

ANWENDER- DOKUMENTATION	Teil 3	MOS
11/87	Textverarbeitung mit nroff	MUTOS 1700

Programmtechnische
Beschreibung

Teil 3

Textverarbeitung mit nroff

AC A 7100/7150

VEB Robotron-Projekt Dresden

Ausgabe: 11/87

Die Ausarbeitung dieser Dokumentation erfolgte durch ein Kollektiv des VEB Robotron-Elektronik Dresden Stammbetrieb des VEB Kombinat Robotron.

Nachdruck und jegliche Vervielfältigung, auch auszugsweise, sind nur mit Genehmigung des Herausgebers zulässig.

Im Interesse einer ständigen Weiterentwicklung werden alle Leser gebeten, Hinweise zur Verbesserung der Dokumentation dem Herausgeber mitzuteilen.

Herausgeber:

VEB Robotron-Projekt Dresden
Leningrader Str. 9
Dresden 8010

(C) VEB Kombinat Robotron

Kurzreferat

Das Textformatierungsprogramm nroff ist das Kernstück der Textverarbeitung im Betriebssystem MUTOS. Es erlaubt dem Anwender die automatische Formatierung vielgestaltiger Texte und ist vorallem für die entwicklungsbegleitende Dokumentationsbearbeitung geeignet.

nroff ist ein kommandoorientiertes Textverarbeitungsprogramm. Seine Kommandosprache enthält wesentliche Elemente einer Programmiersprache. Auf Grund der bewußt sehr einfach gehaltenen Kommandos liegen seine Vorteile in der Möglichkeit, auf spezielle Formatierungserfordernisse zugeschnittene, stilvereinheitlichende Makros zu programmieren.

Inhaltsverzeichnis	Seite
1. EINFÜHRUNG	
2. ALLGEMEINE ERKLÄRUNGEN	
2.1. FORM DER EINGABE	2-1
2.2. GERÄTEANPASSUNG	2-1
2.3. NUMERISCHE PARAMETEREINGABE	2-1
2.4. NUMERISCHE AUSDRÜCKE	2-3
2.5. NOTATION	2-3
3. ZEICHENVORRAT UND SCHRIFTARTEN	
4. SEITENSTEUERUNG	
5. TEXTAUFFÜLLUNG, RANDAUSGLEICH UND ZENTRIERUNG	
5.1. FÜLLEN UND AUSRICHTEN	5-1
5.2. UNTERBROCHENER TEXT	5-1
6. ZEILENVORSCHUB (SPACING)	
6.1. BASISZEILENABSTAND	6-1
6.2. EXTRA-ZEILENABSTAND	6-1
6.3. BLÖCKE VON ZEILENVORSCHÜBEN	6-1
7. ZEILENLÄNGE UND EINRÜCKUNG	

8.	MAKROS, STRINGS, DIVERSIONEN UND TRAPS	
8.1.	MAKROS UND STRINGS	8-1
8.2.	EINGABEINTERPRETATION IM KOPIERMODUS	8-1
8.3.	ARGUMENTE	8-1
8.4.	DIVERSIONEN	8-2
8.5.	TRAPS	8-3
9.	REGISTER	
10.	TABULATOREN, LEITPUNKTE UND FELDER	
10.1.	TABULATOREN UND LEITPUNKTE	10-1
10.2.	FELDER	10-1
11.	EIN- UND AUSGABE-VEREINBARUNG UND ZEICHENÜBERSETZUNGEN	
11.1.	EINGABE-ZEICHENÜBERSETZUNGEN	11-1
11.2.	SONDERFUNKTIONEN	11-1
11.3.	KOMMANDOKENNZEICHEN	11-2
11.4.	ZEICHENUMSCHLÜSSELUNG	11-3
11.5.	TEXTDURCHLAUF OHNE ÜBERSETZUNG	11-3
11.6.	KOMMENTARE UND VERDECKTE NEWLINES	11-3
12.	BEWEGUNGEN, WEITEFUNKTION UND LINIEN	
12.1.	HORIZONTALE UND VERTIKALE BEWEGUNGEN	12-1
12.2.	MERKEN EINER HORIZONTALEN POSITION	12-1
12.3.	LINIENZIEHUNG	12-1

MUTOS 1700

13. ZEICHENKONSTRUKTION

- 13.1. ZEICHENÜBERSCHREIBUNG 13-1
- 13.2. ZEICHENDRUCK OHNE VORSCHUB 13-1
- 13.3. GROSSE KLAMMERN 13-1

14. SILBENTRENNUNG

15. TITELZEILEN

16. AUSGABEZEILENUMERIERUNG

17. BEDINGTE AKZEPTIERUNG DES INPUTS

18. UMSCHALTEN DES FORMATIERUNGSUMFELDES
(ENVIRONMENT)

19. EINFÜGEN VON EINGABEN DER STANDARDEINGABE

20. EIN-ÄUSGABE-FILE-UMSCHALTUNG

21. WEITERE VERSCHIEDENE KOMMANDOS

22. MITTEILUNGEN

ANLAGE A ALPHABETISCHE ZUSAMMENSTELLUNG DER KOMMANDOS

ANLAGE B ZUSAMMENSTELLUNG DER STEUERFOLGEN

ANLAGE C ALLGEMEINE VORDEFINIIERTE ZAHLENREGISTER

ANLAGE D TABELLE DER EINGABENAMEN FUER SONDERZEICHEN

TABELLEN

- 2-1: Maßeinheiten 2-1
- 9-1: Interpolation der Register 9-1

1. Einführung

nroff ist ein Textformatierungsprogramm mit hohem Komfort zur vollautomatischen Herstellung vielgestaltiger Texte und kompletter Dokumentationen.

Built-in-Kommandos und im Text eingestreuten "Steuerfolgen" bilden die Basis des Textaufbereitungskonzeptes von nroff. Darauf aufbauend kann sich der Nutzer der Makrotechnik bedienen, um durch einfachen Aufruf komplizierte Formatierungsaufgaben zu erledigen. Mit Hilfe eines entsprechenden Makropaketes ist es möglich, den Aufwand bei der Texterfassung gering zu halten und eine relativ schnelle Einarbeitung zu gewährleisten. Für normale Dokumentationsaufgaben steht ein Standardmakropaket zur Verfügung.

nroff baut auf den Fähigkeiten von Seriendruckern auf. Die Anpassung an gerätespezifische Besonderheiten, z. B. Zeilenabstand, Zeichenabstand u. a., kann durch Modifikation der Gerätetabelle erfolgen.

Diese Dokumentation beschreibt die Syntax und Wirkungsweise der Kommandos und Steuerfolgen, um den Anwender in die Lage zu versetzen, sich ein eigenes Makropaket zu schreiben oder vorhandene Makrosätze seinen spezifischen Bedürfnissen anzupassen. Es ist im allgemeinen nicht üblich, diese Kommandos und Steuerfolgen, bis auf wenige Ausnahmen, direkt bei der Gestaltung von Texten zu verwenden.

2. Allgemeine Erklärungen

~~~~~

### 2.1. Form der Eingabe

~~~~~

Die Eingabe enthält das zum Druck bestimmte Manuskript, vermischt mit Steuerzeilen, die die Formatierung des Textes in der gewünschten Form veranlassen. Steuerzeilen beginnen mit einem Kommandokennzeichen, normalerweise "." oder "'", gefolgt von einem 1- oder 2-Zeichen-Namen. Dieser Name spezifiziert entweder einen Ruf zu einem Basiskommando oder die Substitution eines nutzerdefinierten Makros. Namen müssen von den nachfolgenden Parametern durch ein Leerzeichen getrennt sein. Steuerzeilen mit nichterkannten Namen werden ignoriert. Zusätzlich können in normale Textzeilen Steuerfolgen, die mit dem internen Escape-Zeichen, normalerweise "\", beginnen, eingebaut werden, um die Ausführung einer Spezialfunktion zu bewirken.

2.2. Geräteanpassung

~~~~~

nroff benutzt als interne Bezugsgröße 240 Einheiten/Inch entsprechend einem angenommenen kleinsten gemeinsamen Vielfachen der horizontalen und vertikalen Schrittweite der verschiedenen Ausgabegeräte. Die tatsächlichen Gerätegegebenheiten werden in einer entsprechenden Gerätetabelle spezifiziert. Mittels der "-T"-Option im nroff-Kommando kann auf die jeweilige Gerätetabelle Bezug genommen werden. (Siehe Handbuch für den Bediener, nroff (1), term (7))

### 2.3. Numerische Parametereingabe

~~~~~

nroff akzeptiert bei einer Vielzahl von Kommandos numerische Parameter mit angehängter Maßeinheit.

Tabelle 2-1: Maßeinheiten

Maßeinheit	Bedeutung	Basiseinheiten
i	Inch	240
c	Zentimeter	240*50/127
P	Pica = 1/6 Inch	240/6
m	Em	C
n	En	C
p	Point = 1/72 Inch	240/72
u	Basiseinheit	1
v	vertikaler Zeilenabstand	V

C ist die Zeichenbreite und V der Zeilenabstand. Beide Größen sind geräteabhängig. Die berechneten Werte werden auf die gerätespezifischen Gegebenheiten aufgerundet.

