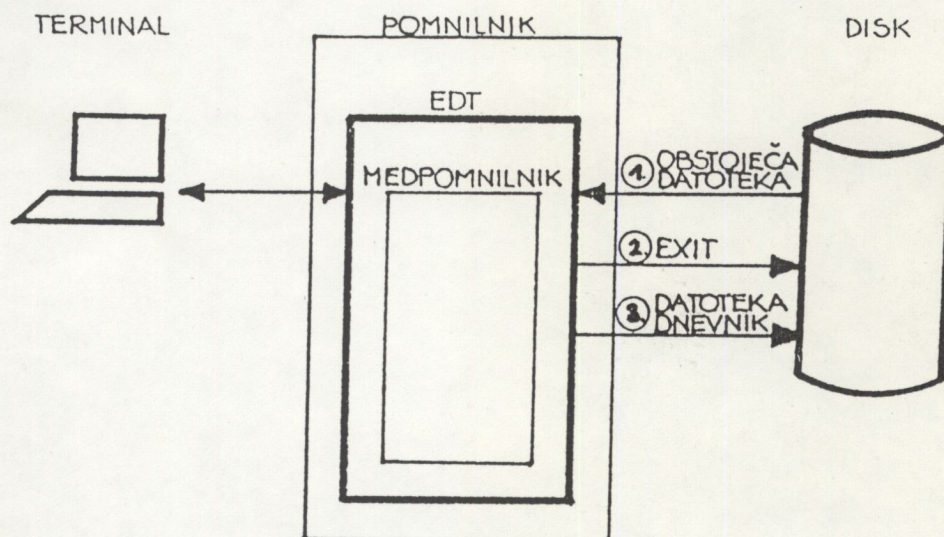


7.2 Princip dela EDT-ja

Kadar urejamo že obstoječo datoteko, EDT ob klicu prepíše vsebino imenovane datoteke v delovno področje v pomnilniku - MAIN buffer. Med urejanjem besedila nato dodajamo nov tekst, izdajamo razne ukaze za urejanje, ki spreminjajo besedilo v POMNILNIKU. Šele ob zaključku urejanja EDT po potrebi prepíše delovno področje nazaj v datoteko na DISK.

Pri kreiranju nove datoteke EDT le pripravi delovno področje in nato pričakuje naše ukaze.



Slika 23: Princip dela EDT-ja

Izbiramo lahko med tremi načini dela:

- vrstični način (Line mode)

EDT se javi z odzivnim znakom * in pričakuje ukaze vrstičnega načina, ki jih moramo odtipkati na tipkovnici. Omogoča nam urejanje besedila po vrsticah s pomočjo vrstičnih števil (EDT vsaki vrstici teksta priredi neko število - line number). Delamo lahko z eno ali več vrstami hkrati. Vrstični način nam omogoča uporabo EDT-ja na terminalih s pomičnim papirjem (hardcopy terminals). To je privzet način dela EDT-ja.

- keypad

Keypad je znakovno usmerjen način dela. Na zaslonu vidimo tekst tako kot izgleda v datoteki. Kazalec lahko premikamo po znakih v vse smeri in dodajamo tekst tako, da ga enostavno odtipkamo na glavni tipkovnici. Ukazov ni potrebno tipkati, ampak jih posredujemo EDT-ju s pritiskom tipk na pomožni tipkovnici in nekaterih tipk na glavni tipkovnici. Ta način lahko uporabljamo samo s terminali tipov VT52, VT100, VT2xx.

- nokeypad

Podobno kot keypad je tudi nokeypad znakovno usmerjen način dela, le da moramo vse ukaze odtipkati na glavni tipkovnici. Primeren za zaslonske terminale, ki ne omogočajo keypad načina.

Izbira načina dela:

vrstični	\$ EDIT file-spec *
keypad	\$ EDIT file-spec * CHANGE
nokeypad	\$ EDIT file-spec * SET NOKEYPAD * CHANGE

Prehod med različnimi načini dela:

vrstični	keypad	*CHANGE
vrstični	nokeypad	*SET NOKEYPAD *CHANGE
keypad	vrstični	CTRL/Z
keypad	nokeypad	CTRL/Z *SET NOKEYPAD *CHANGE
nokeypad	vrstični	EX
nokeypad	keypad	EX *SET KEYPAD *CHANGE

7.3 Zaključek urejanja

Normalen zaključek urejanja:

EXIT

Kadar zaključimo urejanje na enega od načinov opisanih pod prvo točko, EDT prepíše delovno področje (medpomnilnik MAIN) na disk in vrne kontrolo DCL-u.

