



PICCOLINIEN - i vælten

Siden sidste nummer af PICCOLINIEN har der været meget skriveri i aviserne om PICCOLINIEN. Stærkest kom det til udtryk gennem bladet Gymnasieelev, hvor man bl.a. påstod:

- at maskinen overhovedet ikke virkede
- at det var en »skraldet« maskine
- at RC på det nærmeste havde bestykket hele amter til at købe PICCOLINE
- at den desuden ville være totalt uanvendelig til brug i skolerne.

Det var hårde ord, og for de mange der havde anskaffet sig en PICCOLINE, har det nok været chokerende læsning. »Var det en total fejl-investering, man havde foretaget sig? Ville udgiften til eventuelt at bringe udstyret i orden beløbe sig til enorme summer? Var udstyret forældet, allerede inden man havde fået det?«

Sådanne tanker er sikkert gået igennem hovedet på flere af brugerne. Men heldigvis blev påstandene da også kraftigt dementeret i de efterfølgende dages aviser - dels fra talrige lærere rundt omkring i landet - dels fra skoleledere der var ansvarlige for indkøb af maskiner - og dels - og det er måske mest bemærkelsesværdigt - fra det svenske skolevæsen.

Netop på samme tid færdiggjorde den svenske Skolöverstyrelse nemlig sit arbejde med at udvælge de skoledatamater, som man ville yde støtte til. I Sverige får skolerne godtgjort halvdelen af anskaffelsesprisen, hvis de køber en af de godkendte datamater. Man havde testet 38 forskellige maskiner, heriblandt flere

af de store udenlandske mærker og flere danske, og kun 9 blev godkendt - hvoraf de 3 var de svenske maskiner, der deltog. Mange af de store udenlandske maskiner blev underkendt, og fra Danmark blev kun to godkendt: PARTNER og PICCOLINE, begge fra Regnecentralen.

Af de 9 maskiner, der blev godkendt, var der desuden kun 2, der fik direkte ros. Den ene var PICCOLINIEN, der fik speciel ros for udformningen af sit tastatur. Men selvom RC fik denne opbakning, skal kritikken alligevel tages alvorligt. Det er rigtigt, at der har været en række »børnesygdomme« på PICCOLINIEN, men der er vist ikke mange maskiner, der helt undgår sådanne. Disse fejl og u hensigtsmæssigheder er i øjeblikket ved at blive rettet, og alle får i denne tid opdateret deres systemer og maskiner uden nogen udgift for køberen.

Meddelelse om dette er allerede gået ud til alle skoler, der har købt PICCOLINE.

PICCOLINIEN rummer dog som sædvanlig meget andet end artikler om Regnecentralen og PICCOLINIEN. Vi har været glade for, at de første læserreaktioner - i form af opfordringer til at skrive artikler om specielle emner - er begyndt at dukke op, ligesom at strømmen af nye abonnenter på bladet endnu ikke er stoppet.

Alt ialt ser det ud til, at PICCOLINIEN er ved at udvikle sig til et vigtigt medium i debatten omkring EDB i undervisningen.

PICCOLINIEN

LYD PÅ PICCOLINE

PICCOLINE har indbygget en lydgenerator med fire lyd-kanaler, tre tonekanaler (kanal 0, 1 og 2) og en »støj«-kanal (kanal 3).

Der er indbygget en lille forstærker og højttaler i selve PICCOLINENS kabinet, men det er ikke den bedste lyd, man får frem på denne måde. Der er imidlertid mulighed for at hente lyden ud af et DIN stik på PICCOLINENS bagside, og har man en skærm med indbygget forstærker og højttaler, kan man lave et kabel til overførsel af lyden hertil. Man kan selvfølgelig også koble andre forstærkere på og få lyden ud den vej.

(M.h.t. stikforbindelse i AUDIO stikket henvises til manualen).

Fra RcComal80 kaldes lydgeneratoren med det reserverede navn »sound«. Der kan åbnes en datastrøm til lydgeneratoren ved en sætning som:

```
OPEN FILE 5, »SOUND«, WRITE.
```

Derefter kan kommandoer gives ved f.eks.: PRINT FILE 5, CHR\$(X);

Der kan gives tre typer af kommandoer: Lydstyrke, frekvens og støjtype.

Uden at gå i detaljer med det mere tekniske, bringes her en række procedurer, som kan være til hjælp i forbindelse med brug af lydgeneratoren fra RcComal80.

Man kan sætte lydstyrken for hver kanal individuelt. Maksimal lydstyrke fås ved 0 og minimal lydstyrke (stilhed) fås ved 15.

