

PROGRAM

nittinian

BITEN

SP-2

Tecken-
medien

Texteditor
i C99

42
SPELET

Innehåll

Redaktören ...	2
Protokoll från årsmötet	3
Ordföranden ...	3
Stadgar	4
Autofile	5
42-spelet	6-8
Annonser	8
DM 1000 version 3.5	8
Fel i TI-Writer	9
Programbanken	10-12
Forth-jämförelse	13
Texteditor i c99	14-15
Freeware	16
Teckensmeden	17-18
Snabbare HCHAR	18-20
Minefield	21
TI-Writer Option 3	22-24

ISSN 0281-1146

Redaktören . . .

I detta nummer börjar vi med att sätta in checksummor i vissa programlistningar. Vi har haft det uppe i ett par nummer, men inte fått någon direkt respons från medlemmarna. Den här gången är det programmet AUTOFILE som är listat på det viset. Eftersom det programmet kräver både disksystem och expansionsminne tyckte jag att det var lämpligt. Se PB 87-1 sid 4 för en närmare beskrivning.

På mötet i Stockholm den 16 maj visade vi ett nytt tyskt program, **Turbo-Pasc 99**. Det är en Pascal-kompilator som har lånat vissa drag från Borlands Turbo Pascal. Dock naturligtvis med de begränsningar som minnet på 99:an ger.

Myarcs nya Geneve-dator skrev vi om i förra numret. Så värst mycket nytt har inte hänt sen dess egentligen. I tidningen MICROpendium har de börjat testa datorn. I maj-numret kom den första delen av deras recension. De är positiva, men de har tydligen inte fått det nya operativsystemet ännu. Den nya BASIC som ska levereras är tydligen inte heller klar.

Nya stadgarna

som antogs av årsmötet finns tryckta i detta nummer. Det är endast mindre förändringar jämfört med tidigare. I stadgarna framhåller vi nu tydligare än tidigare föreningens ideella karaktär. Vi har också, med de nya stadgarna som grund, gjort en förnyad framställning hos posten om att få sända ut vår post som föreningsbrev. Detta har nu godkänts. Det innebär inbesparingar på en dryg krona per utskickad Programbiten.

Tryckfelsniss

har härjat igen i tidningen. I förra numret blev bilden på Myarcs kort upp-och-nervänd. Orsaken till detta var bristande information från mig till tryckeriet. Jag tror dock inte att det ställde till några större problem för våra vakna läsare.

Freeware

I detta nummer publicerar vi en ny lista över de freeware som föreningen fått från olika håll. Vi har ju tidigare avsatt en del pengar för beställning av sådana program från olika håll. Innan sommaren kommer styrelsen att besluta om vi ska införskaffa ytterligare freeware.

Databas

Det har diskuterats vid olika tillfällen, både i föreningens styrelse och i tidningens spalter, om vi ska skaffa oss en egen databas för de som har modem. Andra dataföreningar håller sig med det. Det är tex mycket vanligt att 99-klubbar i USA har en egen "bulletin board" som innehåller senaste nytt, program som kan hämtas och mycket annat. Det finns förslag om att vi ska skaffa något liknande. Styrelsen kommer att utreda kostnaderna för att hålla en sådan. Dessutom krävs det naturligtvis intresserade som kan administrera det hela. Hör av dig till föreningen om Du är intresserad!

Programmen i detta nummer

tar upp en ganska stor del av tidningen. Hoppas något av dem kan passa någon regnig dag i sommar.

I detta nummer har vi med ett längre program skrivet i c99. Det är det första C-programmet vi publicerar, men inte det sista hoppas jag.

Friska programidér från far och son Hovmöller har vi sett tidigare. Nu är de tillbaka och har också aviserat att de har mera på gång.

Ett par assemblerprogram finns också med.

Skriv i Programbiten

Som vanligt har vi dock plats i spalterna för flera bidrag. Skicka in resultat av ditt programmerande i sommar till Programbiten!

I redaktionen:

Redaktör	Peter Odelryd
Utmaningsredaktör	Anders Persson
Förth-redaktör	Lars-Erik Svahn
Programförmedlare	Börje Häll
Allt-i-allo	Claes Schibler

Föreningens och redaktionens adress:

Föreningen Programbiten
c/o Schibler
Wahlbergsgatan 9, nb
121 46 Johanneshov

Föreningens telefon:

08/761 51 62 tisdagar kl 18-20. Michael Öhman svarar.

Datainspektionens licensnummer: 82100488

Postgiro: 198300-6

Medlemsavgiften för 1987 är 120:-

Annonser, insatta av enskild medlem (ej företag), som gäller försäljning av moduler eller andra tillbehör i enstaka exemplar är gratis.

Övriga annonser kostar 1000 SEK per helsida, 500 SEK per halv sida. För löslad som skickas med tidningen gäller 1 000 SEK per blad.

För kommersiellt bruk gäller följande:

Mångfaldigande av innehållet i denna skrift, helt eller delvis, är enligt lag om upphovsrätt av den 30 december 1960 förbjudet utan medgivande av Föreningen Programbiten. Förbudet gäller varje form av mångfaldigande genom tryckning, duplicering, stencilering, bandinspelning, diskettinspelning etc.

Föreningens tillbehörsförsäljning

Följande finns att köpa för medlemmar genom att motsvarande belopp insätts på postgiro 198300-6.

För TI-99/4A:

Användartips med Mini Memory	60:-
Nittinian T-tröja	40:-
99'er magazine nr 12/82, 1-5, 7-9/83 (per styck)	20:-
Programbiten/Nittinian 1986 (TI 59+99)	80:-
Programbiten/Nittinian 1985 (TI 59+99)	80:-
Programbiten/Nittinian 1984 (TI 59+99)	80:-
Nittinian, årgång 1983 (TI-99/4A)	60:-

För TI-59 mm

Programbiten, årgång 1983 (kalkylatorer)	40:-
Programbiten, årgång 1982 (kalkylatorer)	40:-
Programbiten, årgång 1981 (kalkylatorer)	40:-
Programbiten, årgång 1980 (kalkylatorer)	40:-
Programbiten, årgång 1978/79 (kalkylatorer)	40:-
Programbiten, fem årgångar 1978-83	160:-
Programbiten, sex årgångar 1978-84	220:-
Programbiten, sju årgångar 1978-85	300:-
Katalog med belgiska och engelska program för räknare TI-57, TI-58, TI-59	20:-
Föreningens programmeringsblanketter (TI-59), olika typer, block om 50 blanketter (se PB 83-1 sid 30) per block	11:-
Patenthandlingar TI-59	25:-
Tom magnetkortsplånbok	10:-



PROTOKOLL FÖRT VID ÅRSMÖTE FÖR FÖRENINGEN PROGRAMBITEN DEN 21 FEBRUARI 1987 I MILITÄRHÖGSKOLAN, STOCKHOLM

1 Mötet öppnades av föreningens ordförande Jan Alexandersson

2 Till mötesordförande utsågs Jan Alexandersson. Till mötessekreterare valdes Niels Hovmöller, och till att justera dagens protokoll utsågs John Hanssen och Peter Odelryd.

3 Mötet konstaterades vara utlyst i god tid. Ett drygt 20-tal personer hade infunnit sig, varav 15 röstberättigade medlemmar.

4 Dagordningen upplästes och godkändes.

5 Styrelsens årsberättelse delades ut och godkändes. Det konstaterades att föreningens ekonomi är god, tack vare den goda likviditeten och trots vikande medlemstal och höjda portosatser.

6 Revisionsberättelsen föredrogs.

7 Årsmötet fastställde resultat- och balansräkningen för år 1986 och beviljade styrelsen ansvarsfrihet för det gångna verksamhetsåret.

8 Mötet antog för andra gången det beslut om stadgeändring som föreslogs 1986-11-29.

9 Styrelseval förrättades. Göran Nygren framlade valberedningens förslag. Jan Alexandersson omvaldes till ordförande, Eddy Hedlund blev sekreterare och Mikael Nordlin omvaldes som kassör. Peter Odelryd utsågs till redaktör för Programbiten i ännu ett år. Ovanstående ledamöter framhöll att de hade mycket att göra och att de antog uppdragen under förutsättning att den övriga styrelsen hjälpte dem med arbetet med medlemsmötena och tidningen.

Till övriga styrelseledamöter valdes: Claes Schibler (omval) och Börje Häll (omval till programbankir), Niels Hovmöller (nyval) och Göran Nygren (come-back).

10 Till revisorer för verksamhetsåret 1987 valdes Barbro Nygren och Anders Törnquist, och Lennart Agdeus utsågs till revisorssuppleant.

11 Valberedningen fick följande ledamöter: Åke Olsson (sammankallande) och Michael Öhman.

12 Medlemsavgiften fastställdes till oförändrade 120:- för ett år.

13 Årsmötet uppmanade styrelsen att genom tex annonser, pressreleaser m m försöka höja medlemsantalet, särskilt genom att söka intressera personer som inte har mycket tilläggsutrustning att komma med och utnyttja föreningen.

14 Den formella delen av årsmötet avslutades.

Stockholm 21 februari 1987



Niels Hovmöller

Justeras:

John Hansen

Peter Odelryd

Ordföranden ...

Det vore trevligt om du som har synpunkter på vad föreningen skall syssla med hör av dig. Skriv gärna och berätta vad du saknar och om du vill ha mer eller mindre av något. Speciellt du som tycker att vi skriver om för avancerade saker i tidningen bör höra av dig. Vi har nu fått två nya medlemmar i styrelsen (Niels Hovmöller och Eddy Hedlund) som har sitt mesta kunnande när det gäller BASIC och Extended Basic med kassettbandspelare. Detta bör ge oss bättre möjligheter att hjälpa dig som ej har diskdrive.

Det finns även en frågespalt i tidningen som besvaras av Börje Häll. Du kan skriva och fråga om allt som rör TI-99 och TI-59.

Jag har nu skaffat två nya halvhöjds diskdrivar från USA. Detta betyder att båda två för plats i expansionsboxen. De är tillverkade av FUJITSU och kostade 100 USD per styck. Med porto och moms blev det 950 kr per styck. De är ljusgrå och mycket tystare än andra drivar jag sett. Drivarna klarar DS/DD (360 kb) men med mitt eget TI-kontrollkort kan jag bara använda DS/SD (180 kb).

När nu bra skivor säljs för 10 kr styck är det nog inte värt besväret att köpa ett nytt kontrollkort från CORCOMP eller MYARC. Eftersom det inte finns möjlighet till underkataloger med något av de tillgängliga kontrollkortet blir det ofta svårt att hålla reda på alla filer på samma disk. En annan nackdel med CORCOMP och MYARC är att det finns vissa speciella program som adresserar TI-kontrollkortet direkt och inte med vanlig DSRLNK. Dessa program fungerar endast med TI-kontrollkort.

Både MYARC och CORCOMP kan adressera 4 olika diskdrivar vilket är en nackdel om du skaffar RAM-disk nr 2. Den första RAM-disken kan ställas in för CRU >1000 vilket gör att alla nummer från 1 och uppåt är möjliga. Om du skaffar en andra RAM-disk så måste du välja CRU större än >1100 vilket innebär att den måste ha nr 4- (TI) eller 5- (CORCOMP och MYARC).

Intressantare är då ett nytt hårddiskkontrollkort från MYARC. Detta kostar USD 300 inkl kablar och kan adressera upp till 240 Mb.



Jan Alexandersson
Springarvägen 5, 3 tr
142 00 Trångsund
☎ 08/771 05 69

SIG-Hur funkade det?

av Folke Andersson

Jag, Folke Andersson har ett program som heter "CASS-LOAD" som är freeware. Detta program är mycket intressant i många avseenden. Ett är det sätt på vilket det är lagrat på diskett! Jag har tänkt att med bästa förmåga ta reda på hur det går till! Är Du intresserad så ring under-tecknad!

Allt material, program, hjälpmedel, tips, förklaringar kommer Du som intresserad att få Dig tillsänd, mot att Du sänder mig en formaterad diskett plus portoavgift. Jag kan inte garantera att jag kan ge alla förklaringar men jag vill försöka så gott jag kan.

Du bör ha E/A-modulen och manual, expansionsbox och 32K-minne för att delta i denna övning. Hör av dig innan 15 juli till undertecknad.

Folke Andersson
Pl. 3254
93400 KÅGE
tel 0910/22280 kl 19.00-22.00

STADGAR FÖR FÖRENINGEN PROGRAMBITEN

Årsmötets beslut den 23 februari 1985.

Reviderade den 21 februari 1987.

Kapitel 1. Föreningens syfte.

Föreningen Programbiten har till syfte att samla personer med intresse för programmerbara räknare (företrädesvis TI-59) och person- och hemdatorer (företrädesvis TI-99/4A) i sådana former att unga och äldre kan hjälpa varandra med avseende på maskiner, tillbehör, program, användningsområden m m. Föreningen är öppen för alla.

Föreningen utger, för att främja detta syfte, medlemstidningen PROGRAMBITEN, som i huvudsak avses behandla nämnda datorer och räknare.

Föreningen skall vara ideell. Genom dess verksamhet skall sådana aktiviteter främjas som stimulerar och utvecklar medlemmarnas intresse för datorteknik i allmänhet och nämnda datorer och räknare i synnerhet.

Kapitel 2. Föreningens sammansättning.

Som medlem räknas den som betalt av årsmötet fastlagd medlemsavgift.

Medlem, som bryter mot föreningens stadgar eller på annat sätt uppenbarligen skadar föreningen eller dess syften, kan av styrelsen med enhälligt beslut uteslutas ur föreningen.

Kapitel 3. Föreningens verksamhetsår.

Programbitens verksamhetsår och funktionärernas mandattid är kalenderåret (1 januari – 31 december).

Kapitel 4. Föreningens exekutiva organ.

Programbitens exekutiva organ är:

- a) Föreningsstämman, – ordinarie (= årsmöte)
Föreningsstämman, – extra
- b) Styrelsen
- c) Revisorerna
- d) Årsmötets valberedning.

Kapitel 5. Programbitens funktionärer.

Årsmötet väljer följande funktionärer:

- a) Styrelse, bestående av ordförande, sekreterare, kassör och redaktör, vilka samtliga utses till befattning av årsmötet, samt därutöver minst en och högst fem ledamöter i övrigt.
- b) Två revisorer och en revisorsuppleant.
- c) Valberedning bestående av två ledamöter, varav en sammankallande.

Övriga funktionärer utses av styrelsen.

Programbitens funktionärer skall vara medlemmar i föreningen. Undantagna är revisorerna, som kan vara utomstående.

Funktionärerna är ansvariga inför det organ som utsett dem.

Kapitel 6. Stämman.

Stämman är föreningens högsta beslutande myndighet och bildas av föreningens samtliga medlemmar, som alla har en röst vardera. Stämmans ordförande har utslagsröst utom i valfrågor. Minst en stämma skall hållas årligen.

Årsmötet, dvs ordinarie stämma, skall hållas senast 15 mars.

Stämma är beslutsmässig med det antal medlemmar som infinner sig, förutsatt att kallelse med tid och plats skriftligt utsänts till medlemmarna antingen genom tidningen eller separat, senast 15 dagar innan sammanträdet.

Årsmöte sammankallas av styrelsen.

Extra stämma sammankallas av styrelsen eller av revisorerna på eget initiativ eller när minst 30 medlemmar så fordrar.

