



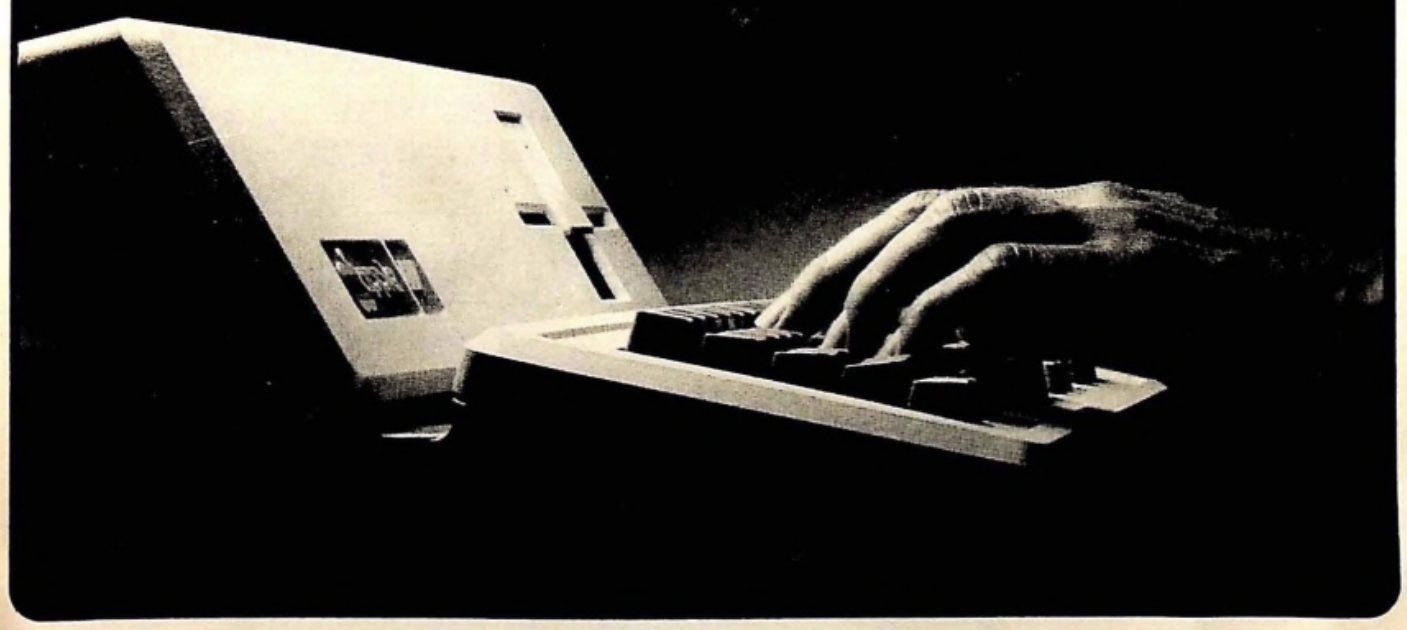
USER NEWS

4.82

organ för Apple User Group of Sweden

17:-

VisiSerien
Bildpackning
Recensioner



DUAL 80/35 TRACK FLOPPY DISK
UNIT FOR THE APPLE II COMPU



- TWO 5¼ INCH 80 TRACK DISK DRIVES IN ONE CABINET
- EITHER DRIVE CAN BE SWITCHED TO 35 TRACK OPERATION
- IN 35 TRACK MODE ALL APPLE SOFTWARE INCLUDING ½ TRACK SOFTWARE CAN BE READ
- IN 80 TRACK MODE TOTAL CAPACITY IS 655,000 BYTES
- CONNECTS TO AN APPLE II DISK CONTROLLER
- MAINS POWERED – TAKES NO CURRENT FROM THE COMPUTER
- POWER AND COMPUTER INTERFACE CABLES INCLUDED

VÄLKOMMEN
TILL EN DEMONSTRATION

Administratörerna

I Stockholm och Västerås
Box 20058, 161 20 Bromma, Tel. 08-985420

APPLE USER NEWS

I detta nummer:

- 6 Nyheter
- 7 Flickan & Odjuret
- 13 Text file fast loader
- 16 Bokrecension
- 20 Nya VisiCalc
- 25 Spel
- 28 DIF
- 30 EZ-draw
- 32 Bildpackning
- 35 VisiDex
- 40 Apple ///
- 47 Bokhyllan
- 52 Nya Monitorkommdon
- 58 DOS-tips
- 60 File disk catalog
- 61 Diskettykning
- 65 Skatteprogram



Redaktion:

Rasmus Möller
Johan Nilsson

Adress:

Dykens väg 22
281 00 Hässleholm

Telefon:

0451-83200 efter 16.00

Bidrag:

125:-/artikelsida

120/år.

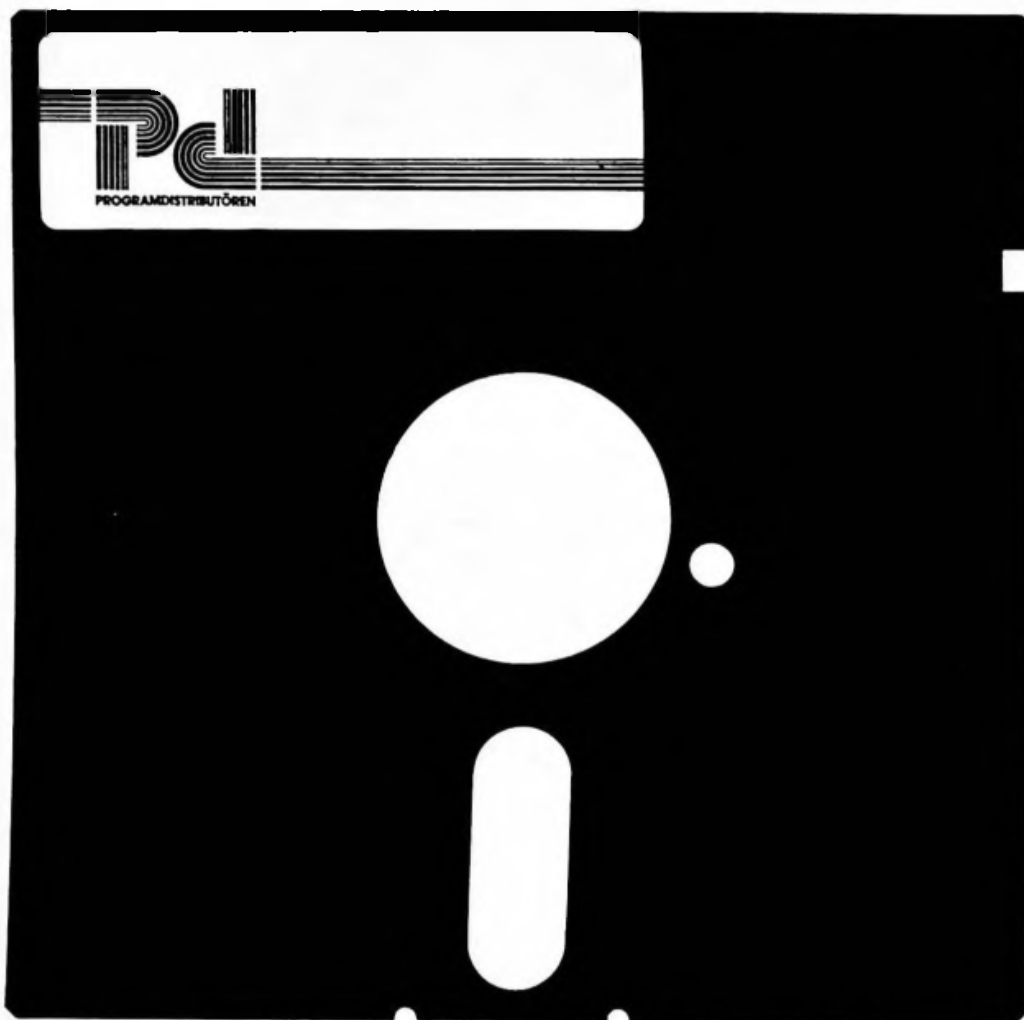
2,382 210

1,2383 15



PROGRAMDISTRIBUTÖREN Box 3009 580 03 LINKÖPING Tel. 013 - 12 12 40

'KATTEN'



Som första företag i Sverige lanserar vi nu 'KATTEN', katalog på diskett.

Där kan du läsa om alla program vi säljer till Apple II. Vi sänder dig 'KATTEN' och vår prislista om du sätter in 25:- på vårt Postgiro 7 05 28 - 5 och skriver namn och adress på blanketten. Du kan även få vår katalog kopierad på en av dina egna disketter. Sänd i så fall in en diskett till oss tillsammans med namn och adress. Vi returnerar omgående!

Störst antal artiklar någonsin - den etiketten sätter vi stolt på detta, det fjärde numret av Apple User News. Naturligtvis vill vi varmt tacka alla artikelförfattarna.

Detta nummer går ju under temat VisiSerien, och innehåller därför flera artiklar därom. Av samma anledning kommer många som köpt Visiprogram att få denna tidning i brevlådan. Vi hoppas att ni och andra kommer att finna tidningen intressant, så intressant att ni inte vill missa nästa nummer (mars 83).

Preliminära utgivningsdagar 83:

1: 9 mars	mtrl. 21 februari
2: 18 maj	mtrl. 2 maj
3: 7 sept.	mtrl. 22 augusti
4: 1 dec.	mtrl. 7 november

Programtävlingen 82

Tyvärr har inte så många bidrag kommit in att vi kan släppa lös juryn för att kora en vinnare. Vi har därför beslutat att förlänga tävlingstiden till den siste februari. Ställ upp - det kostar inget, fränsett tiden.

Medlemsvärvning

Vi inför fr.o.m. detta nummer en medlemsvärvartävling. Den som värvar flest medlemmar till utgången av mars kommer att belönas med ett pris. Ni får dessutom en belöning för varje medlem ni värvar (se talong i tidningen).

Medlemsskap

Många ringer eller skriver och frågar HUR man blir medlem. Vi trodde det framgick i nr.3, men här är det igen:

Medlemsskap är öppet för alla, privatpersoner, företag, institutioner m.fl. Du blir medlem genom att sätta i pengar (för 83 120kr) på föreningens bankgiro: 551-5648, och ange att du vill bli medlem.



Tidningen verkar ha utvecklats i samma snabba takt som utmärker hela datorindustrin. Ser vi tillbaka kan vi konstatera att för drygt ett år sedan var AUN ett fotokopierat ark, i februari i år bestämdes det att en riktig tidning skulle ges ut, och nu - ja du håller i den, tyck själv.

 apple computer

 apple computer

Nyheter

Vi vill be alla företag och privat personer att skriva till oss om Ni har någon nyhet, nya program, agenturer etc. Detta är en förutsättning för denna sida skall bli intressant.

Diversi-DOS

Diversified Software Research meddelar att man utkommit med en ny version på sin Diversi-DOS, vilken bland annat kan erhållas genom föreningen. Denna nya version heter 1-B, och i denna har vissa fel tagits bort. Det gäller främst textfilshanteringen som tidigare inte var felfri. Om Du redan har betalt för version 1-A så kostar denna uppdatering 5 USD, annars kan Du beställa DOSen för USD 30 från Diversified Software Research, Inc, 5848 Crampton Ct., Rockford, Ill 61111, USA.

Aktiedata

Aktiedata är ett program som hjälper till att administrera en aktieportfölj. Programmet åstadkommer en förteckning över innehavet, med uppgifter om nuvärde, anskaffningsvärde, värdeförändringar, branchfördelning m.m. Programmet kräver 48K, två diskettstationer samt skrivare med liggande A4, Epson MX100 eller liknande. Programmet har gjorts och marknadsförs av Dataprodukter i Bollnäs.

SOSCLOCK

SOSCLOCK /// är designad för att ge fullt SOS-kompatibla klock och kalender funktioner. Detta innebär att disketternas katalog (directory) automatiskt kommer att innehålla information om när filerna initierats etc.

SOSCLOCK /// betingar ett pris av USD 60 i USA.

Kan beställas från:
System Fabricators
736 Hermosa Avenue
Hermosa Beach, CA 90254, USA

Nytt CP/M-kort

Appli-Card är en intressant konkurrent till de vanliga Z80 korten som finns på marknaden. Kortet innehåller följande hårdvara:

Z80A eller Z80B processor m clock timer chip.

64K RAM plus från två till åtta K EPROM.

Parallell utgång och utgångs avkodning.

Expansions interface.

Kortet görs och marknadsförs av Lifeboat Associates Inc

Ny VersaForm

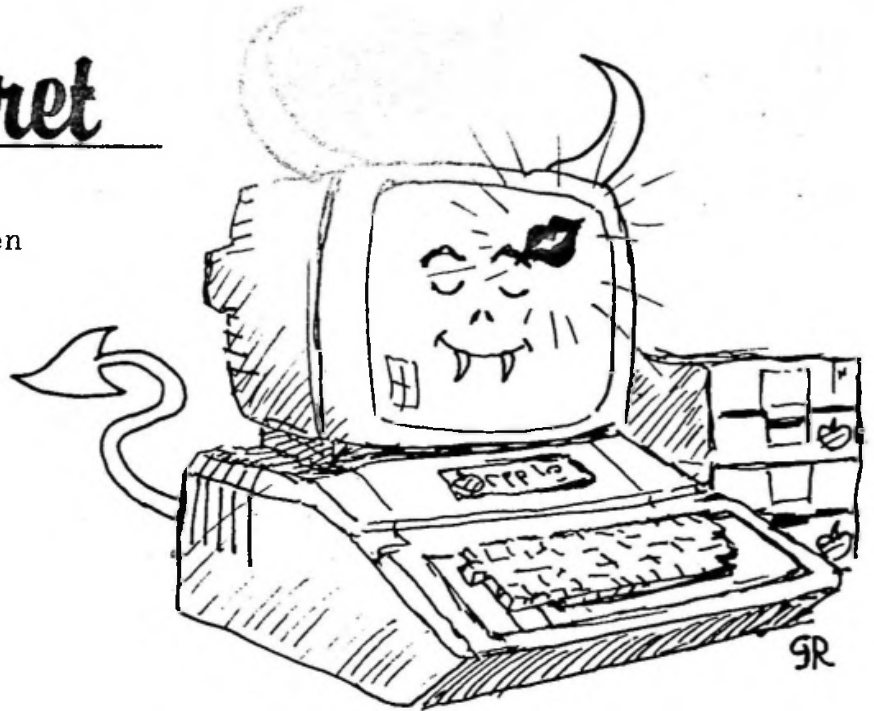
Det i USA populära databasprogrammet (e.g. Business Form Processor) har nu kommit ut i en version för Apple ///, som kan utnyttja både Profile och Corvus hårddisksystem. Apple /// versionen är i många avseenden identiskt med Apple II versionen - kommandostrukturer etc är samma.

Applied Software Technology
14125 Capri Drive
San Jose, CA 95030, USA

Flickan & Odjuret

En datanovis ser på Visiserien

Gunnar Redmalm



Människans möte med maskinen

Vi har väl alla hört den gamla sagan om flickan och odjuret, som berättas i olika versioner över hela världen. Flickan blir först skrämmd av det hotfulla odjuret, men hon kommer över sin rädsla och kysser monstret, som då förvandlas till en ståtlig prins. Sens moralen torde vara att många företeelser visar sitt rätta värde först när man möter dem med en öppen och positiv inställning. Tråkigt nog finns det en hake - alla odjur är inte förtrollade prinsar, och den som en gång kysst fel är föga benägen att göra om försöket.

För många oinvigda är datatekniken ett sådant där odjur som man inte riktigt vet var man har. Den som är van att ägna en mängd tid och intresse åt datorer och dataprogram har kanske svårt att föreställa sig den oro som kan sprida sig på en arbetsplats när någon i ansvarig ställning börjar muttra om datorisering. Att det "bara" är frågan om en mikrodator är ingen ursäkt - tvärt om. Stor- och minidatorerna omges åtminstone av ett skyddande hölje av experter. Men mikrodatorn är ett personligt verktyg som jag själv förväntas använda på ett intelligent sätt - ett verktyg som inte bara är en förlängning av mina manuella färdigheter utan av själva tankeförmågan. I mångas ögon "en otäck liten burk som bara gör allting krångligare eller - ännu värre - tänker bättre än jag".

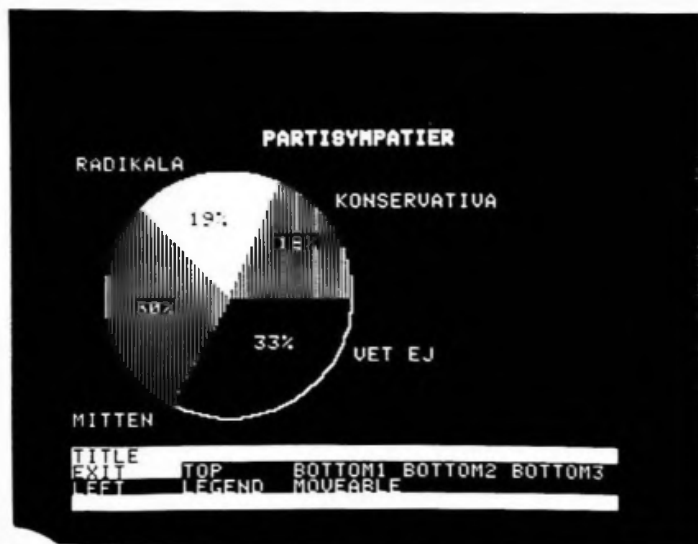
För den som övervinner motståndet och så att säga kysser till öppnar sig en värld av nya möjligheter - förutsatt att man har kysst rätt. Men med fel dator och fel program kan man i stället tycka att man fått sina farhågor bekräftade - datorn förblir det odjur man innerst inne alltid anat att den är. Och så sätter man sig och surar i ett hörn med papper, penna och möjligen en räknedosa, medan den förhatliga manicken står och samlar damm.

Själv hade jag turen att kyssa rätt redan från början, då jag kom att inleda min databana med en Apple II och Visiserien. Apple-datorn har faktiskt aldrig vållat mig några problem (om man bortser ifrån att mitt rymdskepp alltid flyger i bitar varje gång jag försöker lasta ombord jontorpeder), och Visiserien jämnade vägen in i applikationsprogrammets djungel.

Programmen i Visiserien

Visiserien från VisiCorp består för närvarande av sju program, som alla kan köras på Apple-datorer. En del av dem finns även i versioner för andra maskiner, framför allt VisiCalc, som väl är det mest kända.

VisiCalc är Visiseriens slagskepp, enligt uppgift världens mest sålda applikationsprogram. I VisiCalc visar bildskärmen en vidsträckt arbetsyta i matrisform, där man matar in kalkyler på ungefär samma sätt som om man skrev på ett vanligt anteckningspapper. Genom att bygga upp formler av rad-kolumn-koordinater kan man lätt skapa mycket omfattande kalkylmodeller, som omberäknas automatiskt varje gång ett värde ändras.



På så sätt kan man genast se hur varje detaljförändring påverkar kalkylen som helhet. När man väl har byggt upp en modell sparar man den på diskett och kan sedan använda den på nytt varje gång man behöver göra en kalkyl av samma slag. Vid grundarbetet erbjuder programmet en mängd genvägar, som automatisk kopiering av formler, smidiga redigeringsfunktioner och inbyggda standardberäkningar. Det är enkelt att ändra en modell genom att flytta, ta bort eller skjuta in rader och kolumner, eftersom alla formelkoordinater justeras automatiskt.

VisiCalc Advanced Version är en vidareutveckling av VisiCalc för Apple ///, med bl a större utformningsmöjligheter, tangentminne som låter användaren själv sammanföra en lång kommandosekvens till ett par tangentnedslag samt förmåga att kombinera ett flertal kalkylmodeller på önskat sätt. Detta program är ett bra exempel på hur mikrodatorerna börjar leta sig in på områden som hittills varit reserverade för större maskiner.

VisiSchedule bygger i mycket på samma grundtanke som VisiCalc - möjligheten att omedelbart se hur en enskild förändring påverkar helheten. Men medan VisiCalc är ett generellt kalkylhjälpmedel, är VisiSchedule ett specialprogram för tids- och resursplanering. Projektplaneraren matar in alla grunddata, som yrkeskategorier, lön och direkta kostnader, beräknad tidsåtgång för varje delaktivitet etc, som programmet sedan organiserar till en totalplan i form av ett överskådligt, kalenderanpassat diagram. Tidsenheter är antingen eller veckor, kostnader anges i ental och/eller tusental. Man kan kan omfördela, minska eller öka resurser, tid och kostnader som man vill och omedelbart se resultatet för hela planen. Sedan kan man spara planen på diskett och uppdatera den allt eftersom projektet framskrider.

Programmet genererar också utskrivna rapporter av fyra olika typer. Programmet användningsområde är förvånansvärt omfattande. Ett bifogat demonstrationsexempel visar planen för en vattenledningsinstallation - själv använder jag programmet till att planera bokproduktion.

VisiTrend/Plot är ett programpaket bestående av två självständiga moduler som samarbetar med varandra. VisiTrend utför statistiska beräkningar, och VisiPlot visar resultaten grafiskt på skärmen eller i utskrift. VisiPlot, som även kan anskaffas separat, har en mycket bred repertoar av diagramtyper och kan också ta emot data som matas in direkt eller från något annat program, t ex VisiCalc. Diagrammen visas i färg om utrustningen medger det.

VisiFile är ett registerhanteringssystem för lagring, sortering och hämtning av information. Programmet är mycket flexibelt och tillåter omfattande ändringar utan omprogrammering. VisiFile genererar även utskrifter av en mängd olika slag, som rapporter, listor och förteckningar, postetiketter etc.

VisiDex lagrar också information men står ett steg närmare användaren. Programmet kan enklast beskrivas som en elektronisk anteckningsbok, där man kan mata in uppgifter i vilken form man vill. Sökning utförs antingen ostrukturerat eller med hjälp av på förhand angivna nyckelord. Programmet associerar närbesläktade data med varandra på ett sätt som ger intryck av att ligga mycket nära mänskligt tänkande. VisiDex är även en "intelligent" kalender, som påminner användaren om viktiga händelser.

VisiTerm är ett program för datakommunikation. Med hjälp av detta program och ett modem kan man utnyttja sin Apple som terminal till en stordator eller kommunicera med andra mikrodatorer. VisiTerm innehåller en mängd funktioner som underlättar anpassningen till värddatorns krav.

Desktop/Plan, slutligen, är ett hjälpmedel vid affärsplanering, budgetering och finansiell analys. Programmet utför beräkningar, kombinerar uppgifter från olika håll till en totalkalkyl, skriver ut rapporter och skapar stapel- och linjediagram.

Varför Visi?

Var ligger styrkan hos Visiserien? Finns det några speciella egenskaper hos dessa program som skiljer dem från annan datakraft på marknaden?

Enligt min uppfattning har Visiserien framför allt tre starka sidor. Den mest uppenbara är möjligheten att överföra data direkt från det ena programmet till det andra via DIF-filer (Data Interchange Format). Hur detta går till beskriver Göran Bomark i en separat artikel.

Den andra, minst lika viktiga fördelen är att dessa program verkligen fungerar. De verkar ordentligt utprovade och är, så vitt jag kan bedöma, praktiskt taget "bugfria".

Men den egenskap som gjort mest intryck på mig är att programmen är så väl anpassade till användaren. Detta är delvis handböckernas förtjänst. Varje handbok består av tre delar, av vilka den första är en introduktion som ger allmän information om datorn och programmet. Den andra är en lärobok som i en rad lektioner förklarar programmet steg för steg. Den tredje delen är en uppslagsbok som man använder när man väl behärskar programmet i stora drag. Dessutom finns ett referenskort där man snabbt kan friska upp minnet när så behövs.

Handböckerna är mycket detaljerade och förutsätter egentligen inga så helst kunskaper i databehandling. Den som redan har viss erfarenhet av liknande program kan i allmänhet nöja sig med att skumma igenom lektionsdelen och slå upp svaren på frågorna allt eftersom de dyker upp.

En annan finess är systemet med "pekmenyer", där man väljer kommandon genom att peka ut dem med en markör. Att flytta markören känns snart lika hemvant som att borsta tänderna, och eftersom metoden väsentligen är densamma genom hela serien, känner man sig genast hemmastadd när man går över till ett nytt program.

Därmed inte sagt att Visiprogrammen skulle vara lämpade enbart för nybörjare. Dessa program fungerar på flera olika nivåer. Tack vare förvalda standardvärden kan man utan tidigare erfarenhet få ut ganska mycket av t ex VisiCalc redan efter ett par timmars självstudier. Men allt eftersom man tränger djupare in i programmet vidgas möjligheterna successivt, man vågar sig på allt mer avancerade funktioner, och innan man vet ordet av sysslar man med något som ligger mycket nära programmering.

Arbetet med ett Visiprogram är i hög grad en process som verkar i båda riktningar. Att se sina tankar åskådliggjorda på en bildskärm allt eftersom de tar form är både stimulerande och lärorikt. Man börjar med att använda programmet som ett hjälpmedel i den arbetsmetodik man är van att följa. Men så småningom leder programmet in tänkandet i nya banor, man rationaliserar sina metoder och finner snabbare och effektivare vägar som i sin tur leder till nya erfarenheter. Så är t ex VisiSchedule inte bara ett redskap som underlättar den planering man redan gör, utan en vägledare som efter hand lär användaren att planera på ett effektivare sätt.

Den verkliga styrkan hos mikrodatorn ligger inte i att den tänker åt oss, utan att den hjälper oss att tänka själva. Först när vi insett detta kan "odjuret" datorn undergå den stora förvandlingen. Visiserien är ett långt steg på vägen mot den insikten.

```
logon id 4309
password? sesam

user id 4309 logged in at 13:42:21 on 09/03/82

** Datorsystem A nedkopplat torsdag morgon
kl 7:30 till fredag em 3:00

print rapport aktiekurs

Hittierapport 23 oktober 1982
=====
Bolag H ..... 210 25 ned 5 10
Bolag B ..... 90 50 upp 7 50
Bolag F ..... 24 50 ned 8 00

COLF 0 HALF DUPL SET A SECHO
```



Specialist på

VISI SERIEN

VISIFILE™

VISIDEX™

VISICALC™

VISITERM™

VISISCHEDULE™

DESKTOP/PLAN™

VISITREND/PLOT™

datapunkten

Postadress: Box 7156, 103 88 STOCKHOLM
Besöksadress: Malmskillnadsgatan 38
Telefon 08-23 14 30

Windfall



The Apple computer users' magazine

Windfall blir bara bättre och bättre! Tidningen är nu mycket bra och är väldigt prisvärd. Vi får, som vi sagt tidigare, alla nya utgåvor av tidningen. Beställ genom inbetalning till vårt bg.551-5648.