Her er en procedure, der kaldes med angivelse af kanal nummer og den ønskede lydstyrke:

(Det forudsættes selvfølgelig, at datastrømmen er åbnet).

```
0010 PROC styrke(kanal,st)
0020   PRINT FILE 5:
      CHR$(128+32*kanal+16+st);
0030 ENDPROC styrke
```

Skal man have lukket for alt lyd, kan man bruge følgende procedure:

```
0050 PROC stilhed
0060   EXEC styrke(0,15)
0070   EXEC styrke(1,15)
0080   EXEC styrke(2,15)
0090   EXEC styrke(3,15)
0100 ENDPROC stilhed
```

Der skal to kontroltegn til at sætte frekvensen, og det kan klares ved at kalde denne procedure:

```
0120 PROC frekvens(kanal,hz) CLOSED
0130   i:= INT(62500/hz)
0140   b1:= i MOD 16
0150   b2:= i DIV 16
0160   PRINT FILE 5:
      CHR$(128+32*kanal+b1);CHR$(b2);
0170 ENDPROC frekvens
```

Det har naturligvis kun relevans at sætte frekvensen for tonekanalerne – dog har tonekanal 2 en forbindelse til støjkanalen.

Støjtypen kan vælges til såkaldt »hvid støj« eller til »brum«. Samtidig kan man sætte en frekvens for støjkanalen. Man kan enten vælge faste værdier med henholdsvis 0, 1 og 2 som kode, men vælger man 3, vil støjkanalen få »samme« frekvens som lydkanal 2.

I nedenstående procedure kan 'fb' antage 2 værdier: 1 = 'hvid støj' – 0 = 'brum'. for 'fr' medfører 0, 1 og 2 faste frekvenser, mens 3 vil medføre samme frekvens som lydkanal 2:

```
0190 PROC støj(fb,fr)
0200   PRINT FILE 5:
      CHR$(128+64+32+4*fb+fr);
0210 ENDPROC støj
```

Som eksempel på brug af lydgeneratoren er her et lille program, der spiller en skala. Programmet udnytter nogle af de foranstående procedurer.

```
0500 //SKALA
0510 OPEN FILE 5, »SOUND«, WRITE
0520 FOR i:= 1 TO 8 DO
0530   READ hz
0540   EXEC frekvens(0,hz)
0550   EXEC styrke(0,0)
0560   FOR vent:= 1 TO 1000 DO
0570     NEXT vent
0580   EXEC stilhed
0590 NEXT i
0600 CLOSE
0610 DATA 262,294,330,349,392,400,494,523
0620 //SLUT PÅ SKALA
```

Nedenstående eksempel efterligner et pistolskud.

```
1000 //SKUD
1010 OPEN FILE 5, »SOUND«, WRITE
1020 EXEC støj(1,0)
1030 FOR s:= 0 TO 15 STEP 0.1 DO
1040   EXEC styrke(3,INT(s))
1050 NEXT s
1060 CLOSE
1070 //SLUT PÅ SKUD
```

God fornøjelse!

PRINTER 2 VED OPSTART

Når flere PICCOLINER deles om disktestation og skriver, tilsluttes den sidste via printerstikket på disktestationen.

Alle PICCOLINER skal da benytte »printer 2« som aktuel skriver. Det kan derfor være nyttigt at indrette sit system således, at aktuel skriver automatisk sættes til »printer 2« ved opstart.

For at gøre dette, må vi ændre i filen »STARTUP.0«, som indeholder opstartordrer.

Denne fil er normalt sat til »read-only«, dvs. at man kun kan læse fra den og ikke ændre i den. Den skal derfor ændres til »read-write«, der giver mulighed for ændringer. Dette sker ved, at man fra det yderste niveau under CCP/M (når der står A>) giver følgende ordre:

```
A> set startup.0ÆrwÅ
```

Systemet svarer da med meddelelsen:

```
A:STARTUP.0 set to system (SYS),  
read write (RW)
```

Herefter kan man starte RcComal80 systemet op på sædvanlig vis, og STARTUP.0 kan så ændres ved følgende lille RcComal80 program:

```
0010 DIM ordres$ OF 80  
0020 DELETE "startup.0"  
0030 OPEN FILE 1,"startup.0", WRITE  
0050 INPUT "INDTAST ORDRER ADSKILT AF // :  
": ordres$  
0060 PRINT FILE 1: ordres$  
0070 CLOSE FILE 1
```

Programmet er alment, så det kan bruges til at lægge en eller flere opstartsordrer ind i rækkefølge. De enkelte ordrer,

der udføres i den rækkefølge, de er skrevet, skal ved indtastningen adskilles af to skråstreger.

Et eksempel på kørsel kunne være:

```
Fun  
INDTAST ORDRER ADSKILT AF // :  
printer 2//graphics//comal80//graphics no  
END AT 0070
```

Opstartsfilen er nu ændret, og ved næste opstart fra denne diskette vil opstartsfilens ordrer blive udført i den rækkefølge, de er angivet:

Først sættes aktuel skriver til »printer 2«.

Derefter indlæses grafiksystemet.

Så startes RcComal80 op.

Når man er færdig med at køre RcComal80 (dvs. når man skriver BYE), lukkes grafikken igen.

Det skal bemærkes, at metoden med de to skråstreger kun kan benyttes, hvis man kører under CCP/M-version 3.1.

Benytter man en tidligere version, skal ordrene skrives til filen STARTFIL.SUB og på en linie ad gangen. Man kan da benytte følgende program:

```
0010 DIM ordres$ OF 80  
0020 DELETE "startfil.sub"  
0030 OPEN FILE 1,"startfil.sub", WRITE  
0040 REPEAT  
0050 INPUT "INDTAST ORDRE  
- AFSLUT MED RETUR : ": ordres$  
0060 PRINT FILE 1: ordres$  
0070 UNTIL ordres$ IN " "  
0080 CLOSE FILE 1
```

Her indtastes ordrene en ad gangen, og de to skråstreger skal **ikke** anføres.

