

3

5. ÅRGANG

1981

data lære

Udgivet af

FORENINGEN FOR DATALÆRE OG ANVENDELSE AF EDB I UNDERVISNINGEN

ISSN 0107-0002

Er der noget at betænke sig på ?

Indførelsen af datalære i folkeskolen har med skiftende temperaturer været en lunken kartoffel i nu mere end 10 år.

Det er også mere end 10 år siden, at sagen blev taget op på officielt plan gennem nedsættelsen af Johnsen-udvalget i september 1970, og som bekendt anbefalede dette udvalg i sin betænkning fra 1972, at der blev åbnet for datalære som valgfag fra 8.klassetrin, og at der så senere burde tages stilling til evt. placering som selvstændigt obligatorisk fag.

Desværre er det endnu ikke gået, som det blev anbefalet i Johnsen-rapporten, og da der heller ikke har været iværksat systematisk uddannelse af lærere til datalære, står vi i dag i en situation, der sådan set ikke er ret meget anderledes, end den man var i, da Johnsen-udvalget afgav betænkning.

Selvfølge er der sket en del i de 10 år, bl.a. har der været en del vellykkede forsøg med datalære under Folkeskolens Forsøgsråd, og Danmarks Lærerhøjskole har heldigvis fortsat med aktiviteter på området, men bortset fra en række indhentede erfaringer, er startbetingelserne omtrent de samme, som de var i 1972.

Derfor kan det ikke undre, at man ender i den samme konklusion som Johnsen-udvalget:

1.) Der skal åbnes for datalære som valgfag.

2.) Senere skal der tages stilling til en evt. obligatorisk placering af visse emners behandling.

Denne løsning ligger lige for. Man må så håbe, at de forskellige beslutningstagere får øje på den, og ikke drukner sig for meget i principielle diskussioner om datalære for alle eller for de få, fagtrængsler og andre trængsler for ikke at tale om diskussionen om kreative contra boglige indslag i undervisningen - hvor datalære heldigvis kan være begge dele, og derfor må kunne stille alle tilpas

Datalæreforeningen vil primært arbejde for legalisering af datalære som et valgfag.

En indførelse af datalære som valgfag vil efter foreningens opfattelse indebære en række fordele på kort sigt, og det vil ikke hindre den senere omvurdering, hvor der skal tages stilling til en obligatorisk indførelse af visse af emnerne.

Fordele vil bl.a. være:

Datalære kan tages op, hvor forudsætningerne for det er til stede, d.v.s hvor der er interesserede lærere med viden og uddannelse på området, og hvor der er sådanne muligheder for adgang til edb-kraft, at undervisningen kan gennemføres.

Udviklingen på området vil kunne komme til at foregå i en rolig takt, hvor anskaffelse af dataudstyr, efteruddannelse af lærere og fremstilling af undervisningsmaterialer kan finde sted parallelt med og baseret på yderligere indhentede erfaringer.

Nogle vil hævde, at datalære ikke bør placeres som valgfag, idet det så ikke er alle, der får viden om området. Dette er selvfølgelig rigtigt, men er det trods alt ikke bedre, at nogle får en viden om og en holdning til edb, i modsætning til nu hvor stort set ingen hører og lærer noget. Så må det heller ikke glemmes, at placeringen af datalære som valgfag kun skal være åbningen for emnernes behandling i folkeskolen.

Men lad os få den åbning NU. Den er på mange måder en slags forudsætning for, at udviklingen kan gå videre. Og at der åbnes for datalære som valgfag, forhindrer jo ikke den fortsatte diskussion af en evt. obligatorisk placering af nogle emner.



Trods strejke og lock-out på typografområdet er det lykkedes os at få dette blad udgivet. Skulle kvaliteten ikke være lige så høj som den plejer for så vidt angår opsætning og tryk, beder vi om forståelse fra læserne.

Red.

En SIMULATOR

- et forsøg på at pille sjælen ud af data-maskinen.

Når man er nybegynder udi datalæren, har man en række problemer, hvis eksistens man senere er tilbøjelig til at glemme. Jeg er overbevist om, at det største problem nok er, at for at kunne forstå lidt mere komplekse processer, som de fremstår på run-time i en datamaskine, så skal man have forstået, hvad der „ egentlig ” sker i maskinen under udførelsen af et program, m.a.o. man skal have forstået procesdynamikken, og her er det nemlig ikke altid nok, at man har modtaget undervisning i programmering (og forstået principperne, samt opnået en vis øvelse i selv at programmere) og i en datamaskines opbygninger (lager, registre o.s.v.). Problemet er at få forbundet den detailviden, der er erhvervet på disse to områder.

Jeg tror, de fleste lærere i datalære har oplevet elever, som tilsyneladende har forstået principperne i en datamats opbygning, og som samtidigt er i stand til at skrive programmer af den mere enkle slags, men som pludselig laver underlige ting i deres programmer eller ved spørgsmål afslører de mærkeligste vrangforestillinger om, hvad maskinen rent faktisk kan gøre, eller hvad der rent faktisk sker på run-time. En nærmere analyse af problemet vil tit vise — det er min erfaring — at elevens detailviden faktisk er til stede, men det glipper altså med at overføre den til den konkrete situation.

Problemet eksisterer på alle niveauer, hvor der undervises i datalære for begyndere. Jeg husker selv at have haft problemet, da jeg startede datalogi — studiet på universitetet (jeg kendte intet til programmering eller andre datalogiske emner på forhånd). 10 — øren faldt vist først et godt stykke hene i 1. studieår.

Som pædagogisk problem har emnet hidtil været ret upågtet, den almindelige holdning til det har været, at det er noget der løser sig selv, når man har arbejdet længe nok med programmering.

Det gør det vel også — som så mange andre problemer — men det fritager bare ikke emnet for en viderebearbejdning i pædagogisk øjemed.

Mange af os, som underviser i datalære, har matematisk baggrund og glemmer måske tit, at der er en markant forskel på matematik og programmering: matematikken beskæftiger sig næsten udelukkende med statiske strukturer, mens programmering i sin natur er dynamisk. Hvis emnet programmering kun formidles som et statisk emne (syntaks og semantik), er man nødt til at forlade sig på den enkelte elevs selvstændige øvelser i programmering. Frugten af dette arbejde afhænger helt af samme elevs evne til, så at sige, at visualisere en dynamisk proces, nemlig afviklingen af en algoritme. Når det er konstateret, må det også være klart, at ikke alle programmeringsopgaver er lige velegnede som en hjælp til forståelse af samspillet imellem maskine og program.

Som emnet her er beskrevet tager det sig umiddelbart ud som et problem i programmeringsundervisning, som jo kun er en mindre del af undervisningen i datalære, når denne ikke direkte sigter imod en erhvervskompetance. Men emnets tætte sammenhæng med datamatens fysiske organisation gør det velegnet som middel til at få afmystificeret datamaskinen, få trukket sjælen ud af den, eller hvad man nu ønsker at kalde det. Det mål må være et af de fornemste, når man underviser begyndere i datalære.

Det følgende er et bud på, hvordan man kunne tackle problemet i begynderundervisningen i datalære. Emnet hedder: DESIGN & PROGRAMMERING AF EN SIMULATOR TIL EN SIMPEL DATAMASKINE. Undervisningsforløbet er afprøvet i forbindelse med den projektorienterede del af datalære på HF. Emnet var en succes, og jeg er sikker på at vi kom et godt stykke på vejen til afmystifikation af datamaskinen. Jeg har dog ikke foretaget nogen konkrete målinger

af effekten (målt som f.eks. bedre forståelse for programmering og programafvikling). Jeg håber, at andre ville kunne have glæde af emnet, som er særdeles velegnet til gruppearbejde (i hvert fald programmeringsdelen), da delopgaverne i arbejdet falder i ret klart afgrænsede dele, således kan det også blive et godt eksempel på struktureret programmering.

FORUDSÆTNINGER

Emnet var det sidste undervisningsforløb, d.v.s. at eleverne på det tidspunkt fra lærebogen kendte til, at et program gemmes i maskinens lager, at der findes regneregistre (aritmetisk enhed), at man skal bruge en instruktionstæller, at man skal kunne kommunikere med ydre enheder. Desuden havde vi arbejdet lidt med en assembly - kode fra lærebogen. Men nu skulle vi altså igang med vort eget maskinsprog.

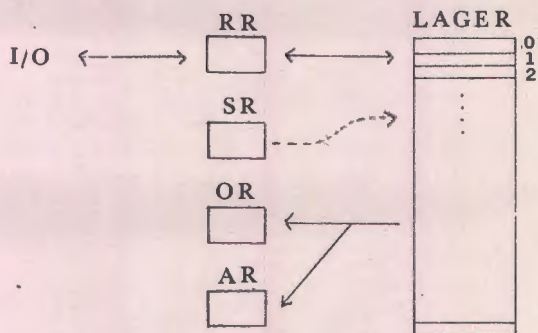
DESIGN

Hvad skulle maskinen kunne? Vi var hurtigt enige om, at den skulle kunne det samme som en lommeregner - mindst. Det blev til +, -, √, *. Desuden: instruktioner til at få data ud og ind af maskinen (i/o - instruktioner) og til at hente og gemme data i lageret. Holdet skulle have lidt hjælp i form af et par småopgaver, for at indse, at vi manglede hop - instruktioner og enkelte andre.