QUIT

Pod drugo točko so navedeni ukazi, ki prav tako zaključijo urejanje, vendar pri teh EDT ne zapiše na disk vsebine medpomnilnika MAIN.

Nasilna prekinitev:

Tretja točka prikazuje ukaze, ki nasilno prekinejo urejanje. Pri tem EDT ne shrani pravilno vsebine medpomnilnika MAIN, temveč dobimo vse spremembe in novi tekst v datoteko-dnevnik (journal file), ki ima običajno isto ime kot datoteka, ki smo jo urejali, le tip je spremenjen (JOU). Uporaba teh ukazov za zaključek urejanja ni priporočljiva!

	vrstični	keypad	nokeypad
1.	* EXIT <RET>	CTRL/Z * EXIT <RET> <PF1><7> EXIT <ENTER>	EX * EXIT <RET>
2.	* QUIT <RET>	CTRL/Z * QUIT <RET> <PF1><7> QUIT <ENTER>	EX * QUIT <RET> QUIT
3.	CTRL/C CTRL/Y	CTRL/C CTRL/Y	CTRL/C CTRL/Y

7.4 Pomoč pri delu (HELP)

Med urejanjem besedil so nam stalno na voljo informacije o posameznih ukazih oziroma njihovih funkcijah. Ni potrebno prekiniti dela, da bi dobili pomoč. V vrstičnem načinu dela, nam je na voljo ukaz

* HELP item <RET>

V keypad načinu pa dobimo pomoč s pritiskom tipke <PF2>:

<PF2> (help)

- Izpišejo se informacije o ukazih.
- Če želimo podrobnejše informacije
- o funkciji neke tipke, enostavno
- pritisnemo izbrano tipko.

<SP> Z urejanjem besedila nadaljujemo tako, da odtipkamo presledek.

V keypad načinu lahko dobimo inf. o ukazih tudi drugače:

<PF1> <7>

Command: HELP item <ENTER>

Pomoč.

•

•

•

Izpišejo se zahtevane inf.

<RET>

Nadaljujemo z delom.

Primer:

* HELP <RET>

7.5 **Zaščita podatkov med urejanjem**

Med urejanjem besedila EDT beleži vse izdane ukaze, ves dodani tekst, skratka vse spremembe v datoteko-dnevnik (journal file). Običajno ima ta datoteka enako ime kot datoteka, ki jo urejamo, tip pa je JOU. S kvalifikatorjem /JOURNAL=file, pa lahko določimo tudi drugo ime datoteke-dnevnika.

Na ta način je omogočeno, da ob morebitnem razpadu sistema (system crash) ali nasilni prekinitvi urejevalnika (npr. s CTRL/C) EDT lahko obnovi večino našega dela. To dosežemo s kvalifikatorjem /RECOVER pri klicu EDT-ja.

§ EDIT/EDT/RECOVER file

EDT file /recover

Kvalifikatorji:

/JOURNAL=file	Spremembe se zapisujejo v imenovano datoteko.
/NOJOURNAL	EDT ne vodi datoteke-dnevnika.
/RECOVER	Omogoča restavriranje stanja pred prekinitvijo.

Če s kvalifikatorjem /JOURNAL ne določimo drugače, EDT kreira datoteko-dnevnik v privzetem direktoriju. Pozor pri urejanju datotek, ki se ne nahajajo v privzetem direktoriju!

Kvalifikator /NOJOURNAL povzroči, da EDT ne vodi dnevnika našega dela, kar pa ni priporočljivo! (uporablja se pri urejanju datoteke-dnevnika)

Primer:

