

TEXVE  
Modul zur Textverarbeitung  
auf dem KC 85-2 und KC 85-3

=====

Berlin, März 1986

IEB Mikroelektronik "Wilhelm Pieck"  
Mühlhausen  
5700 Mühlhausen  
Eisenacher Str. 40

=====

Das Textverarbeitungssystem TEXVE ist ein Markenprodukt des  
IEB Mikroelektronik Mühlhausen und ist urheberrechtlich  
geschützt.

Anfertigung von Softwarekopien jeglicher Art und  
Vervielfältigungen der Beschreibung sind strafbar.  
Wir danken Prof. H. Völz für die Systementwürfe, die Codierung  
und die Erarbeitung dieses Textes.

=====

Inhalt:

1. Kurzbeschreibung
2. Inbetriebnahme
3. TEXOR
  - 3.1. Fileeintritt
    - 3.1.1. Erzeugen eines neuen File
    - 3.1.2. Laden und Aufrufen eines File
    - 3.1.3. Verlassen des Textmode
    - 3.1.4. Die drei Modi
  - 3.2. Textmode
    - 3.2.1. Sonderzeichen
    - 3.2.2. Blättern im Text
    - 3.2.3. Erzeugen von Leerzeilen
    - 3.2.4. Bewegen im File
    - 3.2.5. Einfaches Editieren
    - 3.2.6. Einfügen von Text
    - 3.2.7. Entfernen von Text
    - 3.2.8. Textsymmetrierungen
    - 3.2.9. Rechtsbündiger Text
    - 3.2.10. Tabulator
    - 3.2.11. Die beiden Textfenster
  - 3.3. Ausgabe von Text

- 3.3.1. Randsuche
- 3.3.2. Textausgabe
- 3.3.3. Teiltextausgabe
- 3.4. Veränderungsmodus
  - 3.4.1. Erzeugen von Absätzen
  - 3.4.2. Entfernen von Absätzen
  - 3.4.3. Verlagern von Textteilen
  - 3.4.4. Doppeln von Textteilen
  - 3.4.5. Probleme des Veränderungsmodus
- 3.5. Zeichensuche
- 4. SORED
  - 4.1. Drei Besonderheiten
    - 4.1.1. Bedeutung der Quasizeile
    - 4.1.2. Der Kopfteil
    - 4.1.3. Normale und reziproke Suche
  - 4.2. Erzeugung des File
  - 4.3. Einlesen eines vorhandenen File und Fileanzeige
  - 4.4. Anzeige und Druck eines File
  - 4.5. Sortieren
  - 4.6. Zeichensuche
  - 4.7. Editieren
  - 4.8. Entfernen von Zeilen (delete)
- 5. FILEX
  - 5.1. Die beiden Filestrukturen und ihre Wandlung
    - 5.1.1. TEXOR-Struktur
    - 5.1.2. SORED-Struktur
    - 5.1.3. Wandlung zwischen den Files
  - 5.2. Kassetten-File-Arbeit
    - 5.2.1. Retten von Files
    - 5.2.2. Hinweise zum Umgang mit Aufzeichnungen
    - 5.2.3. Zusammenfügen von Files
    - 5.2.4. Verwenden alter Files
  - 5.3. Zeichen ersetzen
  - 5.4. Zeichenketten ersetzen
  - 5.5. Fileanzeige
- 6. Diagnosesystem
- 7. Initiierung V-24
- 8. Anhang

=====

## TEXVE

Textverarbeitung auf KC 85-2 und KC 85-3

### 1. Kurzbeschreibung

Dieser Modul enthält zwei anspruchsvolle Textverarbeitungsprogramme mit für einen Kleinstrechner sehr hohem Komfort und Überraschend hoher Leistungsfähigkeit. Sie zeichnen sich stark vereinfacht durch folgende Eigenschaften aus:

TEXOR ist ein voll bildschirmorientiertes Textsystem. Es gestattet, umfangreiche Texte

- zu schreiben,
- zu editieren,
- zu verarbeiten,
- zu verändern (Absätze, Textteile verlagern usw.),
- Texte in Zeilenbreiten von 10 bis 255 Zeichen/Zeile in Flatter- bzw. Blocksatz (rechter Rand glatt) auszudrucken.
- Es enthält einen halbautomatischen Silbentrennalgorithmus.
- Es enthält ein Zeichenkettensuchprogramm.

SORED ist ein zeilenorientiertes Sortierprogramm. Es gestattet:

- eine mehrzeilige Quasizeile zu erzeugen, die als Einheit bestehen bleibt,
- Quasizeilen ohne Berücksichtigung der Groß-Kleinschreibung und auch bzgl. Umlaute und ß alphabetisch zu sortieren,
- die Zeilen und Quasizeilen beliebig bildschirmorientiert zu editieren (auch zu löschen).
- Es enthält einen Zeichensuch- und Zeichenantisuchmodus.
- Es enthält einen Kopfteil, der nicht in den Sortiervorgang einbezogen wird.

Die Auswahl zwischen SORED und TEXOR sollte also vor allem nach folgenden Gesichtspunkten erfolgen:

SORED ist effektiv, wenn Textzeilen (auch mehrzeilige) alphabetisch sortiert werden müssen, also bei Adressen, Nachwort- und Telefonverzeichnissen usw.

TEXOR ist effektiv, wenn Texte ohne bevorzugte Zeilenstruktur, also in erster Linie mit Absätzen, geschrieben und editiert werden sollen, also bei Manuskripten, Tabellen, Übersichten, Gedichten usw. TEXOR ist generell universeller, erlaubt aber kein automatisches Sortieren. SORED ist spezieller auf Zeilen und Quasizeilen orientiert. Beide Programme erlauben, vorstrukturierte Tabellen und Formulartexte nachträglich zu ergänzen.

Beide Programme werden durch ein Zusatzprogramm FILEX ergänzt. Es enthält für SORED und TEXOR Möglichkeiten zum

- Austauschen von Einzelzeichen
- Austauschen von Zeichenketten
- Speichern von Files
- Zusammenfügen von Files
- Verändern der Kopfzeilenanzahl von SORED
- Wandlung von SORED- in TEXOR-Files und umgekehrt.

Weiter existieren mit INIT-V-24 Initialisierungsprogramme für unterschiedliche Druckeranschlüsse.

Schließlich existiert ein Diagnoseprogramm "D", welches in Sonderfällen nützlich sein kann.

Alle Programme bauen auf vielfach erprobten Vorgängervarianten in Kassettenversion auf. Damit erzeugte Files können relativ leicht bei TEXVE weiterverwendet werden.

## 2. Inbetriebnahme

Bei ausgeschaltetem Gerät wird der Modul in einen Schacht des Rechners eingeschoben. Zur Vereinfachung werde angenommen, daß nur das Grundgerät mit dem rechten und linken Modulschacht zur Anwendung kommt. Dann wird in den rechten Modulschacht die Schnittstelle zur Schreibmaschine bzw. zum Drucker und in den linken Schacht der Modul TEXVE gesteckt. Nach dem Einschalten erfolgt die Initialisierung des TEXVE-Moduls mit

>SWITCH 0 C1.

Über >MENU erscheint danach auf dem Bildschirm:

```
* HC-CAOS XXXX *
>TEXT
>D
>SWITCH
>MENU
>SAVE
>VERIFY
>LOAD
>COLOR
>MODIFY
>*
```

XXXX ist die in Ihrem Gerät implementierte Version des Betriebssystems. Damit sind alle Voraussetzungen für das Starten des Textverarbeitungssystems gegeben. Der Cursor wird auf die Zeile

>TEXT

bewegt und mit ENTER läuft die entsprechende Initialisierung ab. Dabei werden u. a. die Sondertasten belegt, das kleine Alphabet mit richtiger Umschaltung bereitgestellt und der optimale Speicherbereich bestimmt. Auf dem Bildschirm erscheint danach:

```
#### Textverarbeitung ####

VEB Mikroelektronik Mühlhausen
Version 2.0

-----
I: INIT V 24
F: FILEX
S: SORED
T: TEXOR

STOP HC-CAOS
```

MI 11/2

Befehl: █

Damit wird ein Zeichen als Befehl verlangt. Zugelassen sind:  
I, F, S, T und STOP.

Mit STOP gelangt man zum Menu von CAOS zurück.  
I verlangt die Auswahl zwischen den Varianten der Peripherie (Abschnitt 7). F, S, T führen zu den noch zu beschreibenden Textsystemvarianten. Alle anderen Eingaben werden mit

Befehl: ERROR

Befehl: █

beantwortet. Die Eingabe der Befehle mit kleinen oder großen Buchstaben ist hier ohne Belang.

=====

### 3. TEXOR

TEXOR wird mit T oder t aus dem Start der Textverarbeitung TEXVE erreicht. Nach ca. 3 Sekunden erscheint das Menu:

\*\*\*\*\* TEXOR \*\*\*\*\*

Eingaben	Anzeigen	Ausgaben
N: Neue Daten	H: Menu	R: Randsuche
T: Dateianfang	F: Datei-	e: Flattersatz
B: Dateiende	werte	E: Blocksatz
G: Gehe Absatz-Nr.		

BRK: Hauptmenu

n, t, b, g, z mit kleinem Fenster

A: Erzeuge Absatz	V: Lösche Absatz	
\ rechtsbündig	W: lösche Zeichen	◊, I
Δ: Zentriere	L: Lösche Zeichen	~. ◊, I
□: Tabelliere	D: Lösche	◊ Text
Z: Suche Kette	P: Drucke	◊ Text
fortsetzen	M: Verlagere	◊ Text →
mit Shift CLR	C: Kopiere	◊ Text →

Befehl: █

Es wird einer der folgenden 12 Befehle erwartet:

B, E, F, G, H, L, N, R, T, W, Z (zweiter Teil), BRK.

Deshalb heißt dieser Zustand des Textverarbeitungssystems Befehlsmode. Mit Ausnahme von L, W und Z stehen die Befehle im oberen Teil des Menu. Alle anderen Eingaben werden mit "ERROR"

beantwortet. Mit H (Hilfe) kann jederzeit das oben gezeigte Menu aufgerufen werden.

### 3.1. Fileeintritt

#### 3.1.1. Erzeugen eines neuen Files

Falls noch kein File vorhanden ist, sollte mit

N:Neue Datei (New File)

begonnen werden. Dieser Befehl setzt auch die Zeiger (Pointer) eines alten File (alte Datei) zurück und löscht sie somit scheinbar. Durch das Betätigen von N wird also stets ein neues File begonnen. (bei versehentlichem Löschen, z. B. bei versehentlichem Betätigen von N, besteht die Möglichkeit, das File im Diagnosesystem D zu reaktivieren, siehe Abschnitt 6). Mit N schlägt der Bildschirm auf weiße Schrift auf schwarzem Grund um. Dies zeigt an, daß wir uns im Textmode befinden und alle Tasteneingaben mit Ausnahme von

BRK und ↓

als Texteingaben übernommen werden. Alles, was in dieser Form auf dem Bildschirm sichtbar geschieht, also alle Eingaben und Änderungen, gehen automatisch ins Textfile. Was auf dem Bildschirm zu sehen ist, was oben verschwindet usw., geht ebenfalls automatisch in den Filespeicher.

