

robotron

Systemhandbuch

MUTOS – 8000

Teil III

Textverarbeitung nroff

Programmtechnische Beschreibung
fuer Buerocomputer A5120.16

SYSTEMHANDBUCH

MUTOS 8000

Teil III

Textverarbeitung nroff

~~Ingenieurhochschule Mittweida
— Sektion Informationselektronik —
925 Mittweida, Platz der DSF 17
Fernruf 580~~

1

VEB Robotron-Buchungsmaschinenwerk
Karl-Marx-Stadt
Stand : 12/85

Jens Krause
Am Försterweg 32
O-1260 Strausberg

Die vorliegende Dokumentation entspricht dem Stand 12/85.
Nachdruck, jegliche Vervielfaeltigung oder Auszuege daraus
sind unzuellaessig.

Im Interesse einer staendigen Weiterentwicklung werden alle
Leser gebeten, ihre Vorschlaege bzw. Hinweise dem
VEB Robotron-Buchungsmaschinenwerk
9010 Karl-Marx-Stadt
Annabergerstrasse 93
mitzuteilen.

Fuer das Betriebssystem MUTOS 8000 wurden folgende
Dokumentationen erarbeitet:

Anwendungsbeschreibung MUTOS 8000

Systemhandbuch MUTOS 8000 Teil I

Abschnitt 1 Kommandos
Abschnitt 2,3 Systemrufe, Bibliotheksfunktionen
Abschnitt 4, 5, 7, 8 Special Files, Fileformate, Makropakete,
Systemunterstuetzung

Systemhandbuch MUTOS 8000 Teil II

Abschnitt 1 Kommandointerpreter Shell
Abschnitt 2 Editor
Abschnitt 3 Testhilfsprogramm adb

Systemhandbuch MUTOS 8000 Teil III

Abschnitt 1 Textverarbeitung nroff

Sprachbeschreibungen

Assemblersprachbeschreibung U 8000

Sprachbeschreibung C-Sprache

Das Sachwortverzeichnis ist als Anlage in der
Anwenderbeschreibung enthalten.

Zur Programmentwicklung und Programmtestungen fuer den
Prozessor U 8000 werden zusaetzlich folgende Dokumentationen
angeboten:

Monitortestsystem MON8000

CROSS-Software U8000 zum Betriebssystem UDOS

Assembler U8000ASM

Binder ZLINK

Absolutbinder IMAGER

Preprozessor INCLUDE

<u>Inhaltsverzeichnis</u>		<u>Seite</u>
1.	Einfuehrung	5
2.	Allgemeine Erklarungen	5
2.1.	Form der Eingabe	5
2.2.	Formatierer- und Geratedefinition	5
2.3.	Numerische Parametereingabe	6
2.4.	Numerische Ausdruecke	6
2.5.	Notation	6
3.	Zeichenvorrat und Schriftarten	7
4.	Seitensteuerung	8
5.	Textauffuellen, Randausgleich, Zentrieren	10
5.1.	Fuellen und Ausrichten	10
5.2.	Unterbrochener Text	10
6.	Zeilenvorschub (Spacing)	12
6.1.	Basiszeilenabstand	12
6.2.	Extra-Zeilenabstand	12
6.3.	Bloecke von Zeilenvorschueben	12
7.	Zeilenlaenge und Einruecken	14
8.	Makros, Strings, Diversionen und Positionstraps	14
8.1.	Makros und Strings	14
8.2.	Eingabeinterpretation im Kopiermodus (Copy-Mode)	15
8.3.	Argumente	15
8.4.	Diversionen	16
8.5.	Traps	17
9.	Zahlenregister	20
10.	Tabulatoren, Leitpunkte (Leaders) und Felder	22
10.1.	Tabulatoren und Leitpunkte	22
10.2.	Felder	22
11.	Ein- und Ausgabe-Vereinbarung und Zeichenueber- setzungen	23
11.1.	Eingabe-Zeichenuebersetzungen	23
11.2.	Sonderfunktionen	24
11.3.	Steuerzeichen	24
11.4.	Zeichenumschluesselung	25
11.5.	Textdurchlauf ohne Uebersetzung	25
11.6.	Kommentare und verdeckte Newline	26
12.	Bewegungen, Weitfunktion und Linien	26
12.1.	Horizontale und vertikale Bewegungen	26
12.2.	Funktion zum Bestimmen der Weite einer Zeichen- kette	26
12.3.	Merken einer horizontalen Position	27
12.4.	Linienziehen	27

13.	Zeichenkonstruktion	27
13.1.	Zeichenuberschreibung	27
13.2.	Zeichendruck ohne Vorschub	27
13.3.	Grosse Klammern	27
14.	Silbentrennung	28
15.	Titelzeilen	29
16.	Ausgaben-Zeilenummerierung	30
17.	Bedingtes Akzeptieren der Eingabe	30
18.	Umschalten der Formatierungsumgebung (Environment)	31
19.	Einfuegen von Eingaben der Standardeingabe	32
20.	Input/Output-File-Umschaltung	32
21.	Weitere verschiedene Kommandos	33
22.	Mitteilungen	34
Anlage 1	Alphabetische Zusammenstellung der Kommandos	35
Anlage 2	Zusammenstellung der Steuerfolgen	37
Anlage 3	Allgemeine vordefinierte Zahlenregister	38
Anlage 4	Tabelle der Eingabenamen fuer Sonderzeichen	39

1. Einfuehrung

nroff ist ein Textformatierungsprogramm mit hohem Komfort zum vollautomatischen Herstellen vielgestaltiger Texte und kompletter Dokumentationen.

Im Text eingebaute Kommandos und eingestreute "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 moeglich, den Aufwand bei der Texterfassung gering zu halten und ein relativ schnelles Einarbeiten zu gewährleisten.

nroff baut auf die jeweiligen Faehigkeiten von Seriendruckern auf. Die Anpassung an geratespezifische Besonderheiten, z.B. moeglichen Zeilenabstand, Zeichenabstand u.a., kann durch Modifikation der Geratetabelle erfolgen.