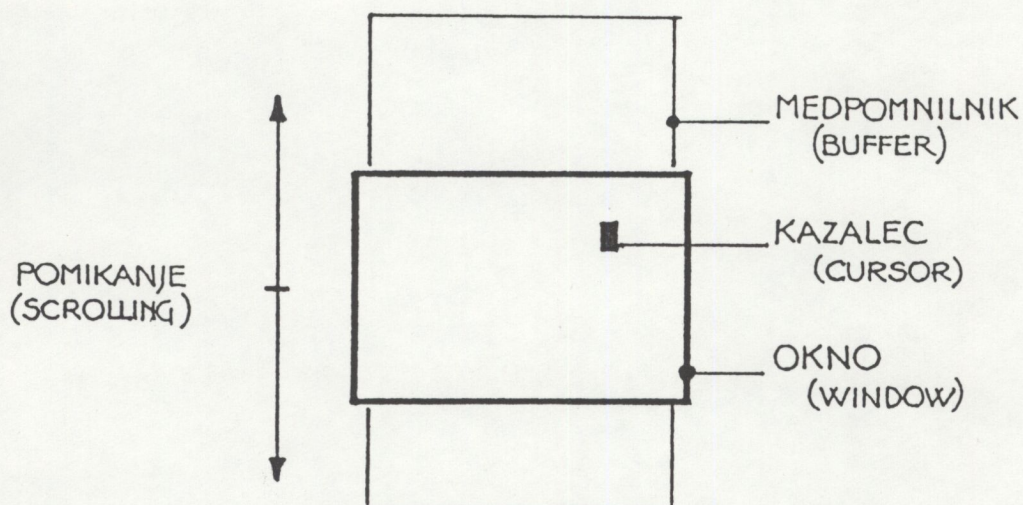
Ko smo urejali datoteko A.TXT, je prišlo do pada sistema. EDT je vse spremembe shranil v datoteko A.JOU, tako da stanje lahko kasneje restavriramo z enim od ukazov:

§ EDIT/EDT/RECOVER A.TXT

§ EDIT/EDT/RECOVER/JOURNAL=A NEW.TXT

7.6 OSNOVE DELA V KEYPAD NACINU

V keypad načinu dela EDT izpiše na zaslon več vrst teksta (običajno 22) in sicer v taki obliki, kot je v datoteki. Zaslon nam predstavlja nekako okno skozi katero gledamo datoteko. Kazalec (cursor) označuje trenutni položaj v datoteki, to je, mesto, kjer se dodaja nov tekst oziroma mesto na katerem delujejo ukazi. Pri spreminjanju položaja kazalca, se ustrezno temu spremeni tudi tekst na zaslonu, podobno kot bi se datoteka premikala pred oknom (scroll).



Slika 24: Okno (window)

7.6.1 Vnos, dodajanje teksta

Ves tekst, ki ga odtipkamo na tipkovnici, se takoj pokaže na zaslonu, na mestu, kjer stoji kazalec (cursor). Pri dodajanju EDT novi tekst vedno vriva med starega (insert mode). Stari del besedila ostane nespremenjen.

Novo vrstico teksta začnemo s pritiskom na tipko <RETURN>. Kadar je kazalec med tekstem, sredi vrstice, in pritisnemo tipko <RETURN>, se vrstica prelomi v dve. Del teksta od kazalca do konca vrstice se prestavi v novo vrsto, ki je takoj za tekočo. Če pa je kazalec na začetku vrstice, <RETURN> vrine prazno vrsto pred tekočo vrstico. Tako lahko dodamo tudi več praznih vrst.

7.6.2 Vnos ukazov

V keypad načinu lahko večino ukazov izdamo z enostavnim pritiskom ene ali več tipk na pomožni tipkovnici (keypad) ali nekaterih tipk na glavni tipkovnici.

PF1 GOLD 20	PF2 HELP 10	PF3 FNDNXT FIND 11	PF4 DEL L UND L 17
7 PAGE COMMAND 7	8 SECT FILL 8	9 APPEND REPLACE 9	— DEL W UND W 18
4 ADVANCE BOTTOM 4	5 BACKUP TOP 5	6 CUT PASTE 6	9 DEL C UND C 19
1 WORD CHNGCASE 1	2 EOL DEL EOL 2	3 CHAR SPECINS 3	ENTER ENTER
0 LINE OPEN LINE 0	• SELECT RESET 16		SUBS 21

VT100

Slika 25: Funkcije tipk na pomožni tipkovnici

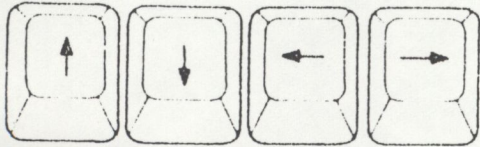
Vsaka tipka na pomožni tipkovnici ima dve funkciji - zgornjo in spodnjo. Zgornjo funkcijo dosežemo samo s pritiskom na to tipko, za doseganje spodnje (alternativne) funkcije pa pritisnemo pred tem tipko GOLD (PF1 na terminalih VT100).