#### 3.1.2. Laden und Aufrufen eines File

Ein vorhandenes File kann im Betriebssystem (siehe Menu, Abschnitt 2) mit >LOAD geladen werden. Danach wird mit >TEXT ins Textverarbeitungssystem gegangen und mit T das Menu von TEXOR aufgerufen. Nun kann in das vorhandene File gesprungen werden, und zwar mit

- T zum Anfang des File (top),
- B zum Ende des File (bottom)
- G zu einem beliebigen Absatz im File  
(über die Abfrage

Zum Absatz:↓

und Eingabe einer Zahl zu dem entsprechenden Absatz).

Mit Farbwechsel ist man so wieder im Textmode. Mit den drei oben genannten Befehlen kann man nach dem Verlassen des Textmode immer wieder in den Textmode und damit zum File zurückkehren.

#### 3.1.3. Verlassen des Textmode

Liegt ein Text vollständig geschrieben vor, so muß man aus verschiedenen Gründen wieder in den Befehlsmode gemäß Menu auf 3.5 zurückkehren. Dies erfolgt durch Betätigen der Taste

BRK

Dabei wird der Bildschirm gelöscht und die Farbe wechselt

zurück auf weiße Schrift auf schwarzem Grund. Mit der Eingabe von F können jetzt die Dateiwerte erhalten werden. Auf dem Bildschirm erscheint folgende Anzeige:

```
File XXXXXX   Byte lang  
noch VVVVVV   Byte frei  
Zeilenlänge:  
ZZZZ Zeichen:
```

Befehl: ■

Hiermit erhalten Sie jederzeit einen Überblick über die Länge des vorhandenen File und wieviel noch eingegeben werden kann. Darüber hinaus wird die jeweils eingestellte Zeilenlänge angezeigt. 60 ist der bei Start automatisch angenommene Wert (default).

Es gibt noch eine weitere Möglichkeit, den Textmode zu verlassen. Hierzu dient das Sonderzeichen

↓  
Es führt in den sogenannten Veränderungsmode (Abschnitt 3.4). Dieses Zeichen sollte wegen seiner stark in das File eingreifenden Änderungsmöglichkeiten bei den ersten Arbeiten noch nicht benutzt werden.

Achtung! Verlassen Sie nie den Textmode (und auch nicht den Befehlsmode) mit der RESET-Taste am Grundgerät. Dabei werden wichtige Zeiger (Pointer) für das File nicht gesetzt und Sie haben Probleme, das File wieder nutzbar zu machen. Hierzu enthält das Diagnosesystem D in gewissem Umfang Möglichkeiten.

#### 3.1.4. Die drei Modi

Wir haben jetzt drei unterschiedliche Zustände des TEXOR kennengelernt. Sie seien hier zusammengefaßt dargestellt.

Befehlsmode wird erreicht aus dem Eingagsmenu (siehe Bild Abschnitt 2) durch den Befehl T. Er ist dadurch gekennzeichnet, daß er mit

Befehl: ■

auf eine Eingabe wartet. Der Befehlsmode sollte immer eingestellt werden, wenn keine Texteingabe erfolgt. Dadurch wird eine erhöhte Sicherheit gegenüber versehentlichen Änderungen des Files erreicht.

Der Textmode (siehe auch Abschnitt 3.2) wird aus dem Befehlsmode über:

B, G, N, T

und, wie später noch beschrieben wird, über

Z

und z. T. über



R und E

erreicht. Er ist durch folgende Fakten gekennzeichnet:

- Die Schrift ist weiß auf schwarzem Grund.
  - Es steht nicht "Befehl:" auf dem Bildschirm.
  - Ein vorhandenes Textfile ist zumindest ausschnittsweise auf dem Bildschirm sichtbar.
  - Alle Tastenbefehle werden auf dem Bildschirm unmittelbar sichtbar, meist als Eingabe eines Zeichens.
- Der Textmode wird mit

BRK bzw.  $\downarrow$

verlassen, wobei  $\downarrow$  erst bei Erfahrungen mit dem System benutzt werden sollte. BRK führt in den Befehlsmode zurück. Gleichwertig mit BRK sind auch die Tastenbetätigungen STOP und Shift CLR (d. h. CLS).

Der Veränderungsmode wird mittels  $\downarrow$  aus dem Textmode angesteuert. Nach Eingabe dieses Zeichens sind vielfältige Veränderungen des Textes, wie Umstellen, Doppeln und Löschen von Textteilen, Erzeugen und Löschen von Absätzen möglich (siehe Abschnitt 3.4). Das auf  $\downarrow$  folgende Zeichen bestimmt hier die Auswahl. Sollten Sie versehentlich  $\downarrow$  betätigt haben, so geben Sie anschließend X ein und mit ERROR landen Sie im Befehlsmode. Abschließend geben Sie den Befehl W zum Löschen des Zeichens  $\downarrow$ .

## 3.2. Textmode

Der Textmode arbeitet mit weißer Schrift auf schwarzem Grund. Er wird erreicht über

- N neues File
- T am Fileanfang
- B am Fileende
- G ab Absatz nn  
(nn = Zahl, Nr. des Absatzes im File)
- Z über eine zu suchende Zeichenkette
- R über Randausgleich, wenn der Rechner keine automatische Trennung findet.

Die Eintritte Z und R werden in Abschnitt 3.3 genauer behandelt.

Im Textmode gehen alle eingegebenen Zeichen direkt auf den Bildschirm und zugleich in den Filespeicher. Die Texteingabe erfolgt, ohne das Ende der Zeile auf dem Bildschirm zu beachten. Der gesamte Bildschirm ist also als eine (unendlich) lange Zeile aufzufassen, die sogar über den Bildschirm unten und oben hinausreicht. Diese Teile sind während der Arbeit im Filespeicher vorhanden, aber momentan wegen der begrenzten Größe des Bildschirms nicht sichtbar. Ein Absatz im üblichen Sinne ist ein in sich fest verbundenes Textstück, das beliebig viele Textzeichen, also auch Interpunktionszeichen, Leerzeichen usw. enthalten kann. Der oben stehende Absatz von "Die Texteingabe..." bis "...nicht sichtbar..." ist als eine einzige lange Zeile fortlaufend



eingetragen worden, ohne auf die durch den Bildschirm vorhandene Zeilenbegrenzung zu achten. In dieser Form existiert er als ein fester Speicherteil. Erst später wird für diesen Textteil (den Absatz) über die Ausgabe die endgültige Zeilenformatierung automatisch erzeugt. Mit dieser Zeilenzerlegung hat der Schreiber des Textes zunächst nichts zu tun. Er schreibt ohne Silbentrennung über das jeweilige Zeilenende des Bildschirms hinweg. Wenn er am unteren Bildschirmrand angelangt ist, rückt der ganze Text einfach automatisch um eine Zeile nach oben und zum Weiterschreiben steht nunmehr immer die jeweils unterste Zeile des Bildfensters neu zur Verfügung. Ein Absatz, das Ende einer "langen" Zeile, entsteht durch Betätigung der Taste

ENTER.

Der neue Absatz beginnt immer mit einer neuen Zeile auf dem Bildschirm. Hier kann sofort mit dem Text oder, wenn leerzeilen gewünscht sind, mit einem vorangestellten Leerzeichen und ENTER begonnen werden.

### 3.2.1. Sonderzeichen

Im Textmode sind über die Tasten F1 bis F6 sowie Shift F1 bis Shift F6 einige wichtige Sonderzeichen zu erreichen. Es gilt:

Taste	Bedeutung	Bedeutung bei Shift + Taste
F1	ä	Ä
F2	ö	Ö
F3	ü	Ü
F4	ß	□
F5	↘	◇
F6	↓	△

Es erweist sich als günstig, wenn Sie auf Ihrem Rechner diese Tasten entsprechend mit einem Aufkleber über den Tasten kennzeichnen.

Besondere Bedeutung haben die Sonderzeichen **↘** und **◇** im Zusammenhang mit dem Veränderungsmodus. Sie werden in Abschnitt 2.4 behandelt. Darüber hinaus bedeutet:

- Tabulator
- ↘ Text rechtsbündig setzen
- △ Symmetrieren von Text

(Hierzu ausführlicher Abschnitt 3.2.8 bis 3.2.10.)

### 3.2.2. Blättern im Text

Wenn ein umfangreicher Text vorliegt, kann jeweils nur ein bestimmter Ausschnitt daraus auf dem Bildschirm sichtbar sein. Zu den unsichtbaren Textteilen gelangt man durch Vor- und Rückblättern. Hierfür existieren drei Befehle:

Shift ↑ blättert eine (Bildschirm-)Seite zurück  
Shift ↓ blättert eine (Bildschirm-)Seite vor  
↑ holt, wenn der Cursor am oberen Bildschirmrand steht, die zuvorstehende Zeile zurück.

Mit diesen Befehlen erreichen Sie sehr schnell jede beliebige Textstelle im File. Die Befehle sind so gestaltet, daß beim (Bildschirm-)seitenweisen bzw. zeilenweisen Rückblättern der Anfang des Textfile erreicht wird. Das Vorblättern geht so weit hinter das Ende des Textfile, daß sofort weiterer Text angefügt werden kann.

Achtung! Die Taste ↓ (Cursor nach unten) ist nicht mit dem Veränderungspfeil T zu Verwechseln.

### 3.2.3. Erzeugen von Leerzeilen

Durch Betätigen der Tasten ENTER bzw. ↓ führen Sie den Cursor zur nächsten Bildschirmzeile abwärts (bei ENTER immer auf den Zeilenanfang). Falls dort kein Text steht, erzeugen Sie scheinbar eine Leerzeile. Diese scheinbare Leerzeile wird jedoch nicht in den Filespeicher übernommen. Überzeugen Sie sich davon, indem Sie z. B. folgenden Text nachschreiben:

```
DIES IST EINE ZEILE <ENTER>
<ENTER>
<ENTER>
DIES STEHT 2 ZEILEN TIEFER <ENTER>
INDE <ENTER>
```

Hierbei soll <ENTER> bedeuten, daß Sie diese Taste betätigt haben. Wenn Sie jetzt mit BRK den Textmode verlassen und anschließend mit dem Befehl T wieder zurückkehren, sind die beiden Leerzeilen verschwunden. (Diese Methode hat Vorteile, z. B. für das Editieren mit Delete.)

Wichtig ist, daß beständige Leerzeilen werden erzeugt, indem am Anfang einer Bildschirmzeile durch Betätigen der Leertaste zunächst ein Leerzeichen (Space) geschrieben wird. Dann ist, nachdem der Cursor um eine Stelle nach rechts gerückt ist, <ENTER> auszuführen.

Sie können sich auch relativ schnell bezüglich Leerzeilen durch Umblättern im File überzeugen (siehe hierzu den folgenden Abschnitt).

#### Hinweis für Kenner:

Der KC-85 stellt auf dem Bildschirm zwei Zeichen unsichtbar dar, und zwar die Codenummern 0H und 20H. Wenn der Bildschirm gelöscht ist, enthält er nur die Zeichen 0 und diese werden nicht in das File übernommen. Dagegen geht das ebenfalls unsichtbare Zeichen 20H in das File ein. Deshalb zu Beginn einer gewünschten Leerzeile ein Space-Zeichen (20H) eingeben.

