

NAS Z80 MYT

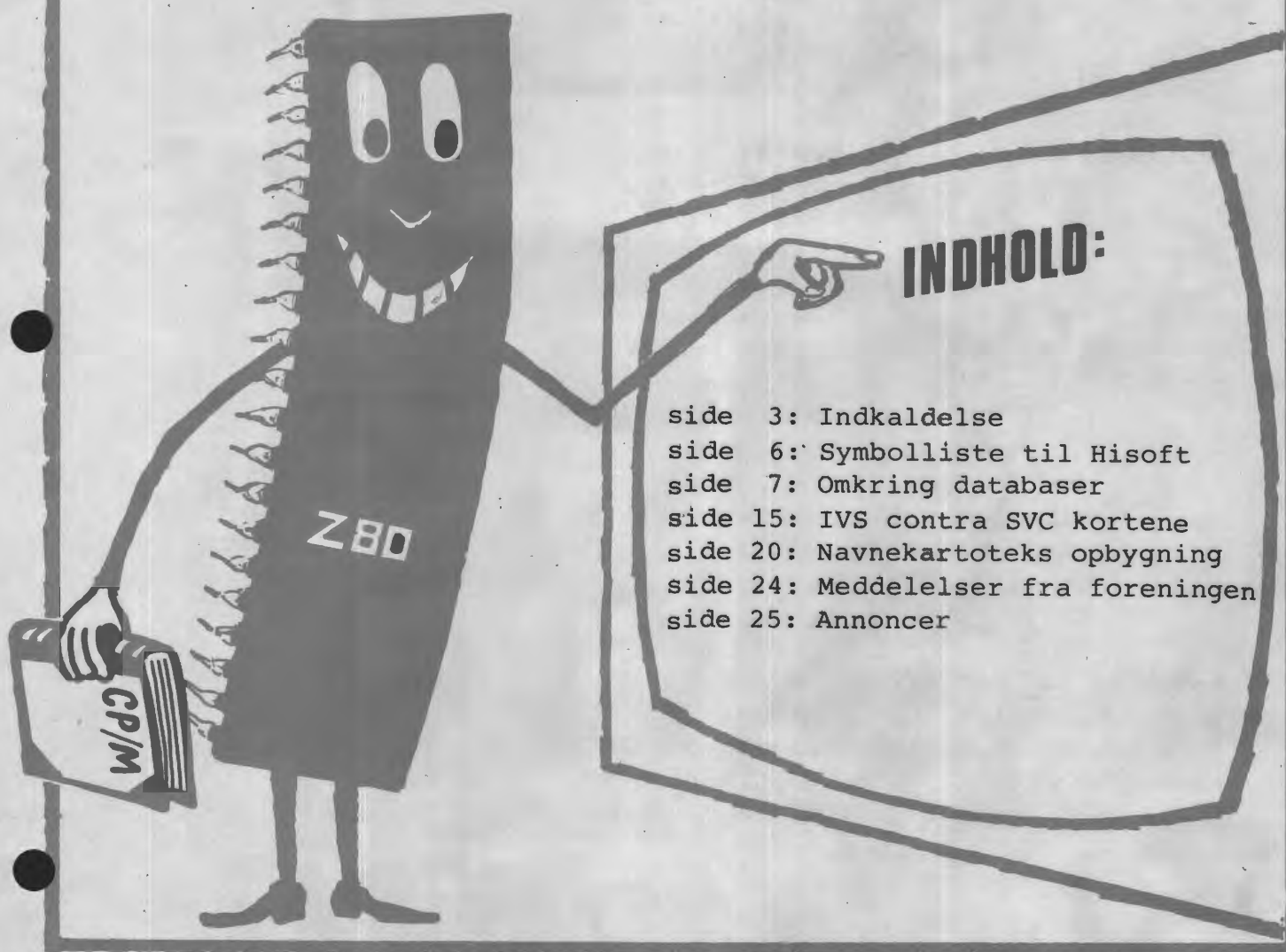
BUS

UDGIVET AF Z80 BRUGERGRUPPEN

5. ÅRGANG NR. 10

December 1984

Bemærk indkaldelsen til stiftende generalforsamling for Z80 Brugergruppens Indkøbsforening til den 6.12.1984. Bestyrelsen og redaktøren takker for indeværende år og ønsker alle foreningens medlemmer en god jul og et godt nytår. Vi ses igen til medlemsmøde den 24.1.1985 og det næste blad forventer vi udsendt midt i januar.



ALMINDELIGE OPLYSNINGER OM FORENINGEN

HENVENDELSE TIL FORENINGEN TIL FORRETNINGSFØREREN:

Benny Thomsen
Hyllingeriis 103
4050 Østby pr. Skibby
Telefon (02) 32 08 27

Hertil skal rettes henvendelse om indmeldelse, adresseforandring, salg af foreningens materialer (bånd, blade og programmer). Øvrige henvendelser af generel art til formanden. Stof og annoncer til foreningens blad sendes til Asbjørn Lind.

Indmeldelsesgebyr: 25.00 kr.
Kontingent 1.1.85 - 1.7.85. 75.00 kr.

Annoncering for medlemmer er gratis i Z80 NYT. For andre 250 kr. pr. A4 side.

Bestyrelsesmedlemmer:

Formand: René Hansen
Bispevangen 6 13.th
2750 Ballerup
Tlf. 02 65 59 76.
Kl. 19.00 - 21.00

Næstformand: Jesper Skavin
c/o K. Hansen
Tagensvej 205 4.th
2400 København NV

Teknisk redatør: Ole Hasselbalch
Vibeskrænten 9
2750 Ballerup
Tlf. 02 97 70 13.

Per Thomsen
Ulspilager 75
2791 Dragør

Frank Damgaard
Kastebjergvej 26A
2750 Ballerup

Redaktør for Z80 NYT:

Sidste frist for indlevering af stoft til næste nummer:

7.1.1985

Asbjørn Lind
Sidevolden 23
2730 Herlev
Tlf. 02 91 71 82. (20.00 - 21.00)
(man-, ons- og torsdag)

INDKØBSFORENING!

Z80 Brugergruppen har nu snart eksisteret i fem år, og køb og salg af varer udgør efterhånden en stor del af arbejdet i foreningen.

Derfor har vi fundet det praktisk at starte en indkøbsforening, og foretager herved :

INDKALDELSE TIL
STIFTENDE GENERALFORSAMLING I

Z80 BRUGERGRUPPENS INDKØBSFORENING

den. 6/12 - 1984 Kl. 19.30
i vores mødelokale på RECKU i forbindelse med
det almindelige medlemsmøde.

DAGSORDEN :

- 1) Valg af dirigent. *Asbjørn Lind*
- 2) Valg af referent. *de H.*
- 3) Ændringer og tilføjelser til foreslåede vedtægter. ✓
- 4) Vedtagelse af vedtægter. ✓
- 5) Valg af bestyrelse og suppleanter. *Fornum* *hestformet*
Valg af revisor og revisor suppleant. *Roué* *Jesper*
- 6) Eventuelt.

Efter generalforsamlingen byder vi på gløg og en hygge sludder, som vi plejer i december.

Vedtægter.

UDKAST TIL VEDTÆGTER FOR INDKØBSFORENINGEN :

PA. 1

Foreningens navn er Z80 Brugergruppens Indkøbsforening.

PA. 2

Foreningens formål er at varetage køb og salg af varer for foreningens medlemmer, og herved støtte Z80 Brugergruppen økonomisk.

PA. 3

Alle medlemmer af Z80 Brugergruppen er automatisk medlem af indkøbsforeningen.

PA. 4

Foreningen ledes af en bestyrelse bestående af formand, næstformand samt 3 menige medlemmer.

Valg af bestyrelse : Formand og næstformand skal vælges på generalforsamlingen blandt Z80 Brugergruppens bestyrelse. De tre menige medlemmer vælges på generalforsamlingen, blandt foreningens medlemmer. Valget gælder for 1 år.

Valg af formand og næstformand : Er der foreslået flere end en kandidat holdes skriftlig afstemning. Opnår ingen kandidat i første omgang over halvdelen af stemmerne, foretages en ny afstemning. Opnår heller ikke ingen ved denne afstemning over halvdelen af stemmerne, foretages bundet omvalg mellem de to, der ved anden afstemning opnåede de største stemmetal.

Valg af bestyrelsesmedlemmer : Er der flere kandidater end 3 foretages et prioreringsvalg. På stemmesedlen opføres 3 kandidater i den rækkefølge man ønsker dem valgt. Ved stemmelighed på 1. stemmer, vil antallet af 2. stemmer være afgørende. Ved stemmelighed på 2. stemmer, vil antallet af 3. stemmer være afgørende. Hvis der er stemmelighed i det samlede antal 1. 2. og 3. stemmer udtages kandidater i rækkefølge efter opnåede 1. stemmer.

PA. 5

Foreningen kan ikke forpligte Z80 Brugergruppen økonomisk.

