

EN DATA BECKER BOG

SCHIEB · WEILER

AMSTRAD OG COMMODORE

**CP/M
ØVELSE**

INDHOLD TO BØGER (PRIS EEN BOG)

DANSK **NORSK**
udgave

Forlag:  , box 105, DK-6950 Ringkøbing

EN DATA BECKER BOG

BRÜCKMANN · ENGLISCH
· GERITS · STEIGERS

AMSTRAD


**464/664 &
6128**

INTERN

Simpelthen bibelen til 464/664 & 6128

DANSK / NORSK

udgave

Forlag: , box 105, DK-6950 Ringkøbing

CP/M ØVELSE

EN DATA BECKER BOG

SCHIEB · WEILER

**AMSTRAD OG
COMMODORE**



INDHOLD TO BØGER (PRIS EN BOG)

DANSK **NORSK**
udgave



NORDIC COMPUTER SOFTWARE
SMEDEGADE · POSTBOX 105 · DK-6950 RINGKØBING
1987

Copyright 1987 Data Becker & Nordic Computer Software
Postbox 105
Smedegade 7
DK-6950 Ringkøbing

Distribution i Norge:
Dataland Norge A/S
Postboks 532, Høyden
N-1501 Moss
Telefon: 032-65340

Sats & tryk: Tarm Bogtryk & Offset A/S

Dansk oversættelse & bearbejdning: NCS

ISBN 87-7283-012-3

Alle rettigheder til den danske version tilhører Nordic Computer Software.
Bogen må under ingen form (fotokopi, aftryk el. lign) reproduceres uden skriftlig tilladelse fra udgiveren. Ligeledes må bogen, eller dele heraf, ikke udbredes via elektroniske medier.

VIGTIGT

Programmer, programeksempler mv. er omfattet af lov om copyright. Disse må kun anvendes til personlige- eller undervisningsformål og må ikke anvendes kommercielt.

Alle programeksempler, tekniske anvisninger osv. i denne bog er omhyggeligt gennemgået for fejl. Trods dette kan der optræde fejl i reproduktionsfasen. Skulle nogle læsere opdage fejl, beder vi Dem rette skriftlig henvendelse til os, så vi har mulighed for at foretage rettelser.

Forord

Denne bog omfatter, som det fremgår af omslaget, CP/M til både Commodore- og Amstrad computere. For at gøre bogen så let læselig som muligt, er det almene stof trykt med bogstaver som dette afsnit.

Afsnit der alene taler om CP/M til Commodore er angivet som her.

Endelig er bogstavtypen den her anvendte, når det er afsnit vedrørende Amstrad alene.

Da vi kan trykke i større oplag ved at slå to bøger sammen, er det lykkedes at holde prisen på samme niveau som een bog ville koste, og du kære læser får en større viden gratis. Idet vi håber bogen falder i smag, ønskes god fornøjelse.

Forlæggeren.

Indholdsfortegnelse & Register

AFSNIT	SIDE
1. COMPUTEREN	9
1.1. Tastaturet	9
1.2. Billedskærmen	11
1.3. Printeren	12
1.3.1. Matrixprintere	12
1.3.2. Typehjulprintere	13
1.3.3. Ink jet-printere	13
1.3.4. Termoprintere	13
1.4. Eet eller nul	14
1.5. Hvordan man tæller binært	14
1.6. Lagring af værdier	15
1.7. Eksterne lagre	17
1.7.1. Disketten	17
1.7.2. Hard disk	19
1.8. Hvad du nu allerede ved	19
2. DRIFTSSYSTEMET	20
2.1. Hvad er et program?	20
2.2. Underprogrammer	21
2.3. CP/M's opgaver	22
2.4. CP/M og de forskellige versioner	23
2.5. CP/M stikordet	23
2.6. Sikker er sikker	27
2.7. Hvad du nu allerede ved	28
3. HVORDAN MAN ARBEJDER MED CP/M	29
3.1. Systemdisketten	29
3.2. Kopiering med een diskettestation	30
3.3. Kopiering med to stationer og CP/M 3.0 (Commodore)	33
3.4. Hvordan man ser på indholdsfortegnelsen	34
3.5. Kopiering med PIP (og to stationer, Amstrad)	35
3.6. Regler for filnavne	37
3.7. Filnavnudvidelser	37
3.8. Hvordan man genfinder en fil	38
3.9. Søgning med spørgsmålstegn (?)	38
3.10. Hvad du nu ved	39
4. INDBYGGEDE KOMMANDOER	40
4.1. Kommandoer, parametre og options	40
4.2. De indbyggede kommandoer	41
4.3. USER	42
4.3.1. USER-områder med CP/M 2.2	42
4.3.2. USER-områder med CP/M 3.0	43
4.4. DIR	44
4.4.1. DIR med parametre	45
4.4.2. Udvidet DIR med CP/M 3.0	46
4.4.3. DIR og dens options	47
4.4.4. DIRSYS	47
4.5. ERASE	48
4.5.1. Sletning med ERA ved CP/M 3.0	49
4.6. Ændringer af filnavne med REN(AME)	49
4.7. TYPE	50
4.8. Hvad du nu ved	51

5.	EXTERNE KOMMANDOER	52
5.1.	Statusangivelser med STAT (kun for Amstrad)	52
5.2.	Midlertidige kommandoer med CP/M 3.0	53
5.3.	SET	53
5.4.	Diskettestations-attributter	55
5.5.	Labels	55
5.6.	Passwords	56
5.7.	PROTECT	57
5.8.	Fil-PASSWORD	58
5.9.	TIME STAMP	59
5.10.	SETDEF	61
5.11.	SHOW	62
5.12.	SUBMIT	63
5.13.	HELP-kommandoen	66
5.14.	Hvad du nu ved	68
6.	ALT OM PIP	69
6.1.	Diskettekopiering	69
6.2.	Kopiering mellem brugerområder	71
6.3.	Tekst- og ikke-tekstfiler	72
6.4.	Sammenkopiering af filer	72
6.5.	Linienummerering	73
6.6.	Ændring af bogstaver	73
6.7.	Søgning af en streng	74
6.8.	Printning af flere filer efter hinanden	74
6.9.	Automatisk sikring af filer	75
6.10.	Overskrivning uden forespørgsel	76
6.11.	Kopiering af systemfiler	76
6.12.	"Rensning" af 8 bit	77
6.13.	Praktiske eksempler	77
6.14.	Hvad du nu ved	78
7.	CP/M INTERNT	80
7.1.	Fremstilling af en sikkerhedsdiskette med CP/M	80
7.2.	Amstrad diskettestationstyringens disketteformater	80
7.3.	Fremmede disketteformater	81
7.4.	CP/M-disketten	89
7.5.	Den medleverede assembler ASM	90
7.6.	Betjeningen af assembleren ASM (Amstrad) resp. MAC (Commodore)	91
7.7.	Hvordan man arbejder med SUBMIT og XSUB	100
7.8.	Lagertildeling	105
7.9.	SETUP-kommandoen (kun Amstrad)	111
7.a.	CP/M INTERNT C-128	114
7.1a.	Generelt om CP/M 3.0 på C-128	114
7.2a.	Systemdiskette for 1571-ejer	114
7.3a.	Den virtuelle station E:	115
7.4a.	COPYSYS på Commodore 128	115
7.5a.	Statuslinien	116
7.6a.	Disketteformaterne	117
7.7a.	Tastaturet	117
7.8a.	Specialfunktioner	118
7.8.1a.	Hvordan aktiveres specialfunktionsmåden	119
7.8.2a.	Hexcode-editoren	119
7.8.3a.	Streng-editoren	120
7.9a.	KEYFIG og hvordan man bruger den	123

8.	KORREKTION AF PIP (AMSTRAD)	126
8.1.	Fejlbeskrivelsen	126
8.2.	Fejlkorrektion	126
8.3.	Lagring af det nye PIP	127
8.a.	"ADDITIONAL UTILITIES" (COMMODORE)	128
8.1a.	Assemblerne MAC og RMAC	128
9.	ALLE CP/M-KOMMANDOER	130
9.1.	ASM (CP/M 2.2) (kun Amstrad)	130
9.2.	COPYSYS	130
9.3.	DATE (CP/M 3.0)	131
9.4.	DDT (kun Amstrad)	132
9.5.	DEVICE (CP/M 3.0)	133
9.6.	DIR (CP/M 2.2 og CP/M 3.0)	134
9.7.	DIRSYS (CP/M 3.0)	135
9.8.	ED (kun Commodore)	136
9.9.	ERA(SE)	137
9.10.	FORMAT (CP/M 2.2 og CP/M 3.0)	137
9.11.	GENCOM (CP/M 3.0)	138
9.12.	GET (CP/M 3.0)	138
9.13.	HELP (CP/M 3.0)	139
9.14.	HEXCOM (CP/M 3.0) (Commodore: Additional Utilities)	140
9.15.	INITDIR (CP/M 3.0)	141
9.16.	KEYFIG (kun Commodore)	141
9.17.	LIB (CP/M 3.0) (Commodore: Additional Utilities)	142
9.18.	LINK (CP/M 3.0) (Commodore: Additional Utilities)	142
9.19.	LOAD (CP/M 2.2 og CP/M 3.0) (kun Amstrad)	142
9.20.	MAC (CP/M 3.0) (Commodore: Additional Utilities)	143
9.21.	MOVCPM (CP/M 2.2) (kun Amstrad)	143
9.22.	PATCH (CP/M 3.0)	144
9.23.	PIP (CP/M 2.2 og CP/M 3.0)	144
9.24.	PUT (CP/M 3.0)	147
9.25.	REN(AME) (CP/M 2.2 og CP/M 3.0)	148
9.26.	RMAC (CP/M 3.0) (Commodore: Additional Utilities)	148
9.27.	SAVE (CP/M 2.2 og CP/M 3.0)	149
9.28.	SET (CP/M 3.0)	149
9.29.	SETDEF (CP/M 3.0)	150
9.30.	SHOW (CP/M 3.0)	150
9.31.	SID (CP/M 3.0) (Commodore: Additional Utilities)	151
9.32.	SUBMIT (CP/M 2.2 og CP/M 3.0)	151
9.33.	STAT (CP/M 2.2) (kun Amstrad)	152
9.34.	SYSGEN (CP/M 2.2) kun Amstrad	154
9.35.	TYPE (CP/M 2.2 og CP/M 3.0) (kun Amstrad)	155
9.36.	USER (CP/M 2.2 og CP/M 3.0)	156
9.37.	XREF (CP/M 3.0) (Commodore: Additional Utilities)	156
10.	FORSKELLE MELLEM AMSTRAD CPC-COMPUTERNE	157
	APPENDIX 1	159
	APPENDIX 2	162
	APPENDIX 3	163
	APPENDIX 4	165

1. Computeren

Jeg misunder dig, fordi du i hvert fald ved, hvad du ikke ved. Jeg kan kun forsøge at forestille mig, hvor meget du ved om computere og alt, hvad der har med dem at gøre. Da jeg ikke kan spørge hver enkelt, begyndes i dette kapitel så at sige med "Adam og Eva", for at vi alle kan have det samme udgangspunkt og derfor rigtig forstå hinanden. Den slags fremgangsmåder findes der overalt, tænk bare på Dansk Standards normer, eller prøv at skrive et program til din computer.

Men lad os begynde. Ved hjælp af en højvidenskabelig metode vil vi forklare, hvad en computer er.

Først stiller vi os dumme an og siger: "En computer er en kasse, hvori der foregår noget". Heldigvis antyder det engelske navn "computer", hvad der foregår. "To compute" kan oversættes til "at regne", dvs., at computeren er en slags regnemaskine. Sådan en taljonglør kan man med fordel anvende til at bearbejde store mængder af tal i bogholderiet eller behandle lange rækker af værdier fra videnskabelige eller tekniske målinger. Skoleelever og studenter kan sikkert også finde andre anvendelser for en sådan elektronisk regneslave.

1.1. Tastaturet

Men - der er et problem. Vi sidder med en kasse, som kan regne. Men hvordan fortæller vi den, hvad den skal regne ud? En mulighed kunne være at hviske den noget i øret. Men den måde at gøre sig forståelig på, for en computer, tilhører endnu science fiction-litteraturens verden, og vi kan ikke benytte den. Derfor beslutter vi os for, et tastatur og tilslutter sådan et bogstavs- og talindlæsningsapparat til kassen.

Da computeren imidlertid har meget mere "på sig" end en skrivemaskine, må tastaturet se lidt anderledes ud. Mellem 60 og over 100 taster findes på et computertastatur.

En Commodore 128 har 92 taster, og de forskellige funktionsblokke er skilt smukt fra hinanden. Som det hører sig til for en komfortabel computer, råder Commodore 128 over et, titalstastatur, f.eks. til indtastning af talkolonner. Den lille prik i midten af 5-tasten er ikke nogen fejl ved udstansningen, men er orientering for dine fingre ved blindskrift. Denne lille prik findes ligeledes på tasterne "F" og "J". Folk, der kan skrive blindskrift, ved hvorfor. På disse to taster skal pegefingrene ligge. Også på andre computere forekommer der hyppigt ved siden af de smukt ordnede bogstaver, tal og tegn til lige sådan et adskilt titalstastatur til hurtig indlæsning af talkolonner og, hvis man har en komfortabel computer, desuden et ekstra felt med såkaldte funktionstaster.

Om din computer har et dansk eller et engelsk tastatur, kan ses med et enkelt blik. Hvis der i næstøverste bogstavrække fra venstre står ASDFGHJKLÆØ, er du lykkelige ejer af en indlæsningsenhed efter dansk smag. Men hvis der kun står

ASDFGHJKL, vil du under visse omstændigheder få det lettere med mange computerprogrammer, fordi de hovedsagelig kommer fra U.S.A.. Du må så give afkald på de danske bogstaver ÆØÅ.

Commodore-ejerne er godt forvirrede nu, for de råder over begge "ord", også selvom bogstavet Æ kun er trykt med lysegråt på tastaturet. Ved hjælp af CAPS-LOCK-tasten (fjerde fra venstre i øverste række) kan du vælge, om du ønsker et amerikansk tastatur (ASCII-tastatur), så må tasten lades urørt, eller du foretrækker et dansk tastatur - så skal CAPS-LOCK-tasten være nedtrykket.

Alt efter dit valg ændrer tegnsættet sig på skærmen. For det danske tastatur gælder der, hvis det altså er til stede, de lysegrå bogstaver på din Commodores tastatur, så de danske bogstaver er der. Ganske vist kan man i CP/M-funktionsmåde kun arbejde med det amerikanske tastatur - af kompatibilitetsgrunde. Kun tegnsættet kan du vælge på denne måde, før du loader CP/M.

Hvad enten det er dansk eller engelsk, er der et par taster, der er meget vigtige for computeriet, og ofte er mærket på samme måde på begge typer tastatur. Fordi disse undtagelser er af betydning for resten af bogen og for alle programmer, skal du vide, hvilke taster, det drejer sig om.

Herunder vil vi nævne de mest anvendte betegnelser for disse taster. De betegnelser, der gælder for Commodore, er hver gang understreget.

VOGNRETUR

Du kender tasten fra den elektriske skrivemaskine, men anvender den på en helt anden måde på computeren. Forkortelserne på denne tast ser for det meste sådan ud:

RETURN (eng. for "vend tilbage")
ENTER (afslutning af en indlæsning)
CR ("carriage return" = eng. vognretur)
← (siger sig selv)

RETURTAST

BS (kommer af "back step")
← (cursor til højre/venstre eller cursor til venstre)

SLETTE- ELLER KORREKTURTAST

DELETE (eng. for "slette")
DEL (forkortelse for DELETE)
RUB OUT (eng. for "viske ud")

SKIFTETAST

SHIFT (skifter til store bogstaver)

SKIFTETASTLÅS

SHIFT LOCK (låser fast på store bogstaver)

Funktionen af den sidste tast afhænger af, hvordan dit tastatur er bygget. Nogle gange skifter den ikke hele tastaturet, men ændrer små til store bogstaver. Det kan være en betydelig lettelse af arbejdet.

På Commodore 128 virker SHIFT-LOCK-tasten på alle tasterne, og virkningen svarer til at holde SHIFT-tasten nede.

På nogle tastaturer findes, til skift af bogstaver, nogle særlige taster, som er mærket:

ALPHA LOCK eller

CAPS LOCK

Hvis du er meget heldig, angives tastaturets øjeblikkelige tilstand yderligere af en lysdiode. Desværre vises SHIFT-LOCK-tastens tilstand ikke på Commodore 128, til gengæld raster den ind, og det kan man godt mærke.

Særligt vigtig er tasten CONTROL, yderst til højre på dit Amstrad-tastatur og yderst til venstre på Commodore, som skal nedtrykkes ved mange styringskommandoer i forbindelse med en bogstavs- eller ciffertast. Den er i reglen mærket:

CTRL

CONTROL

eller

CTL (på Amstrad)

1.2. Billedskærmen

Egentlig skulle overskriften her, hvad angår Commodore, stå i flertal: billedskærmene, for man kan som bekendt tilslutte to skærme på een gang.

Eftersom vi nu har mulighed for at indlæse data i kassen, behøver vi også en metode til igen at få dataene tilbage. Det gavner ikke meget, hvis computeren udfører de mest fantastiske beregninger, men ikke kan meddele, hvad den har fået ud af det. Dataudlæsning over en billedskærm er særdeles miljøvenligt, fordi det går hurtigt og ikke laver megen støj. Computerens data og fejlmeldinger bliver hurtigt synlige, og hvis man vil skrive sine fejl ned, kan en printer hurtigt gå i aktion.

En billedskærm er i grunden et "ombygget" fjernsynsapparat, som fungerer efter samme princip. På "computer-fjernsynet", der kaldes en monitor, bliver der sjældent vist levende billeder, men bogstaver og tal og andre slags tegn. Derfor er kravene til en monitor andre end til et fjernsyn. Opløsningen bør være tydeligt bedre, for ikke at få tåge for øjnene, under de ofte timelange arbejdsperioder.

1.3. Printeren

I den første tid, hvor man havde store computere, med hvad vi i dag ville betragte som mikroskopiske ydeevner, tjente fjernskrivere som udlæsningsmedium for data i varig form. Computeren sender sine data til printeren og bruger der i reglen en stor bunke papir. En yderligere gene udover det store papirforbrug, er den lærrende måde, hvorpå printeren skriver. Men - endnu er der mange mennesker, der forlanger at få information i skriftlig form, og man kommer ikke udenom en printer. Fordi der findes så mange forskellige, giver vi her et kort overblik.

Priser og ydelser svinger indenfor rammer, der knapt kan overskues selv af specialister. Således koster en printer mellem 1500 og 40.000 kroner og byder til gengæld på skrivehastigheder på mellem 15 og 800 karakterer i sekundet.

1.3.1. Matrixprintere

Særdeles udbredt er de såkaldte matrixprintere. De har den største markedsandel og herfra kommer både de billigste og de hurtigste. Det hele begyndte med matrixprintere, der sirligt satte prik ved siden af prik og derved producerede en skrift, som så alt andet end sirlig ud. Det skyldtes skrivemetoden i de "tidlige" år (omkring 1980). Nåle, der kan styres hver for sig, trykker på et farvebånd og dette på papiret, så der her fremkommer en lille prik. Med syv prikker kan man danne et gendeligt bogstav eller tal. Det er blot besværligt at læse længere tekststykker.

Men - de tider er forbi. Bortset fra de helt billige, byder matrixprintere i dag på en skrivebredde på 80 tegn, omkring 100 tegn pr. sekund og et skriftbillede, der måske ikke er egnet til korrespondance, men absolut til interne meddelelser. Matrixprinteren har store fordele: den byder på et stort antal forskellige skrifter og også på rit indlæselige eller programmerbare tegnsæt, der kan slås til eller fra. På den måde bliver kommunikation med Grækenland eller Japan betydeligt lettere.

De bedste matrixprintere klarer hastigheder på indtil 800 tegn i sekundet med en skrivebredde på 132 tegn, men koster ganske vist mellem 30.000 og 40.000 kr. Relativt nye er matrixprintere af den moderne type. De har for det meste 24 nåle og kan dermed, ved nedsat hastighed, nå samme skriftkvalitet som en skrivemaskine. Almindelige værdier er ca. 70 tegn i sekundet til korrespondance og 140-150 tegn ved hurtigskrivning.

1.3.2. Typehjulsprinterne

En anden kategori af printere er typehjulsprinterne. Herunder medregnes skrivecylinder- og kuglehovedprinterne. Skrivecylinder og kuglehovedmaskinerne er for det meste baseret på tidligere skrivemaskinekonstruktioner og kræver ofte store serviceomkostninger. Typehjulsprinterne erstatter skrivecylinderen eller -kuglen med en rund skive, hvorpå de forskellige bogstaver og tegn er anbragt fjedrende. En lille hammer slår typen mod papiret, og det er ikke til at se forskel fra skrivemaskineskrift. Når korrespondancen altså skal se ud "som trykt", kommer en typehjulsprinter på tale.

Hastighed: på de billigste modeller for det meste knap 20 tegn pr. sekund, på topmodellerne 80 tegn.

1.3.3. Ink jet-printere

Betydeligt billigere er de helt nye ink jet-printere. Som en matrixprinter fremstiller de bogstaver og tegn af enkelte prikker. Derunder sker der ingen berøring mellem skrivehovede og papir, men små bitte farvedråber skydes fast på papiret. Skriftbilledets kvalitet er sammenlignelig med en simpel matrixprinters. Specielle fordele ved ink jet-printeren: meget ringe støj med omkring 45 dbA (decibel A) og alligevel 150 tegn i sekundet. En mangel: man kan kun få kopier ved at printe flere gange.

1.3.4. Termoprintere

Den samme mangel er der ved termoprintere. De frembringer tegnene ved hjælp af varmpåvirkning af en papiroverflade. Hertil benyttes et særligt papir, der med tiden bliver gult. Til gengæld er printerne, der også arbejder efter prikmetoden, ekstremt billige og alligevel helt igennem anvendelige til dokumentationsformål. I lommeregnerne under 400-kroners grænsen er de simple termoprintere allerede taget i brug.

Det optimale til hurtig og smuk skrift via en personlig computer er laserprinterne. Deres output er ikke til at skelne fra et professionelt sat og trykt arbejde.

I den gode hensigt at gøre det vanskeligt at komme til at tænde eller slukke ved en fejl, har mange computerfabrikanter fået den geniale ide at gemme afbryderknappen et eller andet sted. Hvis du ikke har fundet afbryderen på din computer, så led et eller andet sted, hvor ingen djævel ville tro den kunne være. Der er den sikkert.

DATALAGER

Vi kan nu lave temmelig meget med vor teoretiske computer. Selve computeren er der, tastaturet, en billedskærm og en printer har vi også. Hvad der mangler, er et sted, hvor dataene kan lagres. Indeni computeren og - i længere perioder - også udenfor. Principielt kan man sige, at måden at lagre data på, er den samme i ethvert

af de medier, computeren anvender, hvad enten det er i arbejdslageret, som netop nu opbevarer min her skrevne tekst, eller på en diskette eller en hard-disk. Hvordan og hvorledes computeren kan lagre noget, skal du nu få at vide.

Da det er en smule teoretisk, er det måske bedst, at du henter en kop kaffe, tager en god slurk og så læser opmærksomt videre. Jeg venter, til du er her igen...

1.4. Eet eller nul

Du vil vel ikke benægte, at en computer er et stykke elektrisk apparatur. Hvis det er tilfældet, så må også informationerne inden i computeren forarbejdes i elektrisk form. Men hvordan klarer vi det? Hvordan skal vi fortælle en computer, hvad vi mener? Dertil kræves en form for fremstilling, som computeren kan forstå. Der findes en formel, som computeren kender: der løber en strøm - der løber ingen strøm. Efter det samme princip fungerer også en elektrisk pære. Løber en strøm, så begynder den straks at lyse. Er der ikke længere nogen strøm, så går den simpelt hen ud. Hvad sådan en dum elektrisk pære kan, kan vor computer selvfølgelig også. Dens konstruktører har indskrevet i dens elektronhjerne, at tilstanden "der løber strøm" er ensbetydende med værdien "eet", ligesom tegnet "1" også har en bestemt værdi for os.

Løber der ingen strøm, betyder det for comp'en "0". Dermed kan den nu skelne mellem to tilstande, hvad der fører til forbavsende resultater

1.5. Hvordan man tæller binært

Til daglig brug er den binære måde at tælle på, altså med eet og nul, ikke særlig egnet. Før du får sagt, hvornår du er født, er du allerede blevet tør i halsen.

Lad os se på, hvordan man arbejder med tal. Der findes forskellige talsystemer: det binære, det decimale, det hexadecimale og flere endnu. Decimalsystemet behersker vi alle. Vi kan knapt nok tænke på anden måde end i titalsværdier. Men hvordan gør vi det? Vi tænker os et tal, f.eks. nul. Skal det være mere, tager vi ettallet og så fremdeles. Er vi kommet til "9", sammensætter vi det næste tal af de to laveste, hvad der giver et "10".

På samme måde kan vi gå frem, når vi kun har to cifre til rådighed. Den første værdi er nullet. Den næste er ettallet. Så er det slut, vi har ikke flere tal at tælle videre med. Altså sammensætter vi det næste tal af de to og får således for værdien to cifrene 10. Tretallet ser sådan ud: 11, og til firtallet behøver vi allerede en ny plads, hvad der giver 100.

For forståelsens skyld er der her endnu en gang en tabel over de binære tal:

nul = 0
een = 1
to = 10
tre = 11
fire = 100
fem = 101
seks = 110
syv = 111
otte = 1000

og således videre lige ud ad landevejen.

Ser man på pladserne i dette talsystem, så kan man med een plads fremstille to værdier, med to pladser fire værdier, med tre pladser otte, med fire pladser seksten og så fremdeles. Fra plads til plads bliver der altid det dobbelte antal muligheder. I vort decimalsystem kender vi jo også pladsernes værdier; blot drejer det sig her altid om tiere. Een plads kan fremstille ti værdier, to pladser hundrede, tre pladser tusinde...

Historien med de to-taller kaldes det binære (totals-) talsystem, og har behaget de folk, der ikke alene har givet os smag for hamburgeren, men også for computeren.

En plads i et tal hedder på amerikansk "digit" og binær kalder super-nationen "binary". Begge dele tilsammen hedder "binary digit" og blev forkortet til "bit".

Nu er imidlertid en eneste plads i et tal temmelig lidt at operere med i et lager, og moderne computere har en hel mængde lagerplads. Holder man sig til een bit, bliver talbetegnelserne for lagerpladserne enormt store. Derfor har man skabt enheder eller "ord", som med eet ord eller tal beskriver en hel række bits. Med et otte-bit ord kan man således beskrive 256 bits og med et seksten-bit ord 65536.

Særdeles hyppigt forekommer enheden med 256 muligheder, hvorfor denne enhed bærer det amerikanske navn "byte".

1.6. Lagring af værdier

For at kunne arbejde rigtigt må computeren kunne lagre tallene. Et tal består, som vi har set, for computeren af en elektrisk tilstand: strøm eller ikke-strøm. Sammenfatter man otte sådanne lagerpladser, får man et otte-bit ord, der kan fremstille ialt 256 værdier. "Tilfældigvis" svarer det nøjagtigt til en byte. Fordi en byte ikke er særlig meget, forekommer også betegnelsen kilobyte. Her svarer et kilo ganske vist ikke nøjagtigt til 1000 bits, men i overensstemmelse med den binære tælle måde til 1024 bits. Til helt store enheder findes også en megabyte.

Ved hjælp af elektroniske omstillinger, er det muligt at læse værdierne ud af en byte igen, bearbejde dem videre og også at indlæse nye værdier. Det er hele computerens hemmelighed - selv om denne hemmelighed også kan spare os for meget

arbejde. Dens fordel er, at den kan foretage disse operationer med meget korte tidsmellemlum og derfor er en ægte og nyttig databearbejder.

Fordi hver lagerplads virkelig må være til stede, har hvert computerlager kun en begrænset plads, hvori data kan deponeres. En almindelige størrelse for arbejds-lageret i en personlig computer, der har en otte-bits processor, dvs. dens egentlige hjerne, er 64 kilobytes eller kort 64K. De nye seksten-bits computere byder på lagerstørrelser mellem 128K og omkring otte megabytes.

Commodore 128 er ganske vist også en otte-bits computer, men byder på 128k bytes - så meget er jo også nødvendigt for CP/M 3.0. Ganske vist kan man udbygge den til 1 megabyte, dette er i alt fald forudset i driftssystemet.

Men det er dog stadig noget uklart, hvordan man f.eks. kan lagre tekst i en computer, hvor det dog kun er tal, der sendes frem og tilbage.

På det punkt er computere helt menneskelige. De benytter, nøjagtig som vi når vi oversætter et fremmedsprog, en ordbog. Indeni computeren er der en tabel, hvori er fastlagt en talværdi for hvert bogstav og tegn. Kommer et bogstav ind i computeren, kigger den efter i tabellen og finder den rigtige binære talværdi. Den kan så bearbejdes videre. Ved udlæsningen fungerer det hele lige omvendt.

For at kunne virke hensigtsmæssigt ikke blot på en computer af et bestemt fabrikat, må man enes om en bestemt "oversættelsestabel". Desværre, set fra et dansk synspunkt, skete denne overenskomst i Amerika og omfatter kun de tegn, der er almindelige der. Det hele kaldes ASCII-koden (American Standard Code for Information Interchange = amerikansk standardkode for informationsudveksling) og indeholder blot ikke de specielle danske bogstaver. Denne ASCII-kode (udtales "aski") omfatter 128 tegn, der hver har sin egen værdi. I appendikset finder du en tabel med alle disse værdier.

Ganske vist har man nu også taget højde for danske og andre udenlandske specielle bogstaver, men anbragt dem på en plads langt "bagude". På den måde får de amerikanske producenter af apparaterne mindst besvær af det, og har alligevel et godt argument for europæiske købere.

Men hvad laver computeren med et helt lager fuldt af værdier? Ingenting. Den er for dum til at begynde på noget. Man må give den en passende kommando. Du kan befale den at forsyne teksten, i sit lager, med en lige højre margin eller at tænde for lyset. Først når du giver hensigtsmæssige kommandoer, kan computeren udrette noget. På det laveste niveau, ser det sådan ud: du siger til computeren: "Tag bit'en i lager "a", adder den til bit'en i lager "b" og læg resultatet i lager "c".

Der findes en hel række af sådanne anvisninger i computerens "Central Processing Unit" (CPU), på dansk kaldet processoren, der er computerens regnehjerne. Ud af mange af disse operationer lader der sig sammensætte komplicerede processer, hvad der er maskinsprogs- eller assemblerprogrammeringens opgave. Det er programmeringssprog, der styrer maskinfunktionerne direkte. Basic, Pascal eller Fortran styrer processoren efter en intern oversættelse af det pågældende program til maskinsprog.

For at løse en bestemt opgave, må hvert enkelt skridt beskrives nøjagtigt for computeren. Det sker ved hjælp af et program, som er en række af kommandoer, der skal udføres efter hinanden. Programmet må forefindes i lageret, for at det kan blive udført; det tager altså også lagerplads op.

Et program forbruger mellem 10 og 120K bytes. Til tider er pladsen i lageret allerede så optaget af programmet, at der ikke er plads tilovers til udføringen og brugerens data. Det betyder så: dette program er for stort til den pågældende computer og kan ikke anvendes i den. Men sådanne tilfælde er sjældne. Hyppigere forekommer det, at brugerens data ikke længere kan være i det indbyggede arbejdslager.

1.7. Eksterne lagre

I de sjældneste tilfælde er det nødvendigt at have et helt program på en gang i lageret. Normalt er det tilstrækkeligt, at de vigtigste dele findes i lageret og resten kun bliver indlæst efter behov. Et program, der arbejder på denne måde, siges at benytte overlægningsteknik. Af den grund findes der i computere, foruden det interne arbejdslager, også eksterne lagringsmuligheder, der ikke er omfattet af så store pladsbegrænsninger og desuden er billigere. Disse såkaldte masselagre kan være: en kassettebåndoptager, en disktestation, eller en hard-disk. Alle tre har deres fordele og ulemper, men de giver alle rigelig plads til data, og kan udveksle disse med computeren på en acceptabel tid.

1.7.1. Disketten

Disketten, der også kaldes en floppy-disk eller bare disk, er en art grammofonplade af samme materiale som et kassettebånd. En tynd skive med magnetiserbar overflade, indhyllet i et beskyttelseshylster mod brugerens snavsede fingre eller støv- og smudspartikler. Langs overfladen af disketten, når den er inde i computeren, kører der et skrive/læsehovede og gør forskellige steder magnetiske.

Der har vi igen to forskellige tilstande: magnetisk eller ikke-magnetisk. Det svarer til lagringsmetoden inde i computeren, og det er også tilsigtet. Dataene i computeren, som foreligger som elektriske tilstande, kan på den måde overføres til disketten. Har en bit tilstanden "strøm", så gør diskettens skrivehovede et lille punkt på overfladen af disketten magnetisk. Har den næste bit tilstanden "ingen strøm", kommer der heller ikke noget på disketten. Ud af rækkefølgen af magnetiske og ikke-magnetiske punkter, kan et program eller en række data læses og igen overføres til computeren. Dermed kan vi lagre data eksternt.

Disketten roterer med 200 til 300 omdrejninger pr. minut og ville efterlade sig et frygteligt datakaos, hvis man blot skrev dataene helt planløst på den. Derfor inddeler man disketten i en række spor, der ligger ved siden af hinanden på disketten, ligesom kørebanerne på en flersporet motorvej.

Vil man aflevere data, så kører skrivehovedet henover det tilsvarende spor og skriver hurtigt løs på det. Et sådant spor er ret langt, og det ville være nødvendigt at vente et pænt stykke tid, før man igen fik læst sine data fra et spor og ind i computeren igen.

For at forkorte denne tid, inddeler man disketten endnu en gang i sektorer (alt efter fabrikat) på fra 128 til 1024 bytes, Amstrad CPC benytter 512 bytes.

Commodore 128 i CP/M-funktionsmåde bruger flere forskellige formater, fordi den er kompatibel med flere. Hvilke det er, omtales senere, så det kommer til at se ud som stykker af en lagkage.

Denne metode at opnotere på er ganske praktisk, men forhindrer blandt andet, at man uden videre kan stikke en diskette fra een computer ind i disktestationen til en anden computer og læse dataene. Hvert firma har sin egen metode for at ærgre os brugere. Store specialister på dette felt er firmaerne Apple og Viktor.

Når du køber nye, uformaterede disketter, må du først formattere dem. Til det formål har hver computer et specielt program, som nøjagtigt nedskriver sporene og sektorerne på den diskette, som computeren benytter.

Hvis du gerne vil ærgre dig lidt, skulle du tage at formattere disketten med dine vigtigste data på. Du vil blive forbavset over, hvor lidt der bliver tilbage af dem. Pas derfor på, og anbring en skrivebeskyttelse over hakket i disketten. De små klæbemærker, der ligger i enhver pakke disketter, kan redde det hele for dig.

Ved tretommers disketter, som Amstrad bruger dem, skubber man en lille plasticskyder, hvad der gør det umuligt for computeren at skrive på denne diskette.

Ligeså klogt er det at tage en sikkerhedskopi af alle vigtige data, og opbevare den på et sted, der er utilgængeligt for børn, hunde, katte, ildebrand, indbrudstyve, jordskælv og lignende ulemper. Selv om du ikke lægger dine diamanter i pengeskabet, skulle du tage at gøre det med dine sikkerhedskopier. Hvordan man ved hjælp af CP/M kan lave sikkerheds- og andre kopier, erfarer du i et følgende kapitel.

For tiden er der forskellige disketteformater på markedet. Tildels forekommer endnu den gamle og ærværdige ottetommers diskette, som engang indledte den elektroniske databehandlings sejrsgang. Særlig udbredt er 5¼-tommers disketten, der normalt anvendes i personlige computere. Dette format er nogenlunde håndterbart, og i mellemtiden passer også næsten to megabytes på sådan en lille plade.

Specielt fra Japan trænger imidlertid nu også 3-tommers og 3½-tommers disketter frem på markedet, som giver løfte om en forbedret beskyttelse, fordi skrive/læsespalten er dækket af en metalklap, så længe disketten ikke er i brug. Desuden har 3'er disketterne et fast plasticomslag, som man kun kan bøjge med vold.

1.7.2. Hard disk

I computere, der hovedsageligt bliver benyttet professionelt, finder man tillige en hard-disk eller fastpladelager. Disse tingester er på størrelse med en 5¼-tomme diskette og fylder også ligeså meget i computerens kabinet (fordi formindskelsesbølgen ikke er til at standse, findes der også allerede 3½-tommers fastpladestationer). Indeni roterer der metalplader med magnetiserbar belægning med omkring 3000 omdrejninger i minuttet, og skrive/læsehovederne flyver i allerlaveste højde henover pladens overflade. Et støvkorn eller en røgpartikel på denne overflade ville have den samme virkning som Matterhorn på en jumbojet, der fløj for lavt.

Derfor leveres hard-disks i et lufttæt aflukket hylster, og der bør man lade dem blive. Som standardkapacitet gælder siden begyndelsen af 80'erne 10M-byte på en hard-disk. Men priserne synker, og 20M-byte pladen holder allerede sit indtog i computerkabinetterne.

Udover den særligt høje kapacitet byder hard-disks også på en afgørende fordel: de er indtil 20 gange hurtigere i tilgang end disketter.

Nå, hvis jeg har talt rigtigt, skulle nu alle en computers bestanddele være omtalt, og du skulle have fået det grundlæggende kendskab til dette meget hjælpsomme, og sommetider også morsomme apparat. I det næste kapitel skal du lære noget om driftssystemet, og hvorfor man overhovedet har brug for sådan et.

1.8. Hvad du nu allerede ved

- * Du kender en computers bestanddele fra tastatur til billedskærm.
- * Du kender de forskellige printertyper, deres fordele og ulemper.
- * Du ved også, hvad et lagringsmedium er.
- * Du kender tallene eet og nul, kan tælle binært og ved, hvordan værdierne lagres.

2. Driftssystemet

I det foregående kapitel har du fået et kort indblik i computerens "hardware", de enkelte dele, den består af. Hvis du ser på sådan en computers materialeværdi, er tingesten ikke særlig meget værd. Grunden: du kan stadig ikke rigtig gøre noget praktisk med den. Rigtig værdifuld bliver en computer først, når du kan lade programmer køre på den, som er vigtige og nyttige for dig. Dertil behøver du det såkaldte "software". Her må vi også skelne mellem forskellige begreber.

Vi beskæftiger os med to typer software: driftssystemet og programmerne. Du skal hele tiden være klar over, at driftssystemet også er et program. Den forskel, jeg nævnte, kommer af de forskellige slags softwares funktion, og jeg skal forklare det nærmere.

Hyppigt hører jeg, at uerfarne mennesker med interesse for computere spørger: "Kan den også tekstbehandle?" eller noget i den retning. Disse mennesker har endnu ikke forstået forskellen mellem et program og et driftssystem, hvorfor der her følger en forklaring, for det tilfælde, at du også skulle have spurgt sådan en gang.

Forestil dig, at det slet ikke drejer sig om en computer, men om den ganske almindelige dagligdag. Her er et stykke smørrebrød. Det består af en skive brød og noget pålæg, der skal smage så godt som muligt. Sådan et stykke smørrebrød er det helt rigtige til at belyse forskellen mellem et driftssystem og et program.

Hvad man altid har brug for, er en skive brød. Overført til computeren svarer det til computeren selv. Fordi smørrebrød nu engang hedder smørrebrød, behøver du ubetinget også noget smør på. Overført på computeren: hvis du har en computer og vil lave noget med den, behøver du et driftssystem. Hvad du lægger på dit smørrebrød, retter sig efter din smag eller hvor sulten du er. Ligesådan med computeren. Hvilket program du anvender for at spille et spil på skærmen eller bearbejde tekster, eller gøre noget andet, afhænger af dit eget personlige behov og smag.

Nu er du forhåbentlig klar over, at spørgsmålet ovenfor var stillet forkert. Hvis computeren er til stede, og et driftssystem forefindes, så kører ethvert program, som er skrevet for denne type computer med dette driftssystem.

2.1. Hvad er et program?

Principielt består ethvert program af flere efter hinanden skrevne kommandoer, der bliver udført af computeren i samme rækkefølge. Computeren behøver befalinger i et sprog, som den kan forstå. Fordi den selv er en maskine, kaldes dettes sprog "maskinsprog". Programmer, der er skrevet i maskinsprog, er uforståelige for "normale" mennesker; men computeren glæder sig. Fordi dette sprog ligesom er skrevet inde på livet af computeren, udfører den rækkefølgen af kommandoer særligt hurtigt.

For "normale" mennesker findes der, til at lette arbejdet, også de såkaldte højere sprog eller programmeringssprog. Du kender sikkert nogle af dem. Særlig kendt er BASIC, men også COBOL eller FORTRAN eller PASCAL eller også det helt nye MODULA 2. I disse højere programmeringssprog anvendes ord, der er forståelige for mennesker som kommandoer, der oversættes indeni computeren til maskinsprog og udføres af computeren som sådan.

Et program er simpelthen en række kommandoer, der nøje fortæller computeren, skridt for skridt, hvad den skal gøre. Længden af et program kan være mellem to kommandolinier og næsten uendeligt mange kommandolinier. Det kommer blot an på lagerkapaciteten i computerens arbejdslager og de eksterne lagre.

Nu er det imidlertid ikke nok, at et program opfylder brugerens krav, f.eks. muliggør tekstbehandling. Det skal også internt, indeni computeren, varetage en række andre opgaver, for at det hele skal kunne lade sig gøre. For eksempel skal det fastslå, hvilken tast på tastaturet, der lige er blevet trykket på, hvilket tegn, der skal fremstilles på billedskærmen eller på printeren, om og hvor på det eksterne lager de netop nødvendige data eller programmer forefindes, eller lade data fra disketten ind i computerens arbejdslager.

Envidere skal det sikres, at de fra disketten læste data kommer til det rigtige sted i arbejdslagret, og at der er tilstrækkelig plads på disketten til nye optegnelser. Desuden skal der føres en indholdsfortegnelse over disketten osv. osv.

2.2. Underprogrammer

Som du ser, er der en hel del arbejde, sådan nærmest ved siden af. Og dette "ved siden af" arbejde skal hver gang programmeres på ny, for hvert enkelt program. For hver del af et program må det fastlægges, hvad der skal ske i computeren. Yderligere besværliggøres det af, at disse opgaver fra computertype til computertype hver gang skal løses på en anden måde. Egentlig er enhver computer en lille smule forskellig fra alle andre. Ville man nu ligesom skrive et længere program i eet stykke, skulle det omskrives til hver enkelt computertype for at køre fornuftigt.

Dette unødvendige arbejde kan man spare sig, når man opdeler programmet i et hovedprogram og flere underprogrammer. Underprogrammerne bliver så kun ladet ind i computerens arbejdslager, når de virkelig behøves.

For at tilpasse programmet til en anden type computer, tager man den ene blok underprogram væk, skriver en ny, og straks kører hele programmet på den nye computer uden begrænsninger. En videre fordel ved denne underprogram-teknik er, at hovedprogrammet slet ikke længere ved, hvordan en bestemt delopgave bliver løst. Det udsteder simpelthen en befaling, f.eks. "tekstudlæsning", og underprogrammet klarer dette arbejde. Om teksten udlæses på en monitor, en printer eller via en fjernskriver, kan være fuldstændigt ligegyldigt for hovedprogrammet. Det udsteder blot befalingen, og arbejdet bliver udført.

Metoden virker, når overførelsen af data til underprogrammet er standardiseret i hovedprogrammet. Det kan man forestille sig som et stafetløb, hvor den næste løber altid står på det nøje forudbestemte sted og overtager staven. I computersprog hedder et sådan normeret og nøje fastlagt overtagelsessted et skæringspunkt. Det udtryk skal forstås helt billedligt, fordi man simpelthen kan skære et underprogram af og sætte et nyt på, uden at hovedprogrammets funktion lider derunder, eller i det hele taget bemærker denne forandring. Alt passer igen nøjagtigt sammen og fungerer gnidningsfrit.

Det er jo egentlig dumt, at programmøren i hvert eneste program på ny skal medtage de interne operationer i sit program. Det giver meget arbejde, koster en bunke tid og penge. Denne ide fik mændene fra DIGITAL RESEARCH også og gik i gang med at skrive et standardprogram for mikrocomputere. De sammenfattede de hyppigste underprogrammer, der er nødvendige for den interne styring, definerede skæringspunkterne, og måden hvorpå data skulle overføres mellem hoved- og driftsprogram, og fik på den måde skabt et fælles driftssystem (også kaldt operativsystem) for de forskellige typer computere.

Indsættes kan dette driftssystem naturligvis kun i computere med processorer, der er identiske eller ligner hinanden meget, fordi det i den sidste ende er programmets maskinsprogkommandoer, der skal forstås og udføres. Det driftssystem, der som det første standardmæssigt stillede de mange interne arbejdsoperationer til rådighed kaldes CP/M. Denne forkortelse står for "Control Program for Micro-Processors", hvad der på dansk ganske simpelt betyder styringsprogram for mikroprocessorer. Dette driftssystem arbejder med microprocessorer af type 8080, 8085 eller Z80.

