

OPERACIJSKI SISTEM FDOS V3.1

ZA MIKRORAČUNALNIK ID 1680

(Navodilo za uporabo)

Prenos trditvin osebam in uporabe v nedogovorjene namene nista dovoljena.

Izdaja	A					List	Stran	J	K	Identifikacijska številka
Št. obvestila	0-151					2				QCA0006PC
Arhiv										Namesto identifikacijske številke



Iskra  
Elektromehanika  
Krani

## KAZALO

	stran
1. UVOD . . . . .	4
2. SISTEMSKE ZAHTEVE . . . . .	6
3. ZAPISI NA DISKETI . . . . .	7
3.1. Inicializacija diskete. . . . .	7
3.2. Opis sektorja imenika . . . . .	8
3.3. Opis podatkovnega sektorja. . . . .	9
3.4. Datoteke. . . . .	9
3.4.1. Kontrolni blok FCB . . . . .	11
3.4.2. Urejevalec datotek . . . . .	13
4. SISTEMSKI NASLOVI. . . . .	23
5. SISTEMSKE RUTINE . . . . .	26
6. FDOS-3 UKAZI . . . . .	35
6.1. Ukaz LDIR . . . . .	36
6.2. Ukaz DATE . . . . .	38
6.3. Ukaz VIEW . . . . .	38
6.4. Ukaz PRINT. . . . .	39
6.5. Ukaz MERGE. . . . .	40
6.6. Ukaz RENAME . . . . .	41
6.7. Ukaz CHGAT. . . . .	42
6.8. Ukaz DELETE . . . . .	43
6.9. Ukaz LOAD . . . . .	44
6.10.Ukaz RUN. . . . .	45
6.11.Ukaz BUILD. . . . .	46
6.12.Ukaz CNVRT. . . . .	46

Izdeja	1					List	Stran	J	K	Identifikacijska številka
Št. obvestila	V-134					3				OCA0006PC
	Iskra Elektromehanika Kranj									Namesto identifikacijske številke
						Arhiv				

6.13.Ukaz COPY	47
6.14.Ukaz CORF	47
6.15.Ukaz CBIN	48
6.16.Ukaz COMP	49
6.17.Ukaz INIT	50
6.18.Uporabniški ukazi	50
<b>7. SEZNAM NAPAK</b>	<b>52</b>

Prenos tretjim osebam in uporaba v nadogovorjene namene nista dovoljena.

Izdača	A				List	Stran	J	K	Identifikacijska številka
Št. obvestila	U-1\$7				4				QCA0006PC
					Arhiv		Namesto identifikacijske številke		



**Iskra  
Elektromehanika  
Kranj**

## 1. UVOD

FDOS V3.1 (Floppy Disk Operating System) je nov operacijski sistem, ki je bil razvit posebej za sistem ID 1680. Po svoji zgradbi, obsegu in načinu delovanja se bistveno razlikuje od do sedanjih operacijskih sistemov FDOS V1. in FDOS V2. Glavni razlici sta v družbi organizaciji datotek (FDOS V3.1 uporablja verižno organizacijo) in v velikem številu ukazov. Sintaksa ukazov in komuniciranje z uporabnikom sta podobna kot pri FDOS V1.0 in FDOS V2.0, tako da je uporabniku prehod na FDOS V3.1 najlažji.

FDOS V3.1 se nahaja na najnižjih sledenih diskete in sa sestavlja štirje dele:

- dopolnilni rezidenčni program (MONITOR),
- program za delo z disketo (DISK HANDLER),
- program za urejanje datotek (FILE MANAGER SYSTEM) in
- program izvrševalcev (EXEC).

Dopolnilni rezidenčni program (MONITOR) dopoljuje programe v pomnilniku EPROM s spremembami, ki so potrebne zaradi razlik med FDOS V3.1 in FDOS V1.0 ter FDOS V2.1. Vsebuje programe za komunikacije z vhodno-izhodnimi enotami ter zasonski izvrševalci (BOOT STRAP LOADER). Ta ob vsakem vklopu sistema naloži v pomnilnik RAM ostale tri dele FDOS V3.1.

Izdaja	A					List	Stran	J	K	Identifikacijska številka
Št. obvestila	B-187					5				0CA0006PC
Iskra Elektromehanika Kranj										Namesto identifikacijske številke
Obr. 6-496						Arhiv				

Program za delo z disketo (DISK HANDLER) vsebuje rutine za komuniciranje s krmilnikom disketne enote in omogoča pisanje na diskete ter branje z disket. V FDOS V3.1 operacijskem sistemu praviloma poteka delo z disketo izključno preko programov za urejanje datotek, ki je praviloma tudi uporabnik DISK HANDLER -ja.

Program za urejanje datotek (FILE MANAGER SYSTEM) omogoča delo z datotekami (glej poslavje 3.4). Vse delo z datotekami se izvaja izključno preko tečajnega programma.

Program izvrševalcev (EXEC) vodi delovanje FIDOS V3.1. Vsebuje tabelo ukazov FIDOS V3.1 operacijskega sistema in sprejema znake, ki jih vnaša uporabnik. Pri tem paži na prekinitveni kontrolni znak CONTROL/G (\$07 HEX). Pri vnosu tega znaka prekine FIDOS V3.1 sleherno izvajanje, zapre vse odprte datoteke in čaka na nov ukaz. Pri dvakratnem zaporednem vnosu CONTROL/G pa se izvrši inicializacija sistema in aktiviranje zasonskega izvrševalca (BOOTSTRAP LOADER), ki ponovno naloži FIDOS V3.1 z diskete v pomnilnik RAM.

## 2. SISTEMSKE ZAHTEVE

FDOS V3.1 zahteva najmanj 8K pomnilnika RAM na naslovih od \$0000 do \$1FFF (za uspešno delo je priporočljivo uporabiti vsaj 16K pomnilnika RAM), poleg tega pa pomnilnik RAM na naslovih od \$C000 do \$DFFF. Naslovi od \$C000 do \$D000 so rezervirani samo za FDOS V3.1 in jih uporabnik ne sme spremnjati. FDOS V3.1 uporablja le majhen del rezidenčnih programov v pomnilniku EPROM in sicer:

- rutina za vzpostavitev začetnega stanja sistema in naložilni program,
- rutina za obdelavo maskirane, programske in nemaskirane prekinitve,
- rutina za sprejem znaka s tastature,
- rutina za prikaz znaka na zaslonu,
- rutina za izpis znaka na tiskalniku in
- prevajalne rutine.

Možna je uporaba enesa do štirih disketnih enot istočasno.

Zaradi enakosti rezidenčnih programov v pomnilniku EPROM je na sistemu ID 1680, ki uporablja FDOS V3.1, vedno mogoče uporabljati tudi stara operacijska sistema FDOS V1.0 in FDOS V2.0. Zamenjava naredimo preprosto tako, da v disketno enoto 0 vstavimo sistemsko disketo s starim operacijskim sistemom in pritisnemo na tipko RESET.

Izdaja	1					List	Stran	J	K	Identifikacijska številka
Št. obvestila	U-197					7		0CA0006PC		
Iskra Elektromehanika Kranj						Arhiv		Namesto identifikacijske številke		
Obr. 6-496										

### 3. ZAPISI NA DISKETI

#### 3.1. INICIALIZACIJA DISKETE

Disketa vsebuje 77 sledi. Vsaka sled je razdeljena na 26 sektorjev. Na disketi je torej 2002 sektorjev ( šestnajst iško \$7D2 ). V vsak sektor lahko zapišemo 128 znakov. To pomeni, da je na disketi lahko do 256 256 znakov.

Za inicializacijo diskete uporabimo ukaz INIT. S pomočjo tega ukaza se vsi dobri sektorji na disketi povežejo v veriso dobrih sektorjev, vsi v času inicializacije slabih sektorji pa so izloženi. FDOS V3.1 zato omogoča tudi uporabo disket, ki vsebujejo slabe sektorje.

Izjema na disketi je sled 0. Vsi sektorji na tej sledi so rezervirani za operacijski sistem in morajo biti dobri. Prva dva sektorja sta rezervirana za nadaljnji razvoj, tretji sektor pa vsebuje naslednje sistemske podatke:

- številko sledi in sektorja začetnega in končnega sektorja iz verise prostih sektorjev,
- število prostih sektorjev,
- število slabih sektorjev, najdenih med procesom inicializacije diskete,
- številko končne sledi in sektorja sistemskesa programa BOOT in

Izdaža	1					List	Stran	J	K	Identifikacijska številka
Št. obvestila	U-417					8	Arhiv	0CA0006PC		
								Namesto identifikacijske številke		
	Iskra Elektromehanika Kranj									

- datum.