UDKAST TIL VEDTÆGTER FOR INDKØBSFORENINGEN :
Forsat.

PA. 6

Ordinær generalforsamling afholdes hvert år sammen med Z80 Brugergruppens ordinære generalforsamling.

Indkaldelse til ordinær generalforsamling med angivelse af dagsorden, skal foretages senest 3 uger før dennes afholdelse. Forslag skal være tilsendt bestyrelsen senest den 15. marts.

Ekstraordinære generalforsamlinger kan indkaldes af formanden eller med angivelse af dagsorden af 3 bestyrelsesmedlemmer eller en 1/5 af medlemmerne med 14 dages varsel.

Ekstraordinær generalforsamling i Z80 Brugergruppen har ingen indflydelse på indkøbsforeningens bestyrelse.

PA. 7

Dagsorden til generalforsamling skal indeholde følgende punkter:

1. Valg af dirigent.
2. Formandens beretning.
3. Det reviderede regnskab fremlægges.
4. Indkomne forslag.
5. Valg af bestyrelse og 2 suppleanter, valg af revisor og revisorsuppleant.
6. Eventuelt.

PA. 8

Generalforsamlingen kan efter indstilling fra bestyrelsen ekskludere medlemmer, der handler mod foreningens interesse.

PA. 9

Foreningen kan opløses på følgende måde : Beslutningen skal foretages på en generalforsamling og bekræftiges af 3/4 af medlemmerne ved skriftlig tilkendegivelse, hvorefter den endelige beslutning om opløsning træffes på en ny generalforsamling.

Eventuelt opsparede midler tilfalder Z80 Brugergruppen.

PA. 10

Disse vedtægter træder i kraft den 6.12.1984 og kan kun ændres på en generalforsamling.

Vedtægtsændringer kan foretages, hvis de er opført som særligt punkt på dagsordenen og 2/3 af de fremmødte medlemmer stemmer for. Foreningen starter den 1.1.1985.



Symboltabel i HISOFT Pascal.

I Z80-NYT nr. 9 omtalte jeg i min artikel om tilkobling af BLS-Pascal's editor, at der var et problem med symboltabellen, der havde en fast størrelse.

For at løse problemet må vi lige se lidt på opbygningen af denne tabel. Den består dels af de prædefinerede navne (WRITE, READ, PROCEDURE o.s.v.) og dels af de navne, der er defineret af brugeren. Når Hisoft'en søger et bestemt navn, starter den bag fra. Adressen på det sidst definerede navn læses i adresse 2FB2+offset, der ved starten på kompileringen bliver initialiseret til den værdi, der står i adresse 30E5+offset. I adresse 2FB4+offset står adressen på den første ledige plads i tabelområdet. Denne bliver initialiseret til værdien, der står i adresse 30E7+offset.

Normalt vil den sidst nævnte adresse pege på området umiddelbart efter det sidste navn i den prædefinerede tabel. Der er imidlertid ikke noget i vejen for at ændre den til at pege på et andet område af lageret, og det er netop på den måde vi kan løse vores problem.

Når man er klar til at indkoble BLS-Pascal's editor (eller måske en anden) starter man Hisoft'en som sædvanlig, idet man på spørgsmålet 'Table size?' svarer 32 (jeg har dårlige erfaringer med mindre). Herefter retter man 30E7+offset (hvor offset nu er 32) til den første adresse efter kompileringen (5697+offset). Tekstbufferens start placeres et stykke ovenover (jeg har valgt adresse 6000H). Denne adresse er samtidig øvre grænse for symboltabellen, så den skal indføres i adresserne 241DH og 2477H.

Man kan selvfølgelig stadig risikere at symboltabellen vokser ud af den på forhånd afsatte plads (i mit tilfælde mere end 2K), så derfor tilføjes i editorens kontrol del en ekstra kommando, der kan flytte tekstbufferen og derved skaffe mere (eller mindre) plads til symboltabellen.

Når man nu er i gang med at lave nye smarte ting, kan man vel ved samme lejlighed lige ændre 'Memory'-kommandoen, så den også giver oplysning om symboltabellen. I min foregående artikel glemte jeg iøvrigt at oplyse, hvor man kunne finde slutningen på den kompilerede kode. Desværre står den ikke direkte noget sted, men under under kompileringen rummer HL' (altså det alternative registerpar HL) næste kompilerings-adresse, så der kan den hentes.

Til slut vil jeg endnu en gang gøre opmærksom på, at jeg har foretaget ændringerne på den specielle Nascom-version HP4TN af Hisoft'en, så alle adresser refererer til denne.

Svend Daugaard Pedersen (333)



DATABASER

Introduktion :

Først og fremmest må vi gøre os klart hvad der menes med begrebet database. Formelt kan det defineres som "en samling af relaterede oplysninger lagret på en sådan måde at adgang til disse sker via et entydigt udvælgelses kriterie fastlagt ved den struktur og metode der er valgt at anvende for databasen."

Dette siger os i første omgang ikke særligt meget, men et glimrende eksempel på en database er en telefonbog, her er en samling telefon abonnenter ordnet og sorteret på en sådan måde at der ikke er to der er helt ens, d.v.s. at her har vi det entydige udvælgelses kriterie. Samtidigt er de enkelte informationer relaterede til hinanden således at et efternavn som regel ikke er meget værd uden et fornavn, et gadenummer er ikke ret meget værd uden gadenavnet osv.osv.

Med andre ord kan vi opbygge en telefonbog efter en række faste retningslinier. F.eks skal efternavn stå først, derefter fornavn og så mellemnavne osv. osv. Altså en række byggesten der tilsammen udgør et individ. Man siger at et individ består af en række felter. Disse felter kan f.eks være :

```
+-----+-----+-----+-----+-----+-----+
! Efternavn ! Fornavn ! Mellemlavne ! Stilling ! Gade ! Nr !
+-----+-----+-----+-----+-----+-----+
```

samt en række flere der måtte være relevante i en telefonbog.

Ved nærmere eftertanke er det heller ikke så meget den formelle definition vi ønsker at klarlægge, men struktur og metode. Det er denne der er interessant i forbindelse med det praktiske arbejde, at implementere en database på en given datamat.

Der findes mildelst talt et hav af forskellige metoder og strukturer og det ville være uoverskueligt at skulle give en fyldestgørende gennemgang af dem alle her, det er der skrevet tykke bøger om.

Generelt kan man sige at der findes 3 hovedgrupper af metoder og strukturer, nemlig : indexerede databaser, listestyrede databaser, transformations databaser, og at de fleste andre metoder og strukturer er afledt af en eller flere af disse 3.

Jeg har valgt kort at beskrive henholdsvis indexerede databaser og transformations databaser for derefter at gå i dybden med en af de mindre kendte typer nemlig en listestyret database.

Indexerede databaser :

De fleste har sikkert hørt udtrykkene B-træ, ISAM, KRAM etc. etc. Det er alle betegnelser der hører hjemme i de indexerede databasers rækker.

KRAM :

Det der kendetegner disse databaser er at søgningen foregår via en nøgle bestående af en eller flere delmængder af de data man ønsker at holde styr på. Med delmængde forstås f.eks en del af et felt eller et helt felt. Denne nøgle er tilknyttet et og kun et individ, f.eks. en telefon abonnent. En nøgle kunne så f.eks være abonnentens efternavn. Denne nøgle gemmer vi i en særskilt fil som vi kalder indexfilen og sammen med nøglen gemmer vi en information der fortæller hvor det tilhørende individ befinder sig i hovedfilen (Den fil hvor vi gemmer alle vores individer). Samtidigt benytter vi en intern struktur i vores indexfil der gør det let og hurtigt for os at finde den nøgle vi gerne vil have fat i. Denne struktur kan f.eks være flg.

Index

```

+---+---+
| A | 33 |      ..
+---+---+
| B | 57 |      ..
+---+---+
| C | ..      +---+-----+-----+-----+
+---+      | 11 | Åkesson | 0 | 86 |
| D | ..      +---+-----+-----+-----+
+---+      ..
| E | ..      ..
.          +---+-----+-----+-----+
.          | 33 | Aakesson | 57 | 3 |
.          +---+-----+-----+-----+
+---+      ..
| Ø | ..      ..
+---+---+      ..
| Å | 11 |      +---+-----+-----+-----+
+---+---+      | 57 | Baldrian | 11 | 5798 |
+---+---+      +---+-----+-----+-----+

```

Her har vi først et under index bestående af bogstaverne fra A til Å, til hver af disse bogstaver, som reelt er nøgler, hører en række pile der peger ned i selve index hvor den rigtige nøgle står, og til hver af disse nøgler hører igen en pil der peger over i hovedfilen på det tilhørende individ samt en pil der peger på næste højere nøgle.

