

R

C B N

ORGAN FOR COMAL BRUGER GRUPPE DANMARK

MAJ 1984 - NR. 5.

EN SPÆNDENDE TID

For den, der interesserer sig for COMAL, forestår der en spændende tid.

Der er udvikling i sprogets udformning.

Udvalg er nedsat til at styre denne og helst standardisere fremskridtene.

Men der er også udvikling i sprogets udbredelse.

Det er en vittighed, at COMAL er verdensberømt i Danmark.

Adskillige tiltag tyder på, at man mange steder uden for COMAL's fædreland er ved at få øje på dette sprogs fortrin frem for andre sprog på samme niveau.

Medvirkende til at styrke denne tendens er formodentlig, at COMAL kan fås på styresystemet CP/M, der som bekendt er vidt udbredt.

En anden årsag dertil er COMMODORE's løsning med en COMAL cartridge - foreløbig til COMMODORE 64.

Med det antal, der er solgt af denne maskine, skulle der være gode chancer for, at COMAL skulle kunne blive allemandseje - også i udlandet.

Ikke mindst i skolerne - og jo da specielt: de danske - er COMAL vidt udbredt - ja, vel stort set enerådende EDB-sprog.

Og hvilken betydning det vil få for sprogets almindelige udbredelse er næppe til at overskue - ganske særlig ikke, når datalære (informatik, eller hvad det skal hedde) nu bliver godkendt som skolefag - først i de ældre klasser og siden i de yngre - først frivilligt og siden obligatorisk.

Et sikkert fremtidsperspektiv, der allerede er en realitet:

I skolen lærer børnene EDB på mikrodatamater - i hjemmet over de sig på hjemmedatamater - begge dele baseret på COMAL.

Jo, COMAL har vind i sejlene!

INDHOLDSFORTEGNELSE

Leder: En spændende tid	1
Anmeldelse	2
VenteprocEDURE	3
Simpelt register	4
Spørgeskemaet	7, 10, 12
Russisk roulet	11
Delmåneder	13

Universitetsbiblioteket, 2. afd.

0108-4925

200002999586



THE DANISH GAZETTE FOR COMAL LANGUAGE

COMAL BRUGER NYT

ANMELDELSE

COMMODORE 64 COMAL-80
Diskette +
Håndbog skrevet af
Børge R. Christensen
COMMODORE DATA A/S
Kr. 430 incl. moms.

Som nævnt i CBN 1983/3 tilbydes stadig flere datamater med COMAL som programmeringssprog.

For nylig har vi fået stillet et eksemplar af COMMODORE 64 til bladets rådighed.

Maskinen er udbygget med farvemonitor, diskettedrev og printer og har nu i nogle uger været afprøvet på kryds og tværs af redaktionen.

Vigtigst for os er, at der fulgte en COMAL med.

Sproget: COMMODORE 64 COMAL-80 rev 00.12 leveres på 2 disketter, der er ganske ens.

Firmaet anbefaler, at man straks gemmer den ene et sikkert sted og kun anvender den anden.

Sagen er, at disketterne ikke kan kopieres. Så hvis man er så ufiks at ødelægge den ene, kan man ty til den anden, medens man gennem sin forhandler (mod et mindre vederlag) får byttet den første.

Til disketterne hører en håndbog skrevet af ingen ringere end COMAL's "fader", Børge R. Christensen, og det er gjort i en kort og præcis stil uden noget omsvøb - lidt for kort, hvis man tænker sig at bruge bogen som noget i retning af en lærebog - Det er den ikke, og det står udtrykkeligt bemærket i forordet, hvor den betegnes som det, den er: en opslagsbog.

For den, der leder efter en lærebog, kan anmelderen røbe, at bogens forfatter også har gjort sig fortjent på det felt! Men udvalget er i øvrigt voksende. Blot drejer det sig om at finde en lærebog, der anvender den COMAL, som brugeren har i sin maskine - Denne bemærkning er desværre ikke helt overflødig endnu. Lad os se, hvad standardiseringsbestræbelserne efterhånden fører til.

COMMODORE 64 COMAL-80 kan for øvrigt også bruges på VIC 20, som er lillebroderen til COMMODORE 64; men dens hukommelse skal jo nok blokkes noget ud!

Denne COMAL-80 er en rev. 00.12, der med enkelte udeladelser svarer til rev. 01.02, som anvendes på de større COMMODORE maskiner, CBM 4032, 8032 og 8096 forsynet med COMAL-board fra fa. INSTRUTEK.

Specielt for denne revision er grafikken og anvendelsen af sprites.

Grafikken er en turtlegrafik, altså et tegnesystem, der anvender en lille trekant, som man forresten selv kan formindske eller forstørre. Den kaldes turtle = skildpadde. Andre lignende grafiksystemer kalder deres "tegnehoved" myre eller mus. Det sidste udtryk anvendes for øvrigt i håndbogen, formodentlig fordi ordet skildpadde er mindre børnevenligt end ordet mus.

Ideen er jo nok, at bevægelseskommandoerne for turtlegrafikken er så enkle, at selv mindre børn kan lære dem og derved blive i stand til at tegne på skærmen ligesom på en tavle.

