SPILLER-MUVS MAJA

# OPERACIJSKI SISTEM DELTA/M ZA ZAČETNIKE

delta računalniški sistemi ®

SPILLER-MUYS MAJA

# OPERACIJSKI SISTEM DELTA/M ZA ZAČETNIKE

IZDAJA 1.0 FEBRUAR 1982

# 1.poglavje

# UPORABA TERMINALA

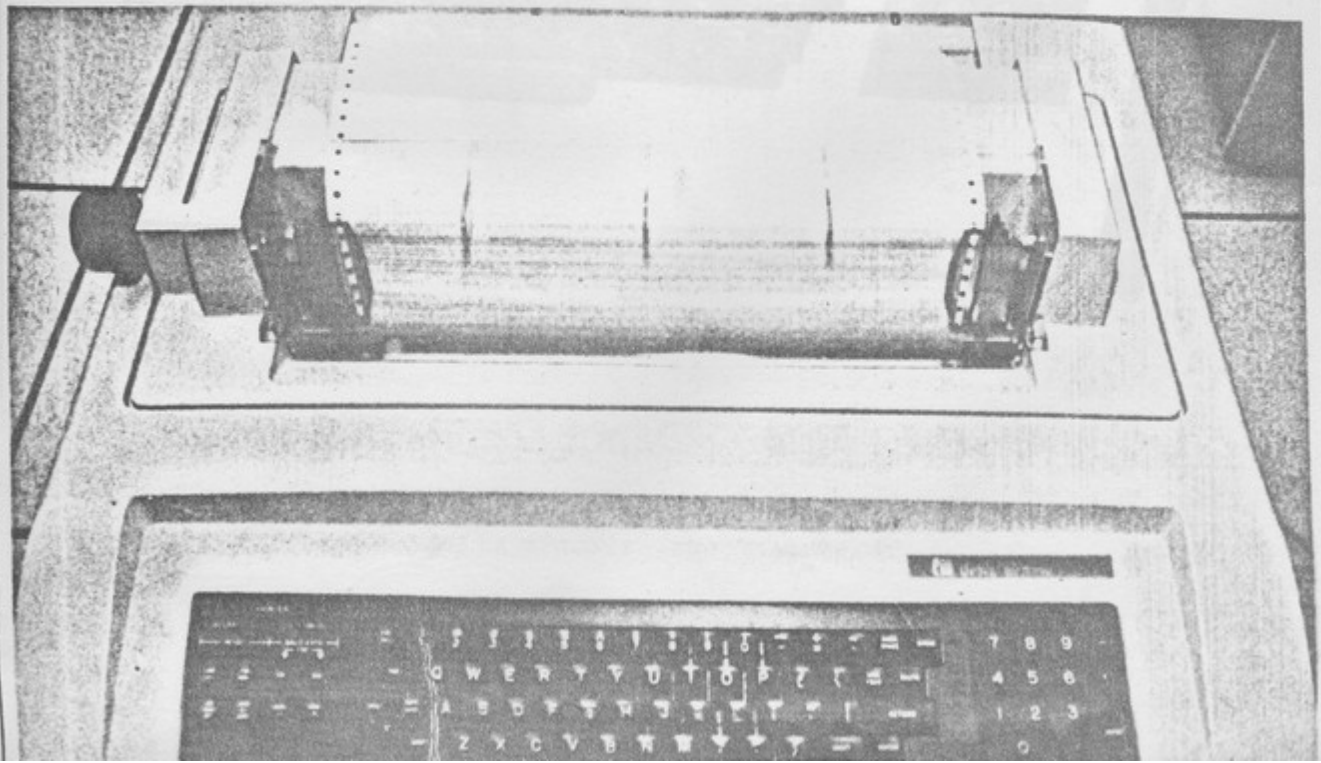
V tem poglavju se bomo seznanili z nacini, kako se prijavimo na operacijski sistem DELTA/M. Operacijski sistem nam omogoca komunikacijo s strojno opremo racunalnika.

Operacijski sistem prevaja jezik, ki ga razumemo mi, v jezik, ki ga lahko razume racunalnik. Prevaja tudi sporocila, ki nam jih posilja racunalnik. Ta dvosmerni komunikacijski proces navadno poteka preko terminala.

Uporabljamo lahko razlicne tipe terminalov. Delijo se na:
*TISKALNI terminal, ki pise na papir in
*ZASLONSKI terminal, ki prikazuje znake na ekranu.

Vecina zaslonskih terminalov ne omogoca trajnega zapisa dela na terminalu.

Na sliki vidimo tiskalni in zaslonski terminal.




računalniški sistemi delta
kopa 2000

## 1.1 UPORABA TERMINALSKE TASTATURE

Vse racunalniske tastature so podobne tistim na pisalnih strojih. Poleg tega pa tastatura vkljucuje se posebne tipke za razlicne funkcije racunalnika. Te tipke niso na vseh termnalih,na istem mestu. Zato preverimo razporeditev tipk vsakic, ko delamo z racunalnikom, ki ga se ne poznamo.

### 1.1.1 funkcijske tipke racunalnika

Nekatere funkcijske tipke, ki nastopajo na tastaturi so opisane v tabeli 1-1. Za specificno informacijo o programski uporabi funkcijskih tipk preverite dokumentacijo za sistemski program, ki ga uporabljte.

### 1.1.2 kontrolni znaki

Kontrolni znak lahko vnesemo s pritiskom na ustrezno tipko, medtem ko pritiskamo na tipko <CTRL>. Enako reagira sistem s pritiskom na stresico <^> nato pa na tipko (primer:<^U>)
Nekateri kontrolni znaki se zaradi njihove funkcije ne izpisejo na terminal.

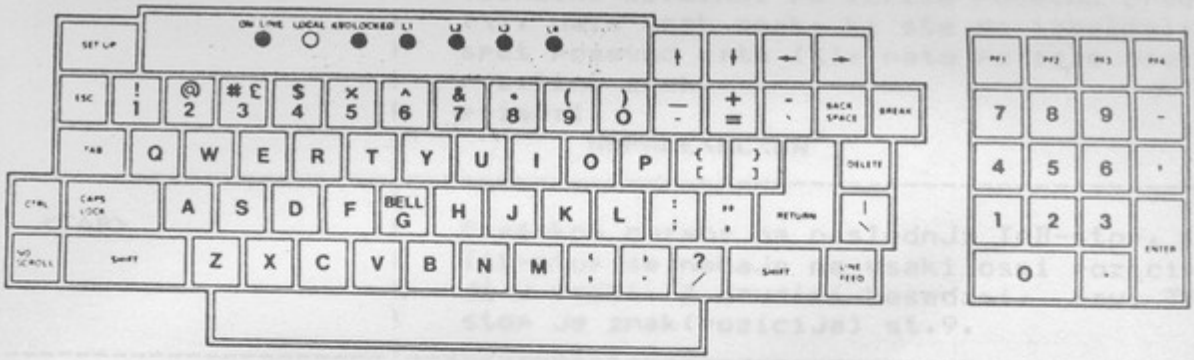


Tabela 1-1 Posebne tipke terminala

| funkcijska tipka | opis |
|---|---|
| <CR>ali<RETURN> | Konca vnos vrste in prenese cursor na 1. mesto v novi vrsti(cursor je utripajoc znak na mestu pisanja) na kratko:kadar hocete nadaljevati vnos v novi vrsti, pritisnite <CR>ali<RETURN> |
| <CTRL/n> | Kombinacija tipke <CTRL> in neke ustrezne crke izvrsi mnozico razlicnih funcij. Vsaka veljavna kombinacija se imenuje kontrolni znak, predstavljen z "n", kjer je "n" spremenljiva crka (tabela 1-2) |
| <RUBOUT>ali<DELETE> | Izbrise zadnji znak v tekoci vrsti. Ce na tipko pritisnemo veckrat, postopoma brise znake proti levi. Vecina zaslonskih terminalov odstrani vsak izbrisan znak z zaslona in premakne mesto pisanja nazaj. Tiskalni terminal pa izpise posevno crto (\), nato vsak znak, ki ste ga izbrisali, spet posevno crto (\), nato pa sele prvi pravilni znak. primer: NAPALE\EL\KA |
| <TAB> | Premakne cursor na naslednji TAB-stop. TAB-stop se nahaja na vsaki osmi poziciji v vrsti. Z drugimi besedami, prvi TAB-stop je znak(pozicija) st.9. |

Pri uporabi razlicnih sistemskih programov ti kontrolni znaki ne delujejo vedno tako kot je opisano. Na primer:nekateri programi ne spoznajo <CTRL/Z> kot izhod iz programa(tabela 1-2).Zaradi tega preverimo pomen kontrolnih tipk pri posameznih sistemskih programih, preden jih uporabljamo.

Tabela 1-2 kontrolni znaki terminala

| znak | opis |
|---|---|
| <CTRL/C> | Poklice monitor, ki razlaga ukaze operacijskemu sistemu. Na znak <CTRL/C> sistem odgovori z monitorskim znakom pripravljenosti: MCR> Ta signal označuje, da je sistem pripravljen sprejeti vnos s terminala. |
| <CTRL/O> | Izmenično prekine in vključi izpis na na terminal. Če vam npr. program izpisuje nezazelene podatke, pritisnite <CTRL/O> za ustavljanje izpisa podatkov! Če ne pritisnete ponovno <CTRL/O>, bo sistem zaustavil ves izpis in nato izpisal znak pripravljenosti (>) in vam s tem dal vedeti, da je izpis končan. |
| <CTRL/S>in<CTRL/Q> | <CTRL/S> ustavi izpis na terminal dokler ne pritisnete <CTRL/Q>, ki ga bo spet nadaljeval. primer:če uporabljate zaslonski terminal, ki izpisuje podatke prehitro, da bi jih lahko prebrali, pritisnite <CTRL/S>, da se izpis ustavi! Ko ste prebrali, kar je na zaslonu, pritisnite <CTRL/Q>, ki nadaljuje izpis. Postopek ponavljajte dokler niste uspeli prebrati do konca. |
| <CTRL/R> | Premakne cursor v novo vrsto (kot <CR>) in prepise tekoco vrsto na terminal brez popravljenih napak. Zato lahko tekst lazje preberemo. Pred zaključitvijo vrste z <CR> pritisnite <CTRL/R>, da se prepričate, če ste napako popravili! primer: NAPALE\EL\KA <CTRL/R> NAPAKA |
| <CTRL/U> | Izbrise celotno tekóco vrsto. To vam omogoca, da na novo vnesete celo vrsto, kadar bi bili posamezni popravki neprakticni. Zapomnite pa si, da morate, pritisniti <CTRL/U> pred <CR>, če hocete izbrisati vrsto. |
| <CTRL/Z> | Prekine izvajanje vecine sistemskih in uporabniskih programov in vrne kontrolo monitorju. |

Zapomnite si razliko med parom ukazov <CTRL/O>-<CTRL/O> in <CTRL/S>-<CTRL/Q>!

* ko drugic pritisnete <CTRL/O> se nadaljuje izpis od mesta, do katerega je med ustavitjijo 'prebral' racunalnik.
  primer:pri izpisu naravnih stevil od 1-100 ste ustavili pri stevilki 42. Med ustavitvijo pa je racunalnik prisel do st.52 in zato nadaljuje izpis podatkov od st.52.
* ko pa pritisnete <CTRL/Q> po <CTRL/S>, se izpis podatkov nadaljuje od tam, kjer ste ga ustavili z <CTRL/S>.
  primer:ustavili ste pri st.42. Izpis se nadaljuje od st.42.

