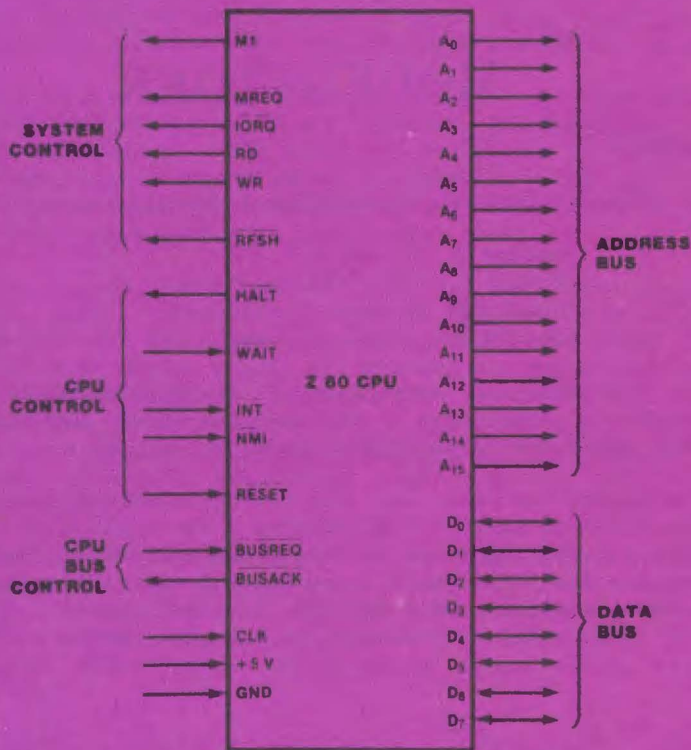




gruppen



1985:1

INDHOLD

Brugermøder i foråret 85	2
Problemer med NEC 765/A og INTEL 8272 FDC	3
8" Double density floppy uden DMA	5
Idiotsikret input i PASCAL	7
Nyheder i biblioteket	13
Oversigt over volume M006, M009, M010, M011 og M012	17
Rettelse til COMAL-80 felteditor	21
Annoncer	22
Adresser, bestilling fra biblioteket	23

MØDEDATOER

BRUGERMØDER

Der er planlagt følgende brugermøder i forårssæsonen 1985:

Onsdag d. 13 februar.

Onsdag d. 13 marts.

Onsdag d. 10 april.

Onsdag d. 8 maj.

Møderne afholdes som sædvanligt i *Medborgerhuset Gimle, Thors-*
havns-gade 21, København S. Dørene åbnes kl 19. Der er endnu ikke
fastlagt emner for de pågældende dage.

På brugermødet den 16/1 var der blandt flere medlemmer interesse
for det modem-projekt, der er startet i det tyske blad MC. Januar-
nummeret (af MC) indeholdt de foreløbige specifikationer samt sel-
ve hardware designet. Man blev enige om foreløbigt at vente på den
tilhørende software. Hvis denne er lige så lovende, er det tanken
evt. at lave et fælles indkøb af bl.a. modem-kreds.

NÆSTE NUMMER

Næste nummer forventes udsendt i løbet af april, så derfor er
deadline indtil videre fastsat til *onsdag den 13 marts*. Indlæg
eller annoncer til bladet sendes til redaktionen. Indlæg modtages
meget gerne på diskette (helst MPS 80 spor, enkeltsidet).

FLOPPY-DISK

PROBLEMER MED NEC-765/A OG INTEL-8272 FLOPPY DISC CONTROLLER

I "BYTE" april 1983, har jeg fundet, og fordansket følgende artikel, der var en fodnote til anmeldelsen af to CP/M-80 kort til IBM PC-datamater.

Ind imellem viser det sig, at diske, der ifølge foreliggende oplysninger burde være "format kompatible", falder ind under MURPHY's LOV og ikke er det. I disse tilfælde kan flere forhold være årsag til at operationerne ikke forløber som forventet.

Een af årsagerne kan henføres til selve FDC-kontrollerkredsen; NEC 765 eller 765A har en intern reset funktion, der hænger nøje sammen med detektering af index-signalet. Prøver man at læse diske, der er formatteret med første sektor liggende for tæt på indexhullet, har kredsen ikke tilstrækkelig tid til at resette intern logik, før end første sektor på sporet skal læses.

Problemet kan løses meget enkelt ved at man, enten tildækker indexhullet, eller indskyder en afbryder i indexlinien fra diskdrevet; herefter kan man efter behov afbryde indexsignalet. Denne "snyde" løsning er mulig fordi NEC-kredsen kan finde soft-sektor formatters fysiske sektorer, ved simpelt hen direkte at læse de enkelte sektorerers header-blocks, dvs. FDC-kredsen kræver ikke index-pulser ved søgning efter de enkelte sektorer. Det vil samtidigt sige, at NEC-765 kun behøver index-signalet, når den skal formattere blanke disketter.

Løsningsmodellen kan eventuelt medføre en komplikation derved, at CP/M systemet nu ikke mere erkender "time-out", fordi denne feature i mange tilfælde erkendes vha. FDC-kredsens læsning af indexpulserne. Kræver man imidlertid "time-out" funktionen, må den laves på anden måde, enten en software "delay-loop", eller en hardware timer.

Efter løsningen af hardware problemerne, konstateredes problemer med andre systemers diske.

Det viser sig, at nogle systemer formatterer diskene på en måde, der ikke kan siges at være helt efter bogen, f.eks. INTERTEC's og MPS's formater med 10 sektorer a' 512 bytes !!; dette format overskrider den tilladte sporelængde med 70 bytes, hvorved inter-record-gap'et mellem sidste og første sektor på de enkelte spor reduceres.

Har man problemer med for små inter-record-gaps, kan det løses på to måder:

Man kan enten formattere kopierings-disketten på sin egen maskine, eller også kan man ned-justere omdrejningshastigheden lidt (ca. 1%) på den fremmede maskines floppy disk drev, for herved at få lidt større inter-record-gaps.