Og resultatet af de lette instrukser er faktisk ret imponerende. Det skyldes jo ikke mindst, at man kan disponere over ikke færre end 64.000 prikker og et væld af farver - ikke alene på de tegnede streger, men også på baggrunden og på rammen uden om det hele.

COMAL BRUGER NYT

Tekst kan anbringes hvorsomhelst på skærmen - også oven i selve tegningerne.

I øvrigt opererer systemet med to skærme, en tekstskærm og en grafikskærm. Man kan "gemme" grafikskærmen og arbejde på tekstskærmen, og samtidig kan man fortsætte arbejdet på den skjulte grafikskærm. Man kan så senere kalde resultatet frem og skifte mellem tekst og tegning efter behag.

Et helt afsnit af håndbogen er helliget begrebet sprites, disse figurer, der kan bevæges hen over skærmen.

Når man har set, hvilke resultater øvede sprites-programmører kan præstere, forstår man, at COMMODURE gør meget ud af at fremhæve maskinens enestående muligheder på dette felt.

For god ordens skyld bør det vel også bemærkes, at yderligere en fremtrædende egenskab ved 64'eren er dens musikfaciliteter, der er så veludviklede, at man - hvis man ellers er vaks nok til det - kan fremtrylle hele kunstværker af vellyd.

COMMODURE 64 er blevet solgt i meget store tal, og med den lette adgang til at udstyre den med COMAL-80, må man formode, at den i sig selv vil være en uvurderlig hjælp ved udbredelsen af dette programmeringsprog.

Yderligere succes for sproget (og maskinen) spår vi, vi vil få at se i den allernærmeste fremtid, når COMAL-80 kommer som cartridge (indstiksmodule) til COMMODURE 64. Der bliver her tale om den nyeste version: 2.00, der vil ligge endnu nærmere op ad det, der forventes at blive standarden, og som vil rumme adskillige udvidelser i forhold til den COMAL, vi kender idag.

S. Chr. H.

VENTEPROCEDURE

Her er en meget enkel procedure, der kan bruges, når man i et program skal indlægge en pause.

Der anvendes parameteroverførsel til bestemmelse af pausens længde, og der åbnes mulighed for, at brugeren kan korte pausen af blot ved at trykke på mellemrumstasten.

```
0010 // list "1.venteprocedure"
0020 PROC vent(sek)
0030 TIME 0
0040 WHILE TIME<60*sek AND
      KEY$<>" " DO NULL
0050 ENDPROC vent
```

Her anføres et par linier, der kan tilføjes, så man kan afprøve proceduren.

```
0060 PRINT "Begynd!"
0070 vent(5)
0080 PRINT "Fortsæt!"
```

C B N



Udgiver:
C B N
Comal Bruger Gruppe
Mindegade 42
8700 Horsens

Tlf. 05 - 62 15 67
Giro 1 75 23 75

Ansvarshavende
redaktør:
S. Chr. Hansen

Tryk:
tekst & tryk
Horsens

Abonnement:
Mindegade 42
8700 Horsens

COMAL BRUGER NYT

SIMPELT REGISTER

Registerprogrammet er et simpelt register der arbejder med en enkelt tekstvariabel (streng). Programmet er ikke ment som et færdigt program for denne register-type, men som en overskuelig illustration af en metode til opbygning af et register.

Meningen med programmet er, at man skal kunne indtaste en række oplysninger i et register og senere kunne finde disse oplysninger frem igen, blot man kender mindre dele af disse oplysninger.

Det kunne være et navnerregister, hvor man så kunne søge på navn, telefonnummer m. m. Eller det kunne være et register over gramofonplader, hvor man kunne søge på en titel eller en del af en titel, hvorefter registeret på skærmen kan udskrive, hvilken lp denne titel indgår i, og hvilke andre titler der er på denne plade.

For at undgå stor diskaktivitet og lange programmer og forklaringer er registeret i dette eksempel blot en tekstvariabel med en længde på 10.000 tegn. Denne variabel læses ind i computeren ved opstart ("hent register"). Derefter kan man arbejde med dette lille register i computeren ("benyt register"). Når man er færdig med indskrivninger og rettelser m. m., gemmes registeret igen på disketten ("gem register").

Når register-variablen er indlæst, kan man oprette nye oplysninger, rette, slette, søge og udskrive registeret på skærmen. Alle disse funktioner kaldes fra proceduren "benyt register", som er en case-struktur.

I proceduren "opret" kan man se, hvordan nye tekster indskrives i registeret. I denne procedure foregår indtastningen af nye tekster i en loop. Hver indtastning i input-sætningen (tekst\$) adderes til temp\$ sammen med en carriage

return (chr\$(13)). På denne måde vil temp\$ indeholde et antal indtastede linier adskilt af en carriage return.

Når indtastningen ønskes afsluttet, trykkes der blot "return". Hvis variabelen tekst\$ er tom, brydes loop'en (exit). Variablen temp\$ indeholder nu hele indtastningen, og temp\$ adderes til registeret (reg\$) sammen med en chr\$(0).

Således kan indtastningen fortsætte, og reg\$ vil nu indeholde en række tekstlinier adskilt af en carriage return (chr\$(13)).