## 1.2 MONITORSKI UKAZI

Ukazi, vpisani preko terminala za kontrolo sistemskih operacij, se imenujejo monitorski ukazi. Z monitorjem komuniciramo tako,da vpisemo ukaz v naslednji obliki:

```
ime ukaza    parametri/kljuc <CR>
```

Ime ukaza

Se sestoji iz treh ali vec crk in zakljucen s praznim mestom. Vsak ukaz natanko doloca monitorsko funkcijo. Monitor prebere samo prve tri crke. Dodatne crke lahko pomagajo uporabniku prepoznati ukaz. (Edina izjema tega tricrkovnega pravila je ukaz HELP.Natipkati moramo celoten ukaz, da ga lahko locimo od monitorskega ukaza HELLO.
Nekaj primerov monitorskih ukazov:

```
ABO[RT]          ustavi tekoci program
DMO[UNT]         logicno odklopi enoto
UFD              ustvari seznam uporabnikovih
                 datotek
```

Parameter

Natancneje doloci objekt ukaza, ki je navadno program ali enota. Eno ali vec praznih mest mora lociti parameter od imena ukaza ali parametre med seboj.
primer: ukazu ABORT, morate dodati tudi ime tekocega programa.
Naslednji ukaz bo pognal program VSOTADVA ob 12:25:00 :

```
RUN VSOTADVA 12:25:00
```

Naslednji ukaz pa bo prekinil izvajanje program VSOTADVA:

```
ABO VSOTADVA
```

Kljuc
Spremeni dejansko funkcijo ukaza ali pa parameter ukaza. Kadar kljuc spremeni funkcijo ukaza, moramo pustiti prazno mesto med imenom ukaza in kljucem.
primer
ime ukaza /kljuc[=vrednost]

Kadar kljuc spremeni parameter ukaza, sledi kljuc neposredno parametru. Nekateri kljuci so doloceni z dodatnim stevilcnim argumentom, ki je oznacen kot [=vrednost].
primer
parameter/kljuc[=vrednost]

Prav tako prazna mesta niso potrebna med posameznimi kljuci:
parameter/kljuc/kljuc
ime ukaza /kljuc/kljuc
primer:
SET /CRT=TI:
Ta ukaz pove sistemu DELTA/M, da je nas terminal zaslonski. TI: je ime enote za terminal, ki ga uporabljamo, to je tisti, preko katerega je bil ukaz vpisan. V tem primeru je CRT kljuc, ki doloca funkcijo ukaza SET.

Preden vnesete ukaz, pa moramo opraviti naslednje:
* preverimo, ce je terminal vkljucen.
* preverimo, ce je tipka <LOCAL/REMOTE> nastavljena na REMOTE.
* upostevamo navodila potrebna za nastavitev terminala in povezavo z racunalnikom(direktna ali telefonska linija).
* pritisnemo <CTRL/C>, da dobimo monitorski znak pripravljenosti (MCR>).

Monitorski znak pripravljenosti in znak pripravljenosti(>) oba oznacujeta, da je monitor pripravljen sprejeti vpis preko terminala. Ce vpisemo <CTRL/C>, ko uporabljamo kak drug sistem kot monitorski, lahko vnesemo nek monitorski ukaz. Sistem izvrsi ta ukaz in se vrne na nivo, kjer je bil pred vpisom <CTRL/C>.

1.2.1 ukaz <HELP>
HELP je edini ukaz(razen HELLO), ki ga lahko vnesemo na terminal preden se prijavimo. Ker so prve tri crke ukaza HELP enake kot pri HELLO, moramo vnesti ukaz HELP v celoti. Ko vnasamo ta ukaz,se na terminal izpise tekst, ki vam pove, kako se lahko prijavimo in kako vnasamo nadaljne ukaze. Celo ce ne potrebujemo takojsnje pomoci, vnesimo ta ukaz vsaj enkrat, tako da lahko ugotovimo znacaj razpolozljivih informacij na vasem sistemu.

1.2.2 ukaz <HELLO>
Ce hocemo zaceti delo na terminalu, moramo vnesti monitorski ukaz HELLO, da se prijavimo. Prijava na sistemu ima vec nalog:
* sistem preveri, ce smo pooblascen uporabnik in registrira naso uporabo sistema.
* omogoca nam dostop do sistema, dokler se ne odjavimo.
* vzpostavi vnaprej dolocene vrednosti in pogoje za delo s terminalom.

Na monitorski HELLO ukaz da računalnik sledeči odgovor:

```
>HEL[LO]<CR>
UIC ali priimek:[s,m]<CR>
                ali
                priimek<CR>
Geslo: geslo <CR>
```

[s,m] imenovan UIC
Sestavljen je iz dveh oktalnih stevil, ki pomenita našo številko pristopa v sistem. Prvo št.(s) pomeni za našo uporabniško skupino, drugo št.(m) pa je naša članska številka v skupini. Navadno sta obe števili ločeni z vejico in zaprta z oglatim oklepajem [s,m]. Ko dobimo naš UIC na sistemu z večuporabniško zaščito, se kreira seznam uporabniških datotek (UFD) z isto številko.

priimek
Namesto UIC-eja lahko vnesemo priimek. Sistem določi naš pravilni UIC iz vnesenega priimka.

geslo
Vsebuje 1-6 črkovnih znakov. Sistem obdrži informacijo vključujoč pravilno geslo za vsak UIC in priimek.
Dostopa do sistema ne moremo dobiti, če ne vnesemo gesla ustrezajočega UIC-eju ali priimku.
Sistem ne napiše na zaslon znakov, ki ste jih vnesli kot Geslo. To nam zagotavlja, da naše geslo ostaja zasebno in da nihče ne more uporabljati naših programov.

primer prijave:

```
>HEL
UIC ali priimek: DELTA <cr>
Geslo:           <cr>

      DELTA-M V1.2 BL26     SISTEM

Dobro jutro
07-Jan-82 10:11 vključen na terminl TT0:

Dobrodošli na DELTA-M V1.2 operacijski sistem
```

Ko sistem prejme pravilno geslo, terminal pokaze sporocilo, ki vkljucuje vrsto sistema, ter datum in cas, ko smo se prijavili. Kar je napisano v zgornjem primeru, se izpise tudi na terminal. Pazite:geslo ni vidno! Terminal nam izpise tudi razlicna sporocila, ki se ponavadi nanasajo na uporabo racunalnika. Znak pripravljenosti (>) v vrstici za sporocilom, pove, da je sporocilo koncano in da je monitor pripravljen sprejemati ukaze.
Naslednje sporocilo pove, da ze nekdo drug uporablja terminal:

```
HEL--vkljucen je drugi uporabnik
```

## MOZNOSTI HITREJSE PRIJAVE

Vecina uporabnikov hoce tipkati cim manj in zeli tudi hiter odgovor racunalnika. Naslednji odstavki opisujejo krajse poti za prijavo.

### Izkljucitev sistemskih sporocil

Ukaz HELLO vam omogoca posebne oblike UIC parametra, ki ne predvaja sporocil (datoteka sporocil pa je lahko napisana tudi tako, da so pomembna sporocila izpisana v vsakem primeru). Obicajna oblika UIC-eja je [g,m]. Ce vas sporocila ne zanimajo,nadomestite vejico z posevno crto (/):

```
[g/m]
```

HELLO dovoljuje stiri UIC oblike:

```
[g/m]
 g/m
[g,m]
 g,m
```

Obliki s posevno crto ne izpiseta sporocil,razen najpomembnejsih. Ko se prijavljate, oglati oklepaji niso potrebni. Toda kadar je UIC parameter kateregakoli drugega ukaza, mora biti v obliki [g,m].

### Izkljucitev vprasanja UIC ali priimek

Druga moznost skrajsanja prijave je, da napisete ukaz HELLO in UIC (ali priimek) v isto vrstico, uporabljajoc posevno crto kot locilo:

```
>HEL 201/312 <CR>
 Geslo:          <CR>
```

Ta ukaz izkljuci vprasanje UIC ali priimek

Avtomaticno izvajanje prijavne ukazne datoteke.
Ko se prijavimo na terminal, sistem avtomaticno priredi SY: vsakemu sistemskemu disku in potem isce po vasem UFD na SY datoteko LOGIN.CMD. Ce jo najde, izvede vse ukaze iz datoteke LOGIN.CMD.Ta oblika je primerna, ce pri vsakem prijavljanju uporabljamo iste ukaze.
Primer: Uporabljamo lahko prijavne ukaze, da nastavimo nas terminal kot zaslonski terminal in za sprejemanje malih crk. Lahko vkljucimo v datoteko tudi ukaz za izpis tekstovne datoteke opozoril za nas same (podobno kot izpise tekstovne datoteke z obvestili vsem prijavljenim uporabnikom). Ko zelimo zamenjati obvestilo, enostavno popravimo datoteko MYLOGIN.TXT. V tem primeru se na zaslonu prikaze naslednje sporocilo:

```
>@LOGIN.CMD
>SET /LOWER=TI:
>SET /CRT=TI:
>PIP TI:=MYLOG.TXT
10:00           OBISK PRI DIREKTORJU
11:00           ZMENEK Z DEKLETOM
12:30           POSLOVNI SESTANEK
>@<EOF>
```

## 1.3 SPOROCILA NAPAK PRI PRIJAVLJANJU

Ko monitor sprejme vpis, ki je napacen, oziroma ga ne prepozna, izpise na vas terminal sporocilo, kot na primer ce se poskusamo prijaviti na terminal, ki ze deluje. Sporocilo izpise monitor tudi v primeru, ko smo vtipkali UIC (lastno ime) ali geslo, ki ga sistem ne pozna:

```
>HEL
UIC ali priimek: 50,1
Geslo:
HEL -- napacen UIC
```

Ta poizkus prijave ni bil uspesen, ker sistem ni prepoznal priimka oziroma UIC -ja(CONRAK) ali pa ker nismo vnesli pravilnega gesla. Za odpravo napake ukrepamo po vsebini sporocila. Vsa sporocila monitorja so razlozena v knjigi "Monitorski ukazi DELTA/M".

Vsako sporocilo, ki ga program izpise na nas terminal, se zacne z imenom ukaza ali programa, ki nam posilja sporocilo. Ko opazimo napako med delovanjem sistemskega programa(npr.urejevalnik teksta) poiscemo obrazlozitev v dokumentaciji za ta program!

## 2.poglavje

## PRIPRAVA PROGRAMA

Da bi lahko izvedli nek program pod operacijskim sistemom DELTA/M, moramo:

* napisati izvorni program v datoteko
* prevesti izvorniprogram,da dobimo prevedeni program
* povezati prevedeni program z sistemom, da dobimo izvedljiv program
* sproziti izvajanje programa(z uporabo monitorskega ukaza RUN).

Vsaka od navedenih dejavnosti se izvaja z najmanj eno datoteko, ki predstavlja podrocje na zunanji pomnilni enoti (disku).

