



ANVISNING UFD-DOS

Förord

Denna bruksanvisning vänder sig till användare av datorer med UFD-DOS. Avsikten är att ge läsaren erforderlig information om användningen av systemprogrammen för UFD-DOS. Datorns och flexskivenhetens funktion behandlas inte i denna bruksanvisning. Separata bruksanvisningar medföljer resp enhet.

I en övergångsperiod kommer delar av kapitel 1 och kapitel 3 - 6 att finnas med i både den tryckta bruksanvisningen för flexskivenheten och UFD-beskrivningen. Kapitelindelningen i denna beskrivningen överensstämmer med den för flexskivenheten.

- o Om din dator är utrustad med UFD-DOS läser du UFD-beskrivningen. Övriga kapitel, som inte denna beskrivning tar upp, läser du i bruksanvisningen för flexskivenheten.
- o Om din ABC 800 inte har UFD-DOS läser du bruksanvisningen för flexskivenheten. Där finns också beskrivningen på de systemprogram och det DOS som är aktuellt för den datorn.

För att köra systemprogrammen till UFD-DOS krävs att datorn har ett UFD-DOS-PROM samt att rätt CMDINT.SYS används.

För att gammal programvara ska fungera i en dator med UFD-DOS krävs ibland särskilda åtgärder. Läs om detta i kapitel 3.18 DRINI.

Observera att den area i minneskartan som benämns "Ledig för POKE" i tidiga utgåvor av bruksanvisningar för Dator ABC 800 används av UFD-DOS som systemvariabelarea. Program som använder denna area måste ändras, för att undvika felfunktion hos DOS.

Innehåll

Allmänt om UFD 1

UFD - Underbibliotek i flera nivåer 1

1.6 Hämtning och lagring av information	3
1.7 Post och fil	3
1.8 Filnamn och filtyp	5
1.9 Adressering av enhet	6
1.10 Beskrivning av styrkort	6
1.11 Bygling av styrkort	8

3 Systemprogram 11

3.1 Allmänt	11
3.2 SYSDIR.SYS	13
3.3 BASICINI.SYS	13
3.4 ADDOPT.ABS	14
3.5 DEVDES.REL	14
3.6 OPTROSH.REL/OPTROSL.REL	14
3.7 CMDINT.SYS	16
3.8 SYSTEM.ABS	17
3.9 COPY.ABS	18
3.10 COPYLIB.ABS	19
3.11 DELETE.ABS	22
3.12 DISKCHK.ABS	23
3.13 DOSGEN.ABS	26
3.14 ERRCOPY.ABS	28
3.15 LIB.ABS	29
3.16 REPROT.ABS	30
3.17 UFD.ABS	31
3.18 DRINI.BAC	33
3.19 START.BAC	35
3.20 DISKCHK.BAC	35
3.21 LIB.BAC	35
3.22 REPROT.BAC	37
3.23 SYS.BAC	37
3.24 PREABS.BAC	40
3.25 NAMEDISK.BAC	41
3.26 UFD.BAC	42
3.27 TERMOPT.REL	42
3.28 SOFTOPT.REL	42
3.29 SETCAL.BAC	43
3.30 Felmeddelanden under DOS	43

4 Kommandon och instruktioner 49

4.1 Allmänt	49
4.2 Kommandon och instruktioner	49

5 Felmeddelanden	63
6 Kommandosammanfattning	67
6.1 Kommandon och instruktioner under BASIC	67
6.2 Kommandon under DOS	69
10 Sakregister	71

Allmänt om UFD

UFD är förkortningen för "User File Directory" som betyder "Användarens bibliotek".

UFD - Underbibliotek i flera nivåer

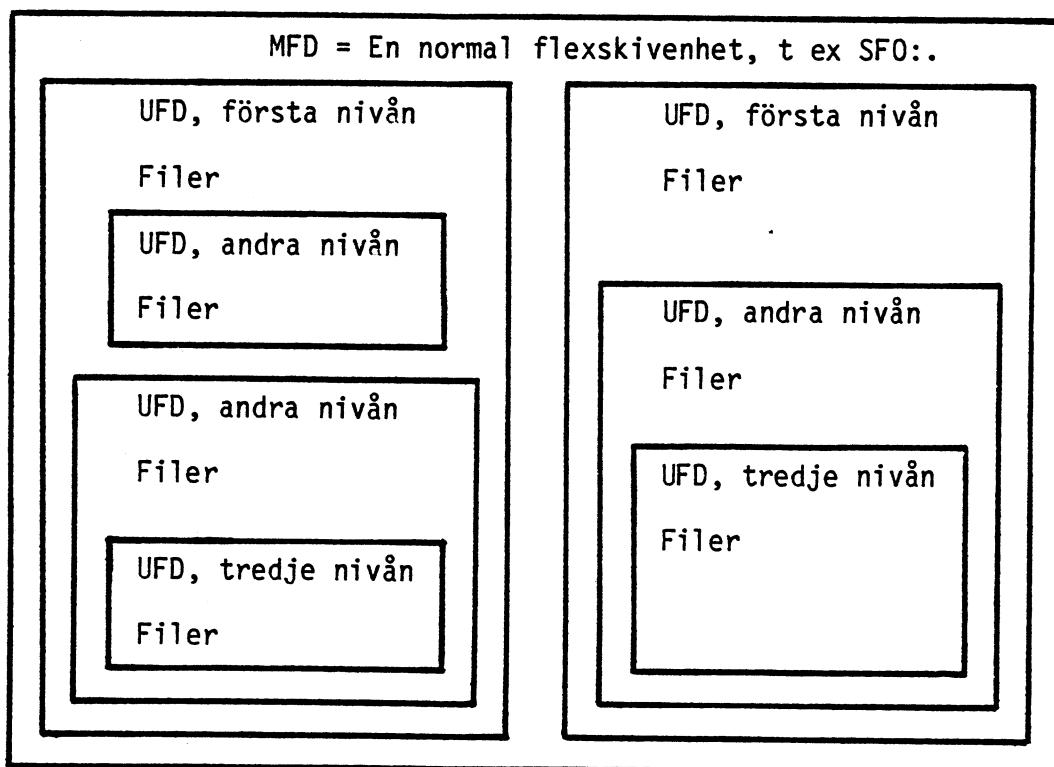
Med särskild programvara är det möjligt att med UFD-DOS skapa och använda underbibliotek i flera nivåer. Denna programvara finns på systemskivan.

Ett UFD-bibliotek kan beskrivas som en fil, som i sin tur är indelad i flera filer, med ett eget separat bibliotek.

Minsta utrymme som fordras för ett UFD-bibliotek är 18 sektorer, eller ett cluster om clusterstorleken är större än 18. Varje UFD-bibliotek kan i sin tur ha underordnade UFD-bibliotek.

Ett UFD-bibliotek behandlas som en extra enhet, UFD:, med ett eget bibliotek. Endast ett UFD-bibliotek kan vara aktivt. Man kan inte samtidigt nå filer i olika UFD-bibliotek. Enhetsnamnet på ett UFD-bibliotek är alltid UFD: oberoende av namnet på just detta UFD-bibliotek.

Programmet UFD, som finns både som UFD.BAC och UFD.ABS för att kunna köras under både BASIC och DOS, används för att skapa ett UFD-bibliotek, radera eller byta namn på ett UFD-bibliotek samt vid förflyttning mellan två UFD-nivåer.



- o Programfil - Fil som innehåller instruktioner (program).

Det finns två sätt att lagra posterna på flexskivan, fil med direktåtkomst och fil med sekvensiell åtkomst.

Definitioner:

- o Direktåtkomst - Metod att läsa eller skriva data i ett minne på sånt sätt att processen är uteslutande beroende av placeringen av dessa data och inte av någon referens till tidigare återvunna eller lagrade data.
- o Sekvensiell åtkomst - Metod att läsa eller skriva data i ett minne på sånt sätt att processen är beroende av placeringen av dessa data samt av en referens till tidigare återvunna eller lagrade data.

I filen med direktåtkomst lagras varje post på en bestämd plats oberoende av övriga poster. Varje post tilldelas ett nummer (postnummer). Om postnumret är känt kan man direkt läsa eller skriva i posten utan att övriga poster berörs.

I filen med sekvensiell åtkomst lagras posterna efter varandra i en lång rad med endast ett skiljetecken (RETURN) mellan posterna. För att kunna läsa en speciell post, måste man läsa samtliga poster framför den aktuella posten. Om man ska ändra en post måste man skriva om hela filen.

För att hantera filer med direktåtkomst används instruktionerna POSIT#, GET# och PUT#. För att skapa och hantera sekvensiella filer används bl a instruktionerna PRINT#, INPUT# och INPUTLINE#.

Innan man kan läsa/skriva i en datafil, måste man öppna filen med OPEN eller PREPARE, varvid filen tilldelas ett filnummer och en filpekare skapas. Filpekaren indikerar var i filen man befinner sig, dvs i vilken position (byte) nästa tecken kommer att läsas/skrivas. Vid OPEN eller PREPARE sätts filpekaren i första positionen (position 0). Filpekaren flyttas med instruktionen POSIT# eller efter en läsning/skrivning.

POSIT# kan också användas för avläsning av filpekarens läge. Efter avslutad läsning/skrivning måste filen stängas med CLOSE.

1.8 Filnamn och filtyp

För att hålla reda på de olika filerna, måste man tilldela varje fil ett filnamn. Filnamnet har följande format:

Filnamn.typ

Filnamnet får bestå av högst åtta tecken, varav det första måste vara en bokstav. I övrigt kan valfria bokstäver och siffror användas. Det är lämpligt att välja filnamnet så att det indikerar innehållet i filen.

Filtypen, som får bestå av högst tre bokstäver och/eller siffror, bör användas för indikering av vilken sorts fil det är frågan om, t ex .TXT för textfiler, .DAT för datafiler, .TMP för temporära filer etc. När det gäller datafiler får valfri typ användas. Nedanstående typer, som används för programfiler, bör dock inte användas.

OBS!

Filnamnet och filtypen skiljs åt med en punkt.

För programfiler används följande filtyper:

- .BAC Programmet lagras i internkodformat. Om man inte anger filtyp och SAVE-kommandot används för lagring av programmet, sätts automatiskt filtypen .BAC. Filer med typen .BAC går snabbare att läsa in eftersom översättningen till internkod inte behöver utföras.
- .BAS Programmet lagras i textform. Om man inte anger filtyp och LIST-kommandot används för lagring av programmet, sätts automatiskt filtypen .BAS. Program med filtyp .BAS tar lite längre tid att läsa in jämfört med program av .BAC-typ. Detta beror på att .BAS-programmen måste översättas till internkod.
- .SYS Programmet är ett systemprogram. Vissa av programmen på systemskivan har filtyp .SYS. Filtypen .SYS bör inte användas för applikationsprogram.
- .ABS Programmet ska köras under skivoperativsystemet DOS. Dessa program är sk absolutadresserade filer. Biblioteksprogrammet LIB.ABS är ett sådant program.
- .REL Programmet som laddas och körs med programmet ADDOPT.ABS. Programmet är relokerbart och består av maskinkod, liksom .ABS-program, men är till skillnad från dessa inte absolutadresserade, utan kan placeras på olika ställen i datorns minne.

I övrigt kan filtypen väljas godtyckligt.

1.9 Adressering av enhet

Följande kringutrustning kan adresseras:

- o Drivenhet 0, som adresseras med DR0:, HD0:, MF0:, **MO0:** eller SF0:
- o Drivenhet 1, som adresseras med DR1:, HD1:, MF1:, **MO1:** eller SF1:, beroende på typ av massminne samt vad som definierats med programmet SYS.BAC
- o Kassettbandspelaren, som adresseras med CAS:
- o Skrivaren, som vanligtvis adresseras med PR: (se bruksanvisning Options-PROM)
- o Bildskärm och tangentbord, som adresseras med CON:

I kommandobeskrivningen anges endast "enhet". En förteckning över de enheter som ingår i systemet fås genom att köra programmet SYS.BAC.

Om "enhet" utelämnas sker sökning på samtliga flexskivenheter, tills kommandot går att utföra. Vill man adressera kassettbandspelaren samtidigt som flexskivenheten är inkopplad, måste alltid CAS: anges. I annat fall kan CAS: utelämnas.

1.10 Beskrivning av styrkort (cont typ)

Styrkortets typ bestäms i första hand av vilket EPROM som sitter på kortet. A6 och A8 betraktas av DOS som identiska.

Enhetsnamnet DR :, som används av många program, kan sättas att motsvara viTken som helst av de andra drivenhetsnamnen.

Styrkort:

- | | | |
|----|--------|--|
| A8 | 8" | Styrkort för 8" flexskivenhet, ABC 838.
Enkel eller dubbel packningstäthet, mjukvaru-
styrd.
Enkel eller dubbelsidig lagring, styrs av flex-
skivan.
Standard enhetsnamn är SF :.
Kortadress är 56Q (46 decTmalt).
Clusterstorlek är 4.
DOS använder sektor 14 som bitmap. |
| A6 | 5 1/4" | Styrkort för 80 cylindern flexskivenhet,
ABC 832.
Endast dubbel packningstäthet.
Enkel eller dubbelsidig lagring, styrs av
omkopplare eller strappningar på kortet. |

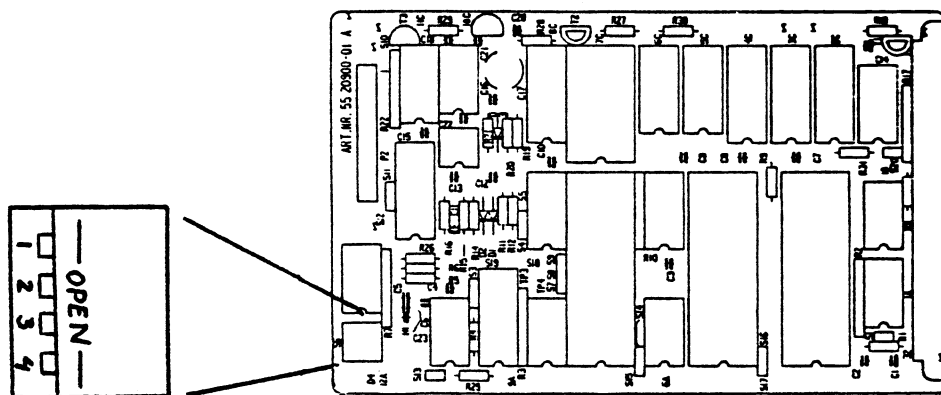
Kan läsa 40 cylinders flexskivor i 80 cylinders
drivenhet, styrs med omkopplare på kortet.
Standard enhetsnamn är MF :.
Kortadress är 54Q (44 decimalt).
Clusterstorlek är 4.
DOS använder sektor 14 som bitmap.

A5 5 1/4" Styrkort för 40 cylinders flexskivenhet.
Enkel eller dubbel packningstäthet, styrs av
omkopplare på kortet.
Enkel eller dubbelsidig lagring, styrs av
omkopplare eller strappningar på kortet.
Standard enhetsnamn är M0 :.
Kortadress är 55Q (45 decimalt).
Clusterstorlek är 1.
DOS använder sektor 6 som bitmap.

De ursprungliga parametrarna sätts vid spänningstillslag eller
RESET.