17:-



April 1982
Apple speeds the news - Games review (Castle Wolfenstein, Threshold, President Elect) - DOS Toolkit problems - Linking Apples to IBM - Home-grown boards boom - Micro-Finesse review - Basketball match analysis - Elements of the Apple, Part III - FMS accounting system review - DOS disc directory, Part II - Apple graphics, Part IV (3D animation graphics) - Apple '82 Education Forum - A structured approach to teaching. PLUS four pages of CompuCopia and five Appletips.



May 1982
A case for Applebus as a new international standard - Games review - Flight Simulator - Hires Planet Plotting - Micro-speed review - Mathmagic review - Update on Printers (special 16-page printer section) - The Stationery Revolution - Understanding Microcomputers (Part IV) - Simulations Enhance Classroom Work - Computers in Business Education Studies - Speedy Way to Handle Histograms. PLUS four pages of CompuCopia and four Appletips.



June 1982
New ways of linking Apples to the outside world - Introduction to Forth, Part I - Games review (The Prisoner, Pinball) - Apples in Medicine - Tasc Compiler review - Micros in process control - Building pictures with machine code - High-speed Apple links to mainframes - Windport cards review - The Last One and CORP program generators reviewed - Book review (Apple II User's Guide) - Teacher's Toolkit and suite of primary school programs reviewed. PLUS four pages of CompuCopia and six Appletips.



July 1982
Games review (Pursuit of the Graf Spee) - Elements of the Apple, Part IV - Apple '82 reviewed - Introduction to Forth, Part II - Making the most of VisiCalc's capabilities - CBasic and MBasic analysed - Ormbeta database reviewed - Crossword Magic reviewed - Make your own user port, Part I - Earth Defence game and listing - Asynchronous data transfer, Part I - School application of Casil - Computers as an aid to concentration - PLUS four pages of CompuCopia and three Appletips.



August 1982
Games review (Bandits, Suicide, Swashbucker, Fly Wars) - Instruction file editor - Teach yourself Morse, Part I - VisiCalc section - Pstext II review - Asynchronous data transfer, Part II - Omnis review - A melody from your micro - Summary of 10 utilities - Make your own user port, Part II - Mah Jong - Number sorting - Elements of the Apple, Part V - Guidelines for buying a school Apple - Educational programs reviewed - PLUS four pages of CompuCopia and two Appletips.



September 1982
Use of CP/M COPY and PIP programs - Games review (Odyssey, Choplifter) - DOS aid to VisiCalc - The VisiCalc phenomenon - Wordscore game (listing) - Tasc compiler review - Med-res graphics, Part I - Snapshot review - Learning Morse, Part II - Button for multiple choice testing - Asynchronous data transfer, Part III - Bag of Tricks review - G-WHIZ review - Medic review - Sorting with Pascal - Memory test program (listing). PLUS four pages of CompuCopia and six Appletips.



October 1982
Games reviews Knight of Diamonds (the second wizardry scenario) and Pig Pen - Think Tank (with listings) - Med-res graphics, Part II (filling in shapes) - Lisa assembler language review - Magic of VisiCalc - VisiCalc Business Forecasting Model review - Cross reference listing program - Apple-vox speech synthesiser review - Morse Code, Part III - Computerised flash card for schools - French Verb program review. PLUS four pages of CompuCopia and seven Appletips.

Text file fast loader

from Call-A.P.P.L.E.
volym V, nr 9, R D Purves

WHILE writing a printer utility for a word processor, I encountered some interesting problems in loading text from disk. The outcome was a new DOS 3.3 command LTXT (Load TeXT) which combines the versatility of READ with the speed of BLOAD. LTXT is similar in conception to the FILEDUMP command described by Lee Reynolds in *Call A.P.P.L.E.* June 1982, but puts the text into memory instead of sending it to the current output device. It is four to five times faster than READ/INPUT for bulk loading of text files.

The text files, created by the word processor I was using, are ordinary sequential files with ASCII carriage return characters signifying the end of a paragraph. Initial attempts to access the files with READ and INPUT came adrift immediately because of commas and colons in the text and consequently a stream of EXTRA IGNORED error messages. It is possible to GET characters one-by-one from disk, but as anyone knows who has tried it, GETting more than a few hundred characters from a text file is a tediously slow business.

An "Input Anything" routine offered more promise. As most readers will know, this is a machine code routine which mimics the INPUT statement, but without rejecting commas and colons. Many problems in reading text files are overcome in a straightforward fashion. A well documented account is given by P. Meyer in *Call A.P.P.L.E. In Depth No.1* "All About Applesoft." Incidentally, Meyer's routine can be considerably shortened by letting the existing ROM routines do more of the work. My version, which fits into only 28 bytes, is shown in listing 1.

... neither IAS nor INPUT can cope with paragraphs longer than 255 characters ...

Unfortunately neither Input Anything nor INPUT itself could cope with paragraphs longer than 255 characters, which is the maximum string length permitted in Applesoft. Even if this limitation could have been sidestepped, tests showed that to input a file of 30000 characters would take more than two minutes.

Since the word processor can read its own files in 30 seconds, it was clear that a better method must exist.

At this point, reference to "Beneath Apple DOS" by D. Worth and P. Lechner showed that an appropriate call to the file manager portion of DOS would do the trick. For those who have not yet obtained this book, the file manager is a set of routines intermediate in level between the DOS command interpreter and RWTS (Read or Write a Track/Sector). Roughly speaking, the command interpreter knows how to figure out what you want and then tells the file manager to do it. The file manager in turn works out which disk sectors are involved and instructs RWTS accordingly. Finally RWTS takes charge of the disk hardware to read or write the sectors. As you proceed down the chain of command, the routines become more versatile but harder to provide with the right information.

Trials with a clumsy machine language driver program to call the file manager proved successful, and the last step was to make some changes to the command interpreter to create the previously mentioned new command LTXT.

As a personal opinion, I have reservations about altering DOS and other standard system programs. Such changes tend to propagate from disk to disk with sometimes very disconcerting results. (How many readers know that a disk INITED while the Program Line Editor is active has an altered DOS on it? A fresh boot from such a disk puts the DOS buffers and HIMEM \$600 bytes lower than normal, even if PLE is not subsequently loaded.) To avoid these surprises, LTXT disables the INIT command. Any attempt to INIT will give a SYNTAX ERROR.

Listing 2 shows how LTXT is activated. The DOS command table and keyword (parameter) table are changed, then the LTXT routine itself overwrites memory formerly used by the file manager's INIT code.

With sequential files, the syntax of the LTXT command is: LTXT filename,Aa,Ll[,Bb,ss,DD,Vv] where the A (address) parameter specifies an address in memory where loading is to occur and the L (length) parameter indicates the number of bytes to

The ... B parameter specifies how many bytes are to be skipped ...

be loaded. The optional B (byte) parameter specifies how many bytes are to be skipped in the file before loading commences. A default value of zero is assumed. S, D and V are the usual slot, drive and volume numbers and are also optional. If you use LTXT in a program, don't forget the preliminary Ctrl-D as a DOS "waker-upper".

Usually we will want to load the whole of a file without knowing what its length is. In this case it suffices to specify a length parameter greater than the file's actual length, since LTXT stops loading automatically at the end of the file. Thus you can set the L parameter to a huge value like 30000 even if the file is only a few bytes long. Prior to loading the text, LTXT puts zeros (\$00) into memory, from the starting address to (starting address + requested length + 1). Subsequently when you are processing the text in memory, the first occurrence of \$00 acts as a flag, to warn you that you have reached the end of the file. LTXT does not check that the A and L parameters point to a "safe" part of memory; *poor choices for these values can lead to DOS being overwritten*, with distressing consequences.

Once loaded, text information is accessible by machine language programs or by a succession of PEEKs. Note that characters retrieved from disk by LTXT have their m.s.b. set, so that A is CHR\$(193), not CHR\$(65). However, in many applications it is more convenient to create Applesoft string variables containing the text. To avoid losing the speed advantages of LTXT, a short machine language routine (listing 3) can be used to scan through the file after it has been loaded into memory, creating ordinary string variables with their m.s.b. off. A demonstration of this routine, named MAKSTR, is shown in listing 4. The string can be assigned to any string variable (simple or array). The original hex data is not altered or moved by MAKSTR, which makes its own copy of each character string.

MAKSTR is relocatable ...

Like Input Anything, MAKSTR is relocatable and can be placed anywhere you have 61 free bytes, merely by altering the value of MAKSTR in line 110.

When a string is constructed from hex data in memory, there can be three outcomes from a scan: a) a normal string is found, terminated by a carriage return character, b) a string is terminated by the end-of-file marker '\$00', or c) after 255 bytes have been incorporated into the string, the 256th byte is not a carriage return or \$00. This last case occurs frequently in word processor files and occasionally in files made from DOS WRITE and PRINT commands. On exit, MAKSTR leaves a return-flag value in location \$02. The three possible values of this flag, corresponding to a), b) and c) above are 13, 0 and 254. Thus the result of the string conversion can be tested by a pair of IF statements: IF PEEK(2) = 0 THEN you've reached the end-of-file. IF PEEK(2) = 254 THEN Applesoft's string variables are too small for the string characters in the file, and you'll have to treat two or more consecutive strings as one block of text.

Lastly, there is a question of random-access files. Since in theory the disk formats of sequential and random access files are the same,

LTXT should work equally well for either type. With the same syntax given earlier, LTXT loads L bytes from a random access file, starting from the beginning (record 0). Furthermore, you can skip over a given number of records before loading, if a preliminary OPEN command is given with a L parameter to tell DOS the record length. In this case an R (record) parameter is used with LTXT, just as with the READ command. LTXT automatically closes the file after loading.

LTXT worked satisfactorily on a number of random access files, but would sometimes stop short after loading only a fraction of the requested length. It turned out that

... DOS had another pitfall for me ...

DOS had another pitfall for me, in the form of "holes" in random access files. A hole is 256 consecutive bytes which a) have never been written to and b) correspond exactly with a disk sector. If you have a record length of 100 and you write a single character to records 0 and 10, then the 999 byte gap between these two characters includes two holes (bytes 256-511 and 512-767). DOS refuses to allocate disk space to holes until you write

some data to the relevant part of the file. The DOS manual alludes somewhat obliquely to this quirk on page 129. When LTXT encounters a hole in a file, it stops loading, under the impression that end-of-file has been reached.

Although it is possible to fill in the holes in a pre-existing random access file, care is needed to preserve data. It is easier to avoid holes right from the outset by writing a single return character to each and every sector that the file may ultimately use. If the record length is RL and you plan a total of NS sectors, then the following code will prevent holes:

```

10 PRINT D$"OPEN FILENAME"
20 FOR BYTE = 0 TO RL*NS
   STEP 256
30 PRINT D$"WRITE FILENAME,
   B"BYTE
40 PRINT
50 NEXT BYTE
60 PRINT D$"CLOSE"

```

A useful side-effect is that since all the file's sectors are now allocated, you cannot get a DISK FULL error while writing to a record in the middle of the file.

forts sid66

LIST

```

100 REM          LTXT
          BY R.D. PURVES
110 T$ = "A884:4C 54 58": REM
   CHANGE INIT TO LTXT
120 T$ = T$ + " NA909:20 7F": REM
   ALLOW PARMS A,L,B,R,S,D,V;
   DISALLOW NEW-FILE CREATION
130 T$ = T$ + " N9D1E:8D AE": REM
   ADDRESS-1 OF PATCH
140 GOSUB 500
150 T$ = "AE8E:20 26 A5 A2 03 8E
   BB B5 CA 8E BC B5 AD 6C AA 8
   D C1 B5 38 6D 72 AA 85 50 AD
   6D AA 8D C2 B5 6D 73 AA 85
   51 AD 72 AA 85 85 AD 73 AA 8
   5 86 A9 00 A8 91 85 E6 85 D0
   02 E6 86 A6 85 E4 50 D0 F2
   A6 86 E4 51 D0 EC 4C 85 A3"
160 GOSUB 500: PRINT : PRINT "LT
   XT IS LOADED AND WILL REMAIN
   PART": PRINT "OF DOS UNTIL
   YOU REBOOT"
170 END
500 T$ = T$ + " ND9C6G": FOR J =
   1 TO LEN (T$): POKE 511 + J
   , ASC ( MID$ ( T$, J, 1)) + 128
   : NEXT : POKE 72,0: CALL -
144: RETURN

```

LIST

```

100 REM          MAKSTR
          BY R.D. PURVES
110 MAKSTR = 768: FOR J = MAKSTR TO
   MAKSTR + 60: READ K: POKE J,
   K: NEXT
120 DATA 32,190,222,32,227,223,3
   2,108,221,133,133,132,134,16
   0,0,177,0,41,127,153,0,2,170
   ,240,15,201,13,240,11,200,20
   8,239,162,0,142,255,2,169,25
   4,168,133,2,200,230,0,208,2,
   230,1,136,208,247,152,160,2,
   32,233,227,76,154,218
130 INPUT "GIVE DECIMAL ADDRESS
   (TRY 53456):";ADDR
140 POKE 1,ADDR / 256: POKE 0,AD
   DR - 256 * PEEK (1)
150 :
160 PRINT : PRINT : PRINT "START
   INC ADDR = " PEEK (0) + 256 *
   PEEK (1)
170 PRINT
180 CALL MAKSTR,A$: PRINT A$
190 PRINT
200 PRINT "LENGTH = " LEN (A$) TAB(
   20)"RFLAG = " PEEK (2)
210 PRINT
220 PRINT "GIVE DECIMAL ADDRESS"
   : INPUT "OR PRESS <RETURN>
   TO GO ON:";ADDR$
230 ON ADDR$ = "" GOTO 160:ADDR =
   VAL (ADDR$): GOTO 140

```

LÅT DIN APPLE VÄXA

RAMKORT SPECIAL:
16K-RAM kort 1150:-
64K-RAM kort 3100:-
32K-RAM kort 1550:-
128K-RAM kort 4700:-

PROGRAM TILL RAMKORT:
Memory Management System 250:-
Versa RAM Disk 800:-
Versa Visicalc Expander 800:-

Vårt 80 kol. kort, U-TERM, har nu Svenska tecken (ÅÄÖ), små/stora bokstäver och 40/80 switch. Det fungerar med PASCAL och CP/M och Du kan göra egna teckenset utan att bränna E-PROM. PRIS: 2200:-.

... och vi har inte glömt APPLE III.

Seriekort	2300:—	
BCD-interface	2300:—	
8 st serieinterface (U-PORT)		4000:—
2 st serieinterface (U-2PORT)		2600:—
A/D-omvandlarsystem (12 bitars)		6900:—
Digitalt I/O-kort med timer (ingår i A/D-omv systemet)		2500:—

Dessa kort levereras komplett med driver-rutiner för APPLE III:s operativsystem, SOS.

Generalagent i Sverige

DATA HJÄLPEN

Kungshöjdsgratan 9, 411 20 Göteborg
031-136814

KÖP KORTEN HOS DIN APPLEFÖRSÄLJARE.

Korten är tillverkade i England av U-MICROCOMPUTERS LTD. Alla priser exkl. moms. Reservation för prisändringar.

BENEATH APPLE DOS // BAG OF TRICKS

Beneath Apple DOS är en bok som mycket ingående beskriver Apple II's diskoperativsystem. Den kom ut 1981, och den innehåller mycket information som aldrig förut publicerats i bokform. För den som tycker om att rota i Apple-systemet är boken en guldgruva, men även för en vanlig användare innehåller den mycket intressant information.

Boken består av 8 kapitel och 3 appendix. Kap. 1 är en kort introduktion. Kap. 2 är en historik över Apple DOS, en beskrivning av alla olika versioner som Apple släppt, samt hur de skiljer sig från varandra. Det kom en eller flera versioner om året, till den 25 aug 1980, då DOS 3.3 släpptes ut. Det är nu över två år sedan, och man kan nog gissa att det inte kommer någon ny version efter DOS 3.3.

Kap. 3 beskriver hur data lagras på disketten på den allra lägsta bit-nivån. Vissa delar av denna beskrivning är felaktig, för en rättelse se Byte, juni 1982, sid 402. Dessa felaktigheter har dock mycket liten betydelse för förståelsen av det hela. Clyckligare är då ett fel på sidan 3-23, som beskriver hur sektorerna "bladas in i" varandra på 16-sektors-disketter. Uppe till vänster står "Physical Sector", ska vara "Logical Sector". I spalterna till höger om denna står nämligen de fysiska sektorer som motsvarar dessa logiska sektorer under de olika operativsystemen DOS, Pascal och CP/M på Apple II. Detta är dock endast detaljer i en i övrigt mycket utförlig och pedagogisk beskrivning hur data lagras fysiskt på disketten, och på vad sätt en 13-sektors och 16-sektors diskett skiljer sig från varandra.

När man nu har fått veta hur enskilda sektorer lagras på disketten, fortsätter kap. 4 med att beskriva hur alla dessa sektorer logiskt knyts ihop till en fungerande diskett, var katalogen finns osv. Mycket av detta står redan i Apples egen DOS 3.3 Manual, men det beskrivs här klarare och fylligare, och dessutom har en del fel i Apples egen manual rättats till. Kapitlet avslutas med en kort diskussion av de vanligaste orsakerna till diskettfel.

Nästa kapitel, nr 5, beskriver i korthet strukturen hos själva DOS, var i minnet de olika delarna ligger osv. En fullständig beskrivning av alla vektorer på sid 3 i RAM-minnet ges också, samt en kort diskussion om vad som händer när man bootar sin disk.

Kap. 6 handlar om hur man använder olika delar av DOS från assemblerspråk. För att ha utbyte av detta kapitel bör man förstås ha viss vana vid assemblerprogrammering. Kapitlet inleds med en beskrivning av hur man direkt använder disk-driven från assembler, helt utan DOS. Sedan beskrivs hur man använder RWTS. Detta beskrivs också i Apple-manualen, men här är beskrivningen fylligare och klarare. Kapitlet avslutas med en beskrivning av en del av DOS som Apple aldrig lämnat ut någon dokumentation på: File Manager. Den har sin egen parameterlista och t.o.m. två egna vektorer på sid 3 i RAM. För den som kan lite CP/M, kan man göra liknelsen att medan RWTS motsvarar BICS, så motsvarar File Manager BDOS. RWTS hanterar endast spår och sektorer. File manager däremot hanterar namngivna filer. Flera program från Apple gör direkta anrop av File Manager, t.ex. FID, RLCAD och DCS Toolkit-assemblern. Med File Manager kan du hantera alla 8 filtyper som finns definierade i DCS helt utan restriktioner. "Avslöjandet" av File Manager är enligt min mening ett av de mest värdefulla avsnitten i denna bok. Allra sist i detta kapitel finns en lista på några korta assembler-rutiner som man kan ha nytta av när man jobbar med DCS.

Kap. 7, som är ganska kort, beskriver några bra knep med DCS. T.ex. BRUN eller EXEC av HELLO-programmet, ta bort pausen vid en mycket lång katalog.

Kap. 8 är det kapitel som innehåller mest information av alla. Det är en mycket ingående, nästan byte-för-byte beskrivning av hur DOS 3.3 fungerar. Om man ska ha fullt utbyte av detta kapitel, bör man disassemblera hela DCS 3.3, och sedan noga läsa igenom detta kapitel, jämföra med disassembleringen, samt ofta göra egna anteckningar. Annars blir kapitlet lätt överskådligt.

Boken avslutas med tre appendix. Appendix A innehåller fem korta utility-program, komplett med assemblerlista. Programmen är inte "färdigputsade", utan tjänar mera som praktiska exempel på hur man använder informationen i Beneath Apple DOS. Programmen är:

DUMP - gör en rå minnesdumpning av ett spår på en diskett.

ZAP - läser/skriver en godtycklig sektor via RWTS.

INIT - formatterar om ett godtyckligt spår på en diskett.

FIS - letar efter track/sector-listor på en diskett.

CCPY - kopierar över en binärfil till en textfil via File Manager.

Nästa appendix är en diskussion om olika metoder att kopieringsskydda disketter. Diskussionen är så allmänt hållen att den inte är till någon större nytta för eventuella "disk-knäckare". Boken avslutas med en ordlista och ett sakregister.

Beneath Apple DOS blev snart en succé, så författarna gav 1982 ut en andra bok - Bag of Tricks. Detta är en bok + en diskett. Disketten innehåller fyra olika program, som alla är vidareutvecklingar av de korta program som fanns med i Beneath Apple DOS. Några av programmen är smått fantastiska. Eller vad sägs om följande:

INIT - formatterar om ett eller flera spår på en diskett, och sparar innehållet hos de sektorer som finns på spåret. Diskettens volymnummer kan ändras, sektorernas hop-bladning (interleaving) kan också ändras.

TRAX - gör en rå hex-dump av ett spår samt analyserar spåret. Programmet är inte avsett för analys av kopieringsskyddade disketter utan snarare för analys av disketter där en eller flera sektorer förstörts. Man kan m.a.o. inte med detta program analysera själva Bag of Tricks -disketten (den är f.ö. skyddad på ett sätt som inget i dag existerande nibble-copy program klarar av. Om disketten inte skulle fungera, kan man mot en mindre kostnad få den utbytt. Man kan t.o.m. köpa ett backup-exemplar innan originalet ev. blivit trasigt).

FIXCAT - reparerar en skadad katalog, tar fram filer som blivit DELETE-ade eller på annat sätt förlorade. FIXCAT gissar även vilken filtyp det är, och gissar förvånansvärt ofta rätt!!

ZAP - det mest omfattande av de fyra programmen. Detta är en diskett-editor som i ett och samma program klarar av att editera 13- och 16-sektors-disketter av alla förekommande typer, dvs DOS, Pascal och CP/M. Man kan ange track/sector som man vill editera. Men man kan också "öppna" en fil. Man anger då ett filnamn, och handskas sedan med sektorer in i filen. Även detta kan göras i såväl DOS, Pascal och CP/M.

Som om inte detta räckte, finns det även möjlighet att programmera ZAP, genom att definiera makron, som kan bestå av godtyckliga kommandon, inkl. andra makron.

Det finns också ett LCOP-kommando, med vars hjälp man kan köra en slinga, dvs upprepa ett eller flera kommandon önskat antal gånger. Dokumentationen på ZAP omfattar hela 47 sidor.

De fyra programmen beskrivs utförligt i var sitt kapitel i boken. Sedan följer ett kapitel som ger exempel på hur man kan använda dessa program. Några exempel: Ändra i random-textfiler i DOS, frigöra oanvänt utrymme i filerna, leta efter dåliga sektorer på en diskett och reparera dessa, kopiera en Pascal-fil till DOS, jämföra två filer, göra ett DOS-löst boot-program.

Apple Doc
SouthWest Data Systems
435.-
044127161

Sensible Software
AppleSoft Optimiser
BEST 450.-

Din dator heter **APPLE!**

 **apple**
Computer



Missa inte
SVERIGES MEST PRISVÄRDA
klientbokförings-system!

Be oss visa Dig **APPLE REVISORSPAKET** för Apple II eller Apple III. Du kommer att finna att världens mest sålda dator har det mest prisvärda bokföringsprogrammet!

Nu med ännu fler finesser.

- Du **SPAR TID** i registreringen tack vare **MOMSAUTOMATIK** och **AUTOMATISK MOTBOKNING**. Slå en tangent: datorn räknar fram in- eller utg moms. Slå en annan tangent: datorn slår själv in motsatt belopp.
- Du ser hela tiden saldo för **KASSA, POSTGIRO** eller **CHECK**. Felsignal ges givetvis vid differens på ett verifikat, men om Du så önskar kan Du boka differens på enskilda verifikat. Du ser hela tiden den totala differensen.
- Du kan **AUTOMATISKT BOKFÖRA** tex sociala kostnader, varuåtgång m m.
- Du kan använda **VALFRI KONTOPLAN** med valfritt utseende på rapporterna.
- Du kan få **KOSTNADSSTÄLLE- / KOSTNADSBÄRARREDOVISNING**.
- Du kan lagra alla belopp på ett urval av **KONTON**, och Du kan i slutet av året ta fram sk **ÅRSHUVUDBOK**.
- Du får i rapporten **JÄMFÖRELSE** med både budget och föregående år.
- Du **AVLÄSER PÅ BILDSKÄRMEN** valfritt kontos saldon och transaktioner, även före rapportutskrift.

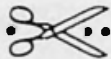
- Du kan ta listor på dataskärm eller på skrivare.
- Du spar massor av tid genom att datorn — på t ex lunch/kafferast — gör **SAMLAD UTSKRIFT** av alla rapporter. De vanligaste rapporterna är Dagbok, Huvudbok, Resultaträkning, Balansräkning och Momsrapport. Övriga rapporter är Bokslutstabla, Saldobalans, Saldofördelning, Nyckeltalsrapporter, Rapportkoncentrat m m.
- Du kan t o m ta fram **DEKLARATIONSBILAGOR DIREKT** från datarapporterna!
- Du använder endast **EN DISKETT PER KUND**.
- Mycket viktigt: Apple ger Dig perfekt **ERGONOMI** tack vare ett separat tiotangentbord.

Du **HÄPNAR** när Du hör priset: Ett komplett **APPLE REVISORSPAKET** med skrivare och program kan Du få för ca 31.000 kronor exkl moms! Du kan leasa för under en tusenlapp i månaden! **SERVICEAVTAL** för 200 kronor i månaden!

NU KAN DU ÄVEN PROVHYRA EN APPLE i tre månader, utan att Du binder Dig att köpa!

Är Du inte nöjd ändå kan Du som tillägg välja världens mest rikhaltiga programutbud för bl a: Tidsredovisning, Skatteberäkning, Utskrift av årsredovisningar, Skatteplanering, Deklarationsutskrifter, Registerhantering, Ordbehandling, Löner, Reskontra samt Visicalc — världens mest sålda dataprogram!

KOM ÖVERENS MED OSS OM LÄMPLIG TIDUNKT FÖR DEMONSTRATION!



Skicka mig en närmare beskrivning av **APPLE REVISORSPAKET**.

