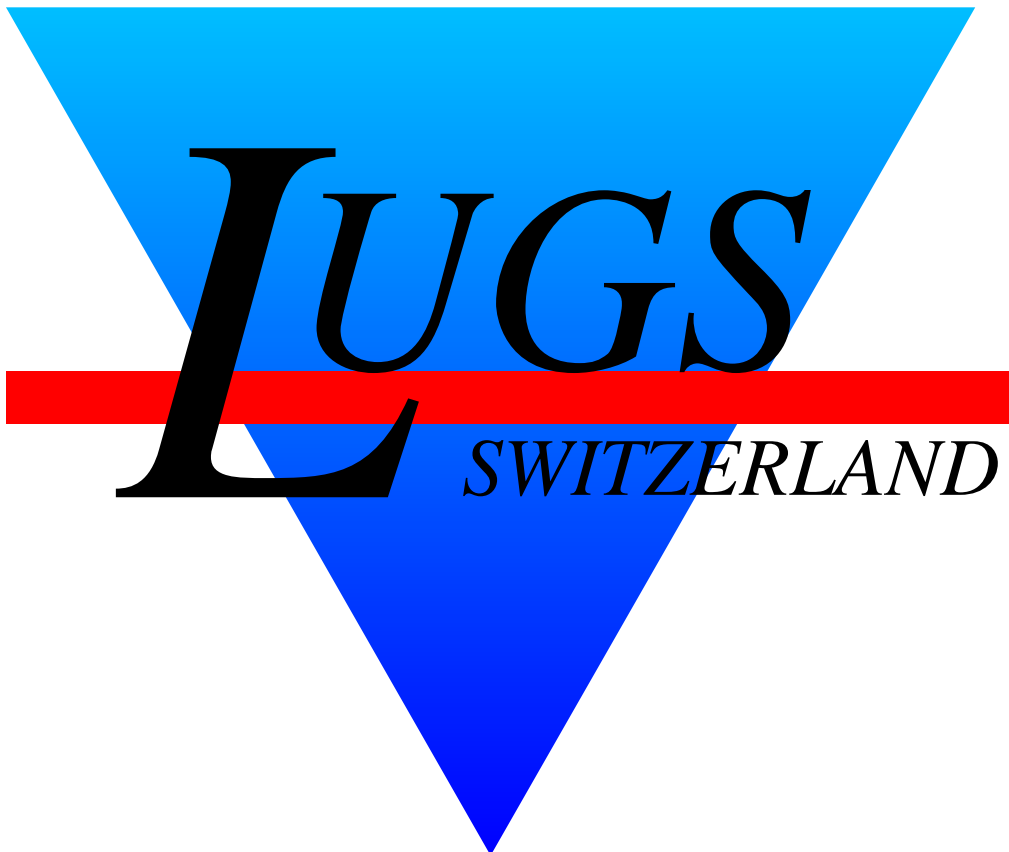


# ed und sed

David Frey



Copyright © 2004 David Frey

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

The author(s) would appreciate a notification of modifications, translations, and printed versions. Thank you.