2.3. CP/M's opgaver

I det væsentlige klarer CP/M følgende grundopgaver: indlæsning af tegn, udlæsning af tegn, forvaltning af den ledige lagerplads på eksterne lagre, læsning af disketteoptegnelser og skrivning af nye optegnelser på diskette, eller andet masselager. Disse tjenesterutiner kan benyttes af alle anvendelsesprogrammer på en ensartet og standardiseret måde.

CP/M må imidlertid gennemføre to forskellige slags arbejder, hvorfor den er opdelt i to store blokke. Den første del BDOS (Basic Disc Operating System = grundlæggende driftssystem for disketter) tager sig af alle opgaver, som uafhængigt af det pågældende computersystem altid skal løses på samme måde. Den anden del af driftssystemet, det såkaldte BIOS (Basic I/O System = grundlæggende system for ind- og udlæsning) indeholder alle de programdele, der er tilpasset til, og nødvendige for et bestemt computersystem.

Enhver computermodel har sine egne metoder til løsningen af bestemte opgaver, og man kan ikke simpelthen overtage et driftssystem fra det ene apparat, og lade det køre på et anderledes. Før sådan et overtaget driftssystem fungerer, må

BIOS'en tilpasses til den nye maskine. Normalt skulle du ikke have nogen problemer med det på din computer, for enhver computer bliver leveret med et tilpasset og passende driftssystem (forhåbentlig da).

2.4. CP/M og de forskellige versioner

Ethvert program, der kan klare sig et stykke tid på computermarkedet, gennemløber også forbedringer og kommer derefter i en forbedret version. Der findes nu engang ingen fejlfrie programmer, og nogle af disse fejl bliver først opdaget, når et program har fundet mangesidig anvendelse, og virkelig alle muligheder bliver afprøvet.

Ankommer der tilstrækkelig mange meddelelser om fejl til programmets leverandør, udbedrer han fejlene, og beslutter sig på et eller andet tidspunkt til at bringe en forbedret version. Som næsten fejlfrit kan man betragte CP/M i version 2.2, som praktisk talt er standarddriftssystemet for alle otte-bits computere. Den nyere version, CP/M 3,0 var altså ingen nyudgave for at rette gamle fejl, men fik en betydelig forbedrelse af sin ydelse, og modtog tillige nye kommandoer. Desuden måtte CP/M tilpasses til de stadigt mere ydelsesdygtige computere og computerteknikker.

2.5. CP/M stikordet

Så, nu tror jeg det er tilstrækkeligt. Lad os lade teorien ligge, og vende os til det virkelige liv, dvs. computeren og dens driftssystemer. Hvad du altså behøver, er en fungerende computer, en diskettestation og en diskette med CP/M driftssystemet på. Er det hele til stede, så tænder du for computeren, skubber disketten med CP/M driftssystemet ind i diskettestation "A" eller station "1" og låser den fast. Denne fremgangsmåde skal du lægge mærke til. Hvis du nemlig tænder eller slukker for computeren, medens der er en diskette i stationen, kan det på grund af strømstødet komme til uønskede bevægelser af skrive/læsehovedet, og din værdifulde diskette er ikke længere til at bruge.

Altså - nu har du tændt for computeren, lagt din CP/M systemdiskette ind og foretaget opstarten efter beskrivelsen i din brugsanvisning. På Amstrad CPC foregår det således, at du tænder (slukker og tænder) og så indtaster kommandoen:

ICPM

Den pæne lodrette streg får du, idet du trykker tasten SHIFT + @

Der er tre måder at starte CP/M på Commodore 128 på. Den første - ikke anbefalesværdige - er at lægge CP/M-systemdisketten i og tænde for computeren. Den anden metode - der kan anbefales - er at trykke på RESET-knappen. Den tredje måde er ikke ret meget mere kompliceret, men kræver ganske vist en indtastning: skriv kommandoen:

BOOT (RETURN)

Brugen af RESET-knappen bør imidlertid blive "standardmetoden" til at nå CP/M-funktionsmåden, den er den sikreste og hurtigste.

Diskettestationen begynder at rasle og klikke, funktionslampen lyser, og kort derefter ser du en melding på billedskærmen. Denne såkaldte startmelding ser lidt forskellig ud på hver computertype. Det ligger i, at denne melding kommer fra BIOS-delen af driftssystemet, altså den del, der må tilpasses specielt til hver enkelt computertype. Enhver computerfabrikant tilpasser den del til sin maskine og udformer startmeldingen efter sine egne ideer. En almindelig startmelding ved CP/M ser sådan ud:

CPM 2.2 - AMSTRAD Consumer Electronics plc.

A>

På CPC 6128 ser denne melding lidt anderledes ud, da det her drejer sig om CP/M 3.0's startmelding.

CP/M Plus - AMSTRAD Consumer Electronics plc.

V 1.0, 61K TPA, 1 (2) Disc drive

Før Commodores specielle startmelding fremkommer på skærmen, ser du på skærmen meldingen "Booting..." og så med blå skrift "Booting CP/M Plus".

Som du ved, kan du tilslutte en skærm med 40 tegn/linie og en skærm med 80 tegn/linie til din Commodore 128. Det er anbefalelsesværdigt at arbejde med en 80 tegns monitor, men også en 40-tegns monitor kan CP/M klare. Med 40/80-display-tasten kan du vælge, om du vil arbejde med 40-tegns skærmen eller med 80-tegns skærmen. Startmeldingen fremkommer stadig på begge skærme, men efter CP/M-stikordet vises kun noget på den valgte skærm.

På 40-tegns skærmen kan en 80-tegns skærm simuleres ved hjælp af vandret scrolling. Med TAB-tasten kan du forskyde skærbilledet. Vi vil hele tiden beskæftige os med 80-tegns skærmen i denne bog, med mindre andet er angivet.

Fremkommer denne melding:

*NO CP/M+.SYS FILE - HIT RETURN TO RETRY
DEL TO ENTER C128 MODE*

så har du helt sikkert lagt den forkerte side af CP/M-systemdisketten i. Vend disketten om og tryk på RETURN-tasten. Så prøver din Commodore igen at boote CP/M.

Der kan imidlertid også fremkomme en indlæsningsfejl. Så fremkommer der ligeledes en melding i den nederste del af skærmen, der lyder således:

*READ ERROR - HIT RETURN TO RETRY
DEL TO ENTER C128 MODE*

Vil du have, at der skal gøres et nyt forsøg på at boote, så behøver du kun at trykke på RETURN-tasten. Er du på den anden side så frustreret over dette mislykkede forsøg, at du hellere vil opgive, så tryk simpelthen på DEL-tasten. Enhver anden tast bliver i øvrigt ignoreret.

Hvis alt går efter planen, så dukker de følgende fem linier op på den nederste del af skærmen:

```
BNKBIOS3 SPR F400 0800
BNKBIOS3 SPR CA00 1600
RESBDOS3 SPR EE00 0600
BNKBDOS3 SPR 9C00 2E00
59K TPA
```

Det er oplysninger om de enkelte BIOS og BDOS-dele, om hvilken adresse, de bliver indlæst på, og hvilken længde de har. Det går egentlig ret hurtigt. 59K TPA betyder, at der er 59k bytes ledige til at load'e programmer og data. TPA betyder i øvrigt Transient Program Area. Er disse ting indlæst, så fremkommer der tillige følgende fire meddelelser, denne gang lidt højere på skærmen:

```
DATA TABLES
COMMON CODE
BANKED CODE
BIOS8502 CODE
```

Først når også disse dele af disketten har kunnet load'es, fremkommer den ovenfor omtalte typiske CP/M-melding på skærmen. Denne ser på Commodore 128 ud som følger:

```
CP/M 3.0 On the Commodore 128 3 JUNE 85
80 column display (resp. 40 column)
```

Du har sikkert lagt mærke til, at man under load'ningen kunne læse forskellige tal i nederste højre hjørne. Selv når startmeldingen er fremkommet, forsvinder disse tal ikke. Den sidste linie på skærmen er statuslinien, der ikke kan nås eller skrives oven i. Tallene i hjørnet angiver, hvilken blok der netop bliver læst eller skrevet. Et R betyder læsning, et W skrivning af en blok. Endvidere står der et A eller B, der angiver, på hvilken diskettestation, der arbejdes.

Denne informationslinje kan du til enhver tid tænde og slukke for med tastkombinationen <CTRL><RUN/STOP>, alt efter hvad der passer dig bedst.

A propos passe: måske behager skærbilledets farve dig ikke? Det er ikke noget problem. Ved at trykke på <CTRL>-tasten og en af tasterne 1-8 kan du vælge den øverst på tasten anførte farve som tegnfarve. Også baggrundsfarven kan du ændre, idet du trykker på <CTRL>-tasten og en af tasterne 1-8 på den numeriske tastgruppe. Hvis du for eksempel vil have sorte tegn på hvid baggrund, så skal du trykke <CTRL>-1 og <CTRL>-2 (numeriske tastgruppe). Det er det hele. *CP/M på Commodoren har imidlertid andre overraskelser parat.*

Dog tilbage til vor startmelding. Denne startmelding betyder for dig, at CP/M-driftssystemet er blevet indlæst korrekt i computerens arbejdslager. Lige herefter melder CP/M sig med sin paratmelding, det såkaldte driftssystem-stikord. Dette stikord ser sådan ud:

A>

Det store "A" fortæller brugeren, at han arbejder med sit systems diskettestation "A" eller station "1", ">"-tegnet er CP/M's egentlige paratmelding. I linien bagved paratmeldingen venter CP/M på indtastningen af en kommando. Nå, vi vil ikke lade CP/M vente så længe, så vi skriver:

A>abcdefgh

Nu står cursoren bagved det sidste bogstav, og der er intet der rører sig, stille hviler havet. Grunden: CP/M ved ikke, om vi allerede er færdige med at indtaste kommandoen eller måske vil skrive lidt mere til. For at afslutte indtastningen må vi også meddele CP/M, at vi er færdige med indtastningen.

Indtastningen af en linie bliver på computeren som på en skrivemaskine afsluttet med et "vognretur". Denne taste hedder på computere hyppigt RETURN eller ENTER, hvad der er en forkortelse af CARRIAGE RETURN eller blot betyder "ind". Så snart du trykker på denne taste, ved computeren, at du er færdig med kommandointastningen og begynder at udføre den tildelte kommando. Du trykker nu på vognretur- eller RETURN-tasten og ser:

A>abcdefgh

ABCDEF GH?

A>

Tja, hvad er der nu sket? CP/M har læst linien, ikke forstået kommandoen og meddelt os dette. Driftssystemet meddeler os sådanne uforståelige kommandoer, idet det gentager det indtastede og forsyner det med spørgsmålstegn. Spørgsmålstegnet betyder så meget som: "Hvad er det for noget ævl? Det kan jeg ikke bruge til noget".

Den gik altså ikke. Lad os forsøge igen og se, om det måske lå i de små bogstaver. Efter boot'ningen kan du indtaste både små og (ved hjælp af SHIFT) store bogstaver. Ved at trykke på SHIFT-LOCK-tasten (CAPS-LOCK på Amstrad) kan også til enhver tid indtaste bare store bogstaver. I øvrigt har tastkombinationen <C>=><SHIFT> nøjagtig samme virkning. Benyt nu <SHIFT-LOCK>-tasten.

A>ABCDEF GH

Som svar derpå får du:

A>ABCDEF GH

ABCDEF GH?

A>

Driftssystemet har altså heller ikke kunnet begribe kommandoen med små bogstaver. Lagde du mærke til før, at kommandolinien blev gentaget med store bogstaver, skønt den blev skrevet med små bogstaver? Det er en egenskab ved CP/M, som vi i mange tilfælde helt bevidst må tage hensyn til. CP/M forvandler først alle bogstaver, som du indtaster, til store bogstaver, og bearbejder dem så.

For at du kan se, at CP/M også virkelig kan arbejde, vil vi nu indtaste en kommando, der også får noget til at ske. Længere tilbage har du læst, at det hører til driftssystemets opgaver at forvalte indlæsningerne på disketten i en indholdsfortegnelse. CP/M anlægger til dette formål på hver diskette en indholdsfortegnelse (på engelsk directory) og viser den på forlangende.

Hvis vi vil se denne indholdsfortegnelse, må vi på en eller anden måde meddele driftssystemet, at vi vil se den, og at det skal afbilde denne indholdsfortegnelse på skærmen. Kommandoen består af de tre bogstaver DIR, hvad der er en forkortelse for directory. Indtast nu denne kommando og afsluttende ENTER:

A> DIR

Lige en bemærkning vedrørende Commodore. Fra BASIC er du måske vant til, at man kan vise indholdsfortegnelsen på skærmen ved hjælp af F3-tasten. For at det ikke skal være så svært at omstille sig, er dette også muligt i CP/M. Tryk simpelthen på F3-tasten. Ved et tryk på F4-tasten vises der kun kommandoordeet DIR på skærmen uden automatisk ENTER-funktion. På den måde kan du indtaste options, men mere herom senere.

Du ser nu din driftssystemdiskettes indhold på skærmen. Kig engang lidt nøjere på det. Helt til venstre angives stationen, i dette tilfælde "A:". Ved siden af står navnene på filer, og adskilt af et mellemrum angives tillige typen af fil. Meget hyppigt ser du tilføjeslen COM, f.eks. ved DDT.COM, PIP.COM, ASM.COM. Denne filbetegnelse giver oplysninger om filens indhold. For os er i øjeblikket COM-filerne af særlig interesse, fordi de indeholder programmer, der er parat til udførelse.

COM er forkortelsen for det engelske ord "command" eller kommando. De således betegnede filer indeholder kommandoer, som kan udføres med det samme. Hvis du indtaster navnet på et sådant program uden tilføjeslen COM bagved driftssystemstikordet, bliver programmet lagt direkte ind i computerens arbejdslager og udført. Det er altså tilstrækkeligt at indtaste PIP, tilføje et RETURN, og straks starter programmet.

2.6. Sikker er sikker

Hvis du stadigvæk arbejder med din originaldiskette, må vi hurtigst muligt fremstille en sikkerhedskopi. Du skulle vænne dig til, aldrig at arbejde med et programs originaler, men altid fremstille en kopi af programmet og opbevare originalen godt og sikkert. Som det sig hør og bør for et driftssystem, der jo skal lette arbejdet, muliggør CP/M fremstillingen af sikkerhedskopier.

2.7. Hvad du nu allerede ved

- * Du kender en simpel forklaring på, hvilke opgaver et driftssystem skal klare.
- * Du er klar over, hvad et program er, og hvorfor der findes underprogrammer.
- * Du kender også CP/M's opgaver.
- * Du ved, hvad CP/M-stikordet er, og hvordan du kan vise dine disketters indholdsfortegnelser.
- * Du er fast besluttet på at tage mindst een sikkerhedskopi af hver eneste vigtig diskette.

3. Hvordan man arbejder med CP/M

3.1. Systemdisketten

Jeg har allerede gjort dig opmærksom på, at du bør lave kopier af dine værdifulde disketter, så at der ikke sker originalen noget ved et eller andet tilfælde. Hvordan man kopierer en hel diskette, skal jeg nu vise dig, og vi benytter dertil CP/M-kommandoer, som først bliver nærmere forklaret senere. Du har brug for to programmer fra CP/M-disketten for at kunne kopiere en hel diskette. Hvis du endnu en gang indlæser DIR og ser nærmere på indholdsfortegnelsen, vil du blandt andet finde to programmer, der hedder:

SYSGEN.COM og PIP.COM i CP/M 2.2,
COPYSYS.COM og PIP.COM i version 3.0
FORMAT.COM og PIP.COM I VERSION 3.0 PLUS (Commodore)

Normalt kopierer man system-CP/M'et ved hjælp af kommandoen COPYSYS. Denne kommando er dog ikke implementeret til Commodore CP/M. - dvs. den er ikke tilstede. Du kan alligevel, hvis du er vant til kommandoen COPYSYS, indtaste denne kommando, men på skærmen fremkommer der så blot en meddelelse om, at kopieringen skal foregå på en anden måde med Commodore 128.

Læg altså systemdisketten ind med den side, der er betegnet "System Disk" opad, og så kan det gå løs. Pas af sikkerhedsgrunde på, at skrivespærringen ikke er fjernet.

Den opgave, der ellers er overtaget af kommandoen COPYSYS, nemlig kopieringen af systemet på de to første spor 0 og 1, bliver nu meget praktisk udført direkte af kommandoen FORMAT. På den måde er disse to første spor altid optaget, men det har ringe indflydelse på diskettens lagringsevne. Endvidere præparerer FORMAT-rutinen også disketten direkte med en BOOT-sektor, så CP/M også kan bootes efter det næste RESET eller indtastningen af kommandoen BOOT. For den uerfarne bruger bliver arbejdet lettere meget herved.

Systemsporene, der ikke er synlige i det "normale" CP/M, får et navn med Commodore 128: det drejer sig om filerne CP/M+.SYS og CCP.COM.

I filen CCP.COM gemmer sig de såkaldte indbyggede (engelsk: resident) kommandoer, altså de kommandoer, der med CP/M bliver forstået, uden at der skal load'es en COM-fil fra disketten. Disse kommandoer er permanent i din Commodore 128's arbejdslager.

Ganske vist ser dette hos firmaet Amstrad lidt anderledes ud med hensyn til CP/M 3.0. COM-filen COPYSYS befinder sig ikke på nogen af de tre diskettesider. Programmet til kopiering af systemets COPYSYS er integreret i programpakken DISCKIT3. Programpakken er meget praktisk, men hvis jeg må have lov at sige det,

så havde det nok ikke været dyrere at levere COPYSYS.COM med. Der mangler også COM-filen til formattering af en diskette. Jeg beskriver i den følgende tekst den "normale" fremgangsmåde ved CP/M 3.0, dvs. når begge COM-filer befinder sig på systemdisketten. Slå efter i kapitel 7 under FORMAT og COPYSYS, hvordan du kan anvende disse kommandoer på Amstrad'en.

Med de to programmer gør du forskellige ting. Med programmet SYSGEN (COPY-SYS) kopierer du CP/M-systemet på de første to spor af hver diskette. Derfor hedder disse to første spor også system-sporene. De bliver læst først ved starten af din computer, og det program der står herunder, bliver ladet ind i computerens arbejdslager. Til system sporene har du, som computerbruger ikke nogen direkte tilgang, af hvilken grund du ikke kan finde nogen CP/M.COM på disketten eller i indholdsfortegnelsen. Du behøver blot at vide, at systemsporene altid er reserveret til CP/M, og at her står alle de programdele, der er nødvendige for driften af en computer.

Hvis du nu vil kopiere en hel diskette, dvs. alle programmer og også CP/M-driftssystemet, har du brug for at kunne tilgå disse to systemspor. Det hjælper programmet SYSGEN (CP/M 2.2) eller i CP/M 3.0 programmet COPYSYS dig med. De er begge specielle programmer, der kan overføre indholdet af systemsporene fra en diskette til en anden.

Før du begynder kopieringsarbejdet, skal du have en frisk formatteret og tom diskette til rådighed. (En diskette formatteres med CP/M-kommandoen FORMAT, idet man indtaster denne kommando bag CP/M-stikordet og følger de anvisninger, der kommer til syne på skærmen.)

3.2. Kopiering med een diskettestation

Hvis du har alt parat, så lægger du din CP/M-originaldiskette på station "A" og lukker døren til stationen. Arbejder du med kun een station, får du lidt mere arbejde, end hvis du kunne arbejde med to stationer. Men det går også på den måde, når du følger de følgende trin. For at kopiere dit driftssystem skubber du originaldisketten ind i stationen og indtaster for Amstrad:

A>SYSGEN (ENTER)

Derefter fremkommer denne melding på skærmen:

Please insert SOURCE disc into drive A
then press any key:

hvor der med SOURCE menes din originaldiskette eller den, hvorfra driftssystemet skal tages. Når du har sat disketten i stationen, trykker du på ENTER, og denne melding fremkommer:

Please insert DESTINATION disc into drive A
then press any key:

Du sætter den nye diskette ind i stationen, som efter et fornyet ENTER påbegynder sit arbejde. Længe varer det ikke, og du ser følgende:

Do you wish to reconfigure another disc (Y/N)

Hvis du her svarer Y for ja, kan du forsyne flere disketter med driftssystemet. Ved et N for nej, lander du igen ved system-stikordet.

Med Commodore indtaster du efter at have lagt originaldisketten i stationen:

A>FORMAT (RETURN)

Derpå fremkommer følgende melding på skærmen

```
C128 FORMAT PROGRAM
15 May 1985
Drive A is a 1571 (1541)
Please select disc type to format
C128 double sided
C128 single sided
C64 single sided
```

Med cursortasterne i den første tastrække udvælges et af de tre formater, hvori du vil have din diskette formateret. Ejere af en 1570'er skal bemærke, at den ikke kan formatere dobbeltsidigt. Det er den lille, men vigtige forskel på 1570'eren og 1571'-eren: 1570'eren kan ikke læse og skrive dobbeltsidigt på disketterne.

Lad os derfor formaterer vor første diskette enkeltsidigt (single sided). Til den ende må du først trykke på cursor-ned tasten i den øverste tastrække, så kan du trykke på RETURN-tasten. På skærmen fremkommer så:

```
Formatting C128 single sided (invers)
Insert Diskette TO BE FORMATTED
in drive A. Type $ when ready,
any other key to abort
```

Det betyder at den diskette, der skal formateres, skal lægges ind i stationen. Skulle du have begået en fejl ved valget af det format, der skal formateres i, så klares den vanskelighed ved at trykke på enhver anden tast end \$-tasten. Men hvis du vil formaterer den indlagte diskette, så trykker du simpelthen på \$-tasten, og så går det løs:

```
Formatting C128 single sided (invers)
```

Hvis der skulle optræde en fejl under formateringen, for eksempel fordi den indlagte diskette ikke er helt i orden, så slettes skærm billedet, og den røde LED-lampe på disktestationen blinker. Man bør så kontrollere sin indtastning (f.eks. om man har forsøgt at kopiere dobbeltsidigt med 1570'eren) og eventuelt også disketterne,

for så at forsøge påny. Men selvfølgelig har vi ikke begået nogen fejl, så FORMAT-programmet udlæser følgende tekst:

Do you want to format another disk?

Det spørgsmål kan du besvare med N for nej, da vi har nok i een formatteret diskette. Hvis du imidlertid vil formatere endnu en diskette, så tryk venligst på Y-tasten, da dialogen med CP/M, foregår på engelsk - desværre må man sige for alle dem, der ikke mestrer det engelske sprog.

Efter at vi nu har formatteret en diskette, processen ser til nu fuldstændigt identisk ud for ejere af to diskettestationer, skal vi så kopiere de to omtalte filer. Det sker ved hjælp af PIP-kommandoen, som vi har viet et ekstra kapitel, fordi den er så omfangsrig og ligeså vigtig for CP/M-anvenderen. Selv om du endnu ikke rigtig ved, hvad du gør, så tag det roligt, snart vil indtastningerne ikke længere være en by i Rusland for dig. For øjeblikket må vi imidlertid nødvendigvis fremstille en sikkerhedskopi af CP/M - sikker er sikker.

Altså lader vi handling følge på ord og indtaster den følgende linie

```
PIP E:=A:CPM+.SYS
```

Bliv ikke bange - efter nogen tids load'ning fremkommer følgende opfordring i den underste linie (ja, rigtigt, statuslinien):

```
Insert Disk E in Drive A
```

Du bliver hermed opfordret til at tage systemdisketten ud og indsætte den nyformaterede diskette (der her betegnes med E) i stationen. Når du har gjort det, så vær ikke kostbar, men tryk på RETURN-tasten. Meldingen forsvinder, og diskettestationen begynder igen at arbejde. Den kopierer for øjeblikket den indlæste fil CPM+.SYS over på den nye diskette. Når arbejdet er afsluttet, fremkommer CP/M-stikordet igen på skærmen. På samme måde skal du også kopiere filen CCP.COM. Det sker du ved at indtaste følgende kommando:

```
PIP E:=A:CCP.COM
```

CP/M er imidlertid ikke så dumt og ved godt, at du stadigvæk har måldisketten med navnet E i stationen. Det opfordrer dig derfor i statuslinien til at indlægge diskette A igen - hvad du selvfølgelig gør og kvitterer for med RETURN-tasten. På ny lægger du E-disketten i, og også denne fil CCP.COM bliver kopieret. Hvis du nu vil se på indholdsfortegnelsen for den "nye" CP/M-diskette, så indtaster du den bekendte kommando DIR. I statuslinien bliver du opfordret til at indlægge A-disketten. Lad nu roligt din diskette forblive i stationen; ved at trykke på RETURN-tasten forvandler du din E-diskette til A-disketten. Det er det hele. På skærmen fremkommer der:

```
A: CPM+  SYS : CCP  COM
```

Dermed er det første skridt gjort. Da vi først og fremmest vil kopiere hele disketten af sikkerhedsgrunde, må vi kopiere alle filer. Det gøres at indtaste følgende kommando:

```
PIP E:=A:.*
```

På skærmen informeres om, hvilken fil der er ved at blive kopieret. I statuslinien bliver du hele tiden opfordret til at skifte diskette, det går ikke anderledes end med en enkelt diskettestation. Følg anvisningerne, skift disketterne og tryk derefter på RETURN-tasten. Når du er færdig, har du allerede kopieret den første side af CP/M-systemdisketten.

```
COPYING -  
CPM+.SYS  
CCP.COM  
HELP.COM  
HELP.HLP  
KEYFIG.COM  
KEYFIG.HLP  
FORMAT.COM  
PIP.COM  
DIR.COM  
COPYSYS.COM
```

Det var den første side - hvis du også (og det bør du) vil kopiere den anden side, så må du hente en diskette til, ligeledes formattere denne og indtaste kommandoen PIP E:=A:.*. Alt det andet bliver vist på skærmen. Når du har kopieret begge sider, skal du helst kun arbejde med kopien af CP/M og opbevare originalen på et sikkert sted - man ved jo ikke, hvad der kan ske.

3.3. Kopiering med to stationer og CP/M 3.0 (Commodore)

Hvis du skulle være den lykkelige ejer af to diskettestationer, så går kopieringen af filer noget lettere. I selve betjeningen ændres praktisk talt ikke noget, bortset fra, at du i stedet for den virtuelle (dvs. "indbildte") station E nu skal angive den reelt eksisterende station B:. Det betyder, at formatteringen forløber på samme måde som beskrevet under III.2, en formattering kan nemlig desværre kun lade sig gøre på A-stationen.

Ved PIP-kommandoen skal du i stedet for PIP E:=A:. skrive:*

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

Der kommer ikke nogen opfordring til at skifte diskette i statuslinien. Før du indtaster denne linie, bør du ganske vist have indlagt en nyformatteret diskette i B-stationen. Har du ikke nogen B-station, eller er den ikke ordentlig forbundet eller endda frakoblet, så får du følgende fejlmelding:

gennemprøv så din indtastning, resp. disktestationernes elektriske forbindelser.

AMSTRAD:

Når du arbejder med to stationer, fungerer processen ligesådan, bortset fra, at du kopierer fra den ene station til den anden. Den tomme diskette lægges i station B og lukker også døren der. Du starter driftssystemet, indtaster bagved CP/M's paratmelding COPYSYS (SYSGEN ved CP/M 2.2) og slutter med et ENTER.

De to programmer melder sig herefter med deres navne og versionsnumre og forlanger indtastning af den station, hvorfra der skal kopieres. Da din CP/M-driftssystemdiskette ligger i station, indtaster du et "A" og bliver straks derefter spurgt, om den rigtige diskette ligger i station A. Hvis svaret er ja, så tryk ENTER, og spørgriet fortsætter.

Som det næste skal du angive den station, hvori din jomfruelige, dvs. nyformattede diskette ligger. Også her behøver du ikke at indlæse noget ENTER, fordi programmet straks, som det næste, vil have at vide, om disketten nu også er lagt i og til bekræftelse udtrykkeligt beder om et ENTER. Når du har fulgt alle instruktionerne og har indlæst det sidste ENTER, begynder station B at køre, og CP/M driftssystemet bliver skrevet på den nye diskette. Derpå følger spørgsmålet, om CPM3.SYS også skal kopieres. Det er vigtige hjælpeprogrammer, og du bør have dem på enhver kopi af din systemdiskette. Så snart station B er faldet til ro, er arbejdet færdigt, og du kunne godt lægge den næste diskette ind til overførelse af driftssystemet. Da vi har mere for endnu med vor driftssystemdiskette, trykker du blot ENTER og afslutter systemkopieringsprogrammet.

3.4. Hvordan man ser på indholdsfortegnelsen

Du har kopieret noget fra A til B og er nysgerrig efter at vide, hvad der i det hele taget står på B-disketten. For at du ikke skal revne af nysgerrighed, indtaster du lige efter CP/M-stikordet:

Dir B:

Hvis du ikke har glemt mellemrummet efter DIR, så vil du få at se, hvordan station B begynder at arbejde, og du får en melding på skærmen. Med DIR B: har du bedt CP/M om at vise indholdet af disketten i station B. Hvis du giver denne kommando med en diskette, som du lige har formatteret uden at have kopieret noget ved hjælp af PIP, fremkommer nu meldingen NO FILE. FILE bliver på dansk til fil, og NO FILE betyder, at der ikke står noget på disketten. Som du erindrer, bliver programmerne på de to første systemspor på hver diskette skjult. Du har derfor ingen direkte adgang til nogen af dem, og kan heller ikke se dem i indholdsfortegnelsen.

3.5. Kopiering med PIP (og to stationer, Amstrad)

Normalt bruges PIP kun til kopiering af filer, når man er i besiddelse af to diskettestationer. Det forholder sig anderledes ved Commodore 128. Man kan ved at opgive diskettestation E, som er en fysisk umulighed (kun stationerne A: - D: er mulige) tilgå en såkaldt virtuel station, som vi og computeren blot forestiller os, men som slet ikke eksisterer i virkeligheden. Man henvender sig hele tiden kun til station A.; men computeren underretter hele tiden om, hvilken diskette man skal lægge ind. Man skal derfor lægge mærke til, hvilken diskette, der er A-disketten, og hvilken der er E-disketten. Tricket med den virtuelle station kan i øvrigt ikke kun anvendes sammen med PIP, men mere herom senere.

Fordi vi nu vil kopiere hele CP/M-systemdisketten og ikke kun driftssystemet, betjener vi os af et andet program. Dette andet program har det uendeligt lange danske navn, processor til dataudveksling mellem ydre enheder, hvad der på engelsk bliver til Peripheral Interchange Processor, og forkortet giver et lystigt PIP. Hvad dette program gør, er overhovedet ikke pip, men derimod uhørt nyttigt.

Du har endnu de to disketter i stationen og taster nu PIP og derefter RETURN (Amstrad: ENTER). Computeren lader nu PIP ind i arbejdslageret, hvorpå dette melder sig med sit eget stikord. Det ser sådan ud på Amstrad:

A>PIP

*

og på Commodore:

A>PIP

CP/M 3 PIP VERSION 3.0

*

Asterisken viser dig, at PIP er parat til at modtage kommandoer. Dette program er særdeles nyttigt og ydedygtigt. Du skal lære PIP's mange muligheder nærmere at kende i kapitel 6.

Før vi nu giver PIP en tilsvarende kommando, skal vi lige tænke på, hvad, det egentlig skal gøre. Filerne fra disketten i station A skal overføres på disketten i station B (Eller til "station E", hvis man kun har en diskettestation (Commodore)). Man kunne også sige, at "B:" får alle de filer, der er lagret på "A:".

Den sidste sætning skulle du se nærmere på og frem for alt indprente dig grundigt. Den er afgørende ved arbejdet med PIP. Når alle filer er overført fra A til B, er indholdet af begge disketter ens. Derfor bliver der brugt lighedstegn i PIP-kommandoer. Nu er der kun det problem tilbage, hvordan vi fortæller PIP, at det skal tage alle filerne og ikke kun en enkelt.

Det er der taget højde for. En asterisk i stedet for et filnavn betyder i PIP så meget som "alle filer". Du kan opfatte asterisken som en joker i et kortspil, der kan indsæt-

tes i stedet for et hvilket som helst kort. Men fordi filer nu engang adskiller sig, for det første ved filnavnet, for det andet ved tilføjelsen, det er de tre små bogstaver bagved punktummet, så må du, for at kopiere alle filerne indlæse asterisk/punktum/asterisk. Det ser sådan ud på skærmen:

B:=A:.* resp. E:=A:*.* (Commodore)*

Idet vi vil være sikre på, at alle filer er blevet overført korrekt, lader vi straks alle filer afprøve. Vi kan overgive arbejdet til PIP, idet vi skriver et "V" i firkantede klammer bagved vor kommando. "V" står for "verify", hvad der betyder at kontrollere, verificere. For at PIP skal forstå denne befaling rigtigt, skal den stå i firkantede klammer. *Imidlertid udfører Commodore 128 et automatisk Verify, så man i praksis kan udelade option'en [V]. Har du indtastet hele kommandoen:*

A>PIP

B:=A:.*COM

så går det også rigtig løs. Som du vil se under kopieringsprocessen, meddeler PIP, hvilken fil, der er blevet kopieret.

På Amstrad ser du følgende CP/M 3.0-indholdsfortegnelse på skærmen

A: CCP COM : DATE COM : DEVICE COM : DIR COM : DUMP COM

A: ED COM : ERASE COM : GENCOM COM : GENCPM COM : GET COM

A: LIB COM : LINK COM : MAC COM : PATCH COM : INITDIR COM

A: PUT COM : RENAME COM : RMAC COM : SAVE COM : SET COM

A: HEXCOM COM : SETDEF COM : SHOW COM : SID COM : SUBMIT COM

A: TYPE COM : XREF COM : COPYSYS COM : FORMAT COM : HELP COM

A: PIP COM

Når arbejdet er gjort, melder PIP sig igen med sit parattegn, asterisken. Hvis du ikke har mere arbejde til PIP, så indtast ENTER, og driftssystemet melder sig igen med sit A-stikord.

Hvis du ikke tror helt på teknikken, kan du med DIR B: se efter, om alle filerne er ankommet til B:-stationen (resp. den anden diskette). Er alt tilfredsstillende, så tag disketten ud af A-stationen, altså din original-diskette, anbring den i disketteomslaget og læg den af vejen på et sikkert sted. For fremtiden arbejder du kun med den kopi, som du lige har fremstillet, og sparer mange ærgrelser hvis noget går galt. Thi: en diskette til at kopiere et nyt system på, koster dig små tyve kroner (3-tommers disketterne til Amstrad er noget dyrere), men et nyt CP/M-driftssystem derimod koster straks adskillige hundrede kroner.

3.6. Regler for filnavne

Hvad en fil, er og hvad den dermed har i sig, har du endnu ikke erfaret.

Som du ved, har enhver fil i CP/M et navn. Det er vigtigt for dig, for at du kan vide, hvad der står i den; for det andet er det vigtigt for CP/M, for at det også skal kunne finde denne fil på dit krav herom. Desværre har CP/M-konstruktørerne indbygget en begrænsning, der f.eks. overhovedet ikke passer mig.

Et filnavn i CP/M kan bestå af maksimalt otte tegn, et punktum og tre tegn yderligere til den såkaldte udvidelse (extension). Det betyder, at du må finde navne til dine filer, der højst er på otte tegn og trods det har en mening, og tillader slutninger om deres indhold.

De tilladte tegn for et filnavn er alle alfabetets bogstaver (minus de specielt danske bogstaver Æ, Ø og Å?) såvel som plus-, minus-, skråstregs-, procent- og dollar-tegnet i en hvilken som helst position indenfor det første navn eller i udvidelsen. Tegnene "mindre end", "større end", punktum, komma, kolon, lighedstegn, semikolon, asterisk, spørgsmålstegn, firkantede klammer start, firkantede klammer slut og udråbstegn, må du ikke benytte i filnavne eller deres udvidelser, da disse tegn har en ganske bestemt betydning for CP/M og kan føre til funktionsfejl.

Ud fra min mangeårige erfaring med CP/M og andre programmer kan jeg råde dig til, at du giver dine filer et meningsfyldt navn, der tillader, at man slutter noget om indholdet. Specielt når du har mange filer stående på din diskette, kan du overhovedet ikke længere finde ud af det med navne som XK2512AC.ABC. Ved valget af dine filnavne behøver du ikke at lade nogen særlig forsigtighed råde, fordi to filer med det samme navn og samme udvidelse ikke kan stå samtidig på den samme diskette. Ganske vist kan filer med samme navn og forskellige udvidelser stå samtidig på en diskette.

3.7. Filnavnudvidelser

Udvidelsen af et filnavn kan du udforme, efter ønske, dvs. du kan kalde hver fil, hvad du vil, men du bør herunder tænke på de typiske CP/M-forkortelser. En tabel over de almindelige forkortelser i CP/M og deres betydning finder du i slutningen af dette kapitel. Der kan du finde programmer med filnavnudvidelsen COM. Det er en forkortelse for COMMAND og betyder, at dette program indeholder kommandokoder, der er øjeblikkeligt udførbare.

Du erindrer nok at PIP og SYSGEN eller COPYSYS er sådanne COM-programmer. Desuden bør du vogte dig for at betegne filer med udvidelserne BAK eller \$\$\$\$. Bak står for "back-up", hvad der benyttes af CP/M-editoren til at betegne sikkerhedskopier. En fil med tre dollar-tegn til sidst opfattes som en midlertidig fil, der bliver omvendt af et program og ved afslutningen af programmet uden tøven bliver slettet. PIP anlægger for eksempel sådanne midlertidige filer og sletter dem til sidst fuldstændigt fra disketten. Pas altså hellere på.

Det er også uklogt at anvende samme filnavne og kun bruge udvidelsen til at tælle filerne. Du kan f.eks. have en tekst og kalde den første fil TEXT.1, den anden fil TEXT.2 og den tredje fil TEXT.3. Hvad der for øjeblikket kan se ganske logisk ud, vil derefter give dig ærgelser. Så snart du nemlig har bearbejdet den første fil, bliver din originalfil automatisk til den nye fil TEXT.BAK, og den bearbejdede version hedder nu TEXT.1. Nu bearbejder du filen TEXT.2, og også her bliver originalfilen til TEXT.BAK og den reviderede version til TEXT.2.

Da der i CP/M ikke kan stå to filer med nøjagtig det samme navn, og den samme udvidelse på een diskette, har driftssystemet uden videre slettet din første BAK-fil og overskrevet den med den anden. Derfor har du nu ingen mulighed for at få teksten i din første original-TEXT.1-fil at se igen.

For at undgå sådanne problemer, skal du kalde din første fil TEXT1.TXT, din anden fil TEXT2.TXT og din tredje fil TEXT3.TXT.

Når du derefter gennemarbejder filerne, får hver af dem sin egen back-up fil, og du kan hente din originaltekst frem igen.

3.8. Hvordan man genfinder en fil

Hvis du benytter din computer intensivt og fremstiller mange forskellige filer, står du muligvis foran et nyt problem. Du kan ikke længere finde filen lige med det samme i indholdsfortegnelsen. Desuden kan det også være ret nyttigt for dig at finde ud af, hvor mange filer med navnet TEXT.TXT, der allerede står på disketten

Du har to muligheder til søge efter disse filer. Som hjælpere hertil tjener i begge driftssystemversionerne asterisken og spørgsmålstegnet. Asterisken har du allerede lært at kende, som et såkaldt jokertegn ved vore kopieringsoperationer med PIP. Spørgsmålstegnet virker på lignende måde. Medens asterisken kan stå i stedet for alle tegn i filnavnet eller udvidelsen, virker spørgsmålstegnet som erstatning for eet tegn på en bestemt position.

Indlæsningen "text?" for eksempel søger og finder alle filer med navnet "text" plus endnu eet tegn. Således ville alle tre filer "TEXT1", "TEXT2" OG "TEXT3" blive fundet. Skulle der desuden stå endnu en fil med navnet "TEXT" på vores diskette, ville den også blive fundet og vist. Det ligger i, at CP/M altid reserverer otte tegn til et filnavn. Hvis dit filnavn er kortere, så udfylder CP/M de resterende pladser med tomme mellemrum. Herunder er "mellemrumstegnene" ligeså gyldige tegn som bogstaver eller tal.

3.9. Søgning med spørgsmålstegn (?)

Eftersom spørgsmålstegnet kan skrives i stedet for ethvert andet tegn, findes ikke alene filerne "TEXT1", "TEXT2" og "TEXT3" frem, men tillige filen "TEXT". Spørgsmålstegnets anvendelse begrænser sig ikke til et enkelt tegn. Du kan ligeså godt

indtaste indtil otte spørgsmålstegn for filnavnet og indtil tre for udvidelsen. Det er besværligt skriveri, som du kan undgå ved at anvende en asterisk for filnavnet og en asterisk for udvidelsen. Driftssystemet er ligeglad med, om du skriver otte spørgsmålstegn for filnavnet eller en asterisk. Internt bliver hver asterisk nemlig alligevel forvandlet til otte spørgsmålstegn, før søgningen begynder.

Man kan også søge mere præcist med asterisken, når man skriver de søgte filers begyndelsesbogstav eller begyndelsesbogstaver foran asterisken. Men hvis man skriver en asterisk og så et par bogstaver, fungerer søgningen ikke. Denne metode, indlæsningen af filnavne med spørgsmålstegn eller asterisker, kan du også anvende sammen med kommandoerne DIR, TYPE, STAT, FILECOPY og PIP (Amstrad) resp. DIR, TYPE, ERASE, SHOW OG PIP (Commodore)

3.10. Hvad du nu ved

- * Du kan nu kopiere en systemdiskette og opbevare originalen korrekt
- * Med asterisken og spørgsmålstegnet som erstatning for manglende bogstaver kan du også bruge en smutvej og gøre livet lettere for dig selv.
- * Du ved, at man altid skal give filer hensigtsmæssige navne, så man let kan finde dem igen.

4. Indbyggede kommandoer

I det sidste kapitel har vi lært et par grundfunktioner ved driftsystemet CP/M at kende; men dermed er vi langt fra færdige. I dette kapitel skal vi beskæftige os nærmere med de indbyggede CP/M-kommandoer, der lyder således: USER, DIR, DIRSYS (CP/M 3.0), ERA, ERASE (CP/M 3.0), REN, RENAME (CP/M 3.0) og TYPE. (Amstrad), resp. USER, DIR, DIRS(YS), ERA(SE), REN(AME) OG TYPE (Commodore)

Desuden tager vi os senere også af de såkaldte transient-programmer (det er CP/M programmer, som du kan se i indholdsfortegnelsen, og som er lagret som COM-filer). Endvidere vil du få udvidet dit kendskab til kommandoerne DIR og PIP, som du har lært at kende i forrige kapitel.

4.1. Kommandoer, parametre og options

Før jeg forklarer dig flere CP/M-kommandoer, skal du lære den grundlæggende forskel mellem en kommando, en parameter og en option. En kommando er et "befalingsord", der beordrer CP/M til at udføre en bestemt handling. En parameter er i reglen et filnavn, der angiver, hvilken fil kommandoen vedrører. En option står i CP/M i firkantede klammer og kan, som navnet allerede antyder, også udelades. Men hvis en option læses med ind, så forandrer den forud indlæste kommandos funktion sig på en bestemt måde.

Et eksempel på en option har du allerede set, dengang vi kopierede din systemdiskette med PIP. Option'en hed "V" og fik PIP til at kontrollere alle de kopierede data.

Alt eftersom hvilke kommandoer det drejer sig om, bliver parametre tastet med ind eller ej. Hvis der skulle være brug for en parameter, og du ikke har indtastet nogen, forlanger CP/M udtrykkeligt, at der anføres en parameter. Du behøver altså ikke at være bange for, at der skal være noget, der ikke virker.

Ved indlæsningen af kommandoer og parametre, må du passe på, at der er mindst eet mellemrum mellem kommandoen og parameteren. Og det er det eneste sted, hvor et mellemrum må dukke op. Hverken i kommandoen, eller i parameteren eller i optionen må der forekomme mellemrum.

Normalt vil du angive eet eller to filnavne efter en kommando som parameter. Du er ikke tvunget til at nøjes så få angivelser. Hvis du ønsker det, kan du skrive hele linien fuld, begynde en ny linie med Control E (↑E) og gå videre på den. CP/M vil læse begge disse linier som een kommandolinie.

4.2. De indbyggede kommandoer

Så snart CP/M er indlæst fra disketten i arbejdslageret, har du to typer kommando til din rådighed. Indlæsningen af de indbyggede kommandoer sker ved Commodore 128 gennem indlæsning af filen CCP.COM, som det allerede tidligere blev nævnt. De indbyggede kommandoer er i din computers arbejdslager og kan med det samme og meget hurtigt kaldes frem, for at udføre en eller anden opgave. De andre kommandoer står som fil på din diskette, og du kan få en liste over dem ved at se på indholdsfortegnelsen. Disse programmer lader sig kopiere eller slette som ganske normale filer, eller, hvis du ved nok om det, også forandre.