Ingen får delta i beslut eller leda sammanträde när fråga om ansvarsfrihet för honom/henne själv behandlas.

Vid årsmöte förekommer ärenden, som av styrelsen hänskjutits till årsmötet eller som skriftligen inlämnats till styrelsen av medlem senast sista dagen av det verksamhetsår som skall redovisas på mötet. Beslut kan endast fattas i frågor som upptagits på föredragningslistan.

Årsmötet skall behandla följande frågor:

- a) Styrelsens berättelse över föregående verksamhetsår.
- b) Revisorernas berättelse över föregående verksamhetsår.
- c) Ansvarsfrihet för styrelsen och funktionärerna.
- d) Val av styrelse med funktionärer.
- e) Val av revisorer.
- f) Val av representanter i valberedningen.

Protokoll skall föras som beslutsprotokoll och skall justeras inom 20 dagar efter mötet av sammanträdesledaren jämte två av årsmötet utsedda justeringsmän. Protokollet skall i justerat skick utsändas till samtliga medlemmar.

Kapitel 7. Föreningsstyrelsen (FS).

FS består av föreningens ordförande, kassör, sekreterare och redaktör samt minst en ledamot därutöver.

FS är beslutsmässig, om minst tre ledamöter är närvarande och om kallelse med tid och plats meddelats senast tre arbetsdagar innan mötet.

FS ordförande har utslagsröst vid lika röstslifför.

Protokoll skall föras. Protokollet justeras av ordförande och utskickas till styrelsemedlemmarna.

Det åligger FS:

- att inför årsmötet ansvara för hela föreningens förvaltning och verksamhet.
- att verkställa årsmötets beslut.
- att handha och ansvara för föreningens medel, bereda inkomna förslag, handha föreningens korrespondens samt i övrigt sköta löpande ärenden.
- att vid behov förrätta interimisval.
- att före utgången av varje verksamhetsår tillse att gällande stadga är utskriven.
- att senast 20 dagar före ordinarie stämma överlämna föreningens årsredovisning till revisorerna.

Kapitel 8. Föreningens revisorer.

Revisorerna

skall granska föreningens förvaltning och verksamhet. skall senast fem dagar före möte om ansvarsfrihet till FS inlämna revisionsberättelse.

Revisorerna äger rätt

- att närhelst de så önskar taga del av samtliga räkenskaper, protokoll och andra handlingar.
- att begära och erhålla upplysningar rörande verksamhet och förvaltning.
- att övervara samtliga föreningsorgans sammanträden med yttranderätt.
- att inkalla samtliga föreningsorgan.

Kapitel 9. Föreningens valberedning.

Årsmötets val förbereds av valberedningen.

Valberedningens förslag skall inlämnas till FS senast tolv dagar innan årsmöte. Förslaget bör helst distribueras med kallelsen till årsmötet.

Kapitel 10. Ändring av stadgarna.

Förslag till ändring av dessa stadgar inlämnas skriftligen till FS senast tolv arbetsdagar före årsmötet, vid vilket FS utlåtande skall föreligga. För slutgiltigt bifall fordras beslut på två på varandra följande stämmor, varav minst ett årsmöte. På det möte som sist hålles skall beslutet ha bifallits med minst 2/3 majoritet.

Kapitel 11. Tolkning av stadgarna.

Årsmötet äger rätt att tolka dessa stadgar. FS kan i samråd med revisorerna avgöra i årsmötets ställe.

Kapitel 12. Föreningens upplösning.

Beslut om föreningens upplösning kan endast fattas på stämma. För slutgiltigt bifall fordras beslut på två på varandra följande stämmor, varav minst ett årsmöte. På det möte som sist hålles skall beslutet ha bifallits med minst 2/3 majoritet.

Kallelse till stämma som behandlar föreningens upplösning skall innehålla ett av initiativtagarna – styrelsen, revisorerna eller grupp av medlemmar om minst 30 – motiverat förslag till nedlägningsbeslut med ett tydligt förslag om fördelningen av föreningens tillgångar.

Autofile

av Per Virving

Torpkillen. 5
741 00 KNIVSTA
018/38 52 01

Det här programmet tyckte jag att jag var så duktig som hade gjort, men så upptäckte jag att sådana här program redan fanns, men gjorda på ett annat sätt. I alla fall kanske det här blir en inspiration åt någon.

AUTOFILE, som jag kallat programmet, kan läsa vilken fil som helst (på disk), oberoende av format, och skriver ut innehållet på skärmen.

Det hela går ut på att man läser diskkatalogen och laddar sen olika värden i minnet beroende på vilket format filen har, dvs INT/FIX, DIS/FIX osv (se även Generell programstartare i Utmaningen 85-1).

Med CALL LOAD-satserna på raderna 290-300 lägger programmet in de olika formaten i OPEN satsen på rad 310.

I minnet lagras de olika parametrarna (Display, Fixed osv) med olika ASCII-koder:

162 = Display
245 = Internal
243 = Variable
250 = Fixed

INPUT,DISPLAY,VARIABLE 80

Ser i minnet ut som:

```
146 INPUT
179 " "
162 DISPLAY
179 " "
243 VARIABLE
179 " "
243
200
2 anger hur många siffror postlängden använder
56 8
48 0
0 slut på raden/satsen
```

2:an talar om hur många siffror 80 består av dvs 2 i det här fallet. Om det tex hade varit en Dis/var 163 fil hade det i stället stått 3 där. Därför finns raderna 240-260 i mitt program.

För att datorn ska veta var i minnet den ska ladda in dessa koder finns rad 160, den letar efter understryknings-tecknet på rad 310 och lagrar sedan den adressen i ADR.

Om Du ÄNDRAR programmet - använd ABSOLUT IN-TE understryknings-tecknet, i varken REM-rader eller någon annanstans.

Programförklaring

100-120 Upstart
130-170 Sök efter laddnings-adress
180-190 Fråga efter drive # och filnamn
200-230 Leta efter filen på disken, är det en program-fil?
240-300 Ladda parametrarna
310-350 Öppna och läs av filen
370 Sätt tillbaks !_!!! på rad 320 !!! VIKTIGT
390-400 En gång till?
410-430 Ett fel har uppstått, skriv vilket
440-460 Subrutin för att vänta på en tangenttryckning

Om datorn inte hittar någon laddningsadress när du kör programmet, kolla först att rad 310 är rätt, om den är det och det ändå inte funkar, prova med att programmet får börja söka efter laddningsadressen vid en lägre värde, tex -1200.

När man tittar på en fil kan man stoppa scrollningen med en tangenttryckning.

Man kan avbryta avläsningen av filen med FCTN S.

```
U 10 ! AUTOFILE
M 20 ! (c) SCS piece 1987
W 30 ! ext.-basic,disk drive
Q 40 ! memory expansion
I 100 CALL INIT
G 110 CALL CLEAR :: DNR=1
S 120 ON ERROR 410
M 130 DISPLAY AT(10,7):"SEARCHING FOR": " LOA
D-ADDRESS"
K 140 K$=CHR$(179):: DI$=K$&CHR$(162):: FI$=K$&CHR$(
250):: VA$=K$&CHR$(243):: IN$=K$&CHR$(245)::
SL$=CHR$(200)
A 150 FOR A=-900 TO -1 :: CALL PEEK(A,B):: DISPLAY
AT(5,12):A :: IF B=95 THEN ADR=A-3 :: GOTO 18
0
Q 160 NEXT A
U 170 DISPLAY AT(14,10)BEEP:"!!ERROR!!": " CAN'T
FIND LOAD-ADDRESS" :: END
K 180 CALL CLEAR :: DISPLAY AT(20,1):"DRIVE?":DNR :
: DISPLAY AT(22,1):"FILENAME:" :: ACCEPT AT(2
0,8)VALIDATE("12")SIZE(-1):DNR
C 190 ACCEPT AT(22,10)SIZE(10):F$
W 200 OPEN #2:"DSK"&STR$(DNR)&".",INPUT ,RELATIVE,I
NTERNAL
G 210 INPUT #2:DF$,TYP,SEK,POSTLEN :: IF DF$="" THE
N CLOSE #2 :: DISPLAY AT(13,7):"FILE NOT FOUN
D" :: GOTO 360 ELSE IF DF$=F$ THEN CLOSE #2 :
: GOTO 230
Q 220 GOTO 210
S 230 IF TYP=5 THEN DISPLAY AT(16,7):"NO DATA FILE!
" :: GOTO 360
I 240 IF POSTLEN<10 THEN LE=1 :: GOTO 270
K 250 IF POSTLEN<100 THEN LE=2 :: GOTO 270
I 260 LE=3
U 270 SL$=CHR$(200)&CHR$(LE)
A 280 T$(1)=DI$&FI$&SL$ :: T$(2)=DI$&VA$&SL$ :: T$(
3)=IN$&FI$&SL$ :: T$(4)=IN$&VA$&SL$
W 290 FOR A=1 TO LEN(T$(TYP)):: CALL LOAD(ADR+A,ASC
(SEG$(T$(TYP),A,A))):: NEXT A
Y 300 FOR A=1 TO LEN(STR$(POSTLEN)):: CALL LOAD(ADR
+LEN(T$(TYP))+A,ASC(SEG$(STR$(POSTLEN),A,1)))
:: NEXT A :: CALL LOAD(ADR+LEN(T$(TYP))+LEN(S
TR$(POSTLEN))+1,0)
Q 310 OPEN #1:"DSK"&STR$(DNR)&". "&F$,INPUT ,!_ !!!
!!!!!!!!!!!!!!!!!!!!!!
W 320 IF EOF(1)THEN CLOSE #1 :: GOTO 370 ELSE INPUT
#1:A$ :: PRINT A$
Q 330 CALL KEY(0,K,S):: IF K=8 THEN PRINT "!! BREAK
!!" :: CLOSE #1 :: GOTO 370
Q 340 IF S=1 THEN CALL WAIT
I 350 GOTO 320
Y 360 DISPLAY AT(12,10)BEEP:"!!ERROR!!"
G 370 CALL LOAD(ADR+2,131):: CALL LOAD(ADR+3,95)::
FOR G=4 TO 28 :: CALL LOAD(ADR+G,131):: NEXT
G
A 380 PRINT :: DISPLAY AT(24,1):"AGAIN? Y" :: ACCEP
T AT(24,8)VALIDATE("YN")SIZE(-1):S$
M 390 IF S$="Y" THEN 180
S 400 END
M 410 CALL ERR(ERC,ERT,ERS,ERL):: IF ERT<0 THEN 430
K 420 PRINT "FILE ERROR IN FILE #":ERT:"LINE":ERL :
: GOTO 370
Q 430 PRINT "PROGRAM ERROR!": "ERROR #":ERC:"LINE":E
RL :: GOTO 370
E 440 SUB WAIT
S 450 CALL KEY(0,K,S):: IF S<1 THEN 450
Q 460 SUBEND
```

Save till disk med lediga sektorer

av Jan Alexandersson

När du har en skiva som är nästan full men har 2 lediga sektorer så skulle man tro att det skall gå att spara ett kort BASIC-program. Om programmet tar mindre än 256 bytes så borde 2 sektorer räcka. En sektor används som filhuvud och en sektor för data.

Detta fungerar inte utan det blir ett felmeddelande. Om du då tittar på skivan med en DISK MANAGER så ser du att en sektor har tagits av programmet medan den sista sektorn är ledig. Om du gör DELETE av denna fil med en äldre version (2.2 eller 3.3) av DM 1000 så har du verkligen ställt till det på skivan eftersom SEKTOR 0 då blir LEDIG. Denna skiva bör genast kopieras fil för fil.

Alla andra sätt att fylla skivan till 100% fungerar dock. Du kan spara ett program på 3 sektorer när det finns 3 lediga sektorer på skivan. Med en DISK MANAGER går det tom att kopiera in en fil på 2 sektorer på en skiva med 2 lediga sektorer.


```

340 FOR J=0 TO 1 : CH=((ASC(SEG$(A$,J+1,1))-48
)%)2)+96 : CALL HCHAR(V,H+J,CH) : CALL HCHAR(
V+1,H+J,CH+1) : NEXT J
350 NEXT H : NEXT V
360 FOR I=2 TO 8 : CALL COLOR(1,2,16) : NEXT I
370 FOR I=10 TO 17 STEP 6 : DISPLAY AT(1,20):CHR
$(136):RPT$(CHR$(131),7):CHR$(134) : FOR J=1
380 DISPLAY AT(J+1,20):CHR$(133) : " : CHR$(1
33) : NEXT J : DISPLAY AT(I+4,20):CHR$(130) :
RPT$(CHR$(131),7):CHR$(129) : NEXT I
390 DISPLAY AT(11,24-(LEN(N1$)/2))SIZE(LEN(N1$)):
N1$ : DISPLAY AT(17,24-(LEN(N2$)/2))SIZE(LEN
(N2$)):N2$
400 SUBEND
410 SUB TERNING(SUM(),TUR,N1$,N2$,TID,T)
420 FOR W=1 TO 9 STEP 4
430 FOR A=1 TO INT(RND*6)+1
440 B=INT(RND*6)+1
450 ON B GOSUB 490,500,510,520,530,540
460 NEXT A
470 NEXT W
480 GOTO 550
490 TERN(INT(W/4)+1)=1 : CALL CHAR(115+W,RPT$("F
",13)&"EFCFCFE"&RPT$("F",24)&"7F3F3F7F"&RPT$("
F",13)) : RETURN
500 TERN(INT(W/4)+1)=2 : CALL CHAR(115+W,"FFFFF
F3E1E1F3"&RPT$("F",36)&"CF87B7CFFFFFFF") : RE
TURN
510 TERN(INT(W/4)+1)=3 : CALL CHAR(115+W,"FFFFE7
C3C3E7FEFCFCFE"&RPT$("F",24)&"7F3F3F7FE7C3C3E
7FFFF") : RETURN
520 TERN(INT(W/4)+1)=4 : CALL CHAR(115+W,"FFFFF
F3E1E1F3FFFFF3E1E1F3"&RPT$("F",12)&"CF87B7CFF
FFCF87B7CFFFFFFF") : RETURN
530 TERN(INT(W/4)+1)=5 : CALL CHAR(115+W,"FFFFE7
C3C3E7FEFCFCFE7C3C3E7FFFFF7C3C3E77F3F3F7
FE7C3C3E7FFFF") : RETURN
540 TERN(INT(W/4)+1)=6 : CALL CHAR(115+W,"FF"&R
P$("E7C3C3E7FF",3)&"FF"&RPT$("E7C3C3E7FF",3)&
"00") : RETURN
550 REM NU SVARAR DU!
560 DISPLAY AT(6,21) : "(1*1)*1"
570 CALL INPUT(A$,B$,C$,SLUT,TID,T)
580 IF A=0 THEN 960
590 FOR D=1 TO 3
600 IF A=TERN(D) THEN 640
610 NEXT D
620 FOR WAIT=1 TO 10 : DISPLAY AT(6,22):"FEL TAL
" : DISPLAY AT(6,21) : "" : CALL SOUND(-10,WA
IT*10+110,0) : NEXT WAIT
630 IF T=TID THEN T=0 : GOTO 960 ELSE 560
640 FOR E=1 TO 3
650 IF E=D THEN 670
660 IF B=TERN(E) THEN 690
670 NEXT E
680 GOTO 620
690 FOR F=1 TO 3
700 IF F=D OR F=E THEN 720
710 IF C=TERN(F) THEN 740
720 NEXT F
730 GOTO 620
740 REM FACIT BERAKNAS
750 IF A$="" THEN FACIT=A*B
760 IF A$="/" THEN FACIT=A/B
770 IF A$="+" THEN FACIT=A+B
780 IF A$="-" THEN FACIT=A-B
790 IF B$="-" THEN FACIT=FACIT-C
800 IF B$="+" THEN FACIT=FACIT+C
810 IF B$="/" THEN FACIT=FACIT/C
820 IF B$="*" THEN FACIT=FACIT*C
830 IF FACIT=INT(FACIT) AND FACIT>0 AND FACIT<43 T
HEN 860 ELSE IF SLUT=1 THEN 960
840 DISPLAY AT(8,21):"TALET AR" : DISPLAY AT(9,2
1):"06ILTIGT" : FOR W=1 TO 700 : NEXT W :
DISPLAY AT(8,20) : "" : DISPLAY AT(9,21) : ""
850 GOTO 560
860 RAD=INT((FACIT+5)/6)*3
870 SPALT=((FACIT+6)-(RAD*2))*3+1
880 CALL GCHAR(RAD,SPALT,CH) : IF CH=128 THEN 840
890 CALL HCHAR(RAD,SPALT,128,2) : CALL HCHAR(RAD+
1,SPALT,128,2)
900 FOR V=RAD-2 TO RAD+3 STEP 2.5
910 FOR H=SPALT-2 TO SPALT+3 STEP 2.5
920 CALL GCHAR(V,H,CH)
930 IF CH=128 THEN P=P+1
940 NEXT H
950 NEXT V
960 IF TUR=1 THEN TUR=2 ELSE TUR=1
970 ON TUR GOSUB 980,1010 : P=0 : GOTO 1040 :
980 FOR I=1 TO P : SUM(1)=SUM(1)+1 : DISPLAY AT
(13,21)SIZE(-7):I:SUM(1) : CALL SOUND(50,440,
I) : NEXT I : DISPLAY AT(1,21):N2$
990 IF SUM(1)>49 AND SEG<>1 THEN CALL FANFAR : S
EG=1