EOF og EOD i RcComal80

Da flere har problemer med End-Of-File og End-Of-Data, vil vi hermed præcisere, hvornår de to funktioner giver værdien SAND:

EOF: Antager værdien SAND første gang, man **forsøger** at læse udover det sidste dataelement i en datastrøm. (Der er således en fejl i RcComal80-manualen s.42, bemærkninger til linie 0020-0050).

EOD: Antager værdien SAND, når det sidste element i datalisten er læst.

Mere skatte- jagt



I sidste nummer af PICCOLINIEN fortalte vi om et nyt program til brug i franskundervisningen, kaldet La Chasse Au Tresor, der er udviklet af Francoise Andersen.

Efter artiklen i bladet har salget af programmet, der hidtil er foregået gennem Regnecentralen og dens forhandlere, været så godt, at Francoise Andersen har besluttet sig for at koncentrere sig helt om videreudviklingen af dette og andre spil. I den forbindelse har hun endvidere besluttet sig for selv at overtage salget af programmet.

Ønsker man derfor fremover at købe La Chasse Au Tresor, bestilles det hos

Francoise Andersen

Tlf. (01) 70 72 51

Telefon-svarer er tilkøbt.

Prisen er naturligvis uændret kr. 800,- + moms.

5-faset EDB-grundkursus for samtlige 670 læsere i en kommune

En del af PICCOLINIENS læsere har hørt om det efteruddannelsesprojekt omkring EDB, som er startet i Helsingør kommune, og har bedt PICCOLINIEN om at beskrive det nærmere.

Vi har derfor besøgt viceskoledirektør Holger Scheibel fra Helsingør kommune, der har været en af hovedkræfterne bag projektet.

Projektet strækker sig over perioden efterår 84 til forår 86, og er delt ind i 5 faser:

- 1) Foredragsrække med 4 datalogiske eksperter, som har redegjort for de historiske og tekniske sider af EDB-teknologien.
- 2) »Taster under figurerne«. Et en-dags kursus, hvor lærerne får lov til at arbejde lidt med EDB-maskiner, primært i form af afprøvning af programmer, tekstbehandling, søgning af data i databaser osv.
- 3) Igen en foredragsrække, men denne gang med samfundsdebattører, der ikke er EDB-specialister.
- 4) En »Kædefase«, hvor hver enkelt skole kan fremsætte ønsker om lokale kursusformer, på baggrund af arbejdet i de tre første faser.

- 5) En studiekredsfasen hvor man, enten på den enkelte skoler eller på tværs af skolerne, kan oprette studiekredse til fordybelse i enkelte emner.

»Man skal dog ikke«, siger Holger Scheibel, »betragte disse fem faser som noget endegyldigt, og tro at man dermed har gjort nok. Det er blot de fem faser, vi på nuværende tidspunkt kan overskue. Jeg vil da f.eks. ikke udelukke, at der i 86/87 bliver taget hul på en eller anden form for kommunalt kursus i datalære, der kan virke som et supplement til lærerhøjskolen kurser«.

Han mener heller ikke, at kurset på nogen måde vil gøre lærerne til fuldbefarne EDB-lærere.

»Men det er vigtigt, at vi gennem kurser af denne art får alle lærerne med, idet vi ikke ønsker at få delt lærergruppen op i to: dem der ved noget om EDB, og dem der overhovedet ikke ved noget om det, og måske endda er bange for det«.

Det er også baggrunden for, at man i Helsingør kommune har valgt, at det skal være samtlige lærere, der kommer på kursus og ikke bare de interesserede, der evt. senere skal varetage undervisning i datalære.



Holger Scheibel

»Vi vil gerne«, siger Holger Scheibel, »have lærerne op på et bevidsthedsniveau om dette emne som gør, at de alle sammen kan deltage i drøftelserne af det. Hidtil har det typisk været matematik/fysik-lærere der vidste noget om

disse ting. Hvad vi gerne vil nu er ikke bare at få alle de skeptiske med, men også alle dem der underviser i fag, hvor det ikke for dem er direkte indlysende, at datalogi kan spille en rolle«.

I den sammenhæng mener Holger Scheibel også, at det måske er vigtigere, at der i hver enkelt klasse står en eller to maskiner, som man kan benytte i den daglige undervisning, fremfor at man opbygger egentlige datalærelokaler. »Men så er det også vigtigt, at lærerne ikke er fuldstændigt fremmede overfor, hvad der foregår inde i disse maskiner, men kan benytte dem som en naturlig del af undervisningen«.

Selvom alle lærere deltager i kurset, opfatter man det ikke som en tjenesteplicht. »Når vi ikke gør det«, fortsætter Holger Scheibel, »hænger det sammen med, at vi hele tiden har haft fælleslærerrådet og den lokale lærerforening med i planlægningen af kurset. Det betød, at vi fra starten kunne lægge det frem, ikke bare som noget der udgik herfra, men som noget alle lærere kunne være med i fra starten. Reaktionen på kurset har da indtil videre også udelukkende været positiv«.