F.eks er Aakesson den første nøgle i rækkefølgen, det tilhørende individ findes i record nr. 3 i hovedfilen og den næste større nøgle findes i record nr. 57 i indexfilen. Her finder vi Baldrian hvis tilhørende individ findes i record 5798 i hovedfilen og hvor næste højere findes i record 11 i indexfilen. I record 11 finder vi Åkesson hvis tilhørende individ findes i record 86 i hovedfilen og som ikke har nogen der er højere hvorfor den peger på 0 som vi i dette tilfælde har defineret til at være ingenting.

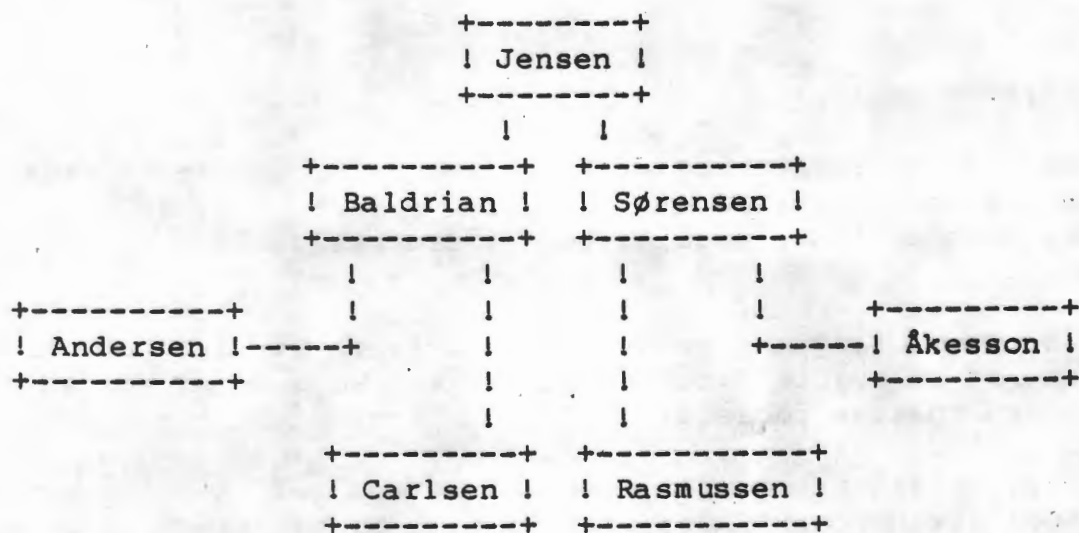
Proceduren for søgning er så at vi tager det første tegn i nøglen og finder det i under indexet, her står så adressen på det første individ der begynder med dette tegn hvorefter vi så læser listen sekventielt igennem efter "næste højere pile" samtidig med at vi sammenligner om vi har fundet det vi søger efter hvorefter vi hopper ud af søge rutinen, eller om første tegn er forskelligt fra det vi søger efter. Hvis dette er tilfældet har vi ikke fundet hvad vi søgte efter og vi må opgive og hoppe ud af søge rutinen med en meddelelse om dette. Dette gøres ligeledes hvis den nøgle vi aktuelt tester er større end den nøgle vi søger efter.

Hvis vi finder den nøgle vi søgte efter kan vi på nøglen aflæse hvilket record nummer vi skal læse i hovedfilen for at få alle oplysninger vi har lagret under denne nøgle.

B-TRÆ :

En anden måde at strukturere sine nøgler på kunne f.eks. være at sætte dem op i et binært træ. I et binært træ findes der, istedet for en pil der peger på "næste højere nøgle", 2 pile der peger efter et sæt fastlagte regler der simplificeret lyder :

Hvis den nøgle vi søger er større end den nøgle vi aktuelt tester går vi til højre i træet, ellers går vi til venstre i træet.



Her er der så igen på hver nøgle hæftet en pil der peger over i hovedfilen og fortæller hvilken record vi skal læse for at få fat i de oplysninger der tilhører den nøgle vi har fundet.

Såvidt så godt om indexerede databaser. Dem der evt. ønsker at vide mere kan læse kapitel 4 i "Algorithms + Datastructures = Programs" af Niklaus Wirth (Prentice Hall, New Jersey 1976) der fortæller om "Dynamic information structures" eller Anders Haraldsson's og Evald Skov Jensen's "Programmering i Pascal" hvor de kan kigge på eksempel 12F der beskriver et binært træ.

Transformations databaser :

Transformations databaser er opbygget på en helt anden filosofi, der hedder Hashing. Denne går ud på at man kan opstille en algoritme der ud fra en nøgle beregner en placering i hovedfilen, således at blot man har nøglen, så kan man altid regne sig frem til placeringen af individet i hovedfilen. Dette er imidlertid en sandhed med modifikationer idet det mig bekendt endnu ikke er lykkedes nogen at opstille en algoritme der entydigt kan identificere et individ, samtidigt med at den kan tage højde for navne kollision, f.eks. er der en del mennesker der hedder Jensen. Dog er der forsket en del i det og det er lykkedes at komme frem til en gruppe af algoritmer der fungerer rimeligt effektivt, idet disse sideløbende ved hjælp af en smule teori fra de indexerede databaser er istand til at håndtere navne kollision. Dog har dette medført at det ikke hver gang er muligt at regne sig frem til placeringen af individet i hovedfilen, men kun et af de individer der evt. måtte være med efternavnet Jensen hvorefter der skal søges sekventielt frem og tilbage for at finde det rigtige, idet man af gode grunde ikke kan forvente at lande der første gang man forsøger.

Jeg vil ikke her komme ind på hashing algoritmer idet dette er en videnskab for sig som jeg selv mener at kende for lidt til, til at jeg kan sige noget fornuftigt om det. Dog kan jeg henvise til David Gries "Compiler Construction for Digital Computers" (Wiley International 1971) afsnit 9.3.3 der omhandler Hashing.

Listestyrede databaser :

Listestyrede databaser bygger på at man har organiseret sine data som en liste der altid er sorteret hvorefter man kan søge den igennem med en binær søgning indtil man har fundet den rigtige nøgle.

Med en binær søgning menes at man bliver ved med at halvere mængden af relevante data indtil dette ikke mere er muligt eller indtil den ønskede nøgle er fundet.

Jeg har valgt at gennemgå en variation af de listestyrede databaser der hedder "Inverteret liste", da jeg synes at denne er et godt eksempel på hvilken filosofi der ligger bag, at den er nem at implementere, at den ikke fylder særlig meget og at søgning i en database opbygget omkring en inverteret liste er hurtig som død og djævel hvis man ønsker søgning efter flere kriterier der hver især hænger sammen.

Et af de steder man har brugt inverteret liste er ved teleadministrationernes såkaldte OPS system, populært kaldet 0034. Det er det system telefonisterne bruger til at trække oplysninger på når man ringer til nummer oplysningen. Det smarte ved dette system er at man kan give en vilkårlig mængde af oplysninger til at udpege det individ man søger hvorefter det stadig er muligt at begrænse antallet af mulige kandidater ret

kraftigt. Hvad angår den før omtalte hurtighed, kan jeg nævne at svartiden på systemet, fra det tidspunkt forespørgslen afsendes fra f.eks. Kbh. til svaret kommer tilbage til afsender af forespørgsel, ikke på noget tidspunkt må overstige 2 sek. Heri inkluderet transmissionstid fra Kbh. til Slet udenfor Århus, hvor det centrale anlæg står, og tilbage igen.

Hvis vi vender tilbage til telefon bogen er den opdelt i en række felter, et for stilling, et for fornavn, et for efternavn etc. etc. Systemet er så at man danner en liste af alle de forskellige muligheder der findes inden for et bestemt felt. Lad os f.eks tage feltet stilling. Her findes tømrer, journalist, specialarbejder osv. osv.

Den liste vi danner indeholder kun feltet specialarbejder 1 gang. Også selvom der er flere telefon abonnenter der er specialarbejdere, ligeledes er tømrer kun repræsenteret en gang osv. osv.

På hvert felt hæfter vi så en række bits, en for hver record der findes i hoved filen og alt efter om f.eks. tømrer findes i den record i hovedfilen der svarer til en given bit, eller ej, så sættes den pågældende bit til 1 hvis den findes og til 0 hvis den ikke findes. Dette medfører en struktur som f.eks. kan se nogenlunde sådan ud :

Index vedr. stilling	Index vedr. efternavn
+-----+-----+	+-----+-----+
! Blikkenslager ! 000001000 !	! Aakesson ! 100000000 !
+-----+-----+	+-----+-----+
! Bødker ! 000000101 !	! Andersen ! 010000000 !
+-----+-----+	+-----+-----+
! Snedker ! 010000010 !	! Baldrian ! 001000000 !
+-----+-----+	+-----+-----+
! Specialarb. ! 001010000 !	! Frederiksen ! 000100000 !
+-----+-----+	+-----+-----+
! Tømrer ! 100100000 !	! Iversen ! 000010000 !
+-----+-----+	+-----+-----+
	! Knudsen ! 000001000 !
	+-----+-----+
	! Larsen ! 000000100 !
	+-----+-----+
	! Sigurdsson ! 000000010 !
	+-----+-----+
	! Åkesson ! 000000001 !
	+-----+-----+

! 0001	! Aakesson	! Hans Henrik	! Tømrer	!
! 0002	! Andersen	! Flemming	! Snedker	!
! 0003	! Baldrian	! Lise Lotte	! Specialarb.	!
! 0004	! Frederiksen	! Arnold	! Tømrer	!
! 0005	! Iversen	! Lars	! Specialarb.	!
! 0006	! Knudsen	! Emil	! Blikkenslager	!
! 0007	! Larsen	! Knud	! Bødker	!
! 0008	! Sigurdsson	! Tjalfe	! Snedker	!
! 0009	! Åkesson	! Christian	! Bødker	! ...