<tipka>
<PF1> <tipka>

zgornja funkcija
spodnja funkcija

Na glavni tipkovnici so še naslednje ukazne tipke: <RETURN>, <DELETE>, <BACK-SPACE>, <LINE-FEED>, tipke s puščicami ter kombinacije s tipko <CTRL>. Pomen posameznih tipk si bomo ogledali v nadaljevanju.

7.6.3 Premikanje po tekstu



Najenostavnejši način premikanja po tekstu je s pomočjo štirih puščic, ki se nahajajo v zgornjem desnem delu glavne tipkovnice. S pritiskom teh tipk se kazalec premakne v izbrani smeri za en znak ali pa tudi za več znakov, če tipko držimo dalj časa (odvisno od nastavitve term. - PONAHLJANJE).

CHAR
CHAR [3]

Premakne kazalec za en znak naprej ali nazaj.

WORD
WORD [1]

Premakne kazalec eno besedo naprej ali nazaj. Besedo lahko definiramo z ukazom SET ENTITY WORD.

<BACK SPACE>

Premakne kazalec na začetek tekoče vrstice. Če je kazalec že na začetku vrste, se premakne na začetek predhodne vrste.

EOL
EOL [2]

Kazalec na konec tekoče/naslednje/predhodne vrste, odvisno od smeri premikanja.

LINE [0]

Kazalec postavi na začetek naslednje/tekoče/predhodne vrste.

SECT [8]

Premakne kazalec za 16 vrstic (section) naprej ali nazaj.

PAGE [7]

Postavi kazalec na začetek naslednje strani -označen s <FF>. Delimiter strani lahko določimo z ukazom SET ENTITY PAGE.

Osnove OS DELTA/V
EDT

na začetek besede
TOP [PF1] [5]

Kazalec na začetek medpomnilnika.

koncu bes
BOTTOM [PF1] [4]

Premakne kazalec na konec medpomnilnika (označen z [EOB]).

ADVANCE [4] *naprej*

Usmeri delovanje funkcij CHAR, WORD, LINE, EOL, SUBSTITUTE, SUBSTITUTE NEXT, FIND in FIND NEXT naprej (tj. desno in proti koncu medpomnilnika).

BACKUP [5] *NAZAJ*

Usmeri delovanje funkcij CHAR, WORD, LINE, EOL, SUBSTITUTE, SUBSTITUTE NEXT, FIND in FIND NEXT nazaj (tj. levo in proti začetku medpomnilnika).

7.6.4 Brisanje delov besedila

<DELETE>

Briše znak pred kazalcem. Kadar se nahaja kazalec na začetku vrstice, briše delimiter vrstice (združi predhodno in tekočo vrsto).

DEL C [,]

Briše znak pod kazalcem. Kadar je kazalec na koncu vrstice, briše delimiter vrstice (združi tekočo in naslednjo vrsto).

DEL W [-]

Briše znake od kazalca do začetka naslednje besede.

<LINE-FEED>

Briše znake od začetka besede do kazalca. Če se nahaja kazalec na prvem znaku besede, briše predhodno besedo.

DEL L [PF4]

Briše znake od kazalca do konca tekoče vrstice, vključno z vrstičnim ločilom (naslednja vrstica se priključi).

<CTRL><U>

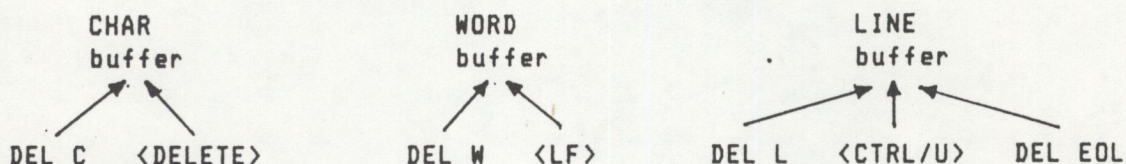
Briše znake od kazalca do začetka tekoče vrste. Če se nahaja kazalec na začetku vrste briše predhodno vrsto.

DEL EOL [PF1] [2]

Briše vse znake od kazalca do konca tekoče vrstice, razen vrstičnega ločila.

7.6.5 Vračanje pobrisanega teksta

EDT shranjuje pobrisani tekst v posebne medpomnilnike za brisanje. Obstajajo trije taki medpomnilniki, in sicer, za brisane znake, za brisane besede in za brisane vrstice. V teh medpomnilnikih se nahaja samo nazadnje pobrisani element teksta.



