

# **Benutzerhandbuch**

**Teil: Textverarbeitung**

**Stand: 19.12.86**



**Änderungsdienst  
zum Benutzerhandbuch (Teil: Textverarbeitung)**

- Zu jeder Änderung der Dokumentation wird eine Änderungsanweisung mit laufender Nummer hinzugefügt. Dort sind alle auszuführenden Änderungen aufgeführt.
  
- Die auf der Änderungsanweisung angegebene Nummer wird nach Einarbeitung der Änderungen auf dem Blatt 'Änderungszustand' eingetragen. Dadurch kann festgestellt werden, ob alle Änderungen in das Dokument eingearbeitet wurden.

**Änderungszustand  
zum Benutzerhandbuch (Teil: Textverarbeitung)**


In diese Felder sind fortlaufend die Nummern der Änderungsanweisungen einzutragen. Dadurch kann festgestellt werden, ob alle Änderungen in das Dokument eingearbeitet worden sind.

---

© GMD 1986

Alle Rechte vorbehalten. Insbesondere ist die Überführung in maschinenlesbare Form sowie das Speichern in Informationssystemen, auch auszugsweise, nur mit schriftlicher Einwilligung der GMD gestattet.

No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the GMD.

---

Autoren :        Monika Fey – McClean  
                  Konrad Klöckner  
                  Werner Metterhausen    u.a.

Texterstellung :

Dieser Text wurde mit der EUMEL – Textverarbeitung erstellt und aufbereitet und mit dem Agfa Laserdrucksystem P400 gedruckt.

---

Gesellschaft für Mathematik und Datenverarbeitung mbH

Bereich/Location Birlinghoven

Schloß Birlinghoven

Postfach 1240

D – 5205 Sankt Augustin 1

Telefon (02241) 14 – 1

Telex 8 89 469 gmd d

Telefax (02241) 14 28 89

Teletex 2627 – 224135 = GMDVV

Bildschirmtext \*43900#

---

## **TEIL 1: Einleitung**

1.1.	Vorwort	1
1.2.	Wichtige Begriffe	4
1.3.	Die Notation in diesem Buch	8
1.4.	Voraussetzungen	10
1.5.	Eine Beispielsitzung	14

## **TEIL 2: Der Supervisor**

2.1.	Steuerkommandos	1
2.2.	Eine Task einrichten	3
2.3.	Informationskommandos	8
2.4.	Übersicht über Supervisor – Kommandos	10

## **TEIL 3: Der Monitor**

3.1.	Der Monitor	1
3.1.1.	Informationskommandos	4
3.2.	Tasksteuerung	7
3.3.	Dateibearbeitung	12
3.4.	Das Archiv	23
3.5.	Kommandos für mehrere Dateien	36
3.7.	Passwortschutz	41
3.8.	Monitor – Kommandos	47

## TEIL 4: Der Editor

4.1.	Ein – und Ausschalten des Editors	2
4.2.	Die wichtigsten Tasten des Editors	5
4.2.1.	Das Tastenfeld	5
4.2.2.	Speicherung von Texten	18
4.2.3.	Schreiben von Texten	19
4.2.4.	Positionieren im Text	23
4.2.5.	Korrigieren im Text	25
4.2.6.	Der Tabulator	32
4.2.7.	Lernen im Editor	35
4.2.8.	Textabschnitte durch Markieren bearbeiten	37
4.2.9.	Der Fenstereditor	39
4.2.10.	Die wichtigsten vorbelegten Tasten	41
4.3.	Die wichtigsten Editor – Kommandos	44
4.3.1.	Der Kommandodialog	44
4.3.2.	Zeile und Textstelle anwählen	45
4.3.3.	Suchen und Ersetzen	47
4.3.4.	Kommandos auf Tasten legen	56
4.3.5.	Texte aus anderen Dateien benutzen	58
4.3.6.	Breitere Zeilen bearbeiten	59
4.3.7.	Die wichtigsten Kommandos	61
4.4.	Fehlersituationen und Abhilfe	66

## TEIL 5: Textkosmetik und Druck

5.0.	Vorwort	1
5.1.	Einführung in die Benutzung der Textkosmetik	2
5.1.1.	Anweisungen für die Textkosmetik und den Drucker	6
5.1.2.	Aufruf der Textkosmetik – Programme	9
5.1.3.	Vorzeitiger Abbruch und Fehlermeldungen	11
5.2.	Lineform/Autoform	13
5.2.1.	Zeilenweise formatieren	16
5.2.1.1.	Interaktive Silbentrennung	16
5.2.1.2.	Automatische Silbentrennung mit autoform	19
5.2.2.	Unterschiedliche Schriften	20
5.2.3.	Veränderung des Schrifttyps	22
5.2.4.	Gesperrt schreiben	24
5.2.5.	Zeilenbreite einstellen	25
5.2.6.	Einfache Tabellen und Aufzählungen schreiben	27
5.2.6.1.	Tabellenanweisungen	31
5.2.6.2.	Einstellen der Tabellenpositionen	34
5.2.6.3.	Blocksatz innerhalb einer Spalte	35
5.2.6.4.	Tabellenspalten auffüllen (Füllzeichen)	36
5.2.6.5.	Tabellenpositionen löschen	38
5.2.7.	Indizes und Exponenten	39
5.3.	Pageform	42
5.3.1.	Seitenweise formatieren	42
5.3.1.1.	Automatische Seitenformatierung	44
5.3.1.2.	Seitenende interaktiv verschieben	45
5.3.2.	Seitenlänge einstellen	49
5.3.3.	Zeilenabstand einstellen	50
5.3.4.	Platz freihalten	52
5.3.5.	Neue Seite beginnen	53
5.3.6.	Kopf- und Fußzeilen	54
5.3.7.	Seiten numerieren	57
5.3.8.	Fußnoten schreiben	60
5.3.8.1.	Fußnoten numerieren	63
5.3.9.	Querverweise	67
5.3.10.	Kombination von Tabellen, Fußnoten und Kopf- bzw. Fußzeilen	69
5.3.11.	Formatierung von Spalten	70
5.4.	Index	75

5.4.1.	Stichwort- und/oder Inhaltsverzeichnisse	75
5.4.1.1.	Worte für 'index' kennzeichnen	77
5.4.1.2.	Nebeneinträge erzeugen	79
5.4.1.3.	Indexdateien zusammenführen	82
5.5.	Outline	83
5.5.1.	Eine Strukturübersicht oder Zusammenfassung erstellen	83
5.6.	Print	85
5.6.1.	Eine Datei drucken	86
5.6.2.	Anweisungen für den EUMEL – Drucker	87
5.6.3.	Blocksatz	89
5.6.3.1.	Randausgleich	89
5.6.3.2.	Seitenausgleich	91
5.6.4.	Schreibfeld verschieben	92
5.6.5.	Zentriert drucken	93
5.6.6.	Rechtsbündig drucken	94
5.6.7.	Übereinander drucken	95
5.7.	Textkosmetik – Makros	96
5.7.1.	Ein Makro – Beispiel	97
5.7.2.	Ein Beispiel mit Makro – Parametern	101
5.7.3.	Makros für Manuskripte	103
5.8.	Textkosmetik für Spezialisten	105
5.8.1.	Schalter – Anweisungen für Kopf- und Fußbereiche	105
5.8.1.1.	Kopf- und Fußbereiche abstellen	107
5.8.2.	Textzeilen markieren	109
5.8.3.	Fußnoten pro Seite zählen	111
5.8.4.	Behandlung falscher Silbentrennungen: Ausnahmelexikon	112
5.8.5.	Voreinstellung ändern: Einige Monitor – Kommandos	114
5.8.5.1.	Trennpunkt einstellen	114
5.8.5.2.	Anzahl Leerzeilen vor Fußnoten einstellen	115
5.9.	Übersicht über die Anweisungen und Kommandos der EUMEL – Textkosmetik	116
5.10.	Fehlersituationen und Abhilfe	121

## **TEIL 6: Spezialitäten**

6.2.	EUMEL – Zeichensatz	3
6.3.	Sortier – Programme	6
6.4	Fonttabellen	9
6.5	Syntax der Kommandos	12

## **Anhang**

	Aufbau und Installation	1
	Druckersoftware einrichten	11

## **Index**

# TEIL 1: Einleitung

## 1.1. Vorwort

Dieses Buch bietet Ihnen eine Anleitung zur Handhabung des Betriebssystems EUMEL in Hinblick auf die Textverarbeitung. Das Buch wendet sich somit an alle, die das leicht erlernbare EUMEL-System zur Erstellung von Texten jeglicher Art und jeden Umfangs nutzen wollen.

Die Anleitung erfordert keinerlei Vorkenntnisse, weder von Computern im allgemeinen noch von EUMEL im besonderen. Neulingen auf dem Gebiet der Benutzung eines Computers sei empfohlen, dieses erste Kapitel, das sich kurz mit der 'grauen Theorie' beschäftigt, mindestens zweimal zu lesen:

- Der erste Durchgang sollte nur einer groben Orientierung dienen. Die Begriffe, die teilweise erst im routinierteren Umgang Sinn gewinnen, sollten Sie 'mal gehört' haben, um die folgenden Kapitel leichter lesen zu können.
- Der zweite Durchgang erscheint uns sinnvoll, wenn Sie sich 'freigeschwommen' haben. Wenn der Umgang mit dem EUMEL-System Ihnen nach einigen Tagen schon vertrauter erscheint, ist es zweckmäßig, sich auf dem Hintergrund der gemachten Erfahrungen die Erklärungen noch einmal durchzulesen. Einige der Begriffe werden erst dann wirklich verständlich und manche Unsicherheit wird beseitigt werden.

Die weiteren Teile des Buches geben dann Anleitung vom ersten Einstieg ins EUMEL-System bis hin zur detaillierten Beschreibung der Textverarbeitung. Alle in Teil 3 und 4 enthaltenen Beispiele sollten Sie sorgfältig durchgehen und am Bildschirm nachvollziehen. Aus dem Teil 5 können Sie nach eigener Einschätzung zunächst die Bereiche auswählen, die Sie selbst für besonders wichtig halten. Sie werden feststellen, daß Sie durch den Umgang mit der EUMEL-Textverarbeitung mit fortschreitender Routine immer mehr der gebotenen Möglichkeiten hinzunehmen werden.

## Was ist ein Betriebssystem ?

Ein Betriebssystem ist eine Sammlung von Programmen, die dem Benutzer eines Computers die Arbeit mit diesem Gerät erst ermöglichen. Die Programme des Betriebssystems stellen die Verbindung zwischen den Bausteinen eines Computers, der Hardware, und den Anwendungsprogrammen eines Benutzers her.

Alle Programme, die diese riesige Lücke schließen und z.B. dafür sorgen, daß der Befehl: *drucke ("diesen brief")* tatsächlich dazu führt, daß der eben noch am Bildschirm verfaßte Brief zum Drucker gesendet und dort ausgedruckt wird, sind sogenannte Systemprogramme, Teile des Betriebssystems.

Dieses Benutzerhandbuch zum Betriebssystem EUMEL wird Ihnen schrittweise die Befehle erklären, die Sie zur Nutzung aller Möglichkeiten der Textverarbeitung mit EUMEL verwenden können, und Ihnen somit einen Teil des Betriebssystems vorstellen.

## Ist EUMEL anders als andere ?

**Ja.** Das Betriebssystem EUMEL (Extendable multi User Microprocessor ELan System) erklärt durch seinen voll ausgeschriebenen Namen einen wesentlichen Unterschied zu anderen Systemen: "Erweiterbares Mehrbenutzer Mikroprozessor ELAN – System."

Während andere auf Mikroprozessoren (und damit auf sogenannte Personal Computer) abgestimmte Betriebssysteme einen Benutzer bei seiner Arbeit unterstützen, ist EUMEL fähig, mehreren Anwendern gleichzeitig die Benutzung eines Computers zu ermöglichen. Natürlich funktioniert EUMEL ebensogut für einen einzigen Benutzer. Es gibt aber bei EUMEL die Möglichkeit, durch Ankoppeln weiterer Bildschirme an den Rechner und ohne Kosten für zusätzliche Software mehreren Benutzern gleichzeitig die Arbeit an diesem Rechner zu ermöglichen.

Zweitens ist EUMEL hardwareunabhängig, das heißt, gleichgültig von welchem Hersteller Ihr Computer stammt, die Bedienung und die Kommandosprache ist immer gleich. Auch können Disketten, die mit einem XY – Rechner beschrieben wurden, von einem ABC – Computer gelesen werden; durchaus keine Selbstverständlichkeit.

Eine weitere Besonderheit des EUMEL – Systems macht alle froh, die damit arbeiten: EUMEL ist durchgängig in der Programmiersprache ELAN gehalten. Auch wenn Sie (noch) nicht programmieren möchten, erleichtert Ihnen ELAN das Leben dadurch, daß Sie schreiben können, was Sie meinen: eine Datei, die einen Geschäftsbrief fix und fertig und druckbereit enthält, heißt nicht etwa:

**\$TXT.PRT**

sondern:

**Angebot an Fa.Müller 1.7.86**

Ein weiterer wichtiger Unterschied wird Ihnen bewußt werden, wenn Sie ein anderes Betriebssystem kennen: Die EUMEL – Textverarbeitung ist kein zusätzliches Programm mit eigener Kommandosprache, welches bei Bedarf geladen werden muß, sondern steht jederzeit, im wahrsten Sinne des Wortes auf Knopfdruck, zu Ihrer Verfügung.

## 1.2. Wichtige Begriffe

- **TASK.** Eine Task ist ein eigenständiger Prozeß innerhalb eines EUMEL – Systems, der entweder zur Verwaltung des EUMEL – Systems oder zu einem Benutzer gehört. Indem jedem Benutzer ein eigener Arbeitsbereich zugewiesen ist, wird verhindert, daß unkontrolliert auf fremde Daten zugegriffen wird. Eine Task hat einen Namen, mit dem sie angesprochen werden kann. Ein EUMEL – System besteht aus mehreren Tasks.

Ein brauchbarer Vergleich mit einem EUMEL – Tasksystem ist ein Firmengebäude: Es besteht aus vielen Räumen und jeder Raum ( = Task ) ist entweder ein normales Arbeitszimmer oder ein Chefbüro oder eine Werkstatt, in der Dienstleistungen für andere erledigt werden.

Eine solche Ordnung zeigt folgendes Tasksystem; der *kursiv gesetzte* Kommentar zeigt die Benennung der 'Dienstposten' eines vergleichbaren Büros:

```

SUPERVISOR (* Zimmerverwalter *) -
-
SYSUR   (* Werkstattmeister *)

ARCHIVE (* Archivar *)

configurator (* Elektriker *)

OPERATOR (* Hausmeister *)

        shutup (* Nachtwächter *)

UR      (* Aufsichtsrat *)

PUBLIC  (* Abteilungsleiter *)

Meier   (* Angestellter *)
Müller  (*      "      *)
Schulze (*      "      *)

```

Bildlich gesprochen stellt eine Task also ein 'Arbeitszimmer' für einen EUMEL – Benutzer dar. Als EUMEL Anwender richten Sie sich Ihre Task selbst ein, indem Sie das Kommando 'begin ("taskname")' geben.

Nachdem Sie dieses Kommando einmal gegeben haben, existiert diese Task unter dem von Ihnen gewählten Namen.

In der Task (also sinngemäß im Arbeitszimmer) arbeiten Sie – insbesondere legen Sie Dateien (= Akten) an. Dateien existieren nur innerhalb einer Task.

Tasks werden durch den SUPERVISOR verwaltet, er regelt den Zugriff auf Tasks. Um Ihre Task zu verlassen, geben Sie das Kommando 'break' an den SUPERVISOR, um sie wieder zu betreten, das Kommando 'continue ("taskname")'.

- **DATEI.** Eine Datei ist eine Menge von zusammengehörigen Daten. Eine Datei in einer Task entspricht einer Akte in einem Arbeitszimmer. Eine Task kann bis zu 200 Dateien enthalten. Jede Datei in einer Task hat einen eigenen Namen, in verschiedenen Tasks dürfen gleichnamige Dateien existieren. Eine Datei ist in Zeilen unterteilt.

Für die Arbeit in einer Datei am Bildschirm muß auf die Datei mit dem Editor zugegriffen werden: 'edit ("dateiname")', danach kann der Inhalt der Datei am Terminal bearbeitet werden (siehe Kapitel 4 und 5).

- **KOMMANDO.** Ein Kommando ist ein Befehl an den Rechner, eine Arbeit zu tun. Welche Kommandos Sie dem Rechner zum jeweiligen Zeitpunkt geben können, hängt davon ab, auf welcher 'Kommandoebene' Sie sich befinden. Als Anhalt gilt:
  - Kommandos auf Supervisor – Ebene betreffen das Tasksystem.
  - Kommandos auf Monitor – Ebene betreffen die eigene Task oder Dateien.
  - Kommandos auf Editor – Ebene betreffen Zeilen, Worte oder einzelne Zeichen der aktuellen Datei.

Auf welcher Ebene Sie sich befinden, werden Sie nach kurzer Gewöhnung leicht am Bildschirm erkennen (siehe Teil 3).

Bei manchen Kommandos muß nicht nur gesagt werden, was getan werden soll, sondern auch, womit es getan werden soll. Eine solche Angabe zum Kommando heißt Parameter.

		Kommando	Parameter
Beispiel:	Lege neue Task an	= begin	("taskname")
	Drucke Datei	= print	("dateiname").
	Suche das Wort ENDE	= down	("ENDE")

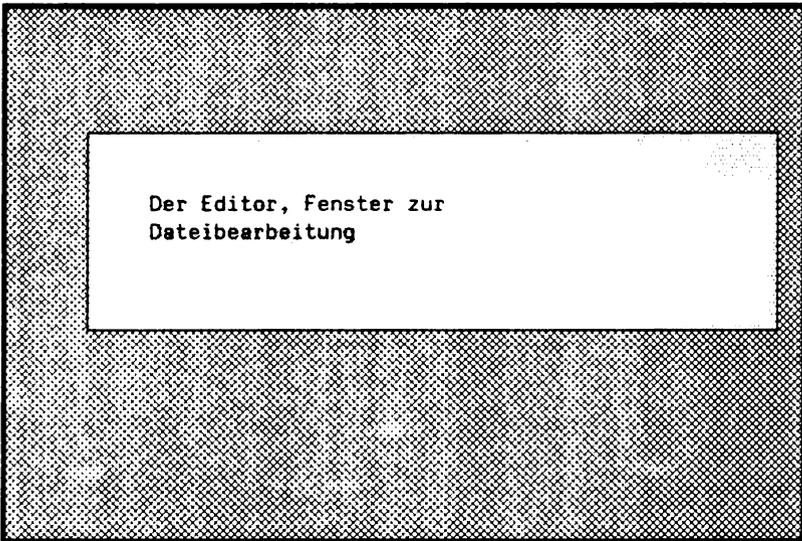
Parameter werden in runde Klammern gesetzt und ggf. durch Kommata voneinander getrennt. Textparameter werden zusätzlich in Anführungsstriche gesetzt.

Ein Kommando kann keinen, einen oder viele Parameter benötigen; die Beschreibung der Kommandos in diesem Buch zeigt jeweils alle Möglichkeiten.

- **SUPERVISOR.** Spezielle Task zur Überwachung eines EUMEL-Systems. Ein Benutzer kann durch die Supervisor-Kommandos Leistungen von dieser Task fordern: neue Task einrichten, Task wiederaufnehmen und diverse Informationen.
- **MONITOR.** Befehlsempfänger in einer Task. Jede Arbeit im EUMEL-System findet in einer Task statt. Die Arbeit mit einem Computer besteht in wesentlichen Teilen im Aufruf von Programmen durch Kommandos. Der Empfänger dieser Kommandos in einer Task ist der Monitor. Der Monitor ist sichtbar durch eine Zeile, in der 'gib kommando' steht. In diese Zeile werden Kommandos und erforderliche Parameter eingegeben.
- **ARCHIVE.** Spezielle Task zur Verwaltung des Diskettenlaufwerks. Da für die längerfristige Datenhaltung und zur zusätzlichen Datensicherung Dateien auf Disketten geschrieben werden, besitzt das EUMEL-System für diese Aufgabe eine besondere Task, die die Bedienung vereinfacht und exklusiven Zugriff auf das Laufwerk garantiert.

- **EDITOR.** Programm zur Dateibearbeitung am Bildschirm. Das Programm wird durch das ( Monitor- ) Kommando 'edit' und die Eingabe des Namens der gewünschten Datei als Parameter gestartet.

Da ein Bildschirm normalerweise auf 80 Zeichen Zeilenbreite und 24 Zeilen beschränkt ist, kann der Editor als Fenster betrachtet werden, das über die möglicherweise weitaus größere Datei bewegt wird und durch das der betrachtete Ausschnitt der Datei bearbeitet werden kann.



## 1.3. Die Notation in diesem Buch

Im weiteren Text werden Sie schrittweise in die Bedienung des Systems eingeführt. Für alle Kommandos und Arbeiten haben wir Beispiele in dieses Buch aufgenommen, die Sie direkt am Rechner nachvollziehen sollten.

Beachten Sie dabei bitte folgende Regeln der Aufschreibung:

- Es gibt eine Reihe von Tasten auf einer Computertastatur, die eine besondere Bedeutung haben. Diese sogenannten Funktionstasten werden ebenso wie besondere Tastenkombinationen explizit als Tasten dargestellt:

SV

ESC e

CR

- Alles, was Sie am Bildschirm Ihres Rechners schreiben oder lesen sollen, ist in Textbereiche, die einen Bildschirm darstellen, eingefaßt.

Beispiel:

```
gib kommando:  
edit ("meine datei")
```

- Innerhalb des Handbuchs sind in der Aufschreibung die Konventionen der Programmiersprache ELAN, in der alle Programme des Betriebssystems geschrieben sind, berücksichtigt. Dabei sind folgende Besonderheiten zu beachten:

- 1) Kommandos werden grundsätzlich klein geschrieben.
- 2) Dateinamen u.ä. werden in Klammern und Anführungsstriche gesetzt. In diesem Buch steht an den Stellen, wo ein Dateiname auftaucht 'dateiname'; den Namen, den Sie tatsächlich verwenden, können Sie frei wählen.
- 3) Falls besondere Begriffe oder Beispiele innerhalb eines normalen Textes auftreten, werden sie in einfache Anführungsstriche gesetzt.

Also: Das Kommando 'edit' benötigt als Parameter einen Dateinamen. Wählen Sie einen Namen und geben Sie 'edit ("dateiname")' ein. Falls Sie den Namen "Geschäftsbrief" gewählt haben, müssen Sie am Bildschirm:

edit ("Geschäftsbrief")

tippen und das Kommando mit der **CR** Taste dem Monitor zur Bearbeitung übergeben:

```
gib kommando :
edit ("Geschäftsbrief")
```

**Die Eingabe von **CR** als 'Auslöser' für die Ausführung von Kommandos wird im weiteren nicht besonders hervorgehoben.**

## 1.4. Voraussetzungen

Neben dem Computer an sich ist die vollständige Installation eines EUMEL – Systems auf diesem Computer Voraussetzung für alle im folgenden beschriebenen Aktivitäten.

Die Beschreibung einer Systeminstallation finden Sie im Anhang I. Im weiteren gehen wir davon aus, daß Ihr Rechner sich in einem Zustand befindet, der durch Eingabe von **SV** oder **CNTL b** (gleichzeitig) die sogenannte EUMEL – Tapete zeigt und Supervisor – Kommandos annimmt.

```
Terminal ?  
  
EUMEL Version 1.8/M  
  
gib supervisor kommando:  
begin("meine erste task")  
  
ESC ? --> help  
ESC b --> begin("")  
ESC c --> continue("")  
ESC q --> break  
ESC h --> halt  
ESC s --> storage info  
ESC t --> task info
```

Weiterführende Information zum Aufbau eines EUMEL – Systems finden Sie im Anhang I.

## Die Funktionstasten des EUMEL – Systems

Die Lage der EUMEL – Funktionstasten entnehmen Sie bitte der speziellen Installationsanleitung zu dem von Ihnen benutzten Gerät.



Positionierungstasten

**SHIFT**

Umschalttaste

**CR**

Eingabe – / Absatztaste

**HOP**

Verstärkertaste

**RUBOUT**

Löschtaste

**RUBIN**

Einfügetaste

**TAB**

Tabulatortaste

**MARK**

Markiertaste

**ESC**

Kommandotaste

**SV**

Supervisortaste

**STOP**

Stoptaste

**WEITER**

Weitertaste

## Task – Organisation

Zum Verständnis der Handhabung des Systems sollten Sie versuchen, eine Vorstellung von der Organisation der Teile zu bekommen.

Die einzelnen Tasks eines EUMEL – Systems 'stehen nicht frei im Raum', sondern sind in einer baumartigen Beziehung organisiert:

```
SUPERVISOR
-
  SYSUR
    configurator
    OPERATOR

    ARCHIVE
UR
  PUBLIC
    Meyer
    Müller
    Schulze
```

Das System besteht aus zwei Zweigen, die nebeneinander liegen:

Dem Systemzweig mit der Wurzel SUPERVISOR

und

dem Benutzerzweig mit der Wurzel UR.

Der Systemzweig stellt Ihnen privilegierte Dienstleistungen zur Verfügung, der Benutzerzweig stellt die normale Arbeitsumgebung dar.

Alle unter diesen Wurzeln liegenden Tasks des EUMEL – Systems haben mindestens einen Vorgänger, es besteht also eine 'Vater – Sohn Beziehung' zwischen allen Tasks des Systems.

Grundsätzlich können Dateien ohne besondere Kommandos zur Vater – Task geschickt und von der Vater – Task geholt werden, aber nicht zu beliebigen anderen Tasks.

'Müller' kann eine Datei an 'PUBLIC' schicken und 'Schulze' kann sie dann dort abholen, aber eine direkte Sendung von 'Müller' nach 'Schulze' ist in der Regel nicht möglich.

Zur Sprechweise: jede Task, über die diese Art von 'Dateivermittlung' abgewickelt werden kann, heißt 'Manager – Task'. Jede Task kann zum 'Manager' erklärt werden.

## 1.5. Eine Beispielsitzung

Der Ablauf zur Erstellung eines Schreibens stellt sich im EUMEL – System wie folgt dar:

**SV**

SUPERVISOR aufrufen

```
Terminal ?  
  
EUMEL Version 1.8/M  
  
gib supervisor kommando:  
begin("meine erste task")  
  
ESC ? --> help  
ESC b --> begin("")  
ESC c --> continue("")  
ESC q --> break  
ESC h --> halt  
ESC s --> storage info  
ESC t --> task info
```

Durch das Kommando 'begin ("meine erste task")', welches durch **CR** abgeschlossen werden muß, wird eine Task mit dem Namen 'meine erste task' im Benutzerzweig, also unterhalb von 'PUBLIC' angelegt. Würde diese Task bereits existieren, so könnten Sie sie mit 'continue ("meine erste task")' an das Terminal holen.

```
gib kommando :  
edit ("Rechnung zum 31.12.86")
```

In der Task eröffnen Sie eine Datei mit dem Kommando 'edit ("dateiname")'. Falls diese Datei neu ist, erfolgt eine Kontrollfrage (zur Kontrolle der gewünschten Schreibweise des Dateinamens), die Sie durch **J** bejahen.

Die Datei ist in diesem Beispiel bereits mit etwas Text gefüllt. Tippen Sie einen beliebigen Text ein und beenden Sie die Bearbeitung dieser ersten Datei durch Drücken der Tasten **ESC** **q** (nacheinander!).

```
..... Rechnung zum 31.12.86 ..... Zeile 1 █  
G M D  
Sankt Augustin  
Schloß Birlinghoven  
  
Sehr geehrte Damen und Herren,  
█
```

Um die Arbeit in der Task zu beenden, geben Sie auch an dieser Stelle **ESC** **q** (nacheinander!) ein.

Nach Verlassen der Task ist wiederum die EUMEL-Tapete auf dem Bildschirm. Jede weitere Aktion wird wiederum von hier aus durch **SV** begonnen. Insbesondere zum Ausschalten des Geräts muß nach **SV** die Task 'shutup' angestoßen werden (siehe auch Anhang I).

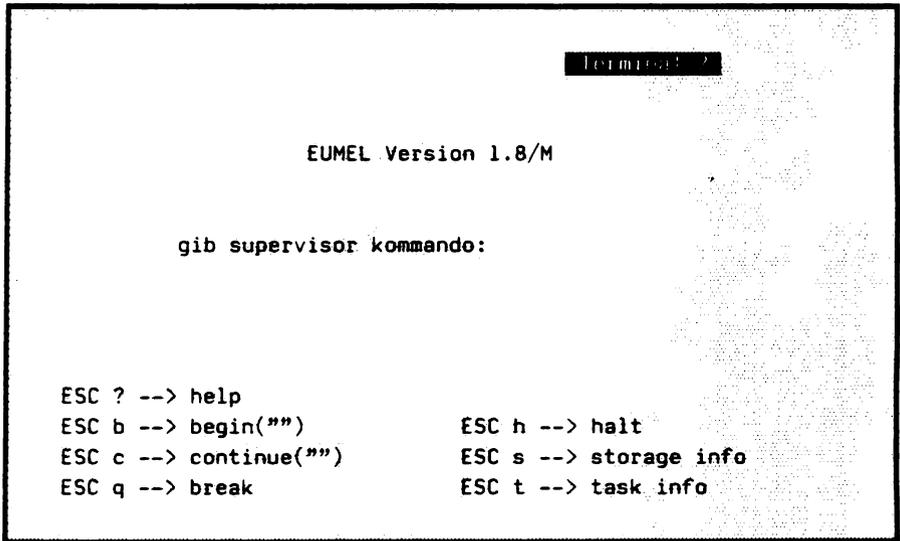
## TEIL 2: Der Supervisor

### 2.1. Steuerkommandos

Jegliche Aktivität im EUMEL-System beginnt mit dem Aufruf des SUPERVISOR durch Drücken der Taste

**SV**

Dieser Tastendruck koppelt Ihr Terminal an den Rechner an. Dieser Vorgang ist auch dann nötig, wenn diese Geräte praktisch eine Einheit bilden.



```
Terminal 2  
EUMEL Version 1.8/M  
gib supervisor kommando:  
  
ESC ? --> help  
ESC b --> begin("")  
ESC c --> continue("")  
ESC q --> break  
ESC h --> halt  
ESC s --> storage info  
ESC t --> task info
```

Die auf den unteren Informationszeilen angezeigten Kommandos stehen nun zur Auswahl. Für alle diese Kommandos gilt, daß sie entweder durch zwei aufeinanderfolgende Tastendrücke **ESC** und Kennzeichen oder auch durch vollständiges Eintippen eingegeben werden können und mit **CR** ausgelöst werden.

Die Eingabe eines falschen Zeichens nach **ESC** oder eines falschen Kommandos wird abgewiesen. Die Eingabe ist dann zu wiederholen.

## Bedeutung der Kommandos:

### 1) Steuerkommandos

ESC b	begin ("taskname")	Task einrichten.
ESC c	continue ("taskname")	Task wieder ankoppeln.
ESC q	break	Terminal abkoppeln.
ESC h	halt	Programmlauf abbrechen.

### 2) Informationskommandos (nur Supervisor)

ESC ?	help	Information.
ESC s	storage info	Belegten Speicherplatz anzeigen.
ESC t	task info	Im System befindliche Tasks anzeigen.

## 2.2. Eine Task einrichten

*Mit dem Kommando 'begin' wird eine neue Task eingerichtet.*

Zunächst koppeln Sie Terminal und Rechner, dann legen Sie eine neue Task an.

Terminal ankoppeln: **SV**

**ESC** **b**

Die Tastenkombination 'ESC b' schaltet den den Einfügemodus ein und positioniert den Cursor passend für die Eingabe des Tasknamens.

```
gib supervisor kommando :  
begin ("█")
```

Dateinamen eintippen:

```
gib supervisor kommando:  
begin ("taskname")
```

Nachdem Sie den Namen eingegeben haben, betätigen Sie die **CR** Taste. Daraufhin meldet sich der Monitor der neuen Task und Sie können beliebige Monitor-Kommandos (siehe Teil 3) eingeben.

Wird eine Task in dieser geschilderten Weise neu eingerichtet, so wird sie automatisch (von der Task SUPERVISOR) als Sohn der Task PUBLIC angelegt.

Soll eine Task nicht als Sohn von PUBLIC, sondern als Sohn einer anderen Task angelegt werden, so ist das Kommando 'begin' mit zwei Parametern zu geben. Die neue Task wird dann als Sohn einer anderen Manager – Task angelegt (siehe Teil 3).

```
gib supervisor kommando:  
begin ("taskname", "name der vatertask")
```

**ACHTUNG:** Die Task, die als Vater – Task angegeben wird, muß eine Manager – Task sein, sonst passiert überhaupt nichts! (s. Kap. 3.1.2.)

## Task wiederankoppeln

*Mit dem Kommando 'continue' wird eine existierende Task an das Terminal angekoppelt.*

Wenn Sie die Arbeit in einer Task wiederaufnehmen wollen, holen Sie die Task mit dem Kommando 'continue' an das Terminal. Dieser Vorgang ähnelt dem Einrichten einer neuen Task:

Terminal ankoppeln: **SV**

**ESC** **c**

Die Tastenkombination 'ESC c' schaltet den den Einfügemodus ein und positioniert den Cursor passend für die Eingabe des Tasknamens.

```
gib supervisor kommando:  
continue ("taskname")
```

Nach dieser Eingabe finden Sie die wiederaufgenommene Task so vor, wie Sie sie verlassen haben.

## Terminal abkoppeln

*Mit dem Kommando 'break' wird das Terminal vom Rechner abgekoppelt.*

Wenn Sie beispielsweise nach einem Informationskommando (siehe Teil 2.3.ff) das Terminal sofort vom Rechner abkoppeln möchten, geben Sie das 'break' – Kommando. Nach 'storage info' geht es jedoch nur mit **SV** weiter.

```
gib supervisor kommando:  
break
```

Nach dieser Eingabe ist das Terminal abgekoppelt. Jede neue Aktivität ist wiederum mit **SV** einzuleiten.

## Laufendes Programm stoppen

*Mit dem Kommando 'halt' wird ein Programm gestoppt, das am betreffenden Terminal läuft.*

Dieses Kommando ist in besonderen Fehlersituationen von Wichtigkeit. Falls Sie ein Programm abbrechen wollen, aber keine regulären Eingaben am Bildschirm mehr möglich sind, so geben Sie zunächst `SV` ein.

Sobald der Supervisor – Bildschirm erscheint, drücken Sie die Tasten

`ESC` `h` (oder tippen 'halt' und drücken 'CR').

```
gib supervisor kommando:  
halt
```

Nach dieser Eingabe wird das an diesem Terminal laufende Programm unterbrochen. Nach dem Abbruch kommen Sie wieder auf die Monitor – Ebene (s. Teil 3).

## 2.3. Informationskommandos

*Mit den Informationskommandos können Informationen zum System abgerufen werden.*

Die folgenden Informationskommandos können direkt an den SUPERVISOR gegeben werden.

Terminal ankoppeln: **SV**

**ESC** **s**

beziehungsweise

```
gib supervisor kommando :  
storage info
```

gibt Auskunft über den belegten Speicherplatz auf dem EUMEL – Hintergrundspeicher.

Das Kommando:

```
gib supervisor kommando :  
task info
```

gibt Auskunft über die Namen der im EUMEL-System befindlichen Tasks und die Struktur des Taskbaums. Verzweigungen im Taskbaum sind durch Einrückungen in den Ebenen des Taskbaums dargestellt.

Alle in dem Schema der Task-Organisation (siehe Teil 1) fettgedruckten Tasks sind auch auf jedem Multi-User-System zu finden, da sie zum Betrieb nötig sind.

Die unterhalb von PUBLIC gelegenen Tasks werden, falls überhaupt schon vorhanden, häufig nach ihrem 'Besitzer' oder der in ihnen erledigten Arbeit benannt sein.

## 2.4. Übersicht über Supervisor – Kommandos

*In diesem Abschnitt werden alle Supervisor- und Task-Kommandos in der ELAN – Notation dargestellt.*

Die Supervisor-Kommandos entsprechen – wie alle anderen Kommandos im EUMEL-System – der ELAN-Syntax (Kommando-Namen werden klein geschrieben, Parameter in Klammern, mehrere Parameter durch Kommata getrennt, TEXT-Parameter in Anführungsstrichen usw.).

### Die ELAN – Notation

Diese Notation dient der präzisen Beschreibung von Konstrukten der Programmiersprache ELAN. Im Anschluß an die teilweise eher informelle Formulierung innerhalb des Kapitels folgt jedem Teil eine Kurzbeschreibung der zu diesem Themenkreis gehörigen Konstrukte.

Eine solche Beschreibung hat z.B. die Form:

`PROC edit (TEXT CONST dateiname)`

Die klein geschriebenen Benennungen von Prozeduren, Parametern etc. sind hoffentlich selbsterklärend, die groß geschriebenen Begriffe sind sogenannte Schlüsselworte und haben folgende Bedeutung:

**OP** Operator  
Ein Operator bewirkt eine elementare Operation. Operatoren werden stets durch Großbuchstaben oder Sonderzeichen dargestellt.

Beispiel: + ( Addition zweier Zahlen)

**PROC** Prozedur  
Programm, welches unter seinem Namen aufrufbar ist, ggf. unter Anfügung von Parametern. **CR** beendet die Eingabe und läßt das Programm ablaufen.

Beispiel: 'edit ("dateiname")'

**CONST** Konstante  
Unveränderbarer Wert.

**VAR** Variable  
Veränderbarer Wert.

**BOOL** Wahrheitswert  
Typ, der nur die Werte TRUE oder FALSE annehmen kann.

**TEXT** Text  
Typ, der alle Buchstaben, Sonderzeichen, aber auch Ziffern umfaßt. Eine TEXT CONST ist somit eine sogenannte Zeichenkette:

"meine datei"  
"\$abc123(XYZ)"  
"abrechnung vom 30.09.86"

Eine im Editor erstellte Datei besteht ausschließlich aus TEXTen. Ein Text wird in Anführungszeichen " " eingeschlossen.

**INT** Integer  
Ganze Zahl. Ein INT CONST ist also irgendeine ganze Zahl. Falls beschrieben ist: 'INT CONST zeilennr', so ist gemeint, daß an dieser Stelle die Zeilennummer der gewünschten Zeile der Datei anzugeben ist, also '25' oder '999'.

**REAL** Real  
Reelle Zahl. Eine REAL CONST bezeichnet eine Zahl mit Dezimalpunkt.

PROC sin (REAL CONST x) => sin (0.5)

**TASK** Task  
Eine TASK CONST bezeichnet eine existierende Task durch einen internen Task – Bezeichner.

#### **THESAURUS**

Ein THESAURUS ist eine Liste von Namen, z.B. eine Liste von Dateinamen.

Folgende Supervisor – Kommandos stehen zur Verfügung:

### ***begin***

PROC begin (TEXT CONST task name)

Richtet eine neue Task als Sohn von PUBLIC ein.

PROC begin (TEXT CONST task name, father task name)

Richtet eine neue Task als Sohn der 'fathertaskname' – Task ein.

### ***break***

PROC break

Das Terminal wird vom Rechner abgekoppelt.

### ***continue***

PROC continue (TEXT CONST task name)

Eine existierende Task wird an das Terminal des Benutzers angekoppelt.

### ***halt***

PROC halt

Das laufende Programm der dem Terminal aktuell zugeordneten Task wird abgebrochen. Natürlich wird die Task nicht gelöscht.

Genauer:

Es wird der Fehler 'halt from terminal' induziert. Normalerweise wird das Programm dadurch wie durch jeden anderen Fehler abgebrochen. Genaueres findet man im Systemhandbuch unter Fehlerbehandlung.

### ***storage info***

PROC storage info

Informationsprozedur über den Hintergrund – Speicher.

### ***task info***

PROC task info

Informiert über alle Tasknamen im System unter gleichzeitiger Angabe der Vater/Sohn – Beziehungen durch Einrückungen.

## TEIL 3: Der Monitor

### 3.1. Der Monitor

*Der Monitor nimmt die Kommandos des Benutzers entgegen, die dieser innerhalb seiner Task gibt. In diesem Kapitel sind die gebräuchlichen Kommandos zur Textverarbeitung beschrieben.*

Der Monitor ist der Empfänger der Kommandos in einer EUMEL-Task. Jede Task und somit jeder aktive Benutzer eines EUMEL-Systems verfügt über einen eigenen Monitor. Halten Sie sich an dieser Stelle vor Augen:

*Der SUPERVISOR ist die Task, die die Aufsicht über alle weiteren Tasks des EUMEL-Systems hat.*

*Der Monitor ist der Befehlsempfänger in Ihrer Task. Jede Task hat einen Monitor.*

Der Monitor in Ihrer Task macht sich direkt bemerkbar durch die Zeile:

`gib kommando :`

Die Kommandos der Monitor-Ebene, die Sie an dieser Stelle eingeben können, lassen sich in Gruppen ordnen:

### Informationskommandos

storage info	Belegten Speicherplatz anzeigen.
task info	Im System befindliche Tasks anzeigen.
task status	Zustand der Task anzeigen.

### Kommandos zur Tasksteuerung

break	Task abkoppeln.
end	Task löschen.
global manager	Task zum Manager machen, d.h. Sohn – Tasks können eingerichtet werden.

### Kommandos zur Dateibearbeitung

copy ("dateiname", "kopie")	Datei kopieren.
edit ("dateiname")	Editor aufrufen.
forget ("dateiname")	Datei löschen.
list	Dateien auflisten.
rename ("dateiname", "neu")	Datei umbenennen.

### Transport von Dateien

fetch ("dateiname")	Datei von Vater – Task holen.
erase ("dateiname")	Datei in Vater – Task löschen.
save ("dateiname")	Datei zu Vater – Task senden.

### Archiv – Kommandos

archive ("name")	Archivlaufwerk reservieren.
fetch ("dateiname", archive)	Datei von Archiv holen.
save ("dateiname", archive)	Datei auf Archiv schreiben.
list (archive)	Inhalt des Archivs listen.
check ("dateiname", archive)	Datei auf Lesbarkeit prüfen.
clear (archive)	Archiv löschen/umbenennen.
format (archive)	Archivdiskette für Benutzung vorbereiten.

### Textverarbeitung

list fonts	Nennt die eingestellten Schrifttypen.
fonttable	Stellt die gewünschte Fonttabelle ein.
lineform ("dateiname")	Zeilenweise formatieren.
autoform ("dateiname")	Automatische Zeilenformatierung.
pageform ("dateiname")	Seitenweise formatieren.
autopageform ("dateiname")	Automatische Seitenformatierung.
index ("dateiname.p")	Stichwortverzeichnis erzeugen.
outline ("dateiname")	Übersicht bzw. Kurzfassung eines Textes erstellen.
print ("dateiname")	Datei drucken.

### Passwortschutz

task password ("geheim")	Passwort für existierende Task festlegen.
begin password ("geheim")	Passwort für neue Task festlegen.
family password ("geheim")	Passwort für mehrere Tasks festlegen.
enter password ("ge/heim")	Passwort für Datei festlegen.

### 3.1.1. Informationskommandos

*Mit den Informationskommandos können Informationen zur eigenen Task bzw. zum gesamten System abgerufen werden.*

Bereits von Teil 2 bekannt sind die Informationskommandos 'ESC s' und 'ESC t'. Auf der Monitor – Ebene ist die abkürzende Schreibweise nicht voreingestellt.

```
gib kommando :  
storage info
```

gibt Auskunft über den belegten Speicherplatz auf dem EUMEL – Hintergrundspeicher.

```
gib kommando :  
task info
```

gibt Auskunft über die Namen der im EUMEL – System befindlichen Tasks und die Struktur des Taskbaums.

### Teil 3: Der Monitor

Auf Monitor-Ebene kann durch zusätzliche Angabe einer Zahl zwischen 1 und 3 Zusatzinformation angefordert werden.

```
gib kommando :  
task info (2)
```

liefert:

```
26.11.86  10:10                                CPU  PRIO  CHAN  STATUS  
SUPERVISOR..... 0001:08:50  0  -   wait  
-..... 0000:00:08  0  2   i/o  
-..... 0000:01:45  0  -   wait  
SYSUR..... 0000:01:48  0  -   wait  
  configurator..... 0000:00:43  0  -   wait  
  OPERATOR..... 0000:00:03  0  -   i/o  
  shutup dialog..... 0000:03:08  0  -   i/o  
  ARCHIVE..... 0000:03:03  0  31  wait  
UR..... 0000:00:43  0  -   wait  
  PUBLIC..... 0000:01:26  0  -   i/o  
  agfa..... 0000:00:11  0  -   i/o  
  werner..... 0000:06:00  0  -  -busy-
```

'task info (1)' entspricht dabei dem Kommando ohne Parameterangabe, '(2)' liefert zusätzlich die verbrauchte CPU-Zeit (= reine Rechenzeit), die Priorität, den Kanal (siehe S.) und den Taskstatus für jede Task des Systems. '(3)' liefert neben diesen Angaben auch noch den belegten Speicherplatz jeder Task. Die Ausführung von task info (3) ist sehr zeitaufwendig!

Um insbesondere den belegten Speicherplatz der eigenen Task anzusehen, aber auch die übrigen der oben erwähnten Angaben, benutzt man das Kommando:

```
gib kommando :  
task status
```

```
24.12.86 18:30 TASK: wenni
```

```
Speicher: 1000K
```

```
CPU-Zeit: 0000.01:11
```

```
Zustand : -busy-, (Prio 1), Kanal 1
```

```
gib kommando :
```

## 3.2. Tasksteuerung

### Task abkoppeln

*Mit dem Kommando 'break' wird eine Task vom Terminal abgekoppelt.*

Durch die Eingabe des Kommandos 'break' auf Monitor-Ebene wird die Task vom Terminal abgekoppelt. Dieses Kommando bewirkt ansonsten keine Veränderungen.

```
gib kommando :  
break
```

Statt 'break' einzutippen, können Sie auch die Tastenkombination

**ESC** **q**

benutzen.