Hvilke registre må vi udstyre maskinen med? Fra lærebogen kendte vi til funktionen af instruktionstælleren (her kaldet SR for styreregister), så en sådan måtte vi have. Men hvad med den aritmetiske enhed? Ofte består den bl.a. af flere forskellige registre med hver sine opgaver, f.eks. additionsregister og multiplikationsregister. For overskuelighedens skyld valgte vi dog at udstyre maskinen med ét register: regneregistret (RR). På dette tidspunkt løb vi ind i en diskussion om datatyper. M.a.o. skulle vi lade vores maskine kunne håndtere forskellige datatyper eller skulle den kun kunne „ knuse tal“? Vi enedes om kun at lade den behandle reelle tal. Det var ikke nogen væsentlig begrænsning i forhold til formålet med undervisningsforløbet - men nok i forhold til vore oprindelige ønsker om, at den skulle kunne mere end en lommeregner. For heller ikke at få kompliceret programmeringen unødigt, enedes vi om at simplificere i/o på den måde, at

al dataoverførsel imellem centralenheden og ydre enheder sker via RR. Yderligere to registre måtte med: OR og AR (se afsnittet „ Ordlængde m.m. ”).

Maskinens registerstruktur fremgår af denne figur:



Nu var tiden inde til at teste funktionsdygtigheden af det design, vi var nået frem til, så vi gik i gang med konstruktion af flere små programeksempler af typen „ indlæs to tal, adder dem og udskriv resultatet ”, udskriv tallene 1, . . . , 10, o.s.v. Her viste det sig hurtigt, at det foreløbige instruktionssæt, vi var nået frem til, ikke slog til, f.eks. havde vi glemt en stop instruktion, vi savnede en instruktion, som kunne anbringe et tal direkte i RR uden at indlæse det først (direkte instruktion), og vi savnede betingede hop instruktioner. Til slut nåede vi frem til dette instruktionssæt:

ADD	(1)	MULD	(7)	SKRIF	(13)
SUB	(2)	DIVD	(8)	HOP	(14)
MUL	(3)	HENT	(9)	HOPN	(15)
DIV	(4)	GEM	(10)	HOPP	(16)
ADDD	(5)	SÆT	(11)	HOPM	(17)
SUBD	(6)	LÆS	(12)	STOP	(18)

Vores instruktionscyklus så således ud:
GENTAG

hent instruktion

SR = SR + 1

udfør instruktion

INDTIL instruktion = STOP

Det er min vurdering, at det betaler sig at bruge god tid på dette sted i forløbet for at sikre sig at eleverne er helt fortrolige med instruktionscyklus'en og for at give dem øvelse i at skrive småprogrammer i assembly - kode. Hvor langt, man vil nå på dette punkt, må afhænge af ti-

den, og i hvor stort omfang man vil involvere eleverne i programmeringsfasen (herom senere).

ORDLÆNGDE m.m.

I instruktionscyklus'en står „ hent instruktion ”. Denne rutine kræver nærmere forklaring, idet den indbefatter både overførsel af et ord i lageret og en afkodning i ordredel og adressedel, til hhv., OR og AR. Det er mindre væsentligt, hvor meget, man vil gøre ud af denne del. Da vi ikke havde gjort særligt meget ud af binære tal – eleverne vidste selvfølgelig, hvad en bit var – fandt vi her en god anledning til at tage emnet op. I første omgang skete det i form af en lille opgave: „ Hvor mange bits skal vi bruge, hvis hver af vore ordrer skal have sit eget bitmønster? Find på et mønster til hver!”. Det viste sig at være en fin anledning til at arbejde lidt med de binære tal, fordi eleverne straks så fordelene i at „ tælle binært ” i stedet for at prøve sig frem med mønstre til de forskellige ordrer. Ordredelen af en instruktion optager altså her 5 bits, resten kan så bruges til adressedelen. Principelt er det selvfølgelig ligegyldigt, hvilken ordlængde man benytter, vi valgte her – af hensyn til programmeringen – at lægge os fast på en bestemt ordlængde, nemlig 16 bits, og brugte lidt tid på at tale om de begrænsninger, det sætter m.h.t. adresseringsområde. Desuden brugte vi lidt tid til at undersøge talområdet som funktion af antallet af bits i et ord – men kun for heltallenes vedkommende, de reelle tal lod vi ligge.

Programmeringen

Denne del af forløbet kan gøres variabel i tidsforbrug, afhængigt af hvor skrap holdet er til at programmere. Men opgaven at programmere en simulator til denne maskine er så velstruktureret en opgave med rutiner på fra 3-4 programlinier til 60-100 linier, at selv svage programmører på holdet kan bidrage med noget. Den vanskeligste del af opgaven – som for læreren ikke skulle være problemer – er at skrive loaderen/assemblyren (her kaldet „indsæt program”). Jeg tog mig af den del af opgaven:

Indsæt program

plads = -1

GENTAG

plads = plads + 1 næste ord i lageret
input ordre

CASE ordre OF

„add”:

„sub”:

(her læses eventuelle operander og det korrekte bitmønster indlægges i lageret).

„stop”:

INDTIL ordre = „stop”

I fællesskab lavede vi på tavlen „ hent ordre ”, som blev noget i retning af:

a = lager (SR) , hent lagerelement
b = mod (a, 1024) ; adressesedlen pilles ud
OR = (a - b) / 1024, 6 forreste bits i OR
AR = b , 10 bagerste bits i AR

Også „ udfør ordre ” blev lavet i fællesskab:

CASE OR OF

WHEN 1 exec add

WHEN 2 exec sub

ENDCASE

Derimod blev rutinerne add, sub o.s.v. samt en initialiseringsrutine skrevet af eleverne, hvorpå vi i fællesskab samlede det hele. I det færdige program består hovedprogrammet af „ indsæt program ” og „ initialiser ” efterfulgt af instruktionscyklus'en (som vist tidligere).

Tilbage stod nu en væsentlig del af programmeringen: nemlig tilføjelsen af yderligere kommentarer for at højne læsbarheden af produktet. Det havde vi et par gode timer med, idet vi måtte samarbejde en mængde forslag til et brugbart hele. Indkøringen gik ret smertefrit – vi havde jo efterhånden fået korrekturlæst programmet ret grundigt. Desværre havde vi ikke COMAL til rådighed, men kun BASIC med alle de ulemper det giver ved konstruktioner af lidt større programmer (for mange GOTO's og for dårlige faciliteter til konstruktion af underprogrammer). Og så ville det jo også have været rart at have haft et udvidet arsenal af datatype, især i en opgave af denne art – men det kan jo heller ikke COMAL TILBYDE!

Den sidste dobbelttime gik med at afprøve først nogle af vore gamle assembly – programmer og med at skrive og afprøve et par nye, et arbejde eleverne kastede sig over med stor iver nu da der var mulighed for at få dem kørt også.

ERFARINGERNE

Vi brugte ialt 8 x 2 timer på projektet, så det var egentlig en del, vi nåede på den tid. Men det er kun muligt, når læreren - som jeg gjorde det i dette tilfælde - går styrende ind i programmeringsfasen. Men man behøver altså ikke afsætte et halvt år til et projekt af denne art! At vi så med fordel kunne have arbejdet med det i endnu længere tid, er en anden sag. Det viste sig nemlig, at de timer vi havde brugt i det foregående halvår på emnet maskinstruktur næsten havde været spildte i forhold til det store udbytte eleverne fik af emnet nu, hvor de for alvor fik lejlighed til at arbejde med det.

Jeg synes også, det er en vigtig erfaring at gøre - og den er der nok mange andre, som har gjort rundt omkring - at selvom eleverne ikke selv udfører hele programmeringsarbejdet, oplever de alligevel at være med, og det er ikke svært at forklare principperne i selv de vanskelige programafsnit. Gruppearbejdet omkring programkonstruktionen er jo også eksemplarisk i forhold til den praksis, der findes overalt i samfundet, hvor der arbejdes med EDB. Hvor vanskeligt det er at samarbejde forskellige folks bidrag til et samlet system - herunder dokumentationen - fik vi også snust lidt til.

Begrebet simulation var én af de fluer, vi fik smækket ved samme lejlighed. Med vores program i lageret opfører maskinen sig nemlig pludselig som en KUK (det kaldte vi vores maskine!) i stedet for som en METRIC. Desværre fik vi ikke tid til at gøre mere ved det emne. Det kunne ellers have været spændende. De erfaringer, eleverne gjorde under konstruktionen med KUK - simulatoren, havde givet gode forudsætninger for at forstå, at simulation af en anden datamat på en datamat ikke er principielt forskelligt fra simulation af et hvilket som helst andet system på en datamat.

Der er måske til slut grund til at nævne, at selvom mikrodatamaterne nu for alvor vælter ind i undervisningssektoren - det var de knapt begyndt på, da mit HF - hold arbejdede med dette emne - så er emnet ingenlunde forældet. Integrerede kredse eller ej: de nye datamaters instruktions-sæt tager sig stadig yderst „klassisk” ud.

FINN H. KRISTIANSEN,
MARSELISBÖRG SEMINARIUM.

Midt i en overgang

- en debatfilm om edb.

Produktion: Bent Barfod Film for
Kommunedata og Statens Filmcentral 1980.

Musik: Jean Michel Jarre.

Tid: 18 min.