Et sæt af linier, der hører sammen vil være adskilt af en chr\$(0). Det er således muligt senere at udskille et sæt sammenhængende linier, f.eks. navn, gade, by, telefonnummer o. s. v.

Vi kan med det samme se på, hvordan vi vil kunne finde et bestemt sæt linier f.eks. navn, gade, by, telefonnummer m. m. ved blot at kende navnet eller en del af navnet, gadens navn eller en del af dette - eller sagt på en anden måde - blot en kombination af tegn, der indgår i en af de linier vi søger, hvorefter vi får samtlige oplysninger udskrevet.

Proceduren "søg" kalder, hvis søgeteksten overhovedet findes i registeret, en ny procedure ("udskriv"). Det første, der sker i proceduren "udskriv", er, at en variabel benævnt pil tildeles en værdi. Dette gøres ved anvendelse af "in"-operatoren. Variablen "pil" tildeles en værdi svarende til den plads i reg\$, hvor søgeteksten starter.

Vor søgetekst er måske et navn, og vi ønsker at få adresse og andre oplysninger på denne person udskrevet.

Vi har nu en pil, der peger ind i registeret, der hvor vor søgetekst står. Det, vi ønsker udskrevet, er samtlige oplysninger, der hører til et sæt linier. Vi tæller der-

COMAL BRUGER NYT

for bøgkens med vor pil, til vi møder en chr\$(0). Der starter vore oplysninger. Når vi har fundet denne chr\$(0), tildeler vi variabelen "start" pilens værdi.

Nu tæller vi så pilen frem, til vi igen møder en chr\$(0). Her slutter vore oplysninger. Variablen "slut" tildeles denne værdi, og vi kan nu udskrive den del af registeret, vi søgte: print register\$(start:slut).

Vi har nu fået vort arbejde udført. Men det kunne jo være, vi søgte på efternavnet "Jensen", og at der i registeret står flere med dette efternavn. Derfor undersøger vi, om søgeteksten forefindes i resten af registeret. Hvis dette ikke er tilfældet, vil "ny pil" få værdien 0 og udskrivningen slutter her. Hvis søgeteksten findes et sted i resten af registeret, tildeles pilen den nye værdi, hvor søgeteksten starter, og hele mællen gentages, indtil alle de sæt af oplysninger, der indeholder søgeteksten, er udskrevet: alle der hedder "Jensen".

Når vi ønsker at slette oplysninger i registeret, anvendes proceduren "slet", som blot kalder proceduren "søg". Variablerne "start" og "slut" anvendes derefter til at udpege den del af reg\$, der skal fjernes. Dette gøres i praksis ved, at man gemmer reg\$ fra starten og frem til starten af det, der skal slettes, sammen med reg\$ fra, hvor det, der skal slettes, slutter og frem til slutningen af reg\$.

Hvis man ikke tidligere har anvendt "in"-operatoren, og ikke har prøvet at udtage dele af en tekstvariabel, kan det anbefales, at man øver sig på det, før man prøver dette lille program.

Meningen med dette lille program-eksempel er at give begyndere mulighed for hurtigt at lave et lille register, som samtidig er nemt at forstå og overskue.

Registerprogrammet skal ikke opfattes som et færdigt register, men som en beskrivelse af et princip, ved hjælp af hvilket man kan søge på samtlige oplysninger i et register. Princippet kaldes en relationsdatabase, og princippet kan anvendes ved skrivning af et mere anvendeligt register.

Senere artikler vil beskrive andre former for registre, hvor oplysninger ligger på disketten, og registeret derfor kan være betydeligt større, men med hurtige søgetider. Næste artikel vil således beskrive teknikken i et "hash"-register.

Ole Holmgren

```

0010 // list "l.reg"
0020 // filnavn se 810 og 870
0030
0040 opstart
0050
0060 PROC program
0070     hent 'register
0080     benyt 'register
0090     gem 'register
0100 ENDPROC program
0110
0120 PROC benyt 'register
0130     LOOP
0140         forklaring
0150         INPUT "Kommando: ": funktion$
0160         CASE funktion$ OF
0170             WHEN "indskriv", "i"
0180                 opret(TRUE)
0190             WHEN "søg", "s"
0200                 søg
0210             WHEN "ret", "r"
0220                 søg
0230             WHEN "udskriv", "u"
0240                 udskriv 'register
0250             WHEN "fjern", "f"
0260                 søg
0270             WHEN "hjælp", "h"
0280                 benyt 'register
0290             WHEN "stop", "*"
0300                 EXIT
0310             OTHERWISE
0320                 bell
0330             ENDCASE
0340         ENDOOP
0350 ENDPROC benyt 'register
0360