Ko smo se prijavili na terminal, nam sistem avtomaticno dovoli dostop do diska. Na tem disku so shranjene vse nase datoteke, razen ce izrecno hocemo drugace. (Ce nas sistem nima vecuporabniske zascite, imamo avtomaticen dostop do sistemskega diska, ki je dostopen vsem uporabnikom.)V tem poglavju se bomo naucili,kako uporabljati nas sistemski disk.

DELTA/M nam omogoca izvajanje programov napisanih v stevilnih racunalniskih jezikih. Vsak jezik ima svoj lastni sistemski program,ki se imenuje izbirnik ali prevajalnik. Ta sistemski program prevaja izvorni program v prevedeni program in preveri, ce je program napisan po pravilih za ta jezik.

Vsak jezikovni zbirnik ali prevajalnik lahko prevede le izvorne datoteke, napisane v tem jeziku: program napisan v pascalu mora biti preveden s pascalovim prevajalnikom. (Vec podatkov dobite v dokumentacijah za posamezne racunalniske jezike.)

Ceprav ta prirocnik prikaze nastanek programa v pascalu,lahko naredimo iste stopnje tudi pri programih pisanih v FORTRANu, COBOLu ali drugih racunalniskih jezikih. Edina razlika je v prevajalniku, ki ga uporabljamo.

## 2.1 VNASANJE IZVORNE DATOTEKE

DELTA/M vrsticni urejevalnik teksta(EDI) je sistemski program, ki ga uporabljate za pisanje izvornih datotek. (EDI lahko uporabljate tudi za tvorbo drugih vrst datotek, tako kot na primer tekstovne datoteke ali podatkovne datoteke.)

V tem poglavju bomo uporabili le nekaj EDI ukazov, s pomocjo katerih bomo vnesli in uredili izvorni program VSOTA.PAS.

### 2.1.1 uporaba urejevalnika EDI za vnasanje in urejanje izvornih datotek

EDI poklicemo na isti nacin kot monitorski ukaz(pritisnete na <CTRL/C>, da dobite monitorski odzivni znak in se prepricate, ce gre vas ukaz v monitor).

```
>EDI<CR>
```

EDI izpise programski odzivni znak:

```
EDI>
```

Za tvorbo nove datoteke z uporabo EDI oznacite ime in tip dateke v naslednji obliki:

```
EDI>ime datoteke.tip datoteke
```

ime datoteke
beseda, ki ima 1-9 crk

tip datoteke
kratica(3 crke), pred katero stoji pika(.). Kratica se navadno nanasa na vsebino datoteke. Na spodnji tabeli se nahaja nekaj standardnih tipov datotek za programiranje izvornih datotek v razlicnih jezikih:

| TIP | JEZIK |
|---|---|
| PAS | PASCAL |
| CBL | COBOL |
| BAS | BASIC |
| FTN | FORTRAN |
| MAC | MACRO |

Naslednji primer, ki ga bomo uporabljali skozi vse poglavje, prikazuje nacin pisanja izvornega programa v pascalu z imenom VSOTA.PAS.
VSOTA.PAS je program, ki zahteva, da natipkate na terminal 2 stevili, ki jih program sesteje in izpise vsoto na terminal.

```
      EDI><CR>
      >EDI VSOTA.PAS<CR>
      [CREATING NEW FILE]
      INPUT
```

Ko EDI sprejme ime nove datoteke, formira prazno datoteko s tem imenom in izpise dve vrstici kot v zgornjem primeru. EDI izpise INPUT in nam da vedeti, da je pripravljen sprejemati vpis na terminal. Vse kar vpisemo, razen kontrolnih znakov, postane del datoteke imenovane VSOTA.PAS. Nato lahko vnesemo izvorni program za VSOTA.PAS. Ko koncamo vsako vrstico z <CR>, EDI shrani vrstico v delovni spomin. Ko koncamo vpis, EDI izpise delovni spomin v datoteko VSOTA.PAS.

Z uporabo funkcijskih tipk(poslavje 1-1), lahko popravimo vse napake v tekoci vrsti preden koncamo z <CR>. Ko je vrstica enkrat koncana in napisana v delovni spomin, moramo za vsako spremembo uporabljati editorske ukaze.

Prav tako lahko vnesemo novo ime in tip datoteke v vrstico, v kateri smo poklicali EDI. Ta hitrejsa moznost tvorbe je prikazana v spodnjem programu. Bodimo pozorni na uporabo tipk:DELETE, CTRL/R IN CTRL/U:

```
>EDI VSOTA.PAS <CR>
[CREATING NEW FILE]
INPUT
   PROGRAM VSOTA; <CR>
   VAR K,L:INTEGER; <CR>
   BEGIN <CR>
      WRITE ('VNESI DVE STEVILI');<CR>
      READLN (K,L); <CR>
      WRITELN('VSOTA = ',K+L:10);<CR>
   END.<CR>
<CR>
*EX<CR>
[EXIT]
>
```

Ko smo vnesli zadnjo vrstico teksta programa, pritisnemo <CR> kot prvi znak v novi vrsti. EDI odgovori z izpisom zvezdice (*). Do zdaj je EDI sprejemal tekst za tvorbo nove datoteke(vhodni nacin). Po pritisku <CR> na zacetku vrste, pa je pripravljen na popravljanje napak(urejevalni nacin). Zvezdica(*) je editorski odzivni znak.

Ukaz EXCIT] ukaze EDI-ju, da izpise VSOTA.PAS na nase podrocje na disku in da vrne kontrolo monitorju.

Ukaz KILL zapre vhodno in izhodno datoteko in izbrise izhodno datoteko. Izloci tudi vse sledi tekocega popravljanja.

2.1.2 urejanje obstojece datoteke

Za urejanje obstojece datoteke z EDI, vnesemo enak ukaz, kot smo sa uporabljali za tvorbo nove datoteke:

```
>EDI VSOTA.PAS
```

Ker je kopija VSOTA.PAS ze na disku, EDI odgovori drugace:

```
>EDI VSOTA.PAS<CR>
[00008 LINES READ IN]
[PAGE 1]
*
```

EDI naredi kopijo VSOTA.PAS in je pripravljen da popravlja napake, kar nakaze z odzivnim znakom(*). Sporocilo [00008 LINES READ IN] nam pove stevilo vrstic, ki jih je EDI spravil v delovni spomin. Vrstice v delovnemu spominu lahko zacnemo urejati. Delovni spomin lahko vsebuje celotno izhodno datoteko, ali pa le njen del, odvisno od velikosti datoteke in delovnega spomina. Za dostop do teksta v tekocem delovnem spominu vnesemo ukaz RENEW! RENEW izpise delovni spomin v izhodno datoteko in ponovno napolni delovni spomin z naslednjim delom teksta.

Notranji vrsticni kazalec doloci vrstico v delovnem spominu, ki naj bo popravljena. Ko EDI bere v delovnem spominu, vrsticni kazalec pokaze vrstico, ki stoji takoj pred prvo vrsto teksta. To nam omogoca, da vkljucimo dodatno vrstico na zacetku. Kazalec lahko premaknemo z iskanjem dolocenega dela teksta, ali z uporabo ukazov, ki premikajo kazalec.

Namescanje in spreminjanje teksta:

EDI ukazi v spodnji tabeli nam omogocajo iskanje in spreminjanje teksta v datoteki. Vecino EDI ukazov lahko skrajsamo na eno ali dve crki. V nadaljevanju je del ukaza, ki ga ni treba vtipkati v oglatih oklepajih.

| ukaz | funkcija |
|---|---|
| L[OCATE] | poisce niz teksta v tekocem delovnem spominu. |
| C[HANGE] | zamenja en niz teksta z drugim. |
| N[EXT] | premakne vrsticni kazalec v naslednjo vrstico (EDI izpise zvezdico(*), toda ne izpise cele vrstice). |
| P[RINT] | izpise tekoco vrstico na vas terminal. |
| T[OP] | premakne vrsticni kazalec na zacetek tekocega delovnega spomina. |
| BO[TTOM] | premakne vrsticni kazalec na konec tekocega delovnega spomina. |
| <CR> | pokaze in izpise na vas terminal naslednjo vrstico datoteke. |
| <ESCAPE> | izpise prejsnjo vrstico na terminal |

Primer EDI popravljanja je prikazan spodaj s pojasnili, kako posamezen ukaz spreminja in namesca vrstice v datoteki in izpise na terminal

```
>EDI VSOTA.PAS<CR>
[00008 LINES READ IN]
[PAGE 1]
*LOCATE VPISI<CR>
  WRITE ('VPISI DVE STEVILI-M, N');
*
```

V zgornjem primeru ukaz LOCATE pokaze in izpise naslednjo vrstico datoteke (za tekoco vrstico) ki vsebuje besedo VPISI.

```
*CHANGE/VPISI/VTIPKAJ/<CR>
  WRITE ('VTIPKAJ DVE STEVILI-M, N');
*
```

V zgornjem primeru ukaz CHANGE spremeni besedo VPISI v VTIPKAJ. Posevne crte locijo star in nov niz teksta. Katerikoli znak, ki ni uporabljen v nizih, se lahko uporabi za omejitev nizov. EDI nato izpise popravljeno vrstico na vas terminal.

```
*NEXT<CR>
*
```

Ukaz NEXT pokaze na naslednjo vrstico v tekstu, vendar je ne izpise na terminal.

```
*PRINT<CR>
    K, L:INTEGER;
*
```

Ukaz PRINT izpise tekoco vrstico na vas terminal

```
*LOCATE (215)<CR>
[*EOB*]
```

Ce EDI doseze konec delovnega spomina, ne da bi nasel iskan niz znakov, izpise [*EOB*]. EDI vrsticni kazalec se na ukaz LOCATE premika po delovnem spominu samo naprej, nikoli pa ne nazaj.

```
*TOP<CR>
*
```

Ukaz TOP premakne vrsticni kazalec na zacetek delovnega spomina (v vrstico pred prvo vrstico teksta). EDI pri tem izpise editorski odzivni znak(*).

```
*<CR>
 BEGIN
*
```

Ko pritisnete na <CR> za zvezdico(*), vam izpise naslednjo vrstico na terminal. <CR> tudi premakne vrsticni kazalec v naslednjo vrstico. V bistvu zdruzuje ukaza NEXT in PRINT. V tem primeru kazalec pokaze prvo vrstico v delovnem spominu.

```
*EXIT<CR>
[EXIT]
>
```

Ukaz EXIT izpise vsebino delovnega spomina in ostanek vhodne datoteke v izhodno datoteko, zapre vhodno in izhodno datoteko in preide iz EDI v MCR. Namesto ukaza EXIT lahko uporabimo znak <CTRL/Z>, ki ga dobimo tako, da pritisnemo hkrati tipki CTRL in Z.

