2 #zz 0** Nun wird der Inhalt von Kategorie 335 zusammen mit einem Präfix angehängt.
- #uw2 b4 p"321 "** Nun wird der Inhalt der Anwendervariable als Kategorie 321 exportiert. Durch den Befehl **b4** beginnt die Ausgabe erst beim eigentlichen Inhalt der Variable. Ansonsten würde die

Zeichenfolge **uw2** mit ausgegeben.

9.3 Satzverknüpfungen

Bis Version 11 war es in ALLEGRO nicht möglich, Satzverknüpfungen durchzuführen. Erst in Version 12 wurden die sogenannten **Nachladebefehle** eingeführt. Seither müssen Bände mehrbändiger Werke nicht mehr hierarchisch in einem großen Datensatz gespeichert werden. Statt dessen bilden sie einzelne Datensätze, die dann bei der Ausgabe - beispielsweise auf dem Bildschirm - über Parameterdateien mit dem übergeordneten Werk verknüpft werden. Ein weiteres Anwendungsgebiet ist die Verknüpfung von Serienstücken mit einem Serienstammsatz.

Bei den untergeordneten Einheiten muß eine Kategorie existieren, in der die Identnummer der übergeordneten Einheit in irgendeiner Form erfaßt ist. Die Verknüpfungsmethodik differiert leider zwischen dem Standard-Kategorienschema und dem in Nordrhein-Westfalen gebräuchlichen sehr stark. In den Parameterdateien wird diese Identnummer nun als Arbeitstext aus der Kategorie extrahiert. Mit diesem Arbeitstext wird versucht über einen Nachladebefehl den entsprechenden Datensatz in den Arbeitsspeicher zu holen. Wenn dies gelingt, kann der Datensatz in der Parameterdatei ausgewertet werden. Wenn er nicht mehr benötigt wird, wird zum Ausgangsdatsatz zurück geschaltet. Zu diesem Zweck existieren die sogenannten **Schaltbefehle**.

Technisch läuft die **Verknüpfung über den Index** der Datenbank. Beim Nachladebefehl wird überprüft, ob im Index Einträge existieren, die mit dem zum Nachladen benutzten Arbeitstext identisch sind oder mit diesem Arbeitstext beginnen. Über diese Indexeinträge werden dann die internen Satznummern ermittelt und die Datensätze können zur Auswertung aus der Datenbank geholt werden.

Da die Auflösung der Verknüpfungen über den Index realisiert wird, war es lange Zeit nicht sinnvoll möglich, auch Personennormsätze, Körperschaftsnormsätze u.ä. über Identnummern mit einer Titelaufnahme zu verknüpfen. Bei der Indexierung der Titelaufnahme konnten ja nur die Identnummern der Normsätze im Register der Datenbank angezeigt werden.. Damit war den Benutzern allerdings nicht geholfen.

Erst mit ALLEGRO Version 14 wurde eine Lösung für dieses Problem gefunden. Im Index der Datenbank werden nun spezielle Indexeinträge erzeugt, die sogenannten **Ersetzungsschlüssel**. Dies sind Indexeinträge, die die Form **Nummer=Klartext** haben. Über diese Ersetzungsschlüssel werden nun bei einem zur Bearbeitung durch eine Parameterdatei aufgerufenen Datensatz im Arbeitsspeicher zunächst die Nummern durch den Klartext ersetzt. Dies funktioniert allerdings nicht bei mehrbändigen Werken u.ä., sondern nur dann, wenn statt einer Nummer eine einzelne normierte Ansetzungsform stehen soll.

Im Gegensatz zu anderen Teilen der Export-Sprache sind die Nachlade- und Schaltbefehle, sowie die Ersetzungsschlüssel im **ALLEGRO-Handbuch** sehr ausführlich dokumentiert. Auf diese Ausführungen in **Kapitel 10.2.6.7 und 10.2.6.8** des Handbuchs sei hier verwiesen.

Dieser kurze Abriß der Export-Sprache strebt keine Vollständigkeit an. Er ist vielmehr als Hilfestellung zum Verständnis der entsprechenden Kapitel des ALLEGRO-Handbuches gedacht. Die hier gemachten Anmerkungen sollen es ermöglichen, die

teilweise recht knappen Ausführungen im Handbuch besser ein zuordnen. Parametrisieren lernt man am besten spielerisch durch Ausprobieren. Sehr nützlich ist dazu die von Heinrich Allers konzipierte Lerndatenbank, die es sowohl auf Diskette als Beilage zu seinem ALLEGRO-Lehrbuch, als auch auf dem ALLEGRO-FTP-Server in Braunschweig gibt.