Ostali sektorji so uporabljeni za imenik datotek. Po inicializaciji imajo prosti sektorji diskete enak format. Prva dva bosta vsakega sektorja vsebujeta številko sledi in številko sektorja naslednjega sektorja v verigi sektorjev. V drugem in tretjem bstu je vpisana losična številka sektorja. Ostali bsti so prazni. Ko vpišemo podatke na disketo, kažeta prva dva bsta vsakega sektorja na naslednji sektor datoteke in ne na naslednji sektor v verigi prostih sektorjev.

FDDOS V3.1 disketa ni inicializirana v pravem IBM formatu. Sektorji na disketi fizično niso v istem zaporedju kot losično. Med branjem in obdelavo podatkov iz enega sektorja se slava premika naprej. Tako preskoči nekaj naslednjih sektorjev, če bi disketa imela standardni IBM format, bi moral biti čas obdelave zelo kratek. Če bi obdelava trajala predolgo, bi se lahko zgodilo, da bi sektor izpadel.

### 3.2. OPIS SEKTORJA IMENIKA

Vsek sektor imenika vsebuje podatke o 5 datotekah. Dokler ti podatki še niso vpisani, je namesto prve črke imena ničla. Prvi štirje bsti v vsakem sektorju imenika vsebujejo podatke o povezovanju sektorjev v verigo.

Izdaja	4				List	Stran	J	K	Identifikacijska številka
Št. obvestila	U-111				9				QCA0006PC
 Iskra Elektromehanika Kranj					Arhiv				Namesto identifikacijske številke

### 3.3. OPIS PODATKOVNEGA SEKTORJA

Prvi štirje biti podatkovnega sektorja vsebujejo informacije o povezovanju sektorjev v veriso. V ostalih bitih je vsebina datoteke. Če je datoteka daljša od enega sektorja, je v prvih dveh bitih vpisana številka sledi in sektorja naslednje sta sektorja datoteke. V zadnjem sektorju datoteke sta prva dva bita enaka nič.

### 3.4. DATOTEKE

Iraz "datoteka" se nanaša na vsako zbirko informacij, ki je zapisana na disketi.

Datoteka lahko vsebuje vse ASCII znake. Dva od teh znakov imata lahko za FDOS V3.1 poseben pomen in je zato pri njuni uporabi potrebna previdnost. To sta:

- ASCII znak za CR (carriage return) v šestnajstičkem zapisu \$0D),
- ASCII znak za HT (horizontal tab skip) v šestnajstičkem zapisu \$09).

Izdaja	A				List	Stran	J	K	Identifikacijska številka
Št. obvestila	0-111				10				OCA0006PC
 Iskra Elektromehanika Kranj					Arhiv				Namesto identifikacijske številke

Znak za CR pomeni konec vrstice v enem zapisu datoteke. Znak za HT pa pomeni, da je bil niz praznih mest (blankov = ASCII \$20) odstranjen iz datoteke. Znaku za HT sledi število, ki pove, koliko praznih mest je bilo odstranjenih (2-127). Ko sistem obdeluje datoteko s kontrolnim znakom HT, avtomatično doda željeno število praznih mest.

Ime datoteke lahko vsebuje največ 8 alfanumeričnih znakov. Vsako ime ima lahko še podaljšek, ki označuje vrsto datoteke. Podaljšek je poljubna kombinacija treh alfanumeričnih znakov. FDOS V3.1 razpozna naslednje podaljške, od katerih so v sedanji verziji uporabljeni samo podaljški CMD in SYS.

BIN	SCR
TXT	DAT
CMD	BAC
BAS	DIR
SYS	PRT
BAK	OUT

Primeri: PROG1.BIN

UPORABNI.TXT

SORT1.X20

TEST5 (ime brez podaljška)

Izdaja	A					List	Stran	J	K	Identifikacijska številka
Št. obvestila	V-137					11				QCA0006PC
	Iskra Elektromehanika Kranj									Nambato identifikacijske številke
Obr. 6-496						Arhiv				

Programi v strojni kodi so zapisani v standardnem Motorolinem S-formatu ali pa v binarnem formatu. Navzven se ločijo v tem, da je pri prvih prva črka zapisa S, pri drugih pa B.

Primer: S11E00204FC603... S-format

B11E00204FC603... binarni format

### 3.4.1. Kontrolni blok FCB

Datoteko kreiramo, beremo in spremenjamo s pomočjo kontrolnega bloka FCB (File Control Block), ki vsebuje vse informacije o datoteki. Ta blok je dolg 192 bytov. Lahko ga kreiramo kjerkoli v uporabiškem RAM pomnilniku.

Vsaka datoteka, ki jo želimo obdelovati, potrebuje svoj kontrolni blok. Ko je obdelava datoteke zaključena, lahko lokacije rezervirane za kontrolni blok, uporabimo za kontrolni blok druge datoteke.

Uporabljamo lahko tudi sistemski kontrolne bloke. Kadar želimo izvajati program, ga vtipkamo preko tastature. Operacijski sistem rezervira za naš program kontrolni blok na naslovu \$C840. Vanj vpiše vse specifikacije programa in prvi sektor programa. Če smo ob programu definirali vhodno datoteko, operacijski sistem rezervira zanje kontrolni blok na naslovu \$C980. Za izhodno datoteko rezervira operacijski sistem kontrolni blok na naslovu \$CAC0. Tako ima uporabnik neposreden dostop do vseh vhodnih

Izdaja	1					List	Stran	J	K	Identifikacijska številka
St. obvestila	U-157					12				QCA0006PC
 Iskra Elektromehanika Kranj										Namesto identifikacijske številke
						Arhiv				

podatkov, prav tako lahko neposredno kreira izhodne podatke.

če uporabimo sistemski kontrolne bloki, ne smemo odpirati datotek za branje ali pisanje, ker to naredi že sam operacijski sistem.

Vsebina kontrolnega bloka je naslednja :

- 1.byte : koda funkcije urejevalca datotek (glej 3.4.2. poslavje),
- 2.byte : stanje napake,
- 3.byte : stanje aktivnosti datoteke,
- 4.byte : številka disketne enote,
- 5.byte-12.byte : ime datoteke,
- 13.byte-15.byte : podaljšek imena datoteke,
- 16.byte : atribut datoteke,
- 17.byte : rezerviran,
- 18.byte-19.byte : začetni naslov datoteke na disketi,
- 20.byte-21.byte : končni naslov datoteke na disketi,
- 22.byte-23.byte : dolžina datoteke,
- 24.byte : indikator dostopenosti datoteke,
- 25.byte : rezerviran,
- 26.byte-28.byte : datum kreiranja datoteke,
- 29.byte-30.byte : kazalec naslednjega bloka FCB,
- 31.byte-32.byte : trenutna pozicija datoteke na disketi,
- 33.byte-34.byte : zaporedna številka zapisa datoteke,
- 35.byte : pozicija naslednjega podatka na disketi,
- 36.byte : pozicija podatka naključno organizirane datoteke na disketi,

Izdaja	4					List	Stran	J	K	Identifikacijska številka
Št. obvestila	0-111					13				QCA0006PC
										Namesto identifikacijske številke
	Iskra Elektromehanika Kranj									
Obr. 6-496										

37.byte-47.byte : delovni vmesni pomnilniki,  
48.byte-50.byte : trenutni naslov imenika datoteke,  
51.byte-53.byte : prvi brisan i vhod imenika,  
54.byte-64.byte : novo ime in podaljšek datoteke,  
65.byte : indikator stisknjenoosti podatkov ter  
66.byte-192.byte : vmesni pomnilnik sektorja datoteke.

### 3.4.2. Urejevalec datotek

Pri delu z datotekami si pomagamo z urejevalcem datotek. Naslov urejevalca je \$D406. Kličemo ga z ukazom JSR. Pred tem mora indeksni register vsebovati naslov kontrolnega bloka FCB željene datoteke. Po končanem izvajanjtu urejevalca je v indeksnem registru ista vrednost, kot je bila pred klicom rutine.

Če želimo zapreti vse odprte datoteke, kličemo rutino na naslovu \$D403 z ukazom JSR. Če pride med zapiranjem datoteke do napake, so vse ostale odprte datoteke nezaključene.

Če se je zgodila med izvajanjem urejevalca datotek napaka, je številka napake shranjena v registru A in v 2.bstu bloka FCB. V posojinem kodnem registru (Conditional Code Register) pa je posojna koda Z (zero) v prvotnem stanju (RESET). To lahko preizkusimo z ukazi BEQ ali BNE.