```

```

1000 RETURN
1010 FOR I=1 TO P : SUM(2)=SUM(2)+1 : DISPLAY AT
(19,21)SIZE(-7):I:SUM(2) : CALL SOUND(50,440,
I) : NEXT I : DISPLAY AT(1,21):N1$
1020 IF SUM(2)>49 AND SEGER<>1 THEN CALL FANFAR :
SEGER=1
1030 RETURN
1040 P=0 : SUBEND
1050 SUB VINJETT
1060 CALL CHAR(35,RPT$("F",16)&"80C0E0F0F8FCFEFF")
1070 CALL CHAR(37,"0103070F1F3F7FFF",38,"FFFEFCFB
F0E0C0B0",34,"FF7F3F1F0F070301")
1080 CALL HCHAR(3,12,37) : CALL HCHAR(3,13,35,2)
1090 FOR I=4 TO 10 : CALL HCHAR(I,15-I,37) : CALL
HCHAR(I,16-I,35,2) : CALL HCHAR(I,18-I,38) :
NEXT I
1100 CALL HCHAR(4,14,35) : FOR I=5 TO 17 : CALL V
CHAR(11,I,35,2) : NEXT I
1110 FOR I=5 TO 15 : CALL HCHAR(I,11,35,4) : NEXT
I
1120 CALL HCHAR(7,18,35,3) : CALL HCHAR(6,18,35,3)
: CALL HCHAR(7,23,37)
1130 FOR I=1 TO 3 : CALL HCHAR(I+2,21-I,37) : CAL
L HCHAR(I+2,22-I,35,I*2+2) : CALL HCHAR(I+2,I
+24,36) : NEXT I
1140 CALL HCHAR(6,24,35,4) : CALL HCHAR(7,24,35,4)
: FOR I=1 TO 5 : CALL HCHAR(7+I,23-I,37) :
CALL HCHAR(7+I,24-I,35,4)
1150 CALL HCHAR(7+I,28-I,38) : NEXT I : CALL HCHA
R(13,18,35,5) : FOR I=18 TO 27 : CALL VCHAR(
14,I,35,2) : NEXT I
1160 CALL HCHAR(13,25,35,3)
1170 W=200
1180 CALL SCREEN(5) : CALL COLOR(1,11,1) : FOR I=2
TO 12 : CALL COLOR(I,16,1) : NEXT I
1190 FOR I=17 TO 22 STEP 2
1200 CALL HCHAR(I,5,35,23)
1210 NEXT I : FOR I=1 TO W : NEXT I
1220 FOR I=8 TO 24 STEP 4
1230 CALL VCHAR(17,I,32,5)
1240 NEXT I : FOR I=1 TO W : NEXT I
1250 FOR I=5 TO 26 STEP 4.12
1260 CALL VCHAR(17,I,35,5)
1270 NEXT I : FOR I=1 TO W : NEXT I
1280 CALL HCHAR(18,7,32) : CALL HCHAR(20,5,32) : C
ALL HCHAR(20,7,35) : FOR I=1 TO W : NEXT I
1290 CALL VCHAR(18,11,35) : CALL HCHAR(20,11,32) :
CALL HCHAR(21,10,32,2) : FOR I=1 TO W : NEX
T I
1300 CALL VCHAR(17,18,32,4) : CALL VCHAR(17,19,32,
4)
1310 FOR I=1 TO W : NEXT I
1320 CALL VCHAR(18,25,32,4) : CALL VCHAR(18,27,32,
4)
1330 DISPLAY AT(24,23):"VANTA"
1340 SUBEND
1350 SUB NAMN(N1$,N2$,TID)
1360 DISPLAY AT(24,1):"NAMN SPELARE 1?" : ACCEPT
SIZE(7)AT(24,18)VALIDATE(" ABCDEFGHIJKLMNOPQR
STUVWXYZAa01234567890-/") : N1$
1370 DISPLAY AT(24,1):"NAMN SPELARE 2?" : ACCEPT
SIZE(7)AT(24,18)VALIDATE(" ABCDEFGHIJKLMNOPQR
STUVWXYZAa01234567890-/") : N2$
1380 IF N1$="" THEN CALL NEWNAME(N1$)
1390 IF N2$="" THEN CALL NEWNAME(N2$)
1400 IF N1$=N2$ THEN CALL NEWNAME(N2$) : GOTO 1400
1410 DISPLAY AT(23,1):"BETANKETID: 1. 30 SEK
2. 2 MIN 3. 1 MILJ AR 2"
1420 ACCEPT AT(24,26)BEEP VALIDATE("123")SIZE(-1):
TID
1430 IF TID=1 THEN TID=150 ELSE IF TID=2 THEN TID=
600 ELSE TID=1.57784E+14
1440 SUBEND
1450 SUB NEWNAME(A$)
1460 ON INT(RND*7)+1 GOSUB 1480,1500,1520,1540,156
0,1580,1600
1470 SUBEXIT
1480 A$="OKAND"
1490 RETURN
1500 A$="NAGON"
1510 RETURN
1520 A$="ANONYM"
1530 RETURN
1540 A$="NOVISEN"
1550 RETURN
1560 A$="NOLLAN"
1570 RETURN
1580 A$="ZERO"
1590 RETURN
1600 A$="SOMEONE"
1610 RETURN
1620 SUBEND
1630 SUB SVENSKABDKST
1640 CALL CHAR(91,"28003B44447C44442B003B444444443
B10003B44447C4444")

```

```

1650 SUBEND
1660 SUB INPUT(A,A$,B,B$,C,SLUT,TID,T)
1670 CAR(1),CAR(3),CAR(5)=49 :: CAR(2),CAR(4)=42 ::
: P,PL=1
1680 GOSUB 1890 :: CALL KEY(O,K,Y):: IF K=13 THEN
1800
1690 IF K=9 THEN P=P+1 :: PL=PL+1 :: CALL SOUND(-1
,660,4)
1700 IF K=9 AND P=6 THEN P=5 :: PL=6
1710 IF K=9 AND PL=4 THEN PL=5 :: GOTO 1680
1720 IF K=8 THEN P=P-1 :: PL=PL-1 :: CALL SOUND(-1
,660,4)
1730 IF K=8 AND P=0 THEN P=1 :: PL=1
1740 IF K=8 AND PL=4 THEN PL=3
1750 IF K=8 OR K=9 THEN 1680
1760 IF P/2=INT(P/2) THEN 1790
1770 IF K<48 OR K>54 THEN 1680 ELSE CAR(P)=K :: GO
SUB 1890 :: P=P+1 :: PL=PL+1 :: CALL SOUND(-1
,660,4):: IF PL=7 THEN PL=6 :: P=5
1780 IF PL=4 THEN PL=5 :: GOTO 1680 ELSE GOTO 1680
1790 IF K<42 OR K>47 OR K=44 OR K=46 THEN 1680 ELS
E CAR(P)=K :: GOSUB 1890 :: P=P+1 :: PL=PL+1
:: CALL SOUND(-1,660,4):: GOTO 1680
1800 REM <ENTER>
1810 FOR I=1 TO 8 :: CALL SOUND(-40,110+20*I,0):: N
EXT I
1820 A=CAR(1)-48
1830 B=CAR(3)-48
1840 C=CAR(5)-48
1850 A$=CHR$(CAR(2))
1860 B$=CHR$(CAR(4))
1870 SUBEXIT
1880 REM MARKÖREN BLINKAR...
1890 CALL HCHAR(6,PL+23,30):: CALL HCHAR(6,PL+23,C
AR(P)):: T=T+1 :: IF T=TID THEN SLUT=1 :: GOT
O 1800
1900 RETURN
1910 SUBEND
1920 SUB FANFAR
1930 FOR J=1 TO 16
1940 CALL SCREEN(J):: CALL SOUND(60,110+20*J,0,110
+40*J,2,110+80*J,4)
1950 NEXT J
1960 SUBEND

```

Obs! Det tyska öret på rad 1810 skall vara upphöjt till (skift-6).

ANNONSER

Säljes:

MULTIPLAN för TI-99/4A 260:-
(modul+disk+manual på 240 sidor)
SPEECH SYNTHESIZER 260:-
PRINTER SEIKOSHA 550A parallellgränssnitt PIO
(begär utskriftsprov mot returporto)

Jan Alexandersson
Springarvägen 5, 3 tr
142 00 TRÅNGSUND
Tel 08/771 05 69

Säljes:

Extended BASIC 600:-
Paint 'n Print (grafikmodul) 600:-
Video Chess, Pirate Adventure 200:-/st
Speech Synthesizer 500:-

Mattias Andersson
Tel 0753/336 70

Säljes:

TI-99/4A + exp-box, SSSD diskdrive + 32K RAM, 96K
RAM-disk m batteri-backup, Parsec, Pole Position,
X-BASIC, E/A-modulen, 40 disketter med nyttoprogram
och spel i X-BASIC och assembler.

Div böcker och disketter med kurser i assembler-program-
mering.

Prisidé: 4 500:- eller högstbjudande.

Miguel Breien
Lekplatsvägen 3
172 32 Sundbyberg
Tel 08/98 24 43 efter 18
08/737 85 90 07-11

Köpes:

Videogames I på modul
Conny Bonet
Nyponstigen 44
152 00 Strängnäs
Tel 0152/175 62 efter kl 17.00

Disk Manager 1000 version 3.5

av Jan Alexandersson

Det har kommit ytterligare en ny version av DM 1000. Vid filhanteringen finns två nya val nämligen:

T (TYPE)

P (PRINT)

TYPE gör det möjligt att direkt från katalogen välja vilken DIS/VAR 80- eller DIS/FIX 80-fil som skall listas på skärmen. Det finns en liten BUG i denna rutin som gör att TYPE inte fungerar om filens första rad är blank. Du får detta om du t ex listar ett BASIC-program till disk.

PRINT fungerar som TYPE men listningen sker mot printer.

En annan förbättring är att buggen med DELETE av fil med 1 sektor har rättats.

Den version av 3.5 som vi nu kan sprida inom föreningen har originalversionen av PROGRAM- och TEXT-filerna men vi saknar den riktiga LOAD-filen. Den vi sänder med kommer från DM 1000 V 2.2. Detta gör att RUN PROGRAM FILE inte fungerar riktigt eftersom DM 1000 letar i LOAD-filen efter ett teckenst. Om det finns någon som har den riktiga filen så sänd in den till föreningen.

Ett annat problem som finns i samtliga versioner av DM 1000 är att du inte kan sända kontrollkoder till printer som är större än 99. Du kan i stället skriva in detta med en sektoreditor t ex DISK-PATCH (som följer med FUNNEL-WEB) eller ADVANCED DIAGNOSTICS. Leta upp sektorn med filhuvudet till MGR1 och editera sektorn.

Vid INPUT av nytt skivnamn kan du krascha programmet om (DEL) används när du är i teckenposition 11. Det blir även konstiga streck på skärmen om du använder (FCTN) S från position 11. Detta problem finns i alla versioner av DM 1000 men märks mer när INPUT-fältet ökas till 11 tecken.

Du kan inte använda DM 1000 för att kopiera TI-FORTH-skivor med filkopiering. Ett exempel på detta är UNIVERSAL DISASSEMBLER som vi har som FREEWARE. Denna har följande katalog:

FORTH	6	DIS/FIX 80
FORTHSAVE	39	PROGRAM
SYS-SCRNS	313	DIS/FIX 128

Om denna skiva kopieras med DM 1000 så kommer alla sektorer mellan 4 och 33 att komma fel. Detta motsvarar SCREEN 1-8. Detta beror på att DM 1000 lagrar filhuvudet sist (sektor 33) medan DM 2 lagrar detta först (sektor 4). Om du använder DISK-kopiering med DM 1000 blir dock kopieringen riktig.

De problem som jag beskrivit här och i tidigare nummer av Programbiten är ej allvarliga utan DM 1000 är utan jämförelse den bästa DISK MANAGER som finns. Alla som har diskdrive bör skaffa den. Vi distribuerar den genom föreningen för 50:-. Du får då en skiva samt hela den tjocka manualen utprintad på papper.

Ny räknare från TI

Som en uppföljare till 74-an som vi presenterade i förra numret har TI nyligen börjat sälja TI-95 Procalc. Det är en avancerad programmerbar räknare. Den arbetar med sex olika displayfönster (se exempel på bilden nedan). På Stor & Liten i Stockholm kostade den 2 295:-. Tillbehören till 74-an passar förstås till 95-an också.



Pressgrannar

TI-Revue heter en tysk tidning som kommer ut varje månad. Den handlar helt och hållet om TI-99/4A. Tidningen ges ut av ett kommersiellt förlag och är professionellt utformad (även om den trycks på ett enklare papper än Programbiten). Numret som avbildas nedan var på 64 sidor, vilket nog är normalt. Innehållet består till stor del av programlistningar, ofta av hög kvalitet. Förutom spel kan man också hitta listningar för nyttoprogram, ofta i assembler. Artiklar om byggen till 99-an brukar det finnas en hel del av. Tyskarna verkar väldigt aktiva när det gäller att bygga tillbehör. De produkter som tagits fram av tyska företag till 99-an finns också recenserade.

