

Verzija 1.0

PARTNER WFG, 2FG, 1FG

Priručnik za uporabnike

Ident: XX XXX XXX

Prva izdaja: Marec 1987

Operacijski sistem: CP/M 3+

Pred vami je preliminarna izdaja PRIROČNIKA ZA UPORABNIKE za mikroračunalnik PARTNER WFG, 2FG, 1FG.

Verjetno se vam bodo ob branju priročnika porodile nove ideje, našli boste boljše razlage, boljše izraze, odkrili morebitne napake in podobno. Za vsako vašo pripombo, poslano na obrazcu v prilogi, se vam zahvaljujemo.

ISKRA DELTA COMPUTERS

Pridržujemo si izključno avtorsko pravico do programskega proizvoda, opisanega v tem priročniku, vključno z vso pripadajočo dokumentacijo.

Pridržujemo si pravico do sprememb brez predhodnega obvestila in ne odgovarjamo za škodne posledice morebitne nevklajenosti informacij v priročniku s proizvodom, ter za morebitne napake.

Jamčimo za uporabnost programskih proizvodov na opremi, ki jo je dobavila in instalirala delovna organizacija ISKRA DELTA COMPUTERS.

Ta priročnik vsebuje tudi tehnično navodilo (Zakon o standardizaciji, Uradni list SFRJ št. 38/77)

ISKRA DELTA COMPUTERS
Proizvodnja računalniških sistemov in inženiring
Parmova 41
61000 LJUBLJANA
JUGOSLAVIJA

KAZALO

=====

1.	OSNOVNI RACUNALNISKI POJMI - UVOD.	1-1 ✓
2.	KAKO JE SESTAVLJEN RACUNALNIK.	2-1 ✓
2.1	Delovni pomnilnik	2-1 ✓
2.2	Procesor.	2-2 ✓
2.3	Krmilna enota	2-3 ✓
2.4	Periferne enote	2-3 ✓
2.4.1	Zaslonski terminal	2-4 ✓
2.4.2	Digitalna tablica in miška	2-4 ✓
2.4.3	Tiskalnik.	2-4 ✓
2.5	Periferni pomnilnik	2-5 ✓
2.5.1	Magnetni trak.	2-5 ✓
2.5.2	Magnetni disk.	2-6 ✓
3.	RACUNALNISKI SISTEMI	3-1 ✓
4.	PROGRAMSKA OPREMA.	4-1 ✓
5.	KAJ PA PARTNER ?	5-1 ✓
5.1	Zgradba mikroročunalnika Partner.	5-2 -
5.1.1	Osrednja enota	5-2 -
5.1.2	Stikala in priključki.	5-2 -
5.1.3	Zaslona	5-3 ✓
5.1.4	Tipkovnica	5-3 ✓
5.1.5	Disketna enota	5-4 ✓
5.1.6	Diskovna enota	5-5 -
5.2	Programska oprema mikroročunalnika Partner.	5-5 ✓
6.	POSTAVITEV IN ZAGON MIKROČUNALNISKEGA SISTEMA PARTNER.	6-1 ✓
6.1	Pregled prevzetega blaga.	6-1 ✓
6.2	Okolje.	6-2 ✓

6.3	Postavitve sistema	6-2
6.3.1	Povezava delov in vključitev sistema	6-2
7.	OSNOVNO DELO S PARTNERJEM.	7-1
7.1	Poslovno delo	7-1
7.2	Grafično delo	7-1
7.3	Aplikacije.	7-1
7.4	Sistemsko delo.	7-1
7.5	Terminal.	7-2
7.6	Ura	7-2
7.7	Programski generator.	7-2
7.8	Izhod v CPM	7-2
8.	OPERACIJSKI SISTEM CP/M 3+	8-1
8.1	Ukazna vrstica.	8-1
8.2	Pojem datoteke.	8-3
8.2.1	Enolični opis datoteke	8-3
8.2.2	Dostop do več datotek.	8-4
8.3	Izbira pogona	8-5
8.4	Izbira uporabniškega področja	8-6
8.5	Datotečni atributi.	8-6
8.5.1	Sistemske in uporabniške datoteke.	8-7
8.6	Predlog za razdelitev delovnih področij	8-8
8.7	Pregled najpomembnejših ukazov CP/M 3+.	8-8
8.7.1	Ukaz DATE.	8-10
8.7.2	Ukaz DIR	8-11
8.7.3	Ukaz ERASE	8-13
8.7.4	Ukaz KOPI.	8-15
8.7.5	Ukaz PIP	8-16
8.7.6	Ukaz RENAME.	8-19
8.7.7	Ukaz SET	8-20
8.7.8	Ukaz SHOW.	8-21
8.7.9	Ukaz SUBMIT.	8-23
8.7.10	Ukaz TYPE.	8-26
8.7.11	Ukaz USER.	8-28

9.	SISTEMSKO DELO	9-1
9.1	Formatiranje disket	9-1
9.2	Zagon računalnika in operacijski sistem	9-2
9.3	Generiranje sistemskega diska / diskete	9-3
9.3.1	Generiranje sis. diska / diskete za WFG	9-4
9.3.2	Generiranje sistemske diskete za 2FG	9-6
9.3.3	Generiranje sistemske diskete za 1FG	9-6
9.4	Instaliranje dokupljene programske opreme	9-7
9.5	Zaščita podatkov pred izgubo.	9-8
10.	PARTNER TERMINAL	10-1 ✓
10.1	SET-UP.	10-2 ✓
10.2	Nastavitev tabulatorskih mest (tabulators).	10-3 ✓
10.3	Nastavitev terminala in tipkovnice (TERM & KBD)	10-4 ✓
10.3.1	Tip terminala (Terminal type).	10-4
10.3.2	Nabor znakov (Terminal language)	10-4
10.3.3	Razporeditev tipk na tipkovnici (kbd type)	10-5
10.3.4	Ponavljanje znaka (auto repeat).	10-5
10.3.5	Zvočni odziv tipkovnice (keyclick)	10-5
10.4	Nastavitev ekrana (screen).	10-5
10.4.1	Širina ekrana (screen lenght).	10-5
10.4.2	Ozadje ekrana (screen background).	10-6
10.4.3	Prehod v novo vrstico (new line)	10-6
10.4.4	Lomljenje vrstic (Auto wrap_around).	10-6
10.5	Shranjevanje in ponoven klic nastavitev	10-6
10.5.1	Vpis v pomnilnik (Save parameters)	10-6
10.5.2	Branje pomnilnika (Recall parameters).	10-6
10.5.3	Začetna nastavitev terminala (Terminal reset)	10-6
11.	KOMUNIKACIJE (opcije).	11-1
11.1	RS-232-C KANAL V.24	11-1
11.2	Opcije.	11-3
11.3	Ukaz DEVICE	11-4
11.4	Izbira in priključitev tiskalnika	11-6

12. NAPAKE IN NJIH ODPRAVLJANJE.	12-1
12.1 Napake ob vključitvi sistema WFG.	12-1
12.2 Napake pri vključitvi sistema 1FG	12-3
12.3 Napake pri vključitvi sistema 2FG	12-4
12.4 Nepravilno delovanje tipkovnice	12-6
12.5 Nepravilno delovanje ekrana	12-6
12.6 Nepravilno delovanje disketnih pogonov.	12-7
12.7 Nepravilno delovanje tiskalnika	12-8
12.8 Sistemska poročila.	12-8

DODATEK A

A. Tehnični podatki	A-1
-------------------------------	-----

DODATEK B

B. Seznam kod alfanumeričnih in posebnih tipk tipkovnice, če je naslovljen terminal tip Partner	B-1
--	-----

DODATEK C

C. Naslovi perifernih enot sistema Partner	C-1
--	-----

DODATEK D

D. Naslovi grafične video plošče	D-1
--	-----

DODATEK E

E. Relativni naslovi prekinitvenih vektorjev v BIOS-u	E-1
---	-----

DODATEK F

F. Način uporabe grafičnih rutin.	F-1
F.1 Navodilo za risanje na grafičnem Partnerju	F-1
F.2 Predlog seznama grafičnih ukazov za grafični Partner	F-4
F.2.1 Knjižnica grafičnih ukazov za Pascal	F-4
F.2.2 Knjižnica grafičnih ukazov za Fortran	F-12
F.2.3 Primer programa za grafiko v Fortranu	F-22

DODATEK G

G. Modul za klic BIOS funkcije 37G-1

DODATEK H

H. Uporaba programa za izdelavo menijevH-1

DODATEK I

I. Ukazna zaporedja Escape (ANSI način)I-1

1. OSNOVNI RACUNALNIŠKI POJMI

1. UVOD

V nekem smislu imajo računalniki zelo kratko zgodovino, ki ni daljša kot 25 - 30 let. Če pa na pojav računalnikov pogledamo širše, sega njihova uporaba v tisti čas, ko je primitivni človek prvič pobral nekaj kamenčkov, saj je bil ob menjavi blaga prisiljen k štetju.

Svet je iz leta v leto postajal bolj zapleten in vedno več je bilo podatkov, ki jih je bilo potrebno obvladovati. Vsi izumi računskih strojev so vse do druge polovice 19. stoletja ekonomsko propadli, saj so bili dragi, lajšali so pa delo le uradnikom, ki so bili cenena in lahko zamenljiva delovna sila.

Leta 1880 so v Ameriki izvedli 11. štetje prebivalstva. Še po štirih letih urejanja podatkov niso prišli do končnih števil. Ko obdelava ni bila končana niti leta 1887, je statistični urad ugotovil, da podatki ne bodo uporabni in da bodo težave z naslednjim štetjem še hujše. To je bilo po vsej verjetnosti prvič, da je skupina ljudi ugotovila, da svet postaja preveč zapleten, da bi ga lahko človeški možgani obvladovali brez zunanje pomoči. Objavili so natečaj za izdelavo računskih strojev in v tridesetih letih tega stoletja so se stvari začele obračati na bolje. Leta 1947 so na Pennsilvanijski elektrotehnični fakulteti sestavili računalnik ENIAC (Electronic Numerical Integrator and Calculator). S tem strojem se pravzaprav začelja obdobje računalnikov. V ZDA so pobudo sprejela velika podjetja. V začetku 50. let so se na trgu pojavili že prvi majhni računalniki, namenjeni gospodarstvu. Njihova cena je bila zelo visoka, sestavni deli nezanesljivi. Kazalo je, da bodo zaradi teh slabosti propadli. Takrat pa so iz družbe Bell Telephone nepričakovano sporočili, da so odkrili transistorje. Tako so računalniki iz preteklosti skočili v sedanost, saj je nova tehnologija omogočila gradnjo zanesljivih in ščasoma tudi cenovno dostopnejših sistemov.

Med ljudmi živi pojmovanje, da je računalnik nekaj zelo zapletenega in nedojemljivega. V resnici je to le avtomat za obdelavo podatkov. Obdelava podatkov pomeni računanje, primerjanje, sortiranje, shranjevanje, vnos in dodajanje podatkov. Ker elektronska vezja zelo hitro delujejo, je tak avtomat visoko zmogljiv, dela mnogo hitreje od mehanskega, zato lahko v enakem času naredi mnogo več. Prav tako, kot moramo programirati mehanski aparat, moramo programirati tudi računalnik.

Bistvena prednost računalnika je v tem, da zelo hitro obdela in prikaže podatke, ki jih potrebujemo. Razen tega je računalnik sposoben na majhnem prostoru shraniti velike količine podatkov in s pomočjo programov priti do njih v zelo kratkem času. Večino ljudi sprva moti to, da so podatki shranjeni tako, da jih ne vidimo. To je motilo tudi prve uporabnike telefona, saj tudi pri telefoniranju ne vidimo sogovornika, danes pa se na to sploh ne spomnimo več.

2. KAKO JE SESTAVLJEN RACUNALNIK

Pogosto se dogaja, da ne vemo, kaj je tista minimalna zgradba, ki ji že lahko rečemo računalniški sistem. Posamezne dele, iz katerih je sestavljen računalnik, imenujemo tudi **pod sisteme računalnika**.

Na sliki 2.1 je prikazan tak model, ki ga imenujemo tudi **centralni oz. procesni del** računalnika.

slika 2.1

2.1. DELOVNI POMNILNIK

Delovni pomnilnik (workin storage) je tisti element, ki ločuje računalnik od navadnih računskih strojev. Vanj lahko zapišemo tako podatke, kot tudi navodila, kako bomo te podatke obdelovali. Kakovost pomnilnika merimo z njegovo kapaciteto, to je številom znakov, ki jih lahko zapišemo v pomnilnik in s hitrostjo, s katero lahko podatke odlagamo v pomnilnik. Najmanjši del pomnilnika je lokacija (pomnilna celica), ki ima enolično določen naslov. Velikost pomnilnika je v vsakem primeru končna in je omejena s številom možnosti

zapisa različnih naslovov.

Cel princip delovanja računalnika je grajen na sposobnosti razpoznavanja dveh električnih stanj - je napetost - ni napetosti. Po dogovoru označujemo ti dve stanji z logično ničlo ("0") ali logično enko ("1"). Računalnik razpozna in "razume" samo jezik ničel in enk in se podreja zakonom Booleove algebre. Eni taki ničli oziroma enki pravimo en bit, ki nam služi kot enota za podajanje množine informacije. Po ustaljenem standardu predstavimo en znak z osmimi biti, kar imenujemo en **byte**. S kombinacijo n ničel in enk lahko predstavimo 2^n različnih znakov. S pomočjo osmih različnih bitov lahko predstavimo vse znake na tastaturi (priloga ASCII znakov). Velikost pomnilnikov je od nekaj Kilo do nekaj 10 Mega znakov oziroma bytov. (kilo = 1000 znakov, mega = 1000000 znakov). Poznamo več tipov delovnih pomnilnikov:

ROM (Read Only Memory) je pomnilnik, iz katerega lahko informacijo samo beremo, ne moremo pa je zapisovati ali spreminjati. Torej v ROM proizvajalec zapiše stvari, ki jih uporabniki računalnikov naj ne bi spreminjali in ki so tako vitalnega pomena za računalnik, da enostavno ne sme obstajati možnost posega v ta del. Informacija pa se mora ohraniti tudi ob odsotnosti električne napetosti.

RAM (Random Access Memory) je pomnilnik, v katerega informacijo lahko pišemo in jo tudi beremo. Pogosto mu pravimo tudi pomnilnik z naključnim dostopom, saj imamo direkten dostop do posameznih lokacij (delov pomnilnika). Odziv je zato zelo hiter (nekaj nano oziroma mikro sec), tehnologija izdelave pa zato zelo zahtevna in draga. Za pomnilnike tipa RAM je značilno, da niso sposobni hraniti informacije ob odsotnosti električne napetosti. Za trajno hranjenje informacije potrebujemo torej drugačen medij; o tem bomo govorili v poglavju 2.4.

2.2. PROCESOR

To je podsistem računalnika, ki izvršuje dejanja, na osnovi podanih ukazov. Sestavljen je iz aritmetične logične enote, kjer se izvršujejo vse operacije in lastnih pomnilniških elementov, ki jim pravimo registri. V registrih se nahajajo trenutni operandi, delni in končni rezultati. Vsi podatki, ki jih v danem trenutku (ob izvrševanju nekega ukaza) ne potrebujemo, se shranijo v delovni pomnilnik, ki igra nekakšno vlogo hladilnika.

Procesor mora znati naslednje:

- klicati operande iz delovnega pomnilnika
- izvajati aritmetične in logične operacije
- premikati vsebine registrov
- vračati informacije iz

registra v delovni
pomnilnik

V ALE prihajajo podatki o operacijah in operandih, iz nje pa izhajajo rezultati operacij. Vsako operacijo, ki jo ALE samostojno izvede, imenujemo **elementarna operacija** (naprimer enostavno seštevanje). Vse druge operacije moramo prevesti (razbiti) na elementarne operacije. Elementarne operacije se izvajajo druga za drugo v določenih časovnih presledkih.

2.3. KRMILNA ENOTA

Ta podsistem nadzoruje in usklajuje delovanje posameznih enot (podsistemov) tako, da se algoritem (navodila), ki je podan računalniku v obliki programa, pravilno izvede. Krmilna enota vodi odvijanje programa v računalniku tako, da analizira korak za korakom vsak elementarni ukaz. Naslov naslednjega ukaza nam pove register, imenovan programski števec. Glede na rezultat analize sporoči, kateri podsistem mora biti v danem trenutku aktiven in kako naj deluje, da bo zagotovljeno pravilno izvajanje. Poleg tega organizira tudi prenos informacij (elementarnih ukazov) in njih obdelavo.

Krmilno enoto sestavljajo posebej organizirana elektronska vezja, ki opravljajo funkcijo pomnilnih celic (registrov) in funkcijo logičnih povezav z drugimi podsistemi. Program izvajamo tako, da posredujemo začetni naslov našega programa (zapiše se v programski števec), krmilna enota pa najde naslednji korak tako, da naslovu že opravljenega koraka prišteje ena. Štetje igra osnovno vlogo pri zaporednem izvajanju programa, odvija pa se v programskem števcu.

2.4. PERIFERNE ENOTE

Za komunikacijo sta vedno potrebna najmanj dva. Ljudje med seboj lahko komuniciramo, če govorimo v istem jeziku. V nasprotnem primeru smo se pač prisiljeni naučiti jezika, ki nam bo omogočal komunikacijo. Samo poznavanje jezika pa nam ne pomaga, dosti ob potrebi po komunikaciji na daljavo. Za to potrebujemo nek medij oziroma napravo, ki nam omogoča tovrstno komuniciranje.

Pri komuniciranju z računalnikom smo na žalost mi tisti, ki se moramo prilagajati in se naučiti jezika, ki ga računalnik razpozna, saj računalnik ni misleče bitje, pač pa samo daje tak vtis zaradi znanja, ki so ga ljudje v obliki programov vložili vanj. **Računalnik ne misli!** Izvaja samo navodila, ki smo mu jih posredovali.

Verjetno si ne predstavljate, da bi računalniku morali posredovati svoje želje s pomočjo zaporedij ničel in enk (v edinem jeziku, ki ga računalnik "razume"). Z računalnikom si želimo v obe smeri komunicirati v jeziku, ki je čim bližji našemu pogovornemu jeziku. Za to pa poleg ustrezne programske opreme potrebujemo še medije, preko katerih nam

Partner WFG, 2FG, 1FG - PRIROČNIK

je taka komunikacija omogočena. Torej mora računalnik imeti naprave, preko katerih lahko komunicira z uporabnikom, izpisuje, prikazuje ter shranjuje podatke. Vse te naprave spadajo v periferije računalnika.

Človeštvo uporablja za predstavitev informacij urejena zaporedja črk, cifer in posebnih znakov (ločila itd.), računalnik pa živi v svetu električnih signalov. Zato potrebujemo nekega posrednika za komuniciranje, ki bo našo informacijo pretvoril v računalniku "razumljiv" jezik in računalnikovo informacijo v nam razumljivo obliko. To nam omogočajo vhodne enote.

Ponavadi na centralni (procesni) del računalnika ni priključena samo ena vhodna oziroma izhodna enota, ampak jih je v splošnem več.

Naštejmo nekaj perifernih enot: zaslonski terminal, grafični terminal, digitalna tablica, miška, tiskalnik, risalnik, zunanji (periferni) pomnilniki, kot so magnetni trakovi, kasete, diskete, diski.

2.4.1. Zaslonski terminal

Zaslonski terminal je namenjen enostavni komunikaciji med človekom in računalnikom. Sestavljen je iz tipkovnice, ki je podobna tipkovnici pisalnega stroja in zaslona, ki spominja na TV zaslon. Preko tipkovnice posredujemo ukaze računalniku. Odtipkane znake vidimo na zaslonu, kar nam omogoča preverjanje in popravljanje ukazov. Prav tako na zaslonu vidimo odgovor (odziv) računalnika na vpisan ukaz, ki je lahko v obliki teksta ali grafične slike.

Slika na ekranu je predstavljena s točkami. Gostota točk je določena z resolucijo. Večja je resolucija, jasnejša je slika.

2.4.2. Digitalna tablica in miška

Ti dve vhodni enoti lahko uporabimo namesto tipkovnice. Pri takem načinu dela imamo nabor ukazov prikazan na ekranu, s premikanjem tablice ali miške pa izberemo željeni ukaz.

Digitalno tablico uporabljamo tudi za prerisovanje risb iz papirja na ekran oz. v računalnikov pomnilnik.

2.4.3. Tiskalnik

Uporabljamo ga v primeru, ko želimo izpis na papir. Izpisovanje je krmiljeno z računalnikom in je relativno počasno, saj je svet mehanike bistveno počasnejši od sveta elektronike. Pisanje s tiskalnikom je zelo podobno pisanju z električnim pisalnim strojem. Papir vstavimo v stroj in ga pomikamo naprej z valjem, na papir pa se odtiskuje izbrani

tekst. Poznamo več vrst tiskalnikov, ki se med seboj razlikujejo po kvaliteti in hitrosti izpisa (matrični tiskalnik, vrstični tiskalnik, laserski tiskalnik...).

2.5. PERIFERNI POMNILNIKI

Delovni pomnilnik je zaradi zahtevne tehnologije izdelave zelo drag. Omenili smo že, da ima vsaka lokacija svoj naslov; dostop do posamezne lokacije pa mora biti izredno kratek. Konstrukcija takega pomnilnika je tehnološko zahtevna. Če gre za delovni pomnilnik tipa RAM, informacijo ob odsotnosti električne napetosti izgubimo. Zato so poskušali iznajti cenen pomnilni medij, ki lahko trajno hrani veliko množico podatkov. Ideja je bila v tem, da bi v delovnem pomnilniku imeli samo program, ki ga trenutno potrebujemo in podatke, ki jih s tem programom obdelujemo. Vsi ostali programi in podatki pa naj bi bili shranjeni na periferni pomnilni enoti, kjer so še vedno razmeroma hitro dosegljivi.

V vsakdanjem življenju lahko delovni pomnilnik primerjamo s človeškimi možgani, periferne pomnilnike pa s knjižnimi omarami.

Izmed množice najrazličnejših metod so se sčasoma v praksi najbolj uveljavili magnetni mediji. Ti pomnilniki so ceneri, dostop do podatkov je počasnejši (nekaj 10 msec), njihova kapaciteta pa je za faktor 10 ali 100 in več večja od kapacitete delovnega pomnilnika. Njihova dobra lastnost je tudi možnost enostavnega transporta.

Informacije zapisujemo na pomnilni medij s pomočjo **bralno pisalnih glav**.

Vse magnetne medije moramo imeti dobro zaščitene pred prahom in magnetnimi vplivi, da na njih ne pride do napak in s tem do uničenja podatkov.

2.5.1. Magnetni trak

Zgodovinsko gledano je bil magnetni trak prvi tip magnetnega pomnilnika. Lahko je to običajna kasetna ali pa več 100 metrov dolg svitek, ki je navit na kolutih. Trak je običajno iz plastike, ki je prevlečena z železooksidno plastjo.

Organizacija podatkov na traku je zaporedna. To pomeni, da moramo podatke tudi brati zaporedno, kar je še posebno neprijetno v primeru, ko potrebujemo podatke, ki se nahajajo na različnih odsekih traku. Zato tudi težko govorimo o času dostopa do posameznih podatkov, saj je ta močno odvisen od tega, kje na traku se nahaja glava za branje. Veliko časa izgubimo s previjanjem traku.

2.5.2. Magnetni disk

Magnetni disk je okrogla kovinska plošča, na katero je nanešen tanek sloj magnetnega medija. Podatki so na mediju zapisani v obliki koncentričnih krogov, ki jih imenujemo sledi (tracks).

Vidimo, da so podatki na disku zapisani in dostopni na drugačen način, kot na magnetnem traku. Za dostop do podatkov, napisanih na disku, je potreben največ en vrtljaj diska. Ker je vsak podatek v bistvu direktno dostopen, pravimo da je disk naprava z direktnim (naključnim) dostopom. Dostop do podatkov na disku traja nekaj 10 msec.

Danes največ uporabljamo tako imenovane neizmenljive trde diske (winchester diske) in gibke diske (floppy disk).

Pri **winchester diskih** so magnetne plošče z glavami neprodušno zaprte in s tem zavarovane pred zunanjimi vplivi. Takšni diski imajo velike pomnilne kapacitete (nekaj 10 M bytov) in ne potrebujejo vzdrževanja.

Diskete so tanke, prožne, okrogle magnetne plošče, zavarovane s kvadratnim papirnim ovitkom. So zamenljive. Zaradi svoje upogljivosti so dobile ime gibki diski. Diskete uporabljajo skoraj vsi mikroročunalniški sistemi, ker so majhne, prikladne za delo, skladiščenje in pošiljanje po pošti ter so relativno poceni. Na disketi lahko shranimo nekaj 100 K bytov.

3. RACUNALNIŠKI SISTEMI

Večina računalnikov ima vse sestavne dele, ki smo jih opisali. Razlikujejo se po hitrosti delovanja, velikosti pomnilnikov in številu perifernih enot. Glede na centralno procesno enoto jih delimo v velike, mini in mikro računalnike.