Funktionsbeskrivning för DIL-omkopplare (S6) på styrkort

Styrkortet 55 20828-01 och 55 20900-01 har plats för fyra stycken DIL-omkopplare för speciella funktioner. Två av omkopplarna är knutna till drivenhet 0 (DRO:) och två till drivenhet 1 (DR1:), men i övrigt är funktionerna helt beroende av den programvara som ingår i styrkortet.



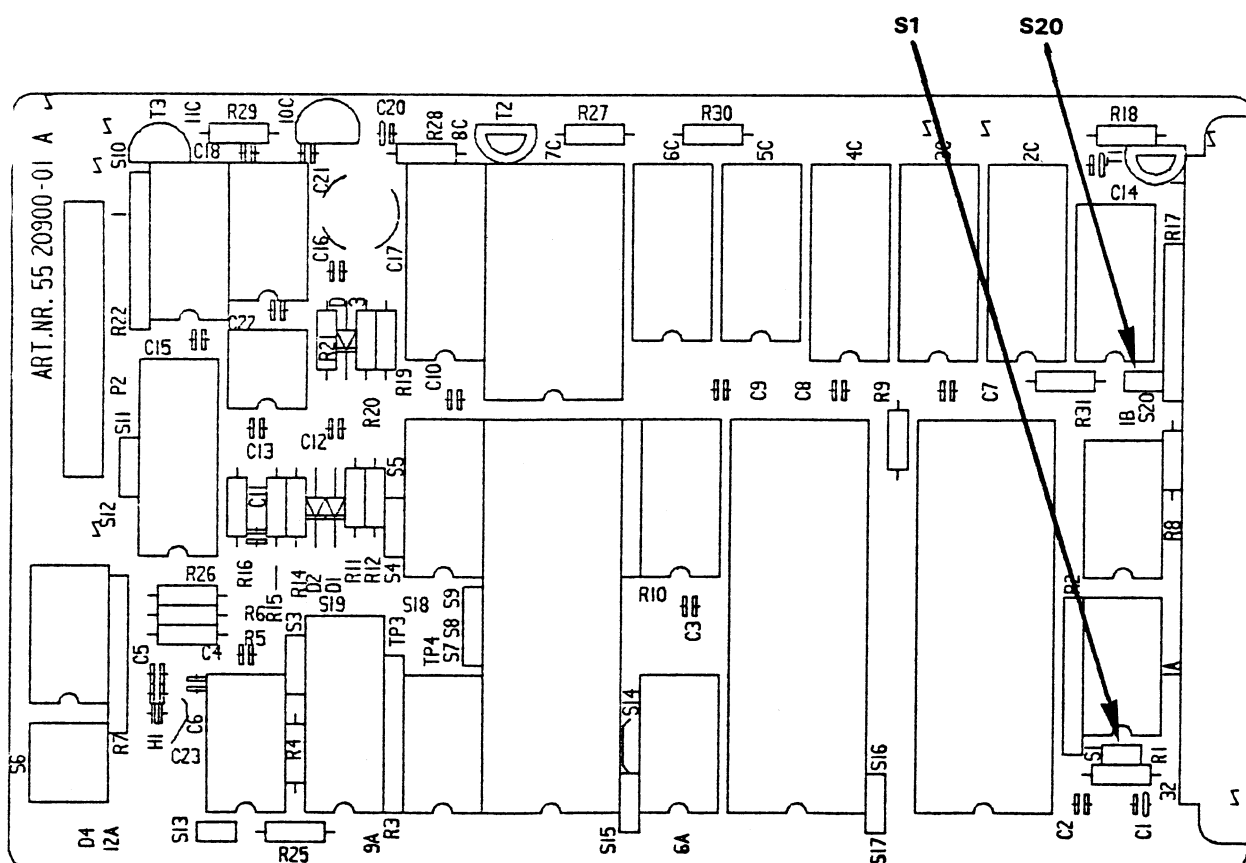
	Programvara för ABC830		Programvara för ABC832		Programvara för ABC838	
	ÖPPEN	SLUTEN	ÖPPEN	SLUTEN	ÖPPEN	SLUTEN
1	* enkel sidig DRO:	dubbel sidig DRO:	enkel sidig DRO:	* dubbel sidig DRO:	ingen funktion	ingen funktion
2	* enkel sidig DR1:	dubbel sidig DR1:	enkel sidig DR1:	* dubbel sidig DR1:	ingen funktion	ingen funktion
3	enkel packnings- täthet DRO:	* dubbel packnings- täthet DRO:	* normal DRO:	830-kompa- tibel utan ver. DRO:	ingen funktion	ingen funktion
4	enkel packnings- täthet DR1:	* dubbel packnings- täthet DR1:	* normal DR1:	830-kompa- tibel utan ver. DR1:	ingen funktion	ingen funktion

* OMKOPPLARENS LÄGE VID LEVERANS

Funktionsbeskrivning för bygel S1 och S20 på styrkort

Styrkort 55 20828-01 och 55 20900-01 har två kontakter (S1 och S20) med kortslutningsbyglar som i vissa lägen ska finnas monterade och i andra avlägsnade. Om kortet kopplas till en dator med UFD-DOS (t ex ABC 802 och ABC 806) gäller följande:

	S1	S20	Adress
ABC 830	Avlägsnas	Monteras	55Q (45 dec)
ABC 832	Monteras	Monteras	54Q (44 dec)
ABC 838	Monteras	Avlägsnas	56Q (46 dec)
Anv ej	Avlägsnas	Avlägsnas	57Q (47 dec)



3 Systemprogram

3.1 Allmänt

Systemskivan innehåller program och rutiner för bl a formatering, kopiering och testning av flexskivorna. Programmen är huvudsakligen av .ABS-typ, dvs de ska köras under skivoperativsystemet DOS (Disk Operating System). Efter övergång till DOS (beskrivs nedan) kan respektive program väljas på två olika sätt. Antingen genom att köra programmet SYSTEM, som presenterar en meny över tillgängliga ABS-program, och därefter trycka in den siffra som motsvarar önskat program (menyval) eller direkt genom att skriva syntaxen för det aktuella programmet. Om du väljer att gå via menyn måste du svara på en del frågor (drivenhet etc) som du slipper om du skriver syntaxen direkt. I det senare fallet anges parametrarna i syntaxen.

Systemskivan innehåller ett antal systemprogram samt två övriga systemfiler, SYSDIR.SYS och BASICINI.SYS. Program med filtyp .BAC är vanliga BASIC-program som startas med kommandot RUN programnamn.

Systemskivan innehåller också ISAM-programmen och beskrivningarna till dem finns i Anvisning ISAM 800. Notera att demoprogrammet för ISAM startas med RUN ISAMDEMO.

Du bör kopiera systemskivan så att du har en reservkopia om något skulle hända med originalet.

Systemskivan innehåller följande filer:

- o SYSDIR.SYS som skapas vid initiering av skiva med programmet DOSGEN.
- o BASICINI.SYS som ger felmeddelanden i klartext. Filen används dessutom för att lagra namnet på maskinkodsprogrammet ADDOPT.ABS och BASIC-strängen "RUN DRINI" som ska läsas in och startas vid autostart.
- o ADDOPT.ABS som startar det relokerbara programmet DEVDES.REL.
- o DEVDES.REL som lägger in parametrar för drivenheterna i skivoperativsystemets variabeltabell.
- o OPTROSH.REL/ som innehåller drivrutiner för hantering av OPTROSL.SYS extra minneskort.
- o CMDINT.SYS som tolkar DOS-kommandona för att .ABS-programmen ska kunna köras.
- o SYSTEM.ABS som presenterar de viktigaste .ABS-programmen och möjliggör menyval av dessa.

- o COPY.ABS som kopierar en fil.
- o COPYLIB.ABS som kopierar flera filer med stöd av biblioteket.
- o DELETE.ABS som raderar filer med stöd av biblioteket.
- o DISKCHK.ABS som testar skivan.
- o DOSGEN.ABS som formaterar skivan samt initierar biblioteket.
- o ERRCOPY.ABS som kopierar felaktiga filer.
- o LIB.ABS som presenterar innehållet på skivorna.
- o REPROT.ABS som rader- och skrivskyddar filer eller tar bort skydd.
- o UFD.ABS som skapar eller raderar UFD-bibliotek.
- o DRINI.BAC som talar om vilket DOS som sitter i maskinen samt sätter DR : att motsvara den enhet som programmet startas från. DRINI avslutas med instruktionen CHAIN "START".
- o START.BAC som avslutar autostartsekvensen på systemskivan.
- o DISKCHK.BAC som laddar och startar DISKCHK.ABS.
- o LIB.BAC som presenterar innehållet på skivorna.
- o LIB.BAS som innehåller källkoden till LIB.BAC.
- o REPROT.BAC som laddar och startar REPROT.ABS.
- o SYS.BAC som ändrar parametrarna för skivoperativsystemet och sparar dessa i filen DEVDES.REL.
- o PREABS.BAC som möjliggör ändring av autostartparametrarna i BASICINI.
- o NAMEDISK.BAC som namnger skivan.
- o UFD.BAC som laddar och startar UFD.ABS.

Systemskivan innehåller dessutom tre program som bara används tillsammans med ABC 806:

- o TERMOPT.REL som innehåller terminal- och teledata-rutin.
- o SOFTOPT.REL som innehåller drivrutin för "Soft-scroll".
- o SETCAL.BAC som sätter den batteridrivna CMOS-kalendern.

.SYS, .ABS, .REL och .BAC anger filtyp. Filtyperna beskrivs utförligt i avsnitt 1.8. Filtyp .ABS anger att programmet ska köras under skivoperativsystemet DOS. Skivoperativsystemet handhar och övervakar dataöverföringen mellan flexskivenheten och datorn. Skivoperativsystemet ligger i en PROM-kapsel i datorn.

För att köra program som ligger under DOS, dvs program med filtypen .ABS, måste systemprogrammet CMDINT.SYS startas. CMDINT innehåller interpretatorn (tolken) för .ABS-programmen (se nedan). CMDINT.SYS laddas in med kommandot BYE.

Under DOS anges felmeddelande enligt följande:

Abort X

Felmeddelanden under DOS överensstämmer till alla delar med de under BASIC, dvs Abort 41 har samma betydelse som Error 41 (Skivan full). Samtliga felmeddelanden finns beskrivna på datorns referenskort och i avsnitt 5. .ABS-programmen innehåller dessutom flera unika felmeddelanden, dvs de gäller för ett speciellt program. Dessa beskrivs i avsnitt 3.30.

3.2 SYSDIR.SYS

Programmet SYSDIR.SYS skapas vid initiering av skiva med DOSGEN. Den finns med för att skapa kompatibilitet med tidigare versioner av DOS för ABC 832 och ABC 838. SYSDIR kan raderas om inte denna kompatibilitet behövs. SYSDIR kan i vissa fall innehålla ett volymnamn på skivan (se DOSGEN.ABS).

3.3 BASICINI.SYS

Programmet BASICINI översätter felkoderna till klartext. Programmet läses in vid påslag och då återställningsknappen RESET trycks in. BASICINI kan kopieras till andra flexskivor med programmet COPYLIB.ABS. Filtypen .SYS anger att det är en systemfil. Du kan kontrollera att BASICINI är inlagd genom att först skriva en bokstav och därefter trycka ner RETURN, varvid datorn ska svara:

Förstår ej

Om datorn istället svarar Error 220 - tryck in RESET och prova igen.

I filen BASICINI lagras den sträng och namnet på den maskinkodsfil som läses in vid autostart (spänningstillslag eller RESET). Dessa läggs in med programmet PREABS.BAC. (se avsnitt 3.24).

3.4 ADDOPT.ABS

Programmet laddar och startar det relokerbara (omplacerbara) programmet DEVDES.REL. Om flera .REL-filer ska startas med ADDOPT så kan de nya filnamnen läggas in från BASIC på följande sätt:

```
OPEN "ADDOPT.ABS" AS FILE 1
PUT #1,CHR$(255)
PUT #1,CHR$(255)+"DEVDES  REL"
PUT #1,CHR$(255)+"FIL2    REL"
```

På detta sätt kan upp till 20 filnamn matas in i ADDOPT.

Observera, att filnamnet måste ha ett speciellt format! Längden på filnamnet måste alltid vara 11 tecken. Inga gemener (små bokstäver) och ingen punkt får ingå. Filnamnet måste börja i position 1 och filtypen måste börja i position 9. Överblivet utrymme måste fyllas med mellanslag.

Efter sista filnamnet måste följande skrivas:

```
PUT #1,CHR$(254)
CLOSE 1
```

VARNING! Felaktig inmatning medför att programmet ADDOPT.ABS förstörs.

3.5 DEVDES.REL

Programmet innehåller parametrar för skivoperativsystemets systemvariabelarea. BASICINI anropar ADDOPT.ABS som i sin tur startar programmet DEVDES.REL. Med hjälp av programmet SYS.BAC kan DEVDES.REL ändras för att få egna parametrar aktiverade vid autostart.

3.6 OPTROSH.REL/OPTROSL.REL

Programmet innehåller drivrutiner för hantering av extra minneskort. BASICINI.SYS anropar ADDOPT.ABS som i sin tur startar programmet OPTROSH.REL. För att OPTROSH.REL ska laddas måste ADDOPT.ABS ändras. Sett från BASIC fungerar denna drivrutin som en "vanlig" flexskivenhet (RAM-floppy) med enhetsbeteckningen RAM:. Fördelen med RAMfloppy jämfört med flexskivenheten är att läsning och skrivning sker mycket snabbt.

Eftersom det inte är någon utökning av primärminnet kan inte större program än tidigare skrivas. Programmen måste rymmas och exekveras i primärminnet (ca 32 kbyte). Detta innebär att stora program även i fortsättningen, måste delas upp i delprogram och instruktionen CHAIN används som logisk avslutning på ett delprogram för att hämta in nästa.

Storleken på ett program kan reduceras genom att t ex texter och konstanter lagras i filer istället för i programmet. Programdelar och datafiler lagras därefter i RAM-floppyn vid start av programmet. Huvudprogrammet lagras som vanligt i primärminnet. Det är möjligt att lagra flera program/datafiler i RAM-floppyn samtidigt. Vid CHAIN laddas programdelen/data från RAM-floppyn till primärminnet. Vid spänningstillslag läggs bibliotekssektorer och bitmap upp. Vid nollställning av primärminnet (CALL(0)) påverkas inte innehållet i extra minnet.

Om du vill radera (formatera), skriver du POKE -6,0, varvid en ny bitmap och nya bibliotekssektioner läggs upp vid nollställning av primärminnet. Detta ger dig möjlighet att gå mellan DOS och BASIC, utan att innehållet i extra minnet förstörs.

Följande extra minnen kan hanteras av denna drivrutin:

Minneskort	64 kbyte RAM	Art nr 190 9626-11
Minneskort	128 kbyte RAM	Art nr 190 9627-11
Dator	128 kbyte	Art nr 190 8089-15

Angående RAM:

GENERELL KOPIERINGSRUTIN

Detta är en generell kopieringsrutin utförd som en flerradig funktion. Den kan användas tillsammans med program som du gjort själv. Inparametrar är ursprungs-enhet (Source#) t ex "DRO:", ny enhet (Dest#) och fil att kopiera (File#) t ex "LIB.BAC".