Et tredje problem kan evt. konstateres i forbindelse med i øvrigt korrekt konstruerede disk-kontrollere som f.eks. CROMEMCO's.

Forskelligt fra ældre systemers simple One-Shot multivibratorer, som er i stand til at synkronisere på datastrømmen indenfor 2 pulser, anvender IBM's PC en PLL data-separator (red: Phase Locked Loop); PLL-separatore er givetvis en bedre løsning pga. mindre følsomhed overfor støj og hurtige hastighedsændringer, men for at kunne låse sig til datastrømmen, kræver disse flere "clock-pulser".

Har man problemer med at læse ældre systemers diske og kan man med sikkerhed udelukke de første to fejlmuligheder, vil det ofte vise sig at være problemer med dataseparatoren der er årsagen.

Disse problemer findes der ingen nemme løsninger på, uden at det kommer til at medføre omfattende hardware indgreb - her er der intet andet at gøre end at forsøge igen, evt. prøve om man kan tackle problemet ved at kopiere via en tredje maskine.

Mogens T. Nielsen

ANDRE PROBLEMER MED NEC-765/A

Udover de ovennævnte problemer med NEC 765 FDC'en findes et enkelt, der mig bekendt ikke tidligere har været omtalt i bladet.

Problemet optræder i forbindelse med SEEK-kommandoen, der anvendes hver gang læse/skrive hovedet skal flyttes. For at udføre en SEEK-operation, kræver FDC'en tre bytes fra CPU'en. Disse tre bytes består af selve kommandoen, drev/side samt nummeret på det søgte spor. Hvis det tager mere end 150 usec (300 usec ved 5.25") for CPU'en at overføre disse tre bytes, kan afstanden mellem første og anden step-puls til drevet være op til 1 msec (2 msec ved 5.25") for kort.

Såfremt CPU'en har disabled interrupt mens den overfører de tre bytes, vil problemet normalt aldrig vise sig; oplever man dog af og til, at drevet ikke finder det ønskede spor i første hug (BIOS'en vil sandsynligvis flytte hovedet til spor 0 og prøve påny), kan man prøve enten at disable interrupt førend udsendelsen af kommandoen, eller sætte step-hastigheden 1 (2) msec op.

Leif Olsen

8" DOUBLE DENSITY FLOPPY UDEN DMA

I Z80 systemer, der ikke anvender en DMA-controller til overførsel af data mellem FDC og lager, har det med de normalt anvendte polling-algoritmer været umuligt at køre 8" DD.

Når data overføres ved polling, skal CPU'en varetage datatransporten; dvs. hver gang der skal overføres en byte, skal CPU'en spørge FDC'en, om den er klar til at sende eller modtage en byte. For den FDC, der anvendes i bl.a. MPS-systemet (NEC 765/A), gælder, at den sætter et signal (MRQ) i FDC-statusregistret, hvorefter CPU'en skal overføre en byte inden 13 usec.

Det tyske computerblad c't bragte i efteråret en speciel algoritme til polling af NEC 765, der mig bekendt ikke er set før. I c't anvendes algoritmen i forbindelse med deres ECB-bus computer PROF-80.

For at forstå ideen i denne algoritme, er man først nødt til at se på hvorledes statusregistret i NEC 765 er opbygget.

```
BIT 0-3   Busy-signal for floppy 0 - 3; altid 0 ved læse/-
          skrive-operationer.
BIT 4     1 = læse/skrive-operation igang.
BIT 5     1 = dataoverførsel igang.
BIT 6     dataretning, 0 = lager -> FDC, 1 = FDC -> lager.
BIT 7     1 = klar til at modtage/afgive data (MRQ-bit).
```

Efter at en læse/skrive kommando er afgivet til FDC'en, vil statusregistret altid have følgende udseende:

```
-----
MSB  ! X ! X ! X ! X ! 0 ! 0 ! 0 ! 0 !  LSB
-----
```

Et X angiver, at den pågældende bit er enten 0 eller 1. Ved læse/skrive operationer er der altså kun fire bit der kan variere.

Sættes disse op i en tabel, skal følgende foretages under pollningen:

MSB	LSB	aktion
0 X X X	0 0 0 0	MRQ ikke sat, læs status igen
1 0 0 X	0 0 0 0	operation slut, stop overførsel
1 0 1 0	0 0 0 0	do
1 0 1 1	0 0 0 0	overfør data fra lager til FDC
1 1 0 X	0 0 0 0	operation slut, stop overførsel
1 1 1 0	0 0 0 0	do
1 1 1 1	0 0 0 0	overfør data fra FDC til lager

Der er altså fire forskellige aktioner, afhængigt af status-registrets indhold.

Anvendes status direkte som indeks i en tabel, skal tabellen have 256 indgange, dvs. den fylder en lagerside. Da de 4 mindst betydende bits altid er nul, vil afstanden mellem de mulige indgange være 16 bytes; der er altså plads til at skrive en lille stump kode i hver mulig indgang.

```
; C = i/o-port for FDC status-register
; H = high-byte af adressen på tabellen (i eks. FF)
; C' = i/o-port for FDC data-register
; HL' = adresse på data i lageret
di          ; disable interrupt
call send_command ; send kommando-bytes til FDC
call rw_poll   ; overfør data til FDC siger stop
call read_result ; læs resultat fra FDC
ei          ;

; her følger tabellen (xxxx: angiver adresse)
rw_poll:
ff00: in l,(c)   jp (hl)      ; fortsæt poll
ff10: in l,(c)   jp (hl)
ff20: in l,(c)   jp (hl)
.
.
.
ff80: ret          ; afslut
ff90: ret
ffa0: ret
ffb0: exx outi exx      ; data lager -> FDC
      in l,(c)   jp (hl)
ffc0: ret
ffd0: ret
ffe0: ret
fff0: exx ini exx      ; data FDC -> lager
      in l,(c)   jp (hl)

end
```

