

** DE 6502 KENNERS ** -- EEN CLUB VOOR 65xx GEBRUIKERS

De vereniging heeft leden in Nederland, België, Duitsland, Frankrijk, Spanje, Portugal, Amerika, Zambia. Het doel van de vereniging is: het bevorderen van de kennisuitwisseling tussen gebruikers van 65xx-computers, zoals KIM, JUNIOR, COMMODORE-64, APPLE CHE-1, PEARCOM, AIM-65, SYM, PET, BBC ATARI, VIC-20, BASIS 108, PROTON-computers, ITT-2020, OSI, ACC 8000, ACORN ELECTRON, SYSTEM 65, PC-100, PALLAS, MINTA FORMOSA, ORIC-1, STARLIGHT, CV-777, ESTATE III, SBC65/68, NCS 6502, KEMPAC System-4, Elektuur SAMSON-65 DOS computer, LASER, etc., etc.

De kennisuitwisseling wordt o.a. gerealiseerd door 5 maal per jaar DE 6502 KENNER te publiceren (1984 en 1985 6 maal zonder contributieverhoging), door het houden van clubbijeenkomsten, door het instandhouden van een cassette-bibliotheek en door het verlenen van paperware-service. Regionale bijeenkomsten worden door leden georganiseerd.

Verschijningsdata DE 6502 KENNER 1985

derde zaterdag van
februari, april, juni,
augustus, oktober, december.

Inlichtingen over de regio- bijeenkomsten:

Gerard van Roekel
Van der Palastraat 11 - C
3135 LK Vlaardingen
Tel.: 010 - 351101

De vereniging is volledig onafhankelijk, is statutair opgericht en ingeschreven bij de Kamer van Koophandel en Fabrieken voor Hollands Noorderkwartier te Alkmaar, onder nummer 634305.

Voorzitter:
Rinus Vleesch-Dubois
Fl. Nightingalestraat 212
2037 N6 Haarlem
Tel.: 023 - 330993

Penningmeester:
John F. van Sprang
Tulp 71
2925 EW Krimpen/IJssel.
Tel.: 01807 - 20589

Leden:
Adri Hankel (05490 - 51151) Hardware/software
Erwin Visschedijk (05490 - 71416) Hardware/software
Promotie
Nico de Vries (010 - 502239) Hardware/software/PET
Erevoorzitter: Siep de Vries
Ereleden : Mw. H. de Vries - Van der Winden
Anton Mueller
Lidmaatschap : Hfl. 45,- per kalenderjaar, postrekening 3757649 t.n.v. Penningmeester KIM Gebruikers Club Ned., Krimpen/IJssel.
Lidm. schap 86: Te voldoen uiterlijk in december 1985.
Advertenties : Tarieven op aanvraag bij de redactie.

Bijeenkomsten van de club

derde zaterdag van
januari, maart, mei,
september, november.

Redactie-adres en informa- ties over paperware etc.

Willeen L. van Pelt
Jacob Jordaensstraat 15
2923 CK Krimpen/IJssel.
Tel.: 01807 - 19881

** DE 6502 KENNER ** -- EEN BLAD VOOR 65xx GEBRUIKERS

DE 6502 KENNER is een uitgave van de KIM Gebruikers Club Nederland. Het blad wordt verstrekt aan leden van de club. DE 6502 KENNER wordt van kopij voorzien door leden van de club, bij de opmaak van een publikatie bijgestaan door de redactie. De inzendingen van programma's dienen voorzien te zijn van commentaar in de listings en zo mogelijk door een inleiding voorafgestaan. Publikatie van een inzending betekent niet dat de redactie of het bestuur enige aansprakelijkheid aanvaardt voor de toepassing ervan. De inzendingen kunnen geschieden in assembly-source-listings, in Basic, in Basicode, Forth, Focal, Comal, Pascal, Fortran, Cobol, Logo Elan, etc. etc.

De leden schrijven ook artikelen over de door hen ontwikkelde hardware en/of aanpassingen daarop. Zij schrijven tevens artikelen van algemene aard of reageren op publikaties van andere inzenders.

DE 6502 KENNER IS EEN BLAD VAN EN DOOR DE LEDEN

Micro-ADE Assembler/Disassembler/Editor is een produkt van Micro Ware Ltd., geschreven door Peter Jennings en bestemd voor alle 6502-computers. De KIM Gebruikers Club Ned. heeft de copyrights verworven nadat ons lid Sebo Woldringh de 4 K KIM-1 versie uitbreidde tot 8 K KIM-1 versie. Adri Hankel paste deze aan voor de JUNIOR. Willeen L. van Pelt stelde een nieuwe 8 K source-listing voor de JUNIOR samen.

De implementatie op andere systemen dan de KIM-1 en JUNIOR kan eenvoudig gebeuren door het aanpassen van de I/O-adressen, welke in de source-listing gemakkelijke te vinden zijn

FATE Foraat-lister/cond. Assembler/Tape-utilities/Editor is de door ons lid Rob Banen geschreven source-listing van een 12 K universeel systeem voor de JUNIOR-computer aan de hand van het universele disk operating systeem van de fa. Proton Electronics te Naarden, nu geschikt voor werken met tapes. FATE wordt beschikbaar gesteld met toestemming van Proton.

In de edities van DE 6502 KENNER worden regelmatig mededelingen gedaan over de door de club georganiseerde bijeenkomsten. Ook worden bestuurlijke mededelingen gedaan, naast informatie over hetgeen in de handel te koop is. Leden die iets te koop hebben of iets zoeken kunnen dit in de edities van DE 6502 KENNER bekend maken. Ook worden wel brieven aan redactie gepubliceerd, evenals specifieke vragen van leden. De edities worden samengesteld op basis van een groot aantal prioriteiten, welke door een redactievergadering worden gehanteerd. Deze vergadering bestaat uit de vaste medewerkers zoals in de colofon vermeld. Het aantal inzendingen is groter dan in een enkele editie van minimaal 48 pagina's is te verwerken. Hierdoor kan het voorkomen dat een inzending eerst na enige tijd kan worden gepubliceerd.

DE CLUB HEEFT BEHOEFTE AAN MEER LEDEN. WIJ WILLEN MEER AAN KUNNEN BIEDEN DAN NU AL HET GEVAL IS. WERF DAAROM EEN LID!

WILT U EEN PRIJSLIJST? STUUR EEN GEFRANKEERDE ENVELOP AAN HET REDAKTIE-ADRES.

Een onafhankelijke jury kent jaarlijks een aantal aanmoedigingspremies toe aan auteurs van gepubliceerde artikelen in DE 6502 KENNER.

De 6502 KENNER is een uitgave van de KIM gebruikers Club Nederland.

Adres voor het inzenden van en reacties op artikelen voor DE 6502 KENNER:
 Willem L. van Pelt
 Jacob Jordaensstraat 15
 2923 CK Krimpen a/IJssel
 Tel.: 01807 - 19881

Vaste medewerkers:
 Willem L. van Pelt
 Gerard van Roekel
 Frans Smeehuijzen
 Jaap van Toledo
 Freelance medewerkers:
 Frans Bakx
 Rob Banen
 Fridus Jonkman
 Gert Klein
 Roger Langeveld
 Anton Mueller
 Gert van Oobroek
 Ruud Uhoff

Gehele of gedeeltelijke overname van de inhoud van DE 6502 KENNER zonder toestemming van het bestuur is verboden. Toepassing van gepubliceerde programma's, hardware etc. is alleen toegestaan voor persoonlijk gebruik.

DE 6502 KENNER verschijnt 6 x per jaar en heeft een oplage van 500 exemplaren.

Copyright (C) 1985 KIM Gebruikers Club Nederland.

De voorpagina is een acuarel van een KIM, geschilderd door:
 Rinus Vleesch Dubois.

In verband met auteurswetgeving en andere maatregelen op het gebied van bescherming van software kan de redactie geen aansprakelijkheid aanvaarde voor inzendingen. Inzendingen dienen afkomstig te zijn van de inzender, tenzij anders aangegeven.

INHOUDSOPGAVE DE 6502 KENNER NR. 39 AUGUSTUS 1985

1. Van de redactie	2.
2. Uitnodiging Landelijke Bijeenkomst Leiden 21 sept 1985	4.
3. <u>COMMODORE 64</u> CBM 64 Tips en Truiks ... Gerard van Roekel	3.
Time of Day Clock; de 6526 nader bekeken ... Ruud H. Uhoff	13.
Automatische Dataregel Teller	46.
4. <u>ACORN ELECTRON</u> Archives Program; een Basic programma ... Simon Voortman	5.
5. <u>BASIC</u> Tokenized Basic Keywords and Addresses CBM 40XX, CBM 80XX ... Nico de Vries	36.
6. <u>DOS65 JUNIOR</u> Another DOS for the JUNIOR ... Coen Kleinoool, France	10.
7. <u>JUNIOR met OS65D V3.3</u> Uitbreidingen op OS65D V3.3 ... Gert Klein	19.
8. <u>JUNIOR</u> Tape I/O ... A.W. den Hartog	33.
9. <u>FORTH</u> FORTH op JUNIOR-computer, deel 3 ... Gert van Oobroek	47.
10. <u>PROTON-computer PC-2</u> Ervaringen met de PC-2 computer van Proton Electronics ... Simon Voortman	35.
11. Vragen van leden	2, 12, 13, 18.
12. Vraag en Aanbod	18.
13. Boekbespreking	12.
14. HCC-computerdagen 16/17 nov 1985 Reductiebon	35.

DE 6502 KENNER omvat per editie minimaal 48 pagina's. In onze club treffen we vogels aan van diverse pluimage. De club is niet alleen een verzameling mensen met allerhande computers rond eenzelfde processor-familie, het is ook de manifestatie van een rijk geschakeerd gebruik van een diversiteit aan programmeertalen en -methoden. Die verscheidenheid wordt nog eens extra omlijst door het voorkomen van alle leeftijden. Jong en oud is actief en leest vooral graag hun blad. Wat mij betreft ontbreekt er nog wel iets essentieels, hetgeen bij de oprichting in 1977 wel degelijk aanwezig was: het vrouwelijk geslacht.

We registreren technisch geschieden, administratief opgeleiden, en anderen. Hun interesses loopen uiteen. Dat alles is voor de redactie van belang om te weten, en een uitdaging om mee te werken, gezien het aantal pagina's. De meeste leden blijken een hogere belangstelling te hebben voor het werken in machinetaal. Niet zo verwonderlijk als je bedenkt dat machinetaal nu eenmaal de grootste verwerkingsnelheid oplevert. Het hoeft ook geen betoog waarom Basic zo populair is. Het is immers de meest aansprekende taal, gemakkelijk te leren en bij de aanschaf van een computer zit deze er meestal reeds in. Na lange tijd van aandrang mijnerzijds heeft ook Forth binnen de club wat meer aandacht gekregen en wordt ook meer gebruikt, omdat het vele malen sneller is dan Basic, al is het aanleren ervan niet voor iedereen even gemakkelijk, en is enig doorzettingsvermogen al voldoende. Merkwaardig is het feit dat de programmeertaal Pascal op het verlanglijstje van velen prijkt, maar er blijkbaar vanwege de kosten nooit afkomt. Je ziet er dan ook vrijwel geen programma's door in het blad, en dat is erg jammer. Ook wat meer aandacht zouden de leden kunnen besteden aan Comal en het publiceren van programma's in die taal. Comal heeft de gebruikersvriendelijke taalkant van Basic gepaard aan het gestructureerd programmeren zoals in Pascal. We zien er echter weinig inzendingen over. Dat behoeft niet zo te zijn, als de leden maar genoeg aandacht daaraan zouden kunnen besteden.

Sommige van onze lezers zijn, naast hun activiteiten met betrekking tot het programmeren, driftig bezig met het ontwikkelen van hardware voor hun systemen. Vooral wat dit betreft is het voor de redactie een gunstiger jaar geweest door het aantal inzendingen op dit gebied. We hebben al de stukken gezien van enkele toekomstige publikaties die er weer mogen zijn. Het vergt veel voorbereiding om het publicatierijp te maken, en veel inspanning van Fridus Jonkman die aan de tekentafel staat, ofschoon hij ook heel graag achter z'n computer zit. Een applausje voor Fridus, die dat voor de inzenders en de lezers over heeft.

Fridus is niet de enige die ook nog werk verricht voor de redactie, de hobby combineren met clubgerichte activiteiten. Je neigt ernaar meteen de bestuurs- en redactieleden in het zonnetje te zetten, en terecht, maar, alhoewel een niet onbelangrijk soort clubdinosaurs met een speciale tik, ik doel eerder op de grote groep altijd nijvere clubmieren die zich voortdurend, bewust of instinctief, bezig houden met het de redactie voorzien van steeds weer publiceerbaar materiaal. Indachtig het gegeven dat daarmee anderen kunnen worden geprikkeld tot het inzenden van hun inspiraties, waardoor zijzelf het genoegen hebben de kennis van elders aangeboden te krijgen. Zoals het mieren ver gaat, hun inspanningen worden eerder op onzichtbare manier beloond dan concreet door de plaatsing van de eigen inzending, waarvoor ze nog veelal hun geduld op de proef gesteld weten. Maar ik blijf hun namen tegenkomen. Ook voor hen mogen de handen wel eens op elkaar.

Toch heeft de redactie niet genoeg. Er is nog veel meer materiaal nodig. Er zijn nog veel meer mensen nodig die medewerking verlenen aan het blad en aan de paperware-service. In de eerste plaats kunnen we voor wat betreft machinetaal denken aan materiaal dat reeds gepubliceerde programma's voor andere dan de eigen machine oazet naar de eigen machine en waaraan zonodig modificaties zijn aangebracht, naast natuurlijk zelf ontwikkelde originele soft-

ware. Dan is er de niet geringe behoefte aan Basic-programma's. Vooral beginnende computerhobbyisten hebben veel aan dit soort publikaties. Hardware-artikelen zijn even welkom. Er zijn ook dokumenterende activiteiten, zoals o.a. tot uitdrukking gebracht in de serie Tokenized Basic Keywords and Addresses, die in een behoefte voorzien. Te denken valt aan het dokumenteren van de assembler van Moser, of de programmeertaal Pascal en al wat dies meer zij. Het betreft hier een bijzonder soort uitoeluiswerk waar niet iedereen tegen bestand is, maar dat toch heel leerzaam kan zijn, en voor anderen heel belangrijk om op gang te komen of om aanpassingen te realiseren.

We hoeven niet altijd te denken aan het ontwikkelen van 12K assembler/editors. Dat hoort er ook bij, maar het kunnen ook minder omvangrijke projecten betreffen. We moeten rekening proberen te houden met het feit dat we beginnende hobbyisten ook een beetje op gang moeten helpen. Dus het sturen van aanpassingen is ook van belang.

De vorige keren heb ik opgeroepen tot het meehelpen aan een aantal belangrijke zaken van de redactie en de paperware-service. Reeds gepubliceerde artikelen moeten worden vertaald naar het Engels. Er zijn echter ook Engelse zaken die vertaling naar het Nederlands behoeven. Hoewel mijn oproep niet zonder reactie is gebleven, ik kan nog meer hulp gebruiken. Immers, het aantal buitenlandse leden is door de ontwikkeling van de Elektuur SAMSON65 DOS-computer in Duitsland nogal opgelopen. Die mensen wil ik ook graag helpen. En nu kan ik keren of wenden, het is alleen mogelijk als de leden van de club gevolg geven aan mijn oproepen.

Zo denk ik ook aan het feit dat het illustreren van het blad een belangrijke zaak is, met name wat de frontpagina betreft. Het is natuurlijk niet zo, dat er onder onze leden niet iemand zou zijn die in staat is voor de frontpagina de plaat te maken die velen van ons leuk zullen vinden of er iets van hun hobby in kunnen herkennen.

En dan nog iets: wie er nog meer ideeën over heeft, die zou ik willen vragen: waar wacht je op?

Het is hier dat ik meteen even kwijt kan dat de redactie blij is met de inzending in Basic voor een Acorn Elektron machine door Simon Voortman, die ook iets te zeggen heeft over de Proton PC-2 computer. En dan onze vriend Drs. Coen Kleipool uit Frankrijk, die de aandacht wil vestigen op het DOS65 voor de JUNIOR-computer, een mooi systeem waar we wel meer van zullen horen, denk ik. In deze editie komt U voor de Commodore-64 een wel wat vreemde eend in de bijt tegen, gezien de listing van het programma van Ruud Uphoff. Niet zo verwonderlijk is het, als men weet dat het om een source gaat die in de Club Assembler Texteditor (CAT) werd geschreven. En, we mogen er gerust een beetje trots op zijn dat we in verband met de activiteiten rond het OS65D V3.3 een artikel van Gert Klein kunnen publiceren dat dat operating-systeem eens flink aanpakte. Een aantal andere publikaties, toch ook van niet geringe betekenis, kunnen het geheel bevredigend larderen.

VERGEET VOORAL NIET UW INZENDINGEN TE DOEN AAN DE REDAKTIE.

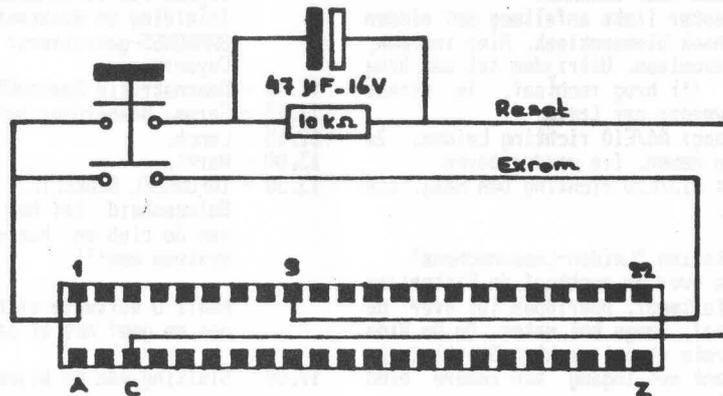
W.L. van Pelt

```
*****
*
*   VRAGEN VAN LEDEN
*
*   Frans Raaijmakers uit Tilburg zit met een probleem.
*   In JUNIOR-boek 2 staat een listing van de monitor-
*   en editorroutines. De monitor is heel simpel. Die
*   zorgt eigenlijk alleen maar voor de sturing van de
*   displays en de afhandeling van het toetsenbord. Ik
*   druk een toets in. De monitor neemt actie. Als het
*   om een adres gaat worden FA en FB bijgewerkt. Als
*   het om data gaat wordt F9 bijgewerkt. Als ik ingeef
*   (AD-0200) (DA-A9), wanneer wordt de adresbus dan
*   0000 0000 0010 000 en de databus 1010 1001 ?
*****
```

===== CBM 64 TIPS EN TRUKS =====

EEN ECHTE RESET VOOR DE COMMODORE 64

Velen van ons zijn inmiddels uit de garantie periode van hun commodore 64. Zodat voor de hardware hobbyisten een aantal tekortkomingen eindelijk verwezenlijkt kunnen worden. Het volgende onderwerp is al in vele variëteiten gepubliceerd in de diverse hobby-bladen. Toch bleven er programma's welke niet te resetten waren en de enige oplossing was om de aan/uit schakelaar te gebruiken. Eigenlijk zijn er maar twee goede oplossingen. Een daarvan is om de resetvektor in de rom te wijzigen. Niet elegant, maar zeker goed werkend. De tweede oplossing is een hardware-matige zoals onderstaande tekening aanseeft. U kunt de reset drukker monteren in de kast en de noodzakelijke verbindingen aanbrengen. Voor wie dit toch niet aandurft is ook een userspoort konnektor te gebruiken en dan aansluiten volgens tekenings.



=====

Als met een kassette rekorder het programma is binnengehaald geeft onderstaand programma dit aan met een pieptoon. Het kan zonder regelnummer worden ingetoetst.

```
POKE54296, 15:POKE54278, 240:POKE54273, 50:POKE54276, 33
```

=====

Indien u gebruikt maakt van schoonmaakschijven voor uw floppy-drive wordt op de verpakking vermeldt dat deze schijf ongeveer 20 seconden moet draaien. Zonder toevoeging van een programma hiervoor zal hij echter na enige seconden weer stoppen en een foutmelding geven. Het programma hieronder helpt u daarbij.

```
10 OPEN 15, 8, 15
20 PRINT#15, "I"
30 INPUT#15, A, B$, C, D
40 N=N+1:IFN<15THEN20
```

=====

Een diskette welke u opnieuw wilt 'FORMATEREN' heeft ongeveer 2,5 minuten hiervoor nodig. Het kan echter veel sneller (ongeveer 15 seconden) indien u gebruik maakt van dezelfde ID code. U toetst dan het volgende in:

```
OPEN15, 8, 15, "N0:NAAM":CLOSE15
```

=====

Het 'SCRATCHEN' van meerdere programma's teselijk wordt mogelijk bij het volgende programma:

```
OPEN15, 8, 15, "S0:NAAM1, NAAM2, NAAM3, NAAMETC":CLOSE15
```

=====

Voor wie nog in de garantie periode zit en toch een redelijke resettoets wil gebruiken kan dit kleine programma zonder regelnummers intoetsen. De 'RESTORE' toets werkt nu als resettoets.

```
POKE792, PEEK(65532):POKE793, PEEK(65533)
```

UITNODIGING CLUBBIJEENKOMST

Datum : Zaterdag 21 september 1985

Lokatie : St. Buurthuis Cornelis Joppensz, Du Rieustraat 4, 2313 GZ LEIDEN. Tel.: 071 - 132462

Entree : Hfl. 10,00 p.p.