Ist keine Maßeinheit angegeben, so ist sie Em (m) für die horizontal orientierten Kommandos ll, in, ti, ta, lt, po, mc sowie \h und \l; v für die vertikal orientierten Kommandos pl, wh, ch, dt, sp, sv, ne, rt, \v, \x und \L; p für das Kommando vs und u für die Kommandos nr, if und ie.

Das Zeichen "|" kann einer Zahl N vorgestellt werden, um die Distanz von der laufenden horizontalen bzw. vertikalen Position zu der Position N zu erzeugen.

2.4. Numerische Ausdrücke

~~~~~

Bei numerischen Eingaben können Ausdrücke mit Operatoren einschließlich Klammern verwendet werden. Es sind die arithmetischen Operatoren

+ , - , / , \* , %(mod)

und die logischen Operatoren

< , > , <= , >= , = (bzw.==), &(UND), :(ODER)

zugelassen. Die Auflösung der Ausdrücke erfolgt, außer bei Klammerung, von links nach rechts, es gibt keinen Operatorvorrang. Bei bestimmten Kommandos wird ein vorgestelltes "+" oder "-" als Inkrement- bzw. Dekrementoperator interpretiert.

#### 2.5. Notation

~~~~~

Numerische Parameter werden in diesem Handbuch auf 2 Arten angegeben:

+_N bedeutet, daß das Argument die Form N, +N, -N haben kann, das Vorzeichen bedeutet ein Inkrement bzw. ein Dekrement des jeweiligen Argumentes, N ein Setzen auf N.

N bedeutet, daß ein angegebenes Vorzeichen algebraischen Charakter besitzt, also tatsächlich als Vorzeichen zu werten ist.

3. Zeichenvorrat und Schriftarten

nroff baut auf dem Zeichenvorrat des KOI7-Codes auf. Insgesamt stehen in der Gerätetabelle jedoch 324 verschiedene Möglichkeiten der Zeichenverschlüsselung zur Verfügung, so daß zusätzliche Zeichen, z. B. Umlaute, konstruiert werden können. Die zusätzlichen Zeichen werden in der Form \(\xx, wobei xx der Name des Zeichens ist, eingegeben. Die Namen der zusätzlichen Zeichen sind in Anlage 3 aufgeführt. Die genaue Behandlung dieser konstruierten Zeichen kann geräteabhängig unterschiedlich erfolgen.

nroff kennt 3 Schriftarten: normal (R), kursiv (I) und Breitdruck (B). Kursiv wird normalerweise durch Unterstreichung realisiert. Falls das Ausgabegerät die Fähigkeit zum Kursivdruck besitzt, wird davon Gebrauch gemacht. Breitdruck ist nur bei dafür vorgesehenen Druckern möglich, ansonsten wird er ignoriert. Zusätzlich dazu ist zur Hervorhebung einzelner Wörter ein Mehrfachdruck möglich (siehe Abschnitt 12.3.).

Kommandoübersicht:

Kommando	Anfangswert	ohne Argument	KZ	Erklärung
.ft F	R	vorher	E	Wechsel der Schriftart F. Analog dazu kann auch die Steuerfolge \fx mit der gleichen Wirkung verwendet werden. Für F können die Buchstaben R, I und B oder die zugeordneten Zahlen 1, 2, 3 verwendet werden.
.uf F	I	I		Setzen der Schriftart Unterstreichung für Schriftart F, F = 2, 3

Unter Anfangswert ist der im Programm initialisierte Wert für den entsprechenden Parameter angegeben.

In der Spalte "ohne Argument" wird die Funktion angegeben, falls das Argument fehlt. Dabei bedeutet "vorher" ein Aufheben der Wirkung des entsprechenden laufenden Kommandos und die Wiederherstellung des früheren Zustandes.

In der Spalte KZ sind die Kurzzeichen mit folgender Bedeutung angegeben:

- B Das Kommando bewirkt einen Break (siehe Abschnitt 5.).
- D Der Modus bzw. die entsprechenden Parameter sind mit der jeweiligen Diversionsebene verbunden.
- E Die Parameter sind Teil des laufenden Environments (siehe Abschnitt 21.).
- v,p,m,n Die jeweilige Standardmaßeinheit für dieses Kommando; fehlt diese Angabe, so wird eine im Kommando angegebene Maßeinheit ignoriert.

4. Seitensteuerung

~~~~~

Da der obere und untere Rand einer Seite nicht automatisch geliefert wird, ist es üblich, dafür 2 Makros zu definieren und diese mit Hilfe von Traps an den Seitenpositionen 0 (oberer Rand) und -N (N vom unteren Rand) aufzurufen (siehe Abschnitt 8.5.).

Der Übergang zur ersten Seite erfolgt entweder mit der ersten nicht in eine Diversion (Abschnitt 8.4.) abgelenkten echten Eingabetextverarbeitung oder dem ersten Break. Die Informationen zur Gestaltung des oberen Randes der ersten Seite müssen vor diesem Übergang festliegen.

Das Kommando .po hat keinen Einfluß auf die Zeilenlänge; es verschiebt die gesamte Zeile nach links bzw. rechts. Mit den Kommandos .mk, .rt und .po, eingebaut in die entsprechenden Seitenanfangs- bzw. Seitenendmakros, kann z. B. ein Mehrspaltendruck realisiert werden.

## Kommandoübersicht:

| Kommando | Anfangs-<br>wert | ohne<br>Argument | KZ  | Erklärung                                                                                                                                                                                                                                                |
|----------|------------------|------------------|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| .pl +_N  | 66               | 66               | v   | Setzen der Seitenlänge auf +_N. Die aktuelle Seitenlänge ist im .p-Register verfügbar.                                                                                                                                                                   |
| .bp +_N  | N = 1            |                  | B   | Neue Seite. Die laufende Seite wird herausgegeben und eine neue Seite wird begonnen. Wenn +_N gegeben ist, wird die neue Seitennummer +_N, sonst wird sie um 1 erhöht.                                                                                   |
| .pn +_N  | N = 1            | ignoriert        |     | Setzen der Seitennummer auf +_N. Die laufende Seitennummer wird im %-Register geführt.                                                                                                                                                                   |
| .po +_N  |                  | vorher           | v   | Seitenoffset. Der aktuelle linke Rand wird auf +_N gesetzt. Der laufende Seitenoffset wird im .o-Register geführt.                                                                                                                                       |
| .ne N    |                  | N = 1            | D,v | Es wird ein verfügbarer Abstand N bis zur nächsten Trapposition benötigt. Ist der Abschnitt kleiner, so wird der Trap sofort ausgeführt. Das Kommando wird verwendet, um z. B. einen neuen Abschnitt nicht auf den letzten Zeilen der Seite zu beginnen. |
| .mk R    |                  | intern           | D   | Merken der aktuellen vertikalen Position in einem internen Register bzw. im Register R, falls angegeben.                                                                                                                                                 |
| .rt +_N  |                  | intern           | D,v | Zurück zu einer markierten vertikalen Position. Ist +_N angegeben, ist die Position +_N entfernt vom Seitenbeginn. Fehlt +_N, ist die Position durch ein vorheriges .mk festgelegt.                                                                      |

## 5. Textauffüllung, Randausgleich und Zentrierung

~~~~~

5.1. Füllen und Ausrichten

~~~~~

Die Worte werden vom Eingabetext gesammelt und in die Ausgabezeile übertragen, bis die Zeile entsprechend der festgelegten Zeilenlänge (Abschnitt 7.) gefüllt ist (Flattersatz). Es wird eine Silbentrennung versucht, um den Rest der Zeile auszunutzen. Anschließend werden die Zwischenräume zwischen den Wörtern der Ausgabezeile vergrößert, um den rechten Rand auszurichten (Blocksatz).

Ein Wort ist eine Zeichenkette, begrenzt durch ein Leerzeichen oder den Anfang bzw. das Ende der Eingabezeile. Ein zusammengehöriges Paar von Wörtern, das nicht getrennt, auch nicht gespreizt werden soll, kann durch Klammerung mit dem "\ "-Zeichen zusammengehalten werden.

In nroff sind die Wortzwischenräume normalerweise nicht gleich, mit der nroff-Kommando-Option -e kann eine Uniformierung der Zwischenräume versucht werden. Textauffüllen, Randausgleich und Silbentrennung (siehe Abschnitt 14.) kann ganz verhindert oder gesteuert werden.

Die Textlänge der letzten Ausgabezeile ist im .n-Register und die Seitenposition dieser Zeile im nl-Register verfügbar. Der Füllungsstand der Seite steht im .h-Register.

".", "?", "!" am Ende einer Eingabezeile werden als Ende des Satzes gedeutet und es wird automatisch ein zusätzliches Leerzeichen eingefügt. Mehrfache Leerzeichen zwischen Wörtern in der Eingabezeile werden einbehalten. Ein Leerzeichen am Anfang einer Eingabezeile bewirkt einen Break, d. h. die bisher gefüllte Zeile wird ausgegeben und eine neue Zeile begonnen.

Im Füll-Modus bewirkt ein \p einen Break am Ende des Wortes, bei dem es steht. Die bisher gesammelte Zeile wird ausgespreizt auf die gesamte Zeilenlänge ausgegeben.

Einer Texteingabezeile, die mit einem Kommandokennzeichen eingeleitet wird, kann der Status einer Steuerzeile durch Vorsatz von \& weggenommen werden. Eine andere Möglichkeit hierfür ist die Translation des Kommandokennzeichens in ein anderes passendes Zeichen mittels .tr (siehe Abschnitt 11.4.).

### 5.2. Unterbrochener Text

~~~~~

Das Kopieren der Eingabezeile im Nichtfüll-Modus kann durch ein \c am Ende der Zeile, z. B. um Kommandos einzufügen, unterbrochen werden. Die nächste entscheidelte Eingabezeile wird dann als Fortsetzung dieser Zeile gehalten. Analog dazu kann im Füll-Modus ein Wort unterbrochen und fortgesetzt werden. Bewirken dazwischen eingebaute Steuerzeilen einen Break, so wird die angefangene Zeile ausgegeben.