2. Allgemeine Erklarungen

2.1. Form der Eingabe

Die Eingabe enthaelt das zum Druck bestimmte Manuskript, vermischt mit Steuerzeilen, die das Formatieren des Textes in der gewuenschten Form veranlassen. Steuerzeilen beginnen mit einem Kommandokennzeichen, normalerweise "." oder "'", gefolgt von einem Namen, der aus einem oder zwei Zeichen besteht. Dieser Name spezifiziert entweder einen Ruf zu einem Basiskommando oder die Substitution eines nutzerdefinierten Makros. Namen muessen von den nachfolgenden Parametern durch ein Leerzeichen getrennt sein. Steuerzeilen mit nichterkannten Namen werden ignoriert.

Zusaetzlich koennen in normale Textzeilen Steuerfolgen, auch als Escape-Folgen bezeichnet, eingebaut werden, die mit dem internen Escape-Zeichen, normalerweise "\", beginnen und die Ausfuehrung einer Spezialfunktion bewirken.

2.2. Formatierer- und Geratedefinition

nroff benutzt als interne Bezugsgroesse 240 Einheiten/Inch entsprechend dem kleinsten gemeinsamen Vielfachen der horizontalen und vertikalen Geratedefinition der verschiedenen Ausgabe-geraete. Die tatsaechlichen Gerategegebenheiten werden in einer Geratetabelle spezifiziert. Mit der "T"-Option im nroff-Kommando kann auf die jeweilige Geratetabelle Bezug genommen werden (Siehe nroff (1)).

2.3. Numerische Parametereingabe

proff akzeptiert bei einer Vielzahl von Kommandos numerische Parameter mit angehaengerter Masseinheit.

Tabelle 1: Masseinheiten

Masseinheit	Bedeutung	Basiseinheiten
i	Inch	240
c	Zentimeter	240*50/127
P	Pica=1/6 Inch	240/6
m	Em (Breite des Buchstaben 'm')	C
n	En (Breite des Buchstaben 'n')	C
p	Point=1/72 Inch	240/72
u	Basiseinheit	1
v	vertikaler Zeilenabstand	V

C ist die Zeichenbreite und V der Zeilenabstand. Beide Groessen sind geraeteabhaengig. Die berechneten Werte werden auf die geraetespezifischen Gegebenheiten aufgerundet.

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

Das Zeichen "l" 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 Ausdruecke

Bei numerischen Eingaben koennen Ausdruecke mit Operatoren einschliesslich Klammern verwendet werden. Es sind die arithmetischen Operatoren

+, -, /, *, %(modulo)

und die logischen Operatoren

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

zugelassen. Das Aufloesen der Ausdruecke 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 tatsaechlich als Vorzeichen zu werten ist.

3. Zeichenvorrat und Schriftarten

nroff kann prinzipiell nur Zeichen des ISO-Code-Zeichenvorrates ausgeben. Zusaeztlich koennen Zeichen konstruiert werden. Die zusaeztlichen Zeichen werden in der Form \(\xx, wobei xx der Name des Zeichens ist, eingegeben. Die Namen der zusaeztlichen Zeichen sind in Anlage 4 aufgefuehrt. Die genaue Behandlung dieser konstruierten Zeichen ist geraeteabhaengig und in den entsprechenden Tabellen zu generieren.

nroff kennt 3 Schriftarten: Normal- (R), Schraeg-(I) und Breitdruck (B). Schraegdruck wird bei nroff durch Unterstreichen realisiert. Breitdruck ist abhaengig von den Faehigkeiten des Ausgabegeraetes. Die Zeichen werden auf doppelte Breite vergroessert. Beim Drucker 1152 wird Breitdruck durch Fettdruck realisiert. Ein Hervorheben einzelner Woerter ist durch Fettdruck moeglich (siehe Punkt 12.3.). Gestattet das Ausgabegeraet die Schriftart "schraeg" (z.B. SD 1157), kann sie mit der Steuerfolge ein- und ausgeschaltet werden.

Soll die Darstellung der Umlaute durch die echten deutschen Umlaute erfolgen, so empfiehlt sich das Umschalten auf verschiedene Schriftarten innerhalb von Makros durchzufuehren, um die Moeglichkeit einer jeweils spezifischen Realisierung der Umlaute zu haben. Zu beachten ist, dass verschiedene Ausgabegeraete (z.B. SD 1157) die Schriftartenumschaltung mit jeder neuen Zeile zuruecksetzen!

Tabelle 2: Kommandoübersicht

Kommando	Anfangswert	ohne Argument	KZ	Erklärung
.ft F	R		E	Wechsel der Schriftart F. Analog dazu kann auch die Steuerfolge mit der gleichen Wirkung verwendet werden. Fuer F koennen die Buchstaben R, I und B oder die zugeordneten Zahlen 1, 2, 3 verwendet werden.
.uf F	I	I		Setzen der Schriftart Unterstreichen fuer Schriftart F, F = 2, 3.

Zur Erklärung der Tabelle:

Unter Anfangswert ist der im Programm initialisierte Wert fuer den entsprechenden Parameter angegeben. In der Spalte KZ sind die Kurzzeichen mit folgender Bedeutung angegeben:

- B Das Kommando bewirkt einen Abbruch der Zeile (Break), (siehe Punkt 5.).
- D Der Modus bzw. die entsprechenden Parameter sind mit der jeweiligen Diversionsebene verbunden.
- E Die Parameter sind Teil der aktuellen Umgebung (siehe Punkt 21.).
- v,p,m,n Die jeweilige Standardmasseinheit fuer 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 ueblich, dafuer 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 7.5.).