## Vstavljanje in brisanje teksta

EDI ukazi na spodnjih tabelah vam omogocajo vstavljanje in brisanje teksta v datoteki.

| ukaz | funkcija |
|---|---|
| I[NSERT] | vstavi eno ali vec novih vrstic v v datoteko. |
| A[DD] | doda tekst na konec ze obstojece vrstice. |
| D[ELETE] | izbrise tekoco vrstico. |
| <ESC> | pokaze in izpise prejsnjo vrstico. <ESC> pa ne izpise vrstice, ki ste jo ravnoar vnesli z ukazom INSERT. |
| R[ETYPE] | zamenja tekoco vrstico z novo. |
| LI[ST] | izpise na terminal vrstice, ki so ostale v delovnem spominu od tekoce vrstice do konca. Po tem ukazu je vrsticni kazalec na zacetku delovnega spomina. |
| TYPE n | izpise na vas terminal n vrstic, toda ne premakne vrsticnega kazalca. |

Primeri editorskega popravljanja:

```
>EDI VSOTA.PAS<CR>
[00008 LINES READ IN]
[PAGE 1]
*
```

EDI poisce zadnjo verzijo VSOTA.PAS. Ce hocete kaksno prejsnjo verzijo, morate ukazu dodati se stevilko verzije.

```
*INSERT<CR>
TA PROGRAM SESTEJE DVE STEVILI<CR>
<CR>
*<ESC>
[*BOB*]
<CR>
TA PROGRAM SESTEJE DVE STEVILI
*
```

Po INSERT ukazu je EDI v vhodnem nacinu. Vstavi tekst takoj pred prvo vrstico delovnega spomina. Po drugem <CR> pa je EDI zopet v urejevalnem nacinu. <ESC> izpise vrstico pred tekoco vrstico. V tem primeru ta vrstica zacenja delovni spomin [*BOB*]. <CR> premakne kazalec v vrstico, ki smo jo ravnokar ustavili in je zdaj prva vrstica teksta.

```
*LOCATE STEVILI<CR>
  WRITE ('VPISI DVE STEVILI-M, N');
*VSOTA !ZNAK ZA VPIS
*PRINT<CR>
  WRITE ('VPISI DVE STEVILI-M, N');!ZNAK ZA VPIS
*
```

Ukaz LOCATE poisce vrstico, kjer je iskani niz, VSOTA doda tej vrstici dolocen tekst in PRINT to vrstco v celoti izpise.

```
*LOCATE VSOTA<CR>
  WRITELN ('VSOTA JE', K+L);
*DELETE<CR>
*
```

Z ukazom LOCATE smo nasli vrstico, ki smo jo z DELETE potem izbrisali. Kazalec pa se premakne v naslednjo vrstico.

```
*<ESC>
  READLN (K, L);
*
```

Po ukazu <ESC> EDI pokaze in izpise prejsnjo vrstico v delovnem spominu na nas terminal.

```
*INSERT<CR>
  WRITE ('VPISI DVE STEVILI-M, N');!ZNAK ZA VPIS<CR>
<CR>
*
```

EDI vstavi novo vrstico. Drugi <CR> prevede EDI v urejevalni nacin.

```
*TOP<CR>
<CR>
TA PROGRAM SESTEJE DVE STEVILI
*RETYPE  PROGRAM IZPISE VSOTO DVEH STEVIL
*
```

Ukaz TOP premakne vrsticni kazalec na zacetek delovnega spomina, <CR> pokaze in izpise naslednjo vrstico, ki je v tem primeru prva vrstica delovnega spomina. Ukaz RETYPE pa zamenja tekoco vrstico z tekstom, ki sledi ukazu.

```
*LIST<CR>
```

Ukaz LIST izpise ves tekst od tekoce vrstice do konca delovnega spomina.

```
*EX<CR>
[EXIT]
>
```

Ko vnesete ukaz EXIT ali <CTRL/Z>, EDI izpise datoteko v nase podrocje na disku.

## 1.3 osnovni EDI ukazi

Tabela 2-1 prikazuje osnovne EDI ukaze. Ti ukazi vsebujejo vse funkcije, ki jih potrebjemo za preprosto urejanje.

Tabela 2-1

| ukaz | opis |
|---|---|
| A[DD] niz | doda niz tekoci vrsti |
| BO[TTOM] | premakne vrsticni kazalec na konec tekocega delovnega spomina |
| C[HANGE]/niz 1/niz 2 | zamenja niz 1 z nizom 2 v tekoci vrstici |
| <CR> | iz vhodnega nacina nas vrne v urejevalni nacin. Ko pa smo v urejevalnem nacinu, nam izpise naslednjo vrstico in premakne vrsticni kazalec v to vrstico |
| CTRL/Z | zapre vhodne in izhodne datoteke in konca urejanje |
| D[ELETE] | brise tekoco vrstico |
| <ESC> | pokaze in izpise prejsnjo vrstico |
| E[XIT] | zapre vhodne in izhodne datoteke in konca urejanje-ista funkcija kot <CTRL/Z> |
| I[NSERT] niz | vstavi nek niz pred naslednjo vrstico ali uvede vhodni nacin, ce je niz izpuscen |
| KILL | zapre vhodno in izhodno datoteko in izbrise izhodno datoteko |
| L[OCATE] niz | poisce naslednjo vrstico, ki vsebuje iskan niz. Iskanje ustavi na koncu tekocega delovnega spomina |
| N[EXT] | premakne vrsticni kazalec v naslednjo vrstico |
| P[RINT] | izpise tekoco vrstico na terminal |
| REN[EW] | izpise tekoci delovni spomin v izhodno datoteko in bere v naslednji delovni spomin tekst iz vhodne datoteke |

| ukaz | opis |
|---|---|
| R[ETYPE] | zamenja tekoco vrstico z dolocenim nizom |
| T[OP] | premakne vrsticni kazalec na zacetek delovnega spomina |
| TYPE n | izpise na terminal naslednjih n vrstic, vendar ne premakne vrsticnega kazalca |

## 2.2 PREVAJANJE IZVORNE DATOTEKE V PASCALU

Ko smo ustvarili izvorno datoteko v PASCALU, VSOTA.PAS v tem primeru, sledi prevajanje programa v obliko, ki jo lahkobere sistem (binarna). Tak program se imenuje prevedeni program (object).

### 2.2.1 tvorba prevedenega programa

Prevajanje zacnete s tem, da poklicete pascalov prevajalnik:

```
>PAS VSOTA.MAC=VSOTA.PAS<CR>
```

Ta ukaz pove prevajalniku, naj prevede izvorno datoteko VSOTA.PAS v prevedeni program imenovan VSOTA.MAC. Kot pokaze primer, oznacite od leve proti desni izhodno datoteko(VSOTA.MAC), enacaj(=) in vhodno datoteko(VSOTA.PAS).

POMNI!
Izhod iz procesa je vedno na levi strani enacaja, vhod v proces pa na desni.(za podroben opis datotek glejte 3.poglavje)

Pascalov prevajalnik je izjema med prevajalniki, ki jih pozna sistem DELTA/M : program moramo prevesti se s prevajalnikom za macro, medtem ko drugi prevajalniki program prevedejo direktno v prevedeno datoteko (.OBJ), zato je ponovno prevajanje nepotrebno ( in nemogoce ).
Pascalski program dodatno prevedemo z ukazom

```
>MAC VSOTA.OBJ=VSOTA.MAC
```

Pascalov prevajalnik ( s pomocjo prevajalnika MAC - prevajalnik za zbirni jezik ) spremeni tip datoteke iz izvorne datoteke (.PAS) v prevedeno datoteko (.OBJ). To pomeni, da ce ne priključimo imenu datoteke tipa datoteke, PASCAL poisce datoteko tipa .PAS z tem imenom. Ce jo najde, uporabi to datoteko kot vhodno. Pascal tvori datoteko tipa .OBJ kot izhod, ce ne dolocite tipa izhodne datoteke.

Vecina DELTA/M sistemskih programov in vsi prevajalniki delujejo na ta nacin. Zato obicajno uporabljamo standardne tipe datotek, kadarkoli je to mogoce!

Privzeti tipi datotek so prikazani na spodnji tabeli

Tabela 2-2

| Datoteka | | Privzeti tip datoteke |
|---|---|---|
| Vhod | Izhod | |
| izvorna datoteka | | .PAS |
| | izpisna datoteka | .LST |
| | prevedeni program | .OBJ |

Z uporabo privzete vrednosti lahko skrajsate zgornji ukaz:

```
>PAS VSOTA=VSOTA<CR>
```

Z dodatnimi stikali za vhodne in/ali izhodne datoteke lahko uvedete posebne prevajalne zahteve. Vsi primeri v tem delu vsebujejo prevajalne privzete vrednosti.

### 2.2.2 nastanek izpisne datoteke

Ko pascalov prevajalnik prevaja izvorni program v prevedeni program, lahko preskrbi tudi izpisno datoteko za ta program. Za nastanek izpisne datoteke vpisemo naslednji ukaz:

```
>PAS VSOTA, VSOTA=VSOTA<CR>
```

Drugi VSOTA na izhodni strani ukaza pove prevajalniku, da naj izvede tudi izpisno datoteko imenovano VSOTA.LST za izvorni program. Izpisna datoteka prikaze tiskovne napake na programu.

Ce sistem vsebuje urejevalnik vrste in izpis na vrsticni tiskalnik, prevajalnik avtomaticno izpise VSOTA.LST v vrsto vrsticnega tiskalnika. To avtomaticno tiskanje lahko preprecite z naslednjim ukazom:

```
>PAS VSOTA, VSOTA/-SP=VSOTA<CR>
```

Stikalo -SP pove prevajalniku naj ne izpise VSOTA.LST na tiskalnik.

Prav tako poslje prevajalnik kopijo VSOTA.LST na nas terminal. Ce nas sistem ne vkljucuje VSOTA.LST, prevajalnik shrani na disk samo datoteko.

Pascalov prevajalnik lahko naredi izpisno datoteko tudi brez tvorbe prevedene datoteke. Ta moznost pa nam onemogoca najti napake v nasem programu, brez tvorbe stevilnih verzij neuporabnih prevedenih programov. Za izvedbe same izpisne datoteke vnesemo naslednji ukaz:

```
>PAS )VSOTA=VSOTA<CR>
>
```

Potem ukazu je VSOTA.LST izpisan na ustreznem tiskalniku in shranjen na nasem delu diska.

Vstaviti pa moramo vejico pred dolocilo za izpisno datoteko v ukazu, drugace bo prevajalnik smatral VSOTA za prevedeni program, namesto za izpisno datoteko.

Naslednji ukaz izpise kopijo VSOTA.LST na nas terminal, vendar je ne izpise na tiskalnik ali shrani na disk:

```
>PAS VSOTA, TI:=VSOTA<CR>
```

2.2.3 aktiviranje pascalovega prevajalnika

Vcasih dobimo na ukaz OMS naslednje sporocilo

```
MCR -- TASK NOT IN SISTEM
```

To pomeni, da OMScalov prevajalnik ni v nasem sistemu. Aktiviramo sa z naslednjim monitorskim ukazom

```
>RUN $OMS<CR>
```

Ta ukaz pove sistemu naj vstavi pascalov prevajalnik v spomin in sa aktivira. Prevajalnik odgovori z odzivnim znakom:

```
PAS>
```

Nato vpisemo ukaz v pascalu kot v primeru:

```
PAS>VSOTA,VSOTA=VSOTA<CR>
PAS>^Z
```

Ko prevajalnik sprejme nase ukaze, znova izpise odzivni znak PAS> . V tem primeru lahko prevajamo naslednje programe v pascalu ali pa zapustimo prevajalnik.

## 2.3 TVORBA IZVEDLJIVEGA PROGRAMA

DELTA/M program za povezovanje spremeni prevedeni program narejen z jezikovnim prevajalnikom v samostojni izvedljivi program. Ta izvedljivi program ostane na disku, dokler nekdo ne vnese monitorskega ukaza za izvajanje programa.