I loopen som börjar på rad 360 kopieras filen tills den är slut eller ett fel inträffar. På rad 410 kontrolleras att felkoden är 37 eller 38, då har kopieringen gått bra och slutet av filen skrivs ut. Om kopieringen gick bra returneras värdet noll, annars returneras felkoden.

```

320 DEF FNCopy(Source#,Dest#,File#)
330   ON ERROR GOTO 400
340   OPEN Source#+File# AS FILE 1
350   PREPARE Dest#+File# AS FILE 2
360   WHILE -1
370     GET #1,Buf# COUNT 253
380     PUT #2,Buf#
390   WEND
400   ON ERROR GOTO 430
410   IF ERRCODE=38 OR ERRCODE=37 THEN 440
420   CLOSE 1,2
430   RETURN ERRCODE
440   ON ERROR GOTO 460
450   PUT #2,Buf#
460   CLOSE 1,2
470   RETURN 0
480 FNEND

```

AUTOSTART KOPIERING

Den ovan beskrivna kopieringsrutinen kan också användas för att kopiera filer vid autostart. Hur detta program ska startas finns beskrivet i kapitel 3.24 PREABS.BAC.

Följande program kopierar de filer som anges i DATA-satserna från flexskiva till enheten RAM:.

```
100 !
110 ! Program för kopiering av filer vid autostart
120 !
130 Source=MF0:
140 Dest=RAM:
150 READ File
160 WHILE File <> "slut"
170   PRINT File,
180   Felkod= FNCopy(Source, Dest, File)
190   IF Felkod <> 0 THEN PRINT "Fel "; Felkod ELSE PRINT
      "Kopiering klar"
200   READ File
210 WEND
220 END
230 !
240 ! Data-satser med filnamn
250 ! Avbryt med raden slut
260 !
270 DATA LIB.BAC
280 DATA REPROT.BAC
300 DATA slut
```

3.7 CMDINT.SYS

Programmet CMDINT.SYS innehåller kommandointerpretatorn för ABS-programmen. En interpretator är ett program som används för tolkning (översättning). CMDINT tolkar DOS-kommandona. CMDINT måste finnas på någon av skivorna i flexskivenheten för att man ska kunna anropa DOS.

Övergång till DOS

Skriv **BYE** och tryck därefter ner **RETURN**, varvid CMDINT läses in i arbetsminnet och följande visas på skärmen:

```
** Disc Operating System - Ver 6.XX **
Ver 6.XX, 1983-02-10
Copyright 1982 Dataindustrier AB
```

När CMDINT's prompt (-) visas på skärmen är systemet redo att ta emot ett DOS-kommando.

Återgång till BASIC

Skriv `▯BAS` och tryck därefter ner RETURN eller tryck bara in datorns RESET-knapp varvid datorns systemmeddelande, t ex "ABC 800", och markören visas på skärmen.

3.8 SYSTEM.ABS

Programmet används för presentation och menyval av .ABS-programmen.

Syntax:

SYSTEM

1. Sätt systemskivan i någon av drivenheterna.
2. Gå in i DOS (se programmet CMDINT.SYS).
3. Skriv SYSTEM och tryck därefter ner RETURN, varvid följande visas på skärmen:

```

** Systemmeny **
Ver 6.XX, 1983-02-10

1. COPY      - Kopiering en fil.
2. COPYLIB   - Kopiering flera filer.
3. DELETE    - Radering flera filer.
4. DISKCHK   - Testning.
5. DOSGEN    - Formatering.
6. ERRCOPY   - Kopiering felaktig fil.
7. LIB       - Bibliotek.
R Återgång till DOS.

Välj (1-7,R)?
```

4. Välj önskad funktion genom att trycka ner motsvarande siffra följt av RETURN. "R" medför återgång till DOS. När RETURN-tangenten trycks ner laddas aktuellt program in.

Det är också möjligt att gå direkt från DOS till ett speciellt program genom att efter BYE-kommandot skriva syntaxen för programmet. Varje ABS-program kan startas med antingen en fullständig eller en förkortad syntax. Om du väljer ABS-program i menyn ovan, dvs går via SYSTEM, eller väljer att skriva den kortare syntaxen, måste du svara på en del frågor (ange vissa parametrar) som du inte behöver ange om du skriver den fullständiga syntaxen. I det senare fallet ingår parametrarna i syntaxen.

3.9 COPY. ABS

Programmet COPY används för kopiering av en fil från en flexskivenhet till en annan. Kopiering av en fil från en enhet till samma enhet med nytt filnamn är också möjlig. Originalets filnamn och typ måste anges. Om originalets enhetsnamn utelämnas sker sökning på alla enheter. Om kopian filnamn utelämnas får kopian samma filnamn och filtyp som originalet. COPY ska köras under DOS.

Syntax:

```
COPY [DRX:]filnamn.typ,DRY:[Kopians filnamn[.typ]]
```

eller

```
COPY
```

Om du väljer COPY i systemmenyn startar programmet på samma ställe som när du skriver enbart COPY.

Om många filer ska kopieras är det enklare att använda COPYLIB (se avsnitt 3.10)

1. Sätt i systemskivan i drivenhet 0 (DRO:,MFO:...) och en formaterad skiva i drivenhet 1 (DR1:,MF1:..).
2. Gå in i DOS (se programmet CMDINT.SYS).
3. Kopiera programmet LIB.BAC enligt följande:
 - o Alternativ 1
Skriv COPY DRO:LIB.BAC,DR1: och tryck därefter ner RETURN. Om du vill att kopian ska få ett nytt namn och/eller filtyp, skriver du det nya namnet följt av filtypen, direkt efter DR1: i exemplet ovan. Fortsätt med punkt 9 nedan.
 - o Alternativ 2
Skriv COPY och tryck därefter ner RETURN.
4. Följande visas på skärmen:

```
** Doscopy **  
Ver 5.XX, 1983-02-10  
Kopiera från ?
```

5. Ange den enhet, t ex MFO:, där originalfilen finns. Om enheten är DRO: eller DR1: räcker det att svara 0 eller 1. Avsluta med RETURN.

```
Kopiera till ?
```

6. Ange den enhet som filen ska kopieras till, på samma sätt som i punkt 5. Avsluta med att trycka ner RETURN, varvid

följande visas på skärmen:

Fil som ska kopieras ?

7. Skriv filnamn.typ och tryck därefter ner RETURN. På skärmen visas:

Kopians namn ?

8. Skriv filnamn.typ och tryck därefter ner RETURN om kopian ska få ett nytt namn. Om du utelämnar filtypen erhåller kopian samma filtyp som originalet. Enbart RETURN gör att kopian får samma filnamn och typ som originalet. Om en fil med kopians namn redan finns på kopieenheten så gäller punkt 9. Om filen inte finns tidigare så visas följande på skärmen:

** Ny fil **

9. När kopieringen är klar kommer CMDINT's prompt (-) eller systemmenyn på skärmen.

3.10 COPYLIB.ABS

Programmet COPYLIB används för kopiering av flera filer. Kopian får samma filnamn och filtyp som originalet. Om originalfilen är rader- och/eller skrivskyddad, tilldelas även kopian detta skydd. COPYLIB ska köras under DOS.

Syntax:

```
COPYLIB [,V] DRX: [filnamn.typ] ,DRY:
```

eller

```
COPYLIB
```

Om du väljer kopieringsprogrammet i systemmenyn, startar programmet på samma ställe som när du skriver enbart COPYLIB.

V står för verifiering. Under verifieringen kontrolleras att kopian överensstämmer med originalet. Hakparentesen anger att uppgiften får utelämnas, kopieringen sker då utan verifiering. Kopieringen utförs från DRX: till DRY:

Om filnamn.typ utelämnas räknas samtliga filer upp. Om en eller flera tecken anges kommer de filnamn som börjar på dessa tecken att räknas upp vid kopieringen från systemskivan. Om tecknen ".B" ges kommer alla filer som har filtyp med första bokstaven B att räknas upp. Ofullständiga filnamn och filtyper kan också kombineras. Om tecknen "D.A" ges kommer filerna DELETE.ABS, DISKCHK.ABS samt DOSGEN.ABS att räknas upp.

1. Sätt i systemskivan i drivenhet 0 och en formaterad skiva i drivenhet 1.
2. Skriv BYE och tryck därefter ner RETURN, varvid systemet övergår till DOS.
3. Skriv den fullständiga syntaxen och ange aktuella parametrar eller den korta varianten och tryck därefter ner RETURN. Om du skriver en fullständig syntax - fortsätt med pkt 6 nedan.
4. Om du enbart skriver COPYLIB (RETURN), visas följande på skärmen:

```

** Kopiering av filer **
  Ver 6.XX, 1983-02-10

Kopiera från?
Kopiera till?
Verifiera? (J/N)
Kopiera fil?

```

5. Ange originalets respektive kopians drivenhet. Svara J (Ja) eller N (Nej) på frågan om verifiering. Ange filnamn.typ eller del av filnamn (se ovan). Enbart RETURN medför att samtliga filer på originalskivan räknas upp. Glöm inte att trycka ner RETURN efter varje avslutad inmatning.
6. Följande visas på skärmen:

```

Kopiering av filer

Tillgängliga kommandon:

A - Kopiera skivan

X - Uthopp, kopiera inget!

I - Kopiera ej resten av skivan

J - Kopiera denna fil

J - Filnamn.typ - Kopiera med nytt filnamn

```

Beroende på om du angav filnamn.typ, endast filnamn osv, räknas samtliga eller endast vissa av filerna upp på skärmen. Efter varje fil måste du svara med något av ovanstående kommandon eller enbart RETURN. Glöm inte att trycka RETURN efter angiven åtgärd.

A - Samtliga återstående filer kopieras.
Exempel:
Skriv COPYLIB DR0:, DR1: och tryck ner RETURN. När första filen visas på skärmen - skriv A och tryck ner

RETURN, varvid samtliga filer kopieras. På detta sätt slipper man ange för varje fil att den ska kopieras.

X - Uthopp. Inga filer kopieras och "-" visas på skärmen. Systemet är klart att ta emot ett nytt DOS-kommando.

I - Återstående filer kopieras inte. De filer som önskas kopierade, dvs markerade med J, kopieras.

J - Filen kopieras.

J = Filnamn.typ - filen kopieras och tilldelas namnet "filnamn.typ".

RETURN - filen kopieras inte och nästa fil visas på skärmen.

7. För varje fil ange A, X, I, J, eller J = filnamn.typ och tryck därefter ner RETURN. Enbart RETURN medför att den aktuella filen inte kopieras och att nästa fil visas på skärmen.

När J för den sista filen, A eller I har angivits, visas följande på skärmen:

```
Fas 2 Kopiering
Filnamn.typ    X    sektorer    kopierade
:              :      :          :
:              :      :          :
:              :      :          :
-
```

8. Om du angav "V" i syntaxen eller svarade "J" på frågan om verifiering, visas följande på skärmen:

```
Fas 2 kopiering
Filnamn 1.typ  X    sektorer    kopierade
Filnamn 2.typ  Y    sektorer    kopierade
:              :      :          :
:              :      :          :
:              :      :          :
Fas 3 Verifiering
Filnamn 1.typ
Filnamn 2.typ
:
:
-
```

9. Kontrollera att filerna har kopierats till skivan i DRY: genom att köra biblioteksprogrammet LIB.ABS.

3.11 DELETE.ABS

Programmet DELETE används för radering av filer.

Syntax:

```
DELETE DRX: [filnamn.typ]
```

eller

```
DELETE
```

Om du väljer raderingsprogrammet i systemmenyn, startar programmet på samma ställe som när du skriver enbart DELETE.

Om filnamn.typ utelämnas räknas samtliga filer på skivan i DRX: (0 eller 1) upp. För varje fil måste du ange åtgärd, dvs radera, radera ej etc. Om såväl filnamn som filtyp anges, åtgärdas endast den angivna filen. Anges endast filnamnet, räknas samtliga filer med det angivna filnamnet (oberoende av filtyp) upp. Anges endast filtyp, räknas samtliga filer med den aktuella filtypen upp. Om endast det första tecknet i filnamnet anges, räknas samtliga filer med det angivna tecknet i början av filnamnet upp. (Jfr COPYLIB).

1. Sätt i systemskivan i drivenhet 0 och skivan med program som ska raderas i drivenhet 1.
2. Gå in i DOS (se programmet CMDINT.SYS).
3. Skriv den fullständiga syntaxen och ange aktuella parametrar eller den korta varianten (DELETE) och tryck därefter ner RETURN. Om du skriver en fullständig syntax - fortsätt med pkt 6 nedan.
4. Om du enbart skriver DELETE (RETURN), visas följande på skärmen:

```
** Borttagning av filer **  
Ver 6.XX, 1983-02-10
```

```
Drivenhet?
```

```
Radera fil?
```

5. Ange i vilken drivenhet skivan är placerad. Ange filnamn.typ eller del av filnamn (jfr COPYLIB). Enbart RETURN medför att samtliga filer på skivan i den angivna drivenheten räknas upp. Glöm inte att trycka ner RETURN efter varje avslutad inmatning.

6. Följande visas på skärmen:

```
Borttagning av filer
Tillgängliga kommandon:
A - Radera resten av skivan
X - Uthopp, radera inget!
I - Radera ej resten av skivan
J - Radera denna fil
```

Beroende på om du angav filnamn.typ, endast filnamn osv, räknas samtliga eller endast vissa av filerna upp på skärmen.

7. Ange för varje fil något av ovanstående kommandon (enbart RETURN medför att filen inte raderas). Glöm inte att trycka ner RETURN efter varje kommando.
8. När J, A eller I har angivits för den sista filen på bildskärmen, visas följande:

```
Fas 2 Radering
Filnamn1.typ  Filen  raderad
Filnamn2.typ  Filen  raderad
:             :      :
:             :      :
:             :      :
-
```

9. Kontrollera att de angivna filerna har raderats genom att köra biblioteksprogrammet LIB.ABS.

3.12 DISKCHK.ABS

Programmet kontrollerar hela skivan i angiven enhet och reparerar fel om så är möjligt. Alla fel skrivs ut på skärmen.

Kontroller sker av lock-out tabellen, bitmap, bibliotekssektorerna samt filhuvudet för varje fil på skivan. Alla sektorer som tillhör en fil kontrolleras, men oanvänt utrymme på skivan kontrolleras inte.

Felaktigheter visas på skärmen, när de hittas, och du kan välja att låta DISKCHK reparera sektorn eller ej. Vissa permanenta fel kan inte repareras inom filen. Använd i så fall ERRCOPY för att kopiera filen och ersätta sektorer som har permanenta fel.

Reparerbara fel är följande:

1. Fel i lock-out tabellen. Repareras.
2. Fel i bitmap. Repareras.
3. Filer som inte är korrekt stängda efter skrivning. Stängs.
4. Felaktiga UFD-huvud-sektorer. Repareras.
5. Sektorer med permanenta checksummafel (går ej att läsa trots upprepade försök) repareras genom att de data som går att läsa från sektorn skrivs tillbaka tillsammans med en korrekt checksumma. Det kan vara felaktiga data, men det är det bästa som går att göra.
6. Formatfel i filhuvudsektorn. Detta kan inte repareras, men operatören kan välja att radera hela filen som har en dålig filhuvudsektor.
7. Felaktigt filnamn i en bibliotekssektor. Kan inte repareras, men operatören kan välja att radera filen.