Denne opdeling i CP/M blev foretaget, fordi de to systemspor på hver diskette, der er reserveret til CP/M, langtfra giver tilstrækkelig plads til alle hjælpeprogrammerne. For dit arbejde med computeren er det imidlertid uden betydning, om du kalder en indbygget (engelsk: resident) kommando eller en midlertidig (engelsk: transient) kommando. Hvis du f.eks. med kommandoen PIP forlanger nogle ganske specielle procedurer, så læser CP/M automatisk filen PIP.COM ind i arbejdslageret og udfører din befaling.

Det eneste problem, du får, er tab af tid på grund af læsningen af filen fra disketten. I alt besidder driftssystemet CP/M 2.2 fem og version 3.0 seks indbyggede kommandoer, hvoraf fire gennemgik en udvidelse ved hjælp af en COM-fil af samme navn. Hvis du i CP/M 3.0 kalder en af disse kommandoer, er det ikke nødvendigt at indtaste det komplette navn hver gang. Du kan forkorte kommandointastningen, som angivet i den nedenstående tabel:

Kommando	Forkortelse	Funktion
DIR	DIR	Viser indholdsfortegnelsen
DIRSYS	DIRS	Viser fortegnelse over systemfiler
ERASE	ERA	Sletter filer
RENAME	REN	Ændrer filnavne
TYPE	TYP	Viser tekstfil
USER	USE	Ændrer brugerområdet

I 2'ers versionen kommer du ikke langt, hvis du skriver navnene på kommandoerne helt ud, fordi driftssystemet ikke forstår dem. Her kan du kun arbejde med forkortelserne.

Det gælder også for en indbygget kommando i CP/M, der ikke har noget navn. Du kan benytte denne kommando til at stille om fra diskettestation A: til station B: eller en anden station. Hvis du i stedet for station A hellere vil have B som den "noterede" (engelsk: default) station, så indtaster du blot:

B: (RETURN)

og CP/M gør din B-station til den noterede station. Alle stationsbetegnelser fra A: til E: er tilladt. Alle andre fører i CP/M til fejlmeldingen:

```
CP/M Error On F: Invalid Drive  
BDOS Function = 14
```

Når du nu arbejder på B-stationen, medens din CP/M-systemdiskette ligger i A, kan du alligevel gribe til CP/M-programmerne på A-disketten. Så skriver du simpelthen stationsbetegnelsen plus kolon foran kommandoen. Det kan se sådan ud:

```
B>A:DIR
```

Ligesådan kan du sætte stationsbogstaver foran en parameter:

```
B:DIR A:FIL.XXX
```

Indtast dog igen A:, hvis du har to stationer og har stillet om til den anden. Du vil undre dig, for disketteindholdsfortegnelsen på en enkelt diskette kan indeles meget fint.

4.3. USER

Det indbyggede program gør det muligt for dig at underopdele et lagermedium i 16 områder. Områdebetegnelserne går fra 0 til 15. Sådan en opdeling er nyttig, når flere brugere deler den samme computer, og deres filer ikke skal blandes sammen, f.eks. i skoler o.l. Man kan også opbevare forskellige programmer og de tilhørende filer i hver sit specielle brugerområde.

Du kan oprette et område til dine tekstfiler, et område til dine BASIC-filer, et område til forretningsbreve osv. osv. Rigtig hensigtsmæssig bliver underopdelingen af et lagermedium egentlig først, når man anvender en hard-disk, som byder på betydeligt mere plads end en diskette, og gør det ret svært at holde styr på alle filerne.

Gennem underopdelingen i områder bliver det også muliggjort, at to filer med nøjagtigt samme navn og samme filnavnudvidelse kan stå på et enkelt lagermedium - ganske vist med hver fil i sit område.

4.3.1. USER-områder med CP/M 2.2

Vi vil også informere dig, kære læser, om hvordan CP/M 3.0's forgængere, dvs. specielt CP/M 2.2, har håndteret visse ting. Det gavner gennemskeligheden, og så er du ikke helt fortabt, hvis du engang hos en ven kommer til at prale med, at du behersker CP/M, og han så har en computer med CP/M 2.2.

Fra flerpladsdriftssystemet MP/M blev muligheden for forskellige brugerområder overtaget af versionen CP/M 2.2. Hvert område arbejder som en separat diskette. Filer bliver oprettet i hvert brugerområde for sig, og de nødvendige programmer skal være til rådighed i dette brugerområde. Mellem områderne skiftes der om med kommandoen

```
USER n
```

n'et står for et tal mellem 1 og 15. Du kan arbejde indenfor et område uden at røre ved filerne i andre områder. For eksempel sletter kommandoen

```
ERA*.*
```

alle filer, men kun indenfor eet brugerområde. Arbejder du med to stationer og kopierer filer fra den ene til den anden, fremkommer disse filer normalt på den anden station i det samme brugerområde. Når du altså kopierer fra område 3A med PIP til B:, havner filerne i område 3B.

Efter enhver koldstart af CP/M begynder driftssystemet i brugerområde nul. Vil du arbejde i et andet, må du indtaste det nye brugerområde. Desværre kan du ikke se på CP/M stikordet, i hvilket område du befinder dig. Det kan du kun få at vide med kommandoen STAT. Heldigvis er det anderledes ved CP/M 3.0

4.3.2. USER-områder med CP/M 3.0

Nu kommer vi til et kapitel, der er vigtigere for brugere af CP/M 3.0. En særstilling indtages af brugerområde nul (0). Programmer og filer, der står og blev defineret som systemfiler, kan kaldes og anvendes af ethvert område. Nærmere herom i et senere afsnit. Når du befinder dig i et nyt område og opretter en ny fil, lægger CP/M automatisk mærke til hvor.

Når du har tændt for din computer og ladet CP/M, er du i område 0. Hvis du vil arbejde i et andet brugerområde, kan du indtaste kommandoen USER. Antag, at du vil til område 3, så indtaster du:

```
USER 3
```

Husk at skrive et mellemrum imellem USER og det pågældende tal. Så snart CP/M har skiftet om til det nye område, ser du, at noget ændrer sig på skærmen. CP/M-stikordet er ikke længere blot et "A", men du ser brugerområdet foran A'et, i dette tilfælde et "3". Nu vil du sikkert også arbejde med den anden station og kunne skifte brugerområder der. For f.eks. at komme til område 3 på station B indtaster du:

```
USER 3B:
```

og ser så CP/M stikordet, der ser sådan ud:

```
3B>
```

Det kan man ikke med Commodore 128's CP/M 3.0. Her må processen opdeles i to indtastninger. Først valg af Userområde, så angivelse af station:

USER 3

B:

Skulle du komme til at indtaste kommandoen USER uden en userbetegnelse, så er det ingen skade til. CP/M 3.0 spørger dig, hvilket userområde det er, du vil have fat i.

A>USER

Enter User_: 4

4A>

Folkene hos Digital Research har tænkt på (næsten) alt, ikke sandt?

Hvis du indtaster kommandoen DIR for at se indholdet af din diskette, ser du "NO FILE". Forudsat naturligvis, at du endnu ikke har oprettet nogen filer i dette område. Du kan gribe til alle indbyggede CP/M-kommandoer fra ethvert område. For atter at komme til brugerområde 0 indtaster du

USER 0

og havner igen i laveste brugerområde. Programmer, der er til rådighed for alle områder, kan du forberede herpå med SET-kommandoen. Hvordan den virker, vises i et senere afsnit. Skiftet mellem brugerområder kan udformes endnu simple, idet du indtaster dit ændringsønske således

B1:

På Commodore kan man gøre det omvendt, altså "1B:". Der er endnu en ting, der bør bemærkes med hensyn til brugerområderne. Når du kopierer en fil med PIP fra "A" til "B" eller omvendt, bliver der kun kopieret indenfor samme område.

Facit: brugerområder kan være en meget hjælpsom ting, særligt når man råder over en hard-disk. Men også på en diskette kan man adskille f.eks. data og programmer klart og ordentligt fra hinanden med forskellige brugerområder. Alligevel bør man i begyndelsen, anvende brugerområderne med forsigtighed, da de kan give forvirring, f.eks. at man ikke kan finde en fil igen, fordi man ikke er helt sikker i anvendelsen af brugerområder.

4.4. DIR

Kommandoen DIR har du lært at kende, i forrige kapitel. DIR-kommandoen er den samme i begge CP/M-versionerne, bortset fra, at den i 3'er udgaven består af to programmer. Den indbyggede kommando DIR, der ligner den fra CP/M 2.2, og det andet af programmet DIR.COM, med betydeligt udvidede hjælpefunktioner.

DIR-kommandoen findes i Commodore i to udgaver. Som indbygget eller som midlertidig kommando. Hvis Dir-kommandoen indtastes med options, altså med firkantede klammer, så load'es filen DIR.COM (det er den midlertidige DIR), ellers nøjes CP/M med den noget enklere udformede indbyggede, altså i arbejdslageret forekommende DIR, og dette er i de fleste tilfælde tilstrækkeligt.

Vi beskæftiger os nu med kommandoen DIR, som den kan benyttes i begge versioner.

Indtaster du kommandoen DIR, uden at tilføje noget andet, fremkommer alle din diskettes filer i det specielle brugerområde. Hvis du skriver DIR i brugerområde 3, vises kun de filer, der er lagret i område 3. Ofte står der med CP/M 3.0 i et brugerområde så mange filer, at langt fra alle kan få plads på en skærm.

I det tilfælde standser udlæsningen på billedskærmen, indtil du har skaffet dig overblik og ved at trykke på en tilfældig tast, giver ordre til at gå videre. Hvis du vil stoppe udlæsningen af indholdsfortegnelsen, trykker du Control C (↑C).

Hvis du arbejder med station A, men gerne ville se, om en fil er på station B, findes der to muligheder for at se på station B's indholdsfortegnelse. Du indtaster:

B:
DIR

og ser indholdsfortegnelsen for station B. Den anden metode går hurtigere. Du kommanderer:

DIR B:

og ser ligeledes indholdsfortegnelsen for station B på din skærm. En ekstra fordel ved den anden metode: du bliver på station A. Hvis du vil printe indholdsfortegnelsen, så indtaster du:

DIR B:

og så Control P (↑P) og ENTER resp. RETURN. Nu bliver hele din indholdsfortegnelse skrevet ud på printeren. Efter at printningen er afsluttet, trykker du endnu engang Control P, og printeren standser.

4.4.1. DIR med parametre

Rigtigt nyttig bliver DIR, når du udnytter de mange ekstra muligheder, der er i den ydedygtige kommando. Lad os antage, at du gerne vil finde frem til en bestemt fil, så indtaster du blot

DIR B:TEXT.TXT

og dette navn vises på skærmen, hvis filen findes på station B. Er filen ikke til stede, får du en fejlmelding fra CP/M.:

No File

Hvis du vil benytte udvidede funktioner af DIR, skal du i version 3.0 arbejde med programmet DIR.COM. Det betyder: hvis du arbejder på station B, men DIR.COM befinder sig på station A, må du ikke glemme at taste stationsbetegnelsen med ind foran DIR. Det kan for eksempel se sådan ud:

A:DIR [options]

Du husker sikkert, at asterisken kan spare os en mængde tastearbejde ved søgning. Med kommandoen DIR og kombinationen med asterisken kan du lade liste en bestemt type af filer. Det fungerer på denne måde:

DIR*.COM

Hvis du indlæser kommandoen på denne måde, får du alle filer at se på skærmen, der har COM som filnavnudvidelse

4.4.2. Udvidet DIR med CP/M 3.0

I de nyere CP/M-versioner kan du nøjagtig ligeså godt søge efter flere forskellige filtyper, dvs. f.eks. samtidig efter filer med udvidelsen .COM og efter filer med udvidelsen .SYS. Den tilsvarende kommando ser således ud:

DIR.COM*.SUB (Amstrad) eller DIR*.COM*.SYS (Commodore)*

Hvis man indlæser den sidste kommandolinie på Commodore, så kan CP/M ikke gøre meget med den, men spørger forvirret:

**.SYS?*

*Det betyder: angivelsen *.SYS er for meget eller forkert! Her havde jeg ventet en option med firkantede klammer, men da den mangler, benytter jeg den indbyggede DIR og ikke den midlertidige. Og for den indbyggede DIR er to angivelser simpelthen for meget som parameter.*

Skal det fungere, må du udtrykkeligt forlange af CP/M, at DIR.COM bliver anvendt.

Det opnås, ved at indtaste stationsbetegnelsen som A: eller B: foran DIR, eller ved at skrive en option i firkantede klammer bagved filnavnet. De to options, der er mulige i dette tilfælde, hedder [FULL] eller [DIR].

4.4.3. DIR og dens options

Du kan angive 18 forskellige options sammen med DIR. Dermed er DIR ligeså yde- dygtigt som PIP og hører til de særligt nyttige CP/M-programmer. Normalt vil du sjældent indtaste mere end een eller to options samtidig. Options til DIR står altid i firkantede klammer.

Når der angives mere end een option, adskilles de to af kommaer eller mellemrum. Hvis option-betegnelsen er entydig, kan du også forkorte option'ens navn til to bog- staver. For at gøre det enklere, kan du udelade den afsluttende firkantede klamme, når den er det sidste tegn i kommandolinien.

Eksempler:

```
DIR*.COM*.SYS[FULL]
DIR* *[NOSORT,SIZE]
DIR*.BAS[USER=5 NOSORT SIZE]
```

Anvendes det midlertidige DIR, bliver først DIR.COM indlæst i TPA'en. Tænk på, at DIR.COM trods alt er 15K bytes stor og først må søges og load'es. Når det er sket, frem- kommer der på skærmen:

Scanning Directory...

Så bliver, alt efter de angivne parametre og options, de filer valgt ud, der skal udlæses. Først når dette er sket, og hvis option'en NOSORT ikke er angivet, kan den alfabetiske sortering af filnavnene påbegyndes. Også sorteringsprocessen bliver meddelt brugeren:

Sorting Directory...

er driftssystemets melding. Så følger udlæsningen af den ønskede indholdsfortegnelse på skærm eller printer. En typisk udprintning ser eksempel sådan ud:

Directory For Drive A: User 0

Name	Bytes	Recs	Attributes	Name	Bytes	Recs	Attributes
CCP	COM	4k	25 SYS RW	COPYSYS	COM	1k	3 DIR RW
CPM+	SYS	23k	182 DIR RW	DIR		15k	114 DIR RW
FORMAT	COM	5k	35 DIR RW	HELP		7k	56 DIR RW
HELP	HLP	83k	664 DIR RW	KEYFIG		10k	75 DIR RW
KEYFIG	HLP	9k	72 DIR RW	PIP		9k	68 DIR RW

4.4.4. DIRSYS

Når du ser på en indholdsfortegnelse i CP/M 3.0, har du sikkert lagt mærke til en melding ved den nederste kant af skærmen, der ser således ud:

SYSTEM FILE(S) EXIST

Dermed gør CP/M opmærksom på, at der foruden de anførte filer også står såkaldte systemfiler på disketten. Ganske vist er der ingen systemfiler på systemdisketten. Systemfiler er sådanne filer, som ligger i brugerområde 0 og er tilgængelige og anvendelige fra ethvert område. Hvis du vil se systemfilerne, skal du give kommandoen DIRSYS eller forkortet DIRS. Blandt de anførte filer fremkommer der så en meddelelse om, at der eksisterer "ikke-systemfiler".

Naturligvis har du med kommandoen DIRSYS de samme muligheder som med kommandoen DIR, du kan bruge asterisken og spørgsmålstegnet, nøjagtig som tidligere omtalt.

4.5. ERASE

Pladsen på din diskette, og antallet af mulige optagelser i hver diskettes indholdfortegnelser, er begrænset.

Helt nøjagtigt har du med Commodore med enkeltsidig formattering 64 optagelsesmuligheder og med dobbeltsidig formattering 128. Det maksimale antal filer og det faktisk tilstedeværende antal bliver også skrevet ud, når du anvender den midlertidige DIR.

Før eller senere kommer du til at mangle plads på disketten, og bliver tvunget til at slette en fil eller endda flere. Driftssystemet CP/M giver mulighed for at slette filer med kommandoen ERA.

Den kommando taster du ind således:

```
ERASE D:NAME
```

hvor D står for stationsbetegnelsen og NAME for din fils navn. Hvis du arbejder med station A og vil slette en fil der, behøver du ikke at indtaste stationsbetegnelsen.

For at slette flere filer på en gang kan du igen benytte de to tegn, asterisken og spørgsmålstegnet. Ganske vist bør du nøje overveje, hvad du gør, når du anvender disse to mægtige kommandoer. Kun alt for let kan det ske, at du sletter de forkerte filer.

Hvis du vil slette alle filer, der har BAK som filnavnudvidelse, så indtaster du:

```
ERA *.BAK
```

og er med CP/M 2.2 fri for alle filer med denne udvidelse. Ved back-upfiler, som de her beskrevne, er fejlen sandsynligvis ikke så alvorlig; men forestil dig engang, at du havde givet denne kommando:

```
ERA *.*
```

og så havde skrevet ENTER, resp. RETURN, bagefter. Tja, det var så det. Dine data er ødelagt.

Men netop fordi denne kommando er så mægtig og kan ødelægge så meget, har CP/M's opfindere indbygget et fikst lille sikkerhedstrin. Så snart du nemlig sletter med to asterisker (*.*), bliver du spurgt, om du mener det alvorligt:

```
ALL (Y/N)
```

Skal det hele virkelig slettes, skriver du et Y for ja og så ENTER. I modsat fald et N for nej, og situationen er reddet.

4.5.1. Sletning med ERA ved CP/M 3.0

Sansynligvis har chefen for DIGITAL RESEARCH en gang ved en fejl selv slettet alle sine filer med sine hemmelig bankkonti. For i den nye CP/M-version spørger programmet ved indlæsning med blot eet jokertegn:

```
ERA *.*BAS
```

straks for at få bekræftet, om virkelig alt skal slettes:

```
ERASE *.*BAS (Y/N)?
```

Før du svarer med Y for ja, bør du endnu engang overveje, hvad det er, du gør. Er du helt sikker på, at kommandoen er rigtigt formuleret, og at du vil slette alle filer af denne type? Har du først tastet dit "Y", er der ingen vej tilbage. Filerne er væk. Pas også på skrivefejl. For eksempel kan man let ved en anslagsfejl komme til at skrive tilføjelsen BAS som PAS. Hvis du sender kommandoen afsted i den form, forsvinder alle PASCAL-filer, der står på din diskette...

Der er to muligheder for at beskytte sig mod den slags ubehagelige overraskelser. Du kan "skrivebeskytte" alle dine vigtige filer. Hvordan det foregår, ser du senere. For det andet kan du sætte option'en CONFIRM efter kommandoen ERASE. CONFIRM betyder at bekræfte, og du kan forkorte option'en til et kort "C". Når du nu indtaster kommandoen således:

```
ERASE *.BAK[C]
```

bliver alle "back-up" filer slettet, men CP/M spørger før hver sletning, om denne fil også virkelig skal slettes.

4.6. Ændringer af filnavne med REN(AME)

Du har hidtil set, at vi henviser til filer ved at angive eller indtaste filnavnene. Nu kunne det jo ske, at et filnavn bagefter ikke mere behagede dig, eller at du vil

kopiere en fil på en anden diskette, hvor der allerede står en fil med det navn. I så fald kan du anvendes kommandoen REN eller RENAME i version 3.0 (på dansk: ombenævne) til at omdøbe dine filer. RENAME-kommandoen lyder.

```
RENAME Ny=Gammel
```

først kommer filens nye navn, så det gamle. Mellem de to navne står et lighedstegn. Når "omdøbningen" skal ske på den noterede station, behøver du ikke at skrive nogen stationsbetegnelse foran filnavnet. Hvis du prøver at give en fil et navn, der allerede er i brug, bliver du advaret af CP/M 2.2:

```
File exists
```

Det nye CP/M er her en smule mere hjælpsomt, og du bliver spurgt, om den gamle fil skal slettes. *Den melding ser i Commodore-CP/M ud som følger:*

```
Error: Not renamed Ny .EXT file already exists, delete (Y/N)?
```

Hvis du besvarer spørgsmålet med et N for nej, er sagen afgjort for RENAME. Alt bliver ved det gamle. Indtaster du Y=ja, så bliver filen NY.EXT slettet og filen GAMMEL.EXT får navnet NY.EXT. Interessant er det, at den indbyggede del af RENAME klarer en simpel ombenævning, men hvis der optræder komplikationer, i dette tilfælde hvis den nye fil allerede eksisterer, så bliver den midlertidige del af RENAME indlæst, før det kan gå videre.

En "simpel" omdøbning af en fil ser således ud

```
REN ny.bas=gl.bas
```

Du kan med RENAME 3.0 igen bruge asterisken og spørgsmålstegnet. Et simpelt eksempel ser således ud:

```
RENAME *.TXT=*.BAK
```

Med den kommando bliver alle back-up filer på disketten forvandlet til filer med udvidelsen TXT, ligegyldigt hvad der står i filerne. Kommandoen RENAME bekymrer sig ikke om indholdet af filer, men kun om navngivningen.

4.7. TYPE

Kommandoen lader filer vise sig på skærmen. Det virker dog kun med tekstfiler, fordi de bliver dannet med ASCII-kodens tegn. Andre filer med udvidelserne COM, REL eller lignende indeholder styretegn, der bringer forvirring i enhver skærmudlæsning og får computeren til at gå agurk. TYPE-kommandoen indtaster du sådan:

```
TYPE A:FIL.XXX
```

Hvis du vil se en fil fra en anden station, skriver du blot en anden stationsbetegnelse forrest.

Til den kommando findes der i CP/M en option:NO PAGE. Den lader den viste tekst scrolle kontinuerligt henover skærmen. Normalt stopper skærmudlæsningen ellers efter 23 linier og går først videre efter et ENTER.

Hvis du har tastet NO PAGE ind, kan du stoppe skærmudlæsningen med Control S (↑S) og få den til at løbe videre med Control Q (↑Q). Hvis du også har indtastet Control P (↑P), så bliver hele skærmudlæsningen skrevet ud på din printer.

I øvrigt er den anden side af CP/M-systemdisketten, altså den side, der er betegnet "Utilities", en assembler-kildefil, som du udmærket kan lade udlæse som et eksempel. Det drejer sig om kildefilen til kommandoen DATEC. For at se kildekoden på skærmen indtastes følgende kommando:

`TYPE DATEC.ASM`

og du får DATEC-kommandoens kildeprogram-listning på skærmen eller printer. Da det er en ASCII-fil, er TYPE-kommandoen i stand til at udlæse filen "fornuftigt", altså uden styretegn.

4.8. Hvad du nu ved

- * CP/M besidder indbyggede kommandoer såvel som sådanne, der først bliver muliggjort af et program på disketten.
- * Kommandoer kan indtastes med parametre eller options.
- * Du ved, hvad alle de indbyggede kommandoer hedder, hvordan de virker, og hvilke options der kun er mulige med midlertidige kommandoer.

V Midlertidige kommandoer

Om de midlertidige kommandoer har du allerede hørt noget. Driftssystemets virkelige magt ligger i de midlertidige kommandoer, der kan findes som COM-filer i indholdsfortegnelsen. Den første del af dette kapitel beskæftiger sig med CP/M 2.2's kommandoer og specielt med STAT(for Amstrad-ejere).

Alt for hyppigt har vi omtalt dem, disse vidunderkommandoer, der bliver indlæst i arbejdslageret, ved at disketten lægges i stationen. Det er ikke uden grund, at disse kommandoer er lagret på diskette: de er simpelthen for store, så de ville tage for megen plads op, hvis de skulle være i arbejdslageret. Og store er de på grund af de mange muligheder, som de giver.

Da midlertidige kommandoer skal søges i indholdsfortegnelsen og så indlæses i arbejdslageret, er de ikke helt så hurtige som deres indbyggede kolleger.

I den anden del, der er betydeligt større, skal du lære CP/M 3.0's mangfoldige muligheder at kende.

5. Externe kommandoer

5.1. Statusangivelser med STAT (kun for Amstrad)

Kommandoen STAT kan på den ene side benyttes til at vise systemets status, på den anden til at ændre apparaturladningen i CP/M. Den enkleste form for kommandoen, der også bliver brugt meget hyppigt, er denne:

STAT

fulgt af et ENTER. Dermed kan man vise, hvor meget plads der er til rådighed på disketten. Det ser sådan ud:

A:R/W,SPACE:89k

Angivelsen betyder, at der er 89 kilobytes til rådighed på disketten, der kan skrives på og læses fra. Det sidste kan du se af betegnelsen R/W, der står for Read/Write og betyder læse/skrive. I modsætning hertil findes der en diskettestatus R/O (Read Only: kun læse), når disketten er forsynet med en skrivespærring. Så kan du ikke skrive noget på disketten, men kun læse, hvad der står på den.

Ligeså interessant som den plads, der er tilbage på en diskette, kan også størrelsen af en fil, eller af en bestemt type filer være. Du kan se størrelsen af en fil, på en hvilken som helst station, når du også indtaster den tilsvarende stationsbetegnelse:

STAT B:DATA.CMD

Så får du følgende oplysninger via skærmen:

```
RECS BYTES EXT   Acc
-----
12   2K   1 R/W B:DATA. CMD
Bytes remaining on A: xK
```

Hvis du benytter asterisken som jokertegn, når du beder om disse oplysninger, således:

STAT B:*.CMD

får du tilsvarende mange meldinger i ovenstående format.

Feltet RECS angiver, hvor mange sætninger filen optager. I feltet BYTES står størrelsen af en fil i enheder på en K, medens man under EXT kan aflæse antallet af sammenhængende blokke.

Hvis du vil forsyne hele disketten med en skrivebeskyttelse, kan du ved hjælp af STAT fastsætte R/O-attributten. R/O er det modsatte af R/W og tillader kun at læse filer. For at virkeliggøre en sådan skrivebeskyttelse indlæses:

STAT B: \$R/O

Nøjagtig ligesådan, hvis du i stedet for en hel diskette vil beskytte en enkelt fil eller en gruppe af filer:

STAT FIL.BAS \$R/O

Da tildelingen af forskellige brugerområder i CP/M 2.2 desværre ikke vises sammen med stikordet, må du benytte STAT-kommandoen, for at fastslå, i hvilket, du befinder dig. Du får straks vist, hvilke andre områder, der ellers indeholder filer, og på hvilket brugerområde, de står. Du indtaster altså:

STAT USR:

og får denne oversigt:

Active User 0

Active Files:0 2 3 5 11 7

Deraf kan du se, i hvilket område, der står filer. Hvis der ikke står mindst een fil i et brugerområde, vises der heller ikke noget i denne oversigt.

5.2. Midlertidige kommandoer med CP/M 3.0

Nu skal vi se på nogle vigtige og hyppigt brugte CP/M-kommandoer. Det er midlertidige kommandoer, som du kan finde i din indholdsfortegnelse forsynet med filnavn-udvidelsen COM. Du ved, hvorledes du skal kalde programmer frem: indtast programnavnet uden udvidelsen og derefter et ENTER, resp. RETURN. Du har tre muligheder for at kalde de midlertidige programmer frem. For det første, når du er i brugerområde 0 og også vil arbejde der. For det andet, når du har gjort programmerne tilgængelige for alle brugerområder med en SET-kommando. For det tredje, kan du bruge kommandoen SETDEF. Om den sidstnævnte kommando, skal jeg senere fortælle dig mere.

5.3. SET

SET-kommandoen overtager forskellige af CP/M 3.0's opgaver. Hovedsagelig bliver den anvendt for at angive forskellige fil-attributter. Det er tilføjelser til filerne, der har en bestemt virkning. For eksempel anvendes SET-kommandoen til at gøre dine CP/M-filer i USER-område 0 tilgængelige for alle USER-områder. Men du kan yderligere beskytte dine filer ved at sige: "Denne fil må kun læses" eller "denne fil er beskyttet af et kodeord".

SET-kommandoen indeholder også options, der har indflydelse på indholdsfortegnelsen. Du kan oprette en indholdsfortegnelse, hvori hver fil får et "tidsstempel", der fortæller, hvornår den første gang blev anlagt, og hvornår der sidst blev arbejdet med den.

Før du kan indrette en indholdsfortegnelse med "timestamps" eller "tidsstempler", må du lade programmet INITDIR.COM køre. Du kan bruge den indretning senere, til automatisk at sikre alle de filer, der blev ændret, på en anden diskette.

Men hvis du vil bearbejde en diskette med INITDIR, så bliver der lavet grundigt om på disketteindholdsfortegnelsen, og der bliver brugt plads. Det er af flere grunde anbefalelseværdigt at anvende kommandoen INITDIR på en diskette, før der endnu befinder sig en eneste fil på den. Det skal være forekommet, at alle optagelser på en diskette er gået tabt under problemer med INITDIR

Præparer en diskette med INITDIR A: for at kunne afprøve vore eksempler på computeren.

For at lære de mange muligheder i SET at kende, kommer der en oversigt over mulighederne:

OPTION	BETYDNING
DIR	Får en systemfil til igen at blive synlig i den normale indholdsfortegnelse
SYS	Gør en fil til SYSTEM-fil
RO	Bestemmer, at en fil kun skal kunne læses
RW	Bestemmer, at en fil skal kunne læses og ændres
ARCHIV=OFF	Sætter ARCHIV-attributten på "slået fra". Det betyder, at denne fil endnu ikke er blevet sikret (arkiveret).

SET PIP.COM[SYS RO]

Dermed er PIP.COM nu en system-fil og tilmed skrivebeskyttet. Hvis du igen vil gøre en således sikret fil frit tilgængelig og ophæve SYSTEM-attributten, skriver du:

SET PIP.COM[DIR W]

Herunder er det ligegyldigt, i hvilken rækkefølge du indtaster options. Hovedsagen er, at du skiller options logisk med et tomt mellemrum eller et komma osv.

5.4. Diskettestations-attributter

Du kan definere hele stationer således, at der kun kan læses fra dem, men ikke skrives på dem. Har du sat en station på "RO", hvad der står for "read only" (kun læse), er det ikke muligt at slette nogen filer med ERASE. RENAME fungerer ikke, og PIP kan ikke kopiere nogen filer på den.

Så snart du indtaster et `Control-C` (tryk Control-tasten ned, hold den nede og tryk "C"), bliver stationens RO-status ophævet igen. Hvis du skriver:

```
SET A:[RO]
```

er station A: skrivebeskyttet. Indtaster du R/W, (dvs. read/write = læse/skrive) i stedet for RO, kan stationen igen benyttes fuldt ud. *At sætte Read-Only attributtet er ensbetydende med at klæbe en skrivebeskyttelsesetikette på.* Dette attribut kan man ikke ophæve ved hjælp af `Control-C`

5.5. Labels

Hvis du har mange disketter, eller flere brugere arbejder med den samme computer, er det nyttigt at give disketterne navne. Du benytter atter SET-kommandoen, men med option'en "NAME=". Diskettenavne har de samme begrænsninger som filnavnene. Dvs., du kan maksimalt benytte otte tegn til navnet og tre til en udvidelse

For at forsyne en diskette med navn indtaster du:

```
SET B:[NAME=BOGHOLD]
```

Specielt for Commodore: Har du kun rådighed over een diskettestation, så fungerer SET-kommandoen naturligvis kun, når filen SET.COM også befinder sig på den diskette, som du vil bearbejde ved hjælp af SET.

Løsningen? Klar nok: du anvender den virtuelle station E:. Indtast altså det følgende; herunder skal diskettesiden med påskriften "Utilities" befinde sig i stationen:

```
SET E:[NAME=BOGHOLD]
```

CP/M henter den nødvendige fil SET.COM og opfordrer til at lægge diskette E: ind i station A:. Har du fulgt denne opfordring, så fremkommer følgende tekst på skærmen:

Label for drive E:

Directory Label	Passwds Reqd	Stamp Create	Stamp Access	Stamp Update
E:BOGHOLD	off	off	off	off

De enkelte tilstande, der er mærket med off, skal jeg gå ind på senere.

Det kan være en betegnelse for en diskette med dit bogholderi på. Hvis du arbejder på station B:, kan du udelade stationsbetegnelsen. Ved anvendelse af E-stationen er angivelse af stationsbetegnelse uomgængelig. Du kan ikke se navnet på en diskette i din indholdsfortegnelse, når du kalder den frem med DIR. Dertil anvender du kommandoen SHOW; en nærmere forklaring kommer senere, og sammen med den option'en "LABEL". Denne option kan forkortes til et simpelt "L", og kommandoen ser sådan ud:

SHOW B:[L] (Amstrad) resp. SHOW E:[L] (Commodore)

hvorunder du igen kan glemme stationsbetegnelsen, såfremt du arbejder med station B:.

For Commodore: CP/M 3.0 opfordrer dig under de kommende eksempler igen og igen til at indlægge henholdsvis diskette A eller diskette E. Det er bedst, at du følger opfordringerne og trykker på RETURN-tasten, hver gang du har lagt den pågældende diskette i stationen.

Nu bliver udlæsningen mere detaljeret, end vi har fået den med SET-kommandoen:

Label for drive E:

Directory Label	Passwds Reqd	Stamp Create	Stamp Access	Stamp Update	Stamp Label	Stamp Created	Stamp Label	Stamp Updated
E:BOGHOLD	off	off	off	off	12/15/82	23:41	12/15/82	23:41

Som på en professionel stor computer får man opgivelser over oprettelsesdato og sidste ajourføring (update) af en fil. Når man forandrer en fil, bliver updatedatoen altid ændret. På den måde kan man skaffe sig nogle fordele, f.eks. med PIP-kommandoen. Man kan med den passende option lade alle på en dag forandrede filer update. Det hele har kun mening, hvis man ved hver eneste brug af computeren også indlæser korrekt dato og tid med DATE-kommandoen, ellers stemmer de angivne datoer jo ikke. Som i vort eksempel.

5.6. Passwords

For at ingen skal bringe forstyrrelse i din smukke disketteorden, findes der i CP/M muligheden for at fastsætte et password. Du beskytter altså dine disketter med et password, som kun du kender.

Så snart du har indlæst et password for en diskette eller for en enkelt fil, forlanger CP/M før hver kommandoudførelse nøjagtigt dette password. Derfor må du hellere lave en fil med alle passwords, for at du selv i længden skal kunne finde rundt i det.

Har du glemt et password, er der ikke længere nogen mulighed for, at du kan få fat i dine data. Pas altså hellere på.

For at forsyne din label med et password, benytter du SET-kommandoen med option'en "PASSWORD=password" eller i modsat fald "PASSWORD=<cr>" (<cr> står for RETURN). Password'et indlæser du på denne måde:

```
SET [PASSWORD=HEMMELIG]
```

Eller for at ophæve et password:

```
SET [PASSWORD=<cr>] (<cr> = ENTER)
```

Commodore: Du kan kun ophæve et password, hvis du også kan indtaste det - det er vel klart. Lad os en gang forsyne DIR.COM på CP/M-sikkerhedsdisketten med et password:

```
SET E:DIR.COM[PASSWORD=DADA]
```

Hvis du vil ophæve password'et, så spørger CP/M:

Password?

Det skal du ikke undre dig over - men ligegyldigt hvor meget du hakker omkring på tastaturet, så fremkommer der ikke et bogstav på skærmen. Af sikkerhedsgrunde, forstår sig: For hvis enhver kunne se, hvad du indtastede, så var den sikkerhed ikke så meget værd. Har du indtastet et forkert password, så udlæses kommandoen ikke, men der fremkommer en fejlmelding:

```
ERROR: Wrong Password
```

Endnu en gang: husk at opbevare dit password et eller andet sted. Hvis du ikke husker det længere, har du ingen adgang til dataene på disketten.

5.7. PROTECT

Du kan ikke alene forsyne din diskettes label med et kodeord, men også enkelte filer eller endda CP/M-kommandoer. Sådan kan du sikre dig, at der ikke er nogen, der ubeføjet leger med dine data, eller får indblik i ting, som ikke kommer ham ved.

Før du nu kan beskytte en enkelt fil eller kommando, må du aktivere PROTECT-funktionen. Det sker ved, at du indlæser den følgende kommando:

```
SET [PROTECT=ON] (Commodore: SET E:[PROTECT=ON])
```

eller omvendt:

SET [PROTECT=OFF] (Commodore: SET E:[PROTECT=OFF])

Når du har indtastet kommandoen kan filerne beskyttes.

Commodore: Når du har indtastet SET E:[PROTECT=ON] er driftssystemet indstillet til at beskytte enkelte filer. Det er vigtigt at bemærke, at der findes forskellige former for beskyttelse, som du bør lære at kende før du giver dig til at prøve dig frem. Derom mere i kapitel V.8.

5.8. Fil-PASSWORD

For at sætte en fil under din personlige beskyttelse går man i princippet frem på samme måde, som ved beskyttelse af et diskettenavn.

Flere disketter kan sættes under eet enkelt passwords beskyttelse, når du bruger en asterisk (*). Så gælder det samme password for alle filer, hvorpå betingelsen passer. Kommandointastningen ser således ud:

SET TEKST.TXT[PASSWORD=hemmelig] (Amstrad)

resp.

SET E:DIRCOM[PASSWORD=hemmelig] (Commodore)

eller udtrykt med "***":

SET*.TXT[PASSWORD=hemmelig]

resp.

SET E:*.COM[PASSWORD=hemmelig]

I det første af de viste eksempler er filen TEKST.TXT beskyttet ved hjælp af "hemmelig" i det andet eksempel får alle TXT-filer password'et "hemmelig".

Yderligere kan fastlægges, hvordan filerne skal beskyttes. Vi har allerede omtalt, at der er forskelle. Dertil findes følgende muligheder:

- READ:** Password'et er nødvendigt for at læse, kopiere, skrive på, slette eller ombenævne filer.
- WRITE:** Password'et er nødvendigt for at skrive på, slette eller ombenævne filen.
- DELETE:** Password'et er kun nødvendigt for at slette.
- NONE:** Der findes ikke noget password. Hvis der allerede findes et password, kan det slettes med denne option.

Indtastningen foregår som følger. Angives der ingen form for beskyttelse, så antages READ at være givet.

```
SET TEKST.TXT[PROTECT=DELETE].  
(Commodore: SETE:TEXT.TXT[PROTECT=DELETE])
```

Dermed har du beskyttet filen TEKST.TXT mod at blive slettet ved en fejltagelse.

For at forebygge misforståelser: Man kan (desværre) ikke beskytte COM-filer på denne måde, før de bliver udført. Hvis man forsyner en COM-fil med et password og beskyttelsestypen READ, får man ved et forsøg på at udføre kommandoen meldingen:

```
CP/M Error On A: Password Error  
BDOS Function = 15 File = DIR .COM
```

Der er ingen anmodning om password før kommandoens udførelse, kommandoen er simpelthen spærret. Først når man vil ophæve password'et, spørger CP/M om det indtastede password, der skal skrives korrekt, fordi beskyttelsen ellers ikke kan ophæves eller ændres. På den måde kan du beskytte hjemmelavede COM- eller SUB-filer mod at blive udført af ubeføjede personer. Du skal selv før udførelsen ophæve beskyttelsen ved hjælp af SET-kommandoen.

5.9. TIME STAMP

Med tids-stempel indretningen kan du få et overblik over, hvor ofte du bruger en bestemt fil. Hvornår du har oprettet og ændret en fil, og holde styr på dine sikkerhedsfiler. Et tidsstempel er ligesom kontroluret på en arbejdsplads, hvormed alle arbejdsdata bliver dokumenteret.

For at kunne dokumentere tid og dato for alle filer, må du først lade programmet INITDIR køre. Dokumentation af tidspunkterne er kun mulig, når indholdsfortegnelsen er organiseret på en bestemt måde. Det gøres med INITDIR. *Læg mærke til, at der går plads tabt i indholdsfortegnelsen, når en diskette behandles med INITDIR; plads der bruges til de tidsangivelser, der skal lagres. På en diskette kan så langtfra længere være 64 filer, snarere halvdelen!*

Du har tre forskellige options på en diskette, du har behandlet med INITDIR; options du kan slå til eller fra ved hjælp af SET-kommandoen. Her skal man tage hensyn til, at ikke alle options aktiveres parallelt.

CREATE=ON: Aktiverer tids- og datomarkering for filerne på en bestemt station.

ACCESS=ON: Aktiverer markeringen for den sidste tilgang til en fil. Du kan kun aktivere enten CREATE eller ACCESS. Når ACCESS vælges til en station, hvortil CREATE allerede er blevet valgt, kobles CREATE automatisk ud.

UPDATE=ON: Sørger for en markering af filerne, hver gang de bliver bearbejdet på ny. At de vises i en indholdsfortegnelse, har ingen virkning på dette tidsstempel.

Hvis du vælger UPDATE og CREATE som dine options, skal du bemærke, at begge tidsstempler bliver nyindstillet ved en behandling af filerne.

Det skyldes, at når man bearbejder en fil, bliver der oprettet en ny, medens den gamle bliver til en BAK-fil (sikkerhedskopi).

Dit computer-kontrolur får du i gang på denne måde:

SET [ACCESS=ON] (Amstrad)

SET E:[ACCESS=ON] (Commodore)

For at se, hvad der kommer ud af denne operation, skriver man kommandoen:

DIR[FULL]

og får dette:

Directory for Drive B:

Name	Bytes	Recs	Attributes	Prot	Update	Access
TEKST.TXT	5K	38	DIR RW	NONE	04/01/85	17:31
BOGHOLD	20k	152	SYS RO	NONE	04/01/85	09:10

Under overskriften ACCESS kan du aflæse, hvornår du sidst har benyttet hvilken fil. Kommandoen for to indførelser i indholdsfortegnelsen ser sådan ud:

SET [CREATE=ON,UPDATE=ON] (Amstrad)

SET E:[CREATE=ON,UPDATE=ON] (Commodore)

En indholdsfortegnelse ville så have denne opbygning:

Directory for Drive B:

Name	Bytes	Recs	Attributes	Prot	Update	Create
TEKST.TXT	5K	38	DIR RW	NONE	04/17/85	10:00
					01/01/85	09:00
BOGHOLD	20K	152	SYS RO	NONE	04/17/85	16:43
					01/01/82	19:21

5.10. SETDEF

CP/M søger efter indtastning af en kommando den tilsvarende fil, der kan udføre CP/M. Det kan være COM-, men f.eks. også SUB-filer. CP/M søger altid på den noterede station, med mindre du har sat et A:, B: eller noget i den retning foran kommandoen. Lad os gennemgå et eksempel på, hvornår SETDEF kunne blive af interesse. Du kan arbejde på station B:, men have alle CP/M-filer eller andre programmer lagret på station A:. Med SETDEF giver du CP/M en rute at søge på, så det kan load'e de ønskede programmer, uden at taste stationsbetegnelsen med ind.

Når du simpelthen indtaster:

```
SETDEF
```

fås oplysninger om, hvordan din søgemåde ser ud, hvilken station, til midlertidige filer og hvilke filer, der bliver søgt efter. *Det kan se således ud på billedskærmen (Com-mode)*

Drive Search Path:

<i>1st Drive</i>	- <i>Default</i>
<i>Search Order</i>	- <i>COM</i>
<i>Temporary Drive</i>	- <i>Default</i>
<i>Console Page Mode</i>	- <i>On</i>
<i>Program Name Display</i>	- <i>Off</i>

Men hvad betyder de enkelte informationer? Et eksempel på en ændring af søgeruten vil være følgende kommando:

```
SETDEF A:
```

Så søger CP/M efter de ønskede filer på A:-stationen, selv når du arbejder på station B:. Kommandoen udvides således:

```
SETDEF A:,*
```

søger CP/M først på A:-stationen og så på den noterede station efter filerne. Aste-risken står for den noterede station. Efter indtastningen af SETDEF-kommandoen fås et uddrag af informationslisten.

Drive Search Path

1st Drive	- A:
2nd Drive	- Default

bliver du belært. Man kan lade informationerne køre ud på printeren, ved at trykke på <CTRL>-P tastkombinationen.

Skal midlertidige filer, altså mellemfiler som PIP f.eks. laver nogle af, på en særlig station, skriver du:

SETDEF [TEMPORARY=C:]

og mellemlerne bliver skrevet på en tredje station eller også på RAM-disk'en. De midlertidige filer kender CP/M på, at de har fået filnavnudvidelsen \$\$\$, men det har vi nævnt tidligere.

Der kan kun søges efter filer, der har en COM- eller SUB-udvidelse. Normalindstillingen ved CP/M 3.0 er søgningen efter COM-filer, der kan ændres således:

SETDEF [ORDER=(SUB,COM)]

SUB-filer kan kaldes frem med kommandoen SUBMIT og indeholder kommandoer, som udføres efter hinanden. Det kommer vi straks til.

5.11. SHOW

Nu til en kommando, der giver mange informationer om pladsen på dine disketter, navnene på disketterne, og antallet af filer pr. USER. Hvis du simpelthen indtaster:

SHOW

ser du alle diskettestationers attributter og den plads, der er tilovers på hver station. Herunder må utilities-disketten befinde sig i stationen

A:RW, Space: 101K
B:RW; Space: 5,934K

Angiver du SHOW med en stationsbetegnelse bagefter, får du kun de statistiske oplysninger om denne ene station.