S stališča uporabnika jih delimo na eno in večuporabniške sisteme. Pri večuporabniških sistemih lahko hkrati uporablja isti računalnik več uporabnikov, na enouporabniškem sistemu pa lahko dela le en uporabnik hkrati. Te sisteme lahko dalje delimo v hišne, osebne in male poslovne sisteme ter jih navadno uvrščamo med mikroračunalnike.

Hišni računalnik je preprost mikro računalnik z majhnim pomnilnikom in skromno periferijo. Ekran navadno nadomešča TV sprejemnik, zunanji pomnilnik pa kasete oz. kasetofon. Ker so počasni in nezanesljivo delujejo, nikakor niso namenjeni profesionalni uporabi. Zaradi cenenosti so dostopni skoraj vsem, ki se želijo spoprijeti z reševanjem preprostih problemov s pomočjo računalnika.

Osebni in mali poslovni računalniki imajo sposobnejše periferne enote (winchester diske in diskete, kvaliteten monitor, kvalitetno tipkovnico), praviloma večji in hitrejši delovni pomnilnik, njihovo delovanje pa je zanesljivo.

4. PROGRAMSKA OPREMA

=====

Omenili smo že, da komuniciramo z računalnikom preko vhodno/izhodnih enot in da se moramo mi naučiti jezika, s katerim lahko posredujemo naše želje računalniku.

Programi, ki omogočajo komunikacijo med človekom in računalnikom, med procesorjem in perifernimi enotami ter vse skupaj povezujejo v "živo", skladno delujočo celoto, imenujemo **operacijski sistem**. Smoter operacijskega sistema je razbremeniti uporabnika in čimbolje izkoristiti zmogljivosti računalnika. Kot dodatek operacijskemu sistemu so **uslužnostni programi** (utilities), ki jih velikokrat poimenujemo kar ukazi operacijskega sistema. Ti nam služijo za delo s podatkovnimi zbirkami na perifernih pomnilnih medijih.

Med programsko opremo spada tudi cel spekter **prevajalnikov**. Najbolj uporabljeni so prevajalniki za višje programirne jezike. Vsak jezik ima svojo slovnico, ki jo prevajalnik prevede v stroju "razumljiv" jezik. Jezik višjih programirnih jezikov je v omejenem okviru zelo blizu našemu naravnemu razmišljanju in izražanju problema.

S pomočjo najrazličnejših jezikov lahko pišemo algoritme za rešitev nekega določenega problema. Take programe imenujemo **aplikacije**. Uporaba aplikacij je tudi za neračunalnikarja enostavna.

5. KAJ PA PARTNER ?

=====

PARTNER je sodoben, interaktivni, mikoračunalniški sistem, namenjen poklicni uporabi. Njegova centralna procesna enota temelji na istem mikroprocesorju, kot pri nekaterih hišnih računalnikih, vendar pa PARTNER le spada v višjo kvaliteto skupino. Zakaj?

Njegovi sestavni deli so zelo skrbno izbrani in preizkušeni. Računalnik je zanesljiv, ima kvalitetnejše periferne enote, uporablja v svetu zelo razširjen operacijski sistem CP/M 3+ (Control Program Monitor), ki nam z obilico kvalitetnih programirnih jezikov omogoča razvoj najrazličnejših aplikacij. Uporabljamo lahko tudi vso v svetu razvito programsko opremo, ki je razvita za ta operacijski sistem. Delo z njim je zelo enostavno in prijetno - to je res pravi **partner** pri vašem delu.

Mikroračunalniška družina PARTNER trenutno obsega tri inačice, ki se razlikujejo samo po zunanjih pomnilnikih:

- WF/G - grafični mikroračunalnik PARTNER z winchester diskom in eno disketno enoto. Ker ima velike pomnilne kapacitete, je široko uporaben na vseh področjih dela.
- 2F/G - grafični mikroračunalnik PARTNER z dvema disketnima enotama je namenjen za obdelavo, ki ne zahtevajo hitrih in obsežnih pomnilnikov.
- 1F/G - grafični mikroračunalnik PARTNER z eno disketno enot. Uporabljamo ga kot šolski računalnik, še posebej pa je zaradi cenenosti primeren kot inteligentni terminal večjim sistemom.

5.1 ZGRADBA MIKRORAČUNALNIKA PARTNER

Slika 5.1.

Standardna konfiguracija PARTNERJA obsega naslednje enote:

- osrednja enota
- tipkovnica

5.1.1. Osrednja enota

Osrednjo enoto sestavljajo zaslon, diskovna enota in disketna enota. Srce računalnika predstavlja centralna procesna enota, pomnilnik, krmilne enote, napajalna enota itd., vendar te za uporabnika niso neposredno pomembne.

Zunanje enote priključimo na hrbtne strani osrednje enote, kjer so stikala in vtičnice.

5.1.2. Stikala in priključki

Slika 5.2.

Na sliki vidimo naslednje (od desne proti levi):

1. vtičnica za tipkovnico,
2. vrtljiv gumb za nastavitev osvetljenosti zaslona,
3. vtičnica z varovalkami za priključni kabel, ki ga na drugi strani vtaknemo v ozemljeno vtičnico z napetostjo 220 V in 50Hz,
4. vtičnica za tiskalnik (J7),
5. stikalo za vklop in izklop napetosti z vgrajeno lučko, ki sveti, ko je PARTNER vklopljen,
6. stikalce RESET, ki služi za postavitve računalnika v začetno stanje,
7. priključnice J6, J8, J9, LAN in TV OUT so vgrajene na željo uporabnika.

5.1.3 Zaslon

Za komuniciranje z uporabnikom je v računalnik vgrajen kvaliteten zaslon, ki daje sliko brez odboja. Zaslon ni namenjen prikazu velikih količin podatkov. Njegov namen je predvsem hiter in učinkovit dialog med uporabnikom in sistemom.

Pri **tekstovnem načinu** so podatki, ki jih prikazuje zaslon, v obliki znakov, sestavljeni iz polja točk (matrike) 8x11. Možen je prikaz velikih in malih črk, števil in posebnih znakov. Uporabniku je na voljo 24 vrstic po 80 ali 132 znakov. Dodani sta še 25. in 26. vrstica za prikaz sistemskih sporočil.

Za prikaz **grafične slike** je poseben pomnilnik, kjer shranimo dve sliki resolucije 1024x512 (256) točk. Sistemska programska oprema omogoča risanje črt, krogov in podobnih elementov. Grafična slika in tekstovni prikaz se lahko prikazujeta na zaslonu istočasno.

Razen ustrezne nastavitve svetlosti (vrtljiv gumb na hrbtni strani osrednje enote) in občasnega čiščenja ne potrebuje uporabnikovih posegov.

5.1.4 Tipkovnica

Tipkovnica je ločen del PARTNERJA in jo lahko namestimo na najustreznejše mesto za udobno delo. Z osrednjo enoto jo povezuje raztegljivi, spiralasti kabel, ki ga vtaknemo v priključnico na hrbtni strani osrednje enote. Tipke so zasnovane ergonomično in so nebleščeče. Tipkovnica ima 83 tipk, ki so razporejene v dve skupini. Na levi strani so

Partner WFG, 2FG, 1FG - PRIROČNIK

tipke kot pri pisalnem stroju, razporeditev je jugoslovanska QWERTZ. Na desni strani so numerične tipke kot pri računalnem stroju za hitrejši vnos numeričnih podatkov. Med njimi so tudi štiri funkcijske tipke, katerih uporaba je odvisna od aplikacije.

Za splošno uporabo morate poznati naslednje tipke:

- < RET > S to tipko vselej zaključimo ukaz in tako damo računalniku vedeti, da želimo izvršitev ukaza.
- < ENT > Ima popolnoma enako vlogo kot < RETURN >.
- < CAPS > Vklopi velike črke (ne pa tudi posebnih znakov), kar označuje rdeča lučka na tipki. Izključimo jih s ponovnim pritiskom na to tipko. Nekatere aplikacije zahtevajo kot odgovor velike črke, zato je bolje imeti to tipko vklopljeno.
- < SHIFT > Če hkrati s to tipko pritisnemo še kakšno drugo tipko, se na zaslon izpišejo velike črke oz. zgornji (posebni) znaki.
- < BS > Zbriše zadnji vnešeni znak. V nekaterih aplikacijah s to tipko premaknemo kurzor na predhodni znak. V besedilu uporabljamo za to tipko znak <BS>.
- < DEL > Zbriše znak, ki je levo od sledilnega znaka.
- < CTL > Ta tipka ima smisel samo, če hkrati z njo pritisnemo še kakšno drugo tipko. Uporabljamo jo za posebne ukaze računalniku.
- < SCR > Pritisk te tipke zaustavi izpisovanje na ekran. S ponovnim pritiskom na to tipko se izpisovanje nadaljuje.

5.1.5. Disketna enota

Pri sistemih WFG in 1FG je disketna enota vgrajena v desnem spodnjem kotu osrednje enote, sistem 2FG pa ima disketni enoti drugo nad drugo na desni strani osrednje enote. Uporabljamo diskete velikosti 5 in 1/4 cole z dvojno gostoto zapisa in z dvostranskim zapisom. Kapaciteta diskete je 644 Kb.

Čeprav je disketa za računalniške pojme robusten medij, zahteva vseeno pazljivost pri uporabi. Zlasti moramo biti pazljivi pri vstavljanju:

- disketo vzamemo iz ovitka tako, da jo primemo za nalepko

Partner WFG, 2FG, 1FG - PRIROČNIK

- z nalepko navzgor jo vložimo v odprtino do konca
- pokrovček zapremo.

Kadar je disketna enota aktivna, sveti na njej rdeča lučka. TAKRAT DISKETE NE SMEMO VZETI VEN! Ko diskete ne potrebujemo več, jo vložimo nazaj v ovitek, da se ne prahi in poškoduje. Na ovitku so tudi navodila, kako ravnati z njimi:

- ne dotikaj se magnetne površine
- ne približuj namagnetnenih predmetov
- ne upogibaj
- pazljivo vlagaj
- hrani pri temperaturi 10 - 50 stopinj C, po možnosti v pokončnem položaju.

Novih disket ne moremo uporabljati, dokler jih ne formatiramo. Formatiranje pomeni, da se na disketo zapišejo sledi in sektorji, kamor se bodo potem zapisovali podatki.

5.1.5. Diskovna enota

Sistem WFG ima nad disketno enoto vgrajeno diskovno enoto, v kateri je disk tipa winchester s kapaciteto 10 milijonov znakov (10 Mb), kar ustreza približno 2700 tipkanim stranem besedila. Vsi podatki, shranjeni na disku, so direktno dostopni, povprečni čas pristopa je 85 ms. Ta enota ne potrebuje uporabnikovih posegov, celo prepovedani so. Kadar sveti lučka na diskovni enoti, pomeni, da se podatki z nje berejo ali zapisujejo.

5.2. PROGRAMSKA OPREMA NA MIKROKONČALNISKEM SISTEMU PARTNER

Zaradi enostavne uporabe boste mikroracunalniški sistem PARTNER radi sprejeli za partnerja pri raznih opravilih. PARTNER lahko reši veliko vaših problemov, saj je v ta namen vrsta naših storkovnjakov razvila veliko število ustreznih aplikativnih programov. S temi programi boste učinkovito opravljali vsa rutinska in zamudna dela. Vsi aplikativni programi so zasnovani tako, da vam bo delo z njimi v zadovoljstvo, vedno znova pa se vam bodo odpirale možnosti uporabe v skladu z vašimi željami.

6. POSTAVITEV IN ZAGON MIKRRORAČUNALNIŠKEGA SISTEMA PARTNER

Odločili ste se za nakup našega mikrroračunalniškega sistema PARTNER. Da ne bi imeli problemov že ob samem začetku, vas bomo opozorili na nekaj ključnih stvari.

6.1 PREGLED PREVZETEGA BLAGA

Mikrroračunalniški sistem PARTNER je ob prevzetju zapakiran v trdno, kartonsko škatlo, v skladu z mednarodnimi standardi.

Pošiljka vsebuje naslednje dele:

Standardna vsebina:

1. Osrednja enota z zaslonom
2. Tipkovnica
3. Priključni kabel
4. Sistemsko disketa
5. 10 praznih, tovarniško novih disket
6. Uporabniški priročnik
7. Seznam blaga v dobavljeni pošiljki

Dodatna vsebina: (odvisno od pogodbe)

1. Programirni jeziki (disketa in priročnik)
2. Aplikacijski programi (disketa in priročnik)
3. Opcije

Ob prejemu dobro preglejte pošiljko, saj je lahko prišlo do morebitne poškodbe pri prevozu. Če je poškodovan katerikoli del ali če se vsebina pošiljke ne ujema s priloženim seznamom blaga, nam pošiljko vrnite, z ustrezno opombo.

Če se zgodi, da poškodbo ali pomanjkljivost odkrijete kasneje, nas v roku 15-ih dni od prejema pošiljke o tem obvestite.

6.2 OKOLJE

PARTNER je sodoben namizni mikroročunalnik. Glede okolja ni zahteven, kljub temu pa moramo, če želimo da bo njegovo delovanje brezhibno, upoštevati naslednje:

- Ohišja sistema in zaslona ne smemo izpostavljati direktnim sončnim žarkom;
- Zaslona naj bo obrnjen tako, da svetloba ne bo padala direktno nanj;
- Močnejših virov toplote ne smemo nameščati v bližino sistema;
- Temperatura okolja naj bo v območju 10-32 st C; če je nižja od 10 st C je treba po vključitvi sistema nekaj časa počakati, da sistem doseže svojo delovno temperaturo;
- Relativna vlažnost okolja naj bo v območju 20-80%, brez kondenza;
- Zgornje strani računalniškega ohišja ter neposredne okolice računalnika ne smemo uporabljati kot odlagalne površine; še posebno ne smemo tja postavljati posod s tekočinami, kovancev, kovinskih sponk, ipd.; če ti predmeti zaidejo v notranjost sistema lahko povzročijo motnje v njegovem delovanju in poškodbe;
- sistema ne smemo postaviti v neposredno bližino virov elektromagnetnih motenj (tudi telefonski aparat !)

6.3 POSTAVITEV SISTEMA

Potem, ko smo v prisotnosti pooblaščenih oseb iz ISKRE DELTE posamezne dele sistema razpakirali ter pregledali, če so vsi in če niso morda poškodovani, se lahko lotimo postavitve sistema.

Mikroročunalniški sistem PARTNER lahko postavimo na katerokoli ustrezno veliko mizo. Posamezne sestavne dele sistema, osrednjo enoto, tipkovnico in tiskalnik povežemo med seboj s kablji. Vse priključnice in stikala so na zadnji strani ohišja sistemske enote (glej sliko 5.2).

6.3.2 Povezava delov in vključitev sistema

Ko smo posamezne dele sistema namestili na primerna mesta, pričnemo z zagonom sistema. Ravnamo po naslednjem postopku:

- Preverimo, če je stikalo za vklop/izklop v položaju 0 (= izklop).
- Priložen omrežni kabel vtaknemo najprej v vtičnico

Partner WFG, 2FG, 1FG - PRIROČNIK

- omrežnega dela na zadnji strani ohišja osrednje enote in nato v ozemljeno šuko vtičnico.
- Spiralni kabel tipkovnice vtaknemo v ustrezno okroglo priključnico na zadnji strani ohišja sistemske enote;
 - Če imate tiskalnik, vtaknite njegov kabel v priključnico J7.
 - Po možnosti vključite tiskalnik v omrežno napajanje preko druge vtičnice kot sistemske enote.

Sistem vključimo s stikalom za vklop/izklop na zadnji strani osrednje enote. Ko pritisnemo na to stikalo, na njem zasveti rdeča lučka. Če imamo priključen tiskalnik, isti postopek ponovimo pri tiskalniku. Važno je zaporedje vključitve oz. izključitve. Najprej vključimo osrednjo enoto, nato tiskalnik. Izključimo v obratnem vrstnem redu. Po pritisku na stikalo za vklop/izklop se na zaslonu pojavi izpis preko celotnega ekrana:

DELTA PARTNER GDP

Takoj za tem pa se na zaslonu pojavi menu z naborom različnih možnosti uporabe PARTNERJA z navodili za nadaljnjo uporabo.

7. OSNOVNO DELO S PARTNERJEM

Po vključitvi sistema, se nam na zaslonu pojavi slika menija, ki prikazuje vrsto izbir. Menujsko delo olajša uporabniku delo z računalnikom. Uporabnik s pritiskom na določeno tipko izbere aktivnost, to je izvajanje nekega programa, oziroma prehod v naslednji menu.

7.1. POSLOVNO DELO

Omogoča urejanje besedil, upravljanje osebne in poslovne informatike, simulacijo poslovnega dogodka.

7.2. GRAFIČNO DELO

To so programi, s katerimi lahko izpisujemo grafikone za prikaz poslovnih dogodkov; kreiramo in popravljamo dvodimenzijske tehniške risbe; interaktivno rišemo slike, katere lahko s pomočjo naprave (RAMTEK-Iskra-Delta-Computers) prerišemo na folije, papir ali preslikamo diapozitive; izdelujemo diapozitive, sestavljene iz teksta in preprostih shem.

7.3. APLIKACIJE

Uporabnik ima možnost vključiti svoje lastne programe v menuje, katere je sam izdelal s pomočjo programa za izdelavo menujev.

Podrobni opis dela s programom za izdelavo menujev vsebuje DODATEK K.

7.4. SISTEMSKO DELO

Omogoča formatiranje diskete, izdelavo menujev in delo z datotekami.

7.5. TERMINAL

Nam omogoča priključitev Partnerja kot VT100 Terminal na drugi računalnik in nastavitve delovanja Partnerjevega terminala.

7.6. URA

Prikaz časa in datuma, ter možnost spremembe datuma in nastavitve časa.

7.7. PROGRAMSKI GENERATOR

Je namenjen tako začetnikom, kot izkušenim programerjem za hitro in lahko urejanje računalniških programov. Vnešene podatke o izgledu zaslonske slike, o načinu zaščite podatkov, avtomatično uporablja pri pisanju računalniških programov.

7.8. IZHOD V CPM

Omogoča izhod v operacijski sistem CP/M 3+. Podrobnejši opis dela z operacijskim sistemom CP/M 3+ je naveden v naslednjem poglavju.

8. OPERACIJSKI SISTEM CP/M 3+

=====

Glavna naloga operacijskega sistema je, da vodi in nadzira naslednje računalniške enote: pomnilnik, diskovno, disketno enoto, terminal, tiskalnik, komunikacijske enote, upravlja s programi in podatki, ki so shranjeni na perifernem pomnilnem mediju, kopira programe in podatke med magnetnimi mediji in delovnim pomnilnikom. Izvaja ukaze, ki jih v ukazni vrstici posredujemo računalniku.

8.1. UKAZNA VRSTICA

Operacijski sistem CP/M 3 PLUS se javi z znakom pripravljenosti A>, na katerega uporabnik odgovori z ukazom, ki je sestavljen iz treh delov:

- ključne ukazne besede
- poljubnega dodatka k ukazu
- pritiska na tipko <RET>

Primer:

A>dir *.txt <RET>

- A> - znak pripravljenosti operacijskega sistema za sprejemanje ukazov
- dir - ključna beseda opravila. Predstavlja ime za program, ki ga mora računalnik izvesti
- *.txt - dodatno določilo v okviru omenjenega ukaza, ki vpliva na njegovo izvedbo
- <RET> - vsak ukaz moramo zaključiti s pritiskom na to tipko. Sele za tem računalnik začne izvrševati ustrezno akcijo

V ukazno vrstico lahko tipkamo z velikimi ali malimi črkami, operacijski sistem CP/M privzame vse kot velike črke.

Ukazno vrstico lahko tudi popravljamo. Za to uporabljamo

Partner WFG, 2FG, 1FG - PRIROČNIK

kontrolne tipke:

Izbira	Funkcija
CTL-A	Premakne sledilni znak za en znak v levo.
CTL-B	Pomakne sledilni znak na začetek ukazne vrstice brez dodatnih vplivov na njeno vsebino. Če je sledilni znak na začetku vrstice, jo prenese CTL-B na njen konec.
CTL-F	Premakne sledilni znak za en znak v desno.
CTL-G	Izbríše znak, na katerem utripa sledilni znak. Ta se ne premakne.
CTL-H	Izbríše znak in premakne sledilni znak za en znak v levo.
CTL-J	Pošlje ukaz do CP/M 3 in postavi sledilni znak na začetek nove vrstice. Učinkuje enako kot tipka RETURN ali pa kot tipka CTL-M.
CTL-K	Izbríše vso informacijo od sledilnega znaka dalje do konca vrstice.
CTL-M	Pošlje ukazno vrstico v CP/M 3 in postavi sledilni znak na začetek nove vrstice. Učinkuje enako kot RETURN ali pa kot tipka CTL-J.
CTL-W	Povrne in prikaže poprej vtipkano ukazno vrstico, tako na ravni operacijskega sistema kot tudi v notranjosti izvedljivih programov, če je CTL-W prvi znak, ki je bil vnešen po predstavitvi sistema na zaslonu. CTL-J, CTL-M, CTL-U in RETURN določajo ukazno vrstico, ki jo moremo priklicati nazaj. Če so v ukazni vrstici znaki, premakne CTL-W sledilni znak na konec ukazne vrstice. Če pritisnete RETURN, bo CP/M 3 izvršil ta ukaz ponovno.
CTL-X	Zbríše vse znake levo od sledilnega znaka in sledilni znak premakne na začetek tekoče vrstice, torej ohrani vse znake desno od položaja sledilnega znaka.

Primer:

A>dor *.txt
DOR?(naš vnos)
(odgovor računalnika)

Pri vnosu ukaza smo naredili napako. Ni potrebno vnesti

ponovno celotnega ukaza, pomagamo si lahko s kontrolnimi tipkami.

```
A>DOR *.TXT_           (CTL-W vrne prejšnjo vrstico)
A>DIR *.TXT            (s CTL-A se pomikamo do napake)
A>DR *.TXT             (s CTL-G zberemo napačen znak)
A>DIR *.TXT            (vpišemo pravičen znak in z <RET>
                       zaključimo ukaz)
```

Včasih prenaša CP/M 3+ informacijo na zaslon prehitro, da bi ji mogli slediti. Posebno dolg prikaz izgine z vrha zaslona prej, kot bi ga bilo mogoče prebrati. Če hočemo povedati sistemu, naj počaka, da bomo lahko prebrali vsebino zaslona, moramo imeti pritisnjeno tipko CTL-S. Tipka CTL-S povzroči prekinitve prenosa informacije na zaslon. Ko smo pripravljeni, pritisnemo CTL-Q za nadaljevanje prenosa prikaza na zaslon. Isto vlogo ima tipka <SCR>.

Če želimo izpis iz zaslona hkrati tudi na tiskalniku, uporabimo tipko CTL-P. Na tiskalnik se izpisujejo vsi znaki, prikazani na zaslonu. S ponovnim pritiskom kombinacije tipk CTL-P prekinemo tak način delovanja.

8.2 POJEM DATOTEKE

Datoteka (file - angl.) je zbirka povezanih informacij, shranjenih na disku. Vsaka datoteka mora imeti enolično določeno ime, da jo operacijski sistem prepozna. Na disku je tudi direktorij. To je seznam, ki vsebuje imena in nekatere druge podatke o vseh datotekah, ki se nahajajo na disku.

V splošnem ločimo dve vrsti datotek: programske (ukazne) in podatkovne datoteke. Programska datoteka vsebuje izvedljiv program, to je zaporedje navodil, ki jih računalnik korakoma izvaja. Podatkovna datoteka je običajno zbirka informacij: seznam imen in priimkov, seznam osnovnih sredstev, izvorni program, vsebina kakšnega dokumenta oz. poljuben tekst itd. Denimo, računalnik ne more izvesti imen in priimkov, lahko pa izvede program, ki izpiše imena in priimke na papir.

8.2.1 Enolični opis datoteke

Operacijski sistem prepozna datoteko po njenem enoličnem opisu (file specification - angl.). Le-ta je sestavljen takole:

```
d:imedat.tip;geslo
```

Partner WFG, 2FG, 1FG - PRIROČNIK

- d:** ime pogona - NEOBVEZEN PARAMETER
To je lahko (pri PARTNERJU) A ali B. A pomeni, da se datoteka nahaja na disku, B pa , da je na disketi. Opis pogona je ločen od imena datoteke z dvopičjem.
- imedat:** ime datoteke je OBVEZNO. Sestavljeno je iz enega do osem znakov. Zaželeno je , da ime datoteke govori tudi o njeni vsebini. Denimo, če vsebuje datoteka seznam kupcev , ji je smiselno dati ime KUPCI, ne pa XXX, ker nam to nič ne pove.
- tip:** je sicer NEOBVEZEN dodatek, vendar ga praviloma uporabljamo. Sestavljen je iz enega do treh znakov. Od imena je ločen s piko. Pove nam, v katero skupino spada datoteka.
Naj navedemo nekaj pogostih tipov datotek:

TXT	tekstovna datoteka
DAT	podatkovna datoteka
BAS	program v BASICU
PAS	program v PASCALU
COM	ukazna datoteka
COB	program v COBOLU

- geslo:** je sestavljeno iz enega do osem znakov. Od imena ali tipa datoteke je ločeno s podpičjem. Funkcija gesla je zaščita datoteke. NEOBVEZEN PARAMETER

Za enolični opis datoteke smemo uporabljati črke angleške abecede in številke. Izogibajmo pa se jugoslovanskim znakom : C, S, 2, D, C, ki imajo posebej definiran pomen in posebnim znakom: ., ; () ? ! itd.

