

VPCG – Hvad er det?

af Jørgen E. G. Nielsen, Orfeus

For mange brugere af EDB er det et dagligt irritationsmoment og en stadig kilde til undren, hvorfor datamaskiner ikke fungerer på samme vis som pladespillere. Hvorfor kan disketter og programmer til en maskintype ikke direkte benyttes på en anden? Det er jo ikke nødvendigt med forskellige pladespillere til de forskellige pladeselskabers produkter.

Denne artikel beskriver nogle af de tekniske grunde, der ligger bag, og anviser en måde at over-

vinde mange af problemerne på, når det drejer sig om maskintyperne Partner og PICCOLINE: ved at benytte programmet VPCG.

Betegnelsen VPCG står for Virtuel Personlig Computer med Grafik. Som navnet antyder, kan programmet få Partner/PICCOLINE til at fungere som en PC, således at de også kan afvikle programmer udviklet til denne maskintype, og vel at mærke også programmer, der udnytter grafik.

I skolesektoren er der i dag mange PICCOLINER, og det har længe været et udbredt ønske at man udover de mange undervisningsprogrammer der findes til denne maskine, også kunne køre det store udbud af DOS-programmer, der findes i dag. Derfor, og for at fremme standardiseringen på skolemarkedet, har Orfeus taget initiativ til at færdiggøre og videreudvikle det VPC-program, der blev distribueret fra RC International. En af de alvorligste begrænsninger ved dette program er dets manglende evne til at kunne håndtere grafikprogrammer, og udnyttelsen af grafik er i dag et af de vigtigste og oftest forekommende virkemidler – især i pædagogiske programmer. VPCG er en gevinst for brugerne, idet man får forøget udbuddet af mulige programmer, når man på Partner/PICCOLINE også kan afvikle allerede eksisterende DOS-programmer. Men samtidig er det en gevinst for programudbydere. Tidligere måtte de lave flere versioner af samme program, for at få det til at køre på de forskellige maskintyper. Det gav især problemer for producenter af undervisningsprogrammer, da skolemarkedet er for lille, til at bære disse ekstra udgifter. Med VPCG er der opnået en vis standard for skolesystemer, idet programmer udviklet med generelle programmeringsværktøjer nu kan køre på to af de mest udbredte maskintyper i skolerne. I den forbindelse skal også nævnes, at Orfeus har en BGI-driver til TurboPascal under udarbejdelse. Med denne driver vil TurboPascal også kunne benytte grafik på RC-maskinerne. Det betyder, at programmer skrevet i dette sprog, fremover kan afvikles på begge maskintyper endda uden brug af VPCG!. Dette modul vil vi vende tilbage til i en senere artikel i PICCOLINIEN.

Hvorfor er PICCOLINE og IBM PC/XT/AT ikke kompatible?

Hardwaren på en Partner/PICCOLINE afviger på næsten alle punkter fra hardwaren på en IBM-maskine. Selv om de inderst inde består af de samme dele, er opbygningen grebet helt forskellig an. Af

tekniske forskelle kan bl.a. nævnes, at Partner/PICCOLINE ingen EGA-controller har, de serielle porte og skærmbufferne ligger på andre absolute adresser, mens IBM omvendt arbejder med fast adresserede skærmlagre (640Kb-grænsen), som man er kommet ud over på Partner/PICCOLINE.

På den anden side ligner styresystemet C-DOS til Partner/PICCOLINE på mange punkter styresystemet (MS/PC-)DOS til de IBM-kompatible maskiner. Mange af de funktioner, der findes i DOS, understøttes af C-DOS. Så en del DOS-programmer vil uden videre kunne køre under styresystemet C-DOS på Partner/PICCOLINE.

Problemerne opstår med programmer, der er skrevet til at »gå udenom« styresystemet. Og netop i to bestemte situationer benyttes denne mulighed ofte.

1. Når man ønsker, at et program skal foretage hurtige udskrifter eller ændringer af tekst på skærmen, kan man ikke klare sig med standardmetoder og systemkald – det går for langsomt. I stedet skrives direkte i lageret på den adresse, hvor skærmlageret findes.
2. Ved grafik-programmering kan man benytte sig af maskinens eget system til at behandle input/output. På IBM kaldes det BIOS (Basic Input/Output System). Men det er også en langsom og affære, så ligesom ved tekst benyttes oftest direkte tilgang til lageret, hvor grafikskærmlageret ligger. (Og det er ikke det samme område som tekstskærmen).

Forsøger man at køre et sådant program på en Partner/PICCOLINE, går det galt.

For det første ligger skærmbufferne et helt andet sted, så programmer, der skriver direkte til skærmen, vil på Partner/PICCOLINE – i heldigste fald – skrive »ud i den blå luft«. Eller mere uheldigt overskrive væsentlige ting i lageret – program, data eller styresystem.

For det andet har Partner/PICCOLINE ikke nogen PC-kompatibel BIOS, så programmer, der benytter denne, vil ofte stoppe med interupt-fejl.

Hvordan klarer VPCG dette?

For at omgå problemet med programmer, der skriver direkte til skærmen, reserverer VPCG det område af lageret, der svarer til IBM-skærmen. Under afviklingen af et program »kopierer« VPCG indholdet af dette område op i Partner/PICCOLINE's skærmlager.

Dette foregår flere gange i sekundet, mens programmet kører. Så alt, hvad der skrives i »IBM-skærmens« område, havner også på Partner/PICCOLINE-skærmen. Det er dog ikke bare en simpel kopiering VPCG foretager. For det første er skærmmatrixerne for tekst forskellige for de to maskintyper. For at tegnene kan vises korrekt på skærmen, skal der derfor foretages en konvertering undervejs.

For det andet er grafikbilleder organiseret på forskellig måde, så de skal også konverteres, før de vil fremtræde ens.

Med hensyn til BIOS-funktioner understøtter C-DOS en del – her iblandt skærmfunktionerne (interupt 10). Det er så VPCG's opgave at genkende og emulere de resterende BIOS-kald.

Disse to funktioner – skærmkopiering/konvertering og BIOS-emulering – er hjørnestene i VPCG, som på denne måde kan få selv de krævede grafikprogrammer til at køre på en maskine, der er fundamentalt anderledes end den, de fra starten var beregnet til.

For at en sådan emulering skal kunne fungere tilfredsstillende, er det for det første nødvendigt med en hurtig maskine – da en del af tiden bliver brugt til skærmmkopiering – og det er Partner/PICCOLINE. For det andet er det nødvendigt med et Multitasking styresystem som C-DOS, idet VPCG-funktionerne skal foregå sideløbende med det kørende program. Derfor er det heller ikke på samme måde muligt at lave det omvendte program, der kan køre Partner/PICCOLINE-programmer på en IBM-maskine, idet denne jo anvender MS-DOS, der ikke indeholder multi-tasking.

VPCG distribueres bl.a. via EDB-HANDELEN for Skoler og Uddannelse tlf. 86 16 98 11 og koster 500 kr. excl. moms. for en enkeltlicens, og 1200,- excl. moms for en skolelicens.



Som sagt fungerer VPCG på DOS-programmer, der "opfører sig pænt" - og det er det som bekendt ikke alt i denne verden, der gør. Nedenfor er anført nogle af de større programsystemer der helt eller næsten gør det, og derfor kan fungere med VPCG:

UniComal v. 2.20 og 3.00
TurboPascal v. 5.0 og 5.5
TurboProlog v.1.1
WordPerfect 5.0
PCWRITE
DSI-TEKST v.4.21
ProDesign II v. 2.5
DesignCAD 3D v.1.0
PCTools
Btrieve 5.00b
CALIS

Derudover virker VPCG (selvfølgelig) på oversatte programmer udviklet i de nævnte programmeringssprog - og derudover på en stor del compilede programmer af anden oprindelse.

Orfeus

Orfeus står for Organisation til fremme af undervisningsprogrammel til skolerne.

Orfeus er en selvejende institution, stiftet af Kommunernes Landsforening, Amtsrådsforeningen i Danmark, Københavns Kommune, Frederiksberg Kommune og Kommunedata i/s.

Orfeus er en landsdækkende organisation, der skal sørge for, at skolerne forsynes med flere og bedre edb-programmer til undervisningsbrug, og at programmerne bliver brugt fornuftigt.

Orfeus har i det lille års tid, det har eksisteret, koncentreret sin aktivitet indefor følgende hovedområder:

1. Standardiserende aktiviteter.

Programmet VPCG er et eksempel på dette område.

Et andet er Skoleprogrammørernes Forum, hvor Orfeus samler undervisere, programmører og designere af edb-baseret undervisningsmateriale. Hensigten er at udveksle erfaringer og nyheder, koordinere udviklingsarbejde og pege på standarder for fremtiden.

Endelig har Orfeus finansieret udviklingen af programmet DINDES, der gør det muligt at præsentere SQL-baserede oplysninger på grafisk form i Videotext-format.

2. Markedsfremmende aktiviteter.

Med projektet '4. klasse bruger edb' vil Orfeus undersøge, om man med en forholdsvis massiv indsats kan sikre en gennemslagskraft og få en bred gruppe lærere så langt i anvendelsen af edb, at det er muligt at forudsige noget om brugen af edb i fremtidens skole. I forsøget deltager ti skoler, hvor man på 4. klassetrin indfører edb som et naturligt redskab i alle fag.

Et tilsvarende projekt på gymnasialt niveau er 'Erhvervsøkonomi'. Formålet er en reel, effektiv undersøgelse af, hvordan edb kan bruges i det nye Erhvervsøkonomi-fag i gymnasiet.

3. Nye muligheder i undervisningen.

I et samlet projekt, der forsøger at vise nye veje for anvendelsen af edb i skolen udvikler Orfeus to programmer: 3DMODEL, et tredimensionelt tegneprogram til PICCOLINE og MFCADCAM, et design-program, hvormed man kan tegne et objekt på dataskærmen. Man kan derefter styre et apparat, MikroFabrikken, så tegningen udskæres.

Til brug i faget kemi udarbejder Orfeus et program med hvilket, man bl.a. kan opbygge molekyler i tre dimensioner.

Orfeus indgår også i et større projekt vedrørende digital billedbehandling, hvor der udarbejdes et programkompleks, der kan behandle digitale billeder, f.eks. satellitbilleder. Som et eksempel på anvendelsen af programmet udarbejdes et undervisningsmateriale omhandlende brugen af satellitbilleder i

geografiundervisningen.

Ligeledes til brug i geografi udarbejder Orfeus i samarbejde med Geologisk Museum et hjælpeværktøj til visualisering af emnerne vulkaner, jordskælv og kontinentalændringer.

4. Opsamling.

Orfeus opsamler og færdiggør undervisningsprogrammer, så de fremstår som professionelle, markedsklare produkter. Her kan nævnes et samlet undervisningsmateriale om datamaskinens fundamentale virkemåde, et Concept-keyboard-program til PICCOLINE og undervisningsprogrammer til elever med forskellige typer handicap. Endelig har Orfeus versioneret KONGEO-programmet (KONstruktions GEOMETri) af Viggo Sadolin, så det kan afvikles på Partner/PICCOLINE.

5. Norden og EF.

Orfeus samarbejder med såvel Norden som EF om versionering, udvikling og standardisering.



Programmering med objekter - med Turbo Pascal 5.5

Af Peter Ferdinand, CPI

Programmering med objekter er ikke noget nyt. Programmeringssprog baseret på objektbegrebet har eksisteret længe. Man kan nævne Simula, Smalltalk 80 og C++, men mange andre findes tilgængelige også på PC-datamater. Det nye er, at der er opstået et behov for anvendelsen af objekter ved programmering, dels fordi programmerne de seneste år er blevet meget større og mere komplicerede og dels fordi behovet for grafiske operationer har vist sig umætteligt. Erfarne programmører har opdaget, at det de foretog sig var en form for objektprogrammering, uden at de dog havde det rigtige værktøj til deres rådighed. Derfor er det en helt naturlig ting, at sproget Pascal udvides til også at indeholde behandling af objekter. Det samme er sket med programmeringssproget C, hvor udvidelsen kaldes C++. Det eneste man kan beklage i denne forbindelse er, at der ikke findes en egentlig standard for, hvorledes objekter implementeres i Pascal. Man kan håbe på, at Turbo Pascal 5.5 er så udbredt, at det vil sætte sin egen standard. I Turbo Pascal er der benyttet en ad hoc løsning, som hviler på både C++ og ObjectPascal fra Macintosh datamater. Der anvendes tre principper ved objektprogrammering i Turbo Pascal:

- indkapsling
- arv
- polymorfisme.

Der findes andre og mere vidtgående principper i de ægte objektprogrammeringssprog som Smalltalk 80, men det vil det være for omfattende at komme ind på i denne sammenhæng. Man skal ikke lade sig forskrække af, at der ved objektprogrammering benyttes andre begreber, end dem man er vant til fra Pascal. F.eks. kaldes en

procedure for en metode. Begreberne har efter min mening klare fordele, fordi de benyttes stringent og på en lidt anden måde end i procedurale programmeringssprog.

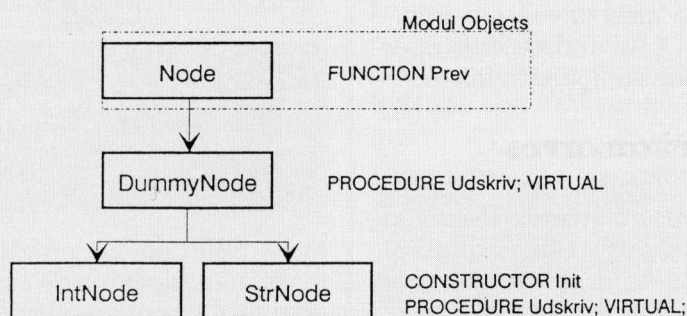
Indkapsling af program og data

Et objekt er kort fortalt en speciel poststruktur, som både indeholder de sædvanlige post-erklæringer og de operationer, som kan udføres på dem. Data- og programstruktur holdes dermed tættere sammen, hvilket både letter formaliseringen af programmerne og vedligeholdelsen. Man opdager hurtigt, at hele ens program kan beskrives ved hjælp af objekterne. Hvis man definerer en datastruktur, som beskriver en cirkel, skal man samtidigt erklære de operationer, der skal udføres på den. Den traditionelle datastruktur ser således ud:

```
EnCirkel = RECORD
  Radius: INTEGER;
  - Placering: EtPunkt;
END;
```

De operationer som udføres på en cirkel er:

```
PROCEDURE VisCirkel(C: EnCirkel);
PROCEDURE SkjulCirkel(C: EnCirkel);
PROCEDURE FlytCirkelTil(C: EnCirkel;
  P: EtPunkt);
```



I Pascal skal dette skrives to adskilte steder, og det er programmørens opgave, at sørge for at data- og programstruktur holdes samlet evt i et modul. Ved objektprogrammering defineres de to områder i samme struktur og er dermed blevet til et objekt:

```
EnCirkel = OBJECT
  Radius: INTEGER;
  Placering: Punkt;
  PROCEDURE Vis;
  PROCEDURE Skjul;
  PROCEDURE FlytTil(P: EtPunkt);
END;
```

Det er ikke mere nødvendigt at bruge parametre ved kald til procedurerne, ligesom man kan ændre metodebetegnelserne til alene at beskrive operationen, da de altid »ved«, hvilken cirkel der refereres til. Anvendelsen af metoderne kan ske på denne måde:

```
VAR
  MinCirkel: EnCirkel;
  NytPunkt: EtPunkt;
BEGIN
  ...
  MinCirkel.Vis;
  MinCirkel.FlytTil(NytPunkt);
OSV
```

Et objekt kan både defineres statisk og dynamisk. Dynamiske objekter vises i figur 3. Dette princip kaldes indenfor objektprogrammering for indkapsling. Man indkapsler sin datastruktur med de operationer, som er gyldige for den. Hvis dette var det eneste nye i TurboPascal 5.5 ville programmer blive lettere at læse og vedligeholde. I programmerings-undervisningen har det stor betydning, da elever vil blive »tvunget« til at se en nøje sammenhæng mellem program- og datastruktur. Det er sikkert dette princip, som vil blive mest anvendt i undervisningen. De to andre principper kan være vanskeligere at forstå.

Objekter kan arves

Det andet princip er arv. Et objekt kan nedarves fra et andet, hvorved det nye objekt arver alle egenskaber fra det gamle, og får tildelt de nye egenskaber fra det nye objekt. Hvis der f.eks. er defineret et objekt - Node - i et modul, som er en definition af en enkelt lænket cirkulær liste, kan objektet se således ud:

```
NodePtr = ÜNode;
Node = OBJECT
  Next: NodePtr;
  function Prev: NodePtr;
end;
```

Metoden Prev returnerer en peger til det forrige element i listen. Den er defineret således:

```
FUNCTION Node.Prev: NodePtr;
VAR
  P: NodePtr;
BEGIN
  P := @Self;
  while PÜ.Next <> @Self do
    P := PÜ.Next;
  Prev := P;
END;
```

Den indbyggede funktion -@Self- returnerer adressen på sit eget objekt. Det er en specielt indbygget funktion, som man kun har behov for i meget få tilfælde, men den er hensigtsmæssig her. Metoden Prev gennemløber alle elementer i listen og stopper et element før det element, den startede fra. Forud-sætningen, for at det kan lade sig

gøre, er, at listen er cirkulær og at den indeholder mere end et element.