”Midt i en overgang” er ikke en undervisningsfilm i sædvanlig forstand, men en debatfilm. Selv betegner Bent Barfod filmen som ”et forsøg på i en munter tone at give informationer og opfordre til at deltage i debatten om edb og samfund”.

Ved en international industrifilmfestival i København i september 1980 fik filmen ikke mindre end 3 priser:

Førsteprisen i gruppen af film fra edb-branchen, den ene af festivalens to Grand Prix'er samt IQ-prisen, der er filmproducenternes egen anerkendelse til en kollega.

Filmene er udmærket og kan absolut anbefales til undervisningsbrug. På grund af sin stil og sine mange spørgsmål kræver den nok i højere grad end andre film at indgå i en sammenhæng i undervisningen, og tid må naturligvis afsættes til diskussion af filmens synspunkter og spørgsmål.

Filmene indeholder mange trickoptagelser. De fleste er fremragende godt lavet og mange er morsomme. Filmene er i hvert fald ikke kedelige.

Filmens speakertekst giver et indtryk af filmens synspunkter og spørgsmål:

”Menneskets egentlige særkende er dets evne til at skabe symboler, altså til at tænke abstrakt. Med så enkle tegn som 0 og 1 kan vi meddele os til hinanden. Der er ingen grænser for, hvad disse symboler sætter os i stand til. Derfor må vi selv trække grænserne, og vi må være sikre på at vore informationer er rigtige, før vi giver dem videre.

De gamle verdensbilleder var højst forunderlige. Jorden var flad som et landkort og søfolkene var svære at lokke ud på de store have, de var bange for at sejle ud over kanten. Så kom Columbus og gjorde jorden til en kugle. Det gav kun ny næring til fantasien, og utrolige var de eventyr, der blev bragt til torvs. Polarforskerens trasken rundt på polerne gjorde folk bange for at jorden skulle tippe over, og Nansen fik skyld for det dårlige vejr i København. Fantasien har altid løbet fra forskningen, men forskningen har flere gange halet ind på fantasien.

Det tekniske eventyr var nøgternt, det søgte viden. Stort var tilbuddet, og idéerne blomstrede. Industrien har givet os velstand, og samlebandet

har gjort bilen til hver mands eje. Men har det bundet vores fantasi og taget interessen for arbejdet fra os? Det, som ikke er endt som gammelt jern, er nu museumsgenstande. Hvis flyvningen havde haft samme tempo på som elektronikken, kunne man i dag have fløjet rundt om jorden på nogen få minutter for en ti'er.

Ikke en gang hundrede år efter at den første pilot lettede fra jorden, landede man på månen præcis, hvor man havde tænkt sig takket være edb. Men også rumstationer bliver gammelt jern, og det, der står på månen, er allerede skrot.

Ligesom det store univers er mikrouniverset, det lille univers, uendeligt. Og jo længere vi trænger ind i det, jo større bliver vor viden.

Mikroprocessoren vil revolutionere vor hverdag og åbne en helt ny verden for os.

Det, som for få år siden fyldte en hel fabrik, kan nu ligge i hjørnet af en tændstikæske og er i stand til at mangedoble vor hjernes kapacitet.

Den menneskelige hjerne er i sig selv en datamaskine, så raffineret, at teknikken endnu ikke har kunnet gøre den efter.

Ligesom TV kan edb kun ses og høres. Der er ingen lugt, der er ingen smag, og der er heller ingen følelse.

Det er ikke nok at kunne trykke på knapperne, vi må trykke på de rigtige - der må være rimelighed i det. Vi må forstå, hvad vi laver, og bevare vore sind.

Det er et par millioner år siden, vi slap den nederste gren, og vores hjerne har tilsyneladende ikke udviklet sig meget siden. Men vi er nysgerrige, og vi elsker at eksperimentere. Men tit bliver vi hængende i en enkelt god idé som hjulet, der har fået lov til at trække et endeløst net af motorveje, viadukter og skinner efter sig.

Hvad er det for et samfund, vi vil have?

Kan vi leve i et lykkeland og være kustoder i det forgangne, eller skal vi springe med på udviklingen? Er fremskridt traditioner på march?

Leder vi i fortiden efter fremtiden, eller bevæger vi os mod et samfund, hvor erfaringen ikke har nogen betydning?

Datamaskinen har det som tryllekunstnerens høje hat. Man kan ikke tage noget ud af den, hvis man ikke har puttet noget i den. Når vi så har bestemt, hvad vi vil putte i den, så kan klaveret skrive bogstaver, og skrivemaskinen spille klaver, og arkitekten kan digte huse.

Men bliver det ikke lidt koldt?

Kan vi bevare romantikken?

Og undgå misbrug?

Er vi bange for at blive brikker i et spil eller symboler i et system?

I skak er symbolerne faste, en bonde er en bonde, og en dronning er en dronning.

Vil vi miste vores identitet?

Vil vi gerne være anonyme?

Bliver vi et nummer i rækken?

Vil edb være i stand til at finde afgrunden i vores sjæl?

Hvad siger folk på gaden?

Hvad siger industrien?

Hvad siger kirken?

Hvad siger kunstneren?

Hvad siger filosofen?

— Enhver er sig selv nærmest.

Har han ret?

Den trojanske krig udbrød fordi prins Paris valgte skønheden frem for visdommen.

Hvad vælger de kommende generationer?

Det er dem, der tegner fremtiden.

Vi er midt i en overgang".

Jørgen Hansen

Filmen udlejes gennem Statens Filmcentral

NYE BØGER

Liber Utbildningsförlaget i Stockholm har udgivet tre bøger, der er udarbejdet af en arbejdsgruppe ved Fortbildningsavdelingen i Göteborg.

Bøgerne er udgivet for at hjælpe den matematik- eller samfundsfaglærer, der skal i gang med emner fra datalæren senest i skoleåret 82/83.

Den første bog er - Datalära, en handledning -. Denne bog er tænkt anvendt som et videreuddannelsesmateriale for lærere, og den indeholder f.eks. metodiske forslag til datalære, både for 9.-10. klasse og for videreuddannelse. Der findes også en litteraturliste over svenske materialer til faget.

Bogen giver i det hele taget en bred baggrund for de aktiviteter inden for datalære, man har tænkt sig i vort naboland.

De to andre udgivelser hedder - Programvara i Datalära -, henholdsvis - Elevhäfte - og - Lärarhandledning med programlistninger til ABC 80+. Listninger til Esselte 100 og 1000 samt til Sharp MZ-80 K fås også.

Her behandles emner som beregninger, simulering, sortering, datahåndtering, grafik, befolkningsprojektioner og husopvarmning. Elevmaterialerne og de tilhørende programmer er beregnet som støtte til den lærer, der ikke selv har mulighed for at udvikle materiale.

Materialet, der er udviklet under ledelse af Rolf Nilsson, virker tiltalende og må kunne gøre god fyldest i den datalæreundervisning, man har tænkt sig i Sverige (fra 12-14 timer og opefter).

Man kan få en god orientering om datalærens vilkår i Sverige, men der er også gode idéer at hente for den interesserede. Det er således værd at anskaffe disse bøger. Forlagets adresse er:

LIBER DISTRIBUTION

Läromedelsorder

162 89 Stockholm

tlf.: 08 - 739 91 00

COMAL -fiduser

Hjørring Gymnasium er med 4 terminaler tilsluttet en RC 7000, og for 3-4 elever er en stor del af fritiden i de sidste par år benyttet i selskab med disse. Vi mener selv efterhånden at have lært COMAL rev. 1.27 at kende ud og ind, og de nye ting, vi opdager hen ad vejen, går da også lidt ud over det, vi kan læse os til. Det var f. eks. et problem for os, at et program, der indeholdt en DEF-sætning, skulle stoppes, når funktionen skulle ændres (graf-program f. eks.). Det lykkedes os at finde følgende smutvej:

```
*CREATE "FNX",0
*LIST
0010 DIM F$(50)
0020 INPUT "F(X)=", F$
0030 OPEN FILE(0,3) "FNX"
0040 WRITE FILE(0) "1234567890 DEF FNF(X)=.F$. "1334102"
0050 CLOSE
0060 ENTER "FNX"
0070 REM PLADS TIL DEF-SÆTN.
```

Problemet ved dette er, at linienummeret i WRITE-sætningen naturligvis ikke ændres ved RENUMBER, og at der derved kan ske mærkelige ting. Det er dog til at overse, idet man f. eks. kunne anbringe definitions-proceduren øverst i sit program, hvor den ville være mindre "sårbar" mht. ændringer i linienumre ved rettelser i programmet. Denne fidus åbner også for mange andre muligheder. Bl. a. er det ved hjælp heraf muligt at execute'e et i en INPUT-sætning indtastet procedurenavn, idet DEF FNF(X)= blot skal udskiftes med EXEC. Der er mange andre anvendelsesmuligheder, men en af ulemperne er, at det tager fhv. lang tid at åbne, skrive i og derefter ENTER datafilen.