Ej reparerbara fel är följande:

1. Alla permanenta fel, förutom checksummafel.
2. Felaktiga data. Data kan vara korrekta ur systemets synpunkt, men kanske inte accepteras av ett applikationsprogram.
3. Felaktigt recordformat. Denna typ av fel gör att DISKCHK skriver ut en varning. Det behöver inte betyda att filen är felaktig ur applikationsprogrammets synpunkt. Repareras inte. Felet kan bero på att ett försök att skriva på filen har misslyckats, och att skrivningen i stället har gjorts på ett annat ställe på skivan. Det kan också vara reserverade men oanvända sektorer som tillhör en fil som inte är korrekt stängd. DISKCHK stänger öppna filer, men oanvänt utrymme inom det sist reserverade clustret kan trots det ge upphov till meddelandet 'Felaktigt recordformat'. (Läs om cluster i avsnitt 3.23).

Syntax:

```
DISKCHK [V,D,N,J,S] DRX:
```

eller

```
DISKCHK
```

Optionerna V,D,N,J och S beskrivs nedan. DRX: är den enhet som ska kontrolleras.

Om du väljer DISKCHK i systemmenyn startar programmet på samma ställe som när du skriver enbart DISKCHK.

1. Sätt i systemskivan i drivenhet 0.
2. Gå in i DOS (se programmet CMDINT.SYS)
3. Skriv den fullständiga syntaxen och ange aktuella parametrar, eller enbart DISKCHK och tryck därefter ner RETURN. Om du skriver en fullständig syntax, fortsätt med punkt 7 nedan.
4. Om du enbart skriver DISKCHK (RETURN), visas följande på skärmen:

```

** Diskcheck **
Ver 6.XX 1983-02-17
Copyright 1982 Dataindustrier AB

Enhet ?
```

5. Ange den enhet som ska testas. Om enheten är DR0: eller DR1: räcker det att skriva 0 eller 1. Tryck därefter ner RETURN, varvid följande visas på skärmen:

```

V - Skriv ut varningar
D - Skriv ut filnamn som checkas
N - Inga ändringar på skivan
J - Korrigera alla funna fel
S - Snabb, varningar checkas ej
Optioner (V,D,N,J,S,ingen)
```

6. Ange de optioner som önskas.
7. Om ingen option ges gäller följande:
 - Alla sektorer som tillhör filer kontrolleras.
 - Varningar skrivs inte ut.
 - Filnamn på korrekta filer skrivs inte ut.
 - Felaktigheter skrivs ut och du får välja om reparation ska utföras.

Utskriften från DISKCHK kan temporärt stoppas genom att en tangent hålls nedtryckt.

En fullständig kontroll av skivan tar några minuter.

Programmet kan avbrytas genom att CTRL hålls nedtryckt medan C trycks ner. Tangentbordet känns av varje gång något skrivs ut på skärmen. Återgång sker då till DOS eller systemmenyn.

8. CMDINT's prompt (-) skrivs ut och DOS är klart för ett nytt kommando.

3.13 DOSGEN.ABS

Programmet används för att initiera skivor, med eller utan en föregående formatering.

Formatering innebär att alla eventuella tidigare filer på skivan raderas och att skivan indelas i sektorer. 5 1/4" skivor måste alltid formateras vid första användningen. Winchester-enheter och 8" skivor är alltid formaterade vid leveransen. 8" skivor ska vara någon av följande typer:

- o Enkelsidig med 26 sektorer/spår och 256 byte/sector.
(BASF märkning: 1D, 26x256 byte/sector)
- o Dubbelsidig med 26 sektorer/spår och 256 byte/sector.
(BASF märkning: 2D, 26x256 byte/sector)

Även andra typer av enkelsidiga eller dubbelsidiga 8" skivor kan användas efter omformatering.

Initiering innebär att alla sektorer på skivan testas med avseende på läsbarhet. När hela skivan är testad skapas ett bibliotek bestående av en sektor bitmap, en sektor lock-out tabell samt 16 sektorer reserverade för filnamn.

I lock-out tabellen markeras de sektorer som används för biblioteket samt de sektorer som vid testningen befinns felaktiga. Samma innehåll skrivs också i bitmap. Detta innebär att felaktiga sektorer är utlåsta från användning. När man skapar nya eller utökar filer på skivan markeras i bitmap vilka sektorer som används. Avslutningsvis skapas filen SYSDIR.SYS på skivan.

Syntax:

```
DOSGEN [F] DRX:[VOLYM.EXT]
```

eller

```
DOSGEN
```

ˆFˆ anger att skivan ska formateras. DRX: är den enhet som innehåller skivan. VOLYM.EXT är ett valfritt volymnamn. Volymnamnet har samma format som ett filnamn dvs högst 8 tecken för namn och högst 3 tecken för filtyp, åtskilda av en punkt. Volymnamnet syns när programmet LIB.BAC körs.

Om du väljer DOSGEN i systemmenyn startar programmet på samma ställe som om du skriver enbart DOSGEN.

1. Sätt systemskivan i drivenhet 0.
2. Sätt skivan som ska formateras i drivenhet 1.
3. Gå in i DOS (se programmet CMDINT.SYS).

4. Skriv den fullständiga syntaxen eller enbart DOSGEN och tryck därefter ner RETURN. Om du skriver en fullständig syntax, fortsätt med punkt 7 nedan.

```
** Disk generation **  
Ver 6.XX, 1983-02-10
```

```
Enhet ?
```

5. Skriv namnet på den enhet som innehåller skivan som ska formateras. Om enheten är DR0: eller DR1: räcker det att skriva 0 eller 1. Avsluta med att trycka ner RETURN.

```
Skall skivan formateras (F/-) ?
```

6. Skriv ett 'F', om skivan ska formateras, och tryck ner RETURN. Ett annat tecken eller enbart RETURN gör att skivan enbart testas och initieras.

7. Följande text visas på skärmen:

```
Skivan i DRX: kommer att rensas helt  
vid initieringen. Är det OK ? <J/N>
```

8. Nu finns en möjlighet att avbryta programmet om t ex vald enhet skulle vara fel. Kontrollera att rätt skiva sitter i DRX:. Skriv J och tryck ner RETURN för att fortsätta, därefter visas följande på skärmen:

```
ABSOLUT säker ?? <J/N>
```

9. Ännu en möjlighet ges att avbryta programmet för att förhindra oavsiktlig radering av en värdefull skiva. Svara J och RETURN för att fortsätta. Om formatering inte är vald i syntaxen eller i menyn ovan gäller punkt 10. Följande visas på skärmen:

```
Skivan formateras !
```

Formateringen tar några minuter.

10. När formateringen är klar, visas följande på skärmen:

```
Skivan testas !
```

```
Nu testas sektor xxxxx
```

Där xxxxx är sektornumret som hela tiden räknas upp på skärmen. Sektorer som befinns felaktiga skrivs ut.

11. När initieringen är klar, visas följande på skärmen:

```
yyyyy användbara sektorer.
```

12. Programmet avslutas och återgång sker till DOS eller systemmeny.

3.14 ERRCOPY.ABS

Programmet används för kopiering av filer med adressmärkningsfel (AM) eller checksummafel (CRC).

Syntax:

```
ERRCOPY [DRX:] Felaktig.typ, [DRY:] Hel.typ
```

eller

```
ERRCOPY
```

Om du väljer kopieringsprogrammet i systemmenyn, startar programmet på samma ställe som när du skriver enbart ERRCOPY.

1. Sätt i systemskivan i drivenhet 0 och den felaktiga skivan i drivenhet 1.
2. Tryck in RESET.
3. Gå in i DOS (se programmet CMDINT.SYS).
4. Skriv den fullständiga syntaxen och ange aktuella parametrar eller den korta varianten (ERRCOPY) och tryck därefter RETURN. Om du skriver en fullständig syntax - fortsätt med pkt 7 nedan.
5. Om du enbart skriver ERRCOPY (RETURN), visas följande på skärmen:

```
** Errcopy **  
Ver 6.XX, 1983-02-10
```

```
Felaktig enhet?  
Kopians enhet?  
Felaktig fil?  
Kopians namn?
```

6. Ange i vilken drivenhet skivan med den felaktiga filen finns. Skriv filens namn och typ. Ange i vilken drivenhet skivan där filen ska lagras finns. Skriv nytt filnamn och ny typ. Enbart RETURN medför att filen lagras med samma namn och samma typ som originalet. Glöm inte att trycka RETURN efter varje avslutad inmatning.

7. Följande visas på skärmen:

```
** Ny fil **  
AM-fel i sektor X  
** Ny fil **  
CRC-fel i sektor X  
Vill du hoppa över sektorn? (J/N)
```

8. Beroende på om du vill hoppa över sektorn eller inte skriv - J eller N och tryck därefter ner RETURN. Om du svarar J stryks den aktuella sektorn och därefter visas "-" på skärmen eller nästa felaktiga sektor. Om du svarar N och felet är ett checksummafel (CRC) - fortsätt med nästa punkt. I annat fall - fortsätt med pkt 11.

9. Följande visas på skärmen:

```
Vill du behålla sektorn som den är? (J/N)
```

10. Beroende på om du vill behålla den felaktiga sektorn eller inte - skriv J eller N och tryck därefter ner RETURN. Om du svarar J ligger sektorn kvar och därefter visas "-" på skärmen eller nästa felaktiga sektor. Om du svarar N - fortsätt med nästa punkt.

11. Följande visas på skärmen:

```
Decimalt fyllnadstal? (0-255)
```

12. Skriv ett tal mellan 0 och 255 och tryck därefter ner RETURN, varvid den felaktiga sektorn fylls med det angivna talet. När samtliga felaktiga sektorer har åtgärdats, kopieras filen. När kopieringen är klar visas "-" på skärmen.

3.15 LIB.ABS

Biblioteksprogrammet LIB.ABS, som körs under DOS, presenterar innehållet på skivorna.

Syntax:

```
LIB [S,N,F,P] [DRX:[filnamn.typ]]
```

1. Sätt i systemskivan i drivenhet 0.
2. Gå in i DOS (se programmet CMDINT.SYS).

3. Skriv LIB eventuellt följt av parametrar och/eller enhet och/eller delar av filnamn.

Om ingen parameter ges visas enbart filnamn på skivorna.
En eller flera av följande parametrar kan ges:

S - Filernas storlekar i antal sektorer skrivs ut.
N - Filernas fysiska filnummer skrivs ut.
F - Filstatus skrivs ut. Skrivskyddade filer markeras med S och raderskyddade filer markeras med R.
P - Utskrift på skrivare.

Om endast biblioteket på en skiva eller enhet ska presenteras anges den enheten som DRX:. Ett ofullständigt filnamn kan ges. Om t ex tecknen LI anges, visas alla filer som börjar på LI. Jämför COPYLIB.ABS. Följande text visas på skärmen:

```
** Library list **  
Ver 6.XX, 1983-02-10  
  
Drive M00:  
  
SYSDIR .SYS BASICINI.SYS ADDOPT .ABS DEVDES .REL  
OPTROSH .REL OPTROSL .SYS ISAMOPT .REL TEROPT .REL  
SOFTOPT .REL CMDINT .SYS SYSTEM .ABS COPY .ABC  
COPYLIB .ABS DELETE .ABS DISKCHK .ABS DOSGEN .ABS  
ERRCOPY .ABS LIB .ABS REPROT .ABS UFD .ABS  
DRINI .BAC START .BAC DISKCHK .BAC LIB .BAC  
LIB .BAS REPROT .BAC SYS .BAC PREABS .BAC  
NAMEDISK .BAC UFD .BAC ISAMDEMO .BAC IDEMO1 .BAC  
DEMOCRE .BAC CREINDEX .BAC ARTIKLAR.DAT ARTIKLAR.ISM  
ARTIKLAR.IS1 SETOPT .BAC SETCAL .BAC  
  
100 av 608 sektorer lediga
```

Om utrymmet på skärmen inte räcker till för att visa alla filnamn kommer markören att stanna i nedre högra hörnet. Varje tryck på mellanslagstangenten visar ytterligare en rad, samtidigt som den översta raden försvinner. Ett tryck på RETURN tangenten avbryter programmet.

4. När presentationen är klar eller om programmet avbryts sker återgång till DOS eller systemmeny.

3.16 REPROT.ABS

Programmet skriv- och raderskyddar filer. Med REPROT kan man också ta bort dessa skydd. REPROT finns inte med i systemmenyn.

Syntax:

```
REPROT [.X][.S][.R] [DRX:] filnamn.typ
```

eller:

```
REPROT
```

Parametrarna X, S och R beskrivs nedan. DRX: är den enhet där filen som ska få ändrad skyddsstatus finns. Om DRX: utelämnas söks filen på samtliga enheter.

1. Sätt en skiva innehållande CMDINT.SYS och REPROT.ABS i någon av drivenheterna.
2. Gå in i DOS (se programmet CMDINT.SYS).
3. Skriv den fullständiga syntaxen eller enbart REPROT och tryck ner RETURN. Om du skriver en fullständig syntax, fortsätt med punkt 6 nedan. Följande visas på skärmen:

```
** File protection utility **  
Ver 6.XX, 1983-12-10
```

```
Filnamn :
```

4. Ange namnet på den fil som ska få ändrad skyddsstatus, eventuellt anges ett enhetsnamn. Avsluta med RETURN. Därefter visas följande på skärmen:

```
X - Avlägsna skydd  
S - Sätt skrivskydd  
R - Sätt raderskydd
```

```
Ange option {X, S, R} :
```

5. Endast en option kan ges. 'X' medför att både skrivning på och radering av filen blir möjligt. 'S' gör filen skrivskyddad. Detta medför också att filen blir raderingsskyddad. 'R' innebär enbart raderingsskydd. Det går fortfarande att uppdatera filen, men inte att radera den.
6. Återgång sker till DOS.

3.17 UFD.ABS

Programmet skapar eller raderar UFD-bibliotek. Byter namn på aktivt bibliotek. Används vid förflyttning mellan två UFD-nivåer. (Läs mer om detta i kapitlet ALLMÄNT OM UFD).

Syntax:

```
UFD [L,M,F,R,C,D] [Enh:] [UFDnamn]
```

eller

```
UFD
```

Optionerna L, M, F, R, C och D beskrivs nedan. Enhet är den flexskivenhet som innehåller det UFD-bibliotek som refereras. UFDnamn är namnet på ett UFD-bibliotek. Filtyp får ej anges.

1. Sätt en skiva innehållande programmen CMDINT.SYS och UFD.ABS i valfri drivenhet.
2. Gå in i DOS (se programmet CMDINT.SYS).
3. Skriv den fullständiga syntaxen och ange aktuella parametrar, eller den kortare varianten, och tryck därefter ner RETURN. Om du skriver en fullständig syntax, fortsätt med punkt 7 nedan.
4. Om du enbart skriver UFD (RETURN), visas följande på skärmen:

```
** User File Directory **  
Ver 6.XX, 1982-02-22  
  
L - Lista aktivt bibliotek  
M - Aktivera master-biblioteket  
F - Aktivera fader-biblioteket  
R - Byt namn på aktivt bibliotek  
C - Skapa nytt bibliotek  
D - Radera aktivt bibliotek  
ingen - Aktivera gammalt bibliotek  
  
Ange option (L,M,F,R,C,D,ingen) :
```

5. Ge någon av följande parametrar följt av RETURN:

- L - Lista namnet på aktivt UFD-bibliotek samt i tur och ordning namnen på överordnade UFD-bibliotek upp till och med MFD.
- M - Aktivera masterbiblioteket samt deaktivera UFD-biblioteket. Enheten UFD: kan därmed inte användas.
- F - Aktivera närmaste överordnade bibliotek. Om detta skulle vara MFD gäller samma förhållande som för M.
- R - Byt namn på aktivt bibliotek. Programmet frågar efter nytt namn. Filtyp får ej anges. Programmet sätter alltid filtypen till .Ufd.