## ed: Ueberblick

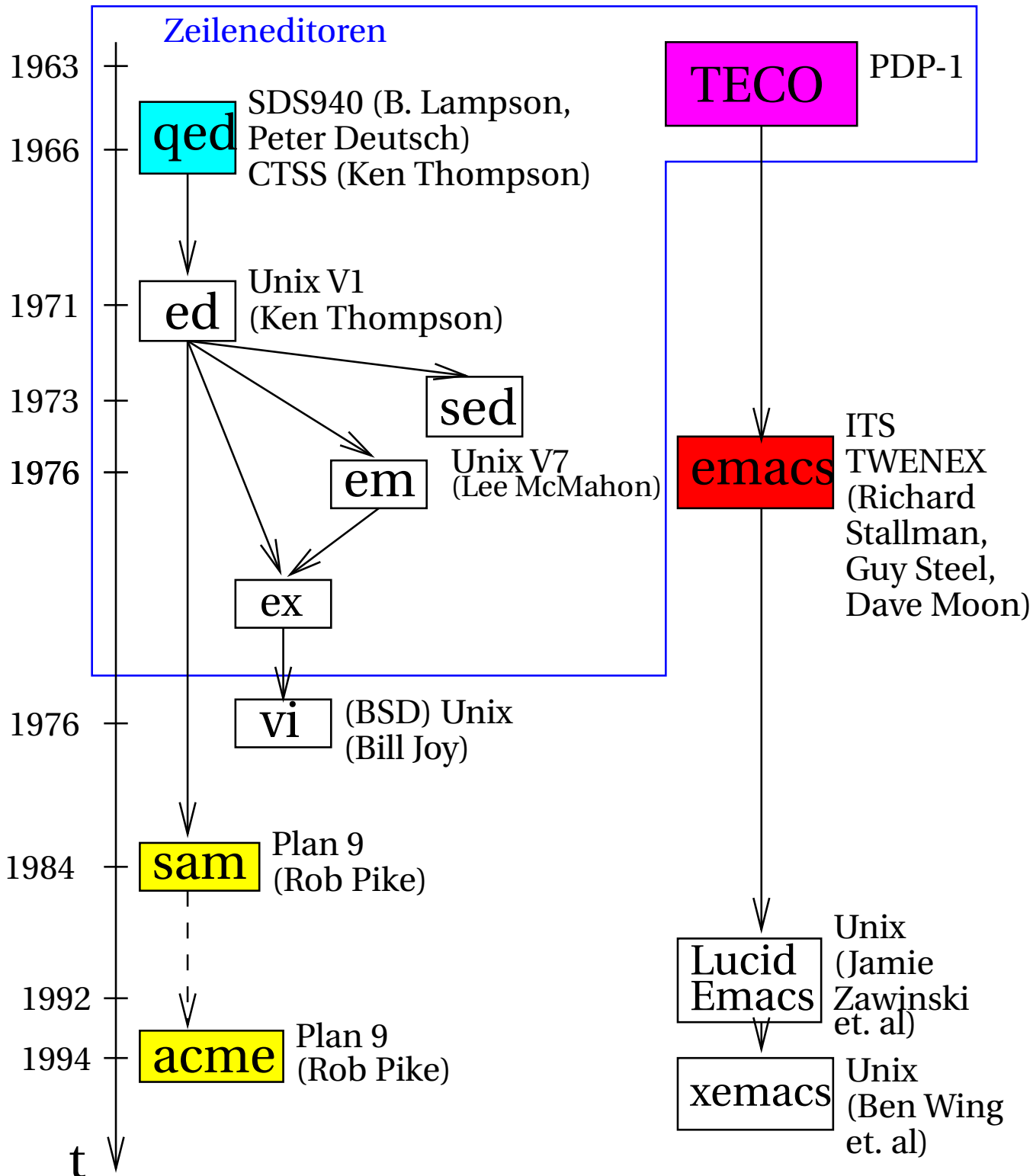
The cycle of editing and testing is very convenient in Visual Basic, though the facilities for viewing and editing text are so primitive that one yearns for a powerful editor like ed.

Brian W. Kernighan

ed ist der Unix-Standard Editor; es ist eine vereinfachte Version des qed-Editors [1]. qed konnte z.B. mehrere Buffer, erweiterte Regular Expressions, rechnen und war mit seinen eigenen Befehlen programmierbar.

Alle diese Editoren sind *Line-Editoren*, d. h. man editiert Linien resp. Linienbereiche.

Sie sind **modal** d. h. die Editoren unterscheiden zwischen *Input* und *Command-Mode*.