Metoden er defineret i standardmodulen Objects. Det skal udbygges med en yderligere egenskab, som er udskrivning af data fra elementet. Der erklæres en abstrakt metode, som arver egenskaberne fra Node.

```
DummyNode = OBJECT(Node)
  PROCEDURE Udskriv;
END;
```

Metoden kan skrives således:

```
PROCEDURE DummyNode.Udskriv;
BEGIN
  ..udskriver indholdet
  af posten..
END;
```

Med objektet DummyNode kan man anvende både de oprindelige egenskaber og den nye. Det vil senere vise sig, hvorfor det er nødvendigt at udføre metodedefinitionen på denne specielle måde.

Der skal bindes en heltals værdi til objektet, så man får en lænket liste med heltals værdier. Det er meget enkelt at udføre. Der defineres endnu et objekt IntNode, som arver alle egenskaber fra DummyNode og dermed også fra Node. Data, hvor heltallet skal gemmes i, defineres som en dobbelt Integer. De to nye metoder er Init, som sætter listeelementets værdi lig med metodens parameter og Udskriv, som skriver værdien ud på skærmen.

```
IntNode = OBJECT(DummyNode)
  Value: LONGINT;
  PROCEDURE Init(V: LONGINT);
  PROCEDURE Udskriv;
END;
```

Det færdige resultat er vist grafisk i figur 1.

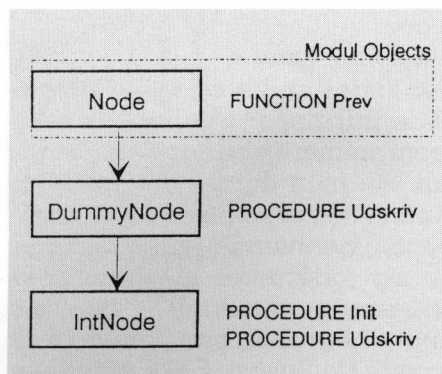
Programmet er ikke helt klar endnu, så ovennævnte program virker ikke umiddelbart. Der mangler en mulighed for at gemme strenge i listen.

I programmeringsundervisningen vil dette princip kunne anvendes, hvor der i forvejen er udviklet en række standardobjekter, som eleverne på en nem måde selv kan

udvide. Eksemplet her er sikkert ikke det bedste, da det er rigeligt abstrakt. Man kan tænke sig objekter til styring af mus, til simple grafiske operationer, som kan udvides til mere komplicerede og objekter til formaterede udskrifter på skærm og printer.

Metoder kan være polymorfe

Det tredje princip er polymorfisme. Polymorfisme er græsk og betyder »mange former«. Metoder med polymorfe (eller virtuelle) egenskaber opfører sig forskelligt afhængig af, hvordan de er defineret i den kaldende datastruktur. Hvis man f.eks. udvikler en lommeregner med anvendelsen af forskellige datatyper, kan man benytte det samme metodekald til udskrivningen uanset datatype. Teknisk set kræves det, at kompilatoren er i stand til at udføre en sen binding af variable på samme måde som en fortolker. Sædvanligvis er dette alene muligt i en fortolker, som først på kørselstidspunktet kender navnene på variable og de anvendte datatyper, og dermed kan afgøre hvorledes udskriften skal finde sted. Turbo Pascal har indbygget en sådan sen binding gennem anvendelsen af virtuelle metoder. For hvert enkelt objekt, som indeholder en virtuel metode, er der defineret en VMT - virtuel metode tabel. Det er denne tabel programmet slår op i, for at afgøre hvilken metode der skal bindes til den aktuelle variabel. Dette sker naturligvis helt automatisk uden at brugeren eller programmøren skal tænke på det. Metoderne skal alene erklæres med ordet -VIRTUAL. Fordelene ved polymorfisme er ikke umiddelbart klare, men ved en omfattende anvendelse af objekter bliver man klar over,



PICCOLINIEN

```

PROGRAM DemoObjects;
(***** *)
(* *)
(* DemoObjects *)
(* Demonstration af brugen af objekter med Turbo Pascal 5.5 *)
(* Copyright (c) Peter Ferdinand juni, 1989 *)
(* Herstedøstergade 20, 2620 Albertslund, Denmark *)
(* *)
(* Objekterne List og Node er defineret i modulet OBJECTS *)
(* *)
(***** *)

USES Objects;

TYPE
  DummyNodePtr = ^DummyNode;
  DummyNode = OBJECT(Node)
    PROCEDURE Udskriv; VIRTUAL;
  END;

  IntNodePtr = ^IntNode;
  IntNode = OBJECT(DummyNode)
    Value: LONGINT;
    CONSTRUCTOR Init(V: LONGINT);
    PROCEDURE Udskriv; VIRTUAL;
  END;

  StringPtr = ^STRING;

  StrNodePtr = ^StrNode;
  StrNode = OBJECT(DummyNode)
    Value: StringPtr;
    CONSTRUCTOR Init(V: STRING);
    PROCEDURE Udskriv; VIRTUAL;
  END;

  DemoList = OBJECT(List)
    PROCEDURE Udskriv;
  END;

VAR
  L: DemoList;

PROCEDURE DummyNode.Udskriv;
BEGIN
  Abstract;
END;

(***** *)
(* Metoder for behandling af heltal *)
(***** *)
CONSTRUCTOR IntNode.Init(V: LONGINT);
BEGIN
  Value:=V;
END;

PROCEDURE IntNode.Udskriv;
BEGIN
  WriteLn('Heltal: ', Value);
END;

(***** *)
(* Metoder for behandling af strenge *)
(***** *)
CONSTRUCTOR StrNode.Init(V: STRING);
BEGIN
  GetMem(Value, Length(V) + 1);
  Value^:= V;
END;

PROCEDURE StrNode.Udskriv;
BEGIN
  WriteLn('Streng: ', Value^);
END;

(***** *)
(* Metoder for udskrivning af datatyper defineret for en liste *)
(***** *)
PROCEDURE DemoList.Udskriv;
VAR
  P: DummyNodePtr;
BEGIN
  P:= DummyNodePtr(First);
  WHILE P <> NIL DO
  BEGIN
    P^.Udskriv;
    P:= DummyNodePtr(Next(P));
  END;
END;
BEGIN
  WriteLn('Opretter en liste med værdier');
  L.Clear;
  L.Append(New(IntNodePtr, Init(1)));
  L.Append(New(IntNodePtr, Init(-1)));
  L.Append(New(IntNodePtr, Init(100)));
  L.Append(New(IntNodePtr, Init(1000)));
  L.Append(New(StrNodePtr, Init('Turbo Pascal 5.5')));
  L.Append(New(StrNodePtr, Init('findes også til Partner og Piccoline')));
  L.Append(New(IntNodePtr, Init(12345678)));

  WriteLn('Listen udskrives på skærm');
  L.Udskriv;
END.

```

at de er uundværlige, når man skal udvikle et effektivt program. En metode gøres virtuel ved at skrive det reserverede ord VIRTUAL efter erklæringen af objektet. Desuden er det nødvendigt med en speciel procedure, som initialiserer datastrukturen og dens virtuelle metode tabel. En sådan procedure erklæres med navnet CONSTRUCTOR. Hvis man skal fjerne en virtuel metode igen benyttes erklæringen DESTRUCTOR.

Da eksemplet skal vise, hvorledes man med det samme metodekald kan oprette både et heltals-element og et streng-element i listen, skal der foretages en ændring i objektet IntNode. Det skal kunne kaldes virtuelt:

```

IntNode = OBJECT(DummyNode)
  Value: LONGINT;
  CONSTRUCTOR Init(V: LONGINT);
  PROCEDURE Udskriv; VIRTUAL;
END;

```

Metoden Init erklæres som en CONSTRUCTOR. Den opretter en virtuel metode tabel, således at objektet kan anvendes ved virtuelle kald, og indsætter heltallet i noden. Udskrivningsproceduren er virtuel, dvs at den kan overskrives af en anden procedure med det samme navn.

For at kunne gemme en strengvariabel i listen erklæres et nyt objekt – StrNode.

```

StringPtr = ^STRING;
StrNode = OBJECT(DummyNode)
  Value: StringPtr;
  CONSTRUCTOR Init(V: STRING);
  PROCEDURE Udskriv; VIRTUAL;
END;

```

Variablen Value er en peger til en streng, Strengen oprettes dynamisk ved at kalde metoden Init. Den opretter den virtuelle metode tabel og allokerer plads i lageret til strengen. Udskrivningen af strengen foregår med proceduren Udskriv, der erklæres som virtuel. Figur 2 viser grafisk hvorledes objekterne nedarves fra overordnede objekter til underordnede. De underordnede arver alle egenskaber fra objekter, som er defineret ovenover.

Det fulde eksempel

I figur 3 er hele programkoden vist. Der oprettes en liste, som består af både heltal og tekststreng. Listen

er ordnet som en sædvanlig FIFO (first in – first out) liste. Listeobjektet er erklæret i modulet Objects, som medfølger gratis ved køb af Turbo Pascal 5.5. I eksemplet benyttes nogle få af de erklærede listemetoder. Det er Clear, som sletter og initialiserer listen og Append, som tilføjer en ny node til den. I eksemplet er listeobjektet udvidet med metoden Udskriv, som udskriver alle data, der findes på skærmen. Objekterne er erklærede som dynamiske, da det giver en enklere mulighed for at oprette dem i listen. De oprettes med en ny version af standard-pascalproceduren New, som tager to parametre. Den første er en pegetil objekterklæringen og den anden er en initialisering af det dynamiske objekt. Læg mærke til, at New her er en function og ikke en procedure. Det er ligeledes nyt i Turbo Pascal 5.5.

Metoden DemoList

Udskriv gennemløber hele listen, og ved hver ny node kaldes proceduren Udskriv. Det undersøges hvilken virtuel metode variabelen tilhører, og denne metode bliver kaldt af systemet. Hvis det er et heltal kaldes IntNode.Udskriv – hvis det er en tekststreng kaldes StrNode.Udskriv. På denne måde udskrives netop den datatype, som er nødvendig. Man opnår en enkel programstruktur og et genbrug af procedurer.

Gennemgangen her er naturligvis ikke en fuldstændig redegørelse for brugen af objekter. Man bør anskaffe Turbo Pascal 5.5 og selv sætte sig ind i de muligheder det giver. Der medfølger en fortræffelig brugervejledning, som er helt uundværlig, når man skal forstå objektprogrammeringen i detaljer. Det er langt mere end en vejledning, det er en god og omfattende forklaring på objektprogrammering, som er skrevet på dansk. Den er både godt skrevet og til at få forstand af.

Turbo Pascal 5.5 findes til alle IBM-kompatible datamater og er desuden overført til RC International's Partner og Piccolinedatamater af CPI. Den kan købes hos EDB-Handelen for Skoler og Uddannelser eller ved at indsende kortet, som findes i dette nummer af Piccolinien.

RC International's nye maskinkoncept

RC900-serien er blevet meget positivt modtaget indenfor undervisningssektoren.

Det skyldes selvfølgelig den enkelte maskines styrke, som beskrevet i sidste nummer af PICCOLINIEN. Men det har vist sig, at det i ligeså høj grad er de forskellige sammenkoblingsmuligheder, der har skabt stor interesse. Vi skal i denne artikel fokusere på sammenkoblinger og koncepter.



RC International er det tredje firma i verden, der har annonceret maskiner, der benytter Intels nyeste 80486 processer.

Med denne processor, kan man afvikle programmer ca. 3 gange så hurtigt som på eksisterende 80386-baserede maskiner.

Som noget nyt, kan RC International tilbyde opgraderingssæt til RC970/20, og RC920/25 og RC990, så de kan udbygges fra 80386-maskiner til 80486.

PICCOLINIEN

Enkeltstående PC'er – 80286, 80386 og 80486

Den absolut enkleste form for opstilling er at have enkeltstående PC'er. Dvs maskiner, der ikke på nogen måde er forbundet, og som kører styresystemet MS-DOS.

Vi vil her typisk pege på RC960 som den almindelige arbejdsplads. RC960 er grundet sin meget hurtige 16Mhz 80286-processor en maskine, der kan afvikle mange MS-DOS programmer i næsten samme tempo som 80386-maskiner kan klare det. Til almindelig undervisning og administration er det idag en perfekt og prisbillig maskine. Maskinen leveres altid med 3.5" 1.44Mb disk og i 3 modeller, henholdsvis

med 40Mb, 20Mb eller uden hard-disk.

Har man brug for mere regnekraft anbefaler vi RC970.

RC970 dækker behovet, hvis man har virkelig krævende opgaver. Det kan f.eks. være Desktop Publishing, omfattende matematiske udregninger (fraktaler o.lign.), CAD-systemer og store databasesystemer.

RC970 kan leveres i en række forskellige versioner, men alle maskiner kan monteres med 3.5" disketter, så man uden problemer kan have både RC960 og RC970 i samme lokale, og med brug af de samme disketter.

Processorhastigheden i en RC970 kan være 16, 20 eller 25Mhz. For de

to sidste gælder det desuden, at de senere kan udbygges/leveres som en 80486-maskine. RC International står nemlig det kun tredje firma i verden der annoncerede brug af denne avancerede seneste processor i INTEL-serien.

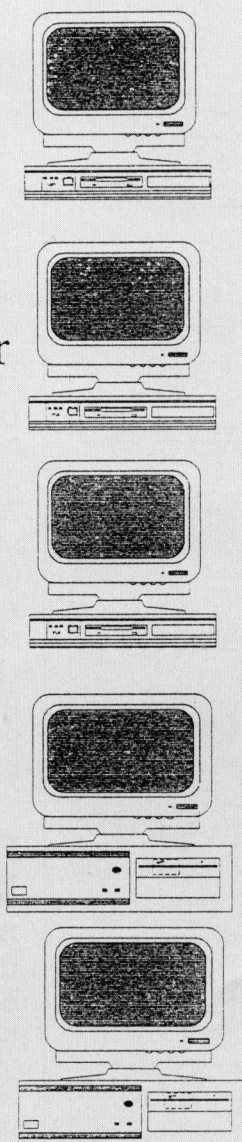
Det skal endelig bemærkes, at disse PC'er ikke er bundet til styresystemet MS-DOS. Som eksempler på alternative styresystemer der kan anvendes (evt med montering af ekstra lager) er C-DOS, OS/2, UNIX o.a.

RC900 PC'er i standard MS-DOS netværk

Alle maskiner i RC900-serien bygger på en standard AT-bus. Det betyder, at alle de forskellige indstik-

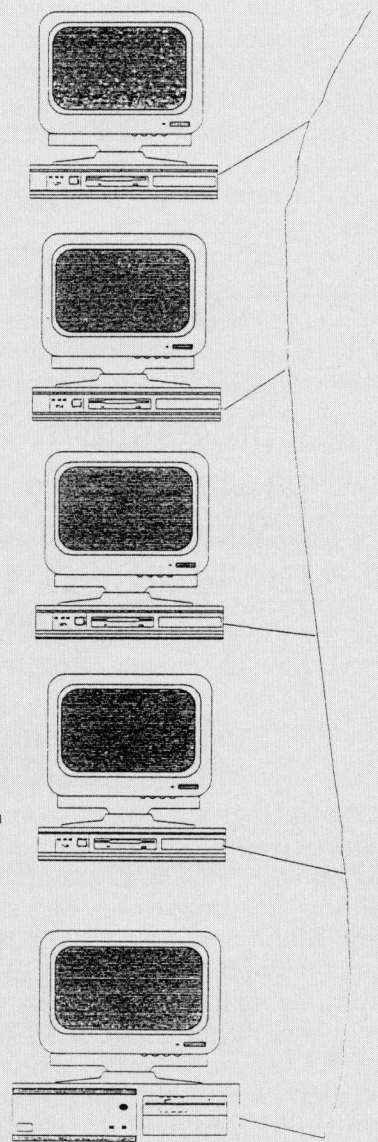
Enkeltstående PC-arbejdspladser

- RC960
- RC970
- MS/DOS
- C-DOS
- UNIX
- OS/2



RC900 - PC'er i standardnet

- MS-NET
- PC-NET
- TOKEN RING
- NOVELL NET



PICCOLINIEN

skort man kan få til denne bus-type, også passer i RC900-seriens maskiner.

Det har betydning i en lang række sammenhænge – dataopsamling, modem, kontrolenheder, MIDI-interface's, osv – og også i forbindelse med netværk. Man er frit stillet med disse maskiner, til at vælge en hvilken som helst netværksløsning bestående af software og tilhørende indstikskort. Som eksempler kan nævnes, Novell-net, Ten-net, Token-ring osv.

RC900 flerbruger UNIX-løsning

Udviklingen indenfor EDB har til tider været svær at følge.

I starten havde man (fysisk) meget

store maskiner, som kun en person kunne arbejde på ad gangen. Sidenhen blev maskinerne kraftigere, og så fandt man ud af, at flere kunne dele de samme ressourcer ved at koble flere skærme på den samme maskine. Så kom PC'erne, og nu skulle den enkelte igen sidde med sin egen maskine. Men nu er også PC'erne blevet så kraftige, at man er begyndt i stigende grad at anvende disse som flerbrugerløsninger.

Alle 80386- og 80486-maskiner i RC900-serien kan anvendes som UNIX flerbrugerdatamater. Valget af model er afhængig af hvor mange samtidige brugere man ønsker at kunne koble på.

– RC970. Som den fysisk mindste

kan RC970 udstyres med op til 4 terminaltilkoblinger i en såkaldt multiplexer.

RC International anbefaler ikke flere samtidige brugere på denne maskine, bl.a. fordi den ikke har indbygget batteri-back-up, hvilket er en stor fordel ved større UNIX-løsninger, hvor det er vigtigt at systemet lukkes korrekt ned.