Kommandoübersicht:

Kommando	Anfangs- wert	ohne Argument	KZ	Erklärung
.br			B	Break. Das Auffüllen der laufenden Zeile wird gestoppt und die Zeile ohne Randausgleich ausgegeben. Mit einem Leerzeichen beginnende Zeilen und Leerzeilen bewirken ebenfalls einen Break.
.fi	ein		B,E	Einschalten des Füll-Modus, Register .u = 1 Füll-Modus .u = 0 Nichtfüll-Modus
.nf			B,E	Ausschalten des Füll-Modus und des Randausgleiches. Eingabezeilen werden ohne Rücksicht auf die Zeilenlänge zur Ausgabe kopiert.
.ad c	ein	ein	E	Einschalten Randausgleich. Im Nichtfüll-Modus wird der Randausgleich bis zum Einschalten des Füll-Modus unterdrückt. c = Typindikator c Randausgleichstyp ----- l linker Rand r rechter Rand c Zentrierung b oder c links und rechts
.na			E	Ausschalten Randausgleich. Der Füll-Modus wird davon nicht betroffen.
.ce N	aus	N = 1	B, E	Zentrierung der nächsten N Eingabezeilen. Falls die Eingabezeile zu lang ist, wird sie links ausgerichtet.

6. Zeilenvorschub (Spacing)

~~~~~

### 6.1. Basiszeilenabstand

~~~~~

Der vertikale Zeilenabstand (v) zwischen zwei aufeinanderfolgenden Normalzeilen ist bei nroff geräteabhängig und ist in der jeweiligen Gerätetabelle anzugeben.

6.2. Extra-Zeilenabstand

~~~~~

Wenn ein Wort eine extra große Vertikalkonstruktion benötigt, z. B. um einen mathematischen Ausdruck in große Klammern zu setzen, kann dieses Wort mit der Spezialfunktion `\x'N'` verbunden werden. Ist N negativ, wird der gewünschte Zeilenvorschub vor der Zeile, die die Spezialfunktion enthält, ausgeführt. Ist N positiv, wird zunächst die Zeile ausgegeben und dann der Zeilenvorschub ausgeführt.

### 6.3. Blöcke von Zeilenvorschüben

~~~~~

Ein Block von Zeilenvorschüben wird in der Regel mit `.sp` vereinbart. Im No-Space-Modus wird `.sp` ignoriert. Der No-Space-Modus dient dem Zweck, die an sich vorgesehene Ausgabe von Leerzeilen, z. B. vor Beginn eines neuen Abschnitts, in bestimmten Situationen zu unterdrücken. Das ist u. a. sinnvoll, wenn inzwischen ein Seitenwechsel aufgetreten ist.

Die Reservierung von Freiraum mittels `.sv` und seine u. U. spätere Ausgabe mittels `.os` ermöglicht es, Leerraum, z. B. zum späteren Einfügen von Bildern, zusammenhängend in den dynamisch aufgebauten Text, unabhängig von eventuellem Seitenwechsel, einzubauen.

Kommandoübersicht:

Kommando	Anfangs- wert	ohne Argument	KZ	Erklärung
.vs N	12	vorher	E,p	Setze den Zeilenabstand auf N Points
.ls N	N = 1	vorher	E	N Leerzeilen werden jeder Ausgabezeile angehängt. Bei Erreichen einer Trapposition wird die Leerzeilenausgabe unterdrückt.
.sp N		N = 1	B,v	Relative Vertikalpositionierung um Größe N. Ist N negativ, wird rückpositioniert, begrenzt durch die obere Seitenkante. Ist N positiv, wird vorwärtspositioniert maximal bis zur nächsten Trapposition. Wird vor allem benutzt, um Leerzeilen einzufügen.
.sv N		N = 1	v	Freihalten eines N-großen Freiraumes innerhalb des Outputs. Ist N kleiner als die Distanz bis zur nächsten Trapposition, so wird die Vorwärtspositionierung ausgeführt; No-Space-Modus spielt dabei keine Rolle. Ist N größer, so wird die Vorwärtspositionierung zurückgehalten (siehe .os).
.os				Ausgabe eines vorher mit .sv geretteten Freiraumes. Der No-Space-Modus wird ignoriert.
.ns			D	Einschalten des No-Space-Modus. Unterdrückung von .sp-Rufen, bei .bp-Rufen wird die Seitennummer nicht weitergeschaltet.
.rs				Wiedereinschalten des Space-Modus
Leerzeile			B	die gleiche Wirkung wie .sp 1

7. Zeilenlänge und Einrückung

Die maximale Zeilenlänge für den Füll-Modus wird mit `.ll` eingestellt. Einrückungen (Indents) werden mit `.in` gesetzt. Eine temporäre Einrückung, die nur für die nächste Zeile gilt, wird mit `.ti` eingestellt. Die Zeilenlänge abzüglich einer eventuellen vereinbarten Einrückung ist die Basis für die Zentrierung (`.ce`).

Die laufende Zeilenlänge wird im Register `.l` und die laufende Einrückung im Register `.i` geführt.

Kommandoübersicht:

Kommando	Anfangswert	ohne Argument	KZ	Erklärung
<code>.ll +_N</code>	6.5in	vorher	E,m	Die Zeilenlänge wird auf <code>+_N</code> gesetzt.
<code>.in +_N</code>	<code>N = 0</code>	vorher	B,E,m	Setzen der Einrückung auf <code>+_N</code> . Die Einrückung gilt für jede Ausgabezeile.
<code>.ti +_N</code>		ignoriert	B,E,m	Temporäre Einrückung. Die nächste Ausgabezeile wird um <code>+_N</code> eingerückt. <code>+_N</code> bezieht sich auf die laufende Einrückung.

8. Makros, Strings, Diversionen und Traps

~~~~~

### 8.1. Makros und Strings

~~~~~

Ein Makro ist ein mit Namen versehener Satz von beliebigen Zeilen. Er wird mit Namen oder mit Trap aufgerufen. Ein String ist eine mit Namen versehene Zeichenkette, die irgendwo durch Angabe des Namens in die Ausgabe eingefügt werden kann. Kommandos, Makros und Strings benutzen innerhalb von proff die gleiche Namensliste. Makro- und Stringnamen können 1 oder 2 Zeichen lang sein. Mit .rn können sie umbenannt und mit .rm gelöscht werden.

Makros werden mit .de und .di definiert und mit .am und .da erweitert; .di und .da bewirken, daß der normale Output in einem Makro abgespeichert wird. Strings werden mit .ds definiert und mit .as erweitert. Makros werden wie normale Kommandos aufgerufen, zuzüglich können bis zu 9 Argumente mitgegeben werden. Die Strings x bzw. xx werden an der gewünschten Stelle durch *x bzw. *(xx eingefügt. String-Bezugnahmen und Makrorufe können geschachtelt sein.

8.2. Eingabeinterpretation im Kopiermodus

~~~~~

Während der Definition und Erweiterung von Strings und Makros (aber nicht bei Diversionen) wird der Input im Kopiermodus gelesen. Der Input wird kopiert, ohne daß eine Interpretation erfolgt, außer,

- die Interpolation der Inhalte der Nummernregister, die durch \n angewiesen wird, wird ausgeführt,
- die Einfügung von Strings, über \\* angewiesen, erfolgt,
- die Substitution von Argumenten, durch \\$ angewiesen, erfolgt,
- versteckte Newlines, die dadurch gekennzeichnet sind, daß ihnen ein \ vorge stellt ist, werden überlesen,
- Kommentare, durch \" gekennzeichnet, werden eliminiert,
- \t und \a werden als Horizontaltabulator bzw. SOH interpretiert,
- \\ wird \,
- \. wird ".".

Diese Interpretationen können durch Vorsatz eines weiteren "\" unterdrückt werden.

### 8.3. Argumente

~~~~~

Beim Aufruf des Makros mit seinem Namen wird der Rest der Zeile als Argumentliste interpretiert. Es sind bis zu 9 Argumente zugelassen. Das Trennzeichen zwischen den Argumenten ist das Leerzeichen. Soll ein Leerzeichen Teil eines Argumentes sein, so kann das Argument in Anführungsstriche eingeschlossen werden. Der Anführungsstrich kann durch Vorsatz eines weiteren Anführungsstriches selbst zum Argument werden.