## Eine Manager – Task erzeugen

*Eine Task kann zum Manager, d.h. zum Kommunikationspartner anderer Tasks, erklärt werden. Insbesondere zwischen Manager – Tasks und anderen, die zueinander in Vater – Sohn – Beziehung stehen, ist ein einfacher Dateitransfer möglich (siehe S.19).*

Normalerweise werden Benutzertasks als Sohn der Task PUBLIC eingerichtet. Es kann jedoch wünschenswert sein, selbst eine Task – Hierarchie aufzubauen und eine vorhandene Task zum Vater einer oder mehrerer in Zukunft einzurichtender Tasks zu machen, um somit auch eine Dateihierarchie mit den benötigten Operationen zu erhalten. Dazu wird diese Task zum 'Manager' erklärt:

```
gib kommando:  
global manager
```

Durch das 'global manager' – Kommando wird implizit ein 'break' – Kommando gegeben, so daß Sie nach Eingabe dieses Kommandos wieder ein Supervisor – Kommando geben können. Wenn Sie nun zu irgendeinem Zeitpunkt diese (zunächst potentielle) Vater – Task wieder ankoppeln ('continue' – Kommando), meldet sich die Task nicht wie gewohnt mit 'gib kommando :', sondern mit:

```
maintenance :
```

um anzudeuten, daß es sich um eine Manager – Task handelt.

Um eine Sohn-Task unterhalb einer Manager-Task einzurichten, wird zur Erzeugung dieser neuen Task nicht nur der gewünschte Name, sondern auch der Name der Vater-Task angegeben.<sup>1)</sup>

```
Terminal 2

EUMEL Version 1.8/M

gib supervisor kommando:
begin("sohntask","vatertask")

ESC ? --> help
ESC b --> begin("")
ESC c --> continue("")
ESC q --> break
ESC h --> halt
ESC s --> storage info
ESC t --> task info
```

In dieser Sohn-Task können dann mit einfachen Kommandos Dateien von der Vater-Task geholt und zur Vater-Task geschickt werden.

---

<sup>1)</sup> Falls keine Vater-Task angegeben wird, so ist die neue Task Sohn der Manager-Task 'PUBLIC'.

Soll eine Task für alle anderen Tasks, nicht nur für Sohn – Tasks, des Gesamtsystems als Kommunikationspartner erreichbar sein, so muß diese Task als freier Manager deklariert werden:

```
gib kommando:  
free global manager
```

Auf eine solche Task kann von jeder anderen Task aus zugegriffen werden, ansonsten gilt das für gewöhnliche Manager – Tasks gesagte.

## Task löschen

*Eine Task kann mit dem 'end' Kommando gelöscht werden (und mit ihr alle Daten).*

Normale Benutzertasks werden meistens zweck – oder themengebunden eingerichtet. Sind die Aufgaben im Zusammenhang mit einer solchen Task erledigt, so sollte die Task gelöscht werden, nachdem alle Daten von Bedeutung auf Diskette gesichert wurden.

```
gib kommando :  
end
```

Wie bei allen Löschungen im EUMEL – System folgt eine Kontrollfrage durch den Monitor:

```
gib kommando :  
end  
task "taskname" löschen (j/n) ?
```

Nur bei der positiven Antwort wird gelöscht, alle Dateien sind unwiderruflich verloren. Als positive Antwort auf derartige Abfragen wirken:

j  J  y  Y

n bzw.  N unterdrücken die angebotene Aktion, andere Eingaben werden abgewiesen.

**ACHTUNG:** Wird eine Manager – Task gelöscht, so werden alle Sohn – Enkel – etc. Tasks ohne zusätzliche Nachfrage gelöscht, falls die Sicherheitsabfrage mit 'j' beantwortet wurde.

## 3.3. Dateibearbeitung

### Datei einrichten

*Mit dem Editoraufruf 'edit' wird eine neue Datei eingerichtet, bzw. eine existierende Datei zur Bearbeitung am Bildschirm gezeigt.*

Eine Datei enthält Texte, die logisch zusammengehören und sie wird über ihren Namen eindeutig gekennzeichnet.

Das EUMEL-System speichert einmal geschriebene Texte, bis sie vom Benutzer gelöscht werden. In der Regel wird nicht nur ein (langer) Text oder ein Programmtext geschrieben, sondern mehrere und unterschiedliche. Um diese auseinanderhalten zu können, versehen wir sie jeweils mit einem Namen, der frei gewählt werden kann. Beispiele für Namen:

"Brief vom 1.12.86"  
"1. Kapitel meines Buches"

Eine Sammlung von Zeichen (also im Normalfall unsere geschriebenen Texte), die mit einem Namen versehen worden ist, nennt man eine **Datei**. Der Editor erstellt also eine Datei, wenn wir einen Text schreiben. Eine Datei kann bis zu 4 000 Zeilen fassen, wobei jede bis zu 32 000 Zeichen lang sein darf.

Einrichten der ersten Datei in Ihrer Task:

Der erste Schritt sollte darin bestehen, daß Sie sich einen vernünftigen Dateinamen ausdenken. Das EUMEL-System legt Ihnen praktisch keine Beschränkungen über Länge oder Form des Dateinamens auf, deshalb sollten Sie sich angewöhnen, Ihre Dateien so zu benennen, daß Sie anhand des Namens auch nach einer Woche oder länger noch erahnen können, was diese Datei enthält.

### Teil 3: Der Monitor

---

Ein guter Name für die erste Datei wäre zum Beispiel: "meine erste Datei" oder "werners test vom 1.12.86". Im weiteren Text steht nur "dateiname" o. ä.. Setzen Sie dafür den von Ihnen gewählten Namen ein.

```
gib kommando:  
edit ("dateiname")
```

```
gib kommando:  
edit ("dateiname")  
  
"dateiname" neu einrichten (j/n)?
```

Drücken Sie 'j', so wird eine neue Datei unter dem von Ihnen eingegebenen Namen eingerichtet. Die Datei ist zunächst leer:

```
..... dateiname ..... Zeile 1
```

Welche Möglichkeiten Sie bei der Bedienung des Editors haben, können Sie in Teil 4 nachschlagen. Schreiben Sie an dieser Stelle nur einige Worte in die Datei, anhand derer der Dateiinhalt wiederzuerkennen ist. Sie können die Tastatur genauso benutzen wie die einer Schreibmaschine.

```
..... dateiname ..... Zeile 1  
Inhalt der ersten Datei. 1234567890
```

Die Datei sollte an dieser Stelle wieder geschlossen werden.

Drücken Sie dazu **ESC** **q**.

Dabei ist gleichgültig, wo der Cursor steht.

Wiederholen Sie das Neuanlegen einer Datei mit einer zweiten Datei "anderer dateiname". Bitte schreiben Sie wiederum einige Zeichen in die Datei.

```
gib kommando :  
edit ("anderer dateiname")
```

Vorschlag zur Eingabe:

```
..... anderer dateiname ..... Zeile 1  
Halten Sie irgendeine Taste gedrückt.....
```

Beenden Sie die Arbeit ebenfalls mit **ESC** **q**.

## Dateinamen auflisten

Mit dem Kommando 'list' werden die Dateinamen der Dateien in der Task ausgegeben.

Vor jedem Dateinamen wird das Datum der letzten Bearbeitung der Datei angezeigt.

```
gib kommando:  
list
```

```
..... |!..... Zeile 1 ■  
01.08.86      "dateiname"  
01.08.86      "anderer dateiname"
```

Auch bei dieser Auflistung der Dateinamen handelt es sich um eine EUMEL-Datei (allerdings um eine schreibgeschützte), die Ausgabe wird also wie gewohnt durch das Kommando **ESC** **q** beendet.

## Datei duplizieren

*Mit dem Kommando 'copy' wird eine existierende Datei dupliziert.*

Eine existierende Datei kann dupliziert werden durch das Kommando :

```
gib kommando:  
copy ("dateiname", "kopienname")
```

Durch dieses Kommando wird eine Kopie der Datei "dateiname" unter dem Namen "kopienname" angelegt, der Inhalt der beiden Dateien ist zunächst identisch. Kontrollieren Sie die Richtigkeit dieser Behauptung, indem Sie nachsehen, ob der Inhalt der kopierten Datei gleich dem Inhalt der Ursprungsdateiname ist:

```
gib kommando:  
edit ("kopienname")
```

## Dateinamen ändern

*Mit dem Kommando 'rename' wird der Name einer Datei geändert.*

Sollte Ihnen der Name einer Datei nicht gefallen, so besteht die Möglichkeit, den Namen zu ändern:

```
gib kommando:  
rename ("dateiname", "neuer dateiname")
```

## Datei löschen

*Mit dem Kommando 'forget' wird eine Datei gelöscht.*

Das Löschen einer Datei wird durch das Kommando:

```
gib kommando:  
forget ("neuer dateiname")
```

eingeleitet. Aus Gründen der Sicherheit erfolgt vor der Ausführung des Kommandos jedoch die Abfrage:

```
gib kommando:  
forget ("neuer dateiname")  
  
"dateiname" löschen ? (j/n)
```

Als positive Antwort auf derartige Abfragen wirken:  j  J  y  Y

n bzw.  N unterdrücken die angebotene Aktion, andere Eingaben werden abgewiesen.

## Dateien verschicken

*Dateien können zur Vater – Task geschickt und von der Vater – Task geholt werden.*

Die Vereinbarung, daß Dateien in einer Task lokal sind, d.h. daß nur in dieser Task Zugriff auf die Daten möglich ist, ist häufig zu einschränkend. So kann es zweckmäßig sein, von mehreren Arbeitsplätzen (= Tasks) aus die wesentlichen Ergebnisse an einer zentralen Stelle zu sammeln oder Ergebnisse aus Tasks, die nur kurzzeitig für eine spezielle Aufgabe eingerichtet wurden, länger aufzubewahren.

Zu diesem Zweck wird eine Benutzertask zum Manager erklärt (siehe S.8) und es werden Söhne dieser Task eingerichtet.

## Datei zur Vater – Task schicken

Mit dem Kommando 'save' wird die Kopie einer Datei zur Vater – Task geschickt.

```
gib kommando:  
save ("dateiname")
```

Wird eine Datei an die Vater – Task gesendet, wird eine Kopie der Ursprungsdateiname unter dem Namen 'dateiname' in der Vater – Task eingerichtet. Danach sind diese beiden, zunächst gleichen Dateien unabhängig voneinander. Änderungen, welcher Art auch immer, haben keinen Einfluß auf die namensgleiche Kopie in der anderen Task.

Falls in der Vater – Task bereits eine Datei mit dem Namen 'dateiname' existiert, sei es durch Zufall oder weil bereits einmal eine 'save' – Operation durchgeführt worden ist, erfolgt eine Abfrage:

```
gib kommando:  
save ("dateiname")  
  
"dateiname" überschreiben ? (j/n)
```

Nur wenn die positive Eingabe 'j' erfolgt, wird die Datei in der Vater – Task durch die eigene Datei überschrieben.

## Datei von der Vater – Task holen

*Mit dem Kommando 'fetch' wird die Kopie einer Datei von der Vater – Task geholt.*

Entsprechend dem Versenden einer Dateikopie können Sie eine Kopie von der Vater-Task holen und ggf., natürlich nach Abfrage, Ihre Datei dieses Namens überschreiben.

```
gib kommando:  
fetch ("dateiname")
```

## Datei in der Vater – Task löschen

*Mit dem Kommando 'erase' wird eine Datei in der Vater – Task gelöscht.*

Soll eine Datei in der Vater – Task gelöscht werden, so kann dieses, dem 'forget' – Kommando analoge Kommando, in der Sohn – Task gegeben werden:

```
gib kommando:  
erase ("dateiname")
```

Falls die Datei in der Vater – Task existiert, wird sie nach Kontrollfrage gelöscht.

```
gib kommando:  
erase ("dateiname")  
"dateiname" loeschen (j/n) j  
  
gib kommando :
```

Anm: Die Task 'PUBLIC' ist grundsätzlich eine Manager – Task. Da Benutzer – Tasks als Sohn von 'PUBLIC' eingerichtet werden, falls Sie nicht als Sohn einer besonderen Manager – Task eingerichtet wurden, beziehen sich 'fetch' – , 'save' – und 'erase' – Kommandos auf 'PUBLIC'.

## 3.4. Das Archiv

*Das Archiv dient der Speicherung von Dateien auf Disketten (Sicherung).*

Das Archiv übernimmt im EUMEL – System die Verwaltung der langfristigen Datenerhaltung. Das Archiv sollen Sie benutzen, um:

- Sicherungskopien wichtiger Dateien außerhalb des Rechners zu besitzen;
- nicht benötigte Dateien außerhalb einer Task zu halten (Speicherplatzersparnis!);
- Dateien auf andere Rechner zu übertragen.

Das Archiv wird im EUMEL – System durch die Task 'ARCHIVE', die das Diskettenlaufwerk des Rechners verwaltet, realisiert. Die Steuerung durch eine Task hat für Sie die erfreuliche Folge, daß die Handhabung des Archivs sich kaum von den schon bekannten Dateioperationen unterscheidet. In den Kommandos wird zusätzlich angegeben, daß das Archiv angesprochen werden soll.

## Archiv – Kommandos

*Der Arbeitsablauf bei Benutzung des Archivs besteht immer aus der Reservierung, dem Lese- oder Schreibzugriff und der Freigabe des Archivs nach Ende der Arbeit. Jede Arbeit mit dem Archiv beginnt mit dem Reservierungskommando.*

Als ersten Schritt der Archivbenutzung müssen Sie das Archiv reservieren, das heißt der Verwaltung Ihres EUMEL-Systems mitteilen, daß Sie die Task 'ARCHIVE', die der Steuerung des Diskettenlaufwerks dient, für Ihre Task arbeiten lassen. Solange für Ihre Task das Archiv reserviert ist, kann keine andere Task das Archivlaufwerk benutzen.

Für die Reservierung müssen Sie bei Benutzung einer schon vorbereiteten oder sogar beschriebenen Diskette den Namen dieser Archivdiskette kennen (er sollte auf dem Diskettenaufkleber stehen) oder vor Benutzung einer neuen Diskette einen Namen festlegen (und auf dem Aufkleber vermerken). Wie gewohnt gibt es keine Vorschriften für die Namensgebung.

Erst nachdem Sie das Reservierungskommando gegeben haben:

```
gib kommando:  
archive ("diskettenname")
```

sollten Sie die Diskette in das Laufwerk einschieben, um zu verhindern, daß ein anderer Benutzer, der das Archiv bereits für sich reserviert hat, auf Ihrer zufällig gleichnamigen Datei arbeitet.

Eine Datei wird mit dem Kommando:

```
save ("dateiname",archive)
```

auf eine Diskette geschrieben und mit dem Kommando:

```
fetch ("dateiname",archive)
```

von einer Diskette geholt.

Das Inhaltsverzeichnis einer Diskette erhalten Sie durch:

```
list (archive)
```

## Benutzung einer neuen Archivdiskette

*Eine neue Diskette muß für die Benutzung vorbereitet (formatiert) werden.*

Vor der erstmaligen Benutzung einer Archivdiskette muß diese formatiert, d.h. in Spuren und Sektoren für die Positionierung des Schreib-/Lesekopfes des Diskettenlaufwerks eingeteilt werden, um überhaupt ein Beschreiben der Diskette zu ermöglichen. Die Einteilung ist geräteabhängig, häufige Formate sind:

40 Spuren zu je 9 Sektoren (360 K)

80 Spuren zu je 9 Sektoren (720 K).

Die Erstbenutzung einer Archivdiskette erfordert nach der Reservierung des Archivs das Kommando:

```
gib kommando:  
format (archive)
```

Erst nach einer Kontrollabfrage:

```
gib kommando:  
format (archive)  
  
Archiv "diskettenname" formatieren ? (j/n)
```

wird tatsächlich formatiert und die Diskette steht mit dem Namen "diskettenname" für Archivoperationen zur Verfügung.

**ACHTUNG:** Wird eine bereits beschriebene Diskette noch einmal formatiert, so sind alle Daten, die auf der Diskette waren, verloren.

Bei einigen Rechnern ist es möglich, die Formatierung zu variieren. Falls beim Formatieren auf einem solchen Rechner ein anderes als das Standardformat erzeugt werden soll, so ist die Codierung des gewünschten Formats mitanzugeben.

Beispiel: Für ein Gerät mit 5 1/2 Zoll Disketten wäre z.B. einstellbar:

- code 0 : Standardformat
- code 1 : 40 Spuren
- code 2 : 80 Spuren
- code 3 : High Density

'format (archive)' erzeugt ebenso wie 'format (0,archive)' eine standard-formatierte Diskette, 'format (3,archive)' erzeugt eine High Density Formatierung.

---

## Diskette löschen / umbenennen

*Bereits benutzte Disketten können wieder gelöscht und auch umbenannt werden.*

Falls Sie den Inhalt einer beschriebenen Archivdiskette löschen oder den Namen einer Diskette ändern wollen, müssen Sie das Archiv unter dem gewünschten Namen reservieren: Falls Sie den Inhalt löschen möchten, tun Sie das unter dem bisherigen und bestehenden Namen. Falls Sie die Diskette umbenennen wollen, reservieren Sie das Archiv unter dem neuen gewünschten Namen. Beachten Sie, daß durch das Umbenennen eines Archivs alle darauf befindlichen Dateien gelöscht werden. Anschließend geben Sie das Kommando:

```
gib kommando:  
clear (archive)
```

Durch die Ausführung des Kommandos erhält die eingelegte Diskette den in der Reservierung angegebenen Namen. Das Inhaltsverzeichnis, das sich auf der Diskette befindet, wird gelöscht. Damit sind die Daten, die sich eventuell auf dieser Diskette befanden, nicht mehr auffindbar. Die Diskette entspricht einer neu formatierten Diskette<sup>1)</sup>.

Eine Neuformatierung ist demnach bei Wiederverwendung der Diskette nicht notwendig.

<sup>1)</sup> Das Kommando 'format' enthält implizit 'clear'.

## Inhaltsverzeichnis der Diskette

Mit 'list (archive)' werden die Dateien auf der Diskette angezeigt.

Eine formatierte Diskette kann nach der Archivanmeldung gelesen oder beschrieben werden. Um zu sehen, welche Dateien auf der Diskette zu holen (= lesen) sind bzw. wieviel Platz zum Beschreiben vorhanden ist, ist es zweckmäßig, zunächst das Inhaltsverzeichnis der Diskette zu betrachten.

```
gib kommando:  
list (archive)
```

Beispiel:

```
..... diskettenname (100 K belegt von 720 K) .....
```

01.05.86	25 K	"rechnungen april"
01.06.86	23 K	"rechnungen mai"
01.07.86	20 K	"rechnungen juni"
01.08.86	32 K	"rechnungen juli"

## Lesen und Schreiben auf Diskette

*Lesen und Schreiben auf der Diskette entspricht den bekannten Operationen zum Senden und Holen von Dateien.*

Das Schreiben einer Datei auf Diskette entspricht dem Übersenden einer Datei an die Vater-Task. Einziger Unterschied ist, daß Sie das Ziel explizit angeben müssen:

```
gib kommando:  
save ("dateiname",archive)
```

Entsprechend funktioniert auch das Lesen einer Datei von der Diskette:

```
gib kommando:  
fetch ("dateiname",archive)
```

Wie auch bei der Kommunikation zwischen Sohn- und Vater-Task werden nur Kopien der Dateien geholt bzw. geschrieben.

## Wechsel der Archivdiskette

Bei Einlegen einer anderen Archivdiskette müssen Sie erneut das Kommando

```
gib kommando:  
archive ("diskettenname")
```

geben, da mit der Archivreservierung zugleich die Prüfung von Diskettenname und –Inhaltsverzeichnis vorbereitet wird.

## Beenden der Archivreservierung

*Nach Benutzung Archiv freigeben!*

Wenn Sie alle gewünschten Arbeiten mit dem Archiv fertiggestellt haben, geben Sie das Archiv wieder frei.

```
gib kommando:  
release (archive)
```

Durch dieses Kommando kann die Task 'ARCHIVE' mit ihren Leistungen von einer anderen Task in Anspruch genommen werden. Falls Sie dieses Kommando nicht gegeben haben aber seit 5 Minuten keine Archivoperation ausgelöst haben, kann eine andere Task durch die Anforderung 'archive("diskettenname")' das Archiv reservieren. Durch diese Maßnahme wird verhindert, daß ein vergeßlicher Benutzer bei einem System mit mehreren Benutzern das Archiv blockiert.

## Fehlermeldungen des Archivs

*Bei Archiv-Operationen kann es zu Fehlersituationen kommen.*

Versucht man, eine Datei vom Archiv zu holen, kann es vorkommen, daß das Archiv-System

Lese-Fehler (Archiv)

meldet und den Lese-Vorgang abbricht. Dies kann auftreten, wenn die Floppy beschädigt oder aus anderen Gründen nicht lesbar ist (z.B. nicht justierte Disketten-Geräte). In einem solchen Fall vermerkt das Archiv-System intern, daß die Datei nicht korrekt gelesen werden kann. Das sieht man z.B. bei 'list (archive)'. Dort ist der betreffende Datei-Name mit dem Zusatz 'mit Lese-Fehler' gekennzeichnet. Um diese Datei trotzdem zu lesen, muß man sie unter ihrem Dateinamen mit dem Zusatz 'mit Lese-Fehler' lesen.

gib kommando:  
fetch ("dateiname mit Lese-Fehler")

Die Datei wird in diesem Fall trotz Lese-Fehler (Informationsverlust!) vom Archiv gelesen.

Um solche Fälle möglichst zu vermeiden, sieht das EUMEL – System die Möglichkeit vor, Archive bzw. Archiv – Dateien nach Beschreiben zu prüfen. Das erfolgt mit dem Kommando

```
gib kommando :  
check ("dateiname", archive).
```

Durch dieses Kommando werden eventuelle Lese – Fehler gemeldet.

Weitere Fehlermeldungen des Archivs:

\* Lesen unmöglich (Archiv)

Die Archiv – Diskette ist nicht eingelegt oder die Tür des Laufwerks ist nicht geschlossen.

= > Diskette einlegen bzw. Tür schließen.

\* Schreiben unmöglich (Archiv)

Die Diskette ist schreibgeschützt.

= > falls wirklich gewünscht, Schreibschutz entfernen.

\* Archiv nicht angemeldet

Das Archiv wurde nicht angemeldet

= > 'archive ("name")' geben.

\* Lese – Fehler (Archiv)

Siehe Lesen unmöglich

\* Schreibfehler (Archiv)

Die Diskette kann nicht (mehr) beschrieben werden.

= > Andere Diskette verwenden.

- \* Speicherengpass  
Im System ist nicht mehr genügend Platz, um eine Datei vom Archiv zu laden, ggf.  
=> ggf. Dateien löschen.
- \* RERUN bei Archiv-Zugriff Das System wurde bei einer Archiv-Operation durch  
Ausschalten bzw. Reset unterbrochen.
- \* "dateiname" gibt es nicht  
Die Datei "dateiname" gibt es nicht auf dem Archiv.  
=> mit 'list(archive)' Archiv prüfen.
- \* Archiv heißt ...  
Die eingelegte Diskette hat einen anderen als den eingegebenen Archivnamen.  
=> Kommando 'archive' mit korrektem Namen geben.
- \* Archiv wird von Task ... benutzt  
Das Archiv wurde von einem anderen Benutzer reserviert.  
=> Abwarten.
- \* "dateiname" kann nicht geschrieben werden (Archiv voll)  
Die Datei ist zu groß für die eingelegte Diskette.  
=> Andere Diskette für diese Datei nehmen.
- \* Archiv inkonsistent  
Die eingelegte Diskette hat nicht die Struktur einer Archiv-Diskette.  
=> 'format (archive)' vergessen.
- \* save/erase wegen Lese-Fehler verboten  
Bei Archiven mit Lese-Fehler sind Schreiboperationen verboten, weil ein Erfolg  
nicht garantiert werden kann.

## 3.5. Kommandos für mehrere Dateien

*Durch Anwendung der besonderen Operatoren 'ALL' und 'SOME' können Sie mehrere Dateien mit einem Kommando behandeln.*

Oft ist es sehr zweckmäßig und erleichternd, einen Befehl für eine ganze Reihe von Dateien wirken zu lassen, wie z.B. beim Archivieren, wenn Sie etwa alle während des Tages veränderten Dateien mit deren neuen Stand auf Diskette schreiben möchten.

Da Tasks einen Namen haben und jede Task ein Inhaltsverzeichnis ihrer Dateien führt, ist es möglich, Listen von Dateien zu benennen.

## Interne Tasknamen

Wenn Sie eine andere als die eigene oder die Vater – Task ansprechen wollen, ist es notwendig, den 'internen Tasknamen' anzugeben. Diese auf den ersten Blick etwas undurchsichtige Forderung hat folgenden Hintergrund:

Durch die in der Einleitung vorgestellte Baumstruktur des EUMEL – Systems ist es ohne besondere Angaben nur möglich, Kommandos zu geben, die die eigene Task ('edit'..) oder die Vater – Task ('save'..) betreffen. Beim Archivieren zum Beispiel wäre es demzufolge erforderlich, eine Datei über den Vater vom Vater vom Vater... an den Sohn des Sohnes... zu schicken, damit die Datei endlich in der Task 'ARCHIVE' landet. Statt dessen verwenden Sie eine Prozedur 'archive', die den internen Taskbezeichner liefert. Damit wird die gewünschte Task intern identifiziert, ohne daß Sie sich darum kümmern müssen.

Wichtige Prozeduren, die interne Taskbezeichner liefern, sind:

myself	Bezeichner der eigenen Task
public	Bezeichner von PUBLIC
father	Bezeichner der Vater – Task <sup>1</sup>
archive	Bezeichner von ARCHIVE
printer	Bezeichner von PRINTER

---

<sup>1</sup> Falls kein besonderer Manager eingerichtet wurde, liefern 'father' und 'public' natürlich dieselbe Task: PUBLIC.

## Dateiverzeichnisse

Jede Task verfügt über ein Verzeichnis der in ihr befindlichen Dateien. Das Verzeichnis Ihrer eigenen Task können Sie mit dem 'list' – Kommando betrachten. Das Verzeichnis einer anderen Task sehen Sie beispielsweise durch das Kommando 'list (archive)'. In diesem Fall müssen Sie dem eigentlichen Kommando den internen Taskbezeichner der gewünschten Task hinzugeben, um das Verzeichnis zu sehen.

Um ein Verzeichnis in Verbindung mit anderen Kommandos benutzen zu können, gibt es besondere Operatoren:

ALL	liefert das gesamte Verzeichnis
SOME	bietet das Verzeichnis zur Auswahl von Einträgen an.

In Verbindung mit einem internen Taskbezeichner wird einer der beiden Operatoren einem Monitor – Kommando als Parameter nachgestellt. Das Kommando wirkt dann nacheinander auf alle im Verzeichnis enthaltenen Dateien.

```
gib kommando:  
fetch (ALL father)
```

Alle Dateien der Vater – Task werden nacheinander geholt, bei Namensgleichheit erfolgt die bekannte Kontrollfrage, ob die gleichnamige Datei in der eigenen Task überschrieben werden soll.

Falls nur einige Dateien des Verzeichnisses bearbeitet werden sollen, wird der Operator 'SOME' dem Taskbezeichner vorangestellt:

```
gib kommando:  
fetch (SOME father)
```

Hier wird zunächst das Dateiverzeichnis der Task angeboten. Streichen Sie alle Dateien, die auf der Diskette sind, aber nicht in Ihre Task geholt werden sollen, aus dem Verzeichnis, indem Sie

- den Dateinamen mit Blanks überschreiben

oder:

- die Zeile mit **HOP** **RUBOUT** löschen

oder:

- mehrere Zeilen markieren, indem Sie zu Beginn des zu markierenden Bereichs 'mark' betätigen und mit Hilfe der Cursor-Tasten den Bereich so weit wie benötigt ausdehnen. Im Anschluß daran können Sie diese Zeilen durch

**ESC** **RUBOUT** oder

**ESC** **p**

löschen.

```
.....  
rechnungen apr 1  
rechnungen mai  
rechnungen juni  
rechnungen juli
```

In obigem Beispiel werden nach dem Kommando 'ESC RUBOUT' (=Löschen der markierten Zeilen) und dem Kommando 'ESC q' (=editieren beenden) die Dateien 'rechnungen juni' und 'rechnungen juli' vom Archiv geholt.

Als weitere Vereinfachung gibt es die Prozedur 'all' als Abkürzung für 'ALL myself'.

Beispiel: alle Dateien auf Archivdiskette schreiben.

```
gib kommando:  
save (all,archive)
```

Für Fortgeschrittene:

Sie können auch aus den Verzeichnissen mehrerer Tasks ein neues Verzeichnis bilden. Zu diesem Zweck sind folgende Mengenoperationen auf Verzeichnisse möglich:

- Differenzmenge
- + Vereinigungsmenge
- / Schnittmenge

Beispiel:

```
fetch (ALL father - ALL myself)
```

Alle Dateien der Vater – Task, die noch nicht in der eigenen Task sind, werden geholt.

## 3.7. Passwortschutz

*Das EUMEL – System ermöglicht Passwortschutz für Dateien, einzelne Tasks und ganze Zweige des Taskbaumes.*

Falls Sie sicherstellen wollen (oder müssen), daß Teile Ihres EUMEL – Systems vor unberechtigter Benutzung geschützt sind, können Sie den Zugriff mit einem Passwort regeln.

Als Passwort können Sie jeden beliebigen Text nehmen. Bedenken Sie jedoch, daß ein wirklich wirksamer Schutz nur dann gewährleistet ist, wenn Sie weder ein triviales Passwort (etwa den eigenen Vornamen) auswählen, noch eines, das Sie selbst nicht behalten. <sup>1)</sup>

**ACHTUNG:** Es gibt ein besonderes Passwort im EUMEL – System: " – ". Dieses Passwort verhindert, daß die Task in der es gegeben wurde (z.B. UR), an ein Terminal geholt wird, es darf folglich nicht für normale Manager – Tasks gegeben werden.

---

<sup>1)</sup> Man darf Passwörter nicht vergessen! Durch Passwörter geschützte Tasks kann niemand – außer durch die Angabe des korrekten Passworts – wieder ankoppeln. Hat man das Passwort vergessen, kann man nur noch die Task löschen.

---

## Eine Task mit Passwort schützen

Das Monitor – Kommando 'task password' sorgt dafür, daß eine Task fortan nur wieder mit einem 'continue' – Kommando 'betreten' werden kann, wenn man vorher das richtige Passwort angibt.

```
gib kommando:  
task password ("rosebud")
```

Versucht nun ein Benutzer, die mit dem Passwort geschützte Task mit dem 'continue' – Kommando an sein Terminal anzukoppeln, wird er zunächst nach dem Passwort gefragt. Nur unter Angabe des Passworts wird die Task angekoppelt.

Bei der Beantwortung des Passworts werden statt der eingegebenen Zeichen Punkte auf den Bildschirm geschrieben. Durch Betätigen von ESC können die getippten Zeichen lesbar gemacht werden.

```
gib supervisor kommando:  
continue("taskname")  
Passwort: .....
```

Der Passwortschutz gewährleistet, daß kein unberechtigter Benutzer direkt an die Dateien und Programme der Task gelangen kann. Es gibt jedoch noch zwei Situationen, die einen unberechtigten Zugang zu Dateien erlauben:

a) Dateien in die Vater – Task schicken:

Transportiert man Dateien in die Vater – Task ('save' – Kommando) können Benutzer auf diese Dateien zugreifen (sofern sie Zugang zu dieser Task haben). Dies kann man verhindern, indem man ein Datei – Passwort angibt. Man beachte, daß das Passwort für Dateien und das oben beschriebene Passwort für Tasks nichts miteinander zu tun haben.

b) Dateien werden in eine Sohn – Task geholt:

Ist die Task als Vater – Task eingerichtet ('global manager' – Kommando), dann ist es möglich, von der Sohn – Task Dateien ('fetch' – Kommando) aus der Vater – Task zu holen, die mit einem Passwort geschützt ist. Darum muß man verhindern, daß unberechtigte Benutzer Söhne einer mit Passwort geschützten Task einrichten können. Das kann man mit dem Kommando

```
maintenance :  
begin password ("geheim")
```

Wird dieses Kommando gegeben, wird man bei dem Versuch, eine Sohn – Task einzurichten, nach einem Passwort gefragt. Beachten Sie, daß das 'begin password' nichts mit dem Task – Passwort und Datei – Passwort zu tun hat.

Man kann einen ganzen Zweig eines EUMEL – Systems durch das Kommando 'family password' vor unberechtigtem Zugriff schützen. Das Kommando:

```
maintenance:  
family password ("geheim")
```

wird dazu (wie gewohnt als Monitor – Kommando) in der Vater – Task des zu schützenden Zweigs des Taskbaumes gegeben. Damit ist das Passwort aller Söhne, Enkel

usw. dieser Task auf 'geheim' gesetzt, falls sie vorher kein Passwort oder das gleiche Passwort wie die aufrufende Task haben. Eine Task in diesem Zweig, die bereits ein eigenes, vom 'family password' verschiedenes Passwort besitzt, behält dieses eigene Passwort.

Bsp: Für 'PUBLIC' wird das Kommando 'family password ("geheim")' gegeben. Dann ist das Passwort von 'PUBLIC' und aller Tasks des Benutzerzweiges auf 'geheim' gesetzt.

Es ist zu beachten, daß bei der Vergabe des 'family password' nur die aktuellen Söhne der Task berücksichtigt werden. Söhne, die nach der Vergabe des 'family password' eingerichtet werden, sind nicht durch dieses Passwort geschützt.

## Passwort löschen

Um ein Passwort zu löschen, geben Sie das Passwort-Kommando mit "" als Parameter:

```
maintenance:  
begin password("")
```

Durch diese Angabe haben Sie den Passwort einen leeren Text als Parameter gegeben, der das bisherige Passwort 'überschreibt'.

## Dateipasswort

Etwas komplizierter gestaltet sich der Passwortschutz für einzelne Dateien einer Manager-Task, da in dieser Anwendung eine Unterscheidung nach Schreib- und Leseschutz vorgenommen wird.

Da in dieser Anwendung nur einige Dateien der Vater-Task vor Lesen ('fetch'), Schreiben ('save', 'erase') oder beidem geschützt werden sollen, benötigt diese Prozedur Angaben über Dateinamen, Schreibpasswort und Lesepasswort.

```
maintenance :  
enter password ("dateiname", "schreibschutz", "leseschutz")
```

Falls die Datei nicht gegen Lesen geschützt werden soll, wird (wie beim Löschen eines Passworts) "" als Lesepasswort angegeben.

Falls Schreiben und/oder Lesen für eine Datei gänzlich verboten<sup>1)</sup> sein soll, so ist "-" als entsprechendes Passwort anzugeben.

Um von einer Sohn-Task eine Datei mit Passwortschutz in der Vater-Task zu lesen oder zu schreiben muß vor dem 'fetch', 'save' oder 'erase' Kommando das 'enter password' Kommando eingegeben werden:

```
gib kommando:  
enter password ("schreibpasswort/lesepasswort")
```

In der Sohn-Task wird also nur ein Passwort eingegeben. Falls wie oben ein '/' in diesem Passwort enthalten ist, wird der erste Teil vor dem '/' als Schreibpasswort und der zweite Teil als Lesepasswort geprüft. Falls kein '/' in dem Passwort enthalten ist, wird das Wort sowohl als Schreib- als auch als Lesepasswort interpretiert.

---

1) Natürlich kann die Datei in der Manager-Task, der sie gehört, normal editiert werden.

Beispiel:

In einer Manager – Task wird eine Datei "texte" eingerichtet, die Textvorlagen enthält. In einigen Sohn – Tasks soll diese Datei geholt (= gelesen) werden können. Die bearbeitete, somit veränderte Datei darf aber nicht zurück in die Vater – Task geschrieben werden.

In der Vater – Task: enter password ("texte", "-", "psw")

In der Sohn – Task : enter password ("psw")

Falls das Passwort in einer Sohn – Task fehlerhaft oder gar nicht eingegeben wurde, erscheint die Meldung :

```
gib kommando :  
fetch ("geschützte datei")  
FEHLER : Passwort falsch
```

Somit kann diese Datei nur von Benutzern, die das Lesepasswort kennen, geholt werden. Ein Überschreiben der Datei ist nicht möglich, da das Schreibpasswort nicht gegeben werden kann (" - " !).

## 3.8. Monitor – Kommandos

### **ALL**

THESAURUS OP ALL (TASK CONST task)

Liefert einen Thesaurus<sup>1)</sup>, der alle Dateinamen der angegebenen Task enthält (auch der Benutzer – Task 'myself').

fetch (ALL father)

THESAURUS OP ALL (TEXT CONST datei)

Liefert einen Thesaurus, der die in 'datei' vorhandenen Dateinamen (jede Zeile ein Name) enthält.

fetch (ALL "dateiliste")

### **archive**

PROC archive (TEXT CONST archivname)

Anmeldung von Archiv – Operationen. 'archivname' wird zur Überprüfung für alle folgenden Archiv – Operationen verwandt, um die unberechtigte Benutzung eines Archivs zu verhindern. Die Anmeldung wird abgelehnt, wenn ein anderer Nutzer das Archiv belegt hat.

archive ("textdiskette")

TASK PROC archive

Liefert den internen Task – Bezeichner für die Verwendung in Dateikommandos.

save ("dateiname", archive)

### **begin password**

PROC begin password (TEXT CONST geheim)

Verhindert das unberechtigte Einrichten einer Sohn – Task.

begin password("gmd")

---

1) Ein Thesaurus ist eine Liste, in diesem Zusammenhang eine Liste von Dateien.  
(Siehe auch 2.4. Die ELAN – Notation)

**break**

PROC break

Die zum Terminal aktuell zugeordnete Task wird abgekoppelt. Sie wird damit zu einer Hintergrund – Task.

**brother**

TASK PROC brother (TASK CONST task)

Liefert den internen Task – Bezeichner der angegebenen "Bruder" – Task.

```
list(brother)
```

**check**

PROC check (TEXT CONST dateiname, TASK CONST task)

Überprüft, ob die Datei 'dateiname' auf dem Archiv lesbar ist.

```
check ("meine datei", archive)
```

PROC check (THESAURUS CONST t, TASK CONST task)

Überprüft, ob die in dem Thesaurus 't' enthaltenen Dateien auf dem Archiv lesbar sind.

```
check (ALL archive, archive)
```

**clear**

PROC clear (TASK CONST task)

Löscht alle Dateien der Task 'ARCHIVE' und benennt die Diskette um, falls ein anderer als der bisherige Diskettenname bei der Reservierung angegeben wurde.

```
archive("disk1"); clear(archive)
```

**copy**

PROC copy (TEXT CONST quelle, ziel)

Kopiert die Datei 'quelle' in eine neue Datei mit dem Namen 'ziel' in der Benutzer – Task.

```
copy("datei", "neue datei")
```

Fehlerfälle: "ziel" existiert bereits  
"quelle" gibt es nicht  
zu viele Dateien

#### **edit**

PROC edit

a) Im Monitor:

Ruft den Editor mit den zuletzt verwandten Dateinamen auf.

b) Im Editor:

Der Dateiname wird erfragt.

Für jedes 'edit' gilt:

Wurde 'edit' zum ersten Mal aufgerufen, nimmt das Fenster den gesamten Bildschirm ein. Bei erneutem 'edit' – Aufruf wird ein Fenster nach rechts unten ab der aktuellen Cursor – Position eröffnet.

PROC edit (TEXT CONST dateiname)

Ruft den Editor mit 'dateiname' auf.

```
edit("handbuch teil3")
```

PROC edit (TEXT CONST dateiname, x, y, xbreite, yhöhe)

Wie obiger 'edit' – Aufruf, jedoch kann das Fenster, in dem 'dateiname' editierbar ist, gesetzt werden. Die Parameter definieren ein Editor – Fenster mit der linken oberen Ecke auf den Bildschirmkoordinaten 'x' und 'y' und einer Zeilenbreite 'xbreite' und 'yhöhe' Zeilen. Wird der Editor mit 'edit ("dateiname")' aufgerufen, wird implizit 'edit ("dateiname", 1, 1, 79, 24)' aufgerufen.

```
edit("notiz",5,5,44,12)
```

PROC edit (THESAURUS CONST t)

Editieren aller in dem Thesaurus 't' enthaltenen Dateien nacheinander.

```
edit (ALL father)
```

#### **end**

PROC end

Die zum Terminal aktuell gehörende Task wird abgebrochen und gelöscht.

**enter password**

PROC enter password (TEXT CONST datei, schreibpass, lesepass)

Die angegebene Datei wird mit Schreib- und Lese Passwort versehen. Die Passwörter werden in der eigenen Task nicht berücksichtigt.

Falls der Schutz total sein soll, so ist für die verbotene Operation "-" als Passwort anzugeben.

```
enter password ("daten","sicher","heit")
```

PROC enter password (TEXT CONST password)

Gibt Schreib- und Lese Passwort für den Austausch mit Manager-Task an. Falls zwei verschiedene Passwörter für Lesen und Schreiben vereinbart sind, so sind sie als ein Text durch "/" getrennt einzugeben.

```
enter password ("lese/schreibpasswort")
```

**erase**

PROC erase (TEXT CONST datei)

Löscht eine Datei mit dem Namen 'name' in der unmittelbaren Vater-Task.

```
erase("alte datei")
```

Fehlerfälle:

```
"datei" gibt es nicht  
Passwort falsch
```

PROC erase (TEXT CONST name, TASK CONST manager)

Löscht eine Datei mit dem Namen 'name' in der Task 'manager'.

```
erase ("dateiname", father)
```

PROC erase (THESAURUS CONST thesaurus)

Löscht die im 'thesaurus' angegebenen Dateien in der Vater-Task. ,

```
erase (ALL myself)
```

```
(* löscht alle Dateien in der Vater-Task, die in der  
Benutzer-Task vorhanden sind *)
```

**PROC erase (THESAURUS CONST thesaurus, TASK CONST manager)**

erase (all, father)

(\* löscht alle Dateien in der Vater-Task, die in der Benutzer-Task vorhanden sind \*)

**father**

**TASK PROC father**

Liefert den internen Task – Bezeichner der Vater – Task der Benutzer – Task.

list(father)

**TASK PROC father (TASK CONST task)**

Liefert den internen Task – Bezeichner von 'task'.

save ("dateiname", father (father))

(\* Kopiert 'dateiname' zum "Großvater" \*)

**fetch**

**PROC fetch (TEXT CONST name)**

Kopieren einer Datei von der Vater – Task in die Benutzer – Task

fetch("sicherungskopie")

Fehlerfälle:

"datei" gibt es nicht

Passwort falsch

zu viele Dateien

**PROC fetch (TEXT CONST name, TASK CONST manager)**

Kopieren einer Datei in die Benutzer – Task von 'manager'.

fetch ("dateiname", /"global")

**PROC fetch (THESAURUS CONST thesaurus)**

Holt alle im 'thesaurus' enthaltenen Dateien von der Vater – Task.

fetch (ALL)

**PROC fetch (THESAURUS CONST thesaurus, TASK CONST manager)**

Zweck: Holt alle im 'thesaurus' enthaltenen Dateien von der 'manager' – Task.

fetch (ALL /"global", /"global")

***forget***

**PROC forget (TEXT CONST datei)**

Löschen einer Datei mit dem Namen 'name' in der Benutzer – Task.

forget ("alte datei")

Fehlerfälle:

"datei" gibt es nicht

**PROC forget (THESAURUS CONST thesaurus)**

Löscht die im 'thesaurus' enthaltenen Dateien in der Benutzer – Task.

forget (SOME myself)

***format***

**PROC format (THESAURUS CONST thes)**

Formatieren von Disketten und Einstellen des Namens.

format(archive)

**PROC format (INT CONST art, THESAURUS CONST thes)**

Formatieren von Disketten im Nichtstandardformat des benutzten Geräts

format(2,archive)

***global manager***

**PROC global manager**

Durch den Aufruf der Prozedur wird die Benutzer – Task zu einem Datei – Manager. Danach können Söhne dieser Task eingerichtet werden.

***list***

PROC list

Listet alle Dateien der Benutzer – Task mit Namen und Datum des letzten Zugriffs auf dem Terminal auf.

PROC list (TASK CONST task)

Listet alle Dateien der angegebenen 'task' mit Namen und Datum der letzten Änderung auf dem Terminal auf.

list (father)

***myself***

TASK PROC myself

Liefert den internen Task – Bezeichner der Benutzer – Task.

save (ALL myself, father)

***public***

TASK PROC public

Liefert den internen Task – Bezeichner von "PUBLIC".

fetch ("dateiname", public)

***rename***

PROC rename (TEXT CONST altername,neuename)

Umbenennen einer Datei von 'altername' in 'neuename'.

rename("altes handbuch","neues handbuch")

**save**

PROC save (TEXT CONST dateiname)

Datei 'dateiname' wird an die unmittelbare Vater – Task übertragen.

```
save("neues handbuch")
```

Fehlerfälle:

```
"neues handbuch" gibt es nicht  
zu viele Dateien  
Passwort falsch
```

PROC save (TEXT CONST name, TASK CONST task)

Datei mit dem Namen 'name' in Task 'task' kopieren

```
save ("dateiname", /"global")
```

Fehlerfälle:

```
"dateiname" gibt es nicht  
zu viele Dateien  
Passwort falsch
```

PROC save (THESAURUS CONST thesaurus)

Kopiert die Dateien, die in 'thesaurus' enthalten sind, in die Vater – Task.

```
save (SOME myself)
```

PROC save (THESAURUS CONST thesaurus, TASK CONST manager)

Kopiert die Dateien, die in 'thesaurus' enthalten sind, in Task 'manager'.

```
save(SOME myself, /"global")
```

**SOME**

THESAURUS OP SOME (THESAURUS CONST thesaurus)

Bietet den angegebenen 'thesaurus' zum Editieren an. Dabei können nicht erwünschte Namen gestrichen werden.

THESAURUS OP SOME (TASK CONST task)

Bietet einen THESAURUS von 'task' zum Editieren an.

THESAURUS OP SOME (TEXT CONST dateiname)

Bietet einen 'thesaurus', der aus 'dateiname' gebildet wird, zum Editieren an.

**task**

TASK PROC task (TEXT CONST task name)

Liefert den internen Task – Bezeichner von 'task name'.

```
save ("dateiname", task ("PUBLIC"))
      =      save ("dateiname", public)
```

**storage info**

PROC storage info

Informationsprozedur über den belegten Hintergrund – Speicher.

**task info**

PROC task info

Informiert über alle Tasknamen im System unter gleichzeitiger Angabe der Vater/Sohn – Beziehungen (Angabe durch Einrückungen).