ZGLEDI:

```
KUPCI
KUPCI.DAT
B:VAJAI.TXT
B:SEZNAM;NADA
A:DOPIŠ.TXT;PASSWORD
```

8.2.2 Dostop do več datotek

Zgodi se, da bi želeli prepisati, brisati, poiskati več datotek hkrati, morda kar vse z določenega pomnilnega medija. V takem primeru jih opišemo z imenom, ki ne določa enolično ene same datoteke, ampak ustreza več datotekam hkrati. Zato uporabimo znaka ? in *. Znak ? lahko nadomesti en znak v enoličnem opisu datoteke, * pa več znakov.

Partner WFG, 2FG, 1FG - PRIROČNIK

ZGLEDI:

Denimo, da imamo na disku datoteke A.DAT, AA.DAT, AAA.DAT, B.DAT, A.TXT IN B.TXT.

?.DAT	zajame A.DAT, B.DAT
*.DAT	zajame A.DAT, B.DAT, AA.DAT, AAA.DAT
A.*	zajame A.DAT IN A.TXT
A*.DAT	zajame A.DAT, AA.DAT, AAA.DAT
.	zajame vse datoteke s tekočega uporabniškega področja

Tega načina se poslužujemo tudi takrat, kadar iščemo datoteko, pa ne vemo natančnega imena, ampak samo nekaj znakov. Denimo, nek dopis lahko iščemo pod imenom DOPIS*.*, pa nam bo računalnik poiskal vse datoteke, katerih ime ustreza tej oznaki.

8.3 IZBIRA POGONA

Pri sistemu PARTNER WFG imamo na izbiro dva pogona: A - diskovni pogon in B - disketni pogon, pri sistemu PARTNER 2FG je A zgornji, B pa spodnji disketni pogon, pri sistemu 1FG pa je disketni pogon označen z A.

Če ne zahtevamo drugače, je naš pogon A, kar nam potrjuje tudi znak A> na zaslonu. Če pa hočemo pri WFG in 2FG pisati ali brati podatke z drugega pogona, moramo spremeniti pogon. To storimo z ukazom:

A>B:

Po vnosu dobimo v novi vrsti znak

B>

Kar pomeni, da smo spremenili pogon in to nam potrjuje tudi lučka na disketni enoti.

Nazaj se vrnemo z ukazom:

B>A:

in spet dobimo na zaslonu znak

A>

8.4 IZBIRA UPORABNIŠKEGA PODROČJA

Partner WFG, 2FG, 1FG - PRIROČNIK

Na disku in disketah se nahajajo vaše datoteke, ki so lahko programi ali podatki. Največje število datotek na disku je 1024 na disketi pa 128. Pregled nad njimi bomo imeli le ob primerni organizaciji po uporabniških področjih. S tem si zagotovimo nemoteno delovanje računalnika, zaščitimo se pred nenamernim brisanjem in pred nezaželenim kopiranjem. Operacijski sistem omogoča, da datoteke razdelimo po uporabniških področjih, jim damo ustrezne attribute in jih zaščitimo.

Disk oziroma disketa je logično razdeljena na 16 uporabniških področij. Njihova skupna velikost je omejena z velikostjo diska (diskete). Označimo jih s številkami od 0 do 15. Številka, ki se izpiše v ukazni vrstici, označuje aktivno delovno področje. Med delovnimi področji prehajamo z ukazom USER ali samo z številko in dvopičjem:

Primer:

Ukaz za prehod iz 3 uporabniškega področja v 4 je :

```
3A>user 4 <RET>
4A>
```

ali

```
3A>4: <RET>
4A>
```

Vsaki datoteki pripada le eno uporabniško področje, s tem so datoteke razdeljene v 16 skupin. Datoteke dobijo svojo uporabniško številko ob kreiranju. Te številke ne moremo spremeniti, razen če celotno datoteko z ukazom PIP prekopiramo na drugo področje.

Pri delu so nam dostopne le datoteke, ki se nahajajo na aktivnem delovnem področju in sistemske datoteke na delovnem področju 0.

Primer:

```
1A>dir
```

Ta ukaz bo izpisal le imena datotek, ki se nahajajo na delovnem področju 1.

8.5 DATOTECNI ATRIBUTI

Ena od možnosti za zaščito datoteke so njeni atributi-oznake. Vsaka datoteka ima najmanj dva atributa: prvi je lahko DIR ali SYS, drugi pa R/W ali R/O.

Atribut DIR pomeni, da je datoteka dostopna samo s svojega

Partner WFG, 2FG, 1FG - PRIROČNIK

uporabniškega področja .Če pa ima datoteka atribut SYS in če je shranjena na uporabniškem področju Ø, je dostopna z vseh uporabniških področij.

Atribut R/O (read only) pomeni, da je datoteka zaščitena proti pisanju oziroma brisanju. Oznaka R/W (read - write) pa pomeni, da te zaščitite ni.

Vsaka nova datoteka ima, če ne določimo drugače (ukaz SET), atributa DIR in R/W.

PRIMER: A>SET PODATKI.DATSDIR,RO

Ta ukaz priredi datoteki PODATKI.DAT atributa DIR in RO.

8.5.1 Sistemske in uporabniške datoteke

Znotraj vsakega uporabniškega področja datoteke razdelimo še v dve skupini: v sistemske in uporabniške. V skupino sistemskih datotek uvrstimo navadno programe in datoteke, ki jih programi uporabljajo, v drugo skupino pa uvrstimo vse ostale. Razdelitev je poljubna in nima posebnega pomena, z izjemo delovnega področja Ø.

V ti dve skupini datoteke uvrščamo z ukazom SET.

Primer:

A1>set profile.sub\$sys

S tem ukazom smo datoteko "profile.sub" uvrstili v skupino sistemskih datotek (dali smo ji atribut SYS).

A1>set *.dat\$dir

Ta ukaz vse datoteke tipa .dat uvrsti v skupino navadnih (nesistemskih) datotek (priredili smo jim atribut DAT).

Za sistemske zbirke pravimo, da imajo atribut SYS, za ostale pa DIR. Pri kopiranju datotek se hkrati kopirajo tudi ti atributi. Sistemske datoteke lahko kopiramo z ukazom PIP le če smo navedli stikalo R.

Primer:

A1>pip a:=b:.*

Ta ukaz bo iz diskete na disk prekopiral le datoteke z atributom DIR, medtem ko z ukazom:

A1>pip a:=b:.*\$r

prekopiramo vse datoteke (sistemske in nesistemske).

8.6 PREDLOG ZA RAZDELITEV DELOVNIH PODROČIJ

Datotek ne moremo poljubno uvrščati na katerokoli delovno področje, pri tem se moramo držati nekega reda.

Najpomembnejše je delovno področje 0, to je sistemsko delovno področje. Tu se nahaja operacijski sistem. Vse datoteke, ki imajo atribut SYS so dostopne s katerega koli drugega področja. Torej bomo na to področje poleg programov operacijskega sistema naložili še vse ostale programe, ki naj bodo na voljo vsem uporabnikom (urejevalnik teksta, prevajalnike, grafične programe in druge aplikacije, če ni drugače določeno v navodilih za instalacijo).

Na delovnem področju 0 ni priporočljivo imeti uporabniških datotek, saj nam le te tu delajo le zmedo. Na tem področju tudi ne priporočamo dela, ker si s tem povečujemo možnost napak.

Delovno področje 15 je namenjeno za MIPOS aplikacije, ostala delovna področja pa so med seboj enakovredna. Če isti sistem uporablja več uporabnikov, vsakemu uporabniku točno določimo njegovo uporabniško področje. Uporabniki, ki nimajo svojega področja, naj delajo na uporabniškem področju 1. Na istem uporabniškem področju naj bodo datoteke, ki smiselno sodijo skupaj. Npr. Področje 2 naj bo namenjeno za tekstovne datoteke, področje 3 za baze podatkov, področje 4 za delo z grafiko ipd.

Velikost diska je omejena na 10M bytov. To je velika pomnilna kapaciteta, vendar se ob delu z diskom hitro zapolni. Da se temu izognemo, poskrbimo, da se na disku nahajajo le datoteke, ki so za delo res potrebne. Datoteke, ki jih le redko uporabljamo ali so le dokumentacija, prekopiramo na diskete, da nam po nepotrebnem ne zavzemajo prostora.

Nekateri programi tvorijo pomožne datoteke. Tudi te nam zasedajo diskovni prostor, zato jih tedensko brišemo. Spoznamo jih po tipih .bak, .tmp, .***, .prn oz. .lst.

8.7 PREGLED NAJPOMEMBNEJSIH UKAZOV CP/M PLUS

Po abecednem redu bomo navedli tiste ukaze operacijskega sistema CP/M, ki jih vsakodnevno uporabljamo, z opcijami, ki jih najpogosteje potrebujemo. Opcije moramo vkleniti med oglati oklepaj in zaklepaj. Na koncu ukazne vrstice lahko oglati zaklepaj izpustimo. Ker na tastaturi ni oglatega oklepaja in zaklepaja, ju nadomeščamo takole:

Partner WFG, 2FG, 1FG - PRIROČNIK

- S oglati oklepaj
- C oglati zaklepaj

8.7.1 Ukaz DATE

NAMEN:

Ta ukaz izpiše trenutni datum, izpisuje datum neprekinjeno ali pa nastavi datum, ki se definira v ukazni vrstici.

ZGRADBA:

DATE	ugotavljanje časa in datuma
DATE mm/dd/11 uu:mm:ss	nastavitev časa in datuma
DATE SET	isto

ZGLEDI:

A>DATE

Sistem odgovori s tekočim datumom in časom, npr.

Wed 11/27/85 08:36:16

A>DATE 11/27/85 09:00:00

Sistem odgovori :

Press any key to set time.

Ko nastopi željeni čas, pritisnemo katerokoli tipko in s tem inicializiramo željeni čas in datum.

A>DATE SET

Sistem odgovori:

Enter today's date (MM/DD/YY):

Vpišemo mesec, dan, leto , nato sistem odgovori:

Enter the time (HH:MM:SS):

Vtipkamo uro, minute in sekunde, nato sistem odgovori:

Press any key to set time

8.7.2 Ukaz DIR

NAMEN:

S tem ukazom pregledamo seznam datotek, ki se nahajajo v direktoriju diska ali diskete. Ukaz DIR prikaže seznam datotek z atributom DIR. Ukaz DIRSYS (okrajšano DIRS) pa seznam datotek s SYS atributom.

ZGRADBA:

DIR
DIRS
DIR B:
DIR eod
DIR Šopcije
DIR eodŠopcije

eod: enolični opis datoteke (poglavje 2.3)
opcije: od opcij bomo navedli le dve: FULL in USER = ALL

ZGLEDI:

1A>DIR

Izpiše seznam datotek z atributom DIR na prvem uporabniškem področju. Če je področje prazno, izpiše:

No File.

Če ni DIR datotek, obstajajo pa datoteke z atributom SYS, izpiše:

SYSTEM FILE(S) EXIST.

A>DIRS B:

Izpiše vse SYS datoteke na disketi, ki se nahajajo na uporabniškem področju 0. Če le-teh ni, obstajajo pa DIR datoteke, izpiše:

NONSYSTEM FILE(S) EXIST.

1A>DIR B:

Izpiše seznam DIR datotek na prvem uporabniškem področju diskete.

Partner WFG, 2FG, 1FG - PRIROČNIK

A>DIR VAJA.TXT

Izpiše VAJA.TXT, če se le-ta datoteka nahaja na tem uporabniškem področju diska, sicer izpiše No File.

A>DIRS *.COM

Izpiše seznam vseh SYS datotek tipa COM

DIR z opcijami:

OPCIJE:

FULL izpiše seznam vseh datotek na disku (disketi) po abecedi, vključno s podatki o njihovi velikosti, atributih, zaščiti ipd.

USER=ALL izpiše iskano datoteko, če se nahaja na kateremkoli uporabniškem področju izbranega pogona.

ZGLEDI:

A>DIR B:SFULL

Izpiše abecedni seznam vseh datotek na disketi

A>DIR SEZNAMSUSER=ALL

Išče datoteko s tem imenom po vseh uporabniških področjih.

8.7.3 Ukaz ERASE

NAMEN:

Ta ukaz briše datoteko ali skupino datotek s tekočega uporabniškega področja, če seveda niso zaščitene z atributom RO ali s kakšno drugo zaščito. Pri brisanju več datotek hkrati zahteva sistem potrditev ukaza. Lahko pa tudi s stikalom CONFIRM dosežemo, da nas sistem za vsako datoteko posebej vpraša ali naj jo zbriše ali ne. Ukaz ERASE lahko okrajšamo v ERA.

ZGRADBA:

ERA d:imedat.tipSC

d: opis pogona vpišemo samo, če je B
ime: lahko tudi *
tip: lahko tudi *
C: okrajšano CONFIRM, ima pomen le, kadar brišemo več datotek hkrati

ZGLEDI:

A> ERA VAJA.TXT

Zbriše datoteko VAJA.TXT, če se le-ta nahaja na tekočem uporabniškem področju, sicer izpiše:

No File.

A>ERA *.DAT

Brisati želimo vse datoteke tipa DAT s tekočega uporabniškega področja. Sistem odgovori:

ERASE *.DAT (Y/N)?

Za potrditev odgovorimo z Y, sicer z N.

A>ERA MEMO*.*SC

Sistem bo začel izpisovati po vrsti imena vseh datotek, ki ustrezajo temu opisu, pri vsaki od njih pa bomo morali z izbiro med tipkama Y ali N potrditi ali negirati brisanje.

Partner WFG, 2FG, 1FG - PRIROČNIK

Na primer:

MEMOP.COM (Y/N) ? N
MEMO.COM (Y/N) ? Y
MEMOHELP.TXT (Y/N) ? Y

itd.

8.7.4 Ukaz KOPI

Program KOPI nam omogoča kopiranje z diskete na disketo na sistemu PARTNER 1FG in s tem nadomešča program PIP.

ZGRADBA:

A>KOPI ime_datoteke

Ime_datoteke je lahko polno ime ali delno ime dopolnjeno s standardnimi razširitvami CP/M (to so ? ali *)

Program interaktivno dela in zahteva menjavanje disket v pogonu. Ta navodila moramo natančno upoštevati.

Primer dela s programom KOPI (kopiranje datotek na enem pogonu)

A>KOPI TP.COM

KOPI --- single floppy PARTNER copy program V2.0

Iskra Delta

Insert source disc and press <RETURN>

?KOPI-I-copying TP.COM

Insert destination disc and press <RETURN>

?KOPI-S-all file(s) copied

To reboot the system insert system disc and press <RETURN>

A>

Primer:

A>KOPI *.*

(kopira vse datoteke)



8.7.5 Ukaz PIP

NAMEN:

PIP je program, ki kopira eno ali več datotek z enega uporabniškega področja na drugo, iz enega pogona na drugega, združuje več datotek v eno ter kopira iz / na pomožne enote. PIP kopira hkrati z datoteko tudi njene attribute. Hkrati s kopiranjem lahko datoteko tudi preimenujemo. Tekstovne datoteke lahko s PIP izpišemo na ekran ali tiskalnik. S PIP lahko kreiramo nove tekstovne datoteke. Če drugače ne definiramo, PIP kopira datoteke s tekočega pogona in uporabniškega področja. Po potrebi dodamo še posebne zahteve, stikala kopiranja.

KOPIRANJE ENE DATOTEKE

ZGRADBA:

PIP d:izhodnadat\$GnC=d:vhodnadat\$stikala

d: pogon; na levi strani novi, na desni stari
 izhodnadat: novo ime datoteke - pišemo samo v primeru,
 ko datoteko tudi preimenujemo
 Gn: n je številka uporabniškega področja, na
 katerega kopiramo
 vhdnadat: ime datoteke, ki jo kopiramo

STIKALA:

PIP ima na izbiro preko 20 stikal. Tu bomo navedli le tiste, ki jih najpogosteje uporabljamo:

Gn n je število med 0 in 15, ki pove, s
 katerega oz. na katero uporabniško področje
 kopiramo
 R kopira tudi datoteke z atributom SYS, ki jih
 brez te opcije zaobide
 V s to opcijo PIP kontrolira, če je kopiranje
 uspešno izvršeno.
 POZOR! PRI KOPIRANJU DATOTEK IZ DISKA NA
 DISKETO IN OBRATNO MORAMO TO OPCIJO OBVEZNO
 VNESTI!
 W prepiše preko datotek z enakim imenom in
 atributom RD, ki jih sicer zaobide
 O za kopiranje programskih datotek
 C selektivno kopiranje

POZOR! NA LEVI STRANI ENACAJA SMEMO UPORABITI SAMO OPCIJO G_n, VSE OSTALE PA LE NA DESNI STRANI.

ZGLEDI:

1. Kopiranje iz diska na disketo:

Iz 0A na 0B:

A>PIP B:=A:SEZNAMSV

Ker se obe uporabniški področji ujemata s tekočim (0), ju ni treba navajati. Opcija V na desni je za kontrolo uspešnega kopiranja.

Iz 1A na 1B:

A>PIP B:SG1C=A:SEZNAMSG1V

Iz 3A na 15B:

1A>PIP B:SG15C=A:VAJA.TXTSG3V

2. Kopiranje iz diskete na disk:

Iz 15B na 3A:

A>PIP A:SG3C=B:VAJA.TXTSG15V

Iz 5B na 1A:

1A>PIP A:=B:VAJA.TXTSG5V

Ker se uporabniško področje, s katerega kopiramo, ujema s tekočim, ga lahko na levi izpustimo.

3. Kopiranje na disket:

Iz 1A na 3A:

A>PIP A:5G3C=A:VAJA.TXT5G1V

KOPIRANJE VEČ DATOTEK:

ZGRADBA:

PIP d:5GnC=d:*.5stikala

d: na desni strani pogon , s katerega kopiramo,
na levi novi pogon
Gn: n je številka uporabniškega področja , na
katerega kopiramo
stikala: velja isto, kot pri kopiranju ene datoteke

POZOR! KER V OPISU DATOTEKE UPORABLJAMO * ALI ?, DATOTEKE NE MOREMO PREIMENOVATI, ZATO TUDI NA LEVI STRANI IMENA DATOTEKE NE PIŠEMO!

ZGLEDI:

A>PIP A:=B:*.COM5VRO

Kopira vse datoteke tipa COM(programske) in atributom SYS (opcija 0) z uporabniškega področja 0 na disketi na uporabniško področje 0 na disku.

A>PIP A:5G10C=A:DBASE.*5G15VRO

Kopira vse datoteke z imenom DBASE s 15. na 10. področje na disku.

A>PIP B:=A:*.5VWRO

Kopira vse datoteke z uporabniškega področja 0 na disku na uporabniško področje 0 na disketi.

8.7.6 Ukaz RENAME

NAMEN:

S tem ukazom damo datoteki novo ime, ki se zapiše v direktorij. Če datoteka, ki jo želimo preimenovali, ne obstaja, sistem javi:

No File

Če se novo ime datoteke ujema z imenom že obstoječe datoteke na istem uporabniškem področju, dobimo naslednje sporočilo: Not renamed:filename.typ already exists, delete (Y/N)?

kar pomeni vprašanje, ali želimo že obstoječo datoteko zbrisati, ali ne. Odgovorimo z Y za pritrditev oziroma z N za negativen odgovor.

Ukaz RENAME lahko okrajšamo na REN.

ZGRADBA:

REN novoimedat=staroimedat

ali

RENAME <RET>

Enter New Name:novoimedat

Enter Old Name:staroimedat

novoimedat= d:ime.tip

staroimedat=ime.tip

Opis pogona lahko izpustimo, če se ujema s privzetim.

ZGLEDI:

A>REN NOVADAT.TXT=STARADAT.TXT

Datoteka z imenom STARADAT.TXT se preimenuje v NOVADAT.TXT na disku.

A>REN B:NOVADAT.TXT=STARADAT.TXT

Datoteka STARADAT.TXT na disketi se preimenuje v NOVADAT.TXT.

8.7.7 Ukaz SET

NAMEN:

Z ukazom SET nastavimo attribute datoteki, zaščitimo datoteke z geslom, aktiviramo zapis datumov kreiranja, popravljanja in zadnjega dostopa do datotek, priredimo ime direktoriju diska ali diskete ter zaščitimo direktorij diska ali diskete z geslom. Tu si bomo ogledali le prvo možnost.

ZGRADBA:

SET eod\$stikala

eod: enolični opis datoteke (poglavje 2.3)
 stikala: atributi DIR, SYS, RO, RW

ZGLEDI:

A>SET ALFA.TXT\$SYS,RO

Z ukazom smo nastavili datoteki ALFA.TXT atributa SYS in RO. Sistem nas o tem obvesti s sporočilom:

A:ALFA .TXT set to system (SYS), Read Only (RO)

Sedaj je datoteka dostopna z vseh uporabniških področij.

A>SET ALFA.TXT\$RW

Atribut RO spremenimo v RW ter s tem odstranimo zaščito.

A>SET B:BETA.DAT\$DIR

Datoteki BETA.DAT na disketi priredimo atribut DIR.

8.7.8 Ukaz SHOW

NAMEN:

SHOW ukaz prikaže razne informacije o diskih in disketah. Tu si bomo ogledali le naslednje:

- koliko je še praznega prostora na disku ali disketi
- pokaže nam številke zasedenih uporabniških področij in število datotek na vsakem uporabniškem področju.

ZGRADBA:

SHOW d:
 SHOW d:SSPACE
 SHOW d:SUSERS

ZGLEDI:

A>SHOW

Sistem sporoči:

A:RW, Space: 9,488k
 B:RW, Space 16k

Ukaz prikaže, da ima enota A 9.488 kilobytov prostora, enota B pa 16 kilobytov.

A>SHOW B:

Sistem sporoči:

B:RW, Space: 16k

Enota B ima 16 kilobytov prostora.

A>SHOW B:SUSERS

Sistem sporoči:

B: Active user : 3
 B: Active Files : 0 3 4 11
 B: # of files : 36 3 3 22
 B: Number of free entries: 53

Partner WFG, 2FG, 1FG - PRIROČNIK

Pri tem ukazu se izpiše:

- številka tekočega uporabniškega področja
 - številke uporabniških področij, kjer se nahajajo datoteke
 - pod vsako številko število datotek na tem področju
- Na koncu izpiše število prostih mest v direktoriju enote.

8.7.9 Ukaz SUBMIT

NAMEN:

Ukaz SUBMIT omogoča izvrševanje skupine ukazov iz datoteke tipa SUB.

Običajno vnašamo ukaze po enega naenkrat in jih tako tudi izvršujemo. Če je potrebno izvrševanje istega zaporedja ukazov več kot enkrat, bo ugodno, zbrati te ukaze v posebni datoteki tipa SUB in jih izvršiti po vrsti, tako kot so bili vnešeni v datoteko, s samo enim ukazom SUBMIT. Datoteki damo svoje ime, njen tip pa mora biti SUB.

Datoteka tipa SUB sme vsebovati ukaze CP/M 3+, vgnezdene ukaze SUBMIT in vhodne podatke za sistemski ukaz ali program.

Argumente smemo vnašati v te datoteke med njihovim izvrševanjem. Vsak argument, ki ga vtipkamo, bo prirejen nekemu parametru v datoteki. Prvi argument nadomesti vsak nastop \$1 v datoteki. Drugi argument stopi na mesto \$2 in tako naprej do zadnjega argumenta, ki more nadomestiti najvišji še dovoljeni parameter \$9.

Npr. datoteka START.SUB pomeni ukaze:

```
ERA $1.BAK
DIR $1
PIP $1=A:$2.COM
```

nato vtipkamo ukaz SUBMIT:

```
A>SUBMIT START SAM TEX
```

SUBMIT potem tvori datoteko z nadomeščenimi parametri in jo izvrši:

```
ERA SAM.BAK
DIR SAM
PIP SAM=A:TEX.COM
```

Če vnesemo manj argumentov v ukaz SUBMIT kot je parametrov v datoteki tipa SUB, preostalih parametrov ne bo v ukazih.

Če pa smo vnesli več argumentov v ukaz SUBMIT, kot pa je na voljo parametrov v datoteki tipa SUB, sistem preostale argumente zanemari.

Za vnos resničnega znaka \$ v datoteko tipa SUB, torej ne kot del parametra, vtipkamo dva dolarska znaka skupaj: \$\$\$. SUBMIT ju nadomesti z enim samim med postavljanjem argumentov na mesto parametrov v datoteki tipa SUB. Za primer vzamemo datoteko AA.SUB, ki naj vsebuje med drugimi vrstico:

```
MAC $1 $$$2
```

nato vtipkajmo ukaz SUBMIT:

A>SUBMIT AA ZZ SZ

prevedena datoteka bo vsebovala:

MAC ZZ \$SZ

Programske vhodne vrstice v datoteki tipa SUB
 =====

Datoteka tipa SUB sme vsebovati programske vhodne vrstice. Vsakemu programskemu vhodu predhodi znak "<" kot to prikazuje naslednji primer:

```
PIP
<B:/*.ASM
<CON:=DUMP.ASM
<
DIR
```

Prve tri vrstice, ki sledijo PIP, so vhodne vrstice za ukaz PIP. Tretjo vrstico predstavlja samo znak <, ki pomeni <cr>. Ta znak RETURN pove ukazu PIP, da naj se vrne v sistem, da bo ta mogel izvršiti ukaz DIR.