Namn:

Telefon:

Adress:

Postadress:

Datarationalisering AB

Fjärdhundragatan 50, 199 31 Enköping • Telefon 0171 - 386 90

NYA AVANCERADE VISICALC (R)

"Vad VisiCalc gör för den enskilda planläggaren utför programmets avancerade version för hela företaget." Så presenterar VisiCorp sin nya skapelse. VisiCorp är programföretaget som skeppat ut mer än 300 000 VisiCalc i världen.

Ny marknad

VisiCalc har nu funnits i dryga tre år och har sålt mer än något annat program. Ide'en har varit: en användare, en dator och en VisiCalc-modell.

På större företag önskar man ofta sammanställa individuella planer till en total plan. Vidare har man i större företag (och även i andra sammanhang) ett behov att låta personer utan större datorvana sammanställa och lämna uppgifter, på ett säkert och effektivt sätt.

Nya avancerade versionen av VisiCalc togs fram med bla dessa krav. Det har säkert varit svårt för VisiCorp att ta med "lagom" många nya funktioner. Deras princip sedan tidigare har varit att ett kalkyl-program ska vara ett kalkyl-program. Därför har VisiCorp tagit fram de övriga programmen i Visi-serien. Där är varje program specialist inom sitt område. Programmen byter sedan data med varandra via det gemensamma dataformatet DIF.

VisiCalc Advanced Version tycks följa den traditionen. Det är ett supberpt kalkyleringsprogram!

"Elektronisk blankett"

De nya funktionerna möjliggör att en person bygger modeller, som sedan används av andra. Det går att skapa "elektroniska" blanketter för budgetering, prognoser, planering av olika slag, tekniska undersökningar, laboratorieförsök mm. Genom att låsa celler för inskrift och tabuleringsmöjligheter, kan modellbyggaren helt styra den kommande inmatningen. Den som fyller i

blanketten behöver inte veta ett dugg om hur modeller är byggd. Vidare finns en speciell hjälpfunktion, som man kan anropa när som helst. Den hjälptext som då dyker upp gäller för det man för tillfället håller på med.

Tangent-minnet innehåller flera, av modellbyggaren bestämda, sekvenser av VisiCalc-kommandon. Den som fyller i en färdig gjord modell kan med en knapptryckning aktivera detta tangent-minne. VisiCalc kommer då att utföra de aktuella kommandon som finns fördefinierade. Lämpligt bli då den "elektroniska blanketten" ska sparas på skivminne.

Detta var lite om VisiCorp's ide'er kring det nya programmet. Låt oss se lite mer i detalj vilka nya funktioner som finns i avancerade VisiCalc.

----- Format-kommandot

Detta kommando används för att sköta utformingen av celler. (Det gamla utformingskommandot FORMAT, görs internt om till detta nya kommando). Kommandot finns även under global-kommandot, och påverkar då alla celler som inte har något enskilt attribut.
Bl a kan man :

- Låsa en cell, så att den helt skyddas mot ändringar. (Mycket värdefullt i typ "elektroniska blanketter".)
- Öppna en cell så att den nås med tabulatorn. (Bra i en inmatningssituation.)
- Gömma cellinnehåll, så att innehållet inte syns på skärmen eller i pappersutskriften.
- Kontrollera datatyp. Olika varianter finns: endast siffror, endast tecken, siffror och tecken men inte formler. (Passar bra då någon ovan person ska jobba med en modell.)
- Få %-tecken i numeriska fält samtidigt som talet multipliceras med 100.
- Debet- och kreditsymboler i numeriska fält.
- Marginaler, både till höger och vänster, inom kolumn. (Användbart för att göra " snygga " modeller).

De flesta av attributen går att kombinera.

----- Olika kolumnbredd

Egentligen räcker det med överskriften. De flesta som jobbat med standard VisiCalc har säkert någon gång önskat att kolumnerna skulle kunna ha haft olika bredd. Nya VisiCalc erbjuder det. Från 1 tecken ända till 125 tecken. Så liten eller stor kan varje kolumn individuellt göras.

----- Upprepnings-kommandot (REPLICATE)

Två nyheter finns här. Det första är att man kan upprepa (kopiera) hela block en eller flera gånger. Detta är tidsbesparande. Det andra är att kommandot nu fått ett underkommando som bl a annat möjliggör:

- upprepning av attribut. Man kan i efterhand utforma celler utan att innehållet går förlorat.
- upprepning av enbart siffervärden utan att de ev. underliggande formlerna följer med. (I standard versionen klarar man det genom att mellanlagra data i DIF-format på diskett.)
- relativ upprepning genom hela området. Är man säker på att alla formler och uttryck kommer att bli relativa i sitt nya område kan man direkt säga till om det innan kommandot utförs. Tidsbesparande.
- icke-relativ upprepning på samma sätt som ovan.

----- Utskrifts-kommandot

Det går att få presentationsfärdiga rapporter. Man kan sätta värden för bl a:

- Sidnumrering
- Överskrifter på varje sida
- Antal rader som ska utnyttjas på varje sida
- Hur många rader långt pappret är (perforeringsöverhopp)

- Antal tecken per rad
- Extra radframmatning
- Speciella styrtecken till skrivaren

Dessa värden sparas tillsammans med arbetsytan på skivminne. Därför behöver man inte, till skillnad från standard VisiCalc, skriva in styrtecken till skrivaren för varje utskrift. Praktiskt.

Nya "alfaslang"-funktioner

Det har tillkommit en del nya ekonomifunktioner för främst ränteberäkning fram och baklänges: nuvärde, framtida värde, annuitet, antal perioder, interränta mm. Det finns också ett antal "tid"-funktioner: antal dagar mellan två datum (klarar av skottår) och liknande.

Tangent-minnet (KEYSTROKE-MEMORY)

Det är väl här som VisiCalc Advanced Version är som mest innovativt.

Tangent-minnesfunktionen kan ses som en makrodefinition. Med det menas att en hel sekvens av VisiCalc-kommandon kan initieras med en knapptryckning. Enklast definieras en sådan sekvens genom att funktionen aktiveras med CTRL-K. Man ger sedan sekvensen ett namn i form av en bokstav (A-Z). Allt man sedan skriver kommer VisiCalc ihåg. Genom att, när definieringen är klar, slå CTRL-K och ange namnet på sekvensen (en bokstav) får man VisiCalc att utföra exakt samma kommandon som man tidigare skrivit. Alla VisiCalc kommandon inklusive markörförflyttningar och Hopp-kommando kan användas, SAMT en till. Med tangent-minnesfunktionen får man ett kommando till: "PAUS". Skriver ^P i tangentminnes sekvens. Därefter skriver man en text som sedan, då tangentminnet aktiveras, kommer att skrivas ut upptill på skärmen, på inmatningsraden. Här har modellbyggaren möjlighet att skriva små meddelanden till den som senare jobbar med modellen. Text "Stoppa in datadiskett i skivminne 2", "Tryck på TAB och fyll i dina värden". Ett sådant meddelande kvitteras med RETURN och sekvensen fortsätter. Varje sekvens kan vara 123 tecken. Observera att även tangent-minnesfunktionen kan ingå i en sekvens. Det går därför att länka sig vidare från en sekvens till en annan. Praktiskt sätt innebär detta att man kan ha flera tusen tecken långa sekvenser.

Tangent-minnet och DIF

Ett bra användningsområde för tangent-minnet är då flera modeller ska sammanställas i en större modell. Genom att "förprogrammera" inläsning och lagring av data (i DIF format) och modeller, kan man enkelt sammanställa flera arbetsytor till en kalkyl. Genom att man använder DIF format kan data hämtas var som helst från arbetsytan, lite här och lite där. När man sedan summerar till en annan modell behöver denna inte alls se ut som de modeller man hämtat data från. Med DIF data bestämmer man själv var på arbetsytan data ska läsas in. Det är förresten inte bara summering som kan användas, det går lika bra att använda vilken operation som helst.

Passar även för en person

Såsom det beskrivits ovan kan det tyckas att nya VisiCalc enbart passar då det finns en som gör modeller och andra som använder dem. Naturligtvis passar programmet även inom samma områden som standard VisiCalc gör. Den VisiCalc-användare som vill göra mer avancerade modeller kommer att uppskatta nya VisiCalc.

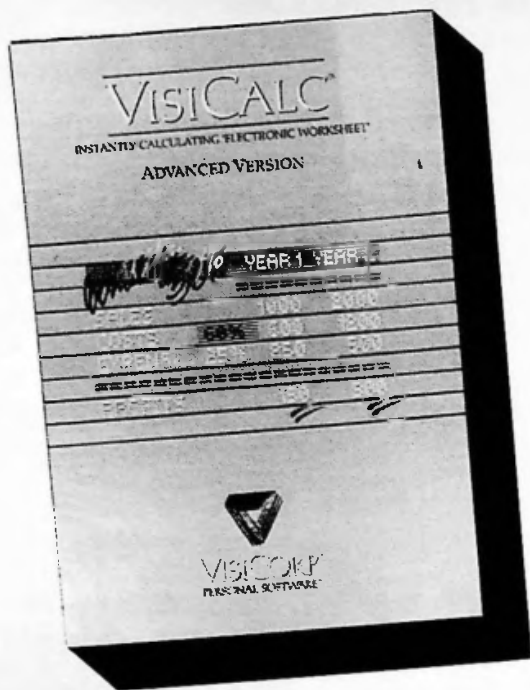
Dator

Under hösten och vintern kommer programmet att finnas enbart till Apple /// datorn. (Programmet kommer aldrig att kunna användas på en vanlig Apple II dator, den är för liten.) Programmet går att använda tillsammans med Apple's Profile skivminne på 5 Mbyte. På en 256k Apple /// får man en arbetsyta på ca 185k beroende på hur många drivrutiner som används.

forts från sid 18

Bag of Tricks innehåller - tvärtemot vad all reklam påstår - inte speciellt mycket information om DOS som inte redan finns i Beneath Apple DOS. Men å andra sidan är de fyra utility-programmen betydligt bättre än vad som annars brukar finnas i den vägen, och de är väl värda sitt pris. Man kanske skulle kunna sammanfatta det hela med att den som vill veta "allt" om DOS bör skaffa Beneath Apple DOS, och den som vill ha i det närmaste full kontroll över sina disketter bör skaffa Bag of Tricks. I Sverige säljs bägge dessa av Programdistributören, den som vill försöka komma undan lite billigare kan förstås skicka efter dem direkt från Quality Software i USA. Adressen kan man hitta i annonser i amerikanska datortidskrifter, kanske då främst de som mer eller mindre inriktar sig på Apple, dvs Micro, Nibble, Windfall, Softside, Apple Orchard, Apple Assembly Line

VISICALC ADVANCED VERSION



**INGENTING
ÄR SÅ BRA
ATT DET
INTE KAN
GÖRAS
BÄTTRE**

VisiCalc är kalkylprogrammet som redan har revolutionerat tänkandet inom mikrodatorbranschen. I dag spar VisiCalc tid och pengar åt mer än 300 000 mikrodatorägare världen över.

Men bra är inte detsamma som perfekt. Det finns faktiskt saker som inte ens VisiCalc klarar av. Därför har VisiCorp utvecklat VisiCalc Advanced Version för Apple ///.

Vad VisiCalc gör för den enskilde planläggaren gör VisiCalc Advanced Version för ett helt företag. Programmet är speciellt utformat för att bearbeta och kombinera ett flertal komplexa beräkningsmodeller. Och samtidigt som kapaciteten mångdubblats är den nya versionen lika smidig och lättanterlig som den gamla.

Några exempel på finesserna hos VisiCalc Advanced Version:

Tredimensionell kapacitet: Ett flertal kalkyluppställningar kan slås samman eller kombineras på önskat sätt.

Utökade funktioner: Större utformningsmöjligheter, t ex olika kolumnbredd, valfritt antal decimaler, flytande π - och %-tecken i celler med värden. Hela sektioner av arbetsytan kopieras med ett enda kommando. Kalender- och ekonomiska funktioner. Tangentminne låter dig sammanföra upp till

123 valfria kommandon till ett enda. Större redigerings- och utskriftsmöjligheter.

Lätt att använda: Förklarande hjälptexter kan i varje läge kallas fram på bildskärmen.



VISICORP™
PERSONAL SOFTWARE™

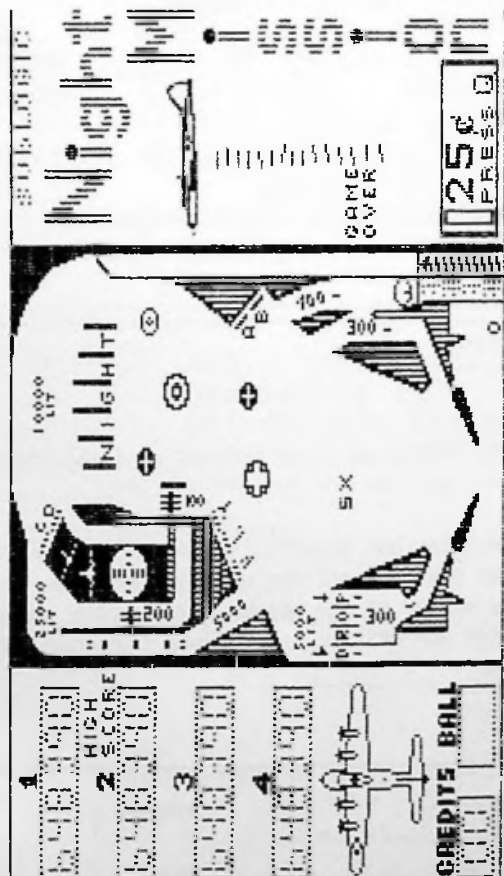
VisiCalc är dagens programvara när den är som bäst. VisiCalc Advanced Version är morgondagens - redan i dag! Be någon av våra återförsäljare om en demonstration.



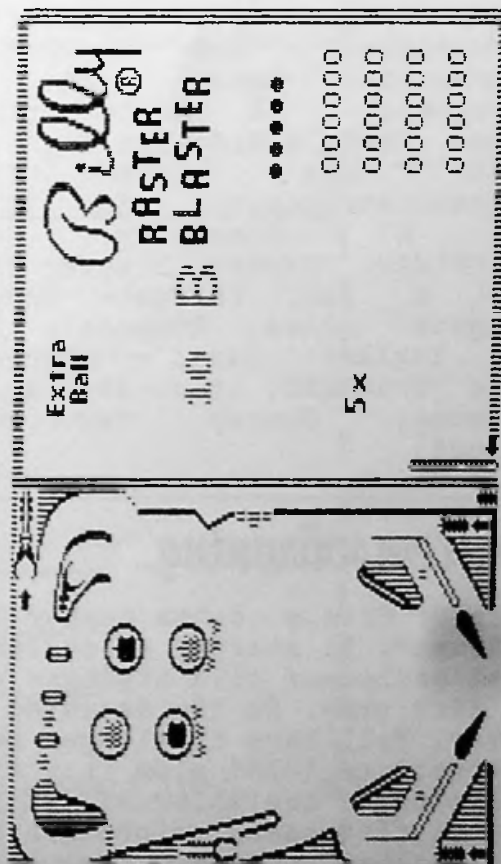
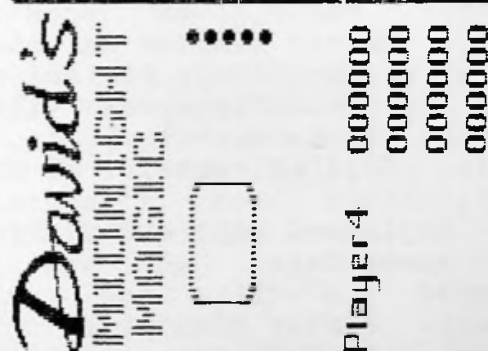
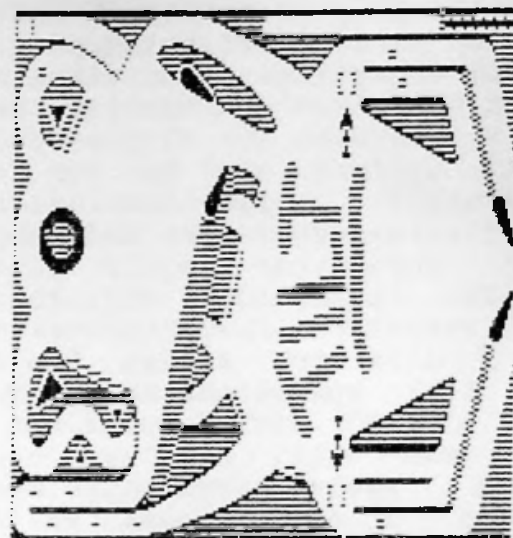
Software of Scandinavia
RIDDERHEIM WÆRNÉR AB

Tel. 08-717 02 70

Minns du också nostalgiskt tiden då du stod vid de tjutande flipperspelen frenetiskt kämpande för att hålla den sista kulan i spel. De gjorde iallafall jag och samlade därför ihop de tre på marknaden förekommande flipperspelen (i Sverige ett): A2-PB1 Night Mission, David's Midnight Magic samt Raster Blaster. I alla tre spelen får man upp en spelplan. A2-PB1 och Raster Blaster visar normalt proportionella spelplaner medan David's Midnight Magic är bredare än normalt. David's Midnight Magic har dessutom fyra flipprar mot de andras mer



normala två. Alla tre spelen simulerar på ett mycket bra sätt bollen rörelse; man kan hålla bollen med flippern uppe, dock studsar bollen hela tiden i Raster Blaster och David's Midnight Magic vilket den inte



gör i A2-PB1 där den ligger absolut stilla. I alla spelen kan man vara upp till fyra spelare samtidigt. A2-PB1 och Raster Blaster har dessutom olika svårighetsgrader. I Raster Blaster två svårighetsgrader, i A2-PB1 egentligen hur många som helst. Här kan man nämligen helt själv bestämma hur flipperspelet skall uppföra sig! Man kan från programmet hoppa in i en modifieringsmeny där man anger alla parametrar själv vilket innebär fantastiska möjligheter till variation.

I alla tre spelen kan man slå till spelplanen eller luta den allt för att få kulan att gå dit man vill. Man kan också frysa spelet och fortsätta senare i A2-PB1 samt David's Midnight Magic, en praktisk detalj. Raster blaster har några mindre buggar i sig. Ibland går kulan igenom flipprarna vilket kan vara störande.

Jag föll direkt för A2-PB1s detaljrikedom och framförallt dess möjlighet till modifiering. Det sammantaget gör att jag utnämner A2-PB1 som klar vinnare. Raster Blaster är också mycket trevligt att spela på, dock blir det tråkigare i längden p.g.a. dess enkelhet i jämförelse med A2-PB1. Programmet hamnar på en andraplats. På sista plats hamnar David's Midnight Magic. A2-PB1 kan köpas från Programdistributören, Box 3009, 580 03 Linköping. Tel. 013-121240. Raster Blaster från Pete & Pam, Waingate Lodge, Waingate Close, Rosendale BB4 7SQ, England. David's Midnight Magic från SBD, 15 Jocelyn Road, Richmond, Surrey, TW9 2TJ, England.

Tips

I förra numrets tips om diskdrivar tog vi upp att man skulle använda speciella rengöringsdisketter för att göra rent läs/skrivhuvudet. Ett problem är dock att få Applens drivar att snurra konstant i de 20 sekunder eller så för att de skall bli rena. Hittar Applen inget den kan läsa i driven stannar den efter ett par sekunder. Det finns dock ett sätt att kringgå det problemet. Man kan använda start/stopp adresserna i DOSen direkt. Värdena är:

	HEX	DEC
Starta	COE9	49375
Stoppa	COE8	49374
Drive1	COEA	49376
Drive2	COEB	49377

Alltså, för att få driven att snurra så skriv från Basic:
CALL-151 för att komma till monitorn
COEA för att välja drive1 el.
COEB för drive2
COE9 för att starta driven
COE8 för att stanna den

forts från sid 46

CONTROL:Ändrar på vissa parametrar i drivrutinen.

Blockenheter har ungefär liknande kommandon. Det som skiljer dessa är att den vid uppstarten av systemet, lägger beslag på den hårdvara den tänker använda, samt att READ och WRITE hanterar data i enheter om 512 tecken.

Medlemsvärning

Vi inför fr.o.m. detta nummer belöningar för alla som värvar medlemmar. Vi startar en medlemsvärvarävling. Den som värvar flest medlemmar till utgången av mars kommer att belönas med ett fint pris. Du får dessutom en belöning för varje medlem du värvar. Fyll bara i talongen och kryssa för ett av alternativen (OBS! glöm inte att fylla i vilket nummer ni önskar om ni beställer en tidning) så skickar vi belöningen så fort vi fått bankgiroinbetalningen. Värva på! Ju större vi blir ju lägre priser kan vi hålla.

Referenskortet

En mycket bra produkt som vi lanserade i det förra numret. För att ni ska veta mer om vad det innehåller så gör vi här en förteckning över innehållet. Kortet är i formatet 95X217 m.a.o. mycket behändigt. Det innehåller så mycket fakta att även den mer avancerade användaren har nytta av det. Priset är fortfarande låga 35:-.

- Sidan 1: 48K minneskarta med viktiga minneslokationer.
- 2: Instruktionssyntax 6502, cykler, 6502 kommando.
- 3: 6502 kommando forts.
- 4: 6502 kommando slut.
- 5: Monitorkommando, hex/dec konverteringstabell, intbasic-adressering.
- 6: IntegerBasic, Applesoft och DOS errormeddelande, Applesoft errorhantering.
- 7: DOS kommando, färgfunktioner
- 8: Applesoft interna koder, Applesoft reserverade ord, speciella tangentfunktioner.
- 9: Basicommando
- 10: Grafikkommando, funktioner
- 11: Funktioner, Basic speciella tecken och operander
- 12: Systemkommando, speciella pokes, peeks och calls
- 13: Jämförande tabell mellan video, 6502, integer Basic (t.ex. vad motsvarar decvärdet 43, hex 2B i resp. fall).
- 14: Se ovan (forts.)
- 15: Likt ovan men Applesoft istället för integer Basic.
- 16: Se ovan (forts.)

Kortet kan som vanligt beställas genom inbetalning till vårt bg. 551-5648.

Skickas till: Apple User Group
Dykens väg 22
281 00 Hässleholm

Mitt Namn:	_____
Adress:	_____
P.Adress:	_____
Den nye medlemmens Namn:	_____
Adress:	_____
P.Adress:	_____
Som belöning vill jag ha:	
<input type="checkbox"/> Windfall (nr: _____)	<input type="checkbox"/> Apple Orchard (nr: _____) 0/0
<input type="checkbox"/> Verbatim Datalife (oinspelad)	

KOMMUNIKATION MELLAN PROGRAM

Det finns i dag många bra programpaket som var för sig löser en uppgift. Men vad gör man om man vill hämta information från sin bokföring för att ge indata till en budgetsimulering? Eller hur använder man sitt lönesystem för att ge indata till en grafisk bild över lönespridningen?

Svaret på båda dessa frågor är "Data Interchange Format (DIF)". DIF är inte namnet på ett program utan är snarare ett koncept jämförbart med exempelvis DIN-normer för olika kontaktdon. DIN-normerna föreskriver exempelvis att man i kontakter av en viss typ ska kunna förvänta sig att vissa signaler återfinns på bestämda stift.

DIF föreskriver på motsvarande sätt att data ska vara arrangerade på ett bestämt sätt så att ett mottagande program kan tolka vad som finns lagrat.

Ett exempel

En DIF-fil förutsätts innehålla data som kan åskådliggöras i tabellform. Exempelvis kan innehållet i ett telefonregister visas på följande sätt:

	POST 1	POST 2	POST 3
NAMN	Ek Sven	Palm John	Asp Erik
ADRESS	Stigen 3	Gränden 4	Gatan 14
POSTNR	123 45	234 56	345 67
POSTADRESS	Xköping	Ystad	Zbyn
TELEFON	012-34567	123-45678	234-56789
NUM.KOD	10	12	34

När man lagrar data i DIF förekommer två viktiga begrepp, nämligen tupler och vektorer. Översätter man dessa begrepp till normal dataterminologi så kan en tupel jämföras med en post och en vektor med en viss term uttagen ur samtliga poster.

I ovanstående exempel kan informationen lagras på två sätt, antingen vertikalt eller horisontellt. Sker lagringen vertikalt så kommer den första tuplen att innehålla samtliga namn, den andra samtliga adresser o s v. Den första vektorn består i detta fall av "Ek Sven, Stigen 3, 123 45, Xköping, 012-34567".

Sker lagringen horisontellt så får man den traditionella lagringsformen med samtliga uppgifter om en person i en post.

Hur en DIF-fil ser ut

En DIF-fil indelas i tre olika typer av poster, nämligen startblock, dataposter och slutblock. Startblocket innehåller följande information:

```
TABLE
O,1
""
VECTORS
O,V          V=antalet vektorer
""
TUPLES
O,T          T=antalet tupler
```

DATA
0,0
"

Dataposterna innehåller var och en följande information:

-1,0
BOT
T1,N1
Sträng1 Första värdet i tuplen
T2,N2
Sträng2 Andra värdet i tuplen
T3,N3
Sträng3 Tredje värdet i tuplen

Tn,Nn
Strängn

T1, T2...Tn betyder typ av data. T kan anta följande värden:

0 = Numeriska data finns lagrade i N1..Nn
1 = Sträng1...Strängn innehåller en teckensträng
-1 = Speciella data. N ska vara 0. Strängvärdet kan anta värdena BOT
(Beginning Of Tuple) eller EOD (End Of Data).

Slutblocket innehåller följande information:

-1,0
EOD

Översätter vi vårt exempel, lagrat horisontellt, så får vi en DIF-fil med följande utseende:

TABLE	"TELEFON"	"Gränden 4"	V
0,1	1,0	1,0	-1,0
"	"NUM.KOD"	"234 56"	EOD
VECTORS	-1,0	1,0	
0,6	BOT	"Ystad"	
"	1,0	1,0	
TUPLES	"Ek Sven"	"123-45678"	
0,4	1,0	0,12	
"	"Stigen 3"	V	
DATA	1,0	-1,0	
0,0	"123 45"	BOT	
"	1,0	1,0	
-1,0	"Xköping"	"Asp Erik"	
BOT	1,0	1,0	
1,0	"012-23456"	"Gatan 14"	
"NAMN"	0,10	1,0	
1,0	V	"345 67"	
"ADRESS"	-1,0	1,0	
1,0	BOT	"Zbyn"	
"POSTNR"	1,0	1,0	
1,0	"Palm John"	"234-56789"	
"POSTADRESS"	1,0	0,34	
1,0			

Visi-programmen och DIF

Vi närmar oss nu pudelns kärna, d v s Visi-programmen. En av hemligheterna med VisiCorps framgångar är just DIF. Praktiskt taget samtliga Visi-program kan kommunicera med varandra med DIF-filer. DIF har även i stor utsträckning blivit en standard utanför VisiCorp. Det finns i dag ett antal programprodukter som kan läsa och skriva DIF-filer.