### 3.2.4. Bewegen im File

Das Bewegen des Cursors im File ist für das Editieren von Texten notwendig. Jedes Editieren von Texten erfolgt nämlich in drei Stufen:

- Durch Blättern im Text wird die zu verändernde Stelle auf dem Bildschirm sichtbar gemacht.
- Der Cursor wird an die zu ändernde Textstelle geführt.
- Die Veränderung wird ausgeführt.

Dies kann auf vielfältige Art erfolgen; hier wird nur die einfachste beschrieben. Weitere Möglichkeiten bestehen mit DELETE und INSERT (Abschnitt 3.2.7; 3.2.8) und Veränderungscode (Abschnitt 3.4).

Für die Cursorbewegung auf dem Bildschirm existieren folgende Tastenbetätigungen:

- ↓ Cursor eine Zeile tiefer
  - ↑ Cursor eine Zeile höher
  - Cursor nach rechts
  - ← Cursor nach links
  - Home Cursor in die linke obere Ecke
  - Shift+← Cursor an den Zeilenbeginn
- Damit ist jeder Punkt auf der Bildfläche schnell erreichbar. Insbesondere ist noch zu beachten, daß durch längeres Drücken der ersten vier Tasten der Befehl automatisch wiederholt wird. Weiterhin ist zu beachten, daß die ersten vier Befehle an den Bildschirmgrenzen einige Besonderheiten bewirken:
- ↓ erzeugt am unteren Bildrand Pseudoleerzeilen (siehe Abschnitt 3.2.3),
  - ↑ holt am oberen Bildrand die vorherstehende Zeile ins Bildfenster und läßt die unterste vom Bildschirm verschwinden.
  - ← Diese Tasten führen innerhalb eines Absatzes automatisch in die Zeile zuvor oder danach.

Achtung! ENTER sollte nicht zum Positionieren des Cursors verwendet werden, da an dieser Stelle eventuell ein Absatz entstehen könnte, der das Textfile ungewollt verfälscht.

### 3.2.5. Einfaches Editieren

Das einfache Editieren von Texten erfolgt dadurch, daß über den vorhandenen Text der neue geschrieben wird. Der Text, der zuletzt auf dem Bildschirm steht, geht automatisch in das File ein, unabhängig davon,

- ob die Änderung als obere Zeile beim Weiterschreiben verschwindet,
- ob im Text geblättert wird oder
- ob der Textmode mit BRK bzw. ↵ verlassen wird.

Mittels

CLR

wird das Zeichen, auf dem der Cursor steht, durch Space

ersetzt und der Cursor rückt eine Position nach links. Bei der Eingabe bewirkt CLR das Löschen des zuletzt eingegebenen Zeichens.

Vorsicht mit ENTER, wenn der Cursor im Textbereich steht!

### 3.2.6. Einfügen von Text

Um in einem File vor allem kleine Textzeilen einzufügen, ist hierfür zunächst einmal an der entsprechenden Stelle Freiraum zu schaffen. Hierzu dient die INSERT-Taste. Mit ihrer Betätigung wird dort, wo der Cursor steht ein Leerzeichen erzeugt und der gesamte rechts stehende Textteil eines Absatzes, also auch die zugehörigen folgenden Bildschirmzeilen, werden um eine Position nach rechts verschoben. Auf diese Weise sind prinzipiell beliebig viele Leerstellen im Text zu erzeugen, auf denen dann der gewünschte Text eingefügt werden kann.

Dem Einfügen von vielen Leerstellen sind einige Grenzen gesetzt:

- Ein Absatzende läßt sich nur bis maximal an das Ende einer Bildschirmzeile verschieben. Darüber hinaus gehen die letzten Zeichen verloren. Dies ist unmittelbar auf dem Bildschirm zu verfolgen. Es läßt sich vermeiden, indem zuvor eine Leerzeile mit Space erzeugt wird, die rückwärts bis an das Ende des Absatzes reicht.
- Ist der Absatz, in dem etwas eingefügt werden soll, hinter der Einfügung länger als auf dem Bildschirm sichtbar ist, so erfolgt dagegen automatisch die richtige Übernahme des Textes in das File.
- In diesen beiden kritischen Fällen ist es jedoch günstiger, Text über den Veränderungsmodus gemäß Abschnitt 3.4 einzufügen.

### 3.2.7. Entfernen von Text

Sollen kleine Textteile gestrichen werden, so existiert dafür die DELETE-Taste. Mit ihrer Betätigung wird das jeweils unter dem Cursor stehende Zeichen entfernt und von rechts rücken alle Zeichen bis zum Absatzende um eine Position auf. In dieser Weise lassen sich im Prinzip auch nahezu beliebig viele Zeichen entfernen.

Für das Entfernen von Text besteht auch die Möglichkeit, Shift DELETE

zu verwenden. Dann werden alle Zeichen vom Zeilenbeginn an bis einschließlich dem Zeichen unter dem Cursor gelöscht, die Zeichen hinter dem Cursor rücken bis zum Zeilenanfang auf. Ähnlich wie bei INSERT (Abschnitt 3.2.6) besteht hier auch die Problematik an den Absatzgrenzen und für das Entfernen umfangreicher Textteile ist der Veränderungsmodus (Abschnitt 3.1) geeigneter. In jedem Fall sind aber auch hier mögliche Fehler sofort auf dem Bildschirm sichtbar.

### 3.2.8. Textsymmetrierungen

Verschiedene Textteile sollen bei einem Ausdruck genau auf der Mitte der Zeile erscheinen. Da aber bei der Texteingabe die Zeilenbreite nicht bekannt ist, und zuweilen auch der gleiche Text in unterschiedlicher Zeilenbreite ausgedruckt werden muß, war es notwendig, eine Sonderform für das Symmetrieren zu finden. Hierzu dient das Zeichen  $\Delta$

am Anfang einer Bildschirmzeile. Der dann folgende Text bis <ENTER> wird, sofern das möglich ist, symmetrisch ausgedruckt. Es werden automatisch rechts und links vom Zeilenrand gleichviel Leerzeichen erzeugt. Ein Text, der wie folgt auf dem Bildschirm steht:

$\Delta$ Mitte <ENTER>

wird beim Ausdruck für jede zulässige Zeilenbreite genau in der Mitte der Zeile positioniert. Auf dem Bildschirm ist dies nur durch das Sonderzeichen  $\Delta$  erkennbar, aber nicht sichtbar zu machen.

Achtung! Ist der hinter  $\Delta$  stehende Text länger als die später formatierte Zeilenbreite, erfolgt bei R (Randsuche), P (Teildruck) und E (Ausdruck) kurzzeitig eine ERROR-Anzeige und danach geht das System an der entsprechenden Stelle in den Textmode über.

### 3.2.9. Rechtsbündiger Text

Vielfach sind ausgewählte Textteile alleinstehend am rechten Rand zu plazieren. Dies ist u. a. bei dem Datum und formelnumerierungen der Fall. Erreicht wird dieses Ziel mit dem Sonderzeichen \. Dieses Zeichen wird nicht gedruckt. Hier werden so viele Leerzeichen (Space) eingefügt, bis der rechts vom Zeichen stehende Text den rechten Textrand genau erreicht. Deshalb wird auch ein Text:

links\rechts

so ausgedruckt, daß das Wort "links" am linken Rand und das Wort "rechts" am rechten Rand bündig erscheint.

Für Zeilen dieser Art, die mit einem Space länger als die formatierte Zeile sind, erfolgt - wie bei den Symmetrierungen - Fehleranzeige.

### 3.2.10. Tabulator

Zuweilen ist es notwendig, Tabellen und Texte auf bestimmten Positionen auszudrucken. Zu diesem Zweck stellt TEXOR einen relativ einfachen Tabulator bereit, der durch das Zeichen  $\square$

begeben ist und sich in der Wirkung ähnlich wie das Symmetrierzeichen verhält. Er ersetzt immer 1 bis 3

Leerzeichen und steuert in dieser Weise die Positionen  
4,8,12,16,20,24 usw.

an. Zwei hintereinanderstehende Tabulatorzeichen, bewirken also 5 bis 7 Leerzeichen, drei Zeichen folglich 8 bis 11 Leerzeichen usw. Das bedeutet, daß ein Tabulator den auf ihn folgenden Text auf die nächste der durch 4 teilbaren Positionen legt. Jedes zusätzlich angefügte Tabulatorzeichen entspricht 4 Leerzeichen (Space). Wird eine Tabulatorzeile in ihrer Gesamtlänge größer, als die Zeilenformatierung vorsieht, erfolgt eine ERROR-Anzeige bei R, P und E.

Die Wirkung der Zeichen \, Δ, □ ist nicht im Textmode, sondern erst bei Ausdruck sichtbar. Treten diese Zeichen in einer Zeile gemischt auf, so bestimmt das erste Zeichen den Charakter der Zeile und die anderen Zeichen werden nicht als Steuerzeichen interpretiert, sondern normal ausgedruckt.

### 3.2.10. Die beiden Textfenster

Für den Textmode existieren zwei Textfenster, ein großes und ein kleines. Sie haben unterschiedliche Eigenschaften und können definiert eingestellt werden:

n,t,b,g führen zum kleinen Textfenster

N,T,B,G führen zum großen Textfenster.

Das kleine Textfenster sollte stets zum Eingeben (erstmaligen Schreiben) von Text verwendet werden. Das große Bildfenster ist dagegen für Textbearbeitung, also z. B. zum Editieren, von Vorteil.

Das kleine Textfenster ist so gewählt, daß auf dem Bildschirm die selteneren Befehle sichtbar stehenbleiben. Es besitzt weiter für schnelles Schreiben den Vorteil, daß - wenn man bei der Eingabe am unteren Rand des Bildschirms angelangt ist, der gesamte Text um eine Zeile hochgerückt werden muß (scrolling). Dies dauert ca. 1/2 Sekunde und während dieser Zeit übernimmt der Rechner keine Tasteneingabe. Das bedeutet, daß entsprechende Tasteneingaben verlorengehen. Mit dem kleinen Fenster wird die Zeit auf ca. 1/10 Sekunde reduziert, so daß kaum eine Eingabe während dieser Zeit verlorengehen kann. Daher für die Eingabe das kleine Textfenster verwenden.

Das große Fenster zeigt dagegen bedeutend mehr Text an. Für das Editieren besteht daher hier ein besserer Überblick, insbesondere werden die in den Abschnitten 3.2.6 und 3.2.7 beschriebenen Grenzen für die Anwendung von INSERT und DELETE weiter hinausgeschoben.

### 3.3. Ausgabe von Text

Sinn und Zweck eines Textverarbeitungssystems ist schließlich das Ausschreiben oder Ausdrucken des im File gespeicherten Textes. Hierbei werden in der Regel unterschiedliche Formatierungen verlangt. Sie betreffen u. a.  
die maximale Länge einer Zeile;

den rechten Rand:

- Flattersatz - rechter Rand unregelmäßig, wie es sich durch die Trennung ergibt,
- Blocksatz - durch Einfügen von Leerzeichen (Space) wird der rechte Rand glatt gemacht.
- die Anzahl der Zeilen pro Seite;
- das Verhalten des Druckers am Ende der Seite.

Diese und weitere Funktionen sind mit TEXOR bei der Ausgabe steuerbar.