Ved denne metode, der nok virker ret uigennemskuelig ved første øjekast, tages der højde for eventuelle "fejltilstande" fra FDC'en. Ser man på den maksimale MRQ-svartid fås følgende:

addr.	instruktion	T-states	kommentar
ff30	in 1,(c)	12	læs status (= 30)
ff32	jp (hl)	4	hop til tabel (addr ff30)
ff30	in 1,(c)	12	læs status (= b0)
ff32	jp (hl)	4	hop til tabel (addr ffb0)
ffb0	exx	4	swap til alternativt reg. sæt
ffb1	outi	12 (16)	overfør data lager -> FDC

ialt: 48 T-states = 12 usec (4 MHz)

En ulempe ved denne metode er, at den optager 256 bytes i lageret; da ikke alle 256 bytes anvendes, kan man dog anvende de resterende til øvrige data.

Leif Olsen

PASCAL

IDIOTSIKRET INPUT I PASCAL.

Den efterfølgende programlistning viser et lille Pascal program, der demonstrerer, hvordan man kan håndtere indlæsning af tekstfelter i PASCAL. Programmet tester hele tiden input af, så man ikke risikerer, at programmet bryder sammen f.eks. af et forkert format. Den viste teknik er meget anvendt i forbindelse med databaser, hvor brugeren udfylder en "fortrykt" formular ved at bevæge sig fra felt til felt, samtidig med at brugeren har mulighed for at rette i allerede indtastede data. Programmet er en pendant til det COMAL-80 program som Lars Peter Larsen bragte i juli nummeret af brugerbladet.

Den centrale procedure i programmet er 'get_streng'. Her indlæses brugerens input direkte fra tastaturet (read(kbd, ch)), og alle kontrolkoder sorteres fra og bruges til kontrol af cursor m.v. I den version af programmet, der bringes i bladet, er kontrolkoderne tilpasset WordStar standard, det vil sige at som pil-taster til cursoren, benyttes cntrl-S, cntrl-E, cntrl-D og cntrl-X. Samtidig har jeg fået plads til pil-tasterne på mit eget tastatur (fabrikat Cherry) der frembringer nogle andre koder, men det er uhyre let at rette i programmet, så de kommer til at passe til et vilkårligt tastatur. 'Pil ned' og 'pil op' får i programmet en lidt andet betydning, idet de bruges til at flytte henholdsvis til næste felt og forrige felt, uanset om de ret fysisk findes på samme linie eller på en anden linie.

Med cntrl-V kan man skifte mellem insert og overstrike mode. I Insert mode indsættes karakteren fra tastaturet på cursorens plads og resten af feltet flyttes en plads til højre. I overstrike mode indsættes karakteren blot istedet for den, der stod der i forvejen. Normalt starter 'get_streng' i insert mode, men hvis tekststrengen i <s> har samme længde som den maksimale længde af feltet, benyttes overstrike mode. Denne sondring er indsat af hensyn til 'get_real', der altid kalder 'get_streng' med maksimal feltlængde. Det virker mest naturligt at rette i tal med overstrike, mens insert er bedst til lidt længere tekststreng.

Programmet giver desuden mulighed for at angive, om der automatisk skal springes til næste felt, hvis cursoren står ved sidste plads i feltet. På den måde kan man befri brugeren for taste CR eller pil ned. (Længe leve dovenskaben).

Programmet benytter sig en del af PASCAL's mængdetype eller som de kaldes på engelsk 'SET'. Denne datatype gør det let at operere med de kendte udtryk fra mængdelæren som f.eks. delmængder og fællesmængder. En mængde, i dette tilfælde en mængde af karakterer, specificeres som enkelt karakterer eller som en rækkefølge af karakterer omkranset af parenteser:

```
legals:= ( . ' ', '-','A'..'Å' . );
```

'legals' sættes lig med mængden bestående af alle de store bogstaver, mellemrum og bindestreg. Herefter er det meget let, at undersøge om en given karakter tilhører denne mængde:

```
IF 'A' IN legals THEN et eller andet
```

Mængdedatotypen bruges i programmet til at specificere hvilke karakterer, man vil acceptere som input i et felt. Desuden skal man angive hvilke karakterer, der betyder, at feltet er færdigt. Alle andre karakterer afvises som input !

Konstruktionen med at angive lovlige karakterer i inputtet gør det meget let at programmere en idiotsikret dialog med brugeren. Hvis man f.eks. har stillet et spørgsmål, og ønsker at få svaret ja eller nej fra brugeren, sætter man blot legals:= (.'j','n'.), og kan så være sikker på, at 'get_streng' kommer med et gyldigt svar. Sætter man yderligere auto:= true, så der automatisk springes til næste felt, behøver man ikke en gang at besvare brugeren med CR.

Jørgen Petersen


```

PROGRAM demo_af_input; (*$A+,C-,V-*)

(* Programmet giver en demonstration af 2 genereller input *)
(* procedurer, der indlæser henholdsvis tekststreng og *)
(* reelle tal. *)

CONST
  BACKSPACE = @008; (* ASCII kode genereret af BACKSPACE *)
  DEL       = @127; (* ASCII kode genereret af DEL-tasten *)

TYPE
  str80 = string(.80.);
  anystr = string(.255.);
  charset = SET of CHAR;

VAR
  str1, str2: string(.50.);
  janej: string(.1.);
  tc: char;
  term: charset;
  n: integer;
  r: real;

(* Returnerer en tekststreng af længden <n> bestående af *)
(* karakteren <c>. *)
FUNCTION cstr(c: char; n: integer): str80;
VAR
  s: string(.80.);
BEGIN
  IF n < 0 THEN n:= 0;
  s(.0.):= chr(n);
  fill(s(.1.), n, c);
  cstr:= s;
END; (* cstr *)

(* Indlæser en tekststreng med mulighed for editering. *)
(* Betydningen af control koderne fremgår af vejledningen, *)
(* der udskrives når programmet køres. (Se sidste i pro- *)
(* gramlisten). *)
(* Ved kald udskrives <s> i positionen <x>,<y>. <l> er den *)
(* maksimale længde på teksstrengen og <legals> de karak- *)
(* terer, der kan indlæses til tekststrengen. <term> er *)
(* de karakterer der afslutter indlæsningen, og <tc> den *)
(* afslutningskarakter brugeren brugte. Hvis <auto> er *)
(* sand afsluttes indlæsningen, når cursoren er ved af- *)
(* slutningen af feltet og <tc> sættes til UM. *)