En anden interessant opdagelse er følgende:

```
*CREATE "FNX",0
*DIM LOOKUP$(240)
*LOOKUP"FNX"

*OPEN FILE(0,1)"FNX
*READ FILE(0)LOOKUP$
*PAGE=0
*LOOKUP$
NAME ATTRIBUTE LENGTH INDEX BLENGTH LBYTE LBLOCK SEQ
RECSIZE NO.REC RAN
FNX ..... X 00005 00132 00006 00002 00005 S
EJERN ..... X 00005 01160 00006 00508 00005 S
PSTR .....
```

Bemærk at der nu kommer "overskrift" på look-up'en (hvilket måske også var forventet, jfr. LOOKUP "\$LPT") Selvom LOOKUP ikke bruges i et program, kan FNX jo godt åbnes, og det giver f. eks. mulighed for automatisk at checke hvilke programmer eller filer, der står i subcataloget, om der er nogle "uregistrerede" filer eller få lookup-udskrifter af bestemte filer under programkørsel, osv. MEN der er visse skavanker! Ikke alle programmer kommer altid med, og til tider dukker pludselig et SYS-program op midt i det hele. Metoden er altså noget usikker, men kan benyttes.

Følgende lille streng-finesse er sikkert gjort af andre, som ville definere en strengvariabel "fra den gale ende":

```
0020 LET A$="....."
0030 PRINT A$
0040 LET A$="*", A$
0050 PRINT A$
0060 LET A$="4=" A$
0070 PRINT A$
*RUN
.....
*****
--4=4=4=4=
```

Denne er brugbar, hvis en streg-ud skrift, ramme el. lignende skal ændres.

Til slut en praktisk lille procedure, der kan bruges, hvis man har brug for at udvælge tilfældige elementer (tal) blandt N mulige samtidig med, at disse skal være forskellige.

(F. eks. MASTER-MIND):

```
0010 DIM A$(10),B$(4)
0020 LET A$="1234567890"
0030 FOR N=1 TO 4
0040 LET D=INT((7-N)*RND(0))+2
0050 LET B$(N)=A$(D)
0060 LET A$=A$(1,D-1),A$(D+1,10)
0070 NEXT N
```

Denne lille ting har vi gang på gang gjort brug af og haft brug for i mange forskellige situationer. Måske er disse finesser alle kendt for læseren, men så kunne dette indlæg vendes til et spørgsmål: Er der andre uofficielle genveje at skyde i COMAL?

Martin Boje Christensen 3. u.F.
Hjørring Gymnasium

***** ARBEJDSGRUPPE

Vi er en arbejdsgruppe, der sammen med Forlaget Systime A/S Herning, har startet med at udarbejde forslag til undervisningsmaterialer til datalære i Folkeskolen.

Vi ønsker at have et godt udgangspunkt for undervisningen, når datalære indenfor overskuelig fremtid bliver indført i folkeskolen.

For ikke at begrænse os til en lille kreds, og dermed arbejde med én snæver ideramme, forestiller vi os, at vi kunne udvide projektgruppen.

Såfremt du er interesseret i at arbejde i denne projektgruppe, enten fordi du har gode idéer, allerede udarbejdede materialer eller sukker efter ny udfordringer, så hører vi gerne fra dig.

Du kan henvende dig både til Forlaget Systime A/S, Solbakken 21, Gjellerup, 7400 Herning, tlf. (07) 11 91 87, eller til mig. Leif Pehrsson, Overlærer
Parkvænget 32
8310 Tranbjerg J.
Tlf. (06) 29 02 20

Programmeringskonkurrence

I anledning af at den internationale dataorganisa- tion IFIP holder sin 3. verdenskonference om anvendelse af datamater i undervisningen, har der været afholdt en konkurrence i programmering for skoleungdom. Den danske konkurrence var organiseret af Datalæreforeningen, der havde nedsat følgende dommerkomite:

Professor H.B. Hansen, Roskilde
Lærer Niels Tovgård, Odense
Lærer Erling Schmidt, Datalæreforeningen

Der indkom i alt 18 forslag, hvoraf de fleste drejede sig om matematik, spil eller fysik og kemi

Vinder blev den 18. årige Kaare Danielsen fra Holte Gymnasium.

Om hans program udtaler dommerkomiteen:

Kaare Danielsen har lavet et program der er imponerende i mere end én henseende. Hans beskrivelse af programmet og dets anvendelsesområde er et forbillede i klarhed og forståelighed. Selve programmet er velstruktureret og velkommenteret, og opfylder i det hele taget de professionelle krav man plejer at stille til en programmørs arbejdsresultat. Programmets almene anvendelighed er særlig imponerende. Det handler om undersøgelse af egenskaberne ved matematiske funktioner. Foruden at få tegnet funktionerne som en kurve, kan man også finde tangenthældning i frit valgte punkter, arealer af lukkede områder, asymptoter, maksimums- og minimumspunkter, o.s.v. Ved sit valg af beregningsmetoder til løsning af disse opgaver viser Kaare Danielsen et dybtgående kendskab til de numeriske metoder man må anvende når man arbejder med en datamaskine - et område der ellers ikke indgår med stor vægt i gymnasiets matematikpensum. Programmets omfattende anvendelighed kan måske bedst illustreres ved at nævne, at flertallet af de opgaver der stilles til studentereksamen i matematik, kan løses mere eller mindre fuldstændigt ved hjælp af Kaares program. Men dette er ikke alt. Hans egne eksempler på programmets anvendelse viser på meget overbevisende måde at programmet kan finde anvendelse inden for en lang række af gymnasiets fag.

Der er derfor ingen tvivl om at Kaare Danielsens program er værdig vinder af konkurrencen.

Vi har bedt Kaare om kort at beskrive funktionen af det vindende-program. Den følger her:

Kort beretning om funktionsanalyseprogrammet.

Programmet er et standardprogram til numerisk funktionsanalyse og graftegning af reelle funktioner af en variabel. Man definerer først den

funktion, man vil undersøge, hvilket kan gøres på forskellige måde. Man kan derefter beregne funktionsværdier.

Analysere funktionen numerisk. Hvis brugeren har defineret endepunkter, beregnes funktionsværdier og differentialkvotienter i disse, ellers undersøges skrå asymptoter og om $f(x) \rightarrow \pm \infty$ for $x \rightarrow \pm \infty$. Derefter undersøges funktionen for nulpunkter, ekstremumpunkter og lodrette asymptoter.

Tegne grafer for funktionen i et defineret interval på skærm eller printer. Programmet finder eventuelt selv y-området og sætter selv enhedsstreger på x og y-aksen.

Man kan ikke blot behandle funktionen selv, men også differentialkvotienten og stamfunktioner til denne. Programmet kan altså tegne differential og integralkurver og integrere. Programmet kan også løse ligninger og uligheder, f.eks. af formen

$$\int_{x_0}^x f(t) dt = a \quad \text{hvor } x_0 \text{ og } a \text{ er konstanter}$$

Programmet kan faktisk løse lidt under halvdelen af opgaverne i et studentereksamenssæt.

Jeg konstruerede programmet for ca. 1 1/2 år siden, fordi jeg ofte havde brug for at lave en graf over en matematik funktion. Samtidig indbyggede jeg funktionsanalysen. Idéen var én gang for alle at lave et standardprogram, der skulle kunne bruges til alt. Da jeg senere skulle lave et geografispeciale og havde brug for grafer i massevis, lavede jeg en version, der kunne anskueliggøre datamængder grafisk.

Programmet har adskillige gange været brugt i undervisningen, først og fremmest fordi det er lettere at lave en edb-graf end at tegne en selv. Simple programmer til f.eks. numerisk løsning af differentiallygninger bliver langt mere værdifulde, når man kan præsentere resultaterne grafisk. Programmet har således været brugt i matematikundervisningen, i fysikundervisningen til anskueliggørelse af forskellige fysiske fænomener, til fysikrapporter m.m.

ANVENDELSE AF RC 702

På Rismølleskolen i Randers anvender man en RC 702 i dansk- undervisningen i 5. læseklasse. Eleverne har anvendt systemet i 3 mdr., og en skolepsykologisk testning har for nylig vist, at eleverne har gjort dobbelt så store fremskridt som normalt indenfor de elementære færdigheder.

COMAL 80 - hvorfor og hvordan

af Børge Christensen

(Fortsat fra sidste nr.)