Npr.: -

-  
-

Izdaja	4					List	Stran	J	K	Identifikacijska številka
Št. obvestila	0-111					14				0CA0006PC
 Iskra Elektromehanika Kranj										Namesto identifikacijske številke
Obr. 6-496						Arhiv				

JSR FMS ; kličemo urejevalca datotek

BNE ERROR

-

-

-

ERROR JSR RPTERR ; izpis številke napake

JMP WARMS ; skok v FDOS V3.1

Urejevalec datotek uporabljamo s pomočjo funkcij. V kontrolnem bloku je za kodo funkcije rezerviran prvi byte. Programer lahko uporablja 23 funkcij urejevalca.

Potek programiranja pri uporabi urejevalca je:

- v prvi byte kontrolnega bloka FCB zapišemo kodo funkcije,
- v indeksni register shranimo naslov bloka FCB,
- z ukazom JSR kličemo urejevalca datotek,
- po končanem izvajanjtu urejevalca lahko kontroliramo, če je bilo izvajanje brez napak (z BEQ ali BNE ukazom).

Kode za funkcije urejevalca so številke od 0 do 22 :

- 0 -Koda povzroči branje ali pisanje naslednjega byta.  
če je datoteka pripravljena za branje, bere urejevalnik naslednji byte datoteke na disketu. Po končanem izvajanjtu urejevalca je željeni byte v registru A. če je datoteka pripravljena za pisanje, zapiše urejevalec vsebino

Izdaja	1				List	Stran	J	K	Identifikacijska številka
Št. obvestila	U-413				15				QCA0006PC
Iskra Elektromehanika Kranj					Arhiv				Namensko identifikacijska številka

registra A v izhodni vmesni pomnilnik, ki sa kasneje doda datoteki na disketu.

Primer:

```
LDX #FCB    ; naslov konstantnega bloka datoteke  
              ; pripravljene za branje  
JSR FMS     ; klicanje urejevalca datotek  
BNE NAPAKA ; kontroliranje napake
```

Branji byte je sedaj v registru A.

```
LDAA BYTE  ; naloži byte  
LDX #FCB    ; naslov konstantnega bloka datoteke  
              ; pripravljene za pisanje  
JSR FMS     ; klicanje urejevalca datotek  
BNE NAPAKA ; kontroliranje napake
```

Byte iz registra A je v izhodnem vmesnem pomnilniku.

- 1 Pripravi datoteko za branje. Če datoteka, določene s kontrolnim blokom FCB, urejevalec ne najde, je po končanem izvajanju urejevalca številka napake zapisana v kontrolnem bloku FCB. Pred klicanjem urejevalca mora biti datoteka definirana v kontrolnem bloku (številka disketne enote, ime datoteke, podaljšek imena), številka kode funkcije 1 pa v 1. bytu kontrolnega bloka FCB.

Izdeja	1					List	Stran	J	K	Identifikacijska številka
St. obvestila	U-411					16				QCA0006PC
Iskra Elektromehanika Kranj						Arhiv				Namesto identifikacijske številke

Primer :

```
LDX #FCB    ; naslov kontrolnega bloka  
-           ; definicija datoteke  
-  
-  
LDAA #1     ; koda funkcije  
STAA 0,X    ; shrani kodo v kontrolni blok  
JSR FMS     ; klicanje urejevalca  
BNE NAPAKA ; kontroliranje napake
```

Datoteka je pripravljena za branje.

- 2 - Pripravi datoteko za pisanje. Datoteka ne sme biti že pripravljena za branje, razen če je najprej zaprta in nato ponovno odprta za pisanje.
- 3 - Pripravi datoteko za popravljanje. Funkcija pripravi datoteko za branje in pisanje. Ko je datoteka pripravljena za popravljanje, lahko izvajamo naslednje štiri operacije :
  - branje zaporedno organizirane datoteke,
  - branje naključno organizirane datoteke,
  - pisanje naključno organizirane datoteke in
  - zaključitev datoteke.
- 4 - Zaključi (zapre) datoteko.

Izdaja	4			List	Stran	J	K	Identifikacijska številka			
Št. obvestila	U-157			17				QCA0006PC			
Iskra Elektromehanika Kranj								Arhiv	Namesto identifikacijske številke		

Primer :

```
LDX #FCB ; naslov kontrolnega bloka  
LDAA #4 ; shrani kodo  
STAA 0,X  
JSR FMS ; klicanje urejevalca  
BNE NAPAKA ; kontroliranje napake
```

Datoteka, ki je bila pred tem pripravljena za branje ali  
pisanje, je tako zaključena.

- 5 -Poišči začetek datoteke. Začetek iščemo samo pri datotekah,  
ki so bile pripravljene za branje. Po končani operaciji  
je datoteka zopet pripravljena za branje.
- 6 -Pripravi (odpre) imenik diskete.

Primer :

```
LDX #FCB ; kontrolni blok  
LDAA DRIVE ; številka disketne enote  
STAA 3,X  
LDAA #6 ; koda funkcije  
STAA 0,X
```

- 7 -Beri informacijski zapis imenika. Imenik diskete mora biti

Izdaja	A					List	Stran	J	K	Identifikacijska številka
St. obvestila	U-151					18				QCA00006PC
 Iskra Elektromehanika Kranj										Namesto identifikacijske številke
						Arhiv				

Pred tem pripravljen za branje (koda 6). Če želimo naslednji vhod imenika, ponovno kličemo urejevalca.

Primer: želimo 3.vhod imenika

LDX #FCB ; naslov kontrolnega bloka

LDAA DRIVE ; številka disketne enote

STAA 3,X

LDAA #6 ; funkcijска koda (pripravi imenik za branje)

STAA 0,X

JSR FMS ; klicanje urejevalca

BNE NAPAKA ; kontroliranje napake

LDAB #3 ; števec vhodov imenika

ZANKA LDAA #7 ; funkcijска koda (beri zapis imenika)

STAA 0,X

JSR FMS ; klicanje urejevalca

BNE NAPAKA ; kontroliranje napake

DEC B ; zmanjšanje števca

BNE ZANKA

V kontrolnem bloku FCB je sedaj 3.vhod imenika.

- 8 -Piši informacijski zapis imenika. Pred tem mora biti imenik diskete pripravljen (koda 6).
- 9 -Beri sektor. Pred klicanjem urejevalca datotek mora biti v kontrolni blok FCB vpisan naslov sektorja, ki ga želimo

Izdaja	1				List	Stran	J	K	Identifikacijska številka
Št. obvestila	U-144				19				0CA0006PC
Iskra Elektromehanika Kranj					Arhiv				Namesto identifikacijske številke

brati.

Primer:

```
LDX #FCB      ; naslov kontrolnega bloka
LDAA TRACK    ; željena sled
STAA 30,X
LDAA SECTOR
STAA 31,X    ; željeni sektor
LDAA #9
STAA 0,X      ; koda funkcije
JSR FMS       ; klicanje urejevalca
BNE NAPAKA   ; kontroliranje napake
```

- 10 -Piši sektor. Programer naj te funkcije ne bi uporabljal. Namesto te je bolj primerna funkcija s kodo 0, ki piše samo en byte, povezavo sektorjev pa opravi urejevalec datotek. Pred klicanjem funkcije s kodo 10 mora biti v kontrolni blok FCB vpisan naslov sektorja, kamor želimo vpisati podatke.
- 11 -Rezervirana koda za razvoj operacijskega sistema.
- 12 -Briši datoteko. V kontrolnem bloku FCB morajo biti pred klicanjem urejevalca vsi podatki o datoteki, ki jo želimo brisati (enota, ime datoteke, podaljšek imena). Datoteka ne sme biti pripravljena za vpisovanje ali branje.

Izdaja	A			List	Stran	J	K	Identifikacijska številka
Št. obvestila	U-187			20				QCA0006PC
				Arhiv	Namesto identifikacijske številke			
 Iskra Elektromehanika Kranj								

- 13 -Primerjaj datoteko. Staro ime datoteke mora biti določeno v kontrolnem bloku (na 4.-bytu enota, na 5.-12.-bytu staro ime, na 13.-15.-bytu pa podaljšek). Prav tako mora biti v kontrolnem bloku določeno tudi novo ime in podaljšek na 54. do 64.-bytu. Podaljšek novesa imena ne sme biti izpuščen!
- 14 -Rezervirana koda za razvoj operacijskega sistema!
- 15 -Beri naslednji zaporedni sektor. Datoteka mora biti pripravljena za branje ali pisanje in ne za popravljanje. Če je datoteka pripravljena za branje, ta koda povzroči preskok bytov trenutno branega sektorja, v kontrolni blok FCB pa se na 65.-192 byte vpriše naslednji zaporedni sektor. Če je datoteka pripravljena za pisanje, bo ostanek vmesnega pomnilnika sektorja datoteke (65.-192.-byte kontrolnega bloka FCB) izpolnjen z ničlami in vpisan na disketo. Naslednji znak datoteke bo napisan v naslednji zaporedni sektor.
- 16 -Pripravi sistemski informacijski zapis. Številka enote mora biti določena v kontrolnem bloku, ki sa ne potrebuje nobena datoteka. Vse spremembe, ki jih želimo vnesti v sistemski informacijski zapis, moramo vpisati v vmesni

Izdaja	A					List	Stran	J	K	Identifikacijska številka
Št. obvestila	U-147					21				OCA0006PC
Iskra Elektromehanika Kranj						Arhiv	Namesto identifikacijske številke			

17 -Poišči byte v sektorju. Datoteka mora biti pripravljena za branje ali popravljanje. Številka željenega byta mora biti na 36.-bytu kontrolnega bloka. Številka byta se nanaša na pozicijo byta v trenutno branem ali popravljenem sektorju. Željeni byte dobimo po končani obdelavi urejevalca v registru A. Številka željenega byta ne sme biti manjša od 4, ker prvi štirje byti niso uporabniški.