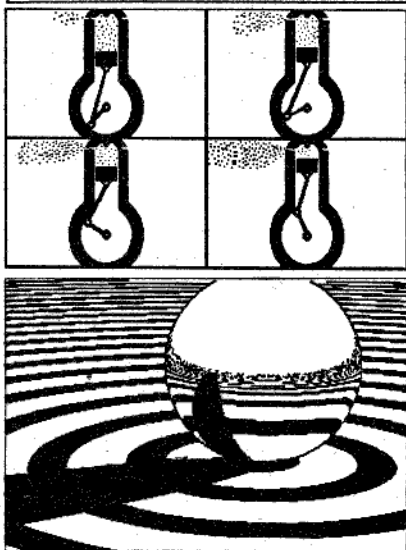
Prenumerationspriset är 80 DMark för 12 nummer.



Nr. 8/86-August

-DM 5,50 / ÖS 46 / BRP 5,50

BAUEN SIE SICH DOCH IHREN PLOTTER SELBST (II)



**DIALOG
TEST &
TECHNIK
ASSEMBLER
TASTATUR-
ABFRAGE
JOYSTICK-
PROPOR-
TIONAL-
STEUERUNG
LISTINGS
TIPS
& TRICKS**

Tidningens adress:

TI-REVIEW
Postfach 1107
D-8044 Unterschleißheim
Västtyskland

RAM-diskbygget

Det bygge som vi nämnt i PB ett par gånger tidigare fortskrider. Korten är så gott som klara, nu väntar vi bara på hållare till batterierna. Dessa har visat sig vara svåra att få tag på. En utförlig rapport får därmed anstå till nästa nummer av tidningen.

En fördel med Horizons RAM-disk är att programvaran till den är lätt att byta eftersom den ligger i RAM. Ibland kan dock operativsystemet "ramla ur", speciellt om batterierna inte håller laddningen. En annan amerikansk firma har nu tagit fram ett EPROM som ersättare. Det ska, enligt leverantören *Genial Computerware*, vara förbättrat jämfört med Horizons operativsystem.

Fel i PB-versionen av TI-Writer

av Bo-Arne Östborg

För ett par månader sedan köpte jag en ny skrivare till 99-an, Star NL-10, en liten japan, som gör det mesta bara man skickar ett antal styrkoder inbakat i texten. Med en sådan skrivare behöver man normalt ej använda utskriftsprogrammet FORMAT i TI-WRITER utan utskriften görs direkt från redigeringsprogrammet med kommandot SkrivFil. TI-WRITER tillåter ju även skrivning av ASCII-koderna 0-31 i texten med visning med särskilda tecken på skärmen.

Men några funktioner, som innehöll ASCII-kod 0 fungerade ej. Eftersom de fungerade från BASIC-program, måste felet finnas i utskriftsprogrammet. Skrivaren har en hex-dump-funktion, som visade, att det tecken, som helt riktigt på skärmen presenteras som ASCII 0, skickas till skrivaren som ASCII 65, dvs A. Den amerikanska originalversionen av TI-WRITER fungerade i detta avseende riktigt. Följaktligen måste felet ha kommit med filerna EDITA1 och EDITA2 på uppdateringsdisketten från PB.

Disassemblering av filerna gav efter många timmars studium svaret. I utskriftsdelen av redigeringsprogrammet har någon ny funktion lagts till, varför de inledande kodraderna stuvats om. Ett villkorligt hopp låter teckenkod 0 falla igenom till ett avsnitt, som definierar om tecken. Jag vet ej om detta avsnitt verkligen används. Men jag har ändrat hoppinstruktionen från JGT till JHE med gott resultat. Tecken 0 sänds nu till skrivaren och jag har ej funnit att någon annan del av programmet skulle ha ändrats.

Jag har gjort ändringen med Anders Törnkvists SEKTOREDIT-program i FORTH så att byte >28 i sektor 110 ändrats från >15 till >14. Men det finns kanske andra versioner med annan sektorfördelning, varför aktuellt kodavsnitt redovisas.

Adr	Kod	Mnemonics	
3AFC	020E	LI R14,1	
	0001		
3B00	C3CE	MOV R14,R15	
3B02	020D	LI R13,>A148	
	A148		
3B06	D820	MOVB @>21E9,@>38C9	Blanktecken (>20)
	21E9		
	38C9		
3B0C	0760	ABS @>38EE	Flagga för undertryckning av styrkod
	38EE		
3B10	1603	JNE >3B18	
3B12	981D	CB \$R13,@>21E9	Jmfr med ASCII >20
	21E9		
3B16	1A0C	JL >3B30	
3B18	0760	ABS @>38EA	Flagga för SkrivFil
	38EA		
3B1C	130F	JEQ >3B3C	
3B1E	04C9	CLR R9	
3B20	D25D	MOVB \$R13,R9	
3B22	1508	JGT >3B34	Här slinker 0 igenom
3B24	0249	ANDI R9,>7C00	
	7C00		
3B28	09A9	SRL R9,10	
3B2A	D829	MOVB @>38CA(R9),@>38C9	Hämtar tecken ur en tabell. Första tecknet är >41 (=A)
	38CA		
	38C9		
3B30	D760	MOVB @>38C9,\$R13	Ersätter skärmens tecken med annat
	38C9		
3B34	058D	INC R13	
3B36	060F	DEC R15	
3B38	15E6	JGT >3B06	
3B3A	1003	JMP >3B42	

Den adress, som skall ändras, är 3B22:

3B22 1408 JHE >3B34

Med tillgång till källkod skulle kanske en bättre revidering kunna göras. Kanske något för Anders Persson?

Programbanken – uppdaterad lista

Här visas en lista på de tillgängliga programmen i PROGRAMBANKEN. Programmen kan köpas! Priset är uppdelat i **två delar**, dels en startkostnad som täcker kostnaden för porto och datamediet, dels en kopieringsavgift för **varje** program.

KASSETT Startkostnad 30:- per kassett
Kopieringsavgift 5:- per program

DISKETT Startkostnad 30:- per diskett
Kopieringsavgift 5:- per program

OBS! Var noggrann när DU anger programnumret. Det är det jag går efter vid kopieringen

Som förut får DU tre program i utbyte när DU skickar in ett program **DU SJÄLV HAR GJORT!**

Ange om ditt program är skrivet i TI-Basic, Extended Basic, Assembler, Pascal eller i Forth. Beskriv noggrant vilken utrustning som behövs (även i själva programmet), hur man kör programmet och vad som behöver göras för att programmet skall fungera. Skriv även vilken modul som erfordras till programmet.

Kodförklaring

De fyra första siffrorna är interna beteckningar (diskett-nummer, programkategori).

Femte siffran anger ursprungsland:

1 = England, USA 2 = Holland, Belgien, Tyskland
3 = Sverige 7 = Finland

De tre sista siffrorna är ett internt löpnummer.

Bokstaven, som finns sist, anger vilket språk som används i programmet

E = Engelska, F = Franska, H = Holländska

Om bokstav saknas så är ledtexten på svenska eller så finns det ingen ledtext.

Dessutom kan ett D finnas allra sist i programnumret och det anger att datafil(er) ingår.

Eventuella bokstäver efter programnumret talar om vilken utrustning som krävs.

C = CALL FILES(1) måste användas om man har diskett-enhet.

D = Diskettenhet
F = Forth
M = MiniMemory
P = Printer
J = Joystick
T = Terminal
Emulator II

E = Minnesexpansion
A = Editor/Assembler
X = Extended-Basic
S = Speech Synthesizer
(pratorn)
K = Personal Record
Keeping

Programkategorier

02 = Operativsystem
10 = Registerprogram
12 = Ordbehandling
14 = Ekonomi
20 = Regression, Kurvanpassning
21 = Statistik, Varians
29 = Statistik, Sannolikhet
30 = Linjär algebra
39 = Allmän matematik
65 = Elektronik
78 = Astronomi
87 = Assemblerrutiner
88 = Assemblerprogram
89 = Subrutiner, subprogram
90 = Basicprogram
91 = Spel
92 = Utbildning
96 = Musik
97 = Demo
99 = Övrigt
29 = Statistik, Sannolikhet
30 = Linjär algebra
39 = Allmän matematik
65 = Elektronik
78 = Astronomi
87 = Assemblerrutiner
88 = Assemblerprogram
89 = Subrutiner, subprogram
90 = Basicprogram
91 = Spel
92 = Utbildning
96 = Musik
97 = Demo
99 = Övrigt

02 **OPERATIV SYSTEM**

10 **REGISTERPROGRAM**
01101001E —X Släktforskningsregister
01101002E — Adressregister
11103069 — Fylläsning på Kasset
11103072 —K Snabbfil till Personal Record
Keeping modulen. PB 85-2
14103045 PDX Telefonregister. PB 84-4
15103092 -DX Tel-Ad. Telefonadresser
17103012 — Nummersändaren. PB 83-4/5

12 **ORDBEHANDLING**
01121001E —P Word-processor
10121010E —C Ordbehandling för TP-printer
11123064 — P:TEXTIN. PB 84-3
11123065 — P:TEXTUT, Lista P:TEXTIN filer. PB 84-3
14123039 —P Nova-Verba
16123007E —X LT-Miniwriter. PB 85-1
17123006 — P:KOPIERA. PB 86-2

14 **EKONOMI**
07141001H —X Annuities
12143006 — Ränteberäkningar
17143015 —K Kontoprogram. Upp till 22 konton.

20 **REGRESSION, KURVANPASSNING**
01201001E — Uträkning av en linjes ekvation efter ett antal punkter fördelade kring en linje
01201002E — Uträkning av riktningskoeff

21 **STATISTIK, VARIANS**
06211001E —C Beräkning av standardavvikelse

29 **STATISTIK, SANNOLIKHET**
09291001H — Statistiska kombinationer etc

30 **LINJÄR ALGEBRA**
06301001E — Matris invertering/multiplikation
14303050 —X Lösning av linjära ekvationssystem med Gauss-eliminationsmetod

39 **ALLMÄN MATEMATIK**
01391001E — Integral, derivata, andragradsekv reella rötter, cp analys, definiera chars, 3-D plot, fakultet och primtalstest
01391002E — Lösning av ekvationssystem
01392001E — Division med valfritt antal decimaler i svaret
09391003H — Differential eq

11393071 — Rita sinuskurva. PB 85-2
 12393005 — Matrisberäkningar, lösning av ekvationssystem
 14393051 —X Math-Pac, Integration, Newton-Raphson, Intervallhalvering, Romberg-integration

65 **ELEKTRONIK**
 01651001E — Beräkning av komponentvärden för resistiv parallellkoppling, kondensator i serie, resonans för spole och kondensator, omvandling frekvens - våglängd, ohm's lag, uträkning av antennlängd
 16653004 — Ohms lag, räknar ut ampere, ohm, volt och watt. PB 85-3

78 **ASTRONOMI**
 01781001E — Beräkning av geostationära satelliters positioner
 15783000 —X Trippelpunkten på 99:an, Tyngdpunkten till en sfärisk triangel. PB 85-1
 16783012 — Stjärnkikeri. PB 86-2

87 **ASSEMBLERROUTINER**
 15873002 EAD K:INPUT, subprogram för inmatning av data från tangentbordet. Källkod. PB 86-2

88 **ASSEMBLERPROGRAM**
 16883011 EAD CSAVE, minnesdumpning till kassett. PB 85-2
 18883018 PEDX Screen Dump. Objektкод.
 18883019 PEDX Screen Dump. Källkod till ovanstående. PB 86-4

89 **SUBROUTINER, SUBPROGRAM**
 12893055 — Screen-Ram. PB 84-5
 13891057 —EX Cursor, ändra utseendet på cursorn
 13891058 — Ensa-komma. PB 84-5
 14893034 — Teckendefinierare
 15893097 — Snabbare grafik med TI-Basic. PB 84-1
 16893003 —EX Miniassembler. Utmaningen 85-2
 16893014 — Datain- utmatning kassett. PB 85-3
 17893019 — Shellsort. 99'an 83-4/5
 17893020 —X Quicksort. 99'an 83-4/5
 18893003 — Riktiga små bokstäver. 99'an 83-4/5
 18893005 — "DISPLAY AT" i TI-Basic. 99'an 83-1
 18893006 — Högerjusterade tal. 99'an 83-3

90 **BASICPROGRAM**
 06901003E —X Sortering och utskrift från flera disketter av upp till 300 program
 06901004E — Diskkatalog med utskrift
 07902002E —P Banner, stor text utskriven på TP-printer
 08901006E PEX Utskrift av titelsida för ex listningar
 08902003E — Stora tecken för 12x28 teckenskärm
 10901008E — Very large characters
 11901011 — Stapeldiagram
 11903063 PDX Lista Basicprogram med Basicprogram. PB 84-4
 11903066 —PX PGM LIST59, Lista TI59 program på TI99/4A. PB 84-2
 11903067 — Screen-Saver, spara skärmen. PB 84-3
 11903068 — Testbild för Din Dator. PB 84-3
 11903075 —DX LISTA ORD. Listar DIS/VAR 80-filer med 55 tecken/rad. PB 85-1
 12901054 —JX Color-Draw, Ritprogram med färger. PB 84-5
 13903022 —DX Fil-Data-Editor, för DIS/VAR 80 filer
 13903027 —X Grafik Exklusive, Kassettbaserat
 13903060 PDX List28, Lista program 28tkn/bredd. PB 85-3
 13903062 —DX P:TILLCOMP, överför P:TEXTIN filer till Companion format. PB 84-4
 14901053 —X Talkonvertering, Hex-Dec-Binärt. PB 84-5
 14902052 —X Färg-Editor. PB 84-5
 14903042 — Kassettinnehåll. PB 84-4

15901089E —DX Merge/Read, listar DIS/VAR 163 filer
 15901090E —EX Mg/Pecker, listar minnes-adresser
 15903005 — Baskonvertering, Hex-Dec-Bin. PB 84-1
 15903076 PDX Listprogram, Listar Basicprogram
 15903077 —SX Trunkering, skapa egna ord med trunkering. PB 84-4
 15903088 —EX List-Size, listar storleken på dina program
 15903093 —DX Char-Adapt, Teckenändring, lagras som DIS/VAR 80 filer
 15903094 —JX Seppo Huikuris Sprite-Maker. PB 84-1
 15903095 EDX Load-Assembler-Bas, omformar Assembler program till Extended-Basic program. PB 84-1
 15903098 —JX Tommy Erikssons Sprite-Maker. PB 86-1
 16901015E —EX Generell programstartare. PB 85-1
 16903006 —DM Diskkatalog i Textmode. PB 84-4
 16903016 —D CATALOG. Listar diskettkatalogen på skärmen. PB 85-1
 16903018 —X Polsk notation. PB 84-2
 17903004 —M Mini-ASMBAS. PB 86-2
 17903011 —JX Kalkylator med 99:an. PB 86-1