I teorien kan der dog via lokalnet (se nedenfor) tilkobles flere.

– RC950 anbefales hvor man har op til 16 terminaltilkoblinger. Alle kan tilsluttes maskinen direkte. RC950 kan leveres med batteri-backup.

– RC990 – den største og nyeste maskine i RC900-serien – anvendes ved behov for op til 48 samtidige brugere. I princippet kan den dog klare langt flere via lokalnet, men for at få en rimelig performance bør man ikke være flere. I undervisningssammenhæng vil dette normalt heller ikke være tilfældet.

Bemærk at RC990 senere kan leveres som 80486-maskine.

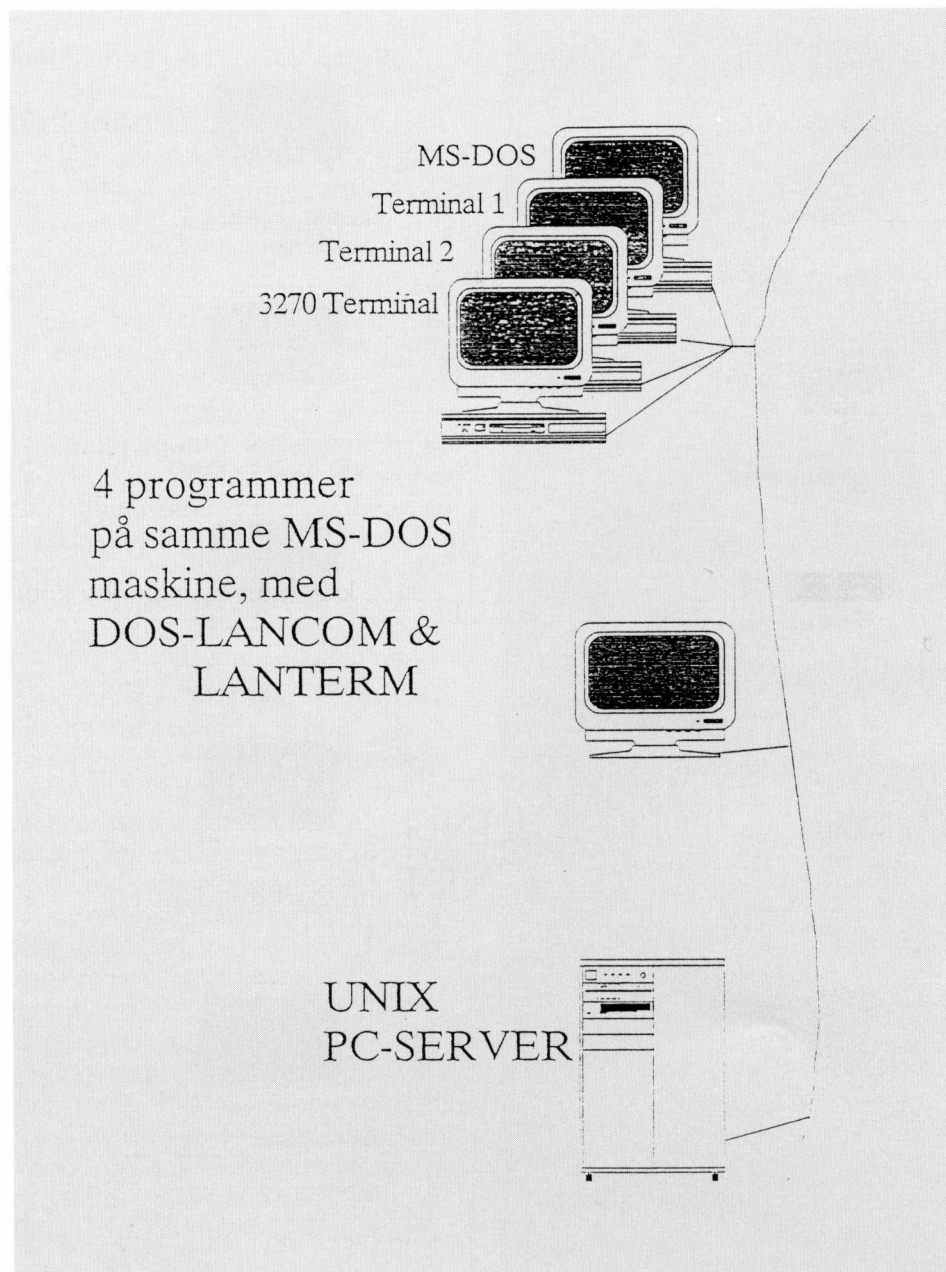
Som terminal kan man enten anvende en dedikeret terminal, eksempelvis RC920, eller man kan benytte en PC'er som f.eks. en anden RC900 eller en Partner/PICCOLINE (se nedenfor).

Det skal endelig nævnes, at mens RC990 altid er født som en UNIX-maskine, så kan man med både RC950 og RC970 starte med en almindelig PC, og sidenhen udbygge den til en UNIX-flerbrugermaskine. Udbygningsprisen er stort set lig med prisforskellen. Man behøver altså med RC International-produkter ikke at købe helt nye produkter, hvis man på et senere tidspunkt vælger at sadle om.

Fordele og ulemper ved UNIX og flerbrugerløsninger

UNIX er idag det absolut mest udbredte styresystem til flerbrugerdatamater, og det benyttes specielt meget i forbindelse med større administrative systemer f.eks. indenfor stat og kommune.

UNIX er stærkt, fordi det – ligesom f.eks. C-DOS – er et multitasking operativsystem, så flere processer



PICCOLINIEN

kan afvikles samtidigt, en nødvendighed i forbindelse med terminaler. UNIX indeholder desuden en række faciliteter til opdelinger af brugerområder, adgangskoder, sikkerhedssystemer osv. Der er desuden udviklet en lang række programmer til UNIX.

UNIX hænger stærkt sammen med programmeringssproget C, der udmærker sig ved at være meget let flytbart. Det betyder, at programmer kan overføres fra en maskintype til en anden, og ofte flyttes fra en type styresystem til en anden uden de store problemer.

Ulempen, set i undervisningsøjemed, ved terminalløsninger er, at det kun er tekst og semigrafik, der overføres til terminalerne. Det betyder, at det kun er hovedarbejds-

pladsen, der kan anvende grafik. Det er ikke den store begrænsning i forbindelse med f.eks. skoleadministration og skolebibliotek, idet man her sjældent benytter grafik. Men i forbindelse med anvendelse af EDB i undervisningen er det et klart handicap, idet mange undervisningsprogrammer benytter grafik. Omvendt er et område som tekstbehandling dog fortsat et meget anvendt program, og bliver det nok i endnu højere grad i fremtiden, og det kan jo sagtens afvikles som terminalløsning.

Så indtil teknikken har udviklet sig, så også grafik kan sendes ud på terminalerne (f.eks. vha. lyslederkabler), vil det nok være vanskeligt at basere et EDB-system til undervisning på en ren UNIX-løsning.

Men RC International har selvfølgelig, som nævnt nedenfor, en stærk kombinationsløsning.

UNIX flerbrugerløsning eller PC-netværk – hvorfor vælge?

At få UNIX og MS-DOS til at fungere sammen på det samme netværk er sin sag – men det er faktisk det, der ligger til grund for RC International's egen netværksløsning.

Konceptet består af følgende:

- En UNIX flerbrugerdatamat
- En række PC'er
- Et RC-lokalnet
- Evt egentlige terminaler (f.eks. RC920)

Alle maskiner er koblet sammen via et lokalnet. Det kan være et 1Mbit RcMikronet eller et 10Mbit Thin Ethernet.

På UNIX-maskinen ligger der så et specielt program udviklet af RC International kaldet PC-SERVER.

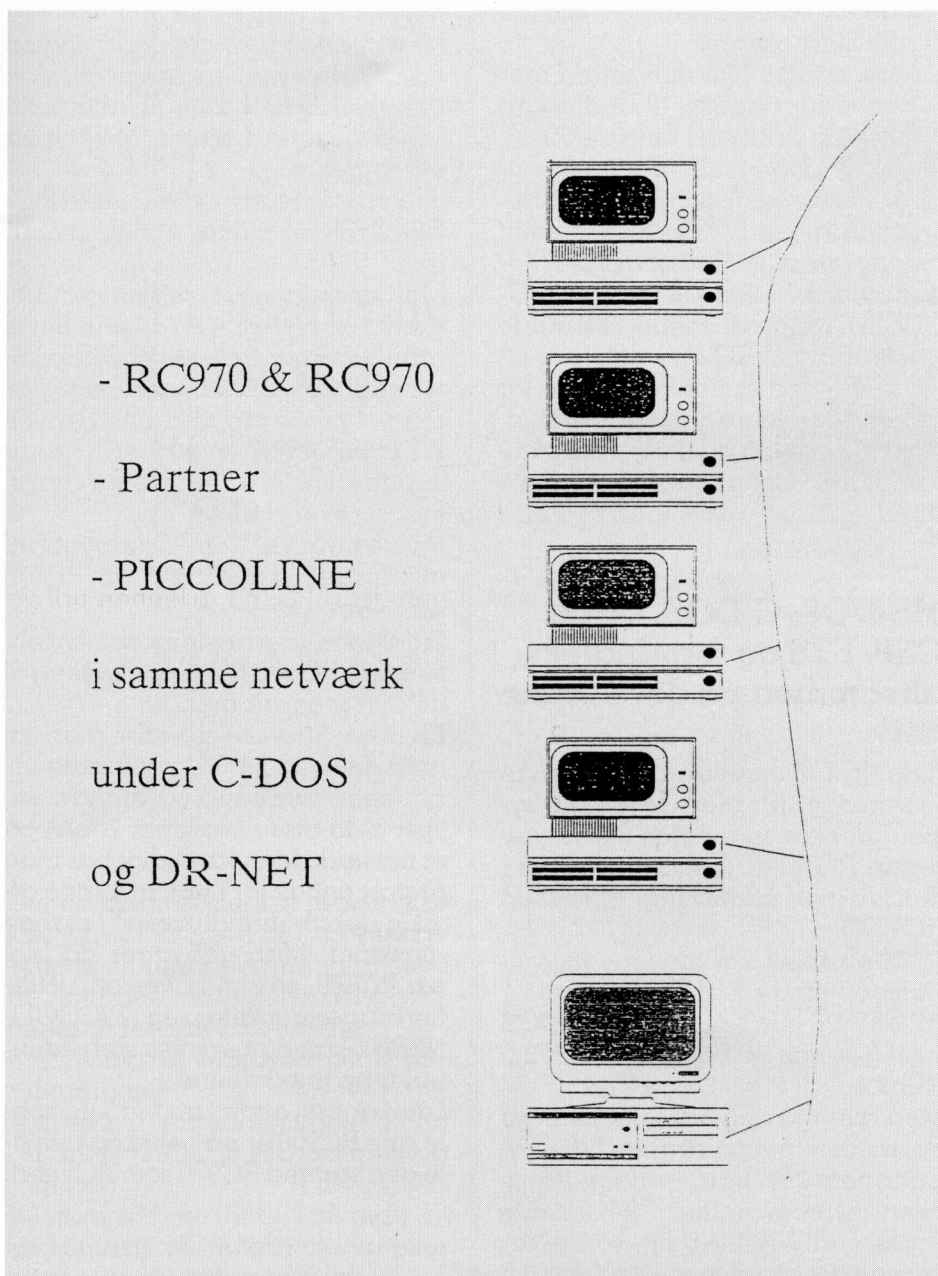
Det sætter UNIX-maskinen i stand til at kunne fungere som almindelig netværter for en MS-DOS maskine.

Dette sker vel at mærke uden at man behøver at dedikere UNIX-maskinen som netværter. Den er stadig fuldt funktionsdygtig som arbejdsmaskine, i modsætning til hvordan de fleste MS-DOS netværter fungerer. RC920-terminaler koblet op mod denne (f.eks. via nettet), kan altså stadig fungere på samme tid.

Er der mere end en UNIX maskine i nettet, kan man selvfølgelig koble sig op imod dem alle, og på den måde navngive diske og printere rundt omkring i nettet, der derefter kan bruges, som om de var private.

4 konsoller under MS-DOS

Men der er en yderligere facilitet, når man benytter denne netværksopbygning. De PC'er, man har koblet op på netværket, kan nemlig, samtidig med at de fungerer som MS-DOS maskiner, også fungere som terminaler. Vha. programmet LANTERM skaber man sig, hvad der virker som 4 konsoller. De 3 af dem er blot terminalopkoblinger mod en eller flere af UNIX-værterne. Herved er man i stand til at arbejde med op til 4 programmer på samme tid på sin MS-DOS maski-



ne, hvoraf de tre ganske vist afvikles på en UNIX-maskine.

En af terminalopkoblingerne kan også være mod en værtsdatamat, der via det offentlige datanet (såkaldt 3270-forbindelse) nåes uden for skolens område. Det sidste er måske især relevant i forbindelse med skoleadministration. Mange er idag koblet op imod en central datamat fra kommunen eller amtet, hvor de kører økonomisystem, samtidigt med at de f.eks. skal køre GAS, DASK, HARDY eller et andet skoleadministrationssystem lokalt. Begge dele vil kunne køre på en RC900-PC samtidigt.

Så når man idag ser to retninger i den fremtidige anvendelse af EDB:

- Flerbrugerløsninger (typisk under UNIX)
- Netværksløsninger

så kan man med RC-net roligt sige: Hvorfor vælge. Man kan nemlig på det samme net både køre almindelig netværk og terminalopkobling – endda samtidigt og endda samtidigt på den samme maskine!

RC970/950 som vært i et Partner/PICCOLINE-net

Såvel RC950, RC960 og RC970 kan køre C-DOS. På RC950 og RC970 er der yderligere lavet en speciel udgave med et net-interface svarende til det der benyttes i Partner/PICCOLINE-net.

Det betyder, at man har mulighed for at forstærke et allerede eksisterende net markant med en særdeles kraftig netvært. Det kan naturligvis lade sig gøre uanset om man benytter et 10Mbit Thin Ethernet eller (det mest udbredte) 1Mbit RCMikronet.

Samtidigt opnår man selvfølgelig, at man får en stærk 80386 maskine indenfor dørene og dermed en indgang til AT-teknologien. Man skal være opmærksom på, at selvom man kører C-DOS på RC950/970 vil man stadig kunne afvikle stort set alle programmer lavet til at køre under MS-DOS. Det er kun 640Kb lagergrænsen der kan give problemer, idet C-DOS fylder mere end MS-DOS.

Hvis problemet opstår, kan man altid blot starte maskinen som en MS-DOS maskine, men så kan den altså ikke samtidig fungere som netvært.

Opdatering til netværk

Har man ikke netværk allerede, kan dette måske være en god anledning til at få det. Netværk opfattes af mange som tidens løsen, idet man har et større og større behov for udveksling af data. Også her i bladet har vi tit omtalt fordele ved brug af netværk.

- Man undgår alle problemerne med de mange disketter, idet alle skolens programmer kan lægges centralt på en eller flere værter.
- Man undgår problemerne kendt fra 4-brugerklynger med at nogle programmer holder disk-filer åbne, og derved blokerer disktestationen for andre.
- Man sikrer sig, at alle elever og lærere har adgang til alle data og programmer uanset om de sidder ved den samme maskine som sidst eller ej.
- Man har fra alle maskiner i mulighed for adgang til alle andre diske og printere i hele nettet.

RC International tilbyder i dette efterår specielle opdateringssæt bestående af en RC950/RC970 samt netværkskort til Partner/PICCOLINE, hvor netdelene til Partner/PICCOLINE sælges til meget favorable priser. Ring til RC International eller en af vore skoleforhandlere for at høre nærmere om dette tilbud, eller mød op på et af vore efterårsseminarer (se andetsteds i dette blad).

MS-DOS, C-DOS, UNIX, CSP, FTS og terminaler – altså sammen på det samme net!

Hvis man blander de to ovennævnte løsninger får man dette store system, der kan indeholde alle nuværende RC International produkter (faktisk incl. mainframe maskinen RC9000).

PC'ere kobles sammen med terminaler op mod en UNIX-maskine. RC950/970 kan bruges som C-DOS vært overfor et Partner/PICCOLINE-net.

Men disse to ting kan eksistere på det samme net på samme tid!

Ydermere kan man – vha af transmissionsprogrammet CSP – koble Partner/PICCOLINE op, så de virker som terminal mod UNIX-maski-

nen. Og dette sker vel at mærke uden at man mister muligheden for at arbejde i de 4 C-DOS-konsoller. Det betyder, at man på samme tid kan afvikle et C-DOS undervisningsprogram på sin PICCOLINE, og være koblet op som UNIX-terminal og f.eks. køre RcTekst på denne. Tilsidst er det også muligt – på samme tid – at lave fil-overførsel fra en maskine efter frit valg i nettet til en anden vha. af filoverførselsprogrammet FTS. På denne måde kan man via nettet udveksle data mellem f.eks. en PICCOLINE og en RC960.

Fremtiden

Måske virker den sidste løsning overvældende, og de fleste tænker nok, at det er alt for meget her og nu.

Men denne tanke går godt i spænd med RC Internationals filosofi. Man behøver ikke at tage skridtet fuldt ud på en gang. Lad os tage et typisk eksempel.

Skolen står idag med en række PICCOLINER – måske i net, måske ikke.