18 -Zapiši byte v sektor. Datoteka mora biti pripravljena za popravljanje. Byte, zapisan v registru A, je zapisan v sektor na lokacijo, ki jo določimo v 35.-bytu kontrolnega bloka FCB. Ta lokacija ne sme biti manjša od štiri.

19 -Rezervirana za razvoj operacijskega sistema.

20 -Rezervirana za razvoj operacijskega sistema.

21 -Pripravi N-ti zapis. Številka željenega zapisa mora biti na 33. in 36.-bytu kontrolnega bloka FCB (16-bitna dvojiška vrednost). Datoteka mora biti pripravljena za branje ali popravljanje. Prvi podatkovni zapis datoteke ima številko 1.

Izdaja	1				List	Stran	J	K	Identifikacijska številka
Št. obvestila	U-187				22				QCA0006PC
Iskra Elektromehanika Kranj					Arhiv		Namesto identifikacijske številke		

- 22 -Pripravi predhodni zapis. Datoteka mora biti pripravljena za branje ali popravljanje. Funkcija za eno zmanjša zaporedno številko zapisa datoteke (33. in 34. byte) in tako kreira dostop do predhodnega zapisa.

Izdaja	1					List	Stran	J	K	Identifikacijska številka
St. obvestila	U-187					23		OCA0006PC		
 Iskra Elektromehanika Kranj						Arhiv		Namesto identifikacijske številke		

#### 4. SISTEMSKI NASLOVI

V tem poslavju so navedeni naslovi FDOS V3.1. ki jih programer lahko uporabi.

\$C080 - \$COFF : vrstični vmesni pomnilnik

Vrstični vmesni pomnilnik lahko vsebuje do 128 bytov. Z rutino INBUF (glej 5. poslavje) so vtipkani znaki shranjeni v vrstičnem vmesnem pomnilniku. V tem pomnilniku ni brisanih znakov. S tipko CARRIAGE RETURN sporočimo, da je vnos podatkov, ki jih želimo shraniti v vrstični vmesni pomnilnik, končan.

\$CC0E - \$CC10 : register za sistemski datum

Na prvem bytu je shranjen dan, ki mu sledita mesec in leto (samo desetice in enice).

\$CC12 - \$CC13 : naslov uporabniške tabele ukazov

Programer lahko shrani v ti dve lokaciji naslov tabele ukazov, ki jih je sam kreiral (glej 6.14. poslavje).

\$CC14 - \$CC15 : kazalec vrstičnega vmesnega pomnilnika

Ti lokaciji vsebujejo naslov naslednjega byta v vrstičnem vmesnem pomnilniku, ki bo obdelan.

Izdaja	A					List	Stran	J	K	Identifikacijska številka
Št. obvestila	0-157					24				0CA0006PC
Iskra Elektromehanika Kranj						Arhiv				Namesto identifikacijske številke
Obz. 6-496										

**\$CC18 : trenutni znak**

Ta naslov vsebuje znak, ki je bil zadnji obdelan z rutino NXTCH (slej 5. poslavje).

**\$CC19 : predhodni znak**

Na tem naslovu je shranjen predzadnji znak, ki je bil obdelan z NXTCH rutino.

**\$CC1B - \$CC1C : vrednost premika**

Vrednost premika, ki je shranjena na tem naslovu je pristeta naslovu, na katerega se bo strojni program začel vpisovati.

**\$CC20 : vrsta napake**

Številko napake shranijo na ta naslov mnoge funkcije urejevalca datotek.

**\$CC22 : izhodni indikator**

Če je izhodni indikator 0, bo sistemski rutina PUTCHR uporabljala OUTCH sistemski rutino, drugače pa rutino OUTCH2 (slej 5. poslavje).

Izdeja	1					List	Stran	J	K	Identifikacijska številka
St. obvestila	U-137					25				OCA0006PC
Iskra Elektromehanika Kranj						Arhiv				Namensko identifikacijska številka

#### \$CC23 : vhodni indikator

če je vhodni indikator 0, bo sistemski rutina GETCHR uporabljala sistemsko rutino INCH, drugače pa rutino INCH2 (glej 5. poslavje).

#### \$CC24 - \$CC25 : izhodni naslov

Beta vsebuje naslov kontrolnega bloka FCB, ki je bil rezerviran za izhodno datoteko. če je vsebina teh dveh lokacij 0, ni bilo izhodne datoteke. Ti dve lokaciji dobita vrednost 0 s pomočjo sistemske rutine RSTRI0.

#### \$CC26 - \$CC27 : vhodni naslov

Beta vsebuje naslov kontrolnega bloka FCB, ki je bil rezerviran za vhodno datoteko. če je vsebina teh dveh lokacij 0, vhodne datoteke ni bilo. Ti dve lokaciji dobita vrednost 0 s pomočjo rutine RSTRI0.

#### \$CC28 : ukazni indikator

Ta naslov ima od nič različno vrednost, če je DOS klican v uporabniškem programu s pomočjo sistemske rutine DOCMND (glej 5. poslavje).

Izdeja	1	2	3	4	5	6	List	Stran	J	K	Identifikacijska številka
St. obvestila	U-157						26				OCA0006PC
Iskra Elektromehanika Kranj											Namensko identifikacijska številka
							Arhiv				

## 5. SISTEMSKE RUTINE

V tem poslavju so opisane rutine operacijskega sistema in jih programer lahko uporabi v svojem programu. Upoštevati je treba, da so vsebine registrov po klicanju sistemske rutine spremenjene, razen če ni posebej omenjeno. Večino rutin kličemo z JSR ukazom. Vrednosti ali statusi so po končani rutini večinoma v enem od registrov.

**\$CD00 (COLDS) : prva startna vhodna točka**

To vhodno točko uporablja samo program BOOT. Ni namenjena uporabi programerjev.

**\$CD03 (WARMS) : druga startna vhodna točka**

To je slavna vhodna točka iz uporabniškega programa v FDOS V3.1. To rutino kličemo z ukazom JMP. Tako vstopimo v slavno zanko FDOS V3.1, ki kontrolira, če je v vrstičnem vmesnem pomnilniku kakšen ukaz. Če ukaz je, sa poskuša FDOS V3.1 izvesti, če pa ukaza ni, se na začetku vrste izpiše na zaslon klicaj, operacijski sistem pa pričakuje vnos podatkov preko tastature. Če smo FDOS V3.1 klicali z rutino DOCMND, dobí kontrolo uporabniški program.

Izdaja	1					List	Stran	J	K	Identifikacijska številka
Št. obvestila	V-157					27				QCA0006PC
Iskra Elektromehanika Kranj						Arhiv				Namensko identifikacijska številka

**\$CH09 (INCH) : vnos znaka**

**\$CH0C (INCH2) : vnos znaka**

Oba rutini uporabljmo za vnos znaka preko tastature. Po končanem vnosu je znak v registru A. Programer lahko spremeni naslovna besedila rutine INCH (na naslovu \$CD0A in \$CD0B) v drug naslov (npr. vnos znaka z diskete). Naslovna beseda rutine INCH2 (\$CD0D in \$CH0E) se ne smeta spremenjati!

**\$CH0F (OUTCH) : izpis znaka**

**\$CH12 (OUTCH2) : izpis znaka**

Rutina povzročita izpis vsebine registra A na zaslon. Programer lahko spremeni naslovna beseda rutine OUTCH (na naslovu \$CD10 in \$CH11) v drug naslov (npr. izpis na tiskalnik). Naslovna beseda rutine OUTCH2 (\$CD13 in \$CD14) se ne smeta spremenjati!

**\$CH15 (GETCHR) : vnos znaka**

Rutina čaka na vnos znaka s tastature. Znak je po izvajanju v registru A. Če je vhodni indikator (na naslovu \$CC23) 0, uporablja rutino rutino INCH, drugače pa rutino INCH2. Registra B in X ostaneta nespremenjena.