Hvis vi nu ønsker at finde en specialarb. så er det bare at gå ind i det index der vedrører stillinger og finde specialarb. her kan vi så ved at teste hver enkel bit se at der findes en specialarb. i record nr. 3 og i record nr. 5 i hovedfilen.

Imidlertid ved vi også at vores specialarbejder hedder Baldrian til efternavn, så nu har vi to søgekriterier.

Det vi så gør, er at gå op i index for stilling og tage specialarb. ud, for derefter at gå op i index for efternavn og tage Baldrian ud. Disse 2 ser sådan ud :

! Specialarb.	! 001010000	!
! Baldrian	! 001000000	!

Hvis vi så laver en LOGICAL AND mellem disse to bit strenge, så får vi en ny bitstreng der udpeger de records hvori der både forekommer specialarb. og Baldrian.

! 001000000	!
-------------	---

I dette tilfælde i record 3. Denne record er så i dette tilfælde den eneste der opfylder begge vores kriterier for udvælgelse.

Denne metode kan i praksis anvendes til ligeså mange index man nu måtte ønske. Blot skal den LOGICAL AND der foretages, foretages på det sidst fremkomne resultat og det næste index, således at der altid kun forekommer 1 index i hver LOGICAL AND, bortset fra den første vi foretager.

So far so good. Nu har vi fundet ud af hvordan vi opbygger og søger i en INVERTERET LISTE. Så kommer det næste problem, hvordan vedligeholder vi så en INVERTERET LISTE. Det er straks en mere besværlig sag, vi kan ikke uden videre indsætte et nyt individ i listen uden at skulle til at sortere, danne nye nøgler og opdatere bit strenge. Det vil ikke kræve ret meget erfaring at fastslå at det vil tage alt for lang tid at gøre "ON LINE", d.v.s. sige at opdatere hvert enkelt individ efterhånden som det bliver tastet ind. Ligeledes vil det også tage for lang tid at ændre blot et enkelt individ, idet vi så evt. også skal ændre nogle af de tilhørende nøgler. Med andre ord må det siges at være mest fornuftigt at samle alle opdateringer (ændringer, sletninger og nye indføjelser) i en stor klump som man så opdaterer i "BATCH".

Så meget om INVERTERET LISTE.

Konklusion :

Konklusionen er at ikke alle database strukturer og metoder er lige velegnede til hvadsomhelst. F.eks er KRAM meget langsom i.f.t f.eks. INVERTERET LISTE hvad angår søgetid, til gengæld er den hurtigere hvad angår vedligeholdelse. B-Træ er hurtigere end begge de ovenstående til både opdatering og søgning, men tilgængæld lægger den en del begrænsninger nøglevalg. Her er INVERTERET LISTE derimod god idet den ikke kræver en entydig identifikation for at finde den rigtige, men den kan f.eks. finde en række mulige kandidater som det så er op til brugeren at vælge imellem. Med andre ord er det ikke muligt at foretrække en metode frem for en anden, det kommer helt an på hvad man skal bruge det til. Generelt kan man fremhæve følgende fordele og ulemper ved de forskellige metoder.

1 : KRAM :

Fordele :

Nem at vedligeholde.

God til at finde flere forskellige kandidater.

Nem at rette om når der går kage i den.

Ulemper :

Fylder meget

Langsom at søge i

Svær at implementere.

2 : B-Træ :

Fordele :

Nem at vedligeholde.

Hurtig at søge i.

Fylder lidt.

Ulemper :

Dårlig til at finde flere forskellige kandidater.

Svær at rette op.

Svær at implementere.

3: INVERTERET LISTE :

Fordele :

Hurtig at søge i.

God til at finde forskellige kandidater.

Nem at rette op.

Nem at implementere.

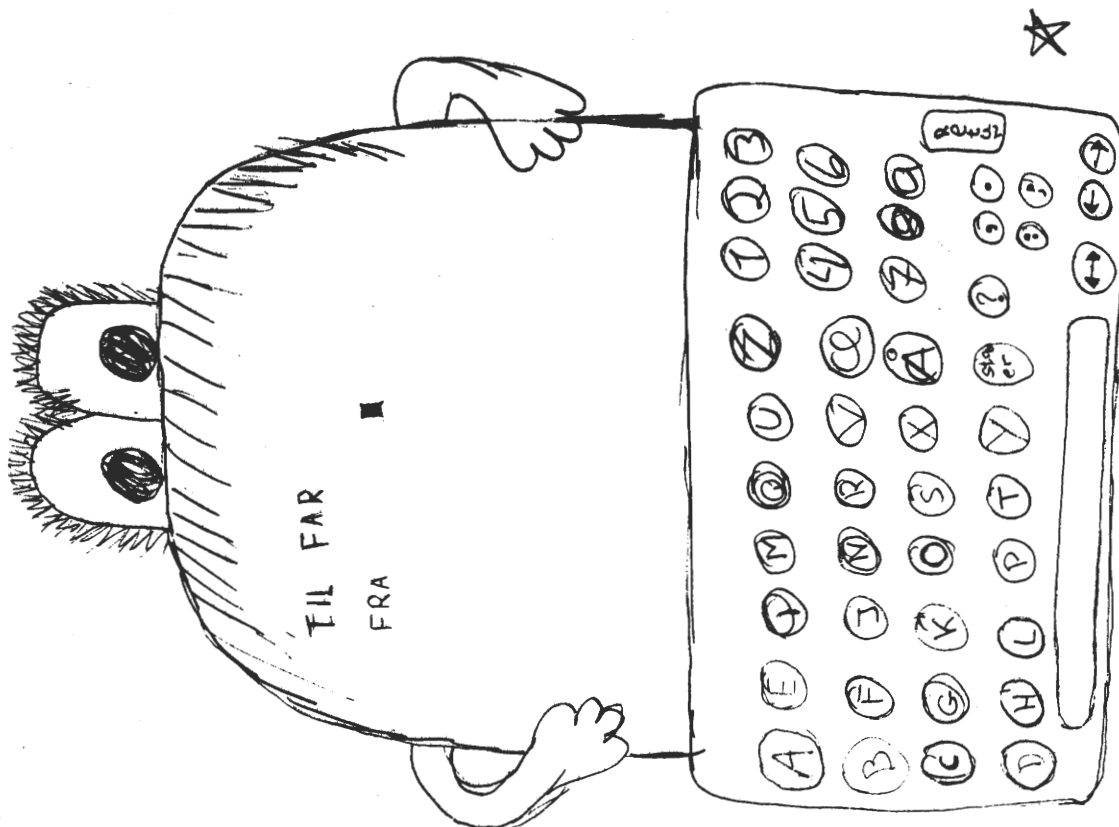
Ulemper :

Fylder meget

Svær at vedligeholde.

Ovenstående er selvfølgelig ikke den fulde sandhed idet disse metoder kan varieres i det uendelige og kombineres på alle mulige forskellige måder således at de lige nøjagtigt opfylder de krav, man måtte kunne stille.

Mads Westermann



Historien om IVC og SVC.

Der er nu begyndt at komme SVC-kort til Danmark fra England til 80-Bussen. Dette kort afløser (?) det tidligere IVC-kort også fra Gemini. Redaktøren har fra Piezodan lånt et SVC-kort i en måneds tid for at kunne vurdere det i forhold til IVC-kortet. Men da der ikke i dette blads spalter har været nogen omtale af IVC-kortet, vil jeg begynde med at beskrive dette.

IVC-kortet er en selvstændig lille computer på et enkelt printkort. Det indeholder en Z80A CPU'enhed, en 6845 CRT-kontroler, 4 Kb karaktergenerator (1/2 ROM og 1/2 RAM), 4 Kb RAM, der deles mellem arbejdslager og skærm. Dette styres af en MONITOR på 2 eller 4 Kb EPROM. Til kommunikation med omverdenen findes et par porte, der via bussen interfacer 80-Bus maskinen.

