

Udgave: November 1987
Forfatter: Henning Godske

Titel: SE Skærmeditor. Reference Manual

Nøgleord: Editor, SE, RC8000, Skærmeditor.

Resume: SE er en skærmorienteret editor til RC8000 med terminaler tilsluttet via RC3600 eller lokalnet.

SE vedligeholder skærbilledet, så det er en kopi af et udsnit af den tekstfil, der editeres.

Copyright 1987, A/S Regnecentralen af 1979

RC Computer A/S

Printed by A/S Regnecentralen af 1979, Copenhagen

Users of this manual are cautioned that the specifications contained herein are subject to change by RC at any time without prior notice. RC is not responsible for typographical or arithmetic errors which may appear in this manual and shall not be responsible for any damages caused by reliance on any of the materials presented.

INDHOLDSFORTEGNELSE

SIDE

1.	INTRODUKTION	1
1.1	Notation	1
1.1.1	Tegn	1
1.1.2	Kontroltegn	1
1.1.3	Tekst	1
1.1.4	Tekstafsnit	1
1.1.5	Linie og tekstlinie	2
1.1.6	Ord	2
1.2	Kontroltaster	2
1.2.1	Prefix	2
1.2.2	Definition af kontroltasternes kommandoer	2
1.2.3	Prefix tastens funktioner	3
1.3	Skærmen	4
1.3.1	Opdeling	4
1.3.2	Markøren	4
1.3.3	Liniearkeringer	4
1.3.4	Statuslinien	4
2.	EDITERINGS TILSTANDE	6
2.1	Skærmeditering	6
2.1.1	Overskrivning	6
2.1.2	Indsætning	6
2.2	Kommandolinie editering	7
2.3	Register kommando editering	7
3.	KOMMANDO UDFØRELSE	8
3.1	Kommando syntaks	8
3.2	Kommando semantik	9
3.3	Parametre	10
3.4	Flag	11
3.5	Fortrydelse og afbrydelse af kommandoer	12
3.6	Typehead	13

INDHOLDSFORTEGNELSESIDE

4.	EDITOR ORDERER	14
4.1	Simple markørflytninger	14
4.1.1	Tegn til højre CR.....	14
4.1.2	Tegn til venstre CL	14
4.1.3	Linie ned LD	14
4.1.4	Linie op LU	15
4.1.5	Linie slutning LE	15
4.1.6	Linie start LS	15
4.1.7	Næste linie NL	15
4.1.8	Skærm bund SB	15
4.1.9	Skærm top ST	15
4.1.10	Skip linie ned SD	16
4.1.11	Tekst slut JB	16
4.1.12	Tekst start JT	16
4.1.13	Næste skærmside NP	16
4.1.14	Forrige skærmside PP	17
4.1.15	Tabulatorstop TR	17
4.1.16	Ord til højre WR	17
4.1.17	Ord til venstre WL	17
4.1.18	Indrykning ID	18
4.2	Udvidede markørflytninger	18
4.2.1	Find tekst FS	18
4.2.2	Find tekst igen FN	19
4.2.3	Find mærke FM	19
4.2.4	Find linie afslutning FL	20
4.2.5	Find side mærke FP	20
4.2.6	Find tegn FC	20
4.2.7	Flyt til linie JL	20
4.2.8	Gå til flag JF	21

INDHOLDSFORTEGNELSE

SIDE

4.3	Indsætning i teksten	21
4.3.1	Indsæt tegn IC	21
4.3.2	Indsæt blanktegn IB	21
4.3.3	Indsæt ny linie IL	22
4.3.4	Indsæt sideskift IP	22
4.3.5	Indsæt tekst IT	22
4.4	Sletning og erstatninger	23
4.4.1	Slet tegn DC	23
4.4.2	Slet ord DW	23
4.4.3	Slet linie DL	23
4.4.4	Find og erstat tekst SS	24
4.4.5	Find og erstat næste SN	26
4.5	Registre	26
4.5.1	Sæt mærke i tekst SM	27
4.5.2	Kopier til register RC	27
4.5.3	Flyt til register RM	28
4.5.4	Indsæt register i tekst RI	28
4.5.5	Slet register RD	28
4.5.6	Vis registerindhold RL	29
4.5.7	Vis benyttede registre RN	29
4.5.8	Udskriv register i fil RW	29
4.5.9	Læs til register fra fil RR	30
4.5.10	Kopier blok til register BC	30
4.5.11	Flyt blok og slet BE	30
4.5.12	Indsæt blok i tekst BI	31
4.5.13	Flyt blok og fjern BM	31
4.5.14	Overskriv blok i tekst BO	31
4.6	Tabulatoren	31
4.6.1	Flyt til stop til højre TR	31
4.6.2	Flyt til stop til venstre TL	32
4.6.3	Modificer tabulatorstop TM	32
4.6.4	Slet tabulatorstop TD	32
4.6.5	Hent tabulatorlinie TG	32
4.6.6	Sæt tabulatorlinie TS	32

INDHOLDSFORTEGNELSE

SIDE

4.7	Afslutning og opdatering af editering	32
4.7.1	Forlad editor EX	33
4.7.2	Forlad uden opdateringer EQ	34
4.7.3	Forlad editor med valgt scope ET EL EU EP	34
4.7.4	Opdater fil EB	35
4.8	Diverse andre ordre	36
4.8.1	Vis hjælpe sider HE HD	36
4.8.2	Gå til kommando tilstand CM	36
4.8.3	Udfør kommando i register XR	37
4.8.4	Definer kontrol tast DK	37
4.8.5	Vis kontrol tast SK	38
4.8.6	Udfør kontrol tast XK	38
4.8.7	Vis linienumre SL	39
4.8.8	Sæt flag SF	39
4.8.9	Slet flag CF	39
4.8.10	Kontrol information CI	40
4.8.11	Fjern blank tegn CB	42
4.8.12	Juster skærmen JS	42
4.8.13	Skriv på statuslinie WT	42
4.8.14	Hent værdi til variabel GV	43
4.8.15	Hent dato til variable GD	43
4.8.16	Hent tidspunkt GT	43
4.8.17	Optæl variabel IV	43
4.8.18	Nedtæl variabel DV	44
4.8.19	Hent linie til variabel PV	44
4.8.20	Hent FP parameter GP	44
4.8.21	Sæt linie mærke fra/til LM	44
4.8.22	Opdater skærbilledet UL US	45
4.8.23	Udfør intet NO	45
5.	KALD OG RESOURCER	46
5.1	Editorkald	46
5.2	Resourcer	47

<u>INDHOLDSFORTEGNELSE</u>	<u>SIDE</u>
6. PARAMETERFILEN	48
7. BEMÆRKNINGER	54
7.1 Timeout	54
7.2 Arbejds området	54
7.3 Attention, afbrydelse og fejl	55
8. HJÆLPEFILENS OPBYGNING	56

INDHOLDSFORTEGNELSE**SIDE****BILAG**

A.	SKÆRMKOMMANDOER	57
A.1	RC822 og RC855-tty standard	57
A.2	Partner ACP ANSI emulator	60
A.3	RC45 ANSI key layout	64
B.	SE ORDRER OVERSIGT	65
C.	ISO-TABEL FOR TEGN BRUGT I SE	68
D.	FEJLMEDDELELSER	69
D.1	Opstart meddelelser	69
D.2	Editerings meddelelser og fejl	71
E.	EKSEMPLER PÅ BRUG AF SE	75
1.	Indtastning af en ny tekst	75
2.	Rettelser i en tekst	76
3.	Gentagelser	78
4.	Ordre	78
5.	Kommandoer	79
6.	Definition af kontroltaster	79
7.	Brug af register	80



1. INTRODUCTION

SE er en skærmorienteret editor til RC8000 med terminaler tilsluttet via RC3600 front-ends eller lokalnet (RCnet). SE vedligeholder skærbilledet, så det er en kopi af et udsnit af teksten i den fil, som editeres.

SE gør det muligt at bladre igennem en tekst, positionere markøren et vilkårligt sted i teksten, foretage indsættelse, sletning og andre rettelser og umiddelbart få resultatet af ændringerne vist på skærmen.

SE er specielt velegnet til editering og indtastning af kildetekster og andre lignende tekster, idet SE bl.a. har ordre til opstilling af indrykkede tekster, og ordre til kolonne og linie kopiering.

I bilag E er brugen af SE illustreret ved eksempler.

1.1 Notation

I denne manual benyttes en række begreber og betegnelser ved beskrivelsen af editorens ordresæt og funktion. I det følgende afsnit gennemgås disse.

1.1.1 Tegn

Som tegn betragtes alle synlige symboler (inkl. tal og bogstaver), der indgår i iso tegnsættet (se bilag C).

1.1.2 Kontroltegn

Som kontroltegn betragtes alle iso-tegn med værdi mellem 0 og 31 samt iso-tegnet med værdien 127.

I tekster editeret med SE benyttes kun kontroltegnene 10 (nl = næste linie '<') og 12 (ff = sideskift '\$'). De resterende kontroltegn markeres i teksten med tegnet '@'.

Tegnet med iso-værdien 0 kan ikke indgå i teksten !

1.1.3 Tekst

En tekst betegner den samling af tegn og kontroltegn, der udgør indholdet af den fil, der editeres (editfilen).

1.1.4 Tekstafsnit

Et tekstafsnit er en nærmere udpeget del af en tekst.

1.1.5 Linie og tekstlinie

En linie udgøres af de tegn, der står på en enkelt linie på skærmen.

En tekstlinie kan derimod bestå af op til flere linier, og udgøres af de tegn der læses mellem 2 'nl' (eller 'ff') kontroltegn fra editfilen.

1.1.6 Ord

Et ord betegner en sammenhængende følge af ikke-delimitortegn. Delimitortegnene udgøres af de tegn, der er angivet i parameterfilen (se afsnit 6. parameter 19) samt afslutningen af en linie.

1.2 Kontroltaster

Kontroltasterne er de taster (eller tastkombinationer) på tastaturet, der sender iso-tegn med værdier mellem 1 og 31 samt værdien 127.

En kontroltast kan enten være en enkelt tast (funktions-tast) eller en enkel tast i kombination med Ctrl-tasten (nedtrykket samtidig).

I det følgende beskrives f.eks. tastkombinationen Ctrl og X med Ctrl-X.

Kontroltasterne benyttes i SE bl.a. til at angive hvilke editerings kommandoer, der ønskes udført (se afsnit 3.).

I SE har alle kontroltaster (og funktionstaster) et nummer mellem 0 og 255. Det er dette nummer, der benyttes ved definition af en tasts kommando. Hvordan en tast tildeles et nummer er beskrevet i afsnit 6. (parameterfilen). Disse numre har ingen betydning ved normal editering (medmindre DK ordren benyttes).

1.2.1 Prefix

I SE udpeges en af kontroltasterne til specielle formål. Denne tast kaldes prefix-tasten (prefix = foranstillet). (Dette er normalt HOME-tasten med værdien 29.)

Prefix-tasten nedtrykkes altid alene (til forskel fra f.eks. Ctrl-tasten og Shift-tasten).

1.2.2 Definition af kontroltasternes kommandoer

Hver kontroltast (eller funktions tast) kan defineres med to kommandoer, der udføres når der trykkes på den pågældende tast i skærmediterings tilstanden.

Den ene kommando udføres når der trykkes på kontroltasten (eller Ctrl - tast kombinationen) alene, mens den anden udføres når der først trykkes på prefix-tasten (se nedenfor) og dernæst kontroltasten.

Kontroltasterne defineres ved opstart fra parameterfilen (se afsnit 6.) og kan senere redefineres enkeltvis v.h.a. ordren DK (Define key; se afsnit 4.8.4)

I bilag A er givet en oversigt over standard definitionerne for kontroltasterne, men den aktuelle definition kan enten fås via SK (Show Key) ordren eller ved at udskrive den aktuelle parameterfil (eller følgende kommando kan f.eks. benyttes: rr par separ, rl par ; hvis editoren er kaldt med 'se')

1.2.3 Prefix-tastens funktioner

I det følgende gennemgås de 6 typer af funktioner, der kan udføres med <prefix>-tasten.

Funktionerne indledes altid med at trykke på prefix-tasten, hvorefter markøren flyttes ned til venstre på statuslinien.

<prefix> <kontroltast>

: Her udføres den kommando, der er defineret for kontroltasten sammen med prefix.

<prefix> <tal> <kontroltast>

: Her udføres den kommando, der er defineret for kontroltasten alene, det antal gange tallet angiver.

<prefix> <tal> <prefix> <kontroltast>

: Her udføres den kommando, der er defineret for kontroltasten sammen med prefix, det antal gange tallet angiver.

<prefix> <ordre> <prefix> eller

<prefix> <ordre> <cr>

: Her udføres den SE ordre der angives. Der kan kun afgives en enkelt ordre i modsætning til den næste prefix funktion. Det er ikke muligt at rette i den indtastede ordre !

<prefix> <prefix>

: Her skiftes til kommandolinie editering. Dette er det samme som at give ordren CM.

<prefix> <kontrol tegn>

: Dette kan kun benyttes når der indtastes tekst på statuslinien. Iso-værdien af kontroltasten (1 til 31) indsættes i den indtastede tekst (se tabel i bilag C for definition af iso-værdierne).

1.3 SKÆRMEN

Terminalens skærm udgør det arbejdsfelt, der benyttes ved kommunikation med SE.

Skærmen består af et antal linier (normalt 24), der hver består af et antal positioner (kolonner; normalt 80).

SE vedligeholder skærbilledet ud fra de interne oplysninger om tilstande m.m. SE kan altså ikke 'se', hvis billedet bliver ødelagt af udefrakommende meddelelser o.a. og heller ikke at der evt. benyttes forkerte kontroltegn til at styre skærmen (p.g.a. forkert parameterfil). Et ødelagt skærbillede kan dog normalt oprettes ved hjælp af editorordren US (se afsnit 4.8.22).

1.3.1 Opdeling

SE opdeler skærmen i en tekstdel og en statuslinie.

Statuslinien er den nederste linie på skærmen, mens tekstdelen består af de øvrige linier på skærmen.

1.3.2 Markøren

Skærmens markør (engelsk: cursor) benyttes af SE til at markere enten den position, hvor SE's interne markør står i teksten (i resten af manualen blot kaldet markøren), eller hvis den står på statuslinien enten til at markere, at der er tastet <prefix>, og at der kan tastes yderligere information til SE i forbindelse hermed, eller til at markere positionen ved indtastning af kommando (se afsnit 3).

1.3.3 Liniemarkeringer

I SE benyttes et sæt af liniemarkeringstegn til at markere en linies afslutning. En linie kan afsluttes på en af 3 måder:

- 1) Linien er afsluttet af et nylinietegn 'nl' <
- 2) Linien er afsluttet af et sideskifttegn 'ff' \$
- 3) Tekstlinien fortsætter på næste linie. +

Udskriften af nylinie-tegnet styres af ordren LM (se 4.8.21)

1.3.4 Statuslinien

Statuslinien i sidste linie på skærmen benyttes af SE til at markere de forskellige editeringstilstande og til at markere for indtastning af information til SE, samt til udskrivning af oplysninger og fejlmeddelelser fra SE.

Følgende tilstande kan angives på statuslinien:

INSERTING : Indsætningstilstand. Se afsnit 4.3.5.

INDENT : Indrykningstilstand. Se afsnit 4.1.8.

COMMAND : Kommandolinie editerings-tilstand. Se afsnit 2.2.

Please wait : En kommando er under udførelse.

Reorganizing : Der foretages reorganisering af det interne arbejdsområdes indhold. Dette sker fordi markøren er flyttet op over arbejdsområdet. Se afsnit 7.2

2. EDITERINGS TILSTANDE

Indholdet i den tekst, der arbejdes med, kan ændres eller slettes, og der kan tilføjes ny tekst. Dette kan foretages på 3 forskellige måder: som skærmeditering, som kommandolinie editering og ved register kommando editering.

Med de forskellige editerings former kan der i princippet foretages nøjagtig den samme editering. Editerings-ordrerne til SE gives blot på forskellig måde.

I det følgende betegner en kommando en samling af ordrer (se afsnit 3.1 for definition af kommandoer)

2.1 Skærmeditering

Her gives ordrene ved tryk på kontroltasterne (og evt. funktionstaster), samt ved benyttelse af prefix tasten i kombination med en kontroltast.