Man ønsker i første omgang at forstærke sit system, ved at gøre det til et netværk med en stærk højkapacitet vært. På lidt længere sigt har man et ønske om at kunne benytte AT-kompatible maskiner, for at kunne afvikle en række programmer skrevet til disse.

Første trin vil så være at udbygge til et netværk, f.eks. med en RC970 som vært.

Sideløbende kan man købe enkeltstående RC960-PC'er til afvikling af de nye programmer.

Efter en årrække mærker man, at anvendelsen af PC-programmerne tager overhånd og at man ønsker at få disse maskiner koblet på et netværk. Men samtidigt har man et stort behov for f.eks. at kunne afvikle tekstbehandling på mange maskiner. Man udbygger da blot sin RC970, så det bliver en UNIX-flerbrugermaskine, og PICCOLINERNE kan da benyttes som terminaler op imod denne.

I den anden ende kan man nu koble sine RC960'er på nettet og benytte den samme RC970 som PC-vært.

PICCOLINIEN

På samme fysiske net

RC990

RC970/RC950

RC990

RC9000

RC45/900

RC960

Partner

PICCOLINE

Programmer

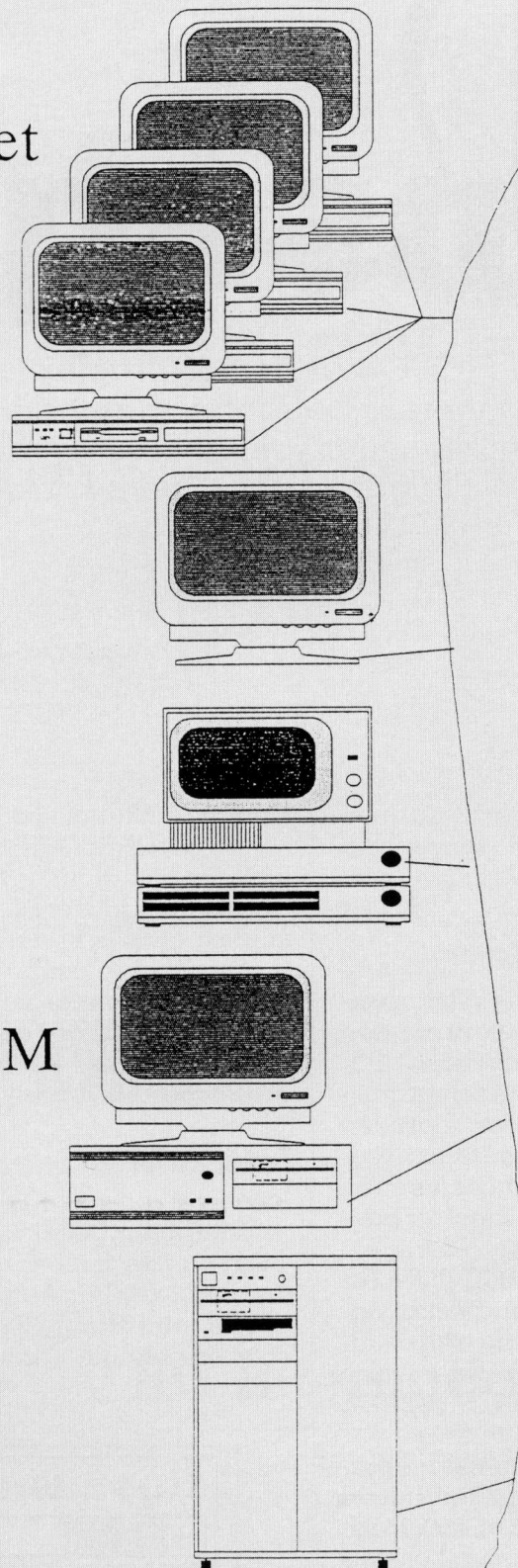
C-DOS

PC-server/LANCOM

CSP

FTS

MS-DOS



Det er klart, at denne artikel ikke kan blive fuldstændigt dækkende over de muligheder man har med RC International-udstyr. Af andre muligheder kan kort nævnes

- afvikling af MS-DOS på UNIX-terminaler
- direkte terminalopkobling af Partner/PICCOLINE uden brug af net

Vil man vil høre mere, kan man altid kontakte RC International eller en af vore skoleforhandlere.

Konsequence

I PICCOLINIEN 4/88 bragtes en artikel, der ud fra nogle generelle betragtninger skulle give et indtryk af, hvilke muligheder computeren havde i musikundervisningen. Det følgende er en mere konkret fortsættelse.

Når man ønsker at arbejde med PICCOLINIEN som sequencer, har man brug for følgende ting:

- Programmet KONSEQUENCE
- MIDI instrument(er)
- Et MIDI interface

Udstyret med disse ting vil man være i stand til at vise eleverne den teknik, der i stor stil anvendes i indspilningsstudier. (Samt selvfølgelig at lade dem selv prøve). Desuden kan sequenceren fungere som realistisk 'skitseblok' for musiklæreren, der ønsker at afprøve ideer før præsentation for eleverne. Yderligere åbnes der med PICCOLINIENS adgang til MIDI en lang række andre muligheder. Denne artikel udgør imidlertid kun en beskrivelse af programmet KONSEQUENCE samt det tilhørende interface.

KONSEQUENCE er en 16 spors sequencer. Dvs. at brugeren har 16 områder, der hver især kan indeholde MIDI informationer. I praksis foregår en optagelse på den måde, at man ved en meget enkel betjening angiver hvilket spor, man ønsker at optage på. Dernæst sættes det hele igang med et tryk på RETUR tasten, hvorefter programmet registrerer, når man spiller på synthesizeren. Princippet i enhver sequencer er så, at man kan lade det netop optagede afspille, mens man

optager endnu et spor med en ny lyd. Med KONSEQUENCE har man mulighed for at nå op på ialt 16 instrumenter. Udover selve det at optage og afspille er der selvfølgelig en lang række andre betjeningsmuligheder. Intet af dette ville imidlertid kunne lade sig gøre, hvis PICCOLINIEN ikke kunne kommunikere med MIDI-udstyr.

Interfacet

Interfacet er naturligvis forsynet med MIDI stik. Der er 1 'MIDI IN' indgang og ikke mindre end 4 identiske 'MIDI OUTS'. At der er hele 4 'OUTS', vil i mange tilfælde kunne spare en såkaldt 'Thru box'.



Ved indspilning af musik er det ofte nødvendigt at benytte en metronom for at være præcis nok. I interfacet er der indbygget elektronik til at levere 'dunk' som metronomslag, hvor det første slag er forskelligt fra de andre. Hvor stor kraft højttaleren skal 'dunke' med, kan indstilles med et potentiometer. Den samme metronomlyd er desuden ført ud til et stik således, at man - hvis ønsket - kan lade metronomlyden indgå sammen med andre lydkilder i en mixer.

Programmet KONSEQUENCE

Ved optagelse rummer KONSEQUENCE flere forskellige muligheder for hjælp til at holde tempoet. Det er bl.a. de beskrevne 'dunks' fra interfacet, som kan aktiveres/deaktiveres fra programmet.

Derudover kan sequenceren bringes til at anslå tangenter via MIDI på de tidspunkter, hvor metronomslag er passende. Lyden afhænger således alene af MIDI instrumentet (f.eks. et 'rim shot' fra en trommemaskine). Andre muligheder åbner sig ved anvendelse af synkronisering, som er en måde, hvormed sequenceren dirigerer trommemaskinen (start, stop og tempo). Den

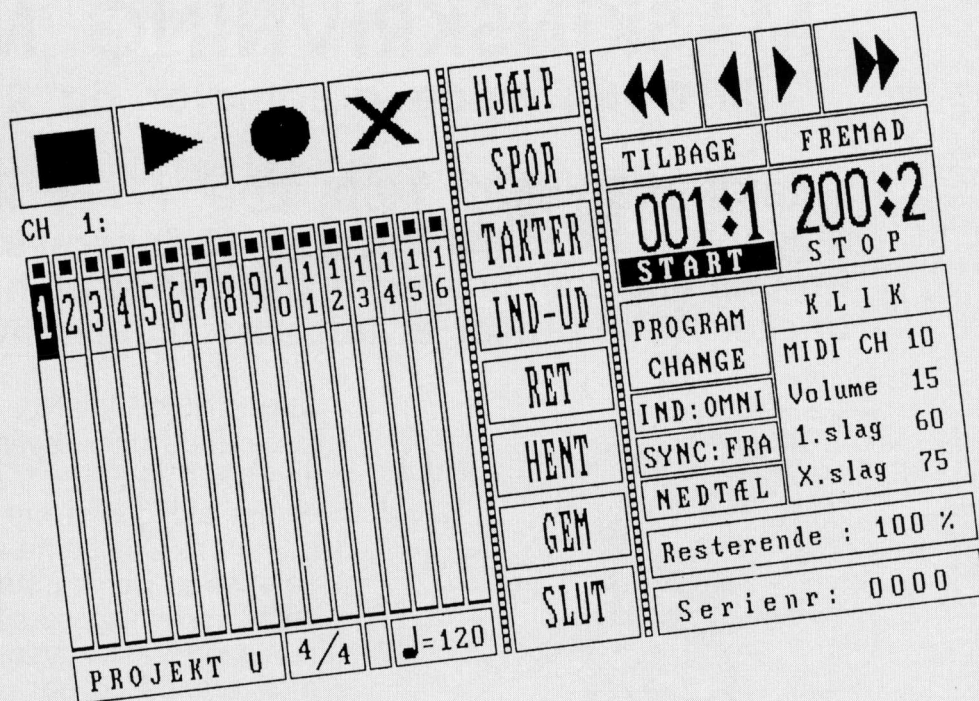
DE BLÅ SIDER

typiske anvendelse af denne funktion er imidlertid ikke som en form for metronom. Normalt vil man indspille et 'trommespor' på trommemaskinen og så lade sequenceren aktivere den på de rette tidspunkter.

Overblik

Når man arbejder med en sequencer, er det vigtigt, at man har styr på MIDI kanalerne. Enhver uorden kan give de mærkeligste udslag. KONSEQUENCE indeholder nogle funktioner, der giver brugeren det nødvendige overblik. Hvert spor kan man således give et navn, og en særlig funktion 'status oversigt' viser sporene (8 af gangen) i en liste på skærmen. Både spor nr, navn og MIDI kanal er opført her samt en længde angivelse, der viser hvor mange 'MIDI-blokke', der ligger på det pågældende spor.

M.h.t. MIDI kommunikation har man under optagelse mulighed for at vælge 'OMNI MODE', hvor der ukritisk optages fra enhver MIDI kanal, eller specificere en bestemt ka-



Vi bestiller til Piccoline/Partner

_____ stk. Konsequence	3700,-
_____ stk. Vejrstationsprogram	800,-
_____ stk. ACP/RcTekst/Mikrologo	975,-
_____ stk. TurboPascal 5.5	900,-
_____ stk. TurboPascal 5.5 + CPI-Graf	1450,-
_____ stk. IEEE-488	390,-
_____ stk. L-disk til PICCOLINE	2700,-
_____ stk. Sæt med 2 LEGO Tryksensor	100,-
_____ stk. "Hold styr på robotterne"	80,75
_____ stk. BOX diskette til denne bog	275,-
_____ stk. I-APL	150,-
_____ stk. FLYTGEO	395,-
_____ stk. 8087 numerisk regneenhed	1600,-
_____ stk. Tegn med Musen vers. 2	600,-
_____ stk. EDDIE	375,-
_____ stk. Tegn med Musen + EDDIE	750,-
_____ stk. "Partner/Piccoline under lup"	128,69

Ovennævnte priser er undervisningspriser og er excl. moms.
 Send mig yderligere informationsmateriale om RC900 serien.

Skole _____
 Navn _____
 Adresse _____
 Postnr. _____ By _____
 Telefon _____
 Underskrift _____

Ufrankeret
 svarforsendelse

48

Postbesørges
 ufrankeret
 RC International
 betaler portoen

RC International

Skole- og undervisningsafd.
 Lautrupbjerg 1
 2750 Ballerup

MASKINSKRIVNING MED EDB

til DOS-styrede datamater og RC-Piccoline

MASKINSKRIVNING MED EDB er et komplet undervisningsprogram til indlæring af 10-fingersystem og blindskrift. Programmet kan bruges af alle, der på en hurtig og sjov måde ønsker at lære eller at forbedre blindskrift.



MASKINSKRIVNING MED EDB er selvinstruerende og kræver ingen forudgående kendskab til EDB. Med sin enkle opbygning er der aldrig tvivl om, hvad der forventes af brugeren. Øvelserne er metodisk opbyggede, således at der fra starten indøves den rette maskinskrivningsteknik, hvilket sikrer en hurtig og sikker indlæring. Det virker meget motiverende og gør undervisningen spændende og afvekslende.

Programmet fungerer ved, at eleven skriver oven i den tekst, der vises på skærmen. Efter hver øvelse fås en bedømmelse på baggrund af hastighed og antal fejl jvf. karaktergivningingen til Folkeskolens Afgangsprøve i maskinskrivning.

Som noget nyt kan programmet fra version 2.0 simulere såvel 5- som 15-minuttersprøven, idet der er 3 forskellige indstillinger til, hvordan eventuelle fejlanslag skal rettes. Endvidere er der mulighed for at gemme resultaterne i en statistik, som kan skrives ud på printer.

NYHED!!

INDHOLD:

- 190 færdige øvelser med en klar metodisk opbygning
- plads til yderligere 224 øvelser til supplerende træning
- mulighed for at gemme elevresultater i en statistik
- tekstbehandlingsafdeling så let, at alle kan finde ud af det
- karaktergivningsnøgle jvf. Folkeskolens Afgangsprøve
- indbygget vejledning i programmet
- lærervejledning, der nøje forklarer programmets opbygning
- udførlig installationsvejledning

Pris: 1350 kr. for diskette + lærervejledning

ST-DATA
Forteledet 12
8240 Risskov
Tlf.: 86 17 94 70

DE BLÅ SIDER

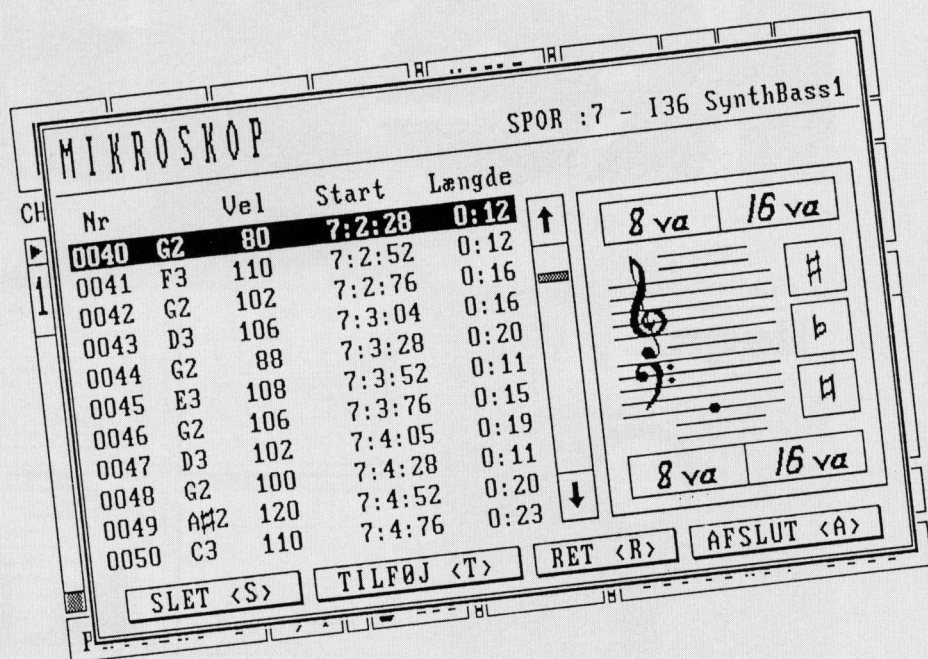
nal. Desuden er der en såkaldt 'soft thru' funktion, der især er anvendelig, hvis man har flere MIDI enheder tilsluttet.

Med den særlige 'Direkte MIDI' funktion kan man angive nogle tal, som derefter sendes ud via MIDI til de tilsluttede instrumenter. Dermed kan man meget enkelt eksperimentere med MIDI kommandoer og desuden illustrere hvorledes MIDI fungerer.

Med 'nedtæl' aktiveret får man 2 takters respit før optagelse, der iøvrigt kan foretages i næsten ethvert tænkeligt tempo (tempoet kan iøvrigt altid ændres). Der er desuden forskellige taktarter at vælge imellem.

Editering

Kvantisering er en funktion, der foretager automatisk tilpasning af anslag således, at de kommer helt præcist. Hvert anslag kan desuden rettes til individuelt på forskellige måder med 'Mikroskopet'. Der kan slettes, tilføjes eller editeres med hensyn til nodeværdi, anslags-



hårdhed, starttidspunkt og længde. Enhver tidsangivelse angives med takt:1/4 node:fininddeling. Finindelingen går fra 0 til 95, mens sequencerens interne præcision er på 1/192 af en 1/4 nodes længde.

Mageligheder

KONSEQUENCE imødegår visse MIDI-tekniske problemer ved, at kunne sende bl.a. 'all notes off' kommandoer.

Program Change er en anden ting, der skulle gøre livet lettere at leve for KONSEQUENCE-brugere. Det er, hvad man overfladisk kunne kalde 'delvis automatisk indstilling af MIDI-udstyr'.