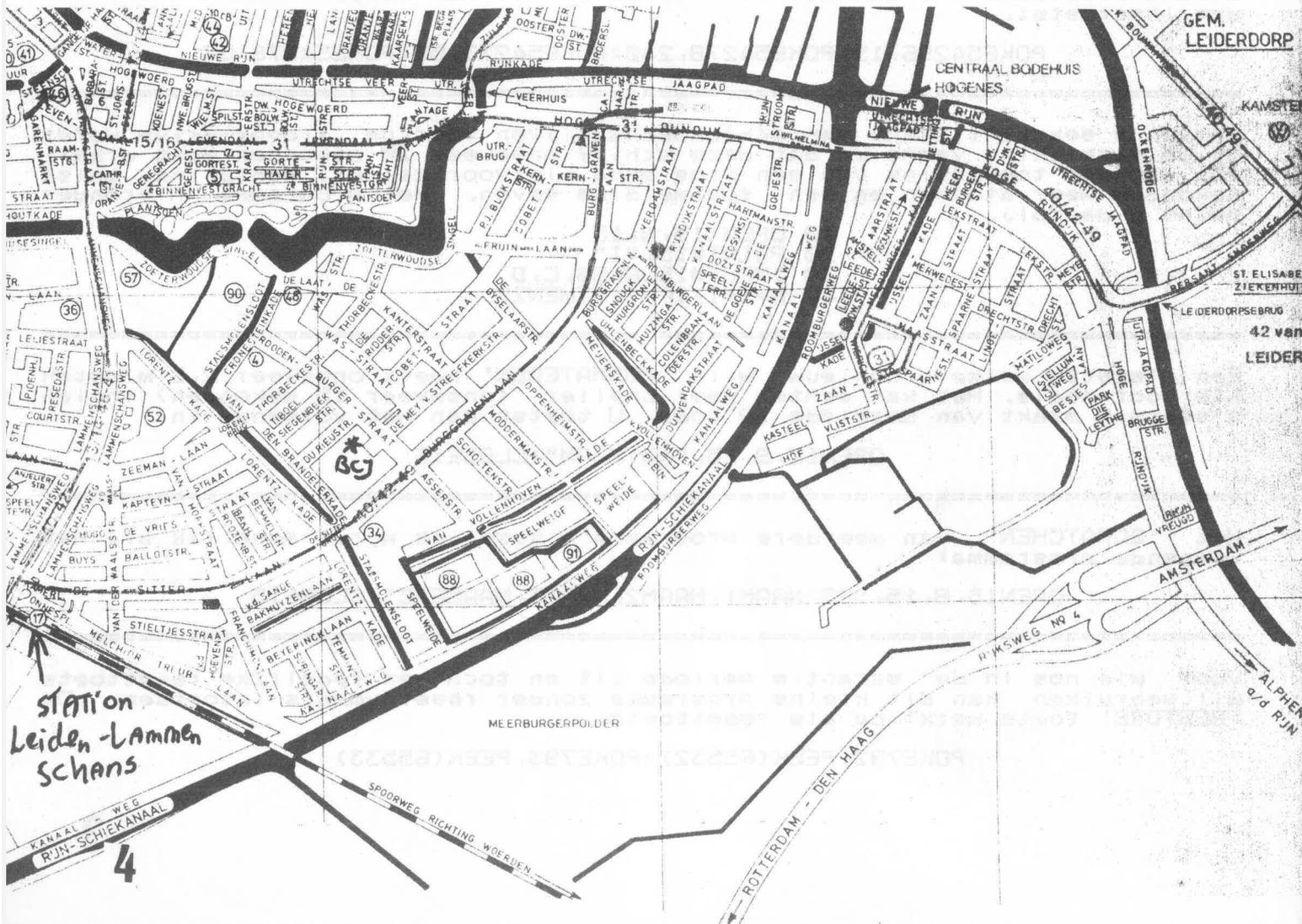
Reisroute :

- per auto - Vanuit A'dam. Op de A4/E10 richting R'dam. 1e afslag Leiden ter hoogte Leiderdorp. Je komt uit op Hoge Rijndijk. Afrijden tot in Leiden. Vlak voor brug links een watertoren. Na die brug loopt vlak langs het water een asfaltweg. **DEZE NIET INSLAAN!** Na ong. 500 meter links asfaltweg met midden berm. Op de hoek bloemenkiosk. Hier inslaan, is de Burgravenlaan. Utrijden tot aan brug en vlak voor (!) brug rechtsaf, 1e straat rechts. Zie verder per trein.
Vanuit Den Haag: A4/E10 richting Leiden. 2e afslag Leiden nemen. Zie verder boven.
Vanuit R'dam: A13/E10 richting Den Haag. Zie verder boven.

- per trein - Uitstappen station "Leiden-Lammenschans". Parkeerplaats over en rechtsaf de Sitterlaan op (brede asfaltweg). Doorlopen tot over de brug en linksaf, langs het water. De Du Rieu straat is eerste straat rechts. Buurthuis is aan rechterkant met ingang aan andere eind van de straat.

PROGRAMMA

- 09.30 - Zaal open. Gastheer: Marcel van Doro
 - 10.00 - Opening
 - 10.15 - **SYSTEMEN ROND DE ELEKTUUR-BUS**
Een rondvlucht boven de oostkant van de JUNIOR en de Elektuur zelfbouwkaarten en koffiedik-kijken via de SAMSON65 DOS-computer.
Inleiding en discussie met twee enthousiaste SAMSON65-gebruikers: Wout van Dinther en Will Cuypers.
 - 11.15 - Demonstratie Samson65.
 - 11.45 - Forum. Bombardeer het forum met Uw vragen.
 - 12.15 - Lunch.
 - 13.00 - Markt.
 - 13.30 - **INFORMEEL GEDEELTE.**
Gelegenheid tot het kennismaken met andere leden van de club en hun systemen. Breng vooral ook Uw systeem mee!!!
- Heeft U software of hardware ontwikkeld? Nee het mee en geef het af aan de redactie.
- 17.00 Sluiting van de bijeenkomst.



ACORN

```
10REM *** Archives program
20REM *** Made by S.Voortman,
30REM *** Beatrixweg 28,
40REM *** 3253 BB Ouddorp
50REM *** (The Netherlands)
60REM *** Phone 01878-3113
70REM *** On an ACORN ELECTRON
80
90REM B%=NUMBER OF RECORDS
100REM C%=COUNTER
110REM TIT#=TITLE STRING,TIM#=TIME STRING
120REM ART#=ARTIST STRING,YEA#=YEAR STRING
130
140B%=0:C%=0
150ON ERROR IF ERR<>17 PROCerror ELSE 180
160PROCswap
170
180MODE6:REPEAT
190PROCmain_program
200IF A%=1THEN PROCnew_input
210IF A%=2THEN PROCchange_records
220IF A%=3THEN PROCsave
230IF A%=4THEN PROCload
240IF A%=5THEN MODE3:PROClist:MODE6
250IF A%=6THEN MODE3:PROCsearch:MODE6
260IF A%=7THEN PROCalphabetical
270IF A%=8THEN PROCadd_records
280IF A%=9THEN PROCdelete_records
290IF A%=10THEN PROCend
300UNTIL FALSE
310
320DEFPROCmain_program:CLS:PRINTTAB(18,3);"MENU"?
330PRINT"You have the following choices:"?
340PRINT"1. New input"
350PRINT"2. Change some records"
360PRINT"3. Save the collection"
370PRINT"4. Load the collection"
380PRINT"5. List the collection"
390PRINT"6. Search on title"
400PRINT"7. Put alphabetical"
410PRINT"8. Add some records"
420PRINT"9. Delete some records"
430PRINT"10 End"?
440INPUT"Select your choice and press RETURN";A%
450ENDPROC
460
470DEFPROCnew_input:CLS
480PRINTTAB(3,3)"New input"?
490INPUT"How much records";B%:PRINT
500IF B%=0 THEN ENDPROC
510C%=0:DIM TIT$(B%+20),ART$(B%+20),TIM$(B%+20),YEA$(B%+20)
520REPEAT:C%=C%+1:PRINT;C%;" ";
530INPUTLINE"Title";TIT$(C%)
540INPUTLINE"Artist";ART$(C%)
550INPUT"Time";TIM$(C%)
560INPUT"Year";YEA$(C%)
570UNTILC%=B%
```

```

580ENDPROC
590
600DEFFPROCchange_records:CLS:C%=0
610PRINTTAB(3,3)"Change some records":IF B%=0 THEN PROCno("change"):ENDPROC
620PRINT"1.Title"'"2.Artist"'"3.Time"'"4.Year"
630INPUT"Select your choice and press RETURN":D%:IF D%<=0 OR D%>4 THEN 630
640INPUT"Wich number":Ch%:IF Ch%=0 ENDPROC
650IF Ch%<0 OR Ch%>B% THEN PRINT"Can't find that record. Try again.":GOTO 620
660ND% GOTO 680,740,800,860 ELSE 620
670
680REM Change title
690PRINT;Ch%;" ";TIT$(Ch%):*FX21,0
700INPUTLINE"New title":TIT$(Ch%)
710IF FNmore THEN 640
720ENDPROC
730
740REM Change Artist
750PRINT;Ch%;" ";ART$(Ch%):*FX21,0
760INPUTLINE"New Artist":ART$(Ch%)
770IF FNmore THEN 640
780ENDPROC
790
800REM Change time
810PRINT;Ch%;" ";TIT$(Ch%);"-";ART$(Ch%);" ";TIM$(Ch%):*FX21,0
820INPUT"New time":TIM$(Ch%)
830IF FNmore THEN 640
840ENDPROC
850
860REM Change year
870PRINT;Ch%;" ";TIT$(Ch%);"-";ART$(Ch%);" ";YEA$(Ch%):*FX21,0
880INPUT"New year":YEA$(Ch%)
890IF FNmore THEN 640
900ENDPROC
910
920DEFFPROCsave:CLS
930PRINTTAB(3,3)"Saving the data":IF B%=0 THEN PROCno("save"):ENDPROC
940I=OPENOUT"records":C%=0:*OPT3,6
950PRINT#I,B%:REPEAT:C%=C%+1
960PRINT#I,TIT$(C%):PRINT#I,ART$(C%)
970PRINT#I,TIM$(C%):PRINT#I,YEA$(C%)
980UNTILC%=B%:CLOSE#I:PRINT'"SAVED"
990PROCwait
1000ENDPROC
1010
1020DEFFPROCload:CLS
1030PRINTTAB(3,3)"Searching for data"
1040I=OPENIN"records":C%=0
1050INPUT#I,B%:IF B% THEN PRINTTAB(3,3);"Loading the data " ELSE ENDPROC
1060DIM TIT$(B%+20),ART$(B%+20),TIM$(B%+20),YEA$(B%+20)
1070REPEAT:C%=C%+1
1080INPUT#I,TIT$(C%):INPUT#I,ART$(C%)
1090INPUT#I,TIM$(C%):INPUT#I,YEA$(C%)
1100UNTILC%=B%:CLOSE#I:PRINT'"LOADED"
1110PROCwait
1120ENDPROC
1130

```

```

1140DEFFPROCadd_records:CLS
1150PRINTTAB(3,3);"Add some records."
1160INPUT"How much records";add%;C%=B%
1170IF add%=0THEN ENDFPROC
1180IF add%>20 THEN PRINT"Not more than 20 new records please!":GOTO 1160
1190REPEAT:C%=C%+1:PRINT;C%;
1200INPUTLINE"Title";TIT$(C%)
1210INPUTLINE"Artist";ART$(C%)
1220INPUT"Time";TIM$(C%)
1230INPUT"Year";YEA$(C%)
1240UNTILC%=B%+add%;B%=C%
1250ENDPROC
1260
1270DEFFPROClist:CLS:page%=15:C%=0
1280VDU19,0,6;0;19,1,0;0;
1290PRINTTAB(3,3);"LIST the collection.":IF B%=0 THEN PROCno("list");ENDPROC
1300PRINT"nr.Title";TAB(41);"Artist";TAB(70);"Time Year"
1310REPEAT:C%=C%+1
1320PRINT;C%;" ";TIT$(C%);TAB(40);"-";ART$(C%);TAB(70);TIM$(C%);TAB(75);YEA$(C%);*FX21,0
1330IF C%=page% THEN page%=page%+15:PRINT'"Press SPACE BAR to continue":REPEATUNTIL GET=32:CLS:PRINTTAB(3,3);"List the collection.":"nr.Title";TAB(41);"Artist";TAB(70);"Time Year"
1340UNTILC%=B%:PRINT'"Press SPACE BAR for MENU":REPEAT:UNTIL GET=32
1350ENDPROC
1360
1370DEFFPROCdelete_records:CLS
1380PRINTTAB(3,3);"Delete some records."
1390IF B%=0 THEN PROCno("delete");ENDPROC
1400INPUT"Number record";C%
1410IF C%<=0 OR C%>B% THEN 1400
1420PRINT;C%;" ";TIT$(C%);"-";ART$(C%)
1430INPUT"Y/N ";ANSW$
1440IF INSTR("Nn",ANSW$) THEN ENDFPROC
1450REPEAT:C%=C%+1
1460TIT$(C%-1)=TIT$(C%);ART$(C%-1)=ART$(C%)
1470TIM$(C%-1)=TIM$(C%);YEA$(C%-1)=YEA$(C%)
1480UNTILC%=B%;B%=B%-1
1490IF FNmore THEN 1400
1500ENDPROC
1510
1520DEFFPROCsearch
1530VDU19,0,6;0;19,1,0;0;:CLS
1540PRINTTAB(3,3);"Search on title.":IF B%=0 THEN PROCno("search");ENDPROC
1550INPUT"Title";tit$
1560found=FALSE:C%=0:REPEAT
1570C%=C%+1:IF found THEN 1590
1580IF TIT$(C%)=tit$ THEN found=TRUE:F%=C%
1590UNTIL C%>=B%
1600IF NOTfound PROCnot_found:GOTO1550
1610IF found PROCfound
1620IF FNmore THEN1550
1630ENDPROC
1640
1650DEFFPROCfound
1660PRINT;F%;" ";TIT$(F%);TAB(38);"-";ART$(F%)
1670ENDPROC
1680

```

```

1690DEFFPROCnot_found
1700PRINT:"Can't find that record. Try again."
1710ENDPROC
1720
1730DEFFPROCalphabetical:CLS
1740PRINTTAB(3,3)"Put in alphabetical order.":IF B%=0 THEN PROCno("alphabetisi
ze"):ENDPROC
1750TIME=0:inc=B% DIV 2
1760REPEAT:IF inc<=0 THEN1810
1770FOR I=inc+1 TO B%
1780PROCpair(I,I-inc)
1790NEXT
1800inc=inc DIV 2
1810UNTIL inc<=0
1820PRINT"it took ";TIME/100;" sec. for ";B%*4;" items."
1830PROCwait
1840ENDPROC
1850
1860DEFFPROCswap(i,j)
1870CALL SWAP%,TIT$(i),TIT$(j)
1880CALL SWAP%,ART$(i),ART$(j)
1890CALL SWAP%,TIM$(i),TIM$(j)
1900CALL SWAP%,YEA$(i),YEA$(j)
1910ENDPROC
1920
1930DEFFPROCerror:CLS
1940PRINT":REPORT:PRINT" at line ";ERL
1950PROCwait
1960ENDPROC
1970
1980DEFFPROCswap
1990FOR EX=0 TO 2 STEP 2:F%=&D00
2000LOFT EX
2010.SWAP% LDA&600:CMP#2:BEQ ok \ No of parameters must be 2
2020JMP &B2BE \ (Arguments)
2030.error JMP&BC0E \ (Type mismatch)
2040.ok LDA&603:CMP&606:BNE error
2050TAY
2060.real CMP#5 :BEQ start
2070.integer CMP#4 :BEQ start
2080.byte CMP#0 :BEQ start2
2090.string CMP#129:BNE error :LDY#4
2100.start DEY
2110.start2 LDX#6
2120.cc LDA&600,X:STA&70,X:DEX:BNE cc
2130.wiss LDA(&71),Y:TAX
2140LDA(&74),Y:STA(&71),Y
2150TXA:STA(&74),Y
2160DEY:BPLwiss
2170RTS
2180] NEXT:ENDPROC
2190
2200DEFFPROCpair(i,j)
2210IF j<=0 THEN ENDPROC
2220IF ART$(i)<ART$(j) THEN PROCswap(i,j):PROCpair(j,j-inc)
2230ENDPROC
2240

```

```

250DEFFPROCwait
260TIME=0:REPEAT:UNTIL TIME=500
270ENDPROC
280
290DEFFPROCno(reason$)
300PRINT"There are no records to ";reason$:PROCwait
310ENDPROC
320
330DEFFNmore
340PRINT"Are there any more records to change":INPUT" <Y/N>";ANSW$
350IF INSTR("Yy",ANSW$) THEN =1 ELSE =0
360
370DEFFPROCend
380CLS:END

```

De Electron heeft de mogelijkheid om fouten te behandelen, en heeft daartoe twee speciale variabelen: ERR en ERL.

ERR geeft het foutnummer van de laatste fout. ERL geeft het reednummer van de laatste fout. ERR 17 is de escape-toets. Als deze toets ingedrukt wordt, springt het programma terug naar het menu. Bij eventuele andere fouten wordt naar de error-procedure gesorogen.

PRINT" = PRINT:PRINT:PRINT

INPUTLINE (r.530) wordt gebruikt om ook titels en/of artiesten met komma's (') binnen te kunnen halen. In plaats van INPUT LINE kan de string met een komma ook met "....." ingevoerd worden (als de Electron een komma tegenkomt bij ingevoerde strings in het INPUT-statement, negeert hij alles rechts van de komma).

Bij PROCno("change") in regel 610 wordt de string 'change' meegegeven bij de PROCno(" ") aanroep. Hierdoor krijgt reason\$ (r.2290) in DEFPROCno(reason\$) de inhoud 'change'.

IF FNmore THEN 640 (r.710). Hier springt de komputer naar DEFFNmore (r.2330).

Regel 2350 is geen fout, maar de manier van BBC-Basic om aan DEFFNmore de waarde 0 of 1 te geven.

In regel 1280 staat nog zo'n leuk BBC-Basic statement: VDU 19,0,6:0:19,1,0:0;

Dit zijn eigenlijk 2 VDU-statements in een: dus VDU 19,0,6:0; en (VDU)19,1,0:0;

Met het eerste statement wordt de achtergrondkleur zwart (=0) geheel-programmeerd in cyaan (=6). Het tweede stuk maakt van de voorgrond kleur wit (=1) de kleur zwart (=0). Het effect van dit alles is dat de hierna volgende tekst op het beeldscherm in zwart op cyaan komt, hetgeen bij 80 karakters op een reed (op een TV) een betere leesbaarheid geeft (naar mijn mening).

Bespreken we nu het alfabetisch sorteren dat plaatsvindt op de artiest. Er wordt een machinetaalroutine gebruikt om het sorteren zo snel mogelijk te maken. (Het sorteren van ong. 85 platen duurt bij mij ca. 14 sec.)

De machinecode wordt aan het begin van het programma (r.160) eenmalig geassembleerd, vanaf de geheugenlokatie &D00 naar boven (tot max. &E00 waar de Basic program area begint). Het 'haakje openen' geeft het begin aan van de machinetaalroutine, het 'haakje sluiten' het einde ervan.

P% is de "pseudo program counter" en wordt door de assembler gebruikt om adressen voor branch en jump instructies te berekenen; tevens het startpunt van de machinecode.

In regel 2010 t/m 2040 wordt gekeken of er twee variabelen van gelijk type aangeboden worden. Zo niet, dan volgt een foutmelding. Er moeten 1, 4 of 5 bytes gewisseld worden. Dit wordt in regel 2050 tot regel 2090 uit het typenummer afgeleid en in Y opgeborgen. Regel 2120 verplaatst de gegevens van het parameterblock op pagina 6 naar het in Basic vrije gedeelte van de zero page. Regel 2130 tot 2160 verwisselen het juiste aantal bytes met tussenooslag in X. De gedachte achter deze manier van swappen is, dat niet de stringinhoud verwisseld wordt (dit zou, als de twee strings niet even lang zijn, geheugenruimte kosten), maar dat in het "string information block" op page 6 de pointers naar de strings verwisseld worden; dit kost geen geheugenruimte.

De regels 2200 tot 2380 geven nog wat DEFPROC's die eerder in het programma aangeroepen worden.

Dan nog het INSTR kommando, om niet te vergeten. Als de string "Yy" in ANSW\$ voorkomt, is INSTR 'waar' (TRUE); en INSTR wordt dus als "Boolean"functie gebruikt (r.2350).

ANOTHER DOS FOR THE JUNIOR

by Coen Kleipool, Val de Périer, F-83310 Cogolin, France. t.(94) 54.43.82

For those of you who think that things come cheaper by the dozen, here is some good news. Another Junior Dos has come out, which makes this Dos65 the fourth of its kind after Elektor's Ohio Dos, Proton Dos and Coen van Nieuwenhoven Dos. Why another Dos? Not so easy to answer but apparently our members were still not satisfied and are willing to improve.

I can wholeheartedly endorse this new Dos65; speed, command structure and flexibility are impressive and are of another class compared to the previous three Dos.

Brouwers Baby. It has been written by our member Ad Brouwer who is a graduate Delft computing student and as such a man of experience on big machines. It is not surprising that this Dos has a lot of similarities with Unix. Unix is basically a multi-user Dos for big systems and Brouwer has boiled this down to a single-user Dos with a simple and easy-to-learn command system. Unix has the reputation to be a powerful tool for the professional and a headache to the occasional user, but this simplified Unix is easy to work with.

Unix, being a multi-user system, has been conceived to work with one or more intelligent terminals and the basic Dos65 also needs a terminal. Later versions were developed to be used with a VDU-card and a screen memory in the computer itself. This solution simplifies matters and reduces costs to some extent, but has the disadvantage that the screen memory occupies valuable computer memory and furthermore microprocessor time is needed to process the screen after each operation. If your computer has only 64 Kbytes of memory, this is a serious consideration and I would myself prefer the terminal solution at any time.

Hardware. To implement Dos65 you will need to construct an Eurocard which carries the 1793 Floppy Controller and a 6821 I/O port. Our member Cees Bootsma has a PC-board available but it is also feasible to produce this in wirewrap or roadrunner. I had a lot of trouble building the PC-board, mainly due to a lack in documentation. The +5 volt and the ground grid need to be improved and some parts do not fit very well in the proper places, but after having made these modifications it works fine.

The 1793 FDC is a very powerful chip. It provides both single and double density (IBM system 34 MFM) and has on-chip track and sector registers which enables automatic sector search. The 372 FDC, used in

Proton Dos, does not have these facilities and is incapable of writing in true double-density.

Double density packs 320 Kbytes on a double-sided 40-tracks diskette and as data are more condensed, you will also get a higher transfer rate ! Dos65 still surprises me with its speed; loading Moser or Micro-ade (8 Kbytes) takes less than 3 seconds and most of this time is spend searching for the proper track. My son uses Proton Dos and together we concluded that Dos65 is at least 4 times as fast.

Software. Dos65 software consists of 4 different parts:

- the monitor
- the kernel
- the utilities
- the editor.

The Monitor. This occupies 4 Kbytes in page F and comprises the basic I/O routines which are very conveniently accessed through jump tables. Monitor commands are special in the sense that they are not needed during normal Dos operation; the enable the user to insert or delete data in memory, call hex- and asccidumps, blockmove and relocate, read and write cassettes, show microprocessor internal registers, search for certain data in memory, etc. All this is very useful during programming and debugging.