SE starter altid i denne tilstand og vil for det meste befinde sig her, idet den egentlige fuldskræmseditering foregår i denne tilstand.

I denne tilstand vil markøren, når denne står i tekstdelen af skærmen, udpege det sted i teksten, hvor næste editering vil blive foretaget.

Markøren kan flyttes til et vilkårligt sted på skærmen og i teksten v.h.a. SE's ordre.

I bilag A gives en oversigt over de enkelte kontroltasters standard funktioner.

2.1.1 Overskrivning

Indtastes tegn, mens markøren står i tekstdelen, vil disse overskrive de tegn, der i forvejen stod på det pågældende sted i teksten.

2.1.2 Indsætning

Skal der indsættes tegn i teksten mellem allerede eksisterende tegn, skal der gøres plads til disse først.

Dette kan gøres ved enten at indsætte et passende antal blanke tegn (mellemrum) eller ved at indsætte en ny tom linie.

Endelig kan hele tekstafsnit indsættes enten fra registre eller ved direkte indtastning i indsætningstilstanden (Insert mode).

Samtlige indsætningsordrer gennemgås i afsnit 4.3 og kontroltasterne, der indeholder disse ordrer, er beskevet i bilag A.

2.2 Kommandolinie editering

Ved kommandolinie editering udføres kommandoen ved indtastning af denne på statuslinien efter 'COMMAND:'.

En kommando kan bestå af indtil flere ordre eventuelt repeteret.

Efter at kommandoen er udført returneres til skærmediterings tilstanden og evt. ændringer vises på skærmen.

For at komme i kommandolinie editerings tilstand fra skærmediterings tilstand trykkes to gange på prefix-tasten (beskrevet i denne manual som <prefix> <prefix>).

2.3 Register kommando editering

Kommandoer kan også udføres fra et register (se afsnit xx.x for definition af registre). Her læses kommandoerne en linie af gangen og udføres som om de blev indlæst i kommandolinie tilstanden. Dog opdateres skærmen først når alle linier fra registret er udført. For at udføre kommandoer i et register skal ordren XR (eXecute Register) benyttes. Denne ordre er beskrevet i afsnit 4.8.3.

3. KOMMANDO UDFØRELSE

3.1 Kommando syntaks

Her gives syntaksen for sammensætningen af SE's ordre til en kommando. Denne syntaks gælder for kommandoer givet ved kommandolinie editering, for kommandoer defineret for kontroltasterne og dermed også for kommandoer givet i parameterfilen, samt for kommandoer givet i et register ved register kommando editering.

```

<kommando>          ::= <ordre>
                        eller
                        <kommandosekvens>
                        eller
                        <gentagkommando>

<kommandosekvens> ::= <kommando>
                        <seperator>
                        <kommandosekvens>

<gentagkommando>  ::= <nummer>
                        <starttegn>
                        <kommando>
                        <sluttegn>
                        eller
                        <kont>
                        <starttegn>
                        <kommando>
                        <sluttegn>

<ordre> <param>    ::= To-bogstavs ordre evt. med
                        parametre.

<seperator>        ::= ,   eller   .

<starttegn>        ::= <   eller   (

<sluttegn>         ::= >   eller   )

<nummer>           ::= tal   eller   &   eller   <var>

<kont>             ::= !   eller   $   eller   &   eller   <var>

```

<param>	::= <navn>
	eller
	<tekst>
	eller
	<nummer>
	eller
	&
	eller
	<var>
<var>	::= % efterfulgt af tal fra 0 til 9
<navn>	::= Navn dannet af indtil 11 tegn, der opfylder syntaksen for 8000 navne.
<tekst>	::= Tekst på indtil 80 tegn (skærmbred- den) evt. indeholdende kontroltegn.

3.2. Kommando semantik

Ordrene i en kommando skrives i den rækkefølge, de skal udføres adskilt af komma eller punktum. Evt. parametre til ordrene skrives umiddelbart efter ordre navnet med en eller flere blanke mellem. Seperatortegnet i FS og SS ordrene skal dog komme umiddelbart efter navnet ellers opfattes det første blanktegn som seperator (se 3.3).

Skal en sekvens af ordrer gentages, omgives disse af et sæt start- og sluttegn og antallet af gentagelser angives umiddelbart foran starttegnet.

Ønskes sekvensen af ordrer gentaget, så længe det er muligt, benyttes ! eller \$ i stedet for et tal. Udførelsen af sekvensen stopper da kun når en af ordrene FS, FN, SS eller SN terminerer med 'String not found' (se dog afsnit 3.5 omkring afbrydelse).

Benyttes ! stopper udførelsen af hele kommandoen, mens kun udførelsen af den sekvens, der omgives af de tilhørende start- og sluttegn afsluttes ved benyttelse af \$.

Ved \$ udskrives 'String not found' ikke !

Start- og slut tegnene skal benyttes parvis: < > eller (), men kan godt indgå blandet i flere niveauer. Start- og sluttegnene må dog ikke indgå i parametrene til en ordre, der indgår i den sekvens (eller en indeholdt undersekvens), der er omgivet af tegnene.

Eks. \$<fs/(*/>,!(ss:<>:<:)) er korrekt, mens
 \$(fs/(*/),!<ss:<>:<:)> ikke vil virke.

Eks. !<fs/(*/,sm,fs/*)//,3(cr),rm>;
 Denne kommando fjerner alle kommentarer i f.eks. et pascal-program, hvis den gives når markøren står på første linie i teksten.

Efter en kommando kan tilføjes kommentarer, hvis disse indledes med ;

En kommando linie kan maximalt indeholde det antal tegn, der kan stå på 1 linie på skærmen (normalt 80).

Eks. ls,25<dl,ld> ;Slet hveranden linie 25 gange

3.3 Parametre

Mange af SE's ordrer kan have en eller flere parametre.

Disse kan opdeles i 4 typer.

- 1) Tekster. En tekst kan bestå af op til 80 iso-tegn (tal, bogstaver og kontroltegn). Teksten kan enten stå i en variabel, læses fra statuslinien når dette markeres eller stå direkte efter ordren. I det sidste tilfælde skal den omgives af to ens seperatortegn. Seperatortegnene kan være ethvert iso-tegn, der ikke indgår i teksten selv og som ikke benyttes af SE (dvs det kan ikke være & % . , () < > \$ eller !).

Hvis intet angives benyttes den tomme tekst. Denne kan også fås, hvis der ikke angives nogle tegn mellem seperatortegnene.

- 2) Tal. Tal benyttes i SE enten til at angive et antal gange en kommando parentes skal udføres, til at angive f.eks et linienummer eller til at angive nummeret på en variabel eller et flag.

Hvis intet angives benyttes normalt værdien 0.

Tal der benyttes af SE skal være positive eller 0 !

Flagene med nummer 0 til 9 kan sættes, mens der kan hoppes til flag mellem 0 og 11. Nogle af flagene sættes automatisk af SE. Nedenfor er angivet brugen af flagene.

- 0 : Benyttes som mærke ved register ordre. Kan også sættes med ordren SM. Flaget markeres i teksten med et inverteret tegn (hvis muligt).
- 1 - 5 : Kan frit benyttes. Ændres ikke af SE.
- 6 : Sættes ved start af en søgeordre.
(FS, FN, SS, SN, FL, FP, FC, FM)
- 7 : Sættes ved sidst fundne position ved en substitute ordre (SS, SN).
- 8 : Sættes ved sidste fundne position ved en find ordre (FS, FN).
- 9 : Kan frit benyttes. Ændres ikke af SE.
- 10 : Sættes på sidste linie i SE's arbejdsområde.
- 11 : Sættes på første linie i SE's arbejdsområde.

Sættes et flag, der allerede er sat, flyttes positionen til det nye sted !.

3.5 Fortrydelse og afbrydelse af kommandoer

Kontrolltasten Ctrl-X kan benyttes til at fortryde og afbryde kommandoer. I SE kaldes denne tast Cancel tasten.

Bemærk: Cancel-tastens værdi er bestemt af parameter-filen, den aktuelle værdi kan ses ved benyttelse af ordren CI. I denne manual antages det at værdien er Ctrl-X.

En kørende repeterende eller længrevarende kommando kan afbrydes ved at trykke Ctrl-X (hvis typeahead er aktiv). Udførelsen af kommandoen vil da stoppe efter et antal sekunder (givet ved timeout, se afsnit 7.1). Kommandoen kan evt. genoptages.

Under indtastning af en kommando kan Ctrl-X benyttes til at fortryde hele kommandolinien.

Ligeledes kan Ctrl-X benyttes til at fortryde indtastning af parametre m.m.

Tastes <prefix> Ctrl-X fortrydes <prefix>.

Teksten der indtastes ved COMMAND, Find m.m. kan editeres med 'slet tegn', 'rubout' eller venstre-pil tasterne.

Ved afbrydelser kan det normalt vælges om kommandoen skal genoptages eller om der skal returneres til skærm editerings tilstand. Er kommandoen blevet afbrudt med Ctrl-X fortsættes udførelsen fra det sted den blev afbrudt. Er kommandoen afbrudt pga. fejl kan den fortsættes med den næste ordre efter den kommando struktur, der blev afbrudt (dvs. efter en evt. parentes struktur).

3.6 Typeahead

I SE er der mulighed for at benytte typeahead dvs. forudtastning.

Er typeahead sat i parameterfilen og er terminalen i stand til at benytte dette, kan der skrives forud. Typeahead har ingen betydning ved normal indtastning af tekst, men ved benyttelse af bl.a. kontroltasterne vil det være muligt at taste videre uden at skulle vente på at den tilhørende kommando bliver færdig-udført.

Ved prefix funktioner er det heller ikke nødvendigt at vente på at markøren står på statuslinien, før de øvrige oplysninger til SE indtastes.

NB. Benyttes taster, der repeterer, vil alle modtagende tegn benyttes af SE, men det er ikke sikkert at f.eks. flytning af markøren på skærmen kan foregå med samme hastighed som tegnene modtages, og der vil derfor være et vist 'efterslæb', der kan gøre det svært at placere markøren.

4. EDITOR ORDRER

I det følgende gennemgås alle de ordrer, der genkendes af SE. Ordreerne er inddelt i grupper efter deres funktion. Hvert underafsnit beskriver den enkelte ordre, der i overskriften er angivet ved den danske betegnelse og ordrens navn (navnet er en memoteknisk forkortelse for den engelske betegnelse, der er angivet i parentes). En samlet alfabetsk oversigt findes i bilag B.

Ved ordre der kan have parametre, er der for hver parameter angivet standard værdien. (Den der benyttes hvis parameteren ikke angives. Skal parameteren angives er der anført -) Samt ledeteksten der udskrives ved brug af & (se 3.3).

4.1 Simple markørflytninger

Flytning af markøren i teksten påvirker aldrig indholdet af denne. Der indsættes dog automatisk blank tegn i en linie, når der indtastes et tegn på markørens position, hvis denne er efter sidste tegn i linien.

4.1.1 Tegn til højre CR (Cursor Right)

Syntaks: cr

Flyt markøren et tegn til højre. Står markøren i sidste kolonne på skærmen, flyttes til første kolonne på næste linie.

Eks. 15<cr>; flyt 15 tegn til højre

4.1.2 Tegn til venstre CL (Cursor Left)

Syntaks: cl

Flyt markøren et tegn til venstre. Står markøren i første kolonne, flyttes til positionen efter sidste tegn på forrige linie.

Eks. 10(cl); flyt 10 tegn til venstre

4.1.3 Linie ned LD (Line Down)

Syntaks: ld

Flyt markøren til næste linie. Er aktuel linie sidste linie i teksten, indsættes en ny linie efter denne. Kolonne positionen ændres ikke.

Eks. ld,ld; flyt 2 linier ned

4.1.4 Linie op LU (Line UP)

Syntaks: lu

Flyt markøren til forrige linie. Er aktuel linie første linie på skærmen flyttes skærbilledet det antal linier op, der er givet ved parameter 14 i afsnit 6 (parameterfilen; Dette er det samme som ved PP ordren).

Eks. 100<lu>; flyt 100 linier op

4.1.5 Linie slutning LE (Line End)

Syntaks: le

Flyt markøren til positionen efter sidste tegn på linien.

Eks. le,cl; flyt til sidste tegn i linien

4.1.6 Linie start LS (Line Start)

Syntaks: ls

Flyt markøren til første kolonne på linien.

Eks. ls,ld; flyt til første kolonne på næste linie

4.1.7 Næste linie NL (New Line)

Syntaks: nl

Flyt markøren til første kolonne på næste linie (se dog ID ordren afsnit 4.1.18). Er aktuel linie sidste linie i teksten, indsættes ny linie efter denne.

Eks. id,nl,id; flyt til første tegn på næste linie

4.1.8 Skærm bund SB (Screen Bottom)

Syntaks: sb

Flyt markøren til sidste linie på skærmen. Kolonne positionen ændres ikke.

4.1.9 Skærm top ST (Screen Top)

Syntaks: st

Flyt markøren til første linie på skærmen. Kolonne positionen ændres ikke.

4.1.10 Skip linie ned SD (Skip Down)

Syntaks: sd

Flyt markøren ned til første kolonne i næste ikke tomme tekstlinie. Som tom linie regnes både linier, der ikke indeholder tegn og linier der kun indeholder blank tegn.

Eks. wt/Algol line: /,gv1,jt,fs/begin/,%1(sd),lu;
Find en linie i en algol kildetekst under hensyntagen til algols linienummererings måde.

4.1.11 Tekst slut JB (Jump Bottom)

Syntaks: jb

Flyt markøren ned efter sidste tegn i teksten.

Ved store tekster tager denne ordre tid, og flytningen markeres derfor på statuslinien med: 'Moving to bottom'.

4.1.12 Tekst start JT (Jump Top)

Syntaks: jt

Flytter til første tegn i teksten.

Ved store tekster tager den ordre tid, og flytningen markeres derfor på statuslinien med: 'Moving to top'.

4.1.13 Næste skærmside NP (Next Page)

Syntaks: np

Flyt markøren ned på næste skærmside i teksten (hvis denne findes).

Der medtages nogle linier (antal bestemt af parameterfilen, se afsnit 6. parameter 14) fra den aktuelle side over på den nye side.

Markørens nye position bestemmes ud fra følgende:

Er markørens aktuelle position på en af de linier, der flyttes med over på ny side, forbliver markøren på denne position. Ellers sættes markøren på første linie på den nye skærmside.

Kolonne positionen ændres ikke.

Eks. 5<np>; flyt 5 sider ned i teksten

4.1.14 Forrige skærmside PP (Previous Page)

Syntaks: pp

Flyt markøren op på forrige skærmside i teksten, hvis denne findes

Der medtages nogle linier (antal bestemt af parameterfilen, se afsnit 6 parameter 14) fra den aktuelle side, over på den nye side.

Markørens nye position bestemmes ud fra følgende:

Er markørens aktuelle position på en af de linier, der flyttes med over på ny side, forbliver markøren på denne position. Ellers sættes markøren på sidste linie på den nye skærmside.

Kolonne positionen ændres ikke.

4.1.15 Tabulatorstop TR (Tabulator Right)

Syntaks: tr

Markøren flyttes til næste tabulatorstop til højre.

Er der ikke flere stop på aktuel linie, flyttes til første stop på næste linie. Er der ikke sat nogle tabulatorstop, flyttes markøren ikke. (Se afsnit 4.6 for yderligere ordre omkring tabulering).

Eks. le,tr; flyt til første tabulatorstop på næste linie

4.1.16 Ord til højre WR (Word Right)

Syntaks: wr

Flyt markøren til starten af første ord til højre.

Er der ikke flere ord på aktuel linie, flyttes til første ord på næste linie. For definition af et ord se afsnit 1.1.6.

4.1.17 Ord til venstre WL (Word Left)

Syntaks: wl

Flyt markøren til starten af første ord til venstre.

Står markøren i et ord flyttes til starten af dette. Er der ikke flere ord på aktuel linie, flyttes til starten af sidste ord på forrige linie.

4.1.18 Indrykning ID (InDent)

Syntaks: id

Indrykningstilstanden (Indent mode) skiftes ved brug af ID ordren.

Tilstanden har betydning for hvordan NL (Næste linie) ordren flytter markøren. Tilstanden markeres på statuslinien med INDENT når indrykningstilstanden er aktiv.