```

COMAL BRUGER NYT

```

0370 PROC opstart
0380   PRINT CHR$(147)
0390   DIM reg$ OF 10000, k$ OF 1, svar$ OF 1
0400   DIM funktion$ OF 80
0410   DIM tekst$ OF 80
0420   DIM temp$ OF 1000
0430   rette'eksisterende:=FALSE
0440   program
0450 ENDPROC opstart
0460
0470 PROC bell
0480   PRINT CHR$(7)+"Fejl"
0490 ENDPROC bell
0500
0510 PROC opret(ny'tekst)
0520   temp$=""
0530   IF NOT ny'tekst THEN
0540     linie:=0
0550     FOR i:=start TO slut DO
0560       IF URD(reg$(i))=13 THEN linie:+1
0570     ENDFOR i
0580     FOR i:=1 TO linie+1 DO PRINT CHR$(145),
0590       slet'del(start-1,slut+2)
0600   ENDIF
0610   LOOP
0620     INPUT "": tekst$
0630     IF tekst$="" THEN EXIT
0640     temp$:+tekst$+CHR$(13)
0650   ENDLOOP
0660   IF NOT ny'tekst THEN
0670     IF start=2 THEN
0680       reg$:=CHR$(0)+temp$+CHR$(0)+reg$(2:LEN(reg$))
0690     ELIF start<LEN(reg$) THEN
0700       reg$:=reg$(1:start-1)+temp$+CHR$(0)+reg$(start:LEN(reg$))
0710     ELSE
0720       reg$:+temp$+CHR$(0)
0730     ENDIF
0740     slut:=start+LEN(temp$)-1
0750   ELSE
0760     reg$:+temp$+CHR$(0)
0770   ENDIF
0780 ENDPROC opret
0790
0800 PROC hent'register
0810   OPEN FILE 4,"artikler",READ
0820   READ FILE 4: reg$
0830   CLOSE
0840 ENDPROC hent'register
0850
0860 PROC gem'register
0870   OPEN FILE 4,"$:artikler",WRITE // Erstat "$" med tegnet for "at"
0880   WRITE FILE 4: reg$
0890   CLOSE
0900   PRINT CHR$(147),"Skal vi fortsætte? (j/n) ",
0910   INPUT svar$
0920   IF svar$ IN "Jj" THEN
0930     PRINT CHR$(147)
0940     program
0950   ELSE

```

COMAL BRUGER NYT

```

0960     END "Program slut"
0970     ENDIF
0980 ENDPROC gem'register
0990
1000 PROC søg
1010     INPUT "Søgetekst: ": tekst$
1020     PRINT
1030     IF tekst$ IN reg$ THEN
1040         udskriv(tekst$)
1050     ELSE
1060         PRINT "Findes ikke"
1070     ENDIF
1080 ENDPROC søg
1090
1100 PROC udskriv(søgetekst$)
1110     pil:=søgetekst$ IN reg$
1120     IF pil THEN
1130         REPEAT
1140             WHILE NOT reg$(pil-1)=CHR$(0) DO pil:-1
1150                 start:=pil
1160             WHILE NOT reg$(pil+1)=CHR$(0) DO pil:+1
1170                 slut:=pil
1180             PRINT reg$(start:slut)
1190             IF funktion$="r" OR funktion$="ret" THEN
1200                 opret(rette'eksisterende)
1210             ELIF funktion$="f" OR funktion$="fjern" THEN
1220                 godkend
1230             ENDIF
1240             ny'pil:=søgetekst$ IN reg$(slut:LEN(reg$))
1250             IF ny'pil THEN vent
1260             pil:=slut+ny'pil
1270         UNTIL NOT ny'pil
1280     ENDIF
1290 ENDPROC udskriv
1300
1310 PROC slet'del(fra,til)
1320     reg$:=reg$(1:fra)+reg$(til:LEN(reg$))
1330 ENDPROC 'slet'del
1340
1350 PROC udskriv'register
1360     PRINT
1370     pil:=1
1380     REPEAT
1390         pil:+1
1400         PRINT reg$(pil),
1410         IF reg$(pil)=CHR$(0) THEN
1420             PRINT
1430             vent
1440         ENDIF
1450     UNTIL pil=LEN(reg$)
1460 ENDPROC udskriv'register
1470
1480 PROC vent
1490     PRINT "Fortsæt = SPACE      Ny kommando = h"
1500     REPEAT
1510         k$:=KEY$
1520     UNTIL k$="h" OR k$=CHR$(32)
1530     IF k$="h" THEN
1540         pil:=LEN(reg$)

```

COMAL BRUGER NYT

```

1550     ny`pil:=FALSE
1560     ELSE
1570     PRINT
1580     ENDIF
1590 ENDPROC vent
1600
1610 PROC forklaring
1620     // skriv
1630     PRINT
1640     PRINT "Følgende kommandoer kan anvendes:"
1650     PRINT
1660     PRINT """"indskriv"" = i"
1670     PRINT """"søg"" = s"
1680     PRINT """"ret"" = r"
1690     PRINT """"udskriv"" = u"
1700     PRINT """"fjern"" = f"
1710     PRINT """"stop"" = *"
1720     PRINT """"hjælp"" = h"
1730     PRINT
1740 ENDPROC forklaring
1750
1760 // PROC skriv
1770 //     FOR n:=1 TO LEN(reg$) DO PRINT ORD(reg$(n));
1780 // ENDPROC skriv
1790
1800 PROC godkend
1810     PRINT "Skal denne record slettes (j/n)? ",
1820     REPEAT
1830         svar$:=KEY$
1840     UNTIL svar$ IN "JjNn"
1850     PRINT svar$
1860     IF svar$ IN "Jj" THEN
1870         slet`del(start-1,slut+2)
1880         slut:=start
1890     ENDIF
1900     PRINT
1910 ENDPROC godkend

```

BRUGSANVISNING

Før registeret kan anvendes, skal der oprettes en fil, som variabelen reg\$ kan gemmes i. Filens navn er - som bemærket i linie 20 - anført i linie 810 og 870. Så hvis man vil bruge et andet navn, end det ("artikler"), der er valgt i programmet, må man ændre det i disse to linier.