Die Informationen zum Bilden des oberen Randes der 1. Seite muessen nroff vor dem Trap zur 1. Seite bekannt sein. Die physische Begrenzung fuer eine Seite ist g raeteabhaengig.

Tabelle 3: Kommandoubersicht

Kommando	Anfangswert	ohne 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	Seitenbeginn. Die laufende Seite wird ausgegeben und eine neue Seite wird begonnen. Ist ±N gegeben, wird die neue Seitennummer ±N.
.pn ±N	N = 1	ignoriert		Setzen der Seitennummer auf ±N. Die laufende Seitennummer wird im %-Register geführt.
.po ±N			v	Seitenverschiebung. Der aktuelle linke Rand wird auf ±N gesetzt. Die aktuelle Seitenverschiebung wird im .o-Register geführt.
.ne N			D,v	Ein verfügbarer Abstand N wird bis zur nächsten Traposition 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 einer Seite zu beginnen.
.mk R		intern	D	Merken des aktuellen vertikalen Platzes 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. Textauffuellen, Randausgleich, Zentrieren

5.1. Fuellen und Ausrichten

Die Worte werden vom Eingabetext gesammelt und in die Ausgabezeile umgesetzt, bis nichts mehr in die Zeile passt. Anschliessend werden die Zwischenraume zwischen den Worten der Ausgabezeile vergrössert, um die Zeile auf die aktuelle Laenge zu spreizen (abzueglich einer eventuellen Einrueckung).

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

In `proff` sind die Wortzwischenraume nicht gleich, mit der `proff-Kommando-Option -e` kann eine Umformierung der Zwischenraume versucht werden. Textauffuellen und Randausgleich (siehe Punkt 12.) koennen ganz verhindert oder gesteuert werden.

Die Textlaenge der letzten Ausgabezeile ist im `.n-Register` und die Seitenposition dieser Zeile im `n1-Register` verfuegbar. Der Fuellungsstand der Seite steht im `.h-Register`.

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

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

Einer Texteingabezeile, die mit einem Kommandokennzeichen eingeleitet wird, kann der Status einer Steuerzeile durch Vorsatz von `\&` genommen werden. Eine andere Moeglichkeit hierfuer ist die Translation des Kommandokennzeichens in ein anderes passendes Zeichen mit `.tr` (siehe Punkt 11.3.).

5.2. Unterbrochener Text

Das Kopieren der Eingabezeile im Nichtfuell-Modus kann durch ein `\c` am Ende der Zeile unterbrochen werden. Die naechste entschluesselte Eingabezeile wird dann als Fortsetzung dieser Zeile gedeutet. Analog dazu kann im Fuell-Modus ein Wort unterbrochen und fortgesetzt werden. Bewirken dazwischen eingebaute Steuerzeilen einen Break, so wird die angefangene Zeile ausgegeben.

Tabelle 4: Kommandoübersicht

Kommando	Anfangs- wert	ohne Argument	KZ	Erklärung
.br			B	Break. Das Auffüllen der laufenden Zeile wird gestoppt. Die Zeile wird ohne Randausgleich ausgegeben. Mit einem Leerzeichen beginnende Zeilen und Leerzeilen bewirken ebenfalls einen Break.
.fi	ein		B,E	Einschalten des Fuell-Modus Register .u = 1 Fuell-Modus, .u = 0 Nichtfuell-Modus.
.nf			B,E	Ausschalten des Fuell-Modus und des Randausgleiches. Eingabezeilen werden ohne Rücksicht auf die Zeilenlänge zur Ausgabe kopiert.
.ad c	ein	ein	E	Einschalten Randausgleich. Im Nichtfuell-Modus wird der Randausgleich bis zum Einschalten des Fuell-Modus unterdrückt. c = Typindikator c Randausgleichtyp ----- l linker Rand r rechter Rand c Zentrieren b oder c links und rechts
.na			E	Ausschalten Randausgleich. Der Fuell-Modus wird davon nicht betroffen.
.ce N	aus	N = 1	B, E	Zentrieren 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 aufeinander folgenden Normalzeilen ist bei nroff gerateabhaengig und in der jeweiligen Geratetabelle anzugeben.

6.2. Extra-Zeilenabstand

Benoetigt ein Wort eine extra grosse Vertikalkonstruktion, kann dieses Wort mit der Spezialfunktion \x'N' verbunden werden. Ist N negativ, wird der gewuenschte Zeilenvorschub vor der Zeile ausgefuehrt, die die Spezialfunktion enthaelt. Ist N positiv, wird zunaechst die Zeile ausgegeben und dann der Zeilenvorschub ausgefuehrt.

6.3. Bloecke von Zeilenvorschueben

Ein Block von Zeilenvorschueben wird in der Regel mit .sp vereinbart. Im No-Space-Modus wird .sp ignoriert. Ein zusammenhaengender Block von Leerzeilen kann mit .sv reserviert werden.

Tabelle 5: Kommandoübersicht

Kommando	Anfangs- wert	ohne Argument	KZ	Erklärung
.vs N	12		E,p	Setzt den Zeilenabstand auf N Points
.ls N	N = 1		E	N Leerzeilen werden jeder Ausgabezeile angehängt. Bei Erreichen einer Trapposition wird die Leerzeilenausgabe unterdrückt.
.sp N			E,v	Relative Vertikalpositionierung um Grösse N. Ist N negativ, wird rückpositioniert-begrenzt durch obere Seitenkante. Ist N positiv, wird vorwaertspositioniert-maximal bis zur naechsten Trapposition.
.sv N			v	Freihalten eines N grossen Freiraumes innerhalb der Ausgabe. Ist N kleiner als die Distanz bis zur naechsten Trapposition, so wird I, die Vorwaertspositionierung, ausgefuehrt; No-Space-Modus spielt dabei keine Rolle. Ist N groesser, so wird die Vorwaertspositionierung zurueckgehalten (siehe .os).
.os				Ausgabe eines vorher mit .sv geretteten Freiraumes.
.ns			D	Einschalten des No-Space-Modus. Unterdruecken 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. Zeilenlaenge und Einruecken

Die maximale Zeilenlaenge fuer den Fuellmodus wird mit `.ll` eingestellt. Einrueckungen werden mit `.in` gesetzt.