Har du någon gång viljat göra höggrafikbilder med cirklar, rektanglar, linjer och även text inblandat men snabbt gett upp för att du tyckt att det varit för krångligt? I så fall kan EZ-draw 3.3 från Sirius software vara något för dej. Programmet erbjuder grafikritning på ett mycket enkelt sätt. När du laddat in programmet så finner man en tom skärm med ett hårkors i mitten. Detta hårkors fungerar som din "pensel". Nu kan man börja rita i stort sett vad man vill. Cirklar t.ex. skapas genom att man går till mittpunkten, slår mellanslag, flyttar ut till ytterkanten av cirkeln och trycker antingen O för ofylld cirkel eller ctrl-O för fylld cirkel och vips har man en cirkel på skärmen. Sambandet med O ctrl-O, ofylld fylld, gäller för all andra figurer också. Förflyttningen av hårkors sker med I J K M, stegningen väljer man i en skala från 1 punkt till 9 punkters förflyttning i taget. Vill du rita en linje slår man ett snedstreck och börjar flytta på hårkors och en linje följer där man flyttar, enklare kan det inte bli.

Det finns funktioner för de flesta figurer, rektanglar, trianglar, paralellogram, cirklar och ellipser. Det finns även ett mycket användbart flyttningskommando med vars hjälp man lätt flyttar omkring sina figurer på skärmen. I programmet finns även en del specialfunktioner där man kan rotera delar av skärmen, vända upp och ner på bilden, trycka ihop den på höjd eller bredd eller expandera den på dito eller rotera 90 grader. Givetvis kan allt fås i färg också; man kan välja färg både på figur och

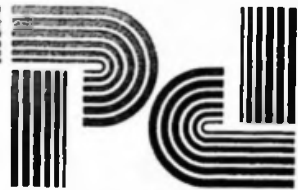
bakgrund.

En mycket trevlig möjlighet i programmet är att det medföljer ett textprogram som ritlar i höggrafik (Higher Text) med vilket man lätt blandar text och grafik. På disketten finns ett otal stilar att välja mellan. Textinskrivning sker genom att trycka shift-P, man får då upp en markör istället för hårkors, för att man skall se skillnad på text och grafikmode. Väl inne i textmode så är det bara att börja skriva text på helt vanligt sätt, dock med den skillnaden att man kan få olika storlekar på stilen. De olika storleksvarianterna som finns är extra breda, extra höga och extra stora både på höjd och bredd, förutom vanlig liten stil. Även i textmode kan man få olika färger på bokstäverna, man kan till och med få fyra extra färger förutom de vanliga (gult, skärt, blå-violett och ljusblått) i "expanded mode" alltså den största stilen. När textinmatningen är färdig trycker man shift-P igen och så är man tillbaka i grafikmode igen.

Sammanfattningsvis kan sägas att programmet är mycket lättanvänt och kraftfullt och behöver man inte 3-D eller rörlig grafik så kan E-Z 3.3 vara ett mycket intressant program. För de 405kr inkl.moms som programmet betingar får de anses som prisvärt.

Programmet kan köpas från:
Programdistributören, Box
3009, 580 03 Linköping.

Vill man hellre ringa är
numret 013-121240.



PROGRAMDISTRIBUTÖREN Box 3009 580 03 LINKÖPING Tel. 013 - 12 12 40



Bag of Tricks™

Don Worth and Pieter Lechner

BAG OF TRICKS

Av Don Worth och Pieter Lechner

Nu finns det ytterligare en bok av författarna till BENEATH APPLE DOS. Över 100 sidor med värdefull information om Apple II DOS, dessutom medföljer en diskett med fyra omfattande utility program.

BAG OF TRICKS är användbar både för nybörjare och avancerade programmerare. Den innehåller många noggranna beskrivningar av hur du kan reparera trasiga disketter, rekonstruera trasiga kataloger, ändra sectorordningen m m.

De fyra utility programmen är:

- 1 TRAX dumpar och visar ett track, antingen 13-sector eller 16-sector, visar den interna formatterings-informationen, och markerar avvikelser från standardformatet.
- 1 INIT formatterar ett eller flera tracks med försöker bibehålla innehållet i oförstörda sectorer. Det ger dig även möjlighet att ändra sectorernas ordning. Det kan öka access-tiden med 40 % eller mer!
- 3 ZAP är en sector editor. Mer än 50 kommandon ger dig möjlighet att hitta, jämföra, ändra och skriva data på skivan. Med kraftfulla macrokommandon kan du programmera ZAP. Du kan flytta och jämföra DOS, CP/M och PASCAL filer.
- 4 FIXCAT automatiserar reparationen av trasiga diskettkataloger. Den kan köras automatiskt eller med manuella kommandon och hittar tappade filer och konstruerar katalogen – helt automatiskt! Du kan också ta bort DOSen från skivan och reparera VTOC.

Bok (engelsk text) och diskett DOS 3.3

395:-

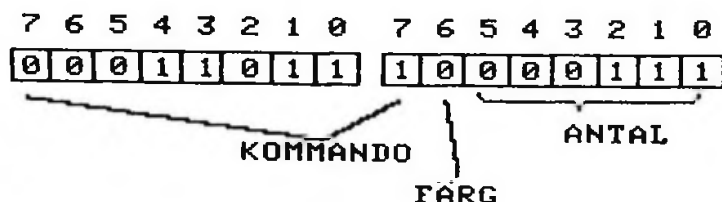
Bildpackning

av *Thomas Ahlström*

Som de flesta människor som arbetar med Apple II så har jag alltid ont om ledigt utrymme på mina disketter. För att kunna få plats med lite mer har jag utvecklat en rutin för att omvandla en grafisk bild till ett datafält. Om man studerar en grafisk bild lite närmare så ser man att bildytan innehåller stora mängder svart och vit information. Vad rutinen PICTURE PACKER gör är att räkna om dessa långa serier av lika bytes och lagra dessa som speciella kommandobytes vilka sedan kan räknas om till en grafisk bild.

En begränsning med denna rutin är att den inte klarar av att representera samtliga färger utan färgerna orange/blått blir istället violett/grönt. Men om man endast kör svart-vita bilder har man inte förlorat något. Nedanstående bild kan ge ett litet hum om hur en bild lagras i minnet, men om du vill veta hur detta går till så refererar jag till REFERENCE MANUAL APPLE II, sidorna 19 och 20.

Data i en packad fil kan se ut enligt följande:



När inte färgerna utnyttjas fyller inte bit 7, inom alla bytes på höggrafiksidorna, någon funktion utan kan nyttjas till annat. En nolla i bit 7 indikerar att byten är en vanlig databyte som skall läggas i grafik-minnet.

En etta markerar att det är en kommandobyte. Bit 6 ger färgen som skall repeteras (1=vitt/0=svart) och bitarna 5 till 0 representerar repetitionens längd. Skulle dessa vara noll så indikerar det att datafilen är slut.

Under packning av en bild så ligger bilden på adress $\$2000$ till $\$3FF7$, Applesoftprogrammet på $\$800$ - ca $\$900$ och det binära packprogrammet på $\$7000$ till $\$7100$.

Vid upppackning BRUN'as filen på adress $\$4000$ och bilden läggs på adresserna $\$2000$ till $\$3FF7$ (sidan 1).

Arbetsordning när du skall genomföra en packning är:

RUN PACKER <CR>

packa vilken bild:MIN BILD <CR>

När diskdriven sedan har stannat så går du ur programmet med ctrl-c och RETURN.

För att du sedan skall kunna återfå din bild, skriv:

HGR <CR>

BRUN MIN BILD.PIX <CR>

Sedan är det bara att försätta som vanligt med ditt program.

Programmen är skrivna i Applesoft och maskinkod med hjälp av assemblern MAC 65.

För att få ett bra grepp om hur detta program fungerar bör du läsa i REFERENCE MANUAL hur grafiken egentligen lagras i minnet.


```

1000 *****
1010 *
1020 * ***** PACK 1.0 ***** *
1030 *
1040 * RUTIN FÖR ATT PLYTTA EN GRAFIK*
1050 * BILD TILL EN PACKAD FIL *
1060 *
1070 * >THOMAS AHLSTRAND*
1080 * >STELLAR KJUKVARA HBK *
1090 *****
1100 *****
1110 I VARIABLER
1120
009D: 1130 TARSET EDU 0FD, 9E
009F: 1140 SOURCE EDU 0FF, A0
00A1: 1150 XLEN EDU 0A1
00A2: 1160 SOURCE EDU 0A2, A3
00A4: 1170 TEMP EDU 0A4
1180
1190 I UPPSTARTNINGSPAS
1200
7000: A9 00 1210 ORS 07000 START AV PROB
7002: 05 9F 1220 LDA 080 I
7004: A9 20 1230 STA SOURCE I SOURCE=02000
7006: 05 A0 1240 LDA 0820 I
7008: A9 39 1250 STA SOURCE+1 I
700A: 05 9D 1260 LDA 0839 I TARSET=04039
700C: A9 40 1270 STA TARSET I
700E: 05 9E 1280 LDA 0840 I
1300 STA TARSET+1 I
7010: A2 00 1310 LDX 00 01LEN=0
7012: 06 A1 1320 STX XLEN 0
1330
7014: A5 A1 1340 LOOP LDA XLEN I SOURCE=X
7016: 18 1350 CLC I SOURCE=XLEN
7017: A5 9F 1360 ADC SOURCE I
7019: 05 A2 1370 STA SOURCE I
701B: A9 00 1380 LDA 00 I
701D: A5 A0 1390 ADC SOURCE+1 I
701F: 05 A3 1400 STA SOURCE+1 I
1410
7021: A5 A2 1420 LDA SOURCEX TESTA OM DET
7023: 09 F9 1421 CMP 08F9 AR SLUT PA
7025: 00 08 1422 BNE EJSLUT GRAFIKBIDA 1
7027: A5 A3 1423 LDA SOURCEX+1
7029: 09 3F 1424 CMP 083F
702B: 00 08 1425 BNE EJSLUT
702D: A5 A4 1426 LDA TEMP
702F: 4C 0C 70 1427 JMP ENDPAGE
1430
7032: A4 A1 1440 EJSLUT LDY XLEN TEST OM REPET
7034: A5 A4 1441 LDA TEMP
7036: 00 3F 1470 CPY 063 BRANS AR NADD
7038: 00 10 1480 BCS ENDL0OP
1490
703A: E6 A1 1490 INC XLEN XLEN=XLEN+1
1510
703C: A0 01 1520 LDY 01 TESTA OM
703E: 01 A2 1530 LDA (SOURCEX), Y BYTE+1 OCH
7040: 29 7F 1540 AND 087F BYTE+2 AR
7042: 08 1550 DEY LIKA.
7043: 05 A4 1560 STA TEMP OM INTE
7045: 01 A2 1570 LDA (SOURCEX), Y AVSLUTA
7047: 29 7F 1580 AND 087F LOOPEN
7049: 03 A4 1590 CMP TEMP
704B: 00 0A 1600 BNE ENDL0OP
704D: 05 A4 1601 STA TEMP
1610
704F: 09 00 1620 CMP 00
7051: F0 C1 1630 BEQ LOOP TESTA OM FARSEN
7053: 09 7F 1640 CMP 087F AR SVART ELLER
7055: F0 8D 1650 BEQ LOOP VIT.
1660
7057: 20 6E 70 1660 ENDL0OP JBR SAVERUT
705A: A5 A1 1670 LDA XLEN
705C: 18 1680 CLC
705D: A5 9F 1690 ADC SOURCE
705F: 05 9F 1691 STA SOURCE
7061: A9 00 1692 LDA 00
7063: A5 A0 1693 ADC SOURCE+1
7065: 05 A0 1694 STA SOURCE+1
7067: A2 00 1695 LDX 00 I
7069: 06 A1 1696 STX XLEN
706B: 4C 14 70 1697 JMP LOOP
1698
706E: A0 00 1670 SAVERUT LDY 00
7070: A6 A1 1680 LDA XLEN
7072: E0 01 1690 CFI 01
7074: 00 03 1900 BNE REPET
7076: 4C 83 70 1910 JMP UPDAT
7079: 09 00 1920 REPET CMP 00
707B: F0 02 1930 BEQ VIT
707D: A9 40 1940 LDA 001000000
707F: 09 80 1950 VIT ORA 0X10000000
7081: 05 A1 1960 ORA XLEN
7083: 01 9D 1970 UPDAT STA (TARSET), Y
7085: E6 9D 1980 INC TARSET
7087: 00 02 1990 BNE 0
7089: E6 9E 2000 INC TARSET+1
708B: 60 2020 RTS
2030 I AVSLUTNING AV PROGRAM
2040
708C: 20 6E 70 2050 ENDPAGE JBR SAVERUT
708F: A9 80 2060 LDA 0880
7091: 01 9D 2070 STA (TARSET), Y
7093: A5 9D 2080 LDA TARSET
7095: 8D 00 03 2090 STA 0300
7098: A5 9E 2100 LDA TARSET+1
709A: 8D 01 03 2110 STA 0301
709D: 60 2120 RTS

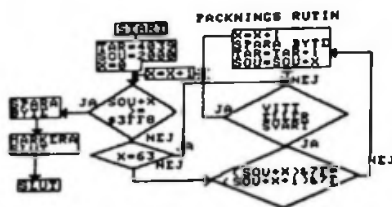
```

SYMBOLER:

```

7032: EJSLUT
7057: ENDL0OP
01=7047
708C: ENDPAGE
7014: LOOP
7079: REPET
704E: SAVERUT
009F: SOURCE
00A2: SOURCEX
009D: TARSET
00A4: TEMP
70E3: UPDAT
00=708B
707F: VIT
00A1: XLEN

```



INBA FIL I ASSEMBLERINGEN

```

1000 *****
1010 *
1020 * ***** UNPACK 1.0 ***** *
1030 *
1040 * RUTIN FÖR ATT PLYTTA EN PACKAD *
1050 * FIL TILL HÖRUPPLÄSNANDE GRAFIK *
1060 * BIDA 1 *
1070 * >THOMAS AHLSTRAND*
1080 * >STELLAR KJUKVARA HBK *
1090 *****
1100 *****
1110 I PROB STARTAR VID ADRESS 04000
1120
1130 ORS 04000
1140
1150 I START AV PROB
1170 LDY 00 POKARE TILL RUTA
1180 BEG DATA
1190
4000: A0 00 1170
4002: F0 08 1180
1190
4004: E6 00 40 1200 LOOP INC DATA+1
4007: 00 03 1210 BNE DATA
4009: E6 0E 40 1220 INC DATA+2
400C: A0 39 40 1230 DATA LDA SLUTPR
400F: 30 0A 1240 BNE KORREKT KORR OM KORR++
1250
4011: 20 2D 40 1260 JBR RUTA RITA UT BYTE
4014: 4C 04 40 1270 JMP LOOP
1280
4017: A8 1290 KORREKT PMS TESTA OM SLUT
4018: 29 3F 1300 AND 083F PA PROB
401B: F0 1B 1310 BEQ SLUT
401C: A4 1320 TAX ANTAL (REPET)
401D: 68 1330 PLA
401E: 29 40 1340 AND 0840 SVART/VITT ?
4020: F0 02 1350 BEQ SVART
4022: A9 7F 1360 LDA 087F VITT!
4024: 20 2D 40 1370 SVART JBR RUTA I LOOP BOM
4027: CA 1380 DEI I RITA UT I
4028: D0 FA 1390 BNE SVART I ANTAL A
402A: 4C 04 40 1400 JMP LOOP
1410
1420 I SUBROUTIN BOM PLACEREN EN BYTE
1430 I PA ADRESS 'RUTA' OCH BEDAN
1440 I KORR POKAREN TILL RUTA MED ETT
1450
402D: 99 00 20 1460 RUTA STA 02000, Y
4030: 08 1470 INY
4031: D0 03 1480 BNE 0
4033: E6 2F 40 1490 INC RUTA+2
4036: 60 1500 .O RTS
1510
1520 I AVSLUT FRAN PROB
1530
4037: A8 1540 SLUT PLA
4038: 60 1550 RTS
1560 SLUTPR

```

SYMBOLER:

```

400C: DATA
4017: KORREKT
4004: LOOP
402D: RUTA
00=403A
4037: SLUTPR
4024: SVART

```

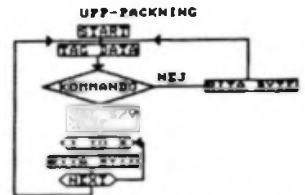
INBA FIL I ASSEMBLERINGEN

ALIST

```

10 REN *****
20 REN >PICTURE PACKER 1.0
30 REN >THOMAS AHLSTRAND*
40 REN >STELLAR KJUKVARA *
50 REN *****
60 REN >ALLA KORREKTURELLA*
70 REN >RÄTTIGHETER *
80 REN >RESERVERADE *
90 REN *****
100 *
110 TEXT I HOME I PRINT CHR(4)"HOMONT,C,0"
115 ORDRS 0000 400
120 PRINT ">>>PICTURE PACKER 1.0<<<<"
130 PRINT "NAMN ETT '0' FRÅNFRÄMMET OM BILDEN REDAN LINNEN PA HÖRUPP *
135 LÄSNANDE BILD BIDA1 RESPECTIVE SKIVA.
140 PRINT "EN '1' ELLER '2' BER <CATALOG> PA
150 BLJ = CHR(14) + "BL0AD"
160 PRINT BLJ"UNPACK, OBJ, 040000
170 PRINT BLJ"PACK. OBJ, 047000
180 I
190 POKE 49233, PEEX (49234)
200 INPUT "PACKA VILKEN BILD: I AR
205 IF AR = "" GOTO 200
210 IF AR = "1" OR AR = "2" THEN PRINT CHR(4)"CATALOG"AR: GOTO 200
220 IF LEFTH (AR,1) = "8" GOTO 230
230 INPUT "KÖLDRIVE: I AR
240 IF KB (>) "1" AND KB (>) "2" GOTO 200
250 INPUT "HALDRIVE: I AR
260 IF KB (>) "1" AND KB (>) "2" GOTO 200
280 POKE 49233, PEEX (49239)
290 IF LEFTH (AR,1) = "8" THEN AR = RIGHT (AR, LEN (AR) - 1): GOTO 340
300 PRINT BLJ"AR", AR2000, B"KB
340 CALL 28672
350 L = PEEX (748) + PEEX (749) + 256 + 1
360 L = L - 16384
370 PRINT CHR(4)"*****AR".PIN, AR4000, L"L", B"PE
380 GOTO 190
400 A = PEEX (222)
408 POKE 49233, PEEX (49234)
410 IF A = 230 THEN PRINT "SVARTT MED CTRL-C": END
420 IF A AND A < 16 GOTO 500
430 PRINT "REL "A" I SMPLEOPT!": END
500 I A = 2: V = 43379
510 Y = V + 1
520 IF PEEX (V) < 128 GOTO 510
530 I = X - 1: IF I GOTO 510
535 INVERSE
540 Y = Y + 1
580 PRINT CHR(4) PEEX (V): I
600 IF A = 2: V = 43379
645 PRINT I NORMAL
570 GOTO 190

```



Du  **apple användare!**

**Medlemmar i AUG kan
alltid räkna med bästa
support och pris**

hos



HANDELSBOLAGET

modern teknik

Stjärnfallsvägen 1, Box 204, 281 01 Hässleholm
Tel. 0451-85585

Auktoriserad
 **apple computer**
® Återförsäljare + Servicecenter

VisiDex (TM) är ett varumärkesskyddat program från VisiCorp, programvaruhuset som har givit oss VisiCalc (R). Programmet är skrivet av Peter Jennings. Många av oss som har hållt på med Apple några år känner igen namnet. Peter Jennings är mannen bakom det oerhört populära programmet MicroChess.

Databas program?

=====

VisiDex är

så olikt traditionella "databas"-program, att det vid första titten kan vara svårt att förstå all problem det löser och ger svar på. Nästan varje människa behöver någon form av korsreferensinformation, och det är det behovet som VisiDex tillfredställer.

Till vad? För vem?

=====

Att kunna söka igenom en kundlista för att få kontakter som ska göras före ett visst datum, att kunna få fram vem i Enköping som har ett hus att sälja OCH kan hjälpa till med finansieringen eller vilken firma som kan köra småspik upp till Norrland OCH har kapacitet ledig nästa vecka. Andra exempel: Alla advokater behöver hålla ordning på kontrakt, pågående förhandlingar, besök i rätten, etc. VisiDex är ett utmärkt hjälpmedel för att hålla reda på sådan kalenderrelaterad information. Journalister håller ofta på med mer än en artikel åt gången, var och en handlar om olika ämnen, folk som ska intervjuas, deadlines och redaktörer. Många författare samlar på uppslag till sina historier. Alla kan ha nytta av VisiDex för att organisera sitt material. Vetenskapliga institutioner samlar referenser om artiklar från olika tidskrifter.

De fyra viktigaste delarna i VisiDex:

- korsreferenser --- 252 tecken är tillgängliga per skärmbild. Det ger omkring 36 nyckelord om sju tecken var. (Det går också att söka på vilken text som helst, man behöver inte nyckelord -- det tar bara något längre tid.)
- enkelhet --- VisiDex programmet kräver ingen planering, som konventionella datahanteringssystem gör. Det är lika fritt att använda som en bit papper eller traditionella indexkort (vilket VisiDex är närbesläktat med). Kommandoraden liknar de som finns i Visicalc och Visiplot, vilka många redan är bekanta med. Kommandona kan fås i klartext m h a en knapptryckning. Lektionsdelen i manualen hjälper också till att göra det enkelt.
- snabbhet --- VisiDex är snabbt. Ingen extra sökning efter sökorden, nyckelorden finns i internminnet, med direktadress till platsen på disketten. Programmet startar t o m upp skivminnet i förväg för att uppstartningstiden inte ska göra åtkomsttiden längre. Sökning efter ICKE-INDEKERADE ord tar något längre tid, men det är ändå bara frågan om sekunder, vilket är betydligt snabbare än att bläddra igenom en låda med indexkort.
- kalender --- programmet kan visa, på skärm eller på skrivare, en sammanfattande kalender för valfri månad mellan 1900 och 2000 och ange ev PM du lagrat under ett visst datum. Det är enkelt att få fullständiga rapporter veckovis. VisiDex kan även skriva ut kalendrar över ett helt år.

Textredigering

=====

Programmet har helt fritt format. Man kan skriva var som helst på skärmen. Kommandon finns för:

- markör upp, ned, framåt och bakåt
- tag bort tecken och sätt in tecken var som helst i texten
- text svart på vitt, vitt på svart eller blinkande
- speciella fält för fast text; lämpar sig för inmatningsbilder, textskärmsblanketter (gör ett register)
- tabulering
- sudda skärmen
- tag bort hel rad och lägg i textbuffert
- lägg in hel rad från textbuffert
- töm textbuffert
- läs in en rad från vanlig textfil (man kan alltså hämta text från andra register och program!)
- skapa nyckelord

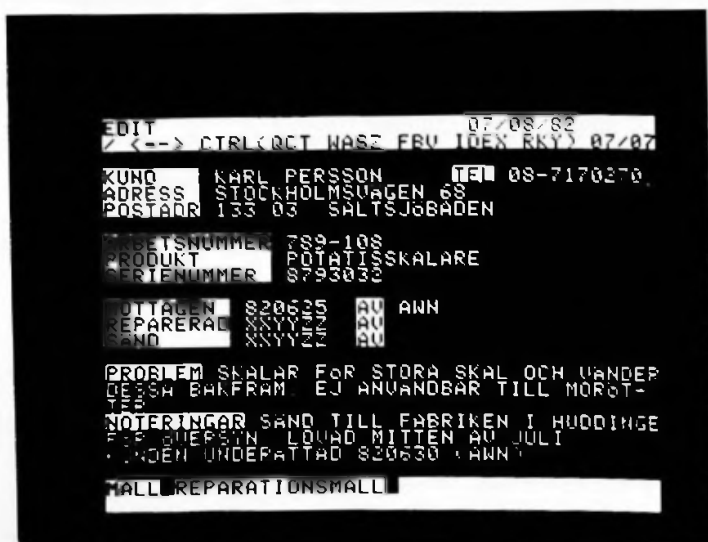


Bild 1: FÖR IN INFORMATION
Man kan betrakta VisiDex som ett elektroniskt kartotek. Som synes ryms minst lika många uppgifter som på ett stort registerkort.

Nyckelord

=====

Att skapa nyckelord är enkelt. Sätt markören på aktuellt ord och tryck på C. Man kan också skapa nyckelord som inte finns på bilden genom att skriva det direkt.

Det går att skapa mycket sammansatta sökbegrepp. Genom att använda tecknet & kan man sätta ihop flera nyckelord och få fram den bild som har alla de aktuella nyckelorden.

Det går också utmärkt att använda sk "wildcards". Bindestreck - och asterisk * kan finnas var som helst i nyckelordet, utom i första, andra och sista tecknet. De båda tecknen kan användas i samma nyckelord. Varje tecken kan också användas mer än en gång i samma nyckelord. Bindestrecket (-) ersätter vilken textsträng som helst i ett nyckelord. Asterisken (*) ersätter endast ett tecken.

Att söka numeriskt går naturligtvis bra det också. För att det ska gå att söka numeriskt, måste skärmbilden innehålla ett fältnamn, ett likhetstecken eller ett kolon och ett talvärde. Det får förekomma blanka tecken, kommatecken och punkter i talet. När man söker på ett numeriskt värde kan man söka på:

- = lika med
- < mindre än
- > större än
- <= mindre än eller lika med

- ... re än eller lika med
- ... >< inte lika med

Man kan också söka på numeriska värden tillsammans med nyckelord och &-tecken.

Det går som synes att skapa sig hur komplexa sökningar som helst. Eftersom du själv bestämmer vilka hänvisningar, nyckelord, din information skall ha, så "minns" VisiDex på samma sätt som du gör. Alla sökningar man gör på skärmen går naturligtvis att få ut på en skrivare.