Hvad er et styresystem

af Hans Lauesen

Det er normalt en længere historie at beskrive, hvad et styresystem er, men her vil vi forsøge at gøre en lang historie kort.

Når man skriver et program til en datamaskine, er det fordi, man gerne vil have maskinen til at løse en bestemt opgave. Med andre ord: man vil gerne have elektronikken i maskinen til at opføre sig på en bestemt måde.

På en datamaskine uden styresystem skal brugerens program selv styre de enkelte komponenter («chips»), således at maskinen kan udføre den opgave, man har brug for.

Dette stiller store krav til programmøren. Han skal nemlig kende de enkelte komponenter i maskineriet, og vide hvordan de styres, og hvordan de er sat sammen. Dette giver mange fejlmuligheder. Desuden er der ikke adgang til oversættere og fortolkere (f.eks. Pascal eller RcComal80), og det betyder, at alle programmer må skrives i maskinkode og indkodes i maskinen f.eks. ved hjælp af vippekontakter på maskinens front.

En anden ulempe er at programmer, skrevet til een maskintype, ikke kan flyttes til andre typer maskiner uden store programændringer. Dette skyldes, at ikke alle maskiner indeholder de samme elektroniske komponenter.

Et styresystem kan løse alle disse problemer.

Styresystemet er, groft sagt, maskinens »sjæl«.

Styresystemet er et systemprogram, som man normalt får med i købet, når man anskaffer en datamat.

Styresystemet indeholder en række procedurer til styring af maskinens elektronik. Disse procedurer kan udnyttes fra brugerens programmer. Procedureerne kan f.eks.:

- Skrive et tegn på skærmen
- Hente et tegn fra tastaturet
- Skrive et tegn på printeren
- Læse eller skrive en sektor på disketten osv.

Hvis man på forhånd har procedurer, der kan løse disse opgaver, er det ikke så svært at programmere, da man ikke længere behøver at bekymre sig om selve elektronikken i maskinen. Man kan stadig programmere i maskinkode, men den eneste elektroniske komponent man behøver at kende er selve mikroprocessoren («hjernen» i datamaten). Al anden elektronik styres ved hjælp af styresystemets procedurer.

For at man kan have glæde af en diskette, må der findes visse regler for, hvordan man gemmer programmer og

data på disketten, og det er styresystemets opgave at sørge for, at disse regler bliver overholdt. Styresystemet indeholder derfor forskellige procedurer, der kan åbne og lukke filer, samt skrive og læse i dem.

Sammen med et styresystem leveres som regel forskellige hjælpeprogrammer. Disse kan f.eks. bruges til kopiering af filer eller udskrive en oversigt over filer på disketten.

Som en hjælp til udvikling af programmer findes der oversættere. En oversætter er et program, der kan oversætte et andet program skrevet i et såkaldt højere programmeringssprog til maskinkode. Pascal er et eksempel på et højere programmeringssprog. Det er således heller ikke nødvendigt for programmøren at kende selve mikroprocessoren, hvis han/hun har en sådan oversætter.

Hvis forskellige maskiner forsynes med det samme styresystem, betyder det, at maskinerne vil opføre sig ens, set fra brugerens synspunkt og fra programmerens synspunkt. De forskelle, der skyldes maskinens elektronik, er nemlig elimineret, idet al styring af elektronikken sker ved hjælp af de procedurer, styresystemet stiller til rådighed.

Dette medfører, at programmer kan flyttes fra en maskine til en anden uden de store vanskeligheder, forudsat at de to maskiner har samme mikroprocessor og samme styresystem. (Dette gælder dog ikke alle programmer, idet der findes programmer, som styrer dele af elektronikken uden om styringssystemet).

For at gøre det let for en bruger at benytte forskellige maskiner, er der mange datamater, der benytter samme styresystem. Et eksempel på et styresystem, der kører på flere forskellige datamater, er Concurrent CP/M-86 (CCP/M-86).

CP/M er en forkortelse for »Control Program/Microprocessors«.

På PICCOLINE anvendes CCP/M-86. CCP/M-86 på PICCOLINE indeholder procedurer til styring af PICCOLINENS elektronik. CCP/M-86 indeholder også en række hjælpeprogrammer, bl.a. til kopiering af filer og udskrivning af en oversigt over filer på disketten.

Et eksempel på en anden maskine, der benytter CCP/M-86, er Regnecentralens PARTNER. Dette betyder, at det er muligt at flytte programmer mellem PICCOLINE og PARTNER, uden at det er nødvendigt at ændre programmerne.

Det er ikke alene en fordel for brugere af en datamat at anvende et standardstyresystem, det er også en fordel for programleverandører. Standardstyresystemerne gør det nemlig muligt for en programleverandør at sælge de samme programmer til mange forskellige brugere, selvom brugerne har forskellige maskiner. Det eneste der kræves er, at styresystemet er det samme.