PROC task info (INT CONST art)

Informiert über alle Tasks im System. Mit 'art' kann man die Art der Zusatz – Information auswählen. Für 'art' sind zur Zeit folgende Werte zugelassen:

art=1: entspricht 'task info' ohne Parameter, d.h. es gibt nur die Tasknamen unter Angabe der Vater/Sohn – Beziehungen aus.

art=2: gibt die Tasknamen aus. Zusätzlich erhalten Sie Informationen über die verbrauchte CPU – Zeit der Task, die Priorität, den Kanal, an dem die Task angekoppelt ist, und den eigentlichen Taskstatus. Hierbei bedeuten:

0	– busy –	Task ist aktiv.
1	i/o	Task wartet auf Beendigung des Outputs oder auf Eingabe.
2	wait	Task wartet auf Sendung von einer anderen Task.
4	busy – blocked	Task ist rechenwillig, aber blockiert.
5	i/o – blocked	Task wartet auf I/O, ist aber blockiert.
6	wait – blocked	Task wartet auf Sendung, ist aber blockiert. Achtung: Die Task wird beim Eintreffen einer Sendung automatisch entblockiert.

art=3: wie 2, aber zusätzlich wird der belegte Speicher angezeigt. (Achtung: Prozedur ist zeitaufwendig!).

```
task info(2)
```

**task status**

PROC task status

Informationsprozedur über den Zustand der eigenen Task. Informiert u.a. über

- Name der Task, Datum und Uhrzeit;
- verbrauchte CPU – Zeit;
- belegten Speicherplatz;
- Kanal, an den die Task angekoppelt ist;
- Zustand der Task (rechnend u.a.m.);
- Priorität.

PROC task status (TASK CONST t)

Wie obige Prozedur, aber über die Task mit dem internen Tasknamen 't'.

task status (father)

**task password**

PROC task password (TEXT CONST geheim)

Einstellen eines Passworts für Benutzertask. Das Kommando 'task password' ist ein Monitor – Kommando. Ist eine Task mit einem Paßwort geschützt, so wird durch den Supervisor nach dem 'continue' – Kommando das Passwort angefragt. Nur nach Eingabe des richtigen Passworts gelangt man in die gewünschte Task. Das Passwort kann durch nochmaligen Aufruf von 'task password' geändert werden, z.B. wenn es in regelmäßigen Abständen geändert werden muß, um personenbezogene Daten zu schützen.

Es gibt keine Möglichkeit, ein einmal eingestelltes Passwort in Erfahrung zu bringen. Sollte das Passwort vergessen werden, kann somit die Task nur noch gelöscht werden.

Wird als Passwort ein '-' – Zeichen eingegeben, so wird verhindert, daß die betreffende Task jemals wieder mit dem 'continue' – Kommando angekoppelt werden kann. Dies ist z.B. für Manager – Tasks sinnvoll.

task password("mein geheimnis")

+

THESAURUS OP + (THESAURUS CONST links, rechts)  
Vereinigungsmenge von 'links' und 'rechts'.

THESAURUS OP + (THESAURUS VAR thes, TEXT CONST name)  
Nimmt den TEXT 'name' in den Thesaurus 'thes' auf.

save (SOME father + "rechnung", archive)

-

THESAURUS OP - (THESAURUS CONST links, rechts)  
Differenzmenge von 'links' und 'rechts'.

THESAURUS OP - (THESAURUS VAR thes, TEXT CONST name)  
Liefert einen Thesaurus aus 'thes', aber ohne den Eintrag 'name'.

save (ALL myself - "rechnung", archive)

/

THESAURUS OP / (THESAURUS CONST links, rechts)  
Zweck: Schnittmenge von 'links' und 'rechts'.

save(ALL myself / ALL father, archive)

TASK OP / (TEXT CONST task name)

Liefert aus einem Tasknamen den internen Tasknamen. '/' kann überall dort eingesetzt werden, wo ein interner Taskname verlangt wird.

fetch ("dateiname", /"global")

## TEIL 4: Der Editor

### 4.0. Vorwort

Mit dem EUMEL-Editor schreiben Sie alle Ihre Texte und Daten. Er bietet vielfältige Möglichkeiten, um Autoren oder Programmierer bei dem Erstellen, Korrigieren und Gestalten von Manuskripten oder Programmen zu unterstützen. Die größte Hilfe beim Schreiben besteht (durch die Speicherfähigkeit von Computern) im beliebig häufigen Zugriff auf einmal geschriebene Informationen. Im Gegensatz zu einer Schreibmaschine können Sie mit dem EUMEL-Editor (beliebig oft) Einfügungen vornehmen, Texte korrigieren, löschen und neu gestalten.

Somit ist das Schreiben von Texten mittels des EUMEL-Systems besonders dann vorteilhaft und zeitsparend, wenn Texte häufig geändert werden oder wenn sie in einer besonders schönen Form gedruckt werden sollen. Weiterhin bietet der Editor Hilfen zum Schreiben an, wie z.B. automatischen Wortumbruch am Zeilenende, eine Einrückungsautomatik, "Lernen" von Texten u.a.m. Zusätzlich kann der Editor in seinen Fähigkeiten erweitert und somit für spezielle Schreibarbeiten angepaßt werden. Aber das soll in einem späteren Kapitel beschrieben werden.

Bei der Entwicklung des Editors wurde besonderer Wert auf einfache Bedienung gelegt: innerhalb von wenigen Minuten können Sie schon Texte schreiben und Daten erfassen und sehen stets auf dem Bildschirm, was mit Ihrem Text passiert. Das Schreiben und Korrigieren werden durch einige wenige, aber leistungsstarke Funktionstasten unterstützt.

Einige Gestaltungsmöglichkeiten für Texte kann man nicht direkt auf dem Terminal "sehen", wie z.B. Proportionalschriften, Fettdruck usw. Solche Leistungen können durch Anweisungen an die Textkosmetik-Programme und den EUMEL-Drucker angefordert werden. Diese Anweisungen müssen in den Text eingefügt werden. Lesen Sie hierzu Teil 5 ("Textkosmetik").

## 4.1. Ein – und Ausschalten des Editors

*Hier beschreiben wir, wie der Editor ein – und ausgeschaltet wird und wie der Editor eine Datei einrichtet.*

Wenn in Ihrer Task auf dem Bildschirm die Aufforderung

```
gib kommando :
```

erscheint, tippen Sie

```
edit ("dateiname")
```

und der EUMEL – Editor wird eingeschaltet. Ist die Datei noch nicht vorhanden, d.h. kein Text unter dem angegebenen Namen im System gespeichert, folgt eine Anfrage, ob eine Datei unter dem eingegebenen Namen neu eingerichtet werden soll:

```
"dateiname" neu einrichten (j/n) ?
```

Dies dient zur Kontrolle von Schreibfehlern, die besonders bei ähnlichen Dateinamen auftreten. Man kann dann das Einrichten der Datei ablehnen, den Dateinamen verbessern und das Kommando erneut geben.

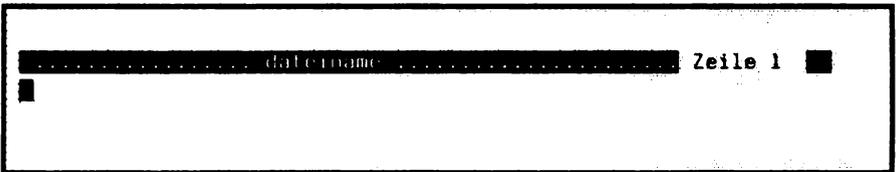
## Teil 4: Der Editor

---

Falls Sie die Datei neu anlegen wollen, bejahen Sie diese Frage mit

j  J  y oder  Y

Es erscheint ein leerer Editorbildschirm. Die oberste Zeile des Bildschirms ist die Titelzeile. In ihr kann nicht geschrieben werden. Sie zeigt jedoch verschiedene nützliche Dinge an: den Namen der Datei, die Nummer der aktuellen Zeile, in der gerade geschrieben wird, Tabulatormarken, Einfügemodus, Lernmodus, Auftrennung usw.



In unserem Fall haben Sie eine neue Datei angelegt. Sie enthält noch keinen Text. In der Titelzeile sind jedoch schon der Name der Datei und die aktuelle Zeilennummer eingetragen. Bei einer neuen Datei ist der Bildschirm unterhalb der Titelzeile leer. Dieser Teil dient als "Schreibfläche". Der Cursor steht dann direkt unter der Titelzeile. Er zeigt immer die aktuelle Schreibposition an. Jetzt kann sofort mit dem Schreiben begonnen werden, ganz wie mit einer normalen Schreibmaschine.

Rufen Sie eine Datei auf, in die Sie schon Text geschrieben haben, zeigt Ihnen der Editor das zuletzt bearbeitete Textstück und Sie können normal weiter schreiben.

Wollen Sie die Schreibarbeit beenden und den Editor ausschalten, so drücken Sie die beiden Tasten

**ESC** **q**

nacheinander. Es erscheint

gib kommando:

und Sie haben damit den Editor verlassen und befinden sich wieder im Monitor.

## 4.2. Die wichtigsten Tasten des Editors

### 4.2.1. Das Tastenfeld

*Auf dem Tastenfeld gibt es einige Tasten, die auf einer Schreibmaschine nicht vorhanden sind.*

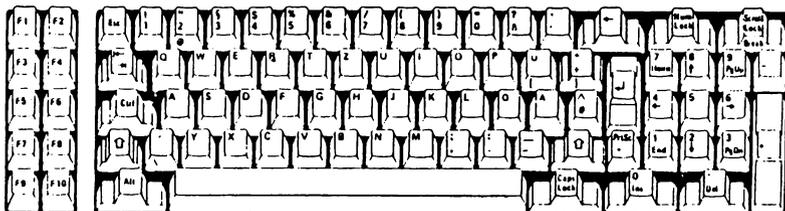
Das Tastenfeld eines EUMEL-Terminals entspricht weitgehend dem einer Schreibmaschine. Sie finden also die Buchstaben a-z und die Ziffern 0-9 auf Tasten. Mit der SHIFT-Taste (Umschalttaste) und gleichzeitigem Drücken einer anderen Taste können Sie die großen Buchstaben und eine Reihe von speziellen anderen Zeichen, die Sonderzeichen genannt werden, schreiben. Die "Zwischenraumtaste" oder Leertaste erzeugt immer ein Leerzeichen.

Nun gibt es in der Praxis zwei unterschiedliche Tastaturen. Zum einen existiert die EDV-Tastatur, die zum Schreiben von Programmen benutzt wird. Sie erkennt man daran, daß keine Umlaute (ä, ö, ü) und kein ß auf den Tasten abgebildet sind. Dafür gibt es Tasten für eckige und geschweifte Klammern. Sollen auf einer solchen Tastatur die Umlaute geschrieben werden, muß man sich eines Tricks bedienen: mit der Taste ESC und nachfolgendem Betätigen einer anderen Taste (z.B. a, o, u) erhalten wir den entsprechenden Umlaut.

In der Regel kann man die Umlaute auf dem Bildschirm eines solchen EDV-Terminals nicht sehen, sondern sie erscheinen als "a", "u", usw. Beim Druck eines Textes werden sie aber richtig dargestellt.

Die andere Tastatur entspricht in der Tastenbelegung weitgehend einer deutschen Schreibmaschine und besitzt Tasten für die Umlaute und ß. Sollen vorwiegend deutsche Texte geschrieben werden, empfiehlt es sich, solch ein Terminal zu verwenden.

## Tastatur



Neben diesen "einfachen" Tasten gibt es die Funktionstasten, die zur Bedienung des Editors (aber auch anderer Programme) notwendig sind. Wo die Tasten auf Ihrem Gerät liegen, hängt von dem jeweiligen Gerätetyp ab. Die Wirkung der Tasten erklären wir in den anschließenden Abschnitten. 1)

1) Es kann sein, daß die Tasten nicht richtig beschriftet sind. Die Installationsanleitung muß dann die Entsprechungen beschreiben. Zusätzlich zu den im folgenden beschriebenen können sich noch weitere Tasten auf Ihrem Terminal befinden, die aber standardmäßig keine besondere Bedeutung für den Editor haben.

## Die Funktionstasten des EUMEL – Systems



Positionierungstasten

**SHIFT**

Umschalttaste

**CR**

Eingabe – / Absatztaste

**HOP**

Verstärkertaste

**RUBOUT**

Löschtaste

**RUBIN**

Einfügetaste

**TAB**

Tabulatortaste

**MARK**

Markiertaste

**ESC**

Kommandotaste

**SV**

Supervisortaste

**STOP**

Stoptaste

**WEITER**

Weitertaste

## Die Wirkung der Funktionstasten

### **SHIFT**

#### Umschalttaste

Wird diese Taste gleichzeitig mit einer anderen betätigt, so wird ein Buchstabe in Großschreibung, bei den übrigen Tasten das obere Zeichen, ausgegeben. So wird z.B. anstelle der "9" das Zeichen ")" ausgegeben.

### **CTRL**

#### Kontroll – /Steuertaste

Mit dieser Taste in Kombination mit Zusatz Tasten können Sonderfunktionen des Systems angewählt werden. Für EUMEL sind folgende drei Tastenkombinationen (wobei die Tasten gleichzeitig betätigt werden müssen) wichtig:

- CTRL a** Anhalten der Bildschirmausgabe
- CTRL b** Wirkung der SV – Taste (bei jedem Rechner)
- CTRL c** Bildschirmausgabe fortführen

**CR**

Eingabetaste / Absatztaste, Carriage Return, kurz: 'CR'

Diese Taste wird im Editor betätigt, um das Ende eines Absatzes zu kennzeichnen. Die kontinuierliche Fließtexteingabe wird durch sie unterbrochen und es wird an den Beginn der nächsten Zeile positioniert. Einrückungen werden beibehalten. Eine Absatzmarke ist im Editor an der Inversmarkierung am rechten Bildschirmrand zu erkennen.

Die 'CR' –Taste ist oft mit einem geknicktem Pfeil nach links gekennzeichnet. Im Kommandomodus (also bei "gib kommando :") wird durch Betätigung dieser Taste ein gegebenes Kommando ausgeführt.

Die sonstige Benutzung dieser Taste außerhalb des Editors wird in der jeweiligen Anwendung beschrieben, z.B. Bestätigung eines Trennvorschlags bei der Silbentrennung.



Tasten für die Positionierung

Positionierung des Cursors um eine Spalten –/Zeilenposition in die jeweilige Richtung.

**HOP**

"Verstärkertaste"; wird als Vorschalttaste bedient.

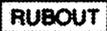
In Kombination mit anderen Funktionstasten wird deren Wirkung verstärkt. (vgl. 4 – 27)

Beispiel:



Steht der Cursor nicht am unteren Bildrand, so wird er dorthin positioniert. Steht er am unteren Bildrand, so wird um einen Bildschirminhalt "weitergeblättert".

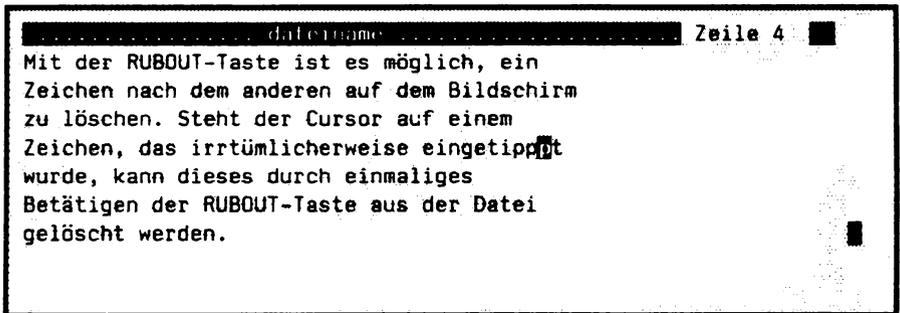
Auch die Funktionen 'RUBIN'/'RUBOUT' werden in Kombination mit der HOP –Taste verstärkt.(vgl. 4 – 28ff)



Löschtaste

Das Zeichen, auf dem der Cursor steht, wird gelöscht. Wenn der Cursor hinter dem letzten Zeichen einer Zeile steht, wie bei fortlaufender Eingabe üblich, wird das letzte Zeichen gelöscht.

Beispiel:



Nach Betätigen der **RUBOUT** - Taste:

```
..... dateiname ..... Zeile 4 ■  
Mit der RUBOUT-Taste ist es möglich, ein  
Zeichen nach dem anderen auf dem Bildschirm  
zu löschen. Steht der Cursor auf einem  
Zeichen, das irrtümlicherweise eingetippt  
wurde, kann dieses durch einmaliges  
Betätigen der RUBOUT-Taste aus der Datei  
gelöscht werden. ■
```

**RUBIN**

Ein – bzw. Ausschalten des Einfügemodus.

Das Betätigen der Taste schaltet in den Einfügemodus. Der Zustand wird durch das Wort "RUBIN" im linken Drittel der Titelzeile der Datei angezeigt. Vor dem Zeichen, auf dem der Cursor steht, wird eingefügt. Nochmaliges Betätigen der Taste schaltet den Einfügemodus aus.

Beispiel:

```
..... dateiname ..... Zeile 5 █
Das Betätigen der Taste schaltet in den
Einfügemodus. Der Zustand wird durch das
Wort "RUBIN" im linken Drittel der Titelzeile
angezeigt. Vor dem Zeichen, auf dem der
Cursor steht, wird eingefügt.
Nochmaliges Betätigen der Taste schaltet den
Einfügemodus aus.
```

Nach Betätigen der **RUBIN** – Taste und Einfügen des Wortes " jetzt":

```
..... RUBIN ..... dateiname .....
Zeile 5 █
Das Betätigen der Taste schaltet in den
Einfügemodus. Der Zustand wird durch das
Wort "RUBIN" im linken Drittel der Titelzeile
angezeigt. Vor dem Zeichen, auf dem der
Cursor steht, wird jetzt eingefügt.
Nochmaliges Betätigen der Taste schaltet den
Einfügemodus aus.
```

**TAB**

Tabulatortaste

Betätigen Sie die 'TAB' – Taste, um vom linken Bildschirmrand auf den Textbeginn in der Zeile bzw. eine Tabellenspalte zu positionieren. Erneutes Betätigen der 'TAB' – Taste positioniert den Cursor auf die nächste eingestellte Tabulator – Position. Die eingestellten Tabulatorpositionen erkennen Sie an den Tabulatorzeichen (Dachzeichen) in der obersten Bildschirmzeile.

Wenn keine TABs gesetzt sind, werden die beiden Schreibgrenzen, linker Bildschirmrand und Ende der Zeile, als *voreingestellte* TABs angesehen.

**MARK**

Ein – bzw. Ausschalten der Markierung.

Bei Betätigung dieser Taste wird in einen speziellen Markierzustand geschaltet. Alles, was Sie jetzt schreiben bzw. durch Bewegen des Cursors in Richtung Dateiende kennzeichnen, steht als *markierter* Bereich für die Bearbeitung zur Verfügung. Zur besseren Sichtbarkeit wird der markierte Bereich invers zum übrigen Text dargestellt.

Wird der Cursor in eine Richtung bewegt, wird das gesamte Textstück zwischen Einschaltzeitpunkt der Markierung und aktueller Cursorposition markiert. Rückwärtsbewegungen des Cursors verkürzen den markierten Bereich wieder.

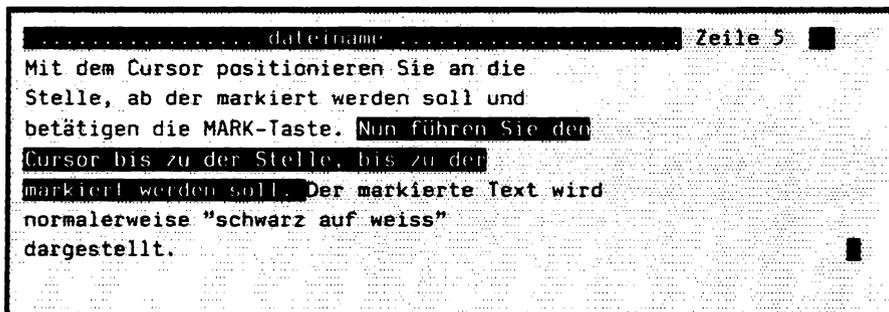
Einen derart markierten Bereich können Sie nun z.B. duplizieren, verschieben, löschen, durchsuchen oder weiterverarbeiten. (vgl. 4 – ff).

Durch erneutes Betätigen der MARK – Taste schalten Sie den Markier – Zustand auch wieder aus.

Beispiel:

Sie wollen einen Textteil markieren, um ihn an eine andere Stelle zu verschieben (evtl. um ihn an dieser Stelle später zu löschen):

Sie positionieren den Cursor auf den Beginn des Textteils, gehen in den Markierzustand durch Betätigen der MARK-Taste und führen nun den Cursor mit Hilfe der Positioniertasten bis zum Ende des zu markierenden Bereichs.



Mit weiteren Kommandos (vgl. ESC-Taste und Kommando-Verarbeitung, 4 – 35) kann der Bereich nun bearbeitet werden.

**ESC**

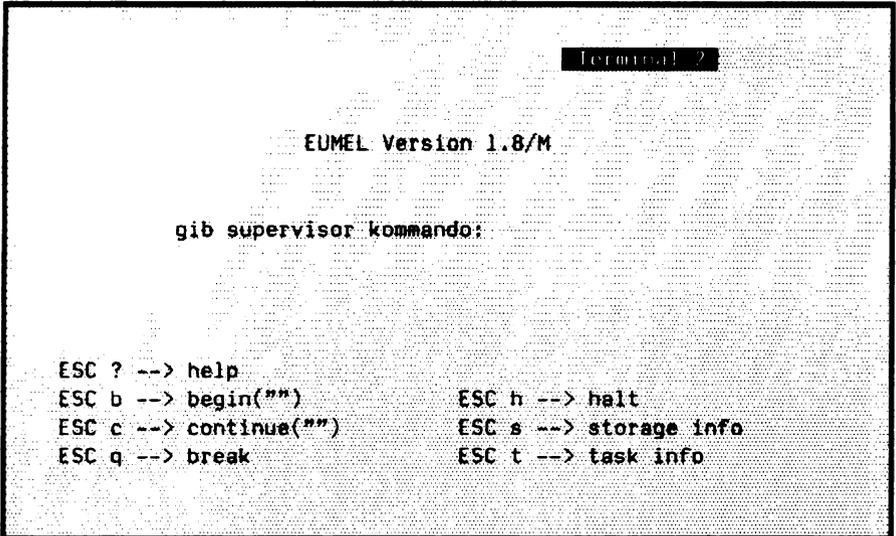
Kommandotaste

Mit der ESC-Taste in Kombination mit einer Folgetaste können Sie vordefinierte Aktionen anwählen. Es gibt Aktionen, die vorprogrammiert zur Verfügung stehen, und Sie selbst können weitere hinzufügen. (vgl. 4 – 35)

SV

SUPERVISOR – Taste im Mehrbenutzer – Betrieb

Betätigen Sie diese Taste im Editor, dann unterbrechen Sie Ihre Editierarbeit und erhalten die Meldung



```
Terminal 2
EUMEL Version 1.8/M
gib supervisor kommando:
ESC ? --> help
ESC b --> begin("")
ESC c --> continue("")
ESC q --> break
ESC h --> halt
ESC s --> storage info
ESC t --> task info
```

Wollen Sie nun im Editor fortfahren bzw. haben Sie irrtümlich die SV – Taste betätigt, dann geben Sie das Kommando

```
gib supervisor kommando :  
continue ("Sekretariat")
```

(falls Ihre Task, in der Sie arbeiteten, wirklich "Sekretariat" hieß!)

Um Ihren in Bearbeitung befindlichen Text wieder vollständig auf dem Bildschirm zu sehen, betätigen die die Tasten

**ESC** **b**

Sie sind wieder an der Stelle, an der Sie den Text mit der SV – Taste verlassen haben, und können normal weiterarbeiten.

Achtung: Die SV – Taste kann, je nach Terminal, durch das Betätigen von zwei Tasten gleichzeitig realisiert sein (oft 'CTRL b'). Beachten Sie die Beschreibung Ihrer Tastatur!

**STOP**

Unterbrechen einer Ausgabe (oft auch als CTRL a realisiert).

Haben Sie diese Taste aus Versehen betätigt, erkennen Sie dies daran, daß der Editor nicht "reagiert". Betätigen Sie die WEITER – Taste (oft auch CTRL c).

**WEITER**

Unterbrochene Ausgabe fortsetzen.

Ein mit der STOP – Taste angehaltene Ausgabe können Sie durch Betätigen der WEITER – Taste fortsetzen.

VORSICHT:Die STOP – Taste unterbricht nur die Ausgabe auf den Bildschirm. Zeichen, die während des STOP eingegeben werden, werden gespeichert und nach 'WEITER' ausgegeben!

## 4.2.2 Speicherung von Texten

*In diesem Abschnitt wird der Begriff "Datei" erklärt und es wird erläutert, wie unterschiedliche Texte auseinandergehalten werden können.*

Das EUMEL-System speichert einmal geschriebene Texte, bis sie vom Benutzer gelöscht werden. In der Regel wird nicht nur ein (langer) Text oder ein Programm geschrieben, sondern mehrere und unterschiedliche. Um diese auseinanderhalten zu können, versehen wir sie jeweils mit einem Namen, der frei gewählt werden kann. Beispiele für Namen:

"Brief vom 1.12.86"

"1. Kapitel meines Buches"

Eine Sammlung von Zeichen (also im Normalfall unsere geschriebenen Texte), die mit einem Namen versehen worden ist, nennt man eine **Datei**. Der Editor erstellt also eine Datei, wenn wir einen Text schreiben. Eine Datei kann bis zu 4 000 Zeilen fassen, wobei jede Zeile bis zu 32 000 Zeichen lang sein darf. Das Produkt aus der Anzahl der Zeilen und den Zeichen pro Zeile kann z.Zt. jedoch 1 000 000 Zeichen (= 1MB) nicht übersteigen.

### 4.2.3. Schreiben von Texten

*Texte werden fortlaufend geschrieben. Absätze werden durch die CR-Taste markiert.*

Nach dieser etwas langen Vorrede können wir endlich losschreiben. Wird ein Zeichen geschrieben, rückt der Cursor automatisch nach rechts auf die nächste Schreibstelle. Durch den automatischen Wortumbruch werden angefangene Worte, die über ein Zeilenende hinausgehen würden, ohne Silbentrennung in die nächste Zeile gebracht.<sup>1)</sup>

Die 'CR' – Taste (bei einer Schreibmaschine bedeutet sie "Wagenrücklauf") braucht also nur noch betätigt zu werden, wenn eine Zeile vorzeitig beendet werden soll, d.h. bei einem Absatz oder einer Leerzeile. Der Cursor wird dabei an den Anfang der nächsten Zeile positioniert. Gleichzeitig erscheint in der vorherigen Zeile am rechten Rand des Bildschirms eine Markierung, die anzeigt, daß hier ein Absatz gemacht wurde.

Darum ist das Betätigen der 'CR' – Taste bei Tabellenzeilen und Programmtexten besonders wichtig, denn hier soll ja jede Zeile separat bleiben. Sie wirkt nur hinter dem letzten Zeichen.

Der Editor ist auf das Schreiben von "normalen" Texten eingestellt. Bei normalen Texten soll ein Wort, welches über das Ende einer Zeile gehen würde, automatisch in die nächste Zeile gebracht werden. Diese Funktion wird "Wortumbruch" genannt.

---

<sup>1)</sup> Nehmen Sie bitte keine Silbentrennung "per Hand" vor. Eingebrachte Trennstriche gelten als Bindestrich und bleiben somit auch bei Umformatierungen erhalten, was unerwünscht ist. Für diese mühevollen Aufgabe gibt es in der Textverarbeitung ein Programm!

Ist kein Wortumbruch erwünscht, zum Beispiel bei der Beschreibung von Programmen, so geben Sie, bevor Sie den Editor aufrufen, im Monitor das Kommando

```
gib kommando :  
word wrap (false)
```

Der Wortumbruch kann durch das Kommando

```
gib kommando :  
word wrap (true)
```

wieder eingeschaltet werden. Der Editor ist standardmäßig auf "Wortumbruch" eingestellt und Sie sollten nur in Ausnahmefällen diese Benutzungsart ausschalten.

Ein Bildschirm faßt (neben der Titelzeile) üblicherweise 23 Zeilen, die mit Text beschrieben werden können. Ist die letzte Zeile voll und muß eine neue Zeile begonnen werden, "rutscht" der Bildschirminhalt automatisch um eine Zeile nach oben. Damit ist Platz für eine Leerzeile, die nun ebenfalls beschrieben werden kann, usw. Keine Angst: die so verschwundenen Zeilen sind natürlich nicht "weg". Da ein Bildschirm immer nur eine beschränkte Anzahl von Zeilen hat, kann der Editor nur einen Ausschnitt aus der Datei zeigen.

## Eintrückungen

*Die Einrückungsautomatik erlaubt bei fortlaufendem Schreiben, die Einrückung zu erhalten.*

Soll ein Text eingerückt werden, so betätigt man entsprechend oft die Leertaste. Die in dieser Zeile geschriebene Einrückung wird automatisch in den folgenden Zeilen beibehalten, bis sie durch die Cursor – Positionierungstasten wieder aufgehoben wird.

Beispiele für Aufzählungen: Einrückung funktioniert automatisch ohne aktive Eingabe von Leerschritten.

The screenshot shows a text editor window with a title bar containing "dateiname" and "Zeile 1". The text inside the editor is as follows:

- Der erste Typ der Aufzählungsform ist die Aufzählung durch Vorstellen eines Sondersymbols. Als Sondersymbole sind die beiden Zeichen "-" und "\*" zugelassen. An ihnen erkennt der Editor eine Aufzählung.
- 12. Weiterhin können Aufzählungen durch Begriffe, gefolgt von einem Punkt oder einer ")", als Aufzählungskriterium verwendet werden.

Aufzählung: Auch diese Möglichkeit steht Ihnen zur Verfügung. Der Editor erkennt, daß Sie hier einen Begriff erläutern wollen.

Wann werden nun Aufzählungen vom Editor erkannt?

Die hier aufgeführten Einzelheiten sollte nur der hieran interessierte Anfänger lesen!

Wenn die Einrückung nicht funktionieren sollte, prüfen Sie die folgenden Punkte, die für das Einrücken erfüllt sein müssen:

1) Die Vorgängerzeile hat eine Absatzmarke.

Wichtig: Innerhalb eines Aufzählungspunktes schaltet die Absatztaste die Aufzählungseinrückung aus!

- 2) "\*" bzw. "-" und mindestes ein Leerzeichen sind die ersten Zeichen in der Zeile.
- 3) "." bzw. ")" und mindestens ein Leerzeichen nach höchstens sieben Zeichen sind die ersten Zeichen in der Zeile.
- 4) ":" und mindestens ein Leerzeichen nach höchstens 19 Zeichen sind die ersten Zeichen in der Zeile.

## 4.2.4. Positionieren im Text

*Um Korrekturen (Überschreiben, Löschen oder Einfügen) vorzunehmen, muß der Cursor, der die aktuelle Schreibposition anzeigt, bewegt werden können. Bei längeren Texten ist es möglich, den Cursor auch auf Zeilen zu positionieren, die (noch nicht) auf dem Bildschirm angezeigt werden. Somit zeigt der Editor nicht nur immer das Ende einer Datei, sondern einen beliebigen Ausschnitt, der auf dem Bildschirm im sogenannten 'Fenster' sichtbar ist.*

Ist eine Korrektur notwendig, positionieren Sie den Cursor auf die Stelle, an der die Korrektur vorgenommen werden soll. Dazu verwenden Sie die Positionierungstasten LINKS, RECHTS, OBEN und UNTEN. LINKS und RECHTS bewegen den Cursor innerhalb einer Zeile. Stößt man mit RECHTS an das Ende einer Zeile, wird der Cursor an den Anfang der nachfolgenden Zeile bewegt.



Ein Zeilenwechsel kann einfacher mit den Tasten OBEN und UNTEN vorgenommen werden. Die Taste OBEN bewegt den Cursor eine Zeile nach oben, die Taste UNTEN entsprechend eine Zeile tiefer.

Was passiert nun, wenn Sie den unteren oder den oberen Rand des Bildschirms erreicht haben, und Sie positionieren darüber hinaus? In diesem Fall wird der Text zeilenweise nach oben oder nach unten verschoben und es erscheint die gewünschte

Zeile, wobei am anderen Rand einige verschwinden". Wir sehen also, daß wir mit den Positionierungstasten den Bildschirm als Fenster über die Datei hinweggleiten lassen können. Den Text selbst können wir uns auf einem langen Band geschrieben vorstellen. Die Zeilennummer, die die Position des Cursors angibt, wird stets in der Titelseile angezeigt.

Vermeiden Sie es, den Cursor über das Textende hinaus nach unten laufen zu lassen. Sie verlängern dadurch Ihren Text um Leerzeilen, die Sie beim Weiterschreiben nicht auffüllen, sondern vor sich herschieben.

Innerhalb einer Zeile ist es etwas anders: Positionieren wir bei einer Zeile, die breiter als der Bildschirm ist, nach rechts, wird nicht das Fenster verschoben, sondern die Zeile 'gerollt'.(vgl. Sie hierzu das Verschieben des Gesamtfensters mit dem 'margin' – Kommando 4 – 59)

## 4.2.5. Korrigieren im Text

*Einfache Korrekturen können durch Überschreiben von Zeichen, Löschen von Zeichen und Einfügen von Zeichen vorgenommen werden.*

### **RUBOUT**

Die einfachste Möglichkeit der Korrektur ist das Überschreiben. Soll z.B. ein Zeichen durch ein anderes ersetzt werden, so positioniert man der Cursor genau über dieses und tippt das richtige Zeichen ein. Das kann natürlich auch mit mehreren Zeichen nacheinander erfolgen.

Korrekturen können Sie gleich beim Schreiben vornehmen, indem Sie die zuletzt geschriebenen Zeichen mit der RUBOUT-Taste löschen. Häufig bemerkt man aber Schreibfehler erst etwas später, so daß man diese Fehler nicht so leicht korrigieren kann. Für solche Zwecke müssen Sie den Cursor an die Textstelle bewegen, an der korrigiert werden soll.

Wollen Sie ein Zeichen löschen, so positionieren Sie den Cursor auf dieses Zeichen und betätigen die Taste RUBOUT. Das Zeichen verschwindet und die Restzeile rückt heran. Sollen mehrere Zeichen gelöscht werden, muß die RUBOUT-Taste entsprechend oft gedrückt werden.

Steht der Cursor hinter dem letzten Zeichen der Zeile, wird immer das letzte Zeichen der Zeile gelöscht. Man kann also mit dieser Eigenschaft eine Zeile "von hinten wegradieren".(vgl. hierzu auch 4 - 37)

**RUBIN**

Fehlende Zeichen können Sie genauso einfach einfügen. Sie bringen den Cursor auf das Zeichen, vor das eingefügt werden soll. Dann drücken Sie die Taste RUBIN. Der Editor gelangt in den Einfügemodus, was in der Titelzeile durch RUBIN angezeigt wird. Er fügt alle Zeichen ein, die jetzt getippt werden (anstatt zu überschreiben). Der Teil der Zeile rechts vom Cursor rückt jeweils um entsprechend viele Stellen nach rechts.

Wichtig ist, daß im RUBIN – Modus der Editor genauso funktioniert wie im Normalzustand (natürlich mit der Ausnahme, daß eingefügt statt überschrieben wird).

Im eingeschalteten RUBIN – Modus können keine Zeichen verloren gehen. Viele Benutzer lassen darum den RUBIN – Modus immer eingeschaltet, um sich vor einem unbeabsichtigten Überschreiben von Texten zu schützen. Sie korrigieren, indem Sie die Verbesserung einfügen und den alten Text löschen.

Durch erneutes Betätigen der RUBIN – Taste beenden Sie den Einfügemodus. Die RUBIN – Taste wirkt wie ein Schalter, der den Einfügemodus ein – und ausschaltet. Allerdings können Sie nur so viele Zeichen in eine Zeile einfügen, bis das letzte Wort der Zeile an das Zeilenende stößt. Das letzte Wort wird am Anfang der folgenden Zeile eingefügt, sofern dort noch Platz ist und es sich nicht offensichtlich um die letzte Zeile eines Absatzes handelt. Andernfalls wird automatisch eine neue Zeile für das angefangene Wort eingefügt.(vgl. Sie hierzu auch 4 – 38)

## Springen und Zeilen einfügen/löschen

*Bewegungen des Cursors sind mit den Positionierungstasten bei größeren "Abständen" etwas mühsam, ebenso bei umfangreichen Löschungen und Einfügungen. Die "Verstärkertaste" HOP ermöglicht es, diese Operationen auf einfache Weise zu beschleunigen. Mit der HOP-Taste kann man das Fenster über der Datei nicht nur zeilenweise, sondern auch um jeweils eine Fensterlänge verschieben. Das nennt man Blättern.*

Wird die HOP-Taste vor einer anderen der schon erklärten Funktionstasten gedrückt, verstärkt sie deren Wirkung. Die HOP-Taste ist eine "Präfix"-Taste: sie wird vor (und nicht gleichzeitig mit, wie z.B. die Umschalttaste SHIFT) einer anderen Taste gedrückt. Zuerst das springende Positionieren:



Zeichenweises Positionieren mit dem Cursor.



Sprung an das rechte Bildschirmende.

Falls die Zeile länger als das Fenster breit ist, wird die Zeile um eine Fensterbreite nach links verschoben.



Sprung an den Bildschirmrand links (ggf. seitlich blättern).



Sprung auf die erste Zeile des Bildschirms.

Nochmaliges Betätigen dieser Tastenkombination positioniert den Cursor (und damit das Fenster in der Datei) um ein Fenster zurück. ("Blättern")



Sprung auf die letzte Zeile des Bildschirms.

Das Blättern erfolgt analog HOP OBEN.

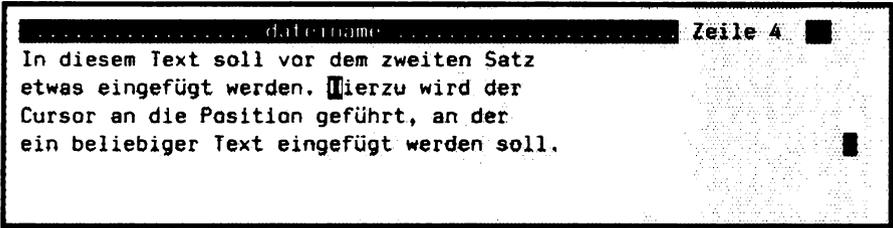


Positioniert das Fenster so, daß die aktuelle Zeile zur ersten des Fensters wird.



Einfügen von Textpassagen. Die HOP-Taste in Verbindung mit RUBIN und RUBOUT wird zum "verstärkten" Löschen und Einfügen verwendet.

Ab der aktuellen Position des Cursors "verschwindet" der restliche Text. Es kann wie bei der anfänglichen Texteingabe fortgefahren werden. Die Anzeige 'REST' in der Titelzeile erinnert daran, daß noch ein Resttext existiert. Dieser erscheint nach einem neuerlichen Betätigen der beiden Tasten HOP RUBIN wieder auf dem Bildschirm (die Anzeige 'REST' verschwindet dann wieder).



Nach Betätigen der Tasten **HOP** und **RUBIN** sieht der Bildschirm wie folgt aus:

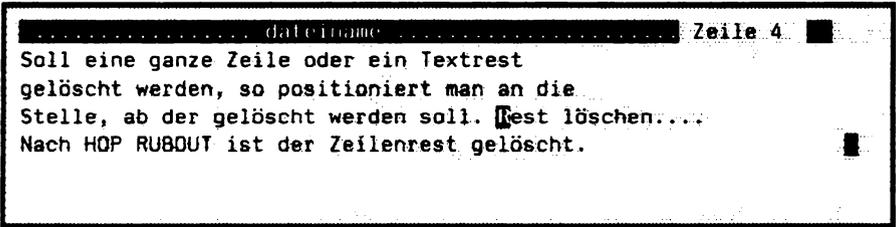


Nun kann beliebig viel Text eingefügt werden. Nochmaliges Betätigen von HOP und RUBIN führt den Text – Rest wieder bündig heran.

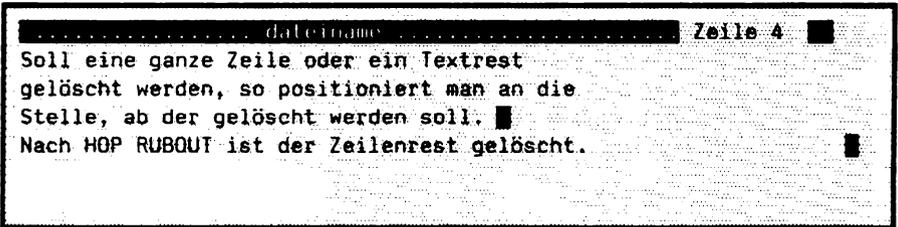
**HOP**

**RUBOUT**

Löscht die Zeile ab Cursor – Position bis Zeilenende.



Nach Betätigen der Tasten **HOP** und **RUBOUT** sieht der Bildschirm wie folgt aus:



Steht der Cursor am Zeilenanfang, wird nach HOP RUBOUT dementsprechend die ganze Zeile gelöscht und die Lücke durch Nachrücken der Folgezeilen geschlossen (HOP RUBOUT betätigen).

## Zeilen aufbrechen und Rückumbruch

*Um grössere Textpassagen einzufügen, betätigt man HOP RUBIN nacheinander. Diese Tastenfolge kann benutzt werden, um eine Zeile bzw. eine längere Textpassage aufzubrechen). HOP RUBOUT am Ende einer Zeile macht einen Rückumbruch.*

**HOP** **RUBIN**

Wie bereits beschrieben, bewirkt HOP RUBIN in einer Zeile, daß der Zeilenrest rechts des Cursors und alle Zeilen unterhalb der aktuellen Zeile scheinbar verschwinden. REST in der Titelzeile erinnert daran, daß ein Teil der Datei nicht sichtbar ist.

Wird unmittelbar nach HOP RUBIN wiederum HOP RUBIN betätigt, wird der vorherige Zeilenrest als eigenständige Zeile dargestellt. Es ist damit eine Aufspaltung einer Zeile in zwei Zeilen vollzogen.

**HOP** **RUBOUT**

Der umgekehrte Fall, nämlich zwei Zeilen zu einer zusammenzufassen (sog. Rückumbruch), ist durch HOP RUBOUT hinter dem letzten Zeichen einer Zeile möglich. Hinter das letzte Zeichen einer Zeile kann einfach mit dem Tabulator positioniert werden.

Das Aufbrechen einer Zeile und der Rückumbruch zusammen angewandt stellen den ursprünglichen Zustand wieder her. Beispiel: Mit HOP RUBIN bricht man eine Zeile auf, der Rest der Zeile und nachfolgende Zeilen verschwinden vom Bildschirm. Erneutes HOP RUBIN stellt den rechten Zeilenteil auf der nächsten Zeile und die nachfolgenden Zeilen auf dem Bildschirm wieder dar. Da der Cursor sich noch immer am rechten Rand der aufgebrochenen Zeile befindet, kann man mit HOP RUBOUT den ursprünglichen rechten Zeilenteil wieder rekombinieren.

## 4.2.6. Der Tabulator

*Eine weitere wichtige Positionierungshilfe innerhalb einer Zeile ist die TAB – Taste. Sie wird u.a. zum Schreiben von Tabellen benötigt. Wie bei einer Schreibmaschine können Tabulatormarken gesetzt bzw. gelöscht werden.*

**TAB**

Der Tabulator hat eine wichtige Funktion für das schnelle Positionieren, auch wenn keine Marken eingestellt wurden. Voreingestellte Tabulatormarken sind nämlich der Textanfang einer Zeile (Einrückung, falls vorhanden) und die Stelle direkt hinter dem letzten Zeichen der Zeile. Betätigt man also die Taste TAB, dann springt die Schreibmarke an die nächste dieser voreingestellten Positionen. So kann man schnell mit dem Cursor an den Anfang oder das Ende einer Zeile gelangen (und z.B. am Zeilenende Zeichen "von hinten" löschen oder dort weiterschreiben).

**HOP**

**TAB**

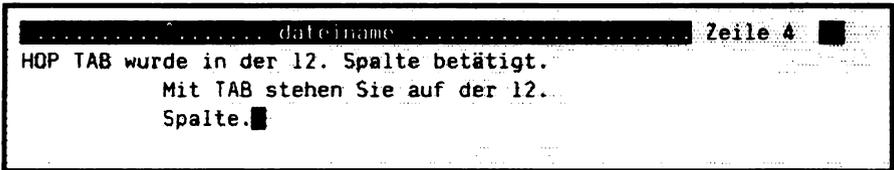
Nun zum Setzen des Tabulators: Sie setzen ihn, indem Sie den Cursor auf die Zeilenposition bringen, in der die Marke plaziert werden soll. Hier betätigen Sie nun HOP TAB. Die Tabulatorsetzung kann man in der Titelzeile an einer Markierung ("Dach" – Zeichen) sehen, falls sie im Fensterbereich ist und die aktuelle Zeile nicht seitlich verschoben ist. Betätigt man nun an irgendeiner Position innerhalb einer Zeile die TAB – Taste, wird der Cursor auf die Position der nächsten Tabulatormarkierung (die sich rechts von dem Cursor befindet) oder eine der voreingestellten Positionen bewegt.

Gesetzte Tabulatormarken können gelöscht werden, indem man mit der TAB – Taste die Position der Tabulatormarke einstellt und dann HOP TAB betätigt. Die Marke ist dann gelöscht, das Dach verschwindet in der Titelzeile.

Tabulatormarkierungen hinterlassen keine Spuren in der Datei, sondern dienen nur als Positionierungshilfen. mit 'HOP TAB' gesetzte Markierungen, die mit 'TAB' angesprungen werden, wirken beim Schreiben von Zahlen wie Dezimaltabulatoren, vgl. Sie dazu 4 – 34.

Beispiel:

Es soll für den Textbeginn eine Tabulatorposition auf die 12. Spalte gesetzt werden. Hierzu wird der Cursor auf die 12. Spalte positioniert und die HOP- und die TAB-Taste nacheinander betätigt. Das "Dach"-Zeichen erscheint in der 12. Spalte in der Titelzeile und von nun an kann durch Betätigen der TAB-Taste diese Position direkt angesteuert werden.



Werden Tabulatormarken gesetzt (HOP TAB), gelten die voreingestellten Tabulatormarken (Anfang und Ende einer Zeile) nicht mehr. Dies ist z.B. bei dem Schreiben von Tabellen notwendig. Andererseits möchte man beim Schreiben von "normalem" Text wieder die voreingestellten Tabulatormarken bedienen können. Mit den Tasten



kann man die gesetzten Tabulatormarken (erkennlich an dem "Dach"-Zeichen in der Kopfzeile) vorübergehend verschwinden lassen. Dann gelten wieder die voreingestellten Marken. Erneutes ESC TAB stellt die gesetzten Tabulatormarken wieder her usw..