Der Eintritt in den Ausgabemodus kann auf drei Arten erfolgen:

- R, r Randsuche
- E, e Ausgabe
- ⌘P, ⌘P Teiltextausgabe

Dabei erfolgt bei e, ⌘P die Ausgabe im Flattersatz und bei E und ⌘P im Blocksatz.

### 3.3.1. Randsuche

Die Randsuche sollte immer vor dem Ausdruck des Textes verwendet werden. Sie verlangt als erstes die Eingabe der Formatierungsgröße für die Zeilenlänge:

Zeilenlänge:

Sie kann beliebig im Bereich von 10 bis 255 Zeichen/Zeile gewählt werden. Eingaben größerer oder kleinerer Werte führen zur Fehleranzeige.

Nach Eingabe dieser Zahl wird der Text auf automatische Zeilenformatierung geprüft. Ist dies möglich, so wird nach kurzer Zeit wieder der Befehlsmodus erreicht. Zwischendurch blinken im Textfenster nur kurz gewisse Textteile auf.

Stößt dagegen die Randsuche auf Probleme, die eine Worttrennung erforderlich machen, so bleibt im kleinen Bildfenster die zu trennende Stelle mit einem Trennvorschlag stehen. Diese vom halbautomatischen Trennalgorithmus vorgeschlagene Stelle wird durch ein ~ angezeigt. Manuell muß nun diese Stelle, falls man einverstanden ist, mit

ENTER

bestätigt werden. Falls die Trennung nicht richtig ist, kann sie mittels ← und → nach rechts und links in gewissen Grenzen verschoben und anschließend mit ENTER bestätigt werden. Nach ENTER läuft die Randsuche weiter und es wird das Ende erreicht, angezeigt durch:

Befehl:

oder es ist erneut eine halbautomatische Silbentrennung zu realisieren.

Bemerkung: Das Zeichen ~ ist ein "weicher" Bindestrich. Es wird an der gewählten Stelle in den Text eingefügt und ist im



Textmode auch als solches zu erkennen. Ausgedruckt wird es wie ein normaler Bindestrich. Es ist aber aus den zuvor genannten Gründen nur für eine feste Zeilenlänge nützlich. Deshalb gibt es den Befehl

L,

der alle "weichen" Bindestriche und weitere Hilfszeichen wieder aus dem Text entfernt.

Für die Formatierung eines Textes ist es folglich sinnvoll, zunächst mit L alle eventuell störenden Sonderzeichen zu löschen und anschließend mit R den Text neu für die Ausgabeformatierung vorzubereiten.

### 3.3.2. Textausgabe

Die vollständige Textausgabe wird, nachdem über R oder r die Zeilenbreite formatiert wurde über E bzw. e realisiert.

E erzeugt einen Blocksatz mit glattem rechten Rand,

e erzeugt einen Flattersatz,

ohne Einfügung von zusätzlichen Leerzeichen.

Nach der Eingabe von E bzw. e wird zunächst die Papierformatierung abgefragt:

Zeilen/Seite:

Hieran schließt sich die Abfrage

Druck J/N:

Für die Druckerausgabe ist sie mit J (groß) zu beantworten. Alle anderen Eingaben führen wieder zur Videoausgabe. Erst nach dieser Eingabe wird der Drucker oder die Schreibmaschine für die Textausgabe angesteuert. Gewisse Textteile (alle jene, wo die Zeilenformatierung und eventuell der rechte Randausgleich wirksam sind) erscheinen dabei zeitweilig auf der oder den obersten Zeilen des Bildschirms. Sobald die eingegebene Zeilenzahl je Seite ausgedruckt wurde, wartet der Rechner auf eine Eingabe. Hier kann ein neues Blatt eingelegt oder bei Rollpapier ein Abstand manuell eingestellt werden.

Mit jeder Taste (außer BRK) wird der Druckvorgang fortgesetzt. Mit BRK wird der Druckvorgang beendet und der Befehlsmode erreicht. Wenn der gesamte Text ausgedruckt ist, kehrt der Rechner automatisch in den Befehlsmode zurück.

### 3.3.3. Teiltextausgabe

Soll nur ein Stück Text ausgegeben werden, z. B. um die Symmetrierung oder den Tabulator zu kontrollieren, so ist wie folgt zu verfahren:

Vor dem Beginn der gewünschten Textausgabe ist mit INSERT das

Hilfszeichen  $\diamond$  einzufügen. Dann ist mit dem Cursor an das gewünschte Textende zu gehen und dort ist nach INSERT  $\uparrow$  einzugeben. Die Eingabe von p oder P bewirkt dann die Ausgabe des so gekennzeichneten Textteiles wie zuvor in Abschnitt 3.3.2 beschrieben. Entsprechend der Zeilenformatierung wird dabei meist noch ein kurzes Stück hinter  $\uparrow$  ausgedruckt. Die beiden Sonderzeichen  $\diamond$  und  $\uparrow$  sind auch noch nach dem Druck im File wiederzufinden. Hierdurch ist ein erneuter Ausdruck leicht möglich. Der Cursor wird genau auf das Zeichen  $\uparrow$  bewegt, das Zeichen wird erneut, gefolgt von p oder P, eingegeben. Damit der Text wieder frei von den beiden Sonderzeichen wird, ist abschließend der Befehl L einzugeben.

### 3.4. Veränderungsmodus

Der Veränderungsmodus wird aus dem Textmode durch Eingabe des Sonderzeichens  $\uparrow$  erreicht. In den meisten Fällen sind dafür aber vorher kleine Kennzeichnungen mit der Raute vorzunehmen. Der Veränderungsmodus hält folgende Möglichkeiten bereit:

- m: Verlagern (move) von Textteilen an andere Stellen
- c: Kopieren (Doppeln; copy) von Textteilen zusätzlich an andere Stellen
- d: Löschen (delete) von Textteilen
- p: Drucken (print) von Textteilen
- a: Erzeugen von Absätzen
- v: Vernichten von Absätzen (Verbinden von Absätzen)

Der Befehlsablauf für p ist bereits in Abschnitt 3.3.3 behandelt und wird daher hier nicht mehr berücksichtigt.

Generell erfolgt die Arbeit im Veränderungsmodus in drei Stufen:

1. Vorbereiten durch Einfügen von 1 bis 2 Rauten  $\diamond$
2. Setzen des Sonderzeichens  $\uparrow$
3. Einleiten und Ausführen des Veränderungsmodus durch a, c, d oder v. Dabei sind große und kleine Buchstaben gleichwertig (Ausnahme p, hier nicht betrachtet!). Alle anderen Eingaben führen zur Fehleranzeige.

Bemerkung: Für die Auswahlzeichen a, c, d, p oder v braucht kein Leerraum erzeugt zu werden.

#### 3.4.1. Erzeugen von Absätzen

Hierbei entfällt das Vorbereiten. Der Cursor wird an die Stelle geführt, wo ein Absatz entstehen soll. Mit INSERT wird Platz für das Sonderzeichen geschaffen. Es wird  $\uparrow$  eingegeben, gefolgt von a, und der Absatz ist nach Bruchteilen von Sekunden auf dem Bildschirm sichtbar.

#### 3.4.2. Entfernen von Absätzen

Unmittelbar hinter dem Textende, wo der Absatz verschwinden soll, wird das Sonderzeichen, gefolgt von einem v, eingegeben. Bruchteile von Sekunden später ist die Wirkung auf dem Bildschirm zu sehen.

Bemerkung: Oft soll zwischen den zu verbindenden Textteilen ein Leerzeichen stehen. Dann ist hinter dem Text zunächst dieses Leerzeichen, gefolgt von  $\downarrow$  und v, einzugeben.

### 3.4.3. Verlagern von Textteilen

Der Textteil, der an eine andere Stelle des Textfiles zu verlagern ist, muß in der Vorbereitung an Anfang und Ende durch Rauten  $\diamond$  begrenzt werden. Für das Einfügen der Rauten ist zuvor mittels INSERT genau ein Leerzeichen (Space) zu erzeugen. Dann wird die Stelle aufgesucht, wo der Text hingebacht werden soll. Mittels INSERT (leerstelle schaffen), Sonderzeichen  $\downarrow$  und m wird der vorher durch Rauten begrenzte Text hier eingefügt.

Auch in diesem Fall ist es günstig, sowohl darauf zu achten, daß an der Stelle, wo der Text entfernt wird, die richtige Anzahl Leerzeichen verbleibt, als auch darauf, daß an der Einfügestelle genau die gewünschten Leerzeichen entstehen. In jedem anderen Fall ist aber auch die Korrektur über INSERT bzw. DELETE hinterher leicht möglich.

### 3.4.4. Doppeln von Textteilen

Das Doppeln erfolgt analog zum zuvor beschriebenen Verlagern von Textteilen, lediglich ist statt m nach dem Sonderzeichen jetzt c einzugeben. Der gekennzeichnete Textteil ist dann sowohl an der alten als auch an der neuen Stelle vorhanden. Dieses Verfahren ist besonders effektiv, wenn Grundstrukturen vervielfältigt werden sollen. Vervielfältigungen in größerer Zahl sind durch eine schrittweise Erhöhung auf das 2-, 4-, 8-, 16-fache günstig zu erhalten. Dazu ist durch Rauten zunächst der zu doppelnde Text zu gruppieren, also z. B.:

$\diamond$  Adresse  $\downarrow$  c

Daraus entsteht:

Adresse Adresse

Mit:

$\diamond$  Adresse Adresse  $\downarrow$  c

folgt:

Adresse Adresse Adresse Adresse

so kann fortgefahren werden, natürlich auch mit komplizierteren Strukturen.

### 3.4.5. Probleme des Veränderungsmodi

Wenn genau die Vorschrift eingehalten wird, läuft der Veränderungsmodi in der Regel völlig problemlos. Da mit ihm

Über tiefgreifende Änderungen im File möglich sind, sollte immer vorher eine Sicherheitskopie des Textes angefertigt werden.

Sollen sehr große Textteile verlagert oder kopiert werden, so kann der begrenzte Speicherbereich Probleme bereiten. Dies wird auf dem Bildschirm mit:

"Speicher zu klein"

angezeigt. Dann sollte der Versuch mit kleineren, geteilten Abschnitten vorgenommen werden. Der Grund für dieses Verhalten liegt darin, daß zunächst der zu verlagernde bzw. zu doppelnde Textteil zusätzlich im Speicher abgelegt werden muß, bevor die Umorganisation erfolgen kann.

Werden bei der Arbeit im Veränderungsmodus Fehler begangen, z. B. die Rauten nicht oder nur teilweise gesetzt, so erfolgt Fehleranzeige. Dabei bleiben aber alle Sonderzeichen im Text erhalten. Dies hat Vor- und Nachteile:

Vorteile, wenn nur Ergänzungen bzw. leichte Korrekturen nötig sind.

Nachteile, wenn die Fehlersuche kompliziert ist und nicht sofort alle in Frage kommenden Stellen auffindig gemacht werden können, wenn z. B. im Text 3 Rauten oder 2 -Zeichen stehen. Dann kann ein falscher Textteil gelöscht oder verlagert bzw. gedoppelt werden. Hier ist also mit großer Sorgfalt zu arbeiten. Es ist daher zu empfehlen, im Zweifelsfall vorher alle Sonderzeichen und mit dem Befehl `W` aus dem Text zu entfernen und erst danach erneut den Veränderungsmodus aufzurufen.