I passiv tilstand flyttes markøren som beskrevet under NL ordren (afsnit 4.1.7).

I aktiv (INDENT) tilstand flyttes markøren som følgende:

- a) Hvis næste linie (linien der skal flyttes til) ikke er tom, flyttes markøren til det første ikke blanke tegn på linien.
- b) Hvis næste linie er tom (linielængde 0 eller kun blanke) sættes markøren på positionen under det første ikke blanke tegn på den første ikke blanke linie ovenfor. Hvis linien ovenfor er fortsat over mere end en skærm-linie (afsluttet med '+'), benyttes den første af disse skærmlinier.

4.2 Udvidede markørflytninger og søgninger

Disse ordre benyttes hvis markøren skal flyttes over et større område eller hvis der skal søges i teksten.

4.2.1 Find tekst FS (Find String)

Syntaks: fs<string>

<string>: & , Find:

Find angivet søgetekst <string> i tekst. Der søges efter angivet søgetekst fra positionen et tegn efter aktuel markør position og videre frem i teksten.

Findes søgeteksten, sættes markøren på første tegn i denne. Findes søgeteksten ikke markeres dette på statuslinien ('String not found'), og markøren sættes efter sidste tegn i teksten.

Søgeteksten <string> kan angives på følgende måder:

- 1) Lige efter ordren omgivet af to ens seperatortegn (et hvert tegn der ikke indgår i søgeteksten bortset fra tegn brugt til adskillelse af ordre samt % og & tegnene) med følgende syntaks:

fs<seperator> <søgetekst> <seperator>

Eks. fs/abc/ ; Find teksten abc
'/' er seperator tegnet

- 2) På statuslinien når der markeres herfor. Ordren har da følgende syntaks:

fs&

- 3) Tekst indeholdt i en variabel <var>.

fs<var>

Eks. fs%4; Find teksten indeholdt i variabel 4

Skal der indgå kontroltegn i søgeteksten kan disse indsættes ved at taste <prefix> <control tegn> når søgeteksten skrives. Dette gælder ved alle 3 angivelsesmåder.

Eks. Søg efter blank efterfulgt af sideskift (Ctrl-L). Ved indtastning af søgetekst tastes:
mellemrum <prefix> <ctrl-L>

Alle kontroltegn værdier undtagen Ctrl-X (cancel) og Ctrl-H (back space) kan benyttes.

Dog vil Ctrl-M ,<cr> (lig 13 decimalt) og Ctrl-Y blive konverteret til Ctrl-J (10 decimalt = new line).

Pas på ikke at taste <prefix> og en funktionstast, der indeholder mere end et kontroltegn !

4.2.2 Find tekst igen FN (Find Next)

Syntaks: fn

Find næste forekomst af søgeteksten, der blev angivet i sidste FS ordre. Søgningen foretages som ved FS ordren

4.2.3 Find mærke FM (Find Mark)

Syntaks: fm

Find evt. sat mærke i teksten. Findes mærket sættes markøren på dettes position, ellers flyttes markøren ikke.

4.2.4 Find linie afslutning FL (Find Line end)

Syntaks: fl

Find næste linie afslutnings tegn ('nl').

Ved at benytte ordren LM angives disse i teksten som <

4.2.5 Find side mærke FP (Find Pagemark)

Syntaks: fp

Find næste sidemærke ('ff').

Sidemærker angives i teksten som \$

4.2.6 Find tegn FC (Find Character)

Syntaks: fc <value>

<value>: - , Character value:

Find tegn med angivet iso-værdi <value> (se bilag C).

Søgningen foretages som angivet i FS ordren.

Eks. fc 9,dc,ib; Find næste 'HT tegn' og erstat det med et blank tegn.

4.2.7 Flyt til linie JL (Jump Linenumber)

Syntaks: jl <line number>

<line number>: 1 , Jump to line:

Flyt markøren til det linienummer <line number>, der er angivet.

Der kan flyttes til enhver linie inden for teksten.

Linierne nummereres fra første linie (1) og fortløbende optalt med 1. Linienumrene ændres dynamisk, hvis der indsættes eller slettes linier. Se iøvrigt SL (Vis linienummer) ordren afsnit 4.8.5.

Eks. jl 1,ls; flyt til første tegn i teksten

Eks. jl& ; flyt til det linienummer, der indtastes på statuslinie, når dette markeres.

4.2.8 Gå til flag JF (Jump Flag)

Syntaks: jf <flag number>

<flag number>: 0 , Jump to flag number:

Markøren flyttes til den position i teksten, hvor flaget med det angivne nummer <flag number> blev sat (se SF ordren 4.8.9 og afsnit 3.4).

4.3 Indsætning i teksten

Ved indsættelse af ny tekst i en linie kan denne blive for lang til at stå på linien, den resterende del vil da blive flyttet ned på en ny linie, der indsættes på skærmen efter den aktuelle. Det markeres i højre kolonne af aktuel linie, at den fortsætter på næste linie på skærmen (se afsnit 1.3.3). I resultatfilen vil linien dog stadig optræde som én lang linie.

4.3.1 Indsat tegn IC (Insert Character)

Syntaks: ic <value>

<value>: - , Character value:

Indsat tegnet med iso-værdien <value> på markørens position, og flyt resten af linien et tegn til højre. Markøren flyttes ikke.

Alle tegn-værdier mindre end 127 og forskellig fra 0 kan indsættes. (25 tolkes som 10 ('nl')).

Værdier mindre end 32 angives på skærmen som @.

Eks. ic 9 ; Indsat et horisontalt tabulator tegn.

Bemærk : Det er ikke muligt at sætte et kontroltegn som første tegn i en fil eller i et register !

4.3.2 Indsat blanktegn IB (Insert Blanck)

Syntaks: ib

Indsat et blanktegn (mellemrum) på markørens position, og flyt resten af linien et tegn til højre. Markøren flyttes ikke.

Eks. 10(ib); indsat 10 blanktegn foran og ved markøren

4.3.3 Indsæt ny linie IL (Insert Linemark)

Syntaks: il

Indsæt et linieafslutningstegn (nl) i teksten på markørens plads. Den evt. resterende del af linien fra markøren og til højre for denne flyttes ned på næste linie, men markøren flyttes ikke.

Er markørens aktuelle position i første kolonne i en linie, indsættes en tom linie før den aktuelle linie, og markøren placeres først på den nye linie.

Eks. il,nl; indsæt ny tom linie efter markøren og sæt markøren først på denne

4.3.4 Indsæt sideskift IP (Insert Pagemark)

Syntaks: ip

Indsæt et sideskift (form feed) i teksten på markørens position.

Sideskift markeres med tegnet \$.

Indsættelsen fortages iøvrigt som ved IL ordren (afsnit 4.3.3)

4.3.5 Indsæt tekst IT (Insert Text)

Syntaks: it ELLER it<seperator><nl><string><seperator>

Tekstindsætningstilstanden (Insert mode) skiftes med denne ordre hvis SE er i skærm editering eller kommando editering. I Register editering benyttes IT ordren til at indsætte tekst direkte.

Er indsætnings tilstanden, under skærm editering, aktiv markeres dette på statuslinien med INSERTING.

Når der skiftes fra passiv til aktiv tilstand slettes teksten fra markørens position (ink.) og resten af skærmen. Tekst der herefter indtastes indsættes mellem den synlige tekst og den tekst der blev slettet fra skærmen.

Når der returneres til passiv tilstand opdateres skærmen således at den fulde tekst atter bliver synlig.

Nogle ordrer vil skifte indsætningstilstanden til passiv hvis de benyttes i aktiv tilstand. Dette vil ske som om der blev udført en IT ordre før den aktuelle ordre.

I bilag B er ordrer, der kan benyttes i aktiv tilstand uden at skifte denne, angivet med Yes i 'Insert mode' kolonnen.

Er aktuel linie i aktiv tilstand den sidste på skærmen, vil der blive tilføjet en ny tom linie efter denne, således at tekstindsætningen kan fortsætte ud over siden.

Benyttes IT ordren ved register editering indsættes de angivne tegn direkte i teksten startende på markørens position.

Lige efter ordren angives <separator>. Al tekst fra næste linie og frem til næste <separator> indsættes. <separator> må ikke være linieafslutnings tegnet !

Eks. it/
 Test tekst, der fylder
 to linier uden afsluttende linieskift/

Eks. it:
 Tekst der indeholder linieskift efter sidste tegn
 :

4.4 Sletning og erstatninger

Denne gruppe af ordrer sletter dele af teksten. Det er muligt at slette en vilkårlig del af teksten. Slettes liniemarkerings tegnene vil linierne blive sammensat (konkateneret). Slettes et tegn eller en linie, hvori der står et flag (mærke) flyttes flaget til det forrige tegn eller den forrige linie.

4.4.1 Slet tegn DC (Delete Character)

Syntaks: dc

Slet tegnet på markørens plads og flyt den resterende del af linien til højre for markøren et tegn til venstre.

Står markøren på positionen lige efter sidste tegn i den aktuelle linie, slettes linieafslutningstegnet (nl), og linien sammensættes med den efterfølgende.

Markøren flyttes ikke ved DC ordren.

4.4.2 Slet ord DW (Delete Word)

Syntaks: dw

Slet alle tegn fra markørens position (inkl.) og frem til starten af næste ord.

For definition af et ord se afsnit 1.1.6.

4.4.3 Slet linie DL (Delete Line)

Syntaks: dl

Slet tegn fra og med markørens position og resten af tekstlinien (dvs. til og med første linieafslutningstegn 'nl' eller 'ff'). En evt. efterfølgende linie sammensættes med den aktuelle.

4.4.4 Find og erstat tekst SS (Substitute String)

Syntaks: ss<sub_string><rep_string><confirm>

<sub_string>: & , Substitute:
 <rep_string>: & , With:
 <confirm> : no , Confirm substitute ?

Find angivet søgetekst <sub_string>, og erstat denne med angivet erstatningstekst <rep_string>.

Søgningen foretages som ved FS ordren (se afsnit 4.2.1), dog startes ved markørens aktuelle position.

Findes søgeteksten slettes denne og erstatningsteksten ind-sættes i stedet.

Søgeteksten <sub_string> og erstatningsteksten <rep_string> kan begge angives på en af følgende måder:

- 1) Omgivet af et seperator tegn der ikke indgår i teksten. Angives begge tekster på denne måde gælder der at der kun må angives et seperator tegn mellem de to tekster og at seperator tegnet derfor skal være ens for begge tekster.

Eks. ss/OK/NOT OK/ ; Erstatte første forekomst af OK med NOT OK

- 2) På status linien når der markeres herfor. Teksten angives da med &.

Eks. ss&& ; Der markeres for indtastning af begge tekster.

Eks. ss/OK/& ; Der markeres kun for indtastning af erstatningsteksten, mens søgeteksten er givet som OK

- 3) I variabel.

Eks. ss%3& ; Søgeteksten er givet ved indholdet af variabel 3 mens der markeres for indtastning af erstatningsteksten.

Enhver kombination af de 3 måder kan benyttes !

Parameteren <confirm> angiver om en erstatning skal bekræftes før den foretages, det markeres på statuslinien for dette når søgeteksten er fundet (skærmen opdateres og markøren sættes ved den fundne tekst).

Svares Ctrl-Y (Yes=JA) udføres erstatningen.

Svares Ctrl-N (No=Nej) afsluttes SS ordren uden at udføre erstatningen. Markøren sættes på positionen én efter første tegn i den fundne søgetekst.

Svares Ctrl-E (End=Slut) stoppes en evt. repeterende kommando. Hvis repetitionen er startet med \$ fortsættes med næste ordre. For at stoppe hele kommandoen skal Ctrl-X benyttes.

Bemærk: Der kan KUN svares med kontrol tegn ! Andre tegn vil blive skrevet direkte på skærmen og overskrive teksten, men de vil dog ikke blive indsat.

Parameteren kan angives på følgende måder:

1) Direkte i ordren. Ønskes bekræftelse angives et ?

Eks. ss/a/b/?

2) Indlæst fra status linien. Ønskes bekræftelse indtastes yes eller y når der markeres herfor.

Eks. ss/a/b/&

3) I en variabel. Ønskes bekræftelse skal variabelen indeholde yes eller y.

Eks. ss/a/b/%1

Angives intet eller no (dog ikke no i stedet for ? direkte i ordren) udføres erstatningen med det samme.

Kontroltegn kan benyttes både i søgeteksten og i erstatningsteksten (se FS ordrens beskrivelse af dette).

Der kan angives tomme tekster både for søgeteksten (markøren flyttes da ikke inden indsættelsen af erstatningsteksten) og for erstatningsteksten (virker da som FS pånær at markøren sættes et tegn efter den fundne tekst).

Eks. ss//%1 ; Indsæt indholdet af variabel 1 på markørens plads.

4.4.5 Find og erstat næste SN (Substitute Next)

Syntaks: sn<confirm>

<confirm> : no , Confirm substitute ?

Udfør søgning og erstatning som ved SS ordren med søge- og erstatningsteksten som angivet i sidste SS ordre.

Angives parameteren <confirm> ikke eller angives tomt svar ved prompt eller angives en tom variabel, benyttes værdien angivet i sidste SS ordre. Ellers benyttes den angivne værdi som beskrevet i SS ordren afsnit 4.4.4.

Eks. sn ;Find og substituer næste søgetekst, der promptes kun for bekræftelse, hvis dette blev angivet i sidste SS ordre.

4.5 Registre

Editoren har et antal registre (max. 10), der dynamisk kan oprettes, nedlægges og navngives af brugeren.

Registerne kan benyttes til at flytte og kopiere tekst-afsnit samt til at flytte dele af teksten til og fra filer. Teksten der kopieres kan både være linie orienteret eller kolonne orienteret (blok orienteret).

Indholdet af et register defineres enten ved indlæsning fra en fil eller ved det tekstafsnit, der angives med mærke- og markørposition på det tidspunkt ordren gives.

Et tekstafsnit angives som teksten mellem et sat mærke (se SM ordren afsnit 4.5.1) og markøren. Markørens position kan være før mærkets !.

Ved de linie orienterede register ordrer (RC,RM) medtages tegnene fra første position og til, men ikke inklusive, sidste position.

Ved de kolonne orienterede ordrer (BC,BE,BM) udpeges de tegn der skal medtages, af de to hjørner i tegnblokken angivet ved markøren og mærket. Begge tegn i hjørnerne medtages. Angives tegn blokken ud over sidste tegn i en linie indsættes blanke i teksten i registret, men ikke i den originale tekst.

De kolonne orienterede ordrer kan kun benyttes på tekstlinier, der ikke er fortsat over flere linier på skærmen. Dette gælder også ved indsætning af disse. Tekst der indsættes ud over højre skærmerkant afskæres.

Registre der bruges ved kolonne orienterede ordrer bør ikke benyttes af de linie orienterede ordrer og omvendt, selvom dette er muligt.

Registernavnet, der ønskes benyttet i de følgende ordrer, angives efter ordren (med blank tegn i mellem). Disse skal overholde formatet for 8000 navne !

Angives intet navn (parameteren er tom) benyttes et forud-defineret register med navnet **default**. Dette register har en særstilling blandt de benyttede registre, idet dette register ikke skal slettes før der kan defineres nyt indhold. Dette er ellers tilfældet for de øvrige registre.

Registrene er i SE implementerede som filer med temporært scope, og der skal derfor være de fornødne ressourcer for at kunne benytte dem.

Antallet af linier, der kan overføres fra teksten til et register, er givet ved størrelsen af SE's arbejdsområde (se CI ordren og afsnit 5.2).

4.5.1 Sæt mærke i tekst SM (Set Mark)

Syntaks: sm

Sæt mærket, til brug for register ordrene, i teksten på markørens position.

Der kan kun sættes et mærke i teksten. Et evt. sat mærke bliver derfor slettet, inden det nye sættes.

Mærket er lig flag nummer 0 !

Benyttes en terminal hvor det er muligt at vise inverterede tegn og er dette angivet i parameterfilen (se afsnit 6 parameter 6) vises mærkets position i teksten, som et inverteret tegn.

Eks. fs/end/,sm,jf6; Sæt mærket ved næste forekomst af end. Markøren flyttes ikke !

4.5.2 Kopier til register RC (Register Copy)

Syntaks: rc <register name>

<register name>: default , To register:

Kopier tekstafsnit til register <register name>.