## Zahlentabellen schreiben: Dezimaltabulator

*Beim Schreiben von Zahlentabellen sollen die Zahlen oft rechtsbündig im Text erscheinen. Dazu bietet der Editor den Dezimaltabulator an.*

Für jede Zahlenkolonne wird die gewünschte Position der Einerstelle (also der letzten Stelle) mit Hilfe eines Tabulators eingestellt. Mit TAB wird der Cursor zur jeweils nächsten Tabulatormarke vorgerückt. Werden nun Ziffern geschrieben, so schreibt man nicht – wie gewohnt – nach rechts, sondern die Ziffern werden nach links eingerückt. Etwas genauer: Beim Drücken einer Zifferntaste wird, solange links vor der Zahl noch ein Blank, eine Zahl, "+", "-" oder ein Punkt sichtbar ist, diese gelöscht und die hierdurch neu entstandene Ziffernfolge rechtsbündig an der Tabulatorposition geschrieben. Das Schreiben von rechtsbündigen Zahlenkolonnen ist so leicht möglich 1):

12	12345,78
1	0,23
12345	1234,00

Es gibt somit vier nützliche Automaten: neben dem automatischen Dezimaltabulator den Wortumbruch, die Einrückautomatik und die Zeileneinfügeautomatik beim einfügenden Schreiben.

---

1) Wird eine Proportionalschrift (Schrift, bei der die Zeichen unterschiedliche Breiten haben) verwendet, sollte man zwischen den einzelnen Zahlenkolonnen mindestens zwei Leerzeichen schreiben. Andernfalls bekommt man – auf Grund der unterschiedlicher Zeichenbreiten – keine rechtsbündigen Kolonnen gedruckt.

## 4.2.7. Lernen im Editor

*Beliebige Folgen von Tastenbetätigungen können gelernt und Tasten zugeordnet werden. Das ist sinnvoll, wenn Sie wiederholt immer die gleichen Tastenbetätigungen ausführen müssen, wie z.B. in Tabellenzeilen etwas einfügen oder wenn dies öfteren gleiche Texte geschrieben werden müssen, wie z.B. ein Absender, Grußformeln usw.*

ESC HOP

Der Lernmodus wird durch Betätigen der Tasten ESC HOP eingeschaltet, es erscheint LEARN als Kontrolle rechts in der Titelzeile). Alle Tastenanschläge werden jetzt bis zum Ausschalten des Lernmodus gelernt. Auch Tastenanschläge wie 'CR'), so daß man kann demnach auch mehrere Zeilen lernen lassen kann.

ESC HOP 'taste' z.B. ESC HOP j

Das Beenden oder Ausschalten des Lernmodus erfolgt durch Drücken der drei Tasten ESC HOP 'taste'. Dabei wird die gelernte Tastenanschlagsfolge, auch Lernsequenz genannt, der Taste 'taste' zugeordnet.

ESC 'taste' z.B. ESC j

Durch späteres Betätigen der Tastenfolge ESC 'taste' kann der gelernte Text an jeder Stelle der Datei geschrieben werden.

Beispiel:

Ein Sachbearbeiter hat jeden Tag 50 mal die Worte 'Gesellschaft für Datenverarbeitung' zu tippen. Er läßt den Editor diese Worte lernen mit

ESC HOP Gesellschaft für Datenverarbeitung ESC HOP m

Die Worte liegen jetzt auf der Taste 'm'. Wird 'm' gedrückt, erscheint ein 'm' auf dem Bildschirm. Mit ESC 'm' erscheinen die obigen Worte. ESC ist also notwendig, um das normale 'm' von der Lernsequenz zu unterscheiden.

Welche Tasten dürfen zum Lernen belegt werden? Alle Tasten, außer

- vom System benutzte Tasten, wie SV, CTRL;
- vom Editor (je nach Anwendung) vorgelegte Tasten, wie die Tasten q oder ESC und HOP;
- durch Programmierung (siehe dieses Kapitel) fest belegte Tasten.

**Praktische Tips:** Man sollte die Tastatur nicht mit Lernsequenzen überlasten, weil man sich zu viele Tasten nicht merken kann. Besser ist es, einige wenige Tasten fest zu belegen und andere für momentane Aufgaben einzusetzen.

Der Einsatz von Lernsequenzen ist besonders sinnvoll für das Schreiben von Textkosmetikanweisungen. Anweisungen wie z.B. 'Unterstreichen einschalten', Schrifttyp – Anweisungen usw. werden zweckmäßigerweise auf Tasten gelegt.

Hat man sich einmal beim 'Lernen' verschrieben, so ist das nicht weiter schlimm: es kann ohne Bedenken korrigiert werden (z.B. mit der Taste RUBOUT). Solche Tastenschläge werden dann allerdings auch gelernt, was aber bei der Benutzung der Lernsequenzen keine Bedeutung hat.

## 4.2.8. Textabschnitte durch Markieren bearbeiten

*Oft ergibt sich die Notwendigkeit, mehrere Zeilen oder ganze Textpassagen zu löschen oder zu verschieben. Hierbei hilft die Taste MARK, mit der man Texte markieren (also kennzeichnen) kann. Die so markierten Texte können dann auf verschiedene Weisen als Ganzes verarbeitet werden.*

### MARK

Durch Drücken der Taste MARK wird die Markierung eingeschaltet und – bei erneuter Betätigung – wieder ausgeschaltet. Der Anfang der Markierung wird "festgehalten" und man kann nun das Markierende durch die Positionierungstasten und die HOP-Taste in Richtung auf das Dateiende verschieben, wobei die dazwischen liegenden Zeichen markiert (in der Regel "schwarz auf weißem Grund" dargestellt) werden.

### ESC

### RUBOUT

Ein so markierter Text kann mit ESC RUBOUT gelöscht werden. Markieren und löschen mit ESC RUBOUT ist eine bequeme und sichere Löschmethode, da man genau sieht, was gelöscht wird.

**ESC****RUBIN**

Der gelöschte Abschnitt ist aber nicht vollständig gelöscht, sondern er kann an anderer (oder an der gleichen) Stelle im Text durch ESC RUBIN wieder eingefügt werden. Der vorsichtig gelöschte Text landet in einem Zwischenspeicher und kann bei Bedarf mit ESC RUBIN wieder aufgerufen werden. Wird erneut vorsichtig gelöscht, so wird der letzte Text des Zwischenspeichers überschrieben. Im Zwischenspeicher ist nur für einen Text Platz. Auf diese Art kann ein Textabschnitt beliebiger Länge an eine andere Stelle des Textes sicher, schnell und bequem verschoben werden. Zusätzlich ist die nachträgliche Korrektur von fehlerhaften Löschungen möglich, weil der Text wieder mit ESC RUBIN reproduziert werden kann.

Mit eingeschalteter Markierung kann auch geschrieben werden. Das markierende Schreiben ist eine besonders vorsichtige Art der Texterstellung, denn der Texteschub bleibt erst durch Ausschalten der Markierung (MARK) wirklich bestehen. Er kann wieder gelöscht (ESC RUBOUT) und an eine andere Stelle gebracht werden (ESC RUBIN). Beim markierenden Schreiben wirkt RUBOUT immer auf das Zeichen vor der Cursorposition.

Hinweis: Positionierungen sind nur innerhalb der Markierung möglich.

## 4.2.9. Der Fenstereditor

*Oft ist es notwendig, mit mehreren Dateien gleichzeitig zu arbeiten, z.B. wenn aus einer Datei etwas in eine andere kopiert werden muß, wenn Fehler durch die Textkosmetik – Programme oder einen Compiler gefunden werden oder wenn man kurz etwas in einer anderen Datei nachschauen will. Zu diesem Zweck bietet der Editor die Möglichkeit, zwei (oder mehr) Dateien zur gleichen Zeit zu bearbeiten.*

Der Editor ermöglicht dem Benutzer wie durch ein Fenster auf den zu bearbeitenden Text zu schauen. Es ist in diesem Zusammenhang nur natürlich, daß man bei der Bearbeitung eines Textes sich die Möglichkeit wünscht, weitere Texte gleichzeitig ansehen zu können. Dies kann notwendig sein, um zu vergleichen, Fehler zu entdecken oder Textteile aus einem Fenster in ein anderes zu übertragen.

Um ein neues Editor – Fenster zu "öffnen", betätigt man im Editor

**ESC** **e**

Betätigt man ESC e ungefähr in der Mitte des Bildschirms, hat man das Fenster auf die neue Datei in der unteren Hälfte des Bildschirms und die "alte" Datei in der oberen Bildschirmhälfte. Zunächst wird der Dateiname erfragt. Nach dessen Eingabe und dem Betätigen der 'CR' Taste wird ein Fenster auf eine andere Datei eröffnet. Die obere linke Ecke des Fensters befindet sich an der aktuellen Cursor – Position. Dabei darf sich der Cursor nicht zu sehr am rechten oder unteren Rand befinden, weil das Fenster sonst zu klein würde. In diesem "Fenster" kann man dann genauso arbeiten wie im "normalen" Editor.

Mit der Tastenfolge

**ESC** **w**

wechselt man von einem Fenster (zyklisch) in das benachbarte. Es gibt eine Hierarchie zwischen den Fenstern in der Reihenfolge, in der eines im anderen eingerichtet worden ist. Gibt man

**ESC** **q**

in einem Fenster, so verschwindet dieses und alle darin eingeschachtelten Fenster, und man befindet sich im übergeordneten Fenster.

Wir schilderten zuvor, daß man mit **ESC RUBOUT** und **ESC RUBIN** Texte verschieben und löschen kann. Zwischen Dateien im Fenstereditor geht dies folgendermaßen:

Durch

**ESC** **p**      oder      **ESC** **d**

schreibt man einen markierten Teil in eine temporäre Datei (einen Zwischenspeicher); durch **ESC p** wird ein markierter Text aus der Ursprungsdatei entfernt und in einen Zwischenspeicher geschrieben. Im Gegensatz dazu wird er durch **ESC d** kopiert. Durch

**ESC** **g**

fügt man ihn in eine andere (oder dieselbe) Datei ein. Im Unterschied zu **ESC RUBIN** wird die temporäre Datei dadurch nicht entleert.

Die Funktionen **ESC d** und **ESC g** leisten auf schnellere Weise dasselbe wie die Kommandos 'PUT ""' und 'GET ""'.

## 4.2.10. Die wichtigsten vorbelegten Tasten

*Lernsequenzen und Kommandos (d.h. ELAN-Programme) können Tasten zugeordnet werden. Da einige Funktionen häufig benötigt werden, sind diese standardmäßig bestimmten Tasten zugeordnet.*

ESC q	Verlassen des Editors bzw. der eingeschachtelten Fenster.
ESC e	Weiteres Editorfenster einschalten.
ESC n	Notizbuch "aufschlagen".
ESC v	Dateifenster auf ganzen Bildschirm vergrößern bzw. Bildschirm rekonstruieren (eingeschachteltes Fenster verlassen).
ESC w	Dateiwechsel beim Fenstereditor.
ESC f	Nochmalige Ausführung des letzten Kommandos.
ESC b	Das Fenster wird auf den linken Rand der aktuellen (ggf. verschobenen) Zeile gesetzt.
ESC →	Zum nächsten Wortanfang.
ESC ←	Zum vorherigen Wortanfang.
ESC 1	Zum Anfang der Datei.
ESC 9	Zum Ende der Datei.

## Lernen

- ESC HOP            Lernen einschalten.
- ESC HOP taste    Lernen ausschalten und Lernsequenz auf 'taste' legen.
- ESC HOP HOP     Gelerntes vergessen. Bedingung ist, daß man die Lernsequenz in der Task löscht, in der man sie hat lernen lassen.

## Operationen auf Markierungen

- ESC RUBOUT      Markiertes "vorsichtig" löschen.
- ESC RUBIN        Vorsichtig mit ESC RUBOUT Gelöschtes einfügen.
- ESC p             Markiertes löschen und in die Notiz – Datei schreiben. Kann mit ESC g an anderer Stelle reproduziert werden.
- ESC d             Duplizieren:  
Markiertes in die Notiz – Datei kopieren (PUT ""), anschließend die Markierung abschalten. Kann mit ESC g beliebig oft reproduziert werden.
- ESC g             MIT ESC p gelöschten oder mit ESC d duplizierten Text an aktuelle Cursor – Stelle schreiben, d.h. Notiz – Datei an aktueller Stelle einfügen (GET "").

## Zeichen schreiben<sup>1</sup>

ESC a	Schreibt ein ä.
ESC A	Schreibt ein Ä.
ESC o	Schreibt ein ö.
ESC O	Schreibt ein Ö.
ESC u	Schreibt ein ü.
ESC U	Schreibt ein Ü.
ESC s	Schreibt ein ß.
ESC (	Schreibt eine [.
ESC )	Schreibt eine ].
ESC <	Schreibt eine {.
ESC >	Schreibt eine }.
ESC #	Schreibt ein #, das auch gedruckt werden kann.
ESC -	Schreibt einen (geschützten) Trennstrich, siehe Textverarbeitung.
ESC k	Schreibt ein (geschütztes) "k", siehe Textverarbeitung.
ESC blank	Schreibt ein (geschütztes) Leerzeichen, siehe Textverarbeitung.

## Kommando auf Taste legen

ESC ESC	Kommandodialog einschalten
ESC ! taste	Im Kommandodialog: Geschriebenes Kommando auf Taste legen.
ESC ? taste	Im Kommandodialog: Auf 'taste' gelegtes Kommando zum Editieren anzeigen.
ESC k	Im Kommandodialog: Das zuletzt editierte Kommando (einzeilige ELAN – Programm) anzeigen.

Eine ausführliche Beschreibung des Kommandodialogs finden Sie im folgenden Kapitel.

---

1) Diese Tasten sind standardmäßig so vorbelegt wie hier aufgeführt, sie können aber von Benutzern und in Anwenderprogrammen geändert werden.

## 4.3. Die wichtigsten Editor – Kommandos

### 4.3.1. Der Kommandodialog

*Einige Operationen kann man nur mühselig mit den bis jetzt beschriebenen Tasten durchführen. Z.B. ist es sehr zeitaufwendig, eine bestimmte Textstelle zu finden. Andere Operationen sind mit den im vorigen Kapitel beschriebenen Tasten überhaupt nicht möglich, wie etwa die Zeilenbreite einzustellen oder Programme aufzurufen, die die zu editierende Datei verarbeiten. Solche Operationen werden durch Kommandos ermöglicht, die man auf Editorebene geben kann.*

Um Kommandos an den Editor geben zu können, schalten wir in den Kommando-  
zustand.

**ESC** **ESC**

Durch zweimaliges Betätigen von ESC erfolgt im Editor die Aufforderung

```

..... dateiname ..... Zeile 4
Mit der ESC-Taste ist es möglich, den Kommandodialog
gjb kommando :

```

Auf dem Bildschirm erscheint eine Kommandozeile, in der der Benutzer  
Kommandos schreiben kann. Durch Betätigen der Taste 'CR' wird das  
Kommando ausgeführt.

## 4.3.2. Zeile und Textstelle anwählen

*Auf der Kommandoebene des Editors können Sie Kommandos erteilen, um an eine beliebige Stelle in der Datei zu positionieren.*

Sie haben einen (größeren) Text erstellt und stehen nun vor dem Problem, für die Korrektur die entsprechenden Textstellen aufzufinden.

### Beispiel:

Bei der Durchsicht eines Ausdrucks Ihres Textes stellen Sie fest, daß Sie sich verschrieben haben. Anstelle von "diese Zeichen" haben Sie "diese Ziichen" geschrieben. Um diese Textstelle anzuwählen, gehen Sie wie folgt vor: Sie positionieren an den Beginn der Datei und betätigen die Tastenfolge

ESC

ESC

Auf dem Bildschirm erscheint:



```
qrb kommando:
```

Sie schreiben nun die zu suchende Textstelle auf:



```
qrb kommando: "diese Zeichen"
```

Durch die Angabe eines TEXTes in Anführungsstrichen wird nach dem eingeschlossenen TEXT 'diese Zeichen' ab der aktuellen Cursor-Position gesucht. Wird 'diese Zeichen' gefunden, bleibt der Cursor auf dem gesuchten Text stehen. Andernfalls steht der Cursor am Ende der letzten Zeile der Datei.

Eine andere Möglichkeit, an eine entferntere Stelle im Text zu kommen, ist die folgende:

**ESC** **ESC**

Es erscheint auf dem Bildschirm:

```
qrb kommando:
```

Sie geben nun die Textzeile an, die Sie suchen:

```
qrb kommando: 134
```

Durch dieses Kommando wird auf die 134. Zeile positioniert.

### 4.3.3. Suchen und Ersetzen

*Auf der Kommandoebene des Editors können Sie wie auf der Monitor-Ebene beliebige Kommandos geben. Diese können Sie zu (ELAN-) Programmen verknüpfen. Zur Erstellung dieser Programme editieren Sie wie gewohnt in der Kommandozeile. Für das Positionieren, Suchen und Ersetzen innerhalb Ihres ELAN-Programms stehen Ihnen Kommandos zur Verfügung. Beliebige ELAN-Programme sind zulässig.*

Die Kommandozeile kann wie eine "normale" Textzeile editiert werden (Positionieren, Überschreiben, Einfügen, Löschen und Markieren). Bevor ein Programm eine Ausgabe erzeugt oder fehlerhafte Kommandos Fehlermeldungen hervorrufen, wird der Cursor in die linke obere Ecke positioniert. Um die Meldungen festzuhalten, sollte das Kommando 'pause' folgen. Diese Meldungen werden dann in der ersten Zeile des Bildschirms angezeigt. Danach ist man wieder im Editor und kann wie gewohnt arbeiten.

Kommandos werden durch ein Semikolon voneinander getrennt.

Beispiel:

```
qrb kommando: ff: "Geschäftsführung"; fefebf "Lieferanten"; rch b a
```

Ihr ELAN-Programm besteht aus zwei Kommandos: zunächst positionieren Sie in die erste Zeile und suchen ab dort nach dem Wort "Geschäftsführung". dann lesen Sie die Datei "Lieferanten" von der Diskette in den Arbeitsspeicher.

Die beiden beschriebenen Kommandos (Text bzw. eine Zeile anwählen) sind Spezialkommandos und können in dieser Form nicht durch ein Semikolon mit anderen Kommandos kombiniert werden. Deshalb gibt es für sie eine ELAN – Form, die es erlaubt, sie mit anderen Kommandos zusammen zu verwenden:

- a) Einen Text ab der aktuellen Cursor – Position suchen (D ist eine Abkürzung für 'DOWN'):



```
gib kommando: "diese Zeichen"
```

(\* Kurzform \*)



```
gib kommando: D "diese Zeichen"
```

(\* Allgemeine Version \*)

- b) Auf eine Zeile positionieren (T ist eine Abkürzung für 'TO LINE'):



```
gib kommando: 127
```

(\* Kurzform \*)

```
gib kommando: I 127
```

(\* Allgemeine Version \*)

Mehrere Kommandos können in der Kommandozeile angegeben werden. Die einzelnen Kommandos müssen in diesem Fall mit ';' voneinander getrennt werden.

Beispiel:

**ESC** **ESC**

schaltet in den Kommandomodus

```
gib kommando: I I; D "noch Zeichen"
```

Diese zwei Kommandos werden nacheinander ausgeführt. Zuerst wird auf die erste Zeile positioniert und dann (von der ersten Zeile ab) nach 'noch Zeichen' gesucht. Damit ist es möglich, die Datei nicht nur ab der aktuellen Zeile zu durchsuchen, sondern die gesamte Datei. Soll nicht in Richtung auf das Dateiende, sondern in Richtung auf den Dateianfang (also nach "oben") gesucht werden, kann man das U-Kommando (Abkürzung für UP) verwenden:

**ESC** **ESC**

```
gib kommando: U "noch ein Text"
```

Ein weiteres Kommando ist das C-Kommando (Abkürzung für 'CHANGE'), mit welchem man einen TEXT sucht und diesen dann ersetzt.

Beispiel:

**ESC** **ESC**

```
qrb kommando: "alte Zeichen" C "neue Zeichen"
```

Ab der aktuellen Cursor-Position wird nach 'alte Zeichen' gesucht. Wird der TEXT gefunden, wird er durch 'neue Zeichen' ersetzt. Der Cursor befindet sich in diesem Fall hinter dem ersetzten TEXT. Wird 'alte Zeichen' dagegen nicht in der Datei gefunden, befindet sich der Cursor (wie beim erfolglosen Suchen mit D) am Ende der letzten Zeile der Datei.

Wie alle anderen Kommandos kann auch das C-Kommando mit anderen Kommandos verbunden werden.

Beispiel:

**ESC** **ESC**

qtb kommando: l 500; "Schreibfehler" C "Schreibfehler"

Hier wird ab der 500. Zeile der Datei nach 'Schreibfehler' gesucht und ggf. ersetzt. Soll ein TEXT nicht nur einmal, sondern bei jedem Auftreten ersetzt werden, benutzt man das CA – Kommando (Abkürzung für CHANGE ALL):

**ESC** **ESC**

qtb kommando: "dieser alte Text" CA "dieser neue Text"

Dadurch wird 'dieser alte Text' bei jedem Auftreten ab der aktuellen Cursor – Position durch 'dieser neue Text' ersetzt.

## Pattern Matcher

*Der Pattern Matcher ist ein Werkzeug zur Mustererkennung. Er dient zur Beschreibung von Texten, die in verschiedenen Ausprägungen auftreten können. Zum Suchen oder Ersetzen wird nicht ein Text fester Gestalt vorgegeben, sondern eine Beschreibung der gesuchten Struktur.*

Häufig werden Sie Texte suchen oder ersetzen wollen, die in einigen Varianten innerhalb eines umfangreicheren Textes auftauchen können.

Beispiel: Gesucht wird 'unser' in verschiedenen Zusammenstellungen, also auch 'unsere' oder 'unserem'. Alle Textstellen, die diesem Muster entsprechen, können in einem Suchverfahren gefunden werden, indem das Muster, welches diese Texte beschreibt, für die Suche benutzt wird:

Suchen nach Begriffen deren genaue Ausprägung unbekannt ist.

```
qrb kommando:D? "unser" * "" + " "
```

Leseweise:

Suche 'unser', gefolgt beliebigen Zeichen plus einem Leerzeichen, oder auch nur einem Leerzeichen.

Dieses Suchkommando liefert Treffer bei 'unser', 'unsere', 'unseres' usw..

## Wie baut man ein Pattern ?

*Texte werden durch ihr Konstruktionsmuster aus bekannten und unbekanntem Teilen beschrieben*

Ein Text, der in seiner konkreten Form nicht bekannt ist, dessen Aufbau jedoch durch ein Muster beschrieben werden kann, besteht aus Teilen, die als:

- bekannte Texte
- unbekannte Texte

bezeichnet werden und die mit dem Operatoren:

'+'      Zusammensetzen  
'OR'     Alternative

kombiniert werden können.

Ein bekannter Text ist z.B. ein Stück eines gesuchten Textes, das als fest vorgegeben betrachtet werden kann, wie etwa der Wortstamm 'unser' in dem obigen Beispiel. Wie gewohnt wird ein solcher bekannter Text, in Anführungsstriche gesetzt, als TEXT CONST "text" notiert.

Demgegenüber ist ein unbekannter Text von nicht näher zu beschreibender Gestalt. Das Muster, welches einen unbekanntem Text beschreibt, steht für irgendeinen einer Vielzahl von Texten, die diesem Muster entsprechen.

Mit der Prozedur:

any

wird das Muster für einen beliebigen Text geliefert.

Im einleitenden Beispiel ist der Wortstamm bekannt, das Teilwort 'unser' kann also im 'Klartext' angegeben werden. Die Endungen sind je nach dem Zusammenhang in dem das gesuchte Wort auftritt verschieden, also zunächst unbekannt.

Ein solcher unbekannter Text kann entweder durch Aufzählung der möglichen Alternativen seiner Erscheinung beschrieben werden oder durch die Prozedur 'any'.

(text + ("er" OR "es" OR "em" OR ..... )

alternative Verknüpfung durch OR

("text" + any + .... )

additive Verknüpfung durch +

Grundsätzlich ist zu beachten, daß der Suchvorgang des Pattern Matcher Zeichenketten untersucht und nicht etwa einzelne Worte und stets nach dem längstmöglichen Muster gesucht wird!

Ein schlecht beschriebener Suchtext kostet somit nicht nur viel Rechenzeit, sondern liefert auch unerwünschte Ergebnisse: z.B. sollte der Artikel 'der' mit einem führenden Leerzeichen als " der" gesucht werden, da andernfalls jedes Wort, das die Silbe 'der' enthält, einen Treffer in der Suche ergibt.

Da die Suche nach unbekanntem Texten viele unerwünschte Ergebnisse liefern könnte, kann die Prozedur any in zweifacher Weise eingeschränkt werden:

D(" d" + any (2) )

Die Länge der unbekanntes Textes wird vorgegeben, indem die Anzahl der Zeichen aus denen der Text besteht, angegeben wird. Die Angabe steht in Klammern hinter 'any'. (In diesem Beispiel genau 2 Zeichen).

D(" d" + any ("aeirs"))

Das Alphabet, aus dem der unbekannte Text bestehen darf, wird angegeben. (In diesem Beispiel darf der Text der einen Treffer ergibt nur aus den Zeichen 'a', 'e', 'i', 'r', 's' bestehen, z.B: der, die, das oder auch dies.)

D(" d" + any (2,"aeirs")

Auch die Kombination der Beschränkungen ist möglich. (Jetzt liefern nur noch 'der', 'die','das' etc. Treffer).

**ACHTUNG:** Das Zeichen '\*' nimmt eine Sonderstellung ein, da es als Abkürzung für 'any' verwandt werden kann. Soll dieses Zeichen im Text gesucht oder ersetzt werden, müssen Sie statt "\*" 'any(1,"\*")' schreiben.

Weitere Informationen zum Pattern Matcher finden Sie im EUMEL-Handbuch zur Programmierung.

## 4.3.4. Kommandos auf Tasten legen

*Oft benutzte Kommandos können auf Tasten gelegt werden. Damit ist es möglich, den Editor auf Ihre speziellen Bedürfnisse einzurichten.*

Oft benutzte Kommandos können mit der Drei – Tastenfolge

`ESC` `!` `'taste'` auf eine Taste gelegt werden.

Beispiel:

`ESC` `ESC` (\* die Kommandozeile erscheint \*)

gib kommando: save (SOME myself)

`ESC` `!` `s` (\* das Kommando 'save (SOME myself)' ist nun auf die Taste 's' gelegt \*)

Wird nun die Taste 's' gedrückt, erscheint das Zeichen 's' auf dem Bildschirm. Mit `ESC s` wird das 'save' – Kommando ausgeführt. Natürlich können auch kompliziertere Kommandos auf Tasten gelegt werden.

Möchten Sie ein Kommando, das auf eine Taste gelegt wurde, verändern, drücken Sie im Kommandodialog (!) die Drei – Tastenfolge

`ESC` `?` `'taste'`

Beispiel:

**ESC** **ESC** (\* in den Kommandodialog gehen \*)

**ESC** **?** **s** (\* es erscheint nun: 'save (SOME myself)' \*)

Dieses Kommando kann nun z.B. verändert und ausgeführt (durch 'CR') oder wiederum auf die gleiche oder eine andere Taste gelegt werden (durch ESC ! 'taste').

Im Editor kann das letzte im Kommandodialog eingegebene Kommando durch 'ESC f' wiederholt werden.

### 4.3.5. Texte aus anderen Dateien benutzen

*Manchmal ist es notwendig, einen Text in eine andere Datei zu schreiben (z.B. wenn man diesen Text noch einmal verwenden will) oder einen Text einer anderen Datei in den zu bearbeitenden Text einzufügen. Die 'GET'- und 'PUT'-Kommandos bieten die Möglichkeit, Texte zwischen Dateien auszutauschen (vergl. auch den Abschnitt über paralleles Editieren).*

Mit dem 'GET'-Kommando können wir Texte aus einer anderen Datei an die aktuelle Schreibposition kopieren.

```
gib kommando: GET "absender"
```

holt den Text 'absender'. Wenn also des öfteren Briefe geschrieben werden, braucht man sich den Absender nur einmalig in die Datei 'absender' zu schreiben und kann diesen mit dem Kommando 'GET' (was man auf eine Taste legen kann) u.U. mehrmals an verschiedenen Stellen in die Datei einfügen.

Mit dem 'PUT'-Kommando können wir zuvor markierte Textteile in eine Datei schreiben.

```
gib kommando: PUT "adressen"
```

schreibt einen markierten Text in die Datei 'adressen'. 'adressen' wird ggf. eingerichtet. Ist die Datei 'adressen' bereits vorhanden, so wird erfragt, ob die Datei gelöscht werden kann, um den markierten Text aufzunehmen (überschreiben). Andernfalls wird der markierte Text an den bereits vorhandenen Text in 'adressen' angefügt. Es ist somit durch mehrmaliges Markieren und das 'PUT'-Kommando möglich, Texte aus einer Datei zu sammeln und in eine neue Datei zu geben.

### 4.3.6. Breitere Zeilen bearbeiten

*Der Editor ist auf eine Zeilenbreite von 77 Zeichen eingestellt. Oft ist es notwendig, mit einer anderen Zeilenbreite zu schreiben, welche man mit dem 'limit'-Kommando einstellen kann. Aber auch die Positionierung innerhalb einer Zeile wird dadurch anders, weil breitere Zeilen nicht als Ganzes auf den Bildschirm passen. In diesem Fall wird "gerollt".*

Eine andere Zeilenbreite stellt man durch 'limit' ein. Beachten Sie, daß die eingestellte Zeilenbreite für die gesamte Datei gilt.

Beispiel:



```
gib_kommando: limit (180)
```

Nun können Sie wie gewohnt schreiben. Allerdings wird die aktuelle Zeile, in der man sich befindet, nicht wie gewohnt am Bildschirmende umgebrochen, sondern erst an der Spalte 180, sofern sie nicht vorher durch die 'CR' - Taste beendet wird. Wird über das rechte Bildschirmende hinaus geschrieben, bleibt die Cursor-Position am Ende des Bildschirms erhalten, aber die Zeile wird beim weiteren Schreiben nach links verschoben, "rollt" also nach links (der Anfang der Zeile verschwindet scheinbar nach links).

Mit der Positionierung verhält es sich ähnlich. Wird mit RECHTS über den rechten Bildschirmrand positioniert, wird die Zeile ebenfalls gerollt. HOP RECHTS bewirkt ein Blättern innerhalb einer einzelnen Zeile nach rechts. Analog verläuft es bei verschobener Zeile, wenn nach links (LINKS bzw. HOP LINKS) positioniert wird.

Beim Schreiben von Tabellen kann es sinnvoll sein, das Fenster vorübergehend auf eine andere Anfangsposition (als 1) einzustellen. Das kann mit dem 'margin' – Kommando erfolgen.

Beispiel:



Das Editorfenster zeigt nun einen Ausschnitt aus der Datei, beginnend ab der Spalte 50. In der Titelzeile wird "M50" angezeigt.

### 4.3.6. Breitere Zeilen bearbeiten

*Der Editor ist auf eine Zeilenbreite von 77 Zeichen eingestellt. Oft ist es notwendig, mit einer anderen Zeilenbreite zu schreiben, welche man mit dem 'limit'-Kommando einstellen kann. Aber auch die Positionierung innerhalb einer Zeile wird dadurch anders, weil breitere Zeilen nicht als Ganzes auf den Bildschirm passen. In diesem Fall wird "gerollt".*

Eine andere Zeilenbreite stellt man durch 'limit' ein. Beachten Sie, daß die eingestellte Zeilenbreite für die gesamte Datei gilt.

Beispiel:



```
gtb kommando: limit 180
```

Nun können Sie wie gewohnt schreiben. Allerdings wird die aktuelle Zeile, in der man sich befindet, nicht wie gewohnt am Bildschirmende umgebrochen, sondern erst an der Spalte 180, sofern sie nicht vorher durch die 'CR' - Taste beendet wird. Wird über das rechte Bildschirmende hinaus geschrieben, bleibt die Cursor-Position am Ende des Bildschirms erhalten, aber die Zeile wird beim weiteren Schreiben nach links verschoben, "rollt" also nach links (der Anfang der Zeile verschwindet scheinbar nach links).

Mit der Positionierung verhält es sich ähnlich. Wird mit RECHTS über den rechten Bildschirmrand positioniert, wird die Zeile ebenfalls gerollt. HOP RECHTS bewirkt ein Blättern innerhalb einer einzelnen Zeile nach rechts. Analog verläuft es bei verschobener Zeile, wenn nach links (LINKS bzw. HOP LINKS) positioniert wird.

Beim Schreiben von Tabellen kann es sinnvoll sein, das Fenster vorübergehend auf eine andere Anfangsposition (als 1) einzustellen. Das kann mit dem 'margin' – Kommando erfolgen.

Beispiel:



Das Editorfenster zeigt nun einen Ausschnitt aus der Datei, beginnend ab der Spalte 50. In der Titelzeile wird "M50" angezeigt.

### 4.3.7. Die wichtigsten Kommandos

*Einige Kommandos sind speziell für die Textverarbeitung im Editor programmiert. Die wichtigsten werden hier vorgestellt.*

**any**

TEXT PROC any

liefert ein Muster beliebiger Gestalt und Länge (also auch der Länge 0) für Suchoperationen.

" ir" + any + "was"

**any**

TEXT PROC any (TEXT CONST alphabet)

liefert den längstmöglichen Text, der aus den in 'alphabet' angegebenen Zeichen besteht.

any ("1234567890") (\* suche Zahlen \*)

**any**

TEXT PROC any (INT CONST laenge)

liefert ein Muster beliebiger Gestalt und der Länge 'laenge'.

" d" + any (2)

**any**

TEXT PROC any (INT CONST laenge, TEXT CONST alphabet)

liefert ein Muster der Länge 'laenge', das nur aus Zeichen aus 'alphabet' besteht.

" d" + any (2,"erias")

**C**

OP C (TEXT CONST muster, ersatz)

Ab der aktuellen Position wird 'muster' in Richtung Dateiende gesucht und durch 'ersatz' ersetzt. Der Cursor steht danach hinter 'ersatz'.

"alt" C "neu"

**CA**

OP CA (TEXT CONST muster, ersatz)

Arbeitet ab der aktuellen Position wie C. Die Aktion wird jedoch bis zum Erreichen des Dateiendes wiederholt. Nach Ausführung ist somit jedes 'muster' durch 'ersatz' ersetzt. Der Cursor steht danach hinter dem letzten Zeichen der Datei.

"alt" CA "neu"

**D**

OP D (INT CONST n)

Positioniert das Fenster n Zeilen vorwärts in Richtung auf das Dateiende.

D 50

OP D (TEXT CONST muster)

Sucht 'muster' vorwärts in Richtung auf das Dateiende. Die Suche beginnt direkt hinter der aktuellen Cursor-Position. Wird 'muster' nicht gefunden, steht der Cursor hinter dem letzten Zeichen der Datei. Wird 'muster' gefunden, steht der Cursor direkt auf dem ersten Zeichen von 'muster'.

D "muster"

**GET**

OP GET (TEXT CONST dateiname)

Kopiert den Inhalt der Datei mit dem angegebenen Namen vor die aktuelle Cursor-Position. Ist ein Teil der Quelldatei markiert, wird nur der markierte Teil kopiert.

GET "quelldatei"

OP G (TEXT CONST dateiname)

Wie GET.

**limit**

OP limit (INT CONST limit)

Setzt die rechte Schreibgrenze auf 'limit'.

limit (50)

**margin**

PROC margin (INT CONST anfang)

Alle Zeilen erscheinen erst ab Spalte 'anfang' im Sichtfenster.

margin (50)

**OR**

TEXT OP OR (TEXT CONST texteins,textzwei)

Liefert ein Muster, wenn *texteins* oder *textzwei* gefunden wird. Die Reihenfolge spielt keine Rolle.

D ("Geschäfts" + ("führung" OR "leitung"))

### **PUT**

**OP PUT (TEXT CONST dateiname)**

Richtet eine Datei mit dem angegebenen Namen ein und kopiert den markierten Textabschnitt in diese.

PUT ("meine hilfdatei")

**OP P (TEXT CONST dateiname)**

Zweck: Wie PUT.

### **T**

**OP T (INT CONST n)**

Positioniert auf die Zeile 'n'.

T 999

### **type**

**PROC type (TEXT CONST zeichenkette)**

Fügt 'zeichenkette' in die aktuelle Position der editierten Datei ein. Besonders nützlich in Verbindung mit der Prozedur 'code', um nicht auf der Tastatur enthaltene Zeichen in den Text zu bringen.

type(code(200))

### **U**

**OP U (INT CONST n)**

Positioniert das Fenster n Zeilen rückwärts in Richtung auf den Dateianfang.

U 100

**OP U (TEXT CONST muster)**

Sucht 'muster' rückwärts in Richtung auf den Dateianfang. Die Suche beginnt links neben der aktuellen Cursor-Position. Vergl. D

U "muster"

***word wrap***

PROC word wrap (BOOL CONST an)

Schaltet den automatischen Wortumbruch an (voreingestellt) bzw. aus.

```
word wrap (true)           (* angeschaltet *)
word wrap (false)          (* ausgeschaltet *)
```

## 4.4. Fehlersituationen und Abhilfe

*Von Zeit zu Zeit werden Sie als Anfänger in Arbeitssituationen geraten, wo Sie nicht weiterwissen. Hier sind einige Tips, wie Sie sich behelfen können.*

Wie helfe ich mir, wenn...

nach  
continue("taskname")

der Monitor nicht

gib kommando:

sagt, sondern "schweigt"?

= > Sie haben die Task bei der letzten Benutzung nicht mit dem Kommando 'break' verlassen (evtl. haben Sie SV betätigt?). Sie sind jetzt im Editor, sehen aber den zuletzt bearbeiteten Textausschnitt nicht. **Betätigen Sie die Tasten**

**ESC b**

und der Text wird neu auf dem Bildschirm ausgegeben.

im Editor kein Tastendruck mehr akzeptiert wird?

- = > Sie haben irrtümlich die STOP – Taste (auch oft als CTRL a realisiert, abhängig vom Terminal), d.h. Anhalten der Bildschirmausgabe betätigt.

Drücken Sie die WEITER – Taste (= CTRL c, d.h. Bildschirmausgabe fortführen). Alle Tastenanschläge, die zwischenzeitlich zu keiner Reaktion führten, werden jetzt ausgegeben.

Je nach Tastatur können STOP und WEITER auch auf anderen Tasten liegen.

der Lernmodus über lange Zeit (ungewollt) eingeschaltet war?

- = > a) Sie merken plötzlich, daß über einen unbestimmt langen Zeitraum alle Ihre Tastenanschläge gelernt wurden (zu erkennen an der "LEARN" – Anzeige in der Überschriftzeile).

Was ist zu tun?

Mit dem Kommando

ESC HOP HOP

vergessen Sie alles Gelernte und schalten den Lernmodus aus.

= > b) Sie beenden den Editor mit ESC q und die Meldung



erscheint auf dem Bildschirm.

Was ist zu tun?

Sie können mit

ESC HOP HOP

das Gelernte sofort vergessen.

Sie zu viele Absatzzeichen in Ihrem Text gesetzt haben und diese entfernen müssen?

= > Sie positionieren in die Zeile, in der die Absatzmarke gelöscht werden soll. Sie betätigen dann die TAB-Taste, um hinter den Text zu positionieren, dann die RUBOUT-Taste. Wenn Sie jetzt die Zeile mit den Cursor-Tasten nach oben oder unter verlassen, verschwindet die Absatzmarkierung.

nach

```
save("dateiname", "vatertask")
```

das Betriebssystem nicht mehr reagiert?

- = > Sie haben die Vater-Task nicht mit dem Kommando 'global manager' in jenem Prozeß zum Empfang von Daten aus anderen Prozessen vorbereitet.

Sie in Ihrer Task das Archive mit dem Kommando

```
archive("archivname")
```

anmelden wollen und das System Ihnen die Meldung

"Fehler: Archive wird von Task "bib" benutzt"

zustellt?

- = > Es gibt zwei Möglichkeiten:

- a) Ein anderer Benutzer benötigt das Archiv-Laufwerk in diesem Moment. Sie müssen warten, bis er seine Arbeit beendet hat.
- b) Ein anderer Benutzer (oder Sie selbst) hat vergessen, mit dem Kommando

```
release(archive)
```

das Archiv in jener Task freizugeben. Falls Sie es selbst waren, holen Sie das nach. Ansonsten kann das Archiv-Kommando wieder erfolgreich gegeben werden, wenn fünf Minuten nicht auf das Archiv zugegriffen wurde.

Sie eine (scheinbare oder echte) Endlosschleife auf einer Taste (z.B. Taste "x") gelernt haben und diese (versehentlich oder bewußt) durch 'ESC x' aktivieren?

- = > Wie immer, wenn Sie eine endlos laufende Task beenden wollen, gelangen Sie mit der SV – Taste in den Supervisor – Modus und mit dem Kommando

'halt'

beenden Sie die Endlosschleife.

Mit

ESC HOP HOP x

wird danach das Gelernte 'vergessen'!

Sie Ihre Datei verlassen wollen und

'ESC q'

(scheinbar) nicht funktioniert?

- = > Sie haben versehentlich den Feststeller für Großbuchstaben (SHIFT LOCK / CAPS LOCK) betätigt und ESC q zeigt keine Wirkung (wie auch andere Tastenkombinationen mit Großbuchstaben evtl. keine Wirkung zeigen).

## TEIL 5: Textkosmetik und Druck

### 5.0. Vorwort

Die Textkosmetik – Programme des EUMEL – Systems bieten Ihnen eine einfach zu erlernende und zu bedienende Möglichkeit, Texte für den Druck zu gestalten (programmtechnisch: formatieren) und zu manipulieren.

Die Textkosmetik – Programme bearbeiten Ihre Dateien, die durch den EUMEL – Editor erstellt wurden. Darum sollten Sie sich zuerst mit dem EUMEL – Editor vertraut machen.

Die Programme sind so konstruiert, daß die meisten Aufgaben durch in den Text eingefügte Anweisungen gesteuert werden. Solche Angaben für die Textkosmetik und den EUMEL – Drucker nennen wir im folgenden kurz **'Anweisung'**. Die Form der Anweisung ist für die Textkosmetik und den EUMEL – Drucker gleich und entspricht der ELAN – Syntax. Beachten Sie den Unterschied zwischen einem Kommando und einer Text – Anweisung: während ein Kommando direkt ausgeführt wird, wird eine in den Text eingebettete Anweisung erst nach dem Aufruf von Textkosmetik – und Drucker – Programmen wirksam.

Die Wirkungsweise der Textkosmetik – Anweisungen ist leicht zu erlernen und kann vor allen Dingen stufenweise erfolgen. Deshalb ein guter Rat für Anfänger: Lesen Sie diesen Teil des Benutzer – Handbuchs erst oberflächlich, so daß Sie ungefähr Bescheid wissen, welche Möglichkeiten die Textkosmetik – Programme bieten. Dann können Sie diejenigen Teile der Textkosmetik auswählen und bei Bedarf anwenden, die Sie für Ihre spezielle Anwendung benötigen.

## 5.1. Einführung in die Benutzung der Textkosmetik

*In diesem Kapitel erhalten Sie eine Übersicht über die verfügbaren Programme der Textkosmetik.*

### Schreiben, Gestalten und Drucken von Texten

*Im EUMEL – System unterscheiden wir zwischen drei Stufen einer Textbehandlung: **Erstellung**, **Gestaltung** und **Druck**. Die Trennung in verschiedene Arbeitsstufen hat den Vorteil, daß Sie sich zu einem Zeitpunkt nur auf einen Arbeitsschritt konzentrieren müssen.*

### Texterstellung bzw. Textbearbeitung

Das Schreiben von Texten wird mit Hilfe des Editors erledigt. In dieser Stufe der Texterstellung können Sie sich ausschließlich auf das Schreiben und die inhaltliche Korrektheit Ihres Textes konzentrieren. Wird ein Text ohne Anweisungen gedruckt, dann erscheint er so, wie er mit dem Editor geschrieben wurde. Bei der Erstellung des Textes können Sie aber auch bereits Textkosmetik – Anweisungen in den Text einfügen.

Es ist wichtig, daß Sie das Kapitel 'Editor'  
sehr gründlich lesen. ■

Druckbild:

Es ist wichtig, daß Sie das Kapitel 'Editor'  
sehr gründlich lesen.

Sie sollten Texte im 'Fließtext' – Modus erstellen, d.h., dann werden Worte, die über Zeilengrenzen gehen, ohne Silbentrennung vom Editor in die nächste Zeile gebracht.