Falls die Zeile nicht zur Aufnahme der Argumentliste ausreicht, kann sie mit Hilfe eines "versteckten Newlines" auf der nächsten Zeile fortgesetzt werden. Wird ein Makro aufgerufen, so wird das bisherige Niveau der Eingabe verlassen. Eventuelle Argumente eines vorhergehenden Aufrufes stehen so lange nicht zur Verfügung, bis das vorhergehende Niveau wieder hergestellt ist. Die Substitution eines Argumentes kann im Makrokörper mit `\$N` angewiesen werden, wobei $1 \leq N \leq 9$ gilt. Falls ein aufgerufenes Argument nicht existiert, wird nichts substituiert.

Beispiel:

```
~~~~~
Definition: .de xx
            Dresden im \\$1 des Jahres \\$2
            ..
Aufruf:     .xx Januar 1985

Ergibt den Text:
            Dresden im Januar des Jahres 1985
```

"\\$" wurde in der Definition durch den Vorsatz eines weiteren "\" vor der Auswertung geschützt (siehe Abschnitt 8.2.). Die Anzahl der Argumente des laufenden Makros steht im `.$`-Register. Ein Stringaufruf führt ebenfalls zu einem Kellern des Input-Niveaus, so daß innerhalb des Strings keine Argumente eines eventuellen vorherigen Makroaufrufs verfügbar sind. Einem durch Trap aufgerufenen Makro können keine Argumente übergeben werden. Die Argumente werden im Kopiermodus auf einen Stack übernommen. Dieser Mechanismus erlaubt keine direkte Einfügung eines längeren Strings in das Argument. Es wird deshalb empfohlen diese Interpolation durch Vorsatz eines weiteren "\" auf die Ebene der Substitution der Argumente innerhalb des Makrokörpers zu verzögern.

8.4. Diversionen

Der verarbeitete Text kann auch statt der normalen Ausgabe einem Makro übergeben werden, so daß er an anderer Stelle wieder zur Verfügung steht. Diese Ausgabeverzweigung wird als Diversion bezeichnet. Diversionen können zur Fußnotenverarbeitung, zum Bestimmen von Textgrößen, zur Erstellung von Inhaltsverzeichnissen u. a. verwendet werden. Ein Diversionstrap kann an einer bestimmten Position innerhalb der laufenden Diversion installiert werden, um einen Makro aufzurufen. Die Register `dn` und `dl` beinhalten die vertikale bzw. die horizontale Größe der jeweils zuletzt abgeschlossenen Diversion.

Diversionen können geschachtelt werden, die jeweilige Niveaustufe wird mit bestimmten Registern und Informationen verbunden.

- Diversionstrap
- No-Space-Modus
- intern markierter Platz (siehe `.mk` und `.rt`)
- die laufende vertikale Position (`.d`-Register)
- der aktuelle Füllungsstand der Seite (`.h`-Register)
- Name der laufenden Diversion (`.z`-Register)

Die maximale Schachteltiefe ist 4.

8.5. Traps

~~~~~

Traps sind Unterbrechungspunkte, die an bestimmten Stellen der Textverarbeitung installiert werden. Sie unterbrechen die bisherige Arbeit und rufen einen Makro auf.

Folgende Traparten sind verfügbar:

- Seitenpositionstrap
- Diversionstrap
- Eingabezeilenzähltrap

Seitenpositionstraps zum Aufruf eines Makros können mit Hilfe des Kommandos `.wh` an irgendeiner vertikalen Position der Seite einschließlich des Seitenbeginns installiert werden. Diese Trapposition kann mittels `.ch` gewechselt werden. Trappositionen am oder unter den unteren Rand haben keinen Effekt. Der mit einem Trap verbundene Makro wird automatisch aufgerufen, wenn die trapauslösende Seitenposition erreicht ist. Die Distanz zu dem jeweilig nächsten Trap ist im `.t`-Register verfügbar. Wenn es keinen weiteren Trap zwischen der laufenden Seitenposition und dem unteren Rand gibt, ist es die Distanz zum unteren Rand. Pro Diversionsebene ist es möglich, jeweils einen Diversionstrap zu installieren. Zusätzlich kann ein Trap zum Aufruf eines Makros nach einer bestimmten Anzahl von Eingabezeilen installiert werden. Gezählt werden dabei alle normalen Textzeilen und durch Makros eingefügter Text.

## Kommandoübersicht:

| Kommando      | Anfangs-<br>wert | ohne<br>Argument | KZ | Erläuterung                                                                                                                                                                                                                                                                              |
|---------------|------------------|------------------|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| .de xx yy     |                  | .yy = ..         |    | Definition oder Neudefinition des Makros xx. Der Makro beginnt mit der nächsten Eingabezeile, diese werden im Kopiermodus übernommen. Er wird beendet durch eine Zeile die mit .yy beginnt, wobei der Makro yy gerufen wird. Fehlt der Parameter yy, so wird der Makro mit ".." beendet. |
| .am xx yy     |                  | .yy = ..         |    | Erweitern des Makros xx (siehe .de)                                                                                                                                                                                                                                                      |
| .ds xx string |                  | ignoriert        |    | Definition einer Zeichenkette xx. Ein beginnender Anführungsstrich wird überlesen, um vorgesetzte Leerzeichen zu ermöglichen.                                                                                                                                                            |
| .as xx string |                  | ignoriert        |    | Erweiterung der Zeichenkette xx                                                                                                                                                                                                                                                          |
| .rm xx        |                  | ignoriert        |    | Löschen des Kommandos, Makros oder Strings xx. Der Name wird aus der Namensliste gelöscht und belegter Speicherplatz freigegeben. Nachfolgende Bezugnahmen werden ignoriert.                                                                                                             |
| .rm xx yy     |                  | ignoriert        |    | Löschen des Kommandos, Makros oder Strings xx bis yy. Existiert yy, wird es ebenfalls gelöscht.                                                                                                                                                                                          |

| Kommando | Anfangs-<br>wert | ohne<br>Argument | KS  | Erläuterung                                                                                                                                                                                                                                                                                                                                                      |
|----------|------------------|------------------|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| .di xx   |                  | Ende             | D   | Abspeichern des Outputs in den Makro xx. Es erfolgt die normale Textverarbeitung außer Seitenoffset (.po). Die Diversion endet, wenn .di oder .da ohne Argument entschlüsselt wird.                                                                                                                                                                              |
| .da xx   |                  | Ende             | D   | Erweiterung der Diversion xx.                                                                                                                                                                                                                                                                                                                                    |
| .wh xx N |                  |                  | v   | Installation eines Traps zum Aufruf des Makros xx auf Seitenposition N. Ein negatives N wird auf das Seitenende bezogen. Ein vorher installierter Trap für N wird überschrieben. N = 0 bezieht sich auf den Seitenbeginn. Fehlt xx, so wird der zuletzt installierte Trap bezogen auf N gelöscht, so daß ein eventuell überschriebener Trap wieder wirksam wird. |
| .ch xx N |                  |                  |     | Wechseln der Trapposition des Makros xx auf N. Fehlt N, wird der Trap gelöscht.                                                                                                                                                                                                                                                                                  |
| .dt N xx |                  | aus              | D,v | Installation eines Diversionstraps auf Position N in der laufenden Diversion zum Aufruf des Makros xx. Ein weiteres dt belegt den Trap neu. Falls keine Argumente gegeben sind, wird der Trap gelöscht.                                                                                                                                                          |

| Kommando | Anfangs-<br>wert | ohne<br>Argument | KZ | Erläuterung                                                                                                           |
|----------|------------------|------------------|----|-----------------------------------------------------------------------------------------------------------------------|
| .it N xx |                  | aus              | E  | Setzen eines Eingabezeilenzähltraps nach der N-ten Eingabezeile zum Aufruf des Makros xx (Steuerzeilen zählen nicht). |
| .em xx   |                  |                  |    | Der Makro xx wird am Ende aller Eingabezeilen aufgerufen.                                                             |