Der er dog (mindst) een forskel ved datamaskiner, som styresystemet ikke kan udligne; nemlig forskellen på disketteformaterne. Der findes nemlig ingen standard for, hvordan man lagrer information på disketter (eller rettere - der findes for mange standarder). Dette gør, at man kun sjældent kan flytte en diskette fra en maskine til en anden med et godt resultat. I de fleste tilfælde må man stille de to maskiner op ved siden af hinanden og flytte programmerne gennem et kabel. Men i det tilfælde vil det så igen være en fordel at anvende et standardstyresystem. F.eks. ligger CCP/M-86 så tæt op ad CP/M, der benyttes på Regnecentralens PICCOLO, at det er muligt at overføre programmer mellem disse to maskiner via et kabel.

GSX – et seriøst bud på en grafik-standard

```

0010 FUNC f(x)
0020   RETURN x*SIN(x'2)
0030 ENDFUNC f
0040
0050
0060 OPEN GRAPHICS 1
0070   forhold:= 3/2
0080 DIM svar$ OF 1,kont(6),ind(10),ud(10),pktud(100,2)
0090
0100 PROC filled_area(REF pktind(,),ant)
0110   kont(1):= 9
0120   kont(2):= ant
0130   kont(4):= 0
0140   GSX kont,ind,pktind,ud,pktud
0150 ENDPROC filled_area
0160
0170 PROC polyline(REF pktind(,),ant)
0180   kont(1):= 6
0190   kont(2):= ant
0200   kont(4):= 0
0210   GSX kont,ind,pktind,ud,pktud
0220 ENDPROC polyline
0230
0240 FUNC xenhed
0250   RETURN 10'(INT(LOG(fx*xmax)/LOG(10)))
0260 ENDFUNC xenhed
0270
0280 FUNC yenhed
0290   RETURN 10'(INT(LOG(fy*yamax)/LOG(10)))
0300 ENDFUNC yenhed
0310
0320
0330 PROC xakse
0340   MOVETO fx*xmin,0
0350   DRAWTO fx*xmax,0
0360   MOVETO xenhed,0
0370   TEXT "!"
0380 ENDPROC xakse
0390
0400 PROC yakse
0410   MOVETO 0, fy*ymin

```

»Hvis man med de sædvanlige R-Comal80 grafik-ordrer er i stand til at udføre GSX-operationerne, hvorfor indfører man da de sidste?»

Svaret er, at man dels får et mere dynamisk værktøj, dels at alle tegneoperationerne bliver udført væsentligt hurtigere, og dels at der selvfølgelig også er nogle ekstra faciliteter, der kan gøre ens tegninger mere spændende.

En af de mest benyttede ordrer i GSX er ordren POLYLINE. Parameteren til denne ordre, er en to-dimensional tal-tabel, der indeholder koordinaterne til en række punkter. Et kald til POLYLINE bevirker, at man får tegnet en kurve gennem de mange punkter i den rækkefølge de står i tal-tabellen. Denne linie kan så ændres på forskellige måder. Feks. findes der en GSX-ordre (SET POLYLINE LINETYPE) til at afgøre om den skal være punkteret (flere forskellige måder), en anden (SET POLYLINE LINE WIDTH) til at ændre bredden af kurven, og flere andre.

En anden ordre der ofte benyttes er FH

Programeksempel, der benytter GSX

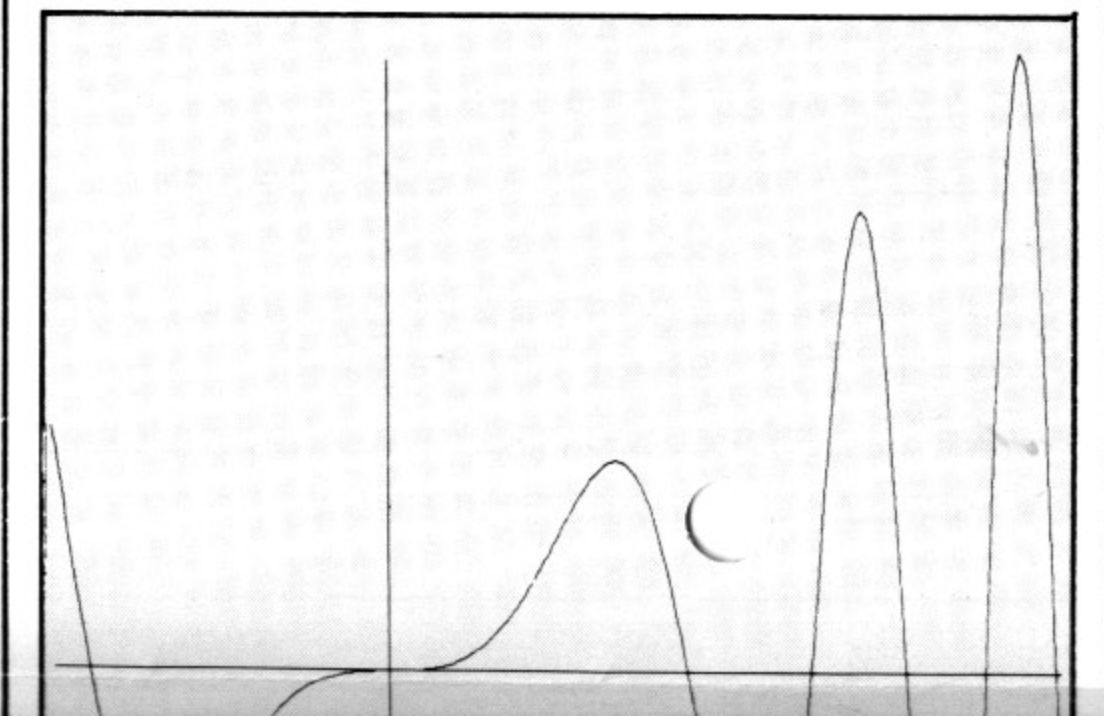
I det blå område er vist et større programeksempel, der bl.a. benytter den nye GSX-ordre. Denne ordre kaldes dog kun via kald af procedurer med navne svarende til de gængse GSX-ordrenavne (der er dog intet krav om at kalde procedureerne disse navne). Er man fortrolig med procedure-begrebet, vil dette nok være den naturlige måde at benytte den nye GSX-ordre på.