## ed: Syntax und Adressen

Syntax eines Kommandos:

$[adr_1[ , adr_2] ] cmd[params]$

eine Adresse:

- . momentane Zeile
- \$ letzte Zeile
- $n$   $n$ te Zeile
- $-n, ^n$   $n$ -Zeilen zurück
- $+n$   $n$ -Zeilen vorwärts
- $, , %$   $\equiv 1, \$$
- $;$   $\equiv . , \$$
- $/RE/$  regular expression vorwärts
- $?RE?$  regular expression rückwärts
- '  $m$  mit  $m$  markierte Zeile

Ein Range besteht aus zwei Adressen:

$a, b$  Zeilen  $[a, b]$

$a; b$  Zeilen, setze zuerst  $a$ , und berechne dann  $b$

## ed: Befehle (1) [2]

a	append
c	change (replace) line
d	delete
e <i>fn,e !cmd</i>	edit file, stdout of <i>cmd</i>
E <i>fn</i>	edit (unconditionally)
f	set default file name
g/ <i>RE/cmds</i>	apply <i>cmds</i> on lines with matching REs.
G/ <i>RE/</i>	interactively edit addressed lines with RE.
h,H	print help on last error, turn off explanations
i	insert text
j	join lines
k <i>m</i>	mark line with letter <i>m</i>
l	print lines
m	move lines

## ed: Befehle (2) [2]

n	number lines
p	print lines
P	toggle command prompt
q,Q	quit (unconditionally)
r <i>fn,r !cmd</i>	read <i>fn</i> , stdout of <i>cmd</i>
s/ <i>RE/rep/[g n]</i>	search & replace first/all (g)/ <i>n</i> th match(es)
t <i>addr</i>	copy (transfer) after <i>addr</i> .
u	undo last command
v/ <i>RE/</i> ,/ <i>VRE/</i>	apply commands <i>not</i> matched by <i>RE</i>
w[ <i>q</i> ] <i>fn,w !cmd</i>	write <i>fn</i> , stdout of <i>cmd</i>
W <i>fn</i>	append to <i>fn</i>
x	put cut buffer after addressed line
y	yank lines to the cut buffer
zn	scroll <i>n</i> lines.

## ed: Befehle (3) [2]

<i>! cmd</i>	execute <i>cmd</i>
<i>#</i>	comment
<i>=</i>	print line number
<i>newline</i>	print addressed line and set address

## sed

sed steht für *Stream Editor*, die gestreamte Variante des Standard Editors, ed.

Somit fallen alle Befehle, die eine Position im Stream benötigen, weg.

Alle interaktiven Befehle fallen auch weg (hHPtxyuz).



## sed: Befehle (1) [3]

! <sup>a</sup>	a	append
-	c	change (replace) line
	d	delete
-	e,E	edit file
-	f	set default file name
!	g/G	copy/append hold space to pattern space
!	h/H	copy/append pattern space to hold space
! <sup>a</sup>	i	insert text
-	j	join lines
-	km	mark line with letter <i>m</i>
	l	print lines unambiguously
	m	move lines

---

<sup>a</sup>'\' als Fortsetzungszeichen anstelle von '.' am Ende.

## sed: Befehle (2) [3]

!	<i>n,N</i>	read/append next line of input into pattern space
	<i>p</i>	print lines
!	<i>P</i>	print up to first embedded newline
	<i>q,Q</i>	immediately quit ( <i>q</i> : print line if auto-print)
!	<i>rfn,Rfn</i>	append text/a line read from <i>fn</i>
	<i>s/RE/rep/[g n]</i>	search & replace first/all ( <i>g</i> )/ <i>n</i> th match(es)
!	<i>tlabel</i>	branch to label on successful substitution.
!	<i>Tlabel</i>	branch to label on unsuccessful substitution.
-	<i>u</i>	undo last command
-	<i>v/RE/,/VRE/</i>	apply commands <i>not</i> matched by <i>RE</i>

## sed: Befehle (3) [3]

!	<i>wfn</i>	write current pattern space to <i>fn</i>
!	<i>Wfn</i>	write the first line of the current pattern space to <i>fn</i>
-	<i>x</i>	put cut buffer after addressed line
!	<i>y/src/dst/</i>	transliterate characters in pattern space
-	<i>zn</i>	scroll <i>n</i> lines.
+	<i>:</i>	label
-	<i>!cmd</i>	execute <i>cmd</i>
	<i>#</i>	comment
	<i>=</i>	print line number
!	<i>;</i>	command separator
!	<i>newline</i>	white space
+	<i>{,}</i>	opening, closing block