## Textkosmetik bzw. Textgestaltung

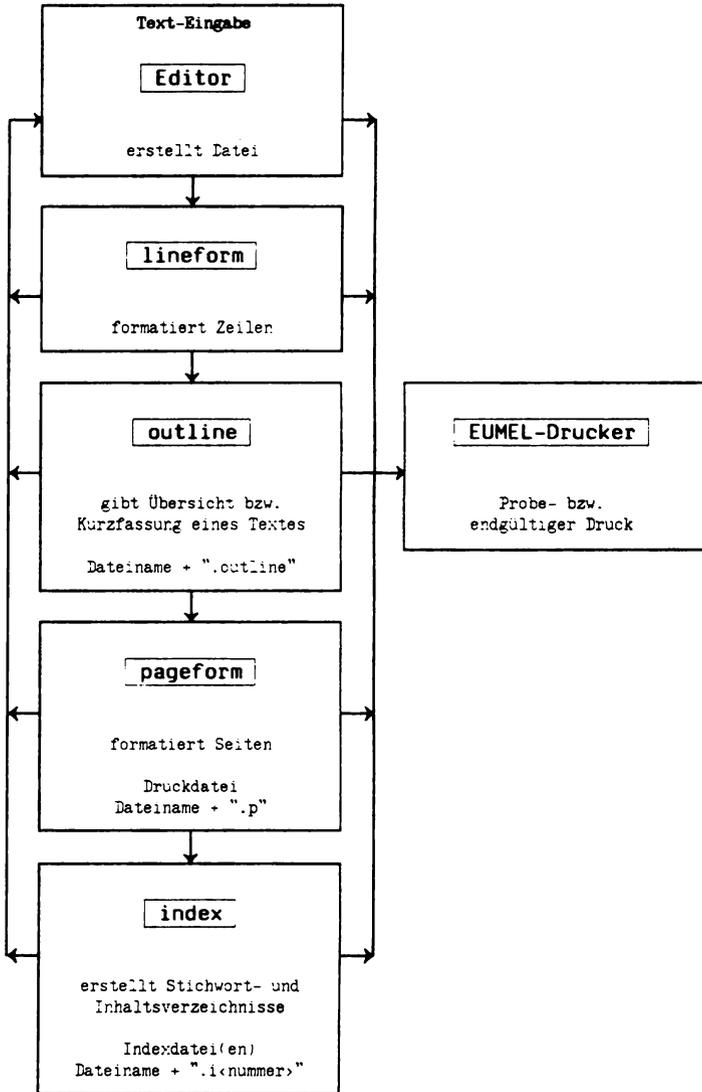
Nachdem Sie einen Text geschrieben haben, können Sie ihn mit Textkosmetik-Programmen gestalten, ohne ihn inhaltlich zu verändern. Dies kann auch vor oder nach eventuellen Korrekturen erfolgen. Die Textkosmetik bietet zur Zeit vier Programme an, die je nach Bedarf eingesetzt werden können:

- - - 'lineform'/'autoform' formatiert einen Text zeilenweise und vollzieht eine Silbentrennung. Weiterhin erlaubt 'lineform'/'autoform' die Verwendung unterschiedlicher Schrifttypen und Schrifthöhen.
- - - 'pageform'/'autopageform' gestattet die Formatierung eines Textes in Seiten (drucktechnisch: "Seitenumbruch"). Dabei berücksichtigt 'pageform'/'autopageform' unterschiedliche Schrifthöhen. Es ist mit 'pageform'/'autopageform' u.a. möglich, die Seiteneinteilung zu bestimmen, eine Seite in Spalten zu formatieren ("Zeitungsformat"), Zeilen am Anfang bzw. Ende jeder Seite einfügen zu lassen, eine Seitennumerierung (drucktechnisch: "Paginierung") zu erhalten und Fußnoten zu gestalten.

- - - 'index' erlaubt die Erstellung von Stichwort- und Inhaltsverzeichnissen aus einer mit 'pageform'/'autopageform' bearbeiteten Datei.
- - - 'outline' holt aus einer Datei alle mit Index-Anweisung gekennzeichneten Überschriften und Stichworte. Es erstellt somit eine Übersicht bzw. Kurzfassung eines Textes.

## Drucken

Zu jedem Zeitpunkt der Texterstellung kann gedruckt werden. Der EUMEL-Drucker beachtet die gleichen Anweisungen wie die Textkosmetik-Programme und noch einige zusätzliche, die nur für die Druckaufbereitung notwendig sind. Spezielle Druckleistungen, wie z.B. verschiedenartige Schrifttypen, können nur auf besonderen Druckern erzeugt werden. Verfügt ein Drucker nicht über eine bestimmte Hardware-Eigenschaft, wird die von ihm geforderte Leistung ignoriert. Somit ist es möglich, Probedrucke für Korrekturen etc. auch auf preiswerten Druckern herzustellen. (siehe hierzu 5.6.1.)



## 5.1.1. Anweisungen für die Textkosmetik und den Drucker

*In diesem Abschnitt wird beschrieben, wie Sie Anweisungen für die Textkosmetik – und Druckprogramme in einen Text einfügen können. Beachten Sie, daß jede Anweisung von '#'-Zeichen eingeschlossen werden muß. Benötigen Sie das '#'-Zeichen in Ihrem Text, müssen Sie es mit 'ESC' schreiben.*

Es gibt zwei Arten von Anweisungen:

- a) Anweisungen, die das gesamte Aussehen eines Manuskripts verändern ("*layout-Anweisungen*"). Zu diesen Anweisungen gehören die Anweisungen `#limit (...)#` (Einstellen der Zeilenbreite), `#linefeed (...)#` (Zeilenabstand), `#page#` (neue Seite) usw. Diese Anweisungen gelten erst ab der nächsten Zeile und Sie sollten sie daher in eine extra Zeile zwischen den Text stellen.

```

#type ("trium8")##limit (11.0)#
#start(5.0,1.5)#
#pagelength(17.4)##pagenr("%",148)##setcount(1)#
#block##pageblock#
#count per page#
#headeven#
#lpos(0.0)##cpos(5.5)##rpos(11.0)#
#table#
                                EUMEL-Benutzerhandbuch
#fillchar(" ")#
#on("u")#           #off("u")#
#table end##clear pos#

#end#
#headodd#
#lpos(0.0)##cpos(5.5)##rpos(11.0)##fillchar(" ")#
#table#
                                Teil 5: Textkosmetik und Druck
#fillchar(" ")#
#on("u")#           #off("u")#
#table end##clear pos#

#end#

```

Das Druckbild (das Ergebnis der Anweisungen) sehen Sie im vorliegenden Benutzerhandbuch.

Anweisungen, die für den Gesamttext gelten sollen, müssen Sie an den Anfang der Datei stellen (noch vor #head#).

- b) Anweisungen, die unmittelbar auf den nachfolgenden Text wirken sollen, wie z.B. #type# (Schrifttyp), #on#/#off# (Modifikationen wie unterstreichen oder fett drucken), #ib#/#ie# (Markierung von Stichworten) usw. Solche Anweisungen werden unmittelbar beachtet und können überall auf einer Zeile stehen (wie in dem folgenden Beispiel).

`#on("underline")#Ausnahmen#off("underline")#` werden bei der Beschreibung der Anweisungen speziell erwähnt. ■

Druckbild:

Ausnahmen werden bei der  
Beschreibung der Anweisungen speziell erwähnt.

Weitere Beispiele für Textkosmetik – Anweisungen:

```
#page#  
#free(3.0)#  
#type("quadrato")#
```

Diese Anweisungen entsprechen – wie alle Kommandos im EUMEL-System – der ELAN-Syntax (u.a. müssen sie klein geschrieben werden; Parameter in runden Klammern; mehrere Parameter werden durch Kommata getrennt; TEXT-Parameter in Anführungsstrichen; REAL-Parameter mit Dezimalpunkt usw.). Leerzeichen spielen (außer in TEXT-Parametern) keine Rolle und können zur besseren Lesbarkeit beliebig verwendet werden.

Die Zeichen, aus denen eine Anweisung besteht, werden bei der Formatierung einer Zeile oder Seite nicht mitgezählt und vom EUMEL-Drucker nicht gedruckt. Eine Zeile, die nur aus Anweisungen besteht, wird ebenso behandelt, auch wenn sie mit **CR** abgeschlossen wird.

## 5.1.2. Aufruf der Textkosmetik – Programme

*In diesem Abschnitt wird beschrieben, wie Sie die Textkosmetik – Programme aktivieren können.*

Sie rufen die Textkosmetik – Programme durch Kommandos auf (d.h. in der 'gib kommando' – Ebene). Sie geben ebenso wie zum Editieren den Namen des Programms und der Datei an.

```
gib kommando:  
lineform ("dateiname")
```

```
oder:  
autoform ("dateiname")  
pageform ("dateiname")  
autopageform ("dateiname")  
outline ("dateiname")  
index ("dateiname")
```

'lineform'/'autoform' können Sie auch vom EUMEL – Editor aus aufrufen. Zu diesem Zweck markieren Sie den zu formatierenden Abschnitt der Datei und geben im Kommando – Zustand ( **ESC** **ESC** drücken) 'lineform' bzw. 'autoform' (ohne Parameter).

Das Programm 'pageform'/'autopageform' erzeugt aus der Eingabedatei eine Druckdatei, die den Namen der angegebenen Eingabedatei mit dem Zusatz '.p' bekommt.

```
gib kommando:  
pageform ("dateiname")
```

Als Ergebnis erhalten Sie: "dateiname.p"

Das Programm 'index' kann nur eine Druckdatei bearbeiten:

```
gib kommando:  
index ("dateiname.p")
```

und erstellt die angeforderten Verzeichnisse in Dateien, die mit dem Zusatz '.i<nummer>' gekennzeichnet werden.

Beispiele: "dateiname.i1", "dateiname.i2" etc.

'outline' erstellt ebenfalls eine neue Datei.

```
gib kommando:  
outline ("dateiname")
```

führt zu dem Ergebnis: "dateiname.outline"

### 5.1.3. Vorzeitiger Abbruch und Fehlermeldungen

*Sie können alle Textkosmetik-Programme vorzeitig abbrechen. Eventuelle Fehlermeldungen werden Ihnen in einem Fenster angezeigt.*

Durch die **ESC** - Taste oder die **SV** - Taste und das Supervisor - Kommando 'halt' können Sie die Textkosmetik - Programme jederzeit vorzeitig abbrechen. Die Eingabedatei steht Ihnen dann unverändert zur Verfügung. Ein vorzeitiger Abbruch kann notwendig sein, wenn Sie ein Programm mit einer falschen Datei aufgerufen haben oder zu viele Fehler gemeldet wurden.

Alle Textkosmetik - Programme melden Fehler, wenn Sie Anweisungen falsch benutzen. Die Fehlermeldungen werden auf dem Bildschirm angezeigt. Bei Beendigung eines Programms wird - falls Fehler entdeckt wurden - automatisch der Fenster - Editor aufgerufen, wobei die Fehlermeldungen im unteren Fenster (das ist das Notizbuch) angezeigt werden, während Ihnen im oberen Fenster die Eingabedatei zur Korrektur angeboten wird.

```
.....datename.....Zeile 1 ■
```

```
#cornerl("-5.0")##on("i")#
```

Sie können alle Textkosmetik-Programme vorzeitig abbrechen.  
Eventuelle Fehlermeldungen werden Ihnen in einem Fenster ange-  
zeigt.

```
#box3("T","2","115.0")##off("i")#
```

```
■
```

```
.....notebook.....Zeile 1 ■
```

FEHLER Zeile 1: Unbekannte Anweisung (ignoriert): cornerl("-5.0")

>>> Bitte Korrigieren

FEHLER Zeile 5: Unbekannte Anweisung (ignoriert):

```
box3("T","2","115.0")
```

>>> Bitte Korrigieren

Um von der Eingabedatei zum Notizbuch – und umgekehrt – zu wechseln, betätigen  
Sie  ESC  w.

## 5.2. Lineform/Autoform

*Die Programme 'lineform' oder 'autoform' formatieren einen Text zeilenweise (ggf. mit Silbentrennung) unter Berücksichtigung von Schrifttyp und Zeilenbreite.*

Zur Zeilenformatierung werden Ihnen zwei Programme (Kommandos) angeboten, die sich nur in ihrem interaktiven Charakter unterscheiden (Behandlung von Silbentrennungen):

--- **autoform:**

Zeilenformatierung mit automatischer Silbentrennung. Sie sollten 'autoform' nur bei Texten einsetzen, in denen einige wenige Trennfehler nicht von großer Bedeutung sind, z.B. bei Probedrucken.

--- **lineform:**

Zeilenformatierung mit Silbentrennung "per Hand", wobei (nach deutschen Trennregeln) ein sinnvoller Trennvorschlag gemacht wird. Die Trennstelle kann interaktiv soweit verschoben werden, wie das zu trennende Wort noch auf die Zeile paßt.

'lineform'/'autoform' hat im wesentlichen vier Aufgaben:

--- **Auffüllen von Zeilen:**

'lineform'/'autoform' kann besonders gut nach Korrekturen eingesetzt werden, bei denen – nach Einfügungen oder Lösungen – nicht vollständige oder zu lange Zeilen in der Datei stehenbleiben können.

--- **Erstellen von Zeilen mit unterschiedlichen Schrifttypen:**

Werden in einer Datei mehrere Schriftarten (#type# – Anweisung) verwendet, berechnet 'lineform'/'autoform' nach der eingestellten Zeilenbreite die Anzahl der Zeichen, die auf eine Zeile passen.

- - - - Bearbeitung unterschiedlicher Zeilenbreiten:  
Manchmal ist es notwendig, die Zeilenbreite zu verändern (#limit#-Anweisung). Dies wird von 'autoform'/'lineform' berücksichtigt.
- - - - Silbentrennung:  
Automatische ('autoform') und interaktive Silbentrennung ('lineform').

'lineform'/'autoform' akzeptiert als Eingabe eine Datei und verändert diese. Dafür wird eine (interne) Zwischendatei benötigt. Deshalb müssen Sie darauf achten, daß noch ausreichend Platz auf dem System ist, der jedoch nur zwischenzeitlich für den Formatierungsschritt benötigt wird.

'lineform' und auch 'pageform' sind auf den ersten Schrifttyp der Fonttabelle, auf eine Zeilenbreite von 16.0 und eine Seitenhöhe von 25.0 initialisiert. Sind die ersten Anweisungen, die das verändern könnten, fehlerhaft, so bleiben diese Werte (wie auch sonst bei ignorierten Anweisungen) erhalten.

'lineform'/'autoform' fragt nach dem Kommando an, mit welchem Schrifttyp und mit welcher Zeilenbreite die Datei formatiert werden soll. Dabei erscheinen zuerst die voreingestellten Anweisungen. Beispiel:

```
LINEFORM (für ... Zeilen): dateiname
```

```
Bitte Schrifttyp: micro
```

```
Zeilenbreite (in cm): 16.0
```

Diese Anweisungen können Sie jetzt durch Ihre gewünschten Anweisungen ersetzen. Diese Informationen werden von 'autoform'/'lineform' in Form von #limit#- und #type#-Anweisungen in der Datei vermerkt, so daß die Anfragen bei weiteren Datei-Bearbeitungen entfallen.

Bei Zeilen, die länger als die angegebene Zeilenbreite sind, werden diejenigen Worte, die über die Zeilenbreite hinausgehen, in die nächste Zeile umgebrochen. Kürzere Zeilen werden aus der nachfolgenden Zeile bis zur Zeilenbreite aufgefüllt. Worte werden jedoch nicht über Absatzgrenzen hinweg verschoben. Deshalb sollten Sie vor Anwendung von 'lineform'/'autoform' darauf achten, daß Absätze richtig markiert wurden. Fehlende Markierungen sollten Sie nachträglich einfügen ( CR am Ende einer Zeile), andernfalls werden Zeilen über Absatzgrenzen zusammengezogen. Dies ist besonders bei Tabellenzeilen unangenehm.

Eintrückungen (Leerzeichen am Anfang einer Zeile) werden von 'lineform'/'autoform' ebenfalls bei der Formatierung von Zeilen eingehalten.

## 5.2.1. Zeilenweise formatieren

### 5.2.1.1. Interaktive Silbentrennung

*'lineform' trennt Silben interaktiv, d.h., es werden Ihnen von 'lineform' Trennungsvorschläge gemacht, die Sie bestätigen oder ablehnen können.*

Paßt ein Wort nicht mehr ganz auf eine Zeile, dann wird es zur interaktiven Trennung angeboten. Bei der Trennung werden Anweisungen innerhalb des Wortes entsprechend berücksichtigt. Die Umgebung dieses Wortes wird zur Erleichterung des Trennvorgangs mit angezeigt. Das Trennzeichen erscheint an einer sinnvollen Stelle im zu trennenden Wort.

Text vor dem Trennwort; das

**Trenn**-wort steht mit diesem Text in dieser Zeile

Der Teil des zu trennenden Wortes, der noch auf die Zeile passen würde, wird markiert angezeigt. Sie können das Trennzeichen mit Hilfe der Positionierungstasten innerhalb des Trennbereichs verschieben. An der gewünschten Trennposition (der Wortteil, der noch auf die Zeile kommen soll, steht links vom Trennstrich) kann die **CR**-Taste betätigt werden. **CR** zeigt dem Programm 'lineform' an, daß an dieser Stelle die Trennung erfolgen soll. 'lineform' fügt an den ersten Teil des Wortes das " - " - Zeichen an und schreibt den abgetrennten Wortteil in die nächste Zeile.

Es stehen folgende Operationen bei der interaktiven Trennung zur Verfügung:

**Taste**

**Bedeutung**

**CR**

Trennen.

←

Trennzeichen um ein Zeichen nach links verschieben.

→

Trennstelle um ein Zeichen nach rechts verschieben.

**HOP** ←

Trennstelle vor das Wort setzen (das Wort wird an dieser Position nicht getrennt).

**HOP** →

Trennstelle an das Ende der Markierung setzen.

**BLANK**

Trennzeichen wird von "–" auf " " umgeschaltet. Dies kann verwendet werden, um Worte, die nicht zusammengeschrieben werden sollen, beim Trennvorgang in zwei Worte aufzuspalten.

–

Schaltet das Trennzeichen von Leerzeichen (" ") wieder auf den Trennstrich ("–") um.

**ESC**

Abbruch von 'lineform'/'autoform'. Die zu bearbeitende Datei steht unverändert zur Verfügung.

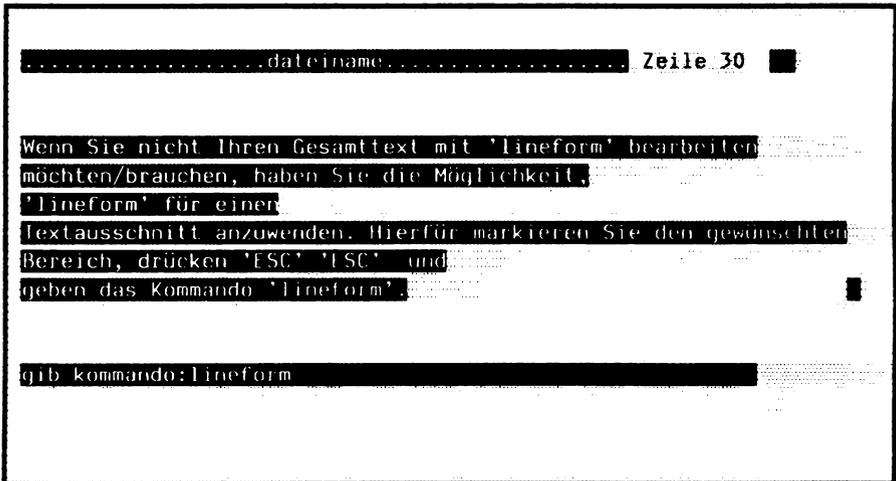
Zwei Besonderheiten sind bei der interaktiven Trennung noch zu beachten:

- Bei Worten mit Bindestrich wird die Trennstelle hinter dem Bindestrich als Leerzeichen angezeigt.
- Bei einer Trennposition zwischen den Zeichen "ck" wird das Zeichen "c" in ein "k" umgewandelt.

Beispiel:            Druk – ker

Sofern es für die Zeilenformatierung notwendig ist, macht die Prozedur 'lineform' bereits erfolgte Trennungen rückgängig (das Trennzeichen wird entfernt und die Wortteile werden wieder zusammengefügt), wenn sich das getrennte Wort (etwa durch Korrekturen oder Veränderungen der Zeilenbreite) nicht mehr am Zeilenende befinden sollte.

Wenn Sie nicht Ihren Gesamttext mit 'lineform' bearbeiten möchten, haben Sie die Möglichkeit, 'lineform' auf einen Textausschnitt anzuwenden. Hierfür markieren Sie den gewünschten Bereich, drücken **ESC** **ESC** und geben im Editor das Kommando 'lineform'. (siehe S. 9)



## 5.2.1.2. Automatische Silbentrennung mit 'autoform'

*'autoform' arbeitet wie 'lineform', nur werden die Silbentrennungen automatisch vorgenommen.*

Ist eine Silbentrennung bei der Formatierung notwendig, übernimmt 'autoform' diese automatisch. Die Trennungen werden in das Notizbuch eingetragen. Nach Beendigung der Formatierung wird die bearbeitete Datei und das Notizbuch zur Kontrolle der Silbentrennungen angezeigt. Die automatische Silbentrennung arbeitet mit einer hohen Trenngüte; allerdings nur für deutsche Texte. Trotzdem kann es vorkommen, daß einige Trennungen, insbesondere bei zusammengesetzten Worten, falsch vorgenommen werden. In einem solchen Fall müssen Sie diese nachträglich mit dem Editor korrigieren. (vgl. Sie dazu auch 5.8.4.)

---

## 5.2.2. Unterschiedliche Schriften

*Unterschiedliche Schrifttypen (Schriftarten) können Sie mit der #type ("schriftname")# – Anweisung anfordern.*

Sie haben die Möglichkeit, mit 'lineform' verschiedenartige Schrifttypen (kurz Typen genannt) verarbeiten zu lassen. Jede Type hat eine bestimmte Höhe und jedes Zeichen hat eine bestimmte Breite. Alle Typen werden auf einer Grundlinie gedruckt.

Es gibt zwei Arten von Schriften:

bei **äquidistanten Schriften** sind alle Zeichen gleich breit (wie bei einer Schreibmaschine). **Proportionalschrift** findet man in gedruckten Büchern. Hier haben unterschiedliche Zeichen auch unterschiedliche Breiten. Die Zeichen ".", "i", "l" sind z.B. schmaler als die Zeichen "w", "o", "m" usw.

Mit der Anweisung:

```
#type ("schriftname")#
```

kann auf einen anderen Schrifttyp umgeschaltet werden (auch mehrmals innerhalb einer Zeile). Diese Type gilt solange, bis wieder eine neue #type ("schriftname")# – Anweisung gegeben wird.

```
#type ("micro")#Jetzt schreiben wir mit einem  
Schrifttyp, der 'micro' heißt. Und jetzt  
#type ("modern15")#schalten wir auf eine an-  
dere Schriftart um. Nun #type ("modern12")#  
möchten wir mit einer größeren Type schrei-  
ben. Um wieder zu unserem gewohnten Schrift-  
typ zu gelangen, schalten wir auf #type  
("trium8")# zurück.
```

Druckbild (ohne 'lineform'):

```
Jetzt schreiben wir mit einem  
Schrifttyp, der 'micro' heißt. Und jetzt  
schalten wir auf eine an-  
dere Schriftart um. Nun  
möchten wir mit einer größeren Type schrei-  
ben. Um wieder zu unserem gewohnten Schrift-  
typ zu gelangen, schalten wir auf  
#type ("trium8")# zurück.
```

Welche Schriftarten Ihnen zur Verfügung stehen, hängt natürlich von dem verfügbaren Drucker ab. Sie können die vorhandenen Schrifttypen mit dem Kommando 'list fonts' erfragen.

Schrifttypen können modifiziert, d.h. verändert, gedruckt werden (vergl. Sie dazu den nächsten Abschnitt). Durch die Angabe einer #type ("schriftname")#-Anweisung werden alle Modifikationen ausgeschaltet.

### 5.2.3. Veränderung des Schrifttyps

Mit der `#on ("..."#`- und `#off ("..."#`-Anweisung können Sie einen Schrifttyp in seinem Aussehen verändern. Die Schrift wird zwar nicht gewechselt, aber verändert gedruckt. Zur Zeit ist unterstrichen, **fett**, *kursiv* und der Druck von **weiß auf schwarz** möglich (abhängig vom eingesetzten Drucker).

Die `#on##off#`-Anweisung wirkt wie ein Schalter, der die gewünschte Schrifttyp-Modifikation ein- bzw. ausschaltet. Die Anweisung `#on#` schaltet die Modifikation ein, `#off#` schaltet sie aus.

Das EUMEL-System ermöglicht es Ihnen,

```
#on ("italic")#kursiv#off ("italic")#
#on ("i")#           #off ("i")#
```

und

```
#on ("underline")#unterstrichen#off ("underline")#
#on ("u")#           #off ("u")#
```

und

```
#on ("bold")#fett#off ("bold")#
#on ("b")#           #off ("b")#
```

und

```
#on ("reverse")#invers (weiß auf schwarz)#off ("reverse")#
#on ("r")#           #off ("r")#
```

zu schreiben.

Druckbild:

Das EUMEL – System ermöglicht es Ihnen,

*kursiv*

und

unterstrichen

und

**fett**

und

**gröÙers (geiß auf schwarz)**

zu schreiben.

Dabei sollten Sie folgendes beachten:

- a) Ein #type# – Anweisung schaltet immer eine vorausgehende Modifikation aus, d.h. ein Schrifttypwechsel macht eventuelle #off ("b")# – , #off ("u")# – , #off ("i")# – und #off ("r")# – Anweisungen überflüssig.
- b) 'lineform'/'autoform' erzeugt eine Warnung, falls Sie vergessen haben, eine Modifikation auszuschalten.
- c) Nicht alle Drucker können die hier angegebenen Modifikationen auch drucken. Welche Modifikationen gleichzeitig eingeschaltet werden können, ist ebenfalls druckerabhängig.

## 5.2.4. Gesperrt schreiben

*Die Silbentrennung an einem Leerzeichen verhindert man durch Verwendung des geschützten Leerzeichens 'ESC' und 'Leertaste'.*

Möchten Sie ein Wort **g e s p e r r t** schreiben, muß natürlich verhindert werden, daß dieses Wort beim Formatieren getrennt wird. Andere Worte, wie z.B. in Formeln, sollten ebenfalls zusammen auf eine Zeile geschrieben werden (z.B. 'sin (x)'). Dies können Sie erreichen, indem Sie nicht das Leerzeichen zwischen die Zeichen schreiben, denn das Leerzeichen bedeutet für 'autoform'/'lineform' immer das Ende eines Wortes. Stattdessen verwenden Sie **ESC** **Leertaste**. Das geschützte Leerzeichen erscheint auf dem Bildschirm zur besseren Identifizierung invers dargestellt bzw. als ein anderes Zeichen (abhängig von Ihrem Gerät). Beim Drucken wird jedoch wieder ein Leerzeichen produziert.



gesperrt

Druckbild:

g e s p e r r t

## 5.2.5. Zeilenbreite einstellen

*Mit der #limit#-Anweisung können Sie die Zeilenbreite einstellen.*

Die #limit#-Anweisung gibt in cm an, wie breit die Zeile sein soll. Beachten Sie, daß diese Anweisung nichts mit dem Editor-Kommando 'limit' zu tun hat. Dieses gibt an, wie viele Zeichen eine Bildschirmzeile lang sein soll.

Die Zeilenbreite wird zusammen mit dem Schrifttyp beim erstmaligen Aufruf von 'autoform'/'lineform' interaktiv erfragt und als #limit#-Anweisung (zusammen mit der #type#-Anweisung) in die erste Zeile der Datei eingetragen. Sie kann in einer Datei mehrmals verändert werden.

Die neue Zeilenbreite gilt immer ab der nächsten Zeile, die der #limit#-Anweisung folgt. Beachten Sie, daß Sie als Parameter in der #limit#-Anweisung eine Zahl mit Dezimalpunkt und Nachkommastelle angeben müssen.

```
#limit(9.0)#
```

Mit der #limit#-Anweisung können Sie Paragraphen in einem anderen Format leicht gestalten. Die rechte Schreibgrenze wird durch die #limit#-Anweisung eingestellt, wobei Sie den linken Rand durch eine entsprechende Einrückung gestalten können.

```
#limit(11.0)#
```

Druckbild (mit 'lineform' bearbeitet):

Mit der #limit# – Anweisung können Sie Paragraphen in einem anderen Format leicht gestalten. Die rechte Schreibgrenze wird durch die #limit# – Anweisung eingestellt, wobei Sie den linken Rand durch eine entsprechende Einrückung gestalten können.

Die folgende Tabelle gibt sinnvolle 'limit' – Einstellungen für die am häufigsten verwendeten Papiergrößen an:

<b>Format</b>	<b>'limit' (Zeilenbreite)</b>	<b>Verbleibender Rand</b>
DIN A4	16.0 cm	je 2.50 cm
DIN A5	12.0 cm	je 1.42 cm
DIN A4 quer	25.0 cm	je 2.35 cm

## 5.2.6. Einfache Tabellen und Aufzählungen schreiben

*Aufzählungen und einfache Tabellen werden automatisch richtig formatiert und gedruckt, wenn Sie sich an einige einfache Regeln halten.*

Verwenden Sie eine Proportionschrift beim Tabellenschreiben, so sind die Spalten in der Regel unterschiedlich breit, selbst wenn Sie eine gleiche Anzahl Zeichen in jeder Spalte schreiben. Dies können Sie durch das Schreiben von einem "Doppelblank" ("Mehrfachblank") vermeiden; für kompliziertere Tabellen gibt es spezielle Tabellenanweisungen. (siehe auch S. 5 – 31)

```
iiii ooooo  
mmm lllll
```

Druckbild:

```
iiii ooooo  
mmm lllll
```

Erste und zweite Spalte stehen nicht untereinander.

Aber mit Doppelblanks:



Druckbild:

```

iiii      ooooo
mmmm     lllll
    
```

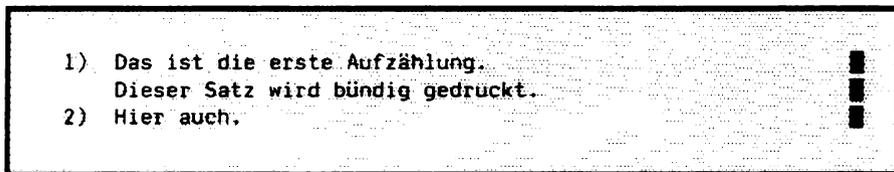
Erste und zweite Spalte stehen jetzt untereinander.

Das Doppelblank dient 'lineform'/'autoform' und dem Drucker als Zeichen, daß die Positionen speziell berechnet und beim Druck berücksichtigt werden müssen. Das gilt nur nach einer Absatzzeile. In seltenen Fällen (insbesondere beim Einsatz von Schrifttypen, die in der Größe stark voneinander abweichen) kann es vorkommen, daß diese Tabellenautomatik nicht funktioniert und Spalten übereinander gedruckt werden. In solchen Fällen müssen Sie die Anzahl der trennenden Doppelblanks erhöhen.

**Praktischer Tip:**

Beachten Sie, daß es für das Funktionieren der "Tabellenautomatik" erforderlich ist, daß jede Tabellenzeile eine Absatzzeile ist. Man sollte diese Zeilen vor dem Druck daraufhin überprüfen oder durch 'lineform'/'autoform' die Datei bearbeiten lassen. Sollten durch die zeilenweise Formatierung einmal wegen fehlender Absatzkennzeichnung zwei Zeilen zusammengezogen sein, können Sie diese leicht mit dem Editor wieder "auseinanderbrechen" ( **HOP** **RUBIN** , **CR** und **HOP** **RUBIN** ).

Ähnliches gilt bei Aufzählungen.



Druckbild:

- 1) Das ist die erste Aufzählung.  
Dieser Satz wird bündig gedruckt.
- 2) Hier auch.

Auch in solchen Fällen wird der gedruckte Text in der Regel richtig eingerückt. Die Tabellenautomatik wirkt nur nach einem Absatz. Hier aber ein Beispiel für eine typische Fehlersituation:

```
#type("normal")#  
1. Aufzählung  
2. Aufzählung  
3. Aufzählung  
  
#type("fett")#M1. Aufzählung
```



Die Einrückbreite wird durch den Schrifttyp bestimmt, der vor der Zeile herrscht, und den ganzen Absatz über beibehalten.

Druckbild:

1. Aufzählung
2. Aufzählung
3. Aufzählung

## **M1Aufzählung**

Das Blank zwischen 'M1.' und 'Aufzählung' reicht nicht aus, um eine Überschreibung zu verhindern. Diesen Fehler können Sie umgehen, indem Sie die #type#-Anweisung in eine gesonderte Zeile stellen. Richtig wäre folgendes (gewünschter Schrifttyp vor die Zeile!):

```
#type("trium8")#  
1. Aufzählung  
2. Aufzählung  
3. Aufzählung  
  
#type("triumbl4")#  
M1. Aufzählung
```



Druckbild:

- 1. Aufzählung
- 2. Aufzählung
- 3. Aufzählung

## M1. Aufzählung

Die genauen Regeln sind etwas kompliziert, so daß sie hier nicht einzeln aufgeführt werden (siehe S. 5–89 unter der Anweisung #block#). Treffen Sie auf einen der seltenen Fälle, wo die Tabellenautomatik nicht funktioniert, können Sie immer noch Tabellen-Anweisungen verwenden.

## 5.2.6.1. Tabellenanweisungen

*Mit den Tabellenanweisungen der Textkosmetik können Sie auf einfache Art Tabellen auch mit Proportionalschriften gestalten.*

Es ist sehr einfach, eine Tabelle in einer äquidistanten Schrift zu schreiben, denn hierbei stimmt das Schriftbild auf dem Terminal weitgehend mit dem späteren Druck überein. Bei einer äquidistanten Schrift ist jedes Zeichen gleich breit – Sie können also "sehen", an welcher Zeilenposition eine neue Spalte beginnt.

Etwas schwieriger sind Tabellen mit Proportionalschriften, da hier jedes Zeichen eine unterschiedliche Breite hat. Sie können somit einer Spaltenbreite nicht direkt "ansetzen", wie breit sie beim Druck wirklich wird. "Einfache" Tabellen können Sie mit dem Mehrfachblank gestalten (siehe S. 5–27). Bei komplizierteren Tabellen müssen Sie die folgenden Tabellenanweisungen benutzen.

Um eine Tabelle zu gestalten, gehen Sie folgendermaßen vor:

- Definieren Sie die Spaltenpositionen der Tabelle mit den folgenden Anweisungen. Für die Punkte bei den Anweisungen müssen Sie entsprechende Parameter einsetzen.

<pre>#l pos (...)# #r pos (...)# #c pos (...)# #d pos (... , ...)# #b pos (... , ...)# #fillchar (...)#</pre>	<pre>(* linksbündig *) (* rechtsbündig *) (* zentrierend *) (* zentrierend um eine Zeichenkette *) (* Blocksatz in einer Spalte *) (* Füllzeichen zwischen Spalten *)</pre>
---	---

Die Zentrierung um eine Zeichenkette ist wie folgt zu verstehen: Die Spalte wird bis zum Anfang der angegebenen Zeichenkette rechtsbündig und ab der Zeichenkette linksbündig geschrieben.

- Schreiben Sie dann die Tabelle. Sie muß von den Anweisungen

```
#table#
```

```
#table end#
```

eingefaßt werden. Die Spalten in der Tabelle müssen Sie durch mindestens zwei Leerzeichen voneinander trennen. Es müssen alle Spalten in einer Tabelle vorhanden sein. Soll einmal eine Spalte leer bleiben, müssen Sie für diese Spalte ein geschütztes Leerzeichen verwenden.

- Da die Spaltenpositionen erhalten bleiben (auch über die Anweisung #table end# hinweg), sollten Sie direkt hinter dem Tabellenende die #clear pos#-Anweisung geben.
- Dann können Sie 'lineform'/'autoform' vornehmen.

```
#r pos (2.2)##c pos (3.8)##l pos (5.8)##d pos (8.8, ".")#
#table#
erste Spalte  zweite Spalte  dritte Spalte  vierte Spalte
rechtsbündig  zentriert      linksbündig    dezi.mal
1234          1234           1234           12.34
12345         12345         12345         123.45
123456        123456        123456        1234.56
#table end# #clear pos#
```

Druckbild:

erste Spalte rechtsbündig	zweite Spalte zentriert	dritte Spalte linksbündig	vierte. Spalte dezi.mal
1234	1234	1234	12.34
12345	12345	12345	123.45
123456	123456	123456	1234.56

Solche Tabellen können Sie in `#head#`, `#bottom#` oder innerhalb von Fußnoten schreiben. Es ist jedoch nicht möglich, eine Fußnote innerhalb dieser Tabelle zu definieren. Ausweg: Tabelle um die Fußnote aufspalten.

## 5.2.6.2. Einstellen der Tabellenpositionen

*Mit den #pos#-Anweisungen können Sie eine bestimmte Position innerhalb der Tabelle einstellen, zugleich aber auch bestimmen, wie die Spalte gedruckt werden soll.*

```
#l pos (5.0)##r pos (10.0)##d pos (15.0, ".")#
```

Die Anweisung oben stellt die erste Spalte der Tabelle auf 5 cm vom Rand ein (linksbündig). Die zweite Spalte endet 10 cm vom Rand, wobei diese Spalte rechtsbündig geschrieben werden soll. Die dritte wird an die Position 15, zentriert um den Dezimalpunkt, gedruckt<sup>1)</sup>.

Beachten Sie, daß ein "Überlappen" von Spalten erfolgen kann (in unserem Beispiel kann die erste Spalte in die zweite hineinschreiben). 'lineform' bzw. 'autoform' meldet bei Spalten – Überschreibungen einen entsprechenden Fehler.

Für jede Spaltenposition nehmen Sie ein Element einer Zeile. Die Elemente müssen Sie beim Schreiben im Editor durch mindestens zwei Leerzeichen voneinander trennen. Auf die erste Spaltenposition wird das erste Element gedruckt, auf die zweite Position das zweite Element usw. Für das Drucken der Spalten wird der eingeschaltete Schrifttyp mit möglicherweise einer Modifikation genommen. Der Schrifttyp und die Modifikation können innerhalb der Tabelle geändert werden<sup>2)</sup>.

Beachten Sie, daß die Tabellenpositionen so lange erhalten bleiben, bis sie explizit gelöscht werden (#clear pos# – Anweisung, siehe S. 5 – 38).

<sup>1)</sup> Spaltenposition < 0.0 und Spaltenposition > 'eingestelltes limit' sind nicht erlaubt.

<sup>2)</sup> Die Zwischenräume zwischen den Spalten werden nicht modifiziert (also z.B. nicht unterstrichen).

### 5.2.6.3. Blocksatz innerhalb einer Spalte

Für Blocksatz innerhalb einer Spalte verwenden Sie die `#b pos (...)#`-Anweisung.

```
#l pos (0.0)##b pos (2.2, 8.0)##l pos (9.0)#
#table#
1. Spalte Die mittlere Spalte wird bis zur Druck- 3. Spalte
1. Spalte position '8.0' in Blocksatz gedruckt. Um 3. Spalte
1. Spalte in dieser Spalte einen Absatz zu bekom- 3. Spalte
1. Spalte men, muß ein geschütztes Leerzeichen am 3. Spalte
1. Spalte Ende der Spalte stehen. 3. Spalte
#table end# #clear pos#
```

Druckbild:

1. Spalte	Die mittlere Spalte wird bis zur Druck-	3. Spalte
1. Spalte	position '8.0' in Blocksatz gedruckt. Um	3. Spalte
1. Spalte	in dieser Spalte einen Absatz zu bekom-	3. Spalte
1. Spalte	men, muß ein geschütztes Leerzeichen am	3. Spalte
1. Spalte	Ende der Spalte stehen.	3. Spalte

## 5.2.6.4. Tabellenspalten auffüllen (Füllzeichen)

Mit der `#fillchar#` – Anweisung können Sie Spaltenzwischenräume füllen.

Angenommen, Sie möchten eine Rechnung erstellen. Die Warenposten sollen linksbündig an der Druckposition '0.0' und die Beträge rechtsbündig an der Position '9.0' gedruckt werden. Zwischen einem Warenposten und dem dazugehörigen Betrag sollen entsprechend viele Punkte ('.') gedruckt werden. Das folgende Druckbild:

```
30 Benutzerhandbücher..... 450, –DM
10 Systemhandbücher..... 150, –DM
```

wird mit

```
#l pos (0.0)##r pos (9.0)##fillchar(".")#
#table#
30 Benutzerhandbücher          450,–DM
10 Systemhandbücher           150,–DM

#table end##clear pos#
```

erreicht. Mit der Anweisung `#fillchar#` stellen Sie das/die Füllzeichen ein. Somit werden entsprechend viele Füllzeichen (anstatt der Leerzeichen) von dem Textende einer Spalte bis zu dem Textanfang der nächsten Spalte gedruckt. Die Füllzeichen bleiben so lange eingestellt, bis erneut die Anweisung `#fillchar#` gegeben wird. Insbesondere bleibt das Füllzeichen – genauso wie auch die eingestellten Spaltenpositionen – über das Tabellenende erhalten. Die Anweisung `#clear pos#` löscht – zusätzlich zu den Tabellenpositionen – auch das eingestellte Füllzeichen (setzt das Zeichen auf ' ' zurück).

Beachten Sie, daß die Füllzeichen direkt gedruckt werden (also ohne Leerzeichen zwischen dem Spaltentext und den Füllzeichen). Möchten Sie einen Zwischenraum zwischen dem Spaltentext und den Füllzeichen haben, dann fügen Sie ein geschütztes Leerzeichen an den Spaltentext an oder setzen eins vor die nachfolgende Spalte.

Die Anweisung `#fillchar#` gilt für Zwischenräume zwischen allen Spalten. Soll nur *ein* Spaltenzwischenraum ausgefüllt werden, müssen Sie die `#fillchar#`-Anweisung in der Tabelle entsprechend geben.

```
#l pos (1.0)##r pos (5.0)##r pos (10.0)#
#table#
1#fillchar(" ")#    3#fillchar(" ")#    4
2#fillchar(" ")#    17#fillchar(" ")#    6
#table end#
```

Druckbild:

```
1.....3          4
2.....17         6
```

Eingeschaltete Modifikationen gelten in diesem Fall auch für die Spaltenzwischenräume.

## 5.2.6.5. Tabellenpositionen löschen

*Mit der #clear pos# – Anweisung löschen Sie alle eingestellten Positionen.*

Sollen gänzlich neue Positionen eingestellt werden, benutzen Sie die Anweisung

```
#clear pos#
```

ohne Parameter. Sie löscht alle eingestellten Tabellenpositionen. Beachten Sie, daß #clear pos# auch das Füllzeichen für die #fillchar# – Anweisung löscht (es wird wieder ein ' ' voreingestellt). Eine einzelne Tabellenposition können Sie z.B. mit

```
#clear pos (10.0)#
```

löschen.

## 5.2.7. Indizes und Exponenten

Mit den Anweisungen #u#, #d# und #e# können Sie Exponenten und Indizes schreiben.

Die Anweisung #u# (steht für 'up') schaltet auf eine Exponenten-Schreibweise um und zwar so lange, bis die Anweisung #e# (steht für 'end') angetroffen wird. Dabei wird automatisch auf den nächst kleineren Schrifttyp umgeschaltet (sofern vorhanden).

a#u#i,k#e#

Druckbild:

$a^{i,k}$

Die #d#-Anweisung ('d' steht für 'down') ist für Indizes gedacht und arbeitet analog zur #u#-Anweisung.

a#d#i,k#e#

Druckbild:

$a_{i,k}$

Die automatische Umschaltung auf den nächst kleineren Schrifttyp erfolgt nur, wenn in der Fonttabelle ein nächst kleinerer Schrifttyp angegeben ist. Sonst wird der eingestellte Schrifttyp für den Exponenten beibehalten.

Nach der `#e#`-Anweisung wird automatisch wieder der Schrifttyp eingestellt, der vor der zugehörigen `#u#`-Anweisung galt. Die `#u#`- und `#e#`-Anweisungen bilden also Klammern. Innerhalb einer Anweisung kann jede beliebige, sinnvolle Textkosmetik-Anweisung stehen. Beachten Sie, daß Anweisungen innerhalb einer Klammer die Zeilenhöhe nicht verändern sollen. Wenn Sie beispielsweise eine `#type#`-Anweisung in eine Klammer schreiben, wird zwar der Index/Exponent in diesem Schrifttyp gedruckt, aber der Drucker geht davon aus, daß die Zeilenhöhe nicht überschritten wird. Deshalb ist es angeraten, nur einen kleineren Schrifttyp innerhalb eines Index/Exponenten zu verwenden. Wie bereits erwähnt, wird auch in diesem Beispiel nach dem Klammerende auf den vorher eingestellten Schrifttyp zurückgestellt.

Die Index/Exponenten – Klammern können auch geschachtelt werden.

`a#u#um 1 hoch#u#noch 1 hoch#e#um 1 zurück#e# Grundlinie`

Druckbild:

a<sup>um 1 hoch</sup><sup>noch 1 hoch</sup>um 1 zurück Grundlinie

Es gelten folgende **Einschränkungen**:

1. Ein Exponent (Index) wird so positioniert, daß es in der Regel keine Überschreibung mit der vorhergehenden (nachfolgenden) Zeile gibt.
2. Bei mehrfachen Exponenten oder Indizes oder bei Umschaltung auf einen anderen Schrifttyp innerhalb eines Exponenten (Index) oder wenn nicht auf einen kleineren Schrifttyp umgeschaltet werden kann, besteht die Möglichkeit, daß der Exponent oder Index über die "normale" Zeile hinausragt. In diesem Fall kann es Überschreibungen geben, die Sie mit der `#linefeed#`-Anweisung ausgleichen können.
3. Eine Exponenten- oder Index-Klammer muß als Ganzes auf einer Zeile stehen.

4. Gleichzeitige Exponenten- und Index-Ausdrücke, die übereinander stehen sollen, sind zur Zeit mit den `#u##d#`-Anweisungen nicht möglich. Jedoch funktioniert folgendes:

`a#u#Exponent#d#Index des Exponenten#e##e#`

Druckbild:

$a^{\text{Exponent}}_{\text{Index des Exponenten}}$

5. Doppelblanks spielen innerhalb einer solchen Klammer keine Rolle, wirken also wie zwei "normale" Leerzeichen und nicht als implizite Positionierung. Innerhalb einer solchen Klammer werden Blanks, sofern die Anweisung `#block#` gegeben wurde, nicht verbreitert.
6. Indizes oder Exponenten sollten nicht mit den Modifikationen `#underline#` und/oder `#reverse#` zusammen verwendet werden, da z.B. ein Unterstreichen von Indizes und Exponenten innerhalb einer unterstrichenen Zeile zu einem solchen Ergebnis führt:

Druckbild:

Indizes und Exponenten     $a_{j,k}$      $a^{i,k}$     sollten nicht unterstrichen werden!

## 5.3. Pageform

### 5.3.1. Seitenweise formatieren

*'pageform'/'autopageform' formatiert eine Datei seitenweise und erledigt Routinearbeiten wie die Platzierung von Fußnoten, Seitennumerierung usw.*

Das Programm 'pageform' können Sie mit dem Kommando

```
gib kommando:  
pageform ("dateiname")
```

aufrufen. 'pageform' erzeugt aus der Eingabedatei (z.B.: "dateiname") eine Druckdatei, deren Name durch ein angehängtes '.p' gebildet wird (z.B.: "dateiname.p").

Die von 'pageform' erzeugte Druckdatei besteht aus der Eingabedatei mit ggf. neu eingefügten Zeilen. Die eingesetzten Zeilen stammen aus #head#-, #bottom#- oder #foot#-Anweisungen. Dadurch erhöht sich die Zeilenanzahl der Datei.

Sie können in Kopf- oder Fußzeilen Seitennummern aufnehmen. Diese Seitennummern werden von 'pageform'/'autopageform' bei Seitenwechseln automatisch erhöht und an eine von Ihnen gekennzeichnete Stelle eingesetzt. Fußnoten können auch durch Nummern gekennzeichnet werden. Querverweise sind ebenfalls möglich.