C B N

Her er et program, der kan bruges til at oprette filen:

```

0010 // list "1.open program"
0020 DIM a$ OF 1, b$ OF 20
0030 PRINT CHR$(147)
0040 INPUT "Hvød skal registeret hedde? ": b$
0050 a$:=CHR$(0)+" "
0060 OPEN FILE 2,b$,WRITE
0070 WRITE FILE 2: a$
0080 CLOSE
0090 PRINT
0100 PRINT "Registeret: ",b$," er klar!"

```


COMAL BRUGER NYT

Når oprettelse af filen er overstået, kan registeret tages i brug.

ENTER (eller LOAD) registerprogrammet og RUN det.

Det melder sig ved at udskrive menuen (linie 1640 - 1720).

Det første, vi må gøre på dette tidspunkt, er naturligvis at indskrive nogle records i registeret. Derfor vælger vi "i", og maskinen er straks klar til at modtage den første linie tekst. Bare skriv løs med anvendelse af sæmtilige tegn, hvis det ønskes; men linien må ikke blive længere end 80 anslag. Har man lyst til at skrive mere, må man trykke RETURN, så kan man fortsætte på næste linie, og det kan gentages, så mange gange man ønsker. Der er altså ingen fastlagt størrelse på en record. Det er netop en af fiduserne ved dette simple register frem for så mange andre mere sofistikerede.

Når alt er indskrevet i denne record, trykker man to gange efter hinanden på RETURN, så er man ude af indskrivningen, menuen dukker op igen, og man kan vælge en ny kommando.

Vi kan nu afprøve registerets næste funktion: søg.

Men først må vi gøre en forudsætning: Lad os sige, at vi har valgt at oprette et kartotek over gram-mofonplader eller musikkassetter, hvor vi indskrives f. eks. grup-pens navn og musikstykkets titel.

Nu vil vi se efter et bestemt musikstykke af The Beatles; men vi behøver ikke at vide, hvad musikstykket hedder. Vi kan blot søge på "Beatles", så kommer efter tur alle de stykker, vi har indskrevet med denne gruppe.

Altså tast "s", og straks fremkommer på skærmen ordet "Søgetekst:"

Nu taster vi "Beatles" (anførelse-tegnene skal naturligvis ikke med!), så kommer den første record, der indeholder navnet Beatles, og maskinen tilføjer:
Fortsæt = SPACE Ny kommando = h

Vælger man SPACE, får man den næste record med Beatles; men vælger man h, kommer menuen frem, og man kan vælge en ny kommando.

Fortsætter man med SPACE, indtil der ikke er flere records med Beatles, får man også menuen og kan vælge en ny kommando.

Hvis vi vil rette i en af vore tekster, vælger vi "r".

Også her bliver vi spurgt om "Søgetekst:"

Så kunne vi f. eks. skrive "yellow", og så finder maskinen frem til "The yellow submarine", hvis den ellers er indskrevet, og cursoren stiller sig på den første linies første plads.

Derefter kan man rette i linien ganske som man ellers kan rette tekst på skærmen.

Når man er færdig med at rette linien, eller hvis man slet ikke vil rette den, trykker man RETURN, og det gør man, lige til man er ude af denne record.

Derefter ser vi igen:
Fortsæt = SPACE Ny kommando = h
Og det er jo så ensbetydende med, at vi - som set under "søg" - kan få den næste record frem, der indeholder ordet "yellow" (hvis der er nogen!). Ellers får vi nok en gang menuen og kommandovalget.

Vil vi se alle records, der findes i registeret - eller i hvert fald nogle af dem, taster vi "u". Så kommer de som perler på en snor adskilt af:
Fortsæt = SPACE Ny kommando = h
Og så kan vi jo fortsætte, så længe vi har lyst, og så længe der er flere records.

COMAL BRUGER NYT

Vi kan også fjerne en record helt fra registeret:

Last "f", så får vi nok engang "Søgetekst:", og vi svarer med noget, der står i den record, vi ønsker slettet. Jo mere vi tager med som søgetekst, desto mere entydigt får vi fat i den rigtige record.

Når den melder sig på skærmen, ledsages den af teksten:
Skal denne record slettes (j/n)?
Der er altså mulighed for at fortryde!

Trykkes der "j", er denne record ude af registeret, næste gang vi kalder det frem.

Trykker vi derimod "n", sker der selvfølgelig ikke noget med den pågældende record. I stedet får man den næste at se - vel at mærke, hvis der er flere, der indeholder søgeordet.

Nu mangler vi blot at få alt skrevet over på disketten.

Det går ganske let: Last "*" eller "stop", så begynder diskdrevet at snurre, og kort tid efter meldes indskrivningen på disk færdig med disse ord:
Skal vi fortsætte (j/n)

Og reaktionen giver jo sig selv - Svares "j", indspilles registeret fra disketten, og hele forestillingen begynder forfra.