Eks. rc& ; Kopier tekst til register. Navnet indtastes på statuslinien når dette markeres.

Eks. rc ; Kopier tekst til register **default**

4.5.3 Flyt til register RM (Register Move)

Syntaks: rm <register name>

<register name>: default , To register:

Flyt tekstafsnit til register <register name>, og fjern afsnittet fra teksten.

Eks. rm afsnit1 ; Flyt tekstafsnit til register med navn afsnit1

4.5.4 Indsæt register i tekst RI (Register Insert)

Syntaks: ri <register name>

<register name>: default , From register:

Indholdet af registret <register name> indsættes i teksten, og markøren sættes på positionen efter den indsatte tekst.

Indholdet af registret ændres ikke.

Eks. ri%1 ; Indholdet af registret med navn angivet i variabelen 1 indsættes i teksten startende på markørens plads.

4.5.5 Slet register RD (Register Delete)

Syntaks: rd <register name>

<register name>: default , Register:

Indholdet af registret slettes, og registrets navn fjernes fra registre kendt af SE.

Bemærk at alle registre slettes, når editoren forlades.

Eks. rd garbage; Registeret garbage fjernes.

Det er nødvendigt at slette et register først, hvis der skal defineres nyt indhold i det. Dette gælder dog ikke for registeret default.

4.5.6 Vis registerindhold RL (Register List)

Syntaks: rl <register name>

<register name>: default , Register:

Indholdet af registret <register name> udskrives en skærmfuld af gangen.

Det angives på statuslinien om det er sidste del af registerteksten der vises. Angives 'Open end line !' betyder dette at teksten er uafsluttet dvs. der ikke er noget linieafslutningstegn. Indsættes dette register (med ri ordren) i teksten vil en evt. efterfølgende linie blive sammensat med indholdet af registerets sidste linie.

Eks. rl ; Vis indholdet af default registeret.

4.5.7 Vis benyttede registre RN (Register Names)

Syntaks: rn

Der udskrives en liste over benyttede registre i følgende format:

<register name> = <file name> <local>

Hvor:

<register name> Angiver navnet på registeret.

<file name> Angiver navnet på den temporære 8000 fil, der benyttes til at gemme registerets indhold.

<local> Angiver om registeret kun benyttes lokalt og dermed bliver slettet når editoren forlades (; local) eller om registret er en 'permanent' 8000 fil der også kan benyttes efter at SE er afsluttet.

4.5.8 Udskriv register i fil RW (Register Write)

Syntaks: rw <register name> <file name>

<register name>: default , From register:

<file name> : - , To file:

Registret <register name> 'udskrives' i en temporær fil med det angivne navn <file name>, hvis der ikke i forvejen eksisterer en temporær fil med dette navn.

Bemærk at et register, hvor indholdet er indlæst fra en fil (med RR ordren), ikke kan udskrives på en ny fil.

Eks. `rw default test;` Register default udskrives i filen `test`

Eks. `rw test ;` Denne ordre udskriver også registeret default i filen `test`, idet der her ikke er angivet et register navn (<file name> skal altid angives)

4.5.9 Læs til register fra fil RR (Register Read)

Syntaks: `rr <register name> <file name>`

<register name>: default , To register:
<file name> : - , From file:

Indholdet af den angivne fil <file name> indlæses til det angivne register <register name>.

Eks. `rr test;` Indlæs filen `test` til default registeret

Eks. `rr par separ, ri par, rd par;` Indsæt filen 'separ' i teksten på markørens plads og slet det benyttede register 'par' igen.

4.5.10 Kopier blok til register BC (Block Copy)

Syntaks: `bc <register name>`

<register name>: default , To register:

Den udpegede tegnblok kopieres til det angivne register <register name>.

Eks. `bc ;` Kopier til default registret

4.5.11 Flyt blok til register og slet BE (Block Erase)

Syntaks: `be <register name>`

<register name>: default , To register:

Den udpegede tegnblok flyttes til det angivne register <register name>. I teksten indsættes blanktegn i stedet for de flyttede tegn.

4.5.12 Indsæt blok i tekst BI (Block Insert)

Syntaks: bi <register name>

<register name>: default , From register:

Indholdet af det angivne register <register name> indsættes i teksten. Øverste venstre hjørne af tegnblokken i registret indsættes på markørens position. Efter indsættelse sættes markøren efter nederste højre tegn i blokken.

4.5.13 Flyt blok til register og fjern BM (Block Move)

Syntaks: bm <register name>

<register name>: default , To register:

Den udpegede tegnblok flyttes til det angivne register <register name> og blokken fjernes fra teksten.

4.5.14 Overskriv blok i tekst BO (Block Overwrite)

Syntaks: bo <register name>

<register name>: default , From register:

Indholdet af det angivne register <register name> indsættes i teksten således at eksisterende tegn overskrives. Øverste venstre hjørne af tegnblokken i registret indsættes på markørens position. Efter indsættelse sættes markøren efter nederste højre tegn i blokken.

4.6 Tabulatoren

Editorens tabulatorstop markeres med ! (udråbstegn) i en intern tabulatorlinie. Tabulatorstop positionerne kan ændres ved at ændre positionerne af tabulatorliniens !.

Tabulatorlinien sættes initialt fra parameterfilen (se afsnit 6 parameter 18).

4.6.1 Flyt til tabulatorstop til højre TR (Tabulator Right)

Syntaks: tr

Markøren flyttes til næste tabulatorstop til højre.

Er der ikke flere stop på aktuel linie, flyttes til første stop på næste linie.

Er der ikke sat nogle tabulatorstop flyttes markøren ikke.

Eks. 4(tr); Flyt 4 tabulator stop til højre

4.6.2 Flyt til tabulatorstop til venstre TL (Tabulator Left)

Syntakst: tl

Markøren flyttes til næste tabulatorstop til venstre.

Er der ikke flere stop på aktuel linie, flyttes til sidste stop på forrige linie.

Er der ikke sat nogle tabulatorstop flyttes markøren ikke.

4.6.3 Modifierer tabulatorstop TM (Tabulator Modify)

Syntaks: tm

Sætter et tabulatorstop i den interne tabulatorlinie svarende til markørens kolonne position.

Eks. ls,10(tm,8(cr)); Sæt tabulatorstop ved hver 8. tegn

4.6.4 Slet tabulatorstop TD (Tabulator Delete)

Syntaks: td

Sletter et evt. tabulatorstop i den interne tabulatorlinie svarende til markørens kolonne position.

Eks. 80(td,tr); Slet alle tabulator positioner.

4.6.5 Hent tabulatorlinie til tekst TG (Tabulator Get)

Syntaks: tg

Henter den interne tabulatorlinie ind i teksten. Den indsættes som en linie på markørens plads (den aktuelle linie flyttes en linie ned).

4.6.6 Sæt tabulatorlinie fra tekst TS (Tabulator Set)

Syntaks: ts

Den interne tabulatorlinie sættes i overensstemmelse med markeringerne på den linie markøren er placeret på. (Linien kopieres til tabulatorlinien).

4.7 Afslutning og opdatering af editering

De følgende ordrer benyttes til opdatering af resultatfilen (se afsnit 5.1 for beskrivelse af editor kald).

Resultatfilen opdateres først når en af de følgende ordrer benyttes og ikke løbende under editeringen !. Den originale editfil slettes først når al opdatering er foretaget (hvis den skal slettes). Al editering foretages i temporære arbejdsfiler (8000 work filer).

Er der ikke angivet en resultatfil ved kaldet af SE, kan der indtastes et navn på statuslinien når dette markeres. Er der angivet en editfil (input file) ved kaldet kan dette navn benyttes ved blot at taste <cr>.

Et nyt navn kan også angives som parameter til ordren.

Når SE forlades, og der gemmes i resultatfilen, angives antallet af linier der er gemt samt filens scope og navnet på det dokument, hvorpå den er gemt.

4.7.1 Forlad editor og opdater resultatfil EX (Editor eXit)

Syntaks: ex <new name>

<new name>: <result file name> , New result file name:

Udskriv den editerede tekst til resultatfilen.

Angives <new name> benyttes dette navn som nyt navn på resultatfilen ellers benyttes det navn der blev angivet ved kaldet. Blev der ikke angivet et resultatfilnavn ved kaldet og er <new name> tom skal der angives et navn til brug for gemningen. Dette gøres når der markeres herfor på statuslinien. Tastes <cr> alene benyttes editfilens navn (angivet i parentes).

Resultatfilen gemmes med scope og på dokument bestemt ud fra følgende regler:

- 1) Er resultatfilen angivet i kaldet af SE, benyttes dennes scope og dokument (hvis muligt).
- 2) Angives resultatfilen ikke i kaldet, benyttes editfilens scope og dokument (hvis muligt).
- 3) Angives ingen filer i kaldet, benyttes temporært scope og et dokument, hvor der er plads.
- 4) Kan resultatfilen ikke oprettes med det angivne scope, benyttes et lavere scope.
- 5) Kan eksisterende resultatfil (evt. samme som editfilen) ikke fjernes gemmes ny resultatfil med et arbejdsnavn (der opgives af editoren).

Eksisterer der en fil på samme baser og med samme navn som resultatfilen slettes denne, hvis det er muligt.

Eks. ex; Forlad SE og opdater resultatfilen.

Fortrydes opdateringen kan der tastes Ctrl-X inden SE afslutter (hvis typeahead er aktiv).

4.7.2 Forlad editor uden opdateringer EQ (Editor Quit)

Syntaks: eq

Forlad editoren uden at opdatere evt. resultatfil.

Eksisterede resultatfilen før kaldet, er denne uændret.

Editoren forlades umiddelbart hvis der ikke er foretaget nogen redigering eller indsættelse af tekst i editfilen (Søgninger m.m. kan foretages). Ellers skal det bekræftes at SE skal forlades uden opdatering af foretagede editeringer. Svares der **yes** eller blot **y** når det markeres på statuslinien forlades editoren, ellers genoptages editeringen.

Eks. eq;

Quit editor, are you sure ? (no)

Tastes <cr> genoptages editeringen. Tastes **y** <cr> forlades SE uden opdatering af filen (bortset fra evt. tidligere backup).

4.7.3 Forlad editor med valgt scope ET EL EU EP

Syntaks: et <new name>
 el <new name>
 eu <new name>
 ep <new name>

<new name>: <result file name> , **New result file name:**

Udskriv den editerede tekst til resultatfilen.

Resultatfilen oprettes med navn, scope og på dokument bestemt ud fra følgende regler:

Navn og dokument bestemmes ud fra samme regler som for EX. Scope bestemmes ud fra ordren og punkt 4 i afsnit 4.7.1. på følgende måde:

ET : Temporært scope.
 EL : Login scope eller lavere.
 EU : User scope eller lavere.
 EP : Project scope eller lavere.

Eksisterer der en fil på samme baser og med samme navn som resultatfilen slettes denne, hvis det er muligt.

Eks. ep test; Gem den editerede fil med project scope under navnet test.

4.7.4 Opdater fil med foretagende ændringer EB (Edit Backup)

Syntaks: eb <new name>

<new name>: <result file name> , New result file name:

Opdater resultatfilen med indtil nu foretagende ændringer i teksten (back up).

Den opdaterede resultatfil gemmes efter samme regler som ved EX ordren (afsnit 4.7.1). Opdateringen overskrives evt. når editoren forlades idet denne efter backup benyttes som ny editfil (input file).

Efter opdateringen genoptages editeringen med markøren på samme position i teksten som før. Registre, flag m.m. er uændrede efter opdateringen.

Angives <new name> vil navnet også blive brugt når editoren forlades, idet navnet benyttes som nyt resultatfil navn.

Back up bør foretages med jævne mellemrum hvis der editeres i længre tid for at undgå maskinfejl m.m.

Eks. dk 7:jt,eb;; Definer tasten med jt ordren til også at udføre backup !

4.8 Diverse andre ordrer

4.8.1 Vis hjælpe sider HE HD ? (Help English, Hjælp Dansk)

Syntaks: he
 hd ELLER ?

Viser hjælpesider fra hjælpefilen (normalt sehld eller sehle).

På statuslinien gives overskriften på næste sides hjælpe-tekst. Denne kan så vælges eller overspringes.

Vælg hjælpeside : <cr> eller y eller Ctrl-Y
Vis næste overskrift: n eller Ctrl-N
Forlad hjælp : e eller Ctrl-E (eller Ctrl-X)

Hjælpefilen er en tekstfil, der kan editeres således at egne tekster evt. kan indføres (se afsnit 8).

Ordren HD eller ? giver danske hjælpetekster, hvis filen sehld er tilstede.

Ordren HE giver engelske hjælpetekster, hvis filen sehle er tilstede o.s.v.

Eks. hd ; Viser hjælpesider fra filen sehld (hvis editoren er kaldt med navnet se).

Eks. ha ; Viser hjælpesider fra filen sehla.

4.8.2 Gå til kommando editerings tilstand CM (Command Mode)

Syntaks: cm <command>

<command>: & , COMMAND:

Normalt benyttes kun ordren cm&. Editoren overgår her til kommandoediterings tilstand og markerer dette på statuslinien med 'COMMAND:', hvorefter der kan indtastes en kommando.

Kommandoen kan også angives direkte efter cm ordren, men kommando-teksten skal da omgives af separator tegn.

Det maximale antal tegn i kommandoteksten er lig skærmens bredde (normalt 80 tegn).

Kommandoteksten kan editeres med 'rubout', 'slet tegn' eller <- tasterne, desuden kan kommandoteksten fortrydes med Ctrl-X.

Denne ordre benyttes normalt kun i parameterfilens tilde-ling af kommandoer til kontroltasterne, men den kan også benyttes på kommandolinien selv. F.eks. kan en repeterende kommando indeholde CM ordren, så brugeren kan foretage individuelle aktioner for hvert gennemløb af kommando-sekvensen.

Eks. `!(cm&,ld) ;` Udfør en ny kommando for hver linie i teksten.

Eks. `cm%1 ;` Udfør kommando i variabel 1;

Eks. `cm/!(fs:test:)/,jt;`

Er det samme som:

`!(fs/test/),jt;`

4.8.3 Udfør kommandoer i register XR (eXecute Register)

Syntaks: `xr <register name>`

`<register name>: default , Execute register:`

Indholdet af registeret `<register name>` læses en linie af gangen. Hver linie udføres som om denne blev udført i kommandoediterings tilstand (pånær XR, IT og DK ordrene).

Opstår der en fejl udskrives de normale fejlttekster sammen med en angivelse af hvilken linie i registeret der var ved at blive udført. Der kan ikke udføres xr ordre i flere niveauer !

Eks. `xr setup;` Udfør kommandoer i registeret setup.

Eks. `gp 1 2, rr setup %1, xr setup;` Udfør kommandoer der står i den fil, der angives som FP parameter nummer 2 ved kaldet af SE.

4.8.4 Definer kontrol tast DK (Define Key)

Syntaks: `dk <key number><command string>`

`<key number> : & , Define key number:`
`<command string>: empty , Enter definition:`

Tildeler en kontrol tast en kommando, således at denne kommando udføres ved aktivering af den pågældende kontrol tast.

Angives `<key number>` og `<command string>` ikke kan den ønskede kontrol tast angives når dette markeres på statuslinien. Dette kan enten være ved at taste kontrol tasten selv (evt. med foranstillet `<prefix>`) eller ved at angive kontrol tastens nummer (se afsnit 6).

Alle kontrol taster og prefix-kontrol taster kan benyttes, men vær opmærksom på at en del af disse som standard er tildelt kommandoer fra parameterfilen. Desuden kan nogle taster være reserveret til andre formål (f.eks attention `<esc>` og `Ctrl-Ø`).

Angives kontrol tastnummeret <command string> direkte i ordren skal denne stå umiddelbart efter <key number> omgivet af separator tegn.

Eks. dk25:wt/Dette er tast 25/:

Alle kontrol tast numre fra 0 til 255, der ikke er tilknyttet en tast på tastaturet (og som ikke benyttes speciel som f.eks prefix og cancel) kan også benyttes, men det er da kun muligt at udføre den tildelte kommando vha. XK (eXecute Key) ordren.

Disse taster kan evt. defineres fra egen fil indeholdende dk ordre ved hver opstart (Benyt key number -1 i parameterfilen. Se afsnit 6).