Nachdem 'pageform' eventuelle Kopf-, Fuß- und Fußnotenzeilen eingefügt hat, berechnet es die Anzahl von Zeilen, die auf eine Seite passen, aus den Angaben für Seitenlänge und Zeilenvorschub und aus der Höhe der eingestellten Schrifttypen (#type# -Anweisung). Dann zeigt 'pageform' das errechnete Seitenende auf dem Bildschirm an. Das Seitenende kann interaktiv verschoben werden, um es an eine gewünschte Stelle zu plazieren und es können Leerzeilen eingefügt/gelöscht werden, um Seiten gleich lang zu machen. Zusätzlich können Sie Seiten in Spalten ("Zeitungsdruck") aufteilen und diese interaktiv formatieren.

Bei mehreren Schrifttypen innerhalb einer Zeile wird als Zeilenhöhe automatisch die des größten Schrifttyps genommen. Dabei müssen Sie bedenken, daß zu Beginn der Zeile immer der Schrifttyp der vorherigen Zeile eingeschaltet ist.

### 5.3.1.1. Automatische Seitenformatierung

*'autopageform' arbeitet wie 'pageform', jedoch werden die Seitenenden automatisch plziert.*

'autopageform' sucht zuerst das rechnerische Seitenende. Ist dort ein Absatz vorhanden, wird die Seite an dieser Stelle beendet. Falls nicht, sucht 'autopageform' nach oben in den nächsten vier Zeilen nach einem Absatz. Wird keiner gefunden, wird die Seite am rechnerischen Seitenende beendet.

Ist die #pageblock# – Anweisung gegeben, wird zuerst nach oben in den vier letzten Zeilen nach einem Absatz gesucht, um dort die Seite zu beenden. Ist dort keiner vorhanden, wird auch über das rechnerische Seitenende hinweg versucht, die Seitenlänge zu plazieren (4 Zeilen). 'autopageform' beachtet in einem solchen Fall die 'pagelength' – Anweisung, indem der Zeilenabstand gestaucht wird.

### 5.3.1.2. Seitenende interaktiv verschieben

*In diesem Abschnitt wird beschrieben, welche interaktiven Möglichkeiten Ihnen 'pageform' bietet, Seiten zu gestalten.*

Auf dem Bildschirm wird das von 'pageform' errechnete jeweilige Seitenende unter Angabe der aktuellen Seitennummer angezeigt. Das Seitenende erscheint ungefähr in der Mitte des Bildschirms und wird durch eine von 'pageform' erzeugte Zeile gekennzeichnet, die auch – nach erfolgter Seitenformatierung – in der Druckdatei zu sehen ist. Der EUMEL-Drucker druckt diese Zeile nicht.

Mehrere Fußnoten innerhalb einer Seite werden von 'pageform'/'autopageform' in der Reihenfolge ihres Auftretens gesammelt und am Ende der Seite plaziert. Für eine entsprechende Trennung der Fußnoten voneinander (z.B. durch Leerzeilen) müssen Sie selbst sorgen.

**#page## - - - - - Ende Seite 215 - - - - - #**

Unter Umständen paßt die Fußnote nicht mehr auf die aktuelle Seite und muß deshalb von 'pageform'/'autopageform' auf die nächste Seite gebracht werden. 'pageform'/'autopageform' geht davon aus, daß die Kennzeichnung der Fußnote in der Zeile unmittelbar vor der Fußnote steht und bringt diese Zeile ebenfalls auf die neue Seite.

Über der Markierung erscheinen die letzten Zeilen der bereits verarbeiteten Seite, darunter die ersten Zeilen der nächsten Seite. Sie können nun mit Hilfe der Positionierungstasten die Markierung und damit das Seitenende nach oben verschieben. Damit vermeiden Sie, daß ein logisch zusammengehöriger Text auseinandergerissen wird und sogenannte "Waisenkinder" entstehen (letzte Zeile eines Abschnittes kommt auf die neue Seite).

Bei der interaktiven Formatierung können Sie die Markierung nicht über das errechnete Ende einer Seite nach unten oder über das vorherige, bereits verarbeitete Seitenende nach oben verschieben.

Haben Sie jedoch zu Beginn die #pageblock# – Anweisung (siehe S. 5 – 91) gegeben, ist es erlaubt, die Seitenende – Markierung auch einige Zeilen über das rechnerische Seitenende hinaus zu bewegen. Betätigen Sie dann CR , wird der Drucker (sofern möglich) den Zeilenabstand auf dieser Seite stauchen. In diesem Fall sollten Sie darauf achten, daß das Seitenende bei einem Absatz immer vor eventuell vorhandene Leerzeilen plaziert wird. Andernfalls werden die Leerzeilen am Ende der Seite als Textzeile mitgezählt und es bleibt entsprechender Platz frei!

Innerhalb einer Fußnote kann die Markierung nicht verschoben werden. In diesem Fall wird interaktiv angefragt, ob die Fußnote auf der nächsten Seite fortgesetzt werden soll. Verneinen Sie die Anfrage, positioniert 'pageform' vor die Fußnote. Von dieser Stelle aus können Sie das Seitenende wie gewohnt verschieben.

Bejahen Sie dagegen die Anfrage nach dem Fußnotenumbruch, plaziert 'pageform' das Seitende an dieser Stelle innerhalb der Fußnote. Der restliche Teil der Fußnote kommt auf die nächste Seite mit einer Anmerkung ('Forts. von letzter Seite')<sup>1)</sup>.

Entstehen bei der Seitenformatierung am Anfang einer Seite Leerzeilen (z.B. durch Plazierung des Seitenendes zwischen zwei Absätzen), so werden diese von 'pageform' automatisch aus der Druckdatei entfernt. Möchten Sie Leerzeilen am Anfang einer Seite, dann sollten Sie die #free# – Anweisung in Verbindung mit der #page# – Anweisung verwenden.

Zusätzlich können Sie Leerzeilen in eine Seite der Druckdatei einfügen und/oder beliebige Zeilen löschen (vergl. b).

---

<sup>1)</sup> Bei fremdsprachlichen Texten sollten Sie nach 'pageform' diese Anmerkungen in der '.p' – Datei ändern.

Folgende Operationen stehen Ihnen bei der interaktiven Seitenformatierung zur Verfügung:

**a) Seitenende verschieben:**

'pageform' berechnet das "rechnerische" Seitenende und zeigt dieses auf dem Bildschirm durch die Markierung an. Die Markierung kann interaktiv verschoben werden:

<b>Taste</b>	<b>Bedeutung</b>
	Seitenende an diese Stelle plazieren.
	Seitenende eine Zeile nach oben verschieben.
	Seitenende eine Zeile nach unten verschieben (wenn vorher nach oben verschoben bzw. wenn #pageblock#-Anweisung gegeben ist).
	Seitenende um einen Bildschirm nach oben verschieben.
	Seitenende um einen Bildschirm nach unten verschieben.
	Abbruch der Seitenformatierung.

**b) Leerzeilen einfügen und/oder Zeilen löschen**

Ist nach den Berechnungen von 'pageform' der Text ungünstig auf der Seite plziert, können Sie in die Seite (der Druckdatei!) Leerzeilen einfügen und/oder Zeilen löschen. Dies kann beispielsweise sinnvoll sein, wenn durch die Löschung einer Zeile ein Absatz noch auf die Seite passen würde oder durch die Einfügung von Leerzeilen ein Absatz auf der letzten Zeile der Seite endet. Oft ist es auch sinnvoll, daß alle Seiten gleich lang sind. In diesem Fall sollten vor Kapiteln und Absätzen Leerzeilen eingefügt oder gelöscht werden.

Um Leerzeilen einzufügen und/oder Zeilen zu löschen, müssen Sie die Markierung wie unter a) beschrieben an die Stelle plazieren, an der die Änderung vorgenommen werden soll.

Taste	Bedeutung
<div style="display: inline-block; border: 1px solid black; padding: 2px 5px; margin-right: 10px;">HOP</div> <div style="display: inline-block; border: 1px solid black; padding: 2px 5px;">RUBIN</div>	Leerzeilen einfügen. Anstatt der Markierung können durch (u.U. mehrmaliges) <div style="border: 1px solid black; padding: 2px 5px; display: inline-block;">CR</div> Leerzeilen eingefügt werden. <div style="border: 1px solid black; padding: 2px 5px; display: inline-block;">HOP</div> <div style="border: 1px solid black; padding: 2px 5px; display: inline-block;">RUBIN</div> beendet den Vorgang (wie Zeileneinfügen im Editor).
<div style="display: inline-block; border: 1px solid black; padding: 2px 5px; margin-right: 10px;">HOP</div> <div style="display: inline-block; border: 1px solid black; padding: 2px 5px;">RUBOUT</div>	Zeile löschen. Die Zeile unmittelbar oberhalb der Markierung wird gelöscht.

Anschließend berechnet 'pageform' die Seite erneut.

#### c) #page# – Anweisung bestätigen/löschen

Wird von der Prozedur 'pageform' eine #page# – Anweisung angetroffen, so wird das gewünschte Seitenende auf Ihrem Bildschirm angezeigt. Die #page# – Anweisung können Sie entweder bestätigen oder löschen.

Taste	Bedeutung
<div style="border: 1px solid black; padding: 2px 5px; display: inline-block;">CR</div>	Seitenende bestätigen.
<div style="border: 1px solid black; padding: 2px 5px; display: inline-block;">RUBOUT</div>	#page# – Anweisung ignorieren. Die Prozedur 'pageform' bearbeitet in diesem Fall die Datei weiter, als ob keine #page# – Anweisung angetroffen wurde.
<div style="border: 1px solid black; padding: 2px 5px; display: inline-block;">ESC</div>	Abbruch der Seitenformatierung.

## 5.3.2. Seitenlänge einstellen

*'pageform'/'autopageform' ist auf ein Schreibfeld von 25.0 cm eingestellt (entspricht einem DIN A4-Schreibfeld). Wünschen Sie eine andere Seitenlänge, müssen Sie die #pagelength#-Anweisung in den Text einfügen.*

```
#pagelength (20.0)#
```

stellt die Seitenlänge auf 20 cm ein.

Beachten Sie, daß

1. die neu eingestellte Seitenlänge immer erst ab der nächsten Seite gilt (die bislang eingestellte Seitenlänge gilt noch für die aktuelle Seite).
2. die eingestellte Seitenlänge am Anfang der Datei (also vor der ersten Textzeile) für die erste Seite gilt.
3. der Dezimalpunkt bei der Seitenlänge mit angegeben werden muß.

Die folgende Tabelle gibt die Seitenlänge für die am häufigsten gewählten Papiergrößen an:

Format	Seitenlänge (in cm)	oberer und unterer Rand
DIN A4	25.0	je 2.35 cm
DIN A5	18.0	je 2.15 cm
DIN A4 quer	16.0	je 2.50 cm

### 5.3.3. Zeilenabstand einstellen

Mit der *#linefeed#*-Anweisung stellen Sie einen Zeilenvorschub relativ zu der *Schriftgröße des eingestellten Schrifttyps* ein.

'pageform'/'autopageform' berechnet die Anzahl der Zeilen pro Seite immer in Abhängigkeit von dem eingestellten Schrifttyp. Haben Sie z.B. eine Schrift gewählt, die doppelt so hoch wie eine Schreibmaschinenschrift ist, bekommen Sie auch entsprechend weniger Zeilen auf eine Seite. Um diesen Berechnungsvorgang brauchen Sie sich in der Regel nicht zu kümmern.

Anders verhält es sich, wenn ein anderer Zeilenabstand als der "normale" Abstand zwischen Zeilen eingestellt werden soll. In diesem Fall wird die *#linefeed#*-Anweisung eingesetzt. Der Parameter gibt an, um welchen Faktor eine *Zeilenhöhe ab der nächsten druckbaren Zeile* erhöht oder verringert werden soll.

```
#linefeed (2.0)#
```

druckt die folgenden Zeilen mit doppeltem Zeilenabstand. Nach Antreffen dieser Anweisung wird die Zeilenhöhe durch  $2 \cdot$  eingestellte Schrifttypgröße errechnet. Es wird also der Zeilenabstand zwischen den Zeilen entsprechend vergrößert, da die Schriftgröße gleich bleibt. Dies entspricht dem zweizeiligen Schreiben bei einer Schreibmaschine (wenn man davon absieht, daß auch hier unterschiedliche Schriftgrößen möglich sind). Ein  $1 \frac{1}{2}$  zeiliges Schreiben wäre mit

```
#linefeed (1.5)#
```

einzustellen.

**#linefeed (0.5)#**

stellt die Zeilenhöhe =  $1/2 * \text{eingestellte Schrifthöhe}$  ein, so daß die Zeilen teilweise ineinander gedruckt werden (was bei manchen Druckern zu nicht lesbaren Resultaten führt). Bei #linefeed (0.0)# werden Zeilen übereinander gedruckt (druckerabhängig).

Beachten Sie, daß die Angabe in der #linefeed# – Anweisung relativ erfolgt. Bei allen anderen Anweisungen der Textkosmetik werden Angaben in Zentimetern verlangt. Die #linefeed# – Anweisung bildet somit eine Ausnahme.

## 5.3.4. Platz freigehalten

Mit der *#free#* – Anweisung können Sie einen zusammenhängenden Teil auf einer Seite freigehalten.

Die *#free#* – Anweisung setzen Sie an solchen Stellen im Text ein, an denen – nach dem Druck – Zeichnungen, Tabellen und ähnliches eingeklebt werden sollen. Sie können sie auch zwischen Absätzen, Kapiteln usw. einsetzen, wenn der Abstand nicht gleich dem Vielfachen der Zeilenhöhe ist. Es wird der in der *#free#* – Anweisung angegebene Platz freigehalten.

```
#free (2.0)#
```

hält zwei Zentimeter frei. Paßt der angeforderte Platz nicht mehr auf die Seite, so wird er auf der nächsten Seite reserviert ('pageform'/'autopageform' platziert das Seitenende vor die *#free#* – Anweisung).

### **Praktischer Tip:**

Sie sollten eine *#free#* – Anweisung allein auf eine Zeile schreiben, damit Sie sie u.U. durch 'pageform' interaktiv entfernen können, wenn die *#free#* – Anweisung ungünstig an den Seitenanfang oder das Seitenende kommt.

### 5.3.5. Neue Seite beginnen

*An einigen Stellen im Text, z.B. zu Beginn eines neuen Kapitels, möchten Sie unbedingt eine neue Seite anfangen. Dies erreichen Sie mit der #page#-Anweisung.*

'pageform' meldet in diesem Fall, nach wie vielen Zentimetern auf der Seite die Anweisung angetroffen wurde. Sie können nun mit **CR** das Seitenende bestätigen oder die Anweisung (in der Druckdatei) löschen. Im letzteren Fall berechnet 'pageform' die Seite neu, als ob die #page#-Anweisung nicht dagewesen wäre.

Gleichzeitig können Sie mit Hilfe der #page#-Anweisung eine neue Seitennummer für die neue Seite einstellen (vergl. Sie dazu die nächsten Abschnitte).

### 5.3.6. Kopf – und Fußzeilen

*Mit den #head# – und #bottom# – Anweisungen können Sie Zeilen am Anfang und Ende jeder Seite einfügen.*

Sie schreiben Zeilen am Anfang ("Kopfzeilen") und Ende ("Fußzeilen") jeder Seite nur einmal und kennzeichnen sie mit Anweisungen. Diese Zeilen fügen 'pageform'/'-autopageform' dann an den entsprechenden Stellen ein.

```
#i.cad#
```

```
    Unser EUMEL-Benutzerhandbuch
```

```
#end#
```



Diese Zeile (also die zwischen den #head# – und #end# – Anweisungen eingeschlossene Zeile) wird von 'pageform'/'-autopageform' an den Anfang jeder Seite in die Druckdatei plaziert.

Entsprechendes gilt für Fußzeilen, die zwischen #bottom# und #end# eingeschlossen werden müssen:

```
#bottom#
```

```
    Autor: I. Listig
```

```
#end#
```



**Praktischer Tip:**

Fügen Sie mindestens eine Leerzeile am Ende eines `#head#` bzw. am Anfang eines `#bottom#` ein, um den eigentlichen Text von den Kopf- bzw. Fußzeilen abzuheben.

'pageform'/'autopageform' zählt die Seiten, beginnend mit der Seitennummer '1'. (Wie man Seitennummern in die Kopf- und Fußzeilen bekommt, erfahren Sie im nächsten Abschnitt). Sie können nun getrennte Kopf- und Fußzeilen für gerade und ungerade Seiten gestalten (wie in diesem Benutzerhandbuch). Dies erfolgt mit den Anweisungen `#headeven#` und `#headodd#` für Seiten mit geraden und ungeraden Seitennummern; ebenso `#bottomeven#` und `#bottomodd#`. Diese Anweisungen müssen ebenfalls jeweils mit einer `#end#`-Anweisung beendet werden.

Sie haben die Möglichkeit, Kopf- und Fußzeilen mehrmals innerhalb einer Datei zu wechseln, um unterschiedliche Beschriftungen zu erhalten (z.B. kapitelweise). Dies ist jedoch nur sinnvoll, wenn es auf einer neuen Seite erfolgt, also unmittelbar *nach* einer `#page#`-Anweisung.

```
#page#
#head#
      Neuer Seitenkopf
#end#
```

Kopf- und Fußzeilen sollen überall gleiches Aussehen haben, unabhängig davon, welche Anweisungen im restlichen Text gegeben werden. Darum werden die bei der Definition einer Kopf- und Fußzeile aktuellen Werte für

```
limit
type
linefeed
```

bei dem Einsetzen der Zeilen berücksichtigt. Für Kopf- oder Fußzeilen können Sie einen anderen Schrifttyp als im restlichen Text verwenden, indem Sie die `#type#`-Anweisung innerhalb eines `#head#`- oder `#bottom#`-Bereiches geben. Beachten Sie, daß nach `#head#`-, `#bottom#` und auch `#foot#`-Bereichen die oben genannten Anweisungen nicht automatisch zurückgestellt werden. Darum sollten Sie vor der `#end#`-Anweisung wieder auf die im übrigen Text verwendeten Werte zurückstellen.

```
#bottom#  
#type ("klein")#  
    Autor: I. Listig  
  
#type ("normal")#  
#end#
```



(Schrifttyp  
zurückstellen):

## 5.3.7. Seiten numerieren

*In den Kopf- und Fußzeilen steht das '%' - Zeichen für die aktuelle Seitennummer.*

Erscheint das '%' - Zeichen innerhalb eines Kopf- oder Fußbereiches, wird von 'pageform'/'autopageform' beim Einsetzen dieser Zeilen auf jeder Seite die aktuelle Seitennummer eingesetzt (sind mehrere '%' - Zeichen vorhanden, wird die Seitennummer mehrmals eingesetzt).

#head#

Seite: - % -

#end#



Wenn Sie die Seitenzahl in der Zeilenmitte oder am rechten Rand plazieren möchten, können Sie die Anweisungen #center# (siehe S. 5 - 93) oder #right# (siehe S. 5 - 94) verwenden.

Durch das Einrichten eines Fußbereiches können Sie die Seitennummern auch am unteren Ende einer Seite erzeugen. Beachten Sie, daß sich bei mehrstelligen Seitennummern die Zeilenlänge durch das Einsetzen vergrößert.

Um zum Beispiel das Vorhandensein einer Folgeseite in einem Fußbereich zu kennzeichnen, müssen Sie das '%' - Zeichen zweimal direkt hintereinander schreiben.

```
#bottom#
```

```
#right# %%
```

```
#end#
```



In dem Beispiel oben wird die Seitenzahl rechtsbündig gedruckt.

Manchmal ist es notwendig und sinnvoll, einen Text in mehreren Dateien zu halten. Bei einer Folgedatei müssen Sie die Seitennummer dann neu setzen. Das erfolgt mit der #pagenr# – oder der #page# – Anweisung.

```
#page (4)#
```

bewirkt eine neue Seite. Die Seitennummer der neuen Seite ist '4'.

Bei einigen Spezialanwendungen benötigen Sie unter Umständen mehr als eine Seitennummer. Beispielsweise soll ein Text nicht nur absolut, sondern auch jede Seite in jedem Kapitel separat durchgezählt werden.

```
#page (4711)#
```

```
#pagenr ("§", 1)#
```

```
#head#
```

```
Mein Buch
```

```
Seite: % Kapitelseite: $
```

```
#end#
```



Die Anweisung `#pagenr ("$$",1)#` veranlaßt, daß ab der nächsten Seite eine neue Numerierung durchgeführt wird. Dabei steht '\$' stellvertretend für die neue Zahl. Die '1' bedeutet, daß bei der Numerierung mit '1' begonnen wird. 'pageform'/'-autopageform' erhöht bei jeder neuen Seite das Zeichen um '1' und setzt es ggf. in die Kopf- und Fußzeilen. Es sind zwei zusätzliche Seitenzeichen (neben dem '%') möglich.

Beachten Sie, daß die neuen Seitennummern immer erst ab der nächsten Seite gelten. Geben Sie die `#page (...)#` – oder die `#pagenr (.....)#` – Anweisung am Anfang der Datei (also vor der ersten Textzeile), gelten die neuen Seitennummern für die erste Seite.

## 5.3.8. Fußnoten schreiben

*Fußnoten werden direkt im Text durch die Anweisungen #foot# und #end# gekennzeichnet. Die Fußnoten plaziert 'pageform'/'autopageform' an das Ende einer Seite.*

Fußnoten schreiben Sie direkt in den Text, am besten an der Stelle, an der später die Fußnote aufgerufen werden soll. Die Fußnote wird von 'pageform'/'autopageform' an das Ende einer Seite, ggf. vor die Fußzeilen, plaziert. Für die Kennzeichnung von Fußnoten und die entsprechende Markierung im Text sind Sie selbst zuständig. Allerdings werden von 'pageform'/'autopageform' bei dem Einsetzen einer Fußnote am Ende einer Seite Unterstriche vor die Fußnoten eingefügt, damit Fußnoten vom laufenden Text abgehoben werden.

```
#foot#
```

```
*) Das ist die erste Anmerkung auf dieser Seite.
```

```
#end#
```

Druckbild:

---

\*) Das ist die erste Anmerkung auf dieser Seite.

Mehrere Fußnoten innerhalb einer Seite werden von 'pageform'/'autopageform' in der Reihenfolge ihres Auftretens gesammelt und am Ende der Seite plaziert. Für eine entsprechende Trennung der Fußnoten voneinander (z.B. durch Leerzeilen) müssen Sie selbst sorgen.

Unter Umständen paßt die Fußnote nicht mehr auf die aktuelle Seite und muß deshalb von 'pageform'/'autopageform' auf die nächste Seite gebracht werden. 'pageform'/'autopageform' geht davon aus, daß die Kennzeichnung der Fußnote in der Zeile unmittelbar vor der Fußnote steht und bringt diese Zeile ebenfalls auf die neue Seite.

Es ist auch möglich, eine Fußnote innerhalb eines Abschnitts zu schreiben, wie z.B. in dieser Zeile#u##)#e#.#foot#  
#u##)#e# Fußnote in einem Abschnitt!  
#end#  
Sie fahren anschließend ohne Unterbrechung mit dem Schreiben Ihres Textes fort.

Druckbild (nach lineform):

Es ist auch möglich, eine Fußnote innerhalb eines Abschnitts zu schreiben, wie z.B. in dieser Zeile<sup>7)</sup>. Sie fahren anschließend ohne Unterbrechung mit dem Schreiben Ihres Textes fort.

In diesem Fall ist es wünschenswert, daß 'lineform' die Zeile, die #foot# vorausgeht, mit der Zeile, die #end# folgt, auffüllt. Dies geschieht unter folgenden Bedingungen:

1. Hinter #foot# darf nichts mehr stehen, also auch kein Absatzzeichen.
2. Es werden so lange Worte von der Zeile nach #end# vor die #foot#-Anweisung plaziert, bis die Zeile gefüllt oder die Zeile nach #end# leerräumt ist.
3. Beachten Sie, daß Textkosmetik-Anweisungen ebenfalls mit über die Fußnote genommen werden. Handelt es sich beispielsweise um eine #type#-Anweisung, kann sich das Aussehen der Fußnote verändern! Darum ist es angeraten, eventuelle Anweisungen, die die Fußnote verändern sollen, innerhalb der Fußnote zu plazieren.

<sup>7)</sup> Fußnote in einem Abschnitt!

Sie sollten vermeiden, umfangreiche Texte in Fußnoten zu schreiben (beispielsweise längere Zitate). Aus programmtechnischen Gründen begrenzt 'pageform'/'autopageform' die maximale Länge von Fußnoten auf einer Seite auf 85% des effektiven Schreibfeldes (effektives Schreibfeld: Seitenlänge minus Länge von #head# – bzw. #bottom# – Zeilen). Nimmt eine Fußnote einen größeren Raum ein, bricht 'pageform'/'autopageform' die Seitenformatierung mit einer Fehlermeldung ab.

### 5.3.8.1. Fußnoten numerieren

*Gleichartige Textteile wie Lehrsätze, Beispiele, Fußnoten usw. werden i. allg. durchnummeriert. Da Sie bei der Abfassung eines längeren Textes ihre genaue Anzahl meist nicht vorausplanen können, übernimmt 'pageform'/'autopageform' die Zählung.*

Durch die #count#-Anweisung wird 'pageform'/'autopageform' veranlaßt, einen internen Zähler (beginnend bei dem Wert 0) zu erhöhen und diesen Wert statt der #count#-Anweisungen in den Text einzusetzen.

#count#

setzt den Wert 1 statt der Anweisung ein. Jede weitere #count#-Anweisung erhöht den internen Zähler und der Zählerwert wird wiederum eingesetzt:

#count#

setzt den Wert 2 ein usw. Dadurch ist es möglich, beliebige Textteile (Kapitel, mathematische Sätze u.a.m.) fortlaufend zu nummerieren, ohne auf die Numerierung beim Schreiben und Ändern des Textes zu achten.

**Anmerkung:**

Trifft 'lineform' auf eine #count#-Anweisung, so wird die Zeile berechnet, als ob drei Ziffern anstatt der Anweisung im Text ständen.

Mit der #value# – Anweisung können Sie den **letzten** erreichten count – Wert nochmals einsetzen. Das ist insbesondere für Fußnoten sinnvoll einsetzbar.

```
Text ..... (#count#)
#foot#
(#value#) Text der Fußnote
#end#
Text .....
```

Das Resultat sähe folgendermaßen aus:

```
Text ..... (3)
Text .....
.....
_____
(3) Text der Fußnote
```

Beachten Sie, daß in diesem Fall die #value# – Anweisung der #count# – Anweisung folgen muß, ohne daß eine weitere #count# – Anweisung dazwischen steht. Das liegt – wie bereits oben erwähnt – daran, daß die #value# – Anweisung immer den letzten #count# – Wert einsetzt.

Das können Sie umgehen, indem Sie die #count# – und #value# – Anweisungen mit einem TEXT – Parameter versehen, der als Kennzeichnung dient.

```
...count ("Merk1")#
```

#count ("Merk1")# arbeitet ebenso wie #count# ohne Parameter und setzt für unser Kapitel hier den Wert 4 ein. Zusätzlich zu dem fortlaufend gezählten Wert (fortlaufende Numerierung der Fußnoten) vermerkt 'pageform'/'autopageform' einen Wert, der bei Bedarf an irgendeiner anderen Stelle im Text durch #value ("Merk1")# wieder aufgerufen werden kann, zum Beispiel, wenn Sie auf eine andere Fußnote verweisen möchten.

```
#count##count#  
#value("Merk1")#
```

Die ersten zwei #count#-Anweisungen produzieren – in unserem Kapitel – die Werte 5 bzw. 6. Die #value#-Anweisung dagegen setzt den vermerkten Wert 4 ein.

Dies ist insbesondere sinnvoll, wenn Sie im Text auf eine Fußnote verweisen möchten.

Beispiel:

Sie schreiben einen mehrseitigen Prospekt über ein neues Produkt. Auf Seite 5 möchten Sie auf eine Fußnote verweisen, die auf einer anderen Seite steht. Dann fügen Sie 'siehe auch Anmerkung (#value("Liefertermin"))#' in Ihren Text ein und fahren mit dem Schreiben fort. 'pageform'/'autopageform' setzt später die entsprechende Zahl für den Verweis ein.

Auf der Seite, auf die Sie Bezug nehmen, sieht das ganze folgendermaßen aus:

```
Der Textverarbeitungskurs ist ein Lernprogramm für Anfänger.  
#(count)("Liefertermin")#  
#foot#  
#(value)("Liefertermin")#  
Der Textverarbeitungskurs wird ab August erhältlich sein.  
#end#  
Das Programm ist auf den neuesten Erkenntnissen der Lehr-  
forschung aufgebaut. Der Kurs umfaßt Lehrbuch, Arbeitsbuch und  
sechs Kassetten.
```

Soll die Zahl für den Verweis bzw. für die Fußnote hochgestellt werden, fügen Sie die Anweisungen #u# und #e# hinzu.

```
#u# (#value("Liefertermin"))#e#
```

Im gedruckten Prospekt sähe es (nach 'lineform') wie folgt aus:

Der Textverarbeitungskurs ist ein Lernprogramm für Anfänger<sup>1)</sup>.  
Das Programm ist auf den neuesten Erkenntnissen der Lehrforschung aufgebaut. Der Kurs umfaßt Lehrbuch, Arbeitsbuch und sechs Kassetten.

Manchmal ist es notwendig (ebenso wie bei der Seitennummer), den internen Zähler neu zu setzen.

```
#setcount (13)##count#
```

produziert den Wert 13.

---

<sup>1)</sup>Der Textverarbeitungskurs wird ab August erhältlich sein.

### 5.3.9. Querverweise

*Mit den Anweisungen #topage# und #goalpage# sind Querverweise möglich, die von 'pageform'/'autopageform' in die Druckdatei eingefügt werden.*

Mit Hilfe von Querverweisen soll auf andere Stellen im Text verwiesen werden, was nur bei längeren Texten üblich ist. Um dem Leser die mühselige Suche nach der Textstelle zu ersparen, gibt man in der Regel die Seitennummer an. Normalerweise steht die Seitennummer vor der Fertigstellung des Textes noch nicht fest. Auch in diesem Fall kann 'pageform'/'autopageform' helfen. Die #topage# – Anweisung verweist auf eine andere Seite im Text, an der sich eine Anweisung #goalpage# befinden muß. Statt der Anweisung #topage# wird die Seitennummer der Seite eingesetzt, auf der sich #goalpage# befindet. Damit jedes #topage# auch sein entsprechendes #goalpage# findet, geben Sie bei beiden Anweisungen einen TEXT – Parameter an.

```
... siehe auch auf Seite #topage("Funktionstasten")# ...
```

Auf einer anderen Seite befindet sich

```
... #goalpage("Funktionstasten")#
```

Nach 'Seite' wird die entsprechende Seitennummer eingesetzt.

Es ist möglich, mehrmals auf die gleiche (Ziel-)Seite zu verweisen. Sie müssen nur darauf achten, daß Sie immer das gleiche Merkmal (TEXT-Parameter) verwenden. Beachten Sie auch, daß die #goalpage#-Anweisungen sich in den Zeilen befinden müssen, die tatsächlich gedruckt werden. Setzen Sie sie nicht in die ersten Zeilen einer Seite oder eines Textes, die Anweisungen für das Layout enthalten.

Die Zahl der Querverweise darf 300 nicht übersteigen.

## 5.3.10. Kombination von Tabellen, Fußnoten und Kopf – bzw. Fußzeilen

*In Fußnoten, #head# – oder #bottom# – Bereichen können Tabellen untergebracht werden.*

```
#head#  
#lpos(0.0)##cpos(5.0)##rpos(11.0)#  
#table#  
Korrekturen      EUMEL – Benutzerhandbuch      S.007  
  
#table end#  
#end#
```

Die obigen Eingaben schreiben an jeden Seitenanfang folgenden Text:

Korrekturen                      EUMEL – Benutzerhandbuch                      S.007

Die Tabelle sollte also vollständig in den oben erwähnten Bereichen enthalten sein.

## 5.3.11. Formatierung von Spalten

*Mit der #columns#-Anweisung ist es möglich, einen Text in Spalten zu formatieren ("Zeitungsdruck").*

Durch die Angabe der #columns#-Anweisung wird 'pageform'/'autopageform' aufgefordert, den Text in Spalten zu formatieren. Die Spaltenbreite müssen Sie mit der #limit#-Anweisung einstellen.

```
#limit (18.0)#  
...  
#columns (2, 2.0)#  
#limit (8.0)#  
...
```

Anfangs schreiben Sie mit einer Zeilenbreite von 18 cm. Dann fordern Sie mit der #columns#-Anweisung zweispaltigen Druck an (zwischen den Spalten sollen 2 cm Abstand sein). Somit muß die #limit#-Anweisung (sie gilt für beide Spalten) auf 8 cm eingestellt werden.

Die interaktive Spaltenformatierung wird von 'pageform' wie gewohnt vorgenommen. Auf dem Bildschirm erscheint nun das Spaltenende, wobei die Nummer der Spalte angezeigt wird. Fußnoten werden spaltenweise eingeordnet und müssen somit die gleiche Zeilenbreite haben wie die restlichen Spalten.

'pageform'/'autopageform' erzeugt in der Druckdatei die Spalten hintereinander. Das folgende Beispiel zeigt einen Ausschnitt aus der Druckdatei mit Kopf- und Fußzeilen bei einem zweispaltigen Druck:

```
head-Zeilen
xx
xx
xx
bottom-Zeilen
#page##----- Ende Seite 1 Spalte 1 ----#
xx
xx
xx
#page##----- Ende Seite 1 Spalte 2 ----#
```

Die zweite Spalte erscheint also ohne Kopf- und Fußzeilen, die jedoch bei der Berechnung berücksichtigt werden. Beachten Sie, daß die Kopf- und Fußzeilen über die Spalten gehen können. Dies erreichen Sie durch geeignete #limit#-Anweisungen in den genannten Bereichen.

Die meisten Drucker plazieren die zweite Spalte im Druckbild neben die erste. Bei einigen wenigen Druckern müssen Sie die Spalten nebeneinander kleben.

Alle Anweisungen funktionieren beim spaltenweisen Formatieren wie üblich. Die #free#-Anweisung z.B. hält entsprechenden Platz in einer Spalte frei. Eine Ausnahme bildet die #page#-Anweisung. Sie vollzieht hier ein Spaltenende. Die #page#-Anweisung mit einem Parameter (welcher die Seitennummer der nächsten Seite angibt) vollzieht dagegen ein Seitenende.

Die #columns end#-Anweisung beendet die spaltenweise Formatierung. Sie wirkt wie eine #page#-Anweisung.

Überschriften (bzw. Textblöcke) über mehrere Spalten hinweg sind nur auf der ersten Seite direkt hinter der #columns#-Anweisung möglich.

#page#

#limit (10.0)#

Überschriften (bzw. Textblöcke) über mehrere Spalten hinweg sind nur auf der ersten Seite direkt hinter der #columns#-Anweisung möglich.

#columns (2,2.0)#

#limit (4.0)#

Die erste Spalte soll nur wenige Zeilen beinhalten. Das vorzeitige Beenden der Spalte erreicht man mit der #page#-Anweisung.

#page#

In der zweiten Spalte kann dann mit dem Schreiben des Textes fortgefahren werden.

.....  
.....  
.....  
.....  
.....

#columns end#

Druckbild (mit 'lineform' bearbeitet):

Überschriften (bzw. Textblöcke) über mehrere Spalten hinweg sind nur auf der ersten Seite direkt hinter der #columns# – Anweisung möglich.

Die erste Spalte soll nur wenige Zeilen beinhalten. Das vorzeitige Beenden der Spalte erreicht man mit der #page# – Anweisung.

In der zweiten Spalte kann dann mit dem Schreiben des Textes fortgefahren werden.

.....  
.....  
.....

Die Zeilen für die zweispaltige Überschrift werden berücksichtigt. Dies gilt jedoch nur unmittelbar hinter der #columns# –Anweisung. Möchten Sie diesen Effekt nochmals erzeugen, beenden Sie mit #columns end#, schreiben die breite Überschrift und schalten die #columns# –Anweisung wieder ein (jeweils unter richtiger Setzung von #limit#).

## 5.4. Index

### 5.4.1. Stichwort – und/oder Inhaltsverzeichnisse erstellen

*Mit dem Programm 'index' können Sie Stichwort- und Inhaltsverzeichnisse erstellen. Stichwortverzeichnisse können sortiert werden. Mehrere Stichwortverzeichnisse können Sie durch 'index merge' zusammenführen.*

Durch den Aufruf von:

```
gib kommando:  
index ("dateiname.p")
```

werden durch Indexanweisungen gekennzeichnete Worte in Dateien, den sogenannten Indexdateien, gespeichert.

Die Worte, die in einen Index übernommen werden sollen, müssen Sie in der Druckdatei für 'index' durch Anweisungen kennzeichnen. Solche Indexanweisungen werden von den anderen Textbe- und -verarbeitungs-Programmen ('lineform', 'pageform', EUMEL-Drucker) ignoriert. Sie können also bei dem Schreiben mit dem Editor gleich festlegen, welche Worte in einen Index aufgenommen werden sollen.

Solche Verzeichnisse von Worten werden im EUMEL-System allgemein als Index bezeichnet. 'index' kann ebenfalls benutzt werden, um ein Inhaltsverzeichnis und/oder ein Verzeichnis aller Abbildungen zu erstellen oder Literaturhinweise zu überprüfen.

Nachdem eine oder mehrere Indexdateien aus einer Druckdatei erstellt sind, werden die Indexdateien auf Anfrage alphabetisch sortiert. Bei einem Inhaltsverzeichnis sollten Sie die Sortierung natürlich ablehnen. Nach der Sortierung werden gleiche Einträge automatisch zusammengefaßt und die entsprechenden Seitennummern nacheinander aufgeführt.

**Praktischer Tip:**

Möchten Sie nur eine Sortierung, aber keine Zusammenfassung von Einträgen, dann lehnen Sie die Sortieranfrage ab. Anschließend können Sie die Indexdatei mit 'lex sort ("indexdateiname")' sortieren. Hierbei bleiben gleiche Einträge erhalten.

### Das Programm

```
gib kommando:  
index merge ("dateiname.i1", "dateiname.i2")
```

erlaubt es Ihnen, zwei durch 'index' erzeugte Verzeichnisse zusammenzuführen und – nach Anfrage – wieder zu sortieren.

### 5.4.1.1. Worte für 'index' kennzeichnen

*Worte, die in einen Index übernommen werden sollen, kennzeichnen Sie mit #ib# und #ie#.*

Da in einem Index – neben dem eigentlichen Worteintrag – die Seitennummer enthalten sein soll, arbeitet das Programm 'index' nur mit einer Druckdatei, d.h. einer Ausgabedatei von 'pageform'/'autopageform'. Die Indexworte werden in Indexdateien gesammelt. Die Indexdateien erhalten den Namen der bearbeiteten Datei, an den ".i" und die Nummer des Index angefügt wird.

```
... Hier wird eine Eigenschaft des #ib(1)#EUMEL-  
Systems#ie(1)# beschrieben. ...
```

Die durch die Anweisungen #ib# und #ie# gekennzeichneten Worte werden mit der dazugehörigen Seitennummer in die erste Indexdatei geschrieben.

Die Einträge in einer Indexdatei werden von den Seitennummern durch mindestens drei Punkte getrennt. Werden diese nicht gewünscht, können Sie sie leicht mit dem Editor entfernen.

Sie haben die Möglichkeit, bis zu neun unterschiedliche Indexdateien zu erstellen, z.B. gehen durch

```
#ib (1)# und #ie (1)#
```

gekennzeichnete Worte in die Indexdatei mit der Nummer 1, durch

```
#ib (9)# und #ie (9)#
```

gekennzeichnete Worte gehen in die Indexdatei mit der Nummer 9. Wenn Sie nur einen Index erstellen müssen, dürfen die #ib#- und #ie#-Anweisungen ohne Parameter benutzt werden, was gleichbedeutend ist mit #ib (1)# und #ie (1)#.

Die durch #ib#- und #ie#-Anweisungen gekennzeichneten Worte können auch über Zeilengrenzen (mit Silbentrennungen) gehen.

```
.... #ib#viele Index-  
Anweisungen#ie# ...
```

'index' zieht getrennte Worte zusammen (hier: 'viele Index-Anweisungen'). Möchten Sie einige Worte in verschiedenen Indexdateien haben, dürfen Sie die #ib#- und #ie#-Anweisungen auch "schachteln". Dies können Sie besonders bei Kapitelüberschriften nutzen.

```
#ib(9)#Eine Anweisung: die '#ib#limit#ie#'-Anweisung#ie(9)#
```

In diesem Beispiel wird das Inhaltsverzeichnis in die Indexdatei '9' gebracht, während der "allgemeine" Index in der Indexdatei '1' gesammelt wird.

## 5.4.1.2. Nebeneinträge erzeugen

*Sie haben die Möglichkeit, an die Seitennummer eines Eintrags einen beliebigen Text anfügen zu lassen.*

Beispiel:

```
EUMEL-System ... 27ff.  
Monitor ..... 13(Def.)
```

Dies wird durch eine weitere Form der #ib# – Anweisung ermöglicht:

```
... der #ib(1,"(Kap.4)")#EUMEL-Editor#ie# ist gut  
geeignet, Texte zu erstellen ...
```

erzeugt den folgenden Eintrag:

Druckbild:

```
EUMEL – Editor ... 1(Kap.4)
```

An einen Eintrag können Sie einen weiteren Text anfügen, um etwa Untereinträge zu bilden:

Druckbild:

EUMEL – System ..... 27

EUMEL – System, komplexes ..... 29

Das wird ebenfalls durch eine andere Form der #ib# – Anweisung ermöglicht:

```
... ist das #ib#EUMEL-System#ie(1," benutzerfreundliches")#  
wirklich ein benutzerfreundliches System ...
```

erzeugt den folgenden Eintrag:

Druckbild:

EUMEL – System, benutzerfreundliches ..... 28

Nach der Erstellung einer Indexdatei können – nach interaktiver Anfrage – die Einträge sortiert werden. Die Sortierung erfolgt alphabetisch nach DIN 5007, Abschnitt 1 und 3.2 (Umlaute werden "richtig" eingeordnet).

Wie bereits erwähnt, können Sie 'index' vielseitig einsetzen:

- a) Erstellung von Stichwortverzeichnissen:  
Wie bereits beschrieben.
- b) Erstellung von Inhaltsverzeichnissen:  
Kapitelüberschriften mit eigenen Indexanweisungen klammern und durch 'index' wie beschrieben verarbeiten.

#ib(9)#6.1. Eine Datei drucken#ie(9)#

Dann sind Sie sicher, daß das Inhaltsverzeichnis bezüglich Seitennummern und Kapitelüberschriften korrekt ist.

- c) Erstellung von Abbildungsverzeichnissen:  
Abbildungsüberschriften – bzw. – unterschritten wie Kapitelüberschriften verarbeiten.
- d) Überprüfung von Literaturhinweisen auf Vollständigkeit:  
Sie klammern alle Literaturhinweise mit gesonderten Indexanweisungen.

#ib(8)#/Meier82/#ie(8)#

und überprüfen dann mit Hilfe dieser Indexdatei die Literaturverweise. So können Sie sichergehen, daß alle Literaturverweise im Text auch in der Literaturliste stehen.

### 5.4.1.3. Indexdateien zusammenführen

*Durch das Programm 'index merge' können Sie eine Indexdatei in eine zweite "einmischen".*

Es ist somit möglich, einen Index zu erstellen, der sich über mehrere Dateien erstreckt, indem Sie 'index' die Druckdateien dieser Dateien bearbeiten und anschließend die entstandenen Indexdateien mit 'index merge' zusammenfassen lassen. Indexdateien können ggf. mit dem Editor bzw. 'lineform' und/oder 'pageform'/'autopageform' bearbeitet und anschließend gedruckt werden.

```
gib kommando:  
index merge ("1.kapitel.il", "2.kapitel.il")
```

Hier wird die Indexdatei des 1. Kapitels in die Indexdatei des 2. Kapitels eingeordnet und auf Wunsch sortiert.

## 5.5. Outline

### 5.5.1. Eine Strukturübersicht oder Zusammenfassung erstellen

*Das Programm 'outline' erstellt aus einem Text eine Zusammenfassung aller (Kapitel-) Überschriften und Stichworte, sofern diese mit Index-Anweisungen gekennzeichnet sind.*

Manchmal sollen Stichworte oder das Inhaltsverzeichnis aus einem Text herausgeholt werden, ohne vorher 'pageform' durchlaufen zu müssen. Das ist dann nützlich, wenn Sie

- Stichworte auf Korrektheit und Vollständigkeit überprüfen möchten;
- die Reihenfolge von Kapiteln überprüfen müssen;
- eine Übersicht durch Kapitel-Überschriften und Stichworte anfertigen möchten;
- einen Text auf logische Zusammenstellung überprüfen.

In solchen Fällen hilft das Programm 'outline', das mit dem Monitor-Kommando

```
gib kommando:  
outline ("dateiname")
```

aufgerufen wird. 'outline' arbeitet ähnlich wie 'index', indem es alle mit #ib# und #ie# markierten Textteile in eine Datei mit dem Zusatz 'outline' schreibt. Im Unterschied zu 'index' muß die Eingabe-Datei keine Druckdatei ('.p' - Zusatz) sein.

Das Programm 'outline' fragt zuerst, mit welcher Indexnummer das Inhaltsverzeichnis versehen ist. Das ist notwendig, weil die Kapitelüberschriften gegenüber Stichwörtern in der 'outline' – Datei hervorgehoben werden (Einrückungen).

Eingabe – Datei ("dateiname"):

```
...
#ib(9)#1. Kapitel#ie(9)#
...
...#ib#Stichwort 1#ie#
#ib#Stichwort 2#ie#...

#ib(9)#1.1. Kapitel#ie(9)#
...
#ib#Stichwort 3#ie#
usw...
```

Druckbild der erzeugten Datei ("dateiname.outline"):

- 1. Kapitel
  - Stichwort 1
  - Stichwort 2
- 1.1. Kapitel
  - Stichwort 3

In diesem Beispiel werden alle Indizes mit Ausnahme der Kapitelüberschrift jeweils in einer Zeile aufgeführt und gegenüber der Kapitelüberschrift eingerückt. Ein neues Kapitel, sofern es dezimal gekennzeichnet ist, wird gegenüber einem Kapitel mit höherer Ordnung eingerückt.