```

```

PROCEDURE get_streng(VAR s: anystr; l, x, y: integer;
                    legals, term: charset; auto: boolean; VAR tc: char);
CONST
    fc = '.';
VAR
    p: integer;
    ch: char;
    overstrike: boolean;

(* Procedure error udsriver en fejltekst et fast sted på *)
(* skærmen. *)

PROCEDURE error(outstr: anystr);
BEGIN
    gotoxy(0,22);
    write(clreol,outstr);
    gotoxy(x+p, y);
END; (* error *)

BEGIN (* get_streng *)
    gotoxy(x, y); write(s, cstr(fc,l-len(s)));
    IF auto THEN p:= 0 ELSE p:= len(s)-1;
    IF len(s) = 1 THEN overstrike:= true ELSE overstrike:= false;
    REPEAT
        gotoxy(x+p, y); read(kbd,ch);
        error('');
        CASE ch OF
            ÛH, ÛS: (* Flyt cursor en plads mod venstre *)
                IF p > 0 THEN p:= p - 1
                ELSE error('Cursor allerede ved begyndelse af felt');
            ÛI, ÛD: (* Flyt cursor en plads mod højre *)
                IF p < len(s) THEN p:= p + 1
                ELSE error('Cursor allerede ved slutningen af feltet');
            ÛA: (* Flyt cursor helt til venstre i feltet *)
                p:= 0;
            ÛF: (* Flyt cursor til sidste karakter i strengen *)
                p:= len(s);
            ÛG: (* Slet karakter under cursor *)
                IF p < len(s) THEN
                    BEGIN
                        delete(s, p+1, 1);
                        write( copy(s, p+1, 1), fc );
                    END ELSE
                        error('Der findes ingen karakterer til højre for cursc
DEL: (* Slet karakteren til venstre for cursor *)
                IF P > 0 THEN
                    BEGIN
                        delete(s, p, 1);

```

```

        write( BACKSPACE, copy(s, p, 1), fc);
        p:= p - 1;
    END ELSE error('Cursor ved begyndelse af felt');
ÜY:      (* Slet resten af feltet *)
    BEGIN
        write( cstr(fc, len(s)-p) );
        delete(s, p+1, 1);
    END;
ÜV:      (* Skift mellem indsæt/overstrike *)
    IF overstrike THEN overstrike:= false
    ELSE overstrike:= true;
    OTHERWISE (* Indsæt karakter i tekststrengen *)
    IF not(ch IN legals + term) THEN error('Ulovlig karakter')
    IF ch IN legals THEN
        IF p < 1 THEN
            BEGIN
                p:= p + 1;
                IF overstrike THEN delete(s, p, 1)
                ELSE IF len(s)=1 THEN delete(s, 1, 1);
                insert(ch, s, p);
                write( copy(s, p, 1) );
            END ELSE error('Feltet er ikke længere');
        END;
    IF auto and (ch IN legals) THEN ch:= ÜM;
    UNTIL ch IN term;
    p:= len(s);
    gotoxy(x+p, y); write(':l-p);
    tc:= ch;
    END; (* get_streng *)

(* Indlæser et reelt tal i position <x>, <y> i et felt af bred- *)
(* den <width> og med <dec> decimaler efter punktummet.      *)
(* Indlæsningen foregår ved kald af <get_streng>.            *)

PROCEDURE get_real( VAR r: real; width, dec, x, y: integer;
                   term: charset; VAR tc: char);

VAR
    s20: string(.20.);
    i, n, fejl: integer;
BEGIN
    str(r:width:dec, s20);
    REPEAT
        get_streng(s20, width, x, y, ( ' ', '+', '-', '.', '0'..'9' ),
                   term, false, tc);
        i:= 1; WHILE (s20(.i.) = ' ') and (i < len(s20)) DO i:= i + 1;
        n:= len(s20); WHILE (s20(.n.) = ' ') and (n > 0) DO n:= n - 1;
        IF n - i >= 0 THEN
            BEGIN
                val( copy(s20, i, n-i+1), r, fejl);
            END;
        END;
    UNTIL fejl = 0;
END;