Svares derimod "n", så slutter programmet høfligt med:
Program slut.

Advarsel!

NB!

Bemærk!

(Nu kan det vist ikke siges tydeligere!)

Husk altid at slutte med "stop" eller "*".

Hvis det bliver glemt, er alt, hvad der er foretaget med registeret i den aktuelle situation spildt. Registeret vil være totalt uforandret.

Altså: Alle rettelser, sletninger og indskrivninger kommer kun med ind i registeret, hvis programmet sluttes korrekt med "stop" eller "*".

P. S.

For den, der er interesseret i at se, hvordan hele registeret ser ud i ASCII værdier, er der indsat en procedure: skriv - i linie 1760 - 1780.

Den kaldes i linie 1620; men den bliver kun virksom, hvis man fjerner remarkstregerne (//) i de nævnte 4 linier.

CBN

SPØRGEBREVKASSEN

Spørgsmål:

I RC Comal-80 findes kommandoen MARGIN, der sætter højre margin ved udskrift på skærmen.

Hvordan gøres det på COMMODORE?

Svar:

MARGIN er ikke implementeret i CBM COMAL-80. Under opstart sættes margin fast til 80 på skærmen og til uendelig på disk og printer.

Brugen af margin kan diskuteres og bliver det.

Oftest ser man i COMAL-programmer på Piccolo, at margin i starten sættes til 0 (svarer til uendelig), fordi margin i RC COMAL-80 er global og er aktiv både på skærm, printer og floppy disk samtidig, hvorfor den som nævnt næsten altid sættes til 0 i starten af et program.

Hvis dette er generelt for næsten alle programmer, er det måske årsagen til, at MARGIN ikke er implementeret i CBM COMAL-80.

COMAL BRUGER NYT

RUSSISK ROULET

Programmet viser værdien af data-
matens evne til at simulere d. v.
s. lade som om.

Spillet "Russisk roulet" går som
bekendt ud på at lade en seks-
skuds tromlerevolver med 1 skarp

patron, hvorefter tromlen roteres.
Så sættes pistolen til tindingen,
og aftrækkeren aktiveres.

Der er 5 chancer ud af 6 for at
overleve.

Prøv dit held!

S. J.

```

0010 // list "1.russisk roulet"
0020 DIM a$ OF 1
0030 REPEAT
0040   PRINT CHR$(147); // slet skærm
0050   nummer:=0
0060   skud
0070   forts
0080 UNTIL valg=0
0090 PROC skud
0100   REPEAT
0110     nummer:+1
0120     valg:=RND(1,6)
0130     PRINT
0140     PRINT "Skud nummer ",nummer
0150     REPEAT
0160       PRINT
0170       INPUT "Hvilket tal skyder du på ": dit'skud
0180       UNTIL dit'skud<7 AND dit'skud>0 AND dit'skud=INI(dit'skud)
0190       IF valg<>dit'skud THEN
0200         PRINT "Du overlevede denne gang"
0210       ELSE
0220         død
0230       ENDIF
0240     UNTIL valg=dit'skud
0250   ENDPROC skud
0260 PROC død
0270   FOR n:=1 TO 3 DO PRINT
0280   PRINT "Du er nu *** d ø d ***"
0290   ENDPROC død
0300 PROC forts
0310   FOR n:=1 TO 3 DO PRINT
0320   PRINT "Vil du spille igen      j/n"
0330   REPEAT
0340     a$:=KEY$
0350     UNTIL a$="j" OR a$="n"
0360     IF a$="j" THEN
0370       valg:=1
0380     ELSE
0390       valg:=0
0400     ENDF
0410   ENDPROC forts

```

COMAL BRUGER NYT

CBN

SPØRGEBREVKASSEN

Spørgsmål:

Jeg har en division: 34947/23.
 Det giver noget i retning af
 1519.43478.
 Jeg skal kun bruge kroner og øre -
 altså heltallet + 2 decimaler.
 Hvordan skal jeg gøre det?

Svar:

Der er to måder at løse problemet
 på.

Enten:

```
0010 PRINT USING "EEEE.EE":
34947/23
0020 PRINT USING "EEEE.EE":
15
```

Udskriver: 1519.43
 15.00

```
0010 PRINT USING "EEEE.EE":
34947/23
0020 PRINT USING "EEEE.EE":
15
```

Udskriver: 1519.435 (matematisk
 oprunding)
 15.000

```
0010 PRINT USING "EEEE.E":
34947/23
0020 PRINT USING "EEEE.E":
15
```

Udskriver: 1519.4
 15.0

Eller:

```
0010 IAL:=34947/23
0020 PRINT INT(IAL*100+.5)/100
```

Udskriver: 1519.43

CBN

SPØRGEBREVKASSEN

Spørgsmål:

Hvordan er forholdene vedr. random
 tal?
 Kan man være sikker på at få et
 tilfældigt tal med:

```
0010 FOR N=1 TO 10
0020 I=RND(1,100)
0030 PRINT I
0040 ENDFOR N
```

Eller får man altid den samme ræk-
 kefølge af tilfældige tal på denne
 måde?

Hvad skal man gøre for at få samme
 rækkefølge gang på gang?