## 5.6. Print

*Der EUMEL-Drucker, der mit dem 'print'-Kommando angesprochen wird, ist eine Software-Schnittstelle zu einem angeschlossenen Drucker. In diesem Kapitel wird erklärt, wie Sie mit dem EUMEL-Drucker eine Datei drucken können und welche speziellen Anweisungen den Drucker steuern.*

Jeder Drucker erbringt "hardwaremäßig" unterschiedliche Leistungen (z.B. Typen und Modifikationen). Diese Leistungen werden durch Eingabe spezieller Zeichenfolgen veranlaßt, die herstellerspezifisch sind.

Um vom EUMEL-System unterschiedliche Drucker auf gleiche Weise ansprechen zu können, wurde eine Software-Schnittstelle geschaffen, die EUMEL-Drucker genannt wird. Der EUMEL-Drucker akzeptiert eine Datei und veranlaßt, daß diese in geeigneter Weise gedruckt wird. Weiterhin beachtet der EUMEL-Drucker die Anweisungen der Textkosmetik. Die Form der Anweisungen der Textkosmetik und des EUMEL-Druckers sind identisch.

## 5.6.1. Eine Datei drucken

*Mit dem Kommando 'print' können Sie dem EUMEL-Drucker eine Datei zum Drucken übergeben.*

```
gib kommando:  
print ("dateiname")
```

In der Regel ist im EUMEL-System (Multi-User) ein "Spooler" installiert, so daß Sie sofort mit der Arbeit fortfahren können. Der EUMEL-Drucker arbeitet in diesem Fall parallel zu Ihren anderen Arbeiten.

## 5.6.2. Anweisungen für den EUMEL – Drucker

Ein Text (eine Datei) kann vom Drucker auch ohne Anweisungen gedruckt werden, etwa für Probdrucke. Für diesen Fall hat der Drucker vernünftige Voreinstellungen. Für einen "normalen" Text brauchen Sie keine speziellen Druckeranweisungen in den zu druckenden Text einzufügen, denn die Anweisungen für die Textkosmetik reichen zur Druckersteuerung aus. Nur wenn besondere Leistungen verlangt werden, wie z.B. Blocksatz oder den gedruckten Text an eine bestimmte Stelle zu plazieren, sind Druckeranweisungen notwendig.

Werden vom Drucker Leistungen verlangt, die hardwaremäßig nicht vorhanden sind, so sorgt der EUMEL – Drucker dafür, daß eine möglichst äquivalente Leistung erbracht wird. Fordern Sie beispielsweise einen nicht vorhandenen Schrifttyp an, wird mit dem Standard – Schrifttyp der jeweiligen Installation gedruckt. Damit ist es Ihnen möglich, einen Text, der eigentlich für einen anderen Drucker bestimmt ist, auf einem Drucker zu drucken, der die geforderte Type nicht kennt.

Wie bereits erwähnt, beachtet der EUMEL – Drucker die gleichen Anweisungen wie die Textkosmetik – Programme, aber einige Anweisungen sind nur für den Drucker implementiert. Eine #type# – Anweisung beispielsweise, die einen bestimmten Schrifttyp anfordert, wird vom EUMEL – Drucker als Befehlsfolge an den angeschlossenen Hardware – Drucker übergeben, sofern der Schrifttyp auf dem Drucker vorhanden ist. Wie die Anweisungen geschrieben werden müssen, wurde in der Beschreibung der Textkosmetik geschildert.

Anweisungen werden nicht gedruckt. Besteht eine Zeile nur aus Anweisungen, so wird diese Zeile vom EUMEL – Drucker nicht gedruckt. Im Gegensatz zu den Programmen der Textkosmetik werden unbekannte oder fehlerhafte Anweisungen vom EUMEL – Drucker ohne Fehlermeldung "verschluckt".

Neben den "normalen" Anweisungen, die nur in "#"-Zeichen eingeschlossen werden, gibt es noch eine andere Form:

**Kommentar – Anweisungen:**

Werden in "#-" und "-#" – Zeichen eingeschlossen. Solche Anweisungen werden ignoriert.

```
.....  
Text.....  
.....  
Kommentar-Anweisungen werden  
beim Drucken ignoriert.  
#---- Ende der Seite 1 ----#
```

Die letzte Zeile erscheint im gedruckten Text nicht.

## 5.6.3. Blocksatz

### 5.6.3.1. Randausgleich

*Die Anweisung #block# bewirkt einen Blocksatz beim Druck.*

Fügen Sie in den Text (meist am Anfang einer Datei) die Anweisung

```
#block#
```

ein, druckt der Drucker ab dieser Stelle alle Zeilen, die nicht mit einem Absatzkennzeichen versehen sind, im Blocksatz. Das heißt, daß durch Vergrößern der Wortabstände alle Zeilen an der gleichen Position enden (rechter Randausgleich). Preiswerte Drucker können dies nur durch Einfügen ganzer Leerzeichen zwischen den Worten vornehmen, was sich oft beim Lesen störend bemerkbar macht. Bei qualitativ hochwertigen Druckern wird dagegen der Blocksatz durch Einfügen kleinerer Abstände zwischen den Worten erreicht.

Der Text einer Zeile wird durch Vergrößern der Wortlücken auf die Zeilenlänge, die durch die #limit# – Anweisung eingestellt ist, verbreitert.

a) Es werden nicht verbreitert:

- Absatzzeilen;
- der Text bis zum letzten Mehrfachblank;
- führende Leerzeichen (Einrückung);
- ein Leerzeichen hinter einer Aufzählung (siehe dazu b);
- geschützte Blanks.

b) Aufzählungen gibt es nur nach einer Absatzzeile:

- "Spiegelstrich" (Bindestrich und Leerzeichen am Anfang der Zeile);
- Doppelpunkt als Ende des ersten Wortes (Position < 20);
- schliessende Klammer oder Punkt als Ende des ersten Wortes (Position < 7), z.B. 1) oder 1.

## 5.6.3.2. Seitenausgleich

*Mit der #pageblock#-Anweisung wird der Drucker veranlaßt, einen Seitenausgleich (ähnlich wie bei der #block#-Anweisung für den rechten Randausgleich) vorzunehmen.*

Durch die automatische oder interaktive Seitenformatierung oder durch einen Fußnotenumbruch von 'pageform'/'autoform' bleiben oft am Ende einer Seite Zeilen leer. Dies können Sie durch die #pageblock#-Anweisung verhindern. Sie veranlaßt den Drucker, Zwischenräume (Fachbegriff: Durchschuß) zwischen den Zeilen einzufügen, so daß alle letzten Zeilen auf allen Seiten auf gleicher Höhe abschließen. Ebenso wie beim Randausgleich hängt die Güte des Druckergebnisses jedoch von den Fähigkeiten des angeschlossenen Druckers ab.

Beachten Sie jedoch, daß manche Verlage so bearbeitete Seiten nicht wünschen, weil bei Verwendung von zu dünnem Papier beim Druck Zeilen "durchscheinen" können, so daß das Lesen erschwert wird.

Ist die Anweisung #pageblock# gegeben, können Sie in 'pageform' die Seitengrenze auch über das rechnerische Seitenende hinaus plazieren. In diesem Fall werden die Zeilen vom Drucker gestauch.

PAGEFORM für x Zeilen: dateiname > dateiname.p

Seitenende verschieben: UP, DOWN / bestätigen: RETURN / Abbruch: FSC

## 5.6.4. Schreibfeld verschieben

*Durch die Anweisung #start# ist es Ihnen möglich, das Schreibfeld beim Druck auf dem Papier an eine andere Stelle zu plazieren.*

Der EUMEL – Drucker plaziert das Schreibfeld auf einem Drucker automatisch derart, daß ein genügender Rand verbleibt. Die Wirkung dieser Voreinstellung ist natürlich abhängig vom Drucker und der Installation. Mit der #start# – Anweisung können Sie die automatische Einstellung verändern.

```
#start (1.0, 2.0)#
```

legt die linke, obere Ecke des Schreibfeldes fest (vom linken Rand 1 cm, vom oberen Rand 2 cm). Die standardmäßige Voreinstellung ist #start (2.54, 2.35)#. Die #start (...)# – Anweisung können Sie nur einmal pro Seite geben.

## 5.6.5. Zentriert drucken

*Mit der #center# – Anweisung können Sie einen Text in der Mitte der Zeile drucken lassen.*

Die #center# – Anweisung zentriert den Text einer Absatzzeile.

`#center#Diese Zeile wird zentriert gedruckt.`

Druckbild:

Diese Zeile wird zentriert gedruckt.

## 5.6.6. Rechtsbündig drucken

*Mit der #right#-Anweisung können Sie einen Teil einer Absatzzeile rechtsbündig drucken.*

Die #right#-Anweisung veranlaßt, daß der nachfolgende Text rechtsbündig gedruckt wird.

```
#head#  
#center#Diese Zeile wird zentriert#right#%  
  
#end#
```



Hierbei wird die Seitenzahl rechtsbündig gedruckt.

Beachten Sie, daß die #center#- und die #right#-Anweisung zusammen verwendet werden können. Beide Anweisungen wirken jedoch nur, wenn sie in einer Absatzzeile stehen.

## 5.6.7. Übereinander drucken

*Mit der #b# – Anweisung können Sie zwei Zeichen übereinander drucken.*

Die #b# – Anweisung veranlaßt, daß zwei aufeinanderfolgende Zeichen, die durch die #b# – Anweisung verbunden sind, übereinander gedruckt werden.

... 0#b# / ...

Druckbild:

0

Das Zeichen '/' wird über das Zeichen '0' gedruckt. 'lineform'/'autoform' nimmt für die Zeilenberechnung nur ein Zeichen. Beachten Sie, daß direkt vor oder nach der #b# – Anweisung keine Anweisung oder kein Blank stehen darf.

## 5.7. Textkosmetik – Makros

*Makros verkürzen Ihren Arbeitsvorgang bei immer wiederkehrenden Textteilen und/oder Anweisungen.*

Unter 'Makro' verstehen wir eine "große" Anweisung, die aus vielen kleinen besteht und die Sie mit Hilfe des Makronamens aufrufen können.

Textkosmetik – Makros kommen zum Einsatz bei:

- immer wiederkehrenden Textteilen;
- immer wiederkehrenden Anweisungssequenzen;
- bei der Erstellung von Manuskripten, deren endgültige Form Sie anfänglich noch nicht kennen oder die Sie noch ändern möchten;
- oder bei Folgen von direkten Drucker – Anweisungen, die bestimmte Leistungen erbringen.

Die Definition von einem oder mehreren Makros wird mit dem Editor vorgenommen. Diese Makro – Datei wird dann geladen. Von diesem Augenblick an "kennen" 'lineform'/'autoform' und 'pageform'/'autopageform' die Makros, d.h. die Textzeilen und/oder Anweisungen, die sich unter dem Makronamen "verbergen".

'lineform'/'autoform' beachtet die Anweisungen, die ggf. in den Makros enthalten sind. Sie erscheinen jedoch nicht in der Datei. Erst 'pageform'/'autopageform' setzt diese in die Druckdatei ein.

## 5.7.1. Ein Makro – Beispiel

*Hier wird Ihnen ein einfaches Beispiel für einen Briefkopf gezeigt.*

Angenommen, Sie schreiben mit dem EUMEL-System Ihre Geschäftsbriefe. Sie haben einen Drucker zur Verfügung, mit dem Sie auch die Briefköpfe erstellen können. Für den Briefkopf schreiben Sie ein Makro `#kopf#` in eine Datei "macro definitionen":

```
##kopf#  
#type("fett und gross")#Firmenname  
#type("fett")#Softwareprodukte  
#type("klein")#Straße  
Stadt  
#type("normal")#  
##macro end#
```

Der Name des Makros ist `#kopf#`. Beachten Sie, daß eine Makro-Definition mit dem Namen des Makros beginnen muß. Der Makroname muß dabei mit einem `*` gekennzeichnet werden, um ihn von "normalen" Text-Anweisungen unterscheiden zu können. Jedes Makro wird mit einer `##macro end#`-Anweisung beendet. Sie dürfen mehrere Makros hintereinander in die Datei schreiben.

Nun müssen Sie das so definierte Makro 'laden':

```
gib kommando:  
load macros ("macro definitionen")
```

Zur Kontrolle können Sie sich die geladenen Makros in das Notizbuch ausgeben lassen:

```
gib kommando:  
list macros
```

Nun haben Sie von jetzt an eine neue Anweisung (mit dem Namen #kopf#) zur Verfügung, mit der Sie einen Briefkopf in jeden Brief drucken können. Sie schreiben nun folgenden Brief:

```
#kopf#  
  
Sehr geehrter Herr ....  
  
usw.
```

Beachten Sie hierbei, daß das Makro in Ihrem Text als Anweisung ohne \* steht. Der Aufruf eines Makros, welches z.B. in einer von 'lineform' zu bearbeitenden Datei steht, unterscheidet sich also nicht von einer "normalen" Textanweisung.

Nachdem Sie mit 'lineform' den Brief zeilenweise formatiert haben, kontrollieren Sie die formatierte Datei. Hier hat sich noch nichts verändert. Die neue Anweisung #kopf# steht unverändert in der Datei. 'lineform' beachtet zwar alle Anweisungen und Textzeilen eines Makros, setzt diese jedoch nicht in die Datei ein. Allerdings ist 'lineform' nicht in der Lage, die #type#- und #limit#-Anweisungen eines Makros zu erkennen, wenn es an erster Stelle in einer Datei steht und in dessen Definition gleich zu Anfang diese Anweisungen korrekt aufgeführt sind. Stattdessen fragt 'lineform' anfangs 'type' und 'limit' an. Das können Sie umgehen, indem Sie mittels 'CR' die Abfrage in 'lineform' ignorieren.

Nun formatieren Sie die Datei, die den Brief enthält, mit 'pageform'/'autopageform'. In der Druckdatei ist nun die Anweisung #kopf# verschwunden. Dort stehen nun die Zeilen des Makrorumpfes. 'pageform'/'autopageform' setzt die Zeilen des Makros in die Druckdatei ein:

```
#type("fett und gross")#Firmenname  
#type("fett")#Softwareprodukte  
#type("klein")#Straße  
Stadt  
#type("normal")#
```

```
Sehr geehrter Herr ...  
usw.
```

**Anmerkung:**

Makros, die den gleichen Namen haben, aber sich durch die Anzahl der Parameter unterscheiden, sind nicht erlaubt. Es ist auch nicht gestattet, Makros innerhalb einer Makro-Definition aufzurufen.

Beachten Sie ferner, daß Makro-Texte so verwendet werden, wie sie mit 'load macros' geladen werden.

```
**textanfang#  
#limit(11.0)#  
#block#  
#pageblock#  
#type("trium8")#  
**macro end#
```

Betätigen Sie in der Makro-Datei nach jeder Zeile die **CR** -Taste (Absatz), dann erhalten Sie nach jedem #...# einen Absatz, was zum Beispiel bei Kapitelüberschriften wünschenswert ist, nicht jedoch bei kleineren Anweisungen, bei denen dann mitten im Satz ein Absatz erschiene. In solchen Anwendungen sollten Sie Makros ohne Absätze speichern. Beachten Sie ferner, daß aus programmtechnischen Gründen eine #foot#- oder die abschließende #end#-Anweisung einer Fußnote nicht in einem Makro enthalten sein darf.

## 5.7.2. Ein Beispiel mit Makro – Parametern

*Makro-Parameter erlauben es Ihnen, immer wiederkehrende Textteile, die sich nur geringfügig voneinander unterscheiden, zu erzeugen.*

Ihnen fällt nun auf, daß Sie Ihr Makro noch etwas verbessern können. Sie möchten das Datum mit in den Briefkopf aufnehmen. Somit editieren Sie Ihre Makro-Datei folgendermaßen (beachten Sie die '\$' – Zeichen):

```
##kopf ($1)#  
#type("gross")#Firmenname  
#type("fett")#Softwareprodukte  
#type("klein")#Straße  
Stadtname  
#type ("normal")#  
  
Stadtname, den $1  
  
##macro end#
```

Damit haben Sie dem #kopf# – Makro einen Parameter gegeben: '\$1'; die Parameter werden numeriert. Ein zweiter Parameter würde '\$2' heißen usw..

Bei der Erstellung eines Briefes müssen Sie die Anweisung #kopf# mit dem jeweiligen Datum in einen Brief schreiben:

```
#kopf ("20.8.1986")#.
```

'pageform'/'autopageform' setzt nun das angegebene Datum direkt hinter 'Stadtname, den' in den Briefkopf ein (in der Druckdatei). Beachten Sie, daß alle Parameter einer Makro – Anweisung in Anführungszeichen stehen müssen (auch Zahlen).

### 5.7.3. Makros für Manuskripte

*Hier wird gezeigt, wie Sie mit Makros Anweisungen formulieren können, die aussagen, um was es sich bei einem Text handelt, und nicht, in welchem Format er gedruckt wird.*

Bei Manuskripten für Artikel, Bücher und Manuals wissen Sie oft vorher nicht, in welchem Format das Manuskript gedruckt werden wird. Zu diesem Zweck ist es ebenfalls nützlich, Makros zu verwenden.

```
##kapitelanfang ($1)#  
#free (2.0)#  
#type ("gross")##ib (9)#$1#ie (9)##type ("normal")#  
  
##macro end#
```

In diesem Beispiel wird ein Makro für den Anfang eines Kapitels definiert. Zwischen zwei Kapiteln sollen hier zwei Zentimeter Zwischenraum bleiben, die Kapitel-Überschrift (als Parameter) wird in einer größeren Schrift gedruckt. Zusätzlich wird die Überschrift für ein Inhaltsverzeichnis in den 9. Index aufgenommen. Nach der Überschrift wird eine Leerzeile eingeschoben, bevor der eigentliche Text anfängt.

Der Anwender dieses Makros schreibt also z.B. folgende Anweisung:

```
#kapitelanfang ("Ein Beispiel fuer Manuskripte")#
```

Beachten Sie, daß die Kapitel – Überschrift nicht länger als eine Textzeile sein darf. Das liegt daran, daß 'lineform'/'autoform' zwar die Zeile bearbeitet, aber nicht in den Text einsetzt. 'pageform'/'autopageform' setzt also die unveränderte – nicht umgebrochene – Textzeile ein.

Sie können nun Makros für die meisten Textstrukturen definieren. Schreibkräfte brauchen dann in der Regel die meisten Text – Anweisungen nicht zu kennen, sondern nur noch eine Anzahl von einfachen Makro – Anweisungen.

Die Makro – Definitionen können jederzeit geändert werden, um wechselnden Bedürfnissen angepaßt zu werden, z.B. wenn ein Verlag ein bestimmtes Schreibformat verbindlich vorschreibt. In diesem Fall brauchen nicht alle Text – Dateien geändert zu werden, sondern nur die Makro – Definitionen.

Ein weiterer Vorteil einer solchen Vorgehensweise ist, daß die Makro – Anweisungen in diesem Fall angeben, **was** eine bestimmte Text – Struktur ist, und nicht, **wie** die Struktur behandelt werden soll.

### **Anmerkung:**

In eine Makro – Definition sollten Sie ggf. #limit# – , #type# – und #linefeed# – Angaben einsetzen, um die Makros unabhängig von der Aufrufstelle zu machen. Ggf. sollten Sie auch die Datei vorher mit 'lineform' bearbeiten, um Trennungen vorzunehmen.

## 5.8. Textkosmetik für Spezialisten

*In diesem Abschnitt werden Ihnen Kommandos und Anweisungen vorgestellt, die in der Regel nur für Spezialfälle benötigt werden.*

### 5.8.1. Schalter – Anweisungen für Kopf – und Fußbereiche

Mit den Textkosmetik – Anweisungen

```
#head off#
```

```
#bottom off#
```

können Sie die Erzeugung von Kopf – oder Fußzeilen abschalten. Mit

```
#head on#
```

```
#bottom on#
```

können Sie diese wieder erzeugen. Beachten Sie, daß diese Anweisungen an der Stelle beachtet werden, an der sie im Text stehen, d.h. diese Anweisungen gelten bereits für die Seite, auf der sie sich bei der 'pageform' – Bearbeitung befinden. Möchten Sie die Kopfzeilen für eine Seite abschalten, dann sollten Sie an dieser Stelle die #head off# – Anweisung geben. Um die Kopfzeilen für die nächste Seite wieder einzuschalten, sollten Sie die #head on# – Anweisung an einer Stelle platzieren, von der Sie sicher sind, daß sie auf die folgende Seite gelangt (im Zweifelsfall nach einer #page# – Anweisung).

## 5.8.1.1. Kopf- und Fußbereiche abstellen

*Mit 'first head' bzw. 'last bottom' können Sie Kopf- oder Fußbereiche auf der ersten (letzten) Seite ab- oder wieder anschalten.*

Manchmal ist es notwendig, die Erzeugung von 'head'-Zeilen auf der ersten Seite (z.B. weil dort ein Briefkopf erscheint) und/oder 'bottom'-Zeilen auf der letzten Seite (weil keine Folgeseite existiert) zu verhindern. Mit dem Monitor-Kommando

```
gib kommando:  
first head (FALSE)
```

können Sie bei 'pageform' die Erzeugung von 'head'-Zeilen auf der ersten Seite jeder Druckdatei abschalten. Die Erzeugung bleibt so lange abgeschaltet, bis sie wieder durch

```
gib kommando:  
first head (TRUE)
```

angeschaltet wird. Das gleiche gilt analog für 'bottom'-Zeilen auf der letzten Seite: Ein- und Ausschalten durch

```
gib kommando:  
last bottom (FALSE)
```

bzw.

```
gib kommando:  
last bottom (TRUE)
```

## 5.8.2. Textzeilen markieren

Mit der Anweisung

```
#mark("markierungszeichen links","markierungszeichen rechts")#
```

können Sie einen Textabschnitt an den Rändern (außerhalb des Schreibfeldes!) mit Texten markieren, wie z.B. im folgenden mit der Anweisung

```
#mark("> ", "<")#
```

- > Dabei gilt der erste Parameter für den linken und der zweite für den rechten Rand. <
- > Beachten Sie, daß Sie einen genügenden Zwischenraum zwischen der Markierung <
- > und dem Rand mit angeben müssen. <
- > <
- > Die Markierung ist insbesondere für Manuals interessant, wo Änderungen gegen <
- > über der letzten Version hervorgehoben werden. Das Markierungszeichen wird neben <
- > den linken und rechten Rand gedruckt (also außerhalb des von #start# und #limit# <
- > begrenzten Textfeldes). Für das Drucken der Markierung wird der/die Schrifttyp/ Modi- <
- > fikationen benutzt, die an der Stelle der #mark# – Anweisung eingeschaltet ist. Der <
- > eigentliche Text bleibt selbstverständlich unberührt. <

Um nur einen Rand zu markieren, kann auch ein leerer Parameter angegeben werden.

```
#type ("pica")##mark ("", " |")##type ("normal")#
```

Mit der speziellen #mark# – Anweisung

```
#mark ("", "")#
```

wird die Markierung ausgeschaltet.

Soll ein Kopf-, Fuß-, Fußnoten- oder Tabellenbereich markiert werden, sollten sich die Markierungsein- und ausschalt-Anweisungen vollständig in dem Bereich befinden.

### 5.8.3. Fußnoten pro Seite zählen

Manchmal wird gewünscht, daß die Fußnoten für jede Seite separat – also für jede Seite von 1 ab – gezählt werden. Das können Sie mit der Textkosmetik – Anweisung

`#count per page#`

erreichen. Sie schaltet von einer fortlaufenden Zählung auf eine seitenweise Zählung um. Diese Anweisung sollte am Dateianfang stehen. Sie kann für die betreffende Datei nicht mehr abgeschaltet werden.

## 5.8.4. Behandlung falscher Silbentrennungen: Ausnahmelexikon

*In das Ausnahmelexikon können fehlerhaft getrennte Worte aufgenommen werden.*

Es kann vorkommen, daß das Silbentrenn-Programm der Textkosmetik einige Worte immer wieder falsch trennt. Um dies zu vermeiden, können Sie diese Worte in ein **Ausnahmelexikon** speichern. Die Worte des Ausnahme-Lexikons werden bei einer Silbentrennung zuerst durchsucht. Wird ein Wort im Lexikon gefunden, dann wird das eigentliche Silbentrenn-Programm nicht mehr ausgeführt.

Die Ausnahmen müssen Sie – wie unten beschrieben – in einer Datei notieren und mit dem Monitor-Kommando

```
gib kommando:  
lade ausnahmen ("dateiname").
```

in das Lexikon laden. Die Ausnahmen müssen Sie folgendermaßen in die Datei schreiben:

```
Sprech-stun-de  
ins-be-son-de-re  
Raum  
Bei-spiel  
...
```



Sie können jederzeit neue Ausnahmen in das Lexikon hinzuladen (wiederum mit 'lade ausnahmen'). In diesem Fall wird angefragt, ob das Lexikon überschrieben werden soll.

Um zu kontrollieren, welche oder wie viele Ausnahmen sich im Lexikon befinden, können Sie

```
gib kommando:  
entlade ausnahmen ("dateiname")
```

geben. Das Lexikon wird dann in "dateiname" geschrieben. Auch hier können Sie weitere Ausnahmen hinzufügen und diese neu laden (aber diesmal überschreiben).

## 5.8.5. Voreinstellungen ändern: Einige Monitor – Kommandos

### 5.8.5.1. Wenige oder viele Silbentrennungen: Trennpunkt einstellen

*Mit dem Kommando 'hyphenation width' können Sie bestimmen, an welchem Punkt Worte zur Trennung angeboten werden. Die Trennbreite können Sie zwischen 4 und 20 Prozent der Zeilenbreite einstellen.*

Viele Silbentrennungen in einem Text erschweren das Lesen. Nehmen Sie keine Silbentrennungen vor, wird der rechte Rand stark "ausgefranst" oder beim Blocksatz ("rechter Randausgleich") müssen viele Zwischenräume zwischen den Worten eingefügt werden. Durch das Monitor – Kommando

```
gib kommando:  
hyphenation width (prozentuale angabe)
```

unmittelbar vor dem Aufruf von 'autoform' oder 'lineform' können Sie den Punkt, an dem die Silbentrennung einsetzen soll, einstellen. Die Klammern enthalten eine ganze Zahl, die für Prozent der Zeilenbreite steht. Minimum sind 4, Maximum 20 Prozent. Beispielsweise stellt 'hyphenation width (5)' den Trennpunkt auf 5% der Zeilenbreite ein (voreingestellt ist 7). Bei einer Angabe von 20 werden somit sehr wenige Worte zur Silbentrennung angeboten, d.h. je größer die Prozentangabe, desto weniger Worte werden zur Trennung angeboten. Die Einstellung des Trennpunktes bestimmt also, ab wann ein Wort zur Silbentrennung untersucht wird. Andererseits bestimmt die Einstellung auch, wieviel Zwischenraum zwischen Worten eingefügt werden muß, um einen rechten Randausgleich zu erzielen.

## 5.8.5.2. Anzahl Leerzeilen vor Fußnoten einstellen

*'number empty lines before foot' stellt die Anzahl der Leerzeilen vor Fußnoten ein.*

Die Anzahl der Leerzeilen vor Fußnoten (voreingestellt ist eine Leerzeile) können Sie durch das Monitor – Kommando 'number empty lines before foot' einstellen.

```
gib kommando:  
number empty lines before foot (3)
```

stellt drei Leerzeilen vor dem Fußnotenblock ein. Beachten Sie, daß diese Einstellung so lange gilt, bis Sie das Monitor – Kommando erneut geben.

## 5.9. Übersicht über die Anweisungen und Kommandos der EUMEL – Textkosmetik

*Zuerst werden die am häufigsten benutzten Kommandos/Anweisungen beschrieben. Danach sind (durch einen Strich getrennt) Kommandos/Anweisungen aufgeführt, die seltener benötigt werden.*

### Kommandos

*Kommandos werden im Monitor gegeben ('gib kommando :').*

<b>Kommando</b>	<b>Bedeutung</b>
lineform ("x")	Formatieren von Zeilen mit interaktiver Silbentrennung.
autoform ("x")	Wie lineform, jedoch werden Silbentrennungen automatisch vorgenommen.
pageform ("x")	Interaktives Formatieren von Seiten, mit Behandlung von Fußnoten, Kopf- und Fußzeilen, Seitennumerierung, Seitenquerverweisen usw. Erzeugt eine Druckdatei (Zusatz '.p').
autopageform ("x")	Wie pageform, jedoch werden die Seitengrenzen automatisch plaziert.
print ("x")	Datei drucken.
print ("x.p")	Eine mit 'pageform' bearbeitete Datei drucken.
-----	-----

index ("x.p")	Erstellt aus einer Druckdatei ein Stichwort – und/oder Inhaltsverzeichnis.
index merge ("a.i1", "b.i1")	Führt Indexdateien zusammen.
outline ("x")	Erstellt eine Übersicht aus Kapitelüberschriften und Stichworten.
hyphenation width (int)	Stellt die Trennbreite für die Silbentrennung ein.
load macros ("x")	Lädt Makros.
list macros	Zeigt geladene Makros.
lade ausnahmen ("x")	Lädt Wörter, die von der Trennhilfe nicht korrekt getrennt werden, in einen Ausnahme – Speicher.
entlade ausnahmen ("x")	Entlädt die Worte aus dem Ausnahme – Speicher in die angegebene Datei.
first head (false)	Schaltet Kopfzeilen auf erster Seite aus.
first head (true)	Schaltet Kopfzeilen auf erster Seite wieder ein.
last bottom (false)	Schaltet Fußzeilen auf letzter Seite aus.
last bottom (true)	Schaltet Fußzeilen auf letzter Seite wieder ein.
number empty lines before foot before foot	Stellt die Anzahl der Leerzeilen vor einer Fußnote ein.

## Anweisungen

Anweisungen werden in die Datei geschrieben. Jede Anweisung muß in Anweisungszeichen eingeschlossen werden. Als Parameter (diese werden in Klammern eingeschlossen) kommen in Frage:

'int' bedeutet eine ganze Zahl: 17, 1, 311;

'real' bedeutet eine Zahl mit Dezimalpunkt (meist cm – Angabe): 0.5, 1.25;

'text' bedeutet eine Zeichen – Angabe. Muß in Anführungszeichen eingeschlossen werden: "%", "meine datei".

Anweisung	Bedeutung
type (text)	Schrifttyp einstellen: #type("trium8")#.
limit (real)	Zeilenbreite einstellen: #limit (16.0)#.
on (text)	Modifikation einschalten: #on("bold")#. Erlaubt sind: b(bold), r(verse), i(talic), u(nderline)
off (text)	Modifikation ausschalten (siehe 'on').
block	Blocksatz (Randausgleich) einschalten.
head	Kopfzeilen (für Seiten mit geraden/ungeraden
(bzw. headeven/headodd)	Seitennummern) definieren.
... - % -	Platzhalter für Seitenzahl
end	Kopfzeilen – Ende (pageform)
bottom	Wie oben, jedoch für Fußzeilen.
(bzw. bottomeven/bottomodd)	
...	
end	Fußzeilen – Ende
pagenr (text, int)	Seitennummer einstellen bzw. zusätzliches Seitenzeichen ab nächster Seite einführen: #pagenr ("% ", 17)#.
foot	Fußnoten – Anfang
...	
end	Fußnoten – Ende
free (real)	Platz freigehalten (in cm): #free (1.27)#.
page	Neue Seite: #page#

page (int)	Neue Seite mit Seitennummer 17: #page (17)#
linefeed (real)	Zeilenhöhe relativ zum eingeschalteten Schrifttyp verändern: #linefeed (1.25)#.
pagelength (real)	Seitenlänge einstellen (ab nächster Seite in cm): #pagelength (24.0)#.
center	Nachfolgenden Zeilentext zentriert drucken.
right	Nachfolgenden Zeilentext rechtsbündig drucken.
u ... e	(steht für up) Exponent schreiben: #u#123#e#.
d ... e	(steht für down) Index schreiben.
start (real, real)	Schriftfeld (linke obere Ecke) einstellen: #start (1.0, 2.0)#.
-----	-----
b	Zwei Zeichen übereinander drucken.
bottom off	Schaltet Fußzeilen aus.
bottom on	Schaltet Fußzeilen ein.
bpos (real, real)	Der Text zwischen den angegebenen Tabellenpositionen wird im Blocksatz gedruckt.
clearpos	Löscht alle Tabellenpositionen.
clearpos (real)	Löscht die angegebene Tabellenposition.
columns (int, real)	Formatieren von Spalten mit Zwischenraum: #columns (3, 1.0)#, 3 Spalten mit 1 cm Zwischenraum.
columnsend	Beendigung der Spaltenformatierung
count	Interner Zähler für Fußnoten wird eingesetzt (pageform).
count (text)	Wie oben, aber der Wert des internen Zählers wird vermerkt: #count ("neue Zahl")#.
count per page	Interner Zähler beginnt bei jeder Seite mit 1.
cpos (real)	Zentrierende Tabellenposition
dpos (real, text)	Um den angegebenen Text zentrierende Tabellenposition, meist Dezimalzeichen: #dpos (13.0, 2.)#
fillchar (text)	Zwischenräume zwischen Tabellenpositionen werden mit dem angegebenen Text beim Drucken ausgefüllt. Beachten Sie, daß das Ausschalten der Füllzeichen durch 'niltext' erfolgt.
goalpage (text)	Stelle, auf die obige Anweisung verweist: #goalpage ("1.Kapitel")#.
head off	Schaltet Kopfzeile(n) aus.
head on	Schaltet Kopfzeile(n) ein.

ib	Anfang eines Stichworts oder einer Kapitelüberschrift kennzeichnen (Ablegen in Indexdatei mit Zusatz '.i1'): #ib#ein Stichwort oder eine Kapitelüberschrift#ie#
ib (int)	Wie oben, jedoch wird Stichwort in angegebener Indexdatei abgelegt.
ib (int, text)	Wie oben, jedoch erhält Eintrag in der Indexdatei den angegebenen Text an die Seitennummer angefügt.
ie	Beendigung der Stichwortmarkierung.
ie (int)	Wie oben (int – Angabe muß der in der ib – Anweisung entsprechen).
ie (int, text)	Wie oben, jedoch wird die Textangabe hinter das markierte Stichwort angefügt.
lpos (real)	Linksbündige Tabellenposition.
mark (text, text)	Markierung rechts und links neben der Schreibfläche ein –/ausschalten.
pageblock	Einschalten des vertikalen Blocksatzes. Falls eingeschaltet, kann mit 'pageform' auch über das (rechnerische) Seitenende formatiert werden.
rpos (real)	Rechtsbündige Tabellenposition.
setcount (int)	Zählerwert setzen: #setcount (17)#
table	Anfang einer Tabelle.
...	
table end	Ende einer Tabelle.
topage (text)	Seitenverweis (die Seitennummer, auf die verwiesen wird, wird eingesetzt): #topage ("1.Kapitel")#
value	Letzter Zählerwert wird eingesetzt.
value (text)	Wie oben, jedoch wird ein vermerkter Zählerwert eingesetzt: #value ("Vermerk")#

## 5.10. Fehlersituationen und Abhilfe

Was können Sie machen, wenn

bestimmte Anweisungen, die den Gesamttext betreffen, erst ab der zweiten Seite wirksam werden?

Textkosmetik – Anweisungen, die ab der ersten Seite für den ganzen Text gelten sollen, müssen Sie als erstes, d.h. in die erste Zeile einer Datei, schreiben. Dies bezieht sich u.a. auf 'pagelength', 'start', 'block', 'pageblock' etc., die noch vor #head# – oder #bottom# – Anweisungen gesetzt werden müssen.

sich der Cursor nicht mehr bewegen läßt?

Eine Möglichkeit besteht darin, daß Sie versehentlich die 'STOP' –Taste (= 'CTRL a' gleichzeitig, d.h. Anhalten der Bildschirmausgabe) betätigt haben. In diesem Fall drücken Sie die 'WEITER' –Taste ('CTRL c' gleichzeitig, d.h. Bildschirmausgabe fortführen). Alle Tastenanschläge, die Sie in der Zwischenzeit vollzogen haben, werden jetzt ausgeführt.

Eine andere Möglichkeit wäre, daß Sie Ihre Datei/Task nicht ordnungsgemäß verlassen haben. Versuchen Sie über die 'SV' –Taste und 'ESC h' wieder auf die Monitor –Ebene zu gelangen, so daß Sie dann auf die Aufforderung 'gib kommando' hin, wieder in Ihre Datei gelangen können.

Sie nur einen Dateiausschnitt löschen, duplizieren oder mit 'lineform' bearbeiten möchten?

Der betreffende Ausschnitt muß markiert werden. Zum Löschen benutzen Sie die Tasten 'ESC RUBOUT'. Der Ausschnitt ist hiermit aber noch nicht 'vollständig verschwunden', sondern Sie können ihn mit 'ESC RUBIN' an gleicher oder anderer Stelle wieder hervorbringen, so lange bis Sie die Tasten erneut benutzen.

Das Duplizieren eines Textbereiches erfolgt nach dem Markieren durch das Betätigen der Tastenfolge 'ESC d'. Hierbei bleibt der Originaltext erhalten und kann beliebig oft dupliziert werden. Den duplizierten Text holen Sie sich mit 'ESC g' an die gewünschte Stelle in Ihrer Datei.

Möchten Sie 'lineform' nur auf einen Ausschnitt anwenden, markieren Sie diesen und geben nach 'ESC ESC' das Kommando 'lineform'.

beim Drucken die letzte bzw. die letzten beiden Zeilen auf einer gesonderten Seite ausgedruckt werden?

a) Sie müssen die Fonttabelle noch einrichten.

oder

b) Sie setzen die #pageblock# – Anweisung zu Beginn des Textes und "stauen" um zwei Zeilen.

oder

c) Sie wählen eine kleineren Schrifttyp.

in Ihrer Datei die Meldung

```
FEHLER: FILE-Überlauf
```

```
gib kommando:  
edit ("dateiname")
```

erscheint und das Abschicken des Kommandos mittels der 'CR' – Taste aber nur zu einer identischen Meldung (siehe oben) führt?

Wenn Sie auf dem oben dargestellten Weg nach mehrmaligem Versuchen nicht wieder in Ihre Datei gelangen, haben Sie die Möglichkeit, mit dem Kommando

```
gib kommando:  
reorganize ("dateiname")
```

Ihre Datei neu zu 'organisieren', um 'Lücken', die durch Einfügen oder Löschen entstanden sind, zu eliminieren. Die Datei beansprucht dann in der Regel auch weniger Speicherplatz.

Sind Sie wieder in Ihrer Datei, empfiehlt es sich, die große Datei in mehrere kleine aufzuteilen. Entweder Sie halbieren oder (besser) Sie dritteln Ihre Textdatei und verteilen den Text auf zwei bis drei Dateien. Zukünftig sollten Sie es sich dann bei umfangreichen Texten zur Regel machen, nur ein logisch zusammenhängendes Kapitel in einer Datei abzulegen. Sie sollten Ihre Datei nur so groß halten, daß Ihnen noch genügend Raum zur Verfügung steht, Prozeduren wie z.B. 'pageform' durchzuführen, durch die sich der Umfang einer Datei (z.T. wesentlich) vergrößert.

bei Ihrem Text einige Wörter mit extrem großem Zwischenraum gedruckt wurden?

Sie haben in diesem Fall wahrscheinlich vergessen, die Absatzmarkierung zu setzen, so daß aufgrund der #block#-Anweisung ein rechter Randausgleich erfolgte, den Sie an dieser Stelle nicht wünschten.

eine Überschreibung erfolgt ist bzw. wenn 'lineform' eine Überschreibung meldet?

Eine solche Überschreibung kann auftreten, wenn Sie mit einer besonders großen Type in Fettdruck (z.B. triumb14) schreiben (vgl. Sie hierzu auch Kapitel 5.2.6.). Um dies zu verhindern, können Sie zum einen die Anzahl der Blanks zwischen den einzelnen Gliedern erhöhen oder zum anderen auch die #type#-Anweisung in die vorhergehende Zeile setzen (n i c h t d i r e k t v o r d e n betreffenden Text).

nach der erfolgten Prozedur 'lineform' Ihre Tabellen durcheinander geraten sind?

Wenn Sie Tabellen schreiben, müssen Sie nach jeder Zeile die Absatzmarke setzen. Sollen bei einigen Tabellenpositionen Spalten leer bleiben, müssen Sie für diese Positionen ein geschütztes Blank einsetzen.

bei der Prozedur 'pageform' das Seitenende nicht interaktiv verschoben werden kann, obwohl am Anfang der Datei eine #pageblock#-Anweisung gegeben wurde?

Eine #page#-Anweisung im Text bewirkt, daß Sie an dieser Stelle das Seitenende nur bestätigen, die Anweisung löschen oder als dritte Möglichkeit, die Seitenformatierung abbrechen können. Für den Fall, daß die #page#-Anweisung nicht mehr zutrifft bzw. falsch gesetzt wurde, sollten Sie die Anweisung löschen und das Seitenende interaktiv plazieren.

Sie bei der Erstellung eines Inhaltsverzeichnisses oder eines Indexes keine Füllzeichen zwischen Text und Seitenangabe haben möchten?

In diesem Fall gehen Sie mit 'ESC ESC' in den Kommandozustand und ändern mit Hilfe von CA (Change All) die Füllzeichen in Leerzeichen um. Verwenden Sie nicht nur einen Punkt, da sonst auch die Punkte zwischen den Ziffern der Kapitelnumerierung verschwänden. Es kann bei einer ungeraden Anzahl von Füllzeichen notwendig sein, nachträglich noch einige Punkte entfernen zu müssen.

```
qtb kommando: ". ." CA ""
```

## TEIL 6: Spezialitäten

### 6.1. Notizbuch

*Das Notizbuch erlaubt es u.a., Fehlermeldungen zwischenzeitig zu speichern und am Ende einer Verarbeitung die Fehlermeldungen zusammen mit dem bearbeiteten Text im Fenstereditor anzuzeigen.*

Das Notizbuch wird von den Programmen 'lineform' und 'pageform' zum Sammeln von Warnungen und Fehlermeldungen verwendet. Wenn das Fenster des Notizbuches an Bildschirm eröffnet ist, können Sie es handhaben wie das gewohnte Editorfenster.

Falls Sie das Notizbuch selbst beim Editieren für Notizen verwenden möchten, drücken Sie statt **ESC** **e** für den Fenstereditor die Tasten **ESC** **n** an beliebiger Stelle des Bildschirms. Durch dieses Kommando wird das Notizbuch statt einer Datei gezeigt. Sie sparen somit die Eingabe eines Dateinamens und können direkt mit allen vorgestellten Editorfunktionen im Notizbuch arbeiten.

..... handbuch Teil 6 ..... Zeile 56

```
#kap("6.2. EUMEL-Zeichensatz")#
```

```
#free(1.0)#
```

```
#zus#
```

```
#cornerl("-5.0")#
```

Das EUMEL-System definiert einen Zeichensatz, der gewährt auf allen Maschinen überall gleich codiert werden. Dadurch ist Dateien und Programme ohne Konvertierungen zwischen EUMEL-System unterschiedlicher Hersteller zu übertragen. Der #ib#EUMEL-dem ASCII-Zeichensatz (DIN 66 003) mit Erweiterungen.  
#box3("T", "2", "115.0")#

..... notebook ..... Zeile 1

FEHLER Zeile 55: Modifikation nicht angeschaltet bei off: b

>>> Anweisung in angegebener Zeilennummer überprüfen

WARNUNG Zeile 55: Umschaltung auf gleichen Schrifttyp: trium8

>>> Schrifttyp wurde darum nicht verändert!

WARNUNG Zeile 75: Überschreibung nach >#ib(9)#6.2.< Fehlende

>>> Bitte fehlende Leerzeichen einfügen

## 6.2. EUMEL – Zeichensatz

*Das EUMEL – System definiert einen Zeichensatz, der gewährleistet, daß Zeichen auf allen Geräten gleich codiert werden. Dadurch ist es z.B. möglich, Dateien und Programme ohne Konvertierungen zwischen EUMEL – Systemen unterschiedlicher Hersteller zu übertragen. Der EUMEL – Zeichensatz beruht auf dem ASCII – Zeichensatz (DIN 66 003) mit Erweiterungen.*

Die Darstellung der einzelnen Zeichen hängt vom Endgerät ab. Die hier aufgeführten Zeichen sind i.A. auf allen Geräten vorhanden. Ein erweiterter Zeichensatz (mit mathematischen, diakritischen und griechischen Zeichen) ist nur auf **Spezialgeräten** verfügbar und wird deshalb hier nicht angegeben.

Beispiele zum Lesen der Tabelle:

code (" ") -> 32

code ("m") -> 109

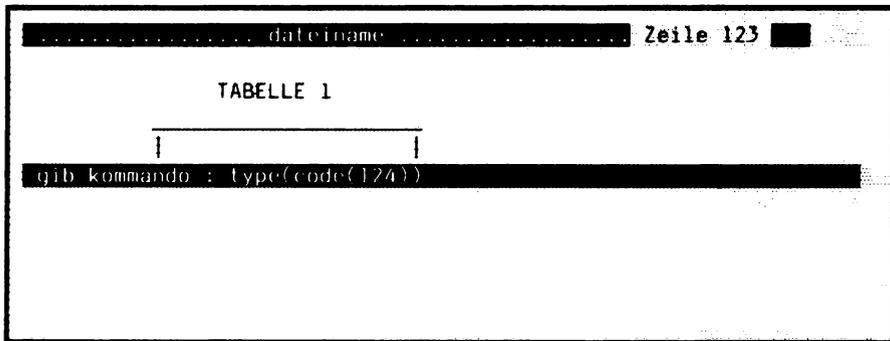
	0	1	2	3	4	5	6	7	8	9
3			SP	!	"	#	\$	%	&	'
4	(	)	*	+	,	-		/	0	1
5	2	3	4	5	6	7	8	9		;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[	\	]	^	_	`	a	b	c
10	d	e	f	g	h	i	j	k	l	m
11	n	o	p	q	r	s	t	u	v	w
12	x	y	z		}	~				
13										
	.									
	.									
20										
21					Å	Ö	Ü	ä	ö	ü
22	k	-	#		SP					
23										
24										
25		ß								

Anmerkungen:

- 1) SP bedeutet Leerzeichen ("blank").
- 2) Die Zeichen 'k', '-' und 'SP' mit den Codes 220, 221, 223 werden für die Zwecke der Textkosmetik benötigt (Trenn- 'k' bei der Umwandlung von 'ck' in 'kk'; Trennzeichen; geschütztes Leerzeichen).
- 3) Das Zeichen '#' (Code 222) ist druckbar, während das Zeichen '#' (Code 35) nicht druckbar ist (Einleitungszeichen für Anweisungen der Textkosmetik und Drucker).
- 4) Das Zeichen SP (Code 223) wird zur besseren Identifizierung invers oder als Unterstreichungsstrich auf dem Terminal dargestellt. In einem Ausdruck erscheint es als ein Leerzeichen.