## 2.5. Zeichensuche

Die Zeichensuche wird aus dem Befehlsmodus erreicht. Je nach Eingabe von `z` oder `Z` entsteht für die Anzeige der gefundenen Zeichen ein kleines oder großes Bildfenster, also mit wenig oder viel Text. Nach der Eingabe von `Z` bzw. `z` erscheint zunächst die Abfrage:

Zeichenkette, (■=dont care):

Hier ist die zu suchende Zeichenkette einzugeben. Es sollen z. B. die Textstellen gesucht werden, wo das Wort "Oxid" steht. Dann kann eingegeben werden:

oxid, Oxid oder OXID.

Die Groß-Kleinschreibung ist nicht wichtig, da sie unberücksichtigt bleibt. Gefunden wird diese Zeichenkette an beliebiger Stelle im Text, also z. B. auch für Wörter wie Monoxid, Oxidation, oxidiert usw.

Bei der Eingabe werden alle Befehlstasten, auch die Cursorfunktionen, unterdrückt, d. h., sie haben keine Wirkung. Ausnahmen sind:

CLR zum Löschen eines falsch eingegebenen Zeichens,

ENTER zum Beenden der Eingabe.

Wenn man nicht weiß, ob eventuell auch irgendwo "Oxyd" im Text steht und diese Stellen mit angezeigt werden sollen, ist das font-care-Zeichen nützlich. Es wird mit Shift und Leertaste erzeugt. Es bedeutet, daß hier ein beliebiges Zeichen bei der Suche stehen darf. Es werden also bei Eingabe von

"ox d"

z. B. gefunden:

oxid, Oxid, OXID, oxyd, Oxyd, OXYD, Oxad, OXTD, OX7D, OX D oder OXID.

Natürlich wird auch gefunden:

"die Box des Hundes" oder "Floxadit".

Das font-care-Zeichen ist sowohl nützlich (für unbekannte Schreibweisen) als auch gefährlich, indem unerwartete Texte gefunden werden.

Nach Abschluß der Eingabe der Zeichenkette mit ENTER beginnt der Suchvorgang. Er kann auf drei Arten verlaufen:

1. Es wird keine Zeichenkette der gewünschten Art im Text gefunden. Dann kehrt der Rechner nach kurzer Zeit in den Befehlsmode zurück.

2. Es wird eine Kette gefunden. Dann wird die Anzeige bei etwa einer Zeile vor der gefundenen Kette begonnen und sie endet, wenn das Bildfenster (groß oder klein) gefüllt ist bzw. am Textende. Da der Rechner sich im Textmode befindet, kann auf dem Bildfenster in gewohnter Weise editiert, umgeblättert usw., aber auch mit BRK, Stop oder CLR in den Befehlsmode zurückgekehrt werden.

3. In vielen Fällen wird die gesuchte Zeichenkette aber nicht nur an der angezeigten Textstelle, die ja die erste gefundene im Text ist, stehen, sondern auch noch später. Dafür ist es möglich, im Text weitersuchen zu lassen. Zu diesem Zweck ist die Tastenkombination

Shift CLR

zu benutzen. Sie kann so oft wiederholt werden, bis das Textende erreicht ist. Danach kehrt der Rechner automatisch in den Befehlsmode zurück.

## 4. SORED

SORED wird mit S oder s aus dem Start der Textverarbeitung TEXVE erreicht. Nach ca. einer Sekunde erscheint das Menu:

\*\*\*\*\* SORED \*\*\*\*\*

Eingaben	Funktionen	- Ausgaben
K:Kopf	S:Sortieren	A:Ausgabe
T:Text	Z:Zeichensuche	H:Menu
:Verbinde	U:Suche ohne	F:Datei-
BRK:Beende	Zeichenkette	werte
	E:Editiere	/

a,u,z ohne Zeilen-Nr.  
BRK:Hauptmenu

Editmode  
INS, DEL, CLR usw..  
ENTER:Übernahme  
BRK:Beende

Befehl:■

Es wird damit einer der folgenden 11 Befehle erwartet:

A, D, E, F, H, K, S, T, U, Z oder BRK.

Bis auf A, U und Z sind große und kleine Buchstaben gleichwertig. Alle anderen Eingaben werden mit

Befehl:ERROR

Befehl:■

beantwortet. D = Entfernen von Zeilen (delete) ist im Menu nicht enthalten.

### 4.1. Drei Besonderheiten

#### 4.1.1. Bedeutung der Quasizeile

SORED ist eine spezielle und für Kleinstrechner entwickelte Kombination von Datenbanksystem und Sortiertechniken. Dies wurde durch die konsequente Einführung und Anwendung des Sonderzeichens erreicht.

Jeder Sortieralgorithmus verlangt eine spezielle Struktur der zu sortierenden Zeilen, Begriffe usw. Entsprechend den Eigenschaften des KC 85 wurde die Grundstruktur so gewählt, daß eine Zeile die Länge zwischen 0 und 32 Zeichen enthalten kann. Dies würde im Prinzip für die meisten Anwendungen ausreichen. Aber bereits bei Adressen ist eine mehrzeilige

Struktur für einen Sortierbegriff günstig. So entstand die Idee der Quasizeile. Sie setzt sich aus prinzipiell beliebig vielen Einzelzeilen zusammen, die aber beim Sortieren weder in der Reihenfolge geändert noch auseinandergerissen werden können, sondern stets als Einheit, nämlich als Quasizeile, erscheinen. Die dazu notwendige feste Verbindung zwischen diesen Zeilen einer Quasizeile wird durch das Verbindezeichen am Ende der Zeilen einer Quasizeile erreicht. Lediglich in der letzten Zeile einer Quasizeile fehlt dieses Zeichen. So wird z. B. eine Adresse mit Telefon-Nr. wie folgt eingegeben:

Max Osternau  
1086 Berlin  
Müllstr. 83  
Tel. 27 886 53

Beim Ausdruck und der Anzeige können die Verbindezeichen unterdrückt werden.

Sortiert wird in diesem Fall, immer in der Reihenfolge der Zeichen beginnend, bei der ersten Filezeile. So würde sich z. B. die folgende Reihenfolge ergeben:

Anton Globau ...  
Max Osternau ...  
Maxim Grüns ...  
Moritz Bergblau ...

#### 4.1.2. Der Kopfteil

Bei Sortieralgorithmen ist es günstig, einen Kopfteil zu definieren, der vom Sortiervorgang immer unberührt am Anfang bleibt und sozusagen das File kennzeichnet. In ihm können dann u. a. eingetragen werden:

Name des File  
Datum der letzten Bearbeitung/Veränderung  
Allgemein gültige Kommentare, Bedeutungen usw.

Aus diesem Grunde wurde ein spezieller Kopfteil mit frei wählbarer Anzahl von Quasizeilen für SORED eingeführt. Zugleich wurde dieser Kopfteil zur Eröffnung des File gewählt, denn bei einem vorhandenen File ist er nur zu editieren, aber nicht neu zu definieren. Um dennoch eine Sicherheit vor Zerstörung eines vorhandenen File zu bekommen, wurde eine zusätzliche Sicherheit eingebaut. Wird nämlich der Befehl K oder k = Kopf eingegeben, so entsteht das folgende Bild

Kopf ■

und nur zwei Tastenbetätigungen führen aus diesem Zustand heraus (alle anderen Tasten bleiben wirkungslos):

1. BRK führt zu dem Ergebnis

Kopf  
Befehl ■

Hier wird also der eventuell versehentlich gegebene Befehl K



wieder aufgehoben und alles, was im File existiert, besteht unverändert weiter.

1. Shift ~~Home~~ führt zu den Ergebnis

Kopf

Jetzt sind alle Zeiger (Pointer) des vorhandenen File zurückgesetzt. Das bedeutet, das alte File ist scheinbar gelöscht und es wird ein neues File mit dem Kopf begonnen. (Diese Eingabe entspricht also dem Befehl N bei TEXOR.) Mit dem Diagnosesystem D ist unter erheblichem Aufwand auch in diesem Zustand das alte File noch zu restaurieren (siehe Abschnitt 6).

#### 4.1.3. Normale und reziproke Zeichensuche

Im Textfile von SORED können Zeichenketten gesucht werden. Dabei können nach Wunsch jene Quasizeilen angezeigt werden, die diese Zeichenkette enthalten (Z) bzw. jene, die diese Zeichenkette nicht enthalten (U). Dies hat vielfältige Vorteile. Hier ein Beispiel: Für eine Tagung sind alle Teilnehmer als File zusammengestellt. In einer Extra-Teilzeile der Quasizeile können jetzt Positionen für Zeichen eingerichtet werden, die z. B. folgendes bedeuten:

Teilnehmer hält Vortrag

Teilnehmer ist Chairman

Teilnehmer hat Beitrag bezahlt

usw. Jetzt ist es relativ leicht, z. B. folgende Teilnehmergruppen getrennt herauszusuchen zu lassen: Teilnehmer ohne Vortrag, Teilnehmer, die noch nicht Tagungsbeitrag bezahlt haben, Chairmen usw. Auch Kombinationen davon sind z. T. möglich.

#### 4.2. Erzeugung des File

Ein File wird mittels

K oder k

und gemäß Abschnitt 4.1.2 eröffnet. Der Kopfteil kann dann als eine oder mehrere Quasizeilen eingegeben werden. Hier hat der Anwender große Freiheit. Jede Teilzeile einer Quasizeile wird mit dem Sonderzeichen ~ beendet, jede Quasizeile selbst dagegen mit ENTER. Der Kopfteil wird insgesamt mit

BRK

abgeschlossen und mit

T oder t

gelangt man dann zur Texteingabe:

Befehl: Text

Damit erfolgt die Eingabe der Textquasizeilen. Sie werden,

renn Textteile verbunden bleiben sollen, wieder mit 7

beendet. Wenn dagegen eine Quasizeile zu Ende ist, erfolgt der Abschluß mit

ENTER

Die Texteingabe wird insgesamt ebenfalls mit

BRK

beendet. Es ist immer wieder möglich, mittels

T oder t

zur Texteingabe zu kommen. Der dann eingegebene Text wird an das Ende des vorhandenen File angehängt.

#### Kommentar 1

Werden in einer Teilzeile mehr als 32 Zeichen versucht einzugeben, so wird automatisch das Verbundzeichen 7 erzeugt und eine neue Zeile begonnen. Diese Maßnahme ist aus Gründen des möglicherweise späteren Editierens notwendig.

#### Kommentar 2

Es ist nicht sinnvoll, ein leeres File ohne Kopf direkt mit T, d. h. Text, zu beginnen. Es kann dann zwar später in einer Quasizeile der Kopfteil mittels Editierens ergänzt werden, aber die so entstehende Filestruktur kann in einigen Spezialfällen zu Irregularitäten in der Filearbeit führen.

#### Kommentar 3

Bei der Eingabe für Kopf und Text sind folgende Korrekturen möglich:

- Eingabe aller Zeichen von TEXOR, einschließlich der Sonderzeichen des Abschnittes 3.2.1. In SORED hat jedoch keines dieser Zeichen eine besondere Bedeutung. Als Sonderzeichen existiert nur 7.