Med SHOW kan du, som det er fortalt ovenfor, lade dine disketters labels vise:

SHOW A:[LABEL]

hvorunder du kan forkorte LABEL til et simpelt "L". På skærmen ser du så:

Label for drive A:

Directory	Passwds	Stamp	Stamp				
Label	Reqd	Create	Update	Label	Created	Label	Updated
BOGHOLD.COM	off	off	off	04/17/85	11:41	04/17/85	11:41

Med endnu en option til SHOW-kommandoen kan vises, hvilke USER-områder, der bruges på disketten, og hvor mange filer, der hører til hvert område. Denne option hedder (let at huske) USER. I tilgift får du vist antallet af ledige pladser i indholdsfortegnelsen

A: Active User: 0

A: Active files: 0 2 11 12

A: _ of files: 22 6 1 1

A: Number of free directory entries: 24

Hvis du indtaster:

SHOW A:[DIR]

får du vist antallet af ledige indholdsfortegnelsespladser, dvs. du får den sidste af de anførte linier på skærmen.

For CP/M 2.2-brugere: SHOW-kommandoen er nært beslægtet med CP/M 2.2-kommandoen STAT, som indeholder flere andre CP/M 3.0-kommandoer. Fra CP/M 2.2 er STAT-kommandoens funktioner under udviklingen af CP/M 3.0 blevet fordelt på flere CP/M 3.0-kommandoer. Herved er disse nye kommandoer blevet mere ydedygtige og betjeningsvenlige.

5.12. SUBMIT

Indtil nu har du indskrevet alle kommandoer ved hjælp af tastaturet. Det er altså meget logisk og godt, men når du skal indtaste de samme kommandoer for at opnå det samme igen og igen, bliver det i længden besværligt. CP/M kan befri dig for dette besvær.

Med kommandoen SUBMIT kan du udføre en række kommandoer, der står i en fil og vil blive udført på samme måde som indtastninger via tastaturet. En fil, hvori der står sådanne rækkefølger af kommandoer, bliver forsynet med filnavnudvidelsen SUB og kan læses og udføres af programmet SUBMIT.

Hvis din computer ikke har noget indbygget ur, der kører på et batteri, og det gælder vel 99% af Commodore 128'erne, må du efter starten indtaste tid og dato på ny. Hvis du vil tvinge dig selv til indtastning hver gang, for at tidsstemplerne skal blive sat rigtigt og kontinuerligt, kan du anvende filen PROFILE.SUB. Denne fil, PROFILE.SUB søges af CP/M efter enhver boot'ning eller reset, og finder det en sådan fil, bliver de i filen forekommende kommandoer udført som en SUBMIT-fil. Man kan sammenligne PROFILE.SUB med AUTOEXEC.BAT-filerne på IBM PC og de kompatible.

Hvis du angiver, at tid og dato skal indtastes, kommer du ikke uden om at gøre det. Filen skrives således:

B:DATE SET

med dit tekstbehandlingsprogram og døber denne fil PROFILE.SUB. Naturligvis kan du angive en anden station. Det afhænger alene af, på hvilken station filen DATE.COM befinder sig. Hvis du glemmer udvidelsen SUB, udfører SUBMIT-kommandoen ingenting.

I øvrigt: Hvis vi vil indtaste en linie, så er det muligt ved hjælp af editoren ED. Det er ikke sjovt at arbejde med denne editor, snarere en plage. Men der findes en anden mulighed for at skrive data fra tastaturet ind i en fil, hvorunder man dog ikke kan editere denne fil. Ved anvendelsen af PIP-kommandoen er det muliggjort:

PIP PROFILE.SUB=CON:

Vent indtil diskettesationen ikke bevæger sig mere. Så kan du indtaste linien og trykke RETURN. Hvis du vil indtaste flere linier, så skriv trøstigt videre. Som sidste line skal du indtaste <CTRL>-Z.

En SUB-fil kan indeholde CP/M-kommandoer, udviklede SUBMIT-kommandoer og indlæsningsdata for et program eller en CP/M-kommando.

Du kan arbejde i en SUB-fil med generelle anvisninger, der bliver anvendt i overensstemmelse med dine indtastninger. Disse generelle anvisninger kaldes parametre, og bliver fremstillet ved hjælp af et dollartegn (\$). Du kan anvende parametre fra \$1 til \$9.

Lad os antage, at du laver en fil, der ser sådan ud:

ERA \$1.BAK

DIR *. \$2

og som hedder DIR.SUB. Med den kommandorækkefølge i filen sletter du alle filer med et bestemt navn og med udvidelsen .BAK. Så lader alle filerne, med en bestemt udvidelse, sig vise. For at nå derhen indtaster du:

SUBMIT TEXT COM (Amstrad)

SUBMIT DIR TEXT COM (Commodore)

Commodore: DIR er her navnet på submitfilen, TEXT og COM er de parametre, der skal overgives til \$1 og \$2. Først bliver alle filer med udvidelsen BAK og indeholdende tegnene \$1 slettet. Derefter får du en oversigt over filer med udvidelsen COM. Filen SUBMIT.COM såvel som SUB-filen befinder sig på den samme disketteside. Ellers må man igen til at arbejde med den virtuelle station E:.

Amstrad: Sletter alle filer med navnet TEXT.BAK og får derefter en oversigt over alle filer med udvidelsen COM. Den oversatte SUBMIT-fil, det vil i praksis sige submit-filen med indsatte parametre, ser således ud:

ERA TEXT.BAK

DIR *.COM

Og CP/M udfører disse kommandoer, nøjagtig som de var indgivet ved hjælp af tastaturet.

Hvis du indtaster færre parametre, end der er forudset i SUBMIT-filen, bliver der ikke taget hensyn til de øvrige parametre. Indtaster du flere parametre, end der står i SUBMIT-filen, bliver der ligeledes ikke taget hensyn til de overtallige. Hvis du vil have et dollartegn til at stå indenfor en SUBMIT-fil, skriver du blot to dollartegn efter hinanden (\$\$).

En SUBMIT-fil kan ikke kun indeholde eenliniers kommandoer, men også kommandoindlæsninger til programmer. Et program kan kaldes frem med een kommandolinie, og med den næste give kommandoer videre til det kaldte program. Et eksempel:

PIP

⋄B:=A:*.COM

⋄

DIR *.COM

Her er en SUBMIT-fil, som indeholder noget nyt. Med den første linie kalder du PIP.COM, giver i den næste linie kommandoen til, at alle COM-filer skal kopieres på station B:; går så ud af PIP og betragter indholdsfortegnelsen. Alle kommandoer, der går direkte til et program, bliver angivet med et "mindre end"-tegn (⋄). Skriver du det, uden at tilføje mere, som i den tredje linie, betyder det RETURN, hvad der igen, i dette tilfælde, slutter PIP. *I øvrigt har brugerne af CP/M 2.2 det ikke ligeså let: vil man videregive parametre til et program, så er det ikke tilstrækkeligt med "mindre end"-tegnet. Her må man betjene sig af kommandoen XSUB (eller et lignende navn) for at nå det samme resultat. CP/M 3.0 på Commodore 128 fritager os for det arbejde*

SUBMIT-kommandoen er ikke kun til leg, men kan anvendes helt igennem fornuftigt. Du kan for eksempel lade printe et vilkårligt antal filer. Kommandofilen skriver du blot sådan:

PIP LST:=FIL1

PIP LST:=FIL2

PIP LST:=FIL3

PIP LST:=FIL4

osv.

eller samme resultat endnu lettere:

PIP LST:=FIL?

Du kalder filen SAMLFILR.SUB og indtaster:

SUBMIT SAMLFILR (RETURN)

Herunder kan filerne også stå på forskellige stationer. Du skal bare skrive stationsbogstaverne foran filnavnene, og SUBMIT sørger selv for resten.

Du kan skrive en kommandofil, der lister alle filer i alle USER-områder på en harddisk med DIR, og så skriver skærmudlæsningen i en ny fil med navnet "indhold". I den fil kan søges en bestemt fil med tekstbehandlingsprogrammets søgekommando eller lade fillisten printe ud.

Med SUBMIT-kommandoen er der ikke nogen grænser for brugeren, der kræves blot fantasi og nogle få programmeringsknep.

5.13. HELP-kommandoen

CP/M 3.0 er med sine mange kommandoer ikke så let at lære som sine forgængere. Særligt option'erne er svære at huske udenad. Heldigvis tilbyder driftssystemet sin hjælp, blot ved et tryk på en knap. Råb om hjælp på engelsk, og straks kommer, hvad du har brug for. Med "råbe" menes der naturligvis skrive på computerens tastatur. Mange nye programmer byder på hjælp i form af de såkaldte "Helpscreens". *Man behøver blot at meddele computeren (programmet), at man har behov for hjælp. Videre må man meddele, hvor denne hjælp er blevet nødvendig.*

Vi vil nu tage den venlige hjælp fra CP/M 3.0 i brug. Fordi vi endnu ikke rigtig ved, hvad det er, vi ikke ved, indtaster vi simpelthen ordet HELP=hjælp:

HELP

og får tilbudt en menu med de forskellige hjælpemuligheder. Så kan du udsøge dig et underpunkt og blive udførligere informeret.

I øvrigt: Commodore 128's tastatur råder over en HELP-tast, som når man kører i BASIC, i tilfælde af en fejl understreger denne fejl på skærmen. I CP/M 3.0 fremkommer efter et tryk på denne tast ordet HELP på skærmen, ganske vist uden automatisk RETURN, hvad der har den fordel, at du kan angive et stikord.

På skærmen fremkommer der med Amstrad følgende tekst, når du benytter HELP uden stikord:

HELP UTILITY V1.1

At "HELP" enter topic [,subtopic]...

EXAMPLE: HELP>DIR EXAMPLES

Topics available:

**COMMANDS CNTRLCHARS COPYSYS DATE DEVICE DIR
DUMP ED ERASE FILESPEC GENCOM GET
HELP HEXCOM INITDIR LIB LINK MAC
PATCH PIP (COPY) PUT RENAME RMAC SAVE
SET SETDEF SHOW SID SUBMIT TYPE
USER XREF
HELP>**

og med Commodore:

HELP UTILITY V1.1

At "HELP" enter topic[,subtopic]...

EXAMPLE: HELP> DIR EXAMPLES

Topics available:

*C128-MODE COMMANDS CNTRLCHARS COPYSYS DATE DEVICE
DIR DUMP ED ERASE FILESPEC GENCOM
GET HELP HEXCOM INITDIR KEYFIG LIB
LINK MAC PATCH PIP (COPY) PUT RENAME
RMAC SAVE SET SETDEF SHOW SID
SUBMIT TYPE USER XREF
HELP>*

Du kan kalde underpunkterne direkte, idet du indtaster:

HELP SETDEF

dermed kommer du direkte til hjælpeinformationerne. Men desværre er engelsk ikke alles modersmål. Vore venner, de amerikanske programmører går ud fra, at hele verden taler engelsk. Efter lang tids overvejelser og mange gennemtyggede stykker tyggegummi har CP/M's opfindere kæmpet sig frem til at skabe en mulighed for, at ikke-amerikanere kan hente hjælpeinformationerne frem på skærmen på andre sprog.

Hjælpeprogrammet består af filerne HELP.COM og HELP.HLP. For at ændre noget her, kalder du filen HELP.COM og indtaster som option EXTRACT:

HELP [EXTRACT]

Du kan også forkorte EXTRACT til et kort "E". Hjælpeprogrammet fremstiller så ud fra filen HELP.HLP en ny fil med navnet HELP.DAT, som du kan ændre med dit tekstbehandlingsprogram efter behag og til dit eget sprog. Gennem option'en EXTRACT er helpeteksterne ikke ligefrem blevet til dansk, men der er skabt mulighed for editering af teksterne.

For at indtaste nye hjælpetekster må du overholde følgende:

Ethvert stikord skal begynde med tre skråstreger (///) og et nummer. Nummeret angiver stikordets hjælpetrin. For eksempel:

```
///1DIR (ENTER (resp. RETURN))  
///2OPTIONS (ENTER)  
///3PARAMETRE (ENTER)  
///4EKSEMPLER (ENTER)
```

Hvis du har ændret det hele eller måske installeret en hjælp til anvendelsen af et lidet brugt program, så savel du filen og kalder igen HELP.COM. Denne gang med option'en CREATE, forkortet til "C". Derpå bliver oprettet en ny HELP.HLP-fil, der indeholder dine ændringer.

Hvem ved, måske er der snart nogen, der tilbyder hjælpetekster på dansk. Behovet er der bestemt.

5.14. Hvad du nu ved

- * Du har lært den vigtige kommando STAT i CP/M 2.2 og kan med den hente de forskellige informationer om dit system frem eller ændre dem.
- * Du har lært CP/M 3.0's midlertidige kommandoer.
- * Du kender et antal options, hvormed du kan forandre eller udvide de midlertidige kommandoers virkning.
- * Du ved, at du kan forsyne filer med en label.
- * Du har set, hvordan man beskytter en hel diskette, en fil eller en bestemt CP/M-kommando mod ubeføjet adgang.
- * Det står også klart, hvordan du kan forsyne filerne med et stempel om tid og dato.
- * Du har lært mulighederne i programmernes søgningsrute at kende.
- * Du ved, hvordan du kan lade en diskettes system-data vise med SHOW.
- * Du kan anvende filen PROFILE.SUB til at automatisere computerens startrutine.

6. Alt om PIP

Nu har du hørt nogle gange om det vidunder, der hedder PIP. For ikke at tro, at jeg overdriver hæmningsløst, skal jeg hurtigt resumere, hvad PIP kan.

- * Overføre en enkelt fil fra en diskette til en anden.
- * Overføre en gruppe af filer fra en diskette til en anden.
- * Kopiere en fil og forsyne den med et andet navn:
- * Formattere tekst til printning.
- * Forkorte for lange linier.
- * Printe en samlig filer på en kommando.
- * Sammenkopiere flere filer til en enkelt.
- * Hente et stykke ud af en tekstfil.
- * Forvandle små bogstaver til store og omvendt.
- * Sætte den ottende eller paritets-bit'en til nul igen.
- * Forsyne en fil med linienumre.
- * Vise en fil på skærmen under overførelsen.
- * Overføre systemfiler.
- * Kopiere filer fra et USER-område til et andet.
- * Automatisk sikre nyoprettede eller ændrede filer.

Nå, er det nok? I hvert fald grund nok til at beskæftige sig udførligt med dette program. Læs hellere dette kapitel og brugsanvisningen tre gange end to gange, og gør dig helt klart, hvad PIP alt sammen kan ordne for dig.

6.1. Diskettekopiering

Næsten enhver brugsanvisning til et eller andet program begynder med det råd først at overspille originaldisketten ved hjælp af PIP til en ny, formatteret diskette og så kun at bruge denne diskette for fremtiden. I et tidligere kapitel har jeg gjort opmærksom herpå og vist, hvorledes det kan lade sig gøre. Fordi PIP's tænkemåde er noget forskellig fra simple menneskers, kommer eksempelet endnu engang her. Således kopierer du fra en diskette til en anden:

PIP

```
*B:TEKST.TXT=A:TEKST.TXT
```

Dermed opnår du, at filen TEKST.TXT fra disketten i station A: bliver skabt endnu engang med det samme navn på station B. Din original forbliver, som den er; du har bare fået en kopi på en anden station.

For at kopiere en hel diskette, indtaster du:

PIP

B:=A:. *

Dermed skovler du alle filerne fra A: til B:, ganske vist uden at kopiere driftssystemet. Det må du skrive med en særlig kommando på hver diskettes systemspor. Kommandoen hedder:

SYSGEN ved CP/M 2.2 og

COPYSYS i CP/M 3.0 (i DISCKIT3!!)

Specielt for Commodore gælder: hvis du ikke har nogen B:-station, så kan du alligevel opnå denne kopiering af alle filer med kommandoen:

PIP E:=A:. **

ganske vist med noget mere besvær, da du hele tiden skal skifte disketter. Ellers er en kopiering fra diskette til diskette slet ikke mulig. I modsætning til andre CP/M-systemer, behøver du med Commodore slet ikke at kopiere systemet specielt med COPY-SYS eller SYSGEN; hvordan man kopierer systemet, har vi jo talt om.

Hvis du har kopieret begge sider af din originaldiskette, besidder du en fuldt anvendelig, dvs. også "bootbar" kopi af originalen. Du kan sætte kopien i stationen og starte computeren. For at forenkle og fremskynde den ovenfor fremstillede procedure findes to muligheder.

Kommandoen PIP kan forsynes med en hel række options, som har de mest forskellige virkninger. Een option hedder "V" og står for "verify", hvad der oversat vil sige "verificere" eller "kontrollere". Så snart PIP arbejder med den option, efterprøver den efter enhver kopieringsproces nøje, om den nye fil nu også svarer til originalfilen. Option'en må atter indtastes med firkantede klammer. For at kopiere og samtidig verificere filen TEXT.TXT indtaster du:

PIP

*B:=A:TEKST.TXT [V]

Det er besværligt først at skulle kalde PIP, og derefter give kommandoen i den anden linie. Set fra CP/M's synspunkt er det ikke nødvendigt. Du kan forkorte den ovenstående kommando således:

PIP B:=A:TEKST.TXT [V]

Så begynder PIP arbejdet, og vender tilbage til stikordet (A>). Bemærk, at jeg ikke har indtastet noget filnavn til station B:. Hvis filnavnet ikke skal ændres ved kopieringen, skal du gøre nøjagtig sådan; men hvis du vil ændre navnet, indtaster du:

PIP B:TEXT1.TXT=A:TEXT.TXT

Dermed kan man på en gang kopiere filer og ændre navnet.

Før du kopierer med PIP, bør du sikre, at der er tilstrækkelig plads. PIP overfører nemlig filen til den nye diskette og opretter en mellemfil, som har samme filnavn, men udvidelsen \$\$\$\$. Først når det ligger fast, at overførelsen var vellykket, ombenævner PIP mellemfilen til det rigtige navn.

Lad os antage, at du kopierer filen TEXT.TXT på diskette B:, men der står endnu, med det samme navn, den gamle version af denne fil. Hvad gør PIP så? Hvis disketten ikke har tilstrækkelig plads mere, får du en fejlmelding. Det ligger i PIP's arbejdsmåde. Det forsøger at skrive den fil, der skal kopieres, selv om den gamle version står der endnu. Deraf behovet for plads. Er der tilstrækkeligt med plads, bliver den nye fil overført og mærket med \$\$\$, derefter bliver den gamle fil slettet og den nye ombenævnt. Hvis din kopieringsoperation en gang ikke skulle lykkes, har du måske ikke plads nok til den nye fil og må først slette den gamle med ERASE.

Ved enhver overførelse ved hjælp af PIP, uden særlige options bliver fil-attributterne som SYS, DIR, RO og RW overført med. Hvis du kopierer en SYSTEM-fil med PIP, er kopien også en SYSTEM-fil.

Hvis du forlanger, at PIP skal overføre en fil, og der allerede på måldisken eksisterer en fil af samme navn, men skrivebeskyttet (RO=read only), så spørger PIP dig, om det må slette denne fil.

Svar på spørgsmålet med "Y" for ja og "N" for nej.

6.2. Kopiering mellem brugerområder

Hvad du til nu har læst, muliggjorde kun overførelsen af filer indenfor et USER-område. Hvis du befinder dig i USER-område 3 og anvender PIP til at kopiere en eller flere filer, så havner kopierne også i USER-område 3 på den anden diskette. Tænk på det, hvis du engang ikke skulle kunne finde dine overførte filer.

For nu at kunne kopiere fra en diskette og samtidig fra et brugerområde til et andet, bruger du option'en "Gn", hvor "n" står for USER-områdets nummer. For at kopiere filen TEKST.TXT fra brugerområde 0 til brugerområde 2 på station B: indtaster du:

PIP B:[G2]=A:TEKST.TXT

Tænk på, at andre CP/M-kommandoer som DIR, ERASE, TYPE osv. kun udfører noget i det udpegede USER-område. Der eksisterer ingen options, der overskrider brugerområder.

6.3. Tekst- og ikke-tekstfiler

Indtil nu er PIP benyttet til at bringe filer fra en diskette til en anden. Du kan også oprette en samlefil af flere enkelte filer, og det endda i een arbejdsgang.

Lige noget om de forskellige typer filer. Principielt skelner CP/M mellem to filformer: tekstfilen og ikke-tekstfilen. En tekstfil opretter du med dit tekstbehandlingsprogram eller den i CP/M indbyggede editor ED (helst ikke) og anvender hertil alle tegn, som dit tastatur sådan kan fremstille. En ikke-tekstfil består af binær kode og bærer i reglen udvidelsen COM.

Man må skelne mellem disse to filarter, fordi CP/M genkender afslutningen af en tekstfil på et Control Z (↑Z). Enhver tekst, der står bagved et ↑Z bliver ikke udtrykt eller på anden måde taget hensyn til. I ikke-tekstfiler derimod er ↑Z et ganske normalt tegn, og CP/M bekymrer sig ikke om dem.

Normalt går PIP ud fra, at det er ikke-tekstfiler, der skal overføres, og får hyppigt ret heri. Men hvis du ved hjælp af PIP sammenfatter flere filer til een, går PIP ud fra, at det drejer sig om tekstfiler. (Fordi det jo for det meste ville være ret urimeligt at ville sammenføje f.eks. COM-filer).

6.4. Sammenkopiering af filer

Som signal til PIP om, at det skal kopiere flere filer, tjener kommaet. Derfor må du ikke bruge noget komma i et filnavn. Ellers bringer du PIP helt i uorden.

En sammenlægning af forskellige filer gøres således:

```
PIP AL.TXT=DEL1.TXT,DEL2.TXT,DEL3.TXT
```

hvis alle filerne står på samme station. Skal samlefilen skrives i et andet USER-område, indtaster du ved CP/M 3.0:

```
PIP AL.TXT[G3]=DEL1.TXT,DEL2.TXT,DEL3.TXT
```

Insisterer du på en verificering efter kopieringen, må du indtaste [V] efter hver enkelt fil, der skal være en del af den nye fil. Dermed ser kommandolinien lidt mere kompliceret ud:

```
PIP AL.TXT[G3]=DEL1.TXT[V],DEL2.TXT[V],DEL3.TXT[V]
```

Under alle omstændigheder er det stadigvæk mindre arbejde end at skulle sammenføje alle filerne "i hånden".

Hvis du vil sammenføje ikke-tekstfiler med PIP, kan der opstå problemer. Som du har læst, tror PIP, hver gang der dukker et ↑Z op, at filen er slut. I ikke-tekstfiler kan ↑Z forekomme hyppigere og vil gøre en overførelse med PIP umulig.

Fordi CP/M's opfindere kendte problemet og gik ud fra, at de fleste COM-filer bliver overført som ikke-tekst-filer, overser PIP nådigt ethvert ↑Z i en COM-fil og kopierer til den bitre ende.

Hvis du vil kopiere filer, der hverken er tekstfiler eller COM-filer, skal du tillige anføre parameteren "O". Som før med firkantede klammer direkte bagved filnavnet.

Du har endda mulighed for, under sammenføjjningen af flere filer til een fil, at skrive noget til. Så skal kopieringskommandoen se sådan ud:

```
PIP AL.TXT=DEL1.TXT,CON:.,DEL2.TXT,CON:.,DEL3.TXT
```

Så begynder PIP at overføre den første fil, standser og venter på din indtastning, inspireret af kommandoen CON:.

Du må indtaste kommandoen for ENTER og "ny linie" (↑J) med håndkraft, for at PIP ikke skal skrive nogen linier oven i hinanden. Så fortsætter du med at indtaste ↑Z. Husk på, at anslagsfejl ikke kan rettes.

6.5. Linienummerering

Et smart påfund for programmører, journalister og alle, der gerne vil have deres tekster eller programmer nummereret linievis, er PIP's "N"-parameter. Når du kopierer en fil med denne parameter eller dens bror "N2", får hver linie sit eget nummer. Hvis du kopierer således:

```
PIP TEKSTNR.TXT=TEKST.TXT[N]
```

kommer dine tekstlinier til at se sådan ud:

- 1: *Dette er din tekst*
- 2: *Nummereret linievis*
- 3: *Praktisk, ikke sandt?*

Bruger du derimod parameteren "N2", kommer det til at se sådan ud:

- 000001 *Dette er din tekst*
- 000002 *Nummereret linievis med N2 som option*
- 000003 *Heller ikke dårligt, vel?*

Linierne bliver talt fortløbende, hvad der ikke lader sig ændre. Altså i rækkefølgen 1,2,3,4,5...

6.6. Ændring af bogstaver

Hvis du i stedet for at bruge linienumre hellere vil lade hele din tekst fremkomme med store bogstaver, er du kandidat til parameteren "U". "U"et" står for "uppercase",

der betyder store bogstaver. Det modsatte får du, når du indtaster parameteren "L" for "lowercase". Så består din tekst af bar små bogstaver. Hvad man kan bruge det til? For eksempel til udprinting af lister.

6.7. Søgning af en streng

Det sker, at printeren strejker, eller at papiret er brugt op midt under printningen. Før du nu printer alle 200 sider ud en gang til, skulle du hellere benytte endnu en PIP-parameter til at få resten af teksten på papiret.

Du kan kopiere en del af en tekst med PIP, idet du meddeler PIP, fra hvilken streng, dvs. kæde af tegn, det skal søge, og ved hvilken streng det atter skal holde op. Begyndelsesstrengen markeres med et "S" for "Start" og slutstrengen med et "Q" for "Quit", hvad betyder "ophøre". Skriv efter hver række af tegn et ↑Z, og PIP gør resten. Det gennemsøger din tekst, til det støder på den angivne tegn række, kopierer og standser ved den anden angivne tegn række. Kommandoen ser således ud:

PIP

```
*TEKSTDEL=TEKST.TXT[Qsøgeord↑Z]
```

Når du skriver sådan, begynder PIP at kopiere ved filens start og stopper, så snart "søgeord" dukker op. Bemærk, at PIP blev kaldt uden nogen option, og alt det andet blev indtastet i den anden linie. Så leder PIP efter søgeordet, med små bogstaver.

Hvis du skriver den ovenstående kommando i een linie, forvandler PIP først "søgeord" til store bogstaver og begynder så at søge. Står dit "søgeord" ikke som "SØGEORD" i din fil, finder PIP ingenting og giver i stedet en fejlmedling. *Denne fejlmedling lyder, til oplysning for alle der ikke selv vil prøve det:*

```
ERROR: START NOT FOUND eller:  
ERROR: QUIT NOT FOUND
```

Et tekstafsnit kopierer du med denne kommando:

PIP

```
*TEKSTDEL=TEKST.TXT[Sbegynd↑ZQslut↑Z]
```

Programmet begynder at kopiere ved ordet "begynd" og hører op dermed ved den første forekomst af ordet "slut". *På den måde kan du f.eks. kopiere tekster fra programlistninger, måske en procedure fra et PASCAL-kildeprogram.*

6.8. Printning af flere filer efter hinanden

Du kender til metoden at printe en fil ud på, hvor du indtaster ↑P og med kommandoen TYPE omdirigerer skærmudlæsningen til printeren. Det er noget omstænde-

ligt og kan løses bedre ved hjælp af PIP. Ethvert ydre stykke apparatur har for PIP sit eget navn. Udlæsningsapparatet bliver angivet for PIP med LST:. Du erindrer, at vi allerede en gang har kaldt indlæsningsapparatet (tastaturet) med CON:. *Kolon'et er vigtigt, for at PIP kan skelne et apparatnavn fra et filnavn. Hvis du vil have en tekstfil frem på printeren, indtaster du*

PIP LST:=TEKST.TXT

Du kan naturligvis også anvende alle options, der har indflydelse på PIP ved denne operation. For at printe flere filer efter hinanden skriver du:

PIP LST:=TEKST1.TXT,TEKST2.TXT,TEKST3.TXT

og teksterne kommer sprøjtende ud af din printer. Hvis du vil have indflydelse på den form, teksten bliver printet ud i, skal du benytte navnet PRN:. Så får du gennemnummererede tekstlinier, otte tegn brede tabulatortrin, og for hver 60 linier bliver der begyndt på en ny side.

Ved slutningen af dette kapitel vises nogle eksempler på, hvad man hensigtsmæssigt kan lave med PIP.

6.9. Automatisk sikring af filer

Nu er det ikke kun meget praktisk at kunne printe data, men ligeså vigtigt at sikre data. Jeg har allerede gjort dig opmærksom på det og taler af smertelig erfaring. På mine vandringer i PIP-brugsanvisningen har jeg opdaget en PIP-funktion, der gør det væsentligt lettere at sikre. Det er "Archiv"-option'en, forkortet "A".

Også her finder man kun ved hyppig prøven sig frem, ud af, hvordan man kan anvende en sådan option.

Særligt når du arbejder med en hard-disk, vil du hurtigt blive klar over denne parameters fordele. Hver fil i CP/M 3.0 har i fil-"hovedet" plads til en bit, der betegnes som "archive flag". Et flag er noget i retning af en markør, der angiver en bestemt tilstand. Hver gang du åbner eller ændrer en fil, bliver dette flag forandret. Altså for eksempel sat til "1" ved ændring af en fil og efter fuldført sikring af dataene igen sat til "0".

På den måde kan PIP kende, hvilke filer på en diskette, der er blevet forandret eller nyoprettet, og kun sikre dem. Dermed bliver du sparet for, hver gang at skulle sikre hele hard-disk'en, hvad der ville koste en bunke tid og penge til ekstra disketter.

Efter en lang arbejdsdag sikres de bearbejdede disketter ved at indtaste:

PIP B:=A:*.TXT[A]

Hvis du vil helgardere dig, tilføjer du også "V"-option'en, og hver fil bliver kontrolleret for fuldstændig overførelse.

```
PIP B:=A*.TXT[AV]
```

Hvis du ikke rører ved originalerne, til de sikrede filer på hard-disk'en, bliver disse ikke påvirket af den næste sikring.

Du kan se i indholdsfortegnelsen, hvilke filer der er sikret og hvilke der ikke er. Dertil benytter du kommandoen DIR med option'erne FULL eller RW. I indholdsfortegnelsen ses de sikrede filer mærket med "Arcv".

6.10. Overskrivning uden forespørgsel

Der er endnu et par detaljer at lægge mærke til, når du arbejder med PIP. Normalt skriver PIP uden at tøve henover en fil på måldisken, når den har det samme navn, som den nye fil. Drejer det sig imidlertid om en fil, der er beskyttet ved hjælp af attributtet R/O (read only), spørger PIP endnu en gang, om der godt må skrives oven i denne fil:

```
DESTINATION FILE IS R/O, DELETE (Y/N)?
```

Skriver du nu "Y" for ja, bliver der skrevet oven i filen på måldisken, og den bliver altså slettet. Hvis du fastholder dit nej ved at trykke "N", afbryder PIP kopieringsfunktionen. Efter "Y" eller "N" behøver du i øvrigt ikke at trykke RETURN.

Ønsker du, at alle filer skal skrives på måldisken, og der ikke skal spørges om lov til at slette, så anvender du "W"-parameteren.

Kommandoen ser således ud:

```
PIP B:=A:TEKST1.TXT,TEKST2.TXT,TEKST3.TXT[W]
```

men dermed overtager du også hele ansvaret for kopieringen og kan ikke komme med nogen bebrejdelser mod PIP. (Hvad der også ville være meningsløst.)

6.11. Kopiering af systemfiler

Hvis du forsøger at kopiere en fil, der besidder SYSTEM-attributtet, bringer du PIP i forlegenhed. Det kan nemlig ikke finde det. Du må hjælpe det lidt og give det "R"-option'en med på vejen. Du kan kombinere de to options. Hvis filer og SYSTEM-filer skal skrives på måldisken uden hensyn til, om de er skrivebeskyttet, indtaster du:

```
PIP B:=A:*.COM[RW]
```

Dermed kopieres alle COM-filer og skriver oven i filer med samme navn på måldisken. Du erindrer nok, at systemfiler er filer, der ikke dukker op i den "normale" indholdsfortegnelse. De vises kun med kommandoen DIRSYS. Systemfiler er tilgængelige i alle brugerområder.

6.12. "Rensning" af bit 8

Det amerikanske tegnsæt (ASCII) bliver repræsenteret ved hjælp af syv bits. Den ottende bit, det er den i computeren anvendte "ordlængde", holdes fri til særlige opgaver. WordStar for eksempel anvender den til at markere mellemrummene, når der skrives i blokke. På den anden side kræver MBASIC, at den ottende bit skal være fri. Hvis du bearbejder et BASIC-program med WordStar og ved en fejltagelse har valgt dokument-funktionsmåden (D), kan dit BASIC-program ikke køre, fordi den ottende bit ikke er fri. Det problem klares, når du anvender PIP's "Z"-option. Du kopierer din fil med PIP, og alt er igen i den skønneste orden. Kommandoen dertil skriver du sådan:

```
PIP SPIL.BAS=SPIL.BAS[Z]
```

Hvis du vil, kan du også tilføje "V"-option'en hertil; men - benyt ikke kommandoen i denne form til at kopiere WordStar-filer. Ellers taber du tekstens formattering, fordi ordmellemrummene, resp. markeringen af dem bliver ødelagt.

6.13. Praktiske eksempler

Nu et par eksempler på, hvordan man hensigtsmæssigt kan indsætte PIP. De forskellige parametre kender du og ved også, hvordan man kan anvende dem.

En tekstudprintning i bedre form end blot med ↑P eller et simpelt LST: giver denne kommandoform:

```
PIP LST:TEKST.TXT[NT8P60]
```

Dermed bliver filen TEKST.TXT sendt til printerens, alle linier bliver gennemnummereret, der er tabulatorstop i hver ottende spalte, og sidelængden er på 60 linier. Hvis du bladrer lidt tilbage, vil du se, at det nøjagtigt svarer til PRN:'s indstillede værdier. Nu ved du, hvad ideen med PRN: er: mindre arbejde!

Hvis du i det ovenstående eksempel tillige vil sørge for, at alle bogstaver fremkommer som små bogstaver, skriver du:

```
PIP LST:TEKST.TXT[NT8P60L]
```

Du tilføjer altså blot "L"-option'en, og straks er det ordnet.

For at rationalisere arkiveringsarbejdet, skriver du i CP/M 3.0 en SUBMIT-fil af det følgende indhold

```
PIP A:=B:*.[WAR]
```

og kalder denne fil ARKIV.SUB. Option'en "WAR" bevirker, at SYSTEM-filer med kopierede, skrivebeskyttede filer uden forespørgsel bliver skrevet henover, og kun sådanne filer bliver kopieret, som endnu ikke er arkiveret. Hvis du rutinemæssigt anvender denne kommando, kan der ikke ske noget med dine filer. Kaldet sker med

```
SUBMIT ARKIV
```

og resten går af sig selv. Du kan også udvide kommandoen. Med option'en "V" bliver det verificeret, om filerne er overført rigtigt, og med parameteren "E" får du vist alt det kopierede på skærmen. "E" bør du kun anvende ved tekstfiler, da du ellers kommer i vanskeligheder.

Videre muligheder vil være at optage DIR- og SHOW-kommandoerne i SUBMIT-filen. Så kan du se, hvilke og hvor mange filer der skal kopieres, og hvor meget plads der endnu er på måldisken.

På Commodore 128 har man - blot på grund af 40- og 80-tegns skærbilledet - flere apparaturkanalbetegnelser til rådighed. Det er de følgende

KEYS = Commodore 128's tastatur

40COL = 40-tegns monitor

80COL = 80-tegns monitor

PRT1 = Serieprinter på IEC-bussen (apparaturnummer 4)

PRT2 = Serieprinter på IEC-bussen (apparaturnummer 5)

Følgende indstillinger til ind-/udlæsning er standard:

CONIN: = KEYS

CONOUT: = 80COL (40COL)

AUXIN = Null Device

AUXOUT = Null Device

LST: = PRT1

Som du kan se, sker printerudlæsning standardmæssigt på serieprinter på IEC-bussen, apparaturadresse 4.

6.14. Hvad du nu ved

- * PIP er et af de mest effektive under-programmer i CP/M.
- * Du kan overføre enkelte filer til en anden diskette.

- * Du kan kopiere hele disketter på een gang.
- * Med PIP kan du frembringe en enkelt fil ud af flere.
- * Du kan hente et stykke ud af en fil.
- * PIP nummererer automatisk linierne i en tekstfil for dig.
- * Med PIP kan man automatisk sikre ændrede filer.
- * Du kan forvandle store bogstaver til små og omvendt.
- * Man kan kopiere mellem de forskellige brugerområder.

7. CP/M internt

7.1. Fremstilling af en sikkerhedsdiskette med CP/M

Før vi kaster os over CP/M, bør du først lave en sikkerhedskopi af din CP/M-diskette. Hvor let kan det ikke ske, at man ved en fejl sletter en fil på disketten, eller at hele disketten bliver ødelagt på uforklarlig vis. (Og tro mig, det forekommer oftere, end det passer dig eller mig.)

For at forhindre, at man roder rundt efter en diskette, der er parat til at køre, bør man lave en kopi, også kaldet backup, af den forhåndenværende CP/M-systemdiskette.

Du ved allerede, hvordan man formatterer en diskette, og hvordan man kopierer enkelte filer eller hele disketten. Hvis du kun har een diskettestation til rådighed, så anvender du kommandoen:

```
DISKCOPY
```

Den rutine formatterer om nødvendigt måldisketten for dig og meddeler dig på skærmen, om du skal indsætte kildedisketten (Source) eller måldisketten (Destination).

Du skal skifte disketten nogle gange under kopieringsprocessen. Hvis du har rådighed over to diskettestationer, anbefales kommandoen COPYDISC, der arbejder hurtigere, fordi det sker uden disketteskift.

Efter du har frembragt een eller to sikkerhedskopier af CP/M-disketten, kan vi begynde.

7.2. Amstrad diskettestationstyringens disketteformater

Amstrad diskettestationstyringen (Amstrad-Floppy) har tre forskellige disketteformater: standard-CPC-formatet, dataformatet og IBM-formatet. Disse tre formater har visse ting tilfælles:

- Der formatteres 40 spor (spor 0 til 39).
- Sektorlængden er på 512 bytes.
- Der kan være 64 optagelser i disketteindholdsfortegnelsen.

Standard-CPC-formatet og dataformatet bliver formatteret hver med ni sektorer pr. spor. De to formater afskiller sig ved, at de to første spor i dataformatet ikke bliver belagt med CP/M, og således er til rådighed til program- og dataopbeva-

ring. IBM-formatet afviger herfra, ved at der kun bliver formatteret otte sektorer pr. spor. Som anvender af driftssystemet CP/M kommer dataformatet altså ikke på tale, eftersom de vigtige spor 0 og 1 ikke bliver forsynet med CP/M.

Vi vil begrænse os til standard-CPC-formatet, da dataformatet ikke har noget at gøre med CP/M, og IBM-formatet (og andre) bliver beskrevet senere. I dette format bliver de to "øverste" spor 0 og 1 anvendt på følgende måde:

- Track 0, Sektor &41 : Boot-sektor.
- Track 0, Sektor &42 : Konfigurationssektor.
- Track 0, Sektor &43 til &47 : Ikke benyttet.
- Track 0, Sektor &48, &49 og
Track 1, Sektor &41 til &49 : CCP og BDOS.
(& indleder hexadecimal tal)

CCP betyder Console Command Processor, BDOS står for Basic Disk Operating System.

Du ved, at du kan formattere dine disketter med den midlertidige kommando FORMAT. Med CPC'en kan du kun formattere disketter i station A:. Da vi har forskellige formater, må du vælge, i hvilket format CP/M 2.2 skal formattere disketter. Der er følgende muligheder:

- FORMAT : Formatterer i standard-CPC-format
- FORMAT I : Formatterer i IBM-format
- FORMAT V : Formatterer i Vendor-format
- FORMAT D : Formatterer i dataformat

Hvis du forsøger med en anden parameter end (S),I,V eller D, så vil FORMAT reklamere over den parameter. (S) betyder, at du ved formattering i system-formatet kan udelade S'et.

7.3. Fremmede disketteformater

Dog, måske er det af interesse at kunne læse, henholdsvis skrive andre formater. Bedst vil det være også at kunne formattere de fremmede formater på CPC'en. Vi har lært, at hos Amstrad omfatter sektorerne 512 bytes. Da diskettestationsstyringen er frit programmerbar, kan man forandre den værdi. Som sektorstørrelser har man 128, 256, 512, 1024 osv. bytes. Hvordan man i det enkelte kan programmere diskettestationsstyringen (PD 765A), er udførligt beskrevet i "Floppy-bogen" til CPC.

Til diskettestationsstyringen kan der tilsluttes to diskettestationer, nu også 5¹/₄-tomme disketter. Egentlig kun interessant for de læsere, der har rådighed over en 5¹/₄-tommes station, da det kun er den, der kan indlæse andre computers CP/M-software og data. Amstrad CPC er den første computer med 3 tommers

stationer.) Men også de læsere, der ”kun” råder over 3-tommers-stationen, turde være forbavsedede over det antal muligheder, som den byder på med sin diskettes-tationsstyring.

Først må det fastslås, at der er hundreder af formater i CP/M for 5¼-tommers disketter. Vi giver dig her en tabel, som firmaet Digital Research har sammenstillet. Tabellen giver oplysninger om de enkelte CP/M-formaters parametre og gør det muligt realisere disse andre formater på CPC'en. Vi begrænser os til formaterne for enkeltsidige 5¼-tommers disketter, da 8 tommers disketter vel næppe eksisterer i større antal længere.

D	BLS B/S	SPT Cap.	BLM BSH	DSM EXM	AL0/1 DRM	OFF CKS	PHM PSH	Sec1 VR	Skew Letz	Nr.
S 128	1	83	18 3 7 0	82	31	80/0	8 3 0 0	0 1	18 5	<1>
S 128	1	83	18 3 7 0	82	63	C0/0	16 3 0 0	0 1	18 4	<2>
S 256	1	92	20 3 7 0	91	63	C0/0	16 3 1 1	0 1	10 1	<3>
S 256	2	92	20 4 F 1	45	63	80/0	16 3 1 1	0 1	10 2	<4>
D 128	1	123	30 3 7 0	122	63	C0/0	16 2 0 0	0 1	30 1	<5>
D 256	1	144	32 3 7 0	142	63	C0/0	16 4 1 1	0 1	16 1	<6>
D 256	1	148	32 3 7 0	147	63	C0/0	16 3 1 1	0 1	16 1	<7>
D 256	1	152	32 3 7 0	151	63	C0/0	16 2 1 1	0 1	16 1	<8>
D 256	1	152	32 3 7 0	151	63	C0/0	16 2 1 1	0 1	16 2	<9>
D 256	1	157	34 3 7 0	156	63	C0/0	16 3 1 1	0 1	17 1	<10>
D 256	2	171	36 4 F 1	84	63	80/0	16 2 1 1	0 1	18 2	<11>
D 256	2	171	36 4 F 1	84	127	C0/0	32 2 1 1	0 0	17 1	<12>
D 512	1	153	32 3 7 0	155	63	C0/0	16 1 2 3	0 1	8 1	<13>
D 512	1	152	32 3 7 0	151	63	C0/0	16 2 2 3	0 1	8 1	<14>
D 512	1	171	36 3 7 0	170	63	C0/0	16 2 2 3	0 1	9 2	<15>
D 512	1	190	40 3 7 0	189	63	C0/0	16 2 2 3	0 1	10 1	<16>
D 512	1	195	40 3 7 0	194	63	F0/0	16 1 2 3	0 0	9 1	<17>
D 512	2	166	36 4 F 1	82	63	80/0	16 3 2 3	0 1	9 1	<18>
D 512	2	190	40 4 F 0	94	63	80/0	16 2 2 3	0 1	10 2	<19>
D 512	2	190	40 4 F 1	94	63	80/0	16 2 2 3	0 1	10 1	<20>
D 512	2	190	40 4 F 1	94	63	80/0	16 2 2 3	0 1	10 2	<21>
D1024	1	185	40 3 7 0	184	63	C0/0	16 3 3 7	0 1	5 1	<22>
D1024	2	160	40 4 F 1	78	63	80/0	16 3 3 7	0 1	5 1	<23>

(Kilde: C'T 6/85, Verlag Heinz Heise GmbH)

En noget uoverskuelig tabel. Ganske vist indeholder den vigtige informationer, uden hvilke vi, ikke kan gennemføre vort forehavende.

Vigtigt for dem, der er interesseret i format <21>: efter læsningen må dataene først komplementeres. Det sker ved at XOR'e med &FF.