10 ALLEGRO-Datenbanken im Internet: Der AVANTI Server

Bei den normalen ALLEGRO-Programmen handelt es sich um monolithische Programme. Sie können nur dann arbeiten, wenn sie ständig Zugriff auf die jeweilig benutzte ALLEGRO-Datenbank haben. Mit AVANTI wird es erstmals möglich mit ALLEGRO eine Client-Server-Architektur zu realisieren, wie sie für Datenbank-Abfragen über das Internet benötigt wird.

In dieser Programmarchitektur schickt der Client an den Server eine Anfrage in einer normierten Abfragesprache. Bei relationalen Datenbanken wird als Abfragesprache SQL verwendet. Die Abfragesprache des AVANTI-Servers orientiert sich an den in PICA verwendeten Befehlen. Der Server schickt die Ergebnisse zurück, die der Client dann dem Benutzer entsprechend aufbereitet präsentiert. Das Client Programm ist vom Server völlig unabhängig und kann in einer beliebigen Programmiersprache erstellt werden, solange es dem Server die Suchanfrage in einer Form übermittelt, die dieser versteht und mit den gelieferten Ergebnissen etwas anfangen kann.

Als Kommunikationsweg zwischen dem Client und dem Server finden in der Regel TCP/IP Sockets Verwendung. Es handelt sich hier um Mechanismen, die bei der Rechnerkommunikation über Internet verwendet werden. Falls der Server auf einem WINDOWS NT Rechner läuft, können auch sogenannte "Named Pipes" benutzt werden.

Als Plattform für den AVANTI-Server findet sinnvollerweise ein UNIX-System oder WINDOWS NT Verwendung. WINDOWS 95 ist nur zum Ausprobieren sinnvoll. Für einen ernsthaften Einsatz ist die Performance zu gering. Der Server läuft auf UNIX-Systemen als Dämon, auf WINDOWS NT Systemen als Systemdienst. Das Programm ist ständig - sozusagen im Standby Betrieb - aktiv und wartet auf Abfragen.

Der Aufruf einer Datenbank über das Internet erfordert noch einen Zwischenschritt. Der Benutzer richtet seine Anfrage zunächst an den **WWW-Server**. Dieser kann die Anfrage nicht aus eigener Kraft bearbeiten, sondern benötigt **Hilfsprogramme**, sogenannte **CGI-Programme**. Mit diesen kommuniziert er über die **CGI-Schnittstelle**. Daher der Name. Das CGI-Programm ist im skizzierten Schema der Client, der die Anfrage an den **AVANTI-Server** weiter reicht. Dieser führt die Anfrage aus und liefert das Ergebnis an das CGI-Programm zurück. Dieses wandelt die erhaltenen Daten in das HTML-Format um und schickt die fertig aufbereiteten Daten über die CGI-Schnittstelle zurück an den WWW-Server, der sie dann endgültig an den Benutzer im Internet übermittelt.

Soweit ganz kurz und vereinfacht die Abläufe. Es würde den Rahmen dieses Papiers sprengen, hier intensiver auf die Spezifikation der CGI-Schnittstelle oder gar auf PERL einzugehen.

Die CGI-Programme, die der WWW-Server benötigt sind auf dem FTP-Server in Braunschweig frei erhältlich. Es müssen lediglich einige kleinere Änderungen durch-

geführt werden, um sie auf lokale Gegebenheiten anzupassen. Die Dokumentation ist ebenfalls auf dem Internet-Server in Braunschweig zu finden. Sie wurden in der im World Wide Web häufig verwendeten Programmiersprache PERL erstellt.

Die Einrichtung der WWW-Schnittstelle erfordert jedoch trotzdem einiges an Hintergrundwissen. Wenn kein engagierter Anwender greifbar ist, der die Einrichtung erledigen kann wird man zwangsläufig auf die Unterstützung eines Rechenzentrums oder einer EDV-Abteilung angewiesen sein.

Was braucht man an Programmen zur Realisierung einer WWW-Anbindung?