9. Register

~~~~~

Eine Vielzahl von Parametern für die Textformatierung sind für den Anwender in vordefinierten Registern, sogenannten Zahlenregistern, verfügbar. Zusätzlich kann der Nutzer sich eigene Register definieren. Die Namen sind ein oder zwei Zeichen lang und werden in einer eigenen Liste geführt, so daß sie sich nicht von Kommandos, Makros oder Zeichenketten unterscheiden müssen. Außer einigen vordefinierten Nur-Lese-Registern können Register gelesen, geschrieben, inkrementiert und dekrementiert werden. Sie können benutzt werden, wenn ein numerischer Wert in den Input eingefügt werden soll und können Teil eines numerischen Ausdrucks sein (siehe Abschnitt 1.4.). Sie sind dann mit Variablen in einer Programmiersprache vergleichbar.

Register können mittels .nr definiert und modifiziert werden, wobei der Name, der zahlenmäßige Wert und der Wert der Erhöhung bei automatischer Inkrementierung bzw. Dekrementierung spezifiziert wird. Die Register können auch bei ihrer Einfügung modifiziert werden. Folgende prinzipielle Schreibweisen sind möglich, wobei x und xx die Namen, N der Wert und M der vereinbarte Inkrementwert für die Register sind.

Tabelle 9-1: Interpolation der Register

Sequenz	Auswirkung auf die Register	eingefügter Wert
\nx	keine	N
\n(xx	keine	N
\n+x	Inkrement um M	N + M
\n-x	Dekrement um M	N - M
\n+(xx	Inkrement um M	N + M
\n-(xx	Dekrement um M	N - M

Beim Einfügen werden die Registerinhalte in folgende ebenfalls vorher zu vereinbarende Formate konvertiert:

Format	Zahlenfolge	
l	0, 1, 2, 3, 4,...	(Standard)
00l	000, 001, 002,...	(dezimal mit Vornullen)
i	0, i, ii, iii, iv,...	(römisch klein)
I	0, I, II, III, IV,...	(römisch groß)
a	0, a, b, c, ..., z, aa,...	(Kleinbuchstaben)
A	0, A, B, ..., Z, AA, ..., ZZ, AAA,...	(Großbuchstaben)

Kommandoübersicht:

Kommando	Anfangs- wert	ohne Argument	KZ	Erklärung
.nr R +_NM			n	Dem Register R wird der vorzeichenbehaftete Wert N zugewiesen. M ist das Inkrement bei automatischer Inkrementierung.
.af R C	arabisch			Festlegung des Formates C für das Register R.
.rr R		ignoriert		Austragen des Registers R. Der intern benötigte Platz wird frei.

10. Tabulatoren, Leitpunkte und Felder
 ~~~~~

10.1. Tabulatoren und Leitpunkte  
 ~~~~~

Die KOI7-Steuerzeichen TAB und SOH (hier verwendet als Kennzeichen für Leitpunkte) können sowohl zur Erzeugung von Horizontalbewegungen als auch zur Erzeugung einer Kette gleicher Zeichen bis zu einer bestimmten Zeilenposition verwendet werden. Die Länge der erzeugten Bewegung bzw. Zeichenkette ist abhängig von mit .ta intern erzeugten Tabulatorstops. Standardmäßig erzeugen Tabulatoren Bewegungen und die Leitpunktzeichen Ketten von Punkten (.). Die Kommandos .tc und .lc bieten die Möglichkeit zwischen Bewegung und Zeichenketten zu wählen. Es gibt 3 Arten von internen Tabulatorstops: linksausgerichtete, rechtsausgerichtete und Zentrierung.

In der folgenden Übersicht ist D die Distanz von der laufenden Position in der Eingabezeile (bei der der Tabulator bzw. das Leitpunktkennzeichen gefunden wurde) bis zum nächsten Tabulatorstop. "Folgende Zeichenkette" besteht aus den Zeichen, die dem Tabulator (oder Leitpunkt) folgen bis zum nächsten Tabulator (oder Leitpunkt) oder dem Zeilenende. W ist die Weite der "folgenden Zeichenkette".

Tabulatorart	Länge der Bewegung bzw. der wiederholten Zeichen	Position der "folgenden Zeichenkette"
links	D	folgend auf D
rechts	D - W	rechtsausgerichtet innerhalb von D
Zentrierung	D - W/2	zentriert bezogen auf D

Tabulatoren und Leitpunkte werden im Kopiermodus nicht interpretiert, \t und \a erzeugen nicht interpretierbare Tabulatoren bzw. Leitpunkte im Normalmodus.

10.2. Felder
 ~~~~~

Ein Feld ist ein Freiraum, der innerhalb des Textes eingebaut wird. Er wird bestimmt von einer Zeichenkette, die innerhalb von "Feldbegrenzungszeichen" gehalten wird. Die Feldlänge wird durch die Distanz der Position auf der Eingabezeile, bei der die Zeichenkette beginnt und dem nächsten Tabulatorstop festgelegt. Alle nachfolgenden Zeichenketten, die innerhalb von Feldbegrenzungszeichen angegeben sind, werden innerhalb des festgelegten Freiraumes zentriert. Die Zeichenketten innerhalb eines Paares von Feldbegrenzungszeichen werden durch Freihaltezeichen, normalerweise dem Leerzeichen, begrenzt. Die Differenz zwischen Feldlänge und Zeichenkette kann negativ sein.

## Kommandoübersicht:

| Kommando | Anfangs-<br>wert | ohne<br>Argument | KZ  | Erklärung                                                                                                                                                                                                                      |
|----------|------------------|------------------|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| .ta Nt   | 0,8i             |                  | E,m | Setzen der Tabulatorstops auf N und der Art auf t. Fehlt t, dann gilt linksausgerichtet. Die Tabulatorstops (N) werden durch Leerzeichen getrennt. Vorzeichen "+" wird als Inkrement zu vorher eingestellten Werten behandelt. |
| .tc c    |                  |                  | E   | Das TAB-Wiederholungszeichen wird c; ohne Argument wird die normale Tabulatorfunktion wiederhergestellt.                                                                                                                       |
| .lc c    | .                |                  | E   | Das Leitpunktzeichen wird c.                                                                                                                                                                                                   |
| .fc ab   | aus              | aus              |     | Das Feldbegrenzungszeichen wird a, das Freihaltezeichen b, falls angegeben, sonst Leerzeichen. Fehlen die Argumente ganz, ist der Feldmodus ausgeschaltet.                                                                     |

## 11. Ein- und Ausgabe-Vereinbarung und Zeichenübersetzungen

### 11.1. Eingabe-Zeichenübersetzungen

Die ASCII-Steuerzeichen TAB (siehe Abschnitt 10.1.), SOH (siehe Abschnitt 10.1.) und Backspace (siehe Abschnitt 11.2.) werden dort diskutiert. Das Newline dient zur Begrenzung der Eingabezeilen. Die Steuerzeichen STX, ETX, ENQ, ACK und BEL werden akzeptiert und können z. B. als Begrenzer benutzt werden, alle anderen werden ignoriert. Das Escape-Zeichen "\ " leitet eine Steuerfolge ein, die eine besondere Bedeutung hat, siehe Anlage 2. Das Zeichen "\ " ist nicht mit dem KOI7-Steuerzeichen ESC zu verwechseln. Das Escape-Zeichen "\ " selbst kann über "\\ " eingegeben werden. Es kann mittels des Kommandos .ec in ein beliebiges anderes Zeichen gewechselt werden. Falls erforderlich kann der Escape-Mechanismus über .eo ausgeschaltet werden und über .ec wieder ein.

Kommandoübersicht:

| Kommando | Anfangs-wert | ohne Argument | KZ | Erklärung                          |
|----------|--------------|---------------|----|------------------------------------|
| .ec c    | \            | \             | -  | Setzen des Escape-Zeichens auf c   |
| .eo      | ein          |               |    | Ausschalten des Escape-Mechanismus |

### 11.2. Sonderfunktionen

Außer im Kopiermodus wird das Backspace-Zeichen durch eine Horizontalbewegung um die Weite eines Leerzeichen nach links ersetzt.

Das Bilden neuer Zeichen durch überschreiben wird im Abschnitt 13. beschrieben.

Unterstreichen in Form von geschlossenen Linien wird im Abschnitt 12.4. beschrieben.

nroff unterstreicht automatisch die Zeichen des Unterstrich-Zeichenvorrates, der in der Gerätetabelle gekennzeichnet ist (normalerweise Groß- und Kleinbuchstaben und Ziffern).

## Kommandoübersicht:

| Kommando | Anfangs-<br>wert | ohne<br>Argument | KZ | Erklärung                                                                                                |
|----------|------------------|------------------|----|----------------------------------------------------------------------------------------------------------|
| .ul N    | aus              | N = 1            | E  | Unterstreichen der nächsten N Eingabezeilen. Es werden jeweils nur Buchstaben und Ziffern unterstrichen. |
| .cu      | aus              | N = 1            | E  | Eine Variante zu .ul, die alle Zeichen unterstreicht.                                                    |