De versioner af COMAL-80, som hidtil er blevet implementeret, er alle udstyret med et simpelt, men ret effektivt fil-håndterings-system. Det lykkedes desværre ikke for arbejdsgruppen at nå frem til et endeligt forslag til et filsystem, skønt man i grunden var tæt på at være enige. På grundlag af allerede eksisterende idéer og notaterne om disse, findes der dog nu en faktisk standard, udtrykt i de allerede kørende systemer. Fælles for disse er, at man kan skrive såvel program- som datafiler i det ydre lager, og at dette kan ske enten i tekst-format (ASCII-format) eller i internt format (binær-format). Man synes også at være enige om, at

```
0100 LIST "filnavn" OF 20, ADDR OF 20, BY# OF 20
0120 REPEAT
0130 PRINT CHR$(147) //CLEAR SCREEN//
0040 FOR I=1 TO 10 DO PRINT //10 LIN. NED//
0050 PRINT "1 = OPRET NY LISTE"
0060 PRINT "2 = INDSET ADRESSAT"
0070 PRINT "3 = SLET ADRESSAT"
0080 PRINT "4 = SKALV LISTE"
0090 PRINT "5 = STOP"
0100 PRINT
0110 PRINT
0120 INPUT ">> ": JOB
0130 //
0140 CASE JOB OF
0150 WHEN 1
0160 EXEC OPRET
0170 WHEN 2
0180 EXEC INDSE1
0190 WHEN 3
0200 EXEC SLET
0210 WHEN 4
0220 EXEC LISTE
0230 WHEN 5
0240 STOP
0250 OTHERWISE
0260 //GENTAG JOBKALD//
0270 ENDCASE
0280 UNTIL FALGE
0290 //
0300 PROC OPRET
0310 OPEN 2,"ADRESSER",RANDOM 80
0320 INPUT "NAVN.....": NAVN$
0330 I:=1
0340 WHILE NAVN$<>" " DO
0350 INPUT "ADRESSE.....": ADR$
0360 INPUT "POSTNR. BY.....": BY$
0370 I:=I+1
0380 WRITE FILE 2,I: NAVN$,ADR$,BY$
0390 INPUT "NAVN.....": NAVN$
0400 ENDWHILE
0410 MAX:=I
0420 WRITE FILE 2,1: MAX
0430 CLOSE
0440 ENDPROC OPRET
0450 //
0460 PROC INDSE1
0470 //OVERLADES TIL LÆSEREN//
0480 ENDPROC INDSE1
0490 //
0500 PROC SLET
0510 //OVERLADES TIL LÆSEREN//
0520 ENDPROC SLET
0530 //
0540 PROC LISTE
0550 PRINT CHR$(147) //CLEAR SCREEN//
0560 PRINT "1 = SKERM"
0570 PRINT "2 = PRINTER"
0580 PRINT
0590 INPUT ">> ": ENHED
0600 IF ENHED=2 THEN SELECT OUTPUT "LP"
0610 OPEN 5,"ADRESSER",RANDOM 80
0620 READ FILE 5,1: MAX
0630 FOR I:=2 TO MAX DO
0640 READ FILE 5,1: NAVN$,ADR$,BY$
0650 PRINT NAVN$
0660 PRINT ADR$
0670 PRINT BY$
0680 FOR LN:=1 TO 3 DO PRINT //3 LIN.//
0690 IF ENHED=1 THEN INPUT BY$ //VENT PÅ RETURN//
0700 NEXT I
0710 CLOSE
0720 SELECT OUTPUT "DS"
0730 ENDPROC LISTE
0740 //
```

LIST "filnavn"

skriver et program ind i det ydre lager i tekst-format, mens

ENTER "filnavn"

omvendt læser et program ind fra det ydre lager, når dette program står skrevet i tekst-format. Hvis et program skrives i det ydre lager med kommandoen:

SAVE "filnavn"

bliver det repræsenteret i internt format og kan kun læses ind i arbejdsområdet med kommandoen:

LOAD "filnavn"

For brugeren er det tilstrækkeligt at vide, at SAVE og LOAD kommandoerne normalt giver anledning til hurtigere operationer end LIST og ENTER kommandoerne, men at LOAD kommandoen sletter det nuværende indhold af arbejdslageret, hvad ENTER kommandoen ikke gør. Hvis man skal sammenflette programmer, er man altså henvist til at bruge LIST/ENTER-parret.

CHAIN

Et program, som står skrevet i en fil, kan også kaldes op og startes af et andet program med sætningen:

CHAIN "filnavn"

Et program, der på denne måde skal bruges som „ydre procedure“, skal stå skrevet i internt format, dvs. det skal være skrevet ind med LOAD-kommandoen. I et enkelt af de systemer, der kører COMAL-80, nemlig CBM's, kan CHAIN også bruges som kommando.

LÆSNING OG SKRIVNING

Hvis man ønsker at skrive eller læse i en datafil, skal denne først åbnes med en sætning, som denne

OPEN 'numerisk udtryk','filnavn','måde'

Sætningen knytter en kanal med nummeret, givet ved 'numerisk udtryk' (en konstant, en variabel eller en formel) til den fil, der er angivet ved 'filnavn'. Filen kan åbnes i forskellige måder, nemlig

READ: læsning i en sekventiel fil
WRITE: skrivning i en sekventiel fil
APPEND: skrivning i forlængelse af allerede oprettet sekventiel fil.
RANDOM 'postlængde': skrivning eller læsning i en fil med direkte tilgang.

Det bemærkes, at postlængden altid skal angives efter nøgleordet RANDOM (antal bytes).

Eksempel:

```
OPEN 2,"MEDLEMMER", RANDOM 80
```

BEMÆRKNING. 'filnavn' kan i nogle versioner indeholde oplysninger om den ydre enhed, i hvilken filen skal skrives eller læses. I RC 702-COMAL-80 og CBM-COMAL-80 kan man endvidere forsyne filnavnet med et præfix, som bevirker, at evt. eksisterende udgave af filen slettes, når der skrives i den påny. I RC 702-COMAL-80 bevirker således sætningen:

```
OPEN 2,"NEW. MEDLEMMER", WRITE
```

at filen MEDLEMMER kan overskrives uden videre, og noget tilsvarende bevirker i CBM-COMAL-80 sætningen:

```
OPEN 2,"0: MEDLEMMER", WRITE
```

Der er planer om at ændre CBM-COMAL-80, så den tillader helt samme skrivemåde som RC 702-COMAL-80.

Ved skrivning i sekventielle filer kan man benytte sætningerne:

```
PRINT FILE 'numerisk udtryk': 'feltliste'  
WRITE FILE 'numerisk udtryk': 'feltliste'
```

PRINT FILE sætningen skriver data på tekstform, mens WRITE FILE sætningen skriver data på binær form. Det numeriske udtryk angiver kanalnummeret på den fil, man ønsker at skrive i. Den anførte 'feltliste' er en liste med de værdier (angivet ved konstanter, variable eller udtryk), som skal skrives i filen.

Ved skrivning i filer med direkte tilgang kan man benytte sætningen:

```
WRITE FILE 'numerisk udtryk', 'postnummer':  
      'feltliste'
```

Eftersom der er tale om filer med direkte tilgang, skal der naturligvis angives postnummer, og det sker med en (numerisk) konstant, en variabel, eller et udtryk.

Eksempler:

```
FOR I:=1 to MAXNR DO WRITE FILE 2:  
      N $ (I), K (I)  
FOR I:=1 TO MAXPOST DO  
  WRITE FILE 5, I: MEDLNR (I), NAVN $ (I),  
      AFD $ (I), LK (I)  
NEXT I  
Ved læsning i sekventielle filer benytter man sætningerne:
```

```
INPUT FILE 'numerisk udtryk': 'felter'  
READ FILE 'numerisk udtryk': 'felter'
```

Her er 'felter' en liste med variable, der får tildelt de indlæste værdier.

Ved læsning i filer med direkte tilgang benytter man sætningen:

```
READ FILE 'numerisk udtryk', 'postnummer':  
      'felter'
```

Eksempler:

```
I:=1  
WHILE NOT EOF (5) DO  
  READ FILE (5), I: MEDLEM $ (I)  
  I:= I+1  
ENDWHILE
```

```
READ FILE 2, 1: MAX  
FOR I:= 2 TO MAX DO  
  READ FILE 2, I: N $ (I), T $ (I)  
NEXT I
```

LUKNING

Når man er færdig med at læse eller skrive i en fil, lukker man den med en af sætningerne:

```
CLOSE  
CLOSE 'numerisk udtryk'
```

Når den første bruges, lukkes alle filer, der måtte være åbnet. Når den anden bruges, lukkes kun den fil, hvis kanalnummer er angivet ved det numeriske udtryk.

Eksempel: ;

```
CLOSE 3
```

SLETNING

I ICL-COMAL-, næsten 80" og i RC 702-COMAL-80 kan man slette filer af alle typer ved at bruge:

```
DELETE "filnavn"
```

der kan optræde både som kommando og som

sætning. I ICL-COMAL findes der nogle særlige regler for filnavnet, når det bruges i en DELETE-kommando eller -sætning.

AFSLUTNING

Ovenstående korte gennemgang gør naturligvis ikke krav på at være udtømmende, men skulle blot give en nogenlunde almen beskrivelse af de vigtigste fil-sætninger og -kommandoer i eksisterende versioner af COMAL-80. Læseren bør hæfte sig ved sætningernes opbygning mere end ved den omstændighed, at én version har et komma her og måske mangler et kolon hist. Som altid er der kun ét sted, man kan forvente - forhåbentligt - at få alle detaljer oplyst, nemlig i firmaernes manualer. Det har dog været meget opmuntrende for undertegnede, at man har kunnet enes så langt, som tilfældet er.

Som eksempel på brugen af filer i COMAL-80 er vedlagt et lille program, der kan bruges til skrivning af adresselister.

Den næste artikel om COMAL-80 hedder: Hvad der ikke kom med i denne omgang.

MANUSKRIPTER

Redaktøren begynder igen at mangle stof. Går du med én eller anden idé, har du prøvet noget indenfor faget datalære eller anvendelsen af edb, er der et af de tidligere indlæg, du godt kunne tænke dig at svare på eller er der andet du godt ville indvie dine kollegaer i, så fat pennen eller skrivemaskinen og send det ind til redaktøren. Du har hele sommeren til det, dead - line for næste nummer er først midt i august.