Zunächst einmal muß natürlich ein funktionsfähiger WWW-Server vorhanden sein. Für die allegro-spezifischen Teile der Anbindung benötigt man folgende Programme.

ACWWW25.EXE Ein selbstentpackendes Archiv auf dem FTP-Server. Es enthält sämtliche CGI-Programme, die zur Anbindung des AVANTI-Servers erforderlich sind. Zu finden ist es im öffentlich zugänglichen Verzeichnis **anwender**.

Die Dokumentation ist im Internet zu finden unter:

<http://www.biblio.tu-bs.de/allegro/doku/acwww25.htm>

AVANTI

Der AVANTI-Server ist den Abonnenten in entsprechenden Verzeichnissen auf dem FTP-Server zugänglich. Für die Versionen für UNIX Betriebssysteme ist ein spezielles etwa 1000 DM teures Abonnement erforderlich. Die WINDOWS Versionen können mit dem normalen ALLEGRO Abo bezogen werden.

PERL

Damit PERL-Programme ausgeführt werden können, muß auf dem System ein PERL Interpreter vorhanden sein. Diesen und eine Vielzahl von freien PERL Programmen gibt es am frischesten in den sogenannten CPAN-Archiven. In Deutschland sind die CPAN-Archive unter anderem auf dem Rechner

<ftp://ftp.rz.ruhr-uni-bochum.de>

im Verzeichnis

[/pub/programming/languages/perl/CPAN/](#)

zu finden. Dort gibt es Versionen sowohl für UNIX, als auch für Win32 (NT/Win95/98)

11 Literaturhinweise, Internetquellen

1. Eversberg, Bernhard

allegro-C : Systemhandbuch. Braunschweig: Universitätsbibliothek der TU Braunschweig.

Das Systemhandbuch ist natürlich für die intensivere Arbeit mit ALLEGRO unverzichtbar. Zur Zeit liegt es in der Version 15 vor.

2. **Allers, Heinrich**

allegro-C Lehrbuch. Version 14. Braunschweig, UB 1995.

Das Lehrbuch ist leider vergriffen. Die zugehörige Lerndatenbank ist aber auf dem FTP-Server an der TU in Braunschweig unter dem Dateinamen **ALLERS.LZH** im Verzeichnis **LEHRBUCH** zu erhalten.

3. **Tews, Annemarie**

allegro-Ouverture : Partitur für Einsteiger. - Braunschweig: UB, Mai 1997

Enthält umgearbeitete Teile des Lehrbuchs von Herrn Allers, beschäftigt sich aber nicht mit Parametrisierung.

4. **Großgart, Ruth u.a.**

Lernbuch für die Einführung von ALLEGRO in der Bibliothek der Friedrich-Ebert-Stiftung. Bonn, FES 1994.

Gibt Katalogisierungshilfen für das nordrhein-westfälische Kategorienschema in der in der FES benutzten Ausprägung. Leider schon lange vergriffen, aber über Fernleihe auf jeden Fall in Bo 133 zu erhalten.

5. **allegro news**. Braunschweig, UB.

Erscheint ungefähr vierteljährlich und gibt einen Überblick über Neuentwicklungen. Es stellt praktisch eine laufende Aktualisierung des Handbuchs da. Jeder Abonnent erhält es. es ist aber auch via Internet unter der Adresse

<http://www.biblio.tu-bs.de/allegro/news/index.htm>

im HTML-Format zu erhalten.

6. Eine Fülle von Informationen bietet der **ALLEGRO WWW-Server** unter der Adresse

<http://www.biblio.tu-bs.de/allegro/index.htm>

7. Vielfältige Informationen und Hilfestellung von Kolleg(inn)en bei Problemen bietet die **ALLEGRO Mailing Liste**. Man kann sich via Internet auf der Seite

<http://www.biblio.tu-bs.de/allegro/email.htm>

anmelden oder eine Mail an **maiser@buch.biblio.etc.tu-bs.de** richten, die nur die Zeile **subscribe allegro** enthält.

8. Es gibt auch eine **Mailing Liste für ALLEGRO Anwender in Nordrhein-Westfalen**, auf der allerdings zur Zeit noch nicht besonders viel los zu sein scheint. Anmelden kann man sich bei **maiser@hbz-nrw.de** mit einer Mail, in der nur die Zeile **subscribe allegro** steht.