The kernel. The kernel is the very cornerstone of the Dos, it is 16 Kbytes long and is read from disk into ram right after startup. Here you will find the vital routines and commands which must be available all the time. There are the typical Unix redirect commands which will allow you to manage in- and output of a command in other directions than the usual keyboard in- and screen out- routines. For instance, you can call input from a disk and redirect the output to the printer or to another disk file, or a modem for that matter.

You can also input commands from a diskfile instead of typing in long and frequent commands on the keyboard such as startup and initialising sequences.

During all these operations there is no need to create, open or close files; Dos65 takes care of all this automatically. Nor is there a "compress" command to fill out empty spaces on your disks, the file handler uses an ISAM (Index Sequential Access Method) to put the file in the first available open space and normally no rearranging is necessary.

At this time Dos65 is incapable of writing 80-tracks disks; in the area where the Track Sector Lists of the 4 open files are located (\$1000 - \$17FF) is insufficient memory space to accommodate 80 tracks, but Ad Brouwer told me he will soon solve this problem.

Utilities. The utilities comprises a number of commands that remain on disk and are only read into memory when needed. Most of them reside in the same place, that is \$ 800 for the smaller and \$ 4000 for the larger programs. Their infrequent use does not merit a permanent stay in active memory. In a Dos computer diskfiles are considered a direct extension of memory and everything is done to occupy the least possible space of active ram which should be available to the user for work in progress.

In any case reading-in a utility only takes a fraction of a second and the software remains in memory until written over by the next utility. The only inconvenience is that it makes the use of two drives almost mandatory. Otherwise a user-drive is not immediately available and you will have to change disks frequently.

But the system can be used with a single diskdrive as a starter. You can read in a utility from the systemdisk, then change disks and execute the utility which now resides in memory with a LC (last command) instruction.

Typical utilities are: delete, list, print or format a file, rename a file, change a filemode, etc. The filemode, which is written on the diskette with the file allows you to protect a file against deleting or writing and to designate it as an ascci-, binary- or command file. You will find this feature very useful once you start using it.

The Brouwer Editor. I will save this item for a next article. Ad Brouwers Editor is a full-screen editor and my 12-year old Hazeltine is not quite ready for its impact. It is a tired old terminal and I will need to inject some fresh blood before he can cope with the full capabilities of this editor.

My comments: A truly excellent Dos which merits to be implemented by all KIM-members. It transforms your Junior into a real professional machine. We will need better hardware and a good manual (especially for Dos beginners) but these are minor problems.

A more serious problem will be to come to a unified Kimclub memory mapping in order to render our club-software compatible. More and more people are scrapping the standard and interface cards, thereby leaving the old junior compatibility. We should now agree on a standard memory-map, starting from page FF downwards.

This should provide us with an uninterrupted ram coverage from \$00 till the start-address of Dos. I have some ideas on this matter and I will come back to this soon.

oooooooooooooooooooo

*****	*****	*****
* VRAGEN VAN LEDEN	* * BOEKBESPREKING	* *
* *****	* * *****	* *
* Ik zoek andere mensen die bezig zijn of bezig zijn	* * Machinetaal 6502.	* *
* geweest met koppeling van MIDI-Synthesizers, zoals	* * Enige tijd geleden is Academic Service met een boek	* *
* DX 7 van Yamaha of de JX 3 P, JX 8 P, Jupiter 6 van	* * op de markt gekomen, een vertaling van het Engelse	* *
* Roland, aan computers.	* * werk van A.P. Stephenson, over het programmeren in	* *
* Zelf zoek ik alle technische specificaties over de	* * machinetaal voor de 6502-processor.	* *
* MIDI-bus en schema's over reeds bestaande interfaces	* * Het geeft een opsomming van de diverse instructies,	* *
* interfaces.	* * waaruit de set van de 6502 bestaat. Het doel van de	* *
* Stuur Uw reactie(s) naar	* * instructies wordt uitgeleend en er worden voorbeelden	* *
* Stefan Soerling	* * gegeven hoe zij gebruikt kunnen worden.	* *
* Waversesteeg 88 A	* *	* *
* B - 1990 HOEILAART	*****	*****
* of naar het adres van de redactie.		

COMMODORE 64 "TIME OF DAY CLOCK"

Ruud Uphoff

IN PLAATS VAN HET VERTROUWDE VIA (6522) IS DE C64 UITGERUST MET TWEE SOORTGELIJK CHIP'S VAN HET TYPE 6526. HET LEEK MIJ GOED DEZE CHIP EENS WAT NADER TOE TE LICHTEN, AANGEZIEN ER ZEER MOOIE DINGEN MEE TE DOEN ZIJN DIE NIET ZO UITVOERIG IN DE HANDBOEKEN VAN COMMODORE BESCHREVEN ZIJN. EEN DAARVAN IS DE "TOD" WAARVAN WE IN BIJGAAND PROGRAMMA GEBRUIK MAKEN.

Voordat U nu alleen de object-code gaat intikken moet ik U een kleine uitdaging doen: Er zit een geweldige blunder in het programma die ik pas na het afdrucken van de source ontdekte. Ik heb hem er toen toch maar in laten zitten, omdat het een prachtig voorbeeld is hoe je ondanks het gebruik van structuren toch nog helemaal de mist in kunt gaan als je tegen een eenvoudige vuistregel zondigd! Aan U om mij te laten weten wat die blunder nu precies is. Er is geen prijs aan verbonden, maar wel zullen we de naam van de eerste goede inzender bekend maken. Laat het weten aan de redactie!

Het programma wordt in de interruptroutine opgenomen met SYS 9*4096 en de klok kan worden gelijk gezet na SYS 9*4096+3. Steeds als U nu de F1-toets ingedrukt houdt, zal de juiste tijd in de linker bovenhoek van het scherm te zien zijn. Na het loslaten van F1 ziet het scherm er weer uit zoals tevoren. Er gaat dus geen informatie van het scherm verloren. Na gebruik van de RESTORE toets is de klok afgekoppeld, maar staat niet stil en U kunt hem met SYS 9*4096 weer aankoppelen.

VEEL PLEZIER

ZIE VOOR LISTING VOLGENDE PAGINA'S

VRAGEN VAN LEDEN

(Stuur het antwoord of de oplossing aan de redactie. deze zorgt ervoor dat de inzender op de hoogte wordt gesteld en er wordt meteen bekeken of erover gepubliceerd kan worden. zodat ook anderen meteen geïnformeerd kunnen worden.)

Marcel van Dorp heeft de volgende wijziging in zijn Elektuur computer JUNIOR aangebracht:

- de adresdecodering is vervangen door een volledige adresdecoder in blokken van vier K, waarbij twee blokken zijn uitgepoord tot 8 blokken van 1 K (K0..K7);
- de standaardmonitor zit nu op FC00..FFFF;
- op de plaats van de Erom op de standaardkaart zit nu een 2016 (2 K statische RAM);
- de 6502 is vervangen door de 65C02 van Rockwell.

Als ik nu een programma oosla op pagina 2,3,1C,1D,1E of 1F (2016) dan gaat dat uitstekend. Als ik het programma laat lopen, dan worden er echter een heleboel bytes veranderd! Dit gebeurt telkens met dezelfde geheugenplaatsen. Een programma in PIA-Ram heeft echter nergens last van!!!

De voedingsspanning heb ik gecontroleerd. Die heeft een waarde van 4,84 V.

Hoe kan dit? Hoe moet ik dit herstellen?

Marcel heeft nog een vraag!

Wie heeft er informatie sociaal gericht op een 8 inch drive gekoppeld aan een BBC, model B?

Marcel van Dorp heeft nog meer te vragen.

Hi, heeft een IBM 3742 gekregen. Dit apparaat bestaat uit twee 8 inch drives, twee toetsenborden, een monitor welke met behulp van een snoegel in tweeën is gedeeld, een flinke voeding, en een microprocessor met bijbehorend geheugen en I/O.

Alle IC's zijn eigen fabrikaat, zodat ik niet weet wat wat is.

Nu wil ik zo'n drive aan mijn JUNIOR hangen, maar ik weet niet hoe. Wat ik wil weten is het volgende:

- Wat doet een floppy disk controller, en kan het ook zonder?
- Wat is het verschil tussen 5 inch en 8 inch?
- Wat is het verschil tussen OHIO en IBM - forraat?

40 Grafische programma's voor de Commodore 64; Leer programmeren met hoge resolutie in Basic. Een nieuw boek vertaald door Nok van Veen. 160 pagina's Hfl. 29,50. ISBN 90 6233 149 1

```

0010 ;* * * * *
0020 ;* COMMODORE 64 PRECISION CLOCK *
0030 ;* (C) RUUD UPHOFF 1985 *
0040 ;* * * * *
0050 ;
0060 ;KERNAL ROUTINES
0070 ;
0080 KERNAL.CHROUT DEF $FFD2
0090 KERNAL.CHRIN DEF $FFCF
0100 KERNAL.MEMTOP DEF $FF99
0110 KERNAL.SCNKEY DEF $FF9F
0120 ;
0130 ;KERNAL PAGE ZERO
0140 ;
0150 LAST.KEY DEF $00C5
0160 QUOTE.MODE DEF $00D4
0170 ;
0180 ;KERNAL ABSOLUTE
0190 ;
0200 SCREEN.AREA DEF $0400
0210 COLOR.AREA DEF $D800
0220 CHARACTER.COLOR DEF $0286
0230 IRQ.RAM.VECTOR DEF $0314
0240 ;
0250 ;CIA #1 LOCATIONS (T.O.D. CLOCK)
0260 ;
0270 TENTH.SECONDS DEF $DC08
0280 SECONDS DEF $DC09
0290 MINUTES DEF $DC0A
0300 HOURS DEF $DC0B
0310 CLOCK.FREQUENCY DEF $DC0E
0320 CLOCK.MODE DEF $DC0F
0330 ;
0340 ;CONSTANTS
0350 ;
0360 BLANK DEF $20
0370 CRLF DEF $0D
0380 INVALID DEF $FF
0390 F1.KEY DEF $04
0400 ;
0410 ORG $9000
0420 STR
0430 OBJ
0440 ;
9000 4C1B90 0450 ENTRY GOTO INIT.CLOCK
9003 4CA790 0460 SET GOTO SET.TIME
0470 ;
9006 20CFFF 0490 EXEC KERNAL.CHRIN
9009 C93A9006 0500 CASE CARRY CMP #'9+1
900D 204191 0510 EXEC SET.BYTE.INVALID
9010 4C1A90C930B003 0520 WHEN /CARRY CMP #'0
9017 204191 0530 EXEC SET.BYTE.INVALID
0540 ENCASE
901A 60 0550 ENPROC

```

```

0560
0570
0580
901B AD1403 0590
901E C942F008 0600
9022 807791 0610
9025 A942 0620
9027 801403 0630
0640
902A AD1503 0650
902D C990F008 0660
9031 807891 0670
9034 A990 0680
9036 801503 0690
0700
9039 A200 0710
903B A090 0720
903D 18 0730
903E 2099FF 0740
9041 60 0750
0760
0770
9042 206391 0780
9045 20E490D02F 0790
904A A200 0800
904C AD0BDC 0810
904F 08 0820
9050 297F 0830
9052 F8 0840
9053 C912D003 0850
9057 38 0860
9058 E912 0870
0880
0890
905A 281003 0890
905D 18 0900
905E 6912 0910
0920
0930
9060 D8 0930
9061 209E90 0940
9064 AD0ADC 0950
9067 209E90 0960
906A AD09DC 0970
906D 208290 0980
9070 AD08DC 0990
9073 209FFF 1000
9076 4C4590 1010
9079 20ED90 1020
907C 004F91 1030
907F 6C7791 1040
1050

```

```

PAG
;
PROC INIT.CLOCK
LDA IRQ.RAM.VECTOR
IF /ZERO CMP #<CLOCK
STA IRQ.COPY
LDA #<CLOCK
STA IRQ.RAM.VECTOR
ENIF
LDA IRQ.RAM.VECTOR+1
IF /ZERO CMP #>CLOCK
STA IRQ.COPY+1
LDA #>CLOCK
STA IRQ.RAM.VECTOR+1
ENIF
LDX #<ENTRY
LDY #>ENTRY
CLC
EXEC KERNAL.MEMTOP
ENPROC
;
PROC CLOCK
EXEC SAVE.CLOCK.AREA
WHILE ZERO EXEC TEST.F1
LDX #0
LDA HOURS
PHP AM/PM FLAG IN BIT 7
AND #*7F
SED
IF ZERO CMP #*12 A.M.=00 A.M.
SEC
SBC #*12
ENIF
IF MINUS PLP THUS P.M.
CLC
ADC #*12 FOR P.M.
ENIF
CLD
EXEC DISPLAY.WITH.COLON HOURS
LDA MINUTES
EXEC DISPLAY.WITH.COLON MINUTES
LDA SECONDS
EXEC DISPLAY.WITHOUT.COLON SECONDS
LDA TENTH.SECONDS
EXEC KERNAL.SCNKEY
ENWHILE
EXEC ENSURE.CLOCK.MODE
EXEC REMOVE.CLOCK
JMP (IRQ.COPY)
POP

```

	1060	PAG
	1070	;
	1080	PROC DISPLAY.WITHOUT.COLON
9082 48	1090	PHA
9083 4A	1100	LSR
9084 4A	1110	LSR
9085 4A	1120	LSR
9086 4A	1130	LSR
9087 209190	1140	EXEC DISPLAY.A.NUMBER
9088A 68	1150	PLA
9088B 290F	1160	AND #0F
908D 209190	1170	EXEC DISPLAY.A.NUMBER
9090 60	1180	ENPROC
	1190	;
	1200	PROC DISPLAY.A.NUMBER
9091 09B0	1210	ORA #00 TO SCREEN MODE
9093 9D0004	1220	STA SCREEN.AREA,X
9096 AD8602	1230	LDA CHARACTER.COLOR
9099 9D00D8	1240	STA COLOR.AREA,X
909C E8	1250	INX
909D 60	1260	ENPROC
	1270	;
	1280	PROC DISPLAY.WITH.COLON
909E 208290	1290	EXEC DISPLAY.WITHOUT.COLON
90A1 A90A	1300	LDA #' :-30
90A3 209190	1310	EXEC DISPLAY.A.NUMBER (COLON)
90A6 60	1320	ENPROC
	1330	;
	1340	PROC SET.TIME
	1350	REPEAT
90A7 20FE90	1360	EXEC SET.HOURS
90AA C924B0F9	1370	UNTIL /CARRY CMP #24
90AE C900D007	1380	CASE ZERO CMP #00
90B2 A912	1390	LDA #12 USE 12H WITH P.M. FLAG
90B4 0980	1400	ORA #80
90B6 4CC490C9139007	1410	WHEN CARRY CMP #13 (PM)
90BD F8	1420	SED
90BE 38	1430	SEC
90BF E912	1440	SBC #12 SET AM WITH PM FLAG
90C1 D8	1450	CLD
90C2 0980	1460	ORA #80 PM FLAG
	1470	ENCASE
90C4 8D0BDC	1480	STA HOURS
	1490	REPEAT
90C7 200491	1500	EXEC SET.MINUTES
90CA C960B0F9	1510	UNTIL /CARRY CMP #60
90CE 8D0ADC	1520	STA MINUTES
	1530	REPEAT
90D1 200A91	1540	EXEC SET.SECONDS
90D4 C960B0F9	1550	UNTIL /CARRY CMP #60
90D8 8D09DC	1560	STA SECONDS
90DB A900	1570	LDA #00
90DD 8D0BDC	1580	STA TENTH.SECONDS
90E0 20ED90	1590	EXEC ENSURE.CLOCK.MODE
90E3 60	1600	ENPROC

```

1610
1620
1630
90E4 ASC5      1640
90E6 C904D002  1650
90EA ASD4      1660
1670
90EC 60        1680
1690
1700
90ED AD0EDC    1710
90F0 0980      1720
90F2 8D0EDC    1730
90F5 AD0FDC    1740
90F8 29F7      1750
90FA 8D0FDC    1760
90FD 60        1770
1780
1790
90FE A948      1800
9100 201091    1810
9103 60        1820
1830
1840
9104 A94D      1850
9106 201091    1860
9109 60        1870
1880
1890
910A A953      1900
910C 201091    1910
910F 60        1920
1930
1940
9110 203391    1950
9113 20D2FF    1960
9116 20D2FF    1970
9119 203B91    1980
911C 2006903011 1990
9121 00        2000
9122 00        2010
9123 00        2020
9124 00        2030
9125 808991    2040
9128 2006903005 2050
912D 290F      2060
912F 008991    2070
2080
2090
9132 60        2100

```

```

PAG
;
PROC TEST.F1
  LDA *LAST.KEY
  IF ZERO CMP #F1.KEY
  LDA *QUOTE.MODE
  ENIF
ENPROC
;
PROC ENSURE.CLOCK.MODE
  LDA CLOCK.FREQUENCY
  ORA ##80 (50 HZ)
  STA CLOCK.FREQUENCY
  LDA CLOCK.MODE
  AND ##F7 (NO ALARM)
  STA CLOCK.MODE
ENPROC
;
PROC SET.HOURS
  LDA #'H
  EXEC GET.DECIMAL.BYTE
ENPROC
;
PROC SET.MINUTES
  LDA #'M
  EXEC GET.DECIMAL.BYTE
ENPROC
;
PROC SET.SECONDS
  LDA #'S
  EXEC GET.DECIMAL.BYTE
ENPROC
;
PROC GET.DECIMAL.BYTE
  EXEC PRINT.CRLF
  EXEC KERNAL.CHROUT
  EXEC KERNAL.CHROUT
  EXEC PRINT.A.BLANK
  IF /MINUS EXEC GET.DECIMAL.DIGIT
  ASL
  ASL
  ASL
  ASL
  STA TEMPORARY
  IF /MINUS EXEC GET.DECIMAL.DIGIT
  AND ##0F
  ORA TEMPORARY
  ENIF
ENIF
ENPROC

```

	2110	PAG
	2120	;
	2130	PROC PRINT.CRLF
9133 48	2140	PHA
9134 A90D	2150	LDA #CRLF
9136 20D2FF	2160	EXEC KERNAL.CHROUT
9139 68	2170	PLA
913A 60	2180	ENPROC
	2190	;
	2200	PROC PRINT.A.BLANK
913B A920	2210	LDA #BLANK
913D 20D2FF	2220	EXEC KERNAL.CHROUT
9140 60	2230	ENPROC
	2240	;
	2250	PROC SET.BYTE.INVALID
9141 203B91	2260	EXEC PRINT.A.BLANK
9144 A93F	2270	LDA #'?
9146 20D2FF	2280	EXEC KERNAL.CHROUT
9149 203391	2290	EXEC PRINT.CRLF
914C A9FF	2300	LDA #INVALID
914E 60	2310	ENPROC
	2320	;
	2330	PROC REMOVE.CLOCK
914F A200	2340	LDX #000
	2350	REPEAT
9151 B07991	2360	LDA SAVE.AREA.1,X
9154 900004	2370	STA SCREEN.AREA,X
9157 B08191	2380	LDA SAVE.AREA.2,X
915A 900008	2390	STA COLOR.AREA,X
915D E8	2400	INX
915E E008D0EF	2410	UNTIL ZERO CPX #8
9162 60	2420	ENPROC
	2430	;
	2440	PROC SAVE.CLOCK.AREA
9163 A200	2450	LDX #000
	2460	REPEAT
9165 B00004	2470	LDA SCREEN.AREA,X
9168 907991	2480	STA SAVE.AREA.1,X
916B B00008	2490	LDA COLOR.AREA,X
916E 908191	2500	STA SAVE.AREA.2,X
9171 E8	2510	INX
9172 E008D0EF	2520	UNTIL ZERO CPX #8
9176 60	2530	ENPROC
	2540	;
9177	2550	VAR 2
9179	2560	VAR 8
9181	2570	VAR 8
9189	2580	VAR 1
	2590	;
	2600	END
918A 918A 00		

TE KOOP:

=====

FLOPPY-DRIVE KAST VOOR 2 X
 NORMALE DRIVES OF 4 X SLIM-
 LINE. MET INGEBOUWDE VOEDING
 5V-5A EN 12V-5A.
 PRIJS: HFL. 60,-.
 TEL.: 01660 - 3560

8" FDOS CONTROLLERKAART MET
 FD 1771 HFL. 90,00
 2 X 8K RAM/ROM KAART MET 2 X
 8K RAM EN IN EPROM'S FDOS1.2
 VOOR 8" HFL. 100,00
 VERKOOP TEN BEHOEVE VAN EEN
 BELGISCH LID.
 VOOR INFORMATIE OF BEZICHTI-
 GING:
 TEL.: 01807 - 19881
 VERZENDKOSTEN ZIJN NIET IN
 DE PRIJS INBEGREPEN.

=====

ALOIS OTTER
 NEUNKIRCHEN, OESTERREICH

ICH HABE DEN SAMSON COMPUTER
 FERTIG AUSGEBAUT. SCHREIBE
 DERZEIT AUF DEM WORDPROZES-
 SOR VON DER LOYS SYSTEMDIS-
 CETTE VON ELECTOR.
 FOLGENDE PUNKTE WAEREN FUER
 MICH BESONDERS WICHTIG:
 1) WIE ARBEITET MAN MIT DEM
 EDITOR?
 2) WIE WERDEN PROGRAMME AUF
 DISK GESPEICHERT OD. VON
 DISK GELADEN?
 3) WIE WERD EIN PROGRAMM GE-
 STARTET?
 4) WELCHE SYSTEMSPEZIFISCHEN
 PUNKTE SIND AUSSERDEM
 NOCH ZU BEACHTEN?

MEIN ENGLISCH IST LEIDER
 NICHT SO GUT. BITTE SCHREIBE
 IHRE ANTWOORT AN DIE REDAK-
 TION IN DEUTSCH.
 ICH WUERDE MICH SEHR FREUEN
 WENN ICH AUF DIESE FRAGEN
 EINE ANTWOORT BEKOMMEN.

=====

PR UITBREIDINGEN OP OS-65D V3.3

G. Klein
Diedenweg 119
6706 CM WAGENINGEN

In nov/dec 1982 publiceerde ELEKTUUR een ontwerp voor een floov-interface ten behoeve van de JUNIOR computer. Deze interface werd ontworpen voor het OHIO SCIENTIFIC OS-65D operating system. Naast een aantal sterke punten heeft OS-65D ook een aantal uitgesproken zwakke kanten. Een groot aantal DOS functies als het schrijven, het verwijderen en veranderen van filenamen in de directory, het formateren van een diskette en het uitlijsten van een directory, kunnen alleen met behulp van het BASIC programma BEXEC* uitgevoerd worden. Uiteraard is dit erg onhandig omdat het betekent dat de BASIC interpreter altijd in het geheugen moet staan en dat een eventueel niet ingetoetst programma overschreven wordt. De hier beschreven uitbreidingen op het DOS elimineren deze tekortkomingen goeddeels.