•••

HP-85

Af Peter B. Yde

HP-85 er en handy mikrodatamat fra Hewlett-Packard. I en enhed (kasse), der vejer 8 kg., er samlet en centralenhed, tastatur, skærm, printer og en kassetteoptager, der kan sammenlignes med en diskstation. Den er programmerbar i en udbygget BASIC-version med bl.a. en fin grafik og gode fejlretningmuligheder. Hele herligheden koster omkr. 30.000 kr. med moms.

HARDWARE

Datamatens dimensioner er 45 x 42 x 16 cm. Og som nævnt vejer det hele kun 8 kg.

Centralenheden indeholder 16 Kbytes, der kan udvides til 32 Kbytes. Efter mine målinger er maskinen en lille smule langsommere end visse andre mikrodatamater, men kun lidt. Den arbejder til gengæld med 12-cifrede tal. Cifferlængden kan dog afskæres til 5 v. hj. a. erklæringen SHORT.

Til forskellige tidspunkter har jeg haft forskellige eksemplarer af HP-85 (jeg har vist arbejdet med 6 maskiner), og jeg har haft lejlighed til at teste den grundigt. Ikke på noget tidspunkt har jeg været ude for, at en HP-85 udviste funktionsfejl. Overhovedet! I 1979 og 1980 har jeg testet en halv snes mikrodatamater, og kun HP-85 har jeg ikke på en eller anden måde haft problemer med, når jeg under problemer foruden funktionsfejl indregner fejlanvisninger i manualer og bøvler med at forbinde enheder og at få dem til at fungere. Da mine undersøgelser (der har været bragt i „Elektronik“ og „Populær Radio“) omfatter markedets vigtigste selvstændige, fuldt udbyggede mikrodatamater, vil det dog være rimeligt at nævne, at der er flere maskiner, jeg i denne forstand har meget lidt at udsætte på. F.eks. fandt jeg på ABC-80 blot en enkelt fejl.

HP-85's tastatur er velforsynet med taster. Der er taster for visse kommandoer som f.eks. NEW (der dog kaldes SCRATCH) og LOAD. Der er fine cursor- (markør-) betjeningsmuligheder. Anslaget af tasterne er behageligt. Maskinen er forsynet med mange tegn, også Æ, Ø og Å. Det kræver dog anvendelse af tre taster f.eks. at indtaste et Æ.

Skærmen er på 5 tommer diagonalt - dvs. lille. Den har 16 linjer á 32 tegn. Benyttes den til grafik, er der imidlertid 192 x 256 punkter på skærmen, hvilket er mange. Punkterne står pænt på skærmen, og det vil sige, at også tekst står meget klart. Man kan desuden ved betjening af en enkelt taste rulle de fire foregående skærmbilleder frem.

HEWLETT-PACKARD laver selv maskinerne fra ende til anden. Maskinens kassetteoptager er således en specialitet for firmaet. Den er praktisk talt lige så hurtig som en floppy diskstation. Det tager kun 4-5 sekunder at overføre et program fra hurtiglager til bånd eller omvendt. Desuden kan data lagres på båndet både sekventielt og som

random acces filer. Der er gode sikkerhedsmuligheder for lagring af programmer og data på bånd.

Hvert bånd er på omkring 200 Kbytes, hvilket også er sammenligneligt med floppy diskens størrelser. De koster imidlertid med moms omkr. 200 kr./bånd, dvs. noget mere end floppy diske. Desuden vil man også kunne savne to stationer, f.eks. hvis man ønsker at tage sikkerhedskopier.

Printeren er en termoskriver, hvis tegn er lavet ud fra en 5 x 7 matrix. Mens kassettebåndene ikke er helt lydløse, er printerens det til gengæld næsten. I programmer skrives der PRINT-sætninger, når udskrifter ønskes på printerens, og DISP-sætninger, når udskrifter ønskes på displayet (skærmen). På printerens er der 32 tegn/linje ligesom på skærmen.

Datamaten kan udbygges med forskellige enheder. Bl.a. fås en række forskellige ROM's (Read Only Memories), der let stikkes ind bag i maskinen. Der er f.eks. en ROM for matrixtegning, hvilket er nyttigt for den, der skal arbejde med omfattende systemer af lineære ligninger. Desuden findes der flere forskellige disktyper, som kan kobles på datamaten, både store og små. Der kan kobles en plotter til. Endelig kan diverse instrumenter og maskiner tilsluttes til maskinen via en serie I/O-porte.

SOFTWARE

Maskinens BASIC er omfattende og god. Den indeholder alle de gængse sætninger, kommandoer og funktioner og flere til. Nedenfor anføres nogle eksempler, der illustrerer, at der findes mere end det sædvanlige i sproget. Det skal understreges, at der kun er tale om eksempler.

Der er f.eks. toner på datamaten, hvilke fås frem v.h.j.a. BEEP-sætninger. Efter en idé fra et langt mere udbygget og bedre program lavet af Mogens Møller Nielsen i Sønderborg på PET-datamaten lavede jeg et lille simpelt program til min 6-årige datter, Mette. Det ser sådan ud:

```

10 PEN 1 @ CLEAR
20 SCALE -2,2,-3/2,3 2
30 DEG
40 ! URET TEGNES
50 FOR M=0 TO 360 STEP 6
60 MOVE SIN(M),COS(M)
70 IDRAW SIN(M)-50,COS(M)/50
90 IF M MOD 5 THEN 100
90 IDRAW SIN(M)/15,COS(M),15
100 NEXT M
110 FOR I=1 TO 12
120 MOVE 1.3*SIN(30*I),1.3*COS(30*I)
130 LABEL VAL#(I)
140 NEXT I
150 ALPHA
160 DISP "INPUT TID: HH MM"
170 INPUT T
180 GOSUB 1000
190 PAUSE
200 PEN -1
210 GOSUB 1000
220 PEN 1
230 GOTO 150
1000 REM TEGNING AF URETS VISERF
1010 MOVE 0,0
1020 H=30*IP(T)
1030 M=6*FP(T)*100
1040 DRAW .7*SIN(H+M*12),.7*COS(H+M*12)
1050 MOVE 0,0
1060 DRAW .9*SIN(M),.9*COS(M)
1070 RETURN

```

Det vælger et tilfældigt tal, som omsættes til et heltal mellem 1 og 10 (sætning 100). Maskinen giver nu lige så mange pip fra sig (sætning 120-140), som heltallet angiver. Det er så Mettes opgave at finde ud af, hvor mange pip, der var tale om og at skrive det tilsvarende ciffer ind på maskinen (sætning 160). Hun bliver belønnet med tre stigende toner (sætning 80), hvis hun svarer rigtigt, og straffet med et kort brum (sætning 180), hvis hun svarer galt. (Det første af tallene efter BEEP angiver tonehøjden, det andet tonens længde. Sætning 80 er en multipel-sætning, hvor adskillelsetegnet er det såk. at-tegn og ikke som på mange andre maskiner kolon.) Med mindre man er vældig stiv i sammenhængen mellem frekvenser og tonehøjder - og hvem er det? -, er det lettest at prøve sig frem, når man vil angive tonehøjde og tone-længde i BEEP'ene.

RND-funktionen synes at være god. Jeg har i hvert fald testet den med det program, H.B. Hansen angav i Datalære nr. 4, september 1979, p 8-10, og der var ikke det ringeste at udsætte på resultatet.

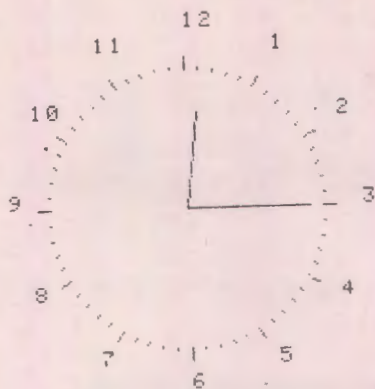
Grafikken er en stor styrke ved datamaten. Følg. program tegner f.eks. et ur, hvori brugeren kan specificere viserens placering:

```

10 PEN 1 @ CLEAR
20 SCALE -2,2,-3/2,3 2
30 DEG
40 ! URET TEGNES
50 FOR M=0 TO 360 STEP 6
60 MOVE SIN(M),COS(M)
70 IDRAW SIN(M)-50,COS(M)/50
90 IF M MOD 5 THEN 100
90 IDRAW SIN(M)/15,COS(M),15
100 NEXT M
110 FOR I=1 TO 12
120 MOVE 1.3*SIN(30*I),1.3*COS(30*I)
130 LABEL VAL#(I)
140 NEXT I
150 ALPHA
160 DISP "INPUT TID: HH MM"
170 INPUT T
180 GOSUB 1000
190 PAUSE
200 PEN -1
210 GOSUB 1000
220 PEN 1
230 GOTO 150
1000 REM TEGNING AF URETS VISERF
1010 MOVE 0,0
1020 H=30*IP(T)
1030 M=6*FP(T)*100
1040 DRAW .7*SIN(H+M*12),.7*COS(H+M*12)
1050 MOVE 0,0
1060 DRAW .9*SIN(M),.9*COS(M)
1070 RETURN

```

Svares der f.eks. „12.15” på spørgsmålet (sætning 160-170), fås følgende tegning (efter kopi-ering over på skriveren):



PEN-, MOVE-, IDRAW, DRAW og LABEL-sætningerne bruges til tegning, og der er væsentlig flere end disse. Lad os tage et par eksempler på anvendelser af disse sætninger: I sætning 60 føres „pennen” over i positionen (sin(M), cos(M)). I sætning 70 og 90 tegnes minut-mærker på skiven, men v.h.j.a. sætning 80, hvor der foregår en modulusregning, tegnes der kun lange mærker med 5-minutters intervaller. I sætning 130 skrives cifrene ind.