### 2.3.1 Popolna oblika ukaza za povezavo

Popolna oblika za povezavo uporablja tri izhodne datoteke in vec vhodnih datotek. Tri izhodne datoteke so:

1. datoteka,ki vsebuje izvedljiv program (tip datoteke:TSK) in ki sa lahko instaliramo in izvajamo
2. datoteka ureditve spomina (MAP), ki vsebuje podatke o velikosti in lesi razlicnih delov programa.
3. datoteka definicij simbolov (STB), ki vsebuje podatke o definicijah globalnih simbolov.(Vecini povezovalnikovih operacij STB datoteka ni potrebna.)

Vhodne datoteke vsebujejo vse prevedene programe in programe, ki so potrebni za tvorbo posameznega izvedljivega programa. Popolna oblika ukaza je:

```
>TKB .TSK, .MAP, .STB= .OBJ(prevedena datoteka)
```

Tabela 2-3 prikazuje razlicne tipe datotek v ukazu za povezavo

Tabela 2-3

| Datoteka | | |
|---|---|---|
| Vhod | Izhod | Privzeti tipi dat. |
| prevedena dat. | | .OBJ |
| | izvajajoca datoteka | .TSK |
| | dat.ureditve spomina | .MAP |
| | dat.defin.simbolov | .STB |

Za program VSOTA.PAS vnesemo naslednji ukaz:

```
>TKB VSOTA.TSK,VSOTA.MAP,VSOTA.STB=VSOTA.OBJ,LB:[1,1]OMSLIB/LB<CR>
```

V tem ukazu je LB psevdonim za enoto, ki vsebuje knjiznico, [1,1] pa je UFD, kjer je knjiznica shranjena. OMSLIB je pascalova sistemska knjiznica.

Z uporabo povezovalnikovih privzetih vrednosti lahko ukaz skrajsamo na:

```
>TKB VSOTA, VSOTA, VSOTA=VSOTA, LB:[1,1]OMSLIB/LB<CR>
```

## 2.3.2 skrajsane oblike ukazov

Kot prevajalnik nam tudi povezovalnik omogoca izpustitev nekaterih izhodnih datotek ali usmeritev izhodnih datotek na terminal. Ce izpustimo datoteko v zacetku ali v sredini seznama izhodnih datotek, moramo vstaviti vejico na mesto imena datoteke, da oznacimo prazno polje. Na vhodni strani ukaza, pa uporabljamo vejico, da locimo eno vhodno datoteko od druge. Tabela 2-4 prikazuje tipe datotek, ki nastanejo pri povezovanju programov.

Tabela 2-4

| Ukaz | Nastale izhodne datoteke |
|---|---|
| >TKB.TSK,.MAP,.STB=.OBJ | vse tri izhodne datoteke |
| >TKB,,.STB=.OBJ | samo STB datoteka |
| >TKB,.MAP,.STB=.OBJ | MAP in STB datoteki |
| >TKB.TSK,,.STB=.OBJ | TSK in STB datoteki |
| >TKB.TSK,.MAP=.OBJ | TSK in MAP datoteki |
| >TKB.TSK=.OBJ | samo TSK datoteka |

### 2.3.3 večvrstični povezovalnikovi ukazi

Veliko vhodnih datotek lahko povzroči, da je povezovalnikov ukaz daljši od ene vrstice. V tem primeru pokličemo povezovalnik v naslednji obliki:

```
>TKB<CR>
TKB>
```

Ko smo vpisali <CR>, monitor aktivira povezovalnik, ki vrne TKB> odzivni znak. Povezovalnik izpiše ta znak po vsaki vrsti vnosa, dokler ne vpišete dve poševni črti (//) in <CR> v začetku vrste. Poševni črti končata delo povezovalnika in nas vrneta v MCR. Enovrstični/ukaz

```
>TKB
TKB>ime.TSK,ime.MAP,ime.STB=ime.OBJ,OBJ,OBJ<CR>
```

lahko vnesemo kot

```
TKB>ime.TSK,ime.MAP,ime.STB=ime1.OBJ
      TKB>ime2.OBJ,ime3.OBJ<CR>
TKB>//<CR>
>
```

### 2.3.4 izpisovanje datoteke ureditve spomina

MAP je ASCII datoteka, ki vsebuje podatke o velikosti in legi delov znotraj programa. Če nas sistem vsebuje urejevalnik vrste, povezovalnik sam po sebi pošlje datoteko na vrstični tiskalnik.

### 2.3.5 povezovalnikova stikala in dodatki

Povezovalnik vsebuje različna stikala in dodatke za kontrolo nastanka izvedljivega programa na disku.

## 2.4 IZVAJANJE PROGRAMA

Da bi sprožili izvajanje programa vtipkamo naslednji monitorski RUN ukaz:

```
>RUN VSOTA<CR>
```

Ta ukaz povzroči, da sistem

* uglotovi lego VSOTA.TSK na vašem sistemskem disku,
* pošlje kopijo izvedljivega programa v spomin,
* izvede program.

Ko smo uporabili to obliko ukaza RUN, izvedljiva oblika programa ostane na disku, pripravljena za ponovno izvajanje, doklerje ne izbrisemo z PIP-om.

Naslednji odstavek prikazuje tri uspesne izvedbe pascalovega programa VSOTA.PAS. VSOTA.PAS je interaktivni program, kar pomeni, da izpise sporocilo na nas terminal in nato caka, da vpisemo odgovor preden izvrsi izracun in izpise rezultat na nas terminal.

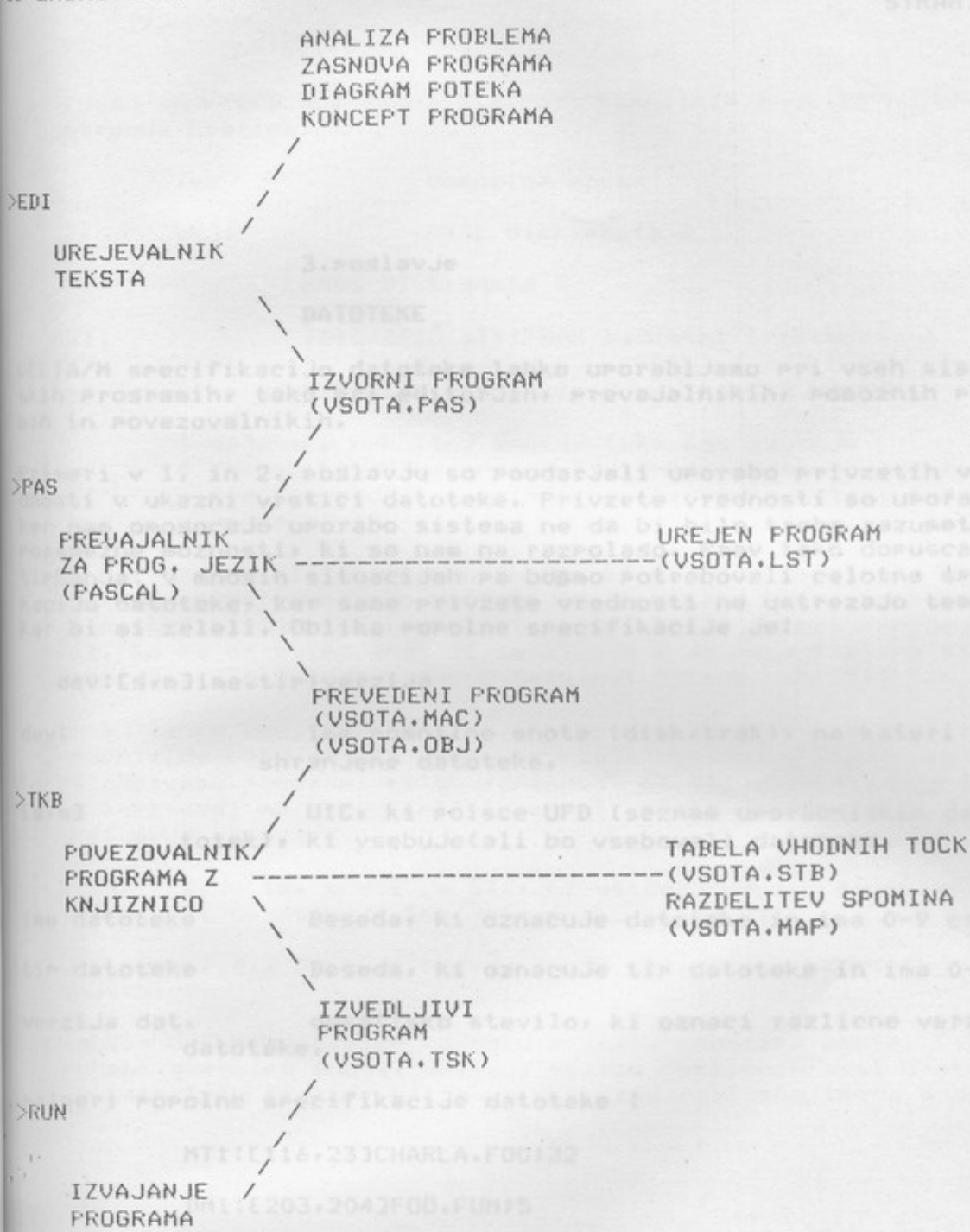
```
>RUN VSOTA<CR>
VNESITE DVE STEVILI  7,3<CR>
VSOTA= 10

>RUN VSOTA<CR>
VNESITE DVE STEVILI  522,628
VSOTA= 1150


>RUN VSOTA<CR>
VNESITE DVE STEVILI  9,16
VSOTA= 25
```

SPODNJA SHEMA PRIKAZUJE POT RAZVOJA PROGRAMA OD ZASNOVE DO IZVAJANJA






Monitorski ukazi uporabljeni za razvoj programa:

```
EDI VSOTA.PAS                          !kreiranje datoteke
PAS VSOTA,VSOTA=VSOTA                  !\ prevajanje programa
MAC VSOTA=VSOTA                        !/ v dveh stopnjah
TKB VSOTA=VSOTA,[1,1]OMSLIB/LB         !povezovanje s knjiznico
RUN VSOTA                              !izvajanje programa
```

Temu sledi se testiranje programa ter popravljanje morebitnih napak. Seveda je potrebno po vsakem popravljanju izvornega programa program ponovno prevesti in povezati s knjiznico.

# 3.poglavje

# DATOTEKE

DELTA/M specifikacijo datoteke lahko uporabljamo pri vseh sistemskih programih, tako pri editorjih, prevajalnikih, pomoznih programih in povezovalnikih.

Primeri v 1. in 2. poglavju so poudarjali uporabo privzetih vrednosti v ukazni vrstici datoteke. Privzete vrednosti so uporabne, ker nam omogocajo uporabo sistema ne da bi bilo treba razumeti vse posamezne moznosti, ki so nam na razpolago. Prav tako dopusca manj tipkanja. V mnogih situacijah pa bo$mo potrebovali celotno specifikacijo datoteke, ker same privzete vrednosti ne ustrezajo temu, kar bi mi zeleli. Oblika popolne specifikacije je:

dev:[g,m]ime.tip;verzija

dev: Ime pomnilne enote (disk,trak), na kateri so shranjene datoteke.