## Aufrufsyntax

Standard sed-Aufruf:

```
sed [-n] -escript|-fscript-file
```

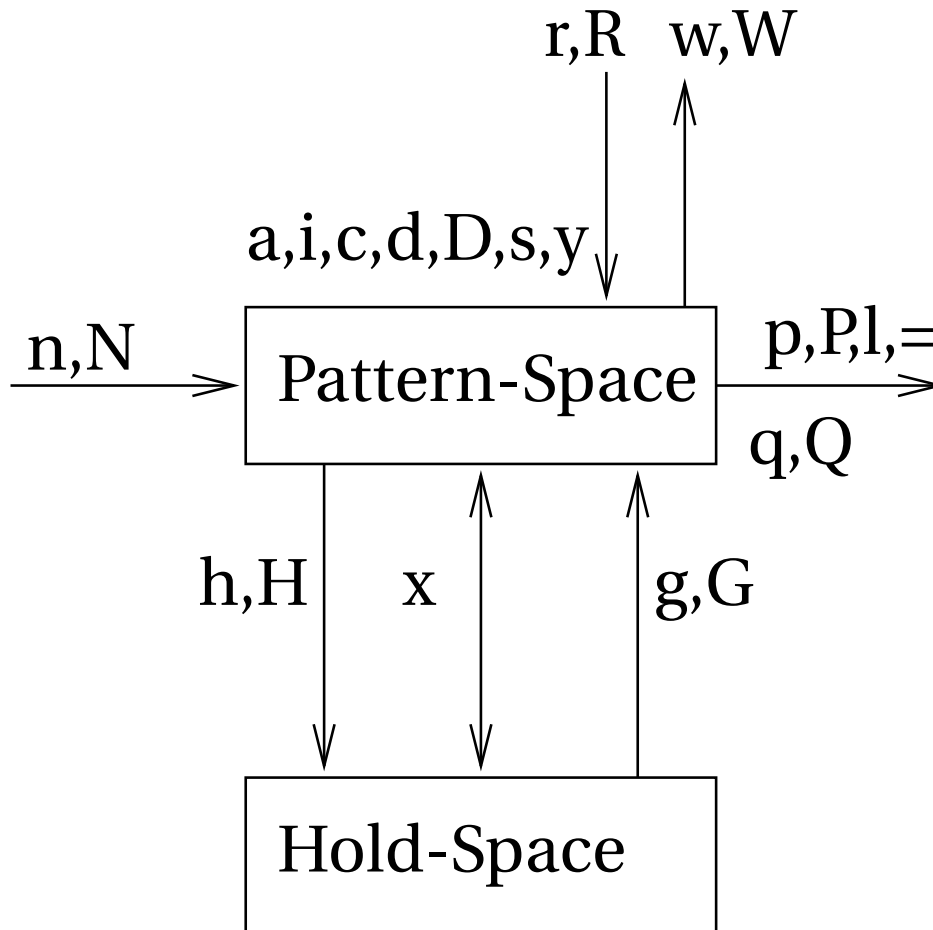
GNU-sed kann wie üblich ein paar Sachen mehr:

```
sed [-nrsu][-i[suffix]][-lN]  
-escript|-fscript-file
```

Aus der Manpage [3]:

- n Pattern-Space nicht automatisch ausgeben
- e Script folgt auf der Command-Line
- f Script folgt im File
- i File in-place (evtl mit Backup) editieren
- l *N* Line-Wrap-Länge auf *N* setzen
- r Perl Regular Expressions (ERE) verwenden
- s separate Files
- u unbuffered

## Hold/Match-Space



$h, g, n$  copy (h,g), read (n)

$H, G, N$  append

$\#$  comment

$:, b, t, T$  labels, branch

$\{, \}$  block

## I/O-Paradigma

Zeilen werden prinzipiell sequentiell gelesen und bearbeitet.