91 **SPEL**
 01911028H — Car driver
 01911029H — Animals
 01911030H — Space Invaders
 02911001E — Master Mind
 02911002E —X Prickskytte med kanon mot en kyckling
 02911003E — Startrek, textadventure
 02911006E — Othello
 02911007E — Robotjakt
 02911008E — 15-spel
 02911009E — Täarningsspel
 02911011E — Keno
 02911012ED —D Star Guard
 03911013E — Miner
 03911014E —C Yahtzee
 03911015E — Backgammon
 03911016E — 3D Tic-tac-toe
 03911017E —X Black-Jack II
 03911018E —SX Not one
 03911019E —X Datorpoker
 03911024E — Fyra i rad (luffarschack)
 04911025E — Startrek
 04911026E —X Hjälp kycklingen över vägen
 04911027E — Katapult
 04911031E — ISOLA. Isolera motståndaren
 06911032E — Vem skjuter den sista roboten
 06911034H —X L-game
 06912004E — Slalom
 06912005E — Killer
 07911035H —X Wari
 07911036H — Ta dig fram i en osynlig labyrint
 07911037H —X Eliza
 07911040H —X Black jack
 07911041H —X Space battle
 07911043H — Enarmad bandit
 07911045E — Labyrint. 2 spelare
 08911039E —X Sprite-jakt
 08911046E —X Damspel
 08911049E —X Swords & sorcery, textadventure
 08911051E —X Ägg-fångst
 09911052E — Deep space
 09911054H — Planetary lander
 09911056H —X Lunar lander
 09911057F — Towers of Hanoi
 09911058E — Breakout, bollspel squashtyp
 09911059H —J Ritprogram i olika färger
 09911060H —JX Treasure hunt, labyrintspel
 09912008F —X Asteriod (tal)
 09912009E — Battlestar
 10911061H —X Kermit
 10911062H —JX Find the gun
 10911063E — Matematikspel
 10911065E — Camel, äventureritt i öknen
 11911055 —X Hängning

11917001 -JX Inkräktarna
 11917002 -JX Pärön
 11917003 -JX Masken
 12911001 -X One Check, brädspel
 12913002 -JX Jockes Hostility
 12913004 -X Giljotinen, ordlek
 12913007 -Tjugoett
 12913009E -Octopus
 12913012 -The Hunt of Aladin
 12913013 -Bowling
 12913014 -Bilrally i Norge
 12913015 -Masken II
 12913016 -Djungelfight
 12913017 -JX Snake, Ormen
 12913018 -JX Zombie
 13913019 -J Kami-Kaze
 13913020 -JX Ubåt
 13913023 -Jumping Jack
 13913024E -Astrostorm
 13913025 -Goblin
 13913026E -Identify the States
 14911037E -X One Check II
 14911046E -X Pricka flygplan
 14911048E -X Spacegame
 14913035 -X Skroting
 14913036 -JX Starport 99
 14913040E -J Blockman
 14913044 -X Mastermind
 15913079 -X Adventurespel, Gör dina egna
 Adventurespel. PB 84-3
 16913001 -Helikopter. PB 86-2
 16913002 -X Lander, labyrintspel. PB 85-4
 16913003 -X Dart. PB 85-4
 16913005 -J Ormar, (liknar Masken). PB 85-4
 16913008 -JX NUC, Atombombspel. Försvara sta-
 den. PB 85-4
 16913013 -X TOTO-G. Galoppspel. PB 85-2
 17913001 -Hangman. PB 83-4/5
 17913002 -Sifferjakt. PB 83-4/5
 17913005 -M Masken för Mini Memory. PB 86-2
 17913010E -X Korsord. PB 86-1
 17913016 -Earth Attack. 99'an 83-1
 17913017 -J Dominion. 99'an 83-2
 17913018 -JX Jagad. 99'an 83-4/5
 18913001 -J Pucman. 99'an 83-3
 18913002 -JX Stockholm Competition. 99'an 83-3
 18913004E -One-Check. 99'an 83-2
 18913011 -JX SHARK. Ta dig till bryggan utan
 att ätas av hajarna
 18913017E -JX Space Bonus. Skjut ner rymdskeppet
 och skaffa dig bonuspoäng

92 UTBILDNING

01921006H -X Arvsanlag
 04921001E -Enkla räkneövningar med de fyra
 räknesätten
 04921002E -Kemifrågor
 04921003E -X Relative IQ-test
 04921004E -Algebra
 04921005E -Projectile problems, beräkn.-spel
 07921007E -Träning i dåtid (imperfekt)
 10921008E -Räkneträning med bråkdelar
 13923029 -X Huvudräkning
 13923030 -X Glosträning, kassettbaserad
 13923031 EDX Byggnadsmekanik
 16923009 EDX Glosträning, diskbaserad
 18923009D -JX Sverige runt. Placera kommunena
 rätt på kartan. PB 86-3
 18923010 -W-ORD-S1. Övning av engelska glosor.
 Endast på kassett
 18923012 -W-ORD-S2. Övning av engelska glosor.
 Något svårare än W-ORD-S1. Endast
 på kassett

96 MUSIK

05962001 -C Johann Strauss. Tales from
 Vienna woods
 05962002 -Never on a sunday
 05962003 -X Berceuse
 -05962004 -X Beethoven. Symfoni nr 5 -
 05962005 -X Serenade
 05962006D -DX Amazing graze

05962007 -CX Beethoven opus 27
 05962008D -DX Time in a bottle
 05962009D -DX You light up my life
 05962010 -Bummle-boogie
 06962001 -The pink panther
 06962006 -Let me call you sweetheart
 06962011D -D Fiddler on the roof
 06962012 -Bach 3. menuet
 07962013 -I'm looking through you
 08961001E -X Killing me softly
 08962014 -X Chopin op 11 3
 12963056 -Tremolo. PB 84-5
 14963033 -TI-Keybord
 14963043E -When the Saints ...
 16963019 -Ta det piano. PB 84-2
 17963003 -Musicbox. Melodierna Bä bä vita lamm,
 Vedhuggaren, Blinka lilla stjärna och
 London bridge
 17963008 -X Sätt musik till dina X-Basicprogram.
 PB 86-2
 17963009 -X Inledningen till Gloriasatsen "Iste
 confessor" av Palestrina. PB 86-2
 18963014 -X ELORGEL.
 18963016 -Keyboard. Spela på tangentbordet.
 Melodin kan sparas. Endast på
 kassett.

97 DEMO

01971001H -C Demo av färger, grafik och ljud
 06971002E -Kaleidoscope
 06972004 -X Halloween (sprites)
 06972005 -X Sprites
 07971003H -Demonstration av olika tecken-
 storlekar
 11973070 -EX Demoladdning av Assembler program
 13973059 -Korsord
 14973038 -X Mönstergrafik
 15971091 -EX Mg/Text, Textmode
 15973003 -M Textmode i TI-Basic. PB 84-4
 15973004 -M Auto-Sprite i TI-Basic. PB 84-4
 15973099 -X Train. Demonstrerar doppler-effekt.
 PB 85-2
 16973010 EAD Tapetest. PB 85-2
 17973007 -X Sprites i cirklar. PB 86-2
 17973013 -DX SORTDEMO. Demo av Quick-, Fast-,
 och Bubblesort. PB 86-3
 17973014 -X SORTDEMO. Demo av Quick-, Fast-,
 och Bubblesort. Kassettversion. PB
 86-3
 18973007 -Att använda grafiken. 99'an 83-2
 18973008 -Sammansatta ledtexter med INPUT.
 99'an 83-4/5

99 ÖVRIGT

01991002H -Utskrift av månadskalender
 01992001E -Månkalender
 07991005H -Morse
 09991003F -Biorytm
 10991007H -Lotto
 11993073 -X Tipsrad. PB 85-2
 11997005 -Slumpning av LOTTO
 12993008 -Toto
 12993010 -Tipsservice
 12993011 -Alkotest
 13993021 -PX Etiketter
 13993028 -X Biorytm II
 14993032 -X Biorytm III
 14993041 -Resultatlista, slalom
 14993049 -X Tips-system
 15993078 -EX Automatisk uppringning
 15993096 -Dagräkning med almanacka
 16993017 -Barnvakten. PB 84-1
 18993013 -X QUETELETINDEX. Visar om du har
 under-, normal- eller övervikt.
 18993015 -LOTTOGRAF. Hjälper Dig att rätta
 Dina lottorader.

Jämförelse mellan PB-Forth och TI-Forth

av Jan Alexandersson

TI-Forth och PB-Forth har en gemensam kärna på 274 ord som tar 7250 bytes av ordlistan. Det finns sedan en påbyggnad som delvis är olika.

PB har möjlighet till laddning både från kassett och disk medan TI endast har disk.

PB kan laddas av fyra olika moduler: XB, MM, EA eller TI-Writer medan TI endast kan laddas av EA.

PB använder mycket smartare programnamn som underlättar laddning enligt följande:

PB: Välj 5 för RUN PROGRAM FILE och tryck sedan på (ENTER) utan att skriva namnet. EA har nämligen en inbyggd rutin som letar efter DSK1.UTIL1 ungefär som XB alltid letar efter DSK1.LOAD. Detta finns ej beskrivet i EA-manualen.

TI: Välj 3 för LOAD AND RUN och skriv sedan DSK1.FORTH. Det är svårare att komma ihåg programmets namn.

I PB-Forth kan man spara hela ordlistan med SAVE-SYSTEM medan man i TI måste spara grunden och påbyggnaden (med BSAVE) i två separata filer. Dessa fördelar med laddningen av PB gör att jag alltid försöker använda PB som grund.

PB har den bästa **editorn** och en bra **disassembler** vilket TI saknar. TI har dock grafikrutiner och flyttalsrutiner som PB saknar. Detta gör att en blandning av TI och PB är den bästa lösningen.

Det kan vara lämpligt att spara PB under CSAVE-SYSTEM och ovanför detta lägga in TI-rutiner. Gör så här:

```
' CSAVE-SYSTEM FENCE !
FORGET TMARG
```

Man kan sedan prova med DP ? som då visar -14268. Använd sedan LOAD av önskade rutiner. Detta kan sedan sparas i kompillerad form med SAVE-SYSTEM för disk eller CSAVE-SYSTEM för kassett.

Några ytterligare fördelar med PB är svenska bokstäver, riktiga små bokstäver och snyggare markör (endast ett diskret rakt streck).

Skilnader

TI	PB	EGEN
VSBW	V!	V!
VSBR	V@	V@
VWTR	VR!	VR!
GPLLNK	GPLLNK	GPLLNK
(även 0 837C ! utföres)		(enligt PB)
DSRLNK	DSRLNK	DSRLNK
(—)	(n —)	(n —)
(TI har 8 inbyggt medan PB kan välja på 8 för DSR och A för underprogram)		
CLS	PAGE	PAGE
	(0 0 GOTOXY (ingår)	
FORMAT-DISK	FORMAT	FORMAT
	(frågar en extra gång om man vill formatera. Detta är en bra extra säkerhet)	(enligt PB)
VFILL	HCHAR	VFILL
VAND	ANDCHAR	VAND
VOR	ORCHAR	VOR
VXOR	XORCHAR	VXOR
MON		MON
(lång rutin i maskinkod)		(: MON 32
TEXT	(TEXT-MODE)	GPLLNK ;)
	(ej komplett saknar bl a VSPTR)	TEXTMODE
		(enligt TI)

GRAPHICS	(GRAPH) (ej komplett saknar bl a VSPTR)	GRAFIKMODE (enligt TI)
SCREEN (bakgrund —)	SCREEN (förgr bakgr —)	SCREEN (bakgrund —) (vit förgrund inbyggd F0 + 7 VR!)
LOAD	LOAD (med CVD som gör att skärmnumret alltid läses som ett decimaltal)	LOAD (enligt PB)
LIST	LIST (med CVD)	LIST (enligt PB)
4 KEYBRD	5 KEYBRD	5 KEYBRD
FLUSH	FLUSH eller FL	FLUSH eller FL
(FL är bättre eftersom skärmen ej behöver läsas tillbaka från disk till RAM i onödan)		
COLD	COLD	COLD
(dessa rutiner är mycket olika varandra trots samma namn)		
SCOPY	COPY (med CVD men utan FLUSH)	COPY (enligt PB)
SMOVE	SCOPY (börj slut ny —) (utan CVD och utan FLUSH klarar ej överlappning)	SMOVE (börj ny ant —) (med CVD och FLUSH klarar överlappning)
.S (skriver lilla ö till vänster)	.S (skriver EMPTY till vänster)	.S (enligt PB)
DUMP (visar text och data)	DIS (visar text, data och opkod)	DUMP och DIS (båda rutinerna behövs)
VLIST	VLIST	VLIST (enligt PB)
BSAVE	SAVE-SYSTEM	SAVE-SYSTEM och BSAVE
S->FAC	—	S->FAC
S->F	—	S->F

Jag har endast fått de två sista flyttalsrutinerna att fungera med Mini Memory om 7200 XMLLNK används. Det står fel i MM-manualen om detta. Har du något förslag hur man ska kunna använda dessa flyttalsrutiner i PB-Forth med modulerna XB eller EA?

Open av P-markerad DIS/VAR 80-fil

av Jan Alexandersson

Om du skriver ett program som skall läsa av en DIS/VAR 80-fil bör du inte i onödan öppna den som UPDATE eftersom programmet undersöker om filen är skrivskyddad. Även om du inte skriver något vid OPEN så är UPDATE ett "default"-värde. Programmet kommer att krascha om du öppnar en skrivskyddad fil med UPDATE även om du bara läser filen i programmet. Följande rad är därför olämplig:

```
100 OPEN #1:FIL$
och bör ändras till
100 OPEN #1:FIL$,INPUT
```

Det är också viktigt att du stänger en öppen fil innan du byter skiva. Du kan annars råka ut för att den stängs oavsiktligt och detta innebär att delar av skivan blir förstörd. Följande saker utöver CLOSE stänger filer och skriver på den skiva som råkar sitta i just då.

```
SAVE och OLD
Editering av programmet
1 (ENTER)
BYE
```

Texteditor i c99

av Mikael Nordlin

Det här är resultatet av mina försök att skriva ett något så när användbart program i programspråket C till 99:an. Det är för fullständighetens skull fråga om användning av C99REL3 av Clint Pulley. Kompilatorn i detta programpaket har något vilseledande versionsnummer 2.1. Blott denna version har möjlighet att använda tvådimensionella variabler.

I mitt program heter den tvådimensionella variabeln 'text'. I Basic skulle en sådan variabel deklarerars så här: DIM text(230,40). Se efter i listningen hur variabeln deklarerars i C! Mitt program använder också några andra utökningar i C99REL3 utöver den tidigare versionen i programbanken, REL1. En utökad variant av ON var GOSUB i Extended Basic används; 'switch...case...default'. Vidare används formaterad utskrift på ett ställe; 'printf'. Detta motsvaras i Extended Basic ungefär av DISPLAY SIZE (FIELDLEN):USING str:var.

C99REL3 behövs alltså för den som avser att knappa in listningen. Jag vill emellertid inte rekommendera någon att göra sig besväret, editorn har alltför många begränsningar för att arbetet ska vara mödan värt. Begränsningen ligger alltså inte i kompilatorn utan i att jag inte använt någon vettig algoritm för editorn, som säkert också har buggar. Se alltså listningen som ett förhoppningsvis inspirerande exempel på hur program kan te sig i C! Skriv ett bättre program själv! C99REL3 finns i programbanken på två disketter till ett sammanlagt pris av 100 kr inkl utskriften manual. Du behöver dessutom Editor/Assembler (eller Funnelweb), minnesexpansion och diskdrive.

Åter till programlistningen! Det är alltså en editor, tänkt att användas vid utveckling av program tex i C. Den har begränsningar i främst följande hänseenden:

- Begränsad textbuffert om 230 rader. Begränsningen är inte så lätt att komma tillrätta med eftersom C99 lagrar lokala variabler i det låga minnet medan globala variabler lagras i det höga minnet tillsammans med den körbara programkoden. Jag har valt att lägga textbufferten i en global tvådimensionell teckenmatris, 'text'. Den upptar ca 9 K av programmets storlek – vilket inte vore möjligt som en lokal variabel – och laddas alltså in från disketten tillsammans med programmet. Textbufferten kan utvidgas något, men jag har inte gjort det nu eftersom jag vill bereda plats i de totalt 24 K som finns i det höga minnet för senare tilläggsrutiner.