Izdaja	A				List	Stran	J	K	Identifikacijska številka
Št. obveznila	U-133				28				QCA0006PC
Iskra Elektromehanika Kranj					Arhiv				Namesto identifikacijske številke



**\$CD18 (PUTCHR) : izpis znaka**

Rutina izpiše vsebino registra A na izhodno napravo. Če izhodni indikator (na naslovu \$CC22) ni nič, uporablja rutina rutino OUTCH2. Če je izhodni indikator 0 in izhodni naslov datoteke (\$CC24 in \$CC25) ni 0, je vsebina izhodnega naslova datoteke uporabljena kot naslov kontrolnega bloka FCB datoteke, ki je že pripravljena za pisanje. Vsebina registra A je dodana tej datoteki. Če je izhodni naslov datoteke 0, uporablja ta rutina rutino OUTCH. Registrov B in X ostane nespremenjena.

**\$CD1E (PSTRNG) : izpis niza znakov**

Register X mora vsebovati naslov prvega znaka niza, ki ga želimo izpisati. Zadnji znak niza, ki ga želimo izpisati, mora biti \$04. Rutina PSTRNG uporablja rutino PUTCHR. Register B ostane nespremenjen.

**\$CD21 (CLASS) : klasifikacija znaka**

Rutina preizkusi, če je znak alfanumeričen. Znak, ki ga želimo kontrolirati, mora biti v registru A. Če je znak alfanumeričen, je po končanem izvajanjtu rutine prenosni bit posojno kodnega registra (carry) 0. Če znak ni alfanumeričen, je prenosni bit 1. Vsi registri so po izvajanjtu rutine nespremenjeni.

Izdaja	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Identifikacijska številka		
St. obvestila	U-147									29							0CA0006PC		
Iskra Elektromehanika Kranj															Arhiv	Namesto identifikacijske številke			

**\$CD24 (PCRLF) : skok na začetek nove vrste**

Ta rutina uporablja rutino PUTCHR. Registra B in X ostane nespremenjena.

**\$CD27 (NXTCH) : beri naslednji znak**

Rutina bere znak, ki smo ga vtipkali preko tastature. Trenutni znak z naslova \$CC18 je shranjen kot predhodni znak z naslovom \$CC19. Znak, česar naslov hrani kazalec vrstičnega vmesnega pomnilnika, je shranjen kot trenutni znak. Kazalec vrstičnega vmesnega pomnilnika kaže na naslednji znak. Rutina NXTCH uporablja rutino CLASS. Željeni znak je v registru A, prenosni bit posojno kodnega registra pove vrsto znaka, registra B in X pa ostane nespremenjena.

Te rutine ne smemo zamenjati s funkcijsko kodo 0, ki bere znak z diskete. Rutina NXTCH pa bere znak, vtipkan preko tastature.

**\$CD2A (RSTRI0) : vzpostaviti začetno stanje (RESET)**  
vhodno/izhodnih vektorjev

Rutine OUTCH, OUTCH2, INCH ter .INCH2 imajo po operaciji enak naslovni del. Vhodni in izhodni indikator imata vrednost 0. Vhodni naslov datoteke in izhodni naslov datoteke imata prav tako vrednosti 0. Registra A in B ostane nespremenjena.

Izdaja	1				List	Stran	J	K	Identifikacijska številka
Št. obvestila	U-157				30				OCA0006PC
Iskra Elektromehanika Kranj					Arhiv				Namesto identifikacijske številke

#### \$CD2D (GETFIL) : beri specifikacijo datoteke

Register X mora pred klicanjem te rutine vsebovati naslov kontrolnega bloka FCB datoteke, kazalec vrstičnega vmesnega Pomnilnika pa mora kazati na prvi znak specifikacije datoteke v vrstičnem vmesnem Pomnilniku. Rutina GETFIL shrani ime, podaljšek in enoto v kontrolni blok.

#### \$CD30 (LOAD) : beri datoteko v Pomnilnik

Predno kličemo to rutino, mora biti v indeksnem registru naslov kontrolnega bloka FCB datoteke, ki je že pripravljena za branje. Datoteka mora biti zapisana v strojni kod. Začetni naslov v Pomnilniku zapisanega programa, določa program sam. Če želimo začetni naslov spremeniti, to storimo s pomočjo vrednosti premika (Loader Offset Value). To vrednost shranimo na naslov \$CC1B in \$CC1C. Ko računalnik bere program v strojni kod, prišteje začetnemu naslovu programa še vrednost premika. Tako dobí naslov, kamor se program prične nalazati. Vsebina programa ostane nespremenjena.

Če vrednosti premika ne želimo prišteti začetnemu naslovu programa, se moramo prepričati, da je vrednost odmika res nič. Ko je program naložen v Pomnilniku, ga operacijski sistem zaključi. Če pride med branjem programa do napake, je kontrola vrnjena operacijskemu sistemu.

Izdaja	1				List	Stran	J	K	Identifikacijska številka
St. obvestila	U-151				31				QCA0006PC
Iskra Elektromehanika Kranj					Arhiv				Namesto identifikacijske številke
Obr. 6-496									

**\$CD33 (SETEXT) : zapisi podaljšek imena**

To rutino lahko kličemo, če imamo v indeksnem registru naslov kontrolnega bloka datoteke, ki ji bomo zapisali podaljšek.  
Register A mora vsebovati kodo, ki označuje podaljšek:

0 - BIN	6 - SCR
1 - TXT	7 - DAT
2 - CMD	8 - BAC
3 - BAS	9 - DIR
4 - SYS	10 - PRT
5 - BAK	11 - OUT

Vse ostale kode so ignorirane. Vrednost indeksnega registra se med izvajanjem te rutine ne spremeni.

**\$CD36 (ADDBX) : registru X prištej register B**

Po seštevanju ostaneta regista A in B nespremenjena.

**\$CD39 (OUTDEC) : izpis desetiškega števila**

V indeksnem registru mora biti naslov skrajno levega byte 16-bitnega števila brez predznaka. V registru B mora biti indikator, ki pove, če so podatki stisnjeni (Space Suppression

Izdaja	4					List	Stran	J	K	Identifikacijska številka
St. obvestila	U-187					32				OCA0006PC
Iskra Elektromehanika Kranj						Arhiv				Namesto identifikacijske številke

Flas). Če je register B od nič različen, so presledki obravnavani kot vodilne ničle. Če je register B nič, se bo izpis števila začel s prvim od nič različnim znakom. Rutina uporablja rutino PUTCHR.

#### \$CD3C (OUTHEX) : izpis šestnajst iškega števila

Indeksnii register mora vsebovati naslov byta, ki sa želimo izpisati. Byte bo izписан z dvema ASCII šestnajst iškima znakoma. Indeksnii in B register se med izvajanjem ne spremeni. Zopet je uporabljena rutina PUTCHR.

#### \$CD3F (RPTERR) : izpis napake

Rutina izpiše številko napake, ki je zapisana v kontrolnem bloku FCB. Rutina uporablja rutino PUTCHR.

#### \$CD42 (GETHEX) : beri i šestnajst iško število

Računalnik prebere število iz vrstičnega vmesnega pomnilnika. Zato mora kazalec vrstičnega vmesnega pomnilnika kazati na prvi znak željenega števila. Ko se rutina izvrši, dobimo v indeksnem registru že vrednost števila. Število ima lahko vrednost od \$0000 do \$FFFF. Kazalec kaže sedaj na znak, ki sledi separatorju števila (npr. vejica), razen če znak ni CR (carriage return), EOL.

Izdaja	A					List	Stran	J	K	Identifikacijska številka
Št. obvestila	U-187					33				QCA0006PC
	Iskra Elektromehanika Kranj					Arhiv				Namesto identifikacijske številke

(end-of-line) ali nič (NULL). Če je bil prvi obdelan znak separator (npr. vejica), je prenosni bit 0, register B in indeksni register pa imata prav tako vrednost nič. Če prebrano število ni bilo v šestnajstičkem zapisu, F10S preskoči znake v vrstičnem vmesnem pomnilniku do prvega separatorja, prenosni bit je 1, nato pa je kontrola vrnjena na naslov, na katerem je bila klicana vrstica.

#### \$CD45 (OUTADR) : izpiši šestnajstički naslov

Indeksni register mora vsebovati naslov skrajno levega byte 2-bytne šestnajstičke vrednosti. Naslov je izpisani s štirimi ASCII šestnajstičkimi ciframi.