Ce se program zaključi prej kot je izčrpal ves svoj vhod, SUBMIT zanemari odvečne vhodne vrstice in izda obvestilo:
 Warning: Program input ignored

Ce pa zahteva program več vhodov kot jih je na voljo v datoteki tipa SUB, pričakuje vnos preostalih vhodov preko tipkovnice.

Dovoljeno je vnašati kontrolne znake v datoteko SUB na običajen način, tako da vtipkamo znak "C" in potem črko, ki naj bo pretvorjena v kontrolni znak. Za vnos znaka "C" moramo to vtipkati dvakrat zapored. Ta kombinacija se pretvori v en sam znak na enak način, kot se prevede \$\$ v znak \$.

Datoteka tipa SUB
 =====

Datoteka tipa SUB sme vsebovati sledeče tipe vrstic:

- Vsak veljaven ukaz iz CP/M 3+
- Vsak veljaven ukaz iz CP/M 3+ s parametri SUBMIT
- Vsako vhodno vrstico s podatki
- Vsako programsko vhodno vrstico s parametri (\$0 do \$9).

Ukazne vrstice v CP/M 3+ ne smejo preseči 128 znakov.

Izvršitev ukaza SUBMIT
 =====

SUBMIT

SUBMIT ImeDatoteke
 SUBMIT ImeDatoteke argument ... argument

Če vtipkamo samo SUBMIT, nas sistem sam povpraša za ostanek ukaza. Vstavimo opis datoteke in argumente.

Če se želimo izogniti stalnemu pisanju ukaza SUBMIT, uporabimo ukaz:

SETDEF SORDER=(COM,SUB)C

S tem smo določili, da se ukazne datoteke (.SUB) kličejo enako kot, običajne programske datoteke (.COM).

Zagonska datoteka (Start-up File) PROFILE.SUB

=====

Vsakokrat ko vklopimo ali postavimo v osnovno stanje (reset) svoj računalnik, bo sistem CP/M 3 sam začel iskati posebno datoteko za izvrševanje, ki se imenuje PROFILE.SUB. Če te datoteke v sistemu ni, bo ta začel z normalno operacijo. Če pa obstaja PROFILE.SUB, bo sistem izvršil ukaze, ki jih ta datoteka vsebuje. Ta datoteka je priročna za uporabo takrat, ko redno izvršujemo določeno serijo ukazov pri vključitvi računalnika.

Z uporabo te lastnosti se bo vedno izvršilo zaželeno zaporedje ukazov prej kot bomo pričeli z rednim delom na računalniku.

8.7.10 Ukaz TYPE

NAMEN:

TYPE izpiše vsebino ASCII datotek na ekran. Izpis datoteke se zaustavi s CTL-S in nadaljuje s CTL-Q, prekinemo pa ga lahko s CTL-C. Izpis je lahko tudi na tiskalnik, če pred <RET> odtipkamo CTL-P. Če navedene datoteke ni, sistem izpiše:

No File

Ukaz lahko okrajšamo v TYP.

ZGRADBA:

TYPE <RET>
TYP eodStikala

eod: enolični opis datoteke (glej poglavje 8.2.1.)
stikala: PAGE in NOPAGE
PAGE: pomeni, da se izpis na ekran zaustavi, ko se ekran napolni, na zaslону se prikaže sporočilo:

Press Return to Continue

kar pomeni, da je treba za nadaljevanje prikaza pritisniti tipko <RET>

NOPAGE: pomeni, da se izpisuje neprekinjeno.

Če opcije izpustimo, je privzeto stanje PAGE, naenkrat se prikaže 24 vrstic besedila na zaslону.

ZGLEDI:

A>TYP TEXT.TXT

Izpiše se vsebina datoteke TEXT.TXT, ki se nahaja na privzeti enoti.

A>TYPE

Če ne navedemo imena datoteke, sistem sporoči:
Enter file:

Partner WFG, 2FG, 1FG - PRIROČNIK

Vpišemo TEXT.TXT in dobimo isto kot v prvem primeru.

A>TYP B:PROGRAM.PASSNOPAGE

Izpiše vsebino datoteke PROGRAM.PAS, ki je na disketi, neprekinjeno na zaslon.

8.7.11 Ukaz USER

NAMEN:

USER spremeni številko uporabniškega področja na številko, ki je navedena v ukazni vrstici.

ZGRADBA:

USER
USER n

n številka zelenega uporabniškega področja (0 .. 15)

ZGLEDI:

A>USER

Sistem zahteva:

Enter user#:11

in odgovori:

11A>

A>USER 13

Sistem odgovori z:

13A>

9. SISTEMSKO DELO

Mikroračunalniški sistem PARTNER vam pomaga pri reševanju zamudnih, rutinskih, ponavljajočih in zahtevnih opravil. Da bo te naloge lahko učinkovito opravil, morate poskrbeti za pravilno organizacijo podatkov (datotek) na diskih. Le tako bo PARTNER res pravi partner pri vašem delu.

Na disku se nahajajo programi, ki so namenjeni reševanju vaših problemov (aplikacije) in programi, ki omogočajo delo računalniku (sistemski programi). Brez sistemskih programov ali pri nepravilni organizaciji le-teh računalnik ne bo deloval.

V tem poglavju bomo opisali vsa sistemska opravila, ki jih mora uporabnik periodično opravljati, da omogoči brezhibno delovanje sistema.

9.1. FORMATIRANJE DISKET

```
-----
!                               OPOZORILO !!!                               !
!                               !                               !
!      NE FORMATIRAJTE DISKET S PROGRAMI IN PODATKI.      !
!                               FORMATIRANJE UNICI VSEBINO DISKETE !                               !
-----
```

Novo diskete niso formatirane, to pomeni, da nanje še ne moremo zapisovati. S formatiranjem se na disketi določijo sektorji in steze, ki pozneje sprejmejo uporabnikove podatke in programe.

Kapaciteta formatirane diskete znaša za uporabnika 146 stez x 18 sektorjev x 256 znakov.

Na novo lahko formatiramo tudi diskete, katerih vsebine ne potrebujemo več, vendar jih želimo uporabljati za

Partner WFG, 2FG, 1FG - PRIROČNIK

shranjevanje drugih podatkov oz. programov. S formatiranjem se namreč vsebina diskete zbríše, zato tako disketo lahko ponovno uporabljamo kot novo.

Za formatiranje disket je na disku zapisan poseben program FORMAT.

Postopek pri formatiranju:

1. Vtipkajte ukaz

A> FORMAT <RETURN>

Na zaslonu se pojavi menu in sporočilo:

Ali je disketa v pogonu B: potrebna formatiranja (D/N)

2. Vtipkajte

"D"

in program FORMAT začne formatirati disketo.

3. Ko je formatiranje uspešno izvedeno (čas formatiranja je približno 3 minute), se na zaslonu prikaže sporočilo:

USPEŠNO FORMATIRANJE

A>

Zdaj je disketa formatirana in lahko sprejme uporabnikove programe in podatke.

9.2. ZAGON RACUNALNIKA IN OPERACIJSKI SISTEM

Operacijski sistem CP/M za delovanje potrebuje naslednje datoteke:

- CPM3.SYS :

Operacijski sistem, ki je sestavljen iz dveh delov. BDOS (Basic Disk Operating System) je osnovni diskovni operacijski sistem, ki skrbi za krmiljenje in upravljanje pomnilnikov. BIOS (Basic Input Output System) ima nalogo krmiljenja vhodno izhodnih enot.

- CCP.COM :

(Console Command Processor) - konzolni krmilni program je procesor konzolnih ukazov. Po nalaganju prevzame vodenje sistema, branje uporabnikovih ukazov in prenašanje na druge module.

- začetni nalagalnik :

Partner WFG, 2FG, 1FG - PRIROČNIK

omogoči nalaganje sistema takoj po vključitvi. Na disku/disketi se ne nahaja v obliki datoteke. Zapisan je na zunanjih sledih diska/diskete.

Za lažje razumevanje si oglejmo dogajanje ob vključitvi računalnika. Ko računalnik vključimo, je delovni pomnilnik prazen - brez operacijskega sistema. Postopek vpisa operacijskega sistema v delovni pomnilnik imenujemo nalaganje sistema.

Poteka v več fazah. Računalnik vsebuje 4K pomnilnika tipa EPROM, v katerem se nahaja preprost program, ki se izvrši takoj ob vključitvi in sproži nalaganje operacijskega sistema. V delovni pomnilnik se najprej vpiše začetni nalagalnik, ki se nahaja na sistemskem disku/disketi. Šele ta program prenese datoteko CPM3.SYS v delovni pomnilnik ter pripravi računalnik za delo.

Konzolni krmilni program CCP.COM skrbi za delo s tipkovnico in poskrbi za izvrševanje vtipkanih ukazov. Brez tega programa sploh ne moremo začeti z delom.

Celoten postopek začetnega nalaganja se ob vključitvi računalnika avtomatično izvede. Po uspešnem zagonu se operacijski sistem javi z znakom pripravljenosti:

A>

oziroma se izvede ukazna procedura definirana z datoteko PROFILE.SUB. Tu se nahaja tudi klic začetnega menija.

9.3. GENERIRANJE SISTEMA DISKA / DISKETE

Sistemsko disketo/disk uporabljamo vedno pri zagonu računalnika. Na njej morajo biti vsi elementi, potrebni za zagon, ki smo jih navedli v prejšnjem poglavju.

Sistemski disk/disketo generiramo po naslednjem postopku:

1. Disk/disketa mora biti formatirana, vendar ni nujno, da je prazna.
2. Na disk/disketo kopiramo datoteki CPM3.SYS in CCP.COM. Uporabimo ukaz PIP oz. KOPI pri sistemu 1FG.
3. S posebnim programom na zunanje sledi diska/diskete zapišemo začetni nalagalnik. Imenujmo ga zapisovalni program.

Različne inačice PARTNERJEV potrebujejo različne operacijske sisteme, nalagalnike in zapisovalne programe. Naslednja tabela prikazuje potrebne elemente za posamezno inačico.

Partner WFG, 2FG, 1FG - PRIROČNIK

inačica	OS	nalagalnik	zapisovalni prog.
disk WFG	CPM3.GDP	WLDR.COM	PUTWSYS.COM
disketa WFG	CPM3.GSR	GFLDR.COM	GPUTFSYS.COM
disketa 2FG	CPM3.2FG	G2FLDR.COM	GPUT2FSYS.COM
disketa 1FG	CPM3.1FG	GFLDR.COM	GPUTFSYS.COM

V tabeli vidimo, da imajo operacijski sistemi pri različnih inačicah različna imena tipov. Tako jih označujemo, da jih ločimo med sabo, vendar moramo poskrbeti, da se na katerikoli kopiji imenuje operacijski sistem CPM3.SYS.

Datoteka CCP.COM je za vse sisteme enaka. Vedno jo naložimo na sistemsko disketo/disk na uporabniško področje 0 in ji priredimo atribut SYS.

9.3.1. Generiranje systemskega diska/diskete za WFG

Pri tej inačici se navadno sistem nalaga z diska. Disketo uporabimo le takrat, ko pride do napak pri nalaganju z diska (glej poglavje 12) ali po uspešnem formatiranju diska.

Disk in systemska disketa sta tovarniško opremljena s sistemom, zato po nakupu računalnika sistema ni potrebno generirati.

Priporočamo vam, da si takoj naredite kopijo systemske diskete in originala za delo ne uporabljate. Original shranite na varno mesto. Iz originalne systemske diskete ne snemajte zaščitne nalepke.

Kopijo systemske diskete naredite po naslednjem postopku:

1. Na disku najdete prazno uporabniško področje in vsebino originalne systemske diskete prekopirate na to področje.

```
PRIMER:
A>SHOW SUSERS (vtipkan ukaz)

A: Active user      : 0 (odgovor sistema)
A: Active files    : 0 3 4 11
A: # of files      : 36 3 3 22
A: Number of free entries: 960
```

Ugotovimo, da so prazna področja 1, 2, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15. Za kopiranje izberimo npr. področje 1.

```
A>PIP A:$G1C=B:*.*$VR (vtipkan ukaz)
```

2. Na prazno, že formatirano disketo prepisemo vsebino izbranega področja:

A>PIP B:=A:*. *SVRG1 (vtipkan ukaz)

3. Zapišemo na zunanje sledi nalagalnik:

A>B:GPUTFSYS (vtipkan ukaz)

S tem smo prekopirali celotno sistemsko disketo. Sistemsko disketa vsebuje vse datoteke (programe), ki se nahajajo tudi na disku. Vse te datoteke potrebujemo za delo.

Če na disku ni sistema (po formatiranju diska), ravnamo po naslednjem postopku:

1. Vstavimo sistemsko disketo v disketni pogon.
2. Resetiramo računalnik in ko se pojavi napis TESTING MEMORY, na tipkovnici pritisnemo tipko CTL-C. Za tem se sistem javi z *.
3. Vnesemo lahko ukaza :
A - povzroči nalaganje sistema diska
F - povzroči nalaganje sistema s systemske diskete

Izberemo ukaz F.

4. Ko se sistem javi z znakom pripravljenosti B>, prekopiramo z diskete potrebne datoteke.

PRIMER: če prekopiramo celotno vsebino systemske diskete, uporabimo ukaz:

B>PIP A:=B:*. *SVRW
Obvezno moramo prekopirati vsaj datoteko CCP.COM.

5. Poskrbimo za pravilno datoteko CPM3.SYS.

PRIMER:

B>PIP A:CPM3.SYS=B:CPM3.GDPSVRW

6. Zapišemo še nalagalnik.

PRIMER:

B>PUTWSYS

7. Resetiramo računalnik. Če smo uspešno naložili sistem, se mora javiti znak pripravljenosti A> ali ustrezni menu.

9.3.2. Generiranje systemske diskete za 2FG

Sistem nalagamo vedno s systemske diskete, ki jo vstavimo v zgornji disketni pogon (A). Priporočamo, da naredite kopijo originalne systemske diskete. Originalno disketo shranite in je ne uporabljajte.

Kopijo systemske diskete naredimo takole:

1. V pogon B (spodnji pogon) vstavimo prazno, formatirano disketo in vsebino originalne systemske diskete prekopi-ramo.

```
A>PIP B:=A:*. *SVR
```

2. Na začetne sledi zapišemo nalagalni program. V pogon A vstavimo originalno systemsko disketo, v pogon B pa kopijo in vpišemo ukaz:

```
A>PUT2FSYS
```

9.3.3. Generiranje systemske diskete za 1FG

Sistem nalagamo vedno le s systemske diskete, ki jo vstavimo v disketni pogon. Priporočamo, da naredite kopijo originalne systemske diskete. Originalno disketo shranite in je ne uporabljajte.

Kopijo systemske diskete naredimo takole:

1. Najprej naredimo kopijo vseh datotek, ki se nahajajo na systemski disketi. Pri sistemu 1FG imamo samo en disketni pogon, kar nam otežuje enostavno kopiranje datotek z ene diskete na drugo. Če vam je dostopen sistem 2FG ali WFG potem kopiranja opravimo na takšnem sistemu (glej prejšnja dva razdelka), drugače uporabimo program KOPI.

PRIMER:

```
A>KOPI *.*
```

2. Na začetne sledi zapišemo še nalagalnik. Najprej ponovno v disketni pogon vstavimo originalno systemsko disketo nakar vpišemo ukaz:

```
A>PUTFSYS
```

Zatem točno sledimo navodilom, ki se izpisujejo na ekranu.

Originalna systemska disketa je praktično polna, vendar pri delu vedno ne potrebujemo vseh datotek (programov). Na kopije lahko naložimo poljubne datoteke, od datotek z

Partner WFG, 2FG, 1FG - PRIROČNIK

originalne sistemske diskete pa samo tiste, ki jih pogosto rabimo (npr.: DIR.COM, PIP.COM, ERASE.COM, TYPE.COM). Na vseh disketah, ki jih uporabljamo kot sistemske, morata obvezno biti datoteki CPM3.SYS in CCP.COM.

Sistemska disketa služi le za zagon sistema. Takrat mora biti v pogonu A sistemska disketa. Med delom lahko sistemsko disketo zamenjamo, tudi s takšnimi na katerih ni sistema, vendar mora na vsaki disketi biti datoteka CCP.COM. Ta datoteka mora biti na uporabniškem področju 0, z atributom SYS.

9.4 INSTALIRANJE DOKUPLJENE PROGRAMSKE OPREME

Računalnik s programi operacijskega sistema bi bil le samemu sebi namen, zato potrebujete še dodatno programsko opremo - aplikativne programe oziroma programska orodja (prevajalnike).

Če ste dokupili dodatne programske izdelke, ste dobili takšen izdelek na disketi. Za takšno disketo velja isto, kot za sistemsko disketo, da je za delo ne uporabljajte. Priporočamo vam, da takoj napravite kopijo, originalno pa uporabljate le v primeru izgube podatkov na kopiji.

Ob disketah dobite tudi pripadajočo dokumentacijo. Če tam ni drugače definirano, potem velja naslednji postopek instalacije :

WFG sistem:

- V disketno enoto vstavimo disketo s programskim izdelkom, ki ga želimo instalirati in vtipkamo ukaz:

```
A>PIP A:=B:*.SVR
```

Pri tem se vse datoteke z diskete preprišejo na disk, na uporabniško področje 0.

- Če datoteke, ki smo jih prekopirali nimajo atributa SYS, jim priredimo takšen atribut z ukazom SET, le tako jih bomo lahko uporabljali na kateremkoli uporabniškem področju.
- Originalno disketo shranimo na varno mesto in je za delo ne uporabljamo. Če določene programe redkeje uporabljamo in imamo disk že zelo zaseden, lahko aplikativne programe kličemo tudi z diskete. Seveda za to uporabljamo kopijo originalne diskete. Če mnogo delamo z disketami je smiselno uporabiti ukaz:

```
A>SETDEF A:,*
```

S tem ukazom določimo vrstni red iskanja datotek na disku. Operacijski sistem najprej pregleda trenutno aktiven disk,

Partner WFG, 2FG, 1FG - PRIROČNIK

zatem pa še disk A. Ta ukaz ponavadi vključujemo, kar v ukazno proceduro PROFILE.SUB.

- Če za delo uporabljate menujsko tehniko, nove programe vključite v menuje.

2FG in 1FG sistem:

- Najprej naredimo kopijo originalne diskete.

Pri sistemu 2FG to naredimo takole: v pogon A vstavimo originalno disketo, v pogon B pa prazno formatirano disketo in vpišemo ukaz:

```
A> pip B:=A:*. *Svr
```

S tem smo prekopirali vse datoteke iz originala na kopijo.

Za sistem 1FG uporabimo za kopiranje ukaz:

```
A>KUPI
```

- Za olajšano delo na kopijo dodamo programe, ki jih dodatno potrebujemo. Npr. pri prevajalnikih še urejevalnik teksta. Tako lahko na isto disketo zlijemo več različnih aplikacij. Priporočamo da na vsako disketo dodatno prekopirate še datoteko CCP.COM.
- Pri delu z disketami hitro napolnimo disketo. Zato na delovnih disketah hranimo le nujno potrebne datoteke, ostale pa brišemo. Pri aplikativnih programih so na disketi dodane tudi datoteke, ki jih za delo ne potrebujemo. (To so priročniki in pomožni programi)

9.5. ZASČITA PODATKOV PRED IZGUBO

Podatki predstavljajo veliko vrednost, zato jih nikakor ne smemo izgubiti. Obstaja pa nevarnost (čeprav zelo majhna), da se disk pokvari, ali pa da pride zaradi človekove napake do brisanja podatkov na disku. Na primer, dali smo računalniku ukaz, da določene podatke briše, kasneje pa ugotovimo, da jih še potrebujemo. Temu se ne moremo izogniti, zato moramo vedno poskrbeti za kopijo podatkov (back-up). Če se torej na disku zaradi napake ali okvare podatki izgubijo, nam še vedno ostanejo njihove kopije na disketah.

Zagotavljanje kopij podatkov, ki so bili vnešeni v računalnik, spada med zelo pomembna opravila. Vrednost podatkov ugotovimo ponavadi šele tedaj, ko jih nimamo več, zato ne preverjamo njihove vrednosti na tak način - raje si

Partner WFG, 2FG, 1FG - PRIROČNIK

redno zagotavljajmo kopije na disketah.

Po vsakokratnem spreminjanju (ažuriranju) podatkov moramo ažurirati tudi kopijo, saj bomo tako ob izgubi osnovnih podatkov imeli najmanj dela z obnovitvijo zadnjega stanja.

Kopije podatkov lahko delamo s programom PIP. Obstaja stikalo (opcija) A, ki omogoča kopiranje samo tistih datotek, ki smo jih spreminjali po zadnjem ažuriranju.

PRIMER:

Želimo narediti kopijo vseh datotek tipa .DAT. Če želimo ugotoviti, katere datoteke smo spreminjali, uporabimo ukaz

A>DIR *.DATSFU

Primer izpisa:

Oznaka Arch označuje datoteke, ki niso bile spreminjane od zadnjega ažuriranja.

Naslednji ukaz kopira le datoteke brez oznake Arch.

A>PIP B:=A:*.DATSAV

Sedaj imajo vse datoteke tipa .DAT oznako Arch. Izgubijo jo takoj, ko takšno datoteko kakorkoli spremenimo.

Če je velikost datoteke na winchester disku večja od kapacitete diskete (664 K Bytov), je ne moremo prekopirati z ukazom PIP. V ta namen je izdelan program BACKUP, ki večje datoteke kopira na več disket - podobno kot na magnetni trak.

Program BACKUP dobite ob aplikativnih programih skupaj z navodili.

Ukaza PIP prav tako ne moremo za delo kopij uporabljati na 1FG sistemu. Na sistemski disketi je dodan program KOPI, ki je opisan v poglavju 8.7.4.

Da boste natančno vedeli, na kateri disketi se nahajajo kopije določenih podatkov vam priporočamo, da jih opremite z nalepkami, ki so priložene disketam. Na nalepko napišite imena datotek in datum kopiranja. S tem načinom bi morali pogostokrat menjavati nalepke, zato lahko priložite tudi list, na katerega predhodno stiskate direktorij (uporabite ukaz DIR in pred pritiskom na tipko <RET> pritisnite CTL-P).

10. PARTNER TERMINAL

=====

Partner je sestavljen iz dveh glavnih digitalnih in več pomožnih vezij. Prvo vezje je CPU s 128K hitrega pomnilnika, drugo pa je sestavljeno iz grafičnega in tekstovnega dela. To vezje sestavlja skupaj z ekranom in tipkovnico terminal Partnerja. Pri večjih računalniških sistemih sta računalniški in terminalski del navadno ločena, pri Partnerju pa sta vgrajena v isto ohišje.

Grafični del je realiziran s pomočjo grafičnega procesorja THOMPSON EF 9367. Grafična slika je shranjena v 128K hitrega pomnilnika (RAM), kar omogoča hkratno shranjevanje dveh slik, vsaka zavzema po 64K hitrega pomnilnika. Grafično sliko lahko krmilimo le programsko, zato so informacije za uporabo grafike zbrane v Dodatku F.

Neodvisno od grafične slike se na ekranu prikazuje tekstovna slika. To pomeni, da imamo na ekranu lahko samo tekst, samo grafično sliko ali pa kombinacijo obojega.

Za shranjevanje teksta je namenjenega 4K pomnilnika in še 4K pomnilnika za attribute. Znakovni generator lahko izpiše 256 različnih vnaprej definiranih znakov. Razen tega je v pomnilniku prostora še za 128 znakov, ki jih definira uporabnik.

Tekstovni del lahko deluje v treh različnih načinih delovanja: VT-100, VT-52 in PARTNER. Način delovanja je odvisen od programa, ki ga izvajamo, zato moramo pred izvajanjem programa ustrezno nastaviti terminal.

Različne vrste znakov, način delovanja in še nekatere druge funkcije terminala nastavimo s tipko SET-UP. Vse terminalske funkcije in njihovo programsko krmiljenje je opisano v Dodatku B.