Ein temporaeres Einruecken, das nur fuer die naechste Zeile gilt, wird mit `.ti` eingestellt. Die Zeilenlaenge abzueglichen einem eventuell vereinbarten Einruecken ist die Basis fuer die Zentrieren (`.ce`).

Die laufende Zeilenlaenge wird im Register `.l` und die laufende Einrueckung im Register `.i` gefuehrt.

Tabelle 6: Kommandoubersicht

Kommando	Anfangswert	ohne Argument	KZ	Erklaerung
<code>.ll ±N</code>	6.5in	bleibt	E,m	Die Zeilenlaenge wird auf $\pm N$ gesetzt.
<code>.in ±N</code>	$N = 0$	bleibt	B,E,m	Setzen der Einrueckung auf $\pm N$. Die Einrueckung gilt fuer jede Ausgabezeile.
<code>.ti ±N</code>		ignoriert	B,E,m	Temporaeres Einruecken. Die naechste Ausgabezeile wird um $\pm N$ eingerueckt. $\pm N$ bezieht sich auf die laufende Einrueckung.

8. Makros, Strings, Diversionen und Positionstraps

8.1. Makros und Strings

Ein Makro ist ein mit Namen versehener Satz beliebiger 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 eingefuegt werden kann. Kommandos, Makros und Strings benutzen innerhalb von `proff` die gleiche Namensliste. Makro- und Stringnamen koennen 1 oder 2 Zeichen lang sein. Mit `.rn` koennen sie umbenannt und mit `.rm` geloescht 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, zuzueglich koennen bis zu 9 aktuelle Parameter mitgegeben werden. Die Strings `x` bzw. `xx` werden an der gewuenschten Stelle durch `*x` bzw. `*(xx` eingefuegt. String-Bezugnahmen und Makrorufe koennen geschachtelt sein.

8.2. Eingabeinterpretation im Kopiermodus (Copy-Mode)

Waehrend der Definition und Erweiterung von Strings und Makros (aber nicht bei Diversionen) wird die Eingabe im Kopiermodus gelesen. Die Eingabe wird kopiert, ohne dass eine Interpretation erfolgt, ausser,

- die Einfuegung der Inhalte der Nummernregister, die durch `\n` angewiesen wird, wird ausgefuehrt,
- die Einfuegung von Strings, ueber `*` angewiesen, erfolgt,
- die Substitution von Argumenten, durch `*` angewiesen, erfolgt,
- versteckte Newlines, die dadurch gekennzeichnet sind, dass ihnen ein `"\"` vorgestellt ist, werden ueberlesen,
- Kommentare, durch `"` gekennzeichnet, werden eliminiert,
- `\t` und `\a` werden als Horizontaltabulator bzw. SOH (`^X`) interpretiert,
- `\"` wird `"`,
- `\"` wird `\"`.

Diese Interpretationen koennen durch Vorsatz eines weiteren `"\"` unterdrueckt werden.

8.3. Argumente

Wird ein Makro aufgerufen, 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 es in Anfuhrungsstriche eingebettet werden. Der Anfuhrungsstrich kann durch Vorsetzen eines weiteren Anfuhrungsstriches selbst zum Argument werden.

Reicht die Zeile nicht zur Aufnahme der Argumentliste aus, kann sie mit Hilfe eines "versteckten Newlines" auf der naechsten Zeile fortgesetzt werden.

Wird ein Makro aufgerufen, so wird das bisherige Niveau des Inputs verlassen. Eventuelle Argumente eines vorhergehenden Aufrufes stehen so lange nicht zur Verfuegung, bis das vorhergehende Niveau wieder hergestellt ist. Die Substitution eines Argumentes kann im Makrokoerper mit `*N` angewiesen werden, wobei $1 \leq N \leq 9$ gilt. Existiert ein aufgerufenes Argument nicht, 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

"*" wird in der Definition durch den Vorsatz eines weiteren "\" vor der Auswertung geschuetzt (siehe Abschnitt 8.2.). Die Nummer des laufenden Argumentes steht im .x-Register. Ein Stringaufruf fuehrt ebenfalls zu einem Kellern des Input-Niveaus, so dass innerhalb des Strings keine Argumente eines eventuellen vorherigen Makroaufrufs verfuegbar sind. Einem durch Trap aufgerufenen Makro koennen keine Argumente uebergeben werden. Die Argumente werden im Kopiermodus auf einen Stack uebernommen. Dieser Mechanismus erlaubt keine direkte Interpolation einer laengeren Zeichenkette.

Es wird deshalb empfohlen, diese Interpolation durch Vorsatz eines weiteren "\" auf die Ebene der Substitution der Argumente innerhalb des Makrokoerpers zu verzoegern.

8.4. Diversionen

Die erstellte Ausgabe kann einem Makro uebergeben werden (Diversion), z.B. zum Zwecke der Fussnotenverarbeitung oder zum Bestimmen der horizontalen und vertikalen Groesse eines Textanschnittes. Ein Diversionstrap kann auf eine spezifizierte vertikale Position gesetzt werden. Die Nummernregister dn und dl beinhalten die vertikale bzw. die horizontale Groesse (Maximum) der letzten Diversion.

Diversionen koennen 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 Fuellungsstand der Seite (.h-Register)
- Name der laufenden Diversion (.x-Register)

Die maximale Schachteltiefe ist 5.

8.5. Traps

Traps sind Behandlungen, die bei Eintreten bestimmter Ereignisse ausgeführt werden und eine Unterbrechung der laufenden Behandlung bewirken.