Der er ikke grund til at trætte med flere eksempler på sætningernes virkemåde. Tag blot programmets fremmedartethed som udtryk for, at maskinens BASIC er særdeles udbygget!

En anden stærk side ved HP-85 er dens fejlbehandling. Laver man grammatiske fejl under indtastningen af programmet, kommer der en fejludskrift, og cursoren går op under det sted, hvor fejlen formodes at være (og i reglen også er). Resten af fejlene findes let under kørselen af programmet. Der er 92 fejlmeldinger. Skulle det stadigvæk knibe, kan man betjene sig af TRACE-ordren, der sikrer, at man følger sætningerne en for en, efterhånden som programmet afvikles. Programmet pyntes let op v.h.j.a. DELETE- og RENUMBER- ordrerne.

Også sætningerne til anvendelse af disken er i orden, og der findes den fortræffelige CHAIN-sætning, hvormed programmer kan bringes til at køre efter hinanden uden brugerens indgriben, på maskinen.

Mens maskinen har brugerdefinérbare taster ligesom HP-lommeregnerne, savnes der GET-sætninger, hvormed et enkelt tegn kan tages ind uden anvendelse af RETURN-tasten. Endvidere er BASIC'en ikke struktureret som COMAL. Det nærmeste, den kommer i den retning, er i form af IF . . . THEN . . . ELSE-sætninger.

Der findes forskellige programpakker til maskinen, bl. a. til lineær programmering, til statistik og til BASIC-træning.

Dokumentationen er pletfri. Den er letlæse-

lig, fejlfri, mangelfri og har perfekte oversigter og registre.

HELHEDSVURDERING

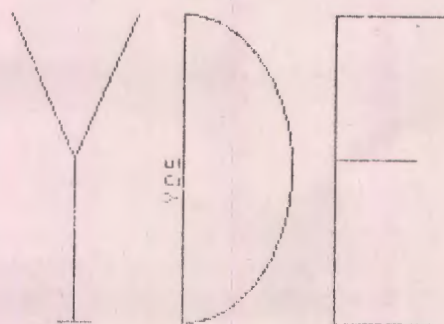
Der er tale om en særdeles handy maskine med en rimelig kapacitet, et hurtigt bånd og en rimelig god skriver, alt i én enhed.

I kraft af en god BASIC, en god fejlbehandling og god dokumentation er den meget behagelig at arbejde med. Den arbejder fejlfrit.

Blandt markedets mikrodatamater er HP-85 i skrivende stund (januar 1981) den, der imponerer mest. For undervisere er især prisen (ca. 30.000 kr.) og skærmens størrelse (5 tommer) vægtige ulemper ved maskinen. For ingeniører uden de store midler er den fin-fin.



Med venlig HP-85 grafisk hilsen



Peter B. Yde

Vedr. manuskripter til Datalære

Maskinskrevne manuskripter er velsete. Håndskrevne manuskripter må være letlæselige, og kun den ene side af papiret bør forsynes med tekst.

Ønske om bibeholdelse af afvigelser fra den "gængse" retskrivning og/eller tegnsætning bedes angivet på manuskriptet.

RC * INFO

RC-NYHEDSORGANET FOR RC-BRUGERE

RC 701 KAN OPDATERES

Da de første RC 700 microdatamatsystemer blev sendt på markedet, havde man endnu ikke fundet på PICCOLO navnet. Men navnet er ikke den eneste ting, der skiller. F. eks. er portnumrene forskellige på RC 701 og på RC 702.

Det er selvfølgelig uhensigtsmæssigt at have to forskellige sæt portnumre, specielt hvis man har RC 701 og RC 702 stående ved siden af hinanden.

Derfor har de pionérkunder, der var først ude og kom i besiddelse af RC 701, fået et tilbud om en mindre ombygning af deres RC 701, så den bringes i overensstemmelse med RC 702.

RINGKØBING AMT VÆLGER RC 700

Allerede for flere år siden begyndte man i Ringkøbing Amt at interessere sig for den kommende udvikling indenfor undervisningssektorens brug af datamatisk udstyr, dvs. specielt MIKRODATAMATSYSTEMER.

Amtet nedsatte en arbejdsgruppe, som fik til opgave at fremlægge et forslag til, hvordan denne udvikling skulle ske indenfor Ringkøbing Amt. Arbejdsgruppen indså hurtigt, at man måtte standardisere skolerne indkøb af udstyr, idet udvikling og udveksling af undervisningsprogrammer ellers ville blive en uoverkommelig opgave.

Amtscentralen i Herning skal fungere som programbibliotek, og skolerne vil her kunne rekvirere programmer på samme måde, som man låner bøger, dias, videobånd m.v. I første omgang distribueres programmerne på disketter, og når brugernes antal bliver tilstrækkelig stort, vil man etablere et egentligt datacenter, hvorfra programmer kan transmitteres via telefonnettet. Dette datacenter skal desuden løse en række administrative opgaver på amtscentralen, f.eks. udlånsstyring.

Da programudveksling ikke umiddelbart er muligt mellem forskellige maskinfabrikanter, besluttede man sig for kun at satse på et Mikrodatamatystem.

I 1980 iværksatte man en kravspecifikation, idet man skulle sikre sig, at det valgte udstyr kunne opfylde skolernes krav. Dette arbejde blev foretaget af en mindre arbejdsgruppe med repræsentanter fra alle de skoleformer, som amtscentralen har blandt sine brugere. Maskinudvælgelsen sluttede med, at man i slutningen af 1980 anskaffede nogle maskiner, for at kunne afprøve dem i praksis på forskellige skoler. Resultatet forelå i april 1981, og der var ingen tvivl om, at det system man ville vælge var RC 700 Mikrodatamatystemet fra A/S Regnecentralen af 1979.

RC 700 vil således være den eneste datamaskine, som amtscentralen vil tilbyde programmene. Ligeledes vil Ringkøbing Amt kunne købe RC 700 til egne skoler, f.eks. gymnasierne, ligesom man ved, at de kommunale folkeskoler vil gøre det samme. Der er ingen tvivl om, at aktiviteterne i Ringkøbing Amt vil få stor betydning for udvikling i resten af landet. Ringkøbing Amt har allerede etableret kontakt til 5-byerne, dvs. skolevæserne i Ålborg, Århus, Odense, Randers og Esbjerg, ligesom flere andre amter vil vælge samme løsning som Ringkøbing Amt.

Yderligere information om aktiviteterne i Ringkøbing Amt, kan fås ved henvendelse til Bent Kristensen, Amtscentralen i Herning.

COMAL 80 PÅ PICCOLO

Distributionen af COMAL-80 til RC 700 brugerne starter nu, og alle vil få tilsendt nærmere oplysning om procedurerne i denne forbindelse direkte. For tiden afprøves og afluses COMAL-80 for højtryk, så den

kan have en høj „færdighedsgrad”, når den kommer ud til brugerne..

Regnecentralens COMAL-80 indeholder selvfølgelig alle de ting, der er med i den såkaldte „COMAL-80-kerne”, som blev defineret af en arbejdsgruppe med bl.a. Børge Christensen, H.B. Hansen, Tom Østerby og en række repræsentanter fra forskellige firmaer. Men ud over COMAL-80-kernen indeholder PICCOLO-versionen af COMAL-80 en række udvidelser, hvoraf en del er inspireret af og bygger på erfaringerne fra de mange RC 7000 installationer landet over.

Bla. vil en RC 7000 bruger kunne nikke gende til behandlingen af filer, hvor PICCOLO har overtaget det meste med få syntaktiske ændringer. Man beholder således alle tre typer af filer: sekventielle filer i ASCII eller binært samt random filer.

Man bruger også stadig PRINT FILE og INPUT FILE ved ASCII filerne og READ FILE og WRITE FILE ved de binære.

CLOSE, CREATE og DELETE genfindes også, og har man haft svært ved at huske på talkoderne i OPEN FILE sætningen på RC 7000 kan man glæde sig, for fremover skriver man simpelthen READ, WRITE, APPEND eller RANDOM i OPEN-sætningen alt efter hvad filen skal bruges til.

Blandt kommandoerne er der også mange gamle bekendte som f.eks. ENTER, LIST, LOAD, SAVE, SIZE o.s.v.

Forbedringer i forhold til RC 7000 er der også en del af, bl.a. kan nævnes, at variabelnavne kan være på 16 tegn, og at der regnes med 13 cifres nøjagtighed i et talområde fra 10E-128 til 10E-127.

Så der er noget at glæde sig til.

NYE SKRIFTER

Det i sidste nummer af RC-INFO omtalte supplement til RC 700 brugermanual, RCSL Nr.: 42-i1599 er kommet.

I dette supplement beskrives bl.a. de nye funktioner til kontrol af tastatur, skærm og terminalport. Endvidere behandles semi-grafikken m.v.



Introduction to CP/M on RC 700
RCSL Nr.: 42-i1622

samt

An introduction to CP/M Features and Facilities

RCSL Nr.: 42-i1610



A Survey of COMAL 80
for the RC 700 Microcomputer
- PICCOLO -

RCSL Nr.: 42-i1714

En oversigt over COMAL 80 på RC 700.