#### \$CD48 (INDEC) : beri desetično število

Rutina bere desetično število (brez predznaka) iz vrstičnega vmesnega pomnilnika. Kazalec vrstičnega vmesnega pomnilnika mora kazati na prvi znak števila. Ko je rutina izvršena, je prenosni bit 0, če je bilo število pravilno. Register B je nič, indeksni register pa vsebuje dvojično kodirano vrednost števila. Kazalec vmesnega pomnilnika kaže na mesto, ki je opisano v rutini GETHEX. Če je prvi preverjan znak separator (npr. vejica), je prenosni bit 0, register B pa je od nič različen, kar pomeni, da število ni bilo najdeno. Indeksni register je nič. Rezultat rutine je zaokrožen na 16-bitno število.

Izdaja	1					List	Stran	J	K	Identifikacijska številka
Št. obvestila	U-187					34				OCA0006PC
Iskra Elektromehanika Kranj						Arhiv				Namensko identifikacijska številka

**\$CD4B (DOCMD) : klicanje FDOS V3.1 kot podrutine**

Uporabnik lahko v programu kliče FDOS V3.1 in izvede željeni ukaz, ki je zapisan v vrstičnem vmesnem pomnilniku. Po operaciji dobí kontrolo zopet uporabniški program. Ukaz v vrstičnem vmesnem pomnilniku se mora končati s kodo \$0D (carriage return) ali \$00 (null).

Rutina omogoča programerju, da iz svojih programov (npr. iz BASIC-a ali zbirnika) uporablja FDOS V3.1 ukaze. Ker se programi za izvajanje FDOS V3.1 ukazov naložijo v pomnilnik na naslovih od \$0000 do \$0FFF (razen ukaza INIT, ki zahteva več prostora) in bi zato prekril uporabnikove programe na tem področju, rutina DOCMD naredi naslednje:

- poišče najvišje 4K pomnilniško področje v sistemu,
- shrani vsebino \$0000 do \$0FFF na najvišje 4K področje,
- pred vrnitvijo v uporabnikov program vrne vsebino najvišjega 4K pomnilniškega področja na naslovih \$0000 do \$0FFF.

S tem je zagotovljeno, da klici DOCMD rutine ne uničijo uporabnikovih programov na naslovih od \$0000 do \$0FFF. Posoj za pravilno delovanje pa je seveda zadostna velikost pomnilnika. Kadar je ta premajhen, opozori FDOS V3.1 uporabnika s posebnim sporočilom, da je vsebina področja od \$0000 do \$0FFF izgubljena.

Izdača	A					List	Stran	J	K	Identifikacijska številka
Št. obvestila	U-157					35				0CA0006PC
Iskra Elektromehanika Kranj						Arhiv				Namesto identifikacijske številke

## 6. FDOS V3.1 UKAZI

Programi za izvajanje vseh FDOS V3.1 ukazov (z izjemo ukaza LDIR) so shranjeni na disketi. Ukazi na disketi se od ostalih datotek razlikujejo v podaljšku imena. Vsí ukazi imajo za imenom in niko še podaljšek CMD. Zato uporabnik sam lahko piše nove ukaze, ne da bi moral spremnijati celoten operacijski sistem. Vsí uporabniški ukazi morajo imeti prav tako podaljšek CMD (glej 6.17. poslavje).

Ukazi so kratice angleških nazivov za posamezne funkcije. Ukaz zaključimo s pritiskom na tipko CARRIAGE RETURN. Kadarkoli pred pritiskom na to tipko lahko uporabimo posebne znake za popravljanje napačno vtipkanega ukaza.

Ukazna vrstica lahko vsebuje do 128 znakov in ima naslednjo obliko:

<ukaz>,<I>:<enota>,<O>:<enota>,<parametri>

kjer je:

<ukaz> eden od FDOS V3.1 ukazov,

<I> ime vhodne datoteke (če je potrebno),

<O> ime izhodne datoteke (če je potrebno),

<enota> številka disketne enote (0 - 3) in

<parametri> parametri po izbiri.

Če številka enote ni navedena pomeni, da je datoteka na disketi v



Izdaja	1	List	Stran	J	K	Identifikacijska številka
Št. obvestila	V-187	36				0CA0006PC
		Arhiv		Namesto identifikacijske številke		

enoti 0.

Pri ukazih MERGE, CHGAT, DELETE, COPY in COMP lahko pri definiciji vhodne in izhodne datoteke uporabimo namesto imena in/ali podaljška zvezdico \*. Zvezdica služi kot univerzalni parameter in pomeni, da se ukaz nanaša na vse datoteke ne sledi na ime in/ali podaljšek. Npr.:

CHGAT \*.\*:1,00

Ukaz bo vsem atributom datotek v enoti 1 dal vrednost 00.

CHGAT,PROG1.\*:1,00

Ukaz bo postavil na 00 atribut vseh tistih datotek disketne enote 1, katerih ime je PROG1, ne sledi na podaljšek.

CHGAT,\*.TXT:1,00

Ukaz povrni, da dobe vrednost 00 vsi atributi datotek v enoti 1, katerih podaljšek je TXT, ne sledi na ime.

#### 6.1. UKAZ LDIR

Ta ukaz povrni prikazovanje vsebine imenika na zaslonu za vsako navedeno disketno enoto.Ukaz ima obliko:

LDIR:<enota>,<parametri>

Izdaja	1					List	Stran	J	K	Identifikacijska številka
Št. obvestila	U-134					37				OCA0006PC
Iskra Elektromehanika Kranj						Arhiv				Namesto identifikacijske številke

če želimo izpis na printer, bomo za parametri navedli L. Za izpis datotek s podaljškom CMD je potreben parameter C, za izpis datotek s podaljškom SYS pa je potreben parameter S. Izpis imenika vsebuje naslednje informacije:

- imena datotek in podaljške imen,
- atr ibute datotek,
- dolžine datotek v sektorjih (hexadecimalne),
- indikator, ki pove,kako je datoteka dosesljiva,
- datum kreiranja datoteke.

Na zaslonu se izpiše naenkrat 20 imen. Če je datotek več, se bo izpisovanje ustavilo. Ob pritisku katerekoli tipke se bo izpis imenika nadaljeval za naslednjih 20 datotek. Ko je imenika konec, se izpiše še datum.

#### Primeri:

- LDIR:1,L      imenik enote 1 se izpiše na tiskalnik,
- LDIR,C      poleg običajnih datotek se izpišejo tudi komandne.
- LDIR,C,S      na ekran se izpišejo komandne datoteke (MERGE.CMD, COPY.CMD, CBIN.CMD in ostale), vse sistemske (ED.SYS, ASM.SYS, BASIC.SYS in URJ.SYS) ter vse ostale datoteke in programi z disketne enote nič.

Izdaja	1				List	Stran	J	K	Identifikacijska številka
Št. obvestila	V-117				38				OCA0006PC
 Iskra Elektromehanika Kranj					Arhiv				Namesto identifikacijske številke

## 6.2. UKAZ DATE

Ukaz uporabljamo, če želimo spremeniti datum na disketi. Ko kreiramo na disketo nove programe ali datoteke, dobija zadnji na disketi napisan datum.

Datum popravljamo na več načinov :

DATE,19.08.81

DATE,21.03..

DATE,03..

## 6.3. UKAZ VIEW

Uporabljamo sa za presledovanje vsebine datotek.

Ukaz ima naslednjo obliko:

VIEW,<ime vhodnega zapisa>:<enota>

Datoteko presledujemo stran za stranjo. Najprej se na zaslonu prikaže prvih 20 vrstic. Po pritisku na tipko LINE FEED se izpiše naslednjih 20 vrstic, dokler datoteka ni izpisana do konca. Če ne želimo presledati celotne datoteke, pritisnemo na tipko RESET in izpisovanje se bo končalo.

Primer : VIEW, COPY.CMD

Izdaja	4				List	Stran	J	K	Identifikacijska številka
St. obvestila	V-117				39				OCA0006PC
 Iskra Elektromehanika Kranj					Arhiv				Namesto identifikacijske številke

Ukaz povroči izpis vsebine ukaza COPY.

VIEW,PROG.MOJ:1

Ukaz izpiše na zaslon vsebino programa PROG.MOJ z disketne enote 1.

#### 6.4. UKAZ PRINT

Ukaz ima obliko:

PRINT,<ime vhodnega zapisa>:<enota>,<parameter>

Ta ukaz je zelo podoben ukazu VIEW. Razlikuje se le v tem, da se proces ustavi, ko je datoteka vsa izpisana. Parameter pove na katerem mediju naj se izpis izvrši. Če želimo izpis na tiskalniku je parameter L, če želimo izpis na zaslonu, pa je parameter C ali pa parametra ne navedemo. Ni pa možno izpisovati istočasno na tiskalnik in na zaslon.