De musikalske frembringelser kan man naturligvis gemme på disk og hente ind igen efter ønske. I den forbindelse opretholder KONSEQUENCE et katalog således, at de angivne navne kan være på op til 28 tegn. (altså omtrent som RcTekst II) Det gør, at man ikke behøver, at kunne huske på korte uforståelige filnavne.

Vejledning

Den til programmet hørende brugermanual er naturligvis på dansk. Den indeholder bl.a. en introduktion til MIDI, samt en punkt

for punkt gennemgang, der leder brugeren gennem den første optagelse.

PICCOLINEN skal være forsynet med min. 512 Kb RAM, hvilket dog i visse tilfælde ikke vil være nok. Programmet kræver således omkring 375 Kb frit brugerlager. Til gengæld benyttes der ikke overlays (hele programmet indlæses i maskinen på en gang). Det har betydning for programmets udførselshastighed især ved anvendelse af net.

KONSEQUENCE supporterer enhver PICCOLINE skærmtypen. På RC741 farvemonitoren udnyttes kun den del af skærbilledet, der svarer til en lavopløsningsskærm. Det anbefales, at man benytter en mus, men alle funktioner kan også udføres v.h.a. tastaturet.

Prisen på kr.

Kr. 3700,-

inkluderer følgende som sælges samlet:

- Programdiskette
- Vejledning
- Interface
- Tastaturskinne

EDB-diskette GEOGRAFI
 PICCOLINE/PICCOLO
 NYREVIDERET vers. 3.1

Programmet består af:

- I Hovedmenu
- II EUROPA, der består af 12 underprogrammer med de europæiske staters beliggenhed og tilhørsforhold som emne, med f.eks. samhandel, politisk samfundssystem og forsvarssystem.
- III KLIMA, der har jordens klimazoner; isæt tempereret, som emne.
- IV BJERGE, der har bjergkæders alder og udseende som emne, bl.a. bjergkædedannelse, erosion og aflejring.

PRIS:

Piccoline: 1 diskette kr. 450

Piccolo : 3 disketter kr. 400

Opdatering (ved fremsendelse

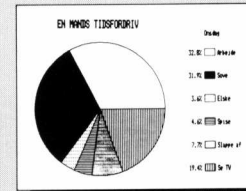
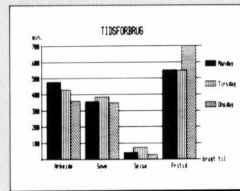
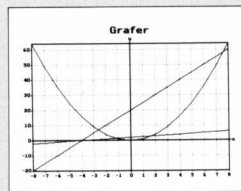
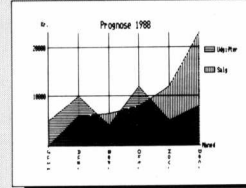
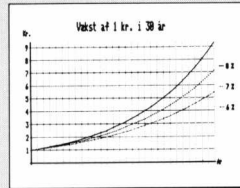
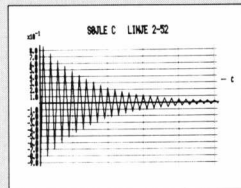
af org. disk.): kr. 75

Programmet forudsætter ingen kendskab til EDB.

J.S.Konsulent Aps, Postbox 4

4040 Jyllinge, tlf. 02 13 23 96

Regn2



Regn2 som indtil nu er købt af ca. 400 danske skoler, kan nu også leveres til PC-kompatible datamater - bl.a. til RC1960 og RC1970.

Regn2 er et elektronisk regneark.

Regn2 kan med fordel benyttes i folkeskolen, på gymnasiet, på HF-kurser, seminarier og i voksenuddannelserne.

Regn2 kan anvendes som et repræsentativt eksempel på, hvad datamater bruges til i erhvervslivet.

Regn2 kan bruges som et redskab i mange fag. I folkeskolen bl.a. til matematik, samtdsorientering, geografi, biologi, fysik/kemi, hjemkundskab og elektronik.

Regn2 kan i gymnasiet indgå i generelle edb-kurser, og eleverne kan benytte det ved udarbejdelsen af rapporter.

Regn2 leveres med en grundig teknisk vejledning, bl.a. indeholdende ideer til anvendelse af regneark i undervisningen.

Regn2 - nu version 2.0 - bl.a. med xy-grafer.

Regn2 koster stadig kun 800 kr. + moms.

Regn2 til både Piccoline/Partner og til PC: Samlet pris 1200 kr. + moms.

Brug Regn2 er et selvinstruerende elevmateriale indeholdende en række kurser, der gør eleven fortrolig med regnearket.

Bestilles til: Piccoline/Partner PC: 5 1/4" diskette 3 1/2" diskette

Stk.: Regn2 Diskette og vejledning: 800 kr.	Stk.: Regn2 demoversion og vejledning: 150 kr *	Stk.: Regn2 ekstra vejledning: 70 kr.
Sæt: Brug Regn2 elevh./klassesæt à 10 stk.: 500 kr.	Stk.: Brug Regn2 elevhæfte/enkelteksemplar: 70 kr.	* Beløbet refunderes ved køb af Regn2. Alle priser er skolepriser og excl. moms.



Skole: _____

Adresse: _____

Postnr.: _____ By: _____

Att. (navn): _____

Sendes til: Jørgen H. Christiansen, Snebærvej 4, 8270 Højbjerg. Tlf.: 06 27 43 09

DE BLÅ SIDER

KONSEQUENCE forhandles af Rockilden ApS, Tlf. 42 36 68 68, men kan også bestilles gennem de blå sider i PICCOLINIEN. Som sædvanlig er prisen excl. moms, og der ydes ikke særlig undervisningsrabat på produkter solgt gennem PICCOLINIEN.

FAKTA

S P R O G L Æ R E R !

GIVER SKOLENS "PICCOLINER" OGSÅ DIG EN HANDESRÆKNING?

HT - D A T A HAR UDVIKLET TRE VIRKELIG GODE TILBUD TIL DIG:

"DANSK MED DATAMASKINEN (1)"

"DANSK MED DATAMASKINEN (2)"

"TYSK MED DATAMASKINEN (1)"

HVER DISKETTE RUMMER FLERE HUNDREDE OPGAVER, OG EN INSTRUKTIV MANUAL GØR DET LET SELV AT FREMSTILLE ARBEJDSSTOF TIL DET AKTUELLE ELEVMATERIALE!

- * KRÆVER INGEN FORUDGÅENDE KENDSKAB TIL EDB
- * STORE MULIGHEDER FOR DIFFERENTIERET UNDERVISNING
- * "KEDELIG GRAMMATIK" GØRES SPÆNDENDE
- * ALLE PROGRAMMER ER "SELVSTYRENDE"
- * ELEVERNE FÅR ØJEBLIKKELIG "RESPONSE"
- * MOTIVATIONEN FOR FAGET ØGES
- * ELEVOPGAVER KAN UDPRINTES TIL KLASSEARBEJDE

BIBLIOTEKSCENTRALEN UDTALER OM "DANSK MED DATAMASKINEN":

- PROGRAMMERNE VIRKER VELOVERVEJEDE OG GENNEMPRØVEDE.
- OPGAVERFORMEN OG DATAMASKINEN GÅR FINT I SPÆND SAMMEN.
- DET ER GJORT SPÆNDENDE, LÆRERIGT OG TOPPROFESSIONELT!

DISKETTE OG MANUAL KR. 1000,- + MOMS.

HT - D A T A
VIEMOSEVEJ 8
7120 VEJLE Ø
75 81 43 72

- KONSEQUENCE
16 spor
Soft thru
Hent/gem katalog
Synkronisering
Anslagseditering
Kvantisering
Kliks
Program change
Reset knap
Status oversigt
Direkte MIDI

- INTERFACE
1 MIDI IN indgang
4 MIDI OUT udgange
Indbygget metronom
Udgang til mixer
Tilsluttes 'local printer' porten

- PRIS KR 3700,-

Disketter

Køb til skolen!
Køb til hele kommunen!

Kvalitet! - Garanteret 100% fejlfri CIS disketter til priser uden for al konkurrence!
5 1/4" = 1,6 Mb.

5 1/4" 48 TPI DS/DD kr. 7.40

(IBM PC/Amstrad/
Commandore/Piccolo)

5 1/4" 96 TPI DS/DD kr. 11.50

5 1/4" 96 TPI DS/HD kr. 14.80

(Piccoline/Partner/
IBM-AT)

(I plasticassette)

3 1/2" 135 TPI MF 2DD kr. 14.80

(i plasticassette)

3 1/2" 135 TPI MF 2HD kr. 27.10

(i plasticassette)

Priserne er pr. stk. i
10 stk. pakninger

Excl. moms - excl. forsendelse

Ved 50 stk.: forsendelse incl.

TYGE KONSULENT - Postbox 4
4040 Jyllinge - 02 13 23 96

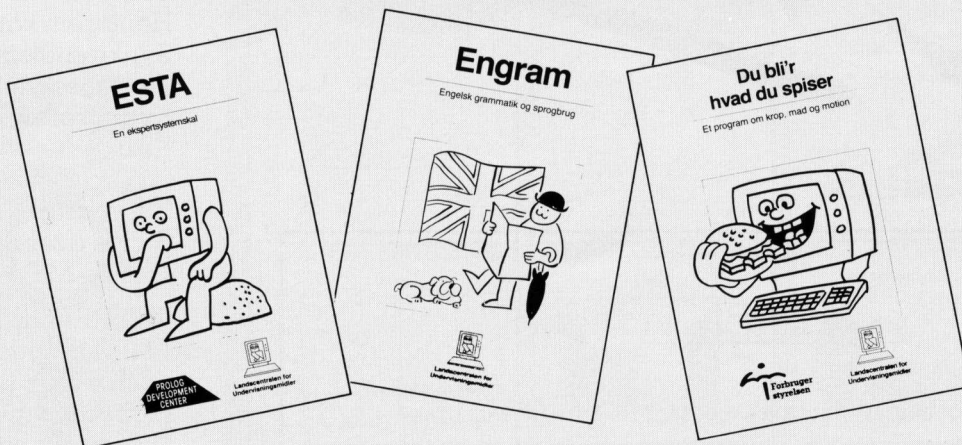
(Bedst efter kl. 15 og mand. kl. 8-12)

DE BLÅ SIDER

Edb i undervisningen

3 gode kort

Ved du, at Landscentralen er Danmarks største producent og distributør af edb-programmer til undervisning.

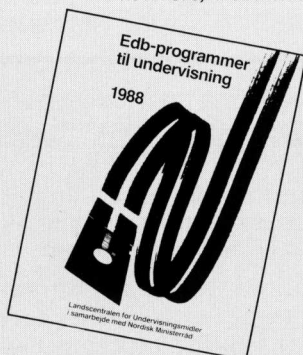


ESTA er skallen til et ekspert-system. materialet består af en diskette med selve styresystemet, to datadisketter med eksempler på små ekspertsystemer, samt skriftligt materiale. Pris kr. **976,-** incl. moms.

ENGRAM er et program kompleks til træning i engelsk grammatik og sprogbrug. Pris **endnu ikke fastsat**.

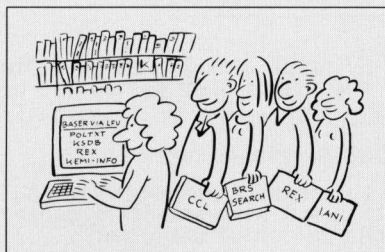
Du bli'r hvad du spiser

En af mulighederne i programmet er at vælge en person evt. sig selv - angive alder, vægt, højde og køn. Derefter vælges mad og drikke for én dag samt de aktiviteter, bl.a. sport/motion, der udføres i løbet af én dag. Programmet kan så udregne hvilken konsekvens det får for vægten samt risikoen for hjerte/karsygdomme og kræft. Pris kr. **850,-** incl. moms.



Edb kataloget

indeholder over 400 beskrivelser af edb-programmer til undervisning. Kataloget ligger også som en base i Landscentralens on-line tjeneste i opdateret stand. Pris kr. **106,15** incl. moms.



Landscentralens databaseservice

Landscentralen kan nu tilbyde skolerne billig adgang til en række af landets professionelle databaser. Indtil videre tilbyder vi: KSDB fra Danmarks Statistik, REX fra Det Kongelige Bibliotek, Kemi-Info fra Datacentralen. Og snart: POLTXT fra Politiken, Ritzaus Bureau og Folketingets Forhandlinger.

"The Marketing Mix"

Serien omfatter følgende

10 programmer:

1. What is Marketing?
2. Marketing through Research?

3. Product Decisions

4. Buying and Selling

5. Using the Media

6. Corporate Image

7. Display and Exhibition

8. The Price Right

9. What is a Brand?

10. Marketing in action

Pris kr. **463,50** (VHS) eller kr. **543,00** (U-matic) pr. program.

MY VIDEO

SVARKUPON

Send mig straks mere materiale om:

- Edb-programmer til undervisning
- Skolernes databaseservice
- Landscentralens store edb-katalog
- Landscentralens databaseservice
- "The Marketing Mix"

Navn: _____

Evt. skole _____

Adresse: _____

Evt. telefonnummer _____

Landscentralen
for Undervisningsmidler

Landscentralen for Undervisningsmidler
Programafdelingen · Ørnevej 30 · 2400 København NV · Tlf.: 31 10 77 33



DE BLÅ SIDER

JNT-DATA
VIBORG

86-672071
86-674180

EDB I FOLKESKOLEN

DANSK

MATEMATIK

DATALÆRE

ENGELSK

VIKARTIME

TYSK

TRÆNING

FYSIK

VARIATION

IDRÆT

DISKETTER

DIFFERENTIERING

BOGFØRING

JNT - DATA FREMSTILLER, FORMIDLER OG SÆLGER UNDERVISNINGSPROGRAMMER. PROGRAMMERNE ER MENUSTYREDE OG DERFOR LETTE AT ANVENDE. DE ER APRØVET AF MANGE ELEVER OG LÆRERE. KRÆVER INGEN EDB INDSIGT. FINDES PÅ MERE END 250 SKOLER.

PROGRAMPRISER 200 KR. - 420 KR. (Excl. moms).

DISKETTER.

KVALITETSDISKETTER TIL PICCOLINE OG PARTNER M.FL.

ATHANA SORTE 105 KR. (ex. moms) for 10 stk.

ATHANA FARVEDE 115 KR. (ex. moms) for 10 stk.

ANDRE TYPER DISKETTER HAVES OGSÅ.

Hermed udbedes

___ stk. ATHANA disketter sorte. ___ stk. ATHANA disketter farvede.

Prisliste _____ Andet _____

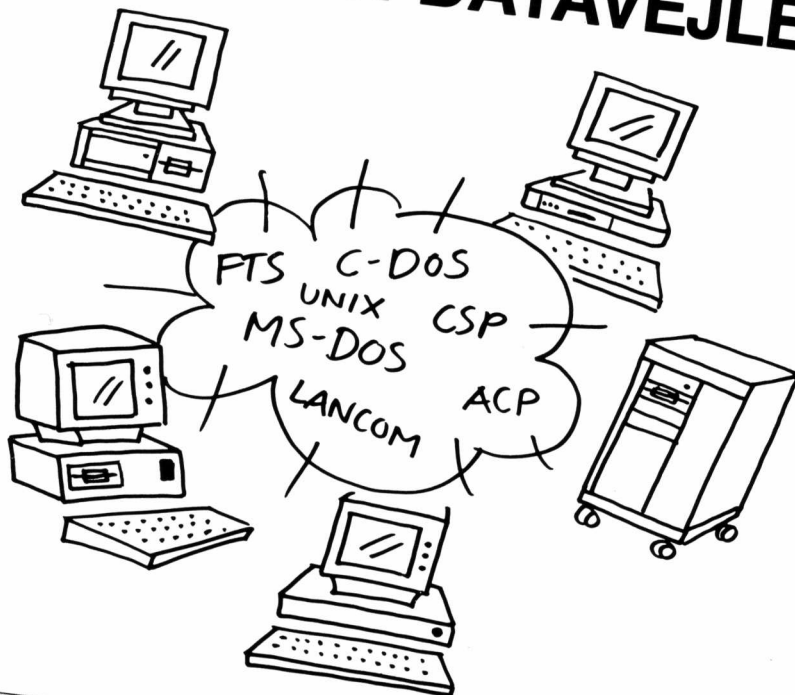
Afsender _____

Adresse _____

Skriv eller ring til

JNT-DATA, Amtmandshøjnen 14, 8800 Viborg, 86-672071 eller 86-674180

RC International & Øyvind Brandt dataservice INVITERER TIL SEMINAR FOR DATAVEJLEDERE



PROGRAM:

1. Introduktion til de nye produkter fra RC 900 serien
2. Eksempler på RC 900 brugt i undervisningen og administrationen
3. Piccoliner i fremtiden
MS-DOS programmer afviklet på Piccoline
UNIX programmer afviklet på Piccoline som terminalarbejdsplads
4. Godt tilbud!

Alle eksempler vil blive vist med RC970 og RC960 sammenkoblet med Piccoline i netværk

Onsdag den 11. oktober 1989
Amtscentralen i Esbjerg
Skolebakken 173
6705 Esbjerg
KL. 13.00-17.00

Onsdag den 25. oktober 1989
Sct. Knuds Gymnasium i Odense
Læssøgade 154
5230 Odense
KL. 12.00-16.00

Onsdag den 1. november 1989
RC International i København
Lyskær 9
2730 Herlev
KL. 12.00-16.00

Onsdag den 15. november 1989
Amtscentralen i Næstved
Troensevej 2
4700 Næstved
KL. 12.00-16.00

Onsdag den 6. december 1989
Amtsgymnasiet i Paderup, Randers
Apollovej
8900 Paderup
KL. 12.00-16.00

TILMELDING PÅ TELEFON 62 24 44 00

PICCOLINE, POLY-PASCAL OG MULTITASKING

Af Morten M. Christensen

PICCOLINE og multitasking

Et velkendt eksempel på Piccolinens muligheder under multitasking-operativsystemet CCP/M (eller C-DOS) er muligheden for at skifte mellem f.eks. en tekstbehandling og et regneark ved at trykke på en tast og derved skifte konsol.