NIEUWE DOS COMMANDO'S

De volgende commando's worden aan het DOS toegevoegd :

DI
Geeft de directory van een diskette

SD <tracknummer>
Vervangt het oude DI commando in OS-65D

FO
Formateert een diskette en schrijft een lege directory op track 12

CR <filenaam>=<eerste track>,<laatste track>
Schrijft een nieuwe filenaam in de directory met bijbehorende track range. Een filenaam mag maximaal 6 karakters lang zijn. Het eerste karakter moet een letter zijn (ASCII)= \$41). Dit commando doet ook nog een controle op onbestaanbare of onlogische tracknummers. De tracks van de file worden niet geinitialiseerd.
Voorbeeld : CR DATA1=01,03
Schrijft een file DATA1 met een lengte van 3 tracks in de directory.

DE <filenaam>
Verwijdert een filenaam uit de directory

CH <filenaam1>=<filenaam2>
Verandert filenaam1 in filenaam2

NU
Geeft het aantal tracks dat de op dat moment aanwezige BASIC of assembler sourcefile in beslag neemt.

Dan nog vier commando's die van belang zijn bij verdere uitbreidingen.

FT
Laadt de FORTH interpreter/comoiler

WF
Warme start FORTH

MA
Laadt de macro assembler van C. Moser

WS
Warme start van de macro assembler

De uitbreidingen nemen ongeveer 1.5 K in beslag en worden door een aangepaste bootstraploader vanaf adres \$E000 in het geheugen

geladen. Dit gebied mag u beslist niet meer beschrijven om het DOS te laten werken.

IMPLEMENTEREN

Om de DOS uitbreidingen op uw systeem werkend te krijgen moet u de volgende procedure volgen :

1. Copieer een syteemdisk (tutorial disk V).
2. Zorg ervoor dat de DOS uitbreidingen vanaf adres \$E000 in het geheugen staan.
3. Doe de copieschijf in drive A en spring naar het DOS.
4. Lees een lege directory in :
A*CA 2E79=11,3 (return)
5. Reset het systeem en breng met het hexadecimale toetsenbord van de JUNIOR de volgende wijzigingen aan :

adres	data
2E79	4F 53 36
	35 44 33
	00 14
2A4B	20 12 E5
2A8B	BD 00 E0
2A95	BD 01 E0
2A9C	BD 03 E0
2AA0	BD 02 E0

En keer met de (PC) toets terug naar het DOS

6. Schrijf de directory weg :
A*SA i2,1=2E79/1 (return)
Schrijf track 1 weg :
A*SA 01,1=2A00/8 (return)
Schrijf de uitbreidingen op track 14 :
A*SA i4,1=E000/8
7. Lees het track 0 R/W utility in :
A*CA 0200=06,4 (return)
A*GD 0200 (return)
Na 'COMMAND' toetst u in :
R4200 (return)
En na nog eens 'COMMAND' :
E (return)
8. Reset het systeem. Nu moet u vanaf adres \$4200 de aangepaste bootstrap intoetsen. Het handigste gaat dit waarschijnlijk met de hand via het toetsenbord van de JUNIOR. Als alles ingetoetst en gecontroleerd is gaan we terug naar het DOS via de (PC) toets.
9. Start opnieuw het track 0 R/W utility :
A*GD 0200 (return)
Na 'COMMAND' toetst u in :
W4200/2200,8 (return)
Als dat gebeurd is moet het mogelijk zijn de comouter met de nieuwe syteemdisk op te starten :
<RST> FF17 <GO> <RUB OUT>
Het systeem meldt zich nu met :
DS-65D V3.4
A*
Nu schrijven we twee filenamen in de directory :
A*CR BEXEC*=15,17 (return)
A*CR COPIER=18,19 (return)
Vervolgens laden we BASIC :
A*BA (return)
10. Na de Ok prompt stoppen we de oude svsteemdisk in drive A en geven het commando RUN"BEXEC*" . We kiezen optie 9 om het syteem te unlocken en stoppen de nieuwe schijf in drive A. Met het commando DISK!"PU BEXEC*" schrijven we de file weg. Tenslotte laden we nog het programma COPIER van de oude systeemschijf en schrijven deze ook op de nieuwe systeemschijf weg.

Of u de overige utilities ook nog op de nieuwe schijf wilt zetten mag u zelf uitmaken. In ieder geval is het nu makkelijk genoeg om de benodigde filenamen in de directory te zetten.

//

PASS 2

```

0010 :*** UITBREIDINGEN OP OS-65D V3.3 ***
0020 :
0030 :GERT KLEIN
0040 :DIEDENWEG 119
0050 :6706 CM WAGENINGEN
0060 :
0070 :***
0080 :
0090 :
0100 :FLOPPY LOAD/WRITE POINTER
0110 :
0120 VECL .DE $FE
0130 VECH .DE $FF
0140 :
0150 :ADRESSEN IN PAGINA 0
0160 :
0170 TEMP1 .DE $E0
0180 TEMP2 .DE $E4
0190 TEMP3 .DE $E5
0200 TEMP4 .DE $E6
0210 :
0220 :ADRESSEN IN OS-65D
0230 :
0240 SECTNUMB .DE $265E TE LEZEN/SCHRIJVEN SECTOR
0250 PAGENUMB .DE $265F AANTAL TE SCHRIJVEN PAGINA'S
0260 RAMLOC .DE $2300 BEVAT HOOGSTE PAGINA IN RAM
0270 LOADHEAD .DE $2754 ZET DE KOP OP DE DISK
0280 SETTK .DE $268C STAPT NAAR TRACK IN ACCU
0290 INITS .DE $2768 INITIALISEERT EEN DISK
0300 HOME .DE $2663 STUURT DE KOP NAAR TRACK 0
0310 SELDRIVE .DE $29C6 SELECTEERT DRIVE IN ACCU
0320 READSECTOR .DE $2967 LEEST EEN SECTOR IN
0330 DUMPSECTOR .DE $27E1 SCHRIJFT EEN SECTOR WEG
0340 UNLOAD .DE $2761 HAALT DE KOP VAN DE DISK AF
0350 ERR? .DE $2A4B DOS FOUTMELDINGSROUTINE
0360 KERNEL .DE $2A51 DOS COMMANDO VERTALER
0370 DIRAD .DE $2E79 ADRES VAN DE DIRECTORY BUFFER
0380 :
0390 :
0400 RECCHA .DE $2340 ALGEMENE INPUT ROUTINE
0410 PRCHA .DE $2343 ALGEMENE OUTPUT ROUTINE
0420 CRLF .DE $2D6A CARRIAGE RETURN/LINE FEED
0430 PRBYTE .DE $2D92 PRINT EEN BYTE
0440 STROUT .DE $2D73 PRINT EEN STRING
0450 READBUFFER .DE $2CE4 LEEST EEN KARAKTER UIT DE INPUTBUFFER
0460 CHECK .DE $2D5B CONTROLE OP EEN "." IN BUFFER
0470 DISTR .DE $2AC6 ;EEN I/O DISTRIBUTOR
0480 :
0490 :
0500 :*** DOS UITBREIDINGEN BEGINNEN HIER ***
0510 :
0520 .BA $E000
0530 :
0540 :*** AANGEPASTE SPRONG TABEL ***
0550 :
E000- 57 53 E2 0560 CONTTABLE .BY 'WS' $E2 $22
E003- 22
E004- 52 45 FC 0570 .BY 'RE' $FC $2B
E007- 2B
E008- 45 4D 2E 0580 .BY 'EM' $2E $2B
E00B- 2B
E00C- 41 53 DD 0590 .BY 'AS' $DD $2A
E00F- 2A
E010- 42 41 E5 0600 .BY 'BA' $E5 $2A
E013- 2A
E014- 43 41 10 0610 .BY 'CA' $10 $2B
E017- 2B
E018- 44 39 BF 0620 .BY 'D9' $BF $2A

```

```

E01B- 2A
E01C- 53 44 28 0630 .BY 'SD' $28 $2B
E01F- 2B
E020- 45 58 36 0640 .BY 'EX' $36 $2B
E023- 2B
E024- 47 4F 45 0650 .BY 'GO' $45 $2B
E027- 2B
E028- 48 4F 62 0660 .BY 'HO' $62 $26
E02B- 26
E02C- 49 4E 54 0670 .BY 'IN' $54 $2B
E02F- 2B
E030- 49 4F 82 0680 .BY 'IO' $82 $2B
E033- 2B
E034- 4C 4F A6 0690 .BY 'LD' $A6 $2B
E037- 2B
E038- 4D 45 C5 0700 .BY 'ME' $C5 $2B
E03B- 2B
E03C- 50 55 DC 0710 .BY 'PU' $DC $2B
E03F- 2B
E040- 58 51 21 0720 .BY 'XQ' $21 $2C
E043- 2C
E044- 53 41 27 0730 .BY 'SA' $27 $2C
E047- 2C
E048- 53 45 42 0740 .BY 'SE' $42 $2C
E04B- 2C
E04C- 46 4F 0750 .BY 'FO'
E04E- 75 E1 0760 .SI FORMAT-1
E050- 44 49 0770 .BY 'DI'
E052- 98 E0 0780 .SI DIRECT-1
E054- 43 52 0790 .BY 'CR'
E056- A5 E2 0800 .SI CREATE-1
E058- 44 45 0810 .BY 'DE'
E05A- 68 E4 0820 .SI DELETE-1
E05C- 43 48 0830 .BY 'CH'
E05E- D5 E4 0840 .SI CHANGE-1
E060- 4E 55 0850 .BY 'NU'
E062- 24 E6 0860 .SI NUMBER-1
E064- 46 54 0870 .BY 'FT'
E066- 49 E6 0880 .SI FORTH-1
E068- 57 46 0890 .BY 'WF'
E06A- 62 E6 0900 .SI WFORTH-1
E06C- 4D 41 0910 .BY 'MA'
E06E- 76 E6 0920 .SI MAETED-1
E070- 00 0930 .BY $00
0940 :
0950 :RUIMTE VOOR EVENTUELE LATERE UITBREIDINGEN
0960 :
E071- 0970 .DS 40
0980 :
0990 :DIRECT GEEFT DE DIRECTORY VAN EEN DISKETTE
1000 :
E099- 20 54 27 1010 DIRECT JSR LOADHEAD :ZET DE KOP OP DE DISK
E09C- A9 12 1020 LDA #12
E09E- 20 BC 26 1030 JSR SETTK :STUUR DE KOP NAAR TRACK 12
EOA1- 20 F4 E0 1040 JSR READDIR :LEES DE DIRECTORY IN
EOA4- 20 73 2D 1050 JSR STROUT :PRINT DE VOLGENDE STRING
EOA7- 20 20 20 1060 .BY $20 $20 $20
EOAA- 2D 2D 20 1070 .BY '--' $20 'Directory' $20
EOAD- 44 69 72
EOB0- 65 63 74
EOB3- 6F 72 79
EOB6- 20
EOB7- 2D 2D 0D 1080 .BY '--' $0D $0A
EOBA- 0A
EOBB- 46 69 6C 1090 .BY 'File name' $20 $20 $20
EOBE- 65 20 6E
EOC1- 61 6D 65
EOC4- 20 20 20
EOC7- 54 72 61 1100 .BY 'Track range' $0D $0A
EOCA- 63 6B 20
EOCD- 72 61 6E
EOD0- 67 65 0D
EOD3- 0A

```

```

E0D4- 2D 2D 2D 1110 .BY '-----'
E0D7- 2D 2D 2D
E0DA- 2D 2D 2D
E0DD- 2D 2D 2D
E0E0- 2D 2D 2D
E0E3- 2D 2D 2D
E0E6- 2D 2D 2D
E0E9- 2D 2D
E0EB- 0D 0A 00 1120 .BY $0D $0A $00
E0EE- 20 11 E1 1130 JSR PRDIRECT :PRINT DE DIRECTORY
E0F1- 4C 61 27 1140 JMP UNLOAD :KOP OMHOOG + EXIT
1150
E0F4- A2 00 1160 READDIR LDX #$00 :ZET HET LABE DEEL VAN
E0F6- 86 FE 1170 STX *VECL :DE LOADPOINTER
E0F8- E8 1180 INX
E0F9- 8E 5E 26 1190 STX SECTNUMB :LEES VAN SECTOR 1
E0FC- AE 00 23 1200 LDX RAMLOC :HOOGSTE PAGINA IN RAM
E0FF- CA 1210 DEX :-1
E100- 86 FF 1220 STX *VECH :IN LOAD VECTOR
E102- 20 67 29 1230 JSR READSECTOR :LEES SECTOR IN
E105- EE 5E 26 1240 INC SECTNUMB :SECTOR 2
E108- AE 00 23 1250 LDX RAMLOC :LOAD POINTER +1 PAGINA
E10B- 86 FF 1260 STX *VECH
E10D- 20 67 29 1270 JSR READSECTOR :LEES SECTOR 2 IN
E110- 60 1280 RTS
1290
E111- A0 00 1300 PRDIRECT LDY #$00 :ZET DE POINTER OP HET
E113- 84 FE 1310 STY *VECL :BEGIN VAN DE DIRECTORY
E115- AE 00 23 1320 LDX RAMLOC
E118- CA 1330 DEX
E119- 86 FF 1340 STX *VECH
E11B- B1 FE 1350 NAME LDA (VECL),Y :LEES BYTE
E11D- C9 23 1360 CMP #$23 :IS HET EEN LEGE PLAATS ?
E11F- F0 10 1370 BEQ NEXTNAME :ZO JA, VOLGENDE FILENAAM
E121- 20 3A E1 1380 JSR PRLINE :PRINT DE FILENAAM UIT
E124- C8 1390 INY
E125- D0 F4 1400 BNE NAME :SPRING TERUG
E127- E6 FF 1410 INCVVEC INC *VECH :VOLGENDE PAGINA
E129- AD 00 23 1420 LDA RAMLOC
E12C- C5 FF 1430 CMP *VECH :LAATSTE PAGINA GEDAAN ?
E12E- 10 EB 1440 BPL NAME :ZO NIET, TERUG
E130- 60 1450 RTS
1460
E131- 98 1470 NEXTNAME TYA :Y-INDEX + 8
E132- 18 1480 CLC
E133- 69 08 1490 ADC #$08
E135- A8 1500 TAY
E136- D0 E3 1510 BNE NAME :SPRING TOT HELE PAGINA GEDAAN IS
E138- F0 ED 1520 BEQ INCVVEC :ANDERS VECTOR AANPASSEN
1530
E13A- 20 43 23 1540 PRLINE JSR PRCHA :PRINT EERSTE KARAKTER
E13D- C8 1550 INDEX INY :VOLGENDE KARAKTER
E13E- 98 1560 TYA
E13F- 29 07 1570 AND #$07 :MASKEER BITS 3-7
E141- C9 06 1580 CMP #$06 :6 KARAKTERS GEDAAN ?
E143- F0 08 1590 BEQ PRLINE2 :ZO JA TRACK RANGE PRINTEN
E145- B1 FE 1600 LDA (VECL),Y :ZO NEE KARAKTER PRINTEN
E147- 20 43 23 1610 JSR PRCHA
E14A- 4C 3D E1 1620 JMP INDEX
1630
E14D- 84 E0 1640 PRLINE2 STY *TEMP1 :REDT Y INDEX
E14F- 20 73 2D 1650 JSR STROUT :PRINT 7 SPATIES
E152- 20 20 2D 1660 .BY $20 $20 $20 $20 $20 $20 $20 $00
E155- 20 20 2D
E158- 20 00
E15A- A4 E0 1670 LDY *TEMP1 :HERSTEL Y INDEX
E15C- B1 FE 1680 LDA (VECL),Y :START TRACK
E15E- 20 92 2D 1690 JSR PRBYTE :PRINT TRACKNUMMER
E161- C8 1700 INY
E162- 84 E0 1710 STY *TEMP1
E164- 20 73 2D 1720 JSR STROUT
E167- 20 2D 2D 1730 .BY $20 '-' $20 $00
E16A- 00

```

```

E16B- A4 E0      1740      LDY *TEMP1
E16D- B1 FE      1750      LDA (VECL),Y          :LAATSTE TRACK
E16F- 20 92 2D   1760      JSR PRBYTE
E172- 20 6A 2D   1770      JSR CRLF             ;RETURN/LINE FEED
E175- 60         1780      RTS
1790      ;
1800      ;
1810      ;BIJ 06-65D STAAT DE DIRECTORY OP TRACK 12.
1820      ;OP ZICHZELF IS DIT AL ONGELUKKIG OMDAT HET
1830      ;BETEKENT DAT FILES DIE TRACK 12 KRUISEN
1840      ;VERBODEN ZIJN. HELAAS IS DAAR NIETS AAN TE
1850      ;DOEN, ALTHANS NIET ALS WE DE ZAAK NOG REDELIJK
1860      ;COMPATIBEL WILLEN HOUDEN. EEN TWEDE MERKWAAR-
1870      ;DIGHEID IS DAT OP TRACK 12, SECTOR 3 EN 4 TWE
1880      ;MACHINETAALROUTINES STAAN DIE DOOR DHIO ALS
1890      ;DE PUT/GET OVERLAYS WORDEN AANGEDUID. BIJ HET
1900      ;WERKEN MET RANDOM FILES MOETEN DEZE ROUTINES
1910      ;AANWEZIG ZIJN, ANDERS IS HET NIET MOGELIJK
1920      ;OM RECORDS TE LEZEN EN TE SCHRIJVEN.
1930      ;DAAROM WORDEN DEZE SECTOREN EERST VAN EEN
1940      ;SYSTEEMDISK GELEZEN OM DAARNA OP DE GEFOR-
1950      ;MATTEERDE SCHIJF TE WORDEN WEGGESCHREVEN.
1960      ;
1970      ;
1980      FORMAT      JSR STROUT      ;PRINT STRING
1990      .BY $0D $0A 'PUT A SYSTEMDISK IN DRIVE A'

E176- 20 73 2D   1980
E179- 0D 0A 50   1990
E17C- 55 54 20
E17F- 41 20 53
E182- 59 53 54
E185- 45 4D 44
E188- 49 53 4B
E18B- 20 49 4E
E18E- 20 44 52
E191- 49 56 45
E194- 20 41
E196- 20 41 4E   2000      .BY ' AND PRESS (RETURN)' $00
E199- 44 20 50
E19C- 52 45 53
E19F- 53 20 3C
E1A2- 52 45 54
E1A5- 55 52 4E
E1A8- 3E 00
E1AA- 20 40 23   2010
E1AD- C9 0D      2020      JSR RECCHA
E1AF- D0 76      2030      CMP #$0D             ;IS HET EEN RETURN ?
E1B1- A9 01      2040      BNE BRANCH          ;ZO NEE, EXIT
E1B3- 20 C6 29   2050      LDA #$01
E1B6- 20 63 26   2060      JSR SELDRIVE        ;SELECTEER DRIVE A
E1B9- 20 54 27   2070      JSR HOME            ;KOP NAAR TRACK 0
E1BC- A9 11      2080      JSR LOADHEAD        ;ZET DE KOP OP DE SCHIJF
E1BE- 20 BC 26   2090      LDA #$11
E1C1- A9 79      2100      JSR SETTK           ;STAP NAAR TRACK 11
E1C3- 85 FE      2110      LDA #$79            ;LOADVECTOR = $2E79
E1C5- A9 2E      2120      STA *VECL
E1C7- 85 FF      2130      LDA #$2E
E1C9- A9 03      2140      STA *VECH
E1CB- 8D 5E 26   2150      LDA #$03            ;OP SECTOR 3 STAAT EEN
E1CE- 20 67 29   2160      STA SECTNUMB        ;LEGE DIRECTORY
E1D1- A9 12      2170      JSR READSECTOR      ;LEES SECTOR IN
E1D3- 20 BC 26   2180      LDA #$12
E1D6- 84 FE      2190      JSR SETTK           ;STAP NAAR TRACK 12
E1D8- AE 00 23   2200      STY *VECL           ;Y=$00
E1DB- CA         2210      LDX RAMLOC          ;TOP VAN HET GEHEUGEN -1
E1DC- 86 FF      2220      DEX
E1DE- 20 67 29   2230      STX *VECH
E1E1- EE 5E 26   2240      JSR READSECTOR      ;LEES SECTOR 3
E1E4- 20 67 29   2250      INC SECTNUMB
E1E7- 20 61 27   2260      JSR READSECTOR      ;LEES SECTOR 4
E1EA- 20 73 2D   2270      JSR UNLOAD          ;KOP OMHOOG
E1ED- 0D 0A 50   2280      JSR STROUT
E1F0- 55 54 20
E1F3- 41 20 42
E1F6- 4C 41 4E
        .BY $0D $0A 'PUT A BLANK DISKETTE IN DRIVE A'