- In einer Teilzeile sind bis zum Abschluß durch 7 oder ENTER zugelassen:

  - Cursor rechts

  - Cursor links

  - DELETE

  - INSERT

Alle anderen Tasten sind ohne Wirkung. Dabei kann mit INSERT die Zeilenlänge von 32 überschritten werden. Hier ist also vom Anwender an die Grenzen des Editierens zu denken. Zum anderen besteht hier als einzige Ausnahme die Möglichkeit, einmal eine längere Teilzeile zu erzeugen.

### 4.3. Einlesen eines vorhandenen File und Fileanzeige

Liegt ein File auf Kassette vor, so wird es im Betriebssystem (Abschnitt 2) mit

>LOAD

eingelesen. Über

>TEXT

und

Ist dann das File verfügbar. Es kann mit T ergänzt werden und mit F können generell die jeweils aktuellen Fileparameter angezeigt werden:

File XXXXXX Byte lang  
noch VVVVVV Byte frei

Kopf ZZZZ Zeilen  
Text TTTT Zeilen

>Kopftext<

Befehl: ■

Der Wert ZZZZ zeigt die Anzahl der im Kopf vorhandenen Quasizeilen und der Wert TTTT die Gesamtzahl aller Quasizeilen des File an. Darunter folgt der vollständige Text des Kopfteiles.

#### 4.4 Anzeige und Druck eines File

Die Anzeige und der Druck von Files bzw. Fileausschnitten beginnt mit der Eingabe von a bzw. A. Hier haben beide etwas unterschiedliche Wirkung. Nach der Eingabe a bzw. A folgt die Abfrage

von Zeile:

Startzeile für die Ausgabe (minimal = 0, maximal = größte vorhandene Zeilenzahl) einzugeben. Eine zu große Zeilenzahl führt zur Anzeige ERROR. Nach der Eingabe der ersten Zeilenzahl wird die Endzeile abgefragt, also

von Zeile:

37

bis:

Hier ist nun minimal dieselbe Zeilenzahl wie für die Startzeile und maximal eine beliebig große Zeilenzahl einzugeben. Wird nämlich hier eine zu große Zeilenzahl eingegeben, so wählt der Rechner automatisch die größtmögliche. Wird eine Zeilenzahl kleiner als die Startzeile eingegeben, erfolgt die Anzeige ERROR. Nach Eingabe der zweiten Zeilenzahl erfolgt die Abfrage

von Zeile:

37

Bis:  
83

Zeilen/Seite:

Hier hat die Eingabe der Bildschirmformatierung zu erfolgen. Es sind Werte von

10 bis 255

zulässig, wobei die Anzahl der ohne Unterbrechung anzuzeigenden bzw. auszudruckenden Quasizeilen + 1 gemeint ist. Wenn die Anzahl der oben eingegebenen Quasizeilen gedruckt bzw. angezeigt ist, wartet die Ausgabe, bis irgendeine Taste betätigt wird. Erst dann wird die Ausgabe fortgesetzt. Lediglich BRK führt sofort zum Ende.

Schließlich folgt die Abfrage, ob die Ausgabe über Bildschirm oder Drucker erfolgen soll:

von Zeile:

37

bis:

83

Zeilen/Seite:

2

Druck J/N:

Nur die Eingabe von J (groß) führt zum Druck, alle anderen zur Videoanzeige.

Die Eingaben für Zeile/Seite und Druck J/N können einfach mit ENTER beantwortet werden. Dann erfolgt die Ausgabe über den Bildschirm mit der zuletzt eingegebenen Zeilenzahl je Seite.

Bei der Ausgabe macht sich nun der Unterschied zwischen a und A bemerkbar. Mit a folgt z. B. die Ausgabe:

Eins

Zwei

Drei

Vier

Während A zu

0004 Eins

0005 Zwei

0006 Drei

0007 Vier

führt. Die Ausgabe bei A enthält also zusätzlich die Nummerierung der Quasizeilen. Darüber hinaus wird bei A (groß) auch das Verbindezeichen dargestellt. Bei a gilt beispielsweise:

Eine

Kuh

hat

Hörner  
und bei A:  
0000 Einer  
FFFF Kuhn  
FFFF hat  
FFFF Hörner

Die zusätzlichen Zeilen einer Quasizeile enthalten also vorn die Fortsetzungszahl FFFF. Die Ausgabe A ist für das deutliche Erkennen der Quasizeilenstruktur und zum Editieren bedeutsam.

#### 4.5. Sortieren

Das Sortieren eines Files erfolgt mit S bzw. s. Dabei wird folgende Reihenfolge der einzelnen Zeichen realisiert:

Space	0	A=a	Q=q	1
!	1	B=b	R=r	□
"	2	C=c	S=s=ß	◇
#	3	D=d	T=t	△
\$	4	E=e	U=u=Ü	↓
%	5	F=f	V=v	~
&	6	G=g	W=w	
'	7	H=h	X=x	
(	8	I=i	Y=y	
)	9	J=j	Z=z	
*	:	K=k		■
+	;	L=l		!
,	<	M=m		┐
-	=	N=n		入
.	>	O=o=ö=ø		—
/	?	P=p		

Für das Alphabet werden also Groß- und Kleinbuchstaben gleichrangig behandelt und die Umlaute sowie ß werden ebenfalls richtig eingeordnet. Für die Interpunktionszeichen, Sonderzeichen und Zahlen ist die Reihenfolge entsprechend den Codewerten realisiert. Bei den Zahlen ist diese Besonderheit besonders zu beachten. So wird z. B. folgende Beispielfolge geordnet:

1  
11  
111  
1110  
1111  
12  
121  
30  
35  
357  
36

Es wird also zunächst nach der ersten Ziffer, dann nach der zweiten, dritten, vierten usw. sortiert. bei Zahlen mit Punkt

bzw. Komma gilt dagegen:

1  
1,03  
11  
11,1  
1110  
2,3  
20,3  
2030

Das Komma bzw. der Punkt besitzen nämlich einen kleineren Codewert. Den kleinsten Codewert für das Sortieren besitzt immer das Leerzeichen (Space).

#### 4.6. Zeichensuche

Die Zeichensuche kann mit Z bzw. z oder U bzw. u eingeleitet werden, und zwar je nachdem, ob alle Quasizeilen angezeigt werden, die eine bestimmte Zeichenkette enthalten (Z, z) oder die Zeichenkette nicht enthalten (U, u). Nach Eingabe von Z, z, U oder u erfolgt die Anzeige:

Zeichenkette, (■=dont care):

■

Für die Bedeutung der einzugebenden Zeichenkette gelten genau die Zusammenhänge von Abschnitt 3.5 bei TEXOR. Die Wirkung der Eingaben von Z, z, U und u ist jedoch sehr unterschiedlich. Es möge im Speicher das kurze File stehen:

Emil  
Anna  
Martha  
Max  
Heinz  
Helmut

Dann folgen bei der Suche nach Zeichenkette "E" (in diesem Fall nur ein Buchstabe) die Anzeigen:

Z	z	U	u
0001 Emil	Emil	0002 Anna	Anna
0005 Heinz	Heinz	0003 Martha	Martha
0006 Helmut	Helmut	0004 Max	Max

Auch für die Suche mit dem dont-care-Zeichen gelten die Aussagen entsprechend Abschnitt 3.5.

Das Zeichensuchen mit großen Buchstaben hat wieder vor allem für das Editieren große Bedeutung. Sollen in einem File z. B. alle Begriffe "Floppy-disc" einheitlich geschrieben werden und wurden zuvor die Varianten "Floppy Disk - Floppy - Floppydisc" u. ä. verwendet, so erhält man mit der Zeichensuche Z "loppy" alle entsprechenden Zeichenketten mit Zeilen-Nr. und kann sie gleichzeitig einheitlich auf dem Bildschirm editieren.

## 1.7. Editieren

Das Editieren erfolgt in fünf Etappen:

1. Sind zunächst die zu editierenden Zeilen mit Zeilen-Nr. auf den Bildschirm zu bringen. Dies kann durch A, Z oder U erfolgen.

2. Ist der Befehl E oder e einzugeben. Danach erscheint unter dem Text folgendes Bild:

```
0003 Floppy Disk
0007 Floppy
0009 Floppydisc
0013 Floppy-disc
```

```
Befehl:
editieren █
```

3. Jetzt kann der Cursor an jede Stelle des Bildschirms bewegt werden. Zulässig sind folgende Tasten:

Home, →, ←, ↓, ↑, Shift ←

Zum Korrigieren des Textes können dann verwendet werden:

CLR, INSERT, DELETE und Überschreiben mit Zeichen.

Diese Änderungen werden deutlich durch geänderte Farbe angezeigt. Sie sind aber zunächst noch nicht übernommen. Damit dies geschieht, ist folgendes notwendig.

4. Die Betätigung von ENTER. Dabei werden aber nur diejenigen Änderungen übernommen, die zu der entsprechenden Quasizeile gehören, in der der Cursor sich zu diesem Zeitpunkt befindet.

5. Der Editmode wird mit BRK verlassen.

Hinweis 1:

Es ist jede Quasizeile einzeln nur durch ENTER zu verändern.

Hinweis 2:

Auch der Befehl Shift DELETE ist für das Editieren vorhanden. Er löscht genau eine Bildschirmzeile. Hierdurch sind vielfältige Besonderheiten möglich. So kann z. B. dadurch besonders einfach eine FFFF-Zeile eingefügt werden. Es ist vorn im Bereich der Zeilen-Nr. FFFF zu schreiben und nach zwei Leerzeichen der dann gewünschte Text. In der Zeile darüber ist das Verbindezeichen zu ergänzen. Mit ENTER wird dann diese Teilzeile an die Quasizeile angehängt.

Hinweis 3:

Im Prinzip kann beim Editieren auch die Zeilennummer zum Beginn der Quasizeile verändert werden. Dann wird der Text, der in dieser Quasizeile steht, zusätzlich in die Zeile kopiert, auf die die geänderte Zeilennummer weist. Auf diese Weise lassen



sich also leicht Grundstrukturen vervielfältigen, für die vorher nur Leerzeilen angelegt werden müssen.

Hinweis 4: Zur Funktion des Editiervorganges in der Zeile, wo der Cursor steht, wird zunächst die Zeilennummer am Beginn gelesen. Dabei treten folgende verschiedene Reaktionen auf:

- Der Rechner findet eine Zeilennummer vor, die gleich oder kleiner als die maximale Zeilenzahl ist. Dann wird der Text von Beginn an gerettet. Findet er am Ende ein Verbindezeichen, so wird auch die nächste Zeile gelesen usw., bis ein Zeilenende ohne Verbindezeichen erkannt wird. Dieser gesamte Text wird in der Zeile unter der vorher gelesenen Zeilennummer abgelegt. In diesem Fall, wo also schon die Zeilennummer bekannt ist und keine weitere Zeilennummer kontrolliert wird, läßt sich folgende Lösung realisieren:

```
0003 Max
0004 Moritz
0005 Heinz
0006 Günter
```

Hierbei wird nach ENTER bei der Anzeige gefunden:

```
0003 Max
FFFF Moritz
FFFF Heinz
FFFF Günter
0004 Moritz
0005 Heinz
0006 Günter
```

- Der Cursor befindet sich in einer FFFF-Zeile. Dann sucht der Rechner nach ENTER weiter oben, bis er eine wirkliche und gültige Zeilennummer findet.
  - Der Rechner findet zu Beginn weder eine gültige Zeilen-Nr. noch eine FFFF-Zeile, dann wird der Editmode mit ERROR verlassen. Dies erfolgt auch, wenn die Zeilennummer nicht vierstellig ist, wenn also z. B. statt
- ```
0004 Moritz
```
- nur
- ```
4 Moritz
```
- oder
- ```
004 Moritz
```
- steht. Dadurch besteht eine beachtliche Sicherheit gegen unbeabsichtigtes Editieren.
- Wird im Editmode ein Verbindezeichen entfernt und bleibt der Cursor in der darüberbefindlichen Zeile, so wird der Rest nach dem fehlenden Verbindezeichen abgehängt. Er geht verloren.