Her er listen over fabrikater til bedre identifikation af de forskellige formater:

- <01>: Xerox
- <02>: Lifeboat R2, TRS-80, Mayon
- <03>: Eurocom
- <04>: Osborne SD
- <05>: Superbrain
- <06>: MC-CP/M Ecma-70, MC-CP/M+ _7
- <07>: Eurocom II Format 2
- <08>: Alphatronics DD, NEC PC 8001
- <09>: Olympia ETX-II, Philips P-2000
- <10>: Spectravideo
- <11>: TRS-M4
- <12>: Bondwell 12
- <13>: IBM CP/M-86
- <14>: ADPS
- <15>: Dec VT-180
- <16>: Rentiki
- <17>: Kaypro II
- <18>: Olympia Boss A
- <19>: SEL Delsy 2000
- <20>: Newbrain, Mayon
- <21>: Superbrain, HKM-ZDOS
- <22>: Osborne DD
- <23>: BASF-7120

Forkortelserne i den første og anden linie har hver følgende betydning:

D=Density. I den spalte finder du formatets "densitet" (dvs."tæthed"), altså S=Single Density (enkelt densitet) og D=Double Density (dobbel densitet). Det er et spørgsmål om tid, før man møder CP/M-formater med firdobbel densitet. De tilsvarende disketter med påskriften Quad-Density eksisterer allerede.

B/S betyder Bytes pr. Sektor. Vi omtalte allerede, at CPC'en lagrer 512 bytes pr. sektor. 128, 256 eller 1024 bytes pr. sektor kan imidlertid også realiseres.

BLS betegner blokstørrelsen (Block Size) i K bytes. Ved et format med 128 bytes pr. sektor og en BLS på 1K byte sammenfattes 8 sektorer til en blok. I CPC'en består en blok af 2 sektorer, altså af 1K byte.

Cap. står Capacity=kapacitet. Her angives det formats lagringskapacitet i K bytes. CPC'en har en kapacitet på 169K bytes.

SPT vil sige Sectors per Track. Man må ikke opfatte ordet sektor fysisk, men forstå det logisk. Da man i begyndelsen havde sektorer på blot 128 bytes, sammenfattede man disse 128 bytes til en "record". SPT angiver, hvor mange af disse records, altså 128-byte enheder, der kan være på eet spor.CPC'en har 9 sektorer på 512 bytes hver, det giver $9 \cdot 512 = 4608$ bytes pr. track. Dividerer vi dette tal, 4608 med 128, så får vi vort resultat SPT: $4608/128 = 36$.

BSH og BLM står for Block-Shift og Block-Mask. Disse to oplysninger fastlægger blokstørrelsen. Controlleren er nødt til at have disse oplysninger.

EXM vil sige Extent-Mask og fastlægger antallet af optagelser i indholdsfortegnelsen.

DSM fastlægger det størst mulige bloknummer.

Under DRM findes det maksimale antal af optagelser i indholdsfortegnelsen (-1!).

AL0/1 står for indholdsfortegnelsesstørrelse, kodet binært. I CPC'en holdes to blokke fri til indholdsfortegnelsen.

CKS angiver, hvor mange logiske sektorer (records) indholdsfortegnelsen består af. I CPC'en er det 16.

OFF står for sporoffset. Her befinder sig antallet af spor, som er reserveret til CP/M.

PSH og PHM giver informationer om blocking og deblocking af fysiske sektorer i logiske sektorer.

Sec1 beskriver den første sektors nummer, lezt den sidste sektors nummer på et spor. I CPC'en er den første sektor, sektor nr. 1, den sidste har nummer 9.

Sidst har vi også en angivelse af den såkaldte skew-faktor SKEW. Kun af interesse for meget viderekomne.

Efter at vi har fået disse oplysninger, vil vi se, hvilke informationer, man kan meddele CPC'en, eller snarere dens Controller PD 765A. Til den ende ser vi på CPC-floppyens systemlager:

Du finder oplysningerne i system-RAM'en på adresserne &A890-&A8A5.

A890,A891	SPT	Records pr. Track (36)
A892	BSH	Block Shift (3)
A893	BLM	Block Mask
A894	EXM	Extend Mask
A895,A896	DSM	Maksimale bloknummer
A897,A898	DRM	Maksimale antal optagelser i indholdsfortegnelse-1 (63)
A899,A89A	AL0/1	Indholdsfortegnelsesstørrelse (C000h), kodet binært, svarer til to blokke
A89B,A89C	CKS	Antal optagelser, der kan afprøves i indholdsfortegnelsen (&0010), 16 optagelser
A89D,A89E	OFF	Sporoffset (2), optagne systemspor
A89F	FSC	Første sektor af hvert spor (041h)
A8A0	PST	Fysiske sektorer pr. track (9)
A8A1	GPS	Længde af GAP3 ved sektoren Read/Write (2ah)
A8A2	GPT	Længde af GAP3 ved formattering af track

A8A3	FLB	Filler-byte ved formattering af track
A8A4	BPS	Bytes/sektor (2), svarer til 512 bytes
A8A5	RPS	Antal records/sektor

Værdierne i klammer angiver de noterede (default-) værdier, altså de forudindstillede værdier.

Format <13>, IBM-formattet, er det eneste, som CPC'en automatisk giver dig rådighed over.

Det er ikke noget problem at realisere et andet format. Man kan programmere controller'en til det hele. Den eneste vanskelighed, er valget af længden på GAP3. GAP3 er et "hul", som anvendes ved disketter, for at udligne små hastighedsvariationer i disktestationen, idet stationen ved skrivning af en sektor med en til vished grænsende sandsynlighed, ikke har samme hastighed som ved formatteringen af denne sektor. Der er så fare for, at man med en langsommere hastighed, kommer til at skrive ind i den (fysisk) næste sektor. Her tjener GAP3-hullerne som stødpuder. Jo større dette GAP3 er, desto færre sektorer kan der være på et spor. Hvis man vælger GAP3 for lille, stiger faren for, at skrive ind i en følgende sektor. Et GAP3 er en række af &4E-koder efter hinanden. Ethvert andet tegn vil være ligeså tænkeligt.

Lad os en gang prøve at realisere Osborne-formatet <22> på vor Amstrad.

De følgende vigtige kendsgerninger kan vi finde i vore tabeller:

1024 bytes pr. sektor, 5 sektorer pr. spor, sektornummer 1 til sektornummer 5. Disse og andre informationer må vi skrive i vor computers systemlager. Vigtigt er det, at de fastsatte 16-bit værdier også lagrer 16 bits. Ellers forskubbes al informationen i Floppy-controlleren, og man opnår ikke det ønskede resultat.

Efter systemlageradresserne &A890-&A8A5, er der skabt basis for at anvende et andet format. Som du kan se af maskinsprogsprogrammet på de følgende sider, kan man kopiere data ind i systemlageret med en tabel og LDIR-kommandoen. Som før anvendes systemrutinerne. De forløber gnidningsløst, som vort program eksempel viser. Først bliver de 40 spor formatteret med hver 5 sektorer. Herefter at dette er sket, skrives der forsøgsvis på alle sektorerne. Det er en test, der skal vise, om alle sektorerne er til rådighed. Det kunne jo være, at GAP3 var blevet valgt for stor eller for lille, eller der er noget andet, der ikke er klappet, som det burde (sådan var det naturligvis også under programudviklingen). Når du ser, hvor hurtigt floppy'en formatterer de 40 spor, og hvor hurtigt der igen skrives på alle spor, så kender du disktestationens sande ydeevne.

Her kommer først assemblerprogrammet til formattering i Osborne-format:

```

10
20 ;formattering i Osborne-format, JS 9/6/85
30 ;
40 ;
50      ORG      _7000      ;startadresse
60 ;
70 ;initialiserer FDC med nye værdier:

```

```

80 ;
90 LD BC,22 ;22 VÆRDIER
100 LD HL,FDC ;TABEL OVER DE NYE VÆRDIER
110 LD DE,_A890 ;FLOPPYSYSTEMLAGERET
120 LDIR
130 ;
140 LD E,0 ;DRIVE
150 LD D,0 ;TRACK
160 LD B,40 ;40 TRACKS
170 L1: PUSH DE ;GEM DE
180 PUSH BC
190 LD HL,FTAB ;TABEL TIL FORMATTERING
200 LD A,D ;TRACK=>D
210 LD B,5 ;5 SEKTORER
220 L0: LD (HL),A ;LAGERSPOR
230 INC HL
240 INC HL
250 INC HL
260 INC HL ;VISER TIL NÆSTE SEKTOR
270 DJNZ L0 ;NÆSTE SEKTOR
280 LD C,_1 ;FØRSTE SEKTOR
290 LD HL,FTAB ;TABEL TIL FORMATTERING
300 RST _18
310 DEFW FORMAT ;FORMATTER SPOR
320 POP BC
330 POP DE
340 INC D ;NÆSTE SPOR
350 DJNZ L1
360 ;
370 JP TEST ;GENNEMPRØV ALLE SEKTORER
380 ;
380 ;
390 ;
400 FORMAT DEFW _C652 ;ADRESSE _C652
410 DEFB 7 ;I FLOPPY ROM'EN
420 ;
430 FTAB: EQU $
440 ;
450 DEFB 0 ;TRACK
460 DEFB 0 ;STATIONENS HOVEDADRESSE
470 DEFB 1 ;SEKTOR
480 DEFB 3 ;3=1024 BYTES
490 DEFB 0,0,3,3 ;SEKTOR 3
500 DEFB 0,0,5,3 ;SEKTOR 5
510 DEFB 0,0,2,3 ;SEKTOR 2
520 DEFB 0,0,4,3, ;SEKTOR 4
530 ;
540 ;
550 ;TABEL OVER CONTROLLER-VÆRDIERNE

```



```

560 ;
570 FDC:  DEF 40,0,3,7,0,184,0,63,0, _C0,0,16
580      DEF 0,3,0,1,5,42,50, _E5,3,8
590 ;
600 ;SKRIV FORSØGSVIS PÅ ALLE SEKTORER
610 ;
620      ORG      _7100      ;NY STARTADRESSE
630 ;
640 TEST: CALL  _BB06      ;VENT PÅ TASTTRYK
650      LD      E,0        ;DRV
660      LD      D,0        ;TRACK
670 LL1  LD      C,5        ;SEKTOR
680      LD      B,5        ;TÆLLER
690      LD      HL, _5000   ;BUFFER, DER LAGRES
700 LL2: RST      _18
710      DEFW    WRITE      ;SKRIV EN SEKTOR
720      JR      NC,FEJL    ;FEJL OPTRÅDT
730      DEC     C          ;NÆSTE SEKTOR
740      DJNZ   LL2
750      INC     D          ;NÆSTE SPOR
760      LD      A,D
770      CP      40         ;ALLE 40 SPOR??
780      JR      NZ,LL1    ;ENDNU IKKE ALLE
790      RET
800 ;
810 WRITE: DEFW  _C64E
820      DEF     7          ;ADRESE _C64E I FLOPPY-
                        ;ROM'EN
830 ;
840 FEJL: EQU S
850      LD      A,7        ;BEEP
860      CALL   _BB5A      ;UDLÆS
870      RET              ;AFBRYD STYRINGEN
880 ;
890 ;
900      ORG     _7200      ;NY STARTADRESSE
910 ;
920      LD      E,0        ;DRIVE
930      LD      D,1        ;SPOR
940      LD      C,2        ;SEKTOR
950      LD      HL, _5000   ;BUFFER
960      RST     _18
970      DEFW    READ      ;LÆS SEKTOR
980      RET              ;SLUT PÅ PROCEDUREN
990 ;
1000 READ: DEFW  _C666
1010      DEF     7          ;ADRESSE _C666 I FLOPPY-
                        ;ROM'EN
1020 ;

```

Du må forstå, hvorfor vi ikke trykker en BASIC-LOADER på dette sted - det ville være meningsløst, for hvis du vil realisere et andet format end f.eks. OSBORNE-formatet, så vil hele BASIC-LOADEREN være værdiløs. De af læserne, der vil indlæse et andet format på deres CPC, må have et vist kendskab til maskinsprog, da deres CP/M ikke uden videre kan klare det nye format. De må ved hjælp af BDOS-rutinerne indlæse dataene pr. maskinsprogsprogram. Et lille tip: Assembler Macro-80 (til Z-80) opretter, som assembleren ASM på CP/M-disketten, COM-filer efter ønske!

Ved en tilsvarende anvendelse af tabellen FDC (linie 570-580) er hvert af de anførte formater mulige. Pas blot på, at værdierne bliver indtastet i den rigtige rækkefølge for systemlageret. Alle værdier, der forekommer i tabellen, men ikke i systemlageret, kan der ses bort fra.

For at forstå, hvor hurtigt 1024 bytes load'es, skulle du prøve følgende: Da alle spor bliver beskrevet efter formatteringen, må et eller andet lagerområde bruges til lagringen. I det følgende eksempel blev lagerområde &5000-&5400 anvendt til det. Indlæs følgende BASIC-linier før formatteringen:

```
DEFINT I
FOR I=&5000 TO &5400
POKE I,I AND &FF
NEXT
CALL &7000 (til start af formatteringen)
```

Dermed er lagerområdet &5000-&5400 blevet belagt med data. (Der tælles opad fra 0 til 255 og begyndes så igen med 0). Dette lagerområde er efter kaldet CALL &7000 blevet lagret på disketten (200 gange).

På adressen &7200 befinder sig endnu en rutine til at indlæse en 1024 bytes stor sektor. Før du kalder denne rutine, bør du slette lagerområdets indhold, ellers kan du ikke kontrollere, om lageret er blevet forsynet med data fra floppy'en:

```
FOR I=&5000 TO &5400
POKE I,0
NEXT
```

Efter at have slettet lagerets indhold, kan du at load'e en tilfældig blok - der står jo det samme overalt på disketten:

```
CALL &7200
```

og lynhurtigt er sektoren indlæst. Du kontrollerer ved at lade lagerområdet vise:

```
FOR I=&5000 TO &5400
PRINT PEEK(I)
NEXT I
```

Som resultat skulle du få:

1 2 3 4 5 6 7 8 9 10 11 12 13...

Det tjener ikke blot som et anskueligt eksempel på, hvor hurtigt data kan skrives, men også som en prøve, om der er blevet lagret og load'et 1024 bytes.

Endnu et lille - under visse omstændigheder meget vigtigt - tip: Da der standardmæssigt kun bliver indlæst 512 bytes store sektorer i CPC'en, er der også kun en buffer med 512 bytes til rådighed for indlæsning af sektorerne. Da denne buffer praktisk talt automatisk udvides til 1024 bytes, når der indlæses 1024 bytes, bliver undertiden skrevet oven i vigtige data eller programdele. Det udgør ikke noget stort problem, fordi du kan omplacere bufferen til et hvilken som helst sted. Vektoren for sektorbufferen står på adressen:

&BE62, &BE63

Man kan opnå dette ved at tilføje følgende linier til assemblerprogrammet:

```
135      LD      HL,SEKBUF
137      LD      (_BE62),HL ;OMPLACER BUFFER
1030 SEKBUF: DS 1024      ;1024 BYTES I BUFFER
```

7.4. CP/M-disketten

Dog, nu tilbage til det disketteformat, som vi ikke først omstændeligt skal simulere. Til Amstrad CPC 464 og CPC 664 leveres der en diskette med, hvorpå der både befinder sig Digital Research's logo og CP/M 2.2. Vi vil først se lidt nærmere på, hvad der altsammen befinder sig på CP/M-disketten. Der tumler sig nemlig ikke alene CP/M-kommandofilerne, selvom de udgør langt den største del.

```
A: MOVCPM COM : PIP COM : SUBMIT COM : XSUB COM
A ED COM : ASM COM : DDT COM : LOAD COM
A: STAT COM : DUMP COM : DUMP ASM : AMSDOS COM
A: FILECOPY COM : SYSGEN COM : BOOTGEN COM : COPYDISC COM
A: CHKDISC COM : DISCCOPY COM : DISCCHK COM : SETUP COM
A FORMAT COM : CSAVE COM : CLOAD COM : EX1 BAS
A: EX2 BAS : ROINTIME DEM
```

På disketten ligger DEMO-filerne for BASIC i AMSDOS, som vi roligt kan forbigå. Foruden de standardmæssige CP/M-2.2 kommandoer befinder sig nogle COM-filer på disketten, som f.eks. CLOAD.COM og SAVE.COM. Disse to kommandofiler tjener til at kopiere henholdsvis CP/M-filer fra diskette til kassettebåndoptager (CSAVE), og fra kassette til diskette (CLOAD). Vil du lagre filen FILECOPY.COM på kassette, så skal du indtaste følgende kommando på computeren:

CSAVE FILECOPY.COM (RETURN)

Det er det hele. På skærmen fremkommer så:

```
CSAVE V2.0
Press REC and PLAY then any key:
Saving FILECOPY.COM block x
```

Hvis du har trykket på de to tilsvarende knapper på din kassettebåndoptager, kan du trykke på en tilfældig tase, og kopieringsprocessen begynder. Udlæsningen "block x" meddeler, fra hvilken blok der for øjeblikket bliver lagret på kassetten. Efter at det hele er blevet afleveret på kassetten, fremkommer der på skærmen:

```
CSAVE V2.0 finished
```

Det kan være praktisk at lagre bestemte filer på kassette, som du igen kan indlæse ved hjælp af CLOAD-kommandoen, da kassetter er et billigt lagermedium, i modsætning til de (endnu) meget dyre 3-tommers disketter. På den måde kan du arkivere. På store computere gør man det samme. Der bliver pladestablerne regelmæssigt kopieret på billige magnetbånd.

7.5. Den medleverede assembler ASM

På disketten, der leveres med disktestation, og selvfølgelig også med CPC 664 og CPC 6128, befinder sig en assembler. Denne assembler kan vi kalde med ASM. Men stop! Glæd dig ikke for tidligt. Som du måske ved, besidder Amstrad'en en Z-80 processor. Måske har du allerede lært at programmere i maskinsprog gennem et møjsommeligt, sveddryppende arbejde. Denne assembler er imidlertid ikke nogen Z-80-assembler, men derimod en 8080-assembler. For dem af jer, der ikke skulle vide det: 8080'eren er ældre end Z-80'eren og nærmest dens fader. Z-80 forstår de programmer, som dens fader forstår. Den har endda et meget større ordforråd end sidstnævnte, men den har en anden mnemokode. Nej, vi har ikke skrevet forkert. Mnemokode kalder man den skrivemåde, som maskinsprogsprogrammørerne koder deres maskinsprogsprogrammer i. Du ved sikkert, at maskinsprogsprogrammer udelukkende består af tal. For eksempel forstår Z-80 processoren kommandoen &41 og kan udføre den. For et menneske er det meget svært at forestille sig noget ved det. Derfor har man udviklet mnemokoderne, som delvis skal erstatte maskinsprogs-koder. Ved Z-80 ser mnemokoden for kommandoen &41 sådan ud:

```
LD B,C
```

Assembleren har til opgave at oversætte mnemokoden, som mennesket "så godt" forstår, til kode som maskinen kan forstå. Processoren kan ikke gøre noget med mnemokoden LD B,C. Her træder assembleren i funktion som tolk eller skæringspunkt, der oversætter menneskesproget til maskinsprog. Assembleren

understøtter tillige mennesket i dets arbejde gennem nogle hjælpeforanstaltninger, som jeg senere kort skal komme ind på.

Du ved, hvad en mnemokode er, og at den ældre 8080 processor er fader til Z-80. Yderligere er du klar over, at Z-80'eren behersker flere kommandoer end 8080'eren. Alle CP/M-programmer er programmeret i 8080-kode, fordi der findes CP/M-computere, som er forsynet med en 8080'er. Det at en Z-80-programmør må begrænse sig til de kommandoer, som en 8080 kan forstå (og dermed må give afkald på nogle virkelig bemærkelsesværdige kommandoer, som Z-80 har). Hvis du vil programmere for de i dag almindelige Z-80-CP/M computere, så kan du bruge det ovennævnte Z-80-LIB. Det er desværre ikke alt, for man har også ændret mnemokoderne ved videreudviklingen af 8080 til Z-80. Lad os erindre kommandoen &41. Der sker i begge processorer ved genkendelsen og udførelsen af denne kommando det samme: C-registerets indhold bliver kopieret over i B-registeret. Mnemokoden ser sådan ud ved Z-80:

LD B,C

Ved 8080 ser den samme (!!) kommando sådan ud:

MOV B,C

Du har ikke gavn af dit Z-80-kendskab, når du vil programmere en 8080, selv hvis du begrænser dig til de kommandoer, som 8080'eren også kan forstå. Altså: ASM-assembleren (Commodore: MAC-assembleren) kan du kun fodre med mnemokoder, som en 8080-assembler også kan forstå.

Vi vil se nærmere på assembleren, der gør det muligt at programmere egne COM-filer.

7.6. Betjeningen af assembleren ASM (Amstrad) resp. MAC (Commodore)

Ved programmering i maskinsprog må man ofte henvise til bestemte lagersteder, f.eks. for at load'e eller tømme registre. Endvidere forekommer det, at man hopper til en lageradresse osv. Forestil dig, at du har skrevet et program, der bgynder på lageradresse \$5000 og vil skifte det til lageradresse \$4000. Så skal du ændre alle hopadresser, bortset fra de relative hop. Hvis du indføjer en kommando i programmet, forskubber alle relative hop sig og alle følgende direkte hop. Til sidst indtræder så det, at programmøren går fra sans og samling. For at lette sådanne arbejder, kan man i assemblere definere "mærker" eller labels, som under assemblyen (dvs. oversættelsen af mnemokoderne til maskinsprog) bliver indsat i programmet. Derefter kan man indføje en kommando på et bestemt sted eller forskyde en rutine eller endda hele programmet - assembleren overtager arbejdet.

Hvordan ser et assembler-program ud. Man har været så venlig at lagre et assemblerprogram på din CP/M-diskette, hvormed vi kan håndtere assembleren. Se nærmere på indholdsfortegnelsen. Vi har en fil med navnet DUMP.COM; det er, en COM-fil, som vi med CP/M kan starte ved at indtaste navnet:

```
DUMP <Filename>
```

ville det syntaktisk korrekte være i dette tilfælde. Dump giver en udskrift af den angivne fil i HEX-format (Hex-dump). Du kan roligt prøve det:

```
DUMP FILECOPY.COM
```

På skærmen fremkommer Hex-dump'en af COM-filen Filecopy, programmet selv på en vis måde. Du kan standse udlæsningen med <Ctrl>/S og atter forsætte med <Ctrl>/Q. Ved hjælp af <Ctrl>/C kan du afbryde udlæsningen. På Commodore kan benyttes NO-SCROLL-tasten, der er mere praktisk end <CTRL>-sekvenserne.

Til Commodore skal man være i besiddelse af Utility-pakken. På den diskette ligger assemblerprogrammerne MAC og RMAC, så vi også kan øve betjeningen af assembleren. Prøv at kigge på indholdet af din utility-diskette:

```
A: LIB COM : LINK COM : MAC COM : RMAC COM : SID COM  
A: CBDOS3 SPR : BNKBDOS3 SPR : CRESBDOS3 SPR : HEXCOM COM :  
XREF COM  
A: CALLVERS ASM : DUMP COM : ZECHOVERS ASM : RANDOM ASM :  
HIST UTL  
A: TRACE UTL : READ ME
```

Her har vi en fil ved navn DUMP.COM, så vi kan fortsætte som ovenfor med at indtaste DUMP-filnavn

DUMP.ASM er maskinsprogsprogrammet for DUMP, sådan som det er kodet af programmøren. Du kan lade filen udlæse på skærmen (eller printeren). Indtast for at opnå det:

```
TYPE DUMP.ASM
```

Du kan se af programmets kreditliste, at firmaet DIGITAL RESEARCH besidder ophavsrettighederne til dette program. Det er ikke mærkeligt, eftersom firmaet har udviklet CP/M. Du kan se en væsentlig forskel mellem assembler- og maskinsprog: man kan indføje kommentarer, som begynder med et semikolon, for at assembleren skal vide, at der ikke er mere at oversætte. Ellers ville assembleren prøve at forstå, for slet ikke at sige oversætte, denne kommando og kunne ikke forstå den og sige: "Her er der noget, der ikke stemmer". Også selv om dette program skulle forekomme dig meget langt, kan du roligt tro mig, når jeg siger, at det er fordømt kort. Tænk på, at driftssystemer også kodes i assembler:

Der findes driftssystemer, der er på 2000 sider eller mere, når man printer dem ud. Også "mindre" programmer, som f.eks. tekstbehandlingsprogrammet TEXTOMAT

til din Amstrad eller Commodore kommer let op på 400 til 500 sider. Du kan se, at programmering i maskinsprog er en meget kostbar sag. Aligevel vil vi prøve at assemblere dette assemblerprogram, for at du for fremtiden skal kunne bruge det selvstændigt. Pas på, at din arbejdsdiskette nu ikke er skrivebeskyttet, for vi må også skrive på den diskette, hvorpå kildeprogrammet befinder sig. Du har vel allerede lavet en sikkerhedskopi af din CP/M-diskette?! (Hvis ikke, så gør det hurtigst muligt.). Vi assemblerer nu vort maskinsprogsprogram:

ASM DUMP (Amstrad) resp. MAC DUMP (Commodore)

Her ser du et vigtigt kendetegn ved assembleren. Du behøver ikke at indtaste nogen udvidelse. Du skal ikke skrive DUMP.ASM, fordi assembleren selv sætter udvidelsen .ASM på. **På Amstrad skærmen vil du se.**

```
CP/M ASSEMBLER - VER 2.0  
0257  
002H USE FACTOR  
END OF ASSEMBLY  
A>
```

Og på Commodore:

```
CP/M MACRO ASSEM 2.0  
0257  
002H USE FACTOR  
END OF ASSEMBLY  
A>
```

Hurtig er den da, vor assembler, ikke sandt? Den opretter nu to filer på Amstrad:

- 1) En protokol under navnet DUMP.PRN**
- 2) En Hex-dump under navnet DUMP.HEX**

og på Commodore tre filer, nemlig foruden de to ovennævnte:

3) *En symboltabel under navnet DUMP.SYM*

Betragt først protokollen ved at indtaste TYPE DUMP.PRN. Assemblerprogrammet listes endnu en gang, med alle de informationer, som assembleren tilføjer. I første spalte står altid den adresse, hvorpå der for øjeblikket kodes - med mindre en label er ved at blive defineret; så står den definerede værdi af label'en i den første spalte. Programmet begynder altså med adressen \$0100 og slutter ved \$0257. Følgelig ved vi, hvilket tal assembleren har udlæst for os (se ovenfor). Næsten alle assemblerprogrammer begynder ved adressen \$0100. Efter at du har set på protokollen, eller fået den skrevet ud på papir, ser du på den anden fil. Hertil indtaster du igen (du ved det sikkert allerede):

TYPE DUMP.HEX

På billedskærmen er en talrække, som assembleren har oprettet. Det er programkoderne. Desuden vises nogle yderligere informationer, såsom den adresse, hvorpå koden senere skal stå osv. *Filen er betydeligt kortere end vor fil DUMP.PRN.* Interessant er også symboltabellen, som bærer filnavnudvidelsen .SYM. Vi kan stadig ikke starte dette maskinprogram, det er jo ikke nogen COM-fil. Der findes endnu et program, som overtager dette arbejde. Det frembringer en COM-fil udfra en HEX-fil, altså en, der kan kaldes og startes af CP/M. Dette program bærer navnet LOAD på Amstrad og HEXCOM på Commodore. Vi prøver at kalde det:

LOAD DUMP resp. HEXCOM DUMP

Her skal eller må ikke anføres nogen udvidelse. LOAD (HEXCOM) sætter i begge tilfælde automatisk udvidelsen ".HEX" på. På skærmen fremkommer:

```
FIRST ADDRESS 0100
LAST ADDRESS 0212
BYTES READ 0113
RECORDS WRITTEN 03
```

På Commodore er der tillige øverst en "overskrift":

HEXCOM VERS: 3.00

Endelig er vi kommet så langt at du har oprettet en COM-fil, filen DUMP.COM, som vi i CP/M kan kalde ved at indtaste:

```
DUMP (Amstrad) resp. DUMP DUMP.COM (DUPM'er sig selv ud)
(Commodore)
```

Det er ikke så enkelt at programmere COM-filer, da man må være godt kendt med CCP og BDOS, fordi man skal ty til disse rutiner.

Nogle informationer om assembleren, for at du skal kunne betjene den helt rigtigt.

Kildeprogrammet til assembleren må udvise følgende syntaks:

```
<linienummer><label>:<kommando><argument>;<kommentar>
```

Linienummeret er en option. Dvs. det behøver ikke at være til stede (det gælder for vort fileksempel). Det bliver alligevel ignoreret af assembleren og kun planlagt, fordi nogle editors automatisk tilføjer disse linienumre, som editoren ED på CP/M-disketten. <Label>: behøver ikke ubetinget at forekomme, men kan stå på dette sted. Ubetinget må i en linie, der ikke kun består af en kommentar, både <kommando> og <argument> være til stede. Til dem slutter sig så eventuelt en ;<kommentar>.

Assembleren forvandler før kodningen alle små bogstaver til bogstaver, sådan som vi er vant til det fra CP/M. Det forenkler programmeringen overordentligt. Undtaget fra den regel er små bogstaver, der står i anførelsestegn og udgør en fast tekst, som f.eks. anvisningen:

DB 'File Dump Version 1.4\$'

De enkelte led i en assemblerlinie skal være adskilt af et tomt mellemrum. Det er skik og brug at lade <label> begynde helt til venstre og så ved hjælp af tabulatoren i det tekstbehandlingsprogram, hvormed du skriver assembler-kildeprogrammet, at gå hen til de andre led. Det er ikke ubetinget nødvendigt, men er til stor fordel for programmets overskuelighed, da man så har tilsvarende led stående under hinanden. Et eksempel:

```
DISKR: ;READ DISK FILE RECORD
```

```
PUSH H! PUSH D! PUSH B!  
LXI D,FCB  
MVI C,READF  
CALL BDOS  
POP B! POP D! POPH  
RET
```

Den passage kan du finde i vort DUMP-program. DISKR er en label, som er markeret af et kolon. Så følger <kommando> og <argument>, hele tiden under hinanden. I een linie har vi en kommentar, indledt af et semikolon, ";".

I rutinen ses en anden egenskab ved ASM-assembleren: forskellige assemblerkommandoer kan adskilles af et udråbstegn.

I ASM kan man mærke kommentarer med et semikolon, eller definere en hel linie som en kommentar, idet man begynder med en asterisk i den første spalte. Det blev indført af dem, der udviklede ASM, for at bevare en vis kompatibilitet med de bestående 8080 assemblyer. Således kan man forsyne enkelte rutiner i assemblerprogrammet med en overskrift, f.eks.:

```
*****  
***          Read Disk File Record          ***  
*****
```

Pas ved anvendelsen af labels på, at du ikke indbygger nogle "reserverede ord". Reserverede ord er for eksempel 8080-processorens mnemokoder, altså ord som MVI, MOV, STA osv.

I ASM-assembleren kan du, ligesom i mange andre assemblyer, bestemme den øjeblikkelige position af program counteren (tælleren) PC under assembleringsprocessen. Det sker automatisk ved programlabels, som i det følgende eksempel:

```
posit: EQU $
```

Argumenter, der forekommer i ASM, kan sammenknyttes med forskellige aritmetiske operander som f.eks.:

```
MVI A,12+2*3
```

Aritmetiske operander er:

+X	Positive tal
-X	Negative tal, svarer til 0-X (altid 16 bits !)
X+Y	Addition af to 16-bits værdier
X-Y	Subtraktion af to 16-bits værdier
X*Y	Produktet af X gange Y
X/Y	Division af argumenterne
X MOD Y	Restfunktionen af divisionen X/Y

Endvidere kan man anvende disse to skifteværdier:

X SHL Y	Skifter 16-bits-værdien X Y positioner til venstre. Bits, der skubbes ud, går tabt.
X SHR Y	Skifter 16-bits-værdien X Y positioner til højre. Bits, der skubbes ud, går tabt.

Endvidere findes der også de logiske operatører:

NOT X	Logisk negation af argumentet X
X AND Y	Logisk og-komposition af argumenterne X og Y
X OR Y	Logisk eller-komposition af argumenterne X og Y
X XOR Y	Eksklusiv eller-komposition af argumenterne X og Y

Alle disse muligheder forekommer dig måske overflødige. Det er de ikke, tværtimod, de er endda særdeles nyttige. Hvis du vil lade akkumulatoren med en adresselabels byte af højere værdi, så kan det gøres ved at skifte denne label 8 bits til højre:

```
MVI A,label SHR 8
```

Hvis du vil have de otte laveste bits, så må du udtrykkeligt "afblænde" de øverste 8 bits, fordi du ellers ville prøve at lade akkumulatorens 8-bits register med en 16-bits værdi. Afblændingen ser således ud:

```
MVI A,label AND 0FFH
```

Eksemplet er det oftest forekommende. Der kan tænkes i tusindvis af andre anvendelser.

Endvidere eksisterer der pseudo-opcoderne

```
ORG, EQU, END, DS, DB, DW, SET, IF og ENDIF
```

(Pseudo-opcoder er assembler-kommandoer, der bliver læst og forarbejdet af assembleren under assemblingsprocessen. Navnet skyldes, at disse kommandoer forekommer i assembler-programmet ligesom de normale opcoder, men ikke kan assembles og forstås af processoren.)

Vil vil ikke forklare de enkelte pseudo-opcoder nærmere, men kun nævne virkningen og funktionsmåden, da de ikke er en CP/M-bogs område.

ORG <startadresse>

Kommandoen definerer startadressen på det program, der skal assembles. ORG skal altid være den første kommando i et program.

EQU <værdi>

EQU tildeler en label en fast værdi. <Værdi> kan også være et aritmetisk udtryk.

Eksempel: Diskm: EQU Diskr+055H
DS

Kommandoen holder i programmet et bestemt område frit, for at kunne lagre data der. Programtælleren sættes tilsvarende højere.

Eksempel: Diskm: DS 100

I eksemplet bliver 100 bytes holdt fri. Startadressen er Diskm, slutadressen er Diskm+99. Den næste kommando begynder ved Diskm+100.

DB

Her bliver de enkelte bytes, der skal defineres, lagret i arbejdslageret. De enkelte bytes skal være adskilt af kommaer. Der kan også forekomme tegnstreng.

Eksempel: Disktxt: DB "Vær venlig alt lægge en diskette i"
Diskt2: DB 23,32,0,3,122
DW

DW definerer ord, altså 16-bits værdier. Også her kan de enkelte 16-bits værdier adskilles med et komma.

Eksempel: Diskadr: DW 03AB9H, label1
END

END definerer afslutningen af det program, der skal assembles. Til specielle formål er det muligt at angive en startadresse til effektueringen af det assemblerede program.

SET

Med SET-kommandoen bliver, ligesom med EQU-kommandoen, en label tildelt en værdi. Ganske vist med en forskel: medens tildelingen er fast med EQU-kommandoen, kan labels, der er defineret med SET-kommandoen ændres under assemblingen. Labels, der er defineret ved hjælp af SET, bliver ikke vist i den venstre spalte under assemblingen.

Eksempel: label1: SET alfa
label1: SET label1 + 32

Hermed er vi stadig ikke stødt på grænserne for ASM's resp. MAC's ydeevne. ASM og MAC kan assemblere betinget. Dvs. kun under bestemte betingelser at afkode og oversætte en bestemt assemblerpassage til maskinsprog. Det opnås ved at bruge kommandoerne IF og ENDIF. (Kommandoer, der erfaringsmæssig sjældent benyttes.)

Du vil skrive et program til at indlæse to værdier, og så enten addere eller multiplicere dem. Algoritmen er den samme for begge programmer. Du kan skrive et program, som under assemblingen spørger et "flag", om programmet skal addere eller multiplicere.

```
;
multi EQU 0 ;programmet skal addere
ORG 0100H
;
;indtast værdier
CALL indtast
;
IF multi
    CALL multir ;kald multiplikationsrutine
ENDIF
IF NOT multi
    CALL addir ;kald multiplikationsrutine
ENDIF
CALL udlæsning
END
```

Når du lader den første rutine multiplicere, så sætter du i den første linie værdien af multi til NOT 0.

Efter at vi har behandlet assembleren så udførligt, vil vi se på, hvordan man kalder ASM resp. MAC:

Man behøver når man kalder ASM eller MAC, ikke anføre udvidelsen ".ASM" med. De to filer <filnavn>.PRN og <filnavn>.HEX bliver frembragt på disketten, på Commodore tillige den tredje fil <filnavn>.SYM. Dem kan man undertrykke! Endvidere kan man anføre, fra hvilken station, de data, der skal assembles, skal tages, eller i hvilken fil de to (tre) filer, der skal frembringes, skal lagres. **Syntaxen i ASM-kommandoen ser således ud:**

ASM <fil>.<ASM-fil><HEX-fil><PRN-fil>

Bortset fra filnavnet <fil> er alle angivelser options, men skal komme i den angivne rækkefølge. Følgende options er mulige:

For ASM-filen: A eller B

A eller B er kildefilens stationsangivelse.

For HEX-filen: A, B eller Z

Her står A og B for stationsangivelse for den station, hvorpå filen skal lagres under navnet <filnavn>.HEX. Yderligere kan man anvise assembleren, at den ikke skal frembringe nogen HEX-fil. Det er praktisk, når du vil foretage en prøvekørsel for at kontrollere syntaksen. Så anfører du på det andet sted et 'Z' som option.

For PRN-filen: A, B, X, eller Z

A eller B er stationsbetegnelser, hvorpå filen <filnavn>.PRN skal frembringes. Denne protokol kan undertrykkes ved at anføre 'Z' (se ovenfor). Protokolfilen udlæses på skærmen med option'en 'X'.

Hvis du vil indlæse en kildefil med navnet Asterix fra station B, men ikke have protokolfilen på billedskærmen eller HEX-filen, måske for at lade assembleren kontrollere dit programs syntax, så skriver du følgende:

ASM Asterix.BZX

For Commodore og assembleren MAC er proceduren anderledes. Der eksisterer fem options: A, H, L, P og S. Angives options, skal et dollartegn sættes som kendetegn:

MAC<fil>\$HB PZ

De enkelte bogstaver har følgende betydning:

<i>A</i>	<i>Station til .ASM-filen (A-O)</i>
<i>H</i>	<i>Station til .HEX-filen (A-O,Z)</i>
<i>L</i>	<i>Station til .LIB-filen (A-O)</i>
<i>P</i>	<i>Station til .PRN-filen (A-O, Z, P, X)</i>
<i>S</i>	<i>Station til .SYM-filen (A-O, Z, P, X)</i>

Som stationsangivelser er stationsbetegnelserne A-O tilladt. Desuden kan man ved nogle filer forudse, at der ikke bliver oprettet en fil (Z-option, for zero). At der følger en udlæsning på skærmen (X-option) eller på printeren (P-option).

Hvis der ikke intastes nogen options, så bliver alle stationsangivelser rettet til standardstationen og alle filer oprettet der. Hvis du vil ændre noget på denne standardind-

stilling (såkaldt *DEFAULT*), så indtastes de pågældende identifikationsbogstaver og option'en lige bagefter. Et dollartegn er obligatorisk. To options skal adskilles af et tomt mellemrum.

RMAC ligner *MAC* temmelig meget i betjeningsstrukturen. Der bliver ikke oprettet nogen *HEX*-fil ved assemblingen, men en *.REL*-fil, der skal behandles med linkerens *LINK*. Med *RMAC* giver man angivelserne til *.LIB*-filerne ved linkerens. Ligeledes er der for *RMAC* kun tre options:

<i>R</i>	Station til <i>.REL</i> -filen (A-O, Z)
<i>S</i>	Station til <i>.SYM</i> -filen (A-O, X, P, Z)
<i>P</i>	Station til <i>.PRN</i> -filen (A-O, X, P, Z)

Angivelserne i den anden spalte har den samme betydning som ved *MAC*. Med *RMAC* skal eventuelle options kendetegnes med et dollartegn, og de enkelte options skal skilles af et tomt mellemrum.

7.7. Hvordan man arbejder med *SUBMIT* og *XSUB*

Det forekommer hyppigt, at bestemte kommandorækkefølger skal indtastes igen og igen. Forestil dig, at du vil assemble en bestemt kildefil, så forvandle *HEX*-filen til en *COM*-fil og til slut starte netop denne nye *COM*-fil for at afprøve den.

Netop i den situation er det smart at oprette en *SUBMIT*-fil, der indeholder de nødvendige kommandoer og udfører dem automatisk. *SUBMIT* betyder at overgive. Programmet *SUBMIT* overgiver denne række af kommandoer til *CCP*, Console Command Processor'en, som tager sig af indlæsning via tastaturet og udførelsen af de indtastede kommandoer. Det sker på denne måde:

Når du fortæller *SUBMIT*, hvilken fil, der skal overgives som *SUBMIT*-fil, så opretter *SUBMIT* en mellemfil *\$\$\$SUB*. I den mellemfil står alt, hvad der hører til *SUBMIT*-filen, med nogle tilføjelser. Før *CCP* overtager en kommando fra tastaturet, ses der efter, om den midlertidige fil *\$\$\$SUB* ikke findes på den noterede station. Hvis det er tilfældet, så bliver en kommandolinie læst ind i *CCP*'s buffer og denne linie slettet fra filen. Efter at kommandolinien er blevet kopieret, bliver befalingen udført. Det gentages, indtil filen *\$\$\$SUB* er blevet tømt, først nu bliver der igen indlæst en kommando fra tastaturet.

Der bliver med *DIR*-kommandoen ikke angivet, hvor meget plads, der endnu er på disketten. **Den information får du via *STAT*-kommandoen (Amstrad) resp. *SHOW*-kommandoen (Commodore).** Vi vil oprette en *SUBMIT*-fil, der sammenfatter *DIR*-kommandoen og *STAT*-, resp. *SHOW*-kommandoen, så at vi med een kommando kan skaffe os oplysninger om diskettens kapacitet og indhold. Dertil opretter vi ved hjælp af tekstbehandling, f.eks. med *ED*, denne *SUBMIT*-fil og kalder den *DISK.SUB*. Alle *SUBMIT*-filer skal bære udvidelsen ".SUB", det er det første bud.

Nå, godt, vor første Submit-fil ser altså sådan ud på Amstrad:

**STAT
DIR**

På Commodore gør vi således

*SHOW
DIR*

Hertil er det bedst at lægge den anden side af P/M-systemdisketten i stationen og kalde ED således:

ED DISK.SUB

ED melder sig med det samme; ED mærker, at du vil anlægge en ny fil og meddeler det med:

*NEW FILE
: *—*

Cursoren blinker, og ED venter på indtastning. ED er svær at betjene. Da vi kun vil indtaste to linier, er der forhåbentlig ikke noget, der går galt. Indtast nu I for at meddele ED, at du vil indføje noget i filen. På skærmen fremkommer:

*: *i
I: —*

Indtast så de to kommandolinier:

*1: show a: (tryk RETURN)
2: dir (tryk RETURN)
3: (tryk CTRL-Z)*

Når det er gjort, skal du gå ud af ED med E-kommandoen:

*: *E (tryk RETURN)*

ED lagrer filen DISK.SUB på diskette. Hvis du ikke stoler på ED, så kan du kigge på filen ved hjælp af TYPE-kommandoen. Indrømmet: det er en meget kort SUBMIT-fil, men man må jo begynde i det små. Lad os engang starte denne første SUBMIT-fil med:

SUBMIT DISK

Du kan se, at vi ikke må anføre udvidelsen ".SUB". Det lyder som om vor station har nok at gøre, men vi ved godt, hvad der foregår. (Oprettelse af \$\$\$SUB-filen, kopiering af indholdet af DISK.SUB, linievis sletning af kommandoerne fra filen \$\$\$SUB osv.) Og det går altså sammen; både diskettepladsen og indholdsfortegnelsen bliver vist på skærmen.

Husk at vi ville assemble en fil, gøre den til en COM-fil ved hjælp af LOAD-(Amstrad) resp. HEXCOM-kommandoen (Commodore) og derefter starte den. Den rækkefølge af kommandoer kan se sådan ud på Amstrad:

```
ASM <filnavn>.<options>  
LOAD <filnavn>  
<filnavn>
```

og på Commodore

```
MAC<filnavn>.<options>  
HEXCOM<filnavn>  
<filnavn>
```

Begreberne <filnavn> og <options> står her for at holde pladsen, som variabler; men hvordan kan man fylde disse variabler? Konstruktørerne af SUBMIT-programmet var klar over problemet, og har fundet på noget, for at udforme SUBMIT-filerne mere praktisk. Når du kalder SUBMIT-filen kan du overdrage disse parametre, idet du skiller parametrene fra hinanden med mindst eet tomt mellemrum. Parametrene kan være simple tal eller også komplekse begreber som f.eks. filnavne. Parametrene nummereres begyndende med eet. I SUBMIT-filen må man mærke disse parametre med et dollartegn (\$), direkte fulgt af parameterens nummer. Samtidig kan de enkelte forlæg gentages, så ofte man ønsker det, i SUBMIT-filen. Ved oprettelsen af filen \$\$\$SUB bliver "pladsholderne" automatisk erstattet af parametrene, så at CCP selvstændigt kan bearbejde dem. **Vor SUBMIT-fil ser nu således ud til Amstrad:**

```
ASM $1.$2  
LOAD $1  
$1 $3
```

og til Commodore

```
MAC $1 $$$2  
HEXCOM $1  
$1 $3
```

Du ser, at den første indtastning, altså den første parameter, forekommer tre gange i SUBMIT-filen. Når vi starter den SUBMIT-fil med vort dump-program og ikke ønsker nogen protokol-fil (en HEX-fil skal der oprettes, fordi LOAD resp. HEXCOM jo behøver den), så kunne kaldet se sådan ud, hvis vi har lagret SUBMIT-filen under navnet "ASSEM.SUB":

```
SUBMIT ASSEM DUMP AAZ DUMP.COM (Amstrad)  
eller  
SUBMIT ASSEM DUMP AB DUMP.COM (Commodore)
```

Den sidste retningslinie viser, hvilken fil der skal dumpes, i vort eksempel DUMP.COM selv.