Nyckeldatum

=====

Som framgått ovan kan man till varje skärmbild lägga ett eller flera nyckelord att söka på. Man kan också "stämpla" sina bilder med datum. När man lagrar undan en datumstämplad bild på disketten, får man möjlighet att tala om, om man vill ha den automatiskt visad då man startar upp programmet. Antag att bilden ha 24 december som nyckeldatum. Frågor som ställs är bl a

- hur många dagar före angivet datum ska bilden visas? (Mellan 0 och 15 dagar före är möjligt.)
- ska bilden visas varje 24:e i månaden?
- ska bilden visas var 4:e fredag varje månad?
- ska bilden visas varje fredag?

Det går att söka på datum på samma sätt som på nyckelord.

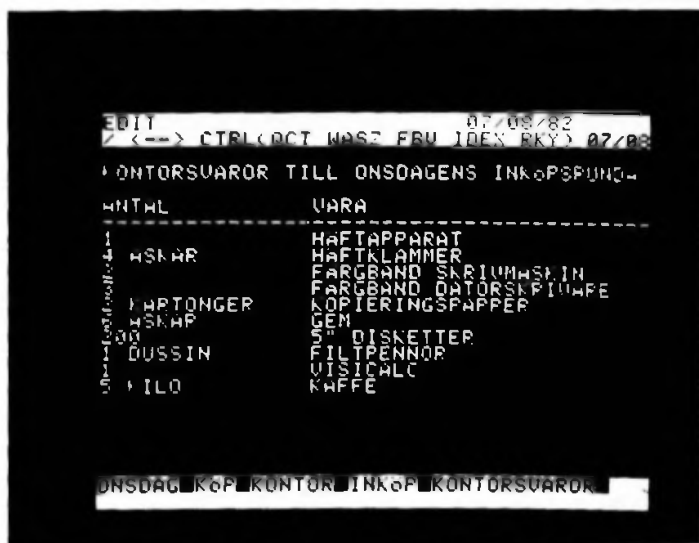


Bild 2: SPARA UPPGIFTER
Ange de nyckelord som de inmatade uppgifterna har anknytning till. Spara skärmens innehåll på diskett.

Realtidsklocka

=====

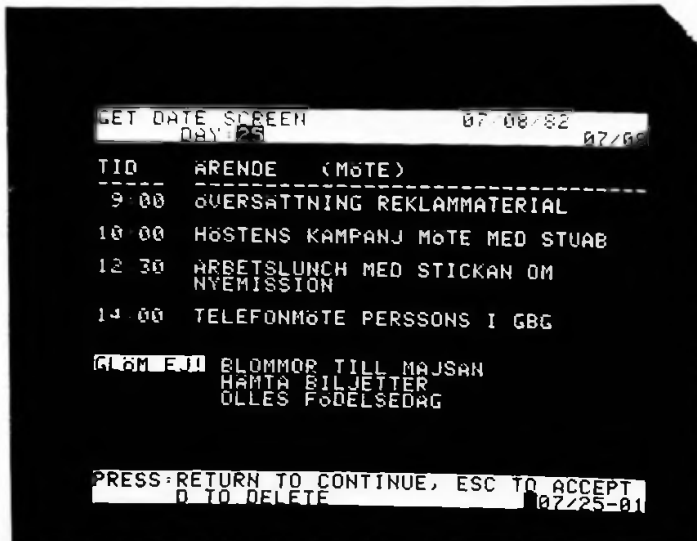
Programmet supportar realtidsklockor, som t ex från Mountain Computer och California Computer System. Många klockor supportar å andra sidan VisiDex. Klock-kortet jag själv använder heter SuperClock II från West Side Electronics. Till kortet finns bl a ett program som ändrar lite i VisiDex så att programmet kan nyttja klockan. Fantastiskt bra. Det enda man behöver göra när programmet startas, är att tala om i vilken utgång, slot, kortet sitter i. Sedan utnyttjar programmet tiden som klockan ger in på sekunden. Det går t o m att använda Appledatorn som väckarklocka. Väcker användaren i tid till viktiga möten. Eftersom man kan "stämpla" skärmbilderna med såväl nyckelord som nyckeldatum, är en realtidsklocka ett bra komplement då den håller reda på tiden åt användaren.

Skillnad mellan VisiDex och traditionella filhanteringssystem
 =====
 Personal Software har även ett "vanligt" filhanteringssystem kallat
 VisiFile (TM). Tabell 1 visar skillnaden mellan VisiDex och VisiFile.

Datorkrav
 =====

VisiDex går att köra på Apple II med minst 48K minne och en flexskivenhet och DOS 3.3, med eller utan språkssystem. Eller Apple /// i Apple II emuleringsmod.

Bild 3: BEGÄR INFORMATION
 VisiDex kan kalla fram önskad information på en mängd olika sätt - efter datum, nyckelord, nummer, siffror mm



VisiDex

VisiFile

VisiDex	!	VisiFile
Olika typer av information som kan vara i samma fil	!	Information av samma typ i en fil, olika typer av inform. i olika filer.
Gjort för snabb åtkomst genom användning av nyckelord eller nyckeldatum eller sökning av vanlig text	!	Gjort för sortering och utskrift av stora mängder data, eller åtkomst genom upp till 24 förplanerade index.
Skriver ut kalendrar, veckorapporter eller etiketter	!	Skriver ut stora adresslistor och rad- och kolumnrapporter med vissa beräkningsmöjligheter.
Gjort för åtkomst, men kan skriva ut enkla, informella rapporter	!	Gjort för mycket flexibla rapporter, men kan användas för direktåtkomst med information på bildskärm.

Tabell 1

Du Norrlänning!

Vi representerar  **apple** i Skellefteådistriktet

Alltid i lager:



Apple III

av ... ström

Inledning

Detta är andra delen av min djupdykning i Apple III. Del 1 dök upp i föregående nummret av Apple User News (Nr 3-82 för att vara exakt). Då tog jag för den som är intresserad upp följande:

En liten historik om Apple

Hårdvaran:

- Utseendet
- Minnet
- Tangentbordet
- Bildskärmen
- Anslutningar till yttervärlden
- Handböckerna
- Hårddiskar

Sedan kom:

- Allmänt om operativsystem
- Kommunikationen med SOS i maskinen
- De tre SOS-filerna och Interrupt.

Den som är intresserad av detta nummer, kan ta kontakt med någon utgivare, som ansvarar för denna blaska.

DRIVRUTINER

Som jag nämnde i föregående nummret, är en drivrutin, ett program i maskinnära kod, som styr en sak som är ansluten till en Apple III.

För att kunna skilja på alla olika drivrutiner, har varje drivrutin ett namn. Detta namn måste börja med en punkt (.) och sedan följas av högst 15 bokstäver.

Vill man från ordbehandlingsprogrammet Apple Writer III skriva ut till den termiska skrivaren SILENTYPE, så anger man att utskriften skall skickas till drivrutinen med namnet .SILENTYPE. Denna drivrutin skickar i sin tur ut all text till skrivaren.

Fördelen med detta är att operativsystemet inte behöver veta vart den skickade utskriften ska hamna. Vissa standardprogram har som utgångsvärde att utskriften skall skickas till den drivrutin, som har namnet .PRINTER. Om man då bara har en SILENTYPE-skrivare ansluten till Apple III, så går det att döpa om Silentyedrivrutinen till .PRINTER varvid det går utmärkt att acceptera utgångsvärdet.

Vi tar ett exempel: Om en person använder programmet Visicalc III (Ett program som används för simuleringar av budgets och liknande saker), och har två skrivare anslutna. En skönskrivare och en SILENTYPE-termisk skrivare. Den termiska skrivaren använder hon (kan i 40

och för sig också vara en han, men det struntar vi i för korta utskrifter när hon arbetar med sina beräkningar. En enstaka gång vill hon få modellen prydligt utskriven på en skönskrivare.

Eftersom man från programmet kan skicka utdata till antingen drivrutinen med namnet .PRINTER eller till en drivrutin med ett av användaren specificerat namn, är det lämpligt att döpa om drivrutinen som går till hårdvaran Silentye till .PRINTER och drivrutinen som går till skönskrivaren till .PERTEC eller något annat lämpligt.

Hade det varit så att hon använt skönskrivaren oftast, så får man döpa om skönskrivaren till .PRINTER och kalla drivrutinen som går till Silentye för .SILENTYPE.

VAD KAN ANVÄNDAREN ÄNDRA I EN DRIVRUTIN?

I en drivrutin, finns det ett antal olika saker som en användare kan ändra. När man skriver ut till en skrivare, kan man ha något som kallas BAUD-rate eller bitar per sekund.

I ett alfabet finns det 96 olika tecken. Det finns också 32 speciella tecken som används för att rensa skärmen mm. Detta blir sammanlagt 128 olika tecken.

LITE OM BITAR OCH BYTE

Den grundenhet, som Apple arbetar med är en BYTE. Den består av åtta bitar som antingen kan vara 0 eller 1 vilket internt i Apple motsvaras av ström eller icke ström. Man kan tänka sig en ramp med åtta stycken lampor, som antingen kan vara tända eller släckta. Har man bara en lampa, så kan den vara ställd på två olika saker. Antingen tänd eller släckt. Har man två lampor, finns det fyra olika lägen. Har man tre lampor, finns det åtta olika enligt följande. "S" betyder släckt och "T" betyder tänd.

	lampnummer			
	3	2	1	
1	S	S	S	De olika varianterna
2	S	S	T	
3	S	T	S	
4	S	T	T	
5	T	S	S	
6	T	S	T	
7	T	T	S	
8	T	T	T	

Är man lite händig i matte, inser man att antal möjliga sätt att ställa en ramp bestående av N lampor, är 2^N eller $2*2*2*2... 0$ s v N gånger. Hade vi tre lampor, så blir det $2*2*2$ (3 stycken tvåor) olika möjligheter. Det vill säga 8 olika möjligheter, vilket var det vi just kom fram till. Atta lampor blir $2*2*2*2*2*2*2*2=256$ olika lägen. Som vi kommer ihåg, fanns det ju åtta bitar i en byte. En byte kan alltså anta 256 olika

värden. Ett exempel är 01001101. Detta är ju inte så speciellt lätt att komma ihåg! man har därför föredragit att gruppera ett sådant byte i två grupper 0100 och 1101. Varje sådan grupp kan inta 16 olika värden. (kommer du ihåg 2*2*2*2?)

0100 är fortfarande lite otympligt att skriva och komma ihåg. Därför skrivs

0000 till 0
0001 till 1
0010 till 2
0011 till 3
0100 till 4
0101 till 5
0110 till 6
0111 till 7
1000 till 8
1001 till 9

Sedan dyker det upp ett litet problem! Vad skall 1010 skrivas till? 10 blir lite otympligt. Det hade varit praktiskt att bara ha en siffra. Men siffrorna har ju tagit slut. Men vi har ju bokstäver!

1010 blir siffran A
1011 blir siffran B
1100 blir siffran C
1101 blir siffran D
1110 blir siffran E
1111 blir slutligen siffran F

Detta sätt att ha 16 stycken olika "siffror" kallas det hexadecimala talsystemet. en siffra är alltså 0 från F.

Det vi var först inne med var det binära talsystemet med sina siffror 0 till 1
Vårt eget talsystem med 10 olika siffror, som vi normalt använder när vi räknar, kallas det decimala.

För att återgå till BYTEN 0100 1101. Detta blir alltså 4 D eller sammanslaget 4D, (appropå ingenting undrar när vi får fyrdimensionell TV?) vilket kan vara mycket enklare att komma ihåg. Ett sätt som ofta förekommer är att skriva ett dollartecken (\$) framför ett hexadecimalt tal för att skilja det från ett decimalt tal om inget annat framgår. Ex: \$4D

7 stycken bitar, kan anta 128 olika värden. Varvid vi är tillbaka där vi började. Det finns 128 olika tecken. Med andra ord: det behövs 7 stycken bitar för att rymma alla möjliga tecken. Varje tecken har en viss unik kod. Denna kod kallas ASCII-kod. Tecknet A har till exempelkoden 0100001 eller \$41; B har koden 01000010 eller \$42 osv.

Den sista överblivna biten, för att det skall bli en hel BYTE, används till olika saker. På Apple /// används den för att tala om om ett tecken är inverterat (svart text på vit bakgrund) i bildskärmsminnet. (Den del av Apple /// minne som används för att lagra de tecken som just syns på bildskärmen.) Vid utskrift betyder det något helt annat o s v.

Baudraten, som vi var inne på är alltså hur många bitar per sekund som sänds till skrivaren. Olika skrivare vill ha en viss specificerad hastighet för att kunna ta emot det som sänds över till den. Detta är olika för olika skrivare.

I de flesta olika drivrutinerna, finns det något som kallas Configuration Block (nedan förkortat CB). Här kan anges olika värden som till exempel vilken hastighet, som en drivrutinen skall använda för att sända ut sina data. Även andra saker som användaren kan behöva ändra för att anpassa systemet till sina speciella behov och hårdvara som har anslutit kan ändras. Dessa data ligger i drivrutinen och laddas samtidigt med den vid uppstart av en Apple ///.

Dessa CB anger man just med dessa hexadecimala koder. Vad de olika koderna står för hittar man antingen i handboken Standard Device Drivers eller den handbok, som följer med den yttre enhet som man köpt.

Drivrutinen hämtar sedan in dessa CB när den används och använder dessa data för att tala om hur den skall bete sig.

Man har också i en drivrutin möjlighet att lägga in en förklarande rad på cirka 80 tecken. I och med att det går att ändra namnen på drivrutinerna och även CB, så kan det vara bra att sätta in en förklaring på vad de aktuella CB är för något samt vilken hårdvara koderna i drivrutinen styr. Bara för att en drivrutin heter .PROFILE, behöver inte koderna styra en PROFILE 5MB hårddisk. Den kan lika väl styra en CORVUS hårddisk. Detta bör då anges i den förklarande raden.

Varje drivrutin har också en unik Device Type och Device Subtype. Device Type talar om vilken kategori en drivrutin tillhör och Device Subtype talar om vilket nummer denna drivrutin har i mängden likadana kategorier. Om man har ändrat om namnen på drivrutiner, eller stöter på en drivrutin med det allmänna namnet .PRINTER, går det att ta reda på vart den går genom att titta på det två olika typerna. Alla drivrutiner med (P) på slutet kan troligen heta .PRINTER. Följande typer är de nu förekommande:

Enhet	Type	Subtype
-------	------	---------

Teckenheter (bara skrivning till)

1. Seriell printer (.PRINTER)	\$41	\$01 (P)
2. Silentype printer (.SILENTYPE)	\$41	\$02 (P)
3. Parallell printer UPIC (.PRINTER)	\$41	\$03 (P)
4. Ljudgörare (.AUDIO)	\$43	\$01

Teckenheter (skriv och läsning)

5. Konsolen (.CONSOLE)	\$61	\$01
6. Grafisk skärm (.GRAFIX)	\$62	\$01
7. RS232 kommunikation (.RS232)	\$63	\$01

8. Parallellt kort UPIC (.PARALLEL) \$64 \$01

Blockenheter:

9. Disk /// (.D1 till .D4) \$E1 \$01
10 Profile hårddisk (.PROFILE) \$D1 \$02
11 Corvus hårddisk (.C1 till .C6) \$F1 \$01

Formatteringsrutiner:

12 Disk /// (.FMTD1 till .FMTD4) \$11 \$01

Den sista saken man kan sätta om en drivrutin, är om den är aktiv eller inaktiv. En drivrutin, som är aktiv, laddas in vid uppstarten av Apple /// från SOS.DRIVER. En drivrutin som är inaktiv, laddas EJ in vid uppstarten, ock kan således inte användas av det program, som man startar upp. Å andra sidan, tar den då heller inte upp någon plats i arbetsminnet på Apple ///. Ett exempel: I programmet Visicalc, kan man lida av akut minnesbrist. Genom att göra en drivrutin till en skrivare med hjälp av hjälpprogrammet SYSTEM UTILITIES inaktiv, och sedan starta om systemet på nytt, går det att få lite mera tillgängligt minne. Sedan när man är klar med sina beräkningar, kan man göra drivrutinen till skrivaren aktiv igen. Att en drivrutin är inaktiv, markeras i SYSTEM UTILITIES med en asterix (*) i slutet på drivrutinnamnet.

ÖVERSIKT ÖVER ALLA DRIVRUTINER.

Nedan kommer en sammanställning på alla drivrutiner som finns. Nummret hänсыftar på nummret i tabellen ovan. Ifall användaren har någon nytta av att ändra på Configuration Block, så tas det upp ungefär vad de gör. Sidhänvisningar till handböckerna där drivrutinerna behandlas tas upp. På vilken diskett, man kan hitta varje drivrutin och vilket filnamn den har. Varje lös drivrutin har alltså ett filnamn när den ligger lagrad löst på en diskett. Detta namn är ej att förväxla med namnet på drivrutinen. Med Read i SYSTEM UTILITIES, går det att läsa in en drivrutin i taget, för att anpassa systemet tills sina speciella behov.

Ex: Drivrutinen till bildskärmen hittas på disketten SYSTEM UTILITIES DATA med filnamnet CONSOLE.DRIVER. Detta namn anger man när man laddar in drivrutinen från SYSTEM UTILITIES. När man sedan tittar på själva drivrutinen, finner man att den har namnet .CONSOLE

-- NUMMER 5 --

NAMN :.CONSOLE
CB :Inga
DISKETT:SYSTEM UTILITIES DATA
FILNAMN:CONSOLE.DRIVER
MANUAL :SDD Sid 27-72

Den drivrutin har hand om all inmatning från tangentbordet och all utskrift till bildskärmen. Den måste finnas med i varje SOS.DRIVER. Denna drivrutin använder sig av två stycken tabeller. En som talar om var varje tecken ligger på tangentbordet och en som talar hur varje tecken ser ut på bildskärmen.

När man köper en amerikansk programvara, så måste det läggas in en svensk tangentbordslayout- och teckenuppsättningstabell som .CONSOLE får använda. Utföres inte detta, så får man Amerikanskt tangentbord (de tecken som man ser på kanten av vissa tangenter) och hakparenteser istället för ÅÄÖ. Det finns olika färdiga tabeller man kan lägga in i SOS.DRIVER och sedan använda. Vill man ibland hålla på med amerikanska tecken, får man se till att man startar upp med en diskett, som innehåller amerikanska tabeller. Vill man ha svenska, så är det bara att starta upp med en diskett, som innehåller en SOS.DRIVER med svenska tabeller.

-- NUMMER 2 --

NAMN :.SILENTYPE
CB :17 STYCKEN
DISKETT:SYSTEM UTILITIES DATA e1 SVENSKA DRIVRUTINER
FILNAMN:SMALL.DRIVER & BIG.DRIVER eller
SILENTYPE.LITEN & SILENTYPE.STOR
MANUAL :Silentypehandboken

Styr hårdvaran Silentype. Denna skrivare skall vara ansluten till port A i baksidan på Apple ///. Det finns i Sverige två olika varianter på denna drivrutin. En med amerikanska tecken och en med svenska dito. Den svenska känns igen på den kommentar, som den har. Den amerikanska varianten har ingen kommentar.

I CB kan man ändra höger och vänster marginal, hur "svart" silentypen skall skriva mm.

Det finns två stycken olika varianter av den svenskar och amerikanska varianten. SMALL.DRIVER är den oftast förekommande. Den innehåller bara den vanliga amerikanska teckenuppsättningen. BIG.DRIVER innehåller tre stycken teckenuppsättningar, som man kan välja mellan.

De svenska motsvarigheterna till de amerikanska drivrutinerna, är SILENTYPE.LITEN och SILENTYPE.STOR. Dessa drivrutiner hittas på disketten SVENSKA DRIVRUTINER FÖR APPLE ///. Har man en amerikansk drivrutin i SOS.DRIVER, så får man ta bort den och lägga in en svensk istället med hjälp av SYSTEM UTILITIES för att få svenska tecken.

-- Nummer 4 --

NAMN :.AUDIO
CB :Inga
DISKETT:SYSTEM UTILITIES DATA
FILNAMN:AUDIO.DRIVER

MANUAL :SDD Sid 129-133

Styr delvis högtalaren i Apple ///. Tar man bort denna drivrutin från en Visicalc /// diskett, så hör man inget när man vandrar in i en vägg.

-- NUMMER 6 --

NAMN :.GRAFIX
CB :Ett antal. Normal användare behöver ej bry sig
DISKETT:SYSTEM UTILITIES DATA
FILNAMN:GRAFIX.DRIVER
MANUAL :SDD Sid 73-98

Denna drivrutin måste vara med för att kunna få grafik på skärmen. Ett exempel på ett program, som använder denna drivrutin är grafikprogrammet BUSINESS GRAPHICS ///.

-- NUMMER 10 --

NAMN :.PROFILE
CB :1 st. Normal användare behöver ej bry sig
DISKETT:PROFILE DRIVER (med i profile-paketet)
FILNAMN:PROFILE.DRIVER
MANUAL :Profile handboken.

Denna måste naturligtvis vara med för att kunna hantera en 5MB PROFILE hårddisk. CB

-- NUMMER 11 --

NAMN :.C1 till .C6
CB :En eller Ingen
DISKETT:CORSOS volym 1
FILNAMN:LOGON.DRIVER och CORVUS.DRIVER
MANUAL :CORVUSHANDBOKEN

Denna drivrutin går till en Corvus hårddisk. Det finns två olika drivrutiner till CORVUS. CORVUS.DRIVER har ingen CB. När man startar upp med denna drivrutin, så måste det finnas ett speciellt uppstartningsprogram, som låter användaren ange sitt namn och kodord.

I LOGON.DRIVER, har man möjlighet att på drivrutinen .C1 ändra CB till den användare, som man automatiskt skall loggas in som.

Det finns en möjlighet att koppla ihop ett antal liknande drivrutiner till en enda. Detta är fallet med de båda corvus-drivrutinerna. Man kan komma åt 6 stycken olika blockenheter via de sex drivrutinerna. Det går inte att ta bort bara en av drivrutinerna. Antingen har man alla sex där, eller så har man inga.

-- NUMMER 12 --

NAMN :.FMTD1 till .FMTD4
CB :Inga.
DISKETT:SYSTEM UTILITIES DATA
FILNAMN:.FMTDX.DRIVER
MANUAL :?

Formatteringsdrivrutiner för de fyra diskettstationerna. Används av SYSTEM UTILITIES för att formatera disketter. Denna drivrutin måste alltså finnas på vare SYSTEM UTILITIES, som man vill använda för att formatera disketter.

KOMMUNIKATION

Och nu kommer det lite svårare! Alla de drivrutiner, som kan kommunicera med olika skrivare och andra datorer.

Vi tar förs de två drivrutinerna, som använder det parallella kortet UPIC. (står för Universal Parallel Interface Card), som man kan sätta i någon av de fyra fria kortkontakterna. Parallell överföring betyder att bitarna åker parallellt i en sladd med ett antal trådar i.

-- NUMMER 3 --

NAMN :.PRINTER
CB :5 STYCKEN
DISKETT:UPIC DRIVER SOFTWARE
FILNAMN:PRINTER.DRIVER
MANUAL :UPIC HANDBOKEN

Användes för att kommunicera bara ut från kortet till en skrivare som har en parallell ingång. Till alla parallella skrivare skall man använda denna drivrutin. Sladden består av 20 stycken pinnar.

Det fodras också att man har en korrekt sladd, så att alla signaler går dit som ska. På sidan 8 i handboken till kortet finns en tabell över hur de vanligaste skrivarnas sladdar skall kopplas. Det finns färdiga sladdar för de flesta behov.

I UPIC handboken på sidan 19 hittas en tabell på hur CB skall ställas in för att det skall fungera med olika skrivare. T ex Epson MX80 och MX100 skall ha följande data:

E8 C8 00 00 0A

På sidan 19 till 22 förklaras exakt vad dessa parametrar styr. Bland annat om AutoLF skall sändas efter CR och vilka fel, som kan uppträda i skrivaren.

-- NUMMER 8 --

NAMN :.PARALLEL
CB :ETT ANTAL STYCKEN
DISKETT:UPIC DRIVER SOFTWARE
FILNAMN:PARALLEL.DRIVER
MANUAL :UPIC HANDBOKEN SID 23 OCH FRAMMÅT

Denna drivrutin används för att upprätta en parallell förbindelse med en parallell enhet. Den sladd som behövs är på 40 pinnar. Detta är inte så ofta förekommande. Denna drivrutin är till för att införliva en tvåvägskommunikation mellan Apple /// och en parallell enhet.

TILLBEHÖREN

Allt för er dator (stor som liten) och ord/textbehandling



REFLEXSKYDD
Tar bort alla reflexer. Passar de flesta bildskärmar.

SKRIVHJUL
80 olika hjul i lager. Stort sortiment svenska tecken.

VERBATIM DISKETTER OCH KASSETTER
Samtliga 8" och minidisketter på lager. I transport- och förvaringsvänlig plastbox. Alla 5 1/4" med förstärkningsring.



FÄRGBAND
TBS färgbandskassetter för Diablo, Qume, IBM, Facit, Tally, Centronic, PET, ABC 80 m fl.

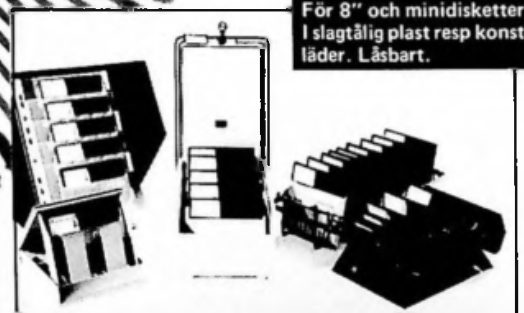


KONCEPTHÅLLARE
Med belysning och elektriskt höj- och sänkbar linjal. (Fotpedal)



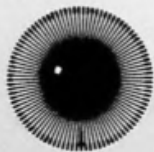
Vår BHJ-policy:
Kvalitet/rätt pris/ snabba leveranser.

DISKETFÖRVARING
För 8" och minidisketter. I slagtålig plast resp konstläder. Läsbart.



Övriga produkter i vårt program är terminalbord, skåp för datamedia, antistatduk, rengöringsdisketter m m. Du finner dessa i kontors- och datorbutiker.

Vårt produktprogram söker ytterligare återförsäljare



BHJ-DATA
Datautrustningar ab

Besök gärna våra permanenta utställningar i Stockholm och Göteborg.

Kammargatan 62,
111 24 Stockholm.
Tel 08/24 94 80.

Johannesbergsgatan 32 B,
412 55 Göteborg.
Tel 031-20 10 11.

Telex
BHJ-Data
14167.