Es sind drei Arten von Traps verfügbar:

- Seitentrap
- Diversionstraps
- Eingabezeilenzaehltrap

Traps zum Aufruf von Makros koennen 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 trapausloesende Seitenposition erreicht ist. Die Distanz zu dem jeweilig naechsten Trap ist im `.t`-Register verfuegbar. Gibt es keinen weiteren Trap zwischen der laufenden Seitenposition und dem unteren Rand gibt, ist es die Distanz zum unteren Rand.

Tabelle 7: Kommandoubersicht

Kommando	Anfangswert	ohne Argument	KZ	Erlaeuterung
.de xx yy		.yy = ..		Definition oder Neudefinition des Makros xx. Der Makro beginnt mit der naechsten Eingabezeile. Diese werden im Kopiermodus uebernommen. 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 Anfuhrungsstrich wird ueberlesen, um vorgeseetzte Leerzeichen zu ermoeeglichen.
.as xx string		ignoriert		Erweitern der Zeichenkette xx.
.rm xx		ignoriert		Loeschen des Kommandos, Makros oder Strings xx. Der Name wird aus der Namensliste geloesch und belegter Speicherplatz freigegeben. Nachfolgende Bezugnahmen werden ignoriert.
.rm xx yy		ignoriert		Loeschen des Kommandos, Makros oder Strings xx bis yy. Existiert yy, wird es ebenfalls geloesch.

Tabelle 7: (Fortsetzung)

Kommando	Anfangs- wert	ohne Argument	KS	Erlaeuterung
.di xx		Ende	D	Abspeichern der Ausgabe in den Makro xx. Es erfolgt die normale Textverarbeitung ausser Seitenverschiebung (.po). Die Diversion endet, wenn .di entschluesst wird.
.da xx		Ende	D	Erweitern 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 fuer N wird ueberschrieben. N = 0 bezieht sich auf den Seitenbeginn. Fehlt xx, so wird der zuletzt installierte Trap bezogen auf N geloesch, so dass ein eventuell ueberschriebener Trap wieder wirksam wird.
.ch xx N				Wechseln der Trapposition des Makros xx auf N. Fehlt N, wird der Trap geloesch.
.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. Sind keine Argumente gegeben sind, wird der Trap geloesch.

Tabelle 7: (Fortsetzung)

Kommando	Anfangs- wert	ohne Argument	KZ	Erlaeuterung
.it N xx		aus	E	Setzen eines Eingabezeilenzaehltraps nach der N-ten Eingabezeile zum Aufruf des Makros xx (Steuerzeilen zaehlen nicht).
.em xx				Der akro xx wird am Ende aller Eingabezeilen aufgerufen.

9. Zahlenregister

Eine Vielzahl von Parametern fuer die Textformatierung sind fuer den Anwender in vordefinierten Registern, sogenannten Zahlenregistern verfuegbar. Zusaetzlich kann der Nutzer sich eigene Register definieren. Die Namen sind ein oder 2 Zeichen lang und werden in einer eigenen Liste gefuehrt, sodass sie sich nicht von Kommandos, Makros oder Zeichenketten unterscheiden muessen. Ausser einigen vordefinierten Nur-Lese-Registern koennen Zahlenregister gelesen, geschrieben, krementiert und dekrementiert werden. Sie koennen benutzt werden, wenn ein numerierter Wert in die Eingabe eingefuegt werden soll und koennen Teil eines numerischen Ausdrucks sein (siehe Punkt 1.4.).

Zahlenregister koennen mit .nr definiert und modifiziert werden, wobei der Name, der zahlenmaessige Wert und der Wert der Erhoehung bei automatischem Inkrementieren bzw. Dekrementieren spezifiziert werden. Die Register koennen auch beim Einfuegen modifiziert werden. Folgende prinzipielle Schreibweisen sind moeglich, wobei x und xx die Namen, N der Wert und M der vereinbarte Inkrementwert fuer die Register sind.

Tabelle 8: Einfuegen der Zahlenregister

Sequenz	Auswirkung auf die Register	eingefuegter 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 Einfuegen werden die Registerinhalte in folgende ebenfalls vorher zu vereinbarende Formate konvertiert:

Format Zahlenfolge

1	0, 1, 2, 3, 4,...	(Standard)
001	000, 001, 002,...	(dezimal mit Vornullen)
i	0, i, ii, iii, iv,...	(roemisch klein)
I	0, I, II, III, IV,...	(roemisch gross)
a	0, a, b, c, ..., z, aa,...	(Kleinbuchstaben)
A	0, A, B, ..., Z, AA, ..., ZZ, AAA,...	(Grossbuchstaben)

Tabelle 9: Kommandoubersicht

Kommando	Anfangswert	ohne Argument	KZ	Erklaerung
.nr R ±NM				Dem Register R wird der vorzeichenbehaftete Wert N zugewiesen. M ist das Inkrement bei automatischen Inkrementieren.
.af R C				Festlegen des Formates C fuer das Register R.
.rr R		ignoriert		Austragen des Registers R. Der intern benoetigte Platz wird frei.

10. Tabulatoren, Leitpunkte (Leaders) und Felder

10.1. Tabulatoren und Leitpunkte

Die ISO-Zeichen TAB und SOH (hier verwendet als Kennzeichen fuer Leitpunkte) koennen sowohl zum Erzeugen von Horizontalbewegungen als auch zur Erzeugung einer Kette gleicher Zeichen bis zu einer bestimmten Zeilenposition verwendet werden. Die Laenge der erzeugten Bewegung bzw. Zeichenkette ist abhaengig von mit .ta intern erzeugten Tabulatorstops. Standardmaessig erzeugen Tabulatoren Bewegungen und Leitpunkte als Ketten von Punkten (..).

Die Kommandos .te und .le bieten die Moeglichkeit zwischen Bewegung und Zeichenketten zu waehlen. Es gibt 3 Arten interner Tabulatorstops: linksausgerichtete, rechtsausgerichtete und Zentrieren.