Ukazi, s katerimi lahko vrnemo (restavriramo) nazadnje pobrisani element besedila:

UND C [PF1] [,]

Vrne znak, ki je bil pobrisan z ukazom DEL C ali <DELETE>.

UND W [PF1] [-]

Vrne besedo, ki je bila pobrisana z DEL W ali <LINE-FEED>.

UND L [PF1] [PF4]

Vrne vrstico, ki je bila brisana z DEL L ali <CTRL/U>.

7.6.6 Obnova zaslona (refresh)

Kadar med urejanjem teksta dobimo kako sporočilo (npr. nekdo nas kliče s pomožnim programom PHONE), se to izpiše na zaslonu in "pokvari" sliko na zaslonu. Vsebino zaslona lahko obnovimo z ukazom REFRESH.

<CTRL/W>
<CTRL/R>

7.6.7 Iskanje nizov

Nek niz znakov v medpomnilniku lahko poiščemo z ukazoma FIND in FNDNXT. Najprej uporabimo ukaz FIND, ki nas vpraša po nizu, ki ga iščemo. Nato z ukazom FNDNXT poiščemo prvo pojavitev tega niza (odvisno od smeri).

Primer:

```
FIND          [PF1] [PF3]
Search for:  niz
FNDNXT       [PF3]
```

Naslednje pojavitve istega niza poiščemo z ukazom FNDNXT.

Ukaz FIND lahko zaključimo tudi z ukazoma ADVANCE ali BACKSPACE ter tako eksplicitno določimo smer iskanja.

7.6.8.2 Premik označenega teksta

Označeni tekst, ki smo ga shranili v medpomnilnik PASTE, lahko premaknemo na katerokoli mesto v delovem medpomnilniku (ali tudi v drugem medpomnilniku) tako, da pred ukazom PASTE premaknemo kazalec na mesto, kamor želimo premakniti označen tekst.

- | | |
|-------------------|-------------------------------------|
| 1. SELECT | Tekst označimo z ukazom SELECT |
| 2. premik kazalca | in uporabo tipk za premik kazalca. |
| 3. CUT | Shrani označeni tekst v PASTE. |
| 4. premik kazalca | Premaknemo kazalec na izbrano mesto |
| 5. PASTE | in vrnemo tekst iz PASTE. |

7.6.8.3 Ukaza REPLACE in SUBSTITUTE

REPLACE [PF1] [9]

Briše označeni tekst in ga zamenja z vsebino medpom. PASTE.

- | | |
|-------------------|-----------------------------------|
| 1. SELECT | Z ukazom SELECT označimo |
| 2. premik kazalca | izbrani tekst in ga |
| 3. CUT | shranimo v medpomnilnik PASTE |
| 4. SELECT | Označimo drugi del teksta, ter |
| 5. premik kazalca | ga z uporabo ukaza REPLACE |
| 6. REPLACE | zamenjamo z vsebino medpom. PASTE |

SUBSTITUTE [PF1] [ENTER]

Poišče niz, ga briše in nadomesti z vsebino medpomnilnika PASTE, ter nato poišče še naslednjo pojavitev niza.

- | | |
|-------------------|--|
| 1. SELECT | Z ukazom SELECT označimo |
| 2. premik kazalca | izbrani tekst in ga |
| 3. CUT | shranimo v medpomnilnik PASTE |
| 4. FIND | Z ukazom FIND definiramo |
| 5. iskalni niz | iskalni niz, ki ga ukaz SUBSTITUTE |
| 6. SUBSTITUTE | zamenja z vsebino medpom. PASTE,
ter nato poišče še naslednjo
pojavitev tega niza. |

7.6.8.4 Sprememba velike/male črke

Z ukazom `CHNGCASE` lahko v celotnem označenem besedilu spremenimo male črke v velike ter obratno. Vsi ostali znaki in razpored teksta ostanejo nespremenjeni.