Hvad skal man gøre, hvis det er nødvendigt, at der skal forekomme et dollartegn i SUBMIT-filen? Jo, man skriver to dollartegn efter hinanden. *I vort eksempel har du set, hvor hurtigt det kan blive nødvendigt at indføje et dollartegn i en SUBMIT-fil.* Prøv at slette alle midlertidige filer, og gør det via tastaturet med:

```
ERA *.$$$
```

I en SUBMITfil ser det mere afskrækkende ud:

```
ERA *.$$$$$
```

men betyder det samme. Da dollartegn forekommer ret sjældent, er det ekstra arbejde minimalt.

Når en kommandolinie fra filen \$\$\$SUB bliver kopieret ind i CCP's lager, bliver linien vist på skærmen. Derefter bliver tastaturet spurgt, om der er blevet trykket på en tast. Er det tilfældet, eller bliver kommandoen i kommandolageret ikke udført eller udført forkert, så afbryder SUBMIT kommandoforløbet og sletter filen \$\$\$SUB.

Specielt for Amstrad:

Hvis en linie i SUBMIT-filen begynder med et tegn, der normalt ikke bør forekomme i et filnavn, så bliver linien vist på skærmen, men bliver ikke udført. På den måde kan du vise kommentarer eller anvisninger for brugeren på skærmen. Det vil være muligt at supplere det følgende i vor SUBMIT-fil ASSEM.SUB:

```
:Nu assembles filen.  
:Vær venlig at have lidt tålmodighed!!  
ASM $1.$2  
LOAD $1  
$1 $3
```

Og SUBMIT-filen er lidt mere afslappet i sin udførelse. Tegn, der bevirker dette er:

```
< > ; : , ? = * -
```

Ved SUBMIT begrænser parametrene sig til CCP. Man kan ikke lave parametre til kaldte programmer som PIP eller ED. Det kan være ønskeligt, bl.a. for at kopiere forskellige filer med PIP eller videregive bestemte anvisninger til ED. Også denne "manko" lader sig overvinde. Hertil findes programmet XSUB, der opfanger alle indtastninger fra tastaturet og overtager forespørgslerne fra \$\$\$SUB-filen. Når vi ved PIP vil gøre angivelser (det er også muligt direkte), så kan det se sådan ud:

```
XSUB  
PIP  
b:=a:*.COM
```

I den sidst linie må du trykke RETURN for at forlade den midlertidige rutine PIP. I PIP er den anvendelsesmåde ikke så praktisk, fordi man direkte havde kunnet udforme kommandoen anderledes, f.eks. PIP b:=a:*.COM for at kopiere alle COM-filer fra station A: på station B:, men ved ED bliver anvendelsen interessant. Det er tilstrækkeligt at kalde XSUB een gang i den første linie. Angivelser, der ellers skulle gøres via tastaturet kommer nu fra filen \$\$\$SUB.

I de gamle CP/M 2.2 versioner var der desuden gemt en fejl i SUBMIT-programmet. Når man overførte data ved hjælp af XSUB, så kunne man ikke overgive nogen styretegn, skønt det var forudset i brugsanvisningen. Det er meget ærgerligt, da man med PIP eller ED ofte vil eller skal sende styretegnet ↑Z. Da fejlen er rettet i din CP/M 2.2 version, behøver du ikke at tænke mere på det.

Specielt for Commodore:

Indenfor SUBMIT-filerne kan man ikke blot gøre angivelser til CCP, men også til programmer, der kaldes som ED eller PIP. Disse indtastninger skal have "mindre end"-tegnet forrest, ellers går det ikke. Vi vil tydeliggøre det med en anvendelse af PIP:

```
PIP
<E:=A:*.COM
<
DIR *.COM
```

Den sidste linie, der blev indtastet til PIP er kun et "mindre end"-tegn, for at forlade den midlertidige rutine PIP. I PIP er anvendelsen ikke så praktisk, fordi man må have udformet kommandoen anderledes, omtrent sådan PIP b:=a:.COM, for at kopiere alle COM-filer fra station A: til station B:, men med ED bliver anvendelsen interessant. Det er tilstrækkeligt at sætte "mindre end"-tegnet forrest, for at overdrage data til indlæsninger af COM-filer. Det var ikke så simpelt med CP/M 2.2. Angivelser, der skulle gøres via tastaturet, kommer nu fra filen \$\$\$SUB*

Vi har hørt om assembleren og SUBMIT-filerne. Vi skal kort omtale en disassembler, som også befinder sig på disketten. Du kalder denne disassembler med DDT på Amstrad og med SID på Commodore. SID er den direkte efterfølger af CP/M 2.2 versionen DDT. Hvis du for eksempel vil have kommandofilen FILECOPY.COM printet ud i disassembler form, så indtaster du blot

```
DDT filecopy.com resp. SID filecopy.com
```

Før du trykker <RETURN>, giver du med <Ctrl>/P den kommando, at det skal skrives ud på printeren.

At disassemble betyder det modsatte af at assemble: maskinkoderne bliver oversat til menneskeværdige mnemokoder. SID (Commodore) er ikke kun en disassembler, men også en monitor. Prøv at disassemble HEXCOM ved indlæsning af diassemblingskommandoen:

```
CP/M 3 SID - Version 3.0
NEXT MSZE PC END
0580 0580 0100 D2FF
_L
0100 JMP 0180
0103 NOP
_L180
0180 LXI H,0000
0183 DAD SP
```

Med D-kommandoen kan du lade et område uddumpe og med G-kommandoen starte programmet. Ved hjælp af A kan du indlæse kildekode med direkt-assembler-metoden. Der findes flere kommandoer, som vi ikke vil gå nærmere ind på. Dem kan du finde i Commodore-håndbogen.

7.8. Lagertildeling

Vi har omtalt slagord som BDOS, CCP, BIOS osv. Begreberne er meget vigtige, især da de så lovende er trykt på bogens omslag. Lad os repetere, hvad vi forstår ved de forskellige begreber:

CCP

CCP står for Console Command Processor. Dette område styrer indlæsningen fra tastaturet og indeholder de indbyggede kommandoer, altså DIR, ERA, REN, SAVE, TYPE, og USER.

BDOS

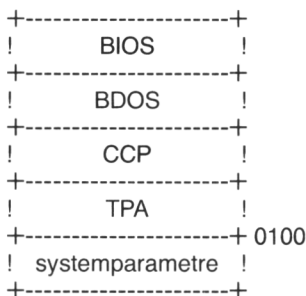
DOS står for Basic Disk Output System. Den del styrer dataudvekslingen mellem computer og disktestationer.

BIOS

BIOS skal være en forkortelse for Basic Input/Output System. BIOS og BDOS arbejder snævert sammen, hvorfor man også hyppigt finder de to partnere under et sammenfattende navn: FDOS.

TPA

Man kalder brugerprogramområdet i computerens lager for TPA. De programmer, der skal udføres, og de data, der skal bearbejdes med disse programmer, bliver forvaltet i dette lagerområde.



Således ser lagerinddelingen i CP/M ud. Brugerområdet (TPA) begynder ved adresse \$0100.

Da forklaringen på, hvordan man programmerer ved CP/M ville sprænge denne bogs planlagte rammer, vil vi kun kort gå ind på programmering i CP/M ved hjælp af BDOS- og BIOS-rutiner.

BDOS indeholder en samling af underprogrammer, der muliggør det grundlæggende arbejde med filer på diskette og med visse former for ydre apparatur. Da CP/M allerede har nogle år på bagen, forekommer der f.eks. stadig rutiner til ind- og udlæsning fra hulkort. Der bliver sikkert også stadigvæk hyppigt gjort brug af dem, men på vor CPC eller Commodore kan vi roligt se bort fra sådanne rutiner.

For at kalde en BDOS-rutine, må man rette sig efter de følgende standardiserede vedtægter:

I processorens C-register bliver den ønskede underrutines kodenummer videregivet til BDOS. I E-registeret, henholdsvis ved 16-bits værdier i DE-registerparret, bliver underrutinen om nødvendigt forsynet med data. Hvis man har forsynet registre på passende måde, hopper man til BDOS på adresse 5 gennem en underprogramskommando (CALL). BDOS fremskaffer så ved hjælp af C-registeret den adresse, hvorpå den ønskede rutine står. Du kan se, at programmer på denne måde også virkeligt kan gøres i stand til at køre på alle computere, fordi man overholder faste regler. Alle computerafhængige sager, som ind- og udlæsning fra tastatur og skærm bliver klaret af BDOS-rutiner, der er udviklet specielt til den pågældende computer. Derfor skal man ved udviklingen af en ny computer "kun" skrive disse rutiner, og straks kører alle CP/M-programmer på denne computer.

To forudsætninger må naturligvis være givet. Computeren skal have en 8080-processor eller en, der er kompatibel dermed, og det skal være muligt også at indlæse disse forhåndenværende programmer. Skulle andet ikke være muligt, så skulle i alt fald teoretisk set CP/M-programmerne kunne køre på denne computer. Ganske vist er det erfaringsmæssigt det mindste problem at overføre forhåndenværende programmer. Dette kan ske: a) ved programmering af Controlleren (se ovenfor) eller: b) idet man med et relativt simpelt program overfører de forhåndenværende programmer ved hjælp af et fælles skæringspunkt for to computere, f.eks. RS-232. Men selv

når der ikke findes noget fælles skæringspunkt, så finder de tålmodige altid en metode, f.eks. kan joystickspporten meget vel være tjenlig til indgangsport for overførelsen af et program.

Men tilbage til BDOS. Hvis BDOS skal tilbagelevere værdier fra BDOS-rutinen til hovedprogrammet, så vil der forefindes 8-bits resultater i processorens akkumulator, 16-bits værdier står som returværdier i HL-registerparret. Ved 16-bits returværdier befinder den lavere bit sig yderligere igen i akkumulatoren, bit'en med den højere værdi befinder sig i B-registeret.

Hvis du for eksempel vil udkøre et tegn på konsollen, så benytter man til det BDOS-rutine nr. 2. Vi udkører nu (i Z-80 mnemokode) et spørgsmålstegn:

```
LD      C,2      ;konsoludkørsel
LD      E,"?"    ;spørgsmålstegn
CALL   5        ;sæt BDOS i gang
```

Hvis du virkelig for alvor vil programmere i maskinsprog med CP/M, så eksisterer der faglitteratur specielt om programmering med CP/M. Vi giver dig her en liste over BDOS-rutinerne med ind- og udlæsningsparametre til at eksperimentere med.

Rutinenavn	C	overtager	giver
Varmstart	0		
Konsolindlæsning	1	A:indlæsning	3
Konsoludlæsning	2	E:tegn	
Læse en hulstrimmel	3		A:tegn
Hulle en strimmel	4	E:tegn	
Tegn på printer	5	E:tegn	
Direkte konsol ind- og udlæsning	6	E: 255 E:tegn	A:0 A:tegn
IOBYTE, spørge om	7		A: IOBYTE
IOBYTE, sætte	8	E: IOBYTE	
Udlæse tegnstreng	9	DE:⇒tegnstreng	
Indlæsning fra buffer	10	DE:⇒buffer	A:tegn
Konsolstatus	11		A:=0 (ingen taste)
CP/M version	12		HL:version
Diskettesystemreset	13		
Fastsættelse af noterede station	14	E:Stationsnummer	
Åbning af fil	15	DE:information	A:255 (error)
Lukning af fil	16	DE:information	A:255 (error)
Søgning af første optagelse ellers	17	DE:information	A:255 (error) A:tegn
Indlæsning af streng	18		A:255/tegn
Sletning af fil	19	DE:information	A:255 (error)

Rutinenavn		C	overtager	giver	
Læsning	af	optagelse20	DE:information	A:0	(OK)
Skrivning	af	optagelse21	DE:information	A:0	(OK)
Opretning	af	fil22	DE:information	A:255(error)	
Ombenævnelser	af	fil23	DE:information	A:255(error)	
Finde	aktive	station24		HL:stat.vektor	
Finde	not.	station25		A:stat.nr.	
Fastsætte		databuffer26	DE:buffer		
Finde		adressetabel27		L:adresse	
Beskytte	not.	station28			
Beskyttede		stationer29		HL:stat.vektor	
Sætte		filoptions30	DE:information	A:255	(error)
Finde		diskparametre31		HL:adresse	
Styre		brugernumre32	E.255	A:nummer	
				E:nummer	
Læsning	af	optagelse33	DE:information	A:0	(OK)
Skrivn.	af	optagelse34	DE:information	A:0	(OK)
Finde		filstørrelse35	DE:information	(i	buffer)
Sætte		beskriver36	DE:information	(i	buffer)
Resette		station37	DE:stat.vektor	A:0	

Denne tabel skal blot vise, omtrent hvordan en programmering med BDOS kan se ud, og hvilke muligheder du har med BDOS. Som allerede sagt er den selvfølgelig ikke tilstrækkelig til programmering, da internt kendskab til de enkelte rutiner er nødvendigt.

Selv om BDOS indeholder rutiner som "beskyttelse af noterede station", så er BIOS's opgaver endnu mere specielle. BIOS består som BDOS af en liste over rutiner - som imidlertid udelukkende tjener til ind- og udlæsning af data. Til dels finder man i BDOS rutiner, der har den samme opgave som rutiner i BIOS, således f.eks. "konsolstatus".

BIOS bidrager i ganske afgørende grad til CP/M's succes, for den gør det muligt for CP/M-programmerne at lade gennemføre de udtalt computerafhængige ind- og udlæsninger fra konsollen ved hjælp af specielle rutiner.

BIOS er underafdelte i fire grove områder:

- Skæringspunktet til BDOS og CP/M-programmerne (f.eks. ved SUBMIT-FILERNE),
- Skæringspunktet til de eksterne lagermedier (disketter),
- Ind- og udlæsning via de af BDOS kontaktede ydre enheder,
- Bufferfunktion overfor data, der så til den passende rette tid bliver stillet til rådighed (SUBMIT-filer).

Som du ved, er det muligt at forskyde BDOS og BIOS i lageret. Rutinerne i BDOS bliver tilgået ved at overgive rutinenummeret i C-registeret - den faste lageradresse &0005 bliver så kaldt. Ved kald af BIOS-rutiner ser det lidt anderledes ud: der findes en hoptabel, hvis rækkefølge ganske vist er fastlagt; imidlertid kan denne hoptabels startadresse forskydes. Vi vil dog først se lidt på denne hoptabel:

00 +offset:	JMP	BOOT	;koldstart
03 +offset:	JMP	WBOOT	;varmstart
06 +offset:	JMP	CONST	;spørg om konsolstatus
09 +offset:	JMP	CONIN	;konsolindlæsning
0C +offset:	JMP	CONOUT	;konsoludlæsning
0F +offset:	JMP	LIST	;printerudlæsning
12 +offset:	JMP	PUNCH	;strimmelhuller
15 +offset:	JMP	READER	;hulstrimmellæser
18 +offset:	JMP	HOME	;skrive/læsehoved på spor 0
1B +offset:	JMP	SELDSK	;valg af station
1E +offset:	JMP	SETTRK	;sætte spor
21 +offset:	JMP	SETSEC	;valg af optegnelssesafsnit
24 +offset:	JMP	SETDMA	;valg af databuffer
27 +offset:	JMP	READ	;læse optegnelssesafsnit
2A +offset:	JMP	WRITE	;skrive optegnelssesafsnit
2D +offset:	JMP	LISTS	;spørg om printerstatus
30 +offset:	JMP	SECTAN	;oversættelse af optegnelssesnummer

Offset står her for forskydningen i lageret, som jeg vil gå nærmere ind på senere. Hvis der ikke er brug for en af de angivne funktioner - dette turde ved CPC'en være tilfældet med de to rutiner, der drejer sig om hulstrimler - så bliver de pågældende lageradresser simpelthen udfyldt med:

```
RET ! NOP ! NOP
```

for at lageret ikke skal forskyde sig hen bagved disse rutiner. Hvis der hoppes til disse rutiner, bliver det derfor straks besvaret med RETURN - der sker altså ingenting. For at finde ud af BIOS's startadresse (offset'et), må man vide, at der på startadresse 0 i bunden af systemet er indeholdt et hop til BIOS's varmstartvektor (den anden vektor i tabellen). Derved bliver det let at finde startadressen. Vi vil prøve at gøre det. Start først CP/M, lad så disassembleren:

```
DDT (Amstrad) resp. SID (Commodore)
```

For at lade lagerområdet disassemble fra adresse indtaster vi følgende i computeren:

```
L0000
```

og vi får også straks svaret fra DDT (Amstrad):

0000 JMP AD03
0003 ADD C
0004 NOP
0005 JMP 8F00
0008 JMP B982

og fra SID (Commodore):

0000 JMP F403
0003 RST 07
0004 NOP
0005 JMP D300
0008 NOP

osv. Det, der interesserer os er altså den første linie: Der bliver hoppet til adressen &AD03, resp. F403, BIOS's hoptabel begynder altså på adressen &A300, resp. F403. Det er alt, hvad der er værd at vide for vort program.

Vil vi hoppe til en vektor, så udregner vi hopadressen således:

Adresse:=hopnummer * 3 + &A300 (Amstrad)

eller

Adresse:=hopnummer * 3 + \$F403 (Commodore)

B982

og fra SID (Commodore):

0000 JMP F403
0003 RST 07
0004 NOP
0005 JMP D300
0008 NOP

osv. Det, der interesserer os er altså den første linie: Der bliver hoppet til adressen &AD03, resp. F403, BIOS's hoptabel begynder altså på adressen &A300DEVICE i CP/M 2.0 bliver erstattet af kommandoen STAT DEV: (mere korrekt kunne man sige, at kommandoen STAT er blevet delt i CP/M 2.2, fordi den er for indviklet).

IX.6 DIR (CP/M 2.2 OG CP/M 3.0)

DIR.COM (midlertidig på diskette) og indbygget kommando (CP/M 3.0)

Viser bestemte filnavne og indholdsfortegnelsen

Indtastningsformat:

Indbygget kommando
DIR
DIR A:
DIR FIL.XXX
DIR AB*.*

Midlertidig kommando:

*DIR[OPTION]
DIR FIL.XXX[OPTION]
DIR AB*.*[OPTION]*

Du kan med BIOS-rutinerne overgive parametre eller modtage dem. Hvis værdier skal leveres til BIOS-rutinerne, så bliver 8-bits værdier overgivet til C-registeret, 16-bits værdier venter registerparret BC på. Du kan se en forskel fra BDOS-rutinerne, der venter på parametre i henholdsvis E-registret eller registerparret DE. Parametre, som BIOS rutinerne leverer tilbage til hovedprogrammet, bliver ligesom med BDOS-rutinerne overgivet til akkumulatoren, hvis det er 8-bits værdier, og til registerparret HL, hvis det er 16-bits værdier. De viderekomne har opdaget en dejlig mulighed: da der hoppes til de enkelte BIOS-rutiner via vektorer i RAM'en, kan man naturligvis ændre disse, dvs. lede hen til en rutine af eget fabrikat, som så forgrener sig tilbage til originalprogrammet.

Vi vil gå ind på nogle egenskaber ved bunden af systemet (adresserne 0000 til &0100 på Amstrad, \$0000 til \$01000 på Commodore). Ved hjælp af kommandoen 10 har vi ladet området &0000 til &0016, resp \$0000 til \$0016 disassemble. Et par JMP-kommandoer tiltrækker sig opmærksomheden; særlig vigtig er hopordren på adresse 5. For til adresse 5 har vi rettet vort underprogramskald, når vi ville kalde en BDOS-rutine! *Altså må BDOS i CPC ligge på adresse &8F00, resp. \$D300 på Commodore, da hopordren på adresse &0005, resp. \$0005 har henvist os dertil. Så vi vil se nærmere på dette lagerområde:*

18F00 (Amstrad) ID3000 (Commodore)

Vi må hoppe en gang til, nemlig til adresse &95A2, resp. \$D9A4.

L95A2 (Amstrad) LD9A4 (Commodore)

For at gøre det kort: der hoppes også til adresse &9F06, resp. \$ECF2 og så til adresse &9F11, resp. \$00E5. Så begyndes med afkodningen af C-registeret, for at hvilken rutine, der skal hoppes til. Det ville bestemt føre for vidt at følge dette forløb videre.

Hermed vil vi afslutte vor "lille" indføring i de vigtigste programmer og utilities, der på Commodore hører under "Additional Utilities"-pakken.

7.9. SETUP-kommandoen (kun Amstrad)

SETUP-kommandoen findes kun i CP/M 2.2, men den er også af interesse for CPC 6128-ejere, da man kan køre CP/M 2.2 på 6128'eren.

SETUP er menustyret og spørger selv om de enkelte punkter i dialogen på skærmen.

Lad os dog kalde kommandoen **SETUP**. Så gennemgår vi de enkelte underpunkter. Efter at du har indtastet **SETUP**, fremkommer følgende på skærmen:

SETUP V2.0

**** Initial command buffer: empty**

Is this correct (Y/N):-

Spørgsmålet "Is this correct (Y/N)" bliver nu stillet ved hver komponent af **SETUP**-kommandoen. Besvares det med Y for ja, så bliver de viste parametre overtaget; ved indlæsning af N for nej, kan du korrigere endnu en gang.

Du kan se, at vor "Initial command buffer" er tom (empty). Når du skriver noget ind i Initial command buffer'en, bliver denne buffer udført ved opstart af CP/M. Hvis du efter at have bootet CP/M vil have en indholdsfortegnelse over systemdisketten på skærmen, for at få et overblik over de midlertidige kommandoer, så kan du gøre det med Initial command buffer'en. Du svarer på spørgsmålet "Is this correct" med et "nej", for at komme til at ændre din Initial command buffer. Efter at du har trykket på N-tasten, fremkommer på skærmen:

Enter new Initial command buffer:-

Vi vil lade bufferen med kommandoen **DIR**. For at en linie kan blive udført, må den afsluttes med et "linieskift". Hvordan kan man indtaste et lineskift her? For alle ASCII-tegn, der er mindre end 32, skal man anvende kontrolsekvenserne ↑A til ↑Z, såvel som ↑(,↑/,↑),↑ og ↑-. For at indtaste et lineskift, må dubrige tegnet ↑M. (Lineskift har ASCII-koden 13, og bogstavet M er det trettende i alfabetet!) Indlæs det følgende i Initial command buffer'en:

DIR↑M

Efter tryk på **ENTER**-tasten, fremkommer igen spørgsmålet **Is this correct (Y/N):-**, hvorpå du skal svare med et Y for ja.

Derefter vises **SIGN-ON-STRING** og spørges om anvisninger hertil. I **SIGN-ON-STRING** er der en mængde styretegn, der bevirker indstilling af 80-tegns funktionsmåde og baggrunds- og kantfarve. Også her må ASCII-tegn mindre end 32 indlæses via ↑(Code). De rette ASCII-koder til indstilling af baggrundsfarven, indstilling af 80-tegns funktionsmåde osv. findes i brugsanvisningens ASCII-tegn tabel.

Efter du har defineret både Initial command buffer'en og **SIGN-ON-STRING**'en, kan du omdefinere tastaturets taster eller forsyne det med ekstra tegn. Hertil tjener punktet:

Keyboard translations

DE angivelser, du gør i dette menupunkt, er de samme som i **BASIC**-kommandoen **KEY DEF**. De rette kodetabeller og tastaturkoderne findes i brugsanvisningen. Også det næste menupunkt, **Keyboard expansions**, har de samme virkninger som **BASIC**-kommandoen **KEY**.

Derefter vises standarddefinitionerne for input/output-udstyret, som kan editeres. Der følger nogle tekniske angivelser til disktestationsstyringen (hvor længe motoren skal køre osv.), samt angivelser for processorens ind- og udlæsningskanaler. Efter du har kvitteret for disse angivelser via tastaturet, stilles spørgsmålet:

Do you want to update your System disc (Y/N):—

Besvarer du spørgsmålet med **Y**, så bliver alle opgivelser sikret på disketten (ved tryk på **N**-tasten tabes alle de nye angivelser). Det sidste spørgsmål, som CP/M retter til dig, er:

Do you want to restart CP/M (Y/N):—

Dermed kan du direkte afprøve virkningerne af dine ændringer på skærmen, idet du svarer **Y** for ja. Så bootes CP/M påny, og hvis du har ændret **SIGN-ON-STRING**'en eller **Initial command buffer**'en, bliver ændringerne straks udført.

Det er jo svært, at uddebattere CP/M, særligt når man selv vil programmere i CP/M. Vi håber, at denne bog har været og vil være dig til god hjælp.

7.a. CP/M internt C-128

7.1a. Generelt om CP/M 3.0 på C-128

CP/M 3.0 er ikke bare CP/M 3.0 - det opdager man igen og igen. Dog må man sige, at fabrikanterne af nye computere holder sig til CP/M-normen, der er blevet foreskrevet af Digital Research. De hardwaremæssige kendsgerninger gør det nødvendigt at ændre et ene eller andet i CP/M eller tilføje noget. Vi vil se på nogle særlige egenskaber ved CP/M 3.0 på Commodore 128, for at kunne udnytte driftssystemet optimalt.

Den vigtigste egenskab er, at CP/M er på en diskette, der både kan læses af 1541'eren, 1570'eren og 1571'eren. Med 1571'eren er den mangel, at man ikke kan udnytte - den dobbeltsidige læsning og skrivning. Man må skifte disketter, fordi COM-filerne ligger på begge diskettesider.

7.2a. Systemdiskette for 1571-ejer

Som ejer af en 1571 bør du benytte dens egenskaber. Du sparer ikke blot arbejde, men også tid og dine nerver, hvis du opretter en passende 1571 systemdiskette. Derfor må du formattere en ny diskette dobbeltsidigt med FORMAT. Det er det vigtigste trin ved fremstillingen af en 1571-systemdiskette. Det er ikke nok at formattere begge sider enkeltsidigt.

Har du gjort det, så må begge diskettesider af originaldisketten kopieres på den nye systemdiskette. Læg derfor side A med påskriften "CP/M System Disk" i stationen og indtast følgende PIP-kommando:

PIP E:=A:.(resp. PIP B:=A:*.*)*

CP/M 3.0 opfordrer dig til regelmæssigt at veksle disketterne. Følg disse opfordringer, og husk, at diskette E er måldisketten, diskette A kildedisketten. Er denne disketteside blevet kopieret, så læg igen originaldisketten med side A ind. Nu skal den anden side af disketten kopieres. Hertil indtaster du følgende kommando:

PIP A:=E:.(resp. PIP B:=E:*.*)*

Det er påkrævet - også hvis du to stationer - at disketten bliver snurret rundt i station A før den egentlige kopieringsproces. Nu er diskette A måldisketten, kildedisketten bliver betegnet med navnet E. Følg opfordringerne til at skifte disketterne.

Er proceduren afsluttet, så har du en systemdiskette, der er beskrevet på begge sider, og som fra dette øjeblik bør anvendes. Det er lettere at arbejde med en sådan diskette, hvorpå alle CP/M-kommandoer er til rådighed.

Allerbedst er det, hvis du tager endnu en kopi af systemdisketten og opbevarer den sikkert i et skab.

7.3a. Den virtuelle station E:

Vi har læst om det og oftere anvendt E: som stationsbetegnelse. I CP/M 3.0 på Commodore 128 er det muligt at tilslutte indtil fire diskettestationer. Det er også forudsat i BASIC. Diskettestationerne har apparaturadresserne 8 til 11, som kan indstilles i resp. på stationen (1571). I CP/M kalder man disse stationer med stationsbetegnelsen, som består af et bogstav og et kolon. Med fire mulige stationer betyder det, at vi kan have stationer med betegnelserne A:, B:, C: og D:.

Hvad betyder den virtuelle femte station med betegnelsen E:? Ordbogen siger om ordet "virtuel":

"som har evne eller mulighed for at være til stede, tilsyneladende"

Det behøver man ikke at føje noget til! Den femte station kan man til enhver tid kalde, men den er ikke til stede i virkeligheden. Den erstattes af station A. Dog skelner CP/M altid mellem, om det går til station A eller station E. Herunder ved CP/M nøjagtigt, om station A er station A, eller om den for øjeblikket spiller station E. Bliver der forlangt den anden af de to stationer, end den som lige repræsenteres af station A, så opfordrer CP/M til skift af disketterne og til tryk på RETURN-tasten.

Takket være denne geniale løsning, er det muligt i CP/M at kopiere disketter med een station. Du kan udføre kommandoer, der ligger på en anden diskette, eller rokere med filer, der er på en anden diskette. Disse ting ville ellers ikke være mulige med en enkelt station, eller med specielle kommandoer. Du bør derfor vænne dig til den "virtuelle station" - selv om den også kun er "virtuel".

De læsere, der har rådighed over to stationer, har en fordel i forhold til andre CP/M-brugere. Kopieringen går meget lettere. Man kan råde over flere data på samme tid osv. Du bør også udnytte denne anden (eller måske endda tredje eller fjerde) station under arbejdet. Alligevel kan det være praktisk at benytte sig af den virtuelle station.

7.4a. COPYSYS på Commodore 128

På enhver "normal" CP/M 3.0-computer bliver systemet, som består af de to øverste spor 0 og 1, kopieret ved indtastning af kommandoen COPYSYS. Det er ikke tilfældet med Commodore 128: kopiering af systemet er lettere, fordi man har anbragt det i de to filer CPM+.SYS og CCP.COM, som man kan kopiere med PIP.

Hvis du har vænnet dig til kommandoen COPYSYS, er det ikke så slemt. Commodore 128 viser på skærmen, at du ikke kan arbejde med kommandoen COPYSYS; der eksisterer nemlig ingen fil ved navn COPYSYS.COM, så du får en fejlmelding i retning af

COPYSYS?

Som omtalt tidligere, bliver systemet i CP/M 3.0 kopieret med de følgende tre kommandoer:

- *FORMAT*
- *PIP E:=A:CPM+.SYS*
- *PIP E:=A:CCP.COM*

Alle indbyggede kommandoer, og bootrutiner etc. bliver kopieret. Ligeledes bliver bootsektoren oprettet på spor 1, sektor 0. Efter indtastning af de tre kommandoer er systemdisketten færdig - dvs. den kan bootes. Men da der ikke er en eneste midlertidig kommando på disketten, er den relativt nytteløs. Du kan imidlertid kopiere alle de COM-filer, du har brug for. Således kan du spare plads på disketten, når du ikke kopierer alle COM-filer.

7.5a. Statuslinien

Den nederste linie på billedskærmen er den såkaldte statuslinie. Her kan du få de vigtigste informationer, der kommer fra systemet. Således bliver du for eksempel opfordret til at indlægge diskette A eller diskette E og trykke på RETURN.

Ingen, der har arbejdet med 3.0 på Commodore 128, er uvidende om, at der i nederste højre hjørne kommer alle mulige tal og bogstaver, der fra tid til anden skifter.

Her gemmer sig et system. På dette sted angives den såkaldte Disk Status. Man kan sige, at det bliver vist, hvilken blok, der netop bliver læst eller skrevet på. Formatet for denne diskettestatusvisning ser sådan ud:

O Ltt ss

Her har de enkelte betegnelser følgende betydning:

- *O* = Operation, *R*=Read (læse) og *W*=skrive (*W*)
- *L* = Fysisk station (*A*, *B*, *C* eller *D*)
- *tt* = tocifret tracknummer (spor) på den blok, der læses eller skrives på
- *ss* = tocifret sektornummer på den blok, der læses eller skrives på

Spor- og sektorangivelse bliver normalt skilt af et mellemrum. Hvis den indlagte diskette er i det såkaldte MFM-format, altså dobbeltsidigt formatteret, så bliver spor- og sektorangivelse ved tilgang til den anden side skilt af en bindestreg (-).

Hvis denne diskettestatuslinie forstyrrer dig, kan du frakoble den med tastkombinationen <CTRL>-<RUN/STOP>. Så må du holde Control-tasten nede. Med den tastkombination kan man igen tilkoble statusvisningen.

Hvis du vil gentage den sidste kommandolinie, så er det tilstrækkeligt, at du trykker på cursor-down-tasten nederst til højre på Commodore tastaturet. Den separate cursor-blok forbliver herunder inaktiv. Den har en anden vigtig funktion. Er *DIR **. * den sidste kommando, som er indtastet, fremkommer teksten på skærmen, når cursor-down-tasten trykkes ned. Der udføres ikke noget automatisk RETURN - du kan editere den sidste indtastning for at rette fejl o.a.

7.6a. Disketteformaterne

1570'eren og 1571'eren er kan læse forskellige formater i forbindelse med CP/M. Ikke eet ord af det følgende gælder for 1541'eren!

Controlleren - der lader sig programmere på 1570'eren og 1571'eren - muliggørde forskellige formater. De følgende formater bliver understøttet af begge stationer (de dobbeltsidige formater naturligvis kun på 1571):

Osborne DD	(1024 bytes/sekter, enkeltsidigt, 5 sektorer/spor)
Epson QX10	(512 bytes/sekter, tosidigt, 10 sektorer/spor)
IBM-8 SS (CP/M 86)	(512 bytes/sekter, enkeltsidigt, 8 sektorer/spor)
IBM 8 DS (CP/M 86)	(512 bytes/sekter, tosidigt, 8 sektorer/spor)
KayPro II	(512 bytes/sekter, enkeltsidigt, 10 sektorer/spor)
KayPro IV	(512 bytes/sekter, tosidigt, 10 sektorer/spor)

Seks formater bliver direkte bakket op. Man kan realisere alle andre formater gennem programmering af controlleren; men hvordan man programmerer denne controller kan ikke forklares i en bog om CP/M. Anden lektur, som f.eks. Floppybogen vil give yderligere oplysninger.

Når du lægger en diskette ind og lukker lemmen, så kører motoren et kort øjeblik. Den intelligente controller forsøger at finde ud af formatet. Herunder tjener angivelserne af bytes pr. sektor og sektorer pr. spor til lokalisering af formatet. Er det ikke muligt at bestemme formatet alene ved hjælp af disse angivelser, så fremkommer en lille box i nederste venstre hjørne med de mulige formater. Ved at scrolle i dette vindue kan meddele systemet, hvilket format der gælder på den nyindlagte diskette. Denne scrolling bliver foretaget med cursor-til-venstre og cursor-til-højre tasterne. Når du har fundet det rigtige format i vinduet, kan du bekræfte det ved at trykke på RETURN-tasten.

Hvis du fremover kun vil anvende det format, så tryk på Control- og RETURN-tasten på samme tid. Så noterer CP/M det godkendte format og spørger ikke mere om det.

7.7a. Tastaturet

Tastaturet på en Commodore 128, der kører i CP/M, er lidt forskelligt fra det tastatur, man kender fra BASIC. Cursortasterne gør ikke alt, hvad de skal. Trykker man på cursor-ned-tasten nederst til højre på tastaturet, så gentages den sidst indtastede kommando. Det har vi allerede nævnt. Hvis man trykker på den tilsvarende tast i den øvre, separate cursorblok, så sker der ikke noget.

En særlig egenskab bør nævnes. Hvis du vil boote systemet på ny, behøver du ikke at lede efter RESET-tasten, du kan opnå den samme effekt med tastkombinationen <CTRL>-<ENTER>, hermed menes ENTER-tasten til højre i decimaltastaturet.

Hver tast på det store 128-tastatur har indtil fire værdier. Disse værdier er den "ushiftede" værdi, den "shiftede" værdi, Control-værdien og CAPS-LOCK-værdien. Som du kan se, skal CAPS-LOCK værdien ikke altid være den samme som den "shiftede" værdi. Den "ushiftede" værdi får du ved simpelthen at trykke på den pågældende tast, når CAPS-LOCK-tasten ikke er rastet ind. Så fremkommer med sort skrift markerede tegn på skærmen. ASCII/DIN-tasten har ingen virkning på tastaturafkodningen i CP/M.-funktionsmåden.

Den "shiftede" værdi fremkommer, ved at trykke på en tast, medens shift-tasten holdes nede. Control-værdien fås tilsvarende ved samtidig at trykke på Controltasten og en anden tast (som i det allerede omtalte eksempel med ENTER-tasten). CAPS-LOCK-funktionsmetoden nås ved at nedtrykke Commodore-tasten og forlades på samme måde ved endnu et tryk. Modsat hvad der sker, når man trykker på CAPS-LOCK, bliver i Commodore-funktionsmåden kun bogstaverne "shiftet", men ikke cifrene og de andre taster. Det er en simuleret CAPS-LOCK-funktionsmåde, sådan som den bør være.

Rækken af ciffertaster ovenover bogstavstasterne tjener, når man samtidig trykker på Control-tasten, til valg af tegnfarver, cursorfarven ændrer sig tilsvarende (foreløbigt). Du kender disse farver fra Commodore-funktionsmåden. Du kan kun vælge otte - ikke alle de seksten tænkelige farver. Hvis du trykker på decimaltastaturets ciffertaster i forbindelse med Control-tasten, ændres baggrundsfarven på 80-tegns skærmen tilsvarende.

Den, der ikke har rådighed over _-tegnet, fordi han ved boot'ningen af CP/M har omstillet til danske tegn, kan få dette tegn ved at trykke på pund-tasten. Pil-opad-tasten frembringer i "ushiftet" tilstand tegnet ↑ og i Control-funktionsmåden den lodrette streg I, som bruges i mange kommandoer. Pil-til-venstre-tasten ovenover Control-tasten frembringer understregningstegnet.

7.8a. Specialfunktioner

Det er ment at belægge tasterne på Commodore 128 med alle mulige forskellige værdier i CP/M 3.0. Dertil findes endda to metoder. Hvordan man ændrer indholdet af HELP-tasten og andre ting, erfarer du i følgende beskrivelse.

Alle specialfunktioner, der beskrives i det følgende når man ved samtidig at trykke på Control- og den højre SHIFT-tast. Hertil kommer den tredje tast, som vælger funktionerne. Det er vigtigt at trykke på den højre SHIFT-tast, den venstre har ikke samme effekt. De følgende tre taster tjener til dette funktionsvalg:

- * ALT-tasten
- * Cursor-til-højre-tasten i cursor-blokken
- * Cursor-til-venstre-tasten i cursor-blokken

Groft sagt, kan man ændre de forskellige tasters ASCII-værdier. Hvis en tast har større værdi end `_80` (128 decimalt), så har den særstatus. En ASCII-værdi højere end 127 ændrer for- eller baggrundsfarve på 40- eller 80-tegns skærbilledet, eller kalder en fastprogrammeret streng frem, omtrent som `HELP`-tasten. Her en tabel over de ASCII-koder, der er større end 127:

80-9F: Funktionstreng 0-15
A0-AF: 80-tegns forgrundsfarver
B0-BF: 80-tegns baggrundsfarver
C0-CF: 40-tegns forgrundsfarver
D0-DF: 40-tegns baggrundsfarver
E0-EF: 40-tegns kantfarver
F0-FF: Specialfunktioner

7.8.1a. Hvordan aktiveres specialfunktionsmåden

Man kan ændre baggrundsfarve (på 80 tegn) via cifertasterne 1-9 i forbindelse med `Control`-tasten. Det betyder, at tasterne skal være belagt med værdierne `_A0-_A7` resp. `_B0-_B7`. Den første tretast-kombination er som følger:

`<CTRL><HSHFT><ALT>`

`<HSHFT>` står for højre `SHIFT`-tast. Denne tastkombination er en ren fingerymnastikøvelse, men det er også meningen, at man ikke kalder disse funktioner ved en fejltagelse. Tastkombinationen tjener til ind- og udkobling af vore med specialfunktioner belagte taster. Hvis du en gang har aktiveret denne tastkombination, vil cursorfarven ikke ændre sig, hvis du trykker på `<CTRL>`-tasten sammen med en af cifertasterne. Man kan heller ikke være muligt for dig at ændre baggrundsfarve på 80-tegns skærmen. Også `HELP`-tasten (prøv selv) er ude af funktion. Efter endnu en gang at trykke på denne tastkombination, er specialkoderne igen tilladt.

7.8.2a. Hexcode-editoren

Lad os se på ASCII-værdierne af nogle taster og naturligvis også ændre dem. Vi begynder med mellemrumstangenten; som vi kender koden på. Tryk på den følgende tastkombination, der gør det muligt at editere en tasts ASCII-kode. På den måde kan du også simulere et dansk tastatur!

`<CTRL><HSHFT><CURSOR-TIL-VENSTRE-TAST>`

Efter du har overstået fingerøvelsen uden brækkede fingre, fremkommer i statuslinien et lille vindue med lys baggrund. Du bliver nu opfordret til at trykke på den tast, hvis ASCII-kode, du vil vide og eventuelt ændre. Kombinationerne med `<CTRL>` og `SHIFT`-tasten er også tilladt her, ellers kan du ikke nå de andre tre tastværdier. Lad os trykke på mellemrumstangenten.

På skærmen fremkommer den hexadecimale ASCII-værdi for den tast, der blev trykket på, i vort tilfælde mellemrumstangenten med ASCII-koden 32, hvad der svarer til 20 i det hexadecimale system. Der blinker en cursor i vinduet. Det er tegn på, at CP/M venter en indtastning. Alle cifre og bogstaverne A-F er tilladt. Enhver anden tast afbryder straks indlæsningsfunktionsmåden - den angivne værdi bliver ikke ændret, og vinduet forsvinder. Ellers kan du ændre tastens værdi. Når du har ændret begge pladser i den tocifrede kode, bliver den nye kode automatisk lagret. Det er altså ikke nødvendigt at trykke på RETURN-tasten.

Du kan tydeliggøre det ved at ændre koden 20 til 21. Vælg altså med <CTRL><HSHFT><CURSOR-T.V.> og <mellemrumstangent> ASCII-koden for mellemrumstangenten. Der fremkommer den aktuelle værdi, 20. Derefter trykker du simpelthen 2 og 1. Vent et øjeblik, det varer lidt, inden systemet har afkodet den nye ASCII-værdi og lagret den i tastaturmatricen. Så forsvinder vinduet straks, og deraf kan du se, at indlæsningen er afsluttet. Til kontrol fremkommer enhver indlæst tastværdi til højre for den gamle ASCII-værdi. Er der skrevet forkert, så tryk på RETURN-tasten, eller en anden tast, så forlader du editeringsmodus uden ændringer.

Lad os vende tilbage til ændringen. Har du trykket på <MELLEMRUM>? Så har du sikkert undret dig over, at der ikke fremkommer noget mellemrum på skærmen, men derimod et udråbstegn. På den måde kan du ændre alle værdier, også det for RETURN-tasten. Da det er temmelig upassende at have et udråbstegn liggende på mellemrumstangenten, vil vi ændre koden tilbage til 20. Nu ved du jo, hvordan det sker, ikke sandt?

En praktisk anvendelse faldt mig ind, da jeg opdagede, hvor nyttig denne mulighed er. Hvis jeg ændrede decimaltastaturets SHIFT-værdier, så havde jeg ikke kun otte, men alle seksten baggrundsfarver.

For at opnå det, må man belægge alle otte SHIFT-værdier på decimaltastaturet med værdierne _B8 til _BF, fordi tastværdierne <CTRL>-1 til <CTRL>-8 er belagt med værdierne _B0 til _B7. Hvis du har gjort det, så står alle seksten mulige baggrundsfarver til rådighed for din 80-tegns skærm. Du kan også som 40-tegns-skærm-bruger programmere disse taster til 40-tegns-farvekoder. Find blot de tilsvarende værdier i tabellen. Hvad der er mere interessant, er koderne _80 til _8F og _F0 til _FF. Disse koder ser vi på nu.

7.8.3a. Streng-editoren

Man kan belægge enkelte taster med strenge - det er kæder af tegn. Der er stadig en af de tre specialfunktionstaster tilbage. Det er tastkombinationen:

<CTRL><HSHFT><CURSOR-T.H.-TAST>

Når du har gennemført denne tastekombination, fremkommer et tydeligt større vindue i statuslinien. En lille bemærkning fra forfatteren: Du har måske lagt mærke til, at CP/M altid vælger to farver til vinduesbaggrunds- og tegnfarve, der træder tydeligt frem fra baggrunden. På den måde er man sikker på en korrekt indtastning, og på at se vinduet.