4.8.5 Vis kontrol tast definition SK (Show Key)

Syntaks: sk <key number>

<key number> : & , Show key number:

Viser den kommando, der er tildelt den angivne kontrol tast eller prefix-kontrol tast <key number>.

Angives <key number> ikke, kan den ønskede kontrol tast angives, når dette markeres på statuslinien. Dette kan enten være ved at taste kontrol tasten selv (evt. med foranstillet <prefix>) eller ved at angive kontrol tastens nummer (se afsnit 6).

Tast <cr> for at afslutte ordren.

Eks. sk 0 ; Vis definitionen af kontrol tast 0

Eks. wt/Start key /,gv1,!(sk%1,iv1); Vis kontrol tasterne
startende med værdien
indlæst til variabel
1. Afslut ved at
taste Ctrl-X.

4.8.6 Udfør kontrol tast kommando XK (eXecute Key)

Syntaks: xk <key number>

<key number> : 0 , Define key number:

Kommandoen der er defineret for kontrol tasten <key number> udføres (se 4.8.4).

Eks. xk 100 ; Udfør kommando defineret for tast nummer 100

Eks. dk 255: ss%1%2? : , %3(xk 255);

4.8.7 Vis linienumre SL (Show Linenumbers)

Syntaks: sl

Vis de aktuelle linienumre for linierne på skærmen. Linienumrene skrives i de 4 sidste kolonner på hver linie, hvorved evt. tekst i disse kolonner overskrives midlertidigt.

SL ordren benyttes til at skifte mellem de to tilstande, hvor der enten vises eller ikke vises linienumre på skærmen.

Linienumrene påvirker ikke teksten når denne gemmes.

De viste linienumre angiver antallet af linier fra starten af teksten, hvor første linie har nummer 1. En tekstlinie, der er fortsat over to linier på skærmen, tæller som to linier med hver sit linienummer.

4.8.8 Sæt flag SF (Set Flag)

Syntaks: sf <flag number>

<flag number>: 0 , Set flag number

Der sættes et flag med det angivne nummer <flag number> på markørens position. Er <flag number> = 0 sættes registermærket.

Eks. sf4; Flag 4 sat på markørens aktuelle position

Eks. sf; Mærket (lig flag 0) sættes.

4.8.9 Slet flag CF (Clear Flag)

Syntaks: cf <flag number>

<flag number>: 0 , Clear flag number

Flaget med det angivne nummer <flag number> slettes. Dette er ikke nødvendigt for at sætte et flag et andet sted !

Eks. cf; Slet flag 0, hvilket er lig mærket.

4.8.10 Kontrol information CI (Control Information)

Syntaks: ci

Ordren benyttes til at udskrive informationer om editoren og den benyttede terminal. Desuden kan kontroltasternes <key number> udskrives.

Der udskrives følgende felter:

Screen Editor: <name> <release> <test>

<name> = Navnet som SE er kaldt med.
 <release> = SE release nummer
 <test> = Angiver om editortest er sat i parameter-filen.

Process: Parent= <name>
 size = <size>
 time = <time>

<name> = Navnet på parent processen
 <size> = Antal halvord i den process der 'kører' SE
 <time> = Antal millisekunder cpu tid brugt i processen

Editor setup: Work size = <size>
 Last col = <col>
 Timeout: Rep= <rtime> Norm= <ntime>
 Temp names= <wrk1> <wrk2>
 Input = <in_file>
 Output= <out_file>
 Scope = <scope>

<size> = Antallet af linier i SE's arbejdsområde. Dette antal er bestemt ud fra processens størrelse. Det kan dog maksimalt blive på 4000 linier.

Ved en process størrelse på 100000 halvord er
 <size> = 610 linier og ved 200000 halvord er
 <size> = 1961 linier.

<col> = Det maksimale antal af tegn i en linie på skærmen.

<rtime> = Antal sekunder mellem timeout ved repeterende kommando startet med ! eller \$.

<ntime> = Antal sekunder mellem timeout ved repeterende kommando startet med antal.

<wrk1> =

<wrk2> = De to temporære filer der benyttes af SE til editeringer.

<infile> = Navnet på editfilen (input file).

<outfile> = Navnet på resultatfilen.

<scope> = Det scope der vil blive benyttet ved EX eller EB ordrene.

Terminal setup: Terminal name <name>
 Typeahead <state>
 Chars out <count>
 <setup>

<name> = Navnet på den benyttede terminal (terminal processen).
 <state> = Active angiver at typeahead er sat i parameter filen og editoren forventer at terminalen er i stand til at bufre input fra tastaturet.
 <count> = Antallet af tegn der er skrevet på terminalen under den aktuelle editering.
 <setup> = Terminalens setup (fra get specification message til terminal process).

Editor state: Display: First = <dfirst>
 Last = <dlast>
 Cursor: x = <xpos>
 y = <ypos>
 Work: First = <wfirst>
 Last = <wlast>
 Stored = <stored>
 Cur line = <line>

<dfirst> = Første linie i arbejdsområdet, der er vist på skærmen.
 <dlast> = Sidste linie i arbejdsområdet, der er vist på skærmen
 <xpos> = Markørens kolonne position.
 <ypos> = Markørens linie position i arbejdsområdet.
 <wfirst> = Første benyttede linie i arbejdsområdet.
 <wlast> = Sidste benyttede linie i arbejdsområdet.
 <stored> = Antal linier gemt i arbejdsfil.
 <line> = Markørens aktuelle absolutte linie placering i teksten.

Flags: Angiver linienummeret hvor flagene 0 til 9 er sat. Er flaget ikke sat angives - . Flag 0 er lig mærket, der benyttes ved register ordre.

Line state: Length = <length>
 Line end mark = <end>
 Cur char = <char>

<length> = Antallet af tegn i den aktuelle linie (den hvor markøren står).
 <end> = Afslutningstegnet for linien:
 10 = New line (ny linie)
 12 = Form feed (sideskift)
 0 = Continue (fortsat linie)
 <char> = Tegnet på markørens plads angivet som decimal ascii værdi.

Current line: Angiver indholdet af den aktuelle linie som decimale ascii værdier.

I nederste linie er angivet hvilken kontrolltast der benyttes som cancel (afbryd). Denne skal benyttes for at afslutte CI ordren.

Tastes en anden kontrolltast (evt. med prefix) udskrives kontrolltast nummeret på denne (se afsnit 6 for definition af kontrolltasternes numre).

4.8.11 Fjern blanktegn CB (Cut Blanks)

Syntaks: cb <position>

<position>: inline , Cut blanks After/Before/Inline:

Denne ordre benyttes til af fjerne blank tegn i teksten.

Parameteren <position> kan have følgende værdier:

- before : Alle blank tegn i tekstlinien før markørens position og/eller indtil første ikke blanke tegn slettes og markøren flyttes ned til næste tekstlinie i samme kolonne.
- after : Alle blank tegn i tekstlinien fra markørens position (inklusive denne) og/eller efter sidste ikke blanke tegn slettes og markøren flyttes ned til næste tekstlinie i samme kolonne.
- inline : Alle blank tegn efter markøren og frem til første ikke blanke tegn slettes. Markøren flyttes ikke.

Eks. cb after; Slet alle blanktegn sidst på linien efter markørens position. Flyt markøren ned til næste tekstlinie.

4.8.12 Juster skærmen JS (Justify Screen)

Syntaks: js

Den linie hvori markøren står, sættes så tæt på midten af skærmen som det er muligt. Resten af skærmen justeres i forhold hertil.

4.8.13 Skriv tekst på statuslinien WT (Write Text)

Syntaks: wt <text>

<text>: & , Write text:

Slet statuslinie og udskriv teksten <text> på denne.

Eks. wt/Prøve/ ; Udskriver 'Prøve' på statuslinien.

4.8.14 Hent værdi til variabel GV (Get Value)

Syntaks: gv <variable number>

<variable number>: 0 , Variable number:

Der indlæses en værdi til variablen angivet ved <variable number>. Markøren flyttes til statuslinien, men denne slettes ikke. Denne ordre benyttes normalt i forbindelse med ordren wt (write text).

Eks. wt/Indtast værdi/,gv2 ; Der indlæses en værdi til variablen 2 efter udskrift af 'Indtast værdi' på statuslinien.

Eks. gv%3; Indlæs værdi til variabel nummer angivet i variabel 3 !

4.8.15 Hent dato til variabel GD (Get Date)

Syntaks: gd <variable number>

<variable number>: 0 , Variable number:

Der indlæses dags dato til den angivne variable <variable number>. Værdien indeholder 8 tegn med formatet 'yy.mm.dd'

Eks. gd5 ; Indlæs dato til variabel 5.

4.8.16 Hent tidspunkt til variabel GT (Get Time)

Syntaks: gt <variable number>

<variable number>: 0 , Variable number:

Der indlæses aktuelt tidspunkt til den angivne variable <variable number>. Værdien indeholder 5 tegn med formatet 'hh.mm'

Eks. gt0,ss//%0; Indsæt aktuel tid på markørens plads.

4.8.17 Optæl værdi i variable IV (Increment Variable value)

Syntaks: iv <variable number>

<variable number>: 0 , Variable number:

Værdien i den angivne variable optælles med 1. Indeholder variablen ikke et tal (i de første tegn) benyttes værdien 0. Resultatet af optællingen sættes højrestillet i 12 tegn i variablen (Det 'gamle' indhold af variablen slettes).

Eks. gd2,iv2,wt%2; Vil udskrive ' 89' , hvis kommandoen bliver udført i 1988.

4.8.18 Nedtæl værdi i variable DV (Decrement Variable value)

Syntaks: dv <variable number>

<variable number>: 0 , Variable number:

Værdien i den angivne variable nedtælles med 1. Indeholder variabelen ikke et tal (i de første tegn) eller er værdien 0 fås resultat værdien 0. Resultatet af optællingen sættes højrestillet i 12 tegn i variabelen (Det 'gamle' indhold af variabelen slettes).

Eks. gd2,iv2,wt%2; Vil udskrive ' 87' , hvis kommandoen bliver udført i 1988.

4.8.19 Hent linie fra tekst til variabel PV (Put Variable)

Syntaks: pv <variable number>

<variable number>: 0 , Variable number:

Tegnene fra markørens aktuelle position i teksten og til enden af linien, kopieres til den angivne variable <variable number>.

Eks. pv8,iv8,ss//%8; Hent tal fra tekst, læg 1 til og indsæt det igen.

4.8.20 Hent FP parameter til variabel GP (Get Parameter)

Syntaks: gp <variable number> <fp parameter number>

<variable number> : - , Variable number:
<fp parameter number>: - , Parameter number:

Der indlæses en FP parameter fra kaldet af SE til variabelen <variable number>. <fp parameter number> angiver hvilken parameter der skal læses. Parameter 0 er første parameter (f.eks. resultat filens navn). Er der ikke angivet en FP parameter med det ønskede nummer sættes indholdet af variabelen til 0 tegn (tom) ellers fylder parameteren altid 12 tegn med parameterværdien højrestillet, hvis det er et tal og venstrestillet, hvis det er et navn.

Eks. gp 1 2; Hent FP parameter 2 til variabel 1.

Eks. gp 1 2,cr%1; Udfør de SE kommandoer, der står i den fil (commfile), der angives som parameter i følgende kald af SE:
se textfil commfile

4.8.21 Sæt linieafslutnings mærke til/fra LM (Line Marks)

Syntaks: lm

Skifter tilstanden mellem at linieafslutnings mærkerne < vises og ikke vises.

4.8.22 Opdater skæmbilledet UL US (Update Line, Update Screen)

Syntaks: ul
 us

Opdater aktuel linie og alle ændringer siden sidste opdatering (UL) eller hele det aktuelle skæmbillede (US).

Opdateringen foretages med det samme, uden at vente på at en evt. kommando er færdigudført

Normalt opdateres skærmen først når en kommando er udført eller når det er påkrævet i f.eks. SS og SN ordrøne.

Eks. !(ss/A/B/,ul); Erstat alle forekomster af A med B til
 enden af filen uden at erstatningen
 skal bekræftes. Hver erstatning vises
 på skærmen inden der fortsættes !

4.8.23 Udfør intet NO (No Operation)

Syntaks: no

Der udskrives teksten "No operation" på statuslinien.

Alle kontroltaster, der ikke defineres i parameterfilen, indeholder ordren NO ved opstart af SE.

5. KALD OG RESOURCER

5.1 Editorkald

SE kaldes online i en FP-proces.

Kaldet har følgende syntaks:

<resultatfil> = <editornavn> <editfil> <parametre>

Den fil der ønskes editeret <editfil>, angives efter editorens navn <editorname>. Angives ingen <editfil> startes med en tom tekst. Som <editfil> kan benyttes alle tekst filer der kan 'ses' fra de proces baser hvorfra SE startes.

Resultatet af en evt. editering kan gemmes i <resultatfil> enten ved backup eller når SE forlades (se afsnit 4.7). Eksisterer der en fil på samme baser i forvejen slettes denne efter opdatering af resultatfilen (inden resultatfilen 'renames' fra et work navn til resultat filens navn). Forsøges resultatet gemt i en fil, hvor der ikke er skrivetilladelse oprettes resultatfilen med et mindre scope (snævre baser).

Angives ingen <resultatfil> kan denne angives når en gemning ønskes foretaget.

Editorens aktuelle navn <editorname> afhænger af det benyttede system. Navnet benyttes bl.a. til at skelne mellem forskellige terminaltyper. Således at man i kaldet af SE angiver hvilken terminaltype, der benyttes, ved at vælge det tilsvarende editornavn.

Angives yderligere parametre efter editfilen kan disse indlæses til SE's variable vha. GP ordren (se 4.8.20).

Ved kaldet af editoren skal følgende filer være tilgængelige for den aktuelle proces:

- | | |
|--|--------------------|
| 1) Editoren selv. | <editornavn> |
| Normalt med navnet 'se'. | |
| 2) Parameterfilen. | <editornavn>+'par' |
| Dvs. editornavnet efterfulgt af 'par'. Normalt 'separ' | |
| 3) Hjælpeteksten (hvis hjælp ønskes) | <editornavn>+'hld' |
| eller hvis engelsk hjælpetekst | <editornavn>+'hle' |
| 4) SE's runtime system | 'serun' |
| 5) SE's rutine bibliotek | 'selib' |

5.2 Resourcer

Ud over de ovenfor nævnte filer, der skal være til rådighed ved kald af editoren, skal der være følgende resourcer:

- 1) Plads til 2 temporære filer på samme dokument som resultatfilen (eller editfilens dokument, hvis resultatfilen ikke eksisterer) med samme størrelse som resultatfilen.
- 2) Plads på et vilkårligt dokument til en temporær fil for hvert nyt register der oprettes.
- 3) Minimum 7 areal processer (inkl. dem til fp og serun).
- 4) Minimum proces-size på 60000 halvord. Benyttes en større 'size' kører editoren mere effektivt og arbejdsområdet bliver større.
- 5) Mindst 4 message buffere.

Startes processen som online-job under BOSS bør tiden sættes til f.eks. TIME 1 0 0 , og størrelsen af primout bør sættes stor f.eks. OUTPUT 200000.

6. PARAMETERFILEN

Til initialisering af de værdier, der i editoren bruges til styring af terminalen, front-end, lokalnet og kontroltast kommandoerne, benyttes en parameterfil. Editoren læser denne fil ved opstarten inden den udskriver opstart teksten.

Som parameterfilens navn benyttes editorens kaldenavn efterfulgt af 'par'. Hvis editoren kaldes med 'se' er parameterfilens navn derfor 'separ'.

Parameterne kan for de flestes vedkommende kommenteres direkte i filen ved at foranstille kommentaren med ;

En tegn sekvens angives i parameterlisten som decimale isoværdier adskilt af komma. Der kan indgå op til 9 tegn i en sekvens.

I det følgende er de enkelte parametre beskrevet. De første af disse skal altid indgå i parameterfilen og altid stå i den angivne linie. Ved beskrivelse af kontroltast kommandoerne er formatet mere frit idet det kun er den afsluttende linie (key number = -1) der altid skal medtages.