```

```

E1F9- 4B 20 44
E1FC- 49 53 4B
E1FF- 45 54 54
E202- 45 20 49
E205- 4E 20 44
E208- 52 49 56
E20B- 45 20 41
E20E- 20 41 4E 2290 .BY ' AND PRESS (RETURN)' $00
E211- 44 20 50
E214- 52 45 53
E217- 53 20 3C
E21A- 52 45 54
E21D- 55 52 4E
E220- 3E 00
E222- 20 40 23 2300 JSR RECCHA
E225- C9 0D 2310 CMP #$0D :IS HET EEN RETURN ?
E227- D0 57 2320 BRANCH BNE FORMATEND :ZO NEE, EXIT
E229- 20 68 27 2330 JSR INITS :INITIALISEER DE DISK
E22C- 20 54 27 2340 JSR LOADHEAD :ZET DE KOP OP DE SCHIJF
E22F- A9 12 2350 LDA #$12
E231- 20 BC 26 2360 JSR SETTK :STAP NAAR TRACK 12
E234- A9 79 2370 LDA #$79 :ZET DE WRITE VECTOR
E236- 85 FE 2380 STA *VECL
E238- A9 2E 2390 LDA #$2E
E23A- 85 FF 2400 STA *VECH
E23C- A9 01 2410 LDA #$01 :SCHRIJVEN NAAR SECTOR 1
E23E- 8D 5E 26 2420 STA SECTNUMB
E241- 8D 5F 26 2430 STA PAGENUMB :LENGTE IS 1 PAGINA
E244- 20 E1 27 2440 JSR DUMPSECTOR :SCHRIJF SECTOR WEG
E247- C6 FF 2450 DEC *VECH :HERSTEL WRITE VECTOR
E249- EE 5E 26 2460 INC SECTNUMB :SECTOR 2
E24C- 20 E1 27 2470 JSR DUMPSECTOR :SCHRIJF SECTOR WEG
E24F- A9 00 2480 LDA #$00 :ZET NIEUWE WRITE POINTER
E251- 85 FE 2490 STA *VECL
E253- AE 00 23 2500 LDX RAMLDC
E256- CA 2510 DEX
E257- 86 FF 2520 STX *VECH
E259- EE 5E 26 2530 INC SECTNUMB :SECTOR 3
E25C- 20 E1 27 2540 JSR DUMPSECTOR :SCHRIJF SECTOR WEG
E25F- EE 5E 26 2550 INC SECTNUMB :SECTOR 4
E262- 20 E1 27 2560 JSR DUMPSECTOR :SCHRIJF SECTOR WEG
E265- A2 07 2570 LDX #$07 :X IS DE LUSTELLER
E267- BD 83 E2 2580 DIRLOOP LDA TABLE, X :ZET FIJLENAAM 'DIRECT'
E26A- 9D 79 2E 2590 STA DIRAD, X ;IN DE DIRECTORY
E26D- CA 2600 DEX
E26E- 10 F7 2610 BPL DIRLOOP
E270- A9 79 2620 LDA #$79 :ZET DE WRITE POINTER
E272- 85 FE 2630 STA *VECL
E274- A9 2E 2640 LDA #$2E
E276- 85 FF 2650 STA *VECH
E278- A9 01 2660 LDA #$01
E27A- 8D 5E 26 2670 STA SECTNUMB :SECTOR 1
E27D- 20 E1 27 2680 JSR DUMPSECTOR :SCHRIJF SECTOR WEG
E280- 4C 61 27 2690 FORMATEND JMP UNLOAD ;KOP OMHOOG + EXIT
: 2700
: 2710 TABLE .BY 'DIRECT' $12 $12
E283- 44 49 52 2720 ;
E286- 45 43 54 2730 ;
E289- 12 12 2740 ;DOS COMMANDO'S DIE VIA BASIC WORDEN GEGEVEN VIA
2750 ;EEN DISK! COMMANDO, WORDEN IN DE EDIT BUFFER
2760 ;GEZET, ECHTER ZONDER AFSLUITENDE RETURN. DEZE
2770 ;ROUTINE DOET DAT ALSNOG, HET PRINCIPE ZIT HEM
2780 ;HIERIN DAT PAGINA 0 BIJ EEN DOS COMMANDO VANUIT
2790 ;BASIC DOOR DE PAGE 0/1 SWAPPER VANAF ADRES $2F79
2800 ;GERED WORDT, OP ADRES $2F95 (HET VIJFDE KARAKTER
2810 ;VAN DE BASIC INPUT BUFFER) KIJKEN WE DAN OF ER
2820 ;EEN ! STAAT (ASCII $21). IS DAT INDERDAAD ZO DAN

```

```

2830 ;MOGEN WE ERVAN UITGAAN DAT BASIC DE TAAL IS
2840 ;WAARIN WE OP DAT MOMENT WERKEN, EN VOEGEN WE DE
2850 ;BETREFFENDE RETURN TOE.
2860 ;
2870 ;
E28B- AD 95 2F 2880 BASTEST LDA $2F95
E28E- C9 21 2890 CMP #$21 ;IS HET EEN ! ?
E290- D0 0E 2900 BNE BASEND
E292- AD 00 23 2910 LDA RAMLOC ;HOOGSTE PAGINA IN RAM
E295- A0 FF 2920 LDY #$FF ;HOOGSTE BYTE IN DEZE PAGINA
E297- 85 FF 2930 STA *VECH
E299- 84 FE 2940 STY *VECL
E29B- C8 2950 INY ;Y=$00
E29C- A9 0D 2960 LDA #$0D ;RETURN
E29E- 91 FE 2970 STA (VECL),Y ;EN ZET DE RETURN
E2A0- 60 2980 BASEND RTS
2990 ;
3000 ;
3010 ; .LI KLEIN3.MOS

FO3 1F93 4292-6225

3010 ;'FILE ALREADY EXISTS' ERROR UITGANG VAN CREATE
3020 ;
E2A1- A0 00 3030 FOUND LDY #$00 ;INDEX IN TABEL
E2A3- 4C 31 E4 3040 JMP PRERROR ;FOUTMELDING
3050 ;
3060 ;
3070 ;CREATE SCHRIJFT EEN FILENAAM IN DE DIRECTORY
3080 ;
3090 ;
E2A6- 20 8B E2 3100 CREATE JSR BASTEST ;BASIC IN GEHEUGEN ?
E2A9- 20 4A E3 3110 JSR TR12 ;STAP NAAR TRACK 12
E2AC- A9 01 3120 LDA #$01
E2AE- 8D 5E 26 3130 STA SECTNUMB ;SECTOR 1
E2B1- 20 53 E3 3140 JSR READ ;LEES SECTOR IN DE DIRECTORY
E2B4- 20 6B E3 3150 JSR COMPAR ;FILENAAM AL IN DE DIRECTORY ?
E2B7- B0 EB 3160 BCS FOUND ;ZO JA, FOUTMELDING
E2B9- EE 5E 26 3170 INC SECTNUMB ;SECTOR 2
E2BC- 20 53 E3 3180 JSR READ ;LEES SECTOR IN DE DIRECTORY
E2BF- 20 6B E3 3190 JSR COMPAR ;FILENAAM AL IN DE DIRECTORY ?
E2C2- B0 DD 3200 BCS FOUND ;ZO JA, FOUTMELDING
E2C4- CE 5E 26 3210 DEC SECTNUMB ;SECTOR 1
E2C7- 20 53 E3 3220 JSR READ ;LEES SECTOR
E2CA- 20 EB E3 3230 JSR GETPLACE ;ZOEK LEGE PLAATS
E2CD- 86 E6 3240 STX *TEMP4 ;RED INDEX
E2CF- F0 0B 3250 BEQ CRA ;SPRING ALS ER EEN LEGE PLAATS WAS
E2D1- EE 5E 26 3260 INC SECTNUMB ;SECTOR 2
E2D4- 20 53 E3 3270 JSR READ ;LEES SECTOR
E2D7- 20 EB E3 3280 JSR GETPLACE ;ZOEK LEGE PLAATS
E2DA- 86 E6 3290 STX *TEMP4 ;RED INDEX
E2DC- 20 0E E4 3300 CRA JSR GETBYTE ;LEES START TRACK UIT BUFFER
E2DF- 85 E4 3310 STA *TEMP2 ;BEWAAR HEM
E2E1- 20 5B 2D 3320 JSR CHECK ;IS DE KOMMA AANWEZIG ?
E2E4- 20 0E E4 3330 JSR GETBYTE ;LEES LAATSTE TRACK UIT BUFFER
E2E7- 85 E5 3340 STA *TEMP3
E2E9- A5 E4 3350 LDA *TEMP2 ;START TRACK
E2EB- 20 FF E3 3360 JSR TCHECK ;CONTROLEER TRACKNUMMER
E2EE- B0 55 3370 BCS BADTRACK ;SPRING OP EEN FOUT
E2F0- A5 E5 3380 LDA *TEMP3 ;LAATSTE TRACK >12 ?
E2F2- C9 12 3390 CMP #$12
E2F4- 90 08 3400 BCC CRB ;ZO NEE, SPRING
E2F6- A5 E4 3410 LDA *TEMP2 ;ZO JA CONTROLEER OF
E2F8- C9 12 3420 CMP #$12 ;START TRACK <12 IS
E2FA- 90 49 3430 BCC BADTRACK ;FOUT ALS DAT ZO IS
E2FC- A5 E5 3440 LDA *TEMP3
E2FE- 20 FF E3 3450 CRB JSR TCHECK ;CONTROLEER TRACKNUMMER
E301- B0 42 3460 BCS BADTRACK ;SPRING OP EEN FOUT
E303- A5 E4 3470 LDA *TEMP2 ;CONTROLEER OF LAATSTE
E305- C5 E5 3480 CMP *TEMP3 ;TRACK >= EERSTE IS
E307- F0 02 3490 BEQ CRG
E309- B0 3A 3500 BCS BADTRACK ;ZO NIET, FOUTMELDING
E30B- A6 E6 3510 CRG LDX *TEMP4 ;HERSTEL X INDEX

```

```

E30D- A0 03      3520      LDY ##03      ;HERSTEL BUFFER POINTER
E30F- 8C E5 2C   3530      STY $2CE5
E312- 20 E4 2C   3540 CRC   JSR READBUFFER ;HAAL KARAKTER UIT BUFFER
E315- C9 3D      3550      CMP #'='      ;DE '=' SLUIT DE FILENAAM AF
E317- F0 06      3560      BEQ CRD       ;SPRING ALS WE HEM GEVONDEN HEBBEN
E319- 9D 79 2E   3570      STA DIRAD,X   ;ZET HET KARAKTER VAN DE FILE-
E31C- E8         3580      INX          ;NAAM IN DE DIRECTORY
E31D- D0 F3      3590      BNE CRC       ;SPRING
E31F- AC E5 2C   3600 CRD   LDY $2CE5     ;INDEX IN INPUT BUFFER
E322- C0 0A      3610 CRE   CPY ##0A     ;6 KARAKTERS GEHAD ?
E324- F0 09      3620      BEQ CRF       ;ZO JA SPRING
E326- A9 20      3630      LDA ##20     ;ZET EEN SPATIE
E328- 9D 79 2E   3640      STA DIRAD,X   ;IN DE DIRECTORY
E32B- E8         3650      INX          ;VERHOOG INDEX EN LUSTELLER
E32C- C8         3660      INY
E32D- D0 F3      3670      BNE CRE       ;EN SPRING
E32F- A5 E4      3680 CRF   LDA *TEMP2    ;ZET START TRACK
E331- 9D 79 2E   3690      STA DIRAD,X   ;IN DE DIRECTORY
E334- E8         3700      INX
E335- A5 E5      3710      LDA *TEMP3    ;ZET LAATSTE TRACK
E337- 9D 79 2E   3720      STA DIRAD,X   ;IN DE DIRECTORY
E33A- A9 01      3730      LDA ##01     ;AANTAL TE SCHRIJVEN PAGINA'S
E33C- 8D 5F 26   3740      STA PAGENUMB
E33F- 20 5F E3   3750      JSR DIRDUMP   ;SCHRIJF DIRECTORY WEG
E342- 4C 61 27   3760      JMP UNLOAD    ;EN EXIT
                3770      ;
E345- A0 16      3780 BADTRACK LDY #BAD-FEXIST ;GEEF BAD TRACK NUMBER
E347- 4C 31 E4   3790      JMP PRERROR   ;FOUTMELDING
                3800      ;
                3810      ;
E34A- 20 54 27   3820 TR12   JSR LOADHEAD   ;ZET DE KOP OP DE SCHIJF
E34D- A9 12      3830      LDA ##12     ;EN STAP NAAR TRACK 12
E34F- 20 BC 26   3840      JSR SETTK
E352- 60         3850      RTS
                3860      ;
                3870      ;
E353- A9 79      3880 READ   LDA ##79      ;LEES EEN SECTOR VAN TRACK 12
E355- 85 FE      3890      STA *VECL    ;IN DE DIRECTORY BUFFER
E357- A9 2E      3900      LDA ##2E
E359- 85 FF      3910      STA *VECH
E35B- 20 67 29   3920      JSR READSECTOR
E35E- 60         3930      RTS
                3940      ;
                3950      ;
E35F- A9 79      3960 DIRDUMP LDA ##79      ;SCHRIJF EEN DIRECTORY NAAR
E361- 85 FE      3970      STA *VECL    ;TRACK 12
E363- A9 2E      3980      LDA ##2E
E365- 85 FF      3990      STA *VECH
E367- 20 E1 27   4000      JSR DUMPSECTOR
E36A- 60         4010      RTS
                4020      ;
                4030      ;
4040      ;COMPAR IS EEN SUBROUTINE DIE DOOR CREATE, DELETE,
4050      ;EN CHANGE WORDT AANGEROEPEN. DEZE ROUTINE CONTRO-
4060      ;LEERT OF EEN GEGEVEN FILENAAM WEL OF NIET AANWEZIG
4070      ;IS IN DE DIRECTORY. IS DAT HET GEVAL, DAN KEERT DE
4080      ;ROUTINE MET DE CARRY VLAG GEZET TERUG. ZO NIET DAN
4090      ;IS DE CARRY NUL. IS EEN FILENAAM GEVONDEN DAN
4100      ;STAAT DE X INDEX ERGENS IN DEZE FILENAAM GERI-
4110      ;DE Y INDEX STAAT GERI-
4120      ;NA DE '=' BEGRENZER IN DE DOS INPUT BUFFER.
4130      ;EN-PASSANT DOET DEZE ROUTINE OOK NOG EEN CONTROLE
4140      ;OP SYNTAX FOUTEN.
4150      ;
4160      ;
E36B- A2 00      4170 COMPAR LDX ##00      ;INDEX=#00
E36D- A0 02      4180      LDY ##02     ;INPUTBUFFER POINTER = 2
E36F- 8C E5 2C   4190      STY $2CE5
E372- 20 E4 2C   4200      JSR READBUFFER ;LEES KARAKTER
E375- C9 20      4210      CMP ##20     ;IS HET EEN SPATIE ?
E377- D0 28      4220      BNE ERRDUT   ;ZO NEE, SYNTAX ERROR
E379- 20 E4 2C   4230 INNAME JSR READBUFFER ;LEES KARAKTER
E37C- C9 0D      4240      CMP ##0D     ;RETURN ?

```

```

E37E- FO 04      4250      BEQ SPACE      ;ZO JA, SPRING
E380- C9 3D      4260      CMP #'=        ;IS HET EEN = ?
E382- D0 16      4270      BNE LCHECK     ;ZO NEE, SPRING
E384- BD 79 2E   4280 SPACE   LDA DIRAD,X    ;KARAKTER UIT DIRECTORY
E387- C9 20      4290      CMP #20        ;IS HET EEN SPATIE ?
E389- D0 05      4300      BNE OK1        ;ZO NEE, INDEX AANPASSEN
E38B- 20 CE E3   4310      JSR FIND)     ;ZOEK FILENAAMBEGRENZER
E38E- 38         4320      SEC           ;FILENAAM GEVONDEN (C=1)
E38F- 60         4330      RTS
                4340      ;
E390- 20 BD E3   4350 OK1     JSR XADJUST    ;PAS X INDEX AAN
E393- A0 03      4360 OK2     LDY #03        ;HESTEL Y INDEX
E395- BC E5 2C   4370      STY $2CE5     ;NAAR INPUTBUFFER
E398- D0 DF      4380      BNE INNAME
                4390      ;
E39A- AC E5 2C   4400 LCHECK   LDY $2CE5
E39D- C0 0A      4410      CPY #0A        ;MEER DAN 6 KARAKTERS GEHAD ?
E39F- D0 05      4420      BNE COMPNAME   ;ZO NEE, VERDER
E3A1- A9 07      4430 ERR0UT  LDA #07        ;SYNTAX ERROR
E3A3- 4C 4B 2A   4440      JMP ERR?
                4450      ;
E3A6- DD 79 2E   4460 COMPNAME  CMP DIRAD,X    ;VERGELIJK MET FILENAAM
E3A9- FO 05      4470      BEQ NEXT       ;VOLGENDE KARAKTER
E3AB- 20 BD E3   4480      JSR XADJUST    ;ANDERS VOLGENDE FILENAAM
E3AE- D0 E3      4490      BNE OK2        ;SPRING ALTIJD
                4500      ;
E3B0- E8         4510 NEXT     INX            ;X+1
E3B1- 8A         4520      TXA
E3B2- 29 07      4530      AND #07        ;MASKEER BITS 3-7
E3B4- C9 06      4540      CMP #06        ;6 KARAKTERS GEHAD ?
E3B6- D0 C1      4550      BNE INNAME     ;VOLGEND KARAKTER
E3B8- 20 CE E3   4560      JSR FIND)     ;ZOEK FILENAAMBEGRENZER
E3BB- 38         4570      SEC           ;FILENAAM GEVONDEN (C=1)
E3BC- 60         4580      RTS
                4590      ;
E3BD- 8A         4600 XADJUST  TXA
E3BE- 29 F8      4610      AND #F8        ;X DEELBAAR MAKEN DOOR 8
E3C0- 18         4620      CLC           ;EN OP VOLGENDE FILENAAM RICHTEN
E3C1- 69 08      4630      ADC #08
E3C3- AA         4640      TAX
E3C4- FO 01      4650      BEQ XOUT       ;SPRING ALS DE HELE PAGINA KLAAR IS
E3C6- 60         4660      RTS
E3C7- 68         4670 XOUT     PLA            ;HAAL TERUGKEERADRES VAN
E3C8- 68         4680      PLA            ;DE STACK
E3C9- 20 CE E3   4690      JSR FIND)     ;ZOEK FILENAAMBEGRENZER
E3CC- 18         4700 XOUT2   CLC           ;FILENAAM NIET GEVONDEN (C=0)
E3CD- 60         4710      RTS
                4720      ;
E3CE- A0 03      4730 FIND)   LDY #03        ;ZOEK DE FILENAAM BEGRENZER
E3D0- 8C E5 2C   4740      STY $2CE5
E3D3- 20 E4 2C   4750 YLOOP   JSR READBUFFER ;LEES KARAKTER
E3D6- C9 0D      4760      CMP #0D
E3D8- FO 04      4770      BEQ YOUT
E3DA- C9 3D      4780      CMP #'=
E3DC- D0 F5      4790      BNE YLOOP     ;ZOEK EEN = OF EEN $0D
E3DE- AC E5 2C   4800 YOUT   LDY $2CE5
E3E1- C0 0B      4810      CPY #0B        ;MEER DAN 6 KARAKTERS ?
E3E3- B0 01      4820      BCS YERR      ;ZO JA, FOUTMELDING
E3E5- 60         4830      RTS
E3E6- A9 07      4840 YERR    LDA #07        ;SYNTAX ERROR
E3E8- 4C 4B 2A   4850      JMP ERR?
                4860      ;
                4870      ;
                4880 ;GETPLACE ZOEKT EEN VRIJE POSITIE IN DE DIRECTORY
                4890 ;
                4900 ;
E3EB- A2 00      4910 GETPLACE LDX #00        ;INDEX = 0
E3ED- BD 79 2E   4920 BETA    LDA DIRAD,X    ;HAAL EERSTE KARAKTER OP
E3F0- C9 23      4930      CMP #23        ;IS HET EEN '#' ?
E3F2- D0 01      4940      BNE GETB      ;ZO NEE SPRING
E3F4- 60         4950      RTS          ;LEGE PLEK GEVONDEN (Z=1)
E3F5- 8A         4960 GETB    TXA            ;X INDEX + 8
E3F6- 18         4970      CLC

```

```

E3F7- 69 08      4980      ADC  #08
E3F9- AA         4990      TAX
E3FA- D0 F1      5000      BNE  GETA      ;SPRING ALS X(<)$00
E3FC- A9 FF      5010      LDA  #FF      ;GEEN LEGE PLAATS (Z=0)
E3FE- 60         5020      RTS
                5030      ;
                5040      ;
                5050      ;DEZE ROUTINE CONTROLEERT OF EEN TRACKNUMMER (<) 0,
                5060      ;(<) 12 EN (<) 40 IS.
                5070      ;
                5080      ;
E3FF- C9 00      5090      TCHECK  CMP  #00      ;(<) 0 ?
E401- F0 09      5100      BEQ  TCHOUT
E403- C9 12      5110      CMP  #12      ;(<) 12 ?
E405- F0 05      5120      BEQ  TCHOUT
E407- C9 40      5130      CMP  #40      ;(<) 40 ?
E409- B0 01      5140      BCS  TCHOUT
E40B- 60         5150      RTS      ;ALLE DRIE JA : C=0
E40C- 38         5160      TCHOUT  SEC      ;C=1
E40D- 60         5170      RTS
                5180      ;
                5190      ;GETBYTE LEEST TWEE ASCII KARAKTERS UIT DE INPUTBUFFER
                5200      ;EN CONVERTEERT HET TOT EEN BCD NUMMER
                5210      ;
E40E- 20 1D E4   5220      GETBYTE  JSR  ASCHEX      ;ASCII -> HEX
E411- 0A         5230      ASL  A      ;HOGE NIBBLE
E412- 0A         5240      ASL  A
E413- 0A         5250      ASL  A
E414- 0A         5260      ASL  A
E415- 85 E0      5270      STA  *TEMP1
E417- 20 1D E4   5280      JSR  ASCHEX      ;ASCII -> HEX
E41A- 05 E0      5290      ORA  *TEMP1
E41C- 60         5300      RTS
                5310      ;
E41D- 20 E4 2C   5320      ASCHEX  JSR  READBUFFER      ;LEES KARAKTER
E420- C9 30      5330      CMP  #30      ; (<) 30 ?
E422- 90 08      5340      BCC  ASCERR?
E424- C9 3A      5350      CMP  #3A      ; (>) 3A ?
E426- B0 04      5360      BCS  ASCERR?
E428- 38         5370      SEC
E429- E9 30      5380      SBC  #30
E42B- 60         5390      RTS
                5400      ;
E42C- A9 07      5410      ASCERR?  LDA  #07      ;SYNTAX ERROR
E42E- 4C 4B 2A   5420      JMP  ERR?
                5430      ;
                5440      ;
                5450      ;PRINT EEN FOUTMELDING GESPECIFICEERD DOOR DE Y INDEX
                5460      ;
                5470      ;
E431- B9 40 E4   5480      PRERROR  LDA  FEXIST.Y
E434- F0 07      5490      BEQ  ENDPRINT
E436- 20 43 23   5500      JSR  PRCHA
E439- C8         5510      INY
E43A- 4C 31 E4   5520      JMP  PRERROR
E43D- 4C 61 27   5530      ENDPRINT  JMP  UNLOAD      ;KOP OMHOOG + EXIT
                5540      ;
                5550      FEXIST  .BY  $0D $0A 'FILE ALREADY EXISTS' $00
E440- 0D 0A 46
E443- 49 4C 45
E446- 20 41 4C
E449- 52 45 41
E44C- 44 59 20
E44F- 45 58 49
E452- 53 54 53
E455- 00
E456- 0D 0A 42   5560      BAD      .BY  $0D $0A 'BAD TRACK NUMBER' $00
E459- 41 44 20
E45C- 54 52 41
E45F- 43 4B 2D
E462- 4E 55 4D
E465- 42 45 52
E468- 00
                5570      ;