In der folgenden Uebersicht ist D die Distanz von der laufenden Position in der Eingabezeile (bei der der Tabulator bzw. das Leitpunktkennzeichen gefunden wurde) bis zum naechsten Tabulatorstop.

"Folgende Zeichenkette" besteht aus den Zeichen, die dem Tabulator (oder Leitpunkt) folgen bis zum naechsten Tabulator (oder Leitpunkt) oder dem Zeilenende. W ist die Weite (Laenge) der "folgenden Zeichenkette".

Tabulatorart	Laenge der Bewegung bzw. der wiederholten Zeichen	Position der "folgenden Zeichenkette"
links	D	folgend auf D
rechts	D - W	rechtsausgerichtet innerhalb von D
Zentrieren	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 in "Feldbegrenzungszeichen" gehalten wird. Die Feldlaenge wird durch die Distanz der Position auf der Eingabezeile, bei der die Zeichenkette beginnt, und dem naechsten 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 Leerzeichen, begrenzt. Die Differenz zwischen Feldlaenge und Zeichenkette kann negativ sein.

Tabelle 10: Kommandoubersicht

Kommando	Anfangswert	ohne 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 Zeichenubersetzungen

11.1. Eingabe-Zeichenubersetzungen

Die ISO - Steuerzeichen TAB (siehe Punkt 10.1.), SOH (siehe Punkt 10.1.) und Backspace (siehe Punkt 11.2.) werden dort diskutiert.

Das Newline dient zum Begrenzen der Eingabezeilen.

Die Steuerzeichen STX, ETX, ENQ, ACK und BEL werden akzeptiert und koennen 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 ISO - Steuerzeichen ESC zu verwechseln. Das Escape-Zeichen "\ " selbst kann ueber "\\ " eingegeben werden. Es kann mit dem Kommando .ec in ein beliebiges anderes Zeichen gewandelt werden. Falls erforderlich, kann der Escape-mechanismus ueber .eo ausgeschaltet werden und ueber .ec wieder ein.

Tabelle 11: Kommandoubersicht

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 Ueberschreiben wird im Punkt 13. beschrieben.

Das Unterstreichen in Form geschlossener Linien wird im Punkt 12.4. beschrieben.

Das Unterstreichen unterstreicht automatisch die Zeichen des Unterstrich-Zeichenvorrates, der in der Gerädetabelle gekennzeichnet ist (normalerweise Gross- und Kleinbuchstaben und Ziffern).

Tabelle 12: Kommandoubersicht

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

11.3. Steuerzeichen

Je nach Bedarf koennen die Steuerzeichen "." und "," in jedes andere Zeichen gewechselt werden, z.B. um gewisse Textstuecke vor der Interpretation zu schuetzen. Zu beachten ist, dass dieser Wechsel mit der Gestaltung der innerhalb dieses Wechsels aufgerufenen Makros uebereinstimmen muss.

Das Nicht-Break-Kommandokennzeichen "" wird verwendet, um in bestimmten Situationen Kommandos zu unterdruecken, die die Break-Funktion einschliessen. Dies ist z.B. bei Kopfzeilen- bzw. Seitenendemarkros erforderlich, die im Fuellmodus mit Traps an Stellen aufgerufen werden koennen, bei denen sich noch Teile der Eingabezeile in Verarbeitung befinden.

Tabelle 13: Kommandoubersicht

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

11.4. Zeichenumschluesselung

Mit des Kommando .tr ist es moeglich bestimmten Zeichen andere Zeichen zuzuordnen. Diese Umschluesseln erfolgt zur Ausgabezeit.

Tabelle 14: Kommandoubersicht

Kommando	Anfangs- wert	ohne Argument	KZ	Erklaerung
.tr abcd	-	-		Umschluesseln der Zeichen a zu b und c zu d. Bei ungerader Anzahl wird das letzte Zeichen in ein Leerzeichen uebersetzt.

11.5. Textdurchlauf ohne Uebersetzung

Eine Eingabezeile, die mit "\!" beginnt, wird im Kopiermodus gelesen und ohne weitere Verarbeitung wieder ausgegeben (ohne "\!").

Sie wird in keiner Weise in die Textverarbeitung (Fuellmodus, Zeilenzaehlung u.a.) einbezogen.

Diese Arbeitsweise kann benutzt werden, um Steuerinformationen an einen Postprozessor weiterzugeben oder waehrend einer Diversion Steuerzeichen in einen Makro abzuspeichern.

11.6. Kommentare und verdeckte Newlines

Eine unhandliche lange Eingabezeile, die aber zusammengehoerig ist (z.B. eine Zeichenkettendefinition oder eine Parameterliste fuer 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 dazu vertikale bzw. horizontale Bewegungen zu erzeugen. N kann negativ sein, eine positive Angabe bedeutet abwaerts bzw. nach rechts. Um unvorherzusehende Verschiebungen zu vermeiden, ist es erforderlich, dass die vertikalen Bewegungen sich im Fuellmodus innerhalb eines Wortes ausgleichen. Folgende zusaetzliche Steuerfolgen fuer vertikale Bewegungen sind zugelassen:

\u	eine halbe Zeile aufwaerts
\d	eine halbe Zeile abwaerts
\r	eine ganze Zeile aufwaerts