Der Editmode erlaubt also einerseits sehr vielfältige Änderungen und ist andererseits mit einem hohen Schutz gegen unbeabsichtigte Änderungen versehen.

## 1.8. Entfernen von Zeilen (DELETE)

Aus einem SORED-File lassen sich gezielt Quasizeilen streichen. Dies erfolgt mittels Eingabe des Befehls

d oder D.

Danach erscheint auf dem Bildschirm

```
Befehl:
Delete von Zeile:
```

■

Hier ist, wie bei der Anzeige, die erste zu löschende Zeilen-Nr. einzugeben, z. B. 3, dann folgt

```
Befehl:
Delete von Zeile:
```

3

```
bis:
```

■

Jetzt ist die letzte zu löschende Zeile einzugeben. Sie muß mindestens so groß wie die erste Zeilen-Nr. sein. Soll nur die eine Zeile gelöscht werden, so ist die gleiche Zeilen-Nr. nochmals einzugeben.

Bemerkung: Da beim Löschen einer Zeile alle folgenden Zeilen automatisch eine um Eins kleinere Nummer erhalten, ist für löschen von mehreren Zeilen wie folgt zu verfahren:

- Entweder bei der höchsten Zeilennummer mit dem löschen beginnen und schrittweise zu den niedrigeren überzugehen oder
- nach jedem Löschen sich durch Anzeige über die jeweils aktuelle Zeilennummer informieren.

=====

## 3. FILEX

FILEX ist ein Hilfsprogramm, mit dem zusätzliche und gemeinsame Funktionen für TEXOR und SORED realisiert sind. Es wird aus dem Grundzustand von TEXVE (Abschnitt 2) durch F bzw. f erreicht. Dadurch erscheint folgendes Menu:

```
***** FILEX *****
```

```
File /   Wandeln   Ersetzen
R:Retten  S: SORED  K:Keften
M:Anfügen T: TEXOR  Z:Zeichen
```

```
BRK:Hauptmenu
```

## Befehl:

Alle Eingaben außer F, H, K, M, R, S, T, Z und ERK führen zu ERROR.

## 5.1. Die beiden Filestrukturen und ihre Wandlung

Mit den Befehlen S und T bzw. s und t lassen sich TEXOR-Files in SORED-Files und umgekehrt verwandeln. Um diese Befehle sinnvoll einzusetzen, sind einige Kenntnisse über die Filestrukturen notwendig.

## 5.1.1. TEXOR-Struktur

Jedes TEXOR-File beginnt bei der Adresse 200H. Dort steht stets der Wert Null, dann folgt der nächste Absatz in ASCII-Zeichen. Jeder Absatz endet wieder mit Null. Am Ende des File stehen nach der letzten Null drei 0FFH. Zusammengefaßt gilt also:

200H=0

Absatz1 0

Absatz2 0

letzter Absatz 0

FF FF FF

Die Adresse nach dem letzten FF ist die Endadresse, welche beim Speichern des File verwendet wird. Wird das File versehentlich durch N gelöscht, so entsteht folgendes Bild:

200H=0

0 FF FF FF Rest von Absatz1 0

Absatz2 0

letzter Absatz 0

FF FF FF

Bei Übergang von TEXVE zu TEXOR wird nun das erste FF gesucht und die Endadresse um drei Zahlen erhöht. Dadurch ist das File scheinbar verschwunden. Findet TEXOR beim Start kein FF, so wird automatisch ein neues File erzeugt. Dies bedeutet, daß, wenn am Ende eines File die FF verlorengehen, dieses File nicht übernommen werden kann, obwohl es im eigentlichen Sinn existiert.

## 5.1.2. SORED-Struktur

Die SORED-Struktur weicht in folgenden Punkten von der TEXOR-Struktur ab:

- Die Absätze sind jetzt durch Quasizeilen ersetzt, wobei zunächst nach höchstens 32 Zeichen einmal das Zeichen 7 (Code 50H) erscheint.
- Das Ende des File ist durch FF XX YY gekennzeichnet, wobei

XX VV die binäre Darstellung der Kopfzeilenzahl ist. Bei zwei Kopfquasizeilen lautet also das Ende FF 02 00. Beim Einlesen von SORED-Files wird geprüft, ob hinter dem ersten FF nicht auch noch zwei weitere FF folgen. Dann wird das File mit ERROR zurückgewiesen. Findet SORED beim Start kein FF, so wird automatisch ein neues File erzeugt. Das Zurücksetzen des Zeigers (Pointer) erfolgt bei SORED durch den Kopfeintrag ganz in Analogie zu TEXOR. Das eigentliche File bleibt also auch hier im Hintergrund bestehen.

### 5.1.3. Wandlung zwischen den Files

Jedes SORED-File ist problemlos in ein TEXOR-File zu wandeln. Es sind einfach nach dem ersten FF zwei weitere anzufügen. Genau das bewirkt der Befehl T. Beim Übergang von TEXOR zu SORED muß dagegen die Anzahl der Kopfzeilen festgelegt werden. Deshalb folgt die Abfrage:

Kopfzeilen:

Anhand des eingegebenen Zahlenwertes werden die beiden letzten FF genau mit dem entsprechenden binären Wert ergänzt. In diesem Sinne kann problemlos mit S auch die Kopfzeilenzahl von SORED-Files geändert werden. Bei der Wandlung eines TEXOR-File in ein SORED-File ist aber zu beachten, daß TEXOR Absatzstruktur und SORED Quasizeilenstruktur besitzt. Die Wandlung mit S wird also immer problemlos erfolgen, aber bei der Anzeige und noch mehr beim Editieren zu SORED gewandelter Files können folglich beachtliche Probleme auftreten.

## 5.2. Kassetten-File-Arbeit

### 5.2.1. Retten von Files

SORED- und TEXOR-Files werden mit R auf Kassette gerettet. Dabei wird der zu rettende Adreßbereich automatisch bestimmt und übernommen. Nach R erscheint nur die Abfrage:

Name:

■ (GROSSEBUCHSTABEN)

Hier muß dem File ein Name gegeben werden, der beim Lesen des File immer wieder mit angezeigt wird und zur Unterscheidung gegenüber anderen Files dient.

Eingelesen wird ein File mit  
>LOAD bzw. >LOAD (Name)

im Betriebssystem.

### 5.2.2. Hinweise zum Umgang mit Aufzeichnungen

Generell ist ein sorgsamer Umgang mit den Files zu empfehlen.

steckt in ihnen doch eine beträchtliche Arbeit, besitzen sie oft einen hohen Wert und sind sie selten wieder vollständig zu erstellen. Deshalb ist es u. a. erforderlich

- Sicherheitskopien anzufertigen, die getrennt und an einem sicheren Ort (vor Kindern, Magnetfeldern usw.) geschützt aufbewahrt werden.

- Beim Eingeben größerer Files zwischendurch zeitweilige Kopien anzufertigen. Kann doch der Strom ausfallen, selbst sehr kurzfristige Impulse (z. B. Schalten von großen Verbrauchern, Gewitter in der Nähe) können im Rechner zu Fehlverhalten führen.

kein Programm ist total gegen Absturz zu schützen. Auch der Kassettenrekorder kann fehlerhaft werden.

Dies alles sind sehr seltene Fehler, aber die durch eine Zwischenaufzeichnung gewonnene Sicherheit hilft Ärger vermeiden.

- Aufzeichnungen der Files gegen unbeabsichtigtes Löschen oder Neuaufzeichnen durch Schreibschutz zu sichern. Hierzu ist das hintere rechte Stück Plaste an der Kassette auszubrechen bzw. bei Datenkassetten der kleine Nippel zu entfernen.

- Der Einfluß von Magnetfeldern auf Kassettenaufzeichnungen ist in der Praxis viel geringer, als oft zu lesen ist. Nur große Transformatoren und alte Farbfernsehempfänger besitzen Felder, die Kassettenaufzeichnungen beeinflussen. Hier sollte eine Entfernung von mindestens 20 bis 30 cm eingehalten werden. Weitaus gefährlicher sind Staub, Schmutz und Fingerabdrücke direkt auf dem Magnetband oder mechanische Beschädigungen, auch durch schlecht eingestellte Kassettenrecorder.

- Optimal für die Datenaufzeichnung sind die Digitalkassetten. Audiokassetten (low noise) sind recht gut, Chromdioxid-Kassetten sind oft weniger geeignet.

### 5.2.3. Zusammenfügen von Files

Soll an ein im Rechner vorhandenes File ein auf Band vorhandenes File angefügt werden, so wird dies mit dem Befehl

M

eingeleitet. Der Cursor verschwindet und jetzt kann das File von Band eingelesen werden. Es wird mit dem Namen und dem Adreßbereich, wo es abgelegt wird, angezeigt. Dabei wird die erste 0 des auf Band befindlichen File auf dem ersten FF im Rechner abgelegt. Auf diese Weise nimmt das kombinierte File in der Gesamtheit die Eigenschaften des nachgeladenen File an, also sowohl TEXOR als auch SORED, und bei SORED die Kopfzeilenzahl des nachgeladenen File. Weiter sind die beiden Files zumindest bei der ersten Anzeige

durch eine Leerzeile getrennt.

#### 5.2.4. Verwenden alter Files

SORED und TEXOR gibt es bzw. gab es Versionen, die von Cassette eingelesen werden mußten. Die dort geretteten Files entsprechen genau der Struktur der Files dieser Modulversion, bis auf zwei Punkte: Die Startadresse liegt nicht bei 200H, sondern z. B.

bei SORED 1.3 bei 99DH,  
bei TEXOR 1.1 bei 9FBH.

Die so erzeugten Files sind auch in der Modulversion verwendbar. Sie müssen jedoch mit Offset geladen werden:

LOAD 200-88D=F973

bzw.

LOAD 200-9FB=F805.

Bei einem TEXOR-File ist danach zusätzlich in FILEX mittels -> TEXOR (T) das neuartige TEXOR-File zu erzeugen. Diese Wandlung ist wegen der etwas unterschiedlichen Struktur des Fileendes notwendig.

#### 5.3. Zeichen ersetzen

Aus verschiedenen Gründen kann es notwendig werden, in einem vollständigen File ein Zeichen durch ein anderes auszutauschen. So mußte z. B. in diesem Manuskript das Zeichen zur Beschreibung verwendet werden. Es ist aber nicht direkt eingetastet. Deshalb wurde es beim Schreiben durch ■ ersetzt. Anschließend, im fertigen Text, mußte das Zeichen wieder gewandelt werden. Andere Fälle treten ein, wenn z. B. bei Zahlen das Komma im deutschen durch den Punkt im englischen ersetzt werden muß. Um dies alles zu ermöglichen, sind zwei Zeichendarstellungen zu kennen:

Das Zeichen selbst und sein ASCII-Code, also z. B.