Sänd mig broschyr på:

- Hela ert sortiment
 Verbatim
 Skrivhjul

- Diskettförvaring
 Koncepthållare
 Färgband
 Reflexskydd

Kontakta mig för besök

Sänd mig uppgift om närmaste återförsäljare

Namn

Företag

Adress

Postadr Tel.

Sedan finns det också en möjlighet till en seriell kommunikation till seriella skrivare. Till detta används den inbyggda seriella porten på baksidan av Apple ///. Den har namnet C. Apple /// följer helt RS232-standarden på hur tåtarna skall dras. Det kan oftast fodras en korskopplare om man skall komunicera till en seriell skrivare.

Skall det upprättas en kommunikation via ett modem till en annan stordator, Apple /// eller dylikt behövs ingen korskopplare.

PROTOKOLL

Det finns kan man säga två stycken grundläggande protokoll. Det finns asynkon och synkron kommunikation. Asynkron kommunikation används när man vill sända ett tecken i taget. Då en skrivare förväntar sig en textmassa är det så gott som alltid asynkront. Detta sätt att kommunicera är det som alla de seriella drivrutiner jag tar upp här använder.

Synkron kommunikation är något som mest IBM stordatorer använder. Där finns det en uppsjö av olika sätt att kommunicera. 3270, 3780, 2780 och 3276 är några exempel på synkrona protokoll. Gäller det något av dessa, så fungerar det inte med någon vanlig drivrutin, utan det behövs speciell programvara.

Ett annat sätt att lösa detta är att placera en burk utanför, som konverterar den synkrona kommunikationen till asynkron kommunikation, varvid då Apple /// kan använda sig av ett vanligt kommunikationsprogram (varför inte Access ///) för att det skall fungera.

Protokoll använder man för att det inte skall inträffa att till exempel Apple /// sänder en massa tecken till en skrivare, som inte kan ta emot dem. Om då inte Apple /// skulle sluta sända tecken när skrivaren inte kan ta emot något mera, så skulle ju all den sända informationen gå förlorad.

Det första protokollet kallas hårdvaruhandskakning. Detta betyder att en speciell pinne på tråden mellan skrivaren och Apple /// användes för att tala om om skrivaren är klar att ta emot data eller inte. Apple /// sänder då bara data då den ser att skrivaren är klar.

Det andra protokollet kallas XON/XOFF protokoll. En Apple /// sänder en massa tecken till en skrivare. Den kan inte ta emot något mera varvid den sänder ett speciellt uppdefinierat tecken (XOFF, som är en speciell ASCII-kod) till Apple ///. Trean slutar då genast att sända tecken och ställer sig istället i ett vänteläge på att ett XON tecken skall sändas av skrivaren. Apple /// börjar då att sända igen, tills det att den återigen tar emot ett XOFF osv.

Detta är alltså en tvåvägskommunikation mellan

skrivaren och Apple.

Slutligen finns det också ett ETX/ACK protokoll. En Apple sänder ett ETX till skrivaren om den kan ta emot ett antal tecken i en följd och väntar tills den svarar ja med ett ACK. Apple /// sänder då över sitt block data.

Alla dessa XOFF, ACK mm är tecken är oftast ASCII-koder från 0 till 31.

— NUMMER 1 —

```
NAMN  :.PRINTER
CB    :5 STYCKEN
DISKETT:SYSTEM UTILITIES DATA
FILNAMN:QUME.DRIVER
MANUAL :SDD Sid 99-105
```

Denna drivrutin skall användas till skrivare som använder hårdvaruhandskakning. I CB kan man ändra kommunikationshastigheten (BAUD-rate) och pariteten. SDD sid 103 beskriver vad man skall ändra för att uppnå olika saker. Det gäller att se till att både skrivaren och drivrutinen är inställda på samma värden. En vanlig rak sladd skall användas och en korskopplare. Oftast går det att ställa om skrivarna för olika protokoll. Det är att föredra om det går att ställa om skrivaren till hårdvaruhandskakning och använda denna drivrutin. Den är nämligen den mindre om man jämför med drivrutinen .RS232

— NUMMER 7 —

```
NAMN  :.RS232
CB    :5 STYCKEN
DISKETT:SYSTEM UTILITIES DATA
FILNAMN:RS232.DRIVER
MANUAL :SDD sid 107 - 133
```

Denna drivrutin supportar alla de tre protokoll, som jag beskrev ovan. Genom att ändra på CB går det att ställa om den för olika protokoll. På sidan 144 i SDD hittar man beskrivningar på hur man ställer in CB för att definiera vilka tecken, som används för ACK mm samt vilket protokoll man använder.

I stället för att använda denna drivrutins möjlighet till hårdvaruhandskakning, bör man använda föregående drivrutin för denna sak.

Vad består SOS.DRIVER av ?

SOS.DRIVER består av två delar. En del är alla olika drivrutiner, som är inlagda där, den andra delen är System Parameters. Där finns följande information:

- * Hur många diskettenheter som används
- * Till vilken drivrutin, som de kort, som sitter i maskinen hör.
- * En tabell med hur tecken ser ut på bildskärmen.
- * En tabell med var tecknen ligger på tangentbordet.

INLÄGGNING AV SVENSKA DRIVRUTINER

Om en användare skall göra om en amerikansk SOS.DRIVER till svenska, skall han utföra följande:

1. Starta upp programmet SYSTEM UTILITIES
2. Välja funktionen 'S' :
systemkonfigurationsprogrammet
3. Först läsa in SOS.DRIVER från den diskett han tänker ändra på med hjälp av programmet SYSTEM UTILITIES till Apple /// interminne.
4. Ta i minnet bort de drivrutiner han inte behöver.
5. Läsa in i minnet de nya drivrutiner han behöver. Dessa hittas antingen på disketten SYSTEM UTILITEIS DATA eller på den diskett, som följer med den aktuella hårdvaran.
5. Editera i minnet de drivrutiner, som behövs editera. Till exempel ändra på kommunikationshastigheten på en drivrutin till en skrivare mm.
6. Editera systemparametrarna i minnet och lägga in en svensk teckenuppsättning och tangentbordslayout.
7. Generera tillbaka detta till den diskett, som SOS.DRIVER lästes in från vid steg nummer 3

DET FINNS TVÅ OLIKA TYPER AV DRIVRUTINER

Det finns två stycken olika typer av drivrutiner. En som hanterar Character Devices och en annan typ som hanterar Block Devices. Med detta förstås att Character Devices (tecken enheter) är enheter, som hanterar information på teckennivå, eller BYTE-nivå vilket kan vara det samma. Ett exempel är alla drivrutiner, som hanterar skrivare. Till dessa sänder man ett tecken i taget. Har man en gång sänt ett tecken, går det inte att komma åt det igen; såvitt inte den enhet, som man sänt tecknet till sänder tillbaka det.

Block Devices (block enheter) är motsatsen. Dessa hanterar informationen i block. Ett block data består av ett antal byte. Det som oftast (hitills alltid) förekommer är att ett block består av 512 tecken. En blockenhet består av ett media, som har möjlighet att lagra informationen på denna nivå. Varje plats för ett block med information, har då ett nummer. Detta nummer används för att tala om var det block data, som man sänder skall komma. När sedan data behöver hämtas, så är det bara att ange vilket blocknummer datat skall hämtas ifrån. Diskettenheter och hårddiskar är exempel på sådana enheter.

En underavdelning av blockenheterna är

formatteringsdrivrutinerna. Dessa används för att formatera disketter mm. Exempel på sådana rutiner är .FMTO1 till .FMTO4, som hittas i SOS.DRIVER på SYSTEM UTILITIES disketten och används vid formatering av disketter, som sitter i någon diskettenhet.

Vad består en drivrutin av?

Om man tittar lite närmare på vad en drivrutin består av, så är det följande delar:

Den första delen kallas Device Information Block och innehåller

```
!-----!  
! PEKARE TILL BÖRJAN AV KODEN, SOM STYR DRIVRUTINEN !  
!-----!  
! INFO OM NAMNET, VERSION, TYP AV DRIVRUTIN !  
!-----!  
! CONFIGURATION BLOCK DATA !  
!-----!
```

Sedan följer:

```
!-----!  
! COPYRIGHTTEXT !  
!-----!  
! PRIVATA VARIABLER FÖR DRIVRUTINEN !  
!-----!  
! EN EVENTUELL AVBROTTSCHANTERE !  
!-----!
```

Dessa två bitar är ungefär likadana för en drivrutin, som går till en block eller teckenenhet. Sedan kommer den del av själva drivrutinen som består av en massa maskinkoder som gör att drivrutinen betar sig korrekt.

En drivrutin, kan utföra ett begränsat antal kommandon. dessa är för en teckenenhet följande:

INIT: Anropas vid uppstarten av systemet, för att iordningställa drivrutinen för användande.

OPEN: Open talar om att drivrutinen behöver använda en viss hårdvara till exempel anslutningskortet i kortkontakt nummer 2. Inga andra drivrutiner får då använda denna hårdvara.

READ: Läser ett specificerat antal tecken (eller byte, som är samma sak) från teckenenheten till Apple /// interminne.

WRITE: Skriver ett specificerat antal tecken.

CLOSE: Avslutar användandet av den hårdvara, som drivrutinen använder, och går det möjligt för andra enheter att använda den.

STATUS: Hämtar in olika information om olika saker från drivrutinen.

Naturligt nog kommer vi denna gång främst att titta på böcker om VisiSerien. Det betyder i detta fall böcker om VisiCalc så gott som uteslutande. Listan är ett utdrag ur Bob Broedels boklista, och den återges efter medgivande av utgivaren av Nibble Magazine, referenstidningen för Applevärlden. Listan kommer att återges i sin helhet i ett senare nummer.

Böcker med ***-märkning har vi tittat närmare på här bredvid.

>> Executive VisiCalc
Addison Wesley publishing, Reading MA 01867
Tillgänglig februari 1983, pris okänt.

>> An Introduction to VisiCalc
Matrixing for Apple & IBM
McGraw-Hill/College
1221 Ave of the Americas, N. Y., NY
252 sidor

>> The Power of VisiCalc (vol 1 och 2) ***
Management Information Systems
3543 N.E. Broadway, Portland OR 97232
88/97 sidor

>> The Power of Visiplot/ Visicalc/ VisiFile
Management Information Systems, se ovan.
154 sidor

>> The Power of VisiCalc real-estate
Management Information Systems, se ovan
166 sidor

>> The VisiCalc Book, Apple edition
Reston/Prentice-Hall Publishing
320 Hudson terrace; Englewood cliffs,
NJ 07632, 400 sidor.

>> VisiCalc: Home & Office companion ***
Osborne/McGraw-Hill
630 Bancroft Way, Berkeley, CA 94710
181 sidor. även gm TRIM (nedan)

>> VisiCalc Applikations
Sybex, gm TRIM Publications
08/540010
213:-, utkommer under december.

THE POWER OF VISICALC

Volym 1 får sägas vara en ren introduktion till VisiCalc, man börjar från "scratch" och lär steg för steg ut konsten att bygga upp kraftfulla modeller.

Volym 2 är då betydligt intressantare, här tar man upp ett antal modeller, och demonstrerar härigenom olika tekniker och finesser. De olika modellerna kan naturligtvis i många fall appliceras direkt, eller med smärre justeringar.

Böckerna från Management Information Source kan även köpas från Apple Pudgetsound Program Library Exchange, A.P.P.L.E. Någon svensk distributör har vi inte kunnat finna.

VISICALC - HOME & OFFICE COMPANION

Färdiga modeller i massor! Mer än 50 stycken modeller av olika slag. Varje modell förklaras grundligt. Inmatning av modellerna sker enkelt då en komplett listning beskriver exakt hur det skall ske. Boken kan tack vare detta användas av såväl nybörjare som mera erfarna VisiCalc användare.

Det finns modeller inom bla följande kategorier: Lån & investeringar, generella affärer, lager kontroll/styrning, annonsering och försäljning, personal administration och avdelningar, personlig ekonomi.

Boken rekommenderas, vi fann den mycket givande.

ANNAT LÄSVÄRT

TRIM Publications presenterar följande nya böcker för Apple. Apple Connection - en ny bok om interfacing (gränssnitt eller.... på svenska), I/O, D/A - A/D omvandling och lite annat. Den kommer att kosta 228:- när den finns tillgänglig i december/ januari. Celestial Basic kan man också få tag på från TRIM i december. Boken handlar om hur man "anlägger" ett planetarium i sin Apple. Pris 238:-

LOGO - det nya utbildningsspråket

En del av nyheterna i Bobs boklista utgörs av böcker om LOGO. För dem av Er som är intresserade av utbildningsprogrammering eller bara vill hänga med kommer här en lista över dessa böcker.

>> Apple Logo
Byte/McGraw-Hill
70 Main Street, Peterborough
NH 03458, 240 sidor
(Boken finns i flera versioner)

>> Learning with Logo
som ovan
230 sidor.

OBS!! Har Du läst några intressanta böcker (som Paul Schlyter i detta nummer) skriv! vi är tacksamma för alla bidrag.

en microdator.

Exempel:

Apple II 48k
12" monitor
Diskdrive 140kB
Programvaror
Visicalc,
Visitrend/Visiplot
och Visifile

925 kr/månad



Exempel:

Apple II 112k
12" monitor
Diskdrive 140kB
Epson MX-80 printer
Programvaror Visicalc,
Visitrend/Visiplot och Visifile

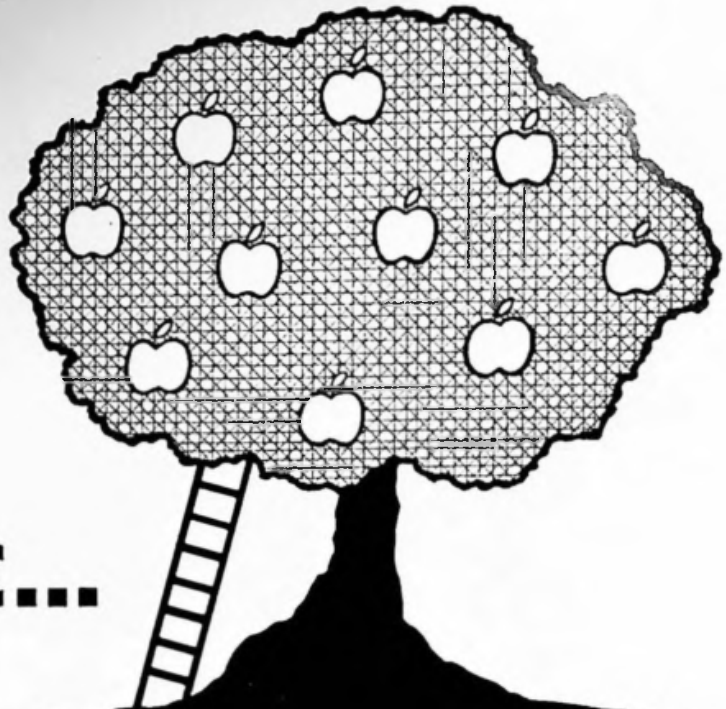
1.600 kr/månad

Samtliga priser exkl. moms. Mastercharge, Köpkort

MICROTECH

ROSENLUNDSGATAN 36 · BOX 17107 · S-10462 STOCKHOLM · TEL 08/690135, ORDER 847375 · TELEX 13188

Nytt
Nummer!



Take a bite...

...of Apple Orchard!

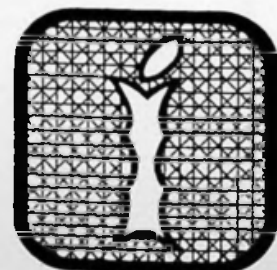
Vi har fått hem ytterligare ett nytt nummer: vol 3 nr 2. Vi har en bra lagersituation så det är bara att beställa! Sätt som vanligt in beloppet på vårt bg.551-5648 och ange vad du önskar.

Volym 1 nr 3

2 nr 1, 2, 3 & 4

3 nr 1, 2

25:-



Applepylär

Disketter
26:-
vid 10st



Äbbisk

svart bakgrund
med logo i sex
färger.

749 X 481 mm

18:-



Slips

blå, vinröd eller brun

64:-



Hals-
smycke

34:-



inklusive kedja. Logo i
sex färger.

Bg 551-5648

Slipsnål 24.50
Mugg 49.50
T-shirt 63.00
Penna (Cross) 154.50

Allt detta kan köpas från
oss. Sätt in pengar på vårt
bankgiro och ange vad du
önskar.

Nya Monitorkommandon

av Paul Schlyter

Monitorn till Apple II finns i två olika versioner. Den version som de flesta av dagens Apple-ägare har kallas autostart-monitorn. Det är den som automatiskt bootar upp en diskett när strömmen slås på. Om du har en mycket gammal Apple II, eller om du har ett Integer-kort, finns där en annan typ av monitor, den s.k. gamla monitorn. Den saknar många av autostartmonitors egenskaper, men den har två kommandon som autostartmonitorn saknar: STEP och TRACE, med vars hjälp man kan avlusa assembler-program. Vore det inte trevligt om man kunde ha autostart-monitorn OCH STEP/TRACE? Jovisst, men det finns ju inte plats. Eller..?

Monitorn har två andra kommandon, READ och WRITE, som läser och skriver på kassettband. Hur länge sen var det du använde kassettband sist? Själv använder jag dem nästan aldrig, och jag tror samma sak gäller de allra flesta som har disk-drivar till sitt Äpple. Tar vi bort dessa kommandon, får vi så mycket ledig plats att det precis räcker för att stoppa in STEP och TRACE i autostart-monitorn. Lista 1 är en assemblerlista som visar hur man gör det hela. Om du inte har en assembler, eller inte vill använda den, kan du istället ta hex-dumparna efter assemblerlistan och knappa in det hela direkt i hex.

Du kanske inte är intresserad av STEP och TRACE, men ändå skulle tycka det vore roligt att bättra på monitorn lite. I så fall kan du ta en titt på Lista 2, som är ett annat förslag på hur man kan bättra på sin monitor. Med dessa ändringar får du ett förbättrat L-kommando (disassembler-kommando). Du kan som förut skriva t.ex. E00L, och få 20 rader. Men du kan också ange ett område som du vill disassemblera, t.ex. E000.E015L. Skulle du vilja disassemblera t.ex. hela DCS, gör du det med 9D00.BFFFL (48K DCS). Hur gjorde man tidigare? 9D00LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL .. ja hur många L skulle man ta? Det var inte lätt att veta.



En annan förbättring med ändringarna i Lista 2 är att den vanliga hex-dumpen nu även ger en ASCII-dump bredvid. Slutligen finns ett helt nytt kommando, en ren ASCII-dump. Om du vill se om ett område i minnet innehåller ASCII-text, gör du enkelt t.ex. A880.ABOOP. Kommandot heter m.a.o. "P".



För att få plats med detta är även i Lista 2 READ och WRITE borttagna. Fig 3 visar exempel på de nya och förbättrade monitor-kommandon man får med ändringarna i Lista 2.

Nu undrar säkert många av er hur man ändrar i monitorn, den ligger ju i ROM! Om du har ett RAM-kort är det hela lätt ordnat, flytta över monitorn till RAM-kortet med monitor-kommandot:

```
C081 C081 D000<D000.FFFFFM C083 C083
```

Detta flyttar inte bara över monitorn, utan även det språk du har i ROM på ditt Äpple. Om du vill flytta bara monitorn, skriver du förstås F800 istället för D000. Efter detta har du monitorn i RAM och kan ändra i den hur du vill. Se bara till att du gör alla ändringar klart innan du provar de nya kommandona.

Men om du inte har något RAM-kort? Du kan fortfarande ändra i monitorn, om du har en ROM-brännare för 2716 EPROMar. Flytta ner monitorn i vanliga RAM-minnet, gör ändringarna där, och bränn sedan ett EPROM med den ändrade monitorn. Men hur ansluter man ett EPROM till Apple? Om du har ett Integer-kort, finns det fyra olika små ställen man kan ändra på, alla är märkta "2716". Antingen ser de ut så här: , då ska du bryta kontakten med t.ex. en liten skruvmejsel. Eller också ser de ut så här: , då ska du löda över mellanrummet så det blir kontakt. Nu kan du byta ut monitorn på Integer-kortet mot ett 2716 EPROM. MEN du måste också byta alla andra ROM-ar mot 2716 på kortet - se därför till att du bränner kopior av dessa också. Samt kanske också en kopia av det gamla monitor-ROMet, ifall du nån gång skulle vilja läsa ett kassettband från Integer Basic.

Om du har ett Apple Language Card, finns där ett monitor-ROM. Det kan också bytas ut mot ett 2716, om du ändrar två punkter på kortet. Båda punkterna är märkta "2716". Den ena finns nertill på mitten av kortet och ser ut så här:  Bryt den förbindelsen med en liten skruvmejsel. Den andra punkten finns upptill strax ovanför monitor-ROMet och ser ut så här:  Löd så att det blir kontakt här. Nu kan du stoppa in ett 2716 i ditt Apple Language Card. Om du har ett annat RAM-kort, kan du tyvärr inte göra detta.

Om du varken har ett RAM-kort eller ett Integer-kort, finns det ändå en utväg. Bränn ditt ROM som vanligt. Men innan du byter ut det mot monitorn på moderkortet, måste du böja upp två pinnar, pinne 18 och pinne 21. Ta sedan två korta stumpar tunn koppartråd och koppla ihop den uppböjda pinne 21 med pinne 24, och den uppböjda pinne 18 med pinne 12. Efter detta kan du stoppa i ditt 2716 direkt på moderkortet. MEN (och detta är viktigt) du kan nu inte använda någon form av RAM-kort, och inte ens ett Integer-kort. Den 2716 som nu sitter på moderkortet kan nämligen aldrig kopplas bort, och kopplar du nu in ett RAM- eller Integer-kort, kommer det att finnas två ROMar (eller ROM + RAM) på samma adress samtidigt. I bästa fall hänger sig Äpplet, i värsta fall bränner du sönder några kretsar. Så om du på detta sätt ansluter ett 2716 direkt på moder-

kortet, måste du vara försiktig i fortsättningen.

Även Applesoft har flera kassetband-kommandon: LOAD, SAVE, STORE, RECALL, SHLOAD. Om man tar bort alla dessa kommandon, får man dels ledigt utrymme, dels fem lediga nyckelord (tokens), och man kan med dessa skapa nya Applesoft-kommandon. Vad sägs t.ex. om

IF ... THEN ... ELSE , RESTORE linjenr , RESUME linjenr ?

Att göra dessa ändringar är förstås också lättast om man har ett RAM-kort, men om man vill kan man även här bränna in det i EPROM.

Mer om detta i ett kommande nummer av Apple User News.

*AB80.A90F

```

AB80- 9D 85 37 60 49 4E 49 D4  .7€INIT
AB88- 4C 4F 41 C4 53 41 56 C5  LOADSAVE
AB90- 52 55 CE 43 48 41 49 CE  RUNCHAIN
AB98- 44 45 4C 45 54 C5 4C 4F  DELETELO
ABA0- 43 CB 55 4E 4C 4F 43 CB  CKUNLOCK
ABAB- 43 4C 4F 53 C5 52 45 41  CLOSEREA
ABB0- C4 45 58 45 C3 57 52 49  DEXECWRI
ABB8- 54 C5 50 4F 53 49 54 49  TEPOSITI
ABC0- 4F CE 4F 50 45 CE 41 50  ONOPENAP
ABC8- 50 45 4E C4 52 45 4E 41  PENDRENA
ABD0- 4D C5 43 41 54 41 4C 4F  MECATALO
ABD8- C7 4D 4F CE 4E 4F 4D 4F  GMONNOMO
ABE0- CE 50 52 A3 49 4E A3 4D  NPR#IN#M
ABE8- 41 58 46 49 4C 45 D3 46  AXFILESF
ABF0- D0 49 4E D4 42 53 41 56  PINTBSAV
ABF8- C5 42 4C 4F 41 C4 42 52  EBLOADBR
A900- 55 CE 56 45 52 49 46 D9  UNVERIFY
A908- 00 21 70 A0 70 A1 70 A0  .!p p!p
    
```

*E000L

```

E000- 4C 28 F1      JMP      ØF128
E003- 4C 3C D4      JMP      ØD43C
E006- 00              BRK
E007- 20 B1 00        JSR      Ø00B1
E00A- 90 05           BCC      ØE011
E00C- 20 7D E0        JSR      ØE07D
E00F- 90 0B           BCC      ØE01C
E011- AA              TAX
E012- 20 B1 00        JSR      Ø00B1
E015- 90 FB           BCC      ØE012
E017- 20 7D E0        JSR      ØE07D
E01A- B0 F6           BCS      ØE012
E01C- C9 24           CMP      #Ø24
E01E- D0 06           BNE      ØE026
E020- A9 FF           LDA      #ØFF
E022- 85 11           STA      Ø11
E024- D0 10           BNE      ØE036
E026- C9 25           CMP      #Ø25
E028- D0 13           BNE      ØE03D
E02A- A5 14           LDA      Ø14
    
```

*AB80.AB00P

```

AB80- .7€INITLOADSAVE RUNCHAINDELETELO
ABA0- CKUNLOCKCLOSEREA DEXECWRITEPOSITI
ABC0- ONOPENAPPENDRENA MECATALOGMONNOMO
ABE0- NPR#IN#MAXFILESF PINTBSAVEBLOADBR
A900- UNVERIFY.!P P!P P P P P PÉ." T"
A920- ."#X"POPÉPÉ.É...ÉPÉ.!Y Q Q
A940- PVDSLRBACIOÉ ...É ...Ü.....
A960- .....LANGUAGE NOT
A980- AVAILABLERANGE ERRORWRITE PROTE
A9A0- CTEDEND OF DATAF ILE NOT FOUNDVOL
A9C0- UME MISMATCHI/O ERRORDISK FULLFI
A9E0- LE LOCKEDSYNTAX ERRORNO BUFFERS
AA00- AVAILABLEFILE TY PE MISMATCHPROGR
AA20- AM TOO LARGENOT DIRECT COMMAND..
AA40- ...Ø3>LADMX.*;- ...FA.A.V.(....
AA60- .....ATCHES.MNT
AAB0- R1
AAA0- .....É...PLESOF
AAC0- TH7;3;4.ÉÜ3!+.,W ,D,*-. -N,U,9,-.
AAE0- ..-Ü3Ü3.,.,.,.,Ü 3Ü3=,I,.,F,Ü3É.P
AB00- .
    
```