Ligesom ved hexkode-editoren fremkommer der altså med streng-editoren et vindue i statuslinien. Hvis du trykker en tast, der har en ASCII-kode værdi mindre end `_80` eller større end `_9F`, så forsvinder vinduet igen, og funktionsmåden er afsluttet. Snupper du imidlertid en tast, der kan opvise en kode imellem `_80` og `_9F`, så bliver det aktuelle indhold af strengen vist på skærmen. Vi kender en tast, der er forsynet med en streng - rigtigt, det er HELP-tasten. Tryk en gang på denne HELP-tast, og straks fyldes skærmen med tekstindholdet:

›Help‹

Tegnet "større end" (`>`) udgør den venstre begrænsning af tegnkedden. Tegnet mindre end tilsvarende den højre. Cursoren står på det højre tegn og blinker. Lad foreløbig tastaturet være i fred. Vi vil først klargøre, hvordan man kan bevæge cursoren, slette og indføje tegn. Det skal den følgende tabel tjene til:

‹CTRL›‹HSFHT› & en af de følgende taster:

RETURN - afslutter indlæsningen

--> - cursor mod højre i vinduet (tasten på cursorblokken)

--< - cursor mod venstre i vinduet (tasten på cursorblokken)

+ - indføjning af et tegn på cursorpositionen

- - sletning af tegnet under cursoren

Man vænner sig til denne måde at editere på. Hvis du skulle komme til at trykke på RETURN-tasten uden det foreskrevne `‹CTRL›` og `‹HSFHT›`, så bliver dette RETURN lagret og også udført, når tastfunktionen kommer til udførelse - pas altså på. Hvis du derpå skriver videre, bliver der skrevet oven i tegnene under cursoren.

Hvis du har nået den højre grænse for vinduet og alligevel skriver videre, så tilføjer systemet automatisk en plads og skubber således hele grænsen en plads, mod højre.

Der er flere tegn til rådighed, end vinduet i statuslinien lader formode. Vinduet råder over 22 tegn; hvis du indføjer mellemrum tit nok, så kan du vandre med cursoren til den højre grænse. Så vil du se, at vinduesindholdet scrolles. Skulle du herunder nå til den virkelige højre grænse for strengen, så springer editoren igen tilbage til den første position i strengen.

Du skulle tage at gennemprøve de enkelte funktioner i cursorbevægelserne og slette- og indføjningsmulighederne. På den måde lærer du bedst at omgås tasterne. Husk at der i CP/M-funktionsmåde ikke er nogen repeat-funktion. At holde en tast nedtrykket bevirker altså ikke nogen gentagelse af den. Her får du besked om, på hvilke taster, der allerede er defineret strenge:

F1:	"F1"	F5:	"F5"
F2:	"F2"	F6:	"F6"
F3:	"DIR<CR>"	F7:	"3 June 85<CR><LF>"
F4:	"DIR"	F8:	"F8"
HELP:	"Help"		

Du kan se, at det hovedsageligt er funktionstasterne, der er forudbelagt. Tænk på, at der kan fordeles 32 strenge på tasterne - det er praktisk at tildele funktionstasterne, som man kalder sammen med <CTRL>-tasten, andre strenge, så man har en tre-dobbel belægning med funktionstasterne. Tænk også ubetinget på, at den kode, der fremkaldes ved indkobling af CAPS-LOCK-funktionsmåden, som du slår til ved at trykke på Commodore-tasten, igen danner en ekstra kode. Her bør man ikke foretage nogen omdefinering, da det kan føre til misforståelser. Her er tillige en liste over foruddefinerede strenge, sådan som systemet opretter dem ved bootningen:

80:	"F1"	90:	"F17"
81:	"F2"	91:	"F18"
82:	"dir<CR>"	92:	"F19"
83:	"dir"	93:	"F20"
84:	"F5"	94:	"F21"
85:	"F6"	95:	"F22"
86:	"F7"	96:	"F23"
87:	"3 June 85<CR><LF>"	97:	"F24"
88:	"F9"	98:	"F25"
89:	"F10"	99:	"F26"
8A:	"F11"	9A:	"F27"
8B:	"<SCREEN LEFT>"	9B:	"F28"
8C:	"<SCREEN RIGHT>"	9C:	"F29"
8D:	"<SCREEN LEFT>"	9D:	"F30"
8E:	"<SCREEN RIGHT>"	9E:	"F31"
8F:	"F16"	9F:	"Help"

Koderne 8B til 9E er tilsyneladende forblevet tomme, men de er belagt med styrekoder til 40-tegns skærmen, derfor automatisk scrolling.

I den første tabel står der, ved siden af ASCII-området F0 til FF, ordet "specialfunktioner". Disse specialfunktioner er f.eks. rebooting - altså genstart af systemet - som nås ved at trykke på tasterne <CTRL> og <ENTER> på een gang.

Gå over i hexkode editoren, og du vil se, at denne tast har ASCII-koden FF. Her er en opstilling af specialfunktioner, som er defineret ifølge CP/M:

F0:	Til-/frakobling af diskettestatus
F1:	Standse/fortsætte skærmvisningen (som <NO SCROLL>)
F2:	40-Tegns
F3:	Skærbillede til venstre (40-tegns)
F4:	Skærbillede til højre (40-tegns)
F5:	MFM Unlock
FF:	Rebooting af systemet

Tre af syv funktioner ligger allerede på forskellige taster. Funktionen F0 er på tastkombinationen <CTRL><RUN/STOP> og bevirker, at diskettestatusvisningen i statuslinien enten bliver tilladt eller forhindret. Den anden funktion: F1 bliver kaldt frem af <NO SCROLL>-tasten. Ved det første tryk fremkommer ordet Pause i statuslinien, og skærmvisningen standses. Når du igen trykker på denne tast, bliver udlæsningen, resp. udførelsen fortsat. Den sidste funktion: FF befinder sig som allerede omtalt på tasten <ENTER> i forbindelse med <CTRL>-tasten.

7.9a. KEYFIG og hvordan man bruger den

De lige beskrevne muligheder er residente, dvs. du kan ændre tastatur og strenge efter behag, men ikke særligt bekvemt. Dog hertil findes også kommandoen KEYFIG, der forenkler al besværet med en komfortabel menuteknik. Indtast engang KEYFIG, programmet melder sig så direkte:

```
C128 SOFT KEYBOARD PROGRAM
3 June 1985

Welcome to the Commodore C128 Keyboard
Definition Program. Do you want help?
```

Et sådant tilbud skal man ikke sige nej til, når man for første gang kalder et program. Derfor svarer vi med et "Y" for ja. På skærmen fremkommer en række tilbud om hjælp:

```
Help is available on the following
topics:
--> done help <--
-->General Usage<--
-->Setting up your work file<--
-->What to do with your work file<--
-->Key values<--
-->Selecting a key to edit<--
-->Logical/Physical colors<--
-->Editing keys<--
-->Assigning/Editing Strings<--
-->Assigning Colors<--
-->Assigning Special Functions<--
-->Assigning HEX values<--
-->Finishing up<--
-->For experts only<--
```

Use the up and down arrows to scroll through the menu; type the return key to select the topic on which you want help.

Se nærmere på de forskellige hjælpetekster, som tjener til almindelig forståeliggørelse. Ved hjælp af cursor-op og -ned tasterne kan du bevæge det markerede felt op og ned, dette går kun med tasterne i cursor-blokken. Benyt også disse taster ved et senere valg. RETURN-tasten tjener til bekræftelse af det udvalgte felt. Det sidste menupunkt fortæller, hvad du netop har lært: at det går meget hurtigere med <CTRL><HSHFT>-kombinationen. Har du fået hjælp nok, vælges første menupunkt "done help", for at komme til det oprindelige KEYFIG-program.

Vil du ombelægge tastatur, eller omdefinere strenge, eller skabe helt nye, simulere det danske tastatur etc., så er det ikke meningen, at arbejdet skal gøres på ny hver gang. Det vil altså være ønskeligt at lagre den ændrede tastaturbelægning. Det kan du gøre med KEYFIG-programmet. Efter at du har forladt HELP-funktionsmåden, fremkommer følgende på C128's skærm:

```
From which of the following sources of
key definitions would you like to work:
Default definitions
Definitions on the CP/M boot disk
Current dfinitions
(Your previous work file)
```

Nu kan du vælge, hvilken tastaturbelægning, der skal være grundlag for dit fremtidige arbejde. Angivelsen i parentes fremkommer derfor kun, hvis du ikke starter KEYFIG for første gang. Du kan vælge, om du vil arbejde med:

```
Tastaturets standardbelægning
At hente tastaturbelægningen fra CP/M-disketten
Den øjeblikkelige tastaturbelægning
Den seneste tastaturbelægning
```

Alt efter omstændighederne henter systemet tastaturbelægningen fra disketten eller ej. I den tredje linie vises, hvilket tastatur der bliver arbejdet på.

Vil du have indblik i, hvor komfortabel denne KEYFIG-utility er, så skal du, efter at følgende menupunkter er vist:

```
Edit a key definition
Set up logical <--> physical colors
Exit and save your work file
```

vælge det andet menupunkt. Så får du (efter at du har forladt 80-tegns skærmen) meget imponerende vist en farvepalet på skærmen, der forestiller logiske og fysiske farver. Du kan ændre denne tildeling ved indtastning af bogstaver. Ændringerne bliver omgående vist.

For igen kort at forklare, hvordan du kan ændre tastaturbelægningen: vælg menupunktet "Edit a key definition". I øvrigt bør du aldrig prøve at lagre en tastaturbelægning, når du ikke er helt færdig, der er tid nok. På billedskærmen fremkommer:

Editing: no key
This key has the 4 values shown below
normal --->
CMDR SHFT->
SHIFTED-->
CONTROL-->
(done editing - exit and save work file

Her kan du køre rundt med markeringsfeltet ved hjælp af cursortasterne i cursor-blokken. Tryk på en tast, og tastens fire forskellige belægningsvisninger vises på skærmen. CMDR SHFT betyder i øvrigt CAPS-LOCK funktionsmåde, som du får ved at trykke på Commodore-tasten.

Særligt imponerende finder jeg det, at KEYFIG f.eks. for til-venstre-tasten øverst til venstre på tastaturet skriver teksten "LEFT ARROW NEXT TO 1" eller for pundtasten "BRITISH POUND" osv. En tast, som overhovedet ikke repræsenterer et printbart ASCII-tegn bliver altså forklaret med en omskrivning til tekst. Også for andre ikke-printbare koder, som f.eks. delete, bliver der udlæst tekster. For delete er det således "RUBOUT". Hvis du vil ændre en af disse fire tastbelægningsvisninger, så bevæg markeringsfeltet derhen og tryk på RETURN-tasten.

ASSIGN a STRING (more than one character)
ASSIGN new (single) character
ASSIGN a color
ASSIGN hex value
ASSIGN a SPECIAL FUNCTION
don't modify this key

Hvor vi med tre-tast kombinationen måtte slå op i vor tabel, hvilken kode vi skulle tildele, så tager KEYFIG dette arbejde fra os. KEYFIG er altså væsentlig behageligere. For den rutinerede bruger kan "kragefodsteknikken" (således kaldet på grund af de stærkt spredte fingre) betegnes som væsentlig hurtigere, fordi man ikke skal opsøge den ønskede funktion ved hjælp af cursortasterne. Det afgør du selv! Lagringen på diskette kan kun KEYFIG klare for dig. Du kan vælge, om den nye tastaturbelægning skal lagres på en separat diskette, eller på CP/M-systemdisketten. Lagrer du belægningen på CP/M-systemdisketten, så bliver den automatisk indlæst efter boot'ningen. For at lagre den, skal du vælge følgende punkter af hovedmenuen:

Exit and save your work file
on CP/M boot disk

Efter lagringen spørger KEYFIG dig helt kærligt:

Do you want to do anything else []

Det spørgsmål kan du besvare med et Y for ja eller et N for nej. Hvis du forlader programmet med et CTRL-C, så bliver der også her stillet et sikkerhedsspørgsmål, for ikke at gå ud af KEYFIG ved en fejl.

8. Korrektion af PIP (Amstrad)

8.1. Fejlbeskrivelsen

I PIP-kommandoen findes en ting, der kan drive en til vanvid. Hvis man arbejder med PIP for at kopiere disketter og så skifter diskette uden at foretage en varm-start med Ctrl-C, så får man meldingen "BDOS ERROR R/O". Det er meget ærgerligt, da man nu må trykke på Ctrl-C tasten og indlæse alting på ny.

Da vi ønsker, at du skal undgå ærgrelser i fremtiden, vil vi give dig en løsning, hvormed meldingen "BDOS ERROR R/O" for evigt kommer til at tilhøre fortiden. Det er, fordi du får direkte indblik i, hvordan man anvender DDT, og hvordan man kan kigge ind i CP/M-COM-filerne og ændre dem.

8.2. Fejlkorrektion

Den ovenfor omtalte ulempe vil vi lade forsvinde, idet vi retter lidt i PIP-programmet. Det er bedst, hvis du kun foretager denne korrektion på din sikkerhedskopi. - Prøv at forestille dig, at du gør en fejl og ikke længere har nogen kopi af CP/M!

Vi vil nu lade CP/M-funktionen Disk-reset kalde fra begyndelsen, for at den eventuelt nyindlagte diskette kan blive initialiseret. Til den ende skal vi rumstere rundt med PIP.COM-filen, bedst med DDT.

```
Indtast:  
DDT PIP.COM
```

Programmet starter ligesom enhver anden COM-fil på adressen \$0100. Du kan lade området disassemble:

```
-I0100  
0100 JMP 04CE  
0103 RET
```

PIP-programmets start ligger altså på 04CE. Det tilføjes endnu et Disk-reset herimellem, hvorfor vi kobler denne lille rutine på enden af PIP-programmet og så hopper til den ved begyndelsen af PIP. Enden af PIP-programmet ligger på \$1DB1. Vi indlæser følgende rutine:

```
-a1db2  
1DB2 MVI C,0D  
1DB4 CALL 0005  
1DB7 JMP 04CE  
1DBA (tryk RETURN)
```


Rutinen kalder altså Disk-reset rutinen. Nu må vi tillige sige i begyndelsen af PIP, at der først skal hoppes til denne rutine. Det gør vi som følger:

```
-a0100  
0100 JMP 1DB2  
0103 (tryk RETURN)
```

8.3. Lagring af det nye PIP

Nu har vi rettet PIP-programmet. Du bør nu afprøve PIP (bedst med alle mulige options). Skulle det ikke fungere som beskrevet, så er der noget, der er gået galt. Men så har du (Gud være lovet) en sikkerhedskopi af CP/M, som du først kan kopiere, og så prøve en gang til. (Du ser, hvor praktisk det er at have en sikkerhedskopi i skabet. Ikke kun på grund af dine egne manipulationer kan du blive nødt til at tag afsked for evigt med en diskette. Forestil dig, at din lille datter eller søn har brugt disketten som skovl i sandkassen!)

Nu skal vi sikre programmet på disketten, fordi det kun befinder sig i rettet tilstand i arbejdslageret. Dertil må du først forlade DDT, idet du trykker på Ctrl-C tasten.

Derefter bliver programmet lagret på disketten igen ved hjælp af SAVE-kommandoen. Det sker med den følgende kommando:

```
SAVE 29 PIP.COM
```

Hvis disketten ikke er skrivebeskyttet, så er det overstået, og PIP er rettet.

Du kan se, at selv så store og betydningsfulde programmer som driftssystemet CP/M fra store firmaer som Digital Research, kan opvise fejl. Det er umuligt at eliminere alle fejl i et program - og der findes ingen fejlfri programmer.

På dette sted ville vi egentlig have omtalt nogle flere fejl i den gængse CP/M 2.2-version samt løsningsforslag; men se: alle disse fejl er allerede blevet rettet i Amstrad-CP/M -versionen. Som du ser, anstrenger firmaerne sig for straks at rette alle fejl, som de er blevet bekendt med.

8.a. "Additional Utilities" (Commodore)

Bliv ikke forskrækket, men ved købet af din Commodore 128 har firmaet Commodore forholdt dig noget. Det er nogle nyttige hjælpeprogrammer (Utilities), som du kan efterbestille hos Commodore, mod ekstrabetaling. Tillige modtager du så en beskrivelse af CP/M, som jo kommer helt til kort i brugsanvisningen.

Om købet af denne programpakke lønner sig, det kan du til slut uddrage af dette kapitel, da vi vil gå ind på de forskellige utility-programmer. For en "normal" bruger, sådan kan man sige det kort og tydeligt, lønner anskaffelsen sig med til visshed grænsende sandsynlighed ikke, thi en "brugsanvisning" til CP/M har du jo nu (i hænderne); de programmer, der er på "Additional Utility"-disk'en er assembler, disassembler, monitor og andre hjælpeprogrammer, der er beregnet på den engagerede CP/M-programmør. Man kunne også udmærket kalde denne programpakke for en udviklingspakke.

For det første altså den helt "normale" anvendelse af de programmer, der leveres i "Additional Utility"-pakken. Man kunne derfor også betegne dette kapitel som:

CP/M for videregående

Senere skal vi se nærmere på nogle af de specielle egenskaber ved pakken.

8.1a. Assemblerne MAC og RMAC

MAC og RMAC er begge assemblere, og er efterfølgere af Digital Researchs assembler ASM. MAC adskiller sig gennemgående fra ASM ved, at MAC er en makroassembler. Assembler-venner vil sikkert vide at sætte pris på dette. RMAC er en specialudgave af MAC; RMAC frembringer en relokerbar kode. Dette halsbrækkende ord betyder på dansk, at den frembringer en frit forskydelig kode, som kræver efterbehandling med programmet LINK. Med RMAC oprettes de dubiøse REL-filer. Mere derom senere. Alle tre assembler udelukkende 8080-mnemokode. Ganske vist har Commodore her tilsyneladende fået en ide. Man har oprettet to LIB-filer, der indeholder en mængde makro'er. Den ene LIB-fil hedder Z80.LIB, den anden har det forjættende navn X6503.LIB. Dine forventninger bliver delvis tilfredsstillt. Ved hjælp af disse ekstra LIB-filer kan du også assemble Z-80-mnemokoder med MAC og RMAC, De fleste mnemokoder ser snarere ud som 8080-mnemokoder. Et LD HL,0 findes ikke, det hedder i 8080-stil LXI H <nn>, der er altså "kun" tilføjet de ekstra Z-80-koder, som f.eks. de relative hop, registre IX eller XY eller udvekslingsregistre. Hvordan disse mnemokoder skal se ud, kan du hurtigt lokke ud af filen Z80.LIB ved et:

TYPE Z80.LIB

Det samme gælder for 6502-programmører. Her har man - med nogle få MAC-specifikke særheder - udtænkt en makrofil, der skal lette programmeringen i 6502. Man burde belønne den gode ide.

Der findes altså to assemblere. Vi vil først beskæftige os med MAC-assembleren.

9. Alle CP/M-kommandoer

9.1 ASM (CP/M 2.2) (kun Amstrad)

ASM.COM

Opretter ud fra en kildefil med 8080- eller Z 80-kommandoer en målfil i HEX-format.

Indtastningsformat
ASM FIL
ASM FIL.BBZ

Beskrivelse

Med dette program bliver der oprettet en hexfil ud fra en fil med 8080- eller Z 80-kommandoer. Denne såkaldte objektfil kan med programmet LOAD indlæses i systemet, som midlertidig kommando.

Anvendelse

Programmet ASM søger altid en fil med udvidelsen ASM, så udvidelsen ikke må indtastes. I alt kan tre parametre indtastes med, som hedder: SHP. S står for Source, hvilket er betegnelsen for station (A..P), hvorpå kildefilen står. H står for HEX og er "pladsholder" for betegnelsen for den station, hvorpå hexfilen skal oprettes. Hvis du vil undertrykke oprettelsen af hexfilen, indtaster du et Z. P står for PRN, og du kan her angive den station, hvorpå PRN-filen skal oprettes. For at undertrykke denne, indtaster du igen et Z eller, hvis du vil se PRN-filen på skærmen, et X.

9.2. COPYSYS

COPYSYS.COM

Kopierer systemsporene og CP/M3.SYS over på en diskette

Indtastningsformat:
COPYSYS (CP/M 3.0)

Beskrivelse:

For at kunne starte CP/M 3.0 fra en diskette, må begge dele af CP/M være til stede. Den første del på systemsporene, den anden som filen CPM3.SYS resp. CPM+.SYS. Disketter, der ikke anvendes til at boote, behøver ingen af delene.

På Commodore 128 er den oprindelige udformning af copysys-kommandoen let ændret. Du kan overføre systemet ved en simpel kopiering med PIP. Hvis du alligevel kommer til at kalde kommandoen COPYSYS, så får du besked om, at kommandoen praktisk talt ikke eksisterer længere.

Anvendelse:

Indtast COPYSYS og så ENTER. Du bliver spurgt om, fra hvilken station der skal kopieres:

Source drive name (or return for default)

Angiv bogstavet for den station, hvori din diskette med CP/M-systemet ligger (normalt station A eller 1). Hvis disketten allerede er i stationen, taster du kun ENTER. Så skriver du målstationens betegnelse og trykker igen ENTER. Hvis du endnu ikke har lagt disketten ind, bliver du opfordret til at gøre det. Er der skrevet på systemsporene, kan du også kopiere CPM3.SYS. Du bliver spurgt:

Do you wish to copy CPM3.SYS?

Svar med et "Y", og CP/M-filen bliver kopieret.

Bemærkning:

COPYSYS er standard-kommandoen til kopiering af systemsporene i CP/M 3.0. Desværre er COM-filen ikke på nogen af de medleverede CP/M-disketter. Til gengæld giver programpakken DISCKIT3 mulighed for at formattere disketter formater og så kopiere systemet.

DISCKIT3 er en meget brugervenligt programpakke, som firmaet Amstrad har udviklet specielt til CPC 6128. Desværre forelå der ved trykningen af denne bog ikke noget materiale om programpakken. Sikkert er, at du med DISCKIT3 kan formattere disketter og kopiere systemets CP/M 3.0. Videre oplysninger om DISCKIT3 får du bedst fra brugsanvisningen.

9.3. DATE (CP/M 3.0)

DATE.COM

Viser dato og tidspunkt og muliggør indtastning deraf.

Indtastningsformat
DATE
DATE CONTINUOUS
DATE SET
DATE MM/DD/YY HH:MM:SS

Beskrivelse:

Med programmet kan indtastes en dato og et klokkeslet i dit system, eller kalde begge disse informationer frem.

Anvendelse:

Hvis du kun indtaster kommandoen DATE, så får du vist dato og klokkeslet.

```
Tue 05/06/86 23:49:17
```

Det er en fast udlæsning. Hvis du vil sammenligne klokkeslettet med, hvad et andet ur viser, indtaster du kommandoen DATE med option'en "C". Så ser du det løbende klokkeslet, men kan herunder ikke lade et andet program køre eller indtaste kommandoer. Tryk på en tilfældig tast for at afbryde programmet.

Du kan indtaste dato og klokkeslet på to måder. Med option'en SET, så beder programmet om værdierne efter hinanden, og kan springe over en indtastning med ENTER. Den anden mulighed ser sådan ud:

```
DATE 05/06/85 23:49:17
```

Indtast et klokkeslet, der ligger ca. eet til to minutter efter den øjeblikkelige tid. Når det tidspunkt, du har angivet, kommer, trykkes på mellemrumstangenten, og uret går.

Hvis din computer ikke har noget batteridrevet ur, og det gælder for Amstrad CPC 6128 og Commodore 128, så må man foretage indtastningen hver eneste gang efter starten. Du kan gøre det lettere for dig selv ved at skrive kommandoen DATE.SET i filen PROFILE.SUB

Commodore: ved at indtaste DATEC kan du undgå at indtaste DATE-kommandoen med C-option'en. Interessant er det også, at kildefilen til DATE-programmet er på systemdisketten under navnet DATE.ASM. Et kig på den er meget lærerigt

9.4. DDT (kun Amstrad)

DDT.COM

Debug-program til indlæsning, udførelse, ændring og test af programmer.

Indtastningsformat
DDT
DDT fil.COM

Beskrivelse:

Programmet DDT erstatter CCP'en og læser den angivne fil ind i arbejdslageret. Med DDT kan der spørges efter den sidste adresse i et program (NEXT) og programtællerens indhold (PC).

Anvendelse:

Programmet skal være til rådighed som DDT.COM på disketten. Indtastes der kun DDT, venter programmet stadig på indtastningen af et filnavn. Med indtastningen af Dn bliver et lagerområde, der begynder med adressen n, vist på skærmen. Til afslutning af programmet benytter man bedst kommandoen GO.

9.5. DEVICE (CP/M 3.0)

DEVICE.COM

Viser og ændrer adgangsvejene til ydre apparatur.

Indtastningsformat:

DEVICE NAMES

DEVICE VALUES

DEVICE LST:=XXXXXX[NOXON,1200]

DEVICE CONOUT:=XXX,XXX

DEVICE CON:[PAGES]

DEVICE CON:[COLUMNS=nn, LINES=mm]

Beskrivelse:

Programmet bruges til at vise driftsystemets ind- og udlæsningskanaler, og ændre dem. De tre logiske kanaler hedder CON:, LST: og AUX:.. Man kan forbinde en logisk kanal til flere stykker ydre apparatur. For eksempel kan data fra computeren samtidig sendes både til printeren og til skærmen. Med DEVICE kan du også bestemme antallet af linier og spalter på skærmen.

Anvendelse:

Navnene på udlæsningsenhederne bliver fastlagt af fabrikanterne. Hvis du indtaster DEVICE med option'en NAMES, fås en linie med navne og dataoverførelses hastigheder for de enkelte stykker apparatur. Hvis du skriver DEVICE VALUES, får du følgende meddelelse:

Current Assignments:

CONIN: = CRT

CONOUT: = CRT

AUXIN: = LPT

AUXOUT: = LPT

LST: = CEN

CONIN: og CONOUT: betyder CONsole INput, tastaturinput, og CONsole OUTput, tastaturoutput. Hvis betegnelsen CON: står alene, heviser den til begge dele. LST: betyder LIST DEVICE og forbinder til printeren.

Med PAGE-option'en kan du fastlægge, hvor mange linier og spalter, der skal vise på skærmen. Hvis du indtaster en bestemt baudhastighed i formatet [nnn], bliver dataudvekslingen med det tilsvarende apparat afviklet med denne hastighed. Som svar på kommandoen DEVICE får du på en gang vist DEVICE NAMES og VALUES.

Bemærk, at kommandoen DEVICE i CP/M 2.0 bliver erstattet af kommandoen STAT DEV: (mere korrekt kunne man sige, at kommandoen STAT er blevet delt i CP/M 2.2, fordi den er for indviklet).

9.6. DIR (CP/M 2.2 OG CP/M 3.0)

DIR.COM (midlertidig på diskette) og indbygget kommando (CP/M 3.0)

Viser bestemte filnavne og indholdsfortegnelsen.

Indtastningsformat:
Indbygget kommando
DIR
DIR A:
DIR FIL.XXX
DIR AB*.*

Midlertidig kommando:
DIR[OPTION]
DIR FIL.XXX[OPTION]
DIR AB*.*[OPTION]

Beskrivelse:

DIR er et meget hjælpsomt program til at finde filer på en diskette og endnu mere på en hard-disk. (**Desværre er der endnu ingen, der tilbyder en hard-disk til Amstrad-computere**). Den indbyggede version arbejder fra enhver station og fra ethvert brugerområde. Jokertegnene "*" og "?" kan benyttes. Hvis der ikke angives nogen option, bliver alle ikke-system filer i området vist.

DIR som midlertidig kommando kan betydeligt mere: vise filer med dato-stempel, sortere i alfabetisk orden osv. En opregning af de forskellige options, som skal indtastes i firkantede klammer, følger nu. Hvis der også står systemfiler i indholdsfortegnelsen, fremkommer meldingen:

```
SYSTEM FILE(S) EXIST
```


OPTIONS:	
option funktion	
ATT	udlæser brugerdefinerede fil-attributter
DATE	viser filer med klokkeslet og dato
DIR	viser ikke-system filer
DRIVE=ALL	viser filerne på alle stationer
DRIVE=A	viser filer på station A:
DRIVE=(A,B)	viser filer på de angivne stationer
EXCLUDE	viser alle filer undtagen den angivne
FF	sidefremføring før udlæsning af indholdsfortegnelsen
FULL	viser filer med fuldstændig beskrivelse
LENGTH=n	o viser skrift efter n linier
MESSAGE	viser stationer og USER-områder fortløbende
NOPAGE	viser indholdsfortegnelsen uden stop
NOSORT	viser filerne usorterede
RO	viser Read Only filer
RW	viser Read/Write filer
SIZE	anfører tillige størrelsen af hver fil
SYS	viser kun system-filer
USER=ALL	viser filer fra alle USER-områder
USER=5	viser filer fra det pågældende USER-område
USER=(3,5)	viser filer fra de pågældende USER-områder

9.7. DIRSYS (CP/M 3.0)

Indbygget kommando

Viser de filer, der er blevet defineret som SYSTEM-filer.

Indtastningsformat:

```
DIRSYS
DIRSYS B:
DIRSYS FIL.XXX
DIRSYS AB?*.*
```

Beskrivelse:

Med DIRSYS vises alle SYSTEM-filer. Det kan man også opnå med DIR og options, men DIRSYS er hurtigere, fordi det er en indbygget kommando. Du kan forkorte kommandoen til:

```
DIRS
```

Desuden får du med DIRSYS også oplysninger om, hvorvidt der foruden systemfilerne også eksisterer "normale" filer, CP/M kalder dem ikke-systemfiler. Er der ingen systemfiler, så fremkommer på skærmen meldingen:

No File

IX.8 DUMP (CP/M 2.2 OG CP/M 3.0)

DUMP.COM

Viser ikke-tekst filer i hexadecimal og ASCII

Indtastningsformat:

DUMP

DUMP FIL.XXX

Beskrivelse:

DUMP er særligt interessant for programmører, der arbejder med assembler. Tekst-filer sendes til skærmen eller printeren med TYPE, filer af typen COM, REL eller OVR med DUMP. Herunder bliver hexadecimalværdierne, og så vidt muligt også ASCII-værdierne gengivet på skærmen. Et tegn, der ikke kan gengives, bliver vist med et punktum (.). Hvis man har aktiveret printeren med <CTRL>-P, så bliver udlæsningen selvfølgelig omdirigeret til printeren.

I kapitel VII findes detaljerede beskrivelser af COM-filerne ASM, MAC, DDT, SID og DUMP.

9.8. ED (kun Commodore)

ED.COM

Editor integreret i CP/M

Indtastningsformat:

ED <filnavn>(<filnavnsudvidelse>)

Beskrivelse:

Den indbyggede editor ED egner sig til korte tekster. Men den har et dårligt rygte, og det ikke uden grund. En kompliceret betjening (selv for professionelle) gør sit dertil. Jeg kender knapt nok nogen, der ville drømme om frivilligt at bruge ED.

Ved indtastning af et filnavn undersøger ED, om filnavnet allerede eksisterer. Hvis det er tilfældet, så bliver filindholdet overført til ED-buffere, og der bliver oprettet en backup-fil. Hvis filen ikke eksisterer, så fremkommer en melding:

NEW FILE

på skærmen. Du forlader editoren inklusive lagring, ved at indtaste kommandoen "E".

9.9. ERA(SE)

Indbygget kommando og ERASE.COM

Sletter een, flere eller alle filer fra disketten.

Indtastningsformat:

Indbygget kommando:

ERASE

ERASE FIL.XXX

ERASE AB?*. *

Midlertidig kommando:

ERASE FIL.XXX[CONFIRM]

Beskrivelse:

På kommandoen ERASE bliver den angivne fil slettet, hvis den ikke er "Read Only" eller skrivebeskyttet. Slette kan man selvfølgelig kun indenfor USER-området. Hvis du bruger "" eller "?" gentager kommandoen sletteanvisningen og spørger, om der virkelig skal slettes.

Anvendelse:

Indtastet uden options kan ERASE arbejde fra enhver station. Ved indtastning med options, er det nødvendigt at have programmet ERASE.COM på stationen. ERASE kan forkortes til ERA. Option'en "CONFIRM" kan forkortes til et "C".

9.10. FORMAT (CP/M 2.2 og CP/M 3.0)

FORMAT.COM

Nyformatterer en diskette.

Indtastningsformat

FORMAT

Beskrivelse:

Enhver diskette, som du anvender i din computer, må være tilsvarende formatteret. Programmet ødelægger ved formateringen alle data på en diskette. FORMAT er ikke noget standard CP/M-program. Derfor kan det være, at det ikke er på din systemdiskette. Søg i så tilfælde efter programmet COPY.COM og prøv, om du kan formattere en diskette med det.

Anvendelse:

FORMAT arbejder, lige indtil du afbryder med Control C (↑C) eller svarer nej på spørgsmålet "Format another disk?". Sådan kan du formattere flere disketter efter hinanden.

Bemærkning:

Med Amstrad CPC 6128 leveres der ikke noget FORMAT i CP/M 3.0. FORMAT er integreret som et underprogram i programpakken DISCKIT3. Vil du imidlertid alligevel have FORMAT som en COM-fil, f.eks, for at indbygge det i SUBMIT-filer, så kan du kopiere dig en COM-fil fra CP/M 2.2-diskette.

På Commodore 128 er forskellige disketteformater mulige. Tilsvarende bliver du ved formattering også spurgt, om du vil formatere enkeltsidigt eller dobbeltsidigt osv. Med FORMAT-kommandoen kan man kun formatere disketter i A-stationen.

6128 leveres der ikke noget FORMAT i CP/M 3.0. FORMAT er integreret som et underprogram TIL-VENSTRE-TAST.

Efter du har overstået fingerøvelsen uden brækkede fingre, fremkommer i statuslinjen et lille vindue med lys baggrund. Du bliver nu opfordret til at trykke på den tast, hvis ASCII-kode, du vil vide og eventuelt ændre. Kombinationerne med <CTRL> og SHIFT-tasten er også tilladt her, ellers kan du ikke nå de andre tre tastværdier. Lad os trykke på mellemrumstangenten.

9.11. GENCOM (CP/M 3.0)

GENCOM.COM

Opretter en speciel version af CP/M 3.0

Indtastningsformat:
GENCOM

Beskrivelse:

Bliver anvendt af programmører for at tilpasse en CP/M-version eller indflette ekstra programdele i CP/M.

9.12. GET (CP/M 3.0)

GET.COM

Henter data fra en fil i stedet for fra tastaturet

Indtastningsformat:
GET CONSOLE INPUT FROM FILE FIL.XXX
GET CONSOLE INPUT FROM CONSOLE
GET CONSOLE INPUT FROM FILE FIL.XXX[SYSTEM]
GET FILE FIL.XXX[NOECHO]

Beskrivelse:

Med GET-kommandoen kan du lade indlæsninger, eller kommandoer, komme fra en hvilken som helst fil i stedet for fra tastaturet.

Anvendelse:

Den første linie ovenfor beordrer CP/M til at udføre den næste kommando eller det næste program, der bliver indtastet via tastaturet. Så snart et input bliver nødvendigt, henter CP/M dette fra filen FIL.XXX. Hvis programmet er slut, eller der ikke er flere kommandoer i filen, vender CP/M tilbage til tastaturinput.

Kommandoen i den anden linie instruerer CP/M om igen at tage imod kommandoer fra tastaturet. Denne kommando kan stå i filen FIL.XXX eller indtastes af brugeren.

Den tredje udformning af kommandoen dirigerer straks tastaturindlæsningen til den nævnte fil, for at denne kan læse og udføre tastaturkommandoerne. Som option kan du anføre (NOECHO) for at undgå, at de udførte kommandoer vises på skærmen.

Kommandoen ligner SUBMIT-kommandoen, hvormed kommandoer kan bestilles fra en fil. For øvede programmører åbner sig en bunke muligheder. Commodore har med en GET-kommando sat en flod af kommandoer i gang på en update-diskette, der retter en fejl i en gammel CCP.COM-fil

9.13. HELP (CP/M 3.0)

HELP.COM og HELP.HLP

Giver hjælpeindstillingen til de enkelte CP/M-kommandoer, med eksempler

Indtastningsformat:

HELP

HELP stikord

HELP stikord underbegreb

HELP stikord[option]

HELP.underbegreb

HELP [option]

Beskrivelse:

Med programmet HELP.COM får du hjælp og oplysninger om CP/M-kommandoerne og -programmerne. Du kan ikke kalde HELP, når du er i gang med et andet program.

Anvendelse:

Når du kalder HELP, får du tilbudt en række stikord. Du kan opsøge det anvendelige, og indtaste det. Herunder kan du forkorte stikordet til to tegn. Er oplysningerne

mere end 23 linier, standser visningen, så snart skærmen er fuld, og du kan få den næste skærmfuld ved at trykke på mellemrumstangenten. Hvis du trykker Control P (1P), før du vælger HELP-funktionen, bliver oplysningerne skrevet ud på printeren.

Hvis du vil ændre på teksten i filen HELP.HLP, skal du indtaste HELP med option'en EXTRACT, så kan du indføje din egen tekst og oprette den nye fil, der kan læses af CP/M, med option'en CREATE.

Kommandoen HELP er til særlig hjælp for begyndere, fordi man ikke altid behøver at have CP/M-brugsanvisningen parat. Også i andre kendte programpakker er eksempelet fra Digital Research blevet efterlignet - hvor mange programmer er der ikke, der har fået indbygget en såkaldt HELP-SCREEN, som skal lette arbejdet.

Særligt for at lære CP/M 3.0+ at kende, bør du beskæftige dig indgående med HELP-teksterne. Du lærer at bruge CP/M 3.0+ og lærer tillige de fleste af kommandoerne, at kende.

Til særlig hjælp er HELP-mulighederne også ved systemer, som man sidder ved for første gang. Ikke sjældent er de enkelte kommandoer lidt eller meget forskellige, fordi de er systemafhængige. Her er det anbefalelsesværdigt at skaffe sig oplysninger via HELP-teksterne, hvad der som regel går meget hurtigere end at rode brugsanvisningen frem fra et eller andet sted.

9.14. HEXCOM (CP/M 3.0) (Commodore: Additional Utilities)

HEXCOM.COM

Opretter en fil, der kan udføres direkte, med filnavnudvidelsen COM

Indtastningsformat:
HEXCOM <filnavn>
HEXCOM

Beskrivelse:

Programmet har du brug for i assemblersprog. HEXCOM forvandler filer, der er oprettet med kommandoen MAC.COM i INTEL-hex-format til filer med udvidelsen COM, der kan udføres. HEXCOM er altså i praksis efterfølgeren af LOAD (CP/M 2.2). Heller ikke med HEXCOM er det nødvendigt at angive nogen filnavnsudvidelse, HEXCOM sætter automatisk udvidelsen ".HEX" efter <filnavn>.

Commodore: Bemærk imidlertid, at HEXCOM ikke forvandler .HEX-filer til COM-filer, som er blevet oprettet af RMAC, altså den assembler, der fremstiller en relokerbar kode. Til dette formål må du anvende programmet LINK.

9.15. INITDIR (CP/M 3.0)

INITDIR.COM

Forbereder en disketteindholdsfortegnelse på at blive forsynet med klokkeslets- og datostempel.

Indtastningsformat:
INITDIR B:

Beskrivelse:

CP/M 3.0 giver mulighed for at forsyne filerne med et klokkeslets- og datostempel. De tilsvarende informationer bliver opbevaret i en separat del af indholdsfortegnelsen. Derfor må hver diskettes indholdsfortegnelse forberedes på passende måde, idet man kalder programmet INITDIR. Med programmet SET vælger du, hvorledes filerne skal mærkes.

Det er bedst at indtaste INITDIR ved nye disketter, fordi optagelserne bruger plads i indholdsfortegnelsen. Hvis du benytter INITDIR på en beskrevet diskette, bør du tage en sikkerhedskopi. Hvis programmet INITDIR bliver afbrudt, f.eks. af strømafbrydelse, taber du samtlige optagelser.

Det er ikke kun af CP/M's systemdiskette, det lønner sig at tage en sikkerhedskopi.

9.16. KEYFIG (kun Commodore)

KEYFIG.COM

Med KEYFIG kan du frit omdefinere tastaturet. Endvidere kan lægges hele strenge på en enkelt tast.

Indtastningsformat:
KEYFIG

Beskrivelse:

KEYFIG er fuldkommen menuorienteret. Man kan i reglen opnå det samme ved hjælp af <CTRL>-<HSHFT>-tastkombinationerne uden KEYFIG. Deres anvendelse er dog mere kompliceret og uoverskuelig. For den øvede bruger sparer metoden tid. KEYFIG indeholder også HELP-tekster. Med KEYFIG er du herre over tastaturet. Du kan tildele enhver tast vilkårlige ASCII-værdier, enten med <Shift> eller <CTRL>. Endvidere kan du definere skærbilledfarver og tildele tasterne specialfunktioner (som for eksempel at reboote systemet etc.). Som kronen på værket kan man også tildele tasterne hele kæder af tegn, som det er tilfældet ved alle funktionstaster og HELP-tasten. Alt i alt en meget nyttig sag. Ganske vist må du benytte denne kommando med det forbehold, at du ikke ubetinget kan forvente et tilsvarende program på andre CP/M-computere.

9.17. LIB (CP/M 3.0) (Commodore: Additional Utilities)

LIB.COM

Opretter og ændrer et "bibliotek" af underprogrammer

Indtastningsformat:
LIB FIL.XXX[options]

Beskrivelse:

Mange compilere og assemblerne RMAC og MACRO-80 benytter separate underprogrammer, der oftest er forsynet med udvidelserne REL eller IRL. En sammenstilling af de vigtigste underprogrammer til et "bibliotek" kan opnås med LIB.

9.18. LINK (CP/M 3.0) (Commodore: Additional Utilities)

LINK.COM

Opretter en fil, der kan udføres, med underprogrammoduler

Indtastningsformat:
LINK FIL, FIL2, FIL3,..[options]

Beskrivelse:

Med LINK kan filerne sammenfattes til een COM-fil.

9.19. LOAD (CP/M 2.2 og CP/M 3.0) (kun Amstrad)

LOAD.COM

Opretter en COM-fil ud fra en HEX-fil

**Indtastningsformat:
LOAD fil (uden filnavnudvidelse!)**

Beskrivelse:

Assembleren ASM anlægger sin i to gennemkørsler oprettede kode i form af en hexadecimal-fil. Denne hexadecimal-fils udformning er defineret af firmaet INTEL. Programmet LOAD bearbejder nu denne HEX-fil til en form, så der ud

af de i ASCII-kode anførte HEX-data bliver en COM-fil, der kan udføres. Man skal huske, at filtypen ".HEX" hverken skal eller kan angives. Denne information er obligatorisk, da assembleren lagrer HEX-filen under navnet fil.HEX.

Avendelse:

LOAD skal kaldes efter hver assemblingsproces, da man ikke kan meget med et programs hexkode. Den nøjere anvendelse af dette og de andre tilhørende programmer kan du slå efter i kapitel VII.

9.20. MAC (CP/M 3.0) (Commodore: Additional Utilities)

MAC.COM

En macroassembler til programmer, der er skrevet i 8080-mnemokode og (nyhed) til en vis grad også til Z-80- og 6502-mnemokode.

Indtastningsformat:
MAC FIL
MAC FIL \$options

Beskrivelse:

Tre filer bliver oprettet af macroassembleren MAC og tillige et macro-bibliotek. Kildeprogrammet har udvidelsen ASM og biblioteket LIB. Den assemblede programkode (objektoden) står i filen med udvidelsen HEX, symboltabellen i SYM-filen og den printbare listning i PRN-filen. I stedet for med de sædvanlige parenteser bliver options mærket med dollartegnet (\$)

Hvis du vil have mere dybtgående oplysninger om assembleren MAC, kan du finde dem i kapitel VII.

9.21. MOVCPM (CP/M 2.2) (kun Amstrad)

MOVCPM.COM

Forskyder systemet i 256-bytes enheder

Indtastningsformat:
MOVECPM <faktor>*

Beskrivelse:

Kommandoen MOVCPM kan forskyde (rellokere) systemets CP/M 2.2 i trin på 256 bytes. <Faktor> kan ligge mellem 64 og 179. Ved angivelse af 179 som <faktor>, frembringes der et standard-CP/M. Tilsvarende bliver ved MOVCPM 178*

oprettet et CP/M, der er forskudt 256 pladser nedad. 'Faktor'en angiver den lagerplads, der er til rådighed for CP/M-kommandoer (midlertidige) og brugerprogrammer i enheder af 256 bytes.

Anvendelse:

Bliver det nødvendigt, at forskubbe systemet, f.eks. fordi BIOS/BDOS overlapper med anvendelsesprogrammet, så kan man opnå dette ved hjælp af MOVCPM. Det forskudte system kan lagres ved hjælp af SYSGEN-kommandoen (se 9.33).

9.22. PATCH (CP/M 3.0)

PATCH.COM

Installerer ændringer i CP/M 3.0 eller programmer

Indtastningsformat:
PATCH FIL
PATCH FIL n

Beskrivelse:

Med programmet PATCH kan man ændre i CP/M-systemet og foretage indføjelser i CP/M eller et af de midlertidige programmer. Denne proces kaldes at patche. Med PATCH bliver ændringerne automatisk indføjet i programmet. Når der er flere patch'ninger henviser man med referencenumre til den tilsvarende patch. Referencenumrene starter med 1.