Baggrunden for dette valg er der to grunde til, 1) at skærmen ikke tager plads fra memoryen i hovedcomputeren og 2) ikke tager CPU tid. Hvis man havde beholdt skærmen som direkte adresserbar fra Z80, ville man opnå større hastighed, men den måtte man igen ned-sætte for at synkronisere med skærmopdateringen for at undgå 'screenflash'. For det andet ville skærmen blive opdateret, selv når det ikke var nødvendigt og det tager CPU tid! Når skærmen skal være 'avanceret' kræves yderligere RAM-plads til software, som tages fra hovedlageret. Hvis der skal scrolles, optages CPU'en i lang tid o.s.v.

Med et selvstændigt kort kan der blive tale om parallelkørsel mellem de to Z80-'maskiner'. Hovedcomputeren afgiver besked om f.eks. scrolling, hvorefter den fortsætter sit arbejde, mens IVC-kortet varetager selve den 'hårde' scrolling. Der spilles heller ikke dyr RAM plads til skærm og software, der jo er placeret på selve IVC-kortet. Når man så yderligere hæfter et keyboard på IVC-CPU'en, frigives der yderligere tid og mere RAM på hovedcomputeren. Det eneste problem, der kan komme på tale, er, hvis man vil indtaste hurtigere end IVC-kortet kan være klar til at modtage bogstaverne eller man fra hovedcomputeren vil skrive hurtige end kortet kan modtage tegnene. Det første problem klares ved en inputbuffer, det andet kan klares ved en buffer i hovedmaskinen med lidt styre-software.

På IVC-kortet sidder IVC-MONITOR version X.Y, der klarer alle problemer på kortet. Denne monitor har haft følgende udvikling, siden det første IVC-kort fremkom.

- 1.0 Er den første udgave af IVC-MONITOR.
- 1.1 En lidt forbedret udgave, der tillader 'nested' escape sekvenser i set, reset og test mode; og sletter nederste linie på skærmen ved 'delete linie og scroll'.
- 1.2 Der blev flyttet lidt om på arbejdslagerets brug. Input buffer blev mindsket til 64 tegn og stakken blev forøget.
- 2.0 Versionsnummer skifter til 2.Y for at vise, at noget helt nyt er sket. Nu kan man bruge et keyboard med funktionstaster. (Rotec - specielle funktionstastatur).
- 2.1 Nuværende versionsnummer. Nogen omstrukturering af koden så der kommer lidt mere fart på opdatering af skærmen. Specielt egnet for CP/M programmer, der undersøger om en nøgle er rørt for hver udskrivning af et tegn til

skærmen. Ved version 2.Y ligger softwaren i 2732 EPROM i modsætning til tidligere versioner, der lå i 2716 EPROM. En ny ESC ordre kommer til i form af spærring af nederste linie på skærmen. Den kan jo passende bruges til at vise, hvad funktionstasterne står for.

3.0 Er der ikke noget der hedder! Men muligheden er åben, fordi SVC-kortets monitorserie starter med 4.0 ! Vigtigt er det, fordi man softwaremæssigt kan undersøge, hvilken monitor der bruges i den aktuelle 80-Bus maskine og indrette den resterende software herefter.

IVC-kortet kan skrive alle standard ASCII-tegn (fra 20H til 7FH) og som et levn fra de gode gamle Nascom-dage, kan man skrive tegn ud enten som inverterede tegn eller i blokgrafik (samme specielle standard som Nascom). Dette kan gøres i to skærbredder (80 eller 48 - igen Nascom). Disse tegn kan koderne fra 80H til FFH. Dette område bliver defineret i RAM (den alternative tegngenerator), så man har også mulighed for at tegne sine egne tegn eller et helt tegnsæt. Dette bliver benyttet, som tidligere beskrevet, af forskellige programmer. Bl.a. HI-RES softwarepakken man kan få fra Henry's i London.

Alle ESC-sekvenser vil ikke blive gennemgået her, men jeg klipper en liste over kommandoerne ind. I denne liste vil de ordre, der kun kan udføres af SVC-kortet blive understreget.

Til IVC-kortet kan som nævnt sættes et parallelkeyboard og en lyspen. Hvis keyboardet skal bruges med funktionstaster, skal det være et keyboard, hvor funktionstasterne sender en escapesekvens af sted til kortet. Denne sekvens skal være (esc) (f) (80H..BDH), hvorefter IVC-MON finder oversættelsen til den tastede nøgle. Denne tegnfølge kan defineres som ALT!, hvad man har lyst til. Hvis man er træt af at skulle skrive spurvognskinneskidtskraberassistent-aspirant, kan dette ord lægges ind i en tast en gang for alle og kaldes frem ved et enkelt tryk. Som nævnt tidligere er der en keyboardbuffer på 64 byte, hvori man kan skrive forud - for programmet eller forud for afhentning af tegn fra IVC-kort, hvis man er hurtig nok.

SVC-kort.

Dette nye kort fra Gemini er en videreudvikling fra IVC-kortet - heldigvis. Det er opad kompatibelt med IVC-kortet - dvs. at alle programmer, der kan køre på IVC, også kan køre på SVC, dog undtaget de programmer, der benytter IVC's 48 tegn skærm, hvor SVC jo kører med 40 tegns bredde som option. Dog skal programmer, der bruger muligheden for at køre brugerdefineret software fra IVC/SVC-kortet, skrives om.

Ud over IVC-kortets muligheder kommer der følgende direkte forbedringer: en Z80B CPU (6 MHz), skærmens RAM er næsten firdoblet, hvilket giver mulighed for en pixelopløsning på 256*256, mulighed for serielt keyboard, en programmerbar UART kan sættes på en ekspansions sokkel og få SVC til at blive en hel selvstændig terminal (dog er der ikke på nuværende tidspunkt software i SVC-MON til at understøtte denne mulighed), en beep'er og et software ur.

Videokortet har 2 Kb arbejdslager og som nævnt 8 Kb skærmlager dog med modifikation, idet tegngeneratorerne ligger i de samme 8 Kb. På følgende side er den fulde kommandotabel for SVC-kortet.

HEX	ASCII	DECIMAL	COMMAND (Special key provided)
-----	-------	---------	--------------------------------

General

07	~G	7	<u>Bell - sound on-board buzzer</u>
08	~H	8	Backspace (BACK SPACE)
0A	~J	10	Linefeed
0D	~M	13	Carriage return (RETURN)

Cursor Movement

1C	~\	28	Cursor left (←)
1D	~]	29	Cursor right (→)
1E	~^	30	Cursor up (↑)
1F	~/	31	Cursor down (↓)
1B 0C	<ESC> ~L	27 12	Cursor home
1B 3D...	<ESC> "="...	27 61...	Cursor addressing

Additional cursor operation

1B 3F	<ESC> "?"	27 63	Return cursor coordinates and character
1B 44	<ESC> "D"	27 68	Delete cursor
1B 45	<ESC> "E"	27 69	Enable cursor
1B 59...	<ESC> "Y"...	27 89...	Define cursor type

Screen editing

0B	~K	11	Delete line and scroll up (SHIFT/↑)
0E	~N	14	Insert line (SHIFT/↓)
16	~V	22	Delete character in line (SHIFT/←)
17	~W	23	Insert character in line (SHIFT/→)
1A	~Z	26	Clear screen & home cursor
1B 16	<ESC> ~V	27 22	Delete character in screen
1B 17	<ESC> ~W	27 23	Insert character in screen
1B 25	<ESC> "~"	27 37	Delete to end-of-screen
1B 2A	<ESC> "~"	27 42	Delete to end-of-line

Screen format

1B 31	<ESC> "1"	27 49	Select 80 wide format
1B 32	<ESC> "2"	27 50	Select 40 wide format
1B 33	<ESC> "3"	27 51	Select user-defined format
1B 34	<ESC> "4"	27 52	Select graphics mode
1B 46...	<ESC> "P"...	27 70...	Define user format
1B 70...	<ESC> "p"...	27 112...	Select display/update partition
1B 42	<ESC> "B"	27 66	Blank screen
1B 56	<ESC> "V"	27 86	Video on (Unblank)
1B 49	<ESC> "I"	27 73	Video Invert screen
1B 4A	<ESC> "J"	27 74	Video normal screen (non-inverted)
1B 41	<ESC> "A"	27 65	Alternate char. generator is default
1B 4E	<ESC> "N"	27 78	Normal character generator is default
1B 4D	<ESC> "M"	27 77	Memory lock on
1B 4F	<ESC> "O"	27 79	Memory lock off

Character set

1B 67...	<ESC> "g"...	27 103...	Set language
1B 43...	<ESC> "C"...	27 67...	Define character
1B 63...	<ESC> "c"...	27 99...	Define character set
1B 62...	<ESC> "b"...	27 98...	Read back character set
1B 47	<ESC> "G"	27 71	Construct block graphics character set
1B 48	<ESC> "H"	27 72	Copy normal char. set to alternate & invert
1B 68	<ESC> "h"	27 104	Copy normal char. set to alternate char.gen