*E000.E015L

```

E000- 4C 28 F1      JMP      ØF128
E003- 4C 3C D4      JMP      ØD43C
E006- 00              BRK
E007- 20 B1 00        JSR      Ø00B1
E00A- 90 05           BCC      ØE011
E00C- 20 7D E0        JSR      ØE07D
E00F- 90 0B           BCC      ØE01C
E011- AA              TAX
E012- 20 B1 00        JSR      Ø00B1
E015- 90 FB           BCC      ØE012
    
```

Fig.3

Lista 1
ASM

```

1010 *****
1020 #
1030 # ANDRINGAR I AUTOSTARTMONITORN #
1040 #
1050 # ATERSTALL S(TEP & T(RACE #
1060 # FRAN GAMLA MONITORN #
1070 #
1080 *****
1090 |-----|
002C: 1100 RTN EDU #2C,2D
002F: 1110 LENGTH EDU #2F
003A: 1120 YSAV EDU #3A
003A: 1130 PC EDU #3A,3B
003C: 1140 XQT EDU #3C
004B: 1150 STATUS EDU #4B
1160
FB00: 1170 INSTDSP EDU #FB00
F954: 1180 PCADJ2 EDU #F954
F956: 1190 PCADJ3 EDU #F956
FA59: 1200 OLDBRK EDU #FA59
FAD7: 1210 REGDSP EDU #FAD7
FE75: 1220 AIPC EDU #FE75
FF3F: 1230 RESTORE EDU #FF3F
FF4A: 1240 SAVE EDU #FF4A
1250
1260 |-----|
1270 | NOPROD
1280 |-----|
1290 | TA BORT R(LEAD & W(RITE
1300 | ORG #FFDC TA BORT WRITE
1310 | HEX B2
1320 | ORG #FFF3
1330 | HEX AC
1340 | ORG #FFDE TA BORT READ
1350 | HEX B2
1360 | ORG #FFF5
1370 | HEX AC
1380
1390 | ATERSTALL S(TEP & T(RACE
1400 | FRAN GAMLA MONITORN
1410 |-----|
1420 | ORG #FFD2 ATERSTALL STEP
1430 | HEX EC
1440 | ORG #FFE9
1450 | HEX C3
1460 | ORG #FFCF ATERSTALL TRACE
1470 | HEX ED
1480 | ORG #FFE6
1490 | HEX C1
1500
1510 | STEP- DCH TRACE-RUTINERNA AR IDENTISKT
1520 | LIKA GAMLA MONITORNS RUTINER, FAST DE
1530 | AR EN SMULA OHFLYTVADE.
1540 | ORG #FEC2
1550 TRACE DEC YSAV
1560 STEPZ JSR AIPC
1570
1570 JMP STEP
1580
1590 ORG #FECD
1600 STEP JSR INSTDSP
1610 PLA
1620 STA RTN
1630 PLA
1640 STA RTN+1
1650 LDX #B
1660 X0INIT LDA INITBL-1,X
1670 STA XQT,X
1680 DEX
1690 BNE X0INIT
1700 LDA (PC,X)
1710 BEQ XBRK
1720 LDY LENGTH
1730 CHP #R20
1740 BNE .1
1750 JMP XJSR
1760 .1 CHP #R60
1770 BEQ XRTS
1780 CHP #R4C
1790 JMP STEP2
1800 ORG #FFFD
1810 STEP2
1820 BNE .1
1830 JMP XJMP
1840 .1 CHP #R6C
1850 BNE .2
1860 JMP XJMPAT
1870 .2 CHP #R40
1880 BEQ XRTI
1890 AND #R1F
1900 EOR #R14
1910 CHP #R04
1920 BEQ XQ2
1930 XQ1 LDA (PC),Y
1940 XQ2 STA XQT,Y
1950 DEY
1960 BPL XQ1
1970 JSR RESTORE
1980 JMP XQT
1990 XBRK CLP OLDBRK
2000 XRTI JMC
2010 PLA
2020 STA STATUS
2030 XRTS
2040 JMP STEP3
2050
2060 ORG #FCC9
2070 STEP3
2080 PLA
2090 STA PC
2100 PLA
2110 PCINC2 STA PC+1
2120 PCINC3 LDA LENGTH
2130 F9 PCADJ3 JSR PCADJ3
2140 STY PC+1
2150 CLC
2160 BCC NEWPCL

```

```

FCD9: 18 2170 XJSR CLC
FCDA: 20 54 F9 2180 JSR PCADJ2
FCDD: AA 2190 TAX
FCDE: 98 2200 TYA
FCDF: 4B 2210 PHA
FCE0: BA 2220 TXA
FCE1: 4B 2230 PHA
FCE2: A0 02 2240 LDY #2
FCE4: 18 2250 XJMP CLC
FCE5: B1 3A 2260 XJMPAT LDA (PC),Y
FCE7: AA 2270 TAX
FCEB: 88 2280 DEY
FCE9: B1 3A 2290 LDA (PC),Y
FCEB: B6 3B 2300 STX PC+1
FCED: B5 3A 2310 NEWPCL STA PC
FCEF: B0 F3 2320 BCS XJMP
FCF1: A5 2D 2330 RTNJMP LDA RTN+1
FCF3: 4B 2340 PHA
FCF4: A5 2C 2350 LDA RTN
FCF6: 4B 2360 PHA
FCF7: 4C D7 FA 2370 JMP REGDSP
FCFA: 38 2380 BRNCH2 SEC
FCFB: B0 D0 2390 BCS PCINC2
FCFD: 20 4A FF 2400 NBRNCH JSR SAVE
FD00: 38 2410 SEC
FD01: B0 CC 2420 BCS PCINC3
FD03: EA 2430 INITBL NOP
FD04: EA 2440 NOP
FD05: 4C FD FC 2450 JMP NBRNCH
FD0B: 4C B3 FB 2460 JMP BRNCH
2470
2480 ORG #FBB3
FBB3: 18 2490 BRNCH CLC
FBB4: A0 01 2500 LDY #1
FBB6: B1 3A 2510 LDA (PC),Y
FBB8: 20 56 F9 2520 JSR PCADJ3
FBBB: B5 3A 2530 STA PC
FBBD: 98 2540 TYA
FBBE: 4C FA FC 2550 JMP BRNCH2
2560 |-----|

```

INGA FEL I ASSEMBLERINGEN

Lista 2
ASM

```

1010 *****
1020 #
1030 # ANDRINGAR I AUTOSTARTMONITORN #
1040 #
1050 # UTBYGGT L-KOMMANDO: #
1060 # XXXXL DISASH 20 LINES #
1070 # XXXX.YYYYYL DISASH RANGE #
1080 #
1090 # UTBYGGD HEX-DUMP: #
1100 # GER AVEN ASCII-DUMP #
1110 #
1120 # REN ASCII-DUMP: #
1130 # XXXX.YYYYYP #
1140 #
1150 *****
1160 |-----|
1170 | MACRO MOVDP
1180 | LDA ?1
1190 | STA ?2
1200 | LDA ?1+1
1210 | STA ?2+1
1220 | ENDM
1230
1240 | MACRO CMPDP
1250 | LDA ?1
1260 | CMP ?2
1270 | LDA ?1+1
1280 | SBC ?2+1
1290 | ENDM
1300 |-----|
003A: 1310 PC EDU #3A,3B
003C: 1320 A1 EDU #3C,3D
003E: 1330 A2 EDU #3E,3F
0040: 1340 A3 EDU #40,41
1350
FB00: 1360 INSTDSP EDU #FB00
F94A: 1370 PRBL2 EDU #F94A
F953: 1380 PCADJ EDU #F953
FCBA: 1390 NXTA1 EDU #FCBA
FDBE: 1400 CROUT EDU #FDBE
FD92: 1410 PRA1 EDU #FD92
FDDA: 1420 PRBYTE EDU #FDDA
FDED: 1430 COUT EDU #FDED
FE75: 1440 AIPC EDU #FE75
1450
1460 |-----|
1470 | NOPROD
1480 |-----|
1490 | TA BORT R(LEAD & W(RITE
1500 | ORG #FFDC TA BORT WRITE
1510 | HEX B2
1520 | ORG #FFF3
1530 | HEX AC
1540 | ORG #FFDE TA BORT READ
1550 | HEX B2
1560 | ORG #FFF5

```

```

FFFS: AC      1570      HEY AC
1580 |-----|
1590 | UTBYGGD HEX-DUMP, AVEN ASCII-DUMP
1600 |-----|
1610 | ANDRINGAR I BAMLÄ HEXDUMP-RUTINEN
1620      ORG RFDAD
FDD: 4C CE FC 1630 MODBCHK JMP MDBCHK NY RUTIN
FDB: EA      1640      NOP
FDB1: EA      1650      NOP
FDB2: EA      1660      NOP
FDB3: 4C FC FC 1670 XAM JMP PRADR NY RUTIN
FDB4: A9 A0 1680 DATADUT LDA # " "
FDB5: 20 ED FD 1690 JSR COUT
FDB6: B1 3C 1700 LDA (A1),Y
FDB7: 20 DA FD 1710 JSR PRBYTE
FDB8: 4C C9 FC 1720 JMP NXTA NY RUTIN
1730
1740 | NY HEXDUMP-RUTIN
1750      ORG RFCC9
FCC9: 20 BA FC 1760 NXTA JSR NXTA1 KLART?
FCC: B0 06 1770 BCS PRASC JA, DUMPA SISTA ASCII
FCE: A5 3C 1780 MDBCHK LDA A1
FCD: 29 07 1790 AND #B07 B BYTES?
FCD2: D0 33 1800 BNE DTOUT NEJ, INTE AN
FCD4: 1810 | TILLÄGG FÖR ASCII-DUMP
1820 PRASC MOVQ A3,A1
FCD4: A5 40 ----> LDA A3
FCD6: B5 3C ----> STA A1
FCD8: A5 41 ----> LDA A3+1
FCD: B5 3D ----> STA A1+1
FCD: A2 02 1830 LDX #2 2 BLANKA
FCD: 20 4A F9 1840 JSR PRBL2
FCE: A0 00 1850 .2 LDY #0
FCE3: B1 3C 1860 LDA (A1),Y HÄMTA BYTE
FCE5: 09 B0 1870 ORA #B0 SATT BIT 7
FCE7: C9 A0 1880 CMP # " KONTROLL-TECKEN?
FCE9: B0 02 1890 BCS .3 NEJ
FCEB: A9 AE 1900 LDA # " JA, ERSATT MED " "
FCE: 20 ED FD 1910 .3 JSR COUT MATA UT ASCII
FCF: 20 BA FC 1920 JSR NXTA1 KLART?
FCF3: 90 01 1930 BCC .4 NEJ
FCF5: 60 1940 RTS JA, SLUT
FCF6: A5 3C 1950 .4 LDA A1
FCF8: 29 07 1960 AND #B07 B BYTES?
FCFA: D0 E5 1970 BNE .2 NEJ, INTE AN
FCFC: 1980 PRADR MOVQ A1,A3 JA, NY RAD
FCFC: A5 3C ----> LDA A1
FCFE: B5 40 ----> STA A3
FD0: A5 3D ----> LDA A1+1
FD02: B5 41 ----> STA A3+1
FD04: 20 92 FD 1990 JSR PRA1 SKRIV ADRESS
FD07: 4C B6 FD 2000 DTOUT JMP DATAOUT
2010 |-----|
2020 | UTBYGGT L-KOMMANDO (DISASM)
2030 |-----|
2040      ORG RFESE
FE5: C9 AE 2050 LIST CMP # " 20 LINJER?
FE6: D0 03 2060 BNE .1 JA
FE62: 4C CD FE 2070 JMP NEWLIST NEJ, NY RUTIN
FE65: 20 75 FE 2080 .1 JSR A1PC
FE68: A9 14 2090 LDA #20 20 RADER
FE6A: 48 2100 .2 PHA
FE6B: 20 B1 FB 2110 JSR LIST.1 DISASM 1 RAD
FE6E: 68 2120 PLA
FE6F: 38 2130 SEC
FE7: E9 01 2140 SBC #1 SNURRA 20 VARV
FE72: D0 F6 2150 BNE .2
FE74: 60 2160 RTS
2170
2180 | NY DISASSEMBLERINGSRUTIN
2190      ORG RFECD
FECD: 20 75 FE 2200 NEWLIST JSR A1PC
FED: 20 B1 FB 2210 .1 JSR LIST.1 DISASM 1 RAD
FED3: 2220 CMPD A2,PC KLART?
FED3: A5 3E ----> LDA A2
FED5: C5 3A ----> CMP PC
FED7: A5 3F ----> LDA A2+1
FED9: E5 3B ----> SBC PC+1
FEDB: B0 F3 2230 BCS .1 NEJ, DISASM NÄSTA RAD
FEDD: 60 2240 RTS JA, SLUT
2250
2260 | DISASSEMBLERA EN RAD
2270      ORG RFBBI
FBB: 20 D0 FB 2280 LIST.1 JSR INSTDSP DISASSEMBLERA
FBB4: 20 53 F9 2290 JSR PCADJ RAKNA UPP PC
FBB7: B5 3A 2300 STA PC
FBB9: B4 3B 2310 STY PC+1
FBBB: 60 2320 RTS
2330 |-----|
2340 | P-KOMMANDO (REN ASCII-DUMP)
2350 |-----|
2360      ORG RFFDE
FFDE: E9 2370 DB RE9 "P" KOMMANDO
2380      ORG RFFFS
FFFS: FC 2390 DB RFC
2400
2410      ORG RFEFD
FFFD: 20 92 FD 2430 .1 JSR PRA1 SKRIV ADRESS
FF0: A2 01 2440 .2 LDX #1 1 BLANK
FF02: 20 4A F9 2450 JSR PRBL2
FF05: A1 3C 2460 .3 LDA (A1,X) HÄMTA BYTE (X=0)
FF07: 09 B0 2470 ORA #B0 SATT BIT 7
FF09: C9 E0 2480 CMP #A0 LITEN BOKSTAV?
FF0B: 90 02 2490 BCC .4 NEJ
FF0D: 29 DF 2500 AND #RDF JA, ÖMVÄNDA TILL STOR
FF0F: C9 A0 2510 .4 CMP # " KONTROLL-TECKEN?
FF11: B0 02 2520 BCS .6 NEJ
FF13: A9 AE 2530 LDA # " JA, ÖMVÄNDA TILL " "
FF15: 20 ED FD 2540 .6 JSR COUT MATA UT ASCII
FF18: 20 BA FC 2550 JSR NXTA1 KLART?
FF1B: B0 0C 2560 BCS .7 JA, SLUT
FF1D: A5 3C 2570 LDA A1
FF1F: 29 1F 2580 AND #A1F 32 TECKEN?
FF21: F0 DA 2590 BEQ .1 JA
FF23: 29 0F 2600 AND #A0F 16 TECKEN?
FF25: F0 D9 2610 BEQ .2 JA
FF27: D0 DC 2620 BNE .3 NEJ
FF29: 4C BE FD 2630 .7 JMP CROUT
2640 |-----|

```

```

#FFDC
FFDC- B2
#FFF3
#FFDE
FFDE- B2
#FFF5
FFFS- AC
#FFD2
FFD2- EC
#FFE9
FFE9- C3
#FFCF
FFCF- ED
#FFE6
FFE6- C1
#FEC2,FEF5
FEC2- C6 34 20 75 FE 4C
FECB- CD FE 4C FB 03 20 D0 FB
FED: 6B B5 2C 6B B5 2D A2 0B
FEDB- B0 02 FD 95 3C CA D0 FB
FEE: A1 3A F0 3F A4 2F C9 20
FEEB- D0 03 4C D9 FC C9 60 F0
FEF: 39 C9 4C 4C FD FE
#FEFD,FF2C
FEFD- D0 03 4C
FF0: E4 FC C9 6C D0 03 4C E5
FF0B- FC C9 40 F0 19 29 1F 49
FF1: 14 C9 04 F0 02 B1 3A 99
FF1B- 3C 00 B8 10 FB 20 3F FF
FF2: 4C 3C 00 4C 59 FA 1B 6B
FF2B- B5 4B 4C C9 FC
#FCC9,FDOA
FCC9- 6B B5 3A 6B B5 3B A5
FCD: 2F 20 5A F9 B4 3B 1B 90
FCD8- 14 1B 20 54 F9 AA 9B 4B
FCE: B8 4B A0 02 1B B1 3A AA
#FFDC
FFDC- B2 2
#FFF3
FFF3- AC
#FFDE
FFDE- E9 1
#FFF5
FFF5- FC 6
#FDAD,FDC2
FDAD- 4C CE FC LNo
FDB: EA EA EA 4C FC FC A9 A0 jJjL00
FDBB- 20 ED FD B1 3C 20 DA FD mÅ1k ZÅ
FDC: 4C C9 FC L10
#FCC9,FD09
FCC9- 20 BA FC B0 06 A5 3C :B0,Xk
FCD: 29 07 D0 33 A5 40 B5 3C ).P3XÅ.<
FCD8- A5 41 B5 3D A2 02 20 4A ZÅ,m". J
FCE: F9 A0 00 B1 3C 09 B0 C9 y .I.<.I
FCEB- A0 B0 02 A9 AE 20 ED FD 0.). mÅ
FCF: 20 BA FC 90 01 60 A5 3C :o.,eXk
FCFB- 29 07 D0 E5 A5 3C B5 40 ).PeXk.<
FD0: A5 3D B5 41 20 92 FD 4C %=.A .ÅL
FDB: B6 FD 6B
#FESE,FE74
FESE- C9 AE 1.
FE6: D0 03 4C CD FE 20 75 FE P.LMÅ u0
FE6B- A9 14 4B 20 B1 FB 6B 3B ).H 1ÅhB
FE7: E9 01 D0 F6 60 1.Pv6
#FECD,FEDD
FECD- 20 75 FE u0
FED: 20 B1 FB A5 3E C5 3A A5 1ÅX>E:Z
FEDB- 3F E5 3B B0 F3 60 ?e10Å6
#FBB1,FBBB
FBB: 20 D0 FB 20 53 F9 B5 PÅ Sy.
FBBB- 3A B4 3B 60 1.16
#FFDE
FFDE- E9 1
#FFF5
FFF5- FC 6
#FEFD,FF2B
FEFD- 20 92 FD .8
FF0: A2 01 20 4A F9 A1 3C 09 ". Jy'<.
FF0B- B0 C9 E0 90 02 29 DF C9 -.I0..)_1
FF1: A0 B0 02 A9 AE 20 ED FD 0.). mB
FF1B- 20 BA FC B0 0C A5 3C 29 :B0,Xk)
FF2: 1F F0 DA 29 0F F0 D9 D0 -.PZ).PvP
FF2B- DC 4C BE FD 0L.Å

```

```

FCER- B0 B1 3A B6 3B B5 3A B8
FCF: B0 B0 20 4B A5 2C 4B 4C
FCFB- D7 B4 7B B0 D0 20 4A FB
FD0: 20 B0 CC EA EA 4C FD FC
FDB: 4C B3 FB
#FBB3,FBC0
FBB3- 1B A0 01 B1 3A
FBBB- 20 56 F9 B5 3A 9B 4C FA
FBC: FC

```


turbo apple

1430:—/
21 02 84

'SÄTT EN TURBO PÅ DITT APPLE-SYSTEM'

När man arbetar med mikrodata är en av nackdelarna att behöva vänta på utskrifter. Speciellt irriterande blir det vid ordbehandling, fakturering och annat interaktivt arbete. Problemet kan lösas på två sätt:

1

Köp en mycket snabb skrivare.
Kostnad: 15.000 kr och uppåt.

2

Köp ett buffrat skrivarkort till Ditt
APPLE. Kostnad: 1975 kr.

Det buffrade skrivarkortet B.PPI—103, från Torsby Microprocessor AB, har sin egen mikroprocessor och 6K RAM.

Texten som skall skrivas ut förs med maskinspråkshastighet över till kortets RAM-minne. Du behöver inte vänta mer än 1—2 sekunder innan kommandot över datorn återgår till tangentbordet eller programmet, förutsatt att buffertkapaciteten inte överskrids.

Kortet skickar tecken till skrivaren så snart den signalerar att den kan ta emot.

B.PPI—103 är det första buffrade skrivarkortet som kan dumpa Hiresgrafik till skrivare, utan att Du behöver ladda in speciella program. Programmet finns lagrat i ett av det EPROM som finns på kortet.

Olika skrivare kräver olika program. EPROM finns till EPSON MX—80, —100, ANADEX, IDS Paper Tider och Centronics 739. Fler kommer.

Du använder B.PPI—103 som andra skrivarkort. Det levereras med färdig 1.5 m lång kabel. Sätt bara i det.

SPECIFIKATION

Storlek: 230 x 77 x 10 mm

Vikt: 130 gram

Ansl. internt: APPLE slot 1—7

Ansl ext: 8 BIT parallell Centronic,

Processor: 6504

Minne: AKN, STROBE, TTL nivå
6K statiskt CMOS RAM

Programminne: 4k EPROM

Strömförbr: 5V 270 mA

Övrigt:

Glasfiberarmerat kretskort, lödmask på båda sidor, komponenttryck, guldpläterade fingerkontakter, 1.5 m kabel inkl. kontakter, "Burnt in" mer än 10 timmar. Följer APPLE's I/O protokoll för Basic, Monitor, Pascal 1.0, Pascal 1.1. Fungerar även med Microsofts CP/M-kort.

Garanti: 1 år på delar och arbete
exkl. moms

Pris: ENDAST 1295:—

inkl. kabel

Utvecklat och tillverkat i Sverige

Kontakta Din APPLE-försäljare

Varumärken:

APPLE är reg. varumärke för APPLE Computer Inc., USA.
CP/M är reg. varumärke för Digital Research Inc., USA.
Microsoft är reg. varumärke för Microsoft Inc., USA.

tmp

Torsby Microprocessor AB Utterbyn 2317 S 68500 Torsby Telefon 0560/42725

12675

Jag har fått en massa förfrågningar sen min förra artikel (som en del tyckte var svår och rörig, ursäkt.. Denna blir värre!) angående om man inte kan läsa in CATALOGen till en variabel i ett BASICprogram. Jovisst, APPLE kan allt (nästan). Efter ett flitigt studerande i APPLE ORCHARD så hittade jag i ett nummer (nr.) vol.1) en artikel om variabler i APPLESOFT. Med hjälp av den och en del kunskaper om DOS blev resultatet som följer.

```

; -----
; ! CAT.&C !
; ! VERSION 2 1982-11-26 !
; ! LEIF HELLSTRÖM !
; -----
;
IOBPTR .EQ $1C
VARPNT .EQ $83
TXTPTR .EQ $B8
TEMP .EQ $F0
LEN .EQ $F2
VTOCREAD.EQ $AFF7      Läser VTOC.
BUFF .EQ $B3B8        Adress till VTOC-buffer.
PTRGET .EQ $DFE3      Se APPLE ORCHARD.
;
;
INIT .OR $9500
LDA $3F6      Adress till &-vektor.
STA NJET+1    Spara undan ev. tidigare
LDA $3F7      &-hopp.
STA NJET+2
LDA #START
STA $3F6
LDA /START
STA $3F7
RTS
START CMP #$43      &C?
BEQ OK      Ja.
JMP NJET    Nej. Hoppa vidare.
OK JSR $00B1    GetChar.
LDA TXTPTR  Spara undan TXTPTR.
STA TEMP
LDA TXTPTR+1
STA TEMP+1
JSR FNDFL   Leta reda på FL%.
STA $01     Spara värde i adress $01.
JSR FNDFLS  Leta reda på adressen
STA TXTPTR till FL$
INY
LDA (VARPNT),Y
STA TXTPTR+1
JSR $03E3   Ladda A och Y med adressen
STY IOBPTR till DOS IOB.
STA IOBPTR+1
LDY #$00
LDA (VARPNT),Y
STA LEN     Spara längden på FL$.
JSR VTOCREAD  Läs VTOC.
LDY #$00    Pekare till en fil i buff
    
```

```

NEXTFL TYA
CLC      Öka pekaren med en fil
ADC #$23 ($23 bytes).
TAY
BNE NOREAD Behövs en ny CATsektor?
LDY #$01
LDA BUFF,Y Spårlänk till nästa CAT.
BEQ NOFILE Slut på CATALOG?
LDY #$04
STA (IOBPTR),Y
LDY #$02
LDA BUFF,Y Sektorlänk till nästa
LDY #$05 CATALOG.
STA (IOBPTR),Y
LDA #$00    Volym = 0
LDY #$03
STA (IOBPTR),Y
JSR $03E3   Hämta IOB-pekare.
JSR $03D9   Läs en sektor.
LDY #$0B
NOREAD LDX BUFF,Y
BEQ NOFILE  =$00? Slut på CATALOG.
INX
BEQ NEXTFL  =$FF? En borttagen fil.
DEC $01     Minska FL%.
BNE NEXTFL  Har vi hittat FL% st
            filer?
INY
INY
INY
TYA
TAX
LDY #$00
LDA LEN
BEQ OVER
MOVE LDA BUFF,X      Flytta filnamnet till FL$
AND #$7F
STA (TXTPTR),Y
INY
INX
CPY LEN
BCS OVER
CPY #$1D
BNE MOVE
BEQ OVER
NOFILE LDA #$01      Ingen fil hittad...
LDY #$00      FL$ = CHR$(1).
STA (TXTPTR),Y
OVER LDA TEMP        Lägga tillbaka TXTPTR.
STA TXTPTR
LDA TEMP+1
STA TXTPTR+1
RTS      Tillbaka till BASIC
FNDFL JSR FINDVAR
        .AS /FL%/
        .HS 00
FNDFLS JSR FINDVAR
        .AS /FL$/
        .HS 00
FINDVAR PLA
        Denna rutin 'stulen' ur
        STA TXTPTR APPLE ORCHARD.
        PLA
        STA TXTPTR+1
        INC TXTPTR
    
```

Software library

```
BNE CALLFP
INC TXTPTR+1
CALLFP JSR PTRGET
STA TXTPTR
STY TXTPTR+1
LDY #$01
LDA (VARPNT),Y
RTS
NJET   JMP $0000      Hopp till ev. tidigare &.
```

Denna rutin läser alltså CATALOGen ifrån ett APPLESOFTprogram med hjälp av ampersand (&) kommandot.

För att använda den gör:

Starta objektfilen (BRUN CAT.&C.OBJ).

HIMEM:38144

FL\$=" (30 mellanslag) "+"

Nu är rutinen initierad, detta behöver bara göras en gång. När man nu vill läsa in ett filnamn så laddar man FL% med ett filnummer och anropar rutinen med &C.