[g,m] UIC, ki poisce UFD (seznam uporabniskih datotek), ki vsebuje(ali bo vseboval) datoteko.

ime datoteke Beseda, ki oznacuje datoteko in ima 0-9 crk.

tip datoteke Beseda, ki oznacuje tip datoteke in ima 0-3 crke.

verzija dat. desetisko stevilo, ki oznaci razlicne verzije iste datoteke.

primeri popolne specifikacije datoteke :

MT1:[116,23]CHARLA.FOO;32

DM1:[203,204]FOO.FUM;5

2.poglavje opisuje ime in tip datoteke. To poglavje pa prikaze ime pomnilne enote in UFD.

## 3.1 IME POMNILNE ENOTE

Pomnilne enote oznacuje disk ali trak na katerem obstaja posamezna datoteka. Ime je sestavljeno iz dveh crk in eno ali dvo mestnega oktalnega stevila, ki mu sledi dvopicje (:). Ce ime ne vkljucuje stevilke, sistem uporabi stevilko 0.

Spodaj so prikazana imena fizicnih pomnilnih enot in njihove ustrezne kratice.

| Ime | Pomnilna enota |
|---|---|
| DK2: | RK05 disk,enota 2 |
| DM1: | RK06 DISK,enota 1 |
| MT1: | TU10,TE10 ali TS03 magnetni trak,enota 1 |

Ime pomnilne enote spada v eno izmed naslednjih vrst:

* dejanska pomnilna enota, taka kot zgornje
* pseudo pomnilna enota, ki lahko predstavlja katerokoli fizicno enoto, z ozirom na to, preko kateri terminal je vnesena

Na primer ime TI: je psevdo enota, ki kaze na terminal, na katerem je bil vnos. Ko vnesemo iz terminala 23 na nasem sistemu, je nasa fizicna vhodna enota terminal TT23: in nas psevdo terminal TI:. Ko pa opravimo vnos iz terminala 6 je nasa fizicna stevilka terminala TT6:, toda psevdo terminal ostane isti TI:

Torej lahko napisemo program, ki bo poslal izpis na TT23:, kar pomeni, da bomo morali vedno sprozitи program s TT23:. Lahko pa napisemo program, ki bo izpisoval na TI: ,kar pomeni, da bo izpisoval na katerikoli terminal, preko katereg smo sprozili izvajanje programa.

Druga psevdo ime enote je SY:, ki ustreza nasemu sistemskemu disku. Vse datoteke v tem prirocniku so na SY:.

Odstavek 2.3.1 opisuje psevdo enoto LB:, enoto ki vsebuje knjiznice.

Tabela 3-1 predstavlja najbolj pogoste pomnilne enote. Crka n pomeni stevilko enote. Obstaja veliko razlicnih vrst diskov in magnetnih trakov, ki imajo ustrezno razlicna imena enot.

Tabela 3-1

| Naprava | Kratica |
|---|---|
| Disk | DBn:<br>DFn:<br>DKn:<br>DLn:<br>DMn:<br>DPn:<br>DRn:<br>DSn:<br>DXn:<br>DYn: |
| Vrsticni tiskalnik | LP: |
| Magnetni trak | MMn:<br>MSn:<br>MTn: |
| Pseudo terminal | TI: |
| Terminal | TTn: |
| Pseudo sistemska enota | SY: |

Ker imena SY:,TI:,TT: in LP: predstavljajo bolj vhodne in izhodne enote, kot medije za shranjevanje, se ponavadi ne pojavljajo v popolni ukazni vrstici. Ko naletimo na eno izmed teh enot v ukazni vrstici, stoji ime enote samostojno na eni strani enacaja. Na primer, naslednji ukaz poslje kopijo VSOTA.PAS iz nasega sistemskega diska na nas terminal:

```
>PIP TI:=VSOTA.PAS<CR>
    {PROGRAM IZPISE VSOTO DVEH STEVIL}
     PROGRAM VSOTA;
     VAR K,L:INTEGER;
     BEGIN
          WRITE ('VTIPKAJ DVE STEVILI');{VNOS PODATKOV}
          READLN (K,L);
          WRITELN('VSOTA=',K+L:10);{IZPIS REZULTATA}
     END.
>
```

## 3.2 DIREKTORIJ UPORABNISKIH DATOTEK (UFD)

Ko se prijavljamo, uporabljamo UIC ali pa priimek. Ta UIC poisce UFD na nasem sistemskem disku (SY:), potem ko ugotovi, ce smo pooblascen uporabnik. UFD je datoteka, ki vsebuje imena vseh datotek shranjenih na nasem seznamu. Noben sistemski program ne more poiskati datoteke, ce ne pozna UFD na katerem je datoteka oznacena.

Izraza UIC in UFD se mnogokrat zamenjujeta, vendar UIC oznacuje porabnika, UFD pa seznam.

Ko potrebujemo datoteko shranjeno v drugih seznamih, lahko uporabimo monitorski SET ukaz, ki spremeni nas UFD (SET/UIC=[g,m]) ali pa oznacimo UFD v specifikaciji datoteke. Monitorski SET ukaz spremeni UFD. Vendar nobena izmed ukazov ne spremeni UIC, s katerim smo se prijavili. Na primer, ce bi radi prepisali na nas terminal datoteko iz drugega UFD, vnesemo naslednji ukaz:

```
>PIP TI:=[302,200]CONRAD.MAC<CR>
```

Ta ukaz povzroci, da se datoteka CONRAD.MAC ki je na nasem sistemskem disku v UFD[302,200] izpise.

V sistemu brez vecuporabniske zascite UFD ustreza UIC-eju, dolocenem v zadnjem SET /UIC ukazu, izpisanem iz terminala, ki ga uporabljate.

Za izpis tekocega UIC-eja na katerikoli sistem, vnesemo ukaz:

```
>SET /UIC<CR>
```

Ko vnesemo ta ukaz, na da bi dolocili UIC, sistem odgovori z:

```
UIC=[g,m]
```

# 4.poglavje

## PIP IN UREJEVALNIK VRSTE

V tem poglavju sta opisna dva pomozna programa: PIP in urejevalnik vrste.Prikazana je njuna uporaba za prepisovanje,iskanje, brisanje datotek, ter za izpis na vrsticni tiskalnik ali na terminal.

### 4.1 PIP

S PIP-om lahko na DELTA/M sistemu z uporabo primernih stikal za izvajmo naslednje operacije:

* prepisovanje datotek iz enega na drug UFD ali enoto
* brisanje datotek
* preimenovanje datotek
* izpisovanje seznama datotek
* brisanje vseh starih verzij

Vsaka enota, ki se nanasa na specifikacijo datoteke, mora biti prijavljena kot javna(dostopna vsem uporabnikom) ali rezervirana le za nas(samo na sistemih z vec uporabnisko zascito) in montirana. Ce uporabimo ukaz PIP pri katerikoli enoti, ki ni montirana ali je rezervirana za drugega uporabnika, PIP izpise naslednje sporocilo:

```
PIP--DEVICE NOT MOUNTED/ALLOCATED
  dd:[g,m]
```

Rezerviranje, montiranje in tvorba UFD so monitorske funkcije izvrsene z naslednjimi monitorskimi ukazi:

* ALLOCATE
* MOUNT
* UFD

V procesu pisnja, urejanja, prevajanja in povezovanja VSOTA.PAS program tvori DELTA/M stevilne datoteke:

* izvorni program (VSOTA.PAS;1)
* urejena verzija programa (VSOTA.LST;2)
* prevedeni program (VSOTA.OBJ;1)
* izvedljivi progrm (VSOTA.TSK;1)
* datoteka ureditve spomina (VSOTA.MAP;1)
* datoteka definicij simbolov (VSOTA.STB;1)
* razlicne verzije zgornjih datotek, ce ste ponovili katerokoli fazo priprave programa

Vse te datoteke ostanejo na disku, dokler jih sami ne izbrisemo. DELTA/M ne izbrise starih verzij datotek.

PIP lahko poklicemo iz monitorja na dva nacina:

1. Enovrsticna oblika, ki izvrsi en PIP ukaz in vrne kontrolo monitoju

```
>PIP /LI<CR>

DIRECTORY DB0:[301,314]
14-FEB-81  09:10

VSOTA.PAS;3      1.      22-FEB-81 11:40
VSOTA.PAS;2      2.      22-FEB-81 12:00
VSOTA.PAS;1      1.      19-FEB-81 11:32
```

2. Oblika, ki da kontrolo PIP-u in nam omogoca izvedbo stevilnih ukazov

```
>PIP<CR>
PIP> /LI<CR>

DIRECTORY DB0:[301,314]
14-FEB-81  09:10

VSOTA.PAS;3      1.      22-FEB-81 11:40
VSOTA.PAS;2      2.      22-FEB-81 12:00
VSOTA.PAS;1      1.      19-FEB-81 11:32

TOTAL OF 4./11. BLOCKS IN 3. FILES

PIP> VSOTA.PAS /PU<CR>
PIP>^Z
>
```

To poglavje uporablja enovrsticne oblike(stevilka 1), ki prikazujejo PIP ukazne operacije. Vsak prikazan ukaz pa je lahko izveden tudi v drugi obliki(stevilka 2)

## 4.2 PIP UKAZNA VRSTICA IN PRIVZETE VREDNOSTI

Oblika PIP ukazne vrstice je razlicna za vsako posamezno funkcijo. Na splosno pa PIP stikala delajo na seznamu specifikacije datotek. Kot privzeto vrednost za ta ukaz uporablja PIP v ukazni vrstici zadnjo navedeno vrednost.

Vnesemo lahko stevilne specifikacije datotek v eni sami ukazni vrstici, ki jih locimo z vejico. Stikalo sledi zadnji specifikaciji datoteke.

V naslednjem primeru bo PIP izbrisal datoteke z imeni:ARRAY.PAS;1 MULT.PAS;1 in HEX.PAS;1.

```
>PIP ARRAY.PAS;1,MULT.PAS;1,HEX.PAS;1 /DE<CR>
```

Pri stikalu DELETE moramo pri vsaki posamezni specifikaciji navesti tudi stevilko verzije datoteke, ki jo hocemo izbrisati.

### 4.2.1 izpis datoteke na nas terminal

Za izpis kopije VSOTA.PAS na nas terminal vnesemo PIP ukaz v naslednji obliki:

```
>PIP TI:=VSOTA.PAS<CR>
```

Ukaz izpise naslednje:

```
{PROGRAM IZPISE VSOTO DVEH STEVIL}
 PROGRAM VSOTA;
 VAR K,L:INTEGER;
 BEGIN
      WRITE ('VTIPKAJ DVE STEVILI');{VNOS PODATKOV}
      READLN (K,L);
      WRITELN('VSOTA=',K+L:10);{IZPIS REZULTATA}
 END.
```

Tabela 4-1 vsebuje PIP stikala, ki so opisana v tem poglavju.