Linie: Parameter:

- | | |
|---|--|
| 1 | Parameter filens version, som skal stemme overens med SE's release dato. Ændres kun ved nye editor releases. |
| 2 | Tegn sekvens der flytter markøren 1 tegn til venstre. (Back space). |
| 3 | Tegn sekvens der sletter resten af den aktuelle linie. (Erase to end of line) |
| 4 | Tegn sekvens der sletter resten af skærmen (inkl. markørens position). (Erase to end of screen) |
| 5 | Tegn sekvens der sætter markøren på angivet position på skærmen. (Cursor addressing) |

Sekvensen består af de tegn, der skal sendes før, mellem og efter positions-tegnene samt disse tegn selv. Positionstegnene indsættes i sekvensen på den plads, hvor de skal sendes.

Et positiontegns decimale værdi beregnes ved:

192 + summen af de funktions værdier, der er anført ud for hver funktion i følgende tabel.

32	Kolonne position
0	Linie position
16	Adder 1 til positions værdi
8	Adder 32 til positions værdi
4	Exclusive-or positions værdi med 140 (octal)
2	Send position som et tegn (tegnværdi lig positions værdi).
0	Send position som 2 cifret decimaltal (2 ascii tegn)

Bemærk: Første kolonne og første linie på skærmen har begge positions værdien 0.

Eks. Ansi markør positionerings sekvens:

Sekvens: <esc>Æ<line>;<column>H
giver: 27, 91, 208, 59, 240, 72

Her beregnes <line> ved:
 $192 + 0 + 16 + 0 = 208$

og <column> ved:
 $192 + 32 + 16 + 0 = 240$

- 6 Tegn sekvens der sætter inverteret udskrift til. Dette benyttes bl.a. til at angive mærkets position i teksten.
- 7 Tegn sekvens der slår inverteret udskrift fra.
- 8 Tegn sekvens der sætter forøget lysstyrke (highlight) til. Dette benyttes bl.a. på statuslinien.
- 9 Tegn sekvens der slår forøget lysstyrke fra.
- 10 Værdien på den kontrol tast der skal benyttes som afbryde-tast (cancel). Dette er normalt Ctrl-X (26).
- 11 Antallet af tegn i en linie på skærmen. Det antal tegn der kan editeres på en linie vil være 1 mindre end dette tal. For nogle terminaler (bl.a. RC822) skal tallet sættes til 1 mindre end der aktuelt er positioner på skærmen for at undgå problemer med skrivning i sidste tegn på skærmen.

- 12 Antallet af linier på skærmen. Dette tal skal
 være nøjagtigt for at scroll funktionen i SE
 kan virke !

- 13 Timeout tider. Der skal angives 2 decimaltal
 adskilt af komma. Første værdi angiver timeout
 tiden for 'uendeligt' repeterende kommandoer (!
 og \$). Anden værdi angiver timeout tiden for
 kommando der gentages et fast antal gange. Ti-
 den angives i sekunder. Se desuden afsnit 3.5
 og 7.1.

- 14 Antallet af overlappende linier ved sideskift
 (PP,NP ordrene) samt ved 'rulning' ved linie op
 (ordren LU; Line Up).

- 15 Kontrol af typeahead (forud indtastning):

 0 = SE forventer ikke at terminalen har type-
 ahead.

 1 = SE forventer at terminalen har typeahead.
 Dette udnyttes bl.a. ved timeout. Idet kom-
 mandoen ikke stoppes medmindre der er
 tastet cancel (Ctrl-X).

- 16 Kontrol af test output fra SE:

 0 = Normal

 1 = Editeringen afsluttes med testoutput. Sæt-
 tes MODE LISTING.YES inden SE kaldes ud-
 skrives en oversigt over editorens resource
 forbrug m.m.

- 17 Angiver hvilken tegnkonverterings tabel (type)
 der skal benyttes. Denne værdi bør normalt kun
 sættes til 2 eller 12.

 2 = Type 2 på lokalnet (ACP, CSP m.m)
 12 = Type 2 på 3600 front end. Her sættes MSB i
 kontroltegn af SE (disse skal altså ikke
 sættes i de i parameterfilen angivne
 sekvenser eller i andre kontroltegn
 sekvenser).

- 18 Tabulator liniens initialværdi. Indeholder et !
 i de positioner der skal benyttes som tabulator
 stop.

- 19 Angiver de tegn der bruges som skilletegn i en
 tekst ved definition af ord. Alle kontroltegn
 benyttes som skilletegn. Denne linie kan ikke
 indeholde kommentar !

- 20 Værdien på den kontrol tast der skal benyttes
 som <prefix> tast. Dette er normalt HOME tasten
 (Ctrl-Å ; værdi=29).

- 21 Den kontrol værdi der benyttes som 'control introducer' ved definition af kontroltasternes tastkoder (se nedenfor). Denne værdi må ikke være benyttet som normal kontrol tast og bør ikke kunne sendes direkte fra terminalens tastatur. Normalt benyttes værdien 0 (nul).

Herefter defineres indholdet af de kontroltaster der ønskes benyttet i SE. Antallet af taster der defineres er frit. Definitionen afsluttes med at definere indholdet af 'kontrol tasten' -1. Der kan angives kommentarer mellem definitions linierne ved at indlede linien med ;

Hver linie definerer indholdet af en kontrol tast. Der benyttes følgende syntaks:

```
<key number>: <Command>
```

Hvor <key number> (er nummeret på den kontrol tast der ønskes defineret og <Command> er indholdet af tasten med samme syntaks og betydning som beskrevet i xx.x.

<key number> kan have en værdi fra -1 til 255 og der kan således defineres 256 forskellige kontroltaster og en initialiserings kommando (-1). Det vil dog normalt ikke være muligt af benytte alle disse fra tastaturet, men de resterende kan da udføres vha. ordren XK (eXecute Key).

Initialiserings kommandoen angivet i den sidste kontrol tast definition (<key number> = -1) udføres altid ved opstart af SE. Denne kan f.eks. indeholde setup af initial tilstande for linie nummervisning og liniemærke visning. Ønskes mange initial kommandoer udført kan denne definition f.eks indeholde indlæsning og udførelse af et register indeholdende kommandoerne.

Eks. -1: rr setup setupfile, xr setup, rd setup;
Udfør kommandoer i filen setupfile.

Sammenknytningen af tastaturets kontroltaster og funktions taster med <key number> foretages ved at tildele hver tast en sekvens af kontroltegn (evt. ved at programmere disse ved opstart af SE. Se nedenfor).

Den sekvens af kontrol tegn en tast skal indeholde for at en kommando med et givet <key number> bliver udført ved tryk på tasten er defineret ved:

<key number> :

- | | |
|--|---|
| 0 - 31 | Tasten skal sende et kontrol tegn med den ønskede værdi. Kontrolværdien 127 konverteres af SE til 0. |
| Værdierne der er brugt til <prefix>, cancel og <control introducer> samt værdien 0 kan ikke benyttes. Idet disse tegn evt. indleder en længre sekvens! | |
| 32 - 63 | Tasten skal sende <control introducer> tegnet efterfulgt af et kontroltegn. Dette må ikke være <control introducer> tegnet selv. Værdien beregnes som 32 + kontrol tegnets værdi. |
| Eks. 33 = <control introducer> <Ctrl-A>
32 = <control introducer> | |
| 64 - 95 | Tasten skal sende to <control introducer> tegn efterfulgt af et kontrol tegn. Værdien beregnes som 64 + kontrol tegnets værdi. |
| 96 - 127 | Tasten skal sende tre <control introducer> tegn efterfulgt af et kontrol tegn. Værdien beregnes som 96 + kontrol tegnets værdi. |
| 128 - 255 | Tasten skal sende en sekvens som beskrevet for <key number> fra 0 til 127 foranstillet med <prefix>. Værdien beregnes som 128 + tidligere beregnet værdi. |

Normalt vil disse <key number> værdier blive benyttet til kommandoer, der udføres fra tastaturet ved først at taste <prefix> og dernæst at taste den ønskede kontrol eller funktions tast.

En anden måde at definere disse <key number> værdier på, er ved at benytte det generelle forhold at hver indledende <control introducer> tegn adderer 32 til værdien, også ud over de tre som angivet for værdierne 96-127. Denne sidste metode er den bedste hvis <key number> værdien skal defineres fuldtud i kontroltasten, da markøren ellers altid vil gå til statuslinien når der modtages et <prefix> tegn.

Bemærk: Det anbefales at key number 8 altid indeholder kommandoen cl da samme tast benyttes til 'back space' på statuslinien. Skal 'slet tegn' kunne benyttes på statuslinien skal den sende værdien 127, der konverteres til key number 0 af SE ellers kan 'slet tegn' benyttes sammen med <prefix>.

Efter definitionen af initialiserings kommandoen (<key number> = -1) og læsning af den efterfølgende kommentar linie, vil resten af filen blive udskrevet på skærmen under opstarten af SE. Kontroltegn i teksten konverteres i henhold til parameteren angivet i linie 17.

Det er derfor muligt at sende en sekvens (af ubegrænset længde) af tegn til terminalen for f.eks. at programmere dens taster til at indeholde de ønskede kontrol sekvenser.

7. BEMÆRKNINGER

7.1 Timeout

Timeout benyttes på to måder i SE, afhængig af om der er typeahead eller ej.

Der udføres timeout på kommandoer, hvor der er mulighed for at de kan arbejde i lang eller uendelig tid. Der kan i parameterfilen angives en tid for både uendeligt repeterende kommandoer (! og \$) og for tællende repeterende kommandoer.

Benyttes der ikke typeahead, stoppes en kommando efter det antal CPU sekunder, der er angivet i parameterfilen (CPU sekunder er det antal sekunder processen bruger). Normalt sættes timeout til 40 - 60 sekunder.

Er typeahead sat aktiv, stoppes kommandoen kun midlertidigt, når den har brugt det givne antal CPU sekunder. Det undersøges om der er trykket en kontroltast. Hvis dette er tilfældet stoppes kommandoen ellers ventes ca. et sekund (sand tid) før der fortsættes. Dette gentages til kommandoen er færdigudført eller der trykkes på cancel tasten (Ctrl-X). Her sættes timeout-tiden normalt til 0.5 - 5 sekunder.

Er typeahead sat uden at terminalen kan benytte dette, kan der kun tages Ctrl-X inden for det sekund der ventes ved hver time out.

7.2 Arbejds området

Den aktuelle tekst på skærmen samt noget af den omgivne tekst opbevares internt i SE i et såkaldt arbejdsområde (work area).

Dette område flyttes af SE ned gennem den fil, der editeres, således at den aktuelle tekst altid er placeret i området.

Markøren kan frit placeres i området, men forsøges den af SE placeret uden for dette, må området flyttes. Da det kun kan flyttes ned gennem filen, (dette skyldes at der ikke kan læses et vilkårligt sted i filen, men kun fremad) kan markøren ikke umiddelbart flyttes op foran det aktuelle område.

Ordre der skal flytte op over områdets første linie vil derfor kræve en reorganisering af teksten. Dette foretages automatisk af SE og markeres på statuslinien med 'Reorganizing'.

Den aktuelle placering og størrelse af arbejdsområdet kan, hvis det ønskes, aflæses vha. ci ordren.

Arbejdsområdets størrelse har også betydning for hvor mange linier der kan flyttes til et register (med ordrene RC, RM, BC, BM, BE) idet mærket, der benyttes til at udpege tekstområdet, skal stå inden for arbejds området når ordren udføres.

7.3 Attention, afbrydelse og fejl

Bliver SE afbrudt ved at der trykkes attention (Esc) kan situationen klares ved at taste på pil-ned tasten. Derefter skal skærbilledet evt. reetableres vha. US ordren.

Bliver terminalens forbindelse (link) fjernet, mens man befinder sig i SE, og har typeahead aktiv, vil det ikke altid være muligt at få kontakt med terminalen igen, da det ikke er muligt at give attention, uden at der er en input message til terminalen. For at kunne benytte terminalen igen skal attention enables for den pågældende terminal (dette sker normalt automatisk !). Før SE benyttes bør man derfor undersøge hvilket link terminalen er tilknyttet og hvilket navn terminalen har (se CI ordren), da disse oplysninger evt. skal bruges ved reetablering af terminalens status.

Bliver SE afbrudt pga. fejl eller lignende uden at forbindelsen til terminalen fjernes, kan SE blot kaldes igen for at reetablere terminalen (Prompt sættes lig BEL, Timeout sættes til 60 sek.).

Normalt vil <CR>-tasten ikke virke, hvis SE afbrydes, der skal i stedet benyttes en tast der sender 'nl' (f.eks pil-ned eller Ctrl-J).

Bemærk: Fjern aldrig terminalens link for at komme ud af SE.

Benyttes SE under TAS kan der uden problemer skiftes til anden session. Skærmen vil da normalt også blive re-etableret ved returnering til den afbrudte session ! (Hvis dette ikke er tilfældet, benyttes US ordren)

8.0 Hjælpefilens opbygning;

Hjælpefilen er opbygget som en række skærbilleder der kan vælges et af gangen.

Et skærbillede består af en valglinie og selve hjælpepeteksten. Valglinien udskrives på statuslinien inden den tilhørende hjælpepetekst evt. bliver udskrevet. Efter udskrift af valglinien ventes på indtastning fra brugeren.

Tastes der <cr> eller y <cr> eller Ctrl-Y udskrives hjælpepeteksten og den næste valglinie (hørende til den næste hjælpepetekst).

Tastes der n <cr> eller Ctrl-N udskrives hjælpepeteksten ikke men næste valglinie udskrives.

Tastes e <cr> eller Ctrl-E returneres fra hjælpe ordren.

En valglinie består af den tekst der skal udskrives på statuslinie (evt. med attributter) afsluttet med et ? .

Al tekst efter valglinien og indtil der mødes et % udgør hjælpepeteksten. Denne tekst udskrives, hvis det vælges ved udskriften af den forrige valglinie.

Følgende attributter kan benyttes i hjælpefilens tekst:

```
$0 : Anuller alle satte attributter.
$1 : Forøget lysstyrke (highlight)
$2 : Inverteret skrift
$3 : Slet skærm
$% : % tegn i hjælpepeteksten
$? : ? i valglinien (afsluttende ? udskrives altid)
$$ : $ tegn i teksten
```

Benyttes attributten \$3 (slet skærm) først i en valglinie udskrives denne øverst på skærmen og den kan da bestå af flere linier (evt en hel side).

Bemærk: Første linie(r) i hjælpefilen skal indeholde valglinien til første hjælpeside !.

Sidste linie i filen skal indeholde en valglinie, der angiver at der ikke er flere hjælpesider !

Navnet på hjælpefilen dannes ved:

<editor navn>+hl+<andet tegn i hjælpe ordren>

Hjælpe ordren er beskrevet i afsnit 4.8.1.

A. SKÆRMKOMMANDOER

I dette bilag er angivet de kontroltaster, der benyttes som standard ved skærm editering samt nogle forslag til brug for Partner ACP og RC45 terminaler. Normalt vil SE være tilpasset den aktuelle terminal vha. parameterfilen.

Parameter filens navn er givet som:

```
separ      : RC822 og RC855-tty
psepar     : Partner ACP terminal emulator
se45par    : RC45 ansi terminal
```

A.1 RC822 og RC855-tty standard

Bemærk at Ctrl-X her benyttes som pil til højre og ikke som cancel tast !! I stedet benyttes Ctrl-T.