## 11.3. Kommandokennzeichen

~~~~~

Je nach Bedarf können die Kommandokennzeichen "." und "," in jedes andere Zeichen gewechselt werden, z. B. um gewisse Textstücke vor der Interpretation zu schützen. Zu beachten ist, daß dieser Wechsel mit der Gestaltung der innerhalb dieses Wechsels aufgerufenen Makros übereinstimmen muß.

Das Nicht-Break-Kommandokennzeichen "'" wird verwendet, um in bestimmten Situationen bei Kommandos, die die Break-Funktion einschließen, diese zu unterdrücken. Dies ist z. B. innerhalb von Kopfzeilen- bzw. Seitenendemakros erforderlich, die im Füll-Modus mittels Traps an Stellen aufgerufen werden können, bei denen sich noch Teile der Eingabezeile in Verarbeitung befinden.

Kommandoübersicht:

Kommando	Anfangs- wert	ohne Argument	KZ	Erklärung
.cc C	.	.	E	Das Steuerzeichen wird C oder, falls nichts angegeben, zurück auf "." gesetzt.
.c2 C	,	,		Das Nicht-Break-Kommandokennzeichen wird C oder, falls nichts angegeben, zurück auf "'" gesetzt.

11.4. Zeichenumschlüsselung

Mittels des Kommandos .tr ist es möglich, bestimmten Zeichen andere Zeichen zuzuordnen. Diese Umschlüsselung erfolgt zur Ausgabezeit.

Kommandoübersicht:

Kommando	Anfangs- wert	ohne Argument	KZ	Erklärung
.tr abcd	-	-		Umschlüsselung des Zeichens a zu b und c zu d. Bei ungerader Anzahl wird das letzte Zeichen in ein Leerzeichen übersetzt.

11.5. Textdurchlauf ohne Übersetzung

Eine Eingabezeile, die mit "\!" beginnt, wird im Kopiermodus gelesen und ohne weitere Verarbeitung wieder ausgegeben (ohne "\!"). Sie wird in keiner Weise in die Textverarbeitung (Füll-Modus, Zeilenzählung u. a.) einbezogen. Diese Arbeitsweise kann benutzt werden, um Steuerinformationen an einen Postprozessor weiterzugeben oder während einer Diversion Steuerzeichen in einen Makro abzuspeichern.

11.6. Kommentare und verdeckte Newlines

Eine unhandliche lange Eingabezeile, die aber zusammengehörig ist (z. B. eine Zeichenkettendefinition oder eine Parameterliste für einen Makroaufruf), kann auf mehrere Zeilen aufgeteilt werden, wenn die Zeilen außer der letzten mit "\(NL)" abgeschlossen werden. Kommentare werden durch "\" eingeleitet. Sie sind innerhalb von Makrodefinitionen sinnvoll. Eine Zeile, die mit "\" beginnt, erzeugt ausgabeseitig eine Leerzeile.

12. Bewegungen, Weitefunktion und Linien

~~~~~

### 12.1. Horizontale und vertikale Bewegungen

~~~~~

Die Steuerfolgen `\v'N'` und `\h'N'` dienen der Erzeugung vertikaler bzw. horizontaler Bewegungen. N kann negativ sein, eine positive Angabe bedeutet abwärts bzw. nach rechts. Um unvorherzusehende Verschiebungen zu vermeiden, ist es erforderlich, daß die vertikalen Bewegungen sich im Füll-Modus innerhalb eines Wortes ausgleichen. Folgende zusätzliche Steuerfolgen für vertikale Bewegungen sind zugelassen:

```
\u   eine halbe Zeile aufwärts
\d   eine halbe Zeile abwärts
\r   eine ganze Zeile aufwärts
```

Funktion zur Bestimmung der Weite einer Zeichenkette

~~~~~

Mit der Steuerfolge `\w'string'` kann die Weite einer Zeichenkette in Basiseinheiten erzeugt werden. Zum Beispiel nimmt das Kommando

```
.ti -\w'1. 'u
```

die Einrückung temporär um die Weite der Zeichenkette "1. " zurück. Zu beachten ist, daß die Maßeinheit u unbedingt anzugeben ist, da sonst das Kommando die Weite in Em's (siehe Abschnitt 2.3.) interpretieren würde. Es werden 2 Register gesetzt, in sb der Anfang und in st das Ende der Zeichenkette relativ zur Zeile.

### 12.2. Merken einer horizontalen Position

~~~~~

Die Steuerfolge `\kx` speichert die laufende horizontale Position in das Register x.

Zum Beispiel erzeugt die Konstruktion `\kxWort\h'|\nxu+2u'Wort` den Doppeldruck des Wortes Wwoorrtt.

12.3. Linienziehung

~~~~~

Die Steuerfolge `\l'Nc'` produziert eine Kette von N Zeichen c nach rechts. Ist c nicht angegeben, so wird der Unterstrich "\_" verwendet. Ist N negativ, so wird zunächst eine Bewegung nach links um die Distanz N ausgeführt, ehe die entsprechende Zeichenzahl ausgegeben wird.

Die Steuerfolge `\L'Nc'` erzeugt eine senkrechte Linie, bestehend aus dem Zeichen c. N ist die Anzahl von Vertikalvorschieben mit dem generierten Zeilenabstand. Ist kein c angegeben, wird der Senkrechtstrich `\(br)` verwendet. Ein positives N erzeugt eine Linie abwärts und ein negatives N eine Linie aufwärts.

### 13. Zeichenkonstruktion

~~~~~

13.1. Zeichenüberschreibung

~~~~~

Mit Hilfe der Steuerfolge `\o'string'` werden die Zeichen der Zeichenkette "string" übereinandergedruckt. Der übereinanderdruck erfolgt in der Weise, daß die einzelnen Zeichen zentriert werden. Die Weite wird von dem längsten Zeichen bestimmt. Die Steuerfolge `\o'o+` produziert `o+`.

#### 13.2. Zeichendruck ohne Vorschub

~~~~~

Die Steuerfolge `\zc` druckt das Zeichen C ohne Zeichenvorschub. Damit ist es ähnlich der Zeichenüberschreibung (siehe Abschnitt 13.1.) möglich, Zeichen zu überdrucken mit dem Unterschied, daß der überdruck linksausgerichtet erfolgt.

13.3. Große Klammern

~~~~~

Die Funktion `\b'string'` wird benutzt, um große Klammern zu konstruieren. Dabei werden die Zeichen der Zeichenkette 'string' senkrecht übereinander gebaut. Das erste Zeichen wird dabei an die Spitze und die folgenden jeweils darunter angeordnet. Mit Hilfe der Funktion für extraweiten Zeilenabstand (siehe Abschnitt 6.2.) ist für den entsprechenden Freiraum zu sorgen.

## 14. Silbentrennung

---

Die automatische Silbentrennung kann aus- bzw. eingeschaltet werden. Bei eingeschalteter Silbentrennung können verschiedene Varianten benutzt werden. Ein Kennzeichen (\%) kann in ein Wort eingebaut werden, um eine gewünschte Trennstelle zu kennzeichnen, oder aber dem Wort vorgestellt werden, um eine Trennung zu verhindern. Zusätzlich kann eine (kleine) Ausnahmewortliste definiert werden.

Es werden nur zusammenhängende Ketten von Alpha-Zeichen für die Silbentrennung berücksichtigt. Wörter, die das Trennzeichen (Minus), Em-Dash (\(em) oder das Silbentrenn-Kennzeichen enthalten, werden, falls erforderlich, immer an dieser Stelle getrennt, unabhängig, ob der Silbentrennmodus ein- oder ausgeschaltet ist.

Kommandoübersicht:

| Kommando     | Anfangswert | ohne Argument | KZ | Erklärung                                                                                                                                                                                                                                                                                                                                                               |
|--------------|-------------|---------------|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| .nh          | ein         | -             | E  | Ausschalten der automatischen Silbentrennung.                                                                                                                                                                                                                                                                                                                           |
| .hy N        | ein, N=1    | ein, N=1      | E  | Einschalten der automatischen Silbentrennung für $N > 1$ und Ausschalten für $N = 0$ . Falls $N = 2$ ist, werden letzte Zeilen einer Seite oder eines Abschnitts (allg. die einen Trap bewirken) nicht getrennt. Ist $N = 4$ , werden die letzten 2 Zeichen, bei $N = 8$ die ersten 2 Zeichen eines Wortes nicht mehr getrennt. $N = 14$ bedingen alle 3 Restriktionen. |
| .hc c        | \%          | \%            | E  | Das Kennzeichen für Silbentrennung wird auf c gesetzt. Es erscheint nicht in der Ausgabe.                                                                                                                                                                                                                                                                               |
| .hw Wort1... |             | ignoriert     |    | Spezifikation von Wörtern mit eingebauten Bindestrichen, die die Trennstelle in den Wörtern markieren (Ausnahmewortliste).                                                                                                                                                                                                                                              |

## 15. Titelzeilen

Die Titelfunktion `.tl` liefert die automatische Plazierung von 3 Feldern (links, rechts, Mitte) auf einer Zeile, deren Länge mit `.lt` festgelegt werden kann. Eine allgemeine Anwendung für `.tl` liegt in den Makros für den oberen und unteren Rand einer Seite.