```

```

    IF fejl <> 0 THEN
    BEGIN
        gotoxy(0,22); write('Ulovligt format på reelt tal');
    END;
END ELSE
BEGIN
    r:= 0.0; fejl:= 0;
END;
UNTIL fejl = 0;
END; (* get_real *)

BEGIN (* Main Program *)
    (* Slet skærmen og udskriv vejledning *)
    write(clrhom);
    gotoxy(0, 1); write('Demonstration af input procedurer i COMPAS PA
gotoxy(0, 2); write( cstr('-', 79) );
gotoxy(0, 4); write('   Skriv bogstaver: ');
gotoxy(0, 6); write('   Skriv alle tegn: ');
gotoxy(0, 8); write(' Svar Ja eller Nej: ');
gotoxy(0,10); write('Skriv et reelt tal: ');
gotoxy(0,12); writeln( cstr('-', 79) );
writeln('Følgende taster er defineret til editering:');
writeln(' ':5,'ÜH, ÜS: Mod venstre           ÜI, ÜD: Mod højre');
writeln(' ':5,' ÜA: Til venstre kant         ÜF: Til højre kan
writeln(' ':5,' ÜG: Slet under cursor       DEL: Slet til højre
writeln(' ':5,' ÜY: Slet til slut af felt   ÜV: Insert/overst
writeln('Følgende taster skifter til et andet felt:');
writeln(' ':5,'ÜE, ÜK: Forrige felt         ÜX, CR: Næste felt');
writeln(' ':5,' ÜZ: Stopper programmet');
write( cstr('-', 79) );

    (* Skriv tekst i felterne *)
    str1:= 'Denne streng skrives ud på skærmen';
    gotoxy(22,4); write(str1);
    str2:= '1234567890 abcdefghijklmnopqrstuvwxyzå !"#$%&()';
    gotoxy(22,6); write(str2);
    janej:= 'J';
    gotoxy(22,8); write(janej);
    r:= 999.9999;
    gotoxy(22,10); write( r:15:5);

    (* Indlæs bruger input *);
    term:= (. ÜE, ÜK, ÜX, ÜM, ÜJ, ÜZ .);
    n:= 1;
    REPEAT
        CASE n OF
            1: get_streng(str1, 50, 22, 4, (. ' ', 'A'..'Ä', 'a'..'ä'.),
                term, false, tc);
            2: get_streng(str2, 50, 22, 6, (. ' '..'ä' .), term, false, tc
            3: get_streng(janej, 1, 22, 8, (. 'J', 'N' .), term, true, tc);

```

```

4: get_real(r, 15, 5, 22, 10, term, tc);
END;
CASE tc OF
  ÜX, ÜM, ÜJ: IF n = 4 THEN n:= 1 ELSE n:= n + 1;
  ÜE, ÜK   : IF n = 1 THEN n:= 4 ELSE n:= n - 1;
END;
UNTIL tc = ÜZ;
gotoxy(0,23);

```

Bemærk: Tegnet Ü skal læses som "pil-op".

BIBLIOTEKET

NYHEDER I BIBLIOTEKET

Jeg har nu overtaget biblioteket fra Jørgen Gad, der har fået for travlt til at stå for biblioteket.

Bestillingsproceduren er dog indtil videre stadig den samme. Det vil sige, at pengene indbetales til brugergruppens giro-konto, og bag på girokortet skriver man så hvilke volumes, man vil have og i hvilket format (40 eller 80 spor). Prisen er fortsat 60 kr. pr. volume uanset format.

På et par af de nye volumes fylder hovedfilen mere end 190K, der er 40 spors formatet kapacitet, så de kan simpelt hen ikke leveres i dette format. Jeg vil senere prøve at se, om det der muligt at dele disse store filer i mindre dele.

Nedenfor en oversigt over nye volumes siden den sidste store oversigt i januar nummeret 1984. Desuden har jeg fundet en del ældre volumes, som ikke tidligere har været katalogiseret. Sammen med den nævnte oversigt fra januar 84, giver det altså en total oversigt over de 215 volumes, der allerede er i biblioteket.

Diskene er sorteret efter hvilke brugergruppe, der har samlet dem. De anvendte forkortelser er:

C###	C Users Group, USA
G###	CP/M gebruikers groep, Holland
K###	UK Users Group
I###	Infonet
M###	MPS-Brugergruppen, Danmark
P###	Piconet
S###	SIG/M, USA
U###	CP/M Users group, USA
Z###	Z80 Brugergruppen

- C018 Utilities V.
Et tekstformateringsprogram, NRO og 2 forskellige "Disk Doctor", bl.a. DISKDOC fra Dr. Dobbs Journal #66, april 1982.
- C201 Software Tools, volume 1 af 12.
Programmerne har sin oprindelse i "Software Tools" af Brian Kernighan og P. J. Plauger, Addison-Wesley 1976.
Volume 1: Cookbook, table of contents, .FTN files.
- C202 Software Tools, volume 2 af 12.
Tools and primitives in BDS C.
- G506 Basic programmer til løsning af matematiske problemer.
- I006 Modemprogrammer for Apple.
- I008 Diverse Pascal programmer.
- K008 Diverse utility programmer.
Bl.a. disassembler, file compare og meget andet godt. NB.
Dette volume var tidligere katalogiseret som M008.
- K014 Diverse utility programmer.
- K015 Small C i version der producerer Z80 assembler koder.
Desuden rettet for fejl dokumenteret i Dr. Dobbs.
- K016 Diverse utility programmer tilpasset CP/M 3.0.
- K017 Tidsskriftindeks for 1978 - 1983.
Omfatter 3170 artikler fra følgende tidsskrifter:
BYTE Byte 3/80 to 12/83
PCW Personal Computer World 5/78 (First Issue) to 12/83
PRAC Practical Computing 9/78 (First Issue) to 12/83
SOFT Soft 6/83 (First Issue) to 12/83
NB. Hovedfilen er så stor at dette volume ikke kan leveres i 40 spors formatet !
- M006 HBIOS version 1.23
Bios til MPS-2000. Omtalt i Brugerbladet 84:3.
- M008 Dette volume er nu katalogiseret som K008 !
- M009 SYSLIB, volume 1 af 4.
Dokumentation i almindelig ASCII-format plus REL-fil.
- M010 SYSLIB, volume 2 af 4.
Help-filer til online dokumentation af SYSLIB.