Funktion:	Kontroltast:	RC822	RC855
Forlad editoren og opdater resultatfil.	<prefix> Ctrl-E		
Forlad editoren <u>uden</u> opdatering.	<prefix> Ctrl-Q		
Opdater resultat og genoptag editering.	<prefix> Ctrl-B		

Tegn til højre.	Ctrl-X		
Tegn til venstre.	Ctrl-H		
Linie ned.	Ctrl-J		
Linie op.	Ctrl-Z		
Linie slutning.	<prefix> Ctrl-X	HOME	
Linie start.	<prefix> Ctrl-H	HOME	
Næste linie.	Ctrl-M	Return	

Skærm bund.	<prefix> Ctrl-J	HOME
Skærm top.	<prefix> Ctrl-Z	HOME
Tekst slut.	<prefix> Ctrl-N	HOME print
Tekst start.	<prefix> Ctrl-P	HOME print-off
Næste skærmside.	Ctrl-N	print
Forrige skærmside.	Ctrl-P	print-off

Ord til højre.	Ctrl-R
Ord til venstre.	Ctrl-W
15 tegn til højre.	<prefix> Ctrl-R
15 tegn til venstre.	<prefix> Ctrl-W

Indsæt blanktegn.	Ctrl-B	
Indsæt linie mærke.	<prefix> Ctrl-M	HOME Return
Indsæt ny linie efter denne linie.	Ctrl-L	
Indsæt ny side efter denne linie.	<prefix> Ctrl-L	
Indsæt tekst (til/fra).	Ctrl-A	

Slet tegn på markørs plads.	Ctrl-@	Rubout	Slet tegn
Slet tegn før markør.	<prefix> Ctrl-@	HOME Rubout	Slet tegn
Slet ord.	<prefix> Ctrl-D		
Slet linie.	Ctrl-D		

Find tekst.	Ctrl-F
Find tekst igen.	<prefix> Ctrl-F
Find mærke.	<prefix> Ctrl-Y
Find og erstat tekst.	Ctrl-S
Find og erstat næste.	<prefix> Ctrl-S

Sæt mærke i tekst.	Ctrl-Y
Kopier til default register.	Ctrl-C
Flyt til default register.	<prefix> Ctrl-C
Indsæt default register i teksten.	Ctrl-V
Vis indhold af default register.	<prefix> Ctrl-V

Tabuler til højre	Ctrl-I	TAB
Tabuler til venstre	Ctrl-E	
Sæt et tabulatorstop.	<prefix> Ctrl-I	HOME TAB

Opdater skærbillede.	Ctrl-U
Indrykning til/fra.	<prefix> Ctrl-A

A.2 Partner ACP ANSI emulator

Function keys for the Partner with ACP ANSI emulator

'Shift' is the SHIFT key together with the function key
 'Ctrl' is the Ctrl key together with the function key
 'Alt' is the Alt key together with the function key
 'Pref' is the Prefix key (home key) and then the function key

KEY:**Funktion:**

F1	Find string
F2	Substitute string
F3	Exit with default scope
F4	Previous Page
F5	Next Page
F6	Current line to bottom of screen
F7	Delete current line
F8	Copy marked text to default register
F9	Insert text from default register
F10	User definable key 1
F11	Set a flag in text (0 to 9)
F12	Set the text mark flag (flag 0)
PRINT	Insert text toggle
'Pref' PRINT	Indent toggle
TEGN IND	Insert space
'pref' TEGN IND	Insert a control character
<cr>	Move cursor to next line
'pref' <cr>	Insert new line character
!<--	Word left
-->!	Word right
'pref' !<--	Tabulator left
'pref' -->!	Tabulator right
<-	Character left
->	Character right
ü	Line up
V	Line down
'pref' <-	Set cursor at start of line
'pref' ->	Set cursor after last character in line
'pref' ü	Set cursor at top of screen
'pref' V	Set cursor at bottom of screen
'pref' 'pref'	Command mode
<<-	Character left
'Ctrl' <<-	Rub out character

KEY:**Funktion:**

'Shift' F1	Find next
'Shift' F2	Substitute next
'Shift' F3	Quit editor with no updata of file
'Shift' F4	Jump to top of file
'Shift' F5	Jump to bottom of file
'Shift' F6	Move current line to the screen top
'Shift' F7	Delte to end of line from current cursor position (inc.)
'Shift' F8	Move the marked text to the default register and delete from text
'Shift' F9	Insert the default register contents as a block
'Shift' F10	User definable key 2
'Shift' F11	Jump to flag in text
'Shift' F12	Define a command in a function key (normaly the user definable keys or unused key numbers)
'Ctrl' F1	Find Page mark
'Ctrl' F2	Show linemarks on screen (toggle)
'Ctrl' F3	Backup current edit file
'Ctrl' F4	Update screen (rewrite)
'Ctrl' F5	Set tab. line from text
'Ctrl' F6	Move current line to the screen midt
'Ctrl' F7	List contents of the default register
'Ctrl' F8	Copy the marked text block to the default register
'Ctrl' F9	Insert the default register contents and overwrite current text
'Ctrl' F10	User definable key 3
'Ctrl' F11	Move to bottom of internal Window
'Ctrl' F12	Show content of function key
'Alt' F1	Find control character
'Alt' F2	Show linenumbers on screen (toggle)
'Alt' F3	Exit with temp scope
'Alt' F4	Update current line
'Alt' F5	Get curret tab. line to text (insert)
'Alt' F6	Move cursor to the center af the line
'Alt' F7	Show names of registers in user
'Alt' F8	Move the marked text block to the default register and delete from text
'Alt' F9	Move the marked text block to the defaut register and insert spaces in text
'Alt' F10	User definable key 4
'Alt' F11	Move to top of internal Window
'Alt' F12	Execute a function key

KEY:

'Pref' F1
'Pref' F2
'Pref' F3
'Pref' F4
'Pref' F5
'Pref' F6
'Pref' F7
'Pref' F8
'Pref' F9
'Pref' F10
'Pref' F11
'Pref' F12

Funktion:

Move cursor to start af last find (or substitute) command
Cut blancks (spaces) at then end of current line and move cursor to next text line
Exit with project scope
Show editor control information, cancel key value and function key numbers
Set a stop in tab. line at current cursor position.

Execute commands in a register
Write a register to a RC8000 file
Read a RC8000 file to a register
User definable key 5
Jump to a line in the text
Find then text mark

Ctrlr-X

Cancel or interrupt a command !

pref :	Indt	startF	cut-B	QUIT	ci	modtab	reg-Ex	reg-W	reg-R	user5	jumpL	FindM	Ichar
alt :		F-char	lineN	temp	Uline	gettab	reg-N	blk-M	blk-E	user4	topW	key-Ex	
ctrl :		F-page	lineM	backup	Upscr	settab	reg-L	blk-C	blk-O	user3	endW	key-Show	
shift:		F-next	S-next	proj	topF	endF	Del-L	reg-M	blk-I	user2	jumpF	key-Def	
Insert		FIND	SUB	SAVE	PREV	NEXT	Del-LE	reg-C	reg-I	user1	set-F	set-M	Iblank

L-start L-end P-top P-end Command

Rub-out

back

Del-W

Del-C

A.3 RC45 ansi key layout

```

-----
Find-N Sub-N Last-S Page-F Flag-J INDENT Mark-F Reg-M Blk-M Blk-E Blk-I Reg-N
FIND SUB Last-F Char-F Flag-S INSERT Mark-S Reg-C Blk-C Blk-O Reg-I Reg-L
-----

```

```

-----
QUIT Scr-M Scr-T Line-M Del-W Show-K Reg-E user1 user4 user6 user8
EXIT Update Scr-E <---- W ----> Blank-I Def-K Key-E user1 user3 user5 user7
-----

```

B. SE order oversight

Name:	Parameters:	Function:	Insert mode:
-------	-------------	-----------	--------------

BC	<reg>	Block Copy (from text)	No
BE	<reg>	Block Erase (from text)	No
BI	<reg>	Block Insert (to text)	No
BM	<reg>	Block Move (from text)	No
BO	<reg>	Block Overwrite (to text)	No
CB	<B/I/A>	Cut Blanks on line	Yes
CF	<flag number>	Clear Flag	Yes
CI		Control Information	Yes
CL		Character Left	Yes
CM	<command>	Command Mode	Yes
CR		Character Right	Yes
DC		Delete Character	Yes
DK	<key number><string>	Define Key	Yes
DL		Delete Line	No
DV	<var number>	Decrement Variable	Yes
DW		Delete Word	Yes
EB	<file>	Edit Back up	No
EL	<file>	Exit Login scope	No
EP	<file>	Exit Project scope	No
EQ		Exit Quit	No
EU	<file>	Exit User scope	No
ET	<file>	Exit Temp. scope	No
EX	<file>	EXit default scope	No
FC	<iso value>	Find Character	No
FL		Find new Line mark	No
FM		Find Mark	No
FN		Find Next string	No
FP		Find Page mark	No
FS	<string>	Find String	No
GD	<var number>	Get Date to variable	Yes
GP	<var number> <param>	Get Parameter to variable	Yes
GT	<var number>	Get Time to variable	Yes
GV	<var number>	Get value to variable	Yes

Name: Parameters: Function: Insert mode:

HD		Help Danish	Yes
HE		Help English	Yes
IB		Insert Blank	Yes
IC	<iso value>	Insert Character	Yes
ID		InDent on/off	Yes
IL		Insert Linemark (nl)	No
IP		Insert Pagemark (ff)	No
IT		Insert Text mode on/off	No
IV		Increment Variable	Yes
JB		Jump Bottom	No
JF	<flag number>	Jump Flag	No
JL	<line number>	Jump Line	No
JS		Justify Screen	Yes
JT		Jump Top	No
LD		Line Down	Yes
LE		Line End	Yes
LM		new Line Mark on/off	Yes
LU		Line Up	No
LS		Line Start	Yes
NL		New Line	Yes
NO		No Operation	Yes
NP		Next Page	No
PP		Previous Page	No
PV	<var number>	Put Variable (from text)	Yes

Name: Parameters: Function: Insert mode:

RC	<reg>	Register Copy (from text)	No
RD	<reg>	Register Delete	No
RI	<reg>	Register Insert (to text)	No
RL	<reg>	Register List	No
RM	<reg>	Register Move (from text)	No
RN		Register Names	No
RR	<reg> <file>	Register Read (from file)	No
RW	<reg> <file>	Register Write (to file)	No
SB		Screen Bottom	Yes
SD		Skip line Down	No
SF	<flag number>	Set Flag	Yes
SK	<key number>	Show Key	Yes
SL		Show Linenumbers	Yes
SM		Set Mark	Yes
SN	<confirm>	Substitute Next	No
SS	<string><string><confirm>	Substitute String	No
ST		Screen Top	No
TD		Tabulator Delete	Yes
TG		Tabulator Get (to text)	No
TL		Tabulator Left	Yes
TM		Tabulator Modify	Yes
TR		Tabulator Right	Yes
TS		Tabulator Set (from line)	Yes
UL		Update Line	No
US		Update Screen	No
WL		Word Left	Yes
WR		Word Right	Yes
WT	<string>	Write Text	Yes
XK	<key number>	eXecute Key	Yes
XR	<reg>	eXecute commands in Reg.	Yes

C. ISO-TABEL FOR TEGN BRUGT I SE

0	1	2	3	4	5	6	7	8	9
0: Ctr-@	Ctr-A	Ctr-B	Ctr-C	Ctr-D	Ctr-E	Ctr-F	Ctr-G	Ctr-H	Ctr-I
:									
10: Ctr-J	Ctr-K	Ctr-L	Ctr-M	Ctr-N	Ctr-O	Ctr-P	Ctr-Q	Ctr-R	Ctr-S
:									
20: Ctr-T	Ctr-U	Ctr-V	Ctr-W	Ctr-X	Ctr-Y	Ctr-Z	Ctr-Æ	Ctr-Ø	Ctr-Å
:									
30: Ctr-Ü	Ctr-__	SP	!	"	§	\$	%	&	'
:									
40: (	)	*	+	,	-	.	/	0	1
:									
50: 2	3	4	5	6	7	8	9	:	;
:									
60: <	=	>	?	@	A	B	C	D	E
:									
70: F	G	H	I	J	K	L	M	N	O
:									
80: P	Q	R	S	T	U	V	W	X	Y
:									
90: Z	Æ	Ø	Å	Ü	_	`	a	b	c
:									
100: d	e	f	g	h	i	j	k	l	m
:									
110: n	o	p	q	r	s	t	u	v	w
:									
120: x	y	z	æ	ø	å	ü	Ctr-@		

Følgende tegn benyttes specielt af SE i teksterne:

Ctrl-J = New line = 'nl' = <
 Ctrl-L = Form Feed = 'ff' = §
 Ctrl-M = Return = <cr> = <

Alle andre kontroltegn udskrives som @ i teksten.

D. FEJLMEDDELELSER

I det følgende er de meddelelser SE udskriver angivet. Først er meddelelserne der kan udskrives ved opstart beskrevet. Dernæst beskrives de meddelelser der udskrives under editering.

D.1 Opstart meddelelser

- ***se Area claims exceeded**
Procesen har for få area-claims
- ***se Buffer claims too small**
Procesen har for få buffer-claims
- ***se Illegal converttable number <number>**
Der er angivet et illegalt tabelnummer i parameterfilen (tilladte tabeller er 1 til 19, normalt benyttes kun tabel 2 eller 12)
- ***se Illegal key definition: <key>**
Der er angivet et ulovligt kontrol tastnummer ved definition af kontroltasterne. Nummeret skal ligge mellem 0 og 255 !
- ***se Illegal process bases**
SE kan ikke oprette arbejdsfiler med de givne baser.
- ***se Incorrect parameterfile <name>**
Filen med navnet <name> indeholder ikke parametre til editoren.
- ***se Input file protected**
Filen, der skal editeres, er reserveret af anden proces eller den ligger på 'beskyttede' baser.
- ***se Internal or hard error: <type> <result>**
Intern fejl i SE. Kan opstå ved bl.a. hardwarefejl og softwarefejl i terminal og front-end samt ved diskfejl. Fejlen bør meddeles til systemansvarlig.
- ***se No or empty parameterfile: <name>**
Parameterfilen <name> findes ikke eller indeholder ikke parametre til SE.
- ***se Not enough temporary resources on <document>**
Der er ikke nok plads på <document> til SE's arbejdsfiler m.m. (se afsnit 5.2)
- ***se Not a file: <parameter>**
Parameteren angivet ved kaldet af editoren er ikke et filnavn.
- ***se Not a text file: <name>**
Den angivne editeringsfil er ikke en tekstfil (content key <> 0).

- ***se Process too small. Min. size = <min size>
Editoren forsøges startet i en for lille proces. <min size> angiver minimum antal halvord i proces.
- ***se Terminal not online
Editoren er kaldt som batch job. Dvs. c og v peger ikke på den samme terminalproces.
- ***se Unknown file: <name>
Den angivne fil findes ikke.
- ***se Wrong version of parameterfile <name>
Den fundne parameterfil er ikke samme version som editoren.

D.2 Editerings meddelelser og fejl

Already executing register

Ordren xr bliver benyttet ved udførelse af kommandoer i register. Dette er ikke tilladt.

Attention on input

Der var attention under indtastning fra terminal. Editeringen genoptages.

Backup file is removed

Editeringen er afsluttet og en tidligere backup fil er slettet, idet den nye fil er gemt på samme baser som backup filen.

Backup OK

Editoren er blevet afbrudt pga. fejl under backup af fil, men backup operationen blev udført og filen er i orden.

Block too big

Det udpegede tekstafsnit er for stor til at kunne flyttes til et register, idet editorens arbejdsområde ikke kan indeholde det antal linier afsnittet fylder. (se afsnit 7.2)

Cannot create file

Det er ikke muligt af oprette den ønskede fil måske p.g.a. resourcemangel.

Cannot make file login

Det er ikke muligt at give den temporære fil scope login.

Cannot make file permanent

Det er ikke muligt at give den temporære fil et permanent scope (user, project eller system).

Cannot remove old file, new file is called: <name>

Det er ikke muligt at slette den oprindelige fil. Den gemte fil får navnet <name>.

Changes after backup not saved in file

Rettelser der er foretages efter sidste backup af filen er ikke gemt.

Changes not saved in file

Rettelser der er foretaget i teksten er ikke blevet gemt.

Character not found

Det søgte tegn blev ikke fundet.

Command timeout

Kommandoen har brugt det antal CPU sekunder, der er sat for timeout (forekommer kun hvis der ikke er typeahead)

Command level too deep

Kommandoer og/eller kommandosekvenser er angivet i for mange niveauer. (Max. 8)

Command line too long

Kommandoen fylder for mange tegn.

Entry protected, file saved with lower scope

Resultatet forsøges skrevet til en fil, hvor der ikke er tilladelse til at skrive. Resultatet gemmes da med et lavere scope (normalt user).

File does not exists

Den søgte fil findes ikke.

File update aborted

Gemning af filen blev afbrudt og editeringen genoptages. Resultat filens navn er blevet sat til angivende navn og det vil blive brugt ved fornyet gemning !

Flag not set

Det angivne flag er ikke blevet sat.

Illegal character

Der blev angivet en ulovlig iso-værdi for et tegn. Der skal gælde: 1 <= iso-værdi <= 127.

Illegal block start

Første position i en kolonne orienteret register ordre er angivet udover første linie i en tekstlinie (linie der fylder mere end en linie på skærmen).