```

```

5580 :
5590 :DELETE VERWIJDEERT EEN FILENAAM UIT DE DIRECTORY
5600 :
5610 :
E469- 20 91 E4 5620 DELETE JSR FINDFILE ;ZOEK FILENAAM
E46C- 8A 5630 TXA ;MAAK X DEELBAAR DOR 8
E46D- 29 F8 5640 AND #$F8 ;X WIJST DAN OP HET EERSTE KARAKTER
E46F- AA 5650 TAX ;VAN DE FILENAAM
E470- A0 00 5660 LDY #$00 ;Y IS DE LUSTELLER
E472- A9 23 5670 LDA #$23 ;'#' KARAKTER
E474- 9D 79 2E 5680 DELLOOP STA DIRAD,X ;ZET HET IN DE DIRECTORY
E477- E8 5690 INX
E478- CB 5700 INY
E479- C0 06 5710 CPY #$06 ;MAX. 6 KARAKTERS
E47B- D0 F7 5720 BNE DELLOOP
E47D- A9 00 5730 LDA #$00 ;START TRACK = 0
E47F- 9D 79 2E 5740 STA DIRAD,X
E482- E8 5750 INX
E483- 9D 79 2E 5760 STA DIRAD,X ;LAATSTE TRACK = 0
E486- A9 01 5770 LDA #$01 ;AANTAL TE SCHRIJVEN PAGINA'S
E488- 8D 5F 26 5780 STA PAGENUMB
E48B- 20 5F E3 5790 JSR DIRDUMP ;SCHRIJF DIRECTORY WEG
E48E- 4C 61 27 5800 JMP UNLOAD ;KOP OMHOOG + EXIT
5810 :
5820 :
E491- 20 73 2D 5830 FINDFILE JSR STROUT ;VOOR ALLE ZEKERHEID !
E494- 0D 0A 41 5840 .BY $0D $0A 'ARE YOU SURE?' $00
E497- 52 45 20
E49A- 59 4F 55
E49D- 20 53 55
E4A0- 52 45 3F
E4A3- 20 00
E4A5- 20 40 23 5850 JSR RECCHA
E4A8- C9 59 5860 CMP #'Y ;IS HET Y ?
E4AA- D0 27 5870 BNE NO/OUT ;ZO NEE, EXIT
E4AC- 20 6A 2D 5880 JSR CRLF
E4AF- 20 8B E2 5890 JSR BASTEST ;BASIC TEST
E4B2- 20 4A E3 5900 JSR TR12 ;ZET DE KOP OP TRACK 12
E4B5- A9 01 5910 LDA #$01 ;SECTOR 1
E4B7- 8D 5E 26 5920 STA SECTNUMB
E4BA- 20 53 E3 5930 JSR READ ;LEES SECTOR
E4BD- 20 6B E3 5940 JSR COMPAR ;FILENAAM AANWEZIG ?
E4C0- B0 10 5950 BCS FINOUT ;ZO JA, SPRING
E4C2- EE 5E 26 5960 INC SECTNUMB ;SECTOR 2
E4C5- 20 53 E3 5970 JSR READ ;LEES SECTOR
E4C8- 20 6B E3 5980 JSR COMPAR
E4CB- B0 05 5990 BCS FINOUT
E4CD- A9 0C 6000 LDA #$0C ;'FILE DOESN'T EXIST' ERROR
E4CF- 4C 4B 2A 6010 JMP ERR?
6020 :
E4D2- 60 6030 FINOUT RTS
6040 :
E4D3- 68 6050 NO/OUT PLA ;HAAL RETURN ADRES VAN DE STACK
E4D4- 68 6060 PLA
E4D5- 60 6070 RTS ;EN SPRING UIT DOS COMMANDO
6080 :
6090 :
6100 :CHANGE VERANDERT EEN FILENAAM IN EEN ANDERE
6110 :
6120 :
E4D6- 20 91 E4 6130 CHANGE JSR FINDFILE ;ZOEK FILENAAM
E4D9- 8A 6140 TXA ;MAAK X DEELBAAR DOOR 8
E4DA- 29 F8 6150 AND #$F8
E4DC- AA 6160 TAX
E4DD- 20 E4 2C 6170 CHA JSR READBUFFER ;KARAKTER NIEUWE FILENAAM
E4E0- C9 0D 6180 CMP #$0D ;EINDE NAAM ?
E4E2- F0 0D 6190 BEQ CHB ;ZO JA SPRING
E4E4- 9D 79 2E 6200 STA DIRAD,X ;ZET KARAKTER IN DE DIRECTORY
E4E7- E8 6210 INX
E4E8- 8A 6220 TXA
E4E9- 29 07 6230 AND #$07
E4EB- C9 07 6240 CMP #$07 ;MEER DAN 6 GEDAAN ?
E4ED- B0 1E 6250 BCS CHERR? ;ZO JA, FOUTMELDING

```

```

E4EF- 90 EC      6260      BCC CHA      ;SPRING ALTIJD
                  6270      ;
E4F1- 8A        6280      CHB      TXA      ;ZOEK HET AANTAL KARAKTERS
E4F2- 29 07     6290      AND #07     ;VAN DE NIEUWE FILENAAM
E4F4- AB        6300      TAY      ;EN ZET DAT IN DE Y INDEX
E4F5- A9 20     6310      LDA #20    ;EEN SPATIE
E4F7- C0 06     6320      CHC      CPY #06    ;ALLE 6 KARAKTERS GEHAD ?
E4F9- F0 07     6330      BEQ CHD    ;ZO JA EINDE
E4FB- 9D 79 2E 6340      STA DIRAD,X ;SPATIE IN DE FILENAAM ZETTEN
E4FE- E8        6350      INX
E4FF- C8        6360      INY
E500- D0 F5     6370      BNE CHC    ;SPRING ALTIJD
                  6380      ;
E502- A9 01     6390      CHD      LDA #01    ;AANTAL TE SCHRIJVEN PAGINA'S
E504- 8D 5F 26 6400      STA PAGENUMB
E507- 20 5F E3 6410      JSR DIRDUMP ;SCHRIJF SECTOR WEG
E50A- 4C 61 27 6420      JMP UNLOAD  ;KOP OMHOOG + EXIT
                  6430      ;
E50D- A9 07     6440      CHERR?   LDA #07    ;SYNTAX ERROR
E50F- 4C 4B 2A 6450      JMP ERR?
                  6460      ;
                  6470      ;
                  6480      .LI KLEIN4.MOS
    
```

F04 12D4 4292-5566

```

6480 ;IN DE ORIGINELE VERSIE VAN OS-65D WORDEN FOUTMELDINGEN
6490 ;VIA NOGAL CRYPTISCHE MEDEDELINGEN ALS 'ERR #0' ENZ.
6500 ;GEGEVEN. NIET AL TE HANDIG DUS. DEZE ROUTINES HEBBEN
6510 ;TOT DOEL DE FOUTMELDINGEN VAN HET DOS WAT GEBRUIKS-
6520 ;VRIENDELIJKER TE MAKEN. IN PLAATS VAN EEN MELDING
6530 ;ALS 'ERR #C' KRIJGT U NU NETJES ' FILE DOESN'T EXIST'
6540 ;ENZ. OP HET SCHERM TE ZIEN.
6550 ;
6560 ;
E512- AA        6570      DISKERROR TAX ;A BEVAT HET FOUTNUMMER
E513- CA        6580      DEX ;GEBRUIK DIT OM DE Y INDEX IN
E514- BC 26 E5 6590      LDY TABLE1,X ;DE ASCII TABEL TE VINDEN
E517- B9 33 E5 6600      PRINTE   LDA TABLE2,Y ;PRINT DE FOUTMELDING
E51A- F0 07     6610      BEQ EPRINT
E51C- 20 43 23 6620      JSR PRCHA
E51F- C8        6630      INY
E520- 4C 17 E5 6640      JMP PRINTE
E523- 4C 61 27 6650      EPRINT   JMP UNLOAD  ;KOP OMHOOG + EXIT
                  6660      ;
                  6670      ;
E526- 00        6680      TABLE1 .BY ERR1-ERR1
E527- 0D        6690      .BY ERR2-ERR1
E528- 1A        6700      .BY ERR3-ERR1
E529- 2E        6710      .BY ERR4-ERR1
E52A- 46        6720      .BY ERR5-ERR1
E52B- 51        6730      .BY ERR6-ERR1
E52C- 61        6740      .BY ERR7-ERR1
E52D- 6E        6750      .BY ERR8-ERR1
E52E- 7F        6760      .BY ERR9-ERR1
E52F- 97        6770      .BY ERRA-ERR1
E530- B0        6780      .BY ERRB-ERR1
E531- C2        6790      .BY ERRC-ERR1
E532- D5        6800      .BY ERRD-ERR1
                  6810      ;
                  6820      TABLE2
E533- 50 41 52 6830      ERR1      .BY 'PARITY ERROR' $00
E536- 49 54 59
E539- 20 45 52
E53C- 52 4F 52
E53F- 00
E540- 52 45 52 6840      ERR2      .BY 'REREAD ERROR' $00
E543- 45 41 44
E546- 20 45 52
E549- 52 4F 52
E54C- 00
E54D- 43 41 4E 6850      ERR3      .BY 'CAN' $27 'T WRITE TRACK 0' $00
E550- 27 54 20
    
```

```

E553- 57 52 49
E556- 54 45 20
E559- 34 52 41
E55C- 43 4B 20
E55F- 30 00
E561- 44 49 53 6860 ERR4 .BY 'DISK IS WRITE PROTECTED' $00
E564- 4B 20 49
E567- 53 20 57
E56A- 52 49 54
E56D- 45 20 50
E570- 52 4F 54
E573- 45 43 54
E576- 45 44 00
E579- 53 45 45 6870 ERR5 .BY 'SEEK ERROR' $00
E57C- 4B 20 45
E57F- 52 52 4F
E582- 52 00
E584- 44 52 49 6880 ERR6 .BY 'DRIVE NOT READY' $00
E587- 56 45 20
E58A- 4E 4F 54
E58D- 20 52 45
E590- 41 44 59
E593- 00
E594- 53 59 4E 6890 ERR7 .BY 'SYNTAX ERROR' $00
E597- 54 41 58
E59A- 20 45 52
E59D- 52 4F 52
E5A0- 00
E5A1- 42 41 44 6900 ERR8 .BY 'BAD TRACK NUMBER' $00
E5A4- 20 54 52
E5A7- 41 43 4B
E5AA- 20 4E 55
E5AD- 4D 42 45
E5B0- 52 00
E5B2- 43 41 4E 6910 ERR9 .BY 'CAN' $27 'T FIND TRACK HEADER' $00
E5B5- 27 54 20
E5B8- 46 49 4E
E5BB- 44 20 54
E5BE- 52 41 43
E5C1- 4B 20 48
E5C4- 45 41 44
E5C7- 45 52 00
E5CA- 43 41 4E 6920 ERRA .BY 'CAN' $27 'T FIND SECTOR HEADER' $00
E5CD- 27 54 20
E5D0- 46 49 4E
E5D3- 44 20 53
E5D6- 45 43 54
E5D9- 4F 52 20
E5DC- 48 45 41
E5DF- 44 45 52
E5E2- 00
E5E3- 42 41 44 6930 ERRB .BY 'BAD SECTOR LENGTH' $00
E5E6- 20 53 45
E5E9- 43 54 4F
E5EC- 52 20 4C
E5EF- 45 4E 47
E5F2- 54 48 00
E5F5- 46 49 4C 6940 ERRC .BY 'FILE DOESN' $27 'T EXIST' $00
E5F8- 45 20 44
E5FB- 4F 45 53
E5FE- 4E 27 54
E601- 20 45 58
E604- 49 53 54
E607- 00
E608- 52 2F 57 6950 ERRD .BY 'R/W ATTEMPT PAST END OF FILE' $00
E60B- 20 41 54
E60E- 54 45 4D
E611- 50 54 20
E614- 50 41 53
E617- 54 20 45
E61A- 4E 44 20
E61D- 4F 46 20
E620- 46 49 4C

```

E623- 45 00

```

6960 :
6970 :
6980 : NUMBER GEEFT HET AANTAL TRACKS DAT EEN SOURCE-
6990 : FILE LANG IS.
7000 :
7010 :
E625- 20 6A 2D 7020 NUMBER JSR CRLF ; RETURN/LINE FEED
E628- AD 7B 3A 7030 LDA $3A7B ; END ADRES L VAN SOURCE FILE
E62B- 38 7040 SEC
E62C- E9 79 7050 SBC #$79 ; - BEGIN ADRES L
E62E- AD 7C 3A 7060 LDA $3A7C ; END ADRES H VAN SOURCE FILE
E631- E9 3A 7070 SBC #$3A ; - BEGIN ADRES H
E633- 4A 7080 LSR A ; DEEL DOOR 8
E634- 4A 7090 LSR A
E635- 4A 7100 LSR A
E636- 1B 7110 CLC
E637- 69 01 7120 ADC #$01 ; TEL ER EEN TRACK BIJ CP
E639- 20 92 2D 7130 JSR PRBYTE ; PRINT AANTAL TRACKS
E63C- 20 73 2D 7140 JSR STROUT
E63F- 20 54 52 7150 .BY $20 'TRACK(S)' $00
E642- 41 43 4B
E645- 28 53 29
E648- 00
E649- 60 7160 RTS
7170 :
7180 :
7190 : DE VIER NU VOLGENDE COMMANDO'S ZIJN NU NOG NIET VAN
7200 : BELANG. ZE ZIJN BEDOELD VOOR HET LADEN VAN FIG-FORTH-
7210 : EN DE MACRO ASSEMBLER VAN MOSER, DIE IK INMIDDELS
7220 : AAN HET OS-65D SYSTEEM HEB AANGEPAST.
7230 :
7240 :
E64A- A9 03 7250 FORTH LDA #$03 ; AANTAL TE LADEN TRACKS - 1
E64C- 8D 7A 31 7260 STA $317A ; IN DOS LAAD ROUTINE
E64F- A9 20 7270 LDA #$20 ; LADEN VAN TRACK 20
E651- 20 EE 2A 7280 JSR $2AEE ; LEES 4 TRACKS IN
E654- A9 04 7290 LDA #$04 ; HERSTEL LAADROUTINE
E656- 8D 7A 31 7300 STA $317A
E659- A9 97 7310 LDA #$97 ; $1697 IS HET RETURN ADRES
E65B- A0 16 7320 LDY #$16 ; IN FORTH NA EEN DOS FOUT
E65D- 20 7D 2A 7330 JSR $2A7D ; ADRES IN FOUTROUTINE ZETTEN
E660- 4C 00 02 7340 JMP $0200 ; KOUDE START FORTH
7350 :
7360 :
E663- A9 97 7370 WFORTH LDA #$97 ; $1697 IS TERUGKEERADRES
E665- A0 16 7380 LDY #$16 ; IN FORTH NA EEN DOS FOUT
E667- 20 7D 2A 7390 JSR $2A7D
E66A- 4C 04 02 7400 JMP $0204 ; WARM START FORTH
7410 :
7420 :
7430 : ENTRY NA MON WORD IN FORTH
7440 :
7450 :
E66D- A9 51 7460 LDA #$51 ; TERUGKEERADRES NA DOS FOUT
E66F- A0 2A 7470 LDY #$2A ; WORDT $2A51
E671- 20 7D 2A 7480 JSR $2A7D
E674- 4C 51 2A 7490 JMP KERNEL ; EN SPRING NAAR KERNEL
7500 :
7510 :
E677- A9 24 7520 MAETED LDA #$24 ; ASSEMBLER LADEN VAN TRACK 24
E679- 20 EE 2A 7530 JSR $2AEE
E67C- 4C 00 02 7540 JMP $0200 ; KOUDE START
7550 :
7560 :
7570 : *** BOOTSTRAP VOOR OS-65D V3.4 ***
7580 :
7590 :
7600 : BREAK EN NMI VECTOREN
7610 :
7620 BRKL .DE $FA7C
7630 BRKH .DE $FA7D
7640 NMIL .DE $FA7A
    
```

```

7650 NMIH      .DE $FA7B
7660 :
7670 :
7680 :DE BOOTSTRAP IN DE 2708 EPROM OP DE BASISKAART VAN
7690 :DE JUNIOR LAADT TRACK 0 VANAF ADRES $2200 IN HET
7700 :GEHEUGEN. DAARNA SPRINGEN WE NAAR ADRES $2200 WAAR
7710 :WE DE REST VAN DE BOOTSTRAP VINDEN.
7720 :
7730 :
7740 :*** BOOTSTRAP BEGINT HIER ***
7750 :
7760      .BA $2200
7770 :
2200- A9 01    7780 BOOTUP   LDA #$01      ;LEZEN VAN SECTOR 1
2202- 8D 5E 26 7790      STA SECTNUMB
2205- 20 BC 26 7800      JSR SETTK    ;STAP NAAR TRACK 1
2208- A9 2A    7810      LDA #$2A     ;DE LOADVECTOR WORDT $2A00
220A- 85 FF    7820      STA *VECH
220C- 20 54 27 7830      JSR LOADHEAD ;KOP OP DE SCHIJF
220F- 86 FE    7840      STX *VECL   ;X=$00
2211- 20 67 29 7850      JSR READSECTOR ;LEES SECTOR IN
2214- EE 5E 26 7860      INC SECTNUMB ;SECTOR 2
2217- A9 06    7870      LDA #$06
2219- 20 BC 26 7880      JSR SETTK    ;STAP NAAR TRACK 6
221C- 20 67 29 7890      JSR READSECTOR ;LEES SECTOR VANAF $3200
221F- EE 5E 26 7900      INC SECTNUMB ;SECTOR 3
2222- A9 00    7910      LDA #$00
2224- 85 FE    7920      STA *VECL   ;DE LOADVECTOR WORDT $0000
2226- 85 FF    7930      STA *VECH
2228- 20 67 29 7940      JSR READSECTOR ;LEES PAGINA 0 IN
222B- A9 01    7950      LDA #$01
222D- 8D 5E 26 7960      STA SECTNUMB ;SECTOR 1
2230- A9 14    7970      LDA #$14
2232- 20 BC 26 7980      JSR SETTK    ;STAP NAAR TRACK 14
2235- A9 00    7990      LDA #$00     ;DE LOADVECTOR WORDT $E000
2237- 85 FE    8000      STA *VECL
2239- A9 E0    8010      LDA #$E0
223B- 85 FF    8020      STA *VECH
223D- 20 67 29 8030      JSR READSECTOR ;LEES SECTOR IN
2240- A9 13    8040      LDA #$13
2242- 20 BC 26 8050      JSR SETTK    ;STAP NAAR TRACK 13
2245- A9 32    8060      LDA #$32     ;DE LOADVECTOR WORDT $3274
2247- 85 FF    8070      STA *VECH
2249- A9 74    8080      LDA #$74
224B- 85 FE    8090      STA *VECL
224D- 20 67 29 8100      JSR READSECTOR ;LEES SECTOR IN
2250- 20 61 27 8110      JSR UNLOAD   ;KOP OMHOOG
2253- A0 31    8120      LDY #$31     ;DEZE INSTRUCTIE PAST
2255- 8C 7B 26 8130      STY $267B   ;DE STEPRATE AAN
2258- A0 BF    8140      LDY #$BF
225A- 20 9E 22 8150 MEMLOOP  JSR RAMTST   ;ZOEK HOOGSTE PAGINA IN RAM
225D- F0 03    8160      BEQ RAMSTORE ;SPRING ALS HIJ GEVONDEN IS
225F- 88      8170      DEY
2260- D0 F8    8180      BNE MEMLOOP
2262- 8C 00 23 8190 RAMSTORE STY RAMLOC
2265- A2 01    8200      LDX #$01
2267- 8E C6 2A 8210      STX DISTR   ;DEVICE#1 GESELECTEERD
8220 :
8230 :DE NU VOLGENDE INSTRUCTIES ZIJN BEDOELD VOOR DE
8240 :INITIALISERING VAN DE 6845 CRT CONTROLLER OP
8250 :DE VDU KAART VAN ELEKTUUR. INDIEN U MET EEN
8260 :ELEKTERMINAL OF EEN SOORTGELIJK RANDAPPARAAT
8270 :WERKT MOET U DE VOLGENDE 14 BYTES VERVANGEN
8280 :DOOR NOP INSTRUCTIES.
8290 :
226A- A9 00    8300      LDA #$00
226C- 8D F7 EF 8310      STA $EFF7
226F- 8D D2 EF 8320      STA $EFD2
2272- 20 35 F4 8330      JSR $F435
2275- 20 30 F3 8340      JSR $F330
8350 :
8360 :
2278- 20 73 2D 8370      JSR STROUT   ;GEEFT DE SYSTEEMIDENTIFICATIE

```

```

227B- 4F 53 2D 8380      .BY '05-65D V3.4' $00
227E- 36 35 44
2281- 20 56 33
2284- 2E 34 00
2287- A9 2E      8390      LDA ##2E      ;ZET DE BREAK VECTOR
2289- 8D 7C FA 8400      STA BRKL
228C- A9 FF      8410      LDA ##FF
228E- 8D 7D FA 8420      STA BRKH
2291- A9 00      8430      LDA #$00      ;ZET DE NMI VECTOR
2293- 8D 7A FA 8440      STA NMIL
2296- A9 FC      8450      LDA #$FC
2298- 8D 7B FA 8460      STA NMIH
229B- 4C 51 2A 8470      JMP KERNEL
                8480      :
                8490      :
229E- 84 FF      8500 RAMTST  STY *VECH      ;ZOEK EEN TE BESCHRIJVEN
22A0- B1 FE      8510      LDA (VECL),Y  ;GEHEUGENPLAATS
22A2- 29 3F      8520      AND ##3F
22A4- 49 3F      8530      EOR ##3F
22A6- 91 FE      8540      STA (VECL),Y
22A8- 85 FD      8550      STA *$FD
22AA- B1 FE      8560      LDA (VECL),Y
22AC- 29 3F      8570      AND ##3F
22AE- C5 FD      8580      CMP *$FD
22B0- 60      8590      RTS
                8600      :
                8610      :
                8620      :
                .EN
    
```

ERVARINGEN MET DE PC-2 COMPUTER VAN PROTON ELECTRONICS

Door : Simon Voortman te Ouddorp

Ik heb een PC-2. Deze heeft 4K RAM-on-board. Daarnaast 6K monitor en een 10K Basic interpreter/compiler, die ook met labels kan werken. Aan de PC-2 kleeft een groot nadeel: er kunnen maar 40 karakters op een regel. Bij diverse andere komputers (bijv. Acorn Atom) kun je aan het einde van een regel (op het scherm), gewoon doortikken op de volgende regel. Zo niet bij de PC-2! Aan het einde van het scherm wordt wel op de volgende regel begonnen, maar de interpreter interpreteert dit gewoon als een regel zonder regelnummer, en geeft dus vrolijk: Error:syntax!