- C - Skapa ett nytt UFD-bibliotek. Programmet frågar efter namnet på det UFD som ska skapas. Filtyp får ej anges. Minst 18 sektorer i följd måste finnas lediga på den skiva där ett UFD ska skapas. Det nya UFD kommer att ligga på nivån närmast under det UFD som är aktivt. Det nya UFD aktiveras.
- D - Radera det UFD som är aktivt och aktivera närmaste överordnade UFD. Inga filer eller UFD-bibliotek får finnas kvar i aktivt UFD. Radera alla filer i aktivt UFD. Filer som ska sparas kan först kopieras över till MFD.
- ingen - Aktivera ett gammalt UFD-bibliotek som ligger på nivån närmast under aktivt UFD. Programmet frågar efter namnet på det UFD som ska aktiveras. Enhetsnamn får ges men inte filtyp.

Om någon av parametrarna R, C eller ingen har givits visas följande på skärmen:

Filnamn :

6. Ange namnet på det UFD som avses. Enhetsnamn får ges, t ex MF1:EGET, men filtyp måste utelämnas.
7. Programmet avslutas.

3.18 DRINI.BAC

Detta program sätter DR : att motsvara den enhet som programmet laddas och startas från. Programmet avslutas med instruktionen CHAIN "start".

Programmet innehåller den rutin som vid uppstart, sätter TIME π till rätt tid enligt CMOS kalendern i ABC 806. (OBS, detta gäller endast ABC806).

BASICINI innehåller strängen "RUN DRINI". Detta medför att programmet startas vid RESET.

Om man har en autostartad programskiva som använder "DR :" för att adressera en drivenhet, så kan den modifieras för att fungera med UFD-DOS på följande sätt:

1. Ladda in programmet DRINI från systemskivan med kommandot LOAD DRINI.
2. Spara programmet på applikationsprogramskivan med kommandot SAVE DRX:DRINI, där DRX: är den enhet som innehåller applikationsprogramskivan.

3. Använd programmet PREABS för att ändra filen BASICINI.SYS på programskivan. Ändra BASIC strängen i filen så att autostart sker på DRINI i stället för START.

När programmet DRINI körs skrivs följande ut på skärmen:

```
DOS är UFD-DOS ver. XX
DR_ : motsvarar MO_ :
```

Vissa applikationsprogram kanske inte tål att utskrift sker på skärmen, t ex om programmet inte inleds med att tömma skärmen. Gör i så fall följande:

- Ladda in programmet enligt punkt 1 ovan.
- Ge kommandot INTEGER.
- Ge kommandot EXTEND.
- Lista programmet fram till initieringsarean (se BASIC-manualen).

```
1160 ! ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **
1170 ! *
1180 T=-1 : F=0 ! True/False
1190 ! *
1200 Conout=T ! Switch T/F here for display/no display
1210 Setold=F ! Switch T/F here for old biased/new linear
    addressing
1220 ! *
1230 Click=T ! Switch T/F here for keyboard click on
    ABC 802/806
1240 ! *
1250 Link="start" ! Name of next program
1260 ! *
```

Om variabeln Conout sätts till 'F' i stället för 'T' så skriver programmet inte ut något på skärmen, men DR_ sätts som beskrivits tidigare.

Vissa program skriver och läser fysiska sektorer på skivan. Program som är skrivna för DOS ABC 6-1x och 8"/800 använder en annan adressering av sektorerna jämfört med vad som gäller för UFD-DOS. Med UFD-DOS kan också den äldre adresseringen användas.

Om variabeln Setold sätts lika med 'T' i stället för 'F' så kommer den äldre adresseringen att gälla för DR_ när programmet är utfört.

Variabeln Click lika med 'F', stänger av klick vid tangenttryckning på tangentbordet ABC 55 eller ABC 77.

Link innehåller namnet på det program som ska startas efter DRINI. Link kan sättas till valfritt filnamn, t ex 'DR1:MITTPROG', 'V24:T' eller 'NUL:'.

När detta är klart, spara programmet på skivan med kommandot SAVE DRINI. Glöm inte att tillfälligt avlägsna skrivskyddet.

3.19 START.BAC

Detta program avslutar autostartsekvensen på systemskivan. Programmet innehåller instruktionerna NO EXTEND, FLOAT samt CHAIN "nul:". Det innebär att datorn, vad gäller inmatning av BASIC-satser, har samma status som om inget program hade utförts.

3.20 DISKCHK.BAC

Detta BASIC-program laddar och startar DISKCHK.ABS. Arbetsgången blir den samma som för DISKCHK.ABS förutom att programmet startas från BASIC med kommandot RUN DISKCHK.

3.21 LIB.BAC

Innehållet på systemskivan, eller på någon annan skiva, presenteras på skärmen med biblioteksprogrammet LIB. Systemskivan innehåller två versioner av LIB, LIB.ABS och LIB.BAC.

LIB.ABS, som ska köras under DOS, beskrivs tidigare i detta kapitel.

LIB.BAC, som körs under BASIC, används på följande sätt:

1. Sätt systemskivan i någon av drivenheterna.
2. Skriv RUN LIB och tryck därefter ner RETURN, varvid följande visas på skärmen:

```
ABC 800 LIB   Version 6.X
=====
1 - Skrivare (Printer)
2 - Storlek
3 - Filstatus
4 - Viss drivenhet
   Välj (1,2,3,4) ?
```

3. Om endast utskrift av filnamn på skivorna önskas, tryck ner RETURN. Om någon av följande parametrar önskas, tryck på motsvarande siffra/siffror och sedan RETURN.

- 1 - Om du vill ha utskriften på skrivare. Ska utskrift ske på någon annan skrivare än den som datorn brukar använda. Måste önskad skrivartyp öppnas innan program-

met LIB körs (se bruksanvisning för Options-PROM).

- 2 - Om du vill ha filstorlekarna i antal sektorer, presenterade på skärmen/skrivaren.
- 3 - Om du vill ha filstatus presenterat på skärmen/skrivaren. Skrivskyddade filer markeras med S. Raderskyddade filer markeras med R.
- 4 - Om du vill ha innehållet på en enda ennet presenterat.

Om du anger t ex 123 RETURN, kommer innehållet på samtliga enheter att listas på skrivaren. För varje fil anges dessutom storlek och filstatus.

4. Om du väljer att trycka på enbart RETURN erhålls följande utskrift på skärmen:

```
ABC 800 LIB   Version 6.X
=====
MOO:          100 av 640 sektorer kvar.

VOLYM:

          Filnamn:   Storlek:  Skriv-Raderskydd:
          -----
          ADDOPT .ABS
          BASICINI.SYS
          CMDINT .SYS
          COPY .ABS
          COPYLIB .ABS
          DELETE .ABS
          DEVDES .REL
          DISKCHK.ABS
          DISKCHK.BAC
          DOSGEN .ABS
          DRINI .BAC
          ERRCOPY .ABS
          LIB .ABS
          LIB .BAC
          =====
PF5=Föregående PF7=Nästa SH-PF5=Föregående enhet
SH-PF7=Nästa enhet
```

Filnamnen presenteras i bokstavsordning. Om skivans innehåll inte får plats på skärmen så kan framstegning ske med PF7 eller mellanslagstangenten. Bakåstegning sker med PF5. Framåt- och bakåstegning till nästa eller föregående enhet görs genom att hålla ner SHIFT-tangenten samtidigt som PF5 resp PF7 trycks ner. Programmet avbryts genom att trycka på RETURN.

5. Om '1' valts i menyn ovan skrivs filerna ut på skrivaren.

6. Om `2` valts i menyn ovan visas i kolumnen "Storlek" hur många sektorer varje fil upptar.
7. Om `3` valts i menyn ovan anges i kolumnerna "Skriv" resp "Raderskydd" om filen har något av dessa skydd.
8. Om `4` valts i menyn ovan visas följande text på skärmen:

Enhet ?

Skriv namnet på den enhet som information önskas från. Om enheten är DRO: eller DR1: räcker det att ange 0 eller 1. Därefter presenteras filnamnen från den aktuella enheten.

LIB.BAC har kompakterats (genomgått squeeze). Detta för att inte programkörningen ska avbrytas med felmeddelandet `Minnet fullt`, som annars kan uppkomma då flera relokerbara filer är laddade när programmet LIB körs.

Den som är intresserad kan se källkoden till LIB-programmet i filen LIB.BAS.

3.22 REPROT.BAC

Detta BASIC-program laddar och startar REPROT.ABS. Arbetsgången är den samma som för REPROT.ABS förutom att programmet startas från BASIC med kommandot RUN REPROT.

3.23 SYS.BAC

Programmet visar tillgängliga enhetsnamn och ger möjligheten att interaktivt ändra parametrar för flexskivenheterna. De ändrade parametrarna kan lagras på skiva i filen DEVDES.REL. SYS.BAC körs under BASIC och används på följande sätt:

1. Sätt systemskivan i valfri enhet.
2. Starta programmet med RUN SYS.
Programmet skriver ut en lista över de enheter som finns initierade i systemet, samt gällande parametrar för flexskivenheterna. Följande visas på skärmen:

```

** System-enheter **
Ver 6.XX, 1982-09-28
Copyright 1982 Dataindustrier AB

```

HighRes minne

System enhetsnamn

```

PR:  V24:  MEM:  CAS:  DRO:  DR1:  DR2:  UFD:
MF0: MF1:  MF2:  MO0:  MO1:  SFO:  SF1:  SF2:
NUL: CON:

```

Parametrar för flexskivenheter

Typ	Cont.-typ	Cs	Clusterstorl.
MF	A6/AB	540	4
MO	A5	550	1
SF	A6/AB	560	4
DR	konverteras till MF		

Vill du ändra parametrar (J/N) ?

De 32 kbyte RAM-minne som utgör MEM: i ABC 802 tolkas av programmet som HighRes-minne. I kolumnen efter Clusterstorlek kommer att stå "Gammal adr" och/eller "Central" om dessa parametrar är satta (se nedan).

- Genom att svara J och trycka på RETURN blir det möjligt att ändra parametrar för flexskivenheterna. Tryck enbart på RETURN för att behålla de ursprungliga parametrarna.

WARNING! Inmatning av felaktiga parametrar gör att existerande filer på skivorna förstörs.

- Parametrarna sparas i filen DEVDES.REL. När ändring av parametrarna är klar visas följande på skärmen:

```

Ska skiv-parametrarna sparas
på skiva (J/N) ?

```

- Om svaret blir J och RETURN skrivs följande på skärmen:

```

Sparas på enhet:      :DEVDES.REL

```

Markören väntar efter `enhet:'. Du ger ett enhetsnamn eller enbart RETURN. Om enbart RETURN ges sker sökning på alla enheter efter en gammal DEVDES.REL.

Nedan ges förklaringar till en del av begreppen som gäller programmet SYS.BAC

Clusterstorlek:

Clusterstorleken är antalet sektorer som motsvaras av en bit i skivans bitmap. Bitmap har alltid 1912 bitar ($8 \times (256 - 17)$). Clusterstorleken måste vara 1, 2, 4, 8, 16 eller 32 sektorer/bit. Den minsta möjliga clusterstorleken beror på det totala antalet sektorer på flexskivenheten.

Kortadress (Cs):

Kortadressen måste vara olika för flexskivenheterna. Det ges i oktal form. Observera att 8" (SF :) har kortadress 56Q och ABC 832 (MF :) har kortadress 54Q. I tidigare versioner av DOS används 54Q till 8" och 55Q till ABC 832. Anvisningar för eventuella ändringar av adresserna finns i kapitel 1.11.

Gammal sektoradressering:

Den gamla adresseringen av sektorer som gäller för DOS "ABC 6-1X" till ABC 830 och Sattco's DOS till 8" "8"/800" kan erhållas med UFD-DOS. Normalt är sektoradresseringen linjär med UFD-DOS.

Central/Lokal:

Denna parameter finns med för kompatibilitet med ABC-Net. En flexskivenhet kan definieras som central när den finns i centralenheten till ABC-Net systemet. (Kallas också ABC-Multiuser-system).

3.24 PREABS.BAC

Programmet används för att skapa eller ändra autostartparametrar i filen BASICINI.SYS. Vid RESET eller spänningstillslag utförs de instruktioner som finns i BASICINI.SYS. Med programmet PREABS kan man välja att lägga in namnet på en maskinkodsfil (.ABS-fil) samt en rad med BASIC-kommandon, eller också enbart en rad med BASIC-kommandon, som ska utföras.

PREABS.BAC körs under BASIC och används på följande sätt:

1. Sätt systemskivan i valfri enhet.
2. Sätt en skiva innehållande filen BASICINI.SYS i en annan enhet.
3. Starta programmet med RUN PREABS. Tryck därefter på RETURN, varpå följande visas på skärmen:

```
ABC 800 PREABS   Version 6.X
=====
Autostart av maskinkodsfil (N)
```

4. Svara J och tryck ner RETURN om den autostartade skivan ska börja med att utföra ett maskinkodsprogram, t ex om drivrutin för RAM: ska laddas eller om parametrar för flexskivenheter ska sättas genom filen DEVDES.REL. Enbart tryck på RETURN gör att enbart BASIC-kommandon kan läggas in. Följande fråga visas på skärmen:

```
Enhet ?
```

5. Ange den enhet som ska få ändrade autostartparametrar. Programmet talar nu om vilken information som förut står i BASICINI:

```
Gamma1 maskinkodsfil: ADDOPT.ABS
Gamma1 BASIC-sträng:  RUN DRINI

Vill du ändra (N)?
```

Om svaret var N på frågan under punkt 3, uteblir den första raden ovan. Svara J och tryck ner RETURN om ändring ska ske. Enbart RETURN avslutar programmet.

6. Mata in önskade data som svar på frågorna:

```
Maskinkodsfil för autostart
Skriv BASIC-sträng (max 158 tkn)
```

7. Inmatningarna avslutas med att trycka ner RETURN.

3.25 NAMEDISK. BAC

Programmet namnger skivan i angiven drivenhet.

NAMEDISK.BAC körs under BASIC och används på följande sätt:

1. Sätt systemskivan i drivenhet 0 och en formaterad men ännu inte namngiven skiva i drivenhet 1.
2. Skriv RUN NAMEDISK och tryck ner RETURN, varvid följande visas på skärmen:

```
** Volymnamn **
```

```
Drive ?
```

3. Tryck ner tangenten `1` följt av RETURN, varvid följande visas på skärmen:

```
Volym:
```

4. Skriv namnet (max 120 tecken), samtliga tecken som finns på tangentbordet kan användas. Tryck därefter ner RETURN.
5. När märkningen är utförd, visas följande på skärmen:

```
Märkningen utförd
```

6. Skriv RUN och tryck därefter ner RETURN, varvid följande visas på skärmen:

```
** Volymnamn **
```

```
Drive ?
```

7. Tryck ner tangenten `1`, varvid följande visas på skärmen:

```
GAMMALT NAMN
```

```
Volym: XXXXXXXXXXXX
```

```
NYTT NAMN (J/N)?
```

Det namn som du tidigare skrev in visas på skärmen. Därefter frågar datorn om du vill ändra namnet.