- Sökmöjligheter saknas, vilket kanske inte gör så mycket idag pga den begränsade textbufferten.

- Insert resp Delete av rader saknas, något som är lätt att lägga till.

Jag har planer på att utöka textbufferten genom segmenterad minneshantering från disk eller RAM-disk samt även möjlighet att direkt läsa Companion-filer, dvs textfiler i INT/VAR 254-format. Editorn ska nu kunna läsa DIS/VAR-filer av valfri maxlängd, men skriver endast sådana filer med standardlängden 80.

FCTN 7, dvs AID, ger hjälpskärm.

FCTN 0 lagrar texten till disk eller skrivare. Om man väljer skrivare har man möjlighet att bestämma vänstermarginal på utskriften. FCTN 0 ska emellertid i allmänhet föregås av FCTN M om man inte vill dumpa ut hela textbufferten, dvs även tomtecknen, eller själv lägga in alla radslutstecken (↵) samt filslutstecknet (↵).

```
#define HOME      0
#define CR        13

/* Key definitions */
#define AID        1
#define CLEAR      2
#define DEL        3
#define INS        4
#define REDO       6
#define ERASE      7
#define LEFT       8
#define RIGHT      9
#define DOWN      10
#define UP        11
#define PROCD     12
#define BEGIN     14
#define BACK      15
#define BLANK     188
#define FCTNL     194
#define FCTNM     195

/* System characteristics */
#define LINES      23 /* max. lines/screen */
#define CMD        24 /* command line pos. */
#define COL        40 /* max. columns */
#define COLM       39 /* COL-1 */
#define MAX        230 /* max. lines */
#define MAXM       229 /* MAX-1 */
#define CHRS       920 /* screenloc:s */
#define MXPOS      9199 /* MAX*COL-1 */

/* *** Mikes text editor i C ***
Ver: Se "purge"
Runtimes-bibliotek:
CSUP
CFIO
PRINTF
*/

/* #include "dsk1.stdio" */

extern fopen(),fclose(),getc(),putc(),fgets(),fputs();
/*
** STDIO : standard c99 I/O definitions
*/
#define FILE      int
#define NULL      0
#define EOL       10
#define FF        12

/* Delete above lines if #include "dsk2.mitec2;c" */

extern printf();

char text[MAX][COL]; /* Global variables */
int pos;

main() /* MITEC;C */
{
char name[21],temp[81];
char k,x,y,xy,yy,col,max;
int key,left,xx;
FILE unit;
purge();
do /* Start main loop - Rewrite */
{
x=(xy=pos/COL)%LINES+1;
y=(yy=pos%COL)+1; /* buffer: 'xy','yy' */
```

```

locate(CMD,1);
printf("Scr= %2d Line= %3d Col= %2d Pos= %5d",
xy/LINES+1,xy,yy,pos);
k=NULL;
do
{
locate(x,y);
putchar('_');
get(&k);
locate(x,y);
putchar(text[xy][yy]);
get(&k);
}
while(!k);
switch(k)
/* ASCII- or CTRL-char? */
{
case DEL:
col=yy;
if(text[xy][COLM] > ' ')
max=COL;
/* Don't join lines */
else max=COLM;
while(col<max)
text[xy][col] = text[xy][++col];
text[xy][max] = ' ';
break;
case INS:
col=COLM;
while(col<yy)
text[xy][col] = text[xy][--col];
text[xy][yy] = ' ';
break;
case ERASE:
purge();
break;
case CLEAR:
if(xy<MAX-LINES) pos=pos+CHRS;
break;
case BEGIN:
pos=pos-pos*CHRS;
break;
case PROCD:
pos=just(pos-CHRS);
break;
case AID:
putchar(FF);
puts("# AID $ Press any key to continue\n\n");
puts("FCTN 1: Delete a character");
locate(5,6);
puts("2: Insert a character");
locate(7,6);
puts("3: Purge textbuffer - Clear screen");
locate(9,6);
puts("4: Next screen - Roll down");
locate(11,6);
puts("5: Home cursor - Same screen");
locate(13,6);
puts("6: Prev. screen - Roll up");
locate(15,6);
puts("8: Show specific line");
locate(17,6);
puts("9: Exit");
locate(19,6);
puts("0: Save text to file");
locate(21,6);
puts("L: Load text");
locate(23,6);
wait("M: Set file markers (auto)");
break;
case REDD:
clrcmd("SHOW line: ");
pos=input(3)*COL;
break;
case BLANK:
clrcmd("DEVICE: ");
gets(name);
puts("ARE you sure?");
key=getchar();
if(key!='y' & key!='Y') break;
xy=yy;left=yy=NULL;
if($name != 'D')
{
clrcmd("LEFT margin: ");
left=input(2);
}
if(unit=fopen(name,"w"))
{
puts("SENDING text");
while(text[xy][y] != ' ' & xy<MAX)
{
xx=left;
while(xx<col) putc(' ',unit);
while(text[xy][y] != ' ')
{
putc(text[xy][y],unit);
if(!yy%COL) xy++;
}
/* Add space to empty line (Ed/Asm-Editor-wise) */
if(!y) putc(' ',unit);
if(putc(EOL,unit)!=EOL)
{
wait("#ERROR: Disk full. Press any key");
break;
}
xy++;
yy=NULL;
}
fclose(unit);
}
else
wait("#ERROR. Press any key");
break;
case CR:
break;
pos=++xy*COL;
case LEFT:
pos=just(--pos);
break;
case DOWN:
if(xy<MAXM) pos=pos+COL;
break;
case UP:
pos=just(pos-COL);
break;
case FCTNL:
clrcmd("LOAD text (DSKn.NAME): ");
gets(name);
if(unit=fopen(name,"r0"))
{
puts("#LOADING file");
xy=1;
while(fgets(temp,81,unit))
{
yy=NULL;
do
{
if(!yy%COL) xy++;
text[xy][y] = temp[yy];
}
while(temp[yy]);
/* Stop loading if buffer full */
if(xy==MAXM) break;
}
}
}

```

```

}
else
wait("#ERROR: File not found. Press any key");
fclose(unit);
pos=HOME;
break;
case FCTNM:
clrcmd("MARKING text");
xy=MAX;
while(--xy > -1)
{
yy=NULL;
while(yy < COL & text[xy][yy++] == ' ')
;
/* Set EOF marker if text */
if(--yy != COL) break;
}
text[xy][0] = ' ';
while(xy--)
{
yy=COL;
if(text[xy][--yy] == ' ')
{
while(yy-->0 & (text[xy][yy] == ' '
| text[xy][yy] == '~'));
text[xy][yy] = '~';
}
}
break;
default:
/* ASCII or RIGHT */
locate(x,y);
if(k==RIGHT) putchar(text[xy][yy]=k);
if(pos<MXPOS) pos++;
}
if(!((k==RIGHT | k==' ' & k=='~')
& pos%CHRS>HOME | k==BACK |
k==LEFT & (k>1 | y>1) |
k==UP & x>1 | k==BEGIN |
(k==DOWN | k==CR) & x<LINES)
& pos<MXPOS)
write(); /* Do it! */
while(k!=BACK);
putchar(FF);
}
purge() /* Empty-buffers */
{
char x,y;
putchar(FF);
locate(10,10);
puts("MITEC v1.0 03/05/87");
locate(13,10);
puts("HELP - Press FCTN AID");
for(x=pos; x<MAX; x++)
for(y=NULL; y<COL; y++) text[x][y]=' ';
putchar(FF);
}
write() /* Dump */
{
int line,top;
char col;
locate(1,1);
for(top=(line=pos/CHRS+LINES)+LINES;
line<top;
line++)
{
col=NULL;
while(col<COL)
putc(text[line][col++]);
}
}
just(arg)
int arg; /* Justify */
{
if(arg<1) return(NULL);
return(arg);
}
clrcmd(text) /* Clear command line */
char *text;
{
char i;
locate(CMD,1);
i=COL;
while(--i) putchar(' ');
locate(CMD,1);
putc('*');
puts(text);
}
get(key) char *key; /* Delay until char */
{
int i; char l;
for(i=1; i<150 & !key; i++)
{
*key=poll(0);
while(*key & (l=poll(0)))
;
}
}
input(field) int field;
{
char key;
int num;
num=NULL;
while(field)
{
key=NULL;
puts("_\b");
get(&key);
puts("_\b");
get(&key);
if(key==CR) break;
if(key<'0' | key>'9') continue;
field--;
num=num*10+key-48;
putc(key);
}
return num;
}
wait(text) char *text;
{
puts(text);
while(!poll(0))
;
}

```

Fairware (freeware)

50:- per skiva för medlemmar

Postgiro 19 83 00-6

DISK MANAGER 99 Version 85-10

1 skiva 2 sidor manual
av Mike Dodd, USA
EM+DS+XB/EA/MM
se PB 86-3.09

DISK MANAGER 1000 Version 3.5

1 skiva 25 sidor manual
av Ottawa User Group
EM+DS+XB/EA/TW
hjälpmedel för disk se PB 86-4.14, Micropendium
86-SEP.44, 86-OKT.44

DISK CAT Version 2.0

1 skiva 1 sida manual
av Mack McCormick, USA
EM+DS+XB/EA/MM/TW
hjälpmedel för disk

NEATLIST Version 1.0

1 skiva 4 sidor manual
av Danny Michael, USA
EM+XB
hjälpmedel för BASIC se PB 86-3.09, Micropendium
85-MAJ.22

SMALL C Version 2.0

2 skivor 22 sidor manual
av Clint Pulley, Kanada
EM+DS+EA(GROM3)
- C99 se PB 86-3.14 och PB 87-2.14
- RUNOFF formaterare
- C99-extra 1 skiva
- C-stuff 1 skiva

MACRO ASSEMBLER Version 5

2 skivor 19 sidor manual
av RA Green, Kanada
EM+DS+XB/EA/TW
programmeringsspråk se Micropendium 86-JUL.39,
86-AUG.45 (saknar egen editor)

UNIVERSAL DISASSEMBLER Version 2.3

1 skiva 6 sidor manual
av Rene LeBlanc, USA
EM+DS+EA/MM
hjälpmedel för assembler

SUPERBUG II Version 2.0

1 skiva manual ingår ej, beställs separat från USA)
av Edgar Dohmann, USA
EM+XB/EA/MM
hjälpmedel för assembler se Micropendium 86-MAJ.42

TK-WRITER

1 skiva 8 sidor manual
av Tom Knight, USA
EM+DS+XB/EA/MM
ordbehandlingsprogram se Micropendium 85-MAJ.22,
85-OKT.53 (fungerar som TI-Writer)

FUNNELWEB Version 3.4 (1987-04-10)

2 skivor 28 sidor manual
av Tony & Will McGovern
EM+DS+XB/EA/MM/TW
menyprogram för TW,EA mm se Micropendium 86-JUL.36
(filer för TW och EA ingår men ej manual). Ny förbättrad
version trots oförändrat versionsnummer

RAPID SCROLL Version 1.4+2.3

1 skiva 8 sidor manual
av Jurgen Switalski, USA
består av 4 fristående program (min EM+DS)
- RAPID SCROLL visar DIS/VAR80 i 40 och 64 kolumner
(XB/EA/MM/TW)
- MENU LOADER skapar meny för start av assemblerp-
rogram (EA GROM3)
- SUPER MAIL LIST adressregister (XB)
- SMANAGER diskatalog (EA/MM)

PRBASE Version 2.0

2 skivor 37 sidor manual
av William Warren, USA
EM+DS+XB/EA/MM/TW
registerhanteringsprogram se Micropendium 86-APR.39
SCREENDUMP Version 84-12
1 skiva 4 sidor manual
av Danny Michael, USA
EM+XB/EA/MM+EPSON se PB 86-3.09

COLIST Version 1.0

1 skiva 11 sidor manual
av Will & Tony McGovern
består av 5 fristående program
- COLIST listar BASIC-program (EM+DS+XB)
- DISK HACKER Version 1.0 för TI:s disk controller
(EM+DS+XB/EA/MM)
- CASSTRANS överför assembler från DSK till kassett
(EM+DS+XB)
- CASSLOAD laddar assembler från kassett (EM+XB)
- DIS/ASS dissassembler i BASIC (XB/EA/MM)
DISK HACKER Version 2.0
1 skiva (samma som ovanstående men för både TI och Cor-
comp diskkontrollkort)

FAST TERM Version 1.16

1 skiva 8 sidor manual
av Paul Charlton, USA
EM+DS+XB/EA/MM/TW+Modem
se Micropendium 85-MAJ.22, 86-NOV.50

PILOT 99 Version 1

2 skivor 17 sidor manual
Programmeringsspråk, främst för datorstödd utbildning.

Övriga program

Av TI frisläppta

TI Forth (disk+manual) 200:-
TI Forth källkod+demo (3 skivor) 120:-
Uppgradering Multiplan/TI-Writer (disk) 50:-
Navarone Super Debugger (disk) 50:-

Skrivna av medlemmar

PB-Forth (kassett eller disk) 100:-
PB-Forth källkod (3 skivor) 120:-
Forth&99 (3 skivor) 120:-
Game-Forth (disk) 50:-
FORIT (disk) 50:-
PB-Disassembler 1 skiva av Tony Rogvall. Hjälpmiddel för
assembler-programmering. Se PB 84-4 (disk) 50:-
Checklist av Lars Thomasson skapar checklistor för Exten-
ded BASIC-program. Se PB 87-1.5 (disk) 50:-

Samlingsdisk eller kassett

Nittinian 1983 50:-
Programbiten 1984 50:-
Programbiten 1985 50:-
Programbiten 1986 50:-

Teckensmeden (Extended BASIC)

Med hjälp av detta program kan du rätt enkelt ta fram ASCII-koderna för omdefinierade tecken i TI-basic och TI-extended Basic. Programmet låter dig skapa 4 nya tecken samtidigt. Det betyder att du kan se hur en sprite kommer att ta sig ut t ex.