Izdaja	1				List	Stran	J	K	Identifikacijska številka
Št. obvestila	U-137				40				OCA0006PC
Iskra Elektromehanika Kranj					Arhiv				Namesto identifikacijske številke

## 6.5. UKAZ MERGE

Ukaz ima obliko :

MERGE,<0>:<enota>,<I1>:<enota>,<I2>:<enota>

Ukaz povzroči združevanje ene ali več starih datotek v eno novo. Če se pri združevanju zgodijo napake, bo sistem javil številko napake in proces bo prekinjen; nove datoteke ne bo. Kadar definiramo samo eno vhodno datoteko, združevanja ni in ukaz MERGE je v tem primeru enak ukazu COPY (glej 6.13).

Če želimo združiti vse datoteke na disketu v eno samo, ima ukaz obliko:

MERGE,<0>:<enota>,\*.\*:<enota>

Primer : MERGE,NOVA.DAT, P1:1,P2,P3:1

Ukaz povzroči kreiranje nove datoteke z imenom NOVA.DAT, ki je sestavljena iz programov P1 iz disketne enote 1, P2 iz disketne enote 0 in P3 iz disketne enote 1. Programi P1, P2 in P3 so združeni v datoteki NOVA.DAT v istem zaporedju, kot smo jih pisali ob ukazu MERGE.

Izdaja	1					List	Stran	J	K	Identifikacijska številka
Št. obvestila	U-157					41				OCA0006PC
Iskra Elektromehanika Kranj						Arhiv				Namesto identifikacijske številke

## 6.6. UKAZ RENAME

Način uporabe ukaza je:

**RENAME <I>:<enota>, <O>:<enota>**

S tem ukazom preimenujemo datoteko. V tretjem polju ukazne vrstice, kjer definiramo novo ime datoteke, ne smemo navesti številke disketne enote, ker je ta navedena že poleg starega imena.

Ukaz se ne bo izvršil v naslednjih primerih :

- če staro ime za datoteko ni pravilno,
- če imenujemo datoteko z imenom, ki sa ima že druga datoteka,
- če je datoteka, ki jo želimo preimenovati, zaščitena pred brisanjem ali vrisovanjem.

Izdaja	1				List	Stran	J	K	Identifikacijska številka
Št. obvestila	V-117				42				OCA0006PC
 Iskra Elektromehanika Kranj					Arhiv				Namesto identifikacijske številke

## 6.7. UKAZ CHGAT

Ukaz ima obliko:

CHGAT,<I>:<enota>,<nova vrednost atributa>

Ukaz je namenjen spremnjanju atributa datoteke. Definirani so naslednji atributi:

- 00 pomeni, da datoteka ni zaščitena
- 01 pomeni, da je datoteka zaščitena pred branjem
- 02 pomeni, da je datoteka zaščitena pred vojsovanjem
- 04 pomeni, da je datoteka zaščitena pred brišanjem  
(permanentna datoteka)

Če želimo spremeniti atribut vsem datotekam na disketu, ima ukaz obliko:

CHGAT,\*.\*:<enota>,<nova vrednost atributa>

Naslednji ukaz spremeni atribut vsem ukazom na disketu:

CHGAT,\*..CMD:<enota>,<nova vrednost atributa>

Atribut datoteke MOJPROG.T21 s premenimo z ukazom:

CHGAT,MOJPROG.T21:<enota>,<nova vrednost atributa>

Izdaja	A					List	Stran	J	K	Identifikacijska številka
Št. obvestila	v-117					43				OCA0006PC
Iskra Elektromehanika Kranj						Arhiv				Namesto identifikacijske številke

## 6.9. UKAZ LOAD

### 6.8. UKAZ DELETE

Ukaz je oblik:

LOAD,<I>:<enota>,<odstik>

S podatkom <I> definiramo naslov začetnega datoteka, ki mora biti zapisana v strojni kod. Če odstik niz definiran, se datoteka vošče. Uporabljamo ga za brisanje ene ali več datotek, ki niso zaščitene pred brisanjem. Navedene datoteke so lahko na različnih enotah. Če katere od datotek sistem ne najde, se bo izpisalo sporočilo o napaki, brisanje pa je prekinjeno. Vrednost \$10000 je sorazmerna.

Tudi v ukazu DELETE lahko uporabimo zvezdico (\*). Če želimo brisati vse datoteke s podaljškom TXT, zapišemo:

DELETE,\*.TXT:<enota>

Primer: DELETE,V1,V2,V3:1,V0:1,V9

Računalnik briše datoteke v istem zaporedju, kot smo jih ob ukazu zapisali. Če katere od datotek ne najde, se brisanje prekine. Če se program prvič zaraža na naslovu \$100, se po ukazu LOAD shranjuje v memorični način na naslov \$1100. V primeru, ko se program začne na naslovu #FOO0, se po izvajanjу sorazmerno ukaza shranji na naslov \$0000. Tako vidimo, da ukaz pritegne začetnemu naslovu še odstik in pricerom shranji na izredenem naslov.

Izdaja	1				List	Stran	J	K	Identifikacijska številka
St. obvestila	V-137				44				QCAQ0006PC
Iskra Elektromehanika Kranj					Arhiv				Namesto identifikacijske številke

## 6.9. UKAZ LOAD

Ukaz pričimo:

LOAD,<I>:<enota>,<odmik>

S pomočjo tega ukaza se vpisuje v pomnilnik datoteka, ki mora biti zapisana v strojni kod. Če odmik ni definiran, se datoteka vpisuje na naslov, ki sa sama določa, drugače se odmik pristeje začetnemu naslovu, ki sa vsebuje datoteka. Tako dobimo naslov, na katerega se bo vpisala datoteka. Če je tako dobljen naslov večji od \$FFFF, se skrajno levi znak ne upošteva. Vrednost \$10000 je spremenjena v \$0000.

Ukaz LOAD lahko uporabimo tako pri datotekah, ki so zapisane v Motorolinem S-formatu, kot pri tistih, ki so zapisane v binarnem formatu. Ukaz sam ugotovi tip datoteke in nadaljuje z ustreznim izvajanjem.

Primer: LOAD,PROG,1000

Če se program PROG začne na naslovu \$100, se po ukazu LOAD shrani v pomnilnik na naslov \$1100. V primeru, ko se program začne na naslovu \$F000, se po izvajajušem delu ukaza shrani na naslov \$0000. Tako vidimo, da ukaz pristeje začetnemu naslovu še odmik in program shrani na izračunani naslov.

Izdaža	A					List	Stran	J	K	Identifikacijska številka
St. obvestila	U-187					45				OCA0006PC
 Iskra Elektromehanika Kranj						Arhiv				Namesto identifikacijske številke

## 6.10. UKAZ RUN

Oblika zapisa ukaza:

RUN,<ime R>:<enota>,<I>:<enota>,<O>:<enota>,<parametri>

kjer je:

<ime R> ime programa, ki ga bo sistem izvajal.

Ta ukaz uporabimo, kadar želimo izvajati program in pri tem definirati vhodno in/ali izhodno datoteko ter parametre. Sistem vpiše program v pomnilnik, nato ga prične izvajati. Začetni naslov programa mora biti šestnajstiško \$100. Če pride do napake, sistem javi številko napake, izvajanje pa je prekinjeno.

Tudi ta ukaz je, enako kot ukaz LOAD, uporaben pri datotekah v Motorolinem S-formatu in pri tistih v binarnem formatu.

Kadar ne potrebujemo sprotne definicije vhodne in/ali izhodne datoteke in parametrov, lahko uporabimo kraješo obliko ukaza RUN:

<ime R>

Ukaz PROGR.BIN je v vseh posledih ekvivalenten ukazu RUN,PROG.BIN. V obeh primerih se bo program z imenom PROG.BIN z diskete naložil v pomnilnik in se izvajal.

Izdaja	A					List	Stran	J	K	Identifikacijska številka
Št. obvestila	U-133					46				QCA0006PC
	Iskra Elektromehanika Kranj					Arhiv				Namesto identifikacijske številke

## 6.11. UKAZ BUILD

Ukaz kličemo:

BUILD,,<ime 0>:<enota>

S tem ukazom kličemo preprosti urejevalec besedila, ki sa uporabljamo samo za kreiranje novih datotek. V slavnem se uporablja za kreiranje kratkih podatkovnih datotek. Vrstica podatkov je dolga do 128 znakov. Ko je vrsta zaključena, je ni mogoč več popravljati. Pritisak na tipko CARRIAGE RETURN povzroči skok na začetek nove vrstice. Kreiranje je zaključeno s tipkama ESC in CARRIAGE RETURN.

Primer: BUILD,,NOV:1

## 6.12. UKAZ CNVRT

Oblika ukaza je:

CNVRT,<I>:<1,<0>

S pomočjo tega ukaza prevedemo datoteke, kreirane v FDOS V1.0 ali FDOS V2.0 operacijskem sistemu, v datoteke, ki jih lahko uporabljamo v FDOS V3.1 sistemu. Datoteka, pisana za FDOS V2.0, mora biti na disketu v disketni enoti 1, pretvorjeno datoteko pa dobimo na disketu v enoti 0.