8. Om du inte vill ändra namnet, tryck ner tangenten `N` följt av RETURN, varvid datorn är klar att ta emot ett nytt kommando.
9. Om du vill ändra namnet, tryck ner tangenten `J` följt av RETURN varvid följande visas på skärmen:

NYTT NAMN

Volym:

10. Skriv det nya namnet (max 120 tkn) och tryck därefter ner RETURN, varvid följande visas på skärmen:

Märkning utförd

3.26 UFD.BAC

Detta BASIC-program laddar och startar UFD.ABS. Arbetsgången blir den samma som för UFD.ABS förutom att programmet startas från BASIC med kommandot RUN UFD.

3.27 TERMOPT.REL

Program endast för ABC 806.

Programmet innehåller drivrutin för att ABC 806 ska kunna användas som Datavisionsterminal och simulering av en ADM3A-terminal. BASICINI.SYS anropar ADDOPT.ABS som i sin tur startar programmet TERMOPT.REL.

För att TERMOPT.REL ska laddas måste ADDOPT.ABS ändras, hur detta går till beskrivs i avsnittet om ADDOPT.ABS.

3.28 SOFTOPT.REL

Program endast för ABC 806.

Programmet innehåller drivrutin för Soft-scroll. För att "Soft-scroll" ska påverka bilden måste BASIC-instruktionen ATTRIBUTE 2 användas. För mer information om ATTRIBUTE se BASIC II Manual för ABC 806. BASICINI.SYS anropar ADDOPT.ABS som i sin tur startar programmet SOFTOPT.REL.

För att denna option ska laddas måste ADDOPT.ABS ändras, beskrivs i avsnittet om ADDOPT.ABS.

Med Softscroll menas att bilden flyttas långsamt och mjukt uppåt på skärmen, t ex vid programlistning.

3.29 SETCAL.BAC

Program endast för ABC 806.

Detta program sätter den batteridrivna CMOS-kalendern. De frågor som ska besvaras i programmet är År, Månad, Datum, Timme, Minut, Sekund, och Veckdagsnummer. SETCAL behöver normalt endast köras vid exempelvis övergång mellan vinter- och sommartid. Om endast realtidsklockan ska sättas, se programexempel i BASIC II Manual för ABC 806 under TIME.

3.30 Felmeddelanden under DOS

Följande felmeddelanden gäller för samtliga .ABS-program:

- o Förstår ej .ABS-programmet finns inte på den sökta enheten.
- o Hittar ej filen Anropad fil finns inte på den sökta enheten.
- o Filnamn saknas .ABS-programmet kräver att ett filnamn ges som inparameter.
- o Felaktigt enhetsnamn Ett enhetsnamn som inte finns i systemet, eller som inte kan användas av .ABS-programmet har använts.

Medanstående felmeddelanden gäller endast vissa program. Inom parentes anges för vilket program meddelandet gäller.

- o Felaktigt filnamn (COPY, COPYLIB, ERRCOPY, REPROT, UFD) Ett filnamn som är ogiltigt har angivits.
- o Hittar inte original-filen ! (COPY, COPYLIB, ERRCOPY) Filen finns ej på refererad enhet.
- o Minnet fullt (COPYLIB) Tabellen med filnamn som har markerats med J är full. Om extra drivrutiner för t ex RAM: är inlänkade minskar tabellens storlek, och därmed det maximala antal filer som kan kopieras på en gång.
- o Kopian finns inte (COPYLIB) Filen finns inte på refererad enhet.

- | | |
|--|--|
| o Du använder ett för gammalt DOS !!
(COPYLIB, DISKCHK) | För att köra systemprogrammen till UFD-DOS krävs att datorn är utrustad med ett UFD-DOS-PROM samt att rätt version av programmet CMDINT.SYS används. |
| o Verifieringsfel
- Kopiera en gång till
(COPYLIB) | Originalfilen och kopiefilen är olika vid verifiering. |
| o Verifieringsfel
(COPYLIB) | Visas på samma rad som filnamnet under verifieringen. |
| o Sektorfel på kopian
(COPYLIB) | Kopian av filen går inte att läsa vid verifiering. |
| o Sektorformatfel på originalfilen
(COPYLIB) | Originalfilen går inte att läsa. |
| o Sektorformatfel på kopian
(COPYLIB) | Kopian av filen går inte att läsa vid verifiering. |
| o Olika fillängd
(COPYLIB) | Originalfilen och kopiefilen är olika vid verifiering. |
| o Original och kopia måste vara olika
(COPYLIB) | Samma enhet har angivits både som original och kopia. |
| o Fel i biblioteket
(COPYLIB, DELETE) | Skivan inte klar eller läsfel. |
| o Felaktigt record-format
(DISKCHK) | Filens placering på skivan stämmer inte med informationen i biblioteket. |
| o Felaktigt filhuvud
(DISKCHK) | Det filhuvud som pekas ut av biblioteket är inte korrekt. |
| o Enheten ej klar
(DISKCHK) | Ingen skiva i drivenheten eller luckan inte stängd. |
| o Skrivskydd
(DISKCHK) | Skivan är skrivskyddad och kan inte repareras. |
| o Hittar ej denna record
(DISKCHK) | Biblioteket refererar till en post som inte finns i filen. |
| o Checksummafel
(DISKCHK) | Felstatus från drivenhetens styrkort: Sektorns checksumma är fel. |

- | | |
|---|--|
| o Data förlorade
(DISKCHK) | Felstatus från drivenhetens styrkort:
Styrkortet kunde inte skriva sektorn. |
| o Fysiskt slut på
enheten
(DISKCHK) | Felstatus från drivenhetens styrkort:
Styrkortet har fått ett kommando som
gäller en sektor utanför skivan. |
| o Okänd status
(DISKCHK, DOSGEN) | Felstatus från drivenhetens styrkort:
Tyder på ett fel på styrkortet. |
| o Filen ej korrekt
stängd
(DISKCHK) | Biblioteket har reserverat ett större
utrymme än vad filen upptar. |
| o Dubbelreserverat
cluster. SKIVAN
EJ SÄKER !
(DISKCHK) | Biblioteket har reserverat samma
utrymme till två eller flera filer. |
| o Bitmap felaktig
(DISKCHK) | Läsfel på bitmap. |
| o Biblioteket
felaktigt
(DISKCHK) | Läsfel på biblioteket. |
| o Lockout-tabell
felaktig
(DISKCHK) | Läsfel på lockout-tabellen. |
| o Cluster utlåst i
bitmap men ej
använt, kan vara
utlåst av DOSGEN.
(DISKCHK) | Ett cluster, som inte används av någon
fil, är utlåst i bitmap. DOSGEN läser
ut sektorer som har läsfel vid
initieringen. |
| o Felaktigt record-
format
Ej kritiskt.
(DISKCHK) | Kan vara överblivet utrymme i det
cluster som är sist reserverat för
filen. |
| o Du måste vara in-
loggad
(DISKCHK, DOSGEN) | Gäller vid körning under ABC-NET. |
| o Du måste vara
super-user
(DISKCHK, DOSGEN) | Gäller vid körning under ABC-NET. |
| o Internt fel
(DISKCHK, DOSGEN) | Kan vara fel på styrkort eller driv-
enhet. |
| o Fel vid NET-operation
(DISKCHK) | Gäller vid körning under ABC-NET. |

- o Enheten ej klar, Ingen skiva i drivenheten eller luckan
kontrollera! ej stängd.
Skriv in "K" <RETURN>
när du är klar !
(DOSGEN)

- o Skivan skrivskyddad! Skrivskyddstejp på (5 1/4") eller
Skriv in "K" <RETURN> Skrivskyddstejp saknas (8")
när du har
kontrollerat!
(DOSGEN)

- o Det finns redan ett Gäller vid körning under ABC-NET.
job
(DOSGEN)

- o Hittar inte Gäller vid körning under ABC-NET.
"DOSGEN.CEN"
(DOSGEN)

- o Felaktig login Gäller vid körning under ABC-NET.
(DOSGEN)

- o Något fel vid ladd- Gäller vid körning under ABC-NET.
ning av "DOSGEN.CEN"
(DOSGEN)

- o Tyvärr kan jag inte En eller flera av de sektorer, som
initiera biblioteket upptas av biblioteket (0-31), är oan-
på denna skiva. vändbara.
Prova med formatering
om du inte redan har
gjort det!!
(DOSGEN)

- o Felaktig option En option, som inte finns till program-
(DOSGEN, REPROT, met, har angivits.
UFD)

- o Kan inte formatera, Ingen skiva i drivenheten eller luckan
kontrollera!!! är öppen vid formatering.
(DOSGEN)

- o CRC-fel i sektor Checksummafel.
xxxxx
(ERRCOPY)

- o AM-fel i sektor Adressmärkesfel, sektorn finns inte.
xxxxx
(ERRCOPY)

- o Fel vid skrivning Kan vara skrivskydd eller fel i biblio-
i biblioteket teket.
(REPROT)

- o Kan inte radera "MFD" (UFD) Aktivt UFD kan inte raderas om inget UFD är aktivt.
- o Kan inte byta namn på "MFD" (UFD) Det finns inte möjlighet att byta namn på aktivt UFD om inget UFD är aktivt.
- o Enhet får ej anges! (UFD) Det är inte tillåtet att skapa ett UFD på en annan enhet.
- o "MFD" har ingen fader! (UFD) Master File Directory är den högsta möjliga nivån.
- o Inget namn angivet! (UFD) UFD-namn måste ges tillsammans med angiven option.
- o Filtyp får ej anges! (UFD) UFD-bibliotek har alltid namn med filtypen .ufd.
- o Filen finns redan! (UFD) Ett försök att skapa ett nytt UFD med samma namn och på samma nivå som ett redan existerande UFD.
- o Ej tillräckligt kontinuerligt utrymme på enheten ! (UFD) Minst 18 sektorer i följd måste finnas lediga på skivan när ett nytt UFD skapas.
- o Fel vid läsning/skrivning på skivan! (UFD) Kan vara checksummafel i ett UFD.
- o Biblioteket måste vara tomt! (UFD) Ett försök att radera ett UFD-bibliotek som innehåller filer eller underordnade UFD-bibliotek har gjorts.
- o Kan ej ställa UFD-pekaren. (UFD) Gäller vid körning under ABC-NET. En referens till ett bibliotek, som ligger utanför användarens behörighet, har gjorts.

4 Kommandon och instruktioner

4.1 Allmänt

Detta kapitel behandlar endast de kommandon och instruktioner som används för lagring och hämtning av information på flex-skiva. För ytterligare information om BASIC II - se BASIC II-manualen.

I det följande anges

- o instruktion - med I. Dessa kan endast användas i program.
- o kommando - med K.
- o instruktion och kommando - med IK.
- o reserverade ord - med fet stil, t ex **LOAD**, **SAVE** och **RUN**.
- o uppgifter som kan utelämnas - inom hakparentes, t ex [enhet:].
- o alternativa uppgifter - med snedstreck, t ex "data"/strängvariabel.
- o ytterligare uppgifter - med punkter, t ex "data"/strängvariabel,...,....

Allmänt gäller att

- o enhet adresseras med RAM:, PR:, V24:, MEM:, CAS:, DRO:, DR1:, DR2:, UFD:, MFO:, MF1:, MOO:, MO1:, SFO:, SF1:, SF2:, NUL:, CON: (se avsnitt 1.9 samt programmet SYS.BAC).
- o om enhet utelämnas, adresseras alltid UFD:, om den finns, och därefter i den ordning som anges ovan.
- o filnamn får bestå av maximalt åtta bokstäver/siffror, varav det första tecknet måste vara en bokstav. Dessutom kan filtyp (3 tecken) användas valfritt som förtydligande av filnamnet (se avsnitt 1.8).
- o filtyp behöver inte anges. Undantag finns dock. Detta anges isåfall vid respektive syntax. Om filtyp utelämnas utförs kommandot i första hand på filtyp .BAC och därefter på .BAS.
- o RETURN måste tryckas ner efter varje avslutad inmatning.

4.2 Kommandon och instruktioner

K LOAD[enhet:] filnamn[.typ]

Kommandot används för laddning av programfilen

"filnamn.typ". Om filen återfinns på den angivna skivan eller på kassetbandet, nollställs först datorns arbetsminne och därefter sker inläsning. Om filen inte återfinns på skivorna erhålls felkoden Error 21 eller felmeddelandet HITTAR EJ FILEN (om systemfilen BASICINI.SYS är aktiverad). Om filen finns på flera skivor laddas filen från den enhet där den först påträffas.

Exempel:

LOAD DRO:DEMO.BAC

Programmet DEMO.BAC på skivan i DRO: överförs till minnet.

LOAD DEMO

Programmet DEMO (.BAC eller .BAS) överförs till minnet, från den skiva där det först påträffas.

LOAD CAS:DEMO

Programmet DEMO (.BAC eller .BAS) på kassetbandet överförs till minnet.

LOAD CAS:

Nästa program på kassetbandet överförs till minnet.

K. RUN[enhet:] Filnamn [.typ]

Kommandot används för laddning och start av programmet "filnamn.typ". Programmet laddas in på samma sätt som när **LOAD** används (se ovan).

Exempel:

RUN DR1:DEMO.BAS

Programmet DEMO.BAS på skivan i DR1: överförs till minnet och därefter startar programkörningen.

RUN LIB

Programmet LIB (.BAC eller .BAS) överförs till minnet från den skiva där det först påträffas. Därefter startar LIB.

RUN CAS:BERÄKN

Programmet BERÄKN på kassettbandet överförs till minnet och därefter startar körningen. Utelämnas filnamnet (endast **RUN CAS:**) överförs nästa program på kassettbandet till minnet.

RUN

RUN medför att programmet i datorns arbetsminne körs. I detta fall kan **LOAD** användas för laddning av programmet.

K SAVE [enhet:] filnamn [.typ]

Kommandot används för lagring av programmet "filnamn.typ". Programmet lagras i internkodformat. Om filtyp inte anges, erhålls filtyp .BAC. Om det sedan tidigare finns en fil med det aktuella filnamnet och den aktuella filtypen på skivan kommer denna att skrivas över.

OBS!

Nya skivor måste formateras innan de kan användas för lagring av program och/eller data (se avsnitt 3.13).

Exempel:

SAVE SFO:PROG.PRO

Programmet PROG.PRO lagras på skivan i SFO:

SAVE PROG

Programmet PROG lagras i internkodformat (typ = .BAC) i den första enhet där detta är möjligt.

SAVE CAS:PROG

Filen PROG lagras i internkodformat (typ = .BAC) på kassettbandet.

K LIST [enhet:] filnamn [.typ]

Kommandot används för lagring av programmet "filnamn.typ". Programmet lagras i textformat. Om inte filtypen anges, erhålls filtyp .BAS. Om det redan finns en fil med det

aktuella filnamnet och den aktuella filtypen kommer denna att skrivas över.

Exempel:

```
LIST SF1:ABC800
```

Programmet ABC800 lagras i textformat på skivan i SF1: Programmet tilldelas filtypen .BAS.

```
LIST ABC800.PRG
```

Programmet ABC800.PRG lagras i textformat på den skiva där detta först är möjligt.

```
LIST
```

LIST medför att innehållet i arbetsminnet visas på bildskärmen.

```
K UNSAVE [enhet:] filnamn [.typ]
```

Kommandot används för borttagning av "filnamn.typ". Om filtyp utelämnas, söker datorn först efter filnamn.BAC på samtliga skivor och sedan efter filen filnamn.BAS.