10.1 SET-UP

Tipko SET-UP uporabljamo za spreminjanje in nastavljanje načina delovanja Partnerjevega terminala. Nekatere nastavitve so določene s programom, ki ga želimo izvajati, druge pa si lahko poljubno nastavimo in si s tem olajšamo delo. Terminal lahko nastavljamo kadarkoli, tudi med izvajanjem programa. Po pritisku te tipke se izvajanje programa začasno prekine, v zadnjih petih vrsticah se izpiše set-up menu. Ponovni pritisk SET-UP tipke vrne staro sliko in obnovi izvajanje prekinjenega programa.

```
-----
                S e t   u p
1234567890123456789012345678901234567890123456789012345
Clear all tabs                Set all tabs to 8
Tabulators      Term. & Kbd.   Screen      Save & Reset
-----
```

slika 10.1 SET UP MENU (tabulators)

Slika 10.1 prikazuje menu za nastavitve tabulatorjev. V zadnji (spodnji) 26 vrstici vedno dobimo informacijo o trenutno aktivnem meniju. Naslov aktivnega menija je izpisan v inverzni sliki (temne črke na svetli podlagi). Naslovi menuev v slovenščini pomenijo sledeče:

- tabulators - Nastavitev tabulatorjev
- ter. & kbd - Nastavitev terminala in tipkovnice
- screen - Nastavitev zaslona
- save reset - Shranjevanja in klic začetnega stanja

Menuje izbiramo s tipkama <PF3> in <PF4>. Ob pritisku tipke PF4 dobimo na ekranu sliko, ki jo ponazarja slika 10.2. Vidimo, da je sedaj aktiven menu "ter. & kbd". Ponovni pritisk tipke PF4 povzroči prehod v naslednji menu. S tipko <PF3> preidemo nazaj v prejšni menu. ("tabulators").

V vsakem meniju je več različnih funkcij. Izpisane so v 24. in 25. vrstici. Ime funkcije, ki je napisano v inverznem načinu označuje funkcijo, ki jo spreminjamo. Med funkcijami izbiramo s tipkama <PF1> in <PF2>. Pritisk tipke PF2 povzroči prehod k naslednji funkciji, <PF1> pa k predhodni. Zraven vsake funkcije je izpisana trenutna nastavitev. Te spreminjamo s tipkama <gor> "puščica gor" in <dol> "puščica dol".

Postopek nastavitve terminala je torej sledeč:

1. Pritisnemo <SET-Up> tipko.
2. S tipkama <PF3> in <F4> izberemo ustrezen meni.
3. S tipkama <PF2> in <PF1> izberemo ustrezno funkcijo.
4. S tipkama <GOR> in <DOL> nastavimo način delovanja.
5. Če nastavljamo več funkcij, ponavljamo točki 2 in 3.
6. S tipko <SET UP> zaključimo delo.

ESC sekvence za programsko nastavljanje funkcij terminala, so zbrane v Priročniku za programerje.

10.2 NASTAVITEV TABULATORSKIH MEST (tabulators)

Tabulatorska mesta so točke v vrstici na ekranu, kamor skoči sledilni znak ob pritisku tipke <TAB>. Ta mesta imajo isto vlogo, kot tabulatorji na pisalnem stroju. Navadno je vsako osmo mesto tabulatorsko mesto. Ta mesta lahko tudi poljubno nastavimo in si s tem prihranimo večkratno zaporedno uporabo tipke <PRESLEDEK> in lažje oblikujemo izpis na ekranu. To je posebej važno pri vnosu teksta. Nekateri urejevalniki teksta ne uporabljajo terminalskih tabulatorskih mest, zato je pri takšnih urejevalnikih potrebno posebej programsko določiti ta mesta.

V prvi vrstici menija (slika 10.1) so tabulatorska mesta označena s svetlejšim izpisom.

Brisanje vseh tabulatorskih mest. (CLEAR ALL TABS)

S tipkami <GOR>, <DOL> ali <PRESLEDEK> odstranimo vsa tabulatorska mesta.

Postavitev tabulatorskih mest na vsako osmo. (SET ALL TABS TO 8)

S tipkami <GOR>, <DOL> ali <PRESLEDEK> določimo, da je vsako osmo mesto tabulatorsko.

Poljubno postavljanje tabulatorskih mest.

S tipko <PF4> preidemo s sledilnim znakom v prvo vrstico menija. S tipkama <LEVO> in <DESNO> izberemo mesto v vrstici. S tipko <GOR> na tem mestu postavimo tabulator, s tipko <DOL> na tem mestu brišemo tabulator.

10.3 NASTAVITEV TERMINALA IN TIPKOVNICE (TERM & KBD)

Partner terminal lahko emulira (posnema delovanje) več različnih terminalov. Deluje lahko tudi z različnimi tipkovnicami in različnimi načini delovanja. Napačna nastavitev funkcij v tem menuju povzroči, da na ekranu dobimo nepravilne slike - kar pomeni, da tipkovnica pošilja napačne znake.

10.3.1 Tip terminala (Terminal type)

Partner deluje v treh različnih standardih za terminale (ANSI, VT52 in PARTNER). Programi so pisani po različnih standardih, zato je treba pred izvajanjem programa ustrezno nastaviti tip terminala. Nekateri programi sami nastavijo pravi tip. Če je tip terminala nepravilno nastavljen, se na ekranu slika nepravilno izpiše (navadno v eni vrstici).

PARTNER terminal zahtevajo programi, ki so bili pisani za starejšo negrafično verzijo Partnerja. Če grafični Partner deluje v PARTNER načinu, je popolnoma kompatibilen (enako deluje), kot negrafični Partner.

V ANSI načinu deluje večina programov. VT52 način se redkeje uporablja, največkrat če Partnerja uporabljamo kot inteligentni terminal drugih računalniških sistemov.

10.3.2 Nabor znakov (Terminal language)

S to funkcijo izbiramo nabor znakov, ki se izpisujejo na ekranu. Na voljo imamo naslednje nabore znakov:

USASCII
UKASCII
SPANISH
FRENCH
GERMAN
ITALIAN
DANISH
SWEDISH
YUGOSLAV

Navadno izberemo isti nabor znakov, kot je napisan na tipkovnici (YUGOSLAV). Izbira je sicer poljubna. Programerji bodo morda izbrali USASCII nabor, ker pri tem naboru dobijo preglednejši izpis (listing).

Ta nastavitev ima vpliv le na ekran, ne pa tudi na izpis znakov na tiskalniku.

10.3.3 Razporeditev tipk na tipkovnici (kbd type)

Uporabimo lahko tipkovnici s QWERTZ ali QWERTY razporeditvijo. (Razporeditev tipk označujemo z tipkami, ki se nahajajo desno od tipke <TAB>). Pravilnost nastavitve ugotovimo, če se tipka <Z> na ekranu odziva z "Z"; če se odziva z "Y" je nastavev napačna.

10.3.4 Ponavljanje znaka (auto repeat)

Funkcija AUTO REPEAT omogoča, da tipka pošilja več istih znakov zaporedoma, če na tipko pritiskamo dlje, kot 0.5 sec. Znakov ne ponavljajo tipke <SET UP>, <RETURN> in <ESC>. Nastavev te funkcije je poljubna.

10.3.5 Zvočni odziv tipkovnice (keyclick)

Če je ta nastavev vključena, se tipkovnica zvočno odziva ob pritisku vsake tipke. Izbira je poljubna.

10.4 NASTAVITEV EKRANA (screen)

Funkcije v tem meniju omogočajo nastavev širine ekrana, ozadja ekrana in način izpisa daljših vrst, kot je širina ekrana. Priporočljiva nastavev je takšna, kot jo prikazuje slika 10.2.

```

-----
                          S e t   u p
1234567890123456789012345678901234567890123456789012345
Auto new line   No                Auto wrap_around   No
Screen width    80                 Srceen background   Normal
Tabulators      Term.& Kbd.         Screen                Save & Reset
-----
  
```

Slika 10.2. SET UP MENU (screen)

10.4.1 Širina ekrana (screen length)

Izbiramo med širinama 80 ali 132 znakov v vrstici. Navadno izberemo 80 znakov v vrstici, ker aplikacije ne podpirajo širine 132.

10.4.2 Ozadje ekrana (screen background)

Izbiramo med svetlimi znaki na temni podlagi in temnimi znaki na svetli podlagi. Izbira je poljubna, vendar je prva možnost prijetnejša za oči.

10.4.3 Prehod v novo vrstico (new line)

Če je ta funkcija vključena tipki <RETURN> in <LF> pošiljata dva znaka: znaka za prehod na začetek vrstice (carriage return) in znak za prehod v naslednjo vrstico (line feed). Ob izključeni pošiljata le svoj znak.

Nastavitev je napačna, če se na ekranu izpisuje le vsaka druga vrstica, ali če se izpisuje vrstica preko vrstice.

10.4.4 Lomljenje vrstic (Auto wrap_around)

Pri izpisovanju vrstic daljših kot 80 oziroma 132 znakov, imamo možnost, da se takšna vrstica izpiše v več vrstic zaslonu. Pri izključeni funkciji, se bodo odvečni znaki izpisovali na zadnjem mestu v liniji.

10.5 SHRANJEVANJE IN PONOVEN KLIC NASTAVITEV (Save & Reset)

Nastavitev, ki jo definiramo lahko shranimo v pomnilnik in s tem dosežemo, da bomo imeli izbrano nastavitev ob vključitvi računalnika. Če nastavitve ne shranimo velja le do izključitve ali resetiranja računalnika.

Vpis v pomnilnik, branje pomnilnika ali reset terminala dobimo s pritiskom tipke <PRESLEDEK> ali <GOR>.

10.5.1 Vpis v pomnilnik (Save parameters)

Ta funkcija omogoča, da nastavitev zapišemo v pomnilnik, ki je med izključitvijo računalnika napajana z baterijo. To omogoča, da ob vključitvi računalnika dobimo vedno željeno nastavitev. Če nastavitve ne shranimo v pomnilnik velja le do izključitve računalnika.

10.5.2 Branje pomnilnika (Recall parameters)

Omogoča branje nastavitvev, ki so zapisane v pomnilniku, in ustrezno nastavi terminal.

10.5.3 Začetna nastavitev terminala (Terminal reset)

Ta funkcija postavi partner terminal v začetno stanje, oziroma v stanje ki je definirano v pomnilniku, zbrise ekran in izstopi iz SET UP programa.

11. KOMUNIKACIJE (opcije)

Operacijski sistem CP/M 3 Plus lahko komunicira s perifernimi enotami preko petih kanalov, ki jih imenujemo logični kanali. Ti kanali so:

- CONIN: zaslon
- CONOUT: tipkovnica
- AUXIN: izhodni kanal za komunikacijo
- AUXOUT: vhodni kanal za komunikacijo
- LST: tiskalnik

Logične kanale lahko povezujemo s tako imenovanimi fizičnimi kanali. Osnovna konfiguracija PARTNERJA ima tri fizične kanale, ki se imenujejo:

- GDP zaslon
- CRT tipkovnica
- LPT tiskalnik

S tako imenovanimi opcijami dodajamo dodatne fizične kanale, ki jih uporabimo za komuniciranje z drugimi računalniki in perifernimi enotami.

Katerikoli fizični kanal lahko priredimo kateremukoli logičnemu kanalu z ukazom DEVICE. Z istim ukazom definiramo prenosno hitrost in protokol komuniciranja.

11.1. RS-232-C KANAL V.24

Vse verzije PARTNERJEV vključujejo en serijski kanal. Ta vmesnik služi za serijski asinhroni prenos, katerega hitrost lahko spreminjamo z ukazom DEVICE od 2400 na 4800 oz. 9600 boudov. Fizično se kanal imenuje LPT.

Preko 25 polne priključnice DB 25, ki se nahaja na zadnji strani osrednje enote in je označena z J7, ponavadi

Partner WFG, 2FG, 1FG - PRIROČNIK

priključimo matrični ali merjetični tiskalnik.

Ce PARTNERJA uporabljamo kot inteligentni sistem, lahko isti kanal uporabimo za povezovanje z gostiteljskim računalnikom.

Možna sta dva protokola:

- a) "XON-XOFF" protokol: Operacijski sistem podpira XON-XOFF protokol, ki razpozna znaka XON (11H) in XOFF (13H). Ko mikroročunalnik PARTNER sprejme od periferne enote (računalnika, tiskalnika), s katerim komunicira po podatkovni liniji, znak XOFF, preneha z oddajanjem podatkov toliko časa, dokler ne sprejme znaka XON.
- b) "PRIPRAVLJEN-ZASEDEN" protokol: Povezani enoti komunicirata na način sprejemanja oz. oddajanja kontrolnih signalov, ki serijskima vmesnikoma tako na eni kot na drugi strani povezave dovoli prenos podatkov ali pa ne.

Ne glede na izbrani protokol so prenosne hitrosti 2400, 4800 oz. 9600 baudov.

slika 11.1 Ožičenje priključnice J7, J8 in J9, ki so oblikovane za vmesnik V.24 (RS-232-C)

11.2 OPCIJE

OPCIJA 1 : vključuje dvojni serijski vmesnik, ki vsebuje dva dodatna asinhrona serijska kanala. Služi nam predvsem za povezovanje mikroracionalniškega sistema PARTNER z drugimi računalniškimi sistemi in vhodnimi enotami (miška, digitalna tablica).

Kanala sta enaka kot kanal LPT. Kanal označen z J8 se fizično imenuje VAX. Možne hitrosti komunikacije so 2400, 4800 in 9600. Kanal označen z J9 se imenuje MOD. preko tega kanala lahko komuniciramo s hitrostmi 2400, 4800 in 9600 baudov.

OPCIJA 2 : vključuje vmesnik z dvema osem bitnim kanaloma, oziroma z enim šestnajst bitnim kanalom. Fizično se kanal imenuje CEN. Ta kanal se uporablja za krmiljenje digitalnih vhodov in izhodov.

slika 11.2: Ožičenje na 40-polni priključnici J6
(Centronics)
Nožice z neparnimi številkami so za ozemljitev signalov (SIGNAL GROUND)

OPCIJA 3 : vključuje vmesnik z enim osembitnim Centronics kanalom. Ta kanal se fizično imenuje CEN. Uporablja se za periferijo, ki potrebuje tak protokol, predvsem so to tiskalniki.

LSYN 0002: sinhroni kanal. Ta kanal omogoča komunikacijo z računalniki, ki uporabljajo sinhroni protokol (IBM). Potrebno je vključiti tudi ustrezen programski paket.

Partner WFG, 2FG, 1FG - PRIROČNIK

LAN: (Local Area Network) je kanal za lokalno mrežo med Partnerji. Mreža povezuje do 64 (priporočljivo do 8) Partnerjev na serijsko linijo s hitrim prenosom podatkov. Partnerji si med seboj delijo diske in disketne enote. Možno je tudi pošiljanje sporočil med sistemi. Realizirana je tudi zaščita datotek in diskov.

TV OPCIJA : TV izhod za prikaz zaslonske slike na televizijskem ekranu. Uporaben je predvsem pri vzgojnoizobraževalnem procesu.

11.3 UKAZ DEVICE

NAMEN:

Omogoča definiranje povezav med logičnimi in fizičnimi kanali. Isti ukaz uporabimo tudi pri definiranju protokolov in hitrosti komunikacij.

PREGLED PARAMETROV FIZIČNIH KANALOV

ZGRADBA:

DEVICE šfizični kanal d logični kanalš

Ukaz brez parametrov nam vrne celotno informacijo o vhodno/izhodnih kanalih. Če navedemo za ukazom še ime določenega kanala, dobimo informacijo le o tem kanalu.

PRIMER 1: Primer izpisa, brez vgrajenih dodatnih opcij

A>DEVICE

Physical devices:

I=Input, O=Output, S=Serial, X=Xon-Xoff

CRT	9600	IOS	LPT	1200	IOSX	9600
	300		NONE		GDP	NONE 0

Current assignments:

CONIN: = CRT
 CONOUT: = GDP
 AUXIN: =
 AUXOUT: =
 LST: = LPT

Enter new assignments or hit RETURN

Če ne želimo spreminjati nastavitvev, pritisnemo tipko <RET>.

Iz slike je razvidno, da je logičnemu kanalu LST prirejen fizični kanal LPT (tiskalnik), logična kanala AUXIN in AUXOUT pa nimata prirejenega fizičnega kanala. Iz slike tudi vidimo, da preko fizičnega kanala komuniciramo s prenosno hitrostjo 1200 in protokolom XON-XOFF (LPT 1200 IOSX).

Partner WFG, 2FG, 1FG - PRIROČNIK

PRIMER 2: Primer izpisa, če imamo vgrajeno opcijo 1, to pomeni dodatna fizična kanala VAX in MOD.

A>DEVICE

Physical devices:

I=Input, O=Output, S=Serial, X=Xon-Xoff

CRT	9600	IOS	LPT	1200	IOS	VAX	9600	IOSX
MOD	300	IOSX		NON		GDP	NON	0

Current assignments:

CONIN: = CRT
 CONOUT: = GDP
 AUXIN: = VAX
 AUXOUT: = VAX
 LST: = LPT

Enter new assignments or hit RETURN

V tem primeru je logičnemu kanalu AUX prirejen fizični kanal VAX, medtem ko fizični kanal MOD nima definiranega logičnega kanala.

PRIMER 3: Če nas zanima samo en fizični kanal (npr.:LPT), dobimo informacijo o tem kanalu z ukazom:

A>DEVICE LPT

Physical Device: LPT
 Baud rate: 1200
 Characteristics: INPUT
 OUTPUT
 SOFT-BAUD
 SERIAL

DEFINIRANJE POVEZAVE MED LOGIČNIM IN FIZIČNIM KANALOM

NAMEN:

Če želimo standardne vhode in izhode spreminjati, uporabimo to obliko ukaza DEVICE. S tem dosežemo, da določen logični kanal pošilja podatke na nek drug fizični kanal (namesto na ekran na tiskalnik, preko logičnega izhoda LPT komuniciramo z drugimi sistemi).

ZGRADBA:

DEVICE logični-kanal:=fizični-kanal\$stikala

STIKALA:

x	- definiramo protokol XON-XOFF
nox	- definiramo protokol PRIPRAVLJEN-ZASEDEN
hitrost	- CP/M dovoljuje hitrosti komuniciranja
	50 75 110 134

Partner WFG, 2FG, 1FG - PRIROČNIK

150	300	600	1200
1800	2400	3600	4800
7200	9600	19200	

Fizičnim kanalom lahko priredimo le vnaprej definirane hitrosti, ki smo jih navedli pri opisu posameznih kanalov.

PRIMER 1: Preko fizičnega izhoda LST (standardni izhod za tiskalnik) želimo komunicirati z gostiteljskim računalnikom s hitrostjo 4800 in protokolom XON-XOFF.

A>DEVICE AUX:=LST\$X,4800

Physical devices:

I=Input,O=Output,S=Serial,X=Xon-Xoff

CRT	9600	IOS	LPT	4800	IOSX	9600	
	300			NONE		GDP	NONE 0

Current assigments:

CONIN: = CRT

CONOUT: = GDP

AUXIN: = LPT

AUXOUT: = LPT

LST: = LPT

Sedaj lahko komuniciramo preko logične enote AUX in fizične enote LPT z gostiteljskim računalnikom. Takšno nastavitvev uporabimo takrat, kadar imamo programe, ki komunicirajo preko logične enote AUX, nimamo pa vgrajene opcije 1 ali LSYN 002. Ta nastavitvev je posebej primerna, če PARTNERJA uporabljamo le kot inteligentni terminal in za to uporabljamo program RMT20.

Podobne nastavitvev uporabljamo tudi takrat, kadar na logični izhod LPT priključujemo digitalno tablico ali miško.

Ponavadi ukaze DEVICE vpišemo v ukazno datoteko PROFILE.SUB, tako da so vhodno/izhodni kanali že ob zagonu sistema pravilno nastavljeni.

11.4. IZBIRA IN PRIKLJUČITEV TISKALNIKA

Tiskalnik mora imeti vmesnik po standardu RS-232-C ali CENTRONICS.

Pri izbiri tiskalnika s serijskim vmesnikom moramo upoštevati:

1. Tiskalnik mora imeti serijski vmesnik, ki ustreza standardu V.24 (RS-232-C)
2. Način delovanja mora biti PRIPRAVLJEN-ZASEDEN ali XON-XOFF
3. Hitrost prenosa mora biti 2400 baudov ali več

Partner WFG, 2FG, 1FG - PRIROČNIK

4. Sprejemati mora znake v ASCII kodi
5. Imeti mora priključni kabel z vtičem, ki ustreza priključnici J7 (konektor DB 25)

Večina tiskalnikov ima za prireditev vmesnika mikrostickala, s katerimi določimo način komunikacije med tiskalnikom in računalnikom. Možnosti nastavitve stikal najdete v priložnem priložniku za tiskalnik. Pri nastavitvi teh stikal upoštevajte naslednje podatke:

- dolžina besede je 8 bitov
- kontrole paritete ni
- število stop bitov je 1.5
- prenosna hitrost je lahko 2400, 4800 ali 9600 boudov

S strani PARTNERJA prilagodite komunikacijo z ukazom DEVICE. Z njim nastavite protokol in hitrost prenosa.

PRIMER: Izberemo hitrost 4800 in protokol XON-XOFF

A>DEVICE LPT5X,4800

Tiskalnik s CENTRONICS vmesnikom priključimo na priključnico J6. CENTRONICS kanal se fizično imenuje CEN, njemu priredimo logični kanal LST z ukazom:

A>DEVICE LST:=CEN

Kabel, ki ga uporabljate za priključitev tiskalnikov, naj bo dolg največ 5m. Najboljši so parnosukalni kabli, ki povečujejo varnost pred motnjami.

12. NAPAKE IN NJIH ODPRAVLJANJE

Če vaš sistem ne deluje ali če je njegovo delovanje moteno, poskusite najprej sami poiskati napako in jo po možnosti odpraviti. Šele, če vam to ne uspe, pokličite servisno službo. Upoštevajte pa, da je vsak poseg v notranjost računalnika nestrokovnjaka nevaren in zato prepovedan.

Do napak prihaja zaradi nepravilnega delovanja ali uporabe programov ali zaradi nepravilnega delovanja strojne opreme. Predvsem pri prvih lahko sami odpravite vzrok.

V tem poglavju so opisane napake, ki se najpogosteje pojavljajo in kako moramo ukrepati.

V primeru, da napake ne morete odpraviti sami, vam bo pomoč nudila naša vzdrževalna služba. Obrnite se na naslov in telefonsko številko, ki sta navedena na ustreznem garancijskem listu.

12.1. NAPAKE OB VKLJUČITVI SISTEMA WFG

Neposredno po vključitvi sistema lahko pride do napak, ki jih lahko sami odpravite. Vzrok tem napakam so manjkajoči programi operacijskega sistema na disku.

Takšne napake odpravljamo s pomočjo systemske diskete s katero vzpostavimo sistem. V pomoč nam je monitorski program, ki se javi z znakom *, to dosežemo tako, da takoj po vključitvi in izpisu TESTING MEMORY pritisnemo tipko CTL-C. Monitorski program pozna ukaza "A" in "F". Ukaz A uporabimo, če želimo sistem nalagati iz diska, B pa v primeru, ko želimo sistem nalagati iz diskete.

Operacijski sistem s systemske diskete je namenjen le za odpravljanje napak in ga v druge namene ne uporabljamo !

1. NAPAKA : Ob vključitvi se sistem ne javi (ni slike)

- Vzrok : Ni napetosti
 Odpravljanje : - Preverite, če je sistem vključen v omrežno napetost 220V.
 Vzrok : Nepravilna nastavitve svetilnosti ekrana
 Odpravljanje : - Nastavite svetilnost ekrana, kot je opisano v poglavju 10.

2. Napaka : Približno 20 sec po vključitvi se na ekranu izpiše text:

**LOADING ERROR FROM HARD DISK TRY TO LOAD SYSTEM FROM FLOPY
 NO SYSTEM ON DISK**

- Vzrok : Na disku manjka datoteka CPM3.SYS
 Odpravljanje : - 1. vstavite sistemsko disketo
 2. vtipkajte znak F in počakajte, da se sistem javi z znakom pripravljenosti B>
 3. vtipkajte ukaz:

PIP A:CPM3.SYS=B:CPM3.GDP\$VR

4. resetirajte sistem

3. Napaka : Takoj po vključitvi se na ekranu izpiše tekst:

no CCP.com, retry?

- Vzrok : Ni konzolnega krmilnega programa (CCP.COM)
 Odpravljanje : - pritisnite katerokoli tipko. Če se sistem javi z znakom pripravljenosti, normalno nadaljujte z delom
 - 1. vstavite sistemsko disketo in pritisnite tipko reset
 2. ko se pojavi izpis TESTING MEMORY, pritisnite tipko CTL-C in počakajte, da se sistem javi z znakom *
 3. pritisnite tipko F
 4. prekopirajte konzolni krmilni program z diskete:

PIP A:=B:CCP.COM\$RV

5. resetirajte sistem

4. Napaka : Takoj po vključitvi se na ekranu pojavi tekst:

NO SYSTEM ON DISK

- Vzrok : Na disku ni začetnega nalagalnika, zapisanega na začetnih sledih diska
 Odpravljanje : - 1. vstavite sistemsko disketo

Partner WFG, 2FG, 1FG - PRIROČNIK

2. pritisnite tipko F in počakajte na znak pripravljenosti B>
3. z diskete na disk prekopirajte začetni nalagalnik:

PIP A:=B:WLDR.COMSVR

4. zapišite začetni nalagalnik na začetne sledi diska:

PUTWSYS

5. resetirajte sistem

5. Napaka : Sistem se javi s sporočilom BDOS Perm.

Vzrok : Prenizka temperatura sistema

Odpravljanje : - ne izključujte sistema in počakajte nekaj minut, da sistem dobi delovno temperaturo. Nato ponovno pritisnite tipko reset. Če s tem ne uspete, pokličite servisno službo.