- M011 SYSLIB, volume 3 af 4.
Sourcekode i Z80 assembler.
- M012 SYSLIB, volume 4 af 4.
Sourcekode i Z80 assembler.
- P012 Telekommunikation/modem
- P014 Diverse utility programmer.
Bl.a. DCON (et ZSID lignende program) og en tidlig udgave af CCPZ.
- P015 Katalog over CP/M User Group volume 1 - 75.
- P016 Sammenstilling af diverse utilities fra CP/M Users Group.
- S026 Games, electrical engineering package, HAM notebook.
Programmerne er i E-BASIC, der inkluderet.
- S029 Documentation addendum #1.
Kataloger, *.DOC, ABSTRACT.* filer fra SIG/M og CPMUG.
- S030 Documentation addendum #2.
- S040 CROMENCO CDOS til CP/M support.
- S055 Documentation addendum #3.
- S061 DIMS, Infomation Management System.
Omtalt i Brugerbladet 83:4.
- S126 ROFF4, version 1.50
Tekstformaterings program skrevet i C.
- S129 Eksempel på dBASE II programmer.
JRT PASCAL programmer.
NB. Disken er pakket på en speciel måde, der kræver at man har LU.COM fra S119.
- S131 Diverse programmer i PASCAL Z.
Disken består af 54 små procedurer.
- S132 Diverse programmer i PASCAL Z.
Blandet gods.
- S133 Diverse programmer i PASCAL Z.
Indeholder bl.a. et program der kan sammensætte en foderplan for kvæg !

- S134 Diverse programmer i PASCAL Z.
Blandet gods.
- S135 Diverse utility programmer.
Bl.a. et program til relokering af programmer til toppen af TPA.
- S139 Modem program, version 7.12
Source teksten er på 204K ! (god fornøjelse)
NB. Sourceteksten fylder så meget, at dette volume ikke kan leveres i 40 spors format.
- S140 Diverse utility programmer.
Bl.a. cross assembler til 6800 og 68000.
- S142 Spil. Bl.a. Pacman.
- S143 Diverse programmer.
Bl.a. indeks for WordStar tekst filer og setup for EPSON og OKI printere.
- S144 Diverse programmer.
Bl.a. grafik rutiner for EPSON MX80 (kvalitet ukendt).
- S145 VFILER, et skærmorienteret filmanipulations program skrevet af Richard Conn.
- S148 Diverse programmer i PASCAL Z.
Bl.a. en LISP fortolker skrevet i Pascal og et kryptograferings program.
- S152 Diverse utility programmer.
Bl.a. et testprogram til diskdrev, skal dog bruges sammen med en special diskette fra DYSAN.
- S154 Forth-83.
Forbedret udgave af Fig-Forth. Hvis man har lyst til at prøve Forth, er det denne udgave man skal bruge.
- S155 dBASE II programmer.
- S156 Diverse utility programmer.
- S160 Diverse programmer.
Bl.a. sprite grafik for TMS9918 på en Apple og et program til at sætte tidsangivelse på filer (hvordan vides ikke).
- S161 Diverse programmer.
Bl.a. 51K med rettelser til Business Master fra vol. U086 - U090.

S162 Concurrent Pascal-S Compiler.
 PL/O Compiler.

Z004 Diverse COMAL programmer.

Jørgen Petersen

--- 0 ---

MPS-Brugergruppen
 Volume 6 - HBIOS version 1.23.

-CATALOG.006 Indhold af MPS Volume 6
 Released 15. maj 1984
 CRC .COM Checksum program. Viser om
 korrekt kopieret.

Index	name	size	crc	Beskrivelse
006.01	BSTRAP	.MAC 10K	FC 52	Boot-strap. Lægges i EPROM fra
006.02	CONFIG	.SYS 6K	5D A2	Alle system afhængige parametre
006.03	FORMAT	.COM 2K	A9 7D	Formaterings program for 40, 80
006.04	FORMAT	.MAC 16K	FE 10	160 spor.
006.05	HBIOS	.MAC 24K	5D 31	Hovedfilen i HBIOS.
006.06	HLOADER	.MAC 4K	31 48	Loader der placeres på diskette
006.07	MACROS	.MAC 6K	6F F7	Macroer der benyttes af HBIOS.
006.08	PUTSYS	.COM 2K	89 11	Bruges istedet for CP/M's SYSGE
006.09	PUTSYS	.MAC 10K	98 68	/
006.10	READ	.ME 8K	B2 4F	Kortfattet (meget!) forklaring.
006.11	SUBMIT	.COM 2K	D1 16	Rettet version af CP/M's SUBMIT
006.12	SYSGEN	.SUB 2K	8B 14	SUBMIT-fil til generering af sy
006.13	XSUB	.COM 2K	EB 6D	

Copyright (c) 1984 by MPS-Brugergruppen, Copenhagen

MPS-Brugergruppen
 Volume 9 - SYSLIB version 2.7 Dokumentation og REL-fil

-CATALOG.009 Indhold af MPS Volume 9
 Released 29. maj 1984
 CRC .COM Checksum program. Viser om diske
 er korrekt kopieret.

Index	name	size	crc	Beskrivelse
009.01	SYSLIB	.REL 14K	1C 54	SYSLIB REL-fil med alle subrout
009.02	SYSLIB	.TXT 108K	80 7B	User and Reference Manual

009.03	SYSLIBHD.TXT	12K	00 C1	Indholdsfortegnelse til SYSLIB.
009.04	SYSLIBI .TXT	10K	49 3A	
009.05	SYSLIBID.TXT	4K	0B A8	
009.06	SYSLIBR .TXT	4K	E7 7E	
009.07	SYSLIBS .TXT	42K	DB 19	Eksempler på brug af SYSLIB
009.08	SYSLIBUG.TXT	124K	25 66	User's Guide for SYSLIB.

Copyright (c) 1984 by MPS-Brugergruppen, Copenhagen

MPS-Brugergruppen

Volume 10 - SYSLIB version 2.7 Help-filer og demo programmer

-CATALOG.010	Indhold af MPS Volume 10 Released 27. maj 1984
CRC .COM	Checksum program. Viser om diske er korrekt kopieret.