SELECT	[.]
.	
.	premik kazalca
.	
CHNGCASE	[PF1] [1]

7.7 SET in SHOW ukazi

Z ukazom SET lahko delovno okolje urejevalnika prilagodimo našim potrebam in željam. Nekaj primerov:

SET CURSOR top:bottom	definiramo področje pomika
SET ENTITY WORD "string"	predpišemo besedna ločila
SET LINES number	število vrstic na zaslonu < 22
SET MODE CHANGE	zaslonski način dela
SET QUIET	napak ne javlja z zvoncem
NOQUIET	napake javlja tudi z zvoncem
SET SCREEN width	širina zaslona - PAKA 80/132
SET SEARCH ...	atributi pri iskanju nizov
SET TRUNCATE	dolge vrste na zaslonu odreže
NOTRUNCATE	dolge vrste na zaslonu lomi
SET WRAP n	določimo max. dolžino vrst

Ukaz SHOW pokaže vrednosti parametrov, ki smo jih z ukazom SET morebiti spremenili.

7.8 Pregled ukazov vrstičnega načina

CHANGE	prehod na zaslonsko urejanje (keypad, nokeypad)
CLEAR	briše vsebino nekega medpomnilnika
COPY	kopira tekst iz obsega-1 pred vrstico obseg-2
DELETE	briše vrstice navedene v obsegu
EXIT	zaključi urejanje, vsebino medpom. MAIN zapiše na disk
FIND	vrstica določena z obsegom postane tekoča vrstica
HELP	izpiše informacije o uporabi EDT-ja
INCLUDE	vsebino izbrane datoteke prepíše v medpomnilnik
INSERT	na mesto določeno z obsegom vnese eno ali več novih vrst
MOVE	premakne del teksta (tudi med različnimi medpomnilniki)
[range]<RET>	impliciten TYPE ukaz.
PRINT	tekst določen z obsegom prepíše v navedeno datoteko
QUIT	zaključi urejanje, vendar ne shrani vsebine MAIN opsijsko shrani datoteko-dnevnik (journal file)
REPLACE	briše del teksta neveden v obsegu in ga zamenja z novim
RESEQUENCE	preštevilči v obsegu navedene vrstice
SUBSTITUTE	niz-1 zamenja z nizom-2 v navedem obsegu
TYPE	izpiše navedeni obseg vrstic
WRITE	prepíše določeni obseg vrstic v navedeno datoteko

7.8.1 Uporaba ukazov vrstičnega načina v keypad načinu

Med delom v keypad načinu lahko uporabljamo ukaze vrstičnega načina, ne da bi za stalno prešli v vrstični način. To dosežemo z ukazom COMMAND, ki izvrši en ukaz vrstičnega načina in ohrani keypad način dela.

COMMAND
[PF1][7]

Command: line-mode-command <ENTER>

7.8.2 Delo z več medpomnilniki

EDT obdeluje vse besedilo v medpomnilnikih, kjer se med urejanjem hrani in kjer se izvajajo tudi vse spremembe. Ob klicu EDT sam kreira medpomnilnik MAIN, ki služi kot glavno delovno področje (kopija obstoječe datoteke).

Poleg medpomnilnika MAIN, EDT sam kreira še medpomnilnik PASTE, kamor se shranjuje tekst, ki smo ga brisali z ukazom CUT. Vsakokrat, ko uporabimo CUT, se medpomnilnik PASTE prepíše z novo vsebino (v njem je vedno zadnja kopija brisanega besedila). PASTE medpomnilnik lahko normalno urejamo, enako kot MAIN, razen tega, da ne moremo uporabljati ukazov CUT in PASTE.

Dodatni medpomnilniki, ki jih lahko kreiramo sami, nam omogočajo delo z več datotekami. Primeri uporabe:

- vsebino ene datoteke razdelimo na več manjših
- vključimo obstoječo datoteko oziroma njen del v datoteko, ki jo urejamo
- iz delov datoteke sestavimo novo datoteko

Obstoječe medpomnilnike pogledamo z ukazom vrstičnega načina SHOW BUFFER, ki izpiše imena medpomnilnikov, ter število vrstic, ki jih vsebujejo. Medpomnilnik v katerem urejamo besedilo je označen z enačajem (=). Če EDT še ni prebral vseh vrstic iz datoteke v medpomnilnik MAIN, se to označi z zvezdico (*) za števcem vrstic.

* SHOW BUFFER

=MAIN	120*	lines
PASTE	no	lines

Nove medpomnilnike kreiramo z ukazom FIND. Novi medpomnilnik postane tudi naš novi tekoči medpomnilnik, v katerem lahko, enako kot v medpomnilniku MAIN, uporabljamo vse ukaze za urejanje besedila. Edina razlika je ta, da EDT ob ukazu EXIT avtomatsko prepíše vsebino medpomnilnika MAIN nazaj na disk, za ostale medpomnilnike pa moramo poskrbeti sami (npr. z ukazom WRITE).