Block graphics

1B 47	<ESC> "G"	27 71	Construct block graphics character set
1B 52...	<ESC> "R"...	27 82...	Reset point X,Y
1B 53...	<ESC> "S"...	27 83...	Set point X,Y
1B 54...	<ESC> "T"...	27 84...	Test point X,Y

256x256 graphics

1B 52	<ESC> "R"...	27 82...	Reset point X,Y
1B 53...	<ESC> "S"...	27 83...	Set point X,Y
1B 54...	<ESC> "T"...	27 84...	Test point X,Y
1B 60...	<ESC> "L"...	27 108...	Draw/Erase/Complement a line
1B 6F...	<ESC> "o"...	27 111...	Draw/Erase/Complement a circle
1B 6D...	<ESC> "m"...	27 109...	Move cursor to X,Y
1B 77	<ESC> "w"...	27 119	Fill a polygon
1B 64...	<ESC> "d"...	27 100...	Graphics screen dump

Keyboard

1B 66...	<ESC> "f"...	27 102...	Define function key(s)
1B 66 64	<ESC> "fd"	27 102 68	Reset function key definitions
1B 66 3F	<ESC> "f?"	27 102 63	Return current function key definitions
1B 6B	<ESC> "k"	27 107	Test keyboard status
1B 4B	<ESC> "K"	27 75	Get keyboard character
1B 58	<ESC> "X"	27 88	Get one line of input
1B 3E...	<ESC> ">"...	27 62...	Download soft-key display

Clock

1B 74...	<ESC> "t"...	27 116...	Set clock time
1B 74 3D..	<ESC> "t="...	27 116 61.	Position clock display
1B 74 45	<ESC> "tE"	27 116 69	Enable clock display
1B 74 44	<ESC> "tD"	27 116 68	Disable clock display
1B 74 3F	<ESC> "t?"	27 116 63	Return clock time

Miscellaneous

1B 5A	<ESC> "Z"	27 90	Return contents of current line
1B 61...	<ESC> "a"...	27 97...	Set/reset attributes
1B 57...	<ESC> "W"...	27 87...	High speed write to display
1B 4C...	<ESC> "L"...	27 76...	Load user program
1B 55	<ESC> "U"	27 85	Execute user program
1B 50	<ESC> "P"	27 80	Return light pen coordinates
1B 76	<ESC> "v"	27 118	Return SVC-MON version number

De nævnt 8 Kb kan deles op på følgende måde:

8 K	+-----+ Alternate character generator	+-----+ Alternate character generator	+-----+
6 K	+-----+ standard character generator	+-----+ standard character generator	+-----+
4 K	+-----+ 40*25 display part 3	+-----+ 80*25 display part 1	+-----+ 256*256 pixel resolution
2 K	+-----+ 40*25 display part 1	+-----+ 80*25 display part 0	+-----+
1 K	+-----+ 40*25 display part 0	+-----+	+-----+

Ved normal brug (i alfanumerisk mode) kan de alternative tegn-områder defineres på forskellig måde. Ved start kopieres tegnsættet fra SVC-MON over i RAM efter det forvalgte nationalitets tegnsæt. Herefter kan man bruge det alternative tegnsæt på fire forskellige måder: 1) inverteret video, 2) halvtone baggrund, 3) reduceret lysstyrke og 4) blinkende eller i en kombination af flere f.eks. er det muligt med blinkende halvtone baggrund. Det er dog kun muligt at vælge en måde for hele skærmen, idet det mest betydende bit i tegnet vælger det alternative karaktersæt. Tegnsættet kan vælges softwaremæssigt enten ved start eller undervejs til følgende sprog: engelsk, svensk, dansk, tysk, fransk og amerikansk.

I alfanumerisk mode kan der skiftes mellem de forskellige dele af displayet (4 for 40 tegns bredde og 2 for 80 tegns bredde) på en måde, så man kan læse i en og skrive i en anden eller omvendt. Man kan også skifte mellem skærmene. Dette ville være særdeles smart, hvis man har en hjælpeskærm liggende i part 1, som man kan kalde ved hjælp af et tryk på en nøgle (måske en funktions-tast). Det kommer med det samme og den oprindelige tekst forsvinder ikke, men kommer tilbage straks efter igen med et fornyet tryk (programmeringsteknik).

Hastigheden på SVC-kortet er blevet meget bedre end IVC-kortets. Jeg har foretaget nogle forsøg og resultatet blev en hastighedsforøgelse på 55 % ved udskrift til skærm!!

Uret på kortet kan sættes ved start og det vil så gå frem til 59:59:59 før det skifter til 00:00:00 - lidt idiotisk. Men det virker, men har dog en stor ulempe: det går i stå, når du slukker for computeren.

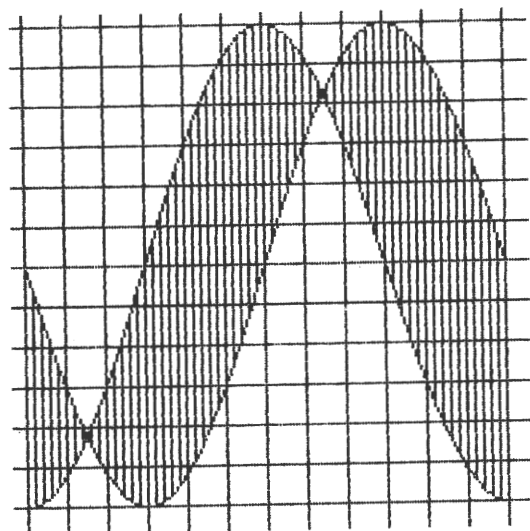
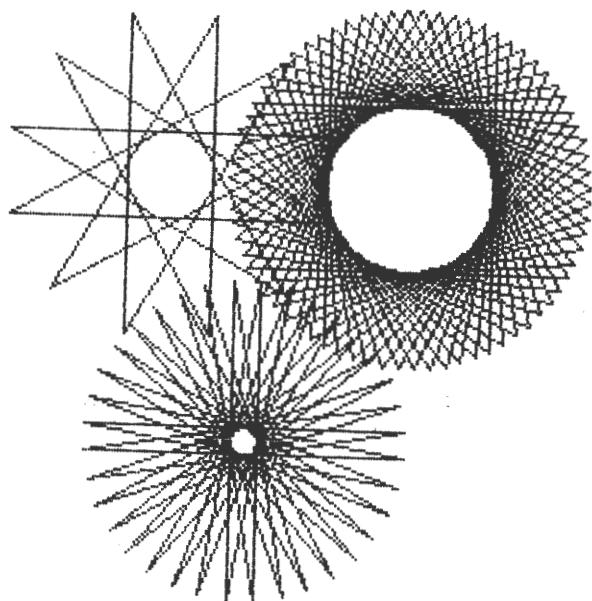
Monitoren er blevet forbedret, så det er vanskeligere at få den til at hænge, men det er dog stadig muligt, selvom det er blevet sværere. Det er gjort ved at SVC-kortet, når det er blevet bedt om at sende data til 80-bussen, stadig undersøger, om der er indkommende data. Hvis der er det afbryder det arbejdet og begynder at behandle de indkomne tegn og kommandoer.

I grafikmode (256*256 pixel) er der indbygget nogle assembler rutiner til stregtegning, cirkeltegning og fyld_område. Disse virker godt og sikkert med undtagelse af stregtegningen, der i specielle tilfælde wrapper rundt og laver nogle utilsigtede streger hen over skærmen. Men er man klar over problemet, kan man i styresoftware undgå at fremprovokere disse utilsigtede streger.

I grafikmode bruges den oprindelige tegngenerator i SVC-MON - det vil sige, at man kun kan bruge engelske tegn ! Lidt ærgeligt! Men det må vente til vi får lavet en dansk udgave af SVC-MON med danske tegn. Antallet af tegn bliver nedsat til 32 pr. linie. De mere avancerede skærmmkommandoer forsvinder i grafikmode (EOS, EOL), derimod får man nogle nye. Blandt andet en der dumper skærmen som en lang tegnfølge - enten lodret eller vandret. Styresoftware skal selvfølgelig sørge for den fornødne rutine til afhentning og initiering af printer m.m.

Herunder er udskrevet nogle tegninger, der er lavet i SVC-kortets grafikmode. De er alle skrevet i MYREsnak, et LOGO ligende sprog udviklet i USA. Det er jo snart sådant, at alle grafiskskærme skal kunnes arbejde i LOGO. Programmet er i sin grundversion lavet af Peter Villadsen til Butler maskinerne.

Asbjørn Lind.



Opbygningen af navnekartotek.

Når Z80-brugergruppen udsender en fortegnelse over klubbens medlemmer, er disse normalt udskrevet efter to forskellige principper. Dels står medlemmerne ordnet alfabetisk og dels efter bopæl (d.v.s postnummer). Hvoledes gøres det?