Index	name	size	crc	Beskrivelse
010.01	HELP .COM	2K	D6 CC	Hovedprogram for help-fil syste
010.02	SYSLIB .HLP	16K	DF D5	Help-filer
010.03	SYSLIB1 .HLP	14K	65 47	/
010.04	SYSLIB2 .HLP	4K	DD 5E	/
010.05	SYSLIB3 .HLP	8K	BF 68	/
010.06	SYSLIB4 .HLP	6K	4D 9C	/
010.07	SYSLIB5 .HLP	6K	C6 6F	/
010.08	SYSLIB6 .HLP	4K	1A AE	/
010.09	SYSLIB7 .HLP	6K	9C 70	/
010.10	SYSLIB8 .HLP	8K	A2 A5	/
010.11	SYSLIB9 .HLP	8K	60 75	/
010.12	SYSLIBA .HLP	8K	E9 66	/
010.13	SYSLIBB .HLP	6K	21 39	/
010.14	SYSLIBC .HLP	10K	4C E8	/
010.15	SYSTEST .MAC	4K	29 D2	Demonstrations programmer.
010.16	SYSTEST1.MAC	4K	8D CF	/
010.17	SYSTEST2.MAC	12K	51 BE	/
010.18	SYSTEST3.MAC	4K	1D D2	/
010.19	SYSTEST4.MAC	6K	1C 40	/
010.20	SYSTEST5.MAC	2K	2B D7	/
010.21	SYSTEST6.MAC	6K	9D F9	/
010.22	ZCPR2 .IDX	12K	0B A7	Oversigt over ZCPR2 og SYSLIB i

Copyright (c) 1984 by MPS-Brugergruppen, Copenhagen

-CATALOG.011

Indhold af MPS Volume 11

Released 27. maj 1984

CRC .COM

Checksum program. Viser om diske
er korrekt kopieret.

Index	name	size	crc	Beskrivelse
011.01	S0FILEIO.MAC	2K	EA B6	SYSLIB Sourcekode
011.02	S1FILEIO.MAC	2K	A6 1C	/
011.03	S2FILEIO.MAC	2K	7A 8E	/
011.04	S3FILEIO.MAC	2K	D8 60	/
011.05	SBBLINE .MAC	4K	5F E1	/
011.06	SBDOS .MAC	2K	A8 85	/
011.07	SBIOS .MAC	4K	99 33	/
011.08	SBLINE .MAC	4K	6D A7	/
011.09	SCAPS .MAC	2K	34 BD	/
011.10	SCAPSTR .MAC	2K	16 30	/
011.11	SCATH .MAC	2K	F5 20	/
011.12	SCCOUT .MAC	2K	43 4A	/
011.13	SCIN .MAC	2K	4F 47	/
011.14	SCLINE .MAC	2K	82 03	/
011.15	SCLOUT .MAC	2K	A1 8B	/
011.16	SCODEND .MAC	2K	2C 8D	/
011.17	SCOMP .MAC	2K	F4 F4	/
011.18	SCOMPHD .MAC	2K	41 56	/
011.19	SCONDIN .MAC	2K	3D 7A	/
011.20	SCOUT .MAC	2K	67 1A	/
011.21	SCPOUT .MAC	2K	C0 0E	/
011.22	SCRC .MAC	4K	2C A7	/
011.23	SCRC1 .MAC	4K	90 ED	/
011.24	SCRC2 .MAC	4K	F5 89	/
011.25	SCRLF .MAC	2K	1A 66	/
011.26	SCST .MAC	2K	10 43	/
011.27	SDIR .MAC	34K	01 31	/
011.28	SEN .MAC	2K	16 AD	/
011.29	SEPRINT .MAC	2K	93 AB	/
011.30	SEPSTR .MAC	2K	36 DB	/
011.31	SEVAL .MAC	4K	3C CC	/
011.32	SEVAL1 .MAC	2K	11 01	/
011.33	SEVAL2 .MAC	2K	37 49	/
011.34	SEVAL3 .MAC	2K	A5 85	/
011.35	SEVAL4 .MAC	2K	4A DC	/
011.36	SFCLOS .MAC	2K	97 C1	/

011.37	SFDEL	.MAC	2K	71	2B	/
011.38	SFEXIST	.MAC	2K	75	8C	/
011.39	SFILEIO	.MAC	10K	E4	DC	/
011.40	SFILL	.MAC	2K	C8	B9	/
011.41	SFMAKE	.MAC	2K	7C	A2	/
011.42	SFNAME	.MAC	6K	D5	AA	/

Copyright (c) 1984 by MPS-Brugergruppen, Copenhagen

MPS-Brugergruppen

Volume 12

- SYSLIB version 2.7 Sourcekode

-CATALOG.012

Indhold af MPS Volume 11

Released 29. maj 1984

CRC .COM

Checksum program. Viser om diske er korrekt kopieret.

Index	name	size	crc	Beskrivelse
012.01	SFOPEN	.MAC	2K 99 81	SYSLIB Sourcekode
012.02	SFREAD	.MAC	2K 2E 50	/
012.03	SFRENAME	.MAC	2K 75 2C	/
012.04	SFWRIT	.MAC	2K A5 FD	/
012.05	SINITFCB	.MAC	2K 8C 63	/
012.06	SINLINE	.MAC	6K 4B C1	/
012.07	SINSFORM	.MAC	4K A6 CD	/
012.08	SINSTR	.MAC	2K F3 78	/
012.09	SLA2HC	.MAC	2K 09 7E	/
012.10	SLADC	.MAC	2K B6 3F	/
012.11	SLCRLF	.MAC	2K 78 B7	/
012.12	SLHL4HC	.MAC	2K 67 23	/
012.13	SLHL5DC	.MAC	2K 3A 52	/
012.14	SLOUT	.MAC	2K 57 A1	/
012.15	SLPRINT	.MAC	2K E2 9B	/
012.16	SLPSTR	.MAC	2K C6 0A	/
012.17	SMA2HC	.MAC	2K D8 50	/
012.18	SMADC	.MAC	4K E8 A2	/
012.19	SMATH	.MAC	6K AE 09	/
012.20	SMHL4HC	.MAC	2K E6 73	/
012.21	SMHL5DC	.MAC	4K 9B 41	/
012.22	SMOVE	.MAC	4K 6C 28	/
012.23	SPA2HC	.MAC	2K 6C 09	/
012.24	SPADC	.MAC	2K 4E 50	/
012.25	SPAUSE	.MAC	4K F6 1E	/
012.26	SPHL4HC	.MAC	2K 3A 4A	/
012.27	SPHL5DC	.MAC	2K BB 6D	/
012.28	SPOUT	.MAC	2K 87 81	/
012.29	SPRINT	.MAC	2K A8 35	/
012.30	SPSTR	.MAC	2K 73 4F	/