Med filnummer menas ett relativt nummer. Om CATALOGEN exempelvis ser ut så här:

DISK VOLUME 254

```
A 002 HELLO
B 002 CAT.&C.OBJ
A 003 TESTPROGRAM
```

Då har HELLO filnummer 1 och TESTPROGRAM har nummer 3.

Skälet till "+" i slutet av FL\$ är att annars kommer APPLESOFTens variabelpekare att peka rätt in i programmet, vilket medför att FL\$ kommer att bli ändrad i själva programmet...? Uhu.. Det var svårt att förklara, jag tror jag visar istället.

```
10 FL$="          "
20 FL%=1 : &C
RUN
```

Bli r...

```
LIST
10 FL$="Filnamn  "
20 FL%=1 : &C
```

Så, det var nog lättare att förstå.

Om det är någon, mot förmodan, som vill ha denna rutin och inte orkar eller kan knappa in den, skriv då på er beställning (när ni beställer andra disketter, vilket naturligtvis alla gör??) så kan jag kopiera över den.

VOL T LEN PROGRAM NAME CODE

```
98 + --- -----+
98 !   BEST OF CIDER PRESS !
98 + --- -----+
```

```
98
98 A 005 BEST OF CIDER PRESS
98 B 005 DISK EDIT+
98 B 005 TAPE EDIT+
98 I 008 B/BSTAT
98 B 003 FRE(X)
98 B 034 LOGO (DO NOT BRUN)
98 I 045 CHECKBOOK DISK
98 B 005 FASTSORT
98 B 002 LAZARUS
98 I 008 PAGE 1/2
98 A 006 HI-RES 21 COLORS
98 I 003 CHECKBOOK NOTES
98 B 009 SINGLE DRIVE COPY
98 I 009 IMPROVED CATALOG
98 I 005 F
98 B 002 SINGING FINGERS
98 I 002 DOS 3.1 OR 3.2?
98 A 051 LIBRARY LIST 10-79
98 I 007 D/A CONVERTER
98 B 002 D/A BIN.A$0.L$10
98 B 003 FAST SPLIT
98 I 011 DOS CHANGER
```

```
99 + --- -----+
99 !   SWEDE #1
99 + --- -----+
```

```
99
99 A 002 HELLO
99 A 003 ELEKTRON
99 B 007 REPLACE:
99 A 020 LAB3DIM
99 A 010 MÄNLANDARE
99 B 004 ED
99 B 002 BHGR
99 B 002 MOONL
99 B 005 AUTO
99 A 009 BLOCKAD
99 A 015 SHAPE30
99 B 003 BLIFE
99 A 005 BLIFE INSTRUKTIONER
99 A 003 BHGR INSTRUKTIONER
99 A 014 AUTORADNR/REPLACE-INS
99 I 046 DIS
99 B 007 BDIS
99 A 006 DISASSEMBLER-INSTRUKT
```

forts sid 60

File disk catalog

av Leif Hellström

På en av våra volymer (#21) finns det ett program som heter FILE DISK CATALOG med tillhörande binärfil FILE DISK CATALOG.X. Ni som har skaffat den volymen har säkert upptäckt att det är en sk. MULTI DISK KATALOG, dvs. en databas för disketter. Denna databas är utomordentligt behändig när ens antal disketter rasar i höjden och man vill snabbt och enkelt hitta en speciell fil. Det finurliga med FILE DISK CATALOG (och de andra MULTI DISK KATALOGERna) är den form av inmatning som finns. Man gör helt enkelt så att när programmet frågar efter disketten, så stoppar man in den och ... vips, programmet läser av katalogen. Fördelen är alltså att det behövs inget knapp-jobb för att få in filnamnen och det går dessutom väldigt snabbt (ungefär som en vanlig catalog). Databasen kan givetvis också sortera, söka och skriva ut innehållet.

När man startar upp programmet så dyker det upp en meny, och på den kan man välja mellan:

1. SORTERA
2. LÄGGA IN
3. SÖKA
4. SKRIVA UT DATA PÅ SKÄRMEN
5. SKRIVA UT DATA PÅ SKRIVARE
6. LADDA/SPARA DATA
7. SLUTA

För att välja någon av dessa alternativ så flyttar du cursorn med hjälp av ESC-tangenten och kvitterar med RETURN.

Ta och titta på alternativ nr. 2, LÄGGA IN. Där finns det en meny till som innehåller:

1. LÄGGA IN FRÅN TANGENTBORDET
2. LÄSA IN FRÅN CATALOGEN
3. ÄNDRA KOMMENTARFÄLTET
4. ÄNDRA DATA
5. TILLBAKA TILL HUVUDMENYN

Det som skiljer FILE DISK CATALOG från många andra MULTI DISK KATALOGER är att man här kan, förutom att den själv läser katalogen, skriva in data (filnamn eller vad man vill) direkt från tangentbordet och man kan sedan ändra alla data hur man vill. Detta är en stor fördel ex. om man vill skilja sina olika disketter med en snygg liten ruta (se klubbens nya lista).

När man nu har läst in alla sina disketter och fyllt upp med små söta rutor så vill man ha ut en lista på skrivaren. Så hade jag tänkt mig... Men, hoppsan, vad händer?? Det kommer ingenting på skrivaren utan bara på skärmen, det måste vara någonting som är fel här..!

Efter sju sorger och åtta bedrävelser hittade jag

felet och lyckad till sist rätta till det. Ni som har fått denna volyme efter Oktober behöver inte oroa er, men Ni andra behöver göra några små ändringar:

Stoppa in volymen (#21).

Skriv: BLOAD FILE DISK CATALOG.X

CALL -151

15EB:85 3E EA

1713:20 97 FE EA EA EA EA EA EA EA

172F:50 52 4F 47 52 41 4D 20 4E 41 4D 45 20

179E:A9 00 20 95 FE EA EA EA

BSAVE FILE DISK CATALOG.X,A\$800,L\$1800

Nu har ni en fungerande version av FILE DISK CATALOG och den går att starta antingen genom BRUN FILE DISK CATALOG.X eller från ett program.

Software library forts.

VOL	T	LEN	PROGRAM NAME	CODE
100	+	---	-----+>	
100	!		IAC #16	!
100	+	---	-----+>	
100				
100			DENNA DISK ÄR TVÄRR	
100			KVADDAD. MEN VI ATR-	
100			KOMMER NÄR VI FAR EN	
100			NY FRÅN IAC..	
101	+	---	-----+>	
101	!		CATTLECAR GALACTICA !	!
101	+	---	-----+>	
101				
101	I	002	HELLO	
101	I	043	APPLESOFT	
101	A	139	CATTLECAR GALACTICA 3	
102	+	---	-----+>	
102	!		DIVERSI DOS	!
102	+	---	-----+>	
102				
102	A	024	HELLO	
102	B	079	ASMDIV	
102	B	008	BUFFER	
102	B	010	PCARDS	

Diskdykning

Under denna rubrik skall vi gå lite djupare in i APPLE USER GROUPs diskettlåda, och undersöka vad som egentligen döljer sig bakom alla namn och nummer.

Om ni har synpunkter på vissa volymer så tveka inte en sekund utan sätt fart på handen och skicka ett (två eller flera) brev om vad och varför det var bra eller dåligt. (Jag skulle bli MYCKET tacksam.)

När man tittar på de befintliga listorna så säger ju namnen ingenting om vare sig kvalitet eller nyttan med programmet. Kvaliteten är ju som de flesta vet varierande... Väckigt varierande. Det finns program som GARBAGE som enbart skriver ut skräp, och det finns program som FILE DISK CATALOG (volym #21) som är en avancerad MULTI DISK CATALOG.

Och det vore ju intressant om man fick veta vad man skulle ha programmet till samt i en del fall hur man använder det.

Vilket alltså är precis vad denna kolumn är avsedd för:

Djupdykning volym #21.

```
+ ---+
!   AAA UTILITY 223   !
+ ---+
I 003 AAA UTILITY 223
I 005 AAB MESSAGE
I 025 APPLEADDIX
A 006 APPLEADDIX NOTES
I 002 AUTOBOOT 16
B 004 AUTOBOOT 16.X
B 022 BASIC-INTEGER
I 019 CATALOG CREATE
I 002 COPY LHS FAST
B 010 COPY LHS FAST.X
I 023 COPY MOGER 1
I 025 COPY MOGER 2
B 005 FILE #1
I 003 FILE DISK CATALOG
B 026 FILE DISK CATALOG.X
B 026 FILE DISK CATALOG.FP
I 009 FREE SECTORS SCOGIN (C)
B 003 M-A$300L$100
B 003 M1-A$800L$180
B 003 M2-A$800L$160
I 018 MAP SECTORS PAYMAR 3
I 015 MEMORY ADRESSES
B 002 MX-A$900L$80
I 008 RENUMBER->NEEDS WORK
I 033 SUBROUTINE PAK
I 016 TYPEWRITER+SOUND ROUTINE
I 019 UPDATE 16
I 019 UPDATE 16 REVISED
I 002 UPDATE 16 WARNING
```

```
A 040 X-FILE CABINET IV
I 024 X-MULTIPLE DISK CATALOG
B 003 RENUMBER MACH
```

APPLEADDIX (APPLEADDIX NOTES + M-,M1-,M2-,MX-)

Detta skall vara en slags databas enligt APPLEADDIX NOTES, men den fungerar inte som den borde (inte alls).

AUTOBOOT 16 (AUTOBOOT 16.X)

Om någon kan komma på meningen med detta program så tala gärna om det för oss andra. Själv brukar jag skriva PR#6.

BASIC-INTEGER

IntegerBASIC är många av de äldre programmen skrivna i, eftersom det var det första språket till APPLE. Om man har en APPLE II+ så har man APPLESOFT i ROM vilket innebär att om man inte kan köra dessa program. Det löstes snabbt med ett IntegerBASIC-kort eller med språkkortet. Om man nu inte har ngt av dessa så kan man starta (med BRUN) denna RAM-VERSION av IntegerBASIC, som då lägger sig snyggt och prydligt inom ramen 48k.

CATALOG CREATE

Detta program skapar en fil med katalogen för APPLEADDIX men då APPLEADDIX inte fungerar så har man inte stor nytta av detta heller.

COPY LHS FAST (COPY LHS FAST.X)

Ett kopierings program som kopierar en diskett till en annan. Programmet är snabbare än COPYA då det bara kopierar använda sektorer.

COPY MOGR 1

Ett program som kopierar enstaka filer. Det fungerar, men om man har FID på SYSTEM MASTER så klarar man sig utan detta.

COPY MOGER 2

Likadant som ovan utom på en punkt. Man får en chans att ångra sig innan den börjar kopiera.

FILE DISK CATALOG (FILE DISK CATALOG.X + FILE #1)

Se speciell artikel i detta nummer.

FREE SECTORS SCOGIN

En nätt liten rutin som talar om hur mycket plats det finns kvar på disketten.

MAP SECTORS PAYMAR 3

Detta program visar filerna på disketten, i vilken ordning och var de ligger. Det går även att få ut en lista på skrivare. Den version som låg här förut var avsedd för DOS 3.2, men efter lite svett och möda är den nu (821107) uppdaterad till DOS 3.3.

MEMORY ADDRESSES

Ett kalkylatorprogram som rör sig med Decimala, Binära samt Hex-tal.

RENUMBER->NEEDS WORK (RENUMBER MACH)

Needs work tror jag står där eftersom det saknades en fil på denna volym för att programmet skulle fungera. Efter att ha lagt dit den filen och provkört programmet kan jag inte hitta ngt fel. Det är alltså som man kan utläsa av namnet ett renumreringsprogram för IntegerBASIC-program.

SUBROUTINE PAK

Ett paket små praktiska rutiner för programmeringshjälp i IntegerBASIC.

TYPEWRITER+SOUND ROUTINE

Om någon har sett programmet THE INFINITE NUMBER OF MONKEYS på volym #97, vet han (hon) vad detta är. Det är en rutin för 'skivmaskineffekt' på bildskärmen, med ljud och fördröjning.

UPDATE 16

Ett program som updaterar en 13-sektors diskett till att boota på en 16-sektors APPLE. OBS! Man måste köra programmet ifrån DOS 3.2 eller med hjälp av DUBBELDOS.

UPDATE 16 REVISED

Troligen en förbättrad version av ovanstående.

UPDATE 16 WARNING

En vänlig varning att man kan inte kopiera hela disketten om man har kört UPDATE 16.

X-FILE CABINET

Ett databasprogram...

X-MULTIPLE DISK CATALOG

Ett hjälpprogram till X-FILE CABINET, som läser katalogen från disketter och lägger in dessa i databasen.

Djupdykning volym #16

- + ---+
 - ! THE DOS DISK !
 - + ---+
 - B 020 BIN.HELLO
 - ◊ A 031 DM II
 - I 018 RWTS.SC
 - A 006 HELLO1
 - A 007 JOHANNES
 - A 003 HELLO
 - B 002 RWTS.OBJ
 - I 025 STATUS.SC
 - B 005 STATUS.OBJ
 - B 007 PD
 - B 002 FTS
 - B 002 INIT
 - B 002 DUMP
 - B 002 ZAP
 - ◊ B 009 LHS CATALOG
 - ◊ B 008 LHS RECST VTOC
 - ◊ B 006 LHS DISK MAP
 - ◊ B 009 LHS COPY
 - A 003 COPY.TRSE
 - A 003 COPY.ADR
 - A 002 COPY.DOS
 - A 005 LANG.CARD-FIX
 - B 002 CAT (NORM)
 - B 002 CAT (2-KOL)
 - B 027 DEMUFFIN
 - A 005 FREE SEC
 - T 028 STATUS.TXT
 - ◊ B 011 DUBBELDOS

BIN.HELLO

Ett helloprogram som läser katalogen och låter användaren välja vilket program som skall köras.

DM II

DiskModify II är ett program som hjälper till då man vill ändra kommandon eller felmeddelanden i DOS. Ex. CATALOG -> CAT.

RWTS.SC

Källkoden till RWTS.OBJ.

HELLO1

JOHANNES

Ett program som 'döper' skivan på en undanskynt plats (VTOC = Spår 17, Sektor 0).

HELLO

RWTS.OBJ

Ett interface-program i maskinkod mellan APPLESOFT och RWTS (RWTS = Read/Write Track, Sector). Se APPLE ORCHARD Vol.1 Nr.3.

STATUS.SC

Källkodsfilen till STATUS.OBJ. Gjord för SC-assembler.

STATUS.OBJ

Ett maskinspråksprogram som på skärmen visar var och i vilken ordning filerna ligger på disketten.

PD

PokeDisk, som upphovsmannen så klurigt tänkte sig, är ett hjälpmedel då man vill/behöver titta på disketten 'nedanför' filnivå. Dvs. direkt på, sektor för sektor.

FTS, INIT, DUMP, ZAP

Fyra små rutiner som finns beskrivna i BENEATH APPLE DOS, en bok som beskriver hur DOSet fungerar. Säljs i Sverige av ProgramDistributören.

LHS CATALOG

En utökad catalog med uppgifter som längd och adresser.

LHS RECST VTOC

Reconstruct VTOC... När man har haft en diskett en längre stund, sparat och tagit bort filer, kan det ibland hända att en eller flera sektorer har blivit 'bortglömda'. Dvs. DOS tror att den sektorn

är upptagen medans det i själva verket inte alls finns ngt där. Det här programmet fixar till sådana fel på disketterna.

LHS DISK MAP

Programmet kontrollerar om de använda sektorerna är 'hela' eller om det är ngt fel.

LHS COPY

Samma program som COPY LHS FAST på volym #21.

COPY.TRSE, -ADR, -DOS

Tre små enkla program som visar användbarheten hos RWTS.OBJ.

CAT (NORM), CAT (2-KOL)

Om man laddar in (BLOAD) CAT (2-KOL) så får man en två-delad catalog, dvs. två filnamn på varje rad. CAT (NORM) är givetvis normalt format.

DEMUFFIN

Motsatsen till MUFFIN. Alltså DOS 3.3 till DOS 3.2.1.

FREE SEC

Programmet modifierar DOS så att CATALOG även visar antalet lediga sektorer.

STATUS.TXT

Källkoden till STATUS.OBJ i: textformat. (Författaren har storhetsvansinne.)

DUBBELDOS

Dubbeldos behöver nog ingen förklaring efter den publicitet den har fått, men det är alltså en DOS som automatiskt växlar mellan DOS 3.3 och DOS 3.2. (Vi tackar Paul Schlyter för att vi får använda detta. Tack, tack.)

LEIF HELLSTRÖM

Djupdykning volym #70

+ -----+
! IAC #12 !
+ -----+
A 002 HELLO
A 016 APPLESOFT LISTER INSTRUCTIONS
B 003 APPLESOFT LISTPRINTER
I 006 AUTHORSHIP
o A 023 CALENDAR
o I 055 CONCENTRATION
o I 017 CONNECTION
v I 062 CRIBBAGE
A 005 HEXCON
I 047 HOOVER DAM
B 004 PAGE DUMP
B 003 SOFTSYMREF
B 012 SOFTSYMREF.S
A 029 STOCK TRADER
B 012 TEDISSEMBLER.S
o I 010 VIDEO TEST

LISTPRINTER

Formatterar utskrift till skrivaren så att 60 teckens bredd samt programnamn och sidnummer blir utskrivet högst upp till höger på varje papper.

AUTHORSHIP

Lägger in REMsatser sist i ett IntegerBASIC program. T ex författarens namn. Dessa REMsatser får radnummer 40 000-40 009. 32 767 är ju annars det högsta radnummer man kan skriva i Integer.

CALENDAR

Ett väl fungerande program som kan visa alla månader, dess datum och dagar från den 1 jan 1582 till Man kan även få en snygg utskrift på skrivaren med hela årets tolv månader.

CONCENTRATION

52 kort läggs ut. Det gäller att vända upp två kort med samma valör. Ett spel typ 'Memory'. (Högupplösande)

CONNECTION

Vartannat drag får man sätta ut ett vertikalt eller horisontellt streck. Det gäller att få så många kvadrater som möjligt.

CRIBBAGE

Kortspel i högupplösande grafik.

HEXCON

Konverterar hex-decimala tal.

HOOVER DAM

Ett spel där man sköter ett kraftverk som skall försörja en stad med elektricitet. (Lågupplösande)

PAGE DUMP

En 'page' (256 b) i taget av minnet visas antingen i hex eller i ASCII.

STOCK TRADER (SOFTSYMREF(.S))

Köp, sälj eller stå över till nästa vecka är i stort sett de regler som gäller för detta 'börsspel'. Spelet omfattar 52 veckor och det gäller alltså att ha gjort så bra aktieaffärer som möjlig under den tidsperioden.

TEDISSEMBLER.S

En källkodsfil till ted-assemblern. Innehållet är...?

VIDEO TEST

Ett program som ritar upp vertikala och horisontella streck i olika kombinationer, avsett för fininställning för de APPLEanvändare som nyttjar en färg TV.

PATRIK ANDERSSON

Skatt

Vi närmar oss, oroväckande snabbt, dagen deklaration. Kanske denna artikel kan rädda magsåret i år.

Eftersom Du har tillgång till en Apple II eller Apple /// så är den största investeringen redan gjord, det blir bara programkostnaden som tillkommer, om Du nu skulle bestämma Dig för att prova det här med skatte-planering/beräkning på dator.

Vi har tittat på följande programvaru paket, vi har dock inte sakgranskat dem, utan bara tittat på funktioner och användbarhet.

COLUMBI SKATT
AB Columbi Data
Box 166
570 82 Målilla
Tel. 0495/212 70

SKATT, TAX m.m.
Snabb-Data i Lund AB
Box 2120
220 02 LUND

SKATT
Datarationalisering AB
Fjärdhundragatan 50
199 31 Enköping

Programmen innehåller en rad olika funktioner i vissa fall. Columbi skatt innehåller:
Skatteberäkning
Beräkning av huskalkyler
Levnadskostnadsbudget
Bilkostnadsbudget
Man planerar att komplettera paketet med 5-årig boendekostnadsprognos. Senaste revisionsdatum är 82.10.20, men man planerar att komma med en ny revision före 15 december.

Programmet är bra ur användarsynpunkt, man kan hela tiden backa med ESC till senast föregående meny. Svarsmöjligheterna vid inmatning är klart och tydligt angivna (ex. J/N, 1-4, 0,1,3) vilket är utmärkt. Dokumentationen omfattande och lättförståelig.

Programmets huvudsyfte är att få ner skatten för deklarerar med B-inkomst och ränteavdrag.

Snabb-Data har sina program separerade, man säljer alltså de olika modulerna för sig.

Skatt 82 är ett beräkningsprogram, det beräknar skatt för fysiska personer bosatta i Sverige 1982. Skatt har också goda möjligheter till skatte-planering, då man kan gå till baka och ändra ett eller flera värden och sedan göra en ny beräkning. Tax-programmet har enbart beräkningsmöjligheterna. Men speciar skatteslag.

Husköp beräknar boendekostnaden vid husköp.

Snabb-Data programmen har också god "svarsmöjlighets" angivning. Dokumentationen är utförlig.

Det program vi erhöll från Datarationalisering i Enköping var av samma typ som Skatt (Snabb-Data) men utan möjlighet att gå tillbaka och göra om beräkningen efter att ha ändrat någon parameter. Tyvärr var materialet något ofullständigt då ingen dokumentation medföljde. Man måste tyvärr också säga att den skvarsmöjlighetsangivningen var bristfällig.

Detta var kort om de skatte/ekonomi program som vi funnit på marknaden. Vi hoppas att kunna återkomma med en ordentlig funktionsgranskning av programmen i ett kommande nummer. Har Du själv erfarenheter så skriv en rad till oss i redaktionen.

ASM

```

1 *****
2 *
3 *   MAKSTR by R. D. PURVES
4 *
5 * Call -A.P.P.L.E. Sept. 1982
6 *
7 *****
8 *
9 * Constructs a string from data at (PNTR)
10 * then advances PNTR for next string
11 *
12 CHKCOM EQU %DEBE ;CHECK FOR COMMA
13 PTRGET EQU %DFE3 ;FIND VARIABLE
14 CHKSTR EQU %DD&C ;CHECK FOR STRING$
15 MAKST EQU %E9E9 ;MAKE-A-STRING
16 SAVD EQU %DA9A ;SAVE DESCRIPTOR
17 *
18 FORPNT EQU %85 ;
19 PNTR EQU %0 ;ADDRESS OF DATA IN MEMORY
20 RFLAG EQU %2 ;RETURN FLAG
21 *
22 *
23 ORG %300 ;DECIMAL 768 FOR CALL
24 JSR CHKCOM
25 JSR PTRGET
26 JSR CHKSTR ;ERROR IF NOT STRING$
27 STA FORPNT
28 STY FORPNT+1 ;SAVE ADDR OF DESCRIPTOR
29 LDY %$0
30 LOOP LDA (PNTR).Y ;READ CHAR
31 AND %$7F ;SET MSB OFF
32 STA %0200.Y ;PUT IN BUFFER
33 TAX ;SAVE AS POSSIBLE STOP-CHAR
34 BEQ EXIT ;END IF %00 OR %80
35 CMP %$0D ;OR CARRIAGE RETURN
36 BEQ EXIT
37 INY
38 BNE LOOP
39 LDY %$00 ;STOP-CHAR. FOR MAKST
40 STX %02FF ;PRETEND A %00 WAS READ
41 LDA %$FE
42 TAY ;FOR INC LOOP INDEX
43 STA RFLAG ;RETURN FLAG = 0,13 OR 254
44 INY
45 INCL INC PNTR ;ADVANCE PNTR
46 BNE NOINC
47 INC PNTR+1
48 NOINC DEY
49 BNE INCL
50 TYA ; 0 => A
51 LDY %$2 ;A,Y POINT AT BUFFER %200
52 JSR MAKST ;BUF -> STRING STORAGE
53 JMP SAVD ;SAVE DESCRIPTOR AT (FORPNT) AND EXIT

```

--End assembly--

61 bytes

Errors: 0

100 REM INPUT ANYTHING
BY R. D. PURVES

110 REM USAGE:- CALL INPT,SVAR\$
120 INPT = 768:: FOR J = INPT TO
INPT + 27: READ K: POKE J,K:
NEXT : DATA 32,190,222,32,2
27,223,32,108,221,133,133,13
2,134,32,111,253,32,57,213,2
00,162,141,32,233,227,76,154
,218

Varför bara utnyttja halva disketten ?



198:-!, moms o. porto tillk.

när Du nu med hjälp av

”HOLE a new diskette”

även kan använda baksidan!

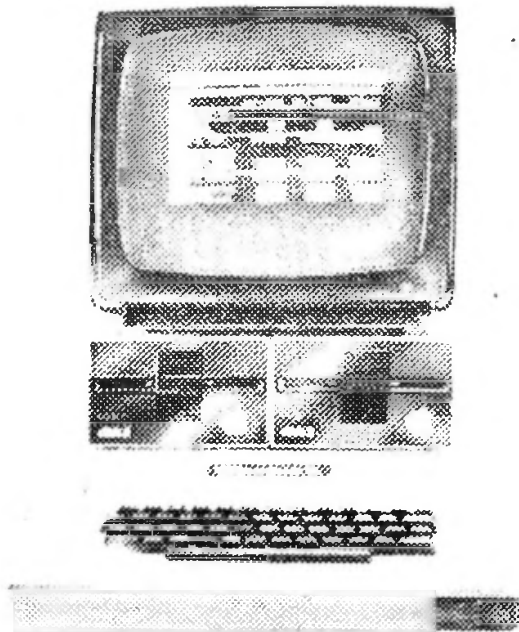


Scandinavian
Electronic and
Computing

Föreningsgatan 30 Box 2083 200 12 Malmö
Telefon 040/782 63

Vi säljer apple och VisiCalc!

microdatorn och plane-
ringsprogrammet med
vars hjälp Du analyserar
och snabbt ändrar Dina
egna uppställningar av
siffermaterial!



VISICALC-PLANERAREN KAN ANVÄNDAS TILL:

EKONOMI: Budgetering, investeringskalkyler, finansiella analyser.

FÖRSÄLJNING: Marknadsplaner, prissättning, offertunderlag och prognoser.

INKÖP: Kalkyler, inköpsplanering.


UTVECKLING: Lönsamhetsanalyser.

FÖRETAGSLEDNING: Prognoser, projektplaner, styrelsepresentationer.

VÄLKOMMEN
TILL EN DEMONSTRATION

Administratörema

I Stockholm och Västerås
Box 20058, 161 20 Bromma, Tel. 08-985420

Vi demonstrerar och säljer även alla andra program gjorda
för  apple microdatorer.