Izdaja	1					List	Stran	J	K	Identifikacijska številka
St. obvestila	V-157					47				QCA0006PC
Iskra Elektromehanika Kranj						Arhiv				Namесто identifikacijske številke

Primer: CNVRT,STAR.1:1,NOV.2

V zgornjem primeru pretvorimo datoteko STAR.1 z disketne enote 1 v datoteko, ki je napisana v operacijskem sistemu FDOS V2.0 na disketni enoti 0. Tako dobljeno datoteko lahko obdelujemo v operacijskem sistemu FDOS V3.0.

#### 6.13. UKAZ COPY

Oblika ukaza: COPY

Ukaz uporabljamo za kopiranje celotne vsebine diskete na drugo disketo. V disketno enoto 0 vstavimo disketo, ki jo želimo kopirati. Disketo, na kateri želimo imeti kopijo, vstavimo v enoto 1. Stara vsebina na disketi z disketne enote 1 bo seveda uničena.

#### 6.14. UKAZ COPF

Oblika ukaza: COPF,<0>:<enota>,<1>:<enota>

Ukaz COPF služi za kopiranje datotek. Povzroči prepis ene ali več datotek na isto ali drugo disketo, ne da bi se pri tem vsebina datotek kakorkoli spremenila.

Primer: COPF,\*..CMD:1,\*..CMD

Izdaja	A					List	Stran	J	K	Identifikacijska številka
Št. obvestila	U-167					48				0CA0006PC
Iskra Elektromehanika Kranj						Arhiv				Namesto identifikacijske številke

V zgornjem primeru se na disketo na enoti 1 prepišejo vse datoteke s podaljškom CMD z diskete na enoti 0.

COPF,PROG1.V20,PROG1.V19

V tem primeru dobimo na disketi, ki jo imamo v disketni enoti 0 kopijo datoteke PROG1.V19, ki ima ime PROG1.V20.

#### 6.15. UKAZ CBIN

Oblika ukaza: CBIN,<0>:<enota>,<1>:<enota>

Ukaz povzroči pretvorbo vhodne datoteke iz Motorolinesa S-formata v izhodno datoteko v binarnem formatu (glej 3.4 poslavje). Strojne kode, napisane z ASCII znaki, so po konverziji napisane z binarnimi znaki. Branje in zapисovanje datotek v binarnem formatu je približno dvakrat hitrejše od branja in zapisovanja datotek, napisanih v S-formatu. Tudi prostor, ki sa zaseda datoteka v binarnem formatu je dvakrat manjši od tistega, ki sa zaseda datoteka v S-formatu. Ukaz CBIN je zato pametno uporabiti pri vseh poslovo uporabljenih programih. Vsi FDOS V3.1 ukazi so shranjeni v binarnem formatu.

Primer: CBIN,PROG.BIN,PROG.S

Izdaja	1				List	Stran	J	K	Identifikacijska številka
Št. obvestila	0-147				49				QCA0006PC
Iskra Elektromehanika Kranj					Arhiv				Namesto identifikacijske številke

V zbornjem ukazu smo program PROG.S pretvorili v nov program. PROG.BIN bo na disketi po izvajanju ukaza zapisan v binarni obliki.

#### 6.16. UKAZ COMP

Oblika ukaza: COMP,<0>:<enota>,<1>:<enota>

Če želimo primerjati datoteke ene diskete z datotekami z enakim ali drugačnim imenom na isti ali drugi disketi, uporabimo ukaz COMP.

Primer: COMP,\*..CMD:1,\*..CMD

V zbornjem primeru smo primerjali vse komandne datoteke (ukaze) na disketi na enoti 0 z ukazi na disketi na enoti 1.

Po primerjanju dveh datotek se pojavi kontrolni izpis. Če sta datoteki enaki, se izpiše število primerjanih bytov datotek, drugače pa računalnik javi napako.

Izdaja	1				List	Stran	J	K	Identifikacijska številka
Št. obvestila	V-187				50				0CA0006PC
Iskra Elektromehanika Kranj					Arhiv				Namesto identifikacijske številke

## 6.17. UKAZ INIT

Oblika ukaza :

INIT,<FDOS V3.1>,<Parametri>

S pomočjo tega ukaza lahko kreiramo novo sistemsko disketo brez ukaznih in sistemskih programov na enoti 1. Sistemski disketa mora biti v enoti 0. FDOS V3.1 je operacijski sistem v strojni kodici na sistemski disketi.

če strojnega programa ne kličemo ob INIT ukazu, bo nova kreirana disketa uporabniška.

Primer: INIT,FDOS.OBJ

INIT

V prvem primeru smo kreirali novo sistemsko disketo na enoti 1, v drugem pa uporabniško disketo.

## 6.18. UPORABNIŠKI UKAZI

To so ukazi, ki jih uporabnik napiše sam. Lahko so na disketi in imajo podaljšek CMD ali pa so vpisani v RAM ali ROM pomnilnik. Začetni naslov ukazov mora biti \$100.

Ukaz, vpisan v RAM ali ROM pomnilnik, je program, ki bo izvajan

Izdaja	A					List	Stran	J	K	Identifikacijska številka
St. obvestila	U-161					51				QCA0006PC
Iskra Elektromehanika Kranj						Arhiv				Namesto identifikacijske številke

ob klicu uporabnika. Uporabniški ukazi in njihove vstopne točke naj bodo shranjeni v tabeli na naslednji način:

FCC /UKAZ/ : ime ukaza (največ 8 znakov)

FCB 0

FDB NASLOV : vstopna točka programa (začetni naslov)

-

-

-

FCB 0 : oznaka za konec tabele

DOS prepozna tako tabelo, ker jo mora uporabnik shraniti na naslovih, ki sta določena za uporabniško tabelo ukazov. To sta naslova \$CC12 in \$CC13 (glej 4. poslavje). DOS presleda uporabniške ukaze takoj za sistemskimi, ki so vpisani v pomnilniku. Ukaze, ki so na disketi, presleda nazadnje. Zato lahko uporabnik piše ukaze z istim imenom kot so tisti na disketi.

Ime ukaza in prilagojče argumente vpisemo preko tastature. DOS jih shrani v vrstični vmesni pomnilnik na naslov \$C080. Če naleti ukaz med izvajanjem na napako, zaradi katere mora izvajanje prekiniti, mora ukaz povrniti po izpisu napake skok na začetni naslov \$C003.

Izdaja	1					List	Stran	J	K	Identifikacijska številka
Št. obvestila	U-187					52				0CA0006PC
Iskra Elektromehanika Kranj						Arhiv				Namesto identifikacijske številke

## 7. SEZNAM NAPAK

1. - funkcionalna koda urejevalca datotek ni prava,
2. - željena datoteka je že uporabljena,
3. - specificirana datoteka že obstaja,
4. - specificirane datoteke ni mogoče najti,
5. - rezervirana za razvoj operacijskega sistema,
6. - v imeniku ni več prostora za specifikacijo nove datoteke,
7. - disketa je polna,
8. - datoteka, ki jo želimo brati, je končana,
9. - napaka pri branju sektorja datoteke,
10. - napaka pri zapisovanju sektorja datoteke,
11. - datoteka ali disketa je zaščitena pred vpisom,
12. - datoteka je zaščitena pred brisanjem,
13. - kontrolni blok FCB je nepravilen,
14. - rezervirana za razvoj operacijskega sistema,
15. - rezervirana za razvoj operacijskega sistema,
16. - disketna enota ni pripravljena,
17. - rezervirana za razvoj operacijskega sistema,
18. - napaden pristop do datoteke (datoteka ni pravilno pripravljena),
19. - pozicija naslednjega podatka na disketi je v kontrolnem bloku FCB napačno določena,
20. - rezervirana za razvoj operacijskega sistema,
21. - nepravilna specifikacija datoteke,
22. - rezervirana za razvoj operacijskega sistema,

Izdaja	1					List	Stran	J	K	Identifikacijska številka
Št. obvestila	V-157					53				QCA0006PC
Iskra Elektromehanika Kranj						Arhiv				Namesto identifikacijske številke

- 23 - rezervirana za razvoj operacijskega sistema,  
 24 - rezervirana za razvoj operacijskega sistema,  
 25 - številka zapisa je napačna - datoteka je uničena,  
 26 - napačen ukaz - ponovno vtiskaj ukaz in  
 27 - napačna strojna oprema (premažjen pomnilnik).

Izdaja	A				List	Stran	J	K	Identifikacijska številka
Št. obvestila	V-184				54				0CA0006PC
Arhiv									Namesto identifikacijske številke
 Iskra Elektromehanika Kranj									