Du kan vrida och vända på de tecken du designar, och invertera dem m m. Programmet skall vara helt självinstruerande, men om det är något som är oklart går det bra att skriva eller ringa till:

Jens Hövmöller
Skestavägen 59
163 53 SPÅNGA
Tel: 08/760 79 77

Obs! I programraderna 150-200 nedan har kontrolltecken omdefinierats. Därför stämmer inte listningen med tecknens utseende för de raderna. Eftersom det handlar om titelskärmen spelar det väl inte så stor roll.

```
100 CALL CLEAR
110 CALL CHAR(126,"0000000000602040")
120 CALL CHAR(128,"0103070F1F3F7FFFFF7F3F1F0F0703
0180C0E0F0BFCFEFFFFFECF8F0E0C080")
130 CALL CHAR(132,"FFFFFFFFFFFFFFFFFECF8F0BFCFE
FF")
140 CALL COLOR(13,1,1)
150 PRINT "EEEEEEEEEEEEEEEEEEEE E
E E E E
E E E E"
160 PRINT " E AE AEAE AEAE AE E AE
AA E AA AE AE E EA E EAA EAA E E
AEAE E EEEAE EE EEEAE E"
170 PRINT " E E E EAA E E E E E A
E A E AA E A E E A EA AE A E AA EA A E
E"
180 PRINT " E AA
E EAAEAE AE AE AE AE
E E E E AE E E AE E E
190 PRINT " E E E EAA E E EAA E E E E E E
EAE E EEEAE E E E E E E E E E E
E E E E A E E E A E E
200 PRINT "EA A A EAAEA AE A E EEEEE
TRYCK ENTE
R"
210 CALL HCHAR(2,2,128):: CALL HCHAR(9,1,128):: C
ALL HCHAR(9,2,132):: CALL VCHAR(10,1,132,5)::
CALL HCHAR(15,1,129):: CALL HCHAR(15,2,132)
220 CALL HCHAR(21,1,128):: CALL HCHAR(21,2,132)::
CALL HCHAR(21,31,132):: CALL HCHAR(21,32,131
):: CALL HCHAR(12,31,132)
230 CALL HCHAR(12,32,131):: CALL HCHAR(12,31,132)
:: CALL HCHAR(8,31,132):: CALL HCHAR(8,32,131
)
240 CALL SCREEN(2):: CALL COLOR(13,11,2,5,11,1,6,
11,1,7,11,1,8,11,1):: CALL KEY(0,K,Y):: IF Y
THEN 250 ELSE 240
250 CALL COLOR(13,2,1)
260 DIM CREM(8,8)
270 CALL SCREEN(11)
280 CALL CHAR(132,RPT$("F",64))
290 CALL MAGNIFY(4)
300 CALL CHAR(91,"002B003B447C4444002B007C4444447
C003B2B3B447C444400000FF00000000")
310 CALL CHAR(64,"60909060")
320 REM #####
330 REM #
340 REM # TECKENSMEDEN #
350 REM #
360 REM # AV #
370 REM #
380 REM # JENS #
390 REM #
400 REM # 1986 #
410 REM #
420 REM #####
430 REM
440 CALL CLEAR
450 RESTORE 620
460 DIM CHR$(16):: DIM CR$(16)
470 FOR I=1 TO 16
480 READ CR$(I),CHR$(I)
490 NEXT I
500 RESTORE 620
510 CALL CHAR(128,"",129,"",130,"",131,"")
```

```
520 FOR I=3 TO 8 :: CALL COLOR(I,2,1):: NEXT I
530 CALL CLEAR
540 IF OP>0 THEN 800
550 FOR I1=1 TO 8
560 FOR I2=1 TO 8
570 CR(I1,I2)=42
580 NEXT I2
590 NEXT I1
600 V=1 :: H=1
610 CALL COLOR(2,16,2)
620 DATA 0,"####"
630 DATA 1,"###-"
640 DATA 2,"##-#"
650 DATA 3,"#-##"
660 DATA 4,"-###"
670 DATA 5,"-##-"
680 DATA 6,"#--#"
690 DATA 7,"#---"
700 DATA 8,"-###"
710 DATA 9,"-##-"
720 DATA A,"-#-#"
730 DATA B,"-#--"
740 DATA C,"---#"
750 DATA D,"---#"
760 DATA E,"----"
770 DATA F,"-----"
780 CALL CHAR(42,"FFB1B1B1B1B1B1FF")
790 CALL CHAR(45,"FFB1B1B1B1B1B1FF")
800 FOR I=9 TO 16
810 CALL HCHAR(I,12,42,8)
820 NEXT I
830 FOR I=1 TO 8
840 FOR J=1 TO 2
850 CH$(I,J)="####"
860 NEXT J
870 NEXT I
875 GOSUB 2190
880 FOR I=21 TO 24 :: CALL HCHAR(I,4,48,16):: CAL
L HCHAR(I,2,107+I):: NEXT I
890 IF CR(V,H)=42 THEN I1=45 :: I2=42 ELSE I1=42
:: I2=45
900 CALL HCHAR(8+V,11+H,I1)
910 CALL HCHAR(8+V,11+H,I2)
920 CALL KEY(0,K,Y):: IF Y=1 OR K>64 AND K<90 THE
N 930 ELSE 890
930 IF (K=13)+(K=68)+(K=83)+(K=69)+(K=88)+(K=32)+
(K=48)=0 THEN 890
940 IF K=68 THEN H=H+1 :: IF H=9 THEN H=1
950 IF K=48 THEN GOSUB 1220
960 IF K=83 THEN H=H-1 :: IF H=0 THEN H=8
970 IF K=88 THEN V=V+1 :: IF V=9 THEN V=1
980 IF K=69 THEN V=V-1 :: IF V=0 THEN V=8
990 IF K=32 THEN IF CR(V,H)=42 THEN CR(V,H)=45 ::
GOTO 890
1000 IF K=32 THEN IF CR(V,H)=45 THEN CR(V,H)=42
1010 IF K=13 THEN 1030
1020 GOTO 890
1030 FOR I1=1 TO 8
1040 FOR I2=1 TO 5 STEP 4
1050 FOR I3=0 TO 3
1060 A$=A$&CHR$(CR(I1,I2+I3))
1070 NEXT I3
1080 I$=((SEG$(A$,4,1)="-")+ (SEG$(A$,3,1)="-")*2+
SEG$(A$,2,1)="-")*4+(SEG$(A$,1,1)="-")*8)
1090 A$=A$&CHR$(I+1)
1100 A$="" :: NEXT I2
1110 NEXT I1 :: CALL SPRITE(2,132,2,92,200)
1120 CALL HCHAR(1,1,32,256):: DISPLAY AT(1,1):"VIL
KET TECKEN VILL 13":"DU DEFINIERA OM 24"
:"": "1"
1130 ACCEPT AT(4,1)VALIDATE("1234")SIZE(-1):OP
1140 CALL HCHAR(1,1,32,256)
1150 DISPLAY AT(20+OP,2):A$
1160 CALL CHAR(127+OP,A$)
1170 OP=OP+1
1180 CALL MAGNIFY(4)
1190 CALL SPRITE(1,128,16,92,200)
1200 A$="" :: A$=""
1210 GOSUB 2180 :: GOTO 890
1220 CALL HCHAR(1,1,32,256):: DISPLAY AT(1,7):"SPE
CIALKOMMANDON"
1230 DISPLAY AT(3,1):"1 SPEGELVANDNING"
1240 DISPLAY AT(4,1):"2 UPPÖCHNERUVANDNING"
1250 DISPLAY AT(5,1):"3 VRIDNING LÅNGS DIAGONALEN"
1260 DISPLAY AT(6,1):"4 INVERTERING"
1270 DISPLAY AT(7,1):"5 SUDDA TECKENFALTET"
1280 DISPLAY AT(8,1):"6 EDITERA GAMMALT TECKEN"
1290 CALL KEY(0,K,Y):: IF K>48 AND K<56 THEN ON K-
48 GOSUB 1310,1470,1610,1770,1880,1960 ELSE 1
290
1300 CALL HCHAR(1,1,32,256):: GOSUB 2190 :: RETURN
1310 REM #####
1320 REM # SPEGELVANDNING #
1330 REM #####
1340 CALL SOUND(-20,880,0,440,0,-3,0)
```

Tecken smiden

TRYCK ENTER

```

1350 FOR I1=1 TO 8
1360 FOR I2=1 TO 8
1370 CREM(I1,I2)=CR(I1,9-I2)
1380 NEXT I2
1390 NEXT I1
1400 FOR I1=1 TO 8
1410 FOR I2=1 TO 8
1420 CR(I1,I2)=CREM(I1,I2)
1430 CALL HCHAR(8+I1,11+I2,CR(I1,I2))
1440 NEXT I2
1450 NEXT I1
1460 RETURN
1470 ! UPP-OCH-NER-VÄNDNING
1480 !
1490 FOR I1=1 TO 8
1500 FOR I2=1 TO 8
1510 CREM(I1,I2)=CR(I1,I2)
1520 NEXT I2
1530 NEXT I1
1540 FOR I1=1 TO 8
1550 FOR I2=1 TO 8
1560 CR(I1,I2)=CREM(9-I1,I2)
1570 CALL HCHAR(8+I1,11+I2,CR(I1,I2))
1580 NEXT I2
1590 NEXT I1
1600 RETURN
1610 REM *****
1620 REM VRIDNING
1630 REM *****
1640 REM
1650 FOR I1=1 TO 8
1660 FOR I2=1 TO 8
1670 CREM(I1,I2)=CR(I1,I2)
1680 NEXT I2
1690 NEXT I1
1700 FOR I1=1 TO 8
1710 FOR I2=1 TO 8
1720 CR(I1,I2)=CREM(I2,I1)
1730 CALL HCHAR(8+I1,11+I2,CR(I1,I2))
1740 NEXT I2
1750 NEXT I1
1760 RETURN
1770 REM *****
1780 REM INVERTERING
1790 REM *****
1800 REM
1810 FOR I1=1 TO 8
1820 FOR I2=1 TO 8
1830 IF CR(I1,I2)=42 THEN CR(I1,I2)=45 ELSE CR(I1,
I2)=42
1840 CALL HCHAR(8+I1,11+I2,CR(I1,I2))
1850 NEXT I2
1860 NEXT I1
1870 RETURN
1880 FOR I1=1 TO 8
1890 FOR I2=1 TO 8
1900 CR(I1,I2)=42
1910 CALL HCHAR(8+I1,11+I2,CR(I1,I2))
1920 NEXT I2
1930 NEXT I1
1940 RETURN
1950 REM *****
1955 REM EDITERING
1960 REM *****
1970 H1=4
1980 CALL HCHAR(1,1,32,256)
1990 DISPLAY AT(1,1):"VILKET TECKEN VILL 13 A A":
"DU TA IN IGEN 1 24 ":"
A A"
2000 ACCEPT AT(2,16)VALIDATE("1234")SIZE(-1):A
2010 CALL CHARPAT(127+A,A*)
2020 FOR I1=1 TO 16
2030 V1=V1+.5 : IF H1=0 THEN H1=4 ELSE H1=0
2040 I2=SEG*(A*,I1,1)
2050 I3=ASC(I2*)

```

```

2060 IF I3>57 THEN I3=I3-7
2070 I3=I3-47
2080 I4=INT(I1/2):: IF I1/2=INT(I1/2) THEN I5=0 ELSE
I5=4
2090 DISPLAY AT(V1+8,H1+10)SIZE(4):CHAR*(I3)
2100 FOR I=1 TO 4
2110 IF SEG*(CHAR*(I3),I,1)="#" THEN CR(V1,H1+I)=4
2 ELSE CR(V1,H1+I)=45
2120 NEXT I
2130 NEXT I1
2140 H1=0 : V1=0
2150 CALL HCHAR(1,1,32,256)
2160 RETURN
2165 REM *****
2170 REM TRYCK PA... FÖR...
2180 REM *****
2190 DISPLAY AT(1,1):"<ENTER> UTSKRIFT AV TECKNET"
2200 DISPLAY AT(2,1):"O VRIDNING INVERTERING MM"
2210 DISPLAY AT(4,1):"STYR MED PILTANGENTERNA":"TR
YCK PA MELLANSLAG FÖR ATT":"INVERTERA RUTAN D
AR":"MARKÖREN FINNS"
2220 RETURN

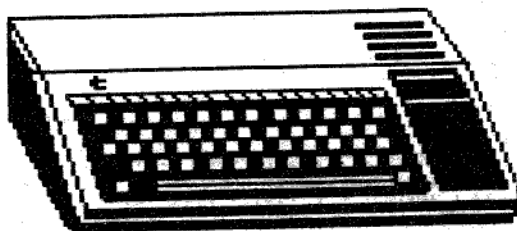
```

ANNONS

Säljes:

TI-99/4A med expansionsbox, disk drive, diskkontrollkort, 32K minnesexpansion, samt RS232-kort. Editor/Assembler, Extended BASIC, PB-Forth, TI-Forth och c99. 6 spelmoduler + disketter med spel och andra program. 4500:- P-kodkort med programvara på disketter 850:-

Martin Flodén
Björkliden 22
183 41 Täby
08/7584610



Snabbare HCHAR

Vill Du snabba upp dina Extended BASIC-program kan följande assembler-program underlätta det. Programmet är skrivet av Bengt Fahlgren. Han har också gjort ett litet exempel som visar hur rutinerna anropas.

```

100 CALL INIT
110 CALL CHAR(130,"FFFFFFFFFFFFFFFF")
120 CALL LOAD("DSK1.HCHAR:REL")
130 CALL CLEAR
140 CALL CHAR(91,"FFFFFFFFFFFFFFFF")
150 CALL CHAR(92,"003C3C3C3C3C3C00")
160 CALL LINK("HCHAR",1,1,91,256)
170 CALL LINK("HCHAR",8,1,92,256)
180 CALL LINK("HCHAR",16,1,91,256)
190 FOR I=1 TO 400 : NEXT I
200 CALL LINK("HCHAR",1,1,92,256)
210 CALL LINK("HCHAR",8,1,91,256)
220 CALL LINK("HCHAR",16,1,92,256)
230 FOR I=1 TO 400 : NEXT I : GOTO 160

```

```

*****
*
*           HCHAR
*
* CALLED FROM TI EXTENDED BASIC
*
* FORMAT "CALL HCHAR(ROW,COL,CHAR-CODE,REPETITIONS)"
*
* ATTEMPTING TO WRITE OUTSIDE SCREEN BORDERS WILL RESULT
* IN AN ERROR MESSAGE EX "CALL HCHAR(1,1,65,769)"-> ERROR
*
*****
*
* WRITTEN 1987 BY BENGT FAHLGREN
*           GULLMARSVAGEN 11
*           S-121 41 JOHANNESHOV
*           SWEDEN
*
* THIS PROGRAM IS PLACED UNDER PUBLIC DOMAIN
*
-----
*
* REGISTERUSAGE:
*
* R0 - TEMPORARY USAGE.
* R1 -  --
* R2 -  --
* R3 -  --
* R4 - ROW
* R5 - COL
* R6 - CHAR-CODE
* R7 - REPETITIONS.
* R8 - SCREEN OFFSET.
* R9 - UNUSED
* R10 - END OF SCREEN IMAGE TABLE. >02FF+1
* R11 - RETURNADDRESS AT BL.
* R12 - UNUSED.
* R13 - --
* R14 - --
* R15 - --
*
*****
* TITLE
NAME   IDT  'HCHAR'
*****
* EQUATES
*
NUMAS6 EQU  >200B
NUMREF EQU  >200C
XMLLNK EQU  >201B
CFI      EQU  >12B8
CIF      EQU  >20
FAC      EQU  >834A
GPLST    EQU  >837C
GPLWS    EQU  >83E0
ERR      EQU  >2034
VSBW     EQU  >2020
*****
*
* WORKSPACE
*
OURWS    BSS  32
*****
*
* DEFINITIONS
*
DEF      HCHAR
*****
* GETNUM - GET NUMERICAL PARAMETER.
*
* FETCHES A NUMERICAL PARAMETER FROM THE CALL-STATEMENT AND CONVERTS IT
* TO AN INTEGER. THE RESULT IS GIVEN IN FAC (>834A).
*****