Illegal block marking

Sidste position i en kolonne orienteret register ordre er angivet udover første linie i en tekstlinie (linie der fylder mere end en linie på skærmen).

Illegal filename, new file is called:

Der forsøges gemt en fil med et navn, der ikke kan benyttes. Filen gemmes med det angivne work navn.

Illegal flag

Der er angivet et ikke tilladt flag. Flag kan have værdier mellem 0 og 9 (dog 0 til 11 ved hop til disse).

Illegal key

Den benyttede tast skal være en kontroltast. Kan også forekomme hvis en kontroltast, der kun indeholder 'control introducer' værdien (se afsnit 6), testes inden en 'normal' kontroltast.

Illegal key number

Der er angivet et tal uden for intervallet 0 til 255.

Illegal name or parameter

Der er angivet et navn eller en parameter, der ikke opfylder kravne for RC8000 navne.

Illegal parameter number

Nummeret på den ønskede FP parameter skal angives som et positivt tal (eller 0).

Illegal repeat count

Tallet foran en parentes er illegalt. Der kan kun angives ! & eller en tal mindre end 1000000.

Illegal variable number

Der er angivet en ikke tilladt variabel. Variable har nummere mellem 0 og 9.

Interrupted

Kommando afbrudt efter tryk på Cancel tasten. Tastes Return fortsættes !

Key can not be defined

Cancel tasten eller prefix tasten er forsøgt tildelt en kommando. Dette er ikke muligt.

Line too long

Den aktuelle tekstlinie er blevet for lang til at ordren under udførelse kan benytte denne. Check linien for korrekt opdatering !

No changes after backup of file

Backup filen indeholder alle rettelser, der er foretaget.

No changes to file

Der er ikke udført nogle rettelser i filen.

No mark set

Mærket er ikke sat.

No more registers

Der er ikke flere registre til rådighed.

Not a positiv number

Det angivne tal er negativt. Editoren kan kun benytte positive tal.

Not a textfile

Der er forsøgt indlæsning af en fil der ikke indeholder tekst.

Only a simple command allowed

Der kan kun angives en enkelt ordre her. Ønskes en kommando bestående af flere ordre udført skal kommando editerings tilstanden benyttes. Denne fås ved <prefix><prefix>.

Overflow on input

Der er tastet for mange tegn forud ved typeahead. Alt indtastet bliver ignoreret !

Page mark not found

Side markering 'ff' blev ikke fundet

Register exists

Det forsøges at oprette et register med et navn, der er i brug i forvejen.

Register is already a file

Det er ikke muligt at udskrive et register på en fil, hvis dette register tidligere er indlæst fra en fil.

Repeat count to big

Der er angivet et tal der er for stort for en repeterende kommando. Der kan max. angives 999999.

Sorry. Help file not available

Den ønskede hjælpefil er ikke tilstede. Prøv HD eller HE.

String not found

Søgeteksten blev ikke fundet.

Text end

Slutningen af teksten blev nået. Det er ikke muligt at udfører ordren ud over sidste linie i teksten.

Unknown command <name>

Den angivne ordre eller kommando kan ikke genkendes evt. pga. forkert syntaks eller fordi parameteren fra en tidligere ordre, der ikke kunne udføres bliver fortolket som en ny ordre.

Unknown register

Det angivne register kendes ikke.

E. EKSEMPLER PÅ BRUG AF SE

I det følgende gennemgås nogle eksempler på brugen af de grundlæggende funktioner i SE. De kontroltaster der benyttes i eksemplerne er angivet for RC822 el. RC855-tty terminaler.

Brugerens indtastninger er markeret med fed skrift.

Eksempel 1. Indtastning af en ny tekst

SE kaldes online (dvs. i/o direkte fra/til terminalen) fra FP.

Ved oprettelse af en ny tekstfil har FP-kommandoen følgende udseende:

eks1 = se

eks1 er navnet på den fil, hvor den indtastede tekst gemmes. se er navnet på editoren.

Efter kaldet udskriver SE først en indledende 'sign on' tekst, denne slettes efter et kort øjeblik og markøren står nu i øverste venstre hjørne af en tom skærm.

Den ønskede tekst kan nu indtastes en linie af gangen, hvor hver linie afsluttes ved tryk på return-tasten.

Tastes der galt, kan fejlen rettes ved pil-venstre tasten (<- , flytter markøren til venstre) og 'slet tegn' tasten (sletter tegnet på markørens plads).

Der indtastes f.eks. følgende algol procedure:

```

procedure transpose (a) order:(n);
array a;
integer n;
begin
  integer i,k;
  for i:=1 step 1 until n do
    for k:=1+i step 1 until n do
      begin
        a(k,i):=w
      end
    end
  end transpose

```

Efter at have indtastet teksten skal den gemmes i filen eks1 som angivet i kaldet. Dette gøres ved at trykke på <prefix>-tasten (HOME el. /<- afhængig af terminaltypen) og dernæst på Ctrl-E (tasten 'CTRL' samtidig med E-tasten). Dette vil bevirke at SE skriver 'Saving eks1' på nederste linie på skærmen (kaldet statuslinien) og derefter afslutter kørslen af editoren.

Eksempel 2. Rettelser i en tekst

Den i eksempel 1 indtastede tekst 'eks1' benyttes igen i dette eksempel.

SE kaldes nu med:

eks2 = se eks1

Efter indledningen står markøren på første tegn i den indlæste tekst (fra filen eks1).

Det første der skal rettes er de 2 et-taller efter 'step'. Dette gøres her ved først at flytte markøren ned til det første et-tal.

Tryk 5 gange på pil-ned tasten (v) og derefter 16 gange på pil-højre tasten (->), markøren flyttes på denne måde 5 linier ned og 16 tegn mod højre. Markøren står nu på et-tallet og det kan rettes ved blot at taste det nye tal f.eks. 2 .

Den type taster der blev benyttes til at flytte markøren betegnes kontroltaster. Disse kan enten som her være en enkelt tast eller det kan være en tastkombination som benyttes i eksempel 1 til at gemme teksten (Ctrl-E).

Det næste et-tal rettes ved at flytte ned til det v.h.a. pil-ned (v) og 2 gange pil højre (->), og dernæst overskrive det med et 2 .

Efter første linie skal der nu indsættes en linie, til beskrivelse af parametren n. Dette gøres ved først at flytte markøren til toppen af teksten v.h.a. <prefix> og Ctrl-P tasterne (først <prefix>-tasten dernæst Ctrl-P). Markøren står nu igen på første tegn i teksten. Trykkes Ctrl-L indsættes en ny linie efter den aktuelle og markøren flyttes ned på denne (bemærk at den første linie forsvinder udenfor skærmen). Der kan derefter indtastes den ønskede tekst: **value n; .**

For at indsætte en manglende erklæring af variablen w skal det første 'begin' i teksten findes. For at gøre dette trykkes Ctrl-F. SE skriver 'Find' på statuslinien og sætter markøren lige efter. Der kan nu indtastes de tegn der skal søges efter. Her indtastes **begin** og der afsluttes med return-tasten. SE vil nu søge efter 'begin' og når tegnene findes sættes markøren på det første af de søgte tegn.

Den nye erklæring skal indsættes foran integer-erklæringen i næste linie. Først flyttes markøren til starten af 'integer' (F.eks. tryk `return` , `->` , `->`), dernæst skal der indsættes blanke foran 'integer'. Et blank-tegn kan indsættes med `Ctrl-B` og den ønskede tomme plads kunne derfor fås ved at trykke det fornødne antal gange på `Ctrl-B`. Dette er dog for besværligt idet antallet af gange en kontroltasts kommando (den funktion den er til-delt) skal udføres kan angives v.h.a. `<prefix>`-tasten. Tasten nemlig `<prefix>` efterfulgt af 7 (skrives på statuslinien) og trykkes dernæst på `Ctrl-B`, indsættes 7 blanktegn. Der er nu plads til at indtaste erklæringen: `real w, .`

Efter det næste 'begin' mangler der 2 linier, disse skal nu indtastes. Først findes det næste 'begin'. Dette kan gøres v.h.a. `<prefix> Ctrl-F` (Her udføres 'Find næste', d.v.s. søg efter det samme som ved sidste Find).

Efter at 'begin' er fundet, flyttes ned på næste line v.h.a. `return`. Markøren står nu først på den linie, hvor de nye linier skal indsættes foran. Tasten nu `Ctrl-A` forsvinder teksten efter markøren og SE skriver `INSERTING` på statuslinien. De nye linier kan nu indtastes:

```
w:=a(i,k);
a(i,k):=a(k,i);
```

Efter at linierne er indtastet trykkes igen `Ctrl-A`. Hvorefter den tekst, der før det første `Ctrl-A` stod efter markøren, skrives efter de nu indsatte linier, og `INSERTING` slettes fra statuslinien.

Den rettede tekst skal nu gemmes i filen `eks2` med scope sat til `user` (hvis det er muligt).

Der trykkes først på `<prefix>`. Markøren står nu på statuslinien. Her kan indtasten en ordre til SE (til forskel fra 7-tallet før). Tasten `eu` (ordren Exit User) efterfulgt af `<prefix>` udføres denne ordre. Teksten gemmes med `user-scope` under navnet `eks2`.

Hvis den netop rettede tekst ikke ønskes gemt, tastes `<prefix> Ctrl-Q` i stedet for `eu` ordren. SE spørger så om man nu også er sikker på at rettelserne skal kaseres 'Quit, are you sure (no)'. Hertil svares `yes` (eller `y`), hvorefter SE fortæller at evt. ændringer ikke er gemt. Tasten blot `return` returneres til SE og editeringen kan genoptages (`(no)` angiver default).

Eksempel 3. Gentagelser

I eksempel 2 blev det benyttet at en kontrolltast (Ctrl-B tasten) kunne gentages et antal gange ved at angive tallet i stedet for at trykke på kontrolltasten, det ønskede antal gange.

Denne mulighed for at gentage udførelsen af en kontrolltast et antal gange kan benyttes for alle kontrol-taster (og <prefix> tast kombinationer).
 Tastes <prefix> 25 'pil ned'-tasten udføres pil-ned 25 gange, d.v.s, markøren flytter 25 linier ned.

Eksempel 4. Ordre

Ud over at kunne udføre editeringer v.h.a. kontrol-tasterne kan editeringen af en tekst i SE også udføres ved benyttelse af ordre fra SE's ordresæt (alle ordre er gennemgået i afsnit 4).

I eksempel 2 blev der benyttet en ordre til at gemme teksten med et bestemt scope (eu ordren). Dette blev her gjort fordi eu ordren ikke umiddelbart kan udføres v.h.a. kontrolltasterne (der er ikke defineret et tast til denne ordre for den aktuelle terminal).

En ordre består af 2 bogstaver evt. efterfulgt af nogle parametre.

Når en ordre ønskes udført trykkes først på <prefix>, dernæst indtastes ordren (på statuslinien) og indtastningen afsluttes med at trykke Return.

Tastes <prefix> ld <cr> (ld er ordren Line Down; Linie ned) svare det nøjagtigt til at trykke på pil-ned tasten.

I eksempel 2 blev der foretaget en søgning efter tegnene 'begin' v.h.a. Ctrl-F tasten. Denne søgning kunne også være foretaget ved benyttelse af ordren fs (Find String) på følgende måde:

<prefix> fs/begin/ <cr>.

Her er 'begin' (omgivet af separatortegnet /) angivet som en parameter til fs-ordren.

Eksempel 5. Kommandoer

Ønskes flere ordre udført lige efter hinanden evt. med gentagelser kan dette gøres ved at skive en kommando til SE.

Tastes <prefix> <prefix> udskriver SE 'COMMAND:' på statuslinien og det er nu muligt at angive den kommando, der ønskes udført af SE.

Står markøren efter COMMAND: på statuslinien kan følgende kommando indtastes: 25(ld) . Når kommandoen udføres, efter at indtastningen af den er afsluttet med return-tasten, flyttes markøren 25 linier ned.

Hele kommandoen udføres før ændringen vises på skærmen.

Som vist kan ordre gentages, hvis disse angives i parentes og antallet af gentagelser står foran parentesen. I parentesen kan flere ordre angives blot de adskilles af et komma.

Ønskes de næste 10 linier rykket 20 tegn ind kan det gøres af følgende kommando:

```
1s,1d,10(20(ib),1d)
```

Først flyttes markøren til starten af den aktuelle linie (1s), derefter flyttes den ned på næste linie (1d). Herefter gentages den yderste parentes 10 gange (en for hver linie). For hver gang gentages ordren ib (indsæt blank) 20 gange, hvorefter markøren flyttes til næste linie.

Ved nogle kommandoer kendes antallet af gange en ordre skal udføres ikke. Dette kan f.eks. være hvis samtlige forekomster af et ord i en tekst skal udskiftes med et andet. I SE's kommandoer kan et ikke nærmere specificeret antal angives med ! (eller \$, se afsnit 3.2.2). Udførelsen af de angivne ordre stopper da først når det ikke er muligt at udføre dem længere, d.v.s., når en søgning resulterer i at det søgte ikke blev fundet.

Gives kommandoen: !(ss/84/85/) i starten af en tekst vil alle forekomster af 84 blive erstattet af 85 (ss ordren er Substitute String; Find og erstat tegn).

Eksempel 6. Definition af kontroltaster

Funktionen af alle kontroltaster i SE defineres ved opstart af editoren. Denne definition foretages ved at tildele de enkelte kontroltaster 2 kommandoer.

Den ene kommando udføres når kontroltasten nedtrykkes, mens den anden udføres, når der først trykkes på prefix-tasten og dernæst på kontroltasten.

Kommandoerne, der er tildelt kontroltasterne, har samme format som de kommandoer, der kan udføres på kommandolinien.

Ved at benytte ordren dk (define key) kan en kontoltast tildeles nye kommandoer, når der er behov for dette.

Tastes <prefix> dk <cr> skriver SE 'Define key (number or presskey): ', der kan nu trykkes på den kontoltast, der ønskes omdefinert f.eks. <prefix> Ctrl-G. Der udskrives så 'Enter definition: ' og tastens nye definition kan indtastes tilsvarende en kommando på kommandolinien. F.eks. kan <prefix> Ctrl-G defineres med: 5(dc),ld.

Hver gang der nu trykkes <prefix> Ctrl-G, slettes 5 tegn og markøren flyttes til næste linie.

Trykkes f.eks. <prefix> 15 <prefix> Ctrl-G slettes 5 tegn (på markørens position) i de næste 15 linier.

Definitionen af en kontoltast kan ses ved at benytte ordren sk (skow key). Trykkes f.eks. <prefix> sk <cr> <prefix> Ctrl-G udskrives 'Key nnn defined as: 5(dc),ld' (nnn angiver et key number, se xx.x)

Bemærk: Alle ordre m.m. kan fortrydes (inden de udføres) ved at taste Ctrl-X

Eksempel 7. Brug af register

I SE er der mulighed for at benytte et sæt registrer til at flytte og gemme dele af teksten.

Et register er et navngivet tekst afsnit. Hvilken tekst et register indeholder defineres v.h.a SE's ordre.

Skal der f.eks. flyttes en del af en tekst fra en sted til et andet benyttes der et register.

Først markeres den del af teksten det drejer sig om v.h.a et mærke i teksten og markøren på følgende måde:

Mærket sættes på første tegn, der skal indgå i den tekstdel, der skal flyttes, dette gøres ved at sætte markøren på tegnet og trykke Ctrl-Y (som er defineret med ordren sm; set mark). Dernæst flyttes markøren ned efter det sidste tegn, der skal medtages.

Derefter udføres register flytte ordren (rm; register move). Denne ordre flytte den markede tekstdel til det register, der angives. Benyttes <prefix> Ctrl-C flyttes til et standard register (kaldet 'default'). Tekstdelen står nu i registret og er fjernet fra teksten.

Skal tekstdelen indsætte et andet sted i teksten flyttes markøren til dette sted og der trykkes Ctrl-V (ri, register insert). Tekstdelen, der står i standard registret, vil derefter blive indsat fra markørens position. Den vil dog også blive stående i registret og kan derfor benyttes igen senere, hvis det ønskes.

Ønskes en tekstdel kopieret til registret i stedet for at blive flyttet, kan det lade sig gøre v.h.a register kopi ordren (rc; register copy). Kopiering til standard registret udføres når der trykkes Ctrl-C. Her bliver teksten altså stående, men der laves et kopi i registeret default.