Tabela 4-1

| stikalo | pomen | opis |
|---|---|---|
| /LI | List | Izpise imena zahtevanih datotek, ki so v dolocenem UFD. |
| /BR | Brief | Izpise skrajsano verzijo seznama narejenega z ukazom /LI. |
| /DE | Delete | Izbrise doloceno datoteko. |
| /PU | Purge | Izbrise vse razen zadnje datoteke (datoteke z najvisjo stevilko verzije). |
| /SP | Spool | Poisce in izpise doloceno datoteko na vrsticni tiskalnik. |
| /RE | Rename | Preimenuje datoteko. |
| /CO | Contiguous | Prepise datoteko v povezan blok na izhodni datoteki. |
| /TR | Truncate | Odreze neuporabljene bloke na koncu datoteke. |
| /CD | Creation Date | Prepise datoteko z datumom,ko je bila narejena,ne pa z tekocim datumom. |
| /SD | Selective Delete | Izbrise doloceno datoteko,potem ko nas za vsako posamezno datoteko vprasa,ce jo zelimo izbrisati. |

4.2.2 /LI-izpise seznam uporabniskih datotek

Vse datoteke so napisane na nasem UFD.PIP /LI ukaz izpise kot v primeru:

```
>PIP /LI

DIRECTORY DL1:[50,1]
7-JAN-82 11:30

VSOTA.PAS;1              1.          07-JAN-82 10:28
VSOTA.PAS;2              1.          07-JAN-82 10:39
VSOTA.PAS;3              1.          07-JAN-82 10:43
VSOTA.OBJ;1              2.          07-JAN-82 10:57
VSOTA.MAC;1              2.          07-JAN-82 10:58

TOTAL OF 7./25.  BLOCKS IN 5. FILES
```

S PIP /LI ukazom dobimo naslednje informacije o datotekah:

* Fizicna enota na kateri so spravljene datoteke in UIC, ki sa vsebuje seznam. Ime enote je privzet sistemski disk(SY:), UIC pa je privzet UFD.
* Datum in cas, ko je PIP napravil izpis seznama.
* Ime, tip in stevilko verzije za vsako datoteko.
* Stevilo blokov, ki jih zasede posamezna datoteka. Blok ima 512 bytov (256 besed).
* Znak, ki pokaze ali je datoteka povezana(contiguous-C), nepovezana(ni znaka) ali zaklenjena(locked-L). Zaklenjena datoteka ponavadi vsebuje pokvarjene podatke. Do njih pridemo z uporabo PIP ukaza za odklepanje(unlock): PIP ime datoteke/UN.
* Datum in cas, ko je bila datoteka narejena.
* Vsota vseh uporabljenih blokov in vsota vseh dodeljenih (rezerviranih blokov.

PIP v povezavi s stikali za kontrolo nam da lahko tudi drugace oblikovan seznam. BR stikalo lahko povzroci, da PIP ustvari seznam v skrajsani obliki. Ukaz PIP /BR izpise seznam v naslednji obliki:

```
>
>PIP /BR


DIRECTORY DL1:[50,1]

VSOTA.PAS;1
VSOTA.MAC;1
VSOTA.PAS;2
VSOTA.PAS;3
VSOTA.OBJ;1
VSOTA.TSK;1
```

FU stikalo povzroci, da PIP izpise seznam v popolnejsi obliki. Primer:

```
DIRECTORY DL1:[50,1]
19-FEB-82 12:47

VSOTA.PAS;1          (133,37)        1./5.                19-FEB-82 12:42
  [50,50]  [RWED,RWED,RWED,R]
VSOTA.MAC;1          (136,40)        2./5.                19-FEB-82 12:43
  [50,50]  [RWED,RWED,RWED,R]
VSOTA.PAS;2          (140,40)        1./5.                19-FEB-82 12:44
  [50,50]  [RWED,RWED,RWED,R]
VSOTA.PAS;3          (110,67)        1./5.                19-FEB-82 12:45
  [50,50]  [RWED,RWED,RWED,R]
VSOTA.OBJ;1          (114,32)        1./1.                19-FEB-82 12:45
  [50,50]  [RWED,RWED,RWED,R]    19-FEB-82 12:46(2.)
VSOTA.TSK;1          (141,35)       13./13.          C    19-FEB-82 12:46
  [50,50]  [RWED,RWED,RWED,R]    19-FEB-82 12:46(2.)
```

Ta oblika nam da naslednje informacije(poleg informacij, ki so bile ze vpisane):

* Stevilo za identifikacijo datoteke, ki se sestoji iz dveh stevil(1437,27). S tem prav tako lahko najdete datoteko v seznamu.
* Stevilo dodeljenih(rezerviranih) blokov(v nasprotju z zasedenemi bloki).
* Varovalni znak, ki pove, kako lahko datoteka postane dostopna tudi ostalim uporabnikom.
* Datum in cas, ko je bila datoteka zadnjic popravljana in stevilo poprav.

Po izpisu seznama v katerikli obliki, PIP vrne kontrolo monitorju.

### 4.2.3 izpis informacij o posameznih datotekah

PIP nam omogoca, da dobimo informacije o eni sami datoteki ali o doloceni skupni datotek v seznamu. Na primer, ce bi radi vedeli, koliko verzij VSOTA.PAS obstaja, uporabimo ukaz:

```
>PIP VSOTA.PAS;* /LI<CR>
```

Kot pokaze primer, je specifikacija datoteke vedno pred stikalom. Ukaz povzroci, da PIP izpise seznam vseh verzij datoteke imenovane VSOTA.PAS. Zvezdica(*) na mestu stevilke verzije pove PIP-u, da naj poisce vse verzije datoteke.

Zvezdica na enem ali vec poljih specifikacije datoteke stoji namesto vsega. To imenujemo tudi posplosena specifikacija datoteke. PIP omejuje uporabo zvezdice v naslednjih primerih:

1. prepis posamezne datoteke
2. izpis seznama
3. ko prepisujete vec datotek, morate uporabiti za specifikacijo izhodne datoteke *.*;* ali privzeto vrednost.
4. Pri stikalih Rename in Enter se lahko namesto katerega koli polja uporabljajo zvezdice(polje ostane isto kot pri izhodni datoteki) lahko pa dolocimo novo ime tega polja.

V vseh primerih, kjer PIP dovoljuje zvezdico v specifikaciji izhodne datoteke, oblika [*,*] oznaci, da je izhodni UIC isti kot vhodni UIC.

Sledeci ukazi pokazejo uporabo zvezdice in njen pomen:

1. *.*;* pomeni vse verzije vseh datotek v tekocem UFD
2. *.DAT;* pomeni vse verzije vseh datotek tipa DAT v tekocem UFD.
3. TEST.*;* pomeni vse verzije vseh tipov datotek z imenom TEST v tekocem UFD.
4. *.* pomeni najnovejso verzijo vseh datotek v tekocem UFD.
5. *.DAT pomeni najnovejso verzijo vseh datotek tipa .DAT v tekocem UFD.
6. TEST.* pomeni najnovejso vseh tipov datotek z imenom TEST v tekocem UFD.
7. TEST.DAT pomeni najnovejso verzijo datoteke TEST.DAT v tekocem UFD.
8. [*,*] pomeni vse skupine UIC s clansko stevilko od 1-377
9. [n1,*] pomeni vse UIC clanske stevilke v skupini n1.
10. [*,n2] pomeni vse skupine UIC s clansko stevilko n2.

### 4.2.4 brisanje datotek

Ko enkrat vemo katere datoteke vsebuje nas UFD, se lahko odlocimo, katere datoteke bomo izbrisali. Na primer, ce hocemo ohraniti le najnovejso verzijo vsake datoteke v ostavku 4.2.2 bomo izbrisali naslednje datoteke:

```
VSOTA.OBJ;1
VSOTA.PAS;1
VSOTA.PAS;2
```

Za brisanje teh datotek s stikalom PIP /delete vpisemo naslednji ukaz:

```
>PIP VSOTA.OBJ;1,VSOTA.PAS;1,VSOTA.PAS;2 /DE<CR>
>
```

Stikalo delete zahteva ali stevilko verzije ali pa zvezdico(*) na mestu verzije. Uporaba zvezdice je neprimerna kadar nas UFD vsebuje datoteke z istim imenom, toda razlicnimi tipi in verzijami, ki jih nocemo izbrisati. V tem primeru moramo tocno dolociti, katero datoteko hocemo izbrisati.

### 4.2.5 brisanje vseh verzij datotek razen zadnje

Ko hocemo izbrisati vse datoteke razen tiste z najvisjo stevilko verzije, uporabimo namesto stikala delete stikalo /PURGE. Sledeci purge ukaz ima isto funkcijo kot zgornji delete ukaz:

```
>PIP VSOTA.* /PU<CR>
>
```

Ukaz purge se ne nanasa na datoteke, ki imajo le eno verzijo. Specifikacija datoteke za stikalo purge ne vsebuje mesta za verzijo. Ko smo vnesli zgornji ukaz, lahko vnesemo naslednji ukaz, s katerim lahko pogledamo, katere datoteke so vam ostale v seznamu.

```
>PIP /LI<CR>


 DIRECTORY DL1:[50,1]
19-FEB-82 12:48

VSOTA.MAC;1          2.          19-FEB-82 12:43
VSOTA.PAS;3          1.          19-FEB-82 12:45
VSOTA.OBJ;1          1.          19-FEB-82 12:45
VSOTA.TSK;1         13.       C  19-FEB-82 12:46

TOTAL OF 17./24. BLOCKS IN 4. FILES
```

Kot kaze izpis, vsebuje UFD samo najvisje stevilke verzij vsake datoteke.

### 4.2.6 selektivno brisanje datotek /SD

PIP ukaz za selektivno brisanje zahteva od nas odgovor preden datoteko dokoncno izbrise. Mozni odgovori (<CR>,^Z,Y,N,Q,G) povzroce naslednje:

| Ukaz | Operacija |
|---|---|
| Y <CR> | izbrise datoteko in nadaljuje |
| Y ^Z | izbrise datoteko in gre v monitor |
| N <CR> | ohrani datoteko in nadaljuje |
| N ^Z | ohrani datoteko in gre v monitor |
| <CR> | ohrani datoteko in nadaljuje |
| ^Z | ohrani datoteko in gre v monitor |
| Q <CR> | ohrani datoteko in se vrne ukazni nacin |
| Q ^Z | ohrani datoteko in gre v monitor |
| G <CR> | izbrise to datoteko in vse ostale kandidate za izbris,izpise izbrisane datoteke in se vrne v ukazni nacin |
| G ^Z | izbrise to datoteko in vse ostale kandidate za izbris,izpise izbrisane datoteke in gre v monitor |

Ukaz za selektivno brisanje lahko vnesemo tudi v naslednji obliki:

```
PIP VSOTA.*;* /SD<CR>
```

PIP odgovarja:

```
DELETE
```

```
>
>PIP VSOTA.*;*/SD
DELETE FILE    DL1:[50,1]VSOTA.MAC;1          [Y/N/G/Q]? N
DELETE FILE    DL1:[50,1]VSOTA.PAS;3          [Y/N/G/Q]? Y
DELETE FILE    DL1:[50,1]VSOTA.OBJ;1          [Y/N/G/Q]? N
DELETE FILE    DL1:[50,1]VSOTA.TSK;1          [Y/N/G/Q]? N
>
```

Glede na nas odgovor,PIP zbrise VSOTA.PAS;1, toda ne zbrise VSOTA.OBJ;1.