9.23. PIP (CP/M 2.2 og CP/M 3.0)

PIP.COM

Kopiere filer og transportere filer mellem ydre apparaturer

Indtastningsformat:
PIP
PIP måfil=kildefil[option]
PIP B:=fil.XXX
PIP a1.txt=TEKST1.TXT,TEKST2.TXT...
PIP AB1?*.*=AB2?*.*[option]

Beskrivelse:

PIP er det mest ydedygtige program i CP/M. Du kan kopiere filer med det og endda fra en diskette til en anden, men også mellem forskellige brugerområder. Kan fremstille en enkelt ud af flere filer. Automatisk sikre dine senest bearbejdede filer, num-

merere linier fortløbende, forvandle alt til store bogstaver og lade den ottende bit sætte til nul. Hvis du har brug for nærmere oplysninger, så slå venligst efter i kapitel VI.

Anvendelse:

PIP kan anvendes med eller uden options. Når PIP er parat til udførelse af kommandoer, viser det sit parategn, asterisken (*). Det opretter målfilen, som en nøje kopi af kildefilen, med mindre der angives bestemte options. De firkantede klammer skal stå direkte, uden mellemrum, bagved filnavnet. I stedet for bogstaverne i filnavne, som du ikke kender, eller hvis du vil kopiere flere filer af en bestemt kategori, kan du anvende "?" for ubekendte tegn i navnet, eller "*" for alle tegn. Her er et par eksempler:

```
Din standardstation er A:  
PIP B:=FIL.XXX[V]
```

Med denne kommando kopierer du filen FIL.XXX fra A: til B: og lader PIP efterprøve de nye datas korrekthed.

```
PIP B:=FIL?.*[V]
```

Sådan kopierer du en række filer med navnet FIL, ligegyldigt hvilken filnavnudvidelse de har, fra A: til B:.

```
PIP B:=*.*[V]
```

Dermed bliver alle filer på en station overspillet på en anden station og kopieringens korrekthed efterprøvet.

```
PIP FILNY.XXX=FILGAML.XXX
```

Med kommandoen kopierer du filen FILGAML.XXX til den nyoprettede fil FILNY.XXX på samme station. Begge filer er herefter samtidig tilstede på disketten.

```
PIP AL.TXT=TEKST1.TXT,TEKST2.TXT,...
```

Således anbringer du flere forskellige filer i een fil, der så indeholder dem allesammen.

```
PIP B:[G]=TEKST.TXT
```

Filen TEKST.TXT bliver kopieret på station B: og i brugerområde "5".

OPTIONS

A Arkiv-funktion. Kopierer kun filer, der er blevet oprettet eller bearbejdet siden sidste arkivering. Funktionerne klokkeslet og dato skal være sat.

- C. Confirm. Forespørger hver fil, om den skal kopieres.
- Dn Slet efter n spalter. PIP sletter alle tegn, der står i filen efter den n'te spalte. Må kun anvendes til tekstfiler.
- E Vis (echo) teksten på skærmen. Indholdet af de kopierede filer bliver vist på skærmen. Må ikke bruges sammen med parameteren "N". Kun anvendelig til tekstfiler.
- F Fjern tegnet for "næste side" (Form Feed). Nogle printere behøver et (↑L) for at kunne gå til næste side. Hvis din printer ikke behøver det, kan du fjerne det pågældende styretegn med "F".
- Gn Hent eller bring en fil fra eller til brugerområde "n". Denne angivelse skal stå direkte efter det første filnavn og er det eneste, der må stå der. Eksemplet:
 PIP B:[G5]=TEKST.TXT[G1]
 kopierer filen TEKST.TXT fra brugerområde "1" til brugerområde "5".
- H Overførelse af hexadecimal data. Optionen bør du altid angive, når du vil overføre hexadecimal filer. PIP efterprøver så om dataene er skrevet i fejlfrit INTEL-format.
- I Ignorer fil-slut-tegnet i HEX-filer. Indtast denne option for alle filer, undtagen den sidste. Sammen med option'en "I" bliver option'en "H" automatisk sat af PIP. Anvend parameteren "H" for den sidste fil.:
 PIP FIL.HEX=PROG1.HEX[I],PROG2.HEX[H]
- K Undertrykker visningen af filnavne under kopieringen.
- L Forvandler alle store bogstaver til små bogstaver. Benyt kun denne option til tekstfiler og anvend desuden parameteren "Z" til WordStar-filer.
- N Numererer linierne i en fil fortløbende. Tallene begynder med 1 i den første linie og bliver for hver linie forhøjet med een. Tallene kan maksimalt være på seks cifre, pladser der ikke anvendes, fremkommer som tomme mellemrum. Et kolon og et mellemrum står bagved cifrene. Må ikke benyttes sammen med option'erne "E" eller "N". Til WordStar-filer skal option'en "Z" tastes med ind.
- N2 Numererer linierne i et BASIC-program.
- O Overførelse af objektfiler. Bruges til at overføre ikke-tekst og ikke-COM filer. Må ikke anvendes til tekstfiler.
- Pn Sætter sidelængden. Forudindstillingen er 60 linier pr. side. Du bør indtaste "F"-option'en med, for at forhåndenværende tegn for sideafslutning (↑L) kan blive slettet. Må kun anvendes til tekstfiler.
- Qxxxx↑Z Afslutning af kopiering efter denne tegnstring. PIP kopierer en fil til og med den indlæste tegnstring. Benyt kommandoen med to kommandolinier, for at tegnstringen ikke skal blive forvandelt til store bogstaver:

PIP
CON:=TEKST.TXT[WEILER]

Hvis du skriver det hele i den første linie efter PIP, fremkommer det indtastede med store bogstaver.

- R Kopiering af SYSTEM-filer. Disse filer bliver ikke vist med DIR og normalt ikke kopieret af PIP. Med "R" kan man ophæve denne begrænsning.
- Sxxx↑Z PIP begynder at kopiere ved denne tegnstreng. Skriv kommandoen på to linier, ellers bliver tegnstrengen forvandlet til store bogstaver. Må kun anvendes til tekstfiler.
- Tn Indstilling af tabulatortrin. Normalt arbejder CP/M med trin på otte tegn. Ved printningen af en fil fungerer det ikke, og trinlængden må defineres."n" angiver antallet af tomme mellemrum for et tabulatortrin. Må kun anvendes til tekstfiler.
- U Forvandler alle små bogstaver til store bogstaver. Må kun anvendes sammen med tekstfiler. Ved WordStar-filer skal option'en "Z" samtidig sættes.
- V Verificer. Denne parameter sørger for, at PIP nøje sammenligner den kopierede fil med den oprindelige.
- W Skriv oven i filer med attributtet Read Only (RO). Disse filer kan normalt kun læses, men ikke ændres eller slettes. Med W"-option'en bliver sådanne filer slettet uden bekræftelse.
- Z Slet paritets-bit'en (8. bit). ASCII tegnsættet bruger kun syv bits. Den ottende bliver anvendt til forskellige opgaver af forskellige programmer. "Z"-option'en er ikke nødvendig ved kopiering efter ASCII-enheder som CON: eller LST:.

9.24. PUT (CP/M 3.0)

PUT.COM

Skriver dataene til printeren eller skærmen i en fil

Indtastningsformat:

PUT CONSOLE OUTPUT TO FILE FIL.XXX[OPTION]

PUT PRINTER OUTPUT TO FILE FIL.XXX[OPTION]

PUT CONSOLE OUTPUT TO CONSOLE

PUT PRINTER OUTPUT TO PRINTER

Beskrivelse:

Normalt sender CP/M de data, der er beregnet på skærmen, til skærmen og dataene, der er beregnet på printeren, til printeren. Med PUT-kommandoen kan dataene tillige skrives i en fil.

Anvendelse:

De to første linier ovenfor meddeler CP/M, at det skal åbne en fil med navnet FIL.XXX og skrive alt, hvad der er beregnet på skærmen eller printeren, derind. Er denne operation afsluttet, vender programmet igen tilbage til den normale CP/M-funktionsmåde. De to sidste linier afbryder PUT-programmet. Dem kan du anvende, hvis du har en SUBMIT-fil, der anvender PUT-kommandoen, og derefter igen skal finde tilbage til den normale funktionsmåde.

Med option'en NO ECHO kan du forhindre, at de overførte data vises på skærmen. Med option'en FILTER kan du forvandle ethvert styretegn til et tegn, der kan trykkes.

Med option'en SYSTEM går ikke kun dataene, men også kommandolinien til filen.

9.25. REN(AME) (CP/M 2.2 og CP/M 3.0)

RENAME indbygget kommando eller RENAME.COM

Ombenævnelse af en fil

```
Indtastningsformat:  
RENAME  
RENAME FIL2.XXX=FIL1.XXX  
RENAME AB?*2=AB?*1
```

Beskrivelse:

Med RENAME ændrer du navnet på en fil - og kun navnet. Indholdet bliver ikke ændret.

Anvendelse:

Når filen ikke er på standard-stationen, kan stationsbetegnelsen indtastes. Hvis du indtaster RENAME uden options spørger programmet efter dem.

Viser det sig under ombenævnelsen, at en bestående fil vil blive overskrevet, så spørger RENAME, om den gamle fil må slettes. Svarer du nej på spørgsmålet, bliver RENAME-kommandoen afbrudt.

9.26. RMAC (CP/M 3.0) (Commodore: Additional Utilities)

RMAC.COM

Macroassembler til programmering i assembler. Commodore: opretter en forskydelig objektkode, der skal efterbearbejdes med LINK.

Indtastningsformat:
RMAC FIL
RMAC FIL \$OPTIONS

Beskrivelse:

Der bliver oprettet tre filer af macroassembleren RMAC tillige med et macro-bibliotek. Kildeprogrammet har udvidelsen ASM og biblioteket LIB. Den assemblede programkode står i filen med udvidelsen REL, symboltabellen i SYM-filen og den listing, der kan printes, i PRN-filen. Med programmet LINK bliver der oprettet et program, der kan udføres, ud fra REL-filen.

Anvendelse:

I stedet for med de sædvanlige klammer, bliver options mærket med dollartegn (\$). Foran dollartegnet skal stå et tomt mellemrum.

9.27. SAVE (CP/M 2.2 og CP/M 3.0)

SAVE.COM

Lagrer indholdet af arbejdslageret i en fil

Indtastningsformat:
SAVE

Beskrivelse:

SAVE-kommandoen opretter en fil med et bestemt lagerområdes indhold. Før lagringen skal et andet program have skrevet noget ind i lageret.

Anvendelse:

Du skal kalde SAVE, FØR du load'er et andet program i arbejdslageret. SAVE anbringer sig automatisk i den øverste ende af CP/M og overgiver så styringen til CP/M igen. Så lader du programmet køre. Så snart det er slut, overtager SAVE igen automatisk kontrollen. Du bliver så spurgt om filnavnet, begyndelses- og slutadressen indenfor arbejdslageret.

9.28. SET (CP/M 3.0)

SET.COM

Sætter fil-attributter, muliggør indtastningen af beskyttelsesord, tildeler navne til disketter og vælger type af klokkeslets- og datostempel.

Indtastningsformat:
SET FIL.XXX[OPTION]
SET AB?*.*[OPTION]
SET B:[OPTION]
SET[OPTION]

Beskrivelse:

SET-programmet bliver indsat til flere formål. Til de vigtigste hører: at sætte arkiv-funktionsmåde og muligheden for at definere filer og hele stationer som Read Only-filer eller -stationer.

For hver diskette kan der indtastes et specielt navn tillige med et beskyttelsesnavn for hver diskette og hver fil. Diskette-navnene kan vises med SHOW.

Som tredje mulighed kan du forsyne filer med et stempel for klokkeslet og dato.

Læs kapitel 5, hvis du behøver nærmere oplysninger.

9.29. SETDEF (CP/M 3.0)

SETDEF.COM

Viser eller definerer søgeruten og kobler skærmmvisningen fra eller til.

Indtastningsformat:
SETDEF
SETDEF B:
SETDEF[option]

Beskrivelse:

Normalt startes et midlertidigt program ved at indtaste navnet og RETURN bagefter. Hvis du vil kalde programmet fra en anden station, angives tillige stationsbetegnelsen.

Hvis du arbejder på station B:, men CP/M-filerne er på station A, kan du bestemme, hvor CP/M skal søge efter filerne. Det kalder man at forudbestemme en søgerute. Dermed kan CP/M forstå, at det først skal søge på station A:, så på station C: og derefter på standard-stationen.

9.30. SHOW (CP/M 3.0)

SHOW.COM

Viser en diskettes options

Indtastningsformat:
SHOW[OPTION]
SHOW B:[OPTION]

Beskrivelse:

Med SHOW kan man lade en diskettes "tekniske data" vise. De oplysninger, du får er: den ubrugte lagerplads på disketten, antallet af ledige pladser i indholdsfortegnelsen, antallet af aktive brugerområder og det valgte brugerområdes nummer. Hertil kommer angivelser af diskettens samlede lagerkapacitet, blokstørrelse, sektorstørrelse og antallet af sektorer pr. spor. Navnet på en diskette kan også vises.

Bemærk ved anvendelse af CP/M 2.2, at også denne kommando bliver realiseret ved hjælp af STAT.

SHOW skal opfattes som et praktisk tillæg til DIR og anvendes på samme måde. Ved hjælp af SHOW kan du få at vide, om enkelte filer, eller endda hele disketten er forsynet med et password.

9.31. SID (CP/M 3.0) (Commodore: Additional Utilities)

SID.COM

Et debugger-program til at load'e, ændre og teste

Indtastningsformat
SID
SID FIL.XXX
SID FIL.XXX TEKST.SYM

Beskrivelse:

Med SID kan du load'e COM- eller HEX-filer i arbejdslageret, betragte dem og ændre dem. Hvis du load'er en SYM-fil, kan programmet køre under nøje kontrol. SID kan også forvandle programmer, der er parat til at køre, til INTEL 8080 mnemokoder.

Desværre for Commodore-brugere befinder program med sig udelukkende på Additional Utility disketten og står således kun til rådighed for købere af denne program-pakke. SID er i øvrigt en videreudvikling af DDT-programmet fra CP/M 2.2.

9.32. SUBMIT (CP/M 2.2 og CP/M 3.0)

SUBMIT.COM

Muliggør indlæsning af kommandoer fra en fil i stedet for fra tastaturet

Indtastningsformat:
SUBMIT
SUBMIT FIL.SUB
SUBMIT FIL FIL1 FIL2 FIL3...

Beskrivelse:

Kommandoer, som ellers indtastes via tastaturet, kan skrives i en SUBMIT-fil og udføres automatisk. Når der ikke står nogen kommando i filen, går kommandoen tilbage til CP/M, og du kan køre videre som normalt.

CP/M søger ved hver ny opstart en fil ved navn PROFILE.SUB og udfører de befalinger, der står der.

Anvendelse:

Du skriver en SUBMIT-fil med dit tekstprogram, idet du hver gang skriver de ønskede kommandoer på een linie. Som udvidelse må filen have SUB, ellers bliver den ikke anerkendt af SUBMIT.

9.33. STAT (CP/M 2.2) (kun Amstrad)

STAT.COM

Angiver lagerbelægningen og de anførte filers skrivebeskyttelses- og systemtilstand på en liste. Endvidere angiver STAT tilstanden af alle aktiverede disktestationer på en liste.

Indtastningsformat:
STAT
STAT FIL.XXX
STAT F??.*
STAT DEV:
STAT DSK:
STAT USR:
STAT FIL.XXX \$SYS

Beskrivelse:

Med STAT kan du lade vise options for dine disketter og de filer, der befinder sig på dem. Kommandoen ligner CP/M-3.0-kommandoen SHOW. Du kan lade den frie lagerplads på disketten vise, længden af bestemte vilkårlige filer i records eller de benyttede brugerområder. Altså indeholder STAT-kommandoen tillige CP/M 3.0-kommandoen DEVICE under eet. Med STAT DSK: kan du vise alle karakteristika for en hvilken som helst diskette vise. Endvidere kan du ved hjælp af STAT-kommandoen sætte filer til \$R/O (Read Only), \$R/W (Read and Write), \$SYS (systemfil) og \$DIR (Directory-File). Den pågældende status bliver også

vist ved hjælp af STAT-kommandoen. STAT indeholder altså tillige CP/M 3.0-kommandoen SET. Som du ser, er STAT meget fleksibel og omfangsrig.

STAT i detaljer

STAT angiver tilstanden af alle for tiden aktive stationer i systemet på en liste som følger:

```
<station>:<beskyttelse>, Space: <n>k
```

STAT **USR**: giver de aktuelle brugernumre såvel som alle brugerområder, der er optaget på disketten.

```
Active User : 5  
Active Files: 0 1 3 5 7
```

STAT **DEV**: svarer til CP/M 3.0-kommandoen **DEVICE** og udlæser en liste over ind- og udlæsningsapparat. En nytildeling er mulig.

```
CON: is CRT:  
RDR: is TTY:  
PUN: is TTY:  
LST: is LPT:
```

STAT <kanal>=<apparat> tildeler en kanal et nyt stykke ind/udlæsningsapparat

```
STAT CON:=CRT:, LST:=TTY:
```

STAT<station>SR/O, henholdsvis STAT<station>SR/W sætter den anførte station på "kun-læse", henholdsvis "skrive/læse" status.

STAT<station> angiver den på den angivne diskette for øjeblikket ledige lagerplads.

```
Bytes Remaining On A: 5k
```

STAT<filnavn.XXX>, eventuelt med jokertegn, angiver lagerforbruget og skrivebeskyttelses- resp. systemtilstanden af den/de angivne filer.

```
Recs Bytes EXT ACC  
144 18k 2 R/W A:(D2.BAS)  
64 2k 1 R/W A:DISC.BAS
```

Hvis en fil er sat i parentes, så er det en systemfil, altså en fil, der er forsynet med \$SYS-option'en. Den fremkommer ikke i den "normale" indholdsfortegnelse.

Ved hjælp af option'en \$S kan du tillige lade størrelsen (Size) af en fil angive, der imidlertid for det meste falder sammen med antallet af records.

STAT<filnavn.XXX>\$<option> sætter filen, eller filerne (hvis der anvendes joker-tegn) til status <option>. Følgende options er mulige

\$R/O - (Read- Only) Sæt skrivebeskyttelse
\$R/W - (Read-Write) Ophæv skrivebeskyttelse
\$SYS - (System) Sæt systemegenskaber
\$DIR - (Directory) Slet systemegenskab

STAT <station>:DSK: lister den i en BIOS-del angivne tabel på følgende måde:

A: Drive Characteristics
1368: 128 Byte Record Capacity
171: Kilobyte Drive Capacity
64: 32 Byte Directory Entries
64: Checked Directory Entries
128: Records/Extent
8: Records/Block
36: Sectors/Track
2: Reserved Tracks

STAT VAL: giver en liste over de med STAT mulige kommandoer. Denne liste ser således ud:

Temp R/O Disk: d:=R/O
Set indicator : d:filename.typ \$R/O \$R/W \$SYS \$DIR
Disk Status : DSK: d:DSK:
User Status :USR:
IObyte Assign :
CON: = TTY: CRT: BAT: UC1
RDR: = TTY: PTR: UR1: UR2:
PUN: = TTY: PTP: UP1: UP2
LST: = TTY: CRT: LPT: UL1:

Anvendelse:

STAT-kommandoen giver mere information om filstørrelse og -fordeling, end kommandoen DIR gør. Endvidere kan man få angivet den øjeblikkelige optagethed af ind/udlæsningsapparat (DEVICES).

Da STAT-kommandoen indeholder CP/M 3.0-kommandoerne SET, DEVICE, SHOW og delvis DIR, kan du se efter betydningen under disse kommandoer.

9.34. SYSGEN (CP/M 2.2) (kun Amstrad)

SYSGEN.COM

Kopierer system-CP/M 2.2 på diskette

Indtastningsformat:
SYSGEN
SYSGEN FIL.XXX
SYSGEN*

Beskrivelse:

SYSGEN skriver resultatet af en MOVCPM-kommando på en diskettes systemspor (se nedenfor). SYSGEN* skriver herunder det umiddelbart forud af MOVCPM oprettede system på disketten. SYSGEN<filnavn> sikrer en af MOVCPM oprettet systemfil på en diskettes systemspor. Endelig spørger SYSGEN uden parametre efter kilde- og måldiskette og kopierer dette system.

Med MOVCPM kan man forskyde CP/M 2.2 i arbejdslageret; mange gange er det nødvendigt. Du kan forskyde CP/M i trin på 256 bytes. Positionen i lageret bliver angivet af en værdi mellem 64 og 179. Syntaksen lyder således:

MOVCPM <størrelse>* som eksempel: MOVCPM 178*

MOVCPM 178* frembringer et CP/M, der i forhold til standard-CP/M er forskudt 256 bytes "nedad". Du kan sikre CP/M på en diskette eller ved hjælp af SYSGEN lagre det direkte som system.

9.35. TYPE (CP/M 2.2 og CP/M 3.0) (kun Amstrad)

TYPE indbygget og TYPE.COM

Viser en tekstfil på skærmen

Indtastningsformat:
TYPE
TYPE FIL.XXX
TYPE AB?.*
TYPE FIL.XXX[NO PAGE]
TYPE AB?.*[NO PAGE]

Beskrivelse:

TYPE-kommandoen lister een eller flere filer på skærmen. Så snart skærmen er fuld, stopper visningen og sættes igen igang med mellemrumstangenten eller RETURN. Hvis du tillige angiver option'en "NO PAGE", standser visningen ikke.

Anvendelse:

Den indbyggede version af TYPE kan du kalde fra station og brugerområde. Hvis du vil afbryde programmet, indtaster du Control C (↑C). Man får en udprintning af programmet, hvis du tillige før RETURN indtaster kommandoen Control P (↑P).

9.36. USER (CP/M 2.2 og CP/M 3.0)

USER indbygget kommando

Forandrer det aktuelle brugerområde

Indtastningsformat:

USER

USER n

Beskrivelse:

Enhver diskette kan underafdeles i 16 forskellige brugerområder. På den måde har man altid et brugerområde, hvor kun de nødvendige filer og programmer står. SYSTEM-filer på brugerflade nul kan kaldes fra alle brugerområder.

Anvendelse:

Det aktuelle brugerområde bliver angivet af CP/M stikordet (1B) og kan ændres ved indtastning af USER. Hvis du ikke angiver det nye brugerområde, spørger programmet selv efter det. Med programmet SHOW kan du lade de aktive brugerområder vise.

9.37. XREF (CP/M 3.0) (Commodore: Additional Utilities)

XREF.COM

Opretter en referenceliste til assemblerprogrammer

Indtastningsformat:

XREF FIL

XREF FIL \$P

Beskrivelse:

Assemblerne MAC og RMAC opretter en alfabetiske liste over alle symboler og deres værdier i et program. XREF forhøjer bekvemmeligheden, ved at det for hvert symbol også angiver, i hvilken programlinie, det kan findes. *Programmet XREF kræver tilstedeværelsen af filerne SYM og PRN. XREF opretter under filnavnet <filnavn>.XREF en assemblerliste (i lighed med <filnavn>.PRN) og tilføjer en udførlig listing af alle forekommende symboler, inklusive angivelse af deres forekomst i programmet under anførelse af linienumre. Man kalder sådan noget for en Cross Reference List. Amerikanerne forkorter gerne cross til bogstavet X, derfor hedder kommandoen XREF.*

10. Forskelle mellem Amstrad CPC-computerne

Firmaet Amstrad, der arbejder sammen med firmaet Locomotive Software, har i den sidste tid sørget for det meste af spændingen på det danske marked. Når alt kommer til alt, har dette firma præsteret i løbet af mindre end eet år at sende 3 (!) computere på markedet, der skulle være kompatible, men alligevel kan opvise nogle vigtige ændringer.

Den først computer i serien er Amstrad CPC 464. Den har en indbygget kasset-tebåndoptager og er meget yndet på grund af dens hurtige og bekvemme BASIC, såvel som dens lave pris. Det varede ikke ret længe (det var under Hannover-Messen i 1985), så præsenterede firmaet Amstrad en ny maskine: CPC 664.

Den vigtigste forskel fra CPC 464 er den indbyggede disktestation. Begge maskiner råder over 64k bytes RAM-lager, hvoraf ganske vist en del går tabt til skærbilledet (16k bytes) og systemvariabler. Brugeren har via BASIC fri rådighed over 42249 bytes. CPC 664 har et let ændret BASIC (V 1.1), der har rettet nogle fejl i CPC-464 BASIC, og tillige har nogle nye kommandoer. Til begge computere, CPC 464 og CPC 664 kræves CP/M 2.2. CP/M 2.2 er den gængse CP/M-version til 64k bytes computere.

Men ikke kun det indre liv har ændret sig, også den ydre fremtoning har gennemgået en forandring: CPC 664 er noget fladere og råder over et andet tastatur med fremhævet cursor-blok. CPC'en er blevet mere professionel. Ganske vist har softwarefirmaerne deres problemer med de to "brødre", for desværre rækker kompatibiliteten ikke så langt. Det er jo klart: når man eliminerer fejl i BASIC'en og tilføjer nye kommandoer, så kan de to computere jo ikke blive kompatible.

Meget hårdere er ganske vist maskinsprogsprogrammørerne ramt, for her er der mest, der har ændret sig. Men heldigvis findes jo CP/M. Her er der slet ikke sket noget for brugeren. Du kan se, hvor smart det er at råde over en standard.

Da de konkurrerende firmaer har bragt computere på markedet, som råder over 128k eller endnu flere bytes, så firmaet Amstrad sig tvunget til også at fremkomme med en 128k bytes computer. Og se, Amstrad overraskede med Amstrad CPC 6128

Også CPC 6128 råder over en indbygget disktestation, det er vel derfor, at det første tal i typebetegnelsen er "6". CPC 6128 ligner også CPC 664 mere end CPC 464. 128k bytes computeren er endnu fladere, man har næsten lyst til at sige superflad. Den har et næsten endnu mere "præprofessionelt" tastatur end CPC 664. Frem for alt cursor-tasterne måtte igen holde for; man har nu placeret dem et helt andet sted. Ganske vist kræver tastaturet en smule tilvæending: COPY-tasten er ikke mere midt i cursor-blokken, og Shift-tasterne befinder sig

ikke i den underste række, men er anbragt ovenover tasterne "CONTROL" og "RETURN". Her kommer man i hvert fald i begyndelsen til at slå forkert en gang imellem.

BASIC'en har ikke forandret sig i CPC 6128-versionen, der er stadig indbygget version 1.1. AMSDOS. Det indbyggede driftssystem for disketterne, har glædeligvis ikke ændret sig i nogen af de tre modeller. I øvrigt er tre-tommers stationerne meget driftssikre og tilforladelige diskettestationer, det er der mange andre stationer, der ikke er.

Endvidere har portene bagpå computeren forandret sig lidt. Diskettestationstilslutningen og printertilslutningen (Centronics) er kvalitativt forbedret. Yderligere råder CPC 6128 over endnu en port med betegnelsen Extension.

Ved købet af DD1 resp. CPC 6128 får du en systemdiskette, hvorpå der befinder sig CP/M 2.2 såvel som programmeringsproget LOGO. Med 6128'eren leveres der endda to disketter, da man ikke kunne anbringe alle filerne i CP/M 3.0 på een diskette. Først og fremmest HELP-teksterne, altså de tekster, som du kan kalde ved hjælp af CP/M 3.0-kommandoen HELP, tager meget plads op. Endvidere får CPC 6128-køberen også en diskette med CP/M 2.2

Den mest betydende forskel mellem den nye CPC og dens forgængere er den lagerplads, der er til rådighed. Den er fordoblet her. Ganske vist har BASIC-programmøren mindre ud af det, idet han stadigvæk kun har 42249 bytes til sin rådighed. Uden tricks er der ikke noget at gøre ved det. Dog, de 128k bytes lagerplads giver computeren mulighed for at gøre springet fra CP/M 2.2- til CP/M 3.0+-computer. CP/M 3.0 er den gængse version for 128k bytes computere. Tilføjes "+" står her for nogle ændringer og tillæg til CP/M 3.0

Med begge CP/M-versioner kan man egentlig arbejde særdeles behageligt. Naturligvis er CP/M 3.0+ noget mere bekvemt, men det ligger i sagens natur, man kører også noget mere bekvemt i den nye Golf end i den gamle "bille". Det er det, man har fremskridt til, og firmaet Digital Research plejer fortsat deres top-program CP/M.

Da denne bog skal tjene som træningsbog for alle CPC-computere, er både CP/M 2.2- og CP/M 3.0-kommandoerne beskrevet. Næsten alle CP/M 2.2-kommandoer lader sig også anvende i CP/M-3.0. Kommandoer, som du kun kan anvende i CP/M 3.0, er mærket i overensstemmelse hermed.

Appendix 1

Omregningstabel

Dezi	Hex	Binär	Dezi	Hex	Binär
-----	---	-----	----	---	-----
#000	\$00	%00000000	#001	\$01	%00000001
#002	\$02	%00000010	#003	\$03	%00000011
#004	\$04	%00000100	#005	\$05	%00000101
#006	\$06	%00000110	#007	\$07	%00000111
#008	\$08	%00001000	#009	\$09	%00001001
#010	\$0A	%00001010	#011	\$0B	%00001011
#012	\$0C	%00001100	#013	\$0D	%00001101
#014	\$0E	%00001110	#015	\$0F	%00001111
#016	\$10	%00010000	#017	\$11	%00010001
#018	\$12	%00010010	#019	\$13	%00010011
#020	\$14	%00010100	#021	\$15	%00010101
#022	\$16	%00010110	#023	\$17	%00010111
#024	\$18	%00011000	#025	\$19	%00011001
#026	\$1A	%00011010	#027	\$1B	%00011011
#028	\$1C	%00011100	#029	\$1D	%00011101
#030	\$1E	%00011110	#031	\$1F	%00011111
#032	\$20	%00100000	#033	\$21	%00100001
#034	\$22	%00100010	#035	\$23	%00100011
#036	\$24	%00100100	#037	\$25	%00100101
#038	\$26	%00100110	#039	\$27	%00100111
#040	\$28	%00101000	#041	\$29	%00101001
#042	\$2A	%00101010	#043	\$2B	%00101011
#044	\$2C	%00101100	#045	\$2D	%00101101
#046	\$2E	%00101110	#047	\$2F	%00101111
#048	\$30	%00110000	#049	\$31	%00110001
#050	\$32	%00110010	#051	\$33	%00110011
#052	\$34	%00110100	#053	\$35	%00110101
#054	\$36	%00110110	#055	\$37	%00110111
#056	\$38	%00111000	#057	\$39	%00111001
#058	\$3A	%00111010	#059	\$3B	%00111011
#060	\$3C	%00111100	#061	\$3D	%00111101
#062	\$3E	%00111110	#063	\$3F	%00111111
#064	\$40	%01000000	#065	\$41	%01000001
#066	\$42	%01000010	#067	\$43	%01000011
#068	\$44	%01000100	#069	\$45	%01000101
#070	\$46	%01000110	#071	\$47	%01000111
#072	\$48	%01001000	#073	\$49	%01001001
#074	\$4A	%01001010	#075	\$4B	%01001011
#076	\$4C	%01001100	#077	\$4D	%01001101
#078	\$4E	%01001110	#079	\$4F	%01001111
#080	\$50	%01010000	#081	\$51	%01010001
#082	\$52	%01010010	#083	\$53	%01010011
#084	\$54	%01010100	#085	\$55	%01010101

Dezi	Hex	Binär
----	---	-----
#086	\$56	%01010110
#088	\$58	%01011000
#090	\$5A	%01011010
#092	\$5C	%01011100
#094	\$5E	%01011110
#096	\$60	%01100000
#098	\$62	%01100010
#100	\$64	%01100100
#102	\$66	%01100110
#104	\$68	%01101000
#106	\$6A	%01101010
#108	\$6C	%01101100
#110	\$6E	%01101110
#112	\$70	%01110000
#114	\$72	%01110010
#116	\$74	%01110100
#118	\$76	%01110110
#120	\$78	%01111000
#122	\$7A	%01111010
#124	\$7C	%01111100
#126	\$7E	%01111110
#128	\$80	%10000000
#130	\$82	%10000010
#132	\$84	%10000100
#134	\$86	%10000110
#136	\$88	%10001000
#138	\$8A	%10001010
#140	\$8C	%10001100
#142	\$8E	%10001110
#144	\$90	%10010000
#146	\$92	%10010010
#148	\$94	%10010100
#150	\$96	%10010110
#152	\$98	%10011000
#154	\$9A	%10011010
#156	\$9C	%10011100
#158	\$9E	%10011110
#160	\$A0	%10100000
#162	\$A2	%10100010
#164	\$A4	%10100100
#166	\$A6	%10100110
#168	\$A8	%10101000
#170	\$AA	%10101010
#172	\$AC	%10101100

Dezi	Hex	Binär
----	---	-----
#087	\$57	%01010111
#089	\$59	%01011001
#091	\$5B	%01011011
#093	\$5D	%01011101
#095	\$5F	%01011111
#097	\$61	%01100001
#099	\$63	%01100011
#101	\$65	%01100101
#103	\$67	%01100111
#105	\$69	%01101001
#107	\$6B	%01101011
#109	\$6D	%01101101
#111	\$6F	%01101111
#113	\$71	%01110001
#115	\$73	%01110011
#117	\$75	%01110101
#119	\$77	%01110111
#121	\$79	%01111001
#123	\$7B	%01111011
#125	\$7D	%01111101
#127	\$7F	%01111111
#129	\$81	%10000001
#131	\$83	%10000011
#133	\$85	%10000101
#135	\$87	%10000111
#137	\$89	%10001001
#139	\$8B	%10001011
#141	\$8D	%10001101
#143	\$8F	%10001111
#145	\$91	%10010001
#147	\$93	%10010011
#149	\$95	%10010101
#151	\$97	%10010111
#153	\$99	%10011001
#155	\$9B	%10011011
#157	\$9D	%10011101
#159	\$9F	%10011111
#161	\$A1	%10100001
#163	\$A3	%10100011
#165	\$A5	%10100101
#167	\$A7	%10100111
#169	\$A9	%10101001
#171	\$AB	%10101011
#173	\$AD	%10101101

Dezi	Hex	Binär
----	---	-----
#174	\$AE	%10101110
#176	\$B0	%10110000
#178	\$B2	%10110010
#180	\$B4	%10110100
#182	\$B6	%10110110
#184	\$B8	%10111000
#186	\$BA	%10111010
#188	\$BC	%10111100
#190	\$BE	%10111110
#192	\$C0	%11000000
#194	\$C2	%11000010
#196	\$C4	%11000100
#198	\$C6	%11000110
#200	\$C8	%11001000
#202	\$CA	%11001010
#204	\$CC	%11001100
#206	\$CE	%11001110
#208	\$D0	%11010000
#210	\$D2	%11010010
#212	\$D4	%11010100
#214	\$D6	%11010110
#216	\$D8	%11011000
#218	\$DA	%11011010
#220	\$DC	%11011100
#222	\$DE	%11011110
#224	\$E0	%11100000
#226	\$E2	%11100010
#228	\$E4	%11100100
#230	\$E6	%11100110
#232	\$E8	%11101000
#234	\$EA	%11101010
#236	\$EC	%11101100
#238	\$EE	%11101110
#240	\$F0	%11110000
#242	\$F2	%11110010
#244	\$F4	%11110100
#246	\$F6	%11110110
#248	\$F8	%11111000
#250	\$FA	%11111010
#252	\$FC	%11111100
#254	\$FE	%11111110

Dezi	Hex	Binär
----	---	-----
#175	\$AF	%10101111
#177	\$B1	%10110001
#179	\$B3	%10110011
#181	\$B5	%10110101
#183	\$B7	%10110111
#185	\$B9	%10111001
#187	\$BB	%10111011
#189	\$BD	%10111101
#191	\$BF	%10111111
#193	\$C1	%11000001
#195	\$C3	%11000011
#197	\$C5	%11000101
#199	\$C7	%11000111
#201	\$C9	%11001001
#203	\$CB	%11001011
#205	\$CD	%11001101
#207	\$CF	%11001111
#209	\$D1	%11010001
#211	\$D3	%11010011
#213	\$D5	%11010101
#215	\$D7	%11010111
#217	\$D9	%11011001
#219	\$DB	%11011011
#221	\$DD	%11011101
#223	\$DF	%11011111
#225	\$E1	%11100001
#227	\$E3	%11100011
#229	\$E5	%11100101
#231	\$E7	%11100111
#233	\$E9	%11101001
#235	\$EB	%11101011
#237	\$ED	%11101101
#239	\$EF	%11101111
#241	\$F1	%11110001
#243	\$F3	%11110011
#245	\$F5	%11110101
#247	\$F7	%11110111
#249	\$F9	%11111001
#251	\$FB	%11111011
#253	\$FD	%11111101
#255	\$FF	%11111111

Appendix 2

CP/M-styretegnene

Kommando	Virkning
↑A	Cursor et tegn til venstre*
↑B	Cursor til begyndelsen af linien eller til slutningen, hvis cursoren allerede står ved begyndelsen
↑C	Afbryd CP/M, gennemfør et reset
↑E	Cursor til den næste linie
↑F	Cursor eet tegn til højre*
↑G	Slet tegnet under cursor'en
↑H	Slet tegnet til venstre for cursor'en
↑I	Cursor til den næste TAB-position
↑J	Udførelseskommando (liniesammenkobling)
↑K	Slet fra cursor til linieende
↑M	Som RETURN
↑P	Til-/frakobling af printer
↑Q	Igangsæt scrolling efter ↑S
↑R	Gentag kommandolinie
↑S	Stands udlæsning på skærm
↑U	Slet alle tegn på linien
↑W	Vis gamle kommandolinie igen
↑X	Slet tegn til venstre for cursor
↑Z	Streng-ende i PIP og ED

* betyder: kun til rådighed i CP/M med "bankswitching"

Appendix 3

Alle PIP-parametre

PIP-parameter

- A Arkiv-funktion. Kopierer kun filer, der er blevet oprettet eller bearbejdet siden den sidste arkivering. Funktionen klokkeslets- og datostempel skal være sat.
- C Confirm. CP/M spørger ved hver enkelt fil, om den må kopieres.
- Dn Slet efter n spalter. PIP sletter alle tegn, der står i filen efter n spalter. Må kun anvendes til tekstfiler.
- E Vis (echo) tekst på skærmen. Indholdet af filer, der lige er kopieret bliver vist på skærmen. Må ikke bruges sammen med parameteren "N". Må kun anvendes til tekstfiler.
- F Fjern tegnet for "næste side" (Form Feed). Visse printere kræver et (↑L) for at kunne gå til den næste side. Hvis din printer ikke behøver det, kan du fjerne det pågældende styretegn med "F".
- Gn Hent eller bring en fil fra eller til brugerområde "n". Denne angivelse skal stå umiddelbart efter det første filnavn og er det eneste, der må stå der. Eksemplet:
PIP B:[G5]=TEKST.TXT[G1]
kopierer filen TEKST.TXT fra brugerområde "1" til brugerområde "5".
- H Overførelse af hexadecimal data. Denne parameter bør du altid indtaste, når du vil overføre HEX-filer. PIP efterprøver så, om dataene er skrevet i fejlfrit INTEL-format.
- I Ignorer "slut-på-filen" tegnet i HEX-filer. Indtast denne parameter for enhver fil undtagen den sidste. Sammen med option'en "I" bliver PIP's option "H" automatisk sat. Anvend parameteren "H" til den sidste fil:
PIP FIL.HEX=PROG1.HEX[I],PROG2.HEX[I],PROG3.HEX[H]
- K Undertryk visningen af filnavne under kopiering.
- L Forvandl alle store bogstaver til små bogstaver. Benyt kun denne parameter til tekstfiler og anvend tillige parameteren "Z" til WordStar-filer.
- N Nummerer linierne i en fil fortløbende. Tallene begynder med 1 i den øverste linie og bliver forhøjet med een for hver linie. Tallene kan højst være på seks cifre; pladser der ikke anvendes fremkommer som tomme mellemrum. Der står et kolon og et mellemrum bagved cifrene. Må ikke bruges sammen med parametrene "E" eller "N". Ved WordStar-filer skal parameteren "Z" angives med:
- N2 Nummererer linierne i et BASIC-program.

- O Overførelse af objektfiler. Bruges til at overføre ikke-tekst- og ikke-COM-filer. Må ikke anvendes til tekstfiler.
- Pn Sætter sidelængden. 60 linier pr. side er forudindstillet. Du bør indtaste "F"-parameteren med, for at eventuelt tilstedeværende tegn for sideafslutning (↑L) kan blive slettet. Må kun anvendes til tekstfiler.
- Qxxxx↑Z Slut med kopieringen efter denne tegnstreng. PIP kopierer en fil til og med denne tegnstreng. Benyt kommandoen med to kommandolinier, for at tegnstrengen ikke skal blive forvandlet til store bogstaver:
- PIP
CON:=TEKST.TXT[WEILER]
- Hvis du skriver det hele i den første linie bagved PIP, fremkommer det indtastede med store bogstaver.
- R Kopier SYSTEM-filer. Disse filer bliver ikke vist med DIR og normalt ikke kopieret. Med "R" kan denne begrænsning ophæves.
- Sxxxx↑Z Begynd ved denne tegnstreng. PIP begynder kopieringen ved tegnstrengen. Skriv kommandoen på to linier, ellers bliver tegnstrengen oversat til store bogstaver. Må kun anvendes til tekstfiler.
- Tn Indstilling af tabulatortrin. Normalt arbejder CP/M med trin på otte tegn. Ved printning af en fil fungerer dette dog ikke, og trinstørrelsen må defineres. "n" angiver antallet af tomme tegn for et tabulatortrin. Må kun anvendes til tekstfiler.
- U Forvandlet til store bogstaver. Forvandler alle små bogstaver til store bogstaver. Må kun anvendes til tekstfiler. Ved WordStar-filer skal "Z"-parameteren samtidig sættes.
- V Verificer. Denne parameter sørger for, at PIP nøje sammenligner den kopierede fil med den oprindelige fil.
- W Skriv oveni filer med Read Only (RO) attributtet. Disse filer kan normalt kun læses, men ikke ændres eller slettes. Med "W" parameteren bliver disse filer slettet, uden at der bedes om bekræftelse.
- Z Slet paritets-bit'en (8. bit). ASCII-tegnsettet benytter kun syv bits. Den ottende bit bliver indsat af forskellige programmer til forskellige opgaver. "Z"-parameteren er ikke nødvendig ved kopiering efter ASCII-enheder som CON: eller LST:.

Appendix 4

SET-parametrene

OPTION	BETYDNING
DIR	Gør en SYSTEM-fil synlig i den normale indholdsfortegnelse igen.
SYS	Gør en fil til en SYSTEM-fil.
RO	Bestemmer, at en fil kun skal kunne læses.
RW	Bestemmer, at fil både skal kunne læses og ændres.
ARCHIV=OFF	Sætter ARCHIV-attributtet på "slukket". Det betyder, at den pågældende fil endnu ikke er blevet sikret (arkiveret). Programmet PIP med option'en [A] kan kopiere filer med attributtet ARCHIV=OFF. Du indtaster PIP-kommandoen med en asterisk for filnavnet, og PIP kopierer alle filer, der er blevet ændret siden den sidste kopiering med PIP og ÖA.

EN DATA BECKER BOG

GERITS . SCHIEB . THRUN

COMMODORE

128

INTERN

Simpelthen bibelen til C128

DANSK / NORSK

udgave

Forlag: NCS, box 105, DK 6950 Ringkøbing

EN DATA BECKER BOG

SCHÜSSEL

COMMODORE


64/128

**HARDWARE-
UDVIDELSE**

LAV SELV COMPUTER-TILBEHØR

DANSK **NORSK**

udgave

Forlag:  , box 105, DK-6950 Ringkøbing

TRE GODE BØGER TIL DIN AMSTRAD

PEEKS & POKES

Uundværlig for programmører. Indføring i maskinsprog.

TIPS & TRICKS

En sand guldgrube for brugere af 464/664 og 6128.
Fortsætter hvor Peeks & Pokes slap.

FØRSTEBOGEN

En let fattelig indføring i 664/6128. Alt puttes ind med
teskeer, enhver fejlfortolkning udelukket.

KATALOG

incl. aktualiseret tillæg
med illustrationer og korte indholdsforklaringer på over
100 bøger og programmer fås hos din

FORHANDLER

eller send kr. 5,00 i frimærker og få det direkte fra



NORDIC COMPUTER SOFTWARE
POSTBOX 105 · DK 6950 RINGKØBING
DANMARK



FLOPPY-BOGEN

1541

1570/1571

MED KOMPLET KOMMENTERET ROM-LISTNING

- *Disk-monitor lige til at taste ind.*
- *Block-execute, USER-kommandoer, autostartfiler m.v.*

ØKONOMISYSTEM

til PC Commodore 64-128 og/eller til Amstrad



Fås hos din forhandler eller direkte hos



ISBN 87-7283-012-3