Mindre velkendt er det, hvordan man som programmør på en forholdsvis simpel måde kan anvende multitasking, ved at designe programmer som drager nytte af operativsystemets muligheder. Almindeligvis er direkte brug af multitasking, multiprogrammering derfor en latent og ubenyttet mulighed.

Her i bladet har tidligere været omtalt, hvordan man kan starte andre processer ved hjælp af styresystemkaldet P_CLI, men vi vil i denne artikel se mere på de enkelte dele, som P_CLI er opbygget af.

Med artiklen følger en listning af et program skrevet i PolyPascal. Programmet er et illustrativt demonstrationsprogram, som viser multiprogrammeringens muligheder.

Inkluderet i demonstrationsprogrammet er en del rutiner, som også kan bruges i mange andre sammenhænge. Bl.a. to lydrutiner skrevet i maskinkode og en speciel errorrutine til Poly-Pascal.

Disse rutiner vil blive beskrevet senere. Først til hvordan multitasking foregår internt i CCP/M.

Multitasking internt

Et operativsystem med multitasking, som CCP/M i Piccoline-computerne, er designet til automatisk og parallelt at kunne administrere en række forskellige opgaver eller jobs - processer! D.v.s. udføre flere programmer samtidigt! Piccolinens hardware sætter den kun i stand til at udføre en ting ad gangen. Multitasking-effekt opnåes derfor ved en fin tidsdeling, hvor Piccolinen skiftevis arbejder med

forskellige processer. De forskellige processer afvikles derfor i teorien snarere sideløbende end samtidigt. I praksis er det anden sag, idet den fine tidsdeling, hvor computeren for Piccolinens vedkommende ca. 40 gange i sekundet skifter proces, gør illusionen, af at de udføres samtidigt, fuldkommen.

Tidsdelingen mellem de forskellige processer sker ved hjælp af en timer, som med et kort tidsinterval udsender et interrupt, et systemtick, som starter procesadministrationen. Her afgøres det, om den afbrudte proces skal have lov til at fortsætte, eller om en anden skal tage over. Afgørelsen foregår ved hjælp af en konstant ajourført og prioriteret liste, som indeholder oplysninger om status på alle processer.

Alle processer har en prioritet. Processen med den højeste prioritet vinder, d.v.s. får lov til at køre. Hvis to processer har samme prioritet, får den som har ventet længst kontrollen.

Hver proces består af program incl. data, herunder specielle datastrukturer som oprettes til at medvirke til administrationen af processen. To specielle datastrukturer indgår i en brugerproces af den type, som vi skal arbejde med. PD'en som på dansk kaldes procesbeskrivelsen og UDA'en, der kaldes brugerdata arealet.

Disse to datastrukturer er nødvendige, for at systemet efter hver systemtick kan gemme oplysninger om den igangværende proces. Således sørges for, at såvel processens status huskes, som at næste gang processen får kontrollen fort-

sættes fra nøjagtigt det punkt, hvor den i sin tid blev afbrudt og med de korrekte værdier i CPU'ens interne registre etc.

Demonstrationsprogrammet

Det listede demonstrationsprogram opretter en subproces, en datterproces, som uafhængigt af hovedprocessen spiller et stykke musik. Hovedprocessen spørger imens brugeren om stykkets navn. Enkelt måske, men alligevel ...

En proces kan skabe en subproces ved et enkelt systemkald.

Subprocessen deler hukommelse med den proces, som skabte den. At datterprocessen har samme lager som moderprocessen, gør kommunikationen mellem dem helt lettil. For Poly-Pascals

vedkommende betyder det, at subprocessen kan bruge alle hovedprogrammets globale variable og konstanter. Subprocessens programkode kan skrives i Pascal og placeres i en alm. procedure. Dette overflødiggør besværlige løsninger med køer og flags m.h.t. datakommunikationen, som ellers ville være nødvendig.

Lidt om typer

I programlistningens deklarerer en type kaldet »regs«. Denne type bruges i forbindelse med CCP/M system kald, og er et alternativ til Poly-Pascal manualens »regpacktype. Et alternativ, som tillader både 8 og 16 bit referencer til 80186'erens interne registre, hvor regpack typen kun tillader 16 bit referencer.

Den alternative type regs i programlistningen svarer til Turbo Pascals type »registers« og anvender en liden kendt syntax, en såkaldt recordvariant, som kort fortalt bl.a. gør det muligt for variable af forskellige typer at dele samme hukommelsesceller. Mere informa-

tion om recordvarianter kan findes i Poly-Pascal manualen.

Længere nede i listningen deklarerer til recordtyper, PDType og UDAType. PDType og UDAType udgør formatet for de to tidligere nævnte datastrukturer, som er procesbeskrivelsen og brugerdata arealet. Selve programmet til subprocessen findes i proceduren MultiProc. Proceduren Create_SubProces opretter datterprocessen.

For at en datterproces kan oprettes, kræves det, at dens tilhørende PD og UDA er initialiseret. Felterne i procesbeskrivelse og brugerdata arealet skal altså indeholde bestemte startværdier. De fleste kræves dog blot nulstillet. Jeg vil i det følgende gennemgå nogle vigtige felter i de to datastrukturer.

Procesbeskrivelsen

PDType

LINK-feltet er et kædefelt, som muliggør oprettelse af et multiplum af subprocesser. Hvis feltet indeholder en ikke-nul værdi, er feltet aktivt og peger på en anden subproces' procesbeskrivelse.

STAT-feltet oplyser om processens status. Værdien skal initialiseres til nul, som betyder at processen er klar.

PRIOR-feltet indeholder processens prioritet (0-255). Et almindeligt Pascal program har altid prioriteten 200. Datterprocessen skal tildeles samme prioritet, da den så vil skiftes med moderprocessen til at få lov til at køre. En hvilken som helst anden værdi vil enten bevirke, at moderprocessen overtager kontrollen helt (hvis PRIOR er større end 200), eller at datterprocessen helt overtager (hvis PRIOR er mindre end 200). Jo mindre PRIOR værdi, desto højere prioritet!

NAVN-feltet oplyser om processens navn, som fylder otte karakterer.

I demonstrationsprogrammet er subprocessen navngivet »MULT-PROC«.

UDA-feltet fortæller om subprocessens placering af UDA-datastrukturen. UDA-strukturens placering angives i antal paragraffer (antal 16 bytes) fra moderprocessens datasegments start, altså et slags offset angivet i paragraffer på UDA-datastrukturen. Det er dette felt, som har ansvaret for kravet om, at

```
(*          PROGRAM LISTNING : POLY-PASCAL MULTITASKING DEMO          *)
(*          Af Morten M. Christensen Juli 1989                          *)
Program MultiTask;
(*SK-,C-*)          (* Slå <Ctrl-C> mulighed fra samt deaktiver Stak- *)
                    (* og overflow check ved procedure kald. Vigtigt! *)

TYPE
Regs = Record      (* Definer speciel record til CPU-registre *)
  Case Integer Of (* Variant giver adgang til både 8 & 16 Bit reg. *)
    0: (AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags: Integer);
    1: (AL,AH,BL,BH,CL,CH,DL,DH: Byte);
  End;
PDType = RECORD    (* Record til Procesbeskrivelsen *)
  LINK: Integer;   (* 00 *)
  TRAD: Integer;   (* 02 *)
  STAT: Byte;      (* 04 *)
  PRIOR: Byte;     (* 05 *)
  FLAG: Integer;   (* 06 *)
  NAVN: Array (.1..8.) Of Char; (* 08 *)
  UDAPARAOFs: Integer; (* 16 *)
  DISK: Byte;      (* 18 *)
  BRUGER: Byte;    (* 19 *)
  RESEVERET1: Integer; (* 20 *)
  MEM: Integer;    (* 22 *)
  RESEVERET2: Array (.1..6.) Of Byte; (* 24 *)
  PARENT: Integer; (* 30 *)
  CNS: Byte;       (* 32 *)
  RESEVERET3: Array (.1..3.) Of Byte; (* 33 *)
  LIST: Byte;      (* 36 *)
  RESEVERET4: Byte; (* 37 *)
  SFLAG: Integer; (* 38 *)
  RESEVERET5: Array (.1..8.) Of Byte; (* 40 *)
END;
UDAType = RECORD   (* Record til brugerdata arealet *)
  RESEVERET1: Integer; (* 00 *)
  DMAOFs: Integer;    (* 02 *)
  DMASEG: Integer;    (* 04 *)
  RESEVERET2: Array (.1..26.) Of Byte; (* 06 *)
  AX: Integer;        (* 32 *)
  BX: Integer;        (* 34 *)
  CX: Integer;        (* 36 *)
  DX: Integer;        (* 38 *)
  DI: Integer;        (* 40 *)
  SI: Integer;        (* 42 *)
  BP: Integer;        (* 44 *)
  RESEVERET3: Array (.1..6.) Of Byte; (* 46 *)
  SP: Integer;        (* 52 *)
  RESEVERET4: Integer; (* 54 *)
  INTO: Array (.1..2.) Of Integer; (* 56 *)
  INT1: Array (.1..2.) Of Integer; (* 60 *)
  RESEVERET5: Array (.1..4.) Of Byte; (* 64 *)
  INT3: Array (.1..2.) Of Integer; (* 68 *)
  INT4: Array (.1..2.) Of Integer; (* 72 *)
  RESEVERET6: Array (.1..4.) Of Byte; (* 76 *)
  CS: Integer;        (* 80 *)
  DS: Integer;        (* 82 *)
  ES: Integer;        (* 84 *)
  SS: Integer;        (* 86 *)
  INT224: Array (.1..2.) Of Integer; (* 88 *)
  INT225: Array (.1..2.) Of Integer; (* 92 *)
  RESEVERET7: Array (.1..8.) Of Byte; (* 96 *)
  SYSTEMSTAK: Array (.1..76.) Of Integer; (*104 *)
  EXT8087: Array (.1..16.) Of Byte; (*256 *)
END;
```

UDA'en skal ligge på en paragrafadresse!

DISK- og BRUGER-feltet indeholder processens aktuelle disk og brugernummer.

CNS og LIST indeholder processens aktuelle konsol og skriver.

SFLAG-feltet afgør, om processen midlertidigt skal sættes i bero, hvis brugeren skifter processen over i en baggrundskonsol. Hvis SFLAG har værdien 1 suspenderes processen. Ellers ikke.

Brugerdata arealet

De to DMA felter i UDA'en bruges til at angive adressen på subprocessens DMA buffer. DMA'en anvendes til disksystemet. Hvis dette skal anvendes, skal 128 bytes reserveres til DMA. Der gøres ikke brug af en DMA i demonstrationsprogrammet.

AX,BX,CX,DX,DI,SI,BP,ES-felterne bruges til at gemme CPU-registre i, når processen suspenderes. Kan evt. anvendes til disses startværdi-

PICCOLINIEN

```

VAR
DATAAREAL : Array (.0..511.) Of Byte; (* Plads til PD & UDA *)
STAK : Array (.0..2047.) Of Integer; (* SubProces's Stak *)
MULTPROCOFS : Integer; (* Offset på SubProces's PD *)
(* SPECIELLE LYD RUTINER OG DATA TIL DEMONSTRATIONSBRUG AF SUBPROCESSEN *)
Procedure Volume (Kanal,Vol:Byte); (* Kanal=0..3, Vol=0..15 *)
Begin
Code
SFA, (* CLI *)
S8A,S56,S06, (* MOV DL,KanalÆBPÅ *)
SD0,SE2, (* SHL DL,1 *)
SFE,SC2, (* INC DL *)
SD0,SE2, (* SHL DL,1 *)
SD0,SE2, (* SHL DL,1 *)
SD0,SE2, (* SHL DL,1 *)
SD0,SE2, (* SHL DL,1 *)
SOA,S56,S04, (* OR DL,VolÆBPÅ *)
S80,SCA,S80, (* OR DL,10000000B *)
SBO,S0C, (* MOV AL,12 *)
SCD,S28, (* INT 28H *)
SFB; (* STI *)
End;
Procedure Frekvens (Kanal:Byte; Frkv:Integer); (* Kanal=0..2, Vol=0..15 *)
Begin
Code
SFA, (* CLI *)
S8B,S4E,S04, (* MOV CX,FrkvÆBPÅ *)
SD1,SE9, (* SHR CX,1 *)
S8A,S00,S00, (* MOV DX,0 *)
S8B,S12,S7A, (* MOV AX,31250 *)
SF7,SF1, (* DIV CX *)
S8B,SC8, (* MOV CX,AX *)
S8A,SF1, (* MOV DH,CL *)
S80,SE6,S0F, (* AND DH,00001111B *)
S8A,S56,S06, (* MOV DL,KanalÆBPÅ *)
SD0,SE2, (* SHL DL,1 *)
SD0,SE2, (* SHL DL,1 *)
SD0,SE2, (* SHL DL,1 *)
SD0,SE2, (* SHL DL,1 *)
SD0,SE2, (* SHL DL,1 *)
SOA,SD6, (* OR DL,DH *)
S80,SCA,S80, (* OR DL,10000000B *)
S51, (* PUSH CX *)
SBO,S0C, (* MOV AL,12 *)
SCD,S28, (* INT 28H *)
S59, (* POP CX *)
SD1,SE9, (* SHR CX,1 *)
SD1,SE9, (* SHR CX,1 *)
SD1,SE9, (* SHR CX,1 *)
SD1,SE9, (* SHR CX,1 *)
S8A,SD1, (* MOV DL,CL *)
S80,SE2,S3F, (* AND 00111111B *)
SBO,S0C, (* MOV AL,12 *)
SCD,S28, (* INT 28H *)
SFB; (* STI *)
End;
CONST
MeloLen = 48;
Melo: Array (.1..MeloLen.) Of Array (.1..2.) Of Integer =
((294,2),(392,2),(392,2),(392,2),(392,2),(392,4),(392,2),(392,2),
(440,2),(440,2),(370,2),(370,2),(392,4),(000,2),(294,2),(392,2),
(392,2),(392,2),(392,2),(392,4),(392,2),(392,2),(440,2),(440,2),
(370,2),(370,2),(392,4),(000,2),(494,2),(492,2),(494,2),(494,2),
(494,2),(492,2),(394,2),(294,4),(392,4),(440,4),(494,2),(392,2),
(392,2),(392,2),(440,2),(440,2),(370,2),(370,2),(392,4),(000,2));
VAR
Hast : Integer;

```

er fra opstart eller kan blot nulstilles. De resterende 5 felter til CPU-register skal sættes til bestemte startværdier! CS og DS skal sættes til subprocessens kode og datasegment. Da datterprocessen deler lager med moderprocessen, skal disse sættes til moderprocessens værdier, som fra Poly-Pascal kan hentes med CSEG og DSEG funktionerne.

Stakken

SS-Feltet skal sammen med SP-feltet sættes til subprocessens stak. Datterprocessens stak kan ikke være den samme som moderprocessens! I stedet må stakken oprettes andetsteds (feltet SYSTEMSTAK bruges internt af CCP/M og kan ikke anvendes til dette formål!). I demonstrationsprogrammet deklareres et array, som reserverer 4K hukommelse i det fælles datasegment til subprocessens stak.

Datterprocessens staksegment falder derfor, modsat modersegmentet, sammen med datasegmentet! SP-feltet skal indeholde et offset fra starten af subprocessens staksegment (som altså er sammenfaldende med datasegmentet) til et sted i stakområdet. Stakområdet skal på dette sted indeholde 2 words med segment og offset på subprocessens programkode (hvilket er henholdsvis kodesegmentet og offset'et på MultiProc-proceduren), samt et enkelt word til CPU status-registret (sættes bare til nul). Denne initierede stak betyder, at CCP/M kan starte processen ved en simpel IRET instruktion.

Det skal lige nævnes, at da 80186 processorens stak vokser nedad (for hver anbringelse af data på stakken formindskes stakpointeren SP), er det hensigtsmæssigt at placere de tre words til initieringen i den højeste ende af stakområdet, d.v.s. i de 3 sidste elementer af stakarrayet.

Proceduren CREATE_SUBPROCES

Som nævnt ovenfor er det proceduren Create_SubProces, som står for selve oprettelsen af datterprocessen. Proceduren starter med at nulstille subprocessens stak samt det areal af hukommelsen, hvor procesbeskrivelsen og brugerdata arealet skal placeres.

Placeringen foregår først ved runtime. To pointerne af typerne PDType og UDATyp sættes til at pege på paragrafadresser i et reserveret areal af hukommelsen. Almindelige variable kan ikke sættes til at starte ved en ny paragraf, som UDA'en kræver. Derfor denne fremgangsmåde med pointerne ved runtime.

Dernæst sættes de af felterne i procesbeskrivelsen og brugerdata arealet, som skal have en startværdi. Da alle felter i starten blev nulstillet behøves ikke yderligere opsætning. Til sidst foretages et systemkald, P_CREATE, som ordner resten. Systemkaldet kræver at CPU registre DS og DX sættes til henholdsvis segment og offset på datterprocessens procesbeskrivelse. Efter kaldet bruges den initierede procesbeskrivelse ikke mere, da den kopieres op i en intern systemliste af ubrugte procesbeskrivelser.