6. Napaka : Sistem se javi z izpisom:

HARD DISK MALFUNCTION >>> RETRY WITH COMMAND A

Vzrok : Z diska se ne da brati

Odpravljanje : Isto kot pri napaki št. 5

12.2 NAPAKE PRI VKLJUCITVI SISTEMA 1FG

1. NAPAKA : Ob vključitvi se sistem ne javi (ni slike)

Vzrok : Ni napetosti

Odpravljanje : - Preverite, če je sistem vključen v omrežno napetost 220V.

Vzrok : Nepravilna nastavitve svetilnosti ekrana

Odpravljanje : - Nastavite svetilnost ekrana, kot je opisano v poglavju 10.

2. Napaka : Približno 20 sec po vključitvi se na ekranu izpiše text:

LOADING ERROR FROM HARD DISK TRY TO LOAD SYSTEM FROM FLOPY NO SYSTEM ON DISK

Vzrok : Na disketi manjka datoteka CPM3.SYS

Odpravljanje : - 1. vstavite originalno sistemsko disketo
2. resetirajte sistem
3. prepisite iz originalne systemske diskete na kopijo datoteko CPM3.SYS z ukazom

KOPI CPM3.SYS

Partner WFG, 2FG, 1FG - PRIROČNIK

- in postopajte po navodilih ukaza KOPI
 4. vstavite kopijo systemske diskete in
 resetirajte sistem

3. Napaka : Takoj po vključitvi se na ekranu izpiše
 tekst:

no CCP.com, retry?

- Vzrok : Ni konzolnega krmilnega programa (CCP.COM)
 Odpravljanje : - pritisnite katerokoli tipko. Če se sistem
 javi z znakom pripravljenosti, normalno
 nadaljujte z delom
 - 1. vstavite originalno systemsko disketo
 2. resetirajte sistem
 3. prepisite iz originalne systemske
 diskete na kopijo datoteko CCP.COM z
 ukazom

KOPI CCP.COM

- in postopajte po navodilih ukaza KOPI
 4. vstavite kopijo systemske diskete in
 resetirajte sistem

4. Napaka : Takoj po vključitvi se na ekranu pojavi
 tekst:

NO SYSTEM ON DISK

- Vzrok : Na disku ni začetnega nalagalnika,
 zapisanega na začetnih sledeh diska
 Odpravljanje : - 1. vstavite originalno systemsko disketo
 2. resetirajte sistem
 3. prepisite iz originalne systemske
 diskete na kopijo začetni nalagalnik z
 ukazom

GPUTFSYS

- in postopajte po navodilih na ekranu
 4. vstavite kopijo systemske diskete in
 resetirajte sistem

12.3 NAPAKE PRI VKLJUCITVI SISTEMA 2FG

1. NAPAKA : Ob vključitvi se sistem ne javi (ni slike)

- Vzrok : Ni napetosti
 Odpravljanje : - Preverite, če je sistem vključen v
 omrežno napetost 220V.

Vzrok : Nepravilna nastavitvev svetilnosti ekrana

Partner WFG, 2FG, 1FG - PRIROČNIK

Odpravljanje : - Nastavite svetilnost ekrana, kot je opisano v poglavju 10.

2. Napaka : Približno 20 sec po vključitvi se na ekranu izpiše text:

**LOADING ERROR FROM HARD DISK TRY TO LOAD SYSTEM FROM FLOPY
NO SYSTEM ON DISK**

Vzrok : Na disketi manjka datoteka CPM3.SYS

Odpravljanje : - 1. v pogon A vstavite originalno sistemsko disketo
2. resetirajte sistem
3. v pogon B vstavite kopijo systemske diskete, ki je povzročila napako
4. prepisite datoteko CPM3.SYS:

PIP B:CPM3.SYS=A:CPM3.2FG5VR

5. vstavite disketo iz pogona B v pogon A in resetirajte sistem

3.Napaka : Takoj po vključitvi se na ekranu izpiše tekst:

no CCP.com, retry?

Vzrok : Ni konzolnega krmilnega programa (CCP.COM)

Odpravljanje : - pritisnite katerokoli tipko. Če se sistem javi z znakom pripravljenosti, normalno nadaljujte z delom

- 1. ukrepajte isto kot pri napaki št. 2 do točke št. 4

2. prekopirajte konzolni krmilni program z diskete:

PIP B:=A:CCP.COMSRV

3. vstavite kopijo systemske diskete iz pogona B v pogon A in resetirajte sistem

4. Napaka : Takoj po vključitvi se na ekranu pojavi tekst:

NO SYSTEM ON DISK

Vzrok : Na disketi ni začetnega nalagalnika, zapisanega na začetnih sledih diska

Odpravljanje : - 1. v pogon A vstavite originalno sistemsko disketo

2. resetirajte sistem

3. vstavite kopijo systemske diskete, ki je povzročila napako v pogon B

4. na začetne sledi napišite začetni nalagalnik (program: GFLDR.COM) z

ukazom:

GPUTFSYS

5. vstavite kopijo systemske diskete iz pogona B v pogon A in resetirajte sistem

12.4. NEPRAVILNO DELOVANJE TIPKOVNICE

Do napak pri delovanju tipkovnice prihaja večinoma zaradi nepravilne nastavitve terminala. V pomoč pri nastavitvi je program MSETUP.

1. Napaka : Sistem ne reagira na znake s tipkovnice.

- Vzrok : Tipkovnica ni (pravilno) priključena.
 Odpravljanje : Kabel, ki vodi od tipkovnice, mora biti vključen v okroglo priključnico označeno s TAST, ki se nahaja na zadnji strani ohišja.
 Vzrok : Nevede pritisnjena tipka SCR (no scroll).
 Odpravljanje : Ponovno pritisnemo tipko SCR.
 Vzrok : Pritisnjena tipka CTL-S
 Odpravljanje : Pritisnite tipko CTL-Q

2. Napaka : Numerična tipkovnica na desni nepravilno deluje

- Vzrok : Numerična tipkovnica je v funkcijskem načinu delovanja
 Odpravljanje : Resetirajte terminal, kot je napisano v poglavju 10.

3. Napaka : Tipkovnica pošilja nepravilne znake

- Vzrok : Nepravilno izbran tip tipkovnice
 Odpravljanje : S SET UP-om izberite pravi tip, kot je opisano v poglavju 10.
 Vzrok : Nepravilno izbran nabor znakov
 Odpravljanje : S SET U-Pom izberite pravi nabor znakov, kot je opisano v poglavju 10.

4. Napaka : Nekontrirano vključevanje zvončka tastature in prižiganje kontrolnih lučk

- Vzrok : Uporaba programa, ki je nekompatibilen z grafičnim PARTNERJEM
 Odpravljanje : Resetirajte sistem in v bodoče ne uporabljajte tega programa

12.5. NEPRAVILNO DELOVANJE EKRANA

1. Napaka : Na ekranu se ne izpišejo vsi znaki

- Vzrok : Preveč zatemnjen ekran

Partner WFG, 2FG, 1FG - PRIROČNIK

Odpravljanje : Nastavite svetilnost ekrana, kot je opisano v poglavju 10.

2. Napaka : Na ekranu se izpisujejo nepravilni znaki

Vzrok : Nepravilno izbran nabor znakov

Odpravljanje : S SET UP-om izberite pravilen nabor znakov, kot je opisano v poglavju 10.

3. Napaka : Nenormalna (nepopolna) slika na ekranu

Vzrok : Napačno izbran tip terminala

Odpravljanje : Izberite pravilen tip terminala, kot je opisano v poglavju 10.

4. Napaka : Izpis na ekranu je v vsaki drugi vrstici

Vzrok : Napačno nastavljen terminal

Odpravljanje : Pravilno nastavite terminal, kot je opisano v poglavju 10.

5. Napaka : Na ekranu ostane grafična slika

Vzrok : Nepravilen izhod iz programa

Odpravljanje : - uporabite program CLG
- resetirajte terminal kot je opisano v poglavju 10.

12.6. NEPRAVILNO DELOVANJE DISKETNIH POGONOV

Ta vrsta napak je najnevarnejša, saj lahko pride do delne ali celotne izgube podatkov na disketi. Izognili se jih boste, če boste upoštevali vsa navodila o delu z disketami.

1. Napaka : Prižge se kontrolna lučka disketnega pogona - sistem ne deluje več

Vzrok : Diskete ni v pogonu

Odpravljanje : Vstavite disketo, kot je opisano v poglavju 5.1.5.

Vzrok : Varovalna ročica ni zaprta

Odpravljanje : Zaprite varovalno ročico

Vzrok : Nepravilno vstavljena disketa

Odpravljanje : Pravilno vstavite disketo

2. Napaka : Na ekranu se pojavi sporočilo:

**Error on B: T-00002, S-00001 Read, Missing address mark
Retry (Y/N) ?**

Vzrok : Neformatirana disketa

Odpravljanje : Formatirajte disketo

Vzrok : Pokvarjena disketa

Odpravljanje : - Vtipkajte znak Y
- Prepišite vsebino neokvarjenih datotek in

formatirajte disketo

12.7. NEPRAVILNO DELOVANJE TISKALNIKA**1. Napaka : Tiskalnik ne deluje**

- Vzrok : Tiskalnik ni vključen.
 Odpravljanje : Vključite tiskalnik.
 Vzrok : Tiskalnik ni priključen na priključnico J7 na zadnji strani ohišja sistema.
 Odpravljanje : Priključite tiskalnik na ustrezno mesto.
 Vzrok : Ni papirja.
 Odpravljanje : Vložite nov papir.
 Vzrok : Tiskalnik je v off-line načinu delovanja
 Odpravljanje : Postavite tiskalnik v on-line način delovanja.

2. Napaka : Tiskalnik tiska nečitljive znake.

- Vzrok : Vezni kabel je razrahljan
 Odpravljanje : Preglejte vezni kabel in priključnici
 Vzrok : Nepravilna priključnica (ni J7)
 Odpravljanje : Tiskalnik pravilno priključite
 Vzrok : Nepravilna prireditev protokola in hitrosti prenosa
 Odpravljanje : Preverite pravilnost nastavitve protokola in hitrosti prenosa

12.8. SISTEMSKA SPOROČILA

Sporočila, ki jih javi operacijski sistem v spodnjih vrsticah, imenujemo sistemska sporočila. Obveščajo nas o napakah, ki so nastale na strojni opremi, ponavadi na diskovnih in disketnih pogonih. Seznam teh sporočil je v knjigi Operating Systems User's Guide, dodatek A.

Sistemska sporočila poskusite odpraviti s pritiskanjem na tipko Y. To ponovite nekajkrat. Če večkratni poskusi ne uspejo, sistemska sporočila zberišemo s pritiskom na tipko N. Pri tem smo nasilno prekinili izvajanje programa, kar je lahko povzročilo izgubo podatkov.

Če se sistemska sporočila pogosto javljajo, poskušajte najprej ugotoviti vzrok, šele nato kličite servisno službo. Vzrok ni nujno okvara na računalniku ali disku/disketi, lahko je razlog takšnim sporočilom zapolnjen disk/disketa ali nepravilno zaščitena datoteka. Pred klicom servisne službe natančno zapišite sistemska sporočila, program ki je povzročil to napako in akcijo, ki ste jo zadnje izvedli. S temi podatki bo serviser lažje in hitreje odkril ter odpravil napako.

DODATEK A

TEHNIČNI PODATKI

Osrednja enota	

Glavni procesor (CPE)	Z80A
Dolžina besede	8 bitov
Sistemski takt	4 MHz
Dinamični pomnilnik	2 x 64 KB RAM
Nezbrisljivi pomnilnik	4 KB EPROM
Krmiljenje DMA	
2 x časomer	
Paralelni vmesnik	
Serijski vmesnik	
Vmesnik za diskovno enoto	
Vmesnik za disketno enoto	
Ura realnega časa s pomožnim baterijskim napajanjem	

Operacijski sistem	CP/M Plus (Verzija 3.0)

Video krmilna enota	

Procesor	Thompson 9367 Signetics 2674
Dinamični RAM	128K
Statični RAM	4K
Serijski kanal za tipkovnico	asinhroni, 300 Baudov
Priključitev na vodilo Partner	

Zaslon	

Diagonala zaslona	31 cm (12")
Prevleka	zeleni fosfor P31
Format zaslona	24 vrstic x 80 znakov (132)
Dodatne tri vrstice za sistemska sporočila	
Točkovna matrika	8 x 11 točk
Direktno naslavljanje kazalca	
Urejevalniške funkcije	
Inverzna slika	
Frekvenčni pas cevi	25 MHz
Frekvenca ponavljanja slike	50 Hz
Grafični prikaz - dve sliki	1024x512 (256) točk

Zunanji pomnilniki

- Diskovna enota (5 1/4)

Kapaciteta:	
Neformatirana	12,76 MB
Formatirana	10 MB

Hitrost prenosa podatkov	5 Mb/s
Poprečni čas pristopa	85 ms
Gostota zapisa	9074 BPI
Poprečni čas med izpadi	11000 ur

- Disketna enota (5 1/4)

Kapaciteta:	
Neformatirana	1 MB
Formatirana	0,66 MB

Hitrost prenosa podatkov	250 Kb/s
Poprečni čas pristopa	150 ms
Gostota zapisa	2938 BPI
Poprečni čas med izpadi	9200 ur

Tipkovnica

Nizki profil, premakljiva
 2 ločeni skupini tipk
 Alfanumerični ASCII Standard
 Ureditev tipk jugoslovsanska QWERTZ
 4 funkcijske tipke
 Možnost predelave na US ASCII

Vmesnik za tiskalnik

Standardni vmesnik V.24 (RS-232-C)	
Hitrost prenosa	2400, 4800 ali 9600 baudov

Paralelni vmesnik Centronics	opcija 3
------------------------------	----------

Komunikacije

Dvoje kanalov V.24 (RS-232-C)	opcija 1
Serijski asinhroni protokol	
Hitrost prenosa	300 - 9600 bau- dov (odvisno od kanala)

Partner WFG, 2FG, 1FG - PRIROČNIK

Napajanje -----	
220 V / 50 Hz, 100 W 2 omrežni varovalki	5 x 25 mm, T 6,3A
Fizične lastnosti -----	
Širina	522 mm
Globina (s tipkovnico)	655 mm
Višina	344 mm
Teža	22 kg
Pogoji okolja -----	
Delovna temperatura	10 - 32 st. C
Relativna vlažnost	20 - 80 %

DODATEK B

SEZNAM KOD ALFANUMERICNIH IN POSEBNIH TIPK
TIPKOVNICE, CE JE NASLOVLJEN TERMINAL TIP
PARTNER

KODIRANJE (HEX)

Oznaka tipke	normal	shift	control	komentar
1	CAN	13	13	ctl X (CP/M)
2	I	0B	0B	kazalec navzgor
3	I	0A	0A	" navzdol
4	-	0B	0B	" levo
5	-	0C	0C	" desno
6	PF1	04	04	ctl D
7	PF2	05	05	ctl E
8	PF3	06	06	ctl F
9	PF4	07	07	ctl G
10	ESC	1B	1B	
11	1 !	31	21	31
12	2 "	32	22	32
13	3 #	33	23	33
14	4 \$	34	24	34
15	5 %	35	25	35
16	6 ^	36	27	36
17	7 &	37	26	37
18	8 *	38	2A	38
19	9 (39	2B	39
20	0)	30	29	30
21	-	2D	5F	2D
22	=+	3D	2B	3D
23	/ ?	2F	3F	2F
24	back sp.	0B	0B	0B kot kazalec levo
25	NUL	00	00	00 ničta koda(ctl a)
26	?	37	37	37
27	B	38	38	38
28	9	39	39	39
29	-	2D	2D	2D
30	TAB	09	09	09 tabulator
31	0	71	51	11
32	W	77	57	17
33	E	65	45	05
34	R	72	52	12
35	T	74	54	14
36	Z	7A	5A	1A
37	U	75	55	15
38	I	69	49	- 09 - kot tabulator
39	D	6F	4F	1F
40	P	70	50	10
41	S	7B	5B	1B
42	D	7C	5C	1C

Partner WFG, 2FG, 1FG - PRIROCNIK

43	DELETE	7F	7F	7F	bríše znak
44	4	34	34	34	
45	5	35	35	35	
46	6	36	36	36	
47	,	2C	2C	2C	
48	CTL				kontrolna tipka
49	CAPS LOCK				velike črke
50	A	61	41	01	
51	S	73	53	13	
52	D	54	44	- 04	-kot PF1
53	F	66	46	- 06	- " PF3
54	G	67	47	- 07	- " PF4
55	H	68	48	- 08	- " kazalec levo
56	J	6A	4A	- 0A	- " LINE FEED
57	K	6B	4B	- 0B	- " kazalec navzg.
58	L	6C	4C	- 0C	- " kazalec desno
59	C	7E	5E	1E	
60	c	7D	5D	1D	
61	RETURN	0D	0D	0D	carriage return
62	< >	3C	3E	3C	
63	1	31	31	31	
64	2	32	32	32	
65	3	33	33	33	
66	ENTER	0D	0D	0D	kot RETURN
67	DC3	03	03	03	ctl C (CP/M)
68	SHIFT				pomik glave
69	Y	79	59	19	
70	X	78	58	18	
71	C	63	43	03	
72	V	76	56	16	
73	B	62	42	02	
74	N	6E	4E	0E	
75	M	6D	4D	- 0D	- kot RETURN
76	, ;	2C	3B	2C	
77	. :	2E	3A	2E	
78	2	60	40	- 00	- kot NUL
79	SHIFT				pomik glave
80	LINE FEED	0A	0A	0A	
81	0	30	30	30	
82	.	2E	2E	2E	
83	space	20	20	20	tipka za presledek

Tipkovnica PARTNER ne vsebuje programsko nastavljenih (koda) tipk. Vse funkcijske tipke (kot so PF1, PF2, itn.) imajo funkcijo, ki jo določa uporabniška in sistemska programska oprema. Namen omenjenih tipk je enostavnejša uporaba (npr., da ni treba vedno uporabljati tipke CTL).

Partner WFG, 2FG, 1FG - PRIROČNIK

DODATEK C

NASLOVI PERIFERNIH ENOT SISTEMA PARTNER

NASLOVI (heksadecimalno) !		FUNKCIJE
80	87	! izklop pomnilnika EPROM
88	8F	! vklop pomnilnika-BANK 1 RAM
90	97	! vklop pomnilnika-BANK 2 RAM
98	9F	! vklop motorjev diskovnih pogonov
A0	A7	!
A8	AF	!
B0	B7	! ura realnega časa
B8	BF	!
C0	C7	! izbira krmilnika DMA
CB	CF	! izbira vezja CTC
D0	D7	! izbira vezja PIO
DB	DF	! izbira vezja SIO1
E0	E7	! izbira vezja SIO2
E8	EF	! izbira prekinit. vektorja za FDC
F0	F7	! izbira krmilnika FDC
F8	FF	!

Naslovi vrat SIO:

kanal	podatek	status	
CRT	0d8h	0d9h	
LPT	0dah	0dbh	
VAX	0e0h	0e1h	opcija 1
MOD	0e2h	0e3h	opcija 1

Naslovi vrat PIO:

kanal	podatek	status	
A	0d0h	0d1h	opcija 2 ali 3
B	0d2h	0d3h	opcija 2 ali 3

Naslovi VIDEO modula so v poglavju DODATEK D.

DODATEK D

NASLOVI GRAFIČNE VIDEO PLOŠČE

Komentar je v angleščini zaradi strokovnosti izrazov.

1. Thomson GDP ports

Hex. naslov

```

20  read : status                               write : command
-----
7 6 5 4 3 2 1 0                                glej tabelo !
d d d d d d d +- light pen end
d d d d d d d +--- vertical blanking
d d d d d d +----- ready
d d d d +----- pen out of display
d d d +----- ligt pen irq
d d +----- vb irq
d +----- ready irq
+----- IRQ

```

21 r/w control register 1

```

-----
7 6 5 4 3 2 1 0
d d d d d d d +--- HIGH = pen down ; LOW = pen up
d d d d d d d +----- HIGH = pen ; LOW = eraser
d d d d d d +----- HIGH = write only
d d d d d +----- HIGH = cyclic screen ( write if pen out )
d d d +----- enable lpen irq
d d +----- enable vb irq
d +----- enable ready irq
+----- not used, read as 0

```

22 r/w control register 2

```

-----
7 6 5 4 3 2 1 0
d d d d d d d d +--- vector type
d d d d d d d +----- vector type
d d d d d +----- HIGH = tilted characters
d d d d +----- HIGH = characters on vert. axis
d d d +----- X
d d +----- X
d +----- X
+----- X

```

23 r/w character size register

```

-----
7 6 5 4 3 2 1 0
d d d d d d d +--- P0
d d d d d d +----- P1
d d d d d +----- P2
d d d d +----- P3 ( scaling on x axis )
d d d +----- Q0
d d +----- Q1
d +----- Q2
+----- Q3 ( scaling on y axis )

```

24 reserved

25 r/w deltax

26 reserved

27 r/w deltay

28 r/w xpos MSBs

29 r/w xpos LSBs

2A r/w ypos MSBs

2B r/w ypos LSBs

2C read XLP write reserved

2D read YLP write reserved

2E reserved

2F read status (no interrupt reset) write reserved

2. Z80 pio ports

30 r/w graphic common control

```

-----
7 6 5 4 3 2 1 0
d d d d d d d +--- display page of memory
d d d d d d +----- write page of mememory
d d d d d +----- write mode ( xor / nor )
d d d d +----- format 0
d d d +----- format 1
d d +----- gdpint ( read only )
d +----- avdint ( read only )
+----- scroll mode

```

31 write : pio port a control register

32 r/w common text attributes

```

-----
7 6 5 4 3 2 1 0
d d d d d d d +---- dot stretch
d d d d d d d +----- cursor mode
d d d d d d +----- monochrome / colour
d d d d +----- force background / blue foreground
d d d +----- reverse screen / green foreground
d d +----- dots / char 0
d +----- dots / char 1
+----- 24 / 16 Mhz text video clock

```

33 write : pio port b control register

3. miscellaneous text registers

34 r/w : character register

35 r/w : attribute register

```

-----
7 6 5 4 3 2 1 0
d d d d d d d +---- blink
d d d d d d d +----- underline
d d d d d d +----- special character
d d d d +----- protect
d d d +----- highlight / red foreground
d d +----- reverse video / green background
d +----- general purpose 2 / blue background
+----- general purpose 1 / red background

```

36 write : graphic scroll read : common input

graphic processor scan lines are offset by the value of this 8 bit register

```

-----
7 6 5 4 3 2 1 0
d d d d d d d +---- undefined
d d d d d d d +----- undefined
d d d d d d +----- undefined
d d d d +----- undefined
d d d +----- avdc access flag
d d +----- undefined
d +----- undefined
+----- graph. pix input

```

4. text video controler (AVDC) ports

```

38      write : init registers          read : interrupt register
-----
all 15  initialization                7 6 5 4 3 2 1 0
registers are sequentially
accessed via a single
address                                d d d d d d d +---- split screen 2
                                         d d d d d d d +---- ready
                                         d d d d d d d +---- split screen 1
                                         d d d d d d d +---- line zero
                                         d d d +----- vertical blank
                                         d d +----- 0
                                         d +----- 0
                                         +----- 0
    
```

```

39      write : command register       read : status register
-----
refer to AVDC manual                7 6 5 4 3 2 1 0
                                         d d d d d d d +---- split scr 2 int
                                         d d d d d d d +---- ready int.
                                         d d d d d d d +---- split scr 1 int
                                         d d d d d d d +---- line zero int
                                         d d d +----- vb int.
                                         d d +----- ready flag
                                         d +----- 0
                                         +----- 0
    
```

- 3A r/w screen start 1 lower register
- 3B r/w screen start 1 upper register
- 3C r/w cursor address lower register
- 3D r/w cursor address upper register
- 3E r/w screen start 2 lower register
- 3F r/w screen start 2 upper register

DODATEK E

RELATIVNI NASLOVI PREKINITVENIH VEKTORJEV V BIOS-U

Naslovi so relativni glede na začetek BIOSa.

```

;; interrupt vector table ;;;;;;;;;;

007E'          DS 6
0084'          INTVEC::
0084'  00B4'   pioA:: dw dумыint
0086'  00B4'   pioB:: dw dумыint

008B'  0000*   fdc::  dw flpint  ; FDC interrupt vector

008A'  0000*   ctc1:: dw motint ; CTC interrupt vector
008C'  00B4'   ctc2:: DW dумыint ; timer
008E'  0000*   ctc3:: dw avdint ; VB

0090'  00B4'  00B4'  sic1:: dw dумыint,dумыint,dумыint,dумыint
0094'  00B4'  00B4'
009B'  00B4'  00B4'   dw dумыint,dумыint,kbdint, kbderr
009C'  0000*  0000*

00A0'  00B4'  00B4'  sic2:: dw dумыint,dумыint,dумыint,dумыint
00A4'  00B4'  00B4'
00A8'  00B4'  00B4'   dw dумыint,dумыint,dумыint,dумыint
00AC'  00B4'  00B4'

00B0'  00B4'   cpioA:: dw      dумыint
00B2'  00B4'   cpioB:: dw      dумыint

00B4'          dумыint:
00B4'  FB      ei
00B5'  ED 4D   reti          ; just return

```

Pri napisu dумыint lahko uporabnik napiše naslov svojega upravljalca prekinitve.