### 4.2.7 Prepisovanje datotek

Prepisovanje datotek je privzeta funkcija PIP-a; to pomeni da PIP izvede prepis operacije,ce vnesete PIP ukazno vrstico brez stikal.Na primer, v naslednjem ukazu prepise datoteko VSOTA.MAP iz vasega UFD na SY: v vas UFD na DK:

```
PIP DK:=VSOTA.MAP<CR>
```

Ukaz vsebuje poziv PIP-u, kateremu sledi specifikacija datoteke v obliki

```
izhodna datoteka=vhodna datoteka
```

vhodna datoteka: datoteka, ki naj bo prepisana
izhodna datoteka: nova kopija datoteke

Ko vnesemo UFD, ime, tip in/ali stevilko verzije datoteke v izhodno datoteko, PIP prepise ime, tip in verzijo datoteke na ustrezna polja v izhodni datoteki. Preden lahko prepisemo datoteko v seznam v drugi enoti, mora direktorij na tej enoti ze obstajati. V sistemu vecuporabniske zascite ze avtomaticno obstaja seznam na SY:. V sistemu brez vecuporabniske zascite pa seznam ne nastane avtomaticno;ce izhodni enoti doloceni v zgornjem primeru ne pripisemo UFD, ki ustreza nasemu UIC,PIP vrne sporocilo:

### 4.2.8 Preimenovanje datotek

PIP /rename stikalo nam omogoca preimenovanje obstojecih datotek. Na primer:

```
>PIP VSOTADVE.*;*=VSOTA.*;* /RE<CR>
>
```

Ta ukaz pove PIP-u naj spremeni imena vsem tipom in verzijam datotek z imenom VSOTA v ime VSOTADVE. Vnesti moramo ali stevilo ali zvezdico namesto vhodne in izhodne verzije. Zvezdice v izhodni specifikaciji datoteke povedo,da sta tip in verzija enaka kot v vhodni datoteki. Seznam vsebuje zdaj naslednje datoteke:

```
DIRECTORY DL1:[50,1]
19-FEB-82 12:49

VSOTADVE.MAC;1      2.          19-FEB-82 12:43
VSOTADVE.OBJ;1      1.          19-FEB-82 12:45
VSOTADVE.TSK;1     13.      C   19-FEB-82 12:46

TOTAL OF 16./19. BLOCKS IN 3. FILES
```

Ne moremoe pa preimenovati same enote, temvec je treba datoteko prepisati iz ene enote v drugo. Ce hocemo preimenovati datoteko in jo prepisati na drugo enoto, vnesemo novo ime v izhodno specifikacijo datoteke v COPY ukazni vrstici. Naprimer:

Ta ukaz pove PIP-u naj prepise vse tipe in verzije z imenom VSOTA, ki so spravljene v vasem UFD na SY: v ekvivalentni UFD na DK: disku,enota 0, kjer naj se imenuje VSOTADVE.

## 4.3 UREJEVALNIK VRSTE

Uporabniki sistema pogosto potrebujejo kopije datotek, ki so izpisane na vrsticnem tiskalniku. Ce hoce prevec uporabnikov naenkrat delati na istem vrsticnem tiskalniku, sistemski program imenovan urejevalnik vrste doloci vrstni red uporabe glede na cas vnesenega ukaza in ostale faktorje ter nam izpise informacijo o mestu nase datoteke v vrsti.

### 4.3.1 tiskanje datotek

Ce hocemo uvrstiti nase datoteke, uporabimo PRINT ukaz urejevalnika vrste.Primer:

```
>PRINT VSOTA.MAP<CR>
```

Pri ukazu PRINT sistem avtomaticno privzame ime fizicne enote LP: in ga ni treba navesti pri opisu datoteke.Ta ukaz zaprosi urejevalnik vrste naj vnese VSOTA.MAP v vrsto datotek, ki cakajo na tiskanje. Navadno se tiska tista datoteka, ki je prva v vrsti.Sistem nam omogoca tudi tiskanje po dolocenem casu ali datumu.

### 4.3.2 izpisovanje datotek v vrsti

Da preverimo, ce je dolocena datoteka v vrsti za tiskanje, vnesimo naslednji ukaz:

```
QUE /LIST<CR>
```

Imena vseh datotek,ki cakajo na tiskanje, bodo izpisana na nas terminal v naslednji obliki:

```
>QUE /LI
** PRINT QUEUES **
PRINT   => LP0:
  [5,5]      PRINT   (2000,1)                    ACTIVE ON LP0:
          DL1:[5,5]KNJIGA.UCB;14
          DL1:[5,5]FORMEK.UCB;1
          DL1:[5,5]DELTAMZAC.UCB;15
```

1. oznaci datoteko,ki se ravnokar tiska, njeno indentifikacijsko stevilko,ime in tiskalnik.
2. izpise imena datotek, ki cakajo v vrsti (urejene so po vrsti kot bodo tiskane).

# Dodatek

## PRIMER PROGRAMA V PASCALU

Ta dodatek nam prikaze razvoj programa v pascalu in manipulacijo s koncnimi datotekami. Prikaze nam, kako lahko hitro tvorimo razlicne verzije iste datoteke in kako lahko odstranimo stare

```
 HEL 50,7/DELTA

      DELTA-M V1.2 BL26   SISTEM
Dober vecer
22-Feb-82 20:54 vkljucen na terminal TT0:

Dobrodosli na DELTA-M V1.2 operacijskem  sistemu


Po koncanem delu pursajte svoje direktorije in jih ne scitite
ker bodo le ti zbrisani !!!!!


>
>PIP /LI
PIP -- NO SUCH FILE(S)

>EDI VSOTA.PAS
[CREATING NEW FILE]
INPUT
PROGRAM AAA(INPUT,OUTPUT);
VAR A,B: INTEGER;
BEGIN
     READLN (A,B);
      WRITELN (A*B:10)
END.

*^Z
[EXIT]
```

```
>PAS VSOTA,TI:=VSOTA

AAA    OMSI Pascal V1.2C RSX  22-Feb-82 20:56 Site # 0000 Page 1
       D  E  L  T  A            -    o t r o s k i   v r t e c   -

  1                        PROGRAM AAA(INPUT,OUTPUT);
  2                        VAR A,B: INTEGER;
  3                        BEGIN
  4     1      1      1          READLN (A,B);
  5     2      1      1           WRITELN (A*B:10)
  6                        END.

Errors detected:  0
Free memory:  4067  words
>EDI VSOTA.PAS
[00006 LINES READ IN]
[PAGE    1]
*PL *
     WRITELN (A*B:10)
*C/*/+
     WRITELN (A+B:10)
*^Z
[EXIT]

>PIP /LI

DIRECTORY DL0:[50,7]
22-FEB-82 20:57

VSOTA.PAS;1          1.          22-FEB-82 20:54
VSOTA.MAC;1          1.          22-FEB-82 20:56
VSOTA.PAS;2          1.          22-FEB-82 20:57

TOTAL OF 3./15. BLOCKS IN 3. FILES

>PIP /LI

DIRECTORY DL0:[50,7]
22-FEB-82 20:58

VSOTA.PAS;1          1.          22-FEB-82 20:54
VSOTA.MAC;1          1.          22-FEB-82 20:56
VSOTA.PAS;2          1.          22-FEB-82 20:57

TOTAL OF 3./15. BLOCKS IN 3. FILES
```

```
>MAC VSOTA=VSOTA
>TKB VSOTA=VSOTA,[1,1]OMSLIB/LB
>RUN VSOTA
12,15
      180
>PAS VSOTA=VSOTA
>MAC VSOTA=VSOTA
>TKB VSOTA=VSOTA,[1,1]OMSLIB/LB
>RUN VSOTA
12,15
       27
>RUN VSOTA
1234
5678
     6912
>
>
>
 PIP /LI

DIRECTORY DL0:[50,7]
22-FEB-82 21:01

VSOTA.PAS;1          1.          22-FEB-82 20:54
VSOTA.MAC;1          1.          22-FEB-82 20:56
VSOTA.PAS;2          1.          22-FEB-82 20:57
VSOTA.OBJ;1          1.          22-FEB-82 20:58
VSOTA.TSK;1          13.      C  22-FEB-82 20:58
VSOTA.MAC;2          1.          22-FEB-82 20:59
VSOTA.OBJ;2          1.          22-FEB-82 21:00
VSOTA.TSK;2          13.      C  22-FEB-82 21:00

TOTAL OF 32./48. BLOCKS IN 8. FILES


>PIP *.*/PU
>PIP /LI

DIRECTORY DL0:[50,7]
22-FEB-82 21:01

VSOTA.PAS;2          1.          22-FEB-82 20:57
VSOTA.MAC;2          1.          22-FEB-82 20:59
VSOTA.OBJ;2          1.          22-FEB-82 21:00
VSOTA.TSK;2          13.      C  22-FEB-82 21:00

TOTAL OF 16./24. BLOCKS IN 4. FILES


 >BYE
 >
 LAHKO NOC
 22-FEB-82 21:01 TT0:   JE IZKLJUCEN
```

## PRIMER PROGRAMA V FORTRANU

Tu si lahko ogledamo razvoj programa v FORTRANU. Program je podoben prejsnjemu, saj ravnotako prebere iz terminala dve stevili in nato izpise njuno vsoto. Pri tem sta uporabljena ukaza TYPE in ACCEPT, izpis na ekran in vnos iz tastature.

```
>HEL 50,6/DELTA

       DELTA-M V1.2 BL26   SISTEM

Dober vecer
22-Feb-82 21:18 vkljucen na terminal TT0:

Dobrodosli na DELTA-M V1.2 operacijskem  sistemu


Po koncanem delu purgajte svoje direktorije in jih ne scitite
ker bodo le ti zbrisani !!!!!


>
>PIP /LI
PIP -- NO SUCH FILE(S)

>EDI SUMA.FTN
[CREATING NEW FILE]
INPUT
        ACCEPT 45,I,J
        TYPE 12 I+J
45      FORMAT (2I5)
12      FORMAT (I6)
        STOP
        END
T
*LI

*
T
*DP
LI
*DP
[*EOB*]
*T
*LI

        ACCEPT 45,I,J
        TYPE 12 I+J
45      FORMAT (2I5)
12      FORMAT (I6)
        STOP
```

```
>FOR SUMA=SUMA
.MAIN.                                                    STRAN    49

FORTRAN IV DIAGNOSTICS FOR PROGRAM UNIT .MAIN.

IN LINE 0002,  ERROR:   SYNTAX ERROR

FOR -- [.MAIN.] ERRORS: 1, WARNINGS: 0
>EDI SUMA.FTN
[00006 LINES READ IN]
[PAGE    1]
*
        ACCEPT 45,I,J
*
        TYPE 12 I+J
*C/2/2,
        TYPE 12, I+J
*^Z
[EXIT]

>FOR SUMA=SUMA
.MAIN.
>TKB SUMA=SUMA,[1,1]FORLIB/LB
>RUN SUMA
12,34
   46
TTO  --  STOP
>RUN
RUN -- SYNTAX ERROR
>RUN SUMA
12
   12
TTO  --  STOP



>PIP *.*/PU
>PIP /LI


DIRECTORY DL0:[50,6]
22-FEB-82 21:23

SUMA.FTN;2          1.          22-FEB-82 21:21
SUMA.OBJ;2          2.          22-FEB-82 21:21
SUMA.TSK;1          34.      C  22-FEB-82 21:22

TOTAL OF 37./44. BLOCKS IN 3. FILES


>BYE
>
LAHKO NOC
22-FEB-82 21:23 TT0:  JE IZKLJUCEN
>
```