Proceduren MULTPROC

MultiProc proceduren indeholder programmet til den nye datterproces.

Det første proceduren gør, er at gemme offset på procesbeskrivelsen i den interne liste af procesbeskrivelser. Når moderprocessen senere vil afbryde datterprocessen, skal dette offset nemlig specificeres.

Herefter følger den egentlige programkode for subprocessen, som henter nogle toner fra en konstant og spiller dem efter tur. Det skal bemærkes, at rutinen er bygget op omkring en Repeat-Until, som aldrig færdiggøres. Hvis computeren nåede til slutningen på proceduren ville den ikke ane hvad den skulle gøre - derfor!

Til at lave lyden har jeg anvendt to procedurer nemlig Volume og Frekvens skrevet in-line i 80186 maskinkode. Rutinerne producer lyden ved hjælp af Piccolines specielle Int 28H,12 rutine.

De to procedurer i listningen er vel-dokumenterede og specielle oplysninger om lyd på Piccoline kan findes i Programmer's Guide udgivet af RC International. Procedureerne er ikke specielt til brug i multitasking sammenhænge og kan tages ud og anvendes andetsteds.

Normalt vil Poly-Pascal foretage stak-check før kald af procedure og funktioner, men da subprocessens stak befinder sig et helt andet sted end Poly-Pascal forventer, vil et sådant check rapportere en fejl, hvor der ikke er nogen. Heldigvis kan stak-check slås fra ved et kompilerdirektiv (K-) og det gøres øverst i listningen, hvor også <Ctrl-C> slås fra, for at undgå at programmet afbrydes u hensigtsmæssigt. Det er nemlig vigtigt, at subprocessen fjernes, før hovedprogrammet afsluttes eller afbrydes.

Nogle begrænsninger

Der eksisterer bestemte begrænsninger for, hvad en datterproces skrevet som en pascal-procedure må beskæftige sig med. Poly-Pascal sætter begrænsninger, idet kald til I/O standard procedurer/funktioner (bl.a. WRITE og READ), brug af real variabler samt WITH-sætninger ikke må anvendes i både moder- og datterproces. Yderligere

```

Procedure MultiProc; (* Proceduren det hele handler om. *)
VAR
  Reg: Regs;
  NodeNr, Vol, Vent, Pause: Integer;

Begin
  Reg.CL:=156; (* P_DADR : Hent adressen på SubProcessens PD *)
  Swint(224, Reg);
  MultiProcOfs:=Reg.AX; (* Gem offset, så at SubProcessen senere kan iden- *)
  (* tificeres og afbrydes af ReMove_SubProces. *)
  (* Spil musik - *)
  Repeat
    NodeNr:=1;
    Repeat
      If Melo(.NodeNr,1.)>0 Then
        For Vol:=0 To 15 Do
          Begin
            Volume(0, Vol);
            Frekvens(0, Melo(.NodeNr,1.));
            For Vent:=1 To Melo(.NodeNr,2.)*Hast Do;
          End
        Else
          For Pause:=0 To 15 Do
            For Vent:=1 To Melo(.NodeNr,2.)*Hast Do;
          NodeNr:=NodeNr+1;
        Until NodeNr>MeloLen;
      Until False; (* Genstarter musik, når aldrig til "end;" !!! *)
    End;
  Const
    EndOfMultiProc : Byte = 0; (* Marker slutning af MultiProc *)

Procedure Create_SubProces; (* Opretter SubProcessen MultiProc *)
Var
  PD : ÜPDType; (* Pointere til procesbeskrivelsen *)
  PDPoint : Integer;
  UDA : ÜUDAType; (* Pointere til brugerdata arealet *)
  UDAPoint: Integer;
  Reg: Regs;
  Lp: Integer;

Begin
  (* Nulstil lager til PD, UDA og stak *)
  For Lp:=0 To 511 Do DATAAREAL(.Lp.):=0;
  For Lp:=0 To 2047 Do STAK(.Lp.):=0;

  (* Sæt pointere til paragraf-memory *)
  PDPoint:=0;
  While ((PDPoint+Ofs(DATAAREAL)) Mod 16)<>0 Do PDPoint:=PDPoint+1;
  UDAPoint:=PDPoint+Size(PDType);

  PD:=PTR(Seg(DATAAREAL), Ofs(DATAAREAL)+PDPoint);
  UDA:=PTR(Seg(DATAAREAL), Ofs(DATAAREAL)+UDAPoint);

  (* Initialiser procesbeskrivelsen *)
  PDÜ.PRIOR:=200; (* Start prioritet *)
  PDÜ.NAVN:='MULTPROC'; (* SubProcess's navn *)
  PDÜ.UDAPARAOfs:=(Ofs(UDAÜ) Div 16); (* UDA's paragraf-placering i DSeg *)

  (* Initialiser brugerdata arealet *)
  UDAÜ.DS:=DSeg; (* Data segment *)
  UDAÜ.CS:=CSeg; (* Kode segment *)
  UDAÜ.SS:=DSeg; (* Stak segment *)
  UDAÜ.SP:=Ofs(STAK)+2*2045; (* Startværdi for Stakpointer *)

  (* Initialiser bruger-Stak til at pege på SubProces-program *)
  Stak(.2045.):=Ofs(MULTPROC); (* Program offset *)
  Stak(.2046.):=CSeg; (* Program segment *)
  Stak(.2047.):=512; (* CPU Flags *)

  (* Opret SubProces *)
  Reg.CL:=144; (* P_CREATE : Skab subprocess *)
  Reg.DX:=Ofs(PDÜ); (* Offset på procesbeskrivelse *)
  Reg.DS:=Seg(PDÜ); (* Segment på procesbeskrivelse *)
  Swint(224, Reg);
End;

Procedure ReMove_SubProces; (* Fjerner, sletter SubProces *)
Var
  APB : RECORD (* APB - Abort Parameter Block *)
    PD: Integer; (* 0 *)
    TERM: Integer; (* 2 *)
    CNS: Byte; (* 4 *)
    NAVN: Array (.1..8.) Of Char; (* 5 *)
  END;
  Reg : Regs;

Begin
  APB.PD:=MULTPROCOfs; (* Informer om offsetadresse på procesbeskrivelse *)
  APB.TERM:=0; (* Termkode=0 : Kun brugerprocesser kan afbrydes *)
  Reg.CL:=157; (* P Abort : Afbryd proces *)
  Reg.DX:=Ofs(APB); (* DX = Offset for Abort Parameter Block *)
  Reg.DS:=Seg(APB); (* DS = Segment for Abort Parameter Block *)
  Swint(224, Reg);
End;

```



```

Procedure Error(Errnr,ErrOfs: Integer); (* Error rutine, kaldes af Poly- *)
                                         (* Pascal hvis der opstår en fejl. *)
Var
  Reg: Regs;
  ErrInSubP: Boolean; (* True = fejl i subprocess, False = fejl i hovedprog. *)
Begin
  ErrInSubP:=((ErrOfs>=Ofs(MultProc)) And (ErrOfs<Ofs(EndOfMultProc)));
  If ErrInSubP Then
    Begin
      (* Error fundet i SubProces *)
      Reg.CL:=143;      (* P_TERM : Afslut SubProces *)
      Reg.DL:=0;       (* Termkode=0 : Kun brugerprocesser kan afbrydes *)
      Swint(224,Reg);
    End;
  End; (* Ingen aktion, hvis Error er opstået i hovedprogram *)

Var
  St : String(.40.);
  ThatsRight: Boolean;
  SvarNr: Byte;
Begin
  EhOfs:=Ofs(Error); (* Aktiver error handler ved angivelse af offset *)
  Writeln;
  Write('Hvilken hastighed (ca. 250 anbefales) ? ');
  Hast:=250;
  Readln(Hast);
  Create_SubProces; (* !!! *)
  Writeln;
  Writeln ('Du hører netop nu et stykke musik. ');
  SvarNr:=1;
  Repeat
    Write ('Hvad hedder stykket ? ');
    Readln(St);
    ThatsRight:=((St='EN PIGE GIK I ENGE') Or (St='en pige gik i enge'));
    If Not ThatsRight Then Writeln('Forkert svar !!!');
    SvarNr:=SvarNr+1;
  Until ThatsRight Or (SvarNr>3);
  Remove_SubProces; (* !!! *)
  Volume(0,15);
End.

```

information kan hentes i Poly-Pascal manualen.

CCP/M sætter yderligere en begrænsning, da to processer ikke (ad »lovlig« vej) begge kan skrive på skærmen eller printer. Hvis noget skal skrives på skærmen fra subprocessen, må det gøres på »ulovlig« vis, f.eks. ved at skrive direkte til skærmhukommelsen (se Programmers Guide).

Hvordan MULTPROC stoppes

Hvordan en proces stoppes, afhænger af hvem der beder den om at stoppe!

Hvis en proces ønsker at fjerne en anden proces, så kan det gøres ved CCP/M kaldet P_Abort. Til P_Abort kaldet skal der i registre DS og DX, anføres adressen på en såkaldt abort parameter blok, som fortæller, hvilken proces man ønsker afbrudt. Desuden skal sættes en såkaldt termkode, som afgør hvilke slags processer, som må kunne afbrydes. Sættes termkoden til nul kan kun normale brugerprocesser afbrydes.

Processen, der ønskes fjernet med P_Abort, kan specificeres på forskellige måder. En måde er at angive offset'et på proces-beskrivelsen i den interne trådliste. Det er denne metode, som anvendes i demonstrationsprogrammet.

Hvis den pågældende proces selv ønsker sig fjernet (afsluttet), gøres dette ved CCP/M-kaldet P_Term. I det tilfælde er det mere enkelt, da ingen speciel parameterblok skal bruges og kun termkoden skal sættes.

En error og errorhandler

I demonstrationsprogrammet er inkluderet en errorhandler, som kaldes, hvis der opstår en fejl. Errorhandleren sikrer, at hvis fejlen er opstået i MultProc, så fjernes datterprocessen. Hvis datterprocessen ikke blev fjernet i tilfælde af fejl, vil den så at sige løbe af sporet og evt. få maskinen til at »crashe«.

Hvis en fejl opstår, vil ErrNr være sat til fejltype og fejlNr. og ErrOfs til offset'et på fejlen. Errorhandleren i demonstrationsprogrammet bruger ErrOfs til at regne ud, om fejlen

er sket i MultProc-proceduren. Aktivering af errorhandleren sker ved at sætte standardvariablen EHOFS til offset'et på errorrutinen. Errorhandlers i Poly-Pascal bruges ikke specielt til multiprogrammering, men også i mange andre sammenhænge. Mere om teknikken bag Poly-Pascal errorhandlers kan findes i dokumentationen i readme-filen til Poly-Pascal. For at kunne bruge errorhandlers skal din Poly-Pascal version være 3.10 eller nyere.

Prøv selv

Færdig med beskrivelsen af teknikken bag demonstrationsprogrammet.

Jeg håber, at denne artikel vil give dig mod på selv at prøve multiprogrammering i Poly-Pascal. Meget virker mere effektivt eller er kun muligt via multitasking.

Opskriften er at lade demonstrationseksemplet, slette lydrutinerne, lyddata og mit hovedprogram, d.v.s. linjerne mellem den sidste »begin-end.«, og derefter at programmere en ny MultiProc procedure. Tilbage står da kun at kæde det hele sammen med dit eget hovedprogram.

Til sidst skal siges, at hvis artiklen har givet dig lyst til at studere CCP/M og multitasking yderligere, kan mere information (på dansk) findes i Gert Hamachers bog »Forstå CCP/M«.

Det er nu det sner! – eller »Regn«er!

Tøbrud for anvendelse af regneark i mange fag.

Af Arne Mogensen og Jørgen H. Christiansen

Regneark er en programtype, der anvendes i større og større udstrækning i flere og flere af skolens fag, og mange lærere landet over er netop nu i gang med at udvikle undervisningsmaterialer til regneark.

I det sidste nummer af bladet PS.NYT (7) fra Projektsamvirket var halvdelen af stoffet artikler om anvendelse af regneark, og PS er blot en af de mange grupper, der tror på, at regnearket nu er ved at slå igennem.

Et regneark er et edb-program, der f.eks. kan bringes til at fungere på samme måde som et budgetschema fra en bank eller sparekasse.

Regnearket er opdelt i lodrette søjler og vandrette linjer som på tegningen herunder.

De enkelte felter i regnearket kaldes celler, og de får navn efter den søjle og den linje, hvor i de befinder sig.

F.eks. C9. = Cellen i søjle C, linje 9. Som det fremgår af fig. 1 er det kun en lille del af regnearket, der er på skærmen på en gang.

Alle regnearkets celler kan enten indeholde tekster, tal eller formler, f.eks. kan nogle celler være fyldt ud som herunder:

	A	B
1		
2	OPGAVE 1	
3	234	
4	615	
5	_____	
6	(A3+A4)	

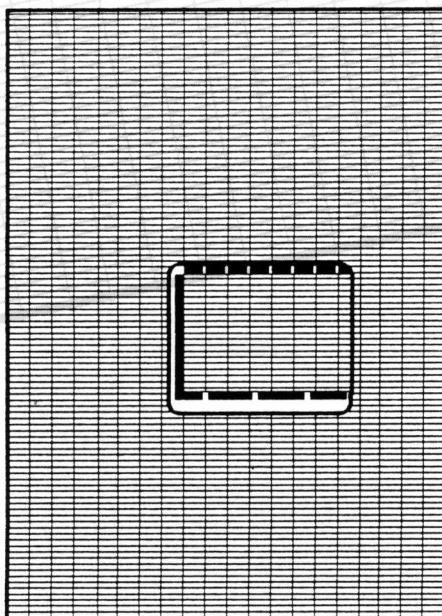
Men skærbilledet viser dette billede:

	A	B
1		
2	OPGAVE 1	
3	234.000	
4	615.000	
5	_____	
6	849.000	

Hvis man nu skriver et andet tal i en af de to celler A3 eller A4, vil facit i celle A6 også ændres.

Netop fordi det er så fleksibelt og enkelt at arbejde med, er regnearket et værktøjsprogram, der er umiddelbart anvendeligt i mange fag.

Fig. 1



Hvorfor anvende regneark?

Nedenstående er blot nogle af de mange argumenter for anvendelse af regneark (med grafik!).

Der er mere at læse om emnet og om anvendelse af regneark i forskellige fag i bogen Vejledning til Regn2.

- Regnearket er et generelt værktøjsprogram, og brugen af det er derfor ikke knyttet til et bestemt fag. Det er tværtimod et redskab, der er velegnet i et tværfagligt samarbejde.
- Regnearket gør det muligt at arbejde med nye faglige områder, som ellers vanskeligt kan behandles i undervisningen.

Ovenstående kan nok synes som postulater, men vi håber gennem eksemplerne at påvise, at de er rigtige.

- Fremkomsten af et nyt redskab til undervisningen giver anledning til revurdering af fagenes indhold og metoder.

F.eks. afskaffede man for år tilbage trigonometrien i folkeskolens matematikundervisning. Med lommeregnerne og regneark til rådighed i undervisningen er det værd at overveje, om man skal genindføre den!

- Regneark opfordrer til eksperimenter med tallene og det grafiske layout, fordi elevernes arbejdsindsats ikke er knyttet til de deraf følgende udregninger og tegninger.

Det giver bedre muligheder for at udforske et område, og undervisningen kan i højere grad lægge op til en diskussion om emnet mellem elever og lærer.

- Regneark er et modelredskab, hvor man ofte på en meget enkel måde er i stand til at anskueliggøre og dynamisk demonstrere vanskelige sammenhænge og ændre i forudsætninger og formler.

En model i et regneark er helt gennemsigtig – i modsætning til de modeller, som man ellers møder.

- Regneark med grafik er et avanceret medium, der gør det muligt for eleverne på meget kort tid at kunne fremstille professionelt udseende grafer, som dem de er vant til at se i de øvrige medier.

Ingen andre redskaber er i den grad i stand til at håndtere store datamængder – for kort efter at kunne vise den afgørende detalje, f.eks. i form af en graf.

- At arbejde med regneark er programmering. Men det er en helt anden, enklere og mere relevant programmeringsmåde end den, der kræver kendskab til et særligt programmeringssprog.

Anvendelse af regneark i undervisningen

I det følgende er anvendelsen belyst ud fra nogle få af de muligheder, som regnearket tilbyder:

- Regnearket som database
- Modelanvendelsen
- De grafiske muligheder.

Et hav af data findes omkring os og venter blot på at blive målt, organiseret i et regneark og derefter taget op til en nærmere behandling/diskussion på klassen.

Blandt fordelene ved at anvende regneark som en database kan nævnes:

- man får et godt overblik over materialet.
- det er let at foretage sammenligninger og beregninger.
- data kan umiddelbart illustreres grafisk.

Klassens egen database:

Et regneark organiseres som database indeholdende oplysninger om klassens elever.

Højde, vægt, skønnummer, afstand til skolen, alder målt i dage, lomme penge, antal søskende, antal kæledyr, antal rigtige matematikopgaver!! osv, osv.

Nogle af de aldersbestemte personlige data kunne måske danne udgangspunkt for selvstændige regneark, hvor man så kan følge udviklingen.