Nu er der sikkert nogle, der tænker, at nu kan de endelig få at vide, hvorledes foreningen har opbygget deres navnekartotek. Til dem må jeg straks sige, at det får de ikke ved at læse denne artikel. Jeg ved det nemlig overhovedet ikke. Det jeg derimod mener at vide, er hvorledes man kunne gøre det.

Problemet med et kartotek af den nævnte type er den sorterede udskrift. Ganske vist findes der en del forskellige rutiner, der kan foretage en sortering, men hvis kartoteket rummer så mange navne, at ikke alle kan være i lageret på een gang, vil det være en temmelig tidkrævende proces at foretage en sortering. Specielt vanskeligt bliver det naturligvis når man ønsker flere forskellige principper for sortering anvendt i udskriften.

Den måde man kan løse problemet på, er at benytte en datafil med direkte tilgang (random access file). Hver post i filen skal nu foruden de almindelige medlemsoplysninger indeholde en hægte (d.v.s. en pegepind) til det næste medlem. Herved bliver de enkelte medlemmer hængt sammen i een lang kæde på en sådan måde, at hvis man starter i den ene ende af kæden og følger denne, vil man få medlemmerne i den rigtige rækkefølge. Denne rækkefølge vil normalt ikke være den samme som den, hvori medlemmerne er blevet indtastet, og dermed heller ikke den samme, som rækkefølgen af posterne i den direkte fil.

Når man har forstået dette princip vil man se, at der naturligvis ikke er noget i vejen for at hver post rummer flere forskellige hægter, der kæder de enkelte medlemmer sammen i overensstemmelse med flere forskellige sorteringsprincipper.

I datafilen vil der normalt være såvel poster, der rummer medlemsoplysninger, som poster, der ikke er i brug (frie poster). De sidste kan dels være poster, der endnu ikke er taget i brug, og dels poster, der er blevet frigjorte fordi medlemmet har meldt sig ud af foreningen. Det er mest praktisk også at kæde disse frie poster sammen.

Foruden disse to grupper af poster, bruges første post i datafilen til at angive starten på de enkelte kæder - såvel medlemskæderne som kæden af frie poster.

Lad os, for at få et bedre overblik over princippet, se på et konkret eksempel. Vi tænker os et meget lille kartotek med plads til kun seks navne, d.v.s. filen rummer syv poster. Vi ønsker at kunne få udskrevet samtlige medlemmer i såvel alfabetisk rækkefølge som medlemsnumrerrækkefølge og efter adresse.

Hver post rummer her fire tal. De første tre rummer hægterne til de tre medlemskæder og det sidste bruges til at hægte de frie poster sammen. I det viste eksempel er der to frie poster.

post 1:	4	4	6	5		
post 2:	6	1	1	-	Lindschow, Sven Erik	nr. 421
					5200 Odense	
post 3:	2	6	4	-	Lind, Asbjørn	nr. 37
					2730 Herlev	
post 4:	3	3	2	-	Hasselbalch, Ole	nr. 8
					2750 Ballerup	
post 5:	-	-	-	7		
post 6:	1	2	3	-	Pedersen, Svend Daugaard	nr. 333
					2000 København F	
post 7:	-	-	-	1		

Man bemærker, at den sidste post i hver kæde er hægget sammen med post 1. Dette giver en mulighed for at afgøre, hvornår man har nået slutningen af en kæde.

Når et nyt medlem skal ind i kartoteket, må man først tage en fri post, hægte denne ud af kæden af frie poster, fylde den med data og endelig hægte den ind i hver af medlemskæderne.

Hvis man omvendt ønsker at fjerne et medlem, må man først hægte den pågældende post ud af medlemskæderne og derpå hægte den ind i kæden af frie poster.

Inden man går igang med at skrive et lille program, der kan foretage disse operationer, er det nyttigt selv at prøve at foretage dem. Prøv f.eks. at indføje vor højt ærede formand i kartoteket (det er jo også noget rod at han ikke er med). Resultatet bør se således ud:

post 1:	5	4	6	7		
post 2:	6	1	1	-	Lindschow, Sven Erik	nr. 421
					5200 Odense	
post 3:	2	6	4	-	Lind, Asbjørn	nr. 37
					2730 Herlev	
post 4:	3	3	5	-	Hasselbalch, Ole	nr. 8
					2750 Ballerup	
post 5:	4	2	2	-	Hansen, Rene	nr. 344
					2750 Ballerup	
post 6:	1	5	3	-	Pedersen, Svend Daugaard	nr. 333
					2000 København F	
post 7:	-	-	-	1		

De to eksempler kan også bruges til at efterprøve afhægning. Prøv f.eks. at fjerne formanden igen (sorry! mr. chairman).

Man ser, at de to operationer "hægte ind" og "hægte ud" er helt centrale i et kartoteksprogram, der anvender kædning. De skrives bedst som procedurer, dels fordi det altid er en god ide at skrive små veldefinerede operationer som procedurer, og dels fordi disse operationer skal bruges flere steder.

Nedenfor er står indledningen til et kartoteksprogram, der viser hvorledes netop disse procedurer kan laves. Programmet kræver nogle yderligere kommentarer.

Jeg ved ikke, om der i Pascal findes standarder for rutiner til behandling af direkte filer, men jeg er sikker på at rutinerne i min Pascal ikke er standard (det er en Hisoft 4TN, der er udvidet med rutiner til at understøtte Nasdicos filsystemet). Derfor må jeg kort fortælle om betydningen af visse rutiner.

SEEK(linje,post) positionerer på postnummer "post" i den fil, til hvilken der er åbnet en forbindelseslinje med nummeret "linje". GET(linje,var) læser denne post ned i variabelen "var" og PUT(linje,var) skriver ud på denne post.

Procedurerne er ret kompakt programmerede, og det kræver derfor nok lidt granskning at forstå dem, men de følger helt princippet som det er blevet skitseret ovenfor.

Til slut vil jeg lige nævne, at princippet med kædning også anvendes af visse diskoperativsystemer til at holde styr på de enkelte sektorer på disketten. Dette gælder vist bl.a. CP/M. En af fordelene herved er, at det er lettere at udvide en fil. Hvis systemet også kæder de frie sektorer sammen, vil sletning af en fil umiddelbart frigive de benyttede sektorer (man skal altså ikke først udfører en pakkeoperation). Ulempen er, at filerne på disketten efterhånden bliver spredt ud over hele disketten, hvilket kan nedsætte indlæsningshastigheden ganske betragteligt.

FORTSÆTTES PÅ SIDE 23

Svend Daugaard Pedersen (333)



Spar penge. Og bekymringer.

3M disketter. Med sikkerhed 100% fejlfri.

Og med livsvarig garanti.

Testet på 327 måder. Og fås til de fleste EDB-systemer. 3M disketter. Én bekymring mindre.

Ring og få et ekstra godt tilbud hos

Z80 brugergruppen

v/ René Hansen

Bispevangen 6 13.th

2750 Ballerup

02 65 59 76

3M
diskettes

Et problem mindre.

Få et ekstra godt tilbud på 3M disketter.

De er med sikkerhed 100% fejlfri. Og

har livsvarig garanti.

Testet på 327 måder.

Og fås til de fleste EDB-systemer.

**3M disketter.
Én bekymring
mindre.**

3M
diskettes

PROGRAM KARTOTEK;

TYPE POST=RECORD

```

    Link: ARRAY[1..3] OF INTEGER; (* sorteringskaeder *)
    FreeLink: INTEGER;           (* haegte til naeste frie post
    Nr: 1..9999;                 (* medlemsnummer *)
    Navn: ARRAY[1..30] OF CHAR;
    Addr: ARRAY[1..20] OF CHAR;
    By: ARRAY[1..20] OF CHAR;
    Tlfnr: ARRAY[1..9] OF CHAR;
END;
```

PROCEDURE LinkIn(memlem: POST);

VAR inf,fgpost,npost: POST;

nynr,fgnr,nnr,i: INTEGER;

passed: BOOLEAN;

BEGIN

SEEK(line,1);GET(line,inf);

nynr:=inf.FreeLink; (* postnummer til det nye medlem *)

SEEK(line,nynr);GET(line,npost);

inf.FreeLink:=npost.FreeLink; (* 1. frie post *)

SEEK(line,1);PUT(line,inf);

FOR i:=1 TO 3 DO

BEGIN (* find for hvert i den rette plads i kaeden *)

SEEK(line,1);GET(line,inf);

nnr:=1;npost:=inf;

REPEAT

fgnr:=nnr;nnr:=npost.Link[i];fgpost:=npost;

SEEK(line,nnr);GET(line,npost);

CASE i OF

1: passed:=(npost.Navn>medlem.Navn);

2: passed:=(npost.Nr>medlem.Nr);

3: passed:=(npost.By>medlem.By)

END;

UNTIL passed OR (nnr=1);

medlem.Link[i]:=nnr;fgpost.Link[i]:=nynr;

SEEK(line,fgnr);PUT(line,fgpost);

END;

SEEK(line,nynr);PUT(line,medlem);

END;

PROCEDURE LinkOut(postnr: INTEGER);

VAR inf,glpost,npost: POST;

nfp,i: INTEGER;

BEGIN

SEEK(line,1);GET(line,inf);

nfp:=inf.FreeLink;inf.FreeLink:=postnr;

PUT(line,inf);

SEEK(line,postnr);GET(line,glpost);

glpost.FreeLink:=nfp;

FOR i:=1 TO 3 DO

BEGIN

SEEK(line,1);GET(line,npost);

WHILE npost.Link[i]<>postnr DO

BEGIN

SEEK(line,npost.Link[i]);GET(line,npost);

END;

npost.Link[i]:=glpost.Link[i];glpost.Link[i]:=1;

PUT(line,npost);

END;

SEEK(line,postnr);PUT(line,glpost);

END;



Angående båndoptagermodem.