Kommandoübersicht:

| Kommando                                   | Anfangs-<br>wert | ohne<br>Argument | KZ  | Erklärung                                                                                                                                                                                                                                                                                                                                       |
|--------------------------------------------|------------------|------------------|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>.tl 'links'<br/>Mitte'rechts'</code> |                  |                  |     | Die Zeichenketten "links", "Mitte" und "rechts" werden entsprechend linksausgerichtet, zentriert und rechtsausgerichtet. Falls in einer Zeichenkette das Seitennummernkennzeichen enthalten ist, (normalerweise %) wird der Inhalt des Registers % eingefügt. Als Begrenzer für die Zeichenketten kann ein beliebiges Zeichen verwendet werden. |
| <code>.pc c</code>                         | %                | -                | -   | Das Seitennummernkennzeichen wird auf C gesetzt oder gelöscht. Die Seitennummer selbst verbleibt im Register %.                                                                                                                                                                                                                                 |
| <code>.lt +_N</code>                       | 65               |                  | E,m | Die Länge des Titels wird auf +_N gesetzt. Die Zeilenlänge und die Titellänge sind voneinander unabhängig.                                                                                                                                                                                                                                      |

16. Ausgabezeilennumerierung

~~~~~

Mit dem Kommando `.nm` kann eine automatische Ausgabezeilennumerierung angewiesen werden. Ist dies der Fall, wird eine 3-stellige Nummer plus ein Leerzeichen den Ausgabezeilen vorangestellt. Jede Zeile erhält einen Offset um 4 Zeichen, die aktuelle Zeilenlänge wird beibehalten. Soll der rechte Rand mit früheren Zeilen übereinstimmen, muß eine Reduktion der aktuellen Zeilenlänge vorgenommen werden. Leerzeilen und anderer Leerraum sowie mit `.tl` erzeugte Zeilen werden nicht nummeriert. Die Numerierung kann temporär mittels `.nu` ausgesetzt werden. Zusätzlich kann eine Zeilennummern-Einrückung `I` sowie ein Freiraum `S` zwischen Nummer und Text vereinbart werden. Desweiteren kann angewiesen werden, daß nur die Zeilen nummeriert werden, deren Nummern ein Vielfaches von `M` sind, bei den anderen erscheint eine entsprechende Einrückung.

Kommandoübersicht:

Kommando	Anfangs- wert	ohne Argument	KZ	Erklärung
<code>.nm +_N M S I</code>		aus	E	Wenn <code>+_N</code> angegeben ist, wird die nächste Ausgabezeile mit <code>+_N</code> nummeriert. Standardwerte sind <code>M = 1</code> , <code>S = 1</code> , <code>I = 1</code> . Eine Nichtziffer kennzeichnet ein fehlendes Argument. Fehlen alle Argumente, wird die Zeilennumerierung ausgeschaltet. Die nächste Zeilennummer wird für evtl. spätere Benutzung im Register <code>.ln</code> gerettet.
<code>.nu N</code>		<code>N = 1</code>	E	Die nächsten <code>N</code> Zeilen werden nicht nummeriert.

17. Bedingte Akzeptierung des Inputs

~~~~~

Die Eingabe von "Text" (Eingabetext, Kommandos, Makros u. ä.) kann von bestimmten Bedingungen abhängig gemacht werden. Folgende Möglichkeiten zur Verzweigung bestehen:

|                                            |                                                                                                                                                                                                                                                                                                                       |
|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>.if c Text</code>                    | Falls die Bedingungen <code>c</code> wahr ist, wird der nachfolgende Text akzeptiert. Handelt es sich um mehrere Zeilen, muß die erste mit <code>{\</code> beginnen und die letzte mit <code>\}</code> enden. <code>c = e</code> bedeutet, die Seitennummer ist gerade, <code>c = o</code> Seitennummer ist ungerade. |
| <code>.if !c Text</code>                   | Ist <code>c</code> falsch, wird Text akzeptiert.                                                                                                                                                                                                                                                                      |
| <code>.if N Text</code>                    | Ist der Ausdruck <code>N &gt; 0</code> , wird Text akzeptiert.                                                                                                                                                                                                                                                        |
| <code>.if !N Text</code>                   | Ist der Ausdruck <code>N &lt;= 0</code> , wird Text akzeptiert.                                                                                                                                                                                                                                                       |
| <code>string1'string2'Text'&lt;30m</code>  | <code>.ie "t"t" .ie "1" .ta 30 .if 'string1'string2'Text</code> Ist die Zeichenkette <code>string1</code> identisch zu <code>string2</code> , wird Text akzeptiert.                                                                                                                                                   |
| <code>string1'string2' Text'&lt;30m</code> | <code>.ie "t"t" .ie "1" .ta 30 .if !'string1'string2' Text</code> Ist die Zeichenkette <code>string1</code> nicht identisch zu <code>string2</code> , wird Text akzeptiert.                                                                                                                                           |
| <code>.ie ...</code>                       | If-Teil von if-else, alle Formen von if sind möglich.                                                                                                                                                                                                                                                                 |
| <code>.el ...</code>                       | Else-Teil von if-else, alle Formen von if sind möglich.                                                                                                                                                                                                                                                               |

Leerzeichen zwischen den Bedingungen und Text werden überlesen. Paarungen von `.ie` und `.el` können auch geschachtelt verwendet werden.

18. Umschalten des Formatierungsumfeldes  
(Environment)

~~~~~

Eine Anzahl von Parametern, in der Kommandozusammenstellung mit E gekennzeichnet, sind in einem Environment zusammengefaßt, das der Nutzer wechseln kann. Damit ist es möglich, innerhalb des Textaufbaus auf ein anderes Grundformat umzuschalten, z. B. innerhalb einer Diversion. Maximal 3 verschiedene Environments sind gleichzeitig nutzbar.

Kommandoübersicht:

Kommando	Anfangs- wert	ohne Argument	KZ	Erklärung
.ev N	N = 0	vorher		Einschalten des Formatierungsfeldes N = 0,1,2

19. Einfügen von Eingaben der Standardeingabe

~~~~~

Die Eingabe kann temporär auf die Standardeingabe umgelegt werden. Bei MUTOS ist das normalerweise die BDE, es kann aber auch ein Filter oder ein File sein. Beabsichtigt wird damit, spezielle Formate, Adressen, Titel u. a. unterzubringen.

Mit 2 Newlines wird die Eingabe wieder zurückgeschaltet.

Kommandoübersicht:

| Kommando   | Anfangs-<br>wert | ohne<br>Argument | KZ | Erklärung                                                                                                                                              |
|------------|------------------|------------------|----|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| .rd prompt | -                | prompt =<br>BEL  |    | Umschalten der Eingabe auf die Standardeingabe bis 2 (NL)s in Folge erscheinen. .rd wird wie ein Makro behandelt; nach prompt können Argumente folgen. |
| .ex        |                  |                  |    | Beenden von nroff. .ex wird genauso behandelt, als wäre EOF im letzten Eingabefile gefunden worden.                                                    |

## 20. Ein-äusgabe-File-Umschaltung

Kommandoübersicht:

| Kommando     | Anfangs-<br>wert | ohne<br>Argument | KZ | Erklärung                                                                                                                                     |
|--------------|------------------|------------------|----|-----------------------------------------------------------------------------------------------------------------------------------------------|
| .so filename |                  |                  |    | Die Eingabe erfolgt von dem angegebenen File. Wird EOF gefunden, wird im Anfangsfile fortgesetzt.<br>so-Kommandos können geschachtelt werden. |
| .nx filename |                  |                  |    | Wie .so, nur daß keine Rückkehr erfolgt.                                                                                                      |
| .cf filename |                  |                  |    | Das angegebene File wird ohne jede Verarbeitung in die Ausgabe kopiert. Bei Fileende wird zur vorhergehenden Eingabe zurückgekehrt.           |
| .of filename |                  | vorher           | B  | Die Ausgabe erfolgt nach dem angegebenen File. Die Rücknahme der Ausgabeumschaltung erfolgt mit .fo ohne Filenamen.                           |
| .pi program  |                  |                  |    | Der Output erfolgt nach "program". Es können keine Argumente übergeben werden.                                                                |

21. Weitere verschiedene Kommandos

Kommandoübersicht:

| Kommando   | Anfangs-<br>wert | ohne<br>Argument | KZ  | Erklärung                                                                                                                                                                                                                                                  |
|------------|------------------|------------------|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| .mc c N    |                  | aus              | E,m | Das Zeichen c wird in einem Abstand von N rechts von der rechten Seitenkante auf den nächsten nicht leeren (außer mit .tl erzeugten) Ausgabezeilen ausgegeben. Falls kein N gegeben ist, wird der vorherige Wert angenommen. Initialisiert ist N mit 2 em. |
| .tm string |                  | (NL)             |     | Die Zeichenkette "string" wird im Kopiermodus gelesen und auf dem Terminal ausgegeben.                                                                                                                                                                     |
| .ig yy     |                  |                  |     | Ignorieren der Eingabezeile; wird wie .de behandelt, nur daß die Eingabe überlesen wird. Der Input wird im Kopiermodus gelesen und Autoinkrementregister werden, falls enthalten, berücksichtigt.                                                          |
| .pm t      |                  |                  |     | Druck der Makros. Name, Größe und Inhalt der verwendeten Makros werden auf dem Terminal ausgegeben.                                                                                                                                                        |
| .fl        |                  |                  | B   | Ausgabe des Output-Puffers; wird hauptsächlich für interaktive Testzwecke benutzt.                                                                                                                                                                         |