Programmet bruges til at tegne grafen for en vilkårlig funktion. Selve funktionsudtrykket er angivet i linie 0020, og man kan skifte til en ny funktion ved at ændre denne linie (ordret RETURN skal blive stående). Man vælger under udførslen af programmet selv sit interval på x-aksen, og hvor mange punkter man vil have udregnet værdier i.

Programmet sørger også for at tegne et koordinatsystems akser med enheder på (ens på begge akser) hvis dette ligger indenfor det område, man arbejder med.

Grafen kan tegnes på to måder:

- 1) Som normal graf
- 2) Som areal-kurve - dvs. en tegning af de arealer, der afskæres af kurven og ligger henholdsvis over og under x-aksen.



```

0380 ENDPROC xakse
0390
0400 PROC yakse
0410   MOVETO 0,fy*ymin
0420   DRAWTO 0,fy*ymax
0430   MOVETO 0,yenhed
0440   TEXT "-"
0450 ENDPROC yakse
0460
0470 PROC lav_vindue
0480   bredde:= xmax-xmin
0490   højde:= ymax-ymin
0500   IF bredde<højde*forhold THEN
0510     fx:= forhold*højde/bredde
0520     fy:= 1
0530   ELSE
0540     fx:= 1
0550     fy:= bredde/(højde*forhold)
0560   ENDIF
0570   WINDOW fx*xmin,fx*xmax,fy*ymin,fy*ymax
0580 ENDPROC lav_vindue
0590
0600 INPUT "Indtast start-x-værdi : ": xmin
0610 INPUT "Indtast slut-x-værdi : ": xmax
0620 INPUT "Indtast antal delepunkter : ": n
0630 DIM koord(0:n+1,2)
0640 ymax:= 0
0650 ymin:= 0
0660 j:= 1
0670 FOR i:= xmin TO xmax STEP (xmax-xmin)/(n-1) DO
0680   koord(j,1):= i
0690   koord(j,2):= f(i)
0700   IF koord(j,2)>ymax THEN ymax:= koord(j,2)
0710   IF koord(j,2)<ymin THEN ymin:= koord(j,2)
0720   j:= j+1
0730 NEXT i
0740 lav_vindue
0750 koord(0,1):= koord(1,1)
0760 koord(n+1,1):= koord(n,1)
0770 INPUT "Ønskes en areal-kurve - svar j/n : ": svar$
0780 PRINT CHR$(12)
0790 IF xmin<=0 AND xmax>=0 THEN yakse
0800 IF ymin<=0 AND ymax>=0 THEN xakse
0810 IF svar$ IN "Jj" THEN
0820   koord(0,2):= 0
0830   koord(n+1,2):= 0
0840   filled_area(koord,n+2)
0850 ELSE
0860   koord(0,2):= koord(1,2)
0870   koord(n+1,2):= koord(n,2)
0880   polyline(koord,n+2)
0890 ENDIF
0900 INPUT svar$
0910 CLOSE GRAPHICS

```

I vores serie om forskellige grafiske muligheder, skal vi i dette nummer beskæftige os med et af de eneste forslag til en standard indenfor grafik-området på mikrodatamater. Det drejer sig om systemet GSX, der er en forkortelse for Graphic System Extension.

Fra og med release 1.3 af RComal80, der fungerer sammen med release 3.1 af CCP/M er der mulighed for at komme i kontakt med GSX-systemet, der er det grafiksystem, der benyttes på en PICCOLINE. Vi skal i denne artikel dels gøre rede for, hvordan denne implementering er sket, og dels give lidt omtale af systemets virkemåde.

Det skal dog med det samme anføres, at GSX-systemet er væsentligt vanskeligere i sin opbygning end de sædvanlige grafikordrer i RComal80, og da man med disse er i stand til at udføre de mest anvendte grafik-operationer i GSX, kan det ikke anbefales uøvede at kaste sig ud i brug af GSX.

Artiklen her henvender sig derfor mere til de øvede programmører, men alle kan selvfølgelig have glæde af at læse den, ligesom de fleste vil kunne benytte det gennemgåede program.

Man kan med det samme stille spørgsmålet:

POLYLINE LINE WIDTH) til at ændre bredden af kurven, og flere andre.