A und 41 oder m und 6D.

Für die Sondertastenbelegungen gelten die Codes:

|       |       |       |       |
|-------|-------|-------|-------|
| ä=80H | ö=81H | ü=82H | ß=83H |
| ✓=84H | †=85H |       |       |
| W=86H | Q=87H | U=88H | □=89H |
| ◇=8AH | Δ=8BH | ~=8CH |       |

Beim Zeichenersetzen wird nun ein Z eingegeben und es folgt:

mit ', '=ASCII, sonst HEX-Wert  
altes Zeichen:

■

Hier kann z. B. bei A entweder ,A oder 41 eingegeben werden, dann folgt

mit ', '=ASCII, sonst HEX-Wert  
altes Zeichen:

↓

A  
neues Zeichen:

Hier folgt nun, wenn A durch A ersetzt werden soll: 86 (denn die ASCII-Werte gelten nur für die Codezahlen 20H = Space bis 7FH). Nach 86 erscheint meist nach sehr kurzer Zeit wieder die Anzeige "Befehl:". Das bedeutet, daß der Austausch erfolgt ist.

Hinweis: Bei beiden Eingaben entfällt das ENTER, es wird automatisch nach zwei Zeilen erzeugt.

Bevor zum File zurückgekehrt werden kann, ist FILEX mit BRK zu verlassen.

#### 3.4. Zeichenkette ersetzen

Häufiger als einzelne Zeichen werden Zeichenketten zu ersetzen sein, z. B. Holland durch Niederlande oder DC durch Gleichstrom. Dies erfolgt mit dem Befehl

und es wird angezeigt:

Zeichenkette (■=dont-care):

Hier ist also die Zeichenkette mit eventueller Verwendung von dont-care-Zeichen (siehe Abschnitt 3.5) einzugeben, z. B.

Zeichenkette (■=dont-care):

Holland

neue Kette:

Hier wird jetzt Niederlande eingetragen und nach kurzer Zeit meldet sich wieder die Abfrage "Befehl:" zum Nachweis dafür, daß der Eintrag erfolgt ist.

#### 3.5. Fileanzeige

Auch in FILEX wurde eine Fileanzeige vorgesehen. Sie wird mit erreicht und zeigt dann, wie bei SORED und TEXOR, die vorhandenen und freien Byte sowie den Filetyp an, also z. B.

File XXXXXX Byte lang  
noch VVVVVV Byte frei  
SORED

=====



## 1. Diagnosesystem

Das Diagnosesystem ist aus dem Betriebssystem zu erreichen, indem

>D XXXX

gestartet wird. Dann ergibt sich folgendes Bild

Adresse 32 ASCII-Zeichen

Adresse 32 ASCII-Zeichen

BSW.

Die erste Adresse eines File ist 200H. Das File wird in spezieller Weise aufgelistet, dabei werden alle Zeichen von 20H (Space) bis 8CH (~) direkt angezeigt.

Für die Codes kleiner 20H, also speziell für die Null, für Trennzeichen zwischen den Absätzen und Quasizeilen, wird

#

erzeugt und für alle Zeichen größer 8CH (also in erster Linie die FF-Zeichen)

%

So ist schnell zu erkennen, wie ein File, das versehentlich gelöscht oder anders verändert wurde, wieder zu rekonstruieren ist. Dies kann mit

>MODIFY Adresse

geschehen. Deshalb ist das Diagnosesystem auch in das Betriebssystem verlegt worden.

Die Anzeige im Diagnosesystem läuft maximal bis zum Ende des RAM-Bereichs, wobei zur Beschleunigung des Ablaufes das Paging verwendet wird.

Die Anzeige kann jederzeit mit

BRK

angehalten und mit

C

fortgesetzt werden. Dagegen führt ENTER nach BRK aus dem Diagnosesystem heraus und es kann, falls notwendig, unmittelbar mit >MODIFY die Änderung vorgenommen werden.

=====

## 1. Initialisierung V 24

Jedes Textverarbeitungssystem ist auch wesentlich durch die verwendete Peripherie begrenzt. Um dieses Textverarbeitungssystem breit verwendbar zu halten, wurden für die Initialisierung der verschiedensten Geräte in jedem Modul mehrere Geräteinitialisierungen installiert.

Das Grundsystem enthält die Varianten

- Schreibmaschine ERIKA S 6005 mit 9600 Baud
  - Printer/Plotter mit 600 Baud
  - Mosaikdrucker 6311 mit IFSS-Schnittstelle.
- Für jedes Gerät existieren ein entsprechender Modul und dessen Programmierung.

Aus dem Grundprogramm TEXVE wird mit

I

die Initialisierung aufgerufen, es folgt dann die Anzeige:

Initialisieren

E: Erika  
I: IFSS  
P: Plotter  
Z: zurück

Es ist je nach verwendetem Gerät E, I oder P einzugeben und alle für den Drucker notwendigen Vorbereitungen sind getroffen. Bei E und I wird angezeigt, wenn der entsprechende Modul für diese Verarbeitung nicht existiert oder im falschen Schacht steckt:

MODULERROR>

Bei P fehlt infolge der Eigenart des verwendeten Moduls dieser Schutz.

Mit Z wird der Modul wieder abgeschaltet.

Es werden auch für andere Gerätevarianten Modulanschlüsse sowie die entsprechenden Initialisierungsprogramme im TEXVE-Modul vorbereitet.

Anhang Textverarbeitung

(gilt für Version 2.1)

1. Hauptmenu führt zu: T: TEXVE  
 S: SORED  
 F: FILEX  
 I: INIT  
 STOP: CAOS

2. TEXVE läßt zu:

Fileeintritt: N: neues File  
 B: Ende eines File (bottom)  
 T: Beginn eines File (top)  
 G: Gehe zum Abschnitt X

Anzeigen: H: Menu (help)  
 F: Fileparameter

Ausgaben: R: Randtest, bei Silbentrennung ist weicher  
 Bindestrich mit ← und → in Grenzen  
 verschiebbar  
 e: Ausgabe mit Flattersatz (exmit)  
 E: Ausgabe mit Blocksatz  
 Z: Zeichensuche  
 " " fortsetzen mit (Shift CLR)  
 " " abbrechen mit BRK

Verlassen: BRK: zum Hauptmenu

Veränderungsmode: ↓a: Erzeuge Absatz  
 ↓v: Verbinde (Absatz)  
 ◇Text↓d: Lösche Text (delete)  
 ◇Text↓p: Ausgabe Text (P: Blocksatz)  
 ◇Text◇...↓c: Kopiere Text nach ↓  
 ◇Text◇...↓m: Verlagere (move) Text nach ↓

Fenster klein: a, t, b, g, z

Sonderzeichen: (nur das erste gilt in einem Absatz)

△ Symmetrie = Setze in Zeilenmitte

↘ Setze Text hinter dem Zeichen rechtsbündig

□ Tabuliere auf die Positionen 4, 8, 12, 16 usw.

Silbentrennung:

Automatische Trennzeichen sind neben Space

! " # \$ % ' ( ) \* + , - . / : ; < = > ?

(hinter ihnen erfolgt die Trennung).

Halbautomatische Trennung als Angebot

- hinter y

- hinter Selbstlaut + einem Mitlaut.

Der Verschieberegion ist bei der Zeilenlänge Z:

$Z(1 - 1/8) - 2$  bis Z

Die Zeilenlänge ist formatierbar von 10 bis 255 Zeichen.

Textmode

Alle Zeichen sind Eingaben.

Betätigen der Cursor-Führungstasten bewegt den Cursor sinngemäß auf dem Bildschirm.

INS, DEL, CLR, HOME - wie im Betriebssystem

Shift ↓ Vorblättern 1 Seite

Shift ↑ Zurückblättern 1 Seite

↑ Cursor am oberen Bildschirmrand - eine Zeile zurückblättern

Sonderzeichen entfernen

W Vernichten (löschen) von: ◇ ↓

l Löschen von: ~ ◇ ↓

### 3. SORED

Eingaben: K: Kopf  
Beginn erst nach Shift Home möglich,  
BRK beendet  
T: Texteingabe  
BRK beendet  
7: Sonderzeichen zum Verbinden von Zeilen  
zu einer Quasizeile am Ende einer Zeile

Anzeigen: A: Ausgabe von Text  
H: Menu (help)  
F: Fileparameter  
Z: Zeichensuche  
(Für Zeichenkette don't care = Shift Space  
und zur Korrektur nur CLR)  
U: Umgekehrte Zeichensuche

Zeilennummer und Verbindezeichen erscheinen bei A, Z, U.  
Dies ist für Edit notwendig. Bei a, z, u ohne Zeilennummer  
und Verbindezeichen.

Funktionen: S: Sortiert den Text  
E: Übergang in Editmode  
Mit Cursor kann auf dem Bildschirm geändert  
werden. Änderungen werden nur dann über-  
nommen, wenn für jede Quasizeile ENTER  
betätigt wird.  
D: delete (löschen) von Zeilen,  
(am besten von hinten beginnen)

### 4. FILEX

Filearbeit: R: Retten eines File  
L: Laden eines File  
A: Anfügen eines gespeicherten File an das im  
Rechner vorhandene (merge)  
F: Fileparameter

Ändern: S: in SORED-File mit festzulegender Zeilenzahl  
T: in TEXOR-File

Ersetzen im File:  
Z: von Einzelzeichen  
K: von Ketten aus einem oder mehreren Zeichen  
BRK: zurück ins Hauptmenu

5. INIT unterschiedlich, je nach EPROM 4

6. d = Diagnosesystem

Eintritt mit:      d:XXXX : (Startadresse)  
                      BRK        : Anhalten  
                      C            : Fortsetzen  
                      ENTER      : Abbrechen

7. HEX-Code der Sonderzeichen

| Zeichen | HEX-Wert | Taste    |
|---------|----------|----------|
| ä       | 80       | F1       |
| ö       | 81       | F2       |
| ü       | 82       | F3       |
| ß       | 83       | F4       |
| \       | 84       | F5       |
| ↑       | 85       | F6       |
| Ä       | 86       | Shift F1 |
| Ö       | 87       | Shift F2 |
| Ü       | 88       | Shift F3 |
| ▣       | 89       | Shift F4 |
| ◇       | 8A       | Shift F5 |
| △       | 8B       | Shift F6 |
| ~       | 8C       | keine    |

Achtung! Die Code 7B bis 7F ergeben: ä ö ü ß. Sie sind nicht über Tasten erreichbar.

8. Sortieren der Zeichen

Hierbei gilt das Alphabet, kleine und große Buchstaben sind gleichwertig

a = A = ä = Ä  
 o = O = ö = Ö  
 u = U = ü = Ü  
 s = ß

Vor den Buchstaben stehen in der folgenden Reihenfolge:

Space ! " # \$ % & ' ( ) \* + , - . /

dann:

0 1 2 3 4 5 6 7 8 9 : ; < = > ?

Nach den Buchstaben:

■ | ~ ^ \_

Die über Tasten nicht erreichbaren Code 7B bis 7F fehlen hier!