## 22. Mitteilungen

~~~~~

Der Output von den Kommandos .tm, .pm und die Anforderung von .rd erfolgen ebenso wie verschiedene Fehlermitteilungen auf der Standardmitteilungsausgabe. Die letztere ist nicht identisch mit dem Standardoutput. Standardmäßig erfolgen zwar beide auf dem Bedienterminal, sie können aber unterschiedlich zugewiesen werden.

Verschiedene Fehler können während der Arbeit von nroff auftreten. Bei weniger ernsten Fehlern wird die Textverarbeitung fortgesetzt. Wenn möglich, wird im Output die Zeile, bei der der Fehler auftrat, markiert. Bei ernsten Fehlern, bei denen eine Fortsetzung wenig sinnvoll erscheint, z. B. beim Zugriff auf nichtvorhandene Files oder beim überschreiten interner Grenzen wird die Verarbeitung abgebrochen und eine entsprechende Mitteilung ausgegeben.

Anlage A Alphabetische Zusammenstellung der Kommandos

Kommando	Abschnitt	KZ	Erklärung
ad	5.	E	Randausgleich
af	9.		Format Zahlenregister
am	8.		Erweitern Makro
as	8.		Erweitern String
bp	4.	B	Seitenbeginn
br	5.	B	Break
c2	11.3.		Wechsel Kommandokennzeichen
cc	11.3.		Wechsel Kommandokennzeichen
ce	5.	B,E	Zentrierung
ch	8.		Wechsel Trapposition
cu	11.2.	E	Unterstreichung
da	8.	D	Erweitern Diversion
de	8.		Makrodefinition
di	8.	D	Diversion
ds	8.		Definiere String
dt	8.	D,v	Diversionstrap
ec	11.1.		Setzen Escape-Zeichen
el	20.		Else-Teil von if-else
em	8.		Endemakro
eo	11.1.		Ausschalten Escapemodus
ev	21.		Umschalten Environment
ex	22.		Beenden nroff
ie	20.		if-else
if	20.		if
ig	24.		Ignorieren Eingabe
in	7.	B,E,m	Einrückung
it	8.	E	Eingabezeilentrap
lc	10.	E	Wechseln Leaderzeichen
ll	7.	E,m	Zeilenlänge
ls	6.	E	mehrzeilige Ausgabe
lt	18.	E,m	Länge Titelzeile
mc	24.	E,m	Zeichenausgabe rechts
mk	4.	D	Merken Seitenposition
na	5.	E	kein Randausgleich
ne	4.	D,v	Test Freiraum
nf	5.	B,E	kein Füll-Modus
nh	17.	E	keine Silbentrennung
nm	19.	E	Zeilennumerierung
nr	9.	u	Definition Zahlenregister
ns	6.	D	No-Space-Modus
nx	23.		Umschalten Eingabedatei
os	6.		Ausgabe Freiraum
pc	18.		Setzen Seitennummernkennzeichen
pi	23.		Ausgabe nach Programm
pl	4.	v	Seitenlänge
pm	34.		Ausgabe Makros
pn	4.		Setzen Seitennummer
po	4.	v	Seitenoffset
rd	19.		Lesen Standardeingabe
rm	8.		Löschen Makro
rn	8.		Umbenennen Makro

Anlage 1 (Fortsetzung)

Kommando	Abschnitt	KZ	Erklärung
rr	9.		Löschen Register
rs	6.	D	Einschalten Space-Modus
rt	4.	D,v	zurück nach Seitenposition
so	23.		Umschalten Eingabedatei
sp	6.	B,v	Leerzeilenausgabe
sv	6.	v	Retten Freiraum
ta	10.	E,m	Tabulatoren setzen
tc	10.	E	Tab-Wiederholungszeichen
ti	7.	B,E,m	temporäre Einrückung
tl	18.		Titelzeile
tm	26.		Mitteilungsausgabe
tr	11.4.		Zeichenumschlüsselung
uf	3.		Schriftart Unterstreichung
ul	11.2.	E	Unterstreichung
vs	6.	E,p	Zeilenabstand
wh	8.		Installieren Trap

Anlage B Zusammenstellung der Steuerfolgen

Steuerfolge	Abschnitt	Bedeutung
\\	11.1.	nicht zu interpretierendes \
\.	8.	siehe de
\&	5.1.	Unterdrückung der Interpretation
\!	11.5.	
\"	11.6.	Beginn Kommentar
\\$N	8.3.	Parametersubstitution
\%	17.	Silbentrenn-Kennzeichen
\(xx	3.	Zeichename xx
X,(xx	8.1.	Stringsubstitution x bzw. xx
\a	11.1.	nicht zu interpretierendes SOH
\b'abc...'	16.3.	Klammerfunktion
\c	5.2.	unterbrochener Text
\d	12.	1/2 Zeile vorwärts
\fx	3.	Wechsel der Schriftart
\h'N'	12.	horizontale Bewegung um N
\kx	14.	Merken der horizontalen Position
\l'Nc'	15.	horizontale Linie
\L'Nc'	15.	vertikale Linie
\nx,\n(xx	9.	Substitution Zahlenregister
\o'abc...'	16.1.	Overstrike-Funktion
\p	5.1.	Break und Ausspreizen der Zeile
\r	12.	1 Zeile rückwärts
\t	11.1.	nicht zu interpretierender TAB
\n	12.	1/2 Zeile rückwärts
\v'N'	12.	vertikale Bewegung um N Einheiten
\w'string'	13.	Interpolation der Weite von "string"
\x'N'	6.2.	extra weiter Zeilenabstand
\zc	16.2.	Zeichendruck ohne Vorschub
\{	20.	Beginn der bedingten Eingabe
\}	20.	Ende der bedingten Eingabe
\Newline	11.6.	ignoriertes Endezeichen

Anlage C Allgemeine vordefinierte Zahlenregister

Registername	Abschnitt	Bedeutung
%	4.	laufende Seitennummer
dl	8.4.	horizontale Weite der letzten Diversion
dn	8.4.	vertikale Größe der letzten Diversion
dw		laufender Tag der Woche (1-7)
dy		laufender Tag des Monats (1-31)
hp	14.	horizontale Position Eingabezeile
ln	19.	Ausgabezeilennummer
mo		laufender Monat (1-12)
nl	5.1.	Vertikalposition der letzten Zeile
sb	13.	Anfangsposition String
st	13.	Endposition String
yr		letzte 2 Ziffern der Jahreszahl
Nur-Lese-Register		
.c		Anzahl der Zeilen der laufenden Eingabedatei
.d	8.4.	laufende vertikale Position der aktuellen Diversion
.h	5.1.	Füllungsstand der laufenden Seite oder Diversion
.i	7.	laufender Ident
.l	7.	laufende Zeilenlänge
.n	5.1.	Textlänge der letzten Ausgabezeile
.o	4.	laufender Seitenoffset
.p	4.	laufende Seitenlänge
.t	8.5.	Distanz zum nächsten Trap
.u	5.1.	Füll-Modus
.v	6.	laufender vertikaler Zeilenabstand
.w	12.2.	Weite des letzten Zeichens

Anlage D Tabelle der Eingabenamen fuer Sonderzeichen

Zeichen	Eingabe	Bezeichnung
-	<code>\(hy</code>	hyphen
o+	<code>\(bu</code>	bullet
[]	<code>\(sq</code>	square
-	<code>\(em</code>	3/4em
-	<code>\(ru</code>	rule
$\frac{1}{4}$	<code>\(14</code>	1/4
$\frac{1}{2}$	<code>\(12</code>	1/2
$\frac{3}{4}$	<code>\(34</code>	3/4
-	<code>\(mi</code>	equation minus
o	<code>\(de</code>	degree
-	<code>\(dg</code>	dagger
'	<code>\(fm</code>	foot mark
'	<code>\(aa</code>	acute accent
`	<code>\(ga</code>	grave accent
-	<code>\(ul</code>	underrule
/	<code>\(sl</code>	slash
ä	<code>\(*a</code>	ae
Ä	<code>\(*A</code>	Ae
ö	<code>\(*O</code>	Oe
ö	<code>\(*o</code>	oe
ü	<code>\(*u</code>	ue
Ü	<code>\(*U</code>	Ue
ß	<code>\(*s</code>	sz
Σ	<code>\(*S</code>	Sigma
> _	<code>\(>=</code>	greater equal
< _	<code>\(<=</code>	lower equal
= _	<code>\(==</code>	identically equal
≈	<code>\(≈=</code>	approx =
≈	<code>\(ap</code>	approximates
≠	<code>\(!=</code>	not equal
->	<code>\(-></code>	right arrow
<-	<code>\(<-</code>	left arrow
^	<code>\(ua</code>	up arrow
v	<code>\(da</code>	down arrow
=	<code>\(eq</code>	equation equal
x	<code>\(mu</code>	multiply
/	<code>\(di</code>	divide
+ -	<code>\(+-</code>	plus-minus
U	<code>\(cu</code>	cup (union)
	<code>\(si</code>	set italic
	<code>\(ri</code>	reset italic
∞	<code>\(if</code>	infinity
+	<code>\(pl</code>	equation plus
(R)	<code>\(rg</code>	registered
(C)	<code>\(co</code>	copyright
	<code>\(br</code>	box vert rule
c/	<code>\(ct</code>	cent sign
=	<code>\(dd</code>	dbl dagger
*	<code>\(**</code>	math *
	<code>\(or</code>	or
o	<code>\(ci</code>	circle