12.2. Funktion zum Bestimmen 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
```

das Einruecken temporaer um die Weite der Zeichenkette "1. " zurueck. Zu beachten ist, dass die Masseinheit u unbedingt anzugeben ist, da sonst das Kommando die Weite in Em's (siehe Punkt 2.3.) interpretieren wuerde. Es werden 2 Register gesetzt, in sb der Anfang und in st das Ende der Zeichenkette relativ zu der Zeile.

12.3. Merken einer horizontalen Position

Die Steuerfolge `\kx` speichert die laufende horizontale Position in das Register `x`.
Zum Beispiel erzeugt die Konstruktion `\kx Wort\h'| \nxu+2n' Wort` den Doppeldruck des Wortes `\kx Wort\h'| \nxu+2n' Wort`.

12.4. Linienziehen

Die Steuerfolge `\l'Nc'` produziert eine Kette von `N` Zeichen `c` nach rechts. Ist `c` nicht angegeben, wird der Unterstrich `"_"` verwendet. Ist `N` negativ, 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 Vertikalvorschüben mit dem generierten Zeilenabstand. Ist kein `c` angegeben, wird der Senkrechtstrich `(\langle br)` verwendet. Ein positives `N` erzeugt eine Linie abwärts und ein negatives `N` eine Linie aufwärts.

13. Zeichenkonstruktion

13.1. Zeichenuberschreibung

Mit Hilfe der Steuerfolge `\o'string'` werden die Zeichen der Zeichenkette `"string"` uebereinandergedruckt. Der Uebereinanderdruck erfolgt in der Weise, dass die einzelnen Zeichen zentriert werden. Die Weite wird vom laengsten Zeichen bestimmt. Die Steuerfolge `\o'o+'` produziert `|o'o+'`.

13.2. Zeichendruck ohne Vorschub

Die Steuerfolge `\zc` druckt das Zeichen `C` ohne Zeichenvorschub. Damit ist es aehnlich der Zeichenuberschreibung (siehe Abschnitt 13.1.) moeglich, Zeichen zu ueberdrucken mit dem Unterschied, dass der Ueberdruck 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 uebereinander gebaut, dabei wird das erste Zeichen an die Spitze und die folgenden jeweils darunter angeordnet. Mit Hilfe der Funktion fuer extraweiten Zeilenabstand (siehe Abschnitt 6.2.) ist fuer den entsprechenden Freiraum zu sorgen.

14. Silbentrennung

Das automatische Silbentrennen kann aus- bzw. eingeschaltet werden. Bei eingeschalteten Silbentrennen koennen verschiedene Varianten benutzt werden. Ein Kennzeichen (\%) kann in ein Wort eingebaut werden, um eine gewuenschte Trennstelle zu kennzeichnen, oder aber dem Wort vorgestellt werden, um ein Trennen zu verhindern. Zusaetzlich kann eine (kleine) Ausnahmewortliste definiert werden.

Es werden nur zusammenhaengende Ketten von Alpha-Zeichen fuer die Silbentrennung beruecksichtigt. Worte, die das Trennzeichen (Minus), Em-Dash (\(em)) oder das Silbentrenn-Kennzeichen enthalten, werden falls erforderlich, immer an dieser Stelle getrennt, unabhængig, ob der Silbentrennmodus ein- oder ausgeschaltet ist.

Tabelle 15: Kommandoubersicht

Kommando	Anfangs- wert	ohne Argument	KZ	Erklaerung
.nh	ein	-	E	Ausschalten der automa- tischen Silbentrennung.
.hy N	ein, N=1	ein, N=1	E	Einschalten der automa- tischen Silbentrennung fuer N > 1 und Ausschalt- ten fuer 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 bedingt alle 3 Restrik- tionen.
.hc C	\%	\%	E	Das Kennzeichen fuer Sil- bentrennung wird auf C gesetzt. Es erscheint nicht in der Ausgabe.
.hw Wort1...		ignoriert		Spezifikation von Wor- ten mit eingebauten Bin- destrichen, die die Trennstelle in den Wor- ten markieren.

15. Titelzeilen

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

Tabelle 16: Kommandoubersicht

Kommando	Anfangswert	ohne Argument	KZ	Erklaerung
<code>.tl 'links' Mitte'rechts'</code>				Die Zeichenketten "links", "Mitte" und "rechts" werden entsprechend linksausgerichtet, zentriert oder rechtsausgerichtet. Ist in einer Zeichenkette das Seitennummernkennzeichen enthalten (normalerweise %), wird der Inhalt des Registers % eingefuegt. Als Begrenzer fuer die Zeichenketten kann ein beliebiges Zeichen verwendet werden.
<code>.pc c</code>	%	-	-	Das Seitennummernkennzeichen wird auf C gesetzt oder geloescht. Die Seitennummer selbst verbleibt im Register %.
<code>.lt +N</code>	65		E,m	Die Laenge des Titels wird auf +N gesetzt. Die Zeilenlaenge und die Titellaenge sind voneinander unabhengig.

16. Ausgabezeilennummerierung

Mit dem Kommando `.nm` kann ein automatisches Ausgabenzeilennummerieren angewiesen werden. Ist dies der Fall, wird eine 3-stellige Nummer plus eines Leerzeichens den Output-Zeilen vorangestellt. Jede Zeile erhaelt eine Verschiebung von 4 Zeichen, die aktuelle Zeilenlaenge wird beibehalten. Soll der rechte Rand mit fruheren Zeilen uebereinstimmen, muss eine Reduktion der aktuellen Zeilenlaenge vorgenommen werden. Leerzeilen und anderer Leerraum sowie mit `.tl` erzeugte Zeilen werden nicht numeriert.

Die Numerierung kann zeitweise mit `.nu` ausgesetzt werden. Zusaetzlich kann ein Zeilennummern-Einruecken `I`, sowie ein Freiraum `S` zwischen Nummer und Text vereinbart werden. Desweiteren kann angewiesen werden, dass nur die Zeilen numeriert werden, deren Nummern ein Vielfaches von `M` sind. Bei den anderen erscheint eine entsprechende Einrueckung.

Tabelle 17: Kommandoubersicht

Kommando	Anfangswert	ohne Argument	KZ	Erklaerung