Databaser over aktuelle emner i tiden:

Mange af de fakta/data, som omgiver os i det moderne samfund er netop karakteristiske derved, at de er under stor forandring. Der kan derfor være flere pædagogiske gevinster ved at arbejde med data af denne type. Ud over de rent matematiske oplevelser, vil eleverne også få et indtryk af at leve i en meget

dynamisk tid i et samfund under stadig forandring.

Mange ting forandrer sig jo nok desværre i disse år i en negativ retning, men specielt på nogle miljømæssige områder kan der være håb om forbedringer, og de data, der er lavet af en klasse i et år, kan derved fint anvendes af andre klasser i efterfølgende år.

Som eksempler på data af denne type kan nævnes:

- Miljøobservationer
- Energiforbrug
- Produktionsdata
- Trafiktællinger
- Priser og kurser
- Vejrobservationer

Flere af de nævnte typer af data findes allerede i form af forskellige statistiske værker, hvorfra de kan indlæses i et regneark for evt. at blive sammenlignet med lokale/aktuelle data.

Andre databaser findes på diskette – klar til brug.

Det gælder f.eks. basen Ressourceoversigter fra Aschehoughs Forlag.

Modeller

Et af de steder, hvor man kan forvente en udbredt anvendelse af regneark, er området modeller.

Det skyldes, at mange matematiske sammenhænge er velegnede til at blive beskrevet ved en eller flere forholdsvis enkle formler, og herved får man en dynamisk model.

Meget ofte er sammenhængene så simple, at det kan være svært at se, at der virkelig er tale om en model. Tag f.eks. modellen for befolkningsudvikling:

I sin grundlæggende form kan den udtrykkes på følgende enkelte vis:

$$\begin{aligned} \text{Befolkning} &= \text{Nuv. befolkning} \\ &+ (\text{Fødte} + \text{Indvandring}) \\ &- (\text{Døde} + \text{Udvandring}) \end{aligned}$$

hvor de variable størrelser igen kan være simple matematiske udtryk. – Antal fødte kan f.eks. udtrykkes ved fødselspromille.

Andre eksempler på anvendelse af modeller, der allerede finder sted, har vi i enkle computerspil, hvor eleverne på grundlag af nogle oplysninger skal tage beslutninger, som så igen får konsekvenser for det videre forløb i spillet.

Disse spil bygger ofte på meget enkle matematiske formler, og vil lige så let kunne laves i et regneark,

med den ekstra fordel, at det bliver klart for eleverne, hvad det er, der foregår. Programmer som FAST FOOD, Säftevandsboden og Diktator er af denne type.

Utroligt mange sammenhænge lader sig omskrive til matematiske modeller, og muligheden for at vildlede/indoktrinere brugeren er omvendt proportional med det anvendte systems gennemskuelighed. Tænk blot på de økonomiske modeller, der anvendes af de politiske partier....

Indenfor mange forskellige fagområder – f.eks. indenfor landbruget – anvendes helt naturligt modeller i den daglige planlægning, og mange af disse modeller vil, evt. i en lettere modificeret form, kunne anvendes i undervisningen.

Det gælder blot om at få fundet og identificeret modellerne.

Udnyttelse af de grafiske fordele

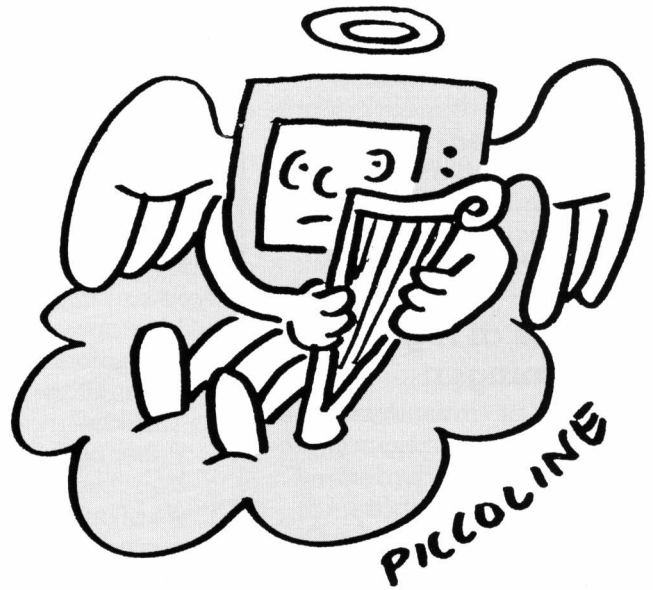
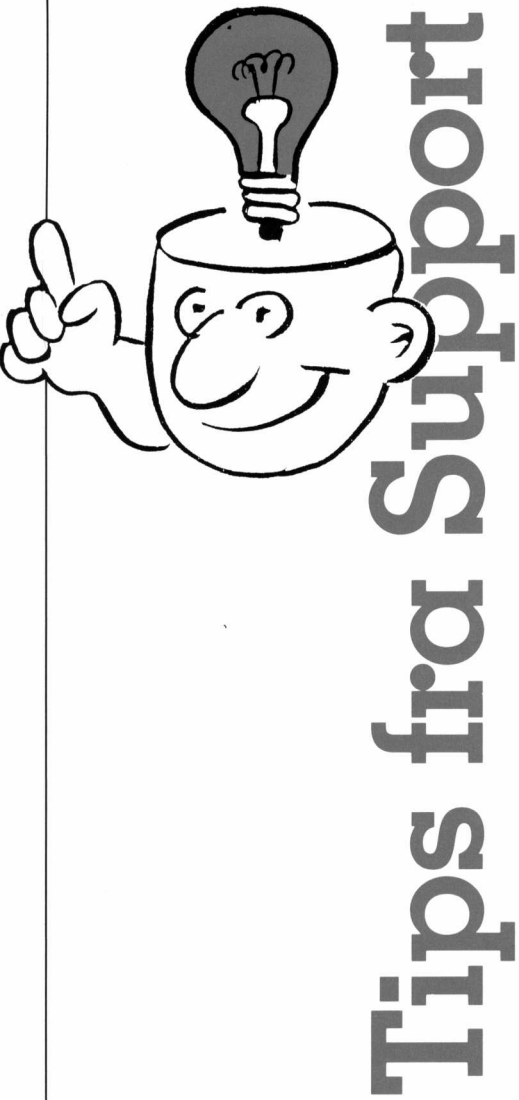
Specielt ved arbejde med meget store og meget små tal kan det være vanskeligt for eleverne at danne sig et indtryk af de indbyrdes relationer mellem tallene, og derved går vigtige oplysninger tabt, og elevens udbytte af at arbejde med den pågældende opgave reduceres til kun at bestå i at udføre de »sædvanlige operationer« – bare med nogle andre tal. Med regnearket har man den store fordel, at de indbyrdes relationer ofte kan ses umiddelbart af den grafiske repræsentation.

Som eksempler kan nævnes arbejde med astronomiske data, hvor f.eks. solsystemets planeters indbyrdes størrelse kan afbildes i et søjlediagram.

Dette gælder også en lang række andre statistiske data, og specielt hvor der er tale om en udvikling, vil denne træde tydeligt frem på en kurve over forløbet.

Også når det gælder om at sammenholde et sæt data med et andet, er grafikken til stor hjælp. F.eks. når man vil sammenholde lokale data med globale data, eller aktuelle data med en given norm.

Artiklens forfattere er ophavs-mænd til regnearket Regn2, som kan anvendes både med Piccoline/Partner og med RC900-serien fra RC International.



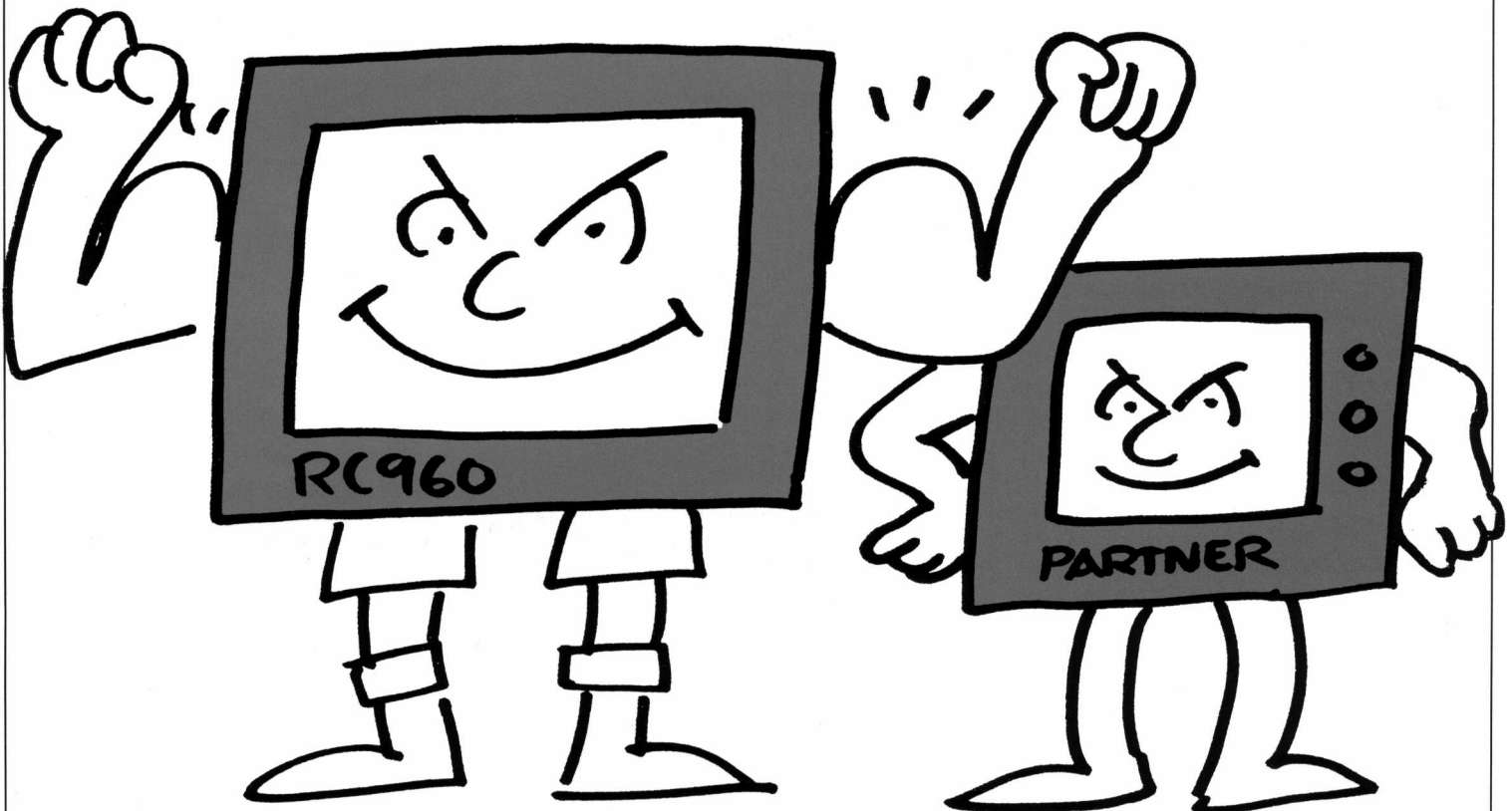
Sidste PICCOLINE!

Tirsdag den 30/8/89 blev faktisk en lidt vemodig dag for alle tilknyttet skoleafdelingen på RC International.

Det blev den dag, da vi for første gang ringede til vores leveringsafdeling for at bestille en PICCOLINE, og fik svaret »desværre, den sidste er netop solgt«.

Hvad var så baggrunden for det? Introduktionen af RC960/RC970 produkterne i sidste nummer af PICCOLINIEN havde skabt så meget interesse, at vi mente man måtte gøre noget for at sælge de ganske mange PICCOLINE, som vi stadig kunne producere.

Vi introducerede derfor et høst-tilbud på disse. Dvs, vi nåede dårlig



PICCOLINIEN

nok at gøre det, før rygterne havde spredt sig, så maskinerne var solgt. Ca 100 maskiner på en uge - det gik stærkt.

Hvad er så fremtiden for PICCOLINE-installationerne, og baggrunden for at stoppe videre produktion af denne maskine.

Fremtiden er lys. Dels gennem de muligheder for sammenkobling, som præsenteres andetsteds i dette blad. Dels fordi Partner jo er kompatibel med PICCOLINE, så denne maskine kan indgå i PICCOLINE installationer fremover.

Det er netop dette, der er baggrunden for at stoppe videre produktion af PICCOLINE. Såvel Partner som PICCOLINE har efterhånden ca 6 år på bagen, og da der sidenhen er kommet mange nye produkter i RC Internationals produktlinie, syntes vi, at tiden var inde til at reducere, så vi kun videreførte et af de to kompatible produkter. Vi valgte så Partner - dels fordi denne maskine også kan leveres med harddisk, og dermed er velegnet som f.eks. vært i et netværk, og dels fordi det er den maskine, der sælges både til skoler og til andre offentlige og private institutioner. PICCOLINE derimod var et rent skoleprodukt.

Så der er stadig mulighed for at udbygge sin maskinbestand, både med PICCOLINE-kompatible produkter, og med andre produkter fra RC International.

C-DOS 6.0 til Partner

Der er blevet introduceret en ny version af C-DOS til Partner.

Version 6.0 adskiller sig markant fra version 5.0, ved at man ikke længere understøtter CCP/M-formatet på disketter, men kun C-DOS (AT) formatet.

Version 6.0 er forbedret på en række områder, og indeholder bl.a. version 2.0 af DR-NET, således at den er kompatibel begge veje med den version af DR-NET, der benyttes under CDOS386 på RC900-maskiner.

Da C-DOS version 6.0 adskiller sig så markant fra tidligere versioner, er der ikke tale om en opdatering fra 5.0 eller 5.2, men om et selvstændigt produkt.

PICCOLINE/PARTNER STANDARD PROGRAMMEL

Herunder er vist et udvalg af de standardprogrammer, som Regnecentralen udbyder til PICCOLINE / Partner. Listen er begrænset til rene PICCOLINE programmer (salgsnr. SW1400-SW1499) og enkelte Partner programmer (salgsnr. SW1500-SW1599), som vi antager har særlig interesse for undervisningssektoren.

Opdatering fra ældre release til nyeste foregår ved at indsende sine originale disketter til:

RC International
att: opdatering
Rosagervej 15
4720 Præstø

Salgsnr.	Betegnelse	Rel.	Ændret	Pris
SW1400	Styresystem CCP/M	3.1a		-
	RcComal80	2.1a		-
SW1402	PolyPascal 3.11	1.2		4.995
SW1403	RcKalk	1.3		1.450
SW1404	ACP750	5.1		2.000
SW1426	RcTeledata	2.0		1.950
SW1433	RcTekst II	3.2		4.800
SW1435	RcFont	1.3		1.285
SW1448	GEM Collection	1.0	*)	1.850
SW1452	RcTekst II, ACP750 & Mikrologo	2.1	*)	2.200
SW1458	Concurrent DOS	5.0		650
SW1495	Myresnak	1.2		500
SW1499	Mikrologo	1.2		1.000
SW1500	Styresystem C.DOS	5.2		-
SW1500.6	Styresystem C.DOS	6.0	ja	3.275
SW1501	RcComal80	3.0		1.800
SW1502	PolyPascal 3.11	1.2		4.995
SW1503	RcKalk	1.3		1.450
SW1504	ACP750	5.1		2.000
SW1520	GraphPlan 1.30	1.1		2.450
SW1523	FilePlan 2.2DA	1.1		2.450
SW1544	VISTA database	1.1	*)	4.800
SW1545	VISTA programmering	1.1	*)	4.600
SW1548	GEM Collection	2.0	*)	1.850
SW1549	GEM Draw	1.0	+))	2.950
SW1550	GEM WordChart	1.0	+))	1.850
SW1551	GEM Graph	1.0	+))	2.750

*) Kræver Concurrent DOS

+) Kræver GEM Collection

Priser er listepreiser ekskl. moms. Opdatering, hvor tal efter decimalpunkt ændres, f.eks. fra rel. 1.1 til 1.2, koster normalt intet. Ændres tallet før decimalpunkt, er prisen kr. 250. I begge tilfælde kan der være fremstillet nye manualer.. Prisen for disse skal i så fald tillægges opdateringsprisen.

Listen er afsluttet den 10/9 1989.

PICCOLINIEN

Vor mand i Havanna



Efter veloverstået arbejde er der intet som at slappe af i skyggen af palmerne på Cuba's brede strande. Sol, strand, vand - og PICCOLINIEN i hånden.

Udgiver:

RC International

Indlæg fra læserne:

Skal sendes til nedenstående adresse:

Redaktion:

Mogens Guildal (ansv.)
Ole Schwander Olsen
Lautrupbjerg 1
2750 Ballerup
Tel.: 42 65 80 00

Salgsafdeling:

Lautrupbjerg 1
2750 Ballerup
Tel.: 42 65 80 00
henviser til nærmeste
forhandler

Supportcenter:

Hovedvejen 9
Tel.: 42 96 52 00
Telefax: 42 96 78 66

Teknisk service:

Glostrup
Hovedvejen 3C
2600 Glostrup
Tel.: 42 96 07 00

Århus
Klamsagervej 19
8230 Åbyhøj
Tel.: 86 25 04 11

Aalborg
Limfjordsvej 14
9400 Nørresundby
Tel.: 98 17 80 44

Odense
Lumbyevej 11
5000 Odense C
Tel.: 66 18 78 15

Grafisk tilrettelægning:

PZ Design

Tryk:

Johnsen+Johnsen a/s,
København