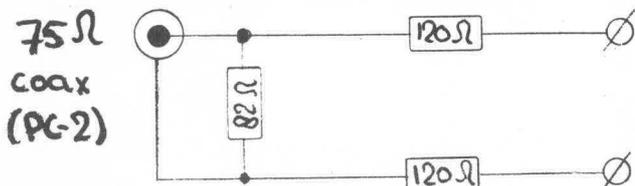
Dit houdt in, dat de teksten korter moeten zijn dan zo'n 21 a 22 karakters. Je kunt wel wat meer op een regel krijgen, als je in plaats van PRINT een vraagteken (?) aanslaat. Bij LISTen staat er gewoon PRINT, maar de tekst schuift door naar de volgende regel. Dit heeft echter het nadeel dat bij veranderen van die regel het stuk tekst op de volgende regel weg is, aangezien alleen de "cursor regel" naar het geheugen gekopieerd is!

Tevens merken de instructies GET en INKEY niet, "wegens de methode van toetsaftasting van de PC-2", hoewel revisie van de GET-instructie er wel aan zou komen.

Dan een instructie die er wel op zit: RENUN. In de documentatie van de PC-2 staat dat de RENUN er niet inzit. Toen ik echter de gereserveerde woorden nakeek, bleek er een RENUN tussen te zitten. En ... ja, hoor! RENUN hernummert netjes de regelnummers, maar dan ook alleen de regelnummers! Sorong-opdrachten (bijv: GOTO 100) worden niet hernummerd! Daar zijn de labels voor.

Als opslagmedium gebruik ik een normale kassetterecorder, en dat gaat prima.

Als monitor gebruik ik een oude (buizen) zw/w TV, met de volgende aanpassing (van 75 ohm naar 300 ohm):



Deze BON is 'n Gulden waard!

... bij inlevering aan de kassa van de Jaarbeurs op 16 en 17 november, tijdens de ...

hoor-dagen 1984

Ons jaarlijks evenement vindt plaats in de Jaarbeurshallen te Utrecht. Geopend van 10.00 tot 17.00 uur. In de HCC-katalogus*, verkrijgbaar in de tijdschriftenkiosk en bij de ingang van de Jaarbeurs, zijn twee gratis toegangsbewijzen opgenomen.

toegang f 5,-

Computer Dagen

*reduktiebon, één per persoon, geldt niet voor aanschaf catalogus HCC, postbus 149, 2250 AC Voorschoten (01717 - 8535)

300Ω symm.
(link kabel)
(TU)

COMMODORE BASIC 4.0 (For CBM 40XX and CBM 80XX).

Tokenized Microsoft Basic Keywords and addresses N. de Vries
 Analogous to the publication for Commodore-64 by A. Mueller.
 DE 6502 KENNER. December 1983. pages 5-8.

COMMANDS

KEYWORDS CORRESPONDING TO B0B2 AND (*)B1AF
 ADDRESSES CORRESPONDING TO B000
 THE ADDRESSES OF ROUTINES FOR COMMANDS ARE THE ADDRESSES
 MINUS 1. BECAUSE THE ROUTINES ARE INVOKED THROUGH RTS.

KEYWORD	TOKEN	ADDR-1			
END	80	B7C7	CONT	9A	B7ED
FOR	81	B6DD	LIST	9B	B62F
NEXT	82	BD18	CLR	9C	B5ED
DATA	83	B882	CMD	9D	BA8D
INPUT#	84	BBA3	SYS	9E	FFDD
INPUT	85	BBBD	OPEN	9F	FFBF
DIM	86	C120	CLOSE	A0	FFC2
READ	87	BC01	GET	A1	BB79
LET	88	B92F	NEW	A2	B5D1
GOTO	89	B82F	GO	* CB	B7AB
RUN	8A	B807	CONCAT	* CC	FF92
IF	8B	B8B2	DOPEN	* CD	FF95
RESTORE	8C	B7B6	DCLOSE	* CE	FF98
GOSUB	8D	B812	RECORD	* CF	FF9B
RETURN	8E	B85C	HEADER	* D0	FF9E
REM	8F	B8C5	COLLECT	* D1	FFA1
STOP	90	B7C5	BACKUP	* D2	FFA4
ON	91	B8D5	COPY	* D3	FFA7
WAIT	92	C962	APPEND	* D4	FFAA
LOAD	93	FFD4	DSAVE	* D5	FFAD
SAVE	94	FFD7	DLOAD	* D6	FFB0
VERIFY	95	FFDA	CATALOG	* D7	FFB3
DEF	96	C4DB	RENAME	* D8	FFB6
POKE	97	C959	SCRATCH	* D9	FFB9
PRINT#	98	BA87	DIRECTORY	* DA	FFB3
PRINT	99	BAA7			

 | MISCELLANEOUS KEYWORDS |
KEYWORDS CORRESPONDING TO B13D

KEYWORD	TOKEN		
TAB(A3	THEN	A7
TO	A4	NOT	A8
FN	A5	STEP	A9
SPC(A6		

COMMODORE BASIC 4.0 (For CBM 40XX and CBM 80XX).

Tokenized Microsoft Basic keywords and addresses

DYADIC OPERATORS

KEYWORDS CORRESPONDING TO B154

PRIORITIES AND ADDRESSES CORRESPONDING TO B094

THE ADDRESSES OF ROUTINES FOR DYADIC OPERATORS ARE THE ADDRESS MINUS 1. BECAUSE THE ROUTINES ARE INVOKED THROUGH A RTS INSTRUCTION.

KEYWORD	TOKEN	ADDR-1	PRTY	
+	AA	C99F	79	addition
-	AB	C988	79	subtraction
*	AC	CB60	7B	multiplication
/	AD	CC47	7B	division
^	AE	D112	7F	exponentiation
AND	AF	C088	50	logical AND
OR	B0	C085	46	logical OR
monadic "--"	AB	D14A	7D	negation
monadic NOT	A8	BECE	5A	logical NOT
)	B1	COB5	64	comparison
=	B2	COB5	64	comparison
<	B3	COB5	64	comparison

FUNCTIONS

FUNCTIONS CORRESPONDING TO B161

ADDRESSES CORRESPONDING TO B066

KEYWORD	TOKEN	ADDR			
SGN	B4	CD6F	TAN	C0	D2D2
INT	B5	CE02	ATN	C1	D32C
ABS	B6	CD8E	PEEK	C2	C943
USR	B7	0000	LEN	C3	C8B2
FRE	B8	C4A8	STR\$	C4	C58E
POS	B9	C4C9	VAL	C5	C8E3
SQR	BA	D108	ASC	C6	C8C1
RND	BB	D229	CHR\$	C7	C822
LOG	BC	CB20	LEFT\$	C8	C836
EXP	BD	D184	RIGHT\$	C9	C862
COS	BE	D282	MID\$	CA	C86D
SIN	BF	D289			

- Note: 1) Bit 7 in the last character of each keyword is set to determine the end of a keyword
 2) Keywords are tokenized by adding x'80' to their relative (hex) position in the table.

```

0010: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
0020: %
0030: %           T A P E - I / O
0040: %
0050: % Datafiles in Basic, Basic programma's editen onder Micro %
0060: % Ade, subroutine libraries voor Basic door Merge, Cross re- %
0070: % ferenca tabels van Basic programma's, auto-execute files %
0080: % en meer van dat leuks voor de KIM en de Junior.
0090: %
0100: %           Door: A.W. den Hartog
0110: %           Tienvoet 2
0120: %           3261 TP Oud Beyerland
0130: %
0140: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
0150:
0160: Bezitters van KIMs of Juniors met een algemene 8x Basic zul-
0170: len bij deze titel waarschijnlijk denken aan met floppydisc
0180: uitgeruste computers die draaien onder Operating Systems als
0190: CP/M, MS-DOS, UCSD p-code, of UNIX waar Basics voor zijn met
0200: commando's als PRINT USING, DRAW, PAINT, OPEN, CLOSE enz. en
0210: met prijzen waar je 20 KIMs voor kunt kopen.
0220: De voorwaarde voor zulke mogelijkheden is echter voornamelijk
0230: software en een 6502 computer is net zo goed programmeerbaar
0240: als een 68000 of 8086 computer. Verder is een floppydisc al-
0250: leen maar een hele snelle cassetteinterface en voor f1250 heb
0260: je al een matrixprinter waar je een grafisch beeldscherm van
0270: 500 bij 800 pixels mee kunt simuleren. Niet langer getreurd
0280: dus en kijken hoe dit alles is te realiseren.
0290:
0300: Wat moet er gebeuren
0310: -----
0320: De sleutel die dit alles mogelijk maakt is het kunnen uit-
0330: wisselen tussen programma's van de karakters die naar het
0340: beeldscherm worden gestuurd en van het keyboard worden ge-
0350: lezen. Dit is te realiseren door de OUTPUT en INPUT roufi-
0360: nes de mogelijkheid te geven om naast de terminal hun karak-
0370: ters ook met de cassetteinterface uit te kunnen wisselen.
0380: Hoe dit tot de genoemde mogelijkheden kan leiden vertel ik
0390: later, ik volsta even met een voorbeeld.
0400: Gegevens kunnen in Basic bewaard worden door ze met out-
0410: put naar tape met PRINT te printen en later met inout van
0420: tape weer binnen te halen met INPUT zoals hieronder:
0430:
0440: Datum=maandag
0450: 10 A$="Hans "+"en Grietje"           Datageneer programma
0460: 20 B=3.14159*((361.27/12)^2.5)
0470: 30 PRINT A$
0480: 40 PRINT B
0490: RUN
0500: Hans en Grietje           "Outout naar tape"
0510: 15623.4211
0520:
0530: Datum=dinsdag
0540: 10 INPUT A$           Geheugen is leeg
0550: 20 INPUT B           Datalees programma
0560: 30 PRINT A$,B
0570: RUN
0580: ? Hans en Grietje           "Inout van tape"
0590: ? 15623.4211
0600: Hans en Grietje 15623.4211   Data zit weer in computer
0610:
0620: Als dit in de praktijk wil werken moet OUTPUT wel de line-
0630: feeds negeren, alsmede de Null karakters ($00) die Basic mee-
0640: zendt voor teletypes die voor de carriage return tijd nodig
0650: hebben en geen karakters kunnen bufferen. De gesavede karak-
0660: ters gaan dienen als keyboard intoetsingen en daar worden
0670: linefeed en $00 ook nooit gebruikt.

```

0680: Het printen van meerdere gegevens in een PRINT statement is
0690: mogelijk maar dan moeten wel komma's worden tussengevoegd dus
0700: PRINT A\$;",";B om later door INPUT te worden geslikt. Het
0710: staat echter veel minder duidelijk.
0720:
0730: De sociale tape routines weten door het geset zijn van be-
0740: paalde flags en het ontvangen van bepaalde karakters of de
0750: input en/of output wel of niet met tape moet gebeuren. Een
0760: Basic programma kan dit zelf verzorgen. Als dit switchen met
0770: de hand moet gebeuren is het aan te raden vier controlcodes
0780: te reserveren (voor wel/niet tape input/output), die dan het
0790: switchen verzorgen.
0800: Aan het einde van dit artikel zijn twee assemblerprogramma's
0810: gegeven, OUTTAP en INTAP, die de tabel/O verzorgen. Hoe deze
0820: in de normale INPUT en OUTPUT routine gebruikt moeten worden
0830: staat hierna vermeld.
0840:
0850: Het formaat van de I/O karakters op de cassette
0860: -----
0870: De karakters die naar tape worden geschreven zullen in blok-
0880: ken moeten worden weggeschreven. Blokken van 1 karakter is
0890: niet economisch door overhead als synchronisatie karakters
0900: en blokken van enkele kilobytes vraagt te veel systeemRAM en
0910: is oneconomisch bij kleine files. Hier is een blok-grootte
0920: van 128 bytes gekozen. Een file is een afgeschermda hoeveel-
0930: heid karakters. Een file die kleiner is dan 128 bytes wordt
0940: toch als een blok van 128 bytes in een record weggeschreven,
0950: aangevuld met wat er toevallig in de buffer stond. Bij het
0960: inlezen worden alleen de serieuze karakters overgenomen. Bij
0970: files die groter dan 128 bytes zijn wordt een serie van re-
0980: cords weggeschreven.
0990: Elk blok wordt afgesloten met een karakter \$1C als 128e kar-
1000: rakter. Als het laatste karakter van een file in de blok-
1010: buffer is gezet en er dus geen karakters voor die file meer
1020: komen wordt deze laatste buffer afgesloten met \$1E in plaats
1030: van \$1C. Normaal wordt de blok-buffer pas op band geschreven
1040: als de buffer vol is. Als het laatste karakter van de file
1050: echter is ontvangen dan wordt de buffer met de laatste karak-
1060: ters van de file gelijk op band gezet.
1070: De eerste record van een file wordt op band gezet met ID=\$01.
1080: de daaropvolgende records worden steeds met een ID geschreven
1090: die 1 hoger is dan die van de voorafgaande record.
1100: Het begin van een file is nu te herkennen aan een ID=\$01 en
1110: het einde is te herkennen aan een record die afgesloten is
1120: met \$1E in plaats van \$1C.
1130: Input van tape en output naar tape kan gelijktijdig gebeuren
1140: omdat elk zijn eigen variabelen en vooral buffer heeft.
1150:
1160: De output naar tape routine OUTTAP
1170: -----
1180: De output wordt naar tape geschreven vanaf het moment dat het
1190: msb van de variabele/flag TAPOUT is geset. OUTTAP controleert
1200: het msb en indien die niet geset is wordt het karakter niet
1210: naar tape gestuurd. Als het msb wel geset is wordt ook bit 6 van
1220: TAPOUT, die voor de initialisatie zorgt, getest. Als die niet
1230: geset is wordt de bufferpointer TPNTD op \$00 gezet en de ID-
1240: teller op \$01 gezet en omdat de initialisatie voor het weg-
1250: schrijven van een nieuwe file dan klaar is wordt ook bit 5
1260: geset zodat er maar eenmaal wordt geïntialiseerd per file.
1270: Als bit 6 en 7(msb) zijn geset wordt gecontroleerd of het weg
1280: te zetten karakter \$0A=Linefeed of \$00=Null is want die moeten
1290: worden genegeerd. Als het karakter gelijk is aan \$1E wordt het
1300: in de buffer gezet om de file af te sluiten en wordt de buffer
1310: naar cassette geschreven, daarna worden bit 6 en 7 van TAPOUT
1320: gereset zodat de output weer alleen naar de terminal gaat tot-
1330: dat bit 7 opnieuw wordt geset voor een nieuwe file.
1340: Als het een gewoon karakter is wordt het mov. de pointer in de
1350: buffer gezet en wordt de pointer opgehoogd. Als de pointer nu

1360: de waarde \$7F heeft is de buffer bijna vol en kan alleen de
1370: bufferafsluiter \$1C er nog bij. Dan wordt de buffer op band ge-
1380: zet en kan de pointer weer naar het begin van de buffer wij-
1390: zen voor een nieuwe record. De ID wordt ook opgehoogd zodat de
1400: volgende record met een hogere ID wordt gesaved.
1410:
1420: De routine voor output naar tape OUTTAP zou als volgt in het
1430: OS opgenomen kunnen worden:
1440:
1450: OUTPUT CMPIM \$1E File-afsluiter alleen naar OUTTAP
1460: BEQ C
1470: JSR OUTTTY Karakter altijd (ook) naar TTY
1480: C JSR OUTTAP Controleert zelf msb. TAPOUT
1490: RTS
1500:
1510: Samenvattend moet output naar tape dus worden gestart door
1520: TAPOUT=\$80 te maken en worden beëindigd door \$1E naar OUTPUT
1530: te sturen.
1540:
1550: De controlcodes die door het OS worden herkend om de output
1560: wel of niet naar tape te sturen moeten altijd een paar din-
1570: gen controleren. Als de output naar tape moet dan mag er al-
1580: leen \$80 in TAPOUT worden gezet als bit 6 niet al is geset.
1590: OUTTAP is dan nl. nog met een file bezig en het resultaat
1600: zou zijn dat die file niet netjes wordt afgesloten omdat de
1610: pointer wordt gereset en de karakters in de buffer dus ver-
1620: loren zijn en de laatste record afgesloten is met \$1C ipv.
1630: \$1E. Als deze file dan wordt gelezen zal de eerste record
1640: van de nu geïntialiseerde file ook worden gelezen want het
1650: karakter \$1E is nog niet voorgekomen. De ID van die eerste
1660: record is echter \$01 en de leesroutine wil een record met
1670: een ID die 1 hoger is dan de laatst gelezen record van de
1680: vorige file, dit loopt dus in het honderd. Het uitzetten van
1690: input naar tape mag dus ook niet gebeuren door \$80 in TAPOUT
1700: te zetten maar door een karakter \$1E naar OUTPUT te sturen.
1710: Als de controlcode echter wordt onderschept in een inte-
1720: ruptroutine (die op een keyboardinterupt reageert) dan kan
1730: het gebeuren dat die interupt gebeurt terwijl de OUTPUT rou-
1740: tine wordt uitgevoerd en als de interuptroutine dan OUTPUT
1750: ook nog eens aanroep geeft dit waarschijnlijk problemen. De
1760: oplossing ligt in zelfdiscipline maar dit beperkt de mogelijk-
1770: heden omdat dan alleen de output naar tape kan worden ge-
1780: stopt als geen uitvoer wordt geleept. Als de uitvoer met een
1790: andere controlcode kan worden onderbroken is er geen probleem
1800: meer. Mijn OS heeft twee flags die resp. geset zijn als er
1810: op keyboardinput wordt gewacht en als de uitvoer tijdelijk
1820: is gestopt, door te testen of een van beide is geset kan het
1830: OS bepalen of het veilig \$1E naar OUTPUT kan sturen.
1840:
1850: De input van tape routine INTAP
1860:
1870: Input van tape halen wordt gestart door het msb van TAPIN
1880: te zetten. Als INTAP wordt aangeropen en bit 6 van TAPIN
1890: is niet geset wordt enige initialisatie gedaan voor het in-
1900: lezen van een nieuwe file zoals de lees-ID=\$01 maken zodat
1910: bij het begin van de file wordt begonnen, de bufferpointer
1920: op \$00 zetten en bit 6 zetten zodat de initialisatie maar
1930: eenmaal wordt gedaan. Ook moet er direkt een record worden
1940: gelezen omdat de buffer leeg is. Als bit 6 en 7 zijn geset
1950: wordt met de pointer een kar. van de buffer gehaald en wordt
1960: de pointer opgehoogd. Als het karakter \$00 is wordt dit gene-
1970: geert en een volgend karakter van de buffer gehaald. Hoe dit
1980: in de record kan komen wordt verderop verteld. Als het karak-
1990: ter \$1C is dan is de buffer leeg en wordt eerst een nieuwe
2000: record gelezen, en de ID-teller opgehoogd zodat de volgende
2010: record de juiste ID heeft. Ook wordt de pointer \$00 gemaakt en
2020: een karakter uit de buffer gehaald. Als het karakter \$1E is
2030: zijn alle karakters van de file gelezen en moet naar de aanvra-
2040: ge teruggekeerd worden met een karakter van het keyboard.
2050: Er is nu een klein probleempje namelijk dat INPUT net beslo-
2060: ten had dat INTAP voor het karakter zorgt omdat bit 7 van
2070: TAPIN geset was. Dit is als volgt opgelost, als \$1E van de
2080: buffer wordt gehaald dan zet INTAP in TAPIN \$00 en gaat dan
2090: met de carry geset terug naar INPUT die in dat geval over-
2100: nieuw begint en dan het karakter van het keyboard haalt om-