012.31	SRAND	.MAC	2K	CD 93	/
012.32	SRIN	.MAC	2K	E6 D2	/
012.33	SSCANNER	.MAC	2K	A4 85	/
012.34	SSORT	.MAC	16K	FF 97	/
012.35	SUD	.MAC	2K	CA 9E	/
012.36	SVERSION	.MAC	2K	8A 45	/
012.37	SZCPR	.MAC	2K	58 C2	/
012.38	SZFNAME	.MAC	22K	53 0E	/
012.39	SZGPINS	.MAC	4K	E9 0A	/

Copyright (c) 1984 by MPS-Brugergruppen, Copenhagen

RETTELSE

RETTELSE TIL COMAL-80 FELTEDITOR - 1984:4

I mit indlæg i brugerbladet i Juli-nr. 1984, om indtastning af data i fastlagte skærmfelter, har jeg fundet en skønhedsfejl og en mere alvorlig fejl.

Den nemme først: Linierne 8520 til og med 8550 skal fjernes. De er et levn fra da jeg hængte de to procedurer sammen med ENTER-kommandoer og har altså noget at gøre her mere.

Den alvorlige fejl viser sig ved at EDITFELT-proceduren "overser" den forreste "#" i print-using masken, når den også indeholder fortegn. Derfor skal den del af proceduren, der scanner masken, rettes ti nedenstående:

Lars Peter Larsen

```

8000 PROC EDITFELT(X#, Y#, REF FELT, REF KEY#,MASKE$) CLOSED
8010  SLUT#:=FALSE; MIN :=10; MAX:=10; FORTEGN:=0
8020  IF POS("+",MASKE$) OR POS("-",MASKE$) THEN FORTEGN#:=1
8030  IF FORTEGN# AND FELT<0 THEN
8040      SIGN#:= -1
8050  ELSE
8060      SIGN#:=1
8070      FELT:=ABS(FELT)
8080  ENDIF
8090  A#:=POS("#",MASKE$)
8100  B#:=LEN(MASKE$)-A#
8110  IF POS(".",MASKE$)<>0 THEN B#:=POS(".",MASKE$)-A#+FORTEGN#-1
8120  WHILE A#<=B# DO
8130      MAX:=MAX*10; A#:+1
8140  ENDWHILE

```

```
8150  A#:=LEN(MASKES)  
8160  IF POS(".",MASKES)<>0 THEN B#:=POS(".",MASKES)-1  
8170  WHILE B#<A# DO  
8180      MIN:=MIN/10; A#:-1  
8190  ENDWHILE
```

ANNONCER

SÆLGES

Følgende kort samt tilbehør til MPS-maskine sælges:

- 1 stk CAI kort, fuldt bestykket til 2 stk MDCR, virker
- 1 stk MDCR, Philips Mini Digital Casette Recorder
- 18 stk Kasettebånd til ovennævnte MDCR

Henvendelse til: Christian Glinnum, Degnemose Alle 18,
2700 Brønshøj, Tlf: (01) 28 26 57

FÆLLESKØB

I forbindelse med bygning af PROF-80 kort (europakort med Z80+ FDC+128K lager, beregnet til ECB-bussen; beskrevet i det tyske blad c't nr 8-10-1984), søges kontakt med andre, der ligesom jeg har haft svært ved at skaffe de lidt mere eksotiske stumper som f.eks. 9229B (fra SMC - dataseparator til FDC; se evt. MPS84:4) og uPD1990 (real-time-clock fra NEC). Ovennævnte kredse er på det nærmeste umulige at opstøve hos de normale løsdelsforhandlere, men kan skaffes via de respektive importører.

Henvendelse til: Leif Olsen, Tlf: (02) 94 98 20

GAMLE NUMRE AF BRUGERBLADET

Hvis du er interesseret i gamle numre af brugerbladet, har vi stadig et lager af de fleste numre. De kan købes for 5 kr. pr. stk. hos kassereren.

ADRESSER

INDMELDELSE OG KONTINGENT

Indmeldelse i brugergruppen kan foretages ved henvendelse til et medlem af bestyrelsen. Kontingentet er pt. 100 kr. årligt. Der opkræves ikke indmeldelses-gebyr.

SOFTWARE

Bestilling af volumes fra brugerbiblioteket sker ved forudbetaling på gruppens giro-konto. Der beregnes 60,- kr. for hvert volume. Der leveres i 40 og 80 spors MPS-format.

Bibliotekar:

Jørgen Petersen
Sofiegade 24
1418 København K
Tlf. (01) 54 91 76 (bedst hverdage kl 18 til 20)

BESTYRELSEN

Formand:

Niels Veilleborg
Finsensvej 11B 2.
2000 København F
Tlf. (01) 87 80 85

Kasserer:

Ken Nørreris
Figenvej 154
4700 Næstved
Tlf. (03) 72 78 34

Sekretær:

Bendt Prüser
Greisvej 100
2300 København S
Tlf. (01) 55 01 27

Redaktion:

Leif Olsen
Kildestrædet 46
2740 Skovlunde
Tlf. (02) 94 98 20

Mads Westermann
Flensborggade 28
1669 København V
Tlf. (01) 31 41 11

GIRO KONTO 1 60 65 81
MPS Brugergruppen
Kassereren
Figenvej 154
4700 Næstved

Oplag: 200

Tryk: Dansk Tidsskrifts Tryk