<code>.nm ±N M S I</code>		aus	E	Ist <code>±N</code> angegeben wird die naechste Ausgabezeile mit <code>±N</code> numeriert. 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 Zeilennummerierung ausgeschaltet. Die naechste Zeilennummer wird fuer evtl. spaetere Benutzung im Register <code>.ln</code> gerettet.
<code>.nu N</code>		<code>N = 1</code>	E	Die naechsten <code>N</code> Zeilen werden nicht numeriert.

17. Bedingtes Akzeptieren der Eingabe

Die Eingabe von "Text" (Eingabetext, Kommandos, Makros u.ae.) kann von bestimmten Bedingungen abhaengig gemacht werden. Folgende Moeglichkeiten 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 = e</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 > 0</code> , wird Text akzeptiert.
<code>.if !N Text</code>	Ist der Ausdruck <code>N <= 0</code> , wird Text akzeptiert.
<code>.if 'string1'string2'Text</code>	Ist die Zeichenkette <code>string1</code> identisch zu <code>string2</code> , wird Text akzeptiert.
<code>.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 <code>if-else</code> . Alle Formen von <code>if</code> sind moeglich.
<code>.el ...</code>	Else-Teil von <code>if-else</code> . Alle Formen von <code>if</code> sind moeglich.

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

18. Umschalten der Formatierungsumgebung (Environment)

Eine Anzahl von Parametern, in der Kommandozusammenstellung mit `E` gekennzeichnet, sind in einer Umgebung zusammengefasst, die der Nutzer umschalten kann. Damit ist es moeglich innerhalb des Textaufbaus auf ein anderes Grundformat umzuschalten, z.B. innerhalb einer Diversion. Maximal 3 verschiedene Umgebungen sind gleichzeitig nutzbar.

Tabelle 18: Kommandouebersicht

Kommando	Anfangswert	ohne Argument	KZ	Erklaerung
<code>.ev N</code>	<code>N = 0</code>			Einschalten des Formatierungsmodus <code>N = 0, 1, 2</code> .

19. Einfuegen von Eingaben der Standardeingabe

Die Eingabe kann zeitweise auf die Standardeingabe umgelegt werden. Bei MUTOS ist das normalerweise die Tastatur, es kann aber auch ein Filter oder ein File sein. Beabsichtigt wird damit, spezielle Formate, Adressen, Titel u.a. unterzubringen. Mit zwei Newlines wird die Eingabe wieder zurueckgeschaltet.

Tabelle 19: Kommandouebersicht

Kommando	Anfangs- wert	ohne Argument	KZ	Erklaerung
.rd prompt	-			Umschalten der Eingabe auf die Standardeingabe bis zwei Newlines in Folge erscheinen. .rd wird wie ein Makro behandelt; nach prompt koennen Argumente folgen.
.ex				Beenden von nroff. .ex wird genauso behandelt, als waere EOF im letzten Eingabefile gefunden worden.

20. Input/Output-File-Umschaltung

Tabelle 20: Kommandouebersicht

Kommando	Anfangs- wert	ohne Argument	KZ	Erklaerung
.so filename				Die Eingabe erfolgt vom angegebenen File. Wird EOF gefunden, ird im Anfangsfile fortgesetzt. so-Kommandos koennen geschachtelt werden.
.nx filename				Wie .so, nur dass keine Rueckkehr erfolgt.
.pi program				Die Ausgabe erfolgt nach "program". Es koennen keine Argumente uebergeben werden.

21. Weitere verschiedene Kommandos

Tabelle 21: Kommandoübersicht

Kommando	Anfangs- wert	ohne Argument	KZ	Erklärung
.mc c H		aus	E,m	Das Zeichen c wird in einen Abstand von N rechts von der rechten Seitenkante auf den naechsten nicht leeren (ausser mit .tl erzeugten) Ausgabezeilen ausgegeben. Ist kein N gegeben, wird der vorherige Wert angenommen. Initialisiert ist N mit 2 em.
.tm string		Nc		Die Zeichenkette "string" wird im Kopiermodus gelesen und auf das Terminal ausgegeben.
.ig yy				Ignorieren der Eingabezeile. Wird wie .de behandelt, nur dass die Eingabe ueberlesen wird. Die Eingabe wird im Kopiermodus gelesen und Autokrementregister werden, falls enthalten, beruecksichtigt.
.pm t				Druck der Makros. Name, Groesse und Inhalt der verwendeten Makros werden auf das Terminal ausgegeben.
.fl			B	Ausgabe des Ausgabe-Puffers. Hauptsächlich fuer interaktive Testzwecke benutzt.

22. Mitteilungen

Die Ausgabe der Kommandos `.tm`, `.pm` und die Anforderung von `.rd` erfolgen ebenso wie verschiedene Fehlermitteilungen auf dem Standard-Error-File. Letzteres ist nicht identisch mit dem Standard-Output. Standardmaessig erfolgen zwar beide auf dem Bildschirm, sie koennen aber unterschiedlich zugewiesen werden.

Verschiedene Fehler koennen waehrend der Arbeit von `nrff` auftreten. Bei weniger ernsten Fehlern wird die Textverarbeitung fortgesetzt. Wenn moeglich, wird in der Ausgabe 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 Ueberschreiten interner Grenzen, wird die Verarbeitung abgebrochen und eine entsprechende Mitteilung ausgegeben.

Anlage 1:

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	Zentrieren
ch	8.		Wechsel Trapposition
cu	11.2.	E	Unterstreichen
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 proff
ie	20.		if-else
if	20.		if
ig	24.		Ignorieren Eingabe
in	7.	B,E,m	Einruecken
it	8.	E	Eingabezeilentrap
lc	10.	E	Wechseln Leaderzeichen
ll	7.	E,m	Zeilenlaenge
ls	6.	E	mehrzeilige Ausgabe
lt	18.	E,m	laenge 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 Fuellmodus
nh	17.	E	kein Silbentrennen
nm	19.	E	Zeilennumerierung
nr	9.		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	Seitenlaenge
pm	34.		Ausgabe Makros
pn	4.		Setzen Seitennummer
po	4.	v	Seitenoffset
rd			Lesen Standardeingabe