PICCOLO-PLOT

Det er altid en fordel at kunne tegne godt - og det kan PICCOLO nu!

Regnecentralen har fundet frem til en kvalitets plotter, der uden videre kan tilsluttes RC 700, og der er udviklet standard plotter routiner i både COMAL og PASCAL.

Den plotter, man har fundet frem til, er en intelligent plotter, der kan tage papirformater op til A3. Den kan mere end bare tegne, bl.a. kan den skrive tekster i forskellig størrelse og i alle vinkler.

Den nye plotter kan også have interesse for RC 7000 brugere, idet den også kan tilsluttes her.

Yderligere oplysninger om plotteren kan fås ved henvendelse til Thorkild Maetoft.

BRUGERMØDER

RC 7000/700 brugergruppen afholder normalt sit årlige møde i første kvartal. Brugergruppens forretningsudvalg har imidlertid besluttet at mødet udsættes til efteråret. Der vil senere blive sendt invitation til alle RC 7000/700 brugerne.

COMAL-PROBLEMLØSNING OG PROGRAMMERING

I sidste nummer af RC-INFO blev det nævnt, at RC 7000 brugergruppen havde fået stillet en række noter til rådighed af Børge Christensen, Tønder. Da disse noter senere er udgivet som bog, har vi besluttet, af hensyn til

forlagets rettigheder, ikke at distribuere disse noter, som oprindeligt planlagt. I stedet henvises man til selv at købe bøgerne: Problemløsning og programmering 1 og 2, fra forlaget Bogika ApS.

MULTI-KARAKTERSÆT I PASCAL

I PASCAL er der indført en forbedring, som gør det muligt for brugerne selv at vælge hvilket karaktersæt, de vil arbejde med. Hvis man efter opstart af loader-disketten trykker på ESC kommer følgende linie til syne:

F (ormat, S (ystem, C (onversion, Q (uit?)

Trykker man så på F,S,C eller Q udføres den tilsvarende funktion.

Ny er conversion, og vælger man den, kommer følgende udskrift:

- 1: Danish
- 2: Swedish
- 3: German
- 4: UK-ASCII
- 5: US-ASCII
- 6: Librarian

Conversion table number:

Man kan nu vælge mellem de nævnte alfabeter, og de fleste giver jo sig selv. Speciel er LIBRARIAN, der er et tegnsæt, udviklet til forskningsbibliotekernes bibliotekssystem, der benytter PICCOLO opkoblet mod RC 8000.

Når det i det hele taget er muligt for PICCOLO at bruge flere forskellige alfabeter, skyldes det, at den integrerede kredse, der indeholder bogstavernes udseende, har flere pladser end dem, der bruges til det sædvanlige alfabet og til de semigrafiske tegn. Gennem en konverteringstabel kan man få de „ekstra” tegn frem, og på disse pladser har man så gemt billederne af de udenlandske tegn og øvrige specialtegn.

OM AT SÆLGE SAND TIL SAHARA

En PICCOLO er fremstillet af komponenter fra mange forskellige fabrikanter i mange lande. Disse komponenter bliver indkøbt i partier af forskellig størrelse, og til alle de komponenter, der anvendes, findes der flere uafhængige fabrikanter og leverandører. Ved at vælge sådanne komponenter til fremstillingen af PICCOLO, kan man være sikker på,

at kunne få fortsatte leverancer også selv om en enkelt af fabrikanterne holdt op med at lave en af de integrerede kredse, der er med i elektronikken i maskinen. Desuden kan man vælge det billigste tilbud, og det vil altid være en fordel, at der er flere leverandører, der konkurrerer om at sælge de samme kredse.

Kikker man derfor ind i flere forskellige RC 702'ere, vil det let kunne ske, at det ikke er præcis de samme integrerede kredse, der sidder i maskinerne. Mange af disse integrerede kredse er iøvrigt mærket med oprindelsesland, og det er jo ikke nogen hemmelighed, at det ofte er navne på lande i Sydøst-asien, der figurerer. Bl.a. kan man støde på Korea, Malaysia og Philipinerne.

Det sidste navn er særlig interessant, for det er muligt, at nogle integrerede kredse er „vendt hjem”. Regnecentralen har nemlig fået en forhandler af PICCOLO på Philipinerne, og der er allerede solgt nogle stykker derude. Der er altså tale om, at billige integrerede kredse sendes til Danmark, bliver samlet til en microdatamat, der så minstanten kan sælges tilbage til et af oprindelseslandene for komponenterne. Det ender med at PICCOLO også sælges i f.eks. Japan og USA.

NYT UDVIDET TASTATUR

Det er nu muligt at få et udvidet tastatur til RC 700. Tastaturet har samme højde og dybde som det oprindelige tastatur, men det er blevet ca. 14 cm. bredere, så der er blevet plads til en selvstændig talblok.

Denne talblok indeholder foruden selve cifrene også taster til return, decimaltegn, mellemrum og minustegn.

Men det er ikke bare et almindeligt tastatur med talblok. Der er nemlig blevet plads til 8 store og 5 små funktionstaster, der hver sender specielle koder til PICCOLO, som så via programmet kan udløse bestemte funktioner. Man har altså selv mulighed for gennem sit program at tillægge hver af de ialt 13 taster en bestemt betydning. Det er således nærmest op til ens egen fantasi, at finde på nyttige anvendelser af disse funktionstaster. Eksempelvis kunne man ved større programsystemer lave dem således, at hver af de store funktionstaster valgte bestemte funktioner i programmet. Det kunne for et kartoteksprogram være funktioner som indsættelse af nye poster, rettelse af poster, fjernelse

af poster, udskrift af poster o.s.v. For hver af disse funktioner kunne så de enkelte små funktionstaster bruges til at vælge det element i posten, der skulle rettes i. Havde man valgt udskrift kunne de samme små taster have en anden betydning, f.eks. udskrift i forskellige formater, sorteret eller ikke sorteret.

Det udvidede tastatur er vel nok først og fremmest lavet efter krav fra den private sektor, der bruger PICCOLO til en lang række systemer til administration, lagerstyring, regnskab og bogholderi m.m. Men det skal nok vise sig, at også undervisningssektoren „falder” for de mange muligheder det nye tastatur tilbyder.

(PS. normalt tastatur: RC721, udvidet: RC722).

TEKNISK SERVICE

Hidtil har alle reparationer af RC 700 foregået på et centralt værksted i Ballerup, men nu bliver Regnecentralens øvrige afdelinger inddraget i servicevirksomheden omkring PICCOLO.

Samtidig vil der blive tilbudt 3 forskellige muligheder for teknisk service, således at de forskellige slags PICCOLO brugere kan væl-

ge den, der passer til deres brug af maskinen og deres økonomi bedst.

De tre muligheder vil fungere efter følgende principper.

1. Reparation efter regning.

Man vil kunne få sin PICCOLO repareret på alle Regnecentralens afdelinger med teknisk service. Man vil kunne bringe eller sende maskinen, henholdsvis hente eller få sendt den reparerede maskine alt efter aftaler.

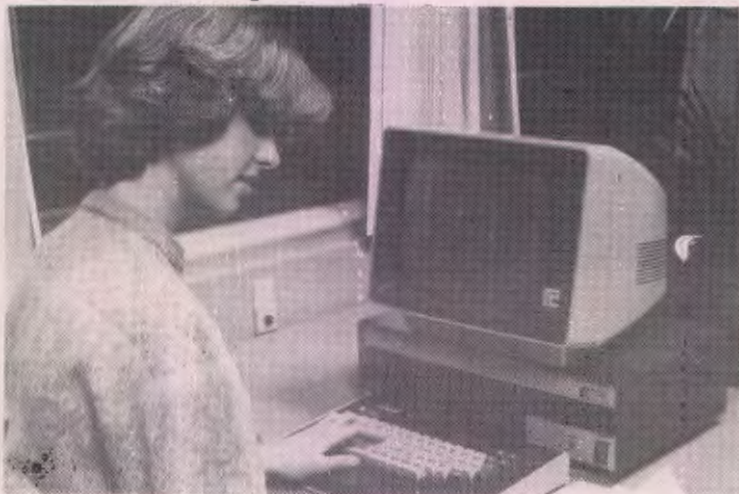
2. Teknisk servicekontrakt.

Ved fejl tilkaldes en tekniker, der reparerer maskinen på stedet eller udskifter defekte dele. Der betales en fast månedlig afgift samt et tilkaldgebyr pr. tilkald, mens selve reparationen er gratis.

Reparationskontrakt.

Ved fejl bringes/sendes maskinen til et teknisk service center, og der repareres den gratis.

Med inddragelsen af Regnecentralens landsdækkende net af teknisk service centre og med muligheden for valg mellem de tre forskellige serviceformer, håber man at have etableret en tilfredsstillende ordning for PICCOLO brugerne.



REGNECENTRALEN
af 1979

LAUTRUPBJERG 1 — 2750 BALLERUP

Telefon: 02 65 80 00

Dansk Data Elektronik

"TAL EDB MED **dde!**"

Hvorfor ikke en professionel datamat til skolebrug?



Spørg efter
SPC/1
til
under
visning

dansk
leverandør
til danske
skoler

dde

Dansk Data
Elektronik ApS
Herlev
Hovedgade 207,
2730 Herlev
02/84 5011