En anden ordre der ofte benyttes er FILLED AREA. Parameteren til denne ordre er magen til parameteren til POLYLINE, men denne ordre tegner en udfyldt polygon omkranset af de angivne punkter. Systemet finder selv ud af at forbinde sidste punkt med første, så der faktisk kommer en polygon ud af det. Ligesom ved POLYLINE findes der til denne ordre også en række hjælpeordrer, der kan angive hvilken type udfyldning (massiv, ternet, stribet o.a.), hvilken farve osv. man ønsker.

Der findes desuden også en række muligheder for at skrive tekster i forskellige højder, evt. drejet i forhold til vandret, med forskellige farver (hvis man har farveskærm) osv.

Hele dette standard-system er udviklet, så det fungerer sammen med enhver maskine, der benytter CCP/M - styresystemet, og f.eks. kan det samme grafiske system benyttes, hvad enten man kører i RComal80 eller i Poly-Pascal.

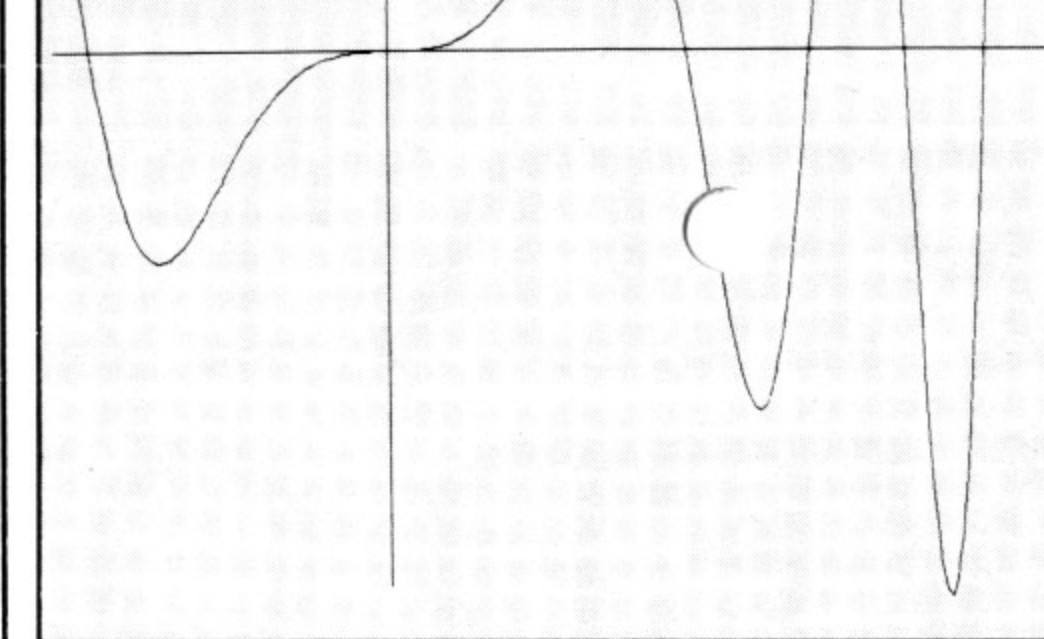
Hvordan er dette system så blevet implementeret i RComal80?

Jo - da man allerede har de fornødne ordrer til grafik i RComal80, har man valgt at lade muligheden for brug af denne mere avancerede facilitet fremkomme ved en enkelt ordre kaldet - GSX!

Denne ordre kaldes så med 5 parametre, hvor den første er en en-dimensionel tal-tabel, hvis værdier angiver hvilken GSX-ordre man ønsker at få udført, hvor mange punkter man har i sit koordinatsæt, osv.

Man kan sige, at der er blevet skabt et »hul« ind til GSX for den mere avancerede bruger.

Hvordan dette hul benyttes i praksis er nærmere beskrevet i det tillæg til manualerne, der udsendes sammen med den nye RComal80-version, men det vil nok også være en god ide for den interesserede bruger, at anskaffe sig Digital Research's GSX-Guide, der i detaljer forklarer GSX-systemet.



I denne forbindelse er der en lille fidus i programmet. De to endepunkter forekommer nemlig umiddelbart to gange. Men hvis man skal tegne en areal-kurve, flyttes det ene af hver af endepunkterne ned på x-aksen. Herved opnår vi, at forbindelsen mellem det første og det sidste punkt ligger oveni x-aksen, så vi får delt den udfyldte kurve i det, der er over og det, der er under x-aksen.

Programmet sørger i proceduren lav_vindue for at lave grafik vinduet på en måde, så en enhed på de to akser ser ens ud. Som programmet står her, vil grafen blive vist på skærmen, men ønsker man tegningen ud på papir, ændres linie 0060 til OPEN GRAPHICS 21, og linie 0070 til forhold: = 2/3.

TEKSTBEHANDLING – en vigtig anvendelse af EDB

Benyttelsen af datamater til avanceret tekstbehandling, er vel nok en af de mest almindelige anvendelser af EDB i dag. Langt de fleste større firmaer og offentlige virksomheder har benyttet denne mulighed for at anvende datamaten som en avanceret skrivemaskine i årevis, og efterhånden som også mange mindre firmaer og skoler – i takt med udbredelsen af mikrodatamater – har fået datakapacitet, er det blevet almindeligt også disse steder.