```

```

GETNUM BLWP @NUMREF      GET NUMERICAL PARAMETER TO FAC.
      BLWP @XMLLNK      CALL ROUTINE IN ROM
      DATA CFI        FOR CONVERSION TO INTEGER.
      B *R11          RETURN TO CALLING ROUTINE.
*****
* CHECK - TESTS IF AN INTEGER IS WITHIN THE LIMITS.
*
* TESTS THE NUMBER IN FAC AGAINST LOWER AND UPPER LIMITS PROVIDED BY THE
* BL-CALL. IF VALUE OUTSIDE LIMITS THE BASIC-PROGRAM IS INTERRUPTED
* WITH THE ERROR-MESSAGE BAD VALUE.
*****
CHECK C @FAC,*R11+ TEST LOWER LIMIT.
      JLT ECODE      JUMP TO ERROR-ROUTINE IF LESS.
      C @FAC,*R11+ TEST UPPER LIMIT.
      JGT ECODE      JUMP TO ERROR-ROUTINE IF GREATER
      B *R11        RETURN.
ECODE LI R0,>1E00      LOAD ERROR-CODE FOR BAD VALUE.
      BLWP @ERR      JUMP TO ERROR-ROUTINE.
*****
*
* MAIN PROGRAM
*
HCHAR LWPI OURWS
      CLR R0          PARAMETER IS NOT AN ARRAY
      LI R1,1         --- NUMBER
      BL @GETNUM      PUT ROW INTO FAC
      BL @CHECK       CHECK ENTRY WITHIN BORDERS
      DATA 1
      DATA 24
      MOV @FAC,R4      MOVE ROW TO R4
      INC R1
      BL @GETNUM      PUT COL INTO FAC
      BL @CHECK       CHECK ENTRY WITHIN BORDERS
      DATA 1
      DATA 32
      MOV @FAC,R5      MOVE COL TO R5
      INC R1
      BL @GETNUM      PUT CHAR-CODE INTO FAC
      BL @CHECK       CHECK ENTRY WITHIN BORDERS
      DATA 30
      DATA 143
      MOV @FAC,R6      MOVE CHAR-CODE TO R6
      INC R1
      BL @GETNUM      PUT REPETITIONS INTO FAC
      MOV @FAC,R7
*****
* CALC SCREEN IMAGE PLACE AND CHECK IF REPETITIONS
* IS WITHIN LIMITS
*
      DEC R4          DECREMENT (ROW-1)
      SLA R4,5        MULTIPLY WITH 32
      DEC R5          (COL-1)
      A R5,R4         ADD TO ROW ADDRESS
      MOV R4,R3
*
      A R7,R3         ADD REP. TO THE LOC. WHERE THE PRINTING SHOULD START
      LI R10,>0300     END OF SCREEN IMAGE TABLE
      C R3,R10
      JGT ECODE      RETURN TO BASIC IF OUT OF LIMIT
      SWPB R6         PUT CHAR-CODE IN MSBY
      MOVB R6,R1
*
      LI R8,>6000      LOAD BASIC SCREEN OFFSET
      AB R8,R1        ADD OFFSET TO CHAR-CODE
RE MOV R4,R0
      BLWP @VSBW
      INC R4          INC SCREEN POSITION
      DEC R7          DEC REMAINING CHAR'S
      JNE RE         ARE THERE MORE CHAR'S TO WRITE?
*
* PREPARE TO JUMP TO BASIC
*
      CLR @GPLST
      LWPI GPLWS
      B *R11
      END

```


TI-WRITER, OPTION 3

av Sören Bernle

Hur skall det program vara konstruerat som går att köra igång med TI-Writers valmöjlighet #3 (användning)?

Ungefär så avslutade jag min assembler-skole-artikel i PB 85-4. Efter som jag nu kan få min logotype utskrivna via TI-Writer, känner jag mig manad att själv svara på ovanstående fråga.

Mina kunskaper i assembler-programmering är som tidigare sagts mycket kläna, så jag beskriver bara hur och var programmet har ändrats. De förklaringar som saknas, får någon annan ge.

Fel i programmet!

Till att börja med, är det ett direkt fel i programlistningen (PB 85-4). Det sista bytet i den bytesrad som står i jämnhöjd med instruktionen LI R3,4, har fått sitt kommatecken felplicerat. Det skall givetvis vara placerat mellan bytena och inte efter.

Vidare är punkten efter PIO i TEXT-direktivet helt onödig.

Inget eget workspace!

Av någon sorts knepig sparsamhet, lät jag programmet klara sig utan ett eget workspace. Och det gick ju tursamt nog bra i assembler, men ack, det gick ej i TI-Writer.

Alltså, har programmet fått ett eget WS (ang. WS, se PB 85-4 sid 8)

Återhopp!

Återhopp-rutinen har ändrats till den rutin, som är beskriven överst på sid 442 i E/A manualen.

Inga tillgängliga subrutiner!

De i programmet mycket nödvändiga subrutinerna: VSBW, VMBW och DSRLNK är inte tillgängliga i TI-Writer, utan dessa måste finnas i själva programmet.

Därför har det i slutet av programmet tillförts två subrutinprogram. Det första ger tillgång till VSBW, VMBW, VSBR, VMBR och VWTR rutinerna, och det sista ger tillgång till DSRLNK rutinen.

Att REF direktivsraden strukits och att END direktivet flyttats ned, tarvar väl ingen förklaring? (Jasså, gör det! Se då 'BIBELN').

Därtill har namnlängdspekaren i huvudprogrammet, getts samma namn som i subrutinerna. Dvs PNTR har ändrats till NMLPNT. Namnändringen kunde förstås lika gärna gjorts tvärtom.

```
*****
*
* Print a LOGOTYPE on a PRINTER *
*
* by using TI-writers option #3 *
*
* 5 dec. 1985 *
*
* S. Bernle *
* P1 280 *
* 260 38 KATTARP *
* Tel 042-93305 *
*
*****
```

```
DEF START,SFIRST,SLAST,SLOAD
START B @FLYT
SFIRST EQU START
SLOAD EQU START
PABBUF EQU >1000
PAB EQU >F80
STATUS EQU >837C
NMLPNT EQU >8356
PDATA DATA >0002,PABBUF,>00DC,>0000,>0003

TEXT 'PIO'
EVEN
WRITE BYTE >03
CLOSE BYTE >01
MYWS BSS >20

FLYT LWPI MYWS
LI R0,PAB Move PAB data into PAB in VDP-RAM
LI R1,PDATA
LI R2,>0E
BLWP @VMBW

LI R6,PAB+9 Store pointer to name length in >8356
MOV R6,@NMLPNT

BLWP @DSRLNK Open file
DATA B

MOVB @WRITE,R1 Change I/O op-code to write
LI R0,PAB
BLWP @VSBW

LI R0,PABBUF Prepare registers to move data
LI R1,DAT1-220 into PABBUF area in VDP-RAM
LI R2,220

LI R3,4 Load loop-counter

LOOP MOV R6,@NMLPNT Restore pointer to name length and
A R2,R1 move 220 bytes into PABBUF in VDP-RAM
BLWP @VMBW

BLWP @DSRLNK Print out
DATA B

DEC R3 Has all been printed out?

VSRPGM BL @WVDPRA
MOVB @VDPDR,@2(R13)
RTWP

* VDP MULTIPLE BYTE READ

VMRPGM BL @WVDPRA
VRDMOR MOVB @VDPDR,*R1+
DEC R2
JNE VRDMOR
RTWP

VTRPGM MOV *R13,R2
MOVB @R2LB,@VDPWA WRITE TO REGISTER BIT
ORI R2,>8000
MOVB R2,@VDPWA
RTWP

* SETS UP VDP ADDRESS

WVDPWA LI R1,>4000 INDICATES WRITE
JMP WVDPAD

WVDPRA CLR R1 INDICATES READ

WVDPAD MOV *R13,R2
MOVB @R2LB,@VDPWA WRITE LOW BYTE
SOC R1,R2 READ/WRITE BIT
MOVB R2,@VDPWA
MOV @2(R13),R1 GET CPU RAM ADDRESS
MOV @4(R13),R2 GET BYTE COUNT
B *R11
```

Auto-start

Att välja #5 3 på TI-writer menyn, är det samma som att välja #5 (RUN PROGRAM FILE) på E/A-menyn. Dvs, att programmet går igång direkt efter att det laddats.

För att datorn skall acceptera att programmet körs på detta sätt, krävs att programmet har genomgått den metamorfos som SAVE UTILITY bjuder på.

Och för att denna SAVE metamorfos, skall kunna genomföras, krävs det att programmet tillföres några DEF symboler och label'ar. (se E/A-manualen 24.5).

SAVE-programmet frågar under körningen efter FILE NAME? Mitt logotype program har, under SAVE-körningen, tilldelats namnet DSK1.UTIL1, vilket jag tycker är väldigt bekvämt, eftersom det redan står på skärmen, efter att man gjort sitt menyval. Dessutom ligger det lagrat på samma diskett som TI-writer-programmet.

Hur vet jag nu allt detta?

Jo, det har Anders Persson från Lund, berättat för mig! Därtill har denna person, sänt mig ett gyllenbrunt kuvert, med mycket behövliga subrutiner uti. Och därutöver gett sitt muntliga tillstånd, till att de får publiceras i PB.

TACK Anders!

I listningen nedan, är BYTE-satserna överhoppade!

Det var allt!

GPLWS EQU >83E0
GPLR11 EQU GPLWS+22

KSCAN DATA VDPWS,SCNPGM

SCNPGM LWPI GPLWS
MOV R11,@VDPWS SAVE GPL RETURN ADDRESS IN R0 VDPWS
BL @SCAN
LWPI VDPWS
MOV R0,@GPLR11 RESTORE RETURN ADDRESS
RTWP

DSRLNK DATA DSRWS,DSRPGM COPY OF THE DSRLNK ROUTINE IN ED/AS
FLNBUF DATA 0,0,0,0 FILE NAME BUFFER
DSRWS BSS >20
POINT TEXT '.' FILE NAME SEPARATOR
IDFLAG BYTE >AA VALID DSR HEADER
EVEN
EQUMSK DATA >2000 POSITION OF EQUAL BIT IN ST AND COND BIT IN GPLST
IOERMS DATA >E000 MASK FOR I/O ERROR CODE IN PAB STATUS BYTE

*
* PERFORMS THE I/O. THE EDITOR ASSEMBLER DSRLNK CAN'T BE USED, SINCE THE
* TI-WRITER EDITOR WILL OVERWRITE THE ED/AS UTILITIES.

JNE LOOP

MOV R6,@NMPLNT Restore pointer to name length

MOVB @CLOSE,R1 Change I/O op-code to close
LI R0,PAB
BLWP @VSBW

BLWP @DSRLNK Close file
DATA 8

CLR R0 So that no errors is reported
LWPI GPLWS
B @>0070 Return

***** Samtliga 880 byte's, har utelmnats! *****
***** Om Du tvunget vill printa ut min logotype, *****
***** så kan Du återfinna dem i PB 85-4. *****

*
* GIVES VDP ACCESS ROUTINES FOR PROGRAMS THAT CAN'T USE THE PREWRITTEN
* UTILITIES.

*
* A-DATA 850503
*

SCAN EQU >000E
VDPWD EQU >8800
VDPWD EQU >8C00
VDPWA EQU >8C02

VDPWS BSS >20
R2LB EQU VDPWS+5 LSBY OF VDPWS R2

VSBW DATA VDPWS,VSWPGM
VMBW DATA VDPWS,VMWPGM
VSBW DATA VDPWS,VSRPGM
VMBR DATA VDPWS,VMRPGM
VNTR DATA VDPWS,VTRPGM

* VDP SINGLE BYTE WRITE

VSWPGM BL @WVDPWA WRITE ADDRESS
MOVB @2(R13),@VDPWD
RTWP

* VDP MULTIPLE BYTE WRITE

VMWPGM BL @WVDPWA
VNTMOR MOVB \$R1+,@VDPWD
DEC R2
JNE VNTMOR
RTWP

* VDP SINGLE BYTE READ

```

*
* THIS IS AN EXACT COPY OF THE ED/AS DSRLNK.
* THERE ARE SOME DIFFERENCES COMPARED TO THE DSRLNK USED BY THE EDITOR ITSELF,
* BUT ONLY IN USAGE OF THE PAD, AND THAT IS RESTORED ANYWAY.
*

DSRPGM MOV  #R14+,R5      FETCH DSR MODE
        SZCB @EQUUMSK,R15
        MOV  @NMLPNT,R0
        MOV  R0,R9
        AI   R9,-8
        BLWP @VSBP        READ DEVICE NAME LENGTH

        MOVB R1,R3
        SRL  R3,8
        SETD R4
        LI   R2,FLNBUF

NAMLOP INC  R0            VDF FILENAME POINTER
        INC  R4            FILENAME LENGTH
        C    R4,R3
        JEQ  ENDNAM
        BLWP @VSBP        READ FILE NAME BYTE
        MOVB R1,#R2+
        CB   R1,@POINT
        JNE  NAMLOP
ENDNAM MOV  R4,R4        NAME LENGTH=0 OR FIRST CHAR='.'?
        JEQ  NOTFOU

        CI   R4,7          MAX DEVICE NAME LENGTH
        JGT  NOTFOU
        CLR  @>83D0        USES SOME PAD ADDRESSES, AS WELL AS GPLWS.
        MOV  R4,@>8354    BUT THAT ISN'T THE PROBLEM, SINCE SAVING THESE VALUES
        INC  R4            AND RESTORING THEM BEFORE BRANCHING TO THE EDITOR
        A    R4,@NMLPNT    DOES NOT MAKE ANY DIFFERENCE.

        LWPI GPLWS
        CLR  R1
        LI   R12,>0F00    CRU BASEADDRESS OF PERIPHERALS

NOTVAL MOV  R12,R12
        JEQ  NOTDIS
        SBZ  0            DISABLE CARD
NOTDIS AI   R12,>100      PERIPHERAL SPACE GAP
        CLR  @>83D0
        CI   R12,>2000    END OF CRU AREA?
        JEQ  ENDCRD
        MOV  R12,@>83D0
        SBD  0            ENABLE CARD
        LI   R2,>4000
        CB   #R2,@IDFLAG
        JNE  NOTVAL
        A    @DSRWS+10,R2
        JMP  CHKDSR

WRDEVI MOV  @>83D2,R2    IN CASE OF WRONG DEVICE
        SBD  0
CHKDSR MOV  #R2,R2      DSR LINK
        JEQ  NOTVAL
        MOV  R2,@>83D2
        INCT R2
        MOV  #R2+,R9    START POINT OF NAME
        MOVB @>8355,R5    DEVICE NAME CHAR COUNT
        JEQ  NOCHAR

        CB   R5,#R2+    NAME LENGTH EQUAL?
        JNE  WRDEVI
        SRL  R5,8
        LI   R6,FLNBUF  EXPECTED DEVICE NAME

CMPNAM CB   #R6+,#R2+    COMPARE CHARS
        JNE  WRDEVI
        DEC  R5
        JNE  CMPNAM
NOCHAR INC  R1            CORRECT NAME FOUND
        BL   #R9
        JMP  WRDEVI      CHECK IF MORE DSR'S WITH SAME NAME

        SBZ  0            DISABLE CARD
        LWPI DSRWS
        MOV  R9,R0        READ STATUS BYTE
        BLWP @VSBP
        SRL  R1,13
        JNE  BADIO
        RTWP

ENDCRD LWPI DSRWS
NOTFOU CLR  R1
BADIO  SWPB R1
        MOVB R1,#R13      ERROR CODE TO CALLING R0
        SOCB @EQUUMSK,R15
        RTWP

SLAST  END

```