Exempel:

```
UNSAVE DR1:PROGRAM1.DEL
```

Filen PROGRAM1.DEL på skivan i DR1: raderas.

```
UNSAVE DR0:PROGRAM2
```

Filen PROGRAM2.BAC eller filen PROGRAM2.BAS på skivan i DR0: raderas.

```
UNSAVE PROGRAM3
```

Filen PROGRAM3.BAC eller filen PROGRAM3.BAS, raderas på den skiva där filen först påträffas.

```
IK KILL "[enhet:] filnamn.typ"
```

KILL kan användas i program (som instruktion) eller som kommando. I båda fallen raderas filen "filnamn.typ". Cita-tionstecken och filtyp måste anges.

OBS!
Filtyp kan vara mellanslag.

Exempel:

```
KILL "DRO:DATA.DAT"
```

Filen DATA.DAT på skivan i DRO:
raderas.

```
100 KILL "DATA.DAT"
```

Filen DATA.DAT raderas på
den skiva där filer först på-
träffas.

```
KILL A=
```

Filen vars namn finns i sträng-
en A= raderas.

```
K MERGE [enhet:] filnamn [-typ]
```

Kommandot används för länkning av programmet
"filnamn.typ", som måste vara lagrat i textformat på ski-
van, och programmet i arbetsminnet. Programmet
"filnamn.typ" laddas in i arbetsminnet utan att det i
arbetsminnet befintliga programmet raderas. Om arbetsmin-
net innehåller ett program med samma radnummer som åter-
finns i det nya programmet, används raden i det nya prog-
rammet.

OBS!
Endast program som sparats med LIST går att ladda in med
MERGE.

Exempel:

```
MERGE DR1:DELPROG.FIL
```

Programmet DELPROG.FIL överförs
från skivan i DR1: till arbets-
minnet.

```
MERGE DELPROG.BAS
```

Programmet DELPROG.BAS överförs
till arbetsminnet, från den
skiva där filen först påträff-
as.

```
MERGE DELPROG.FIL
```

Programmet DELPROG.FIL laddas
in i arbetsminnet.

```
MERGE DELPROG.NXT
```

Programmet DELPROG.NXT laddas
in i arbetsminnet utan att

programmet DELPROG.FIL raderas.
Där samma radnummer förekommer
i båda programmen, gäller
raderna från DELPROG.NXT.

IK CHAIN "[enhet:] Filnamn [.typ]"

CHAIN-instruktionen används för inläsning och start av programmet filnamn.typ. **CHAIN** har samma effekt som **RUN** filnamn men med den skillnaden att kan användas i program.

Exempel:

50 CHAIN "DR1:DELPROG.BAC"

Programmet DELPROG.BAC på skivan i DR1: överförs till minnet och därefter startar programmet.

200 CHAIN "DELPROG"

Programmet DELPROG.BAC eller DELPROG.BAS från den skiva där det först påträffas, överförs till minnet och därefter startas programmet.

100 CHAIN A=

Programmet vars namn finns i strängen A= överförs till minnet och startas.

Programexempel:

```
10 PRINT "NU GÅR PROGRAMMET *EJ NYTT*"
20 FOR I = 1 TO 10
30 PRINT I;",";
35 FOR J = 1 TO 500 : NEXT J : ! DELAY
40 NEXT I
50 CHAIN "NYTT"
60 END

10 PRINT "NU GÅR PROGRAMMET *NYTT*"
20 FOR I = 10 TO 1 STEP -1
30 PRINT I;",";
40 FOR J = 1 TO 500 : NEXT J : ! DELAY
50 NEXT I
60 CHAIN "EJ NYTT"
70 END
```

Spara ovanstående program med namnen "NYTT" respektive "EJNYTT". Programmen laddar in och startar varandra. Föl-

jande utskrift erhålls efter RUN EJNYTT:

```
NU GÅR PROGRAMMET *EJ NYTT*  
1,2,3,4,5,6,7,8,9,10,  
NU GÅR PROGRAMMET *NYTT*  
10,9,8,7,6,5,4,3,2,1,  
NU GÅR PROGRAMMET *EJ NYTT*  
1,2,3,4,5,6,7,8,9,10,  
NU GÅR PROGRAMMET *NYTT*  
10,9,8,7,6,5,4,3,2,1,  
NU GÅR PROGRAMMET *EJ NYTT*
```

Körningen avbryts genom att trycka två gånger på CTRL-C.
(CTRL hålls nedtryckt och C trycks ner).

```
IK NAME "[enhet:] filnamn1.typ" AS "filnamn2.typ"
```

NAME ändrar ett filnamn på skivan från filnamn1 till filnamn2. Citationstecken och filtyp måste anges.

Exempel:

```
NAME "DRO:FÖRE.BAC" AS "EFTER.BAC"
```

Namnet på filen FÖRE.BAC på skivan i DRO: ändras till EFTER.BAC.

```
IO NAME "GAMMAL.DAT" AS "NY.DAT"
```

Namnet på filen GAMMAL.DAT ändras till NY.DAT på den skiva där den först påträffas.

```
NAME A= AS B=
```

Filens namn finns i strängen A=, tilldelas namnet som finns i strängen B=.

```
IK PREPARE "[enhet:] filnamn.typ" AS FILE nr
```

PREPARE skapar en ny fil med namnet "filnamn.typ" på angivna enhet. Filen öppnas för skrivning samt tilldelas filnummer. Tillåtna filnummer är 0 till 255 (0 anger bildskärmen). Filnummer används för att referera till filen.

Filnamnet kan ersättas med en strängvariabel eller ett stränguttryck. Om strängen även innehåller filtyp, behöver inte filtyp anges separat. Om strängen inte innehåller filtyp, anges filtypen inom citationstecken. Om inget skrivs på filen före CLOSE reserveras upp till 4 sektorer beroende på kvarvarande utrymme.

Om **PREPARE** används på en redan befintlig fil, nollställs denna.

Exempel:

```
PREPARE "DR1:TEXT.DAT" AS FILE 1
```

En ny fil med filnamnet TEXT, filtypen DAT och filnumret 1 skapas på skivan i DR1:

```
PREPARE A= + ".BAS" AS FILE 2
```

En ny fil, som får filnamnet lika med innehållet i strängen A=, filtypen .BAS och filnumret 2, skapas på den första skiva där detta är möjligt. Om flexskivenheten inte är inkopplad, skapas filen på kassetbandet.

```
10 PREPARE "TEXT.DAT" AS FILE 3
```

En ny fil med filnamnet TEXT.DAT och filnumret 3, skapas på skivan i den första enhet där detta är möjligt.

```
20 PREPARE B= AS FILE 4
```

En ny fil som får filnamnet lika med innehållet i strängen B= och filnumret 4, skapas på skivan i den första enhet där detta är möjligt.

```
Ik OPEN [enhet:] filnamn.typ" AS FILE nr
```

OPEN öppnar en redan befintlig fil för läsning. Filnumret används för att referera till filen. Filnamnet kan utgöras av en strängvariabel. Tillåtna filnummer är 0 till 255 (0 anger bildskärmen). Samma fil kan öppnas med olika filnummer. Max sju filer får vara öppna samtidigt. Filtyp måste anges (i kommandot/instruktionen eller i strängen).

Exempel:

```
OPEN "DRO:PROG.FIL" AS FILE 5
```

Filen PROG.FIL på skivan i DRO: öppnas för läsning. PROG.FIL tilldelas filnummer 5.

```
OPEN C# AS FILE 6
```

Filen, som anges i strängen C#, tilldelas filnummer 6 och öppnas för läsning.

```
30 OPEN "DR1:DATA.DAT" AS FILE 22
```

Filen DATA.DAT på skivan i DR1:, tilldelas filnummer 22 och öppnas för läsning.

```
40 OPEN D# + ".DAT" AS FILE 7
```

Filen, som anges i strängen D#, och har filtypen .DAT, tilldelas filnumret 7 och öppnas för läsning.

För utökning av en befintlig sekvensiell fil används lämpligen en temporär fil:

10 OPEN "GAMMAL.DAT" AS FILE 1	Öppnar befintlig fil
20 PREPARE "NY.TMP" AS FILE 2	Skapar en temporär fil
30 ON ERROR GOTO 70	Felhantering vid filslut
40 INPUT LINE #1, A#	Läs post
50 PRINT #2, A# ;	Skriv post
60 CLOSE 40	Fortsätt till filslut
70 A# = "NY TEXT"	
80 PRINT #2, A#	Texten NY TEXT till temporär fil
90 CLOSE 1 : CLOSE 2	
100 KILL "GAMMAL.DAT"	Raderar gamla filen
110 NAME "NY.TMP" AS "GAMMAL.DAT"	Ny fil erhåller gamla filnamnet
120 END	

IK CLOSE [nr]

CLOSE stänger filen med angivet filnummer (nr), dvs avslutar läsning och skrivning i filen. Alla filer som öppnas med PREPARE eller OPEN måste stängas innan skivorna tas ur flexskivenheten. Om man är osäker på vilka filer som är öppna, kan man stänga samtliga filer genom att skriva CLOSE utan filnummer.

Exempel:

```
CLOSE 1
```

Filen med filnummer 1 stängs.

```
1000 CLOSE 22
```

Filen med filnummer 22 stängs.

Alla filer stängs.

IK **PRINT # nr "data"/variabel**

PRINT används för skrivning av data på angiven fil. Filnummer används som referens. **PRINT** följs av nummertecknet # och filnumret. Därefter skrivs data (text eller numeriskt uttryck) inom citationstecken, variabeln för aktuellt värde (heleller flyttal) eller strängvariabeln för aktuell sträng. Istället för att skriva **PRINT**, kan man skriva semikolon (;).

Exempel:

```
PRINT #2 "SKRIVTEST"
```

Texten SKRIVTEST lagras i fil 2

```
; #2 B%
```

Innehållet i strängen B% lagras i fil 2.

```
60 PRINT #4 "20.5"
```

Värdet 20,5 lagras i fil 4.

```
70 : #4, B  
80 : #4, C%  
90 : #4, D%
```

Värden på B och C% samt innehållet i strängen D% lagras i fil 4.

IK **PRINT #nr USING "formatsträng"; "data"/variabel [, "data"/variabel, ...]**

PRINT USING används för formaterad utskrift. "Formatsträng" innehåller formateringstecken, som bestämmer uppställning och format för de strängar och tal som ska skrivas. För utförlig information - se BASIC II-manualen.

```
IK INPUT #nr, variabel/strängvariabel
```

```
IK INPUTLINE #nr, strängvariabel
```

Instruktionerna används för läsning av data från sekvensiella filer. Instruktionen ska åtföljas av nummertecknet # och filnumret. Kommatecken behöver inte anges. Därefter skrivs variabeln (t ex A,B%) eller strängvariabeln (t ex C#) vilken ska tilldelas data. Vid läsning av en sträng, kommer en rad (t o m RETURN) att läsas in. Om strängen innehåller mer än 80 tecken, måste detta anges i en DIM-sats. Om INPUTLINE används - läses hela strängen in, dvs även RETURN och LINE FEED överförs.

OBS!

Vid INPUTLINE läggs ett RETURN och ett LINE FEED till den inlästa strängen.

Exempel:

```
20 INPUT #2, A
30 INPUT #2, B%
```

Variablerna A och B%, tilldelas var sitt värde från fil 2.

```
30 INPUT #2, E%
40 INPUT #2, F#
```

Heltalsvariabeln E% tilldelas det därpå följande värdet i fil 2 och strängvariabeln F# tilldelas innehållet till nästa RETURN.

Om man istället för rad 40 skriver:

```
40 INPUTLINE #2, F#
```

Innehållet i strängen F# blir detsamma som i filen. Dessutom tillkommer tecknen för vagnretur (RETURN) och ny rad (LINE FEED).

TIPS

Om man läser en sträng med INPUTLINE bör man ta bort tecknen för vagnretur och ny rad. Detta kan man göra på följande sätt:

```
INPUTLINE A#
A# = LEFT$(A#,LEN(A#)-2%)
```

K CLEAR

CLEAR-kommandot nollställer alla variabler och stänger alla filer. Detta kommando kan man t ex använda när man har glömt vilka filer som är öppna.

OBS!

Glöm inte att stänga alla filer innan skivorna tas ur flexskivenheten. I annat fall kan informationen förstöras.

K NEW

K SCR

NEW och **SCR** har samma funktion. De raderar programmet i datorns arbetsminne och stänger alla filer.

[K PUT #nr,"data"/strängvariabel]

PUT används för skrivning i en fil med direktåtkomst. Vid skrivningen mellanlagras data (text, numeriskt uttryck etc) i en buffert. Data placeras i filen där filpekaren pekar. Första tecknet i filen finns i position 0, andra tecknet finns i position 1 osv.

OBS!

Obegränsat antal tecken kan skrivas, dvs postlängden är (253 byte).

Exempel:

```
10 PUT #1,A=
```

Innehållet i strängen A= lagras i fil 1.

```
20 PUT #2,"KALLE"
```

Ordet KALLE lagras i fil 2.

[K GET #nr,strängvariabel [COUNTtkn]]

GET används för läsning av ett tecken från en fil med direktåtkomst. Tecknet i den position som filpekaren pekar på läses in. Filpekaren flyttas med **POSIT**. Obegränsat antal tecken kan läsas med **COUNT**. **COUNT** används tillsammans med **GET** för läsning av flera tecken från en fil med direktåtkomst. Läsning sker från den position som filpekaren pekar på och angivet antal positioner (tkn) framåt i filen. När sista tecknet i en post har lästs och ytterligare tecken ska läsas, läses automatiskt första tecknet i den därpå följande posten. Postgränserna har i detta fall ingen betydelse.

Exempel:

```
10 GET #3, A
```

Tecknet i den position som filpekaren pekar på, placeras i strängen A.

```
20 GET #3, A COUNT 32
```

32 tecken från och med tecknet i den position som filpekaren pekar på, hämtas från fil 3 och placeras i strängen A.

```
10 POSIT #nr, tkn
```

POSIT används för flyttning av filpekaren till angiven position räknat från filens början (position 0). POSIT kan användas tillsammans med samtliga filhanteringsinstruktioner. Instruktionen används för att peka ut ett speciellt tecken eller första tecknet i en teckenföljd, som ska läsas eller skrivas.

Exempel:

```
10 POSIT #1, N*S  
20 GET #1, A COUNT 5
```

N = postnummer
S = byte/post

Positionerar filpekaren till position $N * S$ på fil 1 samt läser in S stycken tecken.

Exempel:

Fil 1 innehåller följande:

Position	0	1	2	3	4	5	6	7	8	9
Innehåll	A	B	C	D	E	F	G	H	I	J

Följande program läser bokstäverna F, G och H samt visar bokstäverna på skärmen:

```
10 POSIT #1,5  
20 GET #1,A# COUNT3  
30 PRINT A#  
40 END
```

Filpekaren flyttas till position 5 (F)

Tecknen F,G och H läses (positionerna 5-7) och placeras i strängen A#.

Innehållet i strängen A#, dvs F,G och H visas på skärmen.

IK POSIT (nr)

POSIT (nr) används för läsning av filpekarens läge (position). Filnumret måste anges.

Exempel:

```
35 A = POSIT (1)
```

A tilldelas värdet som utgör filpekarens position.