Men der er stadig mange lærere, der ikke er klar over mulighederne ved anvendelse af denne facilitet.

Meget af ens tid som lærer går med at skrive noter, der kopieres og udleveres til eleverne. Ofte benytter man gamle noter, som man klipper og klistrer sammen, retter med hvidt blæk, for derefter at kopiere endnu engang. Til sidst har disse kopier en tendens til at blive ulæselige.

Benytter man et tekstbehandlingssystem, kommer man ud over dette problem, idet man har mulighed for at sammenflette og rette i tidligere skrevet tekst direkte på en dataskærm.

Et tekstbehandlingssystem virker i første omgang som en ganske almindelig skrivemaskine. Forskellen er blot, at papiret her er en dataskærm. Man skriver på nøjagtig samme måde, som man ellers ville skrive, men kan forøge skrivetempoet betydeligt, idet fejl ikke umiddelbart betyder noget. Skriver man forkert et sted, kan man enten med det samme rykke tilbage på skærmen og rette – slet og ret ved at skrive oveni det gamle. Man kan også skrive hele teksten færdig uden tanke for de fejl, man laver undervejs og så til sidst rette dem alle på en gang.

Ligeledes har man mulighed for at omformulere, indsætte nye sætninger midt i det hele, eller flytte (evt. fjerne) en sætning fra et sted til et andet. Det kan godt være fra en hel anden tekst, at man overfører et afsnit.

En vigtig ting er også, at man undervejs ikke behøver at tænke på linieksift.



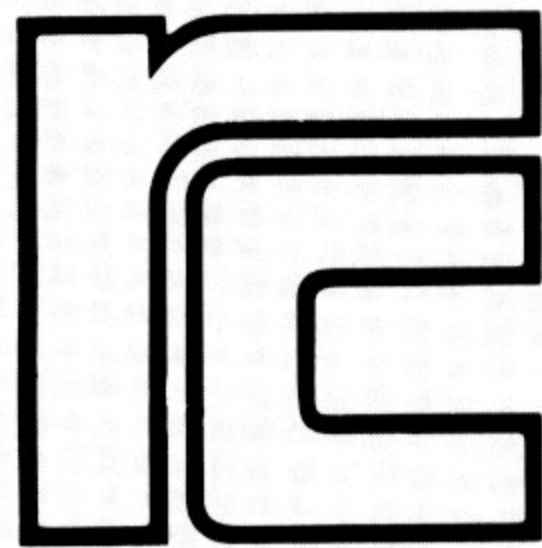
Systemet sørger selv for at skifte linie, på en sådan måde, at det ord, man er i gang med, som måtte være for langt, flyttes ned på næste linie. Når man skriver, behøver man altså udelukkende at koncentrere sig om tastaturet – hvilket måske især er en fordel for uøvede, der ikke kan benytte »tífingersystemet«.

Når selve teksten er færdig, kan man bryde den om og f.eks. lave plads til et billede i den ene side, ved at ændre bredden af teksten i et bestemt afsnit, eller man kan ændre hele teksten, så den får en bredde svarende til en spaltebredde osv.

Når teksten derpå er ombrudt, skriver man den til sidst ud på papir ved hjælp af en skriver. Også her har man forskellige redigeringsmuligheder: at sørge for en lige højremargin, at ændre linieafstand, at få foretaget understregninger, osv.

Mulighederne er utallige, og her er kun beskrevet ganske få af de faciliteter, man har. Med et tekstbehandlingssystem følger også en manual med en fuldstændig beskrivelse af alle de forskellige specialkoder, der sætter en i stand til at lave alverdens ændringer i ens tekst.

Mange bliver i første omgang skræmt af alle disse koder, og tænker at »alt dette lærer jeg aldrig udenad«. Men dette er heller ikke på nogen måde nødvendigt. Langt de fleste lærere har relativt almindelige krav til et tekstbehandlingssystem, og vil i første omgang udelukkende bruge det som en avanceret skrivemaskine. De fleste benytter måske kun en 5-6 forskellige af de muligheder, man har, men allerede dette er nok til, at man får en enorm lettelse, når man skriver tekster.



Udgiver:

Regnecentralen
Hovedvejen 9
2600 Glostrup

Indlæg fra læserne:

Skal sendes til ovenstående
adresse.

Redaktion:

Mogens Guildal (ansv.)
Ole Schwander Olsen

Salgsafdeling:

Hovedvejen 9
2600 Glostrup
Tel.: (02) 96 07 00
henviser til nærmeste
forhandler

Supportcenter:

Hovedvejen 9
2600 Glostrup
Tel.: (02) 96 07 00

Teknisk service:

Glostrup
Hovedvejen 9
2600 Glostrup
Tel.: (02) 96 07 00

Århus

Klamsagervej 19
8230 Åbyhøj
Tel.: (06) 25 04 11

Aalborg

Limfjordsvej 14
9400 Nørresundby
Tel.: (08) 17 80 44

Odense

Henovej 10
5270 Odense N
Tel.: (09) 18 78 15

Grafisk tilrettelægning og tryk:

Johnsen + Johnsen a/s,
København