## Flow-Control

### label

: define label

### go-to

b branch unconditionally

### if

t branch on successful substitution

T branch on unsuccessful substitution

N.B: sed ist *turing-complete*.

## Beispiele (1) [z.T. aus [4]]

### cat

```
sed -n p /etc/motd
```

### cat -n resp. nl -ba

Einfache Lösung:<sup>a</sup>

```
sed -n '=;p' /etc/motd|
```

```
sed -e 'N; s/\([[[:digit:]]\+\)\n/\1 /'
```

### cat -vET

```
sed -n l /etc/motd
```

### cat -s

```
:x
```

```
/^\n*$/ {
```

```
  N
```

```
  bx
```

```
}
```

```
s/^\n\)* /\1/
```

---

<sup>a</sup>mit nur einem Prozess geht es auch [4]

## Beispiele (2) [z.T. aus [4]]

### **tac**

```
sed -ne '2,$G;$p;h' /etc/motd
```

```
original: sed -ne '1!G;$p;h' /etc/motd
```

### **head**

```
sed -n 1,10p /etc/motd
```

### **tail**

```
sed -n '$-10,$p' /etc/motd
```

Es geht, ist aber kein Einzeiler (→ [4]):

### **wc -l**

```
sed -n '$=' /etc/motd
```

### **wc -C**

Es geht, ist aber kein Einzeiler (→ [4]):



## Beispiele (3) [z.T. aus [4]]

### rot13

leicht exotisches Beispiel:<sup>a</sup>

```
sed -e 'y/\
ABCDEF GHIJKLMNOPQRSTUVWXYZ\
abcdefghijklmnopqrstuvwxyz/
NOPQRSTUVWXYZABCDEFGHIJKLM\
nopqrstuvwxyzabcdefghijklm/' /etc/motd
```

---

<sup>a</sup>rot13, caesar 13, tr 'A-MN-Za-mn-z' 'N-ZA-Mn-za-m'



## komplexes Beispiel (Script)

```

1  #!/bin/sed -f
2
3  s,<\([A-Za-z.:/\]\+\)>,<a href="\1">\1</a>,
4  t skip
5
6  s,/\/\([^*]*\)/,<em>\1</em>,g;
7  s,\*\([^*]*\)\*,<b>\1</b>,g;
8  s,_\([^_]*\)\_,<u>\1</u>,g
9
10 :skip
11
12 1i \
13 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">\
14 <html>
15 1{
16 s,^,<head><title>,; s,$,</title></head>,;
17 a \
18 <body>
19 }
20 $a \
21 </body>\
22 </html>
23
24 /^[          ]*[o.-]/{
25     x
26     /^{$/ {
27         s/^/UL/
28         i \
29 <ul>
30     }
31     x

```

```

32     s/^[           ]*[o.-]/<li>/
33 }
34
35 /^$/ {
36     x
37     /^UL$/ {
38         s/^UL//
39         i \
40     </ul>
41     }
42     x
43 }
44
45
46 N
47 /^.*\n *====* *$/ {
48     s/\n *====* *//;
49     s, ^, <h1>, ; s, $, </h1>,
50 }
51
52
53 /^.*\n *----* *$/ {
54     s/\n *----* *//;
55     s, ^, <h2>, ; s, $, </h2>,
56 }
57
58 P
59 D