2110: dat bit 7 van TAPIN gereset is. Als het karakter dat INTAP
2120: van buffer haalt een gewoon karakter is gaat INTAP met dat
2130: karakter en de carry gecleared terug naar INPUT. De INPUT
2140: routine kan nu als volgt worden opgebouwd:
2150:
2160: INPUT BIT TAPIN Moet inoutkarakter van tape komen ?
2170: BPL A
2180: JSR INTAP Haal karakter van tape
2190: BCS INPUT Einde van tape-file bereikt ?
2200: BCC B
2210: A JSR INKEYB Haal karakter van keyboard
2220: B RTS
2230:
2240: Samenvattend moet dus om inout van tape te starten het msb.
2250: van TAPIN worden geset. De input wordt automatisch weer van
2260: het keyboard betrokken als de file is afgelopen.
2270:
2280: De controlcodes die het mogelijk maken vanaf het keyboard te
2290: bepalen of de input van tape of van het keyboard komt hebben
2300: erg weinig werk. Het stoppen van inout van tape mag altijd
2310: en gebeurt door TAPIN=#00 te maken (bit 6 en 7 gereset). Het
2320: beginnen met inout van tape mag in principe ook altijd door
2330: TAPIN \$80 te maken maar als op dat moment de inout al van
2340: tape komt dan komen er op de regel die op dat moment door het
2350: hoofdprogramma wordt ingelezen met INPUT zowel karakters van
2360: de oude als de nieuwe file zodat de eerste regel van de nieu-
2370: we file niet aan het begin van de regel komt. Als dit een
2380: commando is dan gaat het fout. Het is dus aan te raden om
2390: input van tape aanzetten alleen toe te staan als bit 7 nog
2400: niet geset is.
2410:
2420: Naderhand records veranderen
2430: -----
2440: Als een stukje output voor bepaalde doeleinden met output
2450: naar tape is gesaved maar er zijn ongewenste karakters in
2460: gekomen zoals het commando voor die output of de prompt na-
2470: dat de output klaar was dan kunnen deze door twee eigenschap-
2480: pen van INTAP worden verwijderd. Deze eigenschappen zijn het
2490: negeren van het \$00 karakter en het dedecteren van het einde
2500: van een buffer aan de hand van het \$1C (of \$1E) karakter
2510: ipv. door te kijken of de pointer \$7F is. Ongewenste karak-
2520: ters kunnen nu door \$00 worden vervangen of met een blockmo-
2530: ve worden ge"delete". Wat er dus moet gebeuren is de record
2540: opzoeken en laden, de veranderingen aan brengen en dan het
2550: geheel weer save onder dezelfde ID.
2560:
2570: Leesfouten afhandeling
2580: -----
2590: Door het nummeren van de records kan het niet gebeuren dat
2600: een record wordt overgeslagen. Een leesfout tijdens het le-
2610: zen van een file is dus niet fataal. Als elke record ech-
2620: ter maar eenmaal is gesaved dan worden alle daaropvolgende
2630: records genegeerd. Als de file erg lang is en je bent
2640: koffie gaan drinken tijdens het inlezen dan is het handig te
2650: weten welke ID die record had zodat je bij die record opnieuw
2660: kan beginnen. Ik save elke record eenmaal omdat ik bijna nooit
2670: last heb van leesfouten (tok, tok, even afkloppen). Mijn OS
2680: print bij het tegenkomen van een foute ID zowel de gelezen
2690: ID als de gewenste ID, als ik dan verzadigd van de koffie
2700: weer achter de terminal plaats neem en er zijn ondertussen
2710: meer foute ID's geprint dan op het scherm passen dan kan ik
2720: nog steeds zien welke ID de gemiste record heeft. Het is
2730: echter ook mogelijk elke record twee maal te save. Als de
2740: eerste record goed wordt gelezen wordt de tweede vanwege de
2750: ID genegeert, en als de eerste niet goed wordt gelezen wordt
2760: de tweede automatisch wel gelezen. De leesroutine moet het
2770: dan wel uit zichzelf opnieuw proberen, de standaard KIM
2780: routine is dus niet te gebruiken en is sowieso niet als sub-
2790: routine aan te roepen. Het is dus wel handig een zelfoe-
2800: schreven OS te hebben met eigen save en load routines, wat
2810: trouwens ook al geldt voor het implementeren van de 4 con-
2820: trolcodes. Mijn save routine saved altijd twee maal, behalve
2830: als de SINGLE flag geset is die Save zelf naderhand weer re-
2840: set. Je moet zelf maar zien of je een of twee maal wilt sa-
2850: ven, als je het eenmaal doet kan je ook altijd de hele file
2860: twee maal save wat altijd verstandig is ivm. dropouts op de
2870: band.
2880:

2890: Enkele toepassingen
2900: -----
2910: Nu uitgelegd is hoe het save en laden van de karakterI/O gaat
2920: zal ik de toepassingen vertellen die ik bedacht heb en waar
2930: iedereen wel een eigen toepassing aan toe kan voegen.
2940: Het save van data (reals, integers, strings ed.) in Basic is
2950: in het begin al uitgelegd om de interesse te wekken. Ik ben
2960: met het realiseren van INTAP en OUTAP begonnen nadat ik om-
2970: streeks december vorig jaar behoefte kreeg aan een adres-
2980: bestand. In Basic zou dat goed kunnen vanwege de stringmani-
2990: pulatie commando's maar data wegschrijven in files kan mijn
3000: Basic niet of zeer moeilijk. In assembler werd het te veel
3010: werk. Toen ik omstreeks die tijd een programma bekeek waarin
3020: een lange reeks bytes in blokken naar cassette werden geschre-
3030: ven kwam ik op het idee dit ook met karakteroutput te doen.
3040: Ik heb in Basic een simpele namenbestand-manager geschre-
3050: ven die verderop is gelist als onderdeel van deze MicroAde file
3060: waarin ook deze beschrijving staat (wil de redactie dat even
3070: bevestigen?). Dit is gedaan door onder Basic de controlcode
3080: voor output naar tape in te tikken en dan gewoon LIST te ge-
3090: ven en na de OK weer "output naar terminal" te geven. Onder
3100: MicroAde geef je het A(open) commando en "input van
3110: tape" en speelt de onder Basic opgenomen file af. Als de file
3120: is gelezen wordt de input automatisch weer van het keyboard
3130: gehaald door het \$!E karakter waarmee de file eindigde.
3140: Nu zijn een aantal dingen mogelijk als je 8k MicroAde hebt.
3150: Met het V(ind) commando kan je een Cross Reference maken van
3160: de variabelen (of zelfs commando's) met V:(variabele naam):
3170: waarna alle regels waar deze variabele voorkomt verschijnen
3180: (als je eerst LT 10 of zo geeft dat print het V commando ook
3190: geen MA-regelnummers en heb je alleen de Basic regelnummers).
3200: Dit vereist soms wel enige naverwerking met de hand, want als
3210: je met V een variabele zoekt met een korte naam als "R" dan
3220: zullen ook de R in PRINT, FOR ed. worden "gevonden".
3230: Met het V(ervang) commando kan je ook de naam van een varia-
3240: bele consequent in het hele programma vervangen. Het moet
3250: dan wel mogelijk zijn het Basic programma weer in Basic-vorm
3260: te gieten. Dit gaat door het list zonder regelnummers-commando
3270: te geven (LT) en dan voor dat je return indrukt "output naar
3280: tape" te geven. Na het listen weer "output naar terminal" geven.
3290: Onder Basic geef je "input van tape" en speelt de onder MA
3300: opgenomen file af en Basic (en MA al eerder) zal niet beter
3310: weten of je zit het hele programma met de hand in te tikken
3320: (onderhand het wereldrecord tikken verpulverend). De spaties
3330: die LT aan het begin zet worden door Basic genegeert, en als
3340: "LT" of "Ok" meegekomen zijn zal Basic alleen een ongevaar-
3350: lijke foutmelding geven.
3360: Dit brengt me gelijk naar de volgende toepassing namelijk het
3370: er op na houden van een subroutine bibliotheek in Basic, die
3380: ook een andere mogelijkheid inhoudt nl. het "merge"-en van Ba-
3390: sic programma's, te vergelijken met het commando H in MicroAde.
3400: De te mergen programma's worden afzonderlijk geladen onder
3410: Basic en dan met "output naar tape" gelist. Als er een Basic
3420: programma in het geheugen zit kan nu een ander worden ge-
3430: "merge"d door "input van tape" te kiezen en dan de file die
3440: gemaakt is toen het toe te voegen Basic programma onder "out-
3450: out naar tape" werd gelist, af te spelen. Hierbij moet je er
3460: natuurlijk op letten dat beide gedeeltes andere regelnummers
3470: hebben. Dit kunnen mergen is de voorwaarde voor het er op
3480: na kunnen houden van een subroutine bibliotheek. Een voor-
3490: beeld is een aantal grafische routines voor lijn trekken, of
3500: een vlak inkleuren die in meerdere programma's gebruikt wor-
3510: den. Deze routines zullen van tijd tot tijd verbeterd of uit-
3520: gebreed worden en als ze dan in elk programma zitten ingebouwd
3530: dan moet je steeds al die programma's aanpassen. Als ze
3540: echter in een bibliotheek zitten hoeft je ze maar eenmaal aan
3550: te passen. Je hoeft ze pas te linken als je een hoofdprogramma
3560: wilt draaien. Dit linken is zelfs misschien wel in te bouwen in
3570: het programma dat ze nodig heeft, je hoeft dan alleen maar de
3580: betreffende files met routines af te draaien op de cassette-
3590: recorder. Hierbij is de volgende constructie mogelijk:
3600:
3610: Programma dat aangevuld moet worden met subroutine:
3620: 10 REM Hoofdprogramma
3630: :
3640: :
3650: 200 FOR N=1 TO (aantal te mergen subroutine)
3660: 210 POKE (TAPIN,128)

```

3670:      230 END
3680:      240 NEXT N
3690:
3700:      Te lezen bibliotheek-subroutine:
3710:      "500 REM Subroutine
3720:      :
3730:      :
3740:      590 RETURN
3750:      GOTO 240
3760:      ($1E)"
3770:
3780: De file die de aanvulling verzorgt (alleen bij automatisch
3790: mergen) kan niet met listen onder "outout naar tape" alleen
3800: worden gemaakt maar vereist enig editten van de cassette re-
3810: cords. Na het lezen van de vorige paragrafen zal dit echter
3820: niemand onmogelijk zijn.
3830: Het zichzelf uitbreiden van een Basic programma met subrou-
3840: tines dat Artificial Yntelligence-geïnteresseerden leuk
3850: zullen vinden brengt ons automatisch bij de laatste toepas-
3860: sing die ik op het ogenblik kan verzinnen namelijk autoexecute
3870: files. Dit houdt in dat je "input van tape" kiest en dan een
3880: file afsoeelt op de cassetterecorder die alle intoetsingen
3890: bevat die je vanaf dat moment zou moeten geven om een bepaal-
3900: de klus te klaren. Dit ka~ het opstarten van MicroAde zijn
3910: tot op het punt dat het A(ppend) commando is gegeven of het
3920: veranderen van een bepaald programma zodat het onder een be-
3930: paald Operating System of versie daarvan kan draaien. De mo-
3940: gelijkheden hangen sterk van het OS af of eigenlijk van het
3950: programma dat de input verwerkt. MS-DOS bijvoorbeeld heeft een
3960: een soort programmeertaal aan de DOS-commando's toegevoegd
3970: die het o.a. mogelijk maakt halverwege het auto-executeren va-
3980: riabele gegevens aan de persoon achter het toetsenbord te
3990: vragen. Dit kan bijvoorbeeld de tijd en datum of zo zijn die
4000: dan in de juiste adreslocaties wordt gezet. Dit wordt bijv. uit-
4010: gevoerd door een autoexecute file die de boot-routine bij het
4020: aanzetten van de computer altijd uitvoert. Ook kan je hier het
4030: automatisch laden van Basic of zo inzetten zodat je enige se-
4040: condon na het aanzetten van de computer in Basic of zelfs een
4050: draaiend Basic programma zit. De autoexecute files kunnen ook
4060: worden gebruikt voor het creëren van nieuwe commando's in
4070: Basic of MA, opgebouwd met de standaard commando's.
4080:
4090: Al met al geloof ik dat ik de KIM en Junior bezitters die er
4100: over denken hun simpele computertje in te ruilen voor iets
4110: fraais van ca. tienduizend piek, maar dat bedrag toch niet
4120: kunnen opbrengen weer enige hoop heb gegeven. Met enige in-
4130: ventiviteit zoals hierboven ten toon is gespreid valt er nog
4140: best wat van te bakken en kan je nog zeer regelmatig de kick
4150: krijgen van het bedienen van een krachtigere computer.
4160:
4170: Het simpele DataBaseManager programma in Basic
4180: -----
4190:
4200: 10 REM Simpel DataBaseManager programma
4210: 12 N=0:REM Het aantal namen in bestand
4220: 15 DIM A$(500)
4230: 20 PRINT:PRINT
4240: 25 PRINT"1 Toevoeging aan bestand intikken
4250: 30 PRINT"2 Toevoeging aan bestand van band lezen
4260: 40 PRINT"3 Bestand op band zetten
4270: 50 PRINT"4 Bestand printen
4280: 60 PRINT"5 Naam veranderen
4290: 70 PRINT"6 Bestand wissen
4300: 80 INPUT"Geef uw keuze (1-6) : ";Z
4310: 90 ON Z GOTO 100,500,1000,1500,2000,2500
4320: 95 GOTO 20
4330: 99 .
4340: 100 INPUT"Hoeveel records toevoegen : ";Z

```

```

4350: 110 FOR K=N+1 TO N+Z
4360: 120 PRINT K;"=" " : INPUT A$(K-1)
4370: 130 NEXT K
4380: 135 N=N+Z
4390: 140 GOTO 20
4400: 199 .
4410: 500 POKE 57100,128:REM tapin=$DF0C=57100
4420: 510 INPUT Z: REM aantal namen in file
4430: 520 FOR K=N+1 TO N+Z
4440: 530 INPUT A$(K-1)
4450: 540 NEXT K
4460: 550 N=N+Z
4470: 560 GOTO 20
4480: 999 .
4490: 1000 IF N=0 THEN 20
4500: 1001 PRINT"Zet recorder op opnemen en druk toets in " : GET Z$
4510: 1002 PRINT CHR$(10):REM linefeed
4520: 1005 POKE 57097,128:REM tapout=$DF09=57097
4530: 1010 PRINT N: REM aantal namen in file
4540: 1020 FOR K=1 TO N
4550: 1030 PRINT A$(K-1)
4560: 1040 NEXT K
4570: 1050 PRINT CHR$(30):REM End of file=$1E=30
4580: 1060 GOTO 20
4590: 1499 .
4600: 1500 IF N=0 THEN 20
4610: 1510 FOR K=1 TO N
4620: 1520 IF K=20*INT(K/20) THEN GET Z$:GOTO 1540
4630: 1530 PRINT " " :
4640: 1540 PRINT K;" " : A$(K-1)
4650: 1550 NEXT K
4660: 1560 GOTO 20
4670: 1999 .
4680: 2000 INPUT"Geef nummer van record":Z
4690: 2010 IF N=0 OR Z>N OR Z<1 THEN 20
4700: 2020 PRINT"Record luidt " : A$(Z-1)
4710: 2030 INPUT"Geef nieuwe record " : A$(Z-1)
4720: 2040 GOTO 20
4730: 2499 .
4740: 2500 INPUT"Weet u het zeker (Y/N)":Z$
4750: 2510 IF Z$(0)"Y" THEN 2530
4760: 2520 N=0:PRINT"Bestand is gewist"
4770: 2530 GOTO 20
4780:

```

4790: Listing van OUTAP en INTAP routines

```

4800: -----
4810:
4820: Aangeroepen OS subroutines
4830: 1B CC SAVE * $CC1B Save met ID van DSA tot DEA
4840: 7B CC GET * $CC7B Lees file met ID vanaf RSA
4850:
4860: Parameters voor cassetteroutines
4870: 00 DF ID * $DF00 Lees/Save file met ID
4880: 01 DF SINGLE * ID +01 Save eenmaal (zie tekst)
4890: 02 DF DSAL * SINGLE +01 Startadres datagebied
4900: 03 DF DSAH * DSAL +01
4910: 04 DF DEAL * DSAH +01 Eindadres+1 datagebied
4920: 05 DF DEAH * DEAL +01
4930: 06 DF RSAL * DEAH +01 Zet data vanaf RSAL/H
4940: 07 DF RSAH * RSAL +01
4950:
4960: Variabelen voor tapeI/O routines
4970: 08 DF CHAR * RSAH +01 Voor bewaren karakter
4980:
4990: 09 DF TAPOUT * CHAR +01 Flag:Output naar tape
5000: 0A DF TPNTD * TAPOUT +01 Pointer in outputbuffer
5010: 0B DF TIDOUT * TPNTD +01 Save-ID voor buffer
5020:
5030: 0C DF TAPIN * TIDOUT +01 Flag:Inout van tape
5040: 0D DF TPNTI * TAPIN +01 Pointer in inputbuffer
5050: 0E DF TIDIN * TPNTI +01 ID van volgende record
5060:
5070: I/O buffers voor records op tape
5080: 00 DE TBUFD * $DE00 Outputbuffer naar tape
5090: 80 DE TBUFI * TBUFD +80 Inoutbuffer van tape
5100:

```



```

5880: D085 A9 C0          LDAIM $C0      Set bit 6 en 7
5890: D087 8D 0C DF      STA TAPIN
5900: D08A A9 01          LDAIM $01      Eerste record heeft ID=01
5910: D08C 8D 0E DF      STA TIDIN
5920: D08F D0 24          BNE BSR        Lees record en reset pointer
5930:
5940: D091 AE 0D DF      BSC          LDX TPNTI     Haal karakter uit buffer
5950: D094 8D 80 DE      LDAAX TBUFI
5960: D097 8D 08 DF      STA CHAR
5970: D09A E8            INX           Hoog pointer op
5980: D09B 8E 0D DF      STX TPNTI
5990: D09E C9 00          CMPIM $00
6000: D0A0 F0 EF          BEQ BSC       Negeer edit-Null characters
6010: D0A2 C9 1E          CMPIM $1E     End of file ?
6020: D0A4 D0 0B          BNE BSP
6030: D0A6 A9 00          LDAIM $00     Input weer van keyboard
6040: D0A8 8D 0C DF      STA TAPIN
6050: D0AB 68            PLA
6060: D0AC A8            TAY
6070: D0AD 68            PLA
6080: D0AE AA            TAX
6090: D0AF 38            SEC           Input van keyboard halen
6100: D0B0 60            RTS
6110:
6120: D0B1 C9 1C          BSR          CMPIM $:C     Einde van record ?
6130: D0B3 D0 1D          BNE BSE
6140: D0B5 AD 0E0DF      BSR          LDA TIDIN     Record moet ID=TIDIN hebben
6150: D0B8 8D 00 DF      STA ID
6160: D0BB A9 80          LDAIM TBUFI   Nieuw record lezen in buffer
6170: D0BD 8D 06 DF      STA RSAL
6180: D0C0 A9 DE          LDAIM TBUFI   /
6190: D0C2 8D 07 DF      STA RSAH
6200: D0C5 20 78 CC      JSR GET
6210: D0C8 EE 0E DF      INC TIDIN     Volgende record's ID
6220: D0CB A9 00          LDAIM $00     Reset pointer
6230: D0CD 8D 0D DF      STA TPNTI
6240: D0D0 F0 BF          BEQ BSC       Haal karakter uit buffer
6250:
6260: D0D2 68            BSE          PLA
6270: D0D3 A8            TAY
6280: D0D4 68            PLA
6290: D0D5 AA            TAX
6300: D0D6 AD 08 DF      LDA CHAR     File had nog een karakter
6310: D0D9 18            CLC
6320: D0DA 60            RTS
6330:
6340: @

```

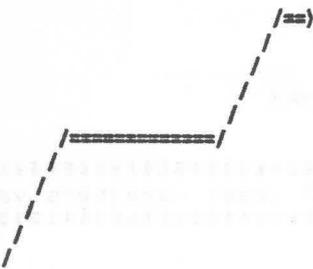
AUTOMATISCHE DATAREGEL TELLER COMMODORE 64

Met dit programma verschijnt in de linkerbovenhoek van het scherm de totale oetelling van de ingetoetste regel.

```

100 PRINT*(CLR)EVEN GEDULD...
105 FORI=886TO1018:READA:CK=CK+A:POKEI,A:NEXT
110 IFCK(1)17539THENPRINT*(DOWN)NIET GOED GELADEN
115 PRINT*DE DATA STATEMENTS ZIJN VERMINKT.*:END
120 SYS886:PRINT*(CLR)(2 DOWN)AUTOMATISCHE DATAREGEL TELLER.*:NEW
886 DATA 173,036,003,201,150,200
892 DATA 001,096,141,151,003,173
898 DATA 037,003,141,152,003,169
904 DATA 150,141,036,003,169,003
910 DATA 141,037,003,169,000,133
916 DATA 254,096,032,087,241,133
922 DATA 251,134,252,132,253,008
928 DATA 201,013,240,017,201,032
934 DATA 240,005,024,101,254,133
940 DATA 254,165,251,166,252,164
946 DATA 253,040,096,169,013,032
952 DATA 210,255,165,214,141,251

```



```

958 DATA 003,206,251,003,169,000
964 DATA 133,216,169,019,032,210
970 DATA 255,169,018,032,210,255
976 DATA 169,058,032,210,255,166
982 DATA 254,169,000,133,254,172
988 DATA 151,003,192,087,208,006
994 DATA 032,205,189,076,235,003
1000 DATA 032,205,221,169,032,032
1006 DATA 210,255,032,210,255,173
1012 DATA 251,003,133,214,076,173
1018 DATA 003

```

FORTH OP JUNIOR COMPUTERS DEEL 3.

door: G. van Opbroek
 Hooglanden 20
 9801 LB Zuidhorn

(GEVOP)

1. Inleiding.

In deze aflevering wil ik iets meer vertellen over de manier waarop FORTH in elkaar zit. Bovendien zal ik trachten duidelijk te maken waarom de Hocus Spokes-achtige opdrachten van Fridus Jonkman [1] nodig zijn om woorden in FORTH vast te zetten (te LOCKEN).

Om een en ander duidelijk te maken begin ik met een beschrijving van de geheugenindeling van een standaard FORTH systeem zoals die op de JUNIOR gebruikt wordt. Ik zal me in deze aflevering voornamelijk beperken tot fig-FORTH en de daarvan afgeleide versies.

2. Geheugen indeling.

In figuur 1. is de geheugenindeling van de FORTH uit de cassette bibliotheek weergegeven. Deze indeling is representatief voor andere op fig-FORTH gebaseerde systemen op een JUNIOR. De RAM-DISC wordt gebruikt als tussenopslag van gegevens. Met behulp van het woord TO-TAPE kan de inhoud hiervan naar cassette geschreven worden en met behulp van het woord FROM-TAPE wordt dit gebied gevuld met gegevens die van de cassette gehaald worden.

Bij systemen met tenminste een floppy-disc of ander random access opslagmedium vervalt meestal de RAM-DISC en kan dit verdeeld worden over de andere blokken.

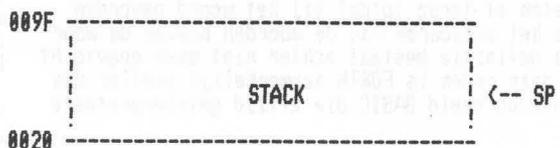
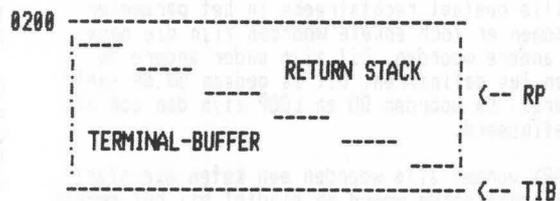
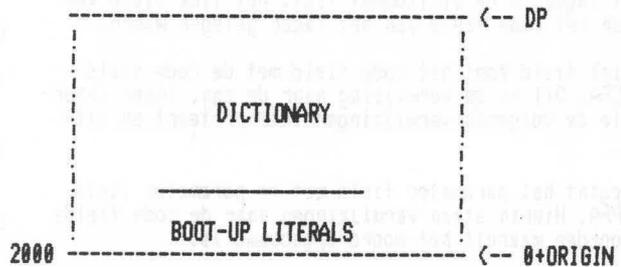
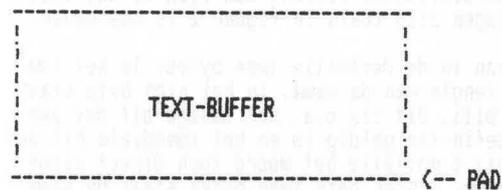