Svar:

Da det ikke fremgår af spørgsmå-
 let, hvilken COMAL-80 version det
 drejer sig om, må vi generelt hen-
 vise til den tilhørende manual;
 men vi vil her gengive et eksempel
 fra CBM COMAL-80 rev. 02.00.

```
0010 LOOP
0020 INPUT "SEED ":SEED
0030 RANDOMIZE SEED
0040 FOR I=1 TO 10 DO
0050 PRINT RND(1,100);
0060 ENDFOR I
0070 ENDLOOP
```

Hvis man prøver at indtaste oven-
 stående program, vil man hurtigt
 kunne se virkningen. Hvis man in-
 put'er en 'seed' på 3 flere gange
 i træk, vil man se, at den samme
 række tilfældige tal genereres
 igen og igen. Hvis man vælger en
 ny 'seed' på f.eks. 5, flere gange
 i træk, vil en ny række tilfældige
 tal blive genereret.

CBN

Indmeldelsesblanketten på
 bagsiden må gerne kopieres!

COMAL BRUGER NYT

DELMÆNGDER

Når man skal lede efter en delmængde i en streng på skærmen, er den såkaldte IN-operator i COMAL 80 af stor betydning. Et af de steder, hvor den kan have betydning, er i adressekartoteker, hvis der skal foretages et bestemt valg, der ikke alene er bestemt af en alfabetisk sortering.

Begrebet delstreng kan lettest forklares ved nogle eksempler: "o" er delmængde af "Comal 80", fordi "o" indgår i strengen. "al" er ligeledes delmængde; men "om" er ikke delmængde. Det er derimod "l 8".

En procedure til den brug er her:

```
0010 DIM tekst$ OF 50, delmængde$ OF 50, tekst1$ OF 50
0020 tekst$:="Jean Frey, Kanadavej 11, 2300 København S"
0030 PRINT tekst$
0040 INPUT " delmængde > ": delmængde$
0050 delmængde'søgning(delmængde$,tekst$,tekst1$)
0060 PRINT delmængde$
0070 PRINT tekst1$
0080
0090
0100 PROC delmængde'søgning(delmængde$,tekst$,REF tekst1$) CLOSED
0110   længde:=LEN(tekst$)
0120   IF delmængde$ IN tekst$ THEN
0130     punkt:=delmængde$ IN tekst$
0140     tekst1$:+tekst$(1:punkt-1)+CHR$(18)+delmængde$+CHR$(146)
0150     tekst$:=tekst$(punkt+1:længde)
0160     delmængde'søgning(delmængde$,tekst$,tekst1$)
0170   ELSE
0180     tekst1$:+tekst$
0190   ENDIF
0200 ENDPROC delmængde'søgning
```

Linierne 0010 - 0090 skal slettes, når proceduren skal indgå som et underprogram i et hovedprogram.

Der er tale om en lukket procedure med parameteroverføring. At den er lukket = <CLOSED> betyder, at det eneste, der kan komme ind og ud af den, er bestemt af de variabelnavne, der står i parenteser efter procedurenavnet. Hvis der samtidig står <REF> foran variabelnavnet, kan det komme både ind og ud. I det aktuelle tilfælde går verdierne i delmængde\$ og i tekst\$ ind; men de kan ikke komme ud igen, mens tekst1\$'s værdi kan komme både ind og ud.

Alle andre variabelnavne, der bruges inde i den lukkede procedure, er ukendte for den del af programmet, der ligger uden for proceduren.

En lukket procedure kan altså sammenlignes med en lukket kasse. Denne kasse kan bruges i alle mulige programmer, så længe den kaldes korrekt.

Der sker følgende: I linie 0020 kommer tekst\$ til at indeholde en adresse. Den kan i følge linie 0010 blive op til 50 pladser lang. Her er den 41 pladser lang. Proceduren kaldes i linie 0050 med 3 parametre, nemlig 1) den delmængde, der skal søges efter,

COMAL BRUGER NYT

2) den tekststreng, der skal søges på, og 3) den tekststreng, der skal returneres med markeringer for de steder, hvor delmængden er fundet.

Sidstnævnte parameter skal altså returneres med det færdige resultat og udskrives i linie 0070.

Man kan også udtrykke det med et billede: De to parameterværdier er råstoffer, og det tredje er produktet, og det skal naturligvis ud af "kassen" og bruges til noget.

I linie 0100 måles længden af tekststrengen, tekst\$. I linie 0120 spørges ved hjælp af IN-operatoren, om den pågældende delmængde (delmængde\$) findes i tekststrengen. Er det ikke tilfældet, springes ned til linie 0180, hvor den streng, der skal returneres fra den lukkede procedure, tekst1\$, modtager indholdet af den oprindelige streng, tekst\$.

Hvis delmængden findes i tekststrengen, tekst\$, bestemmes i linie 0130 punktet, hvor delmængden begynder. Er delmængden f. eks. "a", får <punkt> værdien 3. Er delmængden "ave", får <punkt> værdien 17.

I linie 0140 afskæres tekst\$ umiddelbart før den plæds, der har nummer <punkt>. Denne forreste del indsættes i tekst1\$. Hertil lægges delmængden i revers (sorte bogstaver på lys baggrund). Det er den del, der begynder med CHR\$(18) og slutter med CHR\$(146), der henholdsvis reverserer og ophæver reverseringen.