DODATEK F

NACIN UPORABE GRAFICNIH RUTIN

F.1 NAVODILO ZA RISANJE NA GRAFICNEM PARTNERJU
 =====

Uporabnik ima možnost risanja grafičnih elementov iz aplikativnega programa. Pri tem mora aplikativni program klicati BIOS (del operacijskega sistema za povezavo s periferijo).

Na sledečih straneh je splošno pojasnilo. Za razumevanje je potrebno osnovno znanje o CP/M Plus operacijskem sistemu.

Za ukaze grafičnemu procesorju je v grafičnem Partnerju predvidena bios funkcija 37. Podatki se prenašajo preko vektorja v katerem so: število podatkov, koda grafičnega ukaza in podatki.

Način klicanja grafičnih ukazov:

1. Napolnimo vektor z ustreznimi podatki (glej razlago ukazov),
2. Vstavimo adresu vektorja v register BC,
3. Pokličemo bios funkcijo 37.

Vektor ima naslednjo obliko:

```
vektor$1C ... spodnji byte števila podatkov
vektor$2C ... zgornji byte števila podatkov
vektor$3C ... koda grafičnega ukaza
vektor$4...C podatki
```

Število podatkov je največji indeks zmanjšan za 2 (med podatke štejejo tudi kodo grafičnega ukaza, ne pa samega števila podatkov).

Partner WFG, 2FG, 1FG - PRIROČNIK

PRIMER:

Inicializiraj grafični procesor na 256 pikic po y osi v assemblerju, fortranu in pascalu!

```

a. assembler: LD A,0           ;FMAT := 0;
                  LD (FMAT),A
                  LD HL,VECTOR    ;VECTORS1C := 2
                  LD A,2
                  LD (HL),A
                  INC HL          ;VECTORS2C := 0;
                  LD A,0
                  LD (HL),A
                  INC HL          ;VECTORS3C := 0;
                  LD A,0
                  LD (HL),A
                  INC HL          ;VECTORS4C := FMAT
                  LD A,(FMAT)
                  LD (HL),A
                  LD HL,VECTOR    ;REG.HL = ADDR(VECTOR)
                  CALL BIOS37     ;KLICI FUNKCIJO BIOS 37.
                  .....
    
```

```

;Klic bios rutine 37. (grafični ukazi)
BIOS37:: LD B,H           ;BC=HL
          LD C,L           ;
          PUSH AF          ;spravi reg.A in zastavice
          PUSH DE          ;spravi reg.DE
          LD A,(1)         ;čitaj bios spodnji byte addr
          LD L,A           ;spravi v reg.L
          LD A,(2)         ;čitaj bios zgornji byte addr
          LD H,A           ;spravi v reg.H
          LD DE,37*3       ;postavi v reg.DE odmik do
                          ; graf. funkcije
          ADD HL,DE        ;prištej odmik
          POP DE           ;popravi reg.DE
          POP AF           ;popravi reg.A in zastavice
          JP (HL)         ;skoči na bios funkcijo
    
```

opomba: proceduro BIOS37 lahko prevedemo posebej in linkamo z glavnim programom.

Partner WFG, 2FG, 1FG - PRIROČNIK

```

b. fortran:      BYTE VECOTR,FMAT
                   DIMENSION VECTOR(120)
                   FMAT = 0
                   VECTOR(1) = 2
                   VECTOR(2) = 0
                   VECTOR(3) = 0
                   VECTOR(4) = FMAT
                   CALL BIOS37( VECTOR )
                   END

;Klic bios rutine 37. (grafični ukazi)
BIOS37:: LD B,H          ;BC=HL
         LD C,L          ;
         PUSH AF         ;spravi reg.A in zastavice
         PUSH DE         ;spravi reg.DE
         LD A,(1)        ;čitaj bios spodnji byte addr
         LD L,A          ;spravi v reg.L
         LD A,(2)        ;čitaj bios zgornji byte addr
         LD H,A          ;spravi v reg.H
         LD DE,37*3      ;postavi v reg.DE odklik do
                   ; graf. funkcije
         ADD HL,DE       ;pristej odklik
         POP DE          ;popravi reg.DE
         POP AF          ;popravi reg.A in flage
         JP (HL)         ;skoči na bios funkcijo

```

opomba: procedura BIOS37 lahko prevedemo posebej in linkamo z glavnim programom.

c. pascal:

```

program test;
var fmat : byte;
    vector : array51..127C of byte;
begin
  write('vpiši fmat (0...256, 1...512): ');
  read(FMAT);
  VECTORS1C := 2;
  VECTORS2C := 0;
  VECTORS3C := 0;
  VECTORS4C := FMAT;
  BIOS( 37, ADDR( VECTOR ) );
end.

```

KONEC PRIMERA.

F.2 PREDLOG SEZNAMA GRAFIČNIH UKAZOV ZA GRAFIČNI PARTNER:

Tu je predlog grafičnih ukazov. Uporabnik si lahko sam doda kompleksnejše ukaze. Tu je le zbirka ukazov, ki so že realizirani v Fortranski in Pascalski knjižnici (ti prejme uporabnik na disku - GPFORLIB.REL in GPPASLIB.REL).

F.2.1 KNJIŽNICA GRAFIČNIH UKAZOV ZA PASCAL

procedure ginit (fmat : byte);

inicializira grafični kontroler in nastavi resolucijo:

```
fmat = 1 ---> resolucija je 1024 X 512
fmat = 0 ---> resolucija je 1024 X 256
```

```
vector$1C := 2;
vector$2C := 0;
vector$3C := 0;
vector$4C := fmat;
```

procedure gexit;

podprogram izključi grafični način

```
vector$1C := 1;
vector$2C := 0;
vector$3C := 1;
vector$4C := 0;
```

procedure gclr;

podprogram zbríše grafično sliko

```
vector$1C := 1;
vector$2C := 0;
vector$3C := 3;
vector$4C := 0;
```

procedure gxy(x,y:integer);

premik na pozicijo (x,y) pri čemer je (0,0) levo spodaj

```
vector$1C := 5;
vector$2C := 0;
vector$3C := 4;
vector$4C := low(x);
vector$5C := high(x);
```

```
vector$6C := low(y);
vector$7C := high(y);
```

```
procedure gtext( str :std; s, t :byte);
```

```
  pisanje grafičnega teksta:
```

```
  str vsebina teksta
```

```
  s   velikost črk teksta
```

```
      zgornji del byta - širina znaka
```

```
      št.pikic
```

```
      1 ... 5
```

```
      2 ... 10
```

```
      3 ... 15
```

```
      .....
```

```
      15 ... 75
```

```
      0 ... 80
```

```
      spodnji del byta _ višina znaka
```

```
      1 ... 8
```

```
      2 ... 16
```

```
      3 ... 24
```

```
      .....
```

```
      15 ...
```

```
      0 ... 128
```

```
  t   način pisave - bit 5 in 4: 00 write
```

```
      01 xor
```

```
      10 clear
```

```
      11 clear
```

```
      - bit 1 in 0: 00 normalne črke
```

```
      01 italics črke
```

```
      10 za 90 stopinj obrnjene
```

```
      normalne črke
```

```
      11 za 90 stopinj obrnjene
```

```
      italics črke
```

```
vector$1C := dolžina teksta + 4;
```

```
vector$2C := 0;
```

```
vector$3C := 5;
```

```
vector$4C := s;
```

```
vector$5C := t;
```

```
vector$6C := dolžina teksta (max 120 znakov);
```

```
vector$7..C:= tekst;
```

procedure vbar(l,w,attribute);

nariše blok:

```

l    višina bloka
w    širina bloka
atr  tip polnjenja bloka
      bit 1 in 0: 00 polna črta
                  01 črtkana
                  10 pika
                  11 črta,pika,...
      bit 5 in 4: 00 write
                  01 xor
                  10 clear
                  11 clear

vector$1C := 4;
vector$2C := 0;
vector$3C := 6;
vector$4C := 1;
vector$5C := w;
vector$6C := atr;

```

procedure circ(x,y,r:integer);

nariše krog:

```

x,y  koordinati središča kroga
r    polmer kroga (max 255)

vector$1C := 6;
vector$2C := 0;
vector$3C := 7;
vector$4C := low(x);
vector$5C := high(x);
vector$6C := low(y);
vector$7C := high(y);
vector$8C := r;

```

```
procedure ring(x,y,r,tt:integer);
```

nariše zapolnjen kolobar:

x,y koordinati središča
r polmer zunanlega kroga (max 255)
tt debelina kolobarja (max 255)

```
vector$1C := 7;  
vector$2C := 0;  
vector$3C := 8;  
vector$4C := low(x);  
vector$5C := high(x);  
vector$6C := low(y);  
vector$7C := high(y);  
vector$8C := r;  
vector$9C := tt;
```

```
procedure disc(x,y,r:integer);
```

nariše zapolnjen krog:

x,y koordinati središča kroga
r polmer kroga (max 255)

```
vector$1C := 6;  
vector$2C := 0;  
vector$3C := 9;  
vector$4C := low(x);  
vector$5C := high(x);  
vector$6C := low(y);  
vector$7C := high(y);  
vector$8C := r;
```


procedure draw(x,y: integer; att: byte);

nariše linijo:

```
x,y koordinati končne točke
atr bit 1 in 0: 00 polna črta
                01 črtkana
                10 pike
                11 črta,pika,...
bit 5 in 4: 00 write
            01 xor
            10 clear
            11 clear

vector$1C := 6;
vector$2C := 0;
vector$3C := 10;
vector$4C := atr;
vector$5C := low(x);
vector$6C := high(x);
vector$7C := low(y);
vector$8C := high(y);
```

procedure vect(dx,dy: integer; attr: byte);

nariše linijo :

```
dx,dy relativna razdalja od pozicije,
na kateri je grafični kursor
atr bit 1 in 0: 00 polna črta
                01 črtkana
                10 pike
                11 črta,pika,...
bit 5 in 4: 00 write
            01 xor
            10 clear
            11 clear

vector$1C := 6;
vector$2C := 0;
vector$3C := 11;
vector$4C := atr;
vector$5C := low(dx);
vector$6C := high(dx);
vector$7C := low(dy);
vector$8C := high(dy);
```

procedure scroll(lines: integer; dir: byte);

```

premik slike po y osi

lines  število linij (max 255)
dir    smer: 0 ... dol
        1 ... gor

        vector$1C := 3;
        vector$2C := 0;
        vector$3C := 12;
        vector$4C := low(lines);
        vector$6C := dir;
    
```

function getpix: byte;

```

pove ali je točka, kjer se nahaja kurzor prižgana ali ne
1 ... točka je prižgana
0 ... točka je ugasnjena
rezultat vrne v registru A.

        vector$1C := 1;
        vector$2C := 0;
        vector$3C := 13;
    
```

function cursor(rf: boolean; xc,yc,step: integer): integer;

```

nariše grafičen kurzor, premik kurzorja s puščicami na
tipkovnici in vrne kazalec na novo pozicijo

rf    upoštevanje nove pozicije    1 ... da
        0 ... ne

xc    nova pozicija kurzorja
yc    nova pozicija kurzorja
step  korak pri premiku kurzorja (max 255)

        vector$1C := 7;
        vector$2C := 0;
        vector$3C := 14;
        vector$4C := f1;
        vector$5C := low(xc);
        vector$6C := high(xc);
        vector$7C := low(yc);
        vector$8C := high(yc);
        vector$9C := step;
    
```

procedure blkvect(xp, yp : integer; lt, nm : byte);

risanje verige vektorjev

xp, yp absolutna začetna točka
 lt tip linije
 nm število vektorjev (max 30)

```

vector$1C := 7 + 4 * nm;
vector$2C := 0;
vector$3C := 15;
vector$4C := low(xp);
vector$5C := high(xp);
vector$6C := low(yp);
vector$7C := high(yp);
vector$8C := lt;
vector$9C := nm;
vector$10C := low(x1);
vektor$11C := high(x1);
vektor$12C := low(y1);
vektor$13C := high(y1);
vektor$14C := low(x2);
.....
    
```

procedure qfill(xpos,ypos: integer; xmask,ymask:byte; mode : boolean);

polnjenje omejenih področij v vodoravni ali navpični smeri

xpos pozicija točke
 ypos pozicija točke
 xmask y maska za polnjenje področja
 ymask x maska za polnjenje področja
 mode način polnjenja
 bit 0: 0 vodoravno
 1 navpično

```

vector$1C := 8;
vector$2C := 0;
vector$3C := 16;
vector$4C := low(xpos);
vector$5C := high(xpos);
vector$6C := low(ypos);
vector$7C := high(ypos);
vector$8C := xmask;
vector$9C := ymask;
vector$10C := mode;
    
```

```
procedure pixmem( i : byte );
```

```
  preklop slike (risanje in prikaz)
```

```
  i bit 0: 0 ... prikaz slike 0
           1 ... prikaz slike 1
  bit 1: 0 ... riši na sliko 0
           1 ... riši na sliko 1
```

```
  vector$1C := 2;
  vector$2C := 0;
  vector$3C := 17;
  vector$4C := i;
```

```
procedure getblk( x, y : integer; var bpnt: blkpnt);
```

Čitanje bloka pikic (matrika 128 x 8). Vsak byte vsebuje pokončni stolpec pikic. Bit 0 vsebuje zgornjo pikico, bit 7 pa spodnjo. Pikica podana s koordinatama x in y se nahaja v prvem bytu v bitu 7. Naslednji byte vsebuje pikice desno od predhodnjih.

```
x, y      koordinati začetne točke matrike
           (levi spodnji kot matrike)
```

```
bpnt      kazalec na vektor 128 bytov
```

```
vector$1C := 5;
vector$2C := 0;
vector$3C := 18;
vector$4C := lo(x);
vector$5C := hi(x);
vector$6C := lo(y);
vector$7C := hi(y);
```

F.2.2. KNJIŽNICA GRAFIČNIH UKAZOV V FORTRANU

```

C =====
C Fortranska knjižnica grafičnih ukazov za grafični Partner
C -----
C
C call ginit (fmat)
C
C inicializira grafični kontroler in nastavi resolucijo:
C
C fmat = 1 ---> resolucija je 1024 X 512
C fmat = 0 ---> resolucija je 1024 X 256
C
C      SUBROUTINE GINIT(FMAT)
C      BYTE VECTOR, FMAT
C      DIMENSION VECTOR(4)
C      VECTOR(1) = 2
C      VECTOR(2) = 0
C      VECTOR(3) = 0
C      VECTOR(4) = FMAT
C      CALL BIOS37( VECTOR )
C      RETURN
C      END
C -----
C
C call gexit
C
C podprogram izključi grafični način
C
C      subroutine gexit
C      byte vector
C      dimension vector(4)
C      vector(1) = 1
C      vector(2) = 0
C      vector(3) = 1
C      vector(4) = 0
C      call bios37( vector )
C      return
C      end
C -----
C
C call gclr
C
C podprogram zbriše grafično sliko
C
C      subroutine gclr
C      byte vector
C      dimension vector(4)
C      vector(1) = 1
C      vector(2) = 0
C      vector(3) = 3
C      vector(4) = 0
C      call bios37( vector )

```

```

return
end

```

```

C-----
C
C call gxy(ix,iy)
C
C

```

premik na pozicijo (ix,iy) pri čemer je (0,0) levo spodaj

```

C
C      subroutine gxy(ix,iy)
C      byte vector,ix,iy
C      dimension ix(2),iy(2),vector(7)
C      vector(1) = 5
C      vector(2) = 0
C      vector(3) = 4
C      vector(4) = ix(1)
C      vector(5) = ix(2)
C      vector(6) = iy(1)
C      vector(7) = iy(2)
C      call bios37( vector )
C      return
C      end

```

```

C-----
C
C call gtext( str, s, t )
C
C

```

pisanje grafičnega teksta:

str vsebina teksta

s velikost črk teksta

zgornji del byta - širina znaka

št.pikic

1 (16) ... 5

2 (32) ... 10

3 (48) ... 15

.....

15 (240) ... 75

0 (0) ... 80

spodnji del byta _ visina znaka

1 ... 8

2 ... 16

3 ... 24

.....

15 ...

0 ... 128

t način pisave - bit 5 in 4: 00 write

01 xor

10 clear

11 clear

- bit 1 in 0: 00 normalne črke

01 italics črke

10 za 90 stopinj obrnjene

normalne črke

11 za 90 stopinj obrnjene

italics črke

```

C
C      subroutine gtext( str, s, t )
C

```

```

byte byte, str, s, t, vector
dimension vector(128), str(128)
i=1
10  if (str(i) .eq. 0) goto 20
    i = i + 1
    if (i .le. 128) goto 10
20  continue
    i = i - 1
    vector(1) = i + 4
    vector(2) = 0
    vector(3) = 5
    vector(4) = s
    vector(5) = t
    vector(6) = i
    do 30 j = 1, i
        k = 6 + j
30  vector(k) = str(j)
    call bios37( vector )
    return
end

```

C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C

```
call vbar(1,w,attr)
```

nariše blok:

l višina bloka

w širina bloka

atr tip polnjenja bloka

bit 1 in 0: 00 polna črta

01 črtkana

10 pike

11 črta,pika,...

bit 5 in 4: 00 write

01 xor

10 clear

11 clear

```
subroutine vbar( 1, w, attr )
```

```
byte 1, w, attr, vector
```

```
dimension vector(6)
```

```
vector(1) = 4
```

```
vector(2) = 0
```

```
vector(3) = 6
```

```
vector(4) = 1
```

```
vector(5) = w
```

```
vector(6) = attr
```

```
call bios37( vector )
```

```
return
```

```
end
```

C
C
C
C
C
C
C
C
C
C
C

call circ(x,y,r)

nariše krog:

x,y koordinati središča kroga
r polmer kroga (max 255)

```
subroutine circ(x, y, r )
byte x, y, r, vector
dimension x(2), y(2), vector(8)
vector(1) = 6
vector(2) = 0
vector(3) = 7
vector(4) = x(1)
vector(5) = x(2)
vector(6) = y(1)
vector(7) = y(2)
vector(8) = r
call bios37( vector )
return
end
```

C
C
C
C
C
C
C
C
C
C

call ring(x,y,r,tt)

nariše zapolnjen kolobar:

x,y koordinati središča
r polmer zunanjega kroga (max 255)
tt debelina kolobarja (max 255)

```
subroutine ring(x,y,r,tt)
byte x,y,r,tt,vector
dimension x(2),y(2),vector(9)
vector(1) = 7
vector(2) = 0
vector(3) = 8
vector(4) = x(1)
vector(5) = x(2)
vector(6) = y(1)
vector(7) = y(2)
vector(8) = r
vector(9) = tt
call bios37( vector )
return
end
```


C
C
C
C
C
C
C
C
C

call disc(x,y,r)

nariše zapolnjen krog:
x,y koordinati središča kroga
r polmer kroga (max 255)

```
subroutine disc(x,y,r)
byte x,y,r,vector
dimension x(2),y(2),vector(8)
vector(1) = 6
vector(2) = 0
vector(3) = 9
vector(4) = x(1)
vector(5) = x(2)
vector(6) = y(1)
vector(7) = y(2)
vector(8) = r
call bios37( vector )
return
end
```

C
C
C
C
C
C
C
C
C
C
C
C

call draw(x, y, att)

nariše linijo:
x,y koordinati končne točke
atr bit 1 in 0: 00 polna črta
 01 črtkana
 10 pike
 11 črta,pika,...
bit 5 in 4: 00 write
 01 xor
 10 clear
 11 clear

```
subroutine draw(x, y, att)
byte x,y,att,vector
dimension x(2),y(2),vector(8)
vector(1) = 6
vector(2) = 0
vector(3) = 10
vector(4) = att
vector(5) = x(1)
vector(6) = x(2)
vector(7) = y(1)
vector(8) = y(2)
call bios37( vector )
return
end
```

C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C

call vect(dx,dy,attr)

nariše linijo :

dx,dy relativna razdalja od pozicije,
na kateri je grafični kurzor

atr bit 1 in 0: 00 polna črta
 01 črtkana
 10 pike
 11 črta,pika,...
bit 5 in 4: 00 write
 01 xor
 10 clear
 11 clear

subroutine vect(dx,dy,attr)

byte dx,dy,attr,vector

dimension dx(2), dy(2), vector(8)

vector(1) = 6

vector(2) = 0

vector(3) = 11

vector(4) = attr

vector(5) = dx(1)

vector(6) = dx(2)

vector(7) = dy(1)

vector(8) = dy(2)

call bios37(vector)

return

end

C
C
C
C
C
C
C
C
C
C
C
C
C
C
C

call scroll(lines,dir);

premik slike po y osi

lines število linij (max 255)

dir smer: 0 ... dol

 1 ... gor

omejitev: pri načinu 512 točk po y osi mora biti

število linij deljivo s 4!

subroutine scroll(lines, dir)

byte lines,dir,vector

dimension lines(2), dir(2),vector(6)

vector(1) = 3

vector(2) = 0

vector(3) = 12

vector(4) = lines(1)

vector(5) = lines(2)

vector(6) = dir(1)

call bios37(vector)

return

end

C
C
C
C
C
C
C
C
C

function getpix

pove ali je točka, kjer se nahaja kurzor prižgana ali ne
 1 ... točka je prižgana
 0 ... točka je ugasnjena
 rezultat vrne preko parametra 1

```

subroutine getpix(j)
byte vector,j
dimension vector(3)
vector(1) = 1
vector(2) = 0
vector(3) = 13
call gbio37( vector )
j = vector(1)
return
end

```

C
C
C
C
C
C
C
C
C
C
C

subroutine cursor(rf,xc,yc,step)

nariše grafičen kurzor, premik kurzorja s puščicami na tipkovnici in vrne kazalec na novo pozicijo

```

rf      upoštevanje nove pozicije      1 ... da
                                              0 ... ne
xc      nova pozicija kurzorja
yc      nova pozicija kurzorja
step    korak pri premiku kurzorja (max 255)

```

```

subroutine cursor(rf,xc,yc,step)
byte rf,xc,yc,step,vector
dimension vector(9),xc(2),yc(2)
vector(1) = 7
vector(2) = 0
vector(3) = 14
vector(4) = rf
vector(5) = xc(1)
vector(6) = xc(2)
vector(7) = yc(1)
vector(8) = yc(2)
vector(9) = step
call bioh14( vector )
xc(1) = vector(1)
xc(2) = vector(2)
yc(1) = vector(3)
yc(2) = vector(4)
return
end

```

```

C
C call blkvect( xp, yp, lt, nm )
C
C
C   risanje verige vektorjev
C   xp, yp   absolutna začetna točka
C   lt      tip linije
C   nm      število vektorjev (max 29)
C
C   subroutine blkvect( xp, yp, lt, nm )
C   byte xp,yp,lt,nm,vector
C   dimension xp(58),yp(58),vector(128)
C   if (nm .gt. 29) then nm = 29
C   vector(1) = 7 + 4 * nm
C   vector(2) = 0
C   vector(3) = 15
C   vector(4) = xp(1)
C   vector(5) = xp(2)
C   vector(6) = yp(1)
C   vector(7) = yp(2)
C   vector(8) = lt
C   vector(9) = nm
C   write(3,100)lt,nm
100 format(i5,' ',i5)
C   do 20 i = 1, nm
C     j = i*4
C     k = i*2
C     vector(j+6) = xp(k+1)
C     vector(j+7) = xp(k+2)
C     vector(j+8) = yp(k+1)
C     vector(j+9) = yp(k+2)
20 continue
C   call bios37( vector )
C   return
C   end

C call qfill(xpos,ypos:integer xmask,ymask:byte mode:boolean)
C
C   polnenje omejenih področij v vodoravni ali navpični smeri
C   xpos   pozicija točke
C   ypos   pozicija točke
C   xmask  y maska za polnenje področja
C   ymask  x maska za polnenje področja
C   mode   način polnenja
C           bit 0: 0 vodoravno
C                1 navpično
C
C   subroutine qfill( xpos, ypos, xmask, ymask, mode )
C   byte xpos, ypos, xmask, ymask, mode, vector
C   dimension xpos(2),ypos(2),vector(10)
C   vector(1) = 8
C   vector(2) = 0
C   vector(3) = 16
C   vector(4) = xpos(1)
C   vector(5) = xpos(2)
C   vector(6) = ypos(1)

```

```

vector(7) = ypos(2)
vector(8) = xmask
vector(9) = ymask
vector(10) = mode
call bios37( vector )
return
end

```

C
C
C
C
C
C
C
C
C
C

call pixmem(i)

preklop slike (risanje in prikaz)

```

i bit 0: 0 ... prikaz slike 0
          1 ... prikaz slike 1
bit 1:  0 ... riši na sliko 0
          1 ... riši na sliko 1

```

```

subroutine pixmem( i )
byte i, vector
dimension vector(4), i(2)
vector(1) = 2
vector(2) = 0
vector(3) = 17
vector(4) = i(1)
call bios37( vector )
return
end

```

C
C
C
C
C
C
C
C
C
C
C
C

call getblk(x, y, bpnt)