5 Felmeddelanden

Fel 19- 68: I/O-fel
 Fel 120-129: ISAM fel
 Fel 130-176: Fel vid programkörning
 Fel 180-191: Logiska fel
 Fel 200-211: Allmänna fel
 Fel 220-234: Formella BASIC-fel

Fel (Error)	Meddelande	Kommentar
19	Kan ej öppna fler filer	Sju filer är öppnade
20	För lång rad (>160 tkn)	En rad får innehålla max 160 tecken
21	Hittar ej filen	Filen finns inte eller har sökts under fel namn
32	Filen ej öppnad	
34	Slut på filen	Försökt läsa efter filslut
35	Checksummafel vid läsning	Skivan eller kassettbandet är skadat
36	Checksummafel vid skrivning	Skivan är skadad
37	Felaktigt sektorformat	Fel på skiva eller kasset
38	Sektornummer utanför filen	Försökt att läsa längre än filen medger
39	Filen skrivskyddad	
40	Filen raderskyddad	
41	Skivan full	Filen får ej plats på skivan
42	Enheten ej klar	Ingen flexskiva isatt, luckan öppen eller skrivaren ej ansluten
43	Skivan skrivskyddad	
44	Logisk fil ej öppnad	
45	Fel logiskt filnummer	
46	Fel enhetsnummer	
47	Fel trapnummer	
48	Fel i biblioteket	
49	Felaktigt fysiskt filnummer	
51	Enheten upptagen	
52	Ej till denna enhet	
53	Funktionstangent	Funktionstangent har tryckts ner i INPUT- eller INPUTLINE-sats
54	IEC både sändare och mottagare	IEC-option
55	IEC-mottagare ej aktiv	IEC-option
56	IEC-sändare ej aktiv	IEC-option
57	Tecken från tangentbord ej i tid	
58	Ogiltigt tecken inläst	
64	Felaktigt "NAME"	Nya filnamnet existerar redan
68	Felaktig tidspecifikation	

Fel (Error)	Meddelande	Kommentar
120	Nyckel finns ej	ISAM option
121	Dubblettnyckel	ISAM option
122	Felaktig nyckel	ISAM option
123	Fel vid kontrolläsning	ISAM option
124	Index finns ej	ISAM option
125	Felaktig postlängd	ISAM option
126	Fel ISAM fil version	ISAM option
127	Reserverad kod	ISAM option
128	Slut på minnet i centralen	ISAM option
129	Reserverad kod	ISAM option
130	För stort flyttal	
131	Index utanför tillåtet område	Försök att använda index större än motsvarande DIM
132	För stort heltal	
133	Fel i ASCII-aritmetiskt uttryck	
134	Index utanför strängen	Index för stort eller negativt
135	Negativ "SPACE", "STRING" eller "TAB"<1	
136	För lång sträng	För liten dimension på den mottagande strängen
137	Ej tillåtet öka "DIM"	Ett fält får inte ökas utöver sin ursprungliga längd
138	Fel värde i "ON"-uttryck	
139	"RETURN" utan "GOSUB"	En RETURN-sats påträffad utan att föregående GOSUB-sats har blivit utförd
140	Felaktig "RETURN"-variabel	
141	Data slut	Datalistan har blivit tömd och en READ-sats efterfrågade fler data
142	Felaktigt argument i funktion	
143	Felaktig "SYS"-funktion	
144	Ej tillåten rad	
145	"FNEND" utan föregående "RETURN"	
146	"PRINT USING" fel	Felaktigt format i PRINT USING-sats
147	Felaktiga data	
148	För litet indata	För få data inmatade vid INPUT
149	"RESTORE" ej på en "DATA"-rad	
150	För mycket indata	För många data inmatade vid INPUT
151	"RESUME" utan fel	
176	Grafisk punkt utanför bildskärmen	

Fel (Error)	Meddelande	Kommentar
180	Hittar ej detta radnummer	Referens till ett radnummer som inte finns i programmet
181	Felaktigt in hopp i funktion	
182	"NEXT" eller "WEND" saknas	
183	"FOR" eller "WHILE" saknas	
184	Fel variabel efter "NEXT"	
185	Blandade "FOR"-loopar med samma variabel	
186	"FOR"-loop med lokal variabel ej tillåtet	Gäller i flerradiga funktioner
187	Funktion ej definierad	Anrop till ej definierad funktion
188	Flera funktioner med samma namn	
189	Felaktig funktion	Ej tillåtet att blanda flera "DEF"
190	Fel antal index	Antalet index överensstämmer ej med DIM
191	Ej tilldelningsbar i funktion	Funktionens argument är ej tilldelningsbar i funktion
200	Enheten ej ansluten	
201	Minnet fullt	Datorns primärminne har ej plats för program och data
202	"LIST"-skyddat program	
203	Fel programformat	Programmet är sparat under en ickekompatibel BASIC-version
204	"MERGE" går ej på "BAC"-fil	
205	"COMMON" fel	
206	Använd kommandot "RUN"	
207	Kan ej fortsätta	Gäller GOTO radnr och CON
208	Otillåtet som kommando	Instruktionen kan ej användas som kommando
209	Fel data till kommando	Felaktigt argument till kommandot t ex LIST##
210	Felaktigt tal	Talet innehåller tecken som inte är siffror
211	Precision får ej ändras	Ej tillåtet ändra precision efter tilldelning av variabler
220	Förstår ej	Formellt BASIC-fel
221	Otillåtet tecken efter satsen	Formellt BASIC-fel. Datorn förväntade RETURN, kolon (:) eller utropstecken (!)

Fel (Error)	Meddelande	Kommentar
222	Måste vara först på en rad	
223	Fel antal eller typ av argument	
224	Otillåten blandning av tal och strängar	
225	Ej enkel variabel	Ej tillåtet ha index på variabel t ex i FOR-loop
226	Felaktig sats efter "ON"	Formellt BASIC-fel
227	"," saknas	Formellt BASIC-fel
228	"=" saknas	Formellt BASIC-fel
229	")" saknas	Formellt BASIC-fel
230	"AS FILE" saknas	Förekommer i OPEN- och PREPARE-satser
231	"AS" saknas	Fel i NAME...AS...
232	"TO" saknas	Förekommer i FOR-loopar
233	Radnummer saknas	
234	Felaktig variabel	

6 Kommandosammanfattning

Nedan följer en sammanfattning av samtliga kommandon och instruktioner. Syntaxen följs av en kort beskrivning samt sidhänvisning till den utförliga beskrivningen. Beträffande syntaxens uppbyggnad, se avsnitt 4.1.

6.1 Kommandon och instruktioner under BASIC

K BYE

Sid 16

BYE medför övergång till DOS.

IK CHAIN "[enhet:]filnamn[.typ]"

Sid 54

Instruktionen används för laddning och start av programmet filnamn.typ.

K CLEAR

Sid 60

Kommandot nollställer alla variabler och stänger alla filer.

IK CLOSE [nr]

Sid 57

CLOSE stänger filen med angivet filnummer. Genom att enbart skriva CLOSE (utan filnummer) stängs samtliga filer.

IK GET #nr, strängvariabel [COUNT tkn]

Sid 60

GET används för läsning av ett tecken från en fil med direktåtkomst. Tecknet i den position som filpekaren pekar på läses in. Obegränsat antal tecken (tkn) kan läsas med COUNT.

Läsning sker från den position som filpekaren pekar på och angivet antal positioner (tkn) framåt i filen. Postgränserna har ingen betydelse vid läsningen.

IK INPUT #nr, variabel/strängvariabel

Sid 59

Instruktionen används för läsning av data från filen med angivet filnummer. En rad läses in.

IK INPUTLINE #nr, strängvariabel

Sid 59

Instruktionen används för läsning av data från filen med angivet filnummer. Hela strängen läses in, dvs även RETURN och LINE FEED (ny rad) överförs.

IK KILL "[enhet:]filnamn.typ"

Sid 52

KILL kan användas för radering av filen "filnamn.typ".

K LIST[enhet:] filnamn[.typ]

Sid 51

Kommandot används för lagring av filen "filnamn.typ". Om ingen filtyp anges, lagras filen med filtyp .BAS.

K LOAD[enhet:] filnamn[.typ]

Sid 49

Kommandot används för laddning av programmet "filnamn.typ". Arbetsminnet nollställs innan laddning sker.

K MERGE[enhet:] filnamn[.typ]

Sid 53

Kommandot används för läsning av filen "filnamn.typ". Om samma radnummer återfinns i arbetsminnet och i den nya filen, används raden i den nya filen. MERGE används för länkning av program. Endast program som sparats med LIST går att ladda in med MERGE.

IK NAME "[enhet:] filnamn1.typ" AS "filnamn2.typ"

Sid 55

NAME används för ändring av filnamnet från filnamn1 till filnamn2.

K NEW

Sid 60

Kommandot används för radering av innehållet i datorns arbetsminne och för stängning av samtliga filer.

IK OPEN"[enhet:] filnamn.typ" AS FILE nr

Sid 56

OPEN...AS FILE används för öppning av en befintlig fil. Filen öppnas för läsning. Tillåtna filnummer är 0 till 255. Filnummer 0 är reserverat för bildskärmen. Samma fil kan öppnas med olika filnummer. Max sju filer får vara öppna samtidigt.

IK POSIT #nr,tkn

Sid 61

POSIT används för flyttning av filpekaren till angiven position (tkn) räknat från filens början (position 0). POSIT (nr) kan användas för läsning av filpekarens läge.

IK PREPARE"[enhet:] filnamn.typ" AS FILE nr

Sid 55

PREPARE...AS FILE används för öppning av en ny fil. Filen öppnas för skrivning. Tillåtna filnummer är 0 till 255 (0 anger bildskärmen).

IK PRINT # nr "data"/variabel

Sid 58

PRINT används för skrivning av data på filen med angivet filnummer. Om strängen innehåller mer än 80 tecken, måste detta anges i en DIM-instruktion. PRINT kan bytas ut mot semikolon (;).

IK PRINT # nr USING "formatsträng"; "data"/variabel Sid 58

PRINT USING används för formaterad utskrift av data på filen med angivet filnummer.

IK PUT # nr, "data"/strängvariabel Sid 60

PUT används för skrivning i en fil med direktåtkomst. Data placeras i filen där filpekaren pekar. Obegränsat antal tecken kan skrivas på filen.

K RUN [enhet:] filnamn [.typ] Sid 50

Kommandot används för laddning och start av programmet "filnamn.typ".

K SAVE [enhet:] filnamn [.typ] Sid 51

Kommandot används för lagring av filen "filnamn.typ". Programmet lagras i internkodformat med filtyp .BAC, om ingen filtyp anges.

K SCR Sid 60

Kommandot används för radering av innehållet i datorns arbetsminne och för stängning av samtliga filer.

K UNSAVE [enhet:] filnamn [.typ] Sid 52

Kommandot används för radering av filen "filnamn.typ".

6.2 Kommandon under DOS

▯BAS Sid 17

▯BAS (eller RESET) medför återgång till BASIC.

COPY DRX:filnamn.typ, DRY:[kopians filnamn [.typ]] Sid 18

COPY används för kopiering av filen "filnamn.typ". Filen kopieras från skivan i DRX: till skivan i DRY:. Om kopians filtyp utelämnas, får kopian samma filtyp som originalet.

COPYLIB [,V]DRX:[filnamn.typ], DRY: Sid 19

COPYLIB används för kopiering av flera filer. Kopieringen utförs från DRX: till DRY: Om kopieringen ska åtföljas av verifiering anges V. Vid enbart kopiering anges COPYLIB DRX:...osv. Kommatecknet och V (,V) utelämnas.

DELETE DRX:[filnamn.typ] Sid 22

DELETE används för radering av flera filer på skivan i DRX:

DISKCHK

Sid 23

DISKCHK används för testning av skivor. Läsbarheten kontrolleras och filerna testas med avseende på formatfel.

DOSGEN [F]DRX:

Sid 26

DOSGEN används för formatering av nya skivor samt för radering och testning av gamla skivor.

F - flexskivan i DRX: formateras och testas för filhantering.

DOSGEN DRX: medför att informationen på skivan i DRX: raderas och skivan testas för filhantering.

ERRCOPY [DRX:]Felaktig.typ[DRY:]Hel.typ

Sid 28

ERRCOPY används för kopiering av filer som är behäftade med läsfel. Typ av fel (AM eller CRC) och sektornummer anges.

Den felaktiga sektorn kan hoppas över, behållas som den är eller fyllas med ett heltal mellan 0-255.

LIB

Sid 29

LIB används för presentation av innehållet på skivorna.

REPROT

Sid 30

REPROT används för skriv- och raderskydd av filer.

SYSTEM

Sid 17

SYSTEM används för presentation och val av ABS-program.

UFD

Sid 31

UFD används för skapande eller radering av UFD-bibliotek, ändring av namn på aktivt bibliotek. Används vid förflyttning mellan två UFD-nivåer.

10 Sakregister

Abort	13
ABS	5
ADDOPT	14
Autostart kopiering	16

BAC	5
BAS	5
BASICINI	13
Biblioteksprogram	29, 35
Buffertminne	3
BYE	16
Byglingar	8, 9

CAS:	6
CHAIN	54
CLEAR	60
CLOSE	57
Clusterstorlek	39
CMDINT	16
CON:	6
COPY	18
COPYLIB	19
COUNT	60

Datafil	3
DELETE	22
DEVDES	14
DIL-omkopplare	8
Direktåtkomst	4
DISKCHK	24
DISKCHK.ABS	23
DISKCHK.BAC	35
DOS	11
DOSGEN	26
DRO:	6
DR1:	6
DRINI	33

ERRCOPY	28
---------------	----

Felmeddelande	11, 13, 43, 63
Flexskiva	3
Fil	3
Filnamn	5
Filtyp	5, 11
Formatering	26

Generell kopieringsrutin	15
GET	60

HDO:	6
HD1:	6

Initiering	26
INPUT	59
INPUTLINE	59
Instruktion	49
Interpretator	13

KILL	52
Kommando	49
Kopiering	18, 19
Kopiering av felaktiga filer	28
Kortadress	39

LIB	2, 29
LIB.ABS	29
LIB.BAC	35
LIST	51
LOAD	49

Master File Directory	2
MERGE	53
MFD	2
MFO:	6
MF1:	6
Minneskort	14
MOO:	6
MO1:	6

NAME	55
NAMEDISK	34
NEW	60

OPEN	56
OPTROSH	14
OPTROSL	14

POSIT	61, 62
Post	3
PR:	6
PREABS	40
PREPARE	55
PRINT	58
PRINT USING	58
Programfil	4
PUT	60

Radering	22, 52
Raderskydd	19, 30
REL	5
REPROT.ABS	30
REPROT.BAC	37
RUN	43

SAVE	51
SCR	60
Sekvensiell åtkomst	4
SETCAL	43
SF0:	6
SF1:	6
Skivoperativsystem	11
Skrivskydd	19, 30
SOFTOPT	42
START	35
Styrkort	6
SYS	5
SYS.BAC	37
SYSDIR	13
SYSTEM	17
Systemskiva	11

TERMOPT	42
Tolk	13

UFD	1, 32
UFD.ABS	31
UFD.BAC	42
UFD-bibliotek	1, 31
UNSAVE	52
User File Directory	1

Återgång till BASIC	17
---------------------------	----

Övergång till DOS	16
-------------------------	----



Luxor Datorer AB · Box 923 · 591 29 Motala · Tel. 0141-162 00

REGIONKONTOR:

Stockholm, tel. 08-84 04 90 · Göteborg, tel. 031-42 07 20 · Malmö, tel. 040-18 10 20 · Örnköldsvik, tel. 0660-150 44

66 79206-04