Tekst1\$ indeholder nu den første del af tekst\$ med afsluttet lysende delmængde. I linie 0150 ændres den oprindelige tekststreng, tekst\$, til at bestå af det, der er tilbage af den. Ved at lægge 1 til <punkt> bliver reststrengen den del, der endnu ikke er undersøgt.

Herefter kaldes proceduren rekursivt i linie 160, og næste del, indtil delmængde er fundet, ekspeceres.

Når delmængden ikke mere findes i strengen, gås til linie 0180, og herefter har proceduren udført sin opgave, og resultatet udskrives i linie 0070 med delmængde i revers.

Programmet her er en let omskrivning af det førte:

```

0010 DIM tekst$ OF 50, delmængde$ OF 50, tekst1$ OF 50
0020 tekst$:="Jean Frey, Kanadavej 11, 2300 København S".
0030 PRINT tekst$
0040 INPUT " delmængde > ": delmængde$
0050 delmængde`øgning
0060 PRINT delmængde$
0070 PRINT tekst1$
0080
0090
0100 PROC delmængde`øgning
0110   længde:=LEN(tekst$)
0120   IF delmængde$ IN tekst$ THEN
0130     punkt:=delmængde$ IN tekst$
0140     tekst1$:+tekst$(1:punkt-1)+CHR$(18)+delmængde$+CHR$(146)
0150     tekst$:=tekst$(punkt+1:længde)
0160     delmængde`øgning
0170   ELSE
0180     tekst1$:+tekst$(1:længde)
0190   ENDIF
0200 ENDPROC delmængde`øgning

```

COMAL BRUGER NYT

Her er tale om en procedure uden parameteroverføring. Den arbejder fuldt ud lige så godt som den første; men den kan ikke bruges universelt, fordi alle variable er kendt overalt også i hovedprogrammet.

For at afprøve delmængdesøgning i COMAL 80 kan man bruge dette program:

```

0010 DIM tekst$ OF 1500, delmængde$ OF 1500, tekst1$ OF 1500
0020 delmængde$:="a"
0030
0040 FOR q:=1 TO 960 DO
0050   tekst$:+CHR$(RND(65,93))
0060 ENDFOR q
0070
0080 PRINT tekst$
0090
0100 PRINT
0110
0120 a:=TIME
0130 delmængde`søgning(delmængde$,tekst$,tekst1$)
0140 PRINT tekst1$
0150 b:=TIME
0160
0170 PRINT
0180
0190 PRINT "sekunder      >";INT((b-a)/60)
0200 PRINT "antal hændelser >";(LEN(tekst1$)-LEN(tekst$))/2
0210
0220
0230
0240 PROC delmængde`søgning(delmængde$,tekst$,REF tekst1$) CLOSED
0250   længde:=LEN(tekst$)
0260   IF delmængde$ IN tekst$ THEN
0270     punkt:=delmængde$ IN tekst$
0280     tekst1$:+tekst$(1:punkt-1)+CHR$(18)+delmængde$+CHR$(146)
0290     tekst$:=tekst$(punkt+1:længde)
0300     delmængde`søgning(delmængde$,tekst$,tekst1$)
0310   ELSE
0320     tekst1$:+tekst$
0330   ENDIF
0340 ENDPROC delmængde`søgning

```

Linierne 0240 - 0340 er identiske med linierne 0100 - 0200 i de to forrige eksempler. Bemærk, at man skal dimensionere efter den tekststreng, tekst1\$, der kommer ud af proceduren. Den er længere end den, der kommer ind, hvis den indeholder delmængder. Det forhold kan man benytte til at tælle antallet af hændelser for delmængde-indhold. Se linie 0200.

Eksemplet her, hvor der dannes en streng bestående af 960 tilfældige bogstaver (linierne 0040 - 0060), meddeler samtidig tiden for opgavens løsning.

Angivelsen af delmængdeplacering med reverserede tegn på skærmen tager i dette tilfælde under 3 sekunder.

COMAL BRUGER NYT

C B G
 COMAL BRUGER GRUPPE
 Mindegade 42
 8700 Horsens
 Tlf 05 - 62 15 67
 Giro 1 75 23 75

INDMELDELSE I CBG

Indmeldelse foretages ved udfyldelse af nedenstående blanket, der sendes til ovenstående adresse.

Samtidig indbetales kr. 90,00 - prisen for 1 års abonnement på "COMAL BRUGER NYT", der agtes udgivet med 6 numre pr. år.

Indbetaling kan ske pr. check eller ved overførsel til:

Giro nr. 1 75 23 75, COMAL BRUGER GRUPPE, Mindegade 42, 8700 Horsens.

```

*****
*
*   Undertegnede
*
*
*   Navn:.....
*
*
*   Adresse:
*   Gade:.....
*
*
*   Postnummer:.....   By:.....
*
*
*   Indmelder sig i COMAL BRUGER GRUPPE.
*
*   Kr. 90,00 vedlægges i check ....
*
*                               Sæt venligst X
*
*           indbetales pr. giro ....
*
*****
  
```