* FIND =buffer

Medpomnilnik s katerim delamo zamenjamo pravtako z ukazom FIND.

* FIND =MAIN

Zgornji ukaz FIND povzroči, da postane MAIN naš tekoči medpomnilnik. Ukaz FIND ne vpliva na vsebino medpomnilnikov!

Nove medpomnilnike lahko kreiramo tudi z naslednjimi ukazi:

- INCLUDE

Prepíše vsebino neke datoteke na disku v medpomnilnik. Če medpomnilnik še ne obstaja ga EDT kreira.

* INCLUDE file =buffer

TUDI MAIN
ALI PASTE

Primer:

* INCLUDE OUTSIDER.TXT =NEW

Vsebino datoteke OUTSIDER.TXT vključimo v medpomnilnik NEW, ki postane tudi naš tekoči medpomnilnik.

- COPY

Prekopira text iz enega medpomnilnika v drugega. Če ciljni medpomnilnik še ne obstaja ga EDT kreira. Ciljni medpomnilnik postane tekoči.

* COPY =buffer1 TO =buffer2

- MOVE

Premakne text iz enega medpomnilnika v drugega. Če ciljni medpomnilnik še ne obstaja ga EDT kreira. Ciljni medpomnilnik postane tekoči.

* MOVE =buffer1 TO =buffer2

buffer 1 se briše! pri copy ne

Medpomnilnik in njegovo vsebino brišemo z ukazom vrstičnega načina CLEAR. Za MAIN in PASTE velja, da lahko brišemo njuno vsebino, ne moremo pa brisati samih medpomnilnikov.

* CLEAR buffer

Specifikacija obsega (range) pri delu z več medpomnilniki.

7.8.3 Definiranje ukaznih tipk

EDT omogoča definirati določene tipke oziroma kombinacije tipk kot ukazne tipke, ki jih nato lahko uporabljamo v KEYPAD načinu dela. Funkcije, ki jih tem tipkam priredimo, opišemo s pomočjo nokeypad ukazov ali pa z že obstoječimi keypad ukazi.

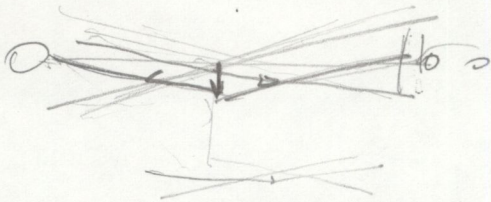
7.8.3.1 Zakaj definiranje ukaznih tipk EDT-ja?

Ugodnosti, ki nam jih definiranje ukaznih tipk omogoča:

- često uporabljana zaporedja ukazov lahko priredimo eni tipki, ter tako zmanjšamo obseg tipkanja.
- uporabljamo lahko nokeypad ukaze, ki nimajo ekvivalenta med keypad ukazi (primer: SHL, SHR, CHGL, CHGU, DATE) *Shift left - ponit levo*
Shift right - ponit desno
- dostopni so nam elementi besedila, ki so definirani samo za nokeypad ukaze (npr.: stavki, odstavki, strani)

7.8.3.2 Tipke, ki jih lahko definiramo:

- tipke na pomožni tipkovnici (keypad)
- funkcijske tipke (PF1, PF2, ...)
- puščice nad glavno tipkovnico
- GOLD+znak
- CONTROL črka (A..Z, [,], \, ~, ^, _)
- GOLD+številka
- GOLD+CONTROL+črka



7.8.3.3 Ukazi za definicijo ukaznih tipk

Določenim tipkam oziroma kombinacijam tipk lahko z ukazom DEFINE KEY priredimo niz ukazov, ki se izvede ob uporabi tipke.

```
* DEFINE KEY key-name AS "function"
```

Imena tipk na pomožni tipkovnici so razvidna iz slike xx.