Čitanje bloka pikc (matrika 128 x 8). Vsak byte vsebuje pokončni stolpec pikic. Bit 0 vsebuje zgornjo pikico, bit 7 pa spodnjo. Pikica podana s koordinatama x in y se nahaja v prvem bytu v bitu 7. Naslednji byte vsebuje pikice desno od predhodnjih.

x, y koordinati začetne točke matrike
(levi spodnji kot matrike)

bpnt kazalec na vektor 128 bytov

```

subroutine getblk( x, y, bpnt )
byte x, y, bpnt, vector
dimension x(2), y(2), bpnt(128), vector(128)
vector(1) = 5
vector(2) = 0
vector(3) = 18
vector(4) = x(1)
vector(5) = x(2)
vector(6) = y(1)
vector(7) = y(2)
call bio128( vector )
do 10 i=1,128
bpnt(i) = vector(i)

```

10

Partner WFG, 2FG, 1FG - PRIROČNIK

```
return  
end
```

Konstrukcija grafične knjižnice:

```
M80 BIOS37 ,= BIOS37  
F80 GPFORLIB,= GPFORLIB  
LIB80 GPFORLIB = GFORLIB, BIOS37/E
```

F.2.3. PRIMER PROGRAMA ZA GRAFIKO V FORTRANU

```

C =====
C glavni program - test grafične knjižnice
C
      byte i1,j1,buffer
      integer xc,yc,xp,yp
      dimension xp(30),yp(30),buffer(128)

C ... brisanje grafične slike
      call gclr

C ... brisanje grafične slike in
C nastavitev resolucije po y osi
      CALL GINIT(1)

C ... nastavitev risanja in prikaza slike
C risanje na sliko 1, prikaz slike 0
      call pixmem( 2 )

C ... premik na točko x=100,y=100
      call gxy( 100, 100 )

C ... nariši blok višine 100 in širine 20
      call vbar( 100,20,0 )

C ... nariši krog na poziciji x=500, y=200 polmer = 50
      call circ( 500,200,50 )

C ... nariši kolobar na poziciji x=600, y=300,
C zunanji polmer=80, notranji polmer=50
      call ring( 600,300,80,50 )

C ... nariši polni krog na poziciji x=700, y=400, polmer=100
      call disc( 700,400,100 )

C ... nastavitev risanja in prikaza slike
      call pixmem( 0 )

C ... premik na točko x=200, y=200
      call gxy( 200,200 )

C ... nariši daljico do točke x=500, y=500
      call draw( 500,500,0 )

C ... nariši vektor do dx=100, dy=-50
      call vect( 100,-50,0 )

C ... premik na točko x=100, y=100
      call gxy( 100, 100 )

C ... prečitaj vrednost točke (0..ugasnjena, 1..prižgana)
      call getpix(i1)

```

Partner WFG, 2FG, 1FG - PRIROČNIK

```

C ... premik na točko x=200, y=250
  call gxy( 200,250 )

C ... prečitaj vrednost točke (0..ugasnjena, 1..prižgana)
  call getpix(j1)

C ... izpisi vrednost točk na ekran v alfa načinu
  write(3,100)i1,j1
100  format( 'i5',' ',i5 )
      call gxy( 1, 1)

C ... čitanje grafičnega kurzorja
  xc = 200
  yc = 150
  call cursor( 1,xc,yc,3)

C ... izpis pozicije grafičnega kurzorja v alfa načinu
  write(3,100)xc,yc

C ... risanje trikotnika
  xp(1) = 100
  yp(1) = 250
  xp(2) = +100
  yp(2) = 0
  xp(3) = 0
  yp(3) = +100
  xp(4) = -100
  yp(4) = -100
  call blvect( xp, yp, 0, 3 )

C ... zapolnenje trikotnika
  call qfill( 150, 300, 255, 255, 1)

C ... nastavitev risanja in prikaza slike
  call pixmem( 0 )

C ... premik na točko x=200, y=200
  call gxy( 0, 107 )

C ... nariši daljico do točke x=500, y=500
  call draw( 227, 100, 0 )

C ... čitaj blok 128x8 pikic iz grafične slike
  call getblk( 100,204,buffer )

C ... izpis vsebine bloka na alfa zaslon
  write( 3,200 ) (buffer(i),i=1,128)
200  format( ' ',16i4 )

C ... izpis teksta 'TEXT' na pozicijo x=300, y=400
  buffer(1) = 'T'
  buffer(2) = 'E'
  buffer(3) = 'X'
  buffer(4) = 'T'
  buffer(5) = 0

```


Partner WFG, 2FG, 1FG - PRIROČNIK

```

call gxy( 300,400 )
call gtext( buffer, 51, 0 )

C ... grafični scroll za 8 točk navzgor
call scroll( 8, 0 )
call cursor( 1,xC,yC,3)

C ... grafični scroll za osem točk navzdol
call scroll( 8, 1 )
call cursor( 1,xC,yC,3)

C ... izstop iz grafičnega delovanja
call gexit
END

```

Prevajanje in linkanje programa:

```

F80 =test
LB0 test,GPFORLIB/S,FORLIB/S,test/N/E

```

Nato izvajamo program TEST. Izhod je tipka ENTER.

DODATEK G

MODUL ZA KLIC BIOS FUNKCIJE 37

To je vmesni modul med uporabniškim programom in BIOS operacijskega sistema CP/M Plus. Kličemo ga iz kateregakoli jezika.

```

                .Z80
;BIOS37.MAC
;Klic bios rutine 37. (grafični ukazi)
;VHOD: HL ... vektor, ki vsebuje vse podatke za izvršitev
;      grafičnega ukaza
bios:  LD B,H      ;BC=HL
        LD C,L      ;
        LD A,(1)    ;čitaj bios spodnji byte addr
        LD L,A      ;shrani v reg.L
        LD A,(2)    ;čitaj bios zgornji byte addr
        LD H,A      ;shrani v reg.H
        LD DE,37*3 ;37*3 ;postavi v reg.DE odmik do
                ; graf. funkcije
        ADD HL,DE   ;prištej odmik
        JP (HL)     ;skoči na bios funkcijo
; klic biosa brez vračanja rezultatov funkcije
BIOS37::push hl    ;spravi registre
        push de
        push bc
        call bios   ;pokliče bios 37
        pop bc      ;restavrira registre
        pop de
        pop hl
        ret         ;return
LOC1:  DW 0         ;adresa 1.parametra
; klic bios funkcij, ki vrnejo rezultate funkcije preko
; parametra 1!
GBIO37::LD (LOC1),HL ;spravi adresu parametra
        CALL BIOS37 ;kliči bios 37
        LD HL,(LOC1) ;rezultat prenesi v parameter
        LD (HL),A
        RET
biohl4::push de
        push bc
        ld (loc1),hl
        call bios
        ex de,hl
        ld hl,(loc1)
        ex de,hl
        ld bc,4
        ldir
        ld hl,(loc1)
        pop bc
        pop de
        ret

```

Partner WFG, 2FG, 1FG - PRIROČNIK

```
bio128::push de
        push bc
        ld (loc1),hl
        call bios
        ex de,hl
        ld hl,(loc1)
        ex de,hl
        ld bc,128
        ldir
        ld hl,(loc1)
        pop bc
        pop de
        ret
        END
```

DODATEK H

UPORABA PROGRAMA ZA IZDELAVO MENUJEV

Preden pristopite k izdelavi menujev priporočamo, da točno definirate izgled menuja in povezave med menuji. Ker lahko iz enega menuja kličemo drugega, menuje povezujemo med seboj. Običajno zgradimo drevesno strukturo menujev.

Izdelava novega menuja poteka v treh fazah:

1. Izdelava ekranske slike
2. Definiranje parametrov za klice programov
3. Vključitev novega menuja v strukturo menujev

Ko vpišemo ime menuja, zaključimo vnos z <RET> tipko. Če je menu nov, to potrdimo z odgovorom "d". Zatem preidemo v prvo fazo dela - izdelavo ekranske slike.

Izdelava ekranske slike

Za ekransko sliko imamo na voljo vse vrstice od 1. do 22. vrstice. Vrstici 23 in 24 sta namenjeni za sporočila. V sliko lahko vključimo vse znake z alfanumeričnega dela tipkovnice. Za pomikanje po ekranu uporabljamo smerne puščice.

Preko numeričnega dela tipkovnice vnašamo ukaze:

1. Splošni ukazi
2. Velikost znakov
3. Način izpisa znakov
4. Posebne funkcije
5. Risanje črt
6. Blok ukazi

SPLDŠNI UKAZI

- <PF1> - Izhod
 <PF2> - Brisanje ekrana. S to tipko zberemo celoten ekran, pred tem odgovorimo z "d".
 <PF3> - Pomoč. Dobimo kratko informacijo o pomenu tipk, na alfanumeričnem delu tipkovnice
 <PF4> - Osvežitev. Ekran se nespremenjen ponovno izpiše.

Partner WFG, 2FG, 1FG - PRIROČNIK

VELIKOST ZNAKOV

Vsi ukazi veljajo za celotno vrstico.

- <7> - Vklop dvojne širine znakov.
- <8> - Vklop zgornjega dela znakov dvojne širine in višine.
- <9> - Vklop spodnjega dela znakov dvojne širine in višine.
- <-> - Vklop standardne velikosti znakov. To je hkrati preklic zgornjih ukazov.

NACIN IZPISA ZNAKOV

Za vsak znak lahko določimo sledeče načine: dvojna svetilnost, inverzni izpis in utripanje. Dovoljena je katerakoli kombinacija načinov izpisa.

- <4> - Vklop/izklop inverzne pisave.
- <5> - Vklop/izklop dvojne svetilnosti.
- <6> - Vklop/izklop utripanja.
- <, > - Izklop vseh načinov.

POSEBNE FUNKCIJE

- <1> - Vklop/izklop sledi. S pomikanjem s smernimi puščicami prenašamo tudi način izpisa znakov. Uporabno za risanje.
- <2> - Vklop/izklop grafičnih simbolov. Namesto malih črk uporabljamo grafične znake.
- <Ø> - Izklop vseh funkcij - postavitev funkcij na začetno stanje.
- <. > - Vklop/izklop risanja črt.

RISANJE ČRT

V tem načinu dela imajo tipke na numeričnem delu tipkovnice drugačen pomen.

- <8> - črta gor
- <2> - črta dol
- <4> - črta levo
- <6> - črta desno
- <. > - izklop risanja črt

BLOK UKAZI

Neko besedilo (blok), katerega smo zapisali na ekran, lahko poljubno prenesemo na drugo pozicijo na ekranu, kopiramo, zapišemo v pomožno datoteko, beremo iz pomožne datoteke in izpišemo, brišemo cel blok.

CTL-B - označimo začetek bloka
 CTL-K - označimo konec bloka
 CTL-V - prenesemo blok na poljubno mesto na ekranu
 CTL-C - kopiramo blok na poljubno mesto
 CTL-Y - brišemo blok
 CTL-H - označbe se brišejo
 CTL-W - vpišemo blok v pomožno datoteko
 CTL-R - beremo blok iz pomožne datoteke in izpišemo

Definiranje parametrov

Z vsakega menija lahko kličemo do 20 programov, ki so označeni s številkami od 1 do 20. Vrstni red vpisa ni važen. Navadno je 20. mesto rezervirano za izstop iz menija.

Ko izberemo katerokoli številko, dobimo naslednjo sliko:

```

Program: _____
Parametri:
Terminal (VT1@0=@, VTS2=1, PARTNER=3):
Področje (@..15):
Klicni znak:
  
```

Program

V to rubriko vpišemo ime programa, katerega želimo klicati iz menija. Kličemo lahko izvršljive programe (.COM) in ukazne datoteke (.SUB). Navesti moramo celotno ime, vključno s podaljškom. (npr.: TP.COM, GRAF.SUB)

Parametri

Pod parametri razumemo vse tisto, kar bi navedli v običajni ukazni vrstici z imenom programa. Če želimo, da se izvaja ukaz DIR *.*SU=ALL, so v tem primeru parametri *.*SU=ALL. Tukaj moramo tudi poskrbeti za povratek v menu po izvršenem klicanem programu. To naredimo tako, da za običajnimi parametri navedemo še klic menija. Sintaksa je:

```
Parametri: ! MENU ime_menuja
```

Primer: Iz menija z imenom TEST želimo klicati ukaz:

```
DIR *.*SU=ALL
```

Po izvršenem programu se želimo vrniti v isti menu, torej v menu TEST.

Partner WFG, 2FG, 1FG - PRIROČNIK

Program: DIR.COM
 Parametri: *.*\$U=ALL ! MENU TEST

Ce pri parametrih ne navedemo ponovnega klica menuja, po končanem programu pademo v operacijski sistem.

Ce iz menuja kličemo drug menu, pri parametrih navedemo samo ime klicanega menuja.

Primer: Kličemo menu z imenom POIZKUS.

Program: MENU.COM
 Parametri: POIZKUS

Ce želimo, da iz menuja preidemo v menu POIZKUS in se nato s pritiskom na katerokoli tipko vrnemo v menu, pod parametre navedemo naslednje podatke:

Program: MENU.COM
 Parametri: POIZKUS \$h ! MENU IME

V primeru, da smo navedli napačne parametre ali ime programa, imajo navedene tipke naslednji pomen:

- CTL-A - pomik na začetek vrstice
- CTL-F - pomik na konec vrstice
- CTL-S - pomik za znak v levo
- CTL-D - pomik za znak v desno
- CTL-G - brisanje znaka pod kazalcem
- CTL-Y - brisanje vrstice od kazalca dalje
- DELETE - brisanje znaka v levo
- SPACE - brisanje znaka v desno

Terminal

Izberemo terminal, za katerega je bil pisan program.

Področje

Vsi programi, katere kličemo iz menujev, morajo biti na delovnem področju Ø, definirani kot sistemski. Na področje, katerega tukaj definiramo, se bodo zapisovale datoteke, ki jih bo general klicani program.

Opozorilo: Nekateri programi zahtevajo, da so podatki na istem področju kot program, zato v takem primeru za delovno področje izberemo Ø.

Partner WFG, 2FG, 1FG - PRIROČNIK

Klicni znak

Na koncu definiramo še klicni znak, s katerim bomo klicali program. Velike in male črke imajo isti pomen.

Primer:

Želimo klicati program TP s tipko <T>. Podatki naj se odlagajo na 3.delovno področje. Po končanem delu v Programu, se želimo vrniti v menu z imenom EDITOR.

```
Program: TP.COM
Parametri: ! MENU EDITOR
Terminal (VT100=#, VT52=1, PARTNER=3):#
Področje (0..15):3
Klicni znak: T
```

Ko definiramo vse klice programov, izstopimo s tipko <RET>. Če želite shraniti nov oziroma spremenjeni menu, na vprašanje:

Ali želiš shraniti menu:

odgovorite z D, sicer z N.

Naslednji korak je definiranje jezika, s katerim bo program Menu javljal napake.

Vključitev novega menuja v strukturo menujev

Po končanem zapisovanju novega menuja na disk, se vrnemo v osnovno sliko. Če smo zgradili nov menu, moramo sedaj ta menu vključiti v strukturo menujev. To storimo tako, da v parametre menuja, iz katerega želimo klicati naš novi menu, vpišemo nov klic menuja.



DODATEK I

UKAZNA ZAPOREDJA ESCAPE (ANSI NAČIN)

V primeru, da ni podan noben parameter, je na tabeli prikazana osnovna vrednost; pri tem "x" določa kateri od številnih priključkov je predviden.

UKAZI	DEFAULT	OPIS
ESC \$Pn A	1	Pomik kurzorja za n vrstic navzgor do zgornjega roba
ESC \$Pn B	1	Pomik kurzorja za n vrstic navzdol do spodnjega roba
ESC \$Pn C	1	Pomik kurzorja za n stolpcev v desno do roba ekrana
ESC \$Pn D	1	Pomik kurzorja za n stolpcev v levo do roba ekrana
ESC \$H	1.1	Pomik kurzorja na izhodiščni položaj (enako kot ESC \$I, 1H)
ESC \$P1;Pc H	1.1	Neposredno nastavljanje kurzorja, P1=vrstica, Pc=stolpec
ESC \$P1;Pc f	1.1	Neposredno nastavljanje kurzorja, P1=vrstica, Pc=stolpec
ESC D		Indeks (pomik gor), (enako kot line feed - to scroll), kurzor mora biti na vrhu ali na dnu območja premikanja
ESC M		Obratni indeks (pomik dol) (obratni line feed)
ESC E		Nova vrstica (return plus line feed with scroll)
ESC 7		Ohrani položaj kurzorja in atributov
ESC 8		Povrne položaj kurzorja in atributov
ESC # 3		Vrstica z dvojno višino (zgornja polovica), dvojno širino
ESC # 4		Vrstica z dvojno višino (spodnja polovica), dvojno širino
ESC # 5		Vrstica z enojno širino, enojno višino
ESC # 6		Vrstica z dvojno širino, enojno višina



Partner WFG, 2FG, 1FG - PRIROČNIK

UKAZI	DEFAULT	OPIS
ESC \$Ps;Ps m	∅	Znakovni atributi - Ps se nanaša na selektivne parametre; izbranih je lahko več parametrov, ki so ločeni s podpičjem (;). Parametri so zbrani v vrstnem redu, so kumulativni in imajo naslednji pomen: ∅ Vsi znakovni atributi so izključeni 1 Naslednji znaki v debelem tisku 4 Naslednji znaki podčrtani 5 Naslednji znaki utripajoči 7 Naslednji znaki v obratnem videu
ESC \$ K	x	Brisanje od kurzorja do konca vrstice, dvojni status je nespremenjen
ESC \$ 1 K		Brisanje od začetka vrstice do kurzorja, dvojni status je nespremenjen
ESC \$ 2 K		Brisanje cele vrstice v kateri je kurzor, dvojni status je nespremenjen
ESC \$ J		Brisanje od kurzorja do konca ekrana, za vse izbrisane vrstice nastavi stanje enojne širine.
ESC \$ 1 J		Brisanje od začetka ekrana do kurzorja, za vse izbrisane vrstice nastavi stanje enojne vrstice
ESC \$ 2 J		Brisanje celotnega ekrana, glede na stanje brisalnega načina; za vse izbrisane vrstice nastavi stanje enojne širine
ESC \$ Ps;Ps q	o	Programabilni LEDs - Ps se nanaša na selektivne parametre; izbranih je lahko več parametrov, ki so ločeni s podpičjem (;). Parametri so zbrani v vrstnem redu in imajo naslednji pomen: ∅ ugasne vse LEDs 1 prižge LED # 1 2 prižge LED # 2 3 prižge LED # 3 4 prižge LED # 4 5 prižge LED # 5
SI in SO		Iz enega od mnogih razpoložljivih naborov znakov izhajata tudi G0 in G1. Nabori znakov so priklicani s kodo SI in SO (shift in, shift out). G0 in G1 odgovarjata kodi SI oziroma SO. Nabora G0 in G1 sta programirana z naslednjim zaporedjem:



Partner WFG, ZFG, 1FG - PRIROČNIK

ESC (A	UKASCII	G0	(designator)
ESC) A	UKASCII	G1	"
ESC (B	UKASCII	G0	"
ESC) B	UKASCII	G1	"
*ESC (E	ŠPANSKI	G0	"
*ESC) E	ŠPANSKI	G1	"
*ESC (F	FRANCOSKI	G0	"
*ESC) F	FRANCOSKI	G1	"
*ESC (G	NEMSKI	G0	"
*ESC) G	NEMSKI	G1	"
*ESC (I	ITALJANSKI	G0	"
*ESC) I	ITALJANSKI	G1	"
*ESC (N	NORVESKI/DANSKI	G0	"
*ESC) N	NORVESKI/DANSKI	G1	"
*ESC (S	ŠVEDSKI/FINSKI	G0	"
*ESC) S	ŠVEDSKI/FINSKI	G1	"
*ESC (Y	JUGOSLOVANSKI	G0	"
*ESC) Y	JUGOSLOVANSKI	G1	"
ESC (Ø	LINIJSKA GRAFIKA	G0	"
ESC) Ø	LINIJSKA GRAFIKA	G1	"

UKAZI	DEFAULT	OPIS
ESC \$ Pt;Pb r	1,24	Nastavi področje delnega pomikanja Pt=zgoraj, Pb=spodaj
ESC \$ r		Povrne področje celotnega pomikanja
ESC H		Nastavi tabulator na trenutnem stolpcu
ESC \$ g	x	Odstrani tabulator na trenutnem stolpcu
ESC \$ 3 g		Odstrani vse tabulatorje
ESC \$ Ø h / Ø 1		Napaka (ignored)
ESC \$ 2Ø h		Nastavi novo vrstico (return povzroči novo vrstico + return)

UKAZI	DEFAULT	OPIS
ESC \$ 2Ø 1		Ponovno nastavi novo vrstico (return ne povzroči nove vrstice, ampak samo return)
ESC \$?1 h		Nastavi kurzorske ključe v aplikacijski način; tudi pomožna tastatura mora biti v aplikacijskem načinu (glej fig 4.3)
ESC \$?1 1		Povrne kurzorske ključe v kurzorski način
ESC \$?2 h		Partner način

Partner WFG, 2FG, 1FG - PRIROČNIK

ESC \$?2 1	Vstop v način VT 52
ESC \$?3 h	Nastavi 132 stolpcev na vrstico
ESC \$?3 l	Nastavi 80 stolpcev na vrstico
ESC \$?4 h	Pomikanje (gladko)
ESC \$?4 l	Pomikanje (skokovito)
ESC \$?5 h	Zaslon s svetlim ozadjem
ESC \$?5 l	Zaslon s temnim ozadjem
ESC \$?6 h	Nastavi relativno področje kurzorja na zgornji levi kot področja delnega premikanja. Relativno izhodišče.
ESC \$?6 l	Nastavi relativno področje kurzorja na zgornji levi kot zaslona. Absolutno izhodišče.
ESC \$?7 h	Preskok (vklop), na koncu vrstice samodejna vrnitev
ESC \$?7 l	Preskok (izklop)
ESC \$?8 h	Samodejno ponavljanje (vklop), ponovi pritisnjen ključ
ESC \$?8 l	Samodejno ponavljanje (izklop)
ESC =	Pomožna tastatura (aplikacijska), (glej fig.4.3)
ESC >	Pomožna tastatura (numerična), (tudi povrne način kurzorja)
ESC c	Postavitev v začetno stanje, izvrši se ponovna vključitev.
ESC # 8	Zaslon napolne z E-ji

* Označuje nove posebnosti dodane standardu VT 100 (tm).

UKAZI ZAPOREDJA ESCAPE (VT 52 NAČIN)

UKAZI	OPIS
*ESC 3	Nasproten video vklopljen
*ESC 4	Nasproten video izklopljen
ESC A	Kurzor gor, ustavi se na vrhu
ESC B	Kurzor dol, ustavi se na dnu
ESC C	Kurzor desno, ustavi se na robu
ESC D	Kurzor levo, ustavi se na robu
*ESC E	Kurzor na izhodišče, zaslon čist
ESC F	Vstop v grafični način (linijske graf.)
ESC G	Izstop iz grafičnega načina
ESC H	Premik kurzorja na izhodišče
ESC I	Obratni vrstični pomik
ESC J	Briše od kurzorja do konca zaslona
ESC K	Briše od kurzorja do konca vrstice
ESC Y	Premakne kurzor na določeno vrstico in stolpec. Številke vrstic in stolpcev so poslani kot ASCII kode, katerih vrednosti so številke in 1FH, npr. 20H velja za prvo vrstico ali stolpec, 28 velja za osmo vrstico ali stolpec etc.

Partner WFG, 2FG, 1FG - PRIROČNIK

ESC =	Vstop v alternativno pomožno tastaturo
ESC >	Izstop iz alternativne pomožne tastature
ESC <	Vstop ANSI načina delovanja

* Označuje nove posebnosti dodane VT100/VT52.



Partner WFG, 2FG, 1FG - PRIROČNIK

PRIPOMBE K PRIROČNIKU (vpiši naslov)

Prosim, da izpolnite in pošljete na naslov:

ISKRA DELTA COMPUTERS
Tržno komuniciranje
Parmova 41, 61000 Ljubljana

Če imate pripombo k priročniku, ali ste v njem odkrili kakršnekoli napake, vas naprošamo, da jih navedete na tem listu.

PRIPOMBE:

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Naslov uporabnika:

DD:

Ulica, kraj:

Ref. oseba:

Telefon:

VAŠE DELOVNO PODROČJE:

- Programer Organizator Vodilni delavec
- študent Drugo

Vaša obstoječa verzija priročnika:

Na osnovi poslanega vam bomo avtomatsko pošiljali vse spremembe v teh navodilih!

Hvala za sodelovanje!

8. Obr. 55-14

Partner WFG, 2FG, 1FG - PRIROČNIK

Mikroračunalniški sistem PARTNER WFG, 2FG, 1FG

PRIROČNIK ZA UPORABNIKE

Prva izdaja

Ident: XX XXX XXX

Izdajatelj:

ISKRA DELTA COMPUTERS, Tržno komuniciranje, Parmova 41, Ljubljana

PARTNER je zaščitni znak ISKRA DELTE COMPUTERS
CP/M 3+ je zaščitni znak podjetja Digital Research Corporation

LJUBLJANA
Marec 1987



01-10-84