```

## komplexes Beispiel (Output)

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
2 <html>
3 <body>
4 <head><title>Title</title></head>
5
6 <h1>Header 1</h1>
7
8 Text <b>bold</b> <u>underlined</u> <em>italic</em>
9 URL <a href="http://www.lugs.ch">http://www.lugs.ch</a>
10
11 <h2>Header 2</h2>
12
13 Text <b>bold</b> <u>underlined</u> <em>italic</em>
14
15 <ul>
16 <li> bullet 1
17 <li> bullet 2
18 <li> bullet bullet 1
19 <li> bullet bullet 2
20 <li> bullet 3
21 </ul>
22
23 </body>
24 </html>
```

## Vorteile

sed ist für das Editieren von Streams ausgelegt; daher ist der Aufwand um etwas zu Editieren gering.

## Nachteile

- ☞ Keine Variablen ( $\Rightarrow$  sh,awk,m4,perl)
- ☞ Kann nicht rechnen ( $\Rightarrow$  sh,awk,m4,perl)
- ☞ (Standard sed) kann nicht *in-place* editieren ( $\Rightarrow$  ed[,perl])

## Zusammenfassung

sed-Programme sind fast ausschliesslich Einzeiler und werden in Pipelines in Shell-Scripts verwendet.

Ab  $\geq 5 \dots 10$ -Zeilen wird es (langsam) kompliziert.

## Kuriosa oder sed-Stresstests

Folgende sed-Scripts wurde von Leuten geschrieben, die Langeweile hatten und/oder mal ein bisschen sed-Programmierung üben wollten [5]:

- dc in sed:  
`/usr/share/doc/sed/examples/dc.sed`
- Infix-Mathematik in sed:  
`/var/local/ftp/pub/copied/  
source-cd-1993/bsd_srcs/usr.  
bin/sed/math.sed`
- Towers of Hanoi in sed:  
`/var/local/ftp/pub/copied/  
source-cd-1993/bsd_srcs/usr.bin/  
sed/hanoi.sed`
- 99-Bottles of Beer in sed:  
`http://www.99-bottles-of-beer.net/  
s.html#sed`
- Sokoban in sed:  
`http://http://aurelio.net/sed/`



## Literatur

- [1] Dennis M. Ritchie. An incomplete history of the QED Text Editor.  
<http://cm.bell-labs.com/cm/cs/who/dmr/qed.html>.
- [2] Free Software Foundation, Inc. *ed manual*, November 1994.  
ed(1).
- [3] Free Software Foundation, Inc., 59 Temple Place – Suite 330, Boston, MA 02111-1307, USA. *sed manual*, 4.1.2 edition, August 2004.  
sed(1).
- [4] Free Software Foundation, Inc., 59 Temple Place – Suite 330, Boston, MA 02111-1307, USA. *sed info pages*, 4.1.2 edition, August 2004.  
info sed.
- [5] various. sed FAQ.  
<http://www.faqs.org/faqs/editor-faq/sed/>.
- [6] Brian W. Kernighan and Rob Pike. *The UNIX*

*Programming Environment*. Prentice–Hall International, Inc., Eaglewood Cliffs, New Jersey, NJ 07632, USA, 1984.

Excellent introduction into Shell Programming, sed, and awk. Discusses programming style, i.e. how to build a filter etc.

ISBN 0-13-937699-2 (hardcover),  
0-13-937681-X (paperback).

- [7] Dale Dougherty and Arnold D. Robbins. *sed & awk*. O’Reilly & Associates, Inc., 103 Morris Street, Suite A Sebastopol, CA 95472, USA, second edition, March 1997.  
ISBN 1-56592-225-5.

- [8] Arnold D. Robbins. *sed & awk Pocket Reference*. O’Reilly & Associates, Inc., 103 Morris Street, Suite A Sebastopol, CA 95472, USA, First edition, January 2000.  
ISBN 1-56592-729-X.

- [9] Brian W. Kernighan. *A Tutorial Introduction to the UNIX Text Editor*, Seventh edition, ca. 1977.  
7<sup>th</sup> edition Unix manual.

`~ftp/pub/doc/OS/Unix/7thEdMan/  
vol2/edtut.{ps,txt}.gz.`

- [10] Lee E. McMahon. *SED – A Non-interactive Text Editor*, Seventh edition, August 1978.

7<sup>th</sup> edition Unix manual.

`~ftp/pub/doc/OS/Unix/7thEdMan/  
vol2/sed.{ps,txt}.gz.`