Falls Sie Zeichen ausgeben möchten, die nicht auf der Tastatur sind, müssen Sie den Code der gewünschten Zeichen zu Hilfe nehmen.

Bewegen Sie den Cursor dazu an die Stelle der Datei, an die das Sonderzeichen geschrieben werden soll und geben Sie nach `ESC` `ESC` ein:



## 6.3. Sortier – Programme

*Es stehen zwei verschiedene Sortier – Programme zur Verfügung: 'sort' (Sortierung nach ASCII – Reihenfolge) und 'lex sort' (Sortierung nach deutschem Alphabet).*

Die Sortierprogramme sortieren eine Datei zeilenweise.

Beispiel:

```
..... dateiname ..... Zeile 1  
Berta ist eine Frau.  
Adam ist ein Mann.  
...
```

```
gib kommando :  
sort ("dateiname")
```

```
..... dateiname ..... Zeile 1  
Adam ist ein Mann.  
Berta ist eine Frau.  
...  
.....
```

Dabei werden die Zeilen – Anfänge solange zeichenweise miteinander verglichen, bis ein Unterschied auftritt und dann ggf. umgeordnet. Werden zwei ungleich lange Zeilen (Anzahl Zeichen/Zeile) miteinander verglichen, dann kann man sich die kürzere Zeile mit Leerzeichen auf die Länge der längeren Zeile verlängert denken.

Die Reihenfolge, in der die Zeilen sortiert werden, erfolgt nach dem ASCII – Zeichensatz in aufsteigender Reihenfolge (vergl EUMEL – Zeichencode):

- das Leerzeichen
- einige Sonderzeichen
- die Ziffern
- einige Sonderzeichen
- große Buchstaben
- einige Sonderzeichen
- kleine Buchstaben
- einige Sonderzeichen
- die Umlaute und ß

Das bedeutet, daß z.B. folgendermaßen sortiert wird:

```
..... dateiname ..... Zeile 1  
Adam  
Ball  
Zuruf  
aber das ist ein Satz  
niemals  
Überlauf  
.....
```

Um zu erreichen, daß große und kleine Buchstaben gleichwertig behandelt werden, kann man das Kommando

```
gib kommando :  
lex sort ("dateiname")
```

geben.

In diesem Fall würde die sortierte Datei folgendermaßen aussehen:

```
..... dateiname ..... Zeile 1  
aber das ist ein Satz  
Adam  
Ball  
niemals  
Überlauf  
Zuruf
```

Man beachte, daß der Umlaut 'Ü' wie 'Ue' behandelt wird (für die restlichen Umlaute gilt eine analoge Behandlung; ebenso wird 'ß' wie 'ss' behandelt). Weiterhin werden alle Sonderzeichen bei der Sortierreihenfolge ignoriert.

## 6.4 Fonttabellen

*Eine Fonttabelle enthält Angaben zu den druckbaren Zeichen.*

Die Einstellung einer Fonttabelle<sup>1)</sup> erfolgt automatisch beim Einrichten der Drucker-Task (siehe Anhang). Um den Namen der in der Task eingestellten Fonttabelle zu erhalten geben Sie ein:

```
gib kommando:  
put(fonttable)
```

Die Ausgabe liefert den Namen der in der Task eingestellten Fonttabelle.

```
gib kommando:  
put(fonttable)  
agfa9  
gib kommando:
```

---

1) Fonttabelle: Beschreibung der druckbaren Schrifttypen.

Um eine neue oder andere Fonttabelle einzustellen, etwa weil verschiedene Drucker benutzt werden können, geben Sie das 'fonttable' Kommando mit dem Namen der gewünschten Fonttabelle als Parameter an:

```
gib kommando:  
fonttable("name der fonttabelle")
```

Eine weitergehende Beschreibung der eingestellten Fonttabelle erhalten Sie durch das Kommando 'list fonts':

```
gib kommando:  
list fonts
```

Durch dieses Kommando erhalten Sie in das Notizbuch eine Aufstellung der Schrifttypen mit Angaben zu Namen der verfügbaren Schrifttypen in der Fonttabelle, Größenangaben zu den Schrifttypen etc.

```
..... notebook ..... Zeile 1
FONTABELLE : "agfa9";
x einheit = 160.0;
y einheit = 160.0;

FONT : "micro", "elanlist", "bulletin22";
  einrueckbreite = 20;
  durchschuss = 7;
  fonthoehe = 30;
  fonttiefe = 8;
  groesserer font = "";
  kleinerer font = "";

FONT : "trium10";
  einrueckbreite = 31;
  durchschuss = 6;
  fonthoehe = 54;
  fonttiefe = 15;
  groesserer font = "trium12";
  kleinerer font = "trium8";
```

Anmerkung:

- Falls mehrere Namen für einen Schrifttyp angegeben sind, können Sie einen beliebigen dieser Namen in der #type# - Anweisung benutzen.
- Größenangaben sind in 'Mikroschritten', d.h. den kleinstmöglichen Schritten des jeweiligen Druckers angegeben und nicht etwa in mm.
- Weitere Informationen entnehmen Sie ggf. dem Systemhandbuch.

## 6.5 Syntax der Kommandos

### *code*

TEXT PROC code (INT CONST zahl)

Wandelt 'zahl' in ein Zeichen um. Falls die Zahl kleiner als 32 oder größer als 254 ist, (siehe Codetabelle) muß mit unerwarteten Ergebnissen gerechnet werden.

```
type(code(92))
```

INT PROC code (TEXT CONST zeichen)

Wandelt 'zeichen' in die zugehörige EUMEL – Codierung um. Falls mehr als ein Zeichen angegeben wird, ist das Resultat ' – 1'.

```
put(code(92))
```

### *list fonts*

PROC list fonts

Listet die Fonts der eingestellten Tabelle ins *notebook*.

PROC list fonts (TEXT CONST fonttable name)

Listet die Fonts der angegebenen Fonttabelle ins *notebook*. Die vorher eingestellte Fonttabelle bleibt jedoch weiter eingestellt.

```
list fonts ("fonttab.alternativ")
```

**lex sort**

PROC lex sort (TEXT CONST datei)

Zeilenweise Sortierung nach (deutscher) lexikographischer Reihenfolge nach DIN 5007.

```
lex sort ("telephonliste")
```

PROC lex sort (TEXT CONST datei, INT CONST anfang)

Wie 'lex sort', jedoch wird bei der Sortierung bei 'anfang' jeder Zeile begonnen.

```
lex sort ("liste",20)
```

**sort**

PROC sort (TEXT CONST datei)

Die Prozedur 'sort' sortiert die Datei 'datei' zeilenweise. Die Sortierung erfolgt nach der Ordnung, die der EUMEL-Zeichencode vorschreibt. Beispielsweise werden Zeilen ("Sätze"), die mit Ziffern beginnen, vor Sätzen, die mit Buchstaben anfangen, eingeordnet. Sätze, die mit großen Buchstaben beginnen, werden vor Sätzen mit kleinen Buchstaben einsortiert. Weiterhin werden die Umlaute und das "ß" nach allen anderen Buchstaben eingeordnet.

```
sort ("liste")
```

PROC sort (TEXT CONST datei, INT CONST anfang)

Sortiert eine Datei wie obige Prozedur, jedoch wird bei der Sortierung nicht der Anfang eines Satzes beachtet, sondern die Position 'anfang'.

```
sort ("liste",10)
```

**type**

PROC type (TEXT CONST zeichenkette)

Fügt 'zeichenkette' in die aktuelle Position der editierten Datei ein. Besonders nützlich in Verbindung mit der Prozedur 'code', um nicht auf der Tastatur enthaltene Zeichen in den Text zu bringen.

```
type(code(200))
```

## Aufbau und Installation

*Diese Installationsanleitung dient nur als Beispiel und soll nicht die Anleitung zu dem von Ihnen benutzten Gerät ersetzen. Insbesondere die hier beschriebene Möglichkeit der Partitionierung ist kein Standard!*

### Allgemeines über das Betriebssystem EUMEL

Zum besseren Verständnis des Installierungsvorganges sei hier kurz der Aufbau des Betriebssystems EUMEL erläutert (Der EUMEL-Kenner mag diesen Abschnitt überspringen) :

Das System besteht im wesentlichen aus den folgenden Komponenten :

- SHard (Software - **Hardware** Interface)
- Systemkern (EUMEL - 0)
- darauf aufbauende Systemteile

Das **SHard** ist der hardwareabhängige Teil des Betriebssystems. Dieser Teil ist verantwortlich für den Informationsfluß zwischen der virtuellen Maschine EUMEL-0 und den einzelnen Hardwarekomponenten (Tastatur, Diskettenlaufwerk, Schnittstellen, usw.).

Der **Systemkern** (auch mit EUMEL-0 oder Urlader bezeichnet) ist der prozessorabhängige Teil des Systems. Er bestimmt im wesentlichen die Leistung des Betriebssystems, da er als virtueller Prozessor den **Befehlsumfang** für den ELAN-Compiler definiert. Dieser **Befehlsumfang** wird dann auf den tatsächlichen **Befehlsvorrat** des hardwaremäßig vorhandenen Prozessors abgebildet.

Die auf dem Kern (EUMEL-0) **aufbauenden Systemteile** sind hardware- und prozessorunabhängig. Sie beinhalten den ELAN-Compiler und alle Tasks, Texte, insertierten ELAN-Pakete, benannte und unbenannte Datenräume eines EUMEL-Systems. Diese Systemteile bilden zusammen mit dem Systemkern EUMEL-0 den **EUMEL-Hintergrund**, d.h. EUMEL-0 ist Teil des EUMEL-Hintergrundes. Momentan werden je nach Rechnertyp Diskette und Festplatte als Hintergrundspeichermedium für EUMEL unterstützt.

Der Begriff EUMEL-Hintergrund (HG) resultiert aus dem Konzept des virtuellen Speichers. Bei diesem Konzept wird der RAM-Speicherbereich der Hardware nur als Pufferbereich nach dem Demand-Paging-Verfahren benutzt, mit Ausnahme der residenten Systemteile (SHard und EUMEL-0). Das macht den Benutzer bezüglich seiner Programme und Daten unabhängig von der eigentlichen Größe des RAM-Speichers. Diese bestimmt lediglich den Durchsatz (Performance) des Systems, d.h. je größer der RAM-Bereich Ihres Rechners ist, desto schneller arbeitet das EUMEL-System.

EUMEL-0 befindet sich auf der ersten Hintergrunddiskette. Das Laden von EUMEL-0 und des restlichen EUMEL-Hintergrundes kann daher in einem Arbeitsgang geschehen.

Der Auslieferungs-Hintergrund ist noch nicht konfiguriert und stellt ausschließlich die im Handbuch beschriebenen Leistungen des Betriebssystems zur Verfügung. Er besteht im wesentlichen aus insertierten ELAN-Paketen, die den Leistungsumfang des Systems bestimmen (Single- oder Multiuser, mit oder ohne Textverarbeitung). Steht dieser Hintergrund auf mehreren Disketten (Multiuser-Hintergründe), dann sind diese fortlaufend numeriert. Die erste Hintergrunddiskette hat die Nummer 0, um daran zu erinnern, daß sich auf dieser Diskette auch der Systemkern EUMEL-0 befindet.

Ein Hintergrund kann natürlich auch die Systemsicherung eines größeren Systems mit z.B. mehreren Megabytes sein. Sollten Sie später einmal Ihr bestehendes System durch ein anderes ersetzen wollen (Hintergrund überschreiben), so brechen Sie während des Hochfahrens beim Speichertest das System durch Tastendruck ab und laden durch Anwahl von 2 'neuen Hintergrund vom Archiv laden' im Startdialog das neue System.

**ACHTUNG:** Dabei gehen alle Daten des alten Systems unwiederbringlich verloren! (Siehe dazu auch 'Laden eines EUMEL-Hintergrundes'.)

Zur besseren Verständlichkeit sollen an dieser Stelle noch die Begriffe 'Systemstart' und 'Systemabschaltung' des EUMEL – Systems erläutert werden :

### **Systemstart :**

Wenn ein EUMEL – System gestartet wird (auch 'Hochfahren' genannt), und dies gilt auch für kleinere Diskettensysteme wie den 'Generierungseumel' (siehe Seite 4), wird zunächst das SHard geladen; dazu erscheint auf dem Bildschirm eine entsprechende Meldung. Das SHard versucht nun den Systemkern vom Archivmedium (in der Regel Diskette) zu laden. Ist keine entsprechende Diskette eingelegt, so wird versucht, EUMEL – 0 vom Hintergrundmedium (Festplatte) zu laden.

Danach wird EUMEL – 0 aktiv; auf dem Bildschirm erscheinen Angaben zu verfügbaren Kanälen, RAM – und Hintergrundspeicher – Größe. Dann führt EUMEL – 0 einen Speichertest durch, was daran zu erkennen ist, daß eine Folge von Sternchen (\*) auf den Schirm geschrieben wird. Wird währenddessen eine beliebige Taste gedrückt, dann gelangt man nach dem Speichertest in den 'Startdialog'.

Nach dem Speichertest bzw. nach Wahl von  'Systemstart' im Startdialog wird der Hintergrund aktiviert, was je nach Größe des Systems und Art des Hintergrundmediums unterschiedlich lange dauert.

### **Systemabschaltung :**

Bevor man den Rechner ausschaltet, sollte jedes EUMEL – System ordnungsgemäß abgeschaltet werden. Das geschieht durch den Befehl 'shutup', den man in der Multiuser – Version von EUMEL im privilegierten Zweig des Taskbaumes erteilen muß. Nur dann ist gewährleistet, daß der aktuelle Zustand Ihres Systems gesichert ist.

Andernfalls gilt das System als abgebrochen, was daran zu erkennen ist, daß sich das System beim nächsten Systemstart mit 'RERUN' meldet. Dann kann nur am letzten Fixpunkt wieder aufgesetzt werden, und Ihre in letzter Zeit (normalerweise ca. 15 Minuten) gesammelten Daten können verloren sein.

## Installation des EUMEL – Systems

### A: Erforderliche Disketten

- EUMEL – Systemdiskette : "Generierungseumel XY"<sup>1)</sup>
- EUMEL – Hintergrunddisketten : "HG0" ... "HGn"
- EUMEL – Archivdisketten : "std..."
- EUMEL – Archivdiskette : "XY" (Typabhängig)
- MS – DOS – Diskette : "EUMELstart"

Die Diskette "Generierungseumel XY" ist ein kleines, jedoch vollständiges EUMEL – System. Auf diesem System laufen nach dem Hochfahren Programme ab, die im Dialog mit dem Benutzer die Generierung einer oder mehrerer EUMEL – Partitionen ermöglichen, diese Bereiche bezüglich schlechter Spuren untersuchen und das SHard auf der jeweiligen Partition installieren. **Bei der Generierung darf diese Diskette nicht schreibgeschützt sein !**

Die Hintergrunddisketten "HG0" ... "HGn" beinhalten das eigentliche Betriebssystem EUMEL. Es sind dies der Systemkern EUMEL – 0 und die darauf aufbauenden Systemteile (siehe Seite 1).

Die Standardarchivdisketten "std..." beinhalten ELAN – Programmpakete und Fonttabellen, die Sie nach erfolgter Installation des Betriebssystems z.B. zum Zwecke einer Druckerinstallation oder erweiterter Rechenfunktionen benötigen werden. Sie finden dazu detaillierte Informationen in Ihrem Benutzer – und Systemhandbuch.

Die Archivdiskette "XY" beinhaltet ELAN – Programmpakete, die Funktionen, die nicht zu den Standardleistungen von EUMEL bzw. der vorliegenden SHard – Version zählen.

---

1) XY steht für die Typbezeichnung eines Rechners wie: XT, AT, M24 usw.

Die Anzahl der ausgelieferten Disketten ist auch typabhängig, da z.B. 'EUMELstart' nur benötigt wird, falls tatsächlich eine Partitionierung möglich ist.

Die Diskette "EUMELstart" ist eine MS-DOS Diskette und beinhaltet Kommando-Dateien. Falls Sie auch eine MS-DOS Partition eingerichtet haben, dann gewährleisten diese das Aktivieren einer EUMEL Partition mit gleichzeitigem Systemstart von MS-DOS aus.

#### B: Partitionieren der Festplatte / Installation des SHard

Wenn Sie bereits ein Betriebssystem auf Ihrer Festplatte installiert haben, müssen Sie darauf achten, daß noch ausreichend Platz für ein EUMEL-System übrig ist. Da z.B. MS-DOS standardmäßig die gesamte Festplatte belegt, muß dieses System gesichert, mit dem MS-DOS-Kommando 'fdisk' gelöscht und entsprechend kleiner neu eingerichtet werden. Sie können auch bei der EUMEL-Installation alle bereits bestehenden Systeme löschen; dazu bietet Ihnen der Generierungseumel die Option *Löschen der gesamten Partitionstabelle* an. Dabei gehen jedoch alle bestehenden Daten verloren. Achten Sie also darauf, daß sie alle Daten vorher gesichert haben !

Hinweis: Bei Festplatten mit einer Kapazität über 32 Megabyte kann die Installation des SHard zu Problemen führen (Fehlermeldung *Platte kann nicht gelesen werden* bei der Suche nach schlechten Spuren). Richten Sie daher Ihre EUMEL-Partition(en) auf den ersten 32 Megabyte ein.

Um nun die Partitionierung für Ihr EUMEL-System vorzunehmen, legen Sie die Diskette 'Generierungseumel' in das Boot-Laufwerk. Sollte die Diskette mit einer Schreibschutzmarke versehen sein, dann müssen Sie diese vorher entfernen.

Schalten Sie nun den Rechner ein bzw. betätigen Sie den Tastatur-RESET, wenn Ihr Gerät bereits eingeschaltet ist, indem Sie die Tasten **CTRL** **ALT** **DEL** gleichzeitig betätigen.

Der Generierungseumel meldet sich zunächst mit folgender SHard – Meldung:

Setup - SHard für EUMEL auf XY und Kompatiblen V.x.x  
Copyright (C) 1985,86  
EUMEL wird vom Hintergrund geladen

Danach erscheinen die EUMEL – 0 Meldungen zu HG –, RAM – und Pufferkapazität, bezogen auf den Diskettenhintergrundes des Generierungseumel.

**ACHTUNG:** Der Generierungseumel soll während des Speichertests (Sternchen) nicht unterbrochen werden. Geschieht dies versehentlich doch, dann fahren Sie fort, indem Sie im Startdialog die Taste **[1]** für Systemstart betätigen. Dann wird normal mit der Installation fortgefahren. Wählen Sie keinesfalls **[2]** 'neuen Hintergrund vom Archive laden', solange sich die Diskette 'Generierungseumel' im Archivlaufwerk befindet.

Nach dem Hochfahren des 'Generierungseumel' wird Ihnen eine Tabelle angezeigt, der Sie entnehmen können, ob bereits Partitionen auf der Festplatte eingerichtet und wie diese spezifiziert sind.

Angezeigt werden neben Größe, Start – und Endspur der einzelnen Partitionen auch eine Typ – Nr.; für EUMEL – Partitionen werden in aufsteigender Reihenfolge die Typ – Nummern 69 bis 72, für MS – DOS je nach Größe der eingerichteten Partition die Nummer 1 oder 4 vergeben. Die Typ – Nummern der eingerichteten Partitionen sollten Sie sich merken, da diese Angaben später von Bedeutung sind, wenn das Gesamtsystem für Partitionswechsel vorbereitet wird. Richten Sie mehrere EUMEL – Partitionen ein, dann können Sie diese ausschließlich über die Typ – Nummern identifizieren !

Außerdem wird die gerade aktive Partition durch einen entsprechenden Eintrag in der Tabelle kenntlich gemacht. **"Aktiv" ist die Partition, die nach dem nächsten Einschalten des Rechners bzw. nach dem nächsten Tastatur – RESET gebootet würde.**

Sie können nun eine der folgenden Funktionen auswählen :

- Generieren einer EUMEL – Partition
- Aktivieren einer Partition
- Löschen einer EUMEL – Partition
- Löschen der gesamten Partitionstabelle
- Generierung beenden

Beim Generieren einer EUMEL – Partition werden lediglich Angaben zu Größe und Startzylinder abgefragt. Dafür werden Vorgaben gemacht, die Sie bestätigen können, indem Sie die  CR Taste betätigen.

Beim Neueinrichten orientiert sich die Vorgabe für die Partitionsgröße an dem größten zusammenhängenden Freiraum auf Ihrer Platte, die Vorgabe für den Startzylinder orientiert sich dann an dem kleinsten zusammenhängenden Freiraum, auf dem eine Partition der gewählten Größe eingerichtet werden kann.

**ACHTUNG:** Soll eine EUMEL – Version installiert werden, die nur 16 Megabyte verwalten kann (1.7.3 bzw. 1.8.0), dann darf die Partition nicht größer eingerichtet werden. Es kann hier keine generelle Aussage über die Anzahl der zu reservierenden Spuren gemacht werden, da sehr verschiedene Plattenaufteilungen angeboten werden. Entnehmen Sie die entsprechenden Angaben bitte dem Festplatten – Handbuch Ihres Hardwareherstellers.

Das Löschen einer EUMEL – Partition geschieht nur logisch, nicht physisch, das heißt es wird nur der Eintrag in der Partitionstabelle gelöscht. Sollten Sie später an gleicher Stelle eine neue Partition einrichten und vorher diesen Bereich physisch nicht überschrieben haben, dann würde nach dem Hochfahren des Rechners das alte System wieder gestartet. Die Meldung 'kein EUMEL – System gefunden'(siehe unten) erscheint dann nicht.

Haben Sie Ihre EUMEL – Partition(en) eingerichtet, dann achten Sie darauf, daß Sie Ihren Generierungseumel ordnungsgemäß wieder verlassen, da es sich hier, wie bereits erwähnt, um ein vollständiges EUMEL – System mit Fixpoint/Rerun – Logik handelt. Das 'shutup' wird automatisch ausgeführt, wenn Sie die Funktion '0. Generierung beenden' wählen.

Wenn die Meldung 'ENDE' auf Ihrem Bildschirm erscheint, ist dieser Schritt der Installation beendet. Sie haben nun eine (oder mehrere) EUMEL – Partitionen eingerichtet und das SHard installiert. Bitte entfernen Sie jetzt die Diskette 'Generierungseumel' aus dem Diskettenlaufwerk.

## Laden eines EUMEL – Hintergrundes

Im nächsten Schritt wird auf Ihrer Festplatte das EUMEL – System installiert, d.h. es wird ein Hintergrund auf der Festplatte erzeugt.

Dazu müssen Sie nach dem ordnungsgemäßen Beenden des Generierungseumel und Entfernen der Diskette aus dem Laufwerk den Tastatur – RESET betätigen. Dies geschieht entweder durch gleichzeitiges Betätigen der Tasten **CNTL** **ALT** **DEL** auf der Tastatur oder durch AUS – und wieder EIN – Schalten des Rechners (Bitte warten Sie einen Augenblick zwischen dem AUS – und EIN – Schalten).

Das System meldet sich nach kurzer Zeit mit folgender SHard – Meldung:

```
SHard für EUMEL auf XY, V x.x  
Copyright (C) 1985,86  
kein EUMEL-System gefunden
```

Sie legen nun die Hintergrunddiskette HG0 in das Boot – Laufwerk und betätigen eine Taste.

Der Systemkern wird nun geladen und es erscheinen die bereits oben erwähnten Angaben zu HG-, RAM- und Pufferkapazität, sowie zu den angeschlossenen Kanälen, diesmal jedoch bezogen auf die eingerichtete Festplatten-Partition. Während des Speichertests drücken Sie bitte erneut eine Taste, um in den Startdialog zu gelangen und damit zu verhindern, daß EUMEL-0 versucht, das System zu starten. Sollten Sie dies versäumen, so erscheint die Meldung 'HG ungültig'. Sie haben dann erneut die Möglichkeit, durch Betätigen einer Taste in den Startdialog zu gelangen.

Hier wählen Sie den Menüpunkt  2 'neuen Hintergrund vom Archiv laden' und bestätigen die Frage 'Alten Hintergrund überschreiben' mit  j für 'ja'.

Es erscheint ein Zähler auf dem Bildschirm, der die gelesenen Blöcke anzeigt. Verteilt sich Ihr Hintergrund auf mehrere Disketten, dann müssen Sie bei der Frage 'Neues HG-Archiv eingelegt' die nächste Diskette einlegen und mit  j quittieren. Bitte beachten Sie dabei genau die Numerierung der HG-Disketten !

Es können bei beschädigten Disketten Lesefehler auftreten; dann gibt das System eine der Meldungen 'Harter Lesefehler' bzw. 'Softerror' aus. Bei letzterem konnte der entsprechende Sektor nach mehrmaligem Versuch noch gelesen werden. Bei einem harten Lesefehler können Sie die Diskette nicht verwenden.

Wenn alle Disketten eingelesen sind, müssen Sie ein letztes mal den Tastatur-RESET betätigen, um das System zu starten. Vergessen Sie nicht, vorher die Hintergrunddiskette aus dem Diskettenlaufwerk zu entfernen.

Wenn Sie jetzt während des Hochfahrens keine Taste drücken, dann startet der Lader durch und das EUMEL-System meldet sich mit 'System aufgebaut'. Dies dauert beim Auslieferungshintergrund wenige Sekunden, kann jedoch bei größeren Systemsicherungen auch mehrere Minuten in Anspruch nehmen; verlieren Sie dann bitte nicht allzu schnell die Geduld.

Da der Auslieferungs-Hintergrund unkonfiguriert ist, gelangt das System beim ersten Hochfahren nach der Installation sofort in den 'configurator'. Sie müssen jetzt den Kanal 1 entsprechend der vorhandenen Tastatur als "PC.german" oder "PC.ascii" konfigurieren. Sollten Sie eine EUMEL-Version 1.7.3 benutzen und diesen Konfigurationsdatenraum nicht zur Verfügung haben, dann konfigurieren Sie den Kanal 1 als "PC" und Terminal. Näheres dazu finden Sie im Systemhandbuch (Teil 1).

## Die einzelnen Schritte der Installation im Überblick

1. Die Diskette 'Generierungseumel' in das Laufwerk stecken
2. Rechner einschalten oder Tastatur – RESET mit **CTRL**, **ALT** **DEL**
3. EUMEL – Partition einrichten
4. Generierung beenden und auf 'ENDE' – Meldung warten
5. Diskette 'Generierungseumel' entnehmen
6. Tastatur – RESET
7. Die Meldung 'Kein EUMEL – System gefunden' abwarten. Wenn die Meldung 'EUMEL wird vom Hintergrund geladen' erscheint, dann weiter bei 9.
8. Erste Hintergrunddiskette (HG0) einlegen und Taste drücken
9. Beim Speichertest eine Taste betätigen, um in den Startdialog zu gelangen.
10. Menüpunkt **2** anwählen : Neuen Hintergrund vom Archiv laden
11. Eventuell weitere HG – Disketten nach entsprechender Aufforderung einlegen und mit **j** quittieren
12. Tastatur – RESET nach entsprechender Aufforderung
13. Nach dem Hochfahren des Systems Konfiguration lt. Systemhandbuch vornehmen
14. Ggf. in der Task 'SYSUR' ELAN – Pakete für Partitionswechsel insertieren.

Dazu – Falls eine EUMEL – Version 1.7.3 benutzt wird, zunächst das Kommando *free global manager* in der Task 'configurator' absetzen

- Archivdiskette "XY" einlegen und anmelden : *archive °("XY")*
- Datei "XY install" von Archivdiskette lesen :  
*fetch ("XY install", archive)*
- Insertierung starten : *run*

## Druckersoftware einrichten

Um mit Ihrem EUMEL-System einen Drucker betreiben zu können, müssen Sie außer dem Anschluß des Druckers mit einem passenden Kabel auch die passende Software für diesen Drucker zur Verfügung stellen. Zu diesem Zweck dienen die Druckeranpassungen.

Das Standardarchive "std.printer" enthält Druckeranpassungen für die Ansteuerung diverser gebräuchlicher Druckertypen. Soll einer dieser Drucker an das EUMEL-System angeschlossen werden, so muß zuerst eine Task "PRINTER" als Sohntask von "SYSUR" mit dem Supervisor-Kommando

```
begin ("PRINTER", "SYSUR")
```

eingerrichtet werden. In dieser Task müssen dann die folgenden Schritte vollzogen werden:

- Archiv anmelden:  
    archive ("std.printer")
- Druckeranpassung vom Archiv holen:  
    fetch ("printer.druckertyp", archive)
- Zeilennummerngenerierung bei der Insertierung abschalten:  
    check off
- Druckeranpassung insertieren:  
    insert ("printer.druckertyp")

Beispiel:

```
archive ("std.printer")
fetch ("printer.epson.fx", archive);
check off;
insert ("printer.epson.fx")
```

Nach der Insertierung wird zuerst nach dem Druckerkanal gefragt. Dieser sollte mit der Gerätetabelle 'transparent' konfiguriert sein. Dann werden ggf. druckerspezifische Fragen zur Papierbreite, Positionierungsart oder ähnlichem gestellt, die mit 'j' oder 'n' beantwortet werden müssen. Dabei werden alle Alternativantworten zu der jeweiligen Frage hintereinander angeboten, bis eine Alternative mit 'j' beantwortet wird.

Als letzter Schritt kommt die Aufforderung das Archiv mit der passenden Fonttabelle einzulegen. Diese Fonttabelle, eine Beschreibung aller darstellbaren Zeichen in allen druckbaren Schrifttypen, ist meistens auf derselben Diskette wie die Druckeranpassung.

Wenn die Generierung beendet ist, muß im Multi-User Betrieb in allen bestehenden Tasks – insbesondere in der Task 'PUBLIC' – die Fonttabelle mit dem fonttable-Kommando eingestellt werden.

Beispiel:

```
fonttable("fonttab.epson.fx")
```

Von jeder Task aus kann danach mit dem Kommando

```
print ("dateiname")
```

wird eine Datei ausgedruckt werden.

Das Einstellene einer Fonttabelle ist insbesondere Voraussetzung für 'lineform', 'pageform' etc.

Befindet sich keine passende Druckeranpassung für den anzuschließenden Drucker auf dem Standardarchiv "std.printer", so sollte die Druckeranpassung "printer.std" benutzt werden. Diese Druckeranpassung ist eine universelle Druckeranpassung für alle Drucker, die mit ASCII-Code 13 ein 'Carriage Return' (d.h. Bewegung des Druckkopfes an den linken Rand) und mit ASCII-Code 10 eine Zeilenschaltung von 1/6 Zoll vornehmen. Mit ihr kann dann in einem Schrifttyp (entweder 10 oder 12 Zeichen pro Zoll, je nachdem welche Fonttabelle eingestellt ist) gedruckt werden. So erhält man wenigstens eine Minimalansteuerung des Druckers.

## **Druckersoftware im Single – User einrichten**

Die Installation der Druckersoftware im Single – User erfolgt ähnlich wie die im Multi – User. Hier brauchen nur die Schritte durchgeführt zu werden, die im Multi – User in der Task "PRINTER" durchgeführt werden müssen. Eine Task "PRINTER" braucht nicht eingerichtet zu werden.

---

# Index

/	3–40, 57
Abbildungsverzeichnisse	5–81
Abbruch	5–11
Abbruch mit ESC	5–11
Absatz	4–19
ALL	3–38, 47
äquidistanten Schriften	5–20
Anweisung	5–1
Anweisungen für den EUMEL – Drucker	5–87
Anweisungen, Kommentar –	5–87
any	4–61
Archiv	3–23
Archivdiskette	3–26
archive	3–2, 47
Auffüllen von Zeilen	5–13
Aufruf der Textkosmetik – Programme	5–9
Aufruf eines Makros	5–98
Aufzählung	4–21 , 5–27, 90
Aufzählungskriterium	4–21
Ausnahmelexikon	5–112
Ausschalten des Geräts	1–15
autoform	3–3 , 5–13, 19
Automatische Seitenformatierung	5–44
Automatische Silbentrennung	5–19
autopageform	3–3 , 5–3
begin	2–2, 13
begin password	3–3, 43, 47
Betriebssystem	1–2
Blättern	4–27, 60
Blocksatz	5–89
Blocksatz innerhalb einer Spalte	5–35
break	2–2, 13 , 3–2, 48
Breitere Zeilen	4–59
Briefkopf	5–97
brother	3–48
C	4–51, 62

CA	4 – 51, 62
CHANGE	4 – 50
CHANGE ALL	4 – 51
check	3 – 2, 34, 48
clear	3 – 2, 28, 48
clear pos – Anweisung	5 – 38
code	6 – 12
continue	2 – 2, 13
copy	3 – 2, 48
Cursor	4 – 3, 19, 23
D	4 – 48, 62
Datei	1 – 5 , 3 – 12 , 4 – 18
Dezimaltabulator	4 – 34
Doppelblank	5 – 27
DOWN	4 – 48
Druckdatei	5 – 10, 77
Druckerkanal	A – 12
edit	3 – 2, 49 , 4 – 2
Editor ausschalten	4 – 4
EDV – Tastatur	4 – 5
Ein – bzw. Ausschalten der Markierung	4 – 13
Ein – bzw. Ausschalten des Einfügemodus.	4 – 12
Eine Datei drucken	5 – 86
Einfügemodus	4 – 26
Einfügen von Textpassagen	4 – 28
Einfügen von Zeichen	4 – 25
Eingabetaste / Absatztaste	4 – 9
Einrückung	5 – 89
Einrückungautomatik	4 – 21
Einrückungen	5 – 15
Ein – und Ausschalten des Editors	4 – 2
end	3 – 2, 11, 49
enter password	3 – 3, 50
entlade ausnahmen	5 – 113
erase	3 – 2, 50
ESC >	4 – 41, 43
ESC ?	2 – 2
ESC A	4 – 43
ESC – Abbruch	5 – 11
ESC b	2 – 2 , 4 – 41
ESC blank	4 – 43

## Index

---

ESC c	2-2
ESC d	4-42
ESC e	4-41
ESC ESC	4-43
ESC f	4-41, 57
ESC g	4-42
ESC h	2-2
ESC HOP	4-35, 42
ESC HOP HOP	4-42
ESC HOP taste	4-35, 42
ESC k	4-43
ESC n	4-41
ESC o	4-43
ESC p	4-42
ESC q	2-2, 4-41
ESC RUBIN	4-38, 42
ESC RUBOUT	4-37, 42
ESC s	2-2, 4-43, 56
ESC t	2-2
ESC TAB	4-33
ESC ? taste	4-43, 57
ESC U	4-43
ESC v	4-41
ESC w	4-41
ESC k	4-43
ESC -	4-43
ESC #	4-43
EUMEL – Drucker	5-85
EUMEL – Editor	4-1
EUMEL – Hintergrundes	A-1
EUMEL – Zeichensatz	6-3
Exponenten	5-39
family password	3-3, 44
father	3-51
fehlerhafte Anweisungen	5-87
Fehlermeldungen	5-11
Fenster	5-11, 4-23
Fenster – Editor	5-11
fetch	3-2, 51
fett	5-22
fillchar	5-36

first head	5 – 107
Fontttabelle	5 – 40
fonttable	3 – 3
forget	3 – 2, 52
format	3 – 52
format (archive)	3 – 2
formatieren	5 – 1
Formatierung für Seiten	5 – 46
Formatierung spaltenweise	5 – 71
Formatierung von Spalten	5 – 70
free global manager	3 – 10
Füllzeichen	5 – 36
Fußnoten	5 – 60
Fußnoten, Leerzeilen davor	5 – 115
Fußnoten numerieren	5 – 63
Fußnoten pro Seite zählen	5 – 111
Fußzeilen	5 – 54
gerollt	4 – 59
geschütztes Leerzeichen	5 – 32
Gesetzte Tabulatormarken	4 – 32
Gesperrt schreiben	5 – 24
GET	4 – 58, 63
gib kommando	3 – 1
global manager	3 – 2, 8, 52
halt	2 – 2, 13
help	2 – 2
HOP LINKS	4 – 60
HOP RECHTS	4 – 60
HOP RUBIN	4 – 31
HOP RUBOUT	4 – 31
HOP TAB	4 – 32
HOP – Taste	4 – 27
hyphenation width	5 – 114
Index	5 – 4, 57
index	3 – 3
Index – Anweisungen	5 – 75ff, 75, 83
Indexdateien	5 – 77
Indexdateien zusammenführen	5 – 82
index merge – Kommando	5 – 82
Indizes	5 – 39
Inhaltsverzeichnis	5 – 75

Interaktive Silbentrennung	5 – 16
Kommando	1 – 5, 6
Kommandodialog	4 – 44
Kommando pause	4 – 47
Kommandotaste	4 – 14
Kommandozeile	4 – 44, 47
Kommandozustand	4 – 44
Kontroll – /Steuertaste	4 – 8
Kopf – und Fußbereiche abstellen	5 – 107
Kopf – und Fußbereiche Schalter – Anweisungen für	5 – 105
Kopf – und Fußzeilen	5 – 54
kursiv	5 – 22
lade ausnahmen	5 – 112
last bottom	5 – 107
layout – Anweisungen	5 – 6
LEARN	4 – 35
Leerzeile	4 – 19
Leerzeilen am Seitenanfang	5 – 46
Leerzeilen einfügen	5 – 47
Leerzeilen vor Fußnoten	5 – 115
Lernen	4 – 36
Lernmodus	4 – 35
Lernsequenz	4 – 35, 36
Lese – Fehler (Archiv)	3 – 33
lex sort	5 – 76 , 6 – 8, 13
limit	4 – 59, 63
limit – Einstellungen	5 – 26
limit – Kommando	4 – 59
lineform	3 – 3 , 5 – 13
lineform	3 – 3
lineform auf einen Textausschnitt	5 – 18
lineform / autoform	5 – 3, 13
list	3 – 2, 53
list (archive)	3 – 2
list fonts	3 – 3 , 6 – 12
list macros	5 – 98
load macros	5 – 97
Löschen von Zeichen	4 – 25
Löschtaste	4 – 10
Makro – Datei	5 – 96
Makro – Definition	5 – 97

Makroname	5 – 97
Makro – Parameter	5 – 101
Makrorumpf	5 – 99
Manager	1 – 13 , 3 – 8
margin	4 – 60, 63
MARK	4 – 37
Markieren und löschen	4 – 37
Markierung	4 – 37
Markierzustand	4 – 13
Mehrfachblank	5 – 27, 89
Monitor	1 – 6
Monitor	3 – 1
Mustererkennung	4 – 52
myself	3 – 53
Nebeneinträge erzeugen	5 – 79
Neue Seite beginnen	5 – 53
Notizbuch	5 – 11, 19 , 6 – 1
number empty lines before foot	5 – 115
Operationen bei der interaktiven Trennung	5 – 17
OR	4 – 63
outline	3 – 3 , 5 – 4, 10, 83
pageform	3 – 3 , 5 – 3, 42
Parameter	1 – 6
Passwort	3 – 42
Pattern Matcher	4 – 52
Platz freihalten	5 – 52
Positionieren	4 – 23
Positionierung des Cursors	4 – 9
Positionierungshilfe	4 – 32
Positionierungstasten	4 – 23
print	3 – 3 , 5 – 85, 86
PRINTER	A – 11
Proportionalschrift	4 – 34 , 5 – 20, 27
PUBLIC	1 – 13 , 3 – 22, 53
PUT	4 – 58, 64
Querverweise	5 – 67
Randausgleich	5 – 89
REAL – Parameter	5 – 8
Rechtsbündig drucken	5 – 94
release	3 – 32
rename	3 – 2, 53

REST	4 – 29, 31
RUBIN	4 – 26
RUBOUT	4 – 25
Rückumbruch	4 – 31
save	3 – 2, 30, 54
Schreibarbeit beenden	4 – 4
Schreiben von Texten	4 – 19
Schreibfehler	4 – 25
Schreibfeld	5 – 92
Schreibfeld verschieben	5 – 92
Schreibposition	4 – 3
Schrifthöhe	5 – 50
Schrifttyp	5 – 14
Schrifttypen, unterschiedliche	5 – 20
Schrifttyp – Modifikation	5 – 22
Seitenausgleich	5 – 91
Seitenende interaktiv verschieben	5 – 45
Seitenende verschieben	5 – 47
Seitenlänge einstellen	5 – 49
Seiten numerieren	5 – 57
Seitennummer	5 – 57, 77
Seitenweise formatieren	5 – 42
Setzen des Tabulators	4 – 32
SHard	A – 1
SHIFT – Taste	4 – 5
shutup	1 – 15
Silbentrennung	5 – 13, 114
Silbentrennungen, Behandlung von falschen	5 – 112
SOME	3 – 38, 54
Sonderzeichen	4 – 5
sort	6 – 6, 13
Spaltenende	5 – 71
Spaltenformatierung	5 – 70
Sprung an das rechte Bildschirmende	4 – 27
Sprung an den Bildschirmrand links	4 – 28
Sprung auf die erste Zeile des Bildschirms	4 – 28
Sprung auf die letzte Zeile des Bildschirms	4 – 28
Stichwortverzeichnis	5 – 75
storage info	2 – 2, 13 , 3 – 2, 55
Strukturübersicht	5 – 83
Suchen und Ersetzen	4 – 47

Supervisor	1 – 6
SUPERVISOR – Taste im Mehrbenutzer – Betrieb	4 – 15
SV	5 – 11
Systemabschaltung	A – 3
Systemstart	A – 3
T	4 – 48, 51, 64
TAB	4 – 32, 34
Tabellenanweisungen	5 – 31
Tabellenautomatik	5 – 28, 29
Tabellen, einfache	5 – 27
Tabellenpositionen, Einstellen von	5 – 34
Tabellenpositionen löschen	5 – 38
Tabellenspalten auffüllen	5 – 36
Tabulator	4 – 32
Tabulatormarken	4 – 32, 33
Tabulatortaste	4 – 13
task	1 – 4, 12 , 3 – 55
task info	2 – 2, 13 , 3 – 2, 55
task password	3 – 3, 42, 56
task status	3 – 2, 56
Tastenbelegung	4 – 5
Tastenfeld	4 – 5
Texte markieren	4 – 37
Texte suchen	4 – 52
Textkosmetik	5 – 2
Textkosmetik – Makros	5 – 96
Textkosmetik – Programme	5 – 1, 3
Textkosmetik – Programme aktivieren	5 – 9
TEXT – Parameter	5 – 8
Textstelle anwählen	4 – 45
Titelzeile	4 – 3
TO LINE	4 – 48
Trenngüte	5 – 19
Trennpunkt einstellen	5 – 114
Trennungsvorschläge	5 – 16
type	6 – 14
type	4 – 64
U	4 – 64
U – Kommando	4 – 49
Übereinander drucken	5 – 95
Überschreiben	4 – 25

Überschreiben von Zeichen	4 – 25
Überschriften in Spalten	5 – 72
Umlaute	4 – 5
Umschalttaste	4 – 8
Unterbrechen einer Ausgabe	4 – 17
Unterschiedliche Schrifttypen	5 – 20
Unterschied zwischen Kommando u.Text – Anweisung	5 – 1
Unterstreichung	5 – 22
UP	4 – 49
Veränderung des Schrifttyps	5 – 22
Verstärkertaste	4 – 9, 27
Voreingestellte Tabulatormarken	4 – 32
Voreinstellungen ändern	5 – 114
Vorhandensein einer Folgeseite	5 – 57
vorzeitiger Abbruch	5 – 11
WEITER – Taste	4 – 17
word wrap	4 – 65
word wrap (false)	4 – 20
word wrap (true)	4 – 20
Worte für index kennzeichnen	5 – 77
Wortlücken, Vergrößern der	5 – 89
Wortumbruch	4 – 19
Zahlentabellen	4 – 34
Zeichen löschen	4 – 25
Zeichenweises Positionieren	4 – 27
Zeilenabstand	5 – 50
Zeilenabstand einstellen	5 – 50
Zeilenbreite	5 – 14
Zeilenbreite einstellen	5 – 25
Zeilenformatierung	5 – 13
Zeilen löschen	5 – 47
Zeilennummer	4 – 24
Zeilenvorschub	5 – 50
Zeilenwechsel	4 – 23
Zeilenweise formatieren	5 – 16
Zentriert drucken	5 – 93
Zusammenfassung	5 – 83
Zwischenspeicher	4 – 38
#block# – Anweisung	5 – 89
#bottom off#	5 – 105
#bottom on#	5 – 105

#bottom# – Anweisung	5 – 54
#b pos – Anweisung	5 – 31
#b# – Anweisung	5 – 95
#center# – Anweisung	5 – 93
#clear pos# – Anweisung	5 – 32, 38
#columns end# – Anweisung	5 – 71
#count# – Anweisung	5 – 63
#c pos – Anweisung	5 – 31
#d pos – Anweisung	5 – 31
#d# – Anweisung	5 – 39
#end# – Anweisung	5 – 54
#e# – Anweisung	5 – 39
#fillchar – Anweisung	5 – 31
#free# – Anweisung	5 – 52
#goalpage# – Anweisung	5 – 67
#head off#	5 – 105
#head off# – Anweisung	5 – 106
#head on#	5 – 105
#head on# – Anweisung	5 – 106
#head# – Anweisung	5 – 54
#ib# – Anweisung	5 – 77
#ie# – Anweisung	5 – 77
#limit# – Anweisung	5 – 25
#limit# – Anweisung für Spalten	5 – 70
#linefeed# – Anweisung	5 – 50
#linefeed# – Anweisung	5 – 40
#l pos – Anweisung	5 – 31
#mark# – Anweisung	5 – 110
#off# – Anweisung	5 – 22
#on# – Anweisung	5 – 22
#pageblock# – Anweisung	5 – 46, 91
#pagelength# – Anweisung	5 – 49
#pagenr	5 – 59
#page# – Anweisung	5 – 48, 53
#page# – Anweisung bestätigen/löschen	5 – 48
#page# – Anweisung für Spaltenende	5 – 71
#page# – Anweisung, mit neuer Seitennummer	5 – 53
#right# – Anweisung	5 – 94
#r pos – Anweisung	5 – 31
#start# – Anweisung	5 – 92
#table end# – Anweisung	5 – 32

## Index

---

#table# – Anweisung	5 – 32
#topage# – Anweisung	5 – 67
#type# – Anweisung	5 – 87
#u# – Anweisung	5 – 39
% – Zeichen	5 – 57