Siden det sidste Z80-Nyt udkom, har jeg ikke modtaget en eneste forespørgelse om modem. Min konklusion er da, at der ingen interesse er, og vi dropper fremstillingen. Det vil dog være muligt, at låne en film mod et passende depositum. Henvend dig til René eller undertegnede, så skal vi se, hvad vi kan gøre.

Vi står så i den situation, at det slet ikke bliver så nemt med overførsel af programmer fra programbiblioteket. Vi må vel lade dem trykke, når det drejer sig om kildetekster til basic, pascal eller assembler. Det bliver noget vanskeligere for de programmer, der ligger som objektkoder. Skyldes det mon, at det er sådan, at mange af vore medlemmer har anskaffet sig et disksystem??

Det kan lade sig gøre med Kansas City, men det skal tilpasses til ens egen computer. Jeg kører selv uden besvær 2400 Baud, men det er også en god båndoptager, specielt trimmet til de høje frekvenser. Jeg kan endda køre 4800 Baud med en god, gammel Revox, men så kører jeg også med stor båndhastighed. Er der nogen, der har eksperimenteret med dette? Jeg tror, der kunne være interesse for en sådan konstruktion.

Jeg så gerne nogle flere sendte artikler, så vi kan tilrettelægge et godt blad.

O.H.

Jeg har ikke modtaget noget svar fra min henvendelse vedrørende de basicprogrammer, der bliver overført på mellembølge. Derfor er der intet nyt i denne sag, og jeg tror heller ikke der vil ske mere, før jeg skriver igen. Det drejede sig om den mellembølgestation, der bruges af Radio Hilversum. Er der nogle der kunne komme med nogle oplysninger?

Det er mig bekendt, at der er forsøgt dataoverførsel på en af vore lokalradioer, så der er intet nyt under solen. Hvem har nogle gode kontakter, så vi også kunne prøve dette?

O.H.



Jeg har en strømforsyning stående, som et af vore medlemmer, for lang tid siden, ville have beskrevet. Jeg har ikke navn og adresse på vedkommende, så derfor denne efterlysning. Vedkommende boede vist i Hellerup. Gider du sætte dig i forbindelse med mig.

Venlig hilsen Ole Hasselbalch.



Korrektion: I sidste nummer omtalte jeg nye volumen i brugerbiblioteket. Under Sig/M 162 skrev jeg bare CCP, men det kan jo være mange ting. Og det det ikke var var CCP - Command processor men derimod Concurrent Compiler Pascal-S !! Undskyld. Asbjørn.



Announce *** Announce *** Announce *** Announce

Video outputenheder der tilbydes os, er normalt enten for store eller med begrænset muligheder. Sammen med et par kolleger har jeg i 2 år bygget Nascom på europakort. Hertil har vi nu udviklet et videokort til maskinen. Printet er allerede lagt ud og fremstillet ifå eksemplarer, men nu kommer så spørgsmålet. Hvor mange vil være interesserede i at købe et sådant print, hvis det er gennempletteret og med loddemaske?

Hvis du er interesseret så send et brev eller kort til mig (ingen penge endnu - tak).

Hvis tilslutningen er stor nok, vil jeg få det lavet - der skal være 20 aftagere. Prisen kommer til at ligge omkring 200 kr for printet, programrom 150 og 8 Kb karakterrom 200 kr.

Prommerne er option, så dem behøver man ikke at købe, hvis man selv vil skrive softwaren.

ITC kort specifikationer: Intel 8085 CPU, Intel videokontroller og Intel portkreds. Fast program og fast karaktersæt i Eprom. 80*25 display. 4 forskellige tegnsæt med hver 128 tegn: Normal, kursiv, bold og grafik. Visual attributes: Blink, highlight og invers. Data kan scrolles, slettes, indrammes på hele skærmen eller i vinduer. Koderne omfatter bl.a. indsæt space, slet karakter, slet to end of line/screen, cursor til x,y.

Kortet er standard eurokort med målene 10*16 cm. Data overføres via en parallelport på printet. Udgangen er komposite video.

T. Bundgård, Baneleddet 13, 8320 Mårslet

CP80 matrixprinter, 80 CPS, højopløsningsgrafik, danske tegn, traktorfeed, Epson kompatibel 3900.00 kr.

Elektron 'Mini-BBC' med samme farver, grafik og lyd (næsten) som en BBC/B. Nypris 4000.00 kr. sælges for 3300.00 kr.

Lars Jensen, Nørrebrogade 221,3 2200 N. 01 85 31 51 (efter 16).

Starprinter (DP 8480) matrixprinter sælges for 2800.00 kr.
Jens Wenberg 06 82 68 80

Nascom I (lettere defekt) med T2, Nassys 1 & 3, 2bufferkort, 1 RAM A kort, Nascom I tastatur + ekstra tastatur med numerisk blok og cursorpile + diverse tilbehør (hard- og software) 1200.00 kr.

8 stk 4118 statisk RAM 400.00 kr.

Henvendelse Finn Christensen 02 643104

2 stk. Friden Flexwriter terminaler med hulstrimmellæser og - puncher sælges eller byttes med elektronisk hardware. tilbud til Erik Rosenlund, Vognborgsvej 1, 2400 NV 01 19 04 08

Nascom II med 1 drive Midicos, monteret i chassis med motherboard, 64 Kb RAM, 4 PIO, blæser, power 5V/5A, 12V/1A. 1 meget lidt defekt Motorola printer med 5 ruller papir. Keyboard indbygget i kasse. 9" monitor. Software + manualer. samlet pris 9000 kr.
Henvendelse hos Bo Issov 02 94 64 92 ★

Nascom 2 med 32/64 Kb, EPROM-kort, amatørmodem, floppydiskkontroller, 8 " drive. Indbygget i lækkert aluminiumskasser. Meget software på disketter og i EPROM (incl manualer). Printer og Monitor kan evt. følge med. Seriøse henvendelser til medlem 300
Benny Sørensen 02 64 05 90 (efter 18) ★

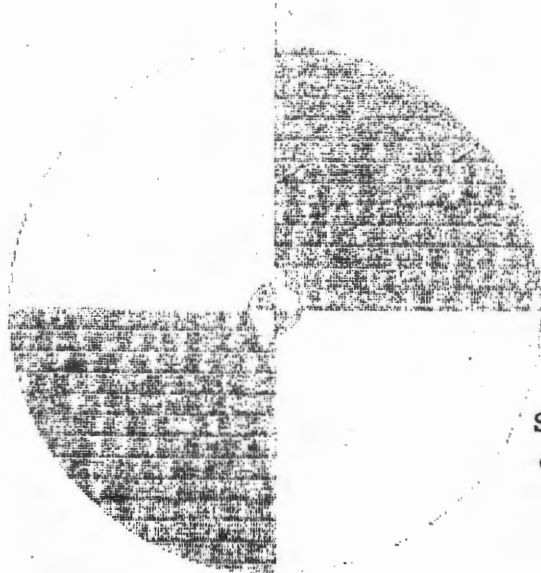
Nascom 2 med CP/M, 64 Kb, 8" DDS (1.2 Mb) diskettestation, Itoh printer. Soroc terminal med to intensiteter, kan køre Wordstar. CP/M og Nassys 3 samt mange anvendelsesprogrammer til begge systemer. Prisiidé 13000.00 kr.
Philips minidigital båndoptager med 6 bånd med prog. 500.00 kr.
Bo Sørensen 02 81 38 46 (privat) 02 88 22 22 lok4412 (arbejde) ★

Ønskes: NAP i 2*2716 EPROM incl. manual Moritsen 01 10 52 47 ★

IVC-kort ønskes Torsten Jacobsen 01 56 59 26 ★

Ønskes: Skrivemaskine, der kan sættes til en computer og samtidig bruges som almindelig skrivemaskine Simon Tsizzig 01 34 85 36 ★

Husk medlemmøden den 24.1.1985
Det sædvanlige sted
R E K U



Skærmkopi af SVC-kort, hvor linittegning, cirkeltegning og udfyldning er benyttet.