Funkcijo tipke zapišemo s pomočjo NOKEYPAD ukazov. Pri tem velja:

- pika (.) na koncu definicije pomeni, da se funkcija izvede takoj ob pritisku tipke
- vprašaj (?) v definiciji omogoča vnos vrednosti šele pri uporabi tipke (odzivni znak pišemo med narekovaje)
- nokeypad ukaz EXT nam omogoča v definicijah tipk uporabiti tudi ukaze vrstičnega načina

Primeri:

```
* DEFINE KEY GOLD Q AS 'EXT QUIT.'  
* DEFINE KEY GOLD E AS 'EXT EXIT.'  
* DEFINE KEY GOLD 14 AS 'SHR.'  
* DEFINE KEY GOLD 15 AS 'SHL.'  
* DEFINE KEY GOLD S AS 'S/?"Subs: "/?" With: "/.'  
* DEFINE KEY GOLD I AS 'EXT INCLUDE ?"Include File: ".'
```

Handwritten notes:
- "SHR." and "SHL." are annotated with "vprašaj" (question mark) and "vprašaj (stanje)" (question mark state).
- "Include File: ".'" is annotated with "vnesti - razširitev" (insert - extension).

SET CURSOR 3; 18 "teh linij se gane obrot"

7.8.4 Uporaba MACRO-jev

Macro je niz ukazov vrstičnega načina dela, ki jih izvajamo kot celoto enostavno z uporabo imena, ki mu je pridruženo. Ime macro-ja definiramo kot veljaven ukaz vrstičnega načina z ukazom DEFINE MACRO.

* DEFINE MACRO name

Ukaze, ki sestavljajo telo macro-ja napišemo v medpomnilnik z istim imenom:

* FIND =name

.

.

.

telo macro-ja

CTRL/Z

* FIND =MAIN.

EDT bo izvedel ukaze, ki sestavljajo macro, vedno ko njegovo ime uporabimo kot ukaz vrstičnega načina.

* name

7.8.4.1 Kaj lahko vsebujejo macro-ji:

- SET ukaze
- ukaze vrstičnega načina
- DEFINE KEY ukaze
- definicije in klice drugih macro-jev

Macro-je lahko kreiramo kadarkoli med urejanjem teksta in jih nato po potrebi uporabljamo do konca dela z EDT-jem. Ko zaključimo urejanje z EDT-jem, se macro-ji, ki so vsebovani v medpomnilnikih, "pozabijo".

Če jih želimo shraniti za nadaljnjo uporabo, jih moramo zapisati v datoteko (napr. z ukazom WRITE), ter nasledjič samo definirati njihovo ime, ter jih vključiti v ustrezni medpomnilnik. Lahko pa macro-je in njihove definicije vključimo kar v zagonsko datoteko in so nam tako vedno na razpolago.

OPOZORILO

Kadar kreiramo macro z istim imenom kot obstoječ ukaz vrstičnega načina, EDT pri njegovi uporabi izvede macro in NE ukaza z istim imenom.

Vrednosti, ki jih v macro-ju uporabljamo, moramo v njemu tudi definirati. Torej, ne moremo prenašati parametrov! To je obenem tudi glavna pomanjkljivost macro-jev.

7.8.5 Uporaba zagonskih datotek

Zagonske datoteke (Start-up command files) vsebujejo ukaze vrstičnega načina dela, ki se izvedejo ob vsakem klicu EDT-ja, še preden se urejevalnik javi in tako omogočajo pripravo okolja pred začetkom dela.

Tipični ukazi, ki jih vsebujejo zagonske datoteke:

- INCLUDE ukazi s katerimi vključimo vsebino raznih datotek v medpomnilnike na ta način lahko v medpomnilnike vključimo npr. macro-je
- DEFINE KEY ukazi s katerimi (re)definiramo funkcije ukaznih tipk
- DEFINE MACRO ukazi definirajo nek medpomnilnik kot macro - pridružijo mu ime, ki ga uporabljamo kot ukaz vrstičnega načina
- SET ukazi za spremembe posameznih parametrov EDT-ja npr. SET MODE CHANGE začne delo v keypad načinu

Kadar želimo uporabiti zagonsko datoteko, navedemo njeno ime pri klicu EDT-ja.

```
$ EDIT/COMMAND=startup_file file
```

EDT pri vsakem klicu pregleda naš trenutni direktorij in če vsebuje datoteko z imenom `EDTINI.EDT` jo avtomatsko uporabi kot zagonsko datoteko. To lahko preprečimo z uporabo kvalifikatorja /NOCOMMAND.

```
$ EDIT/NOCOMMAND file
```

7.8.6 Primerjava macrojev in zagonskih datotek

- oboji vsebujejo ukaze vrstičnega načina
- zagonska datoteka se izvede samo enkrat - pred začetkom dela
- macro-jev imamo lahko več in jih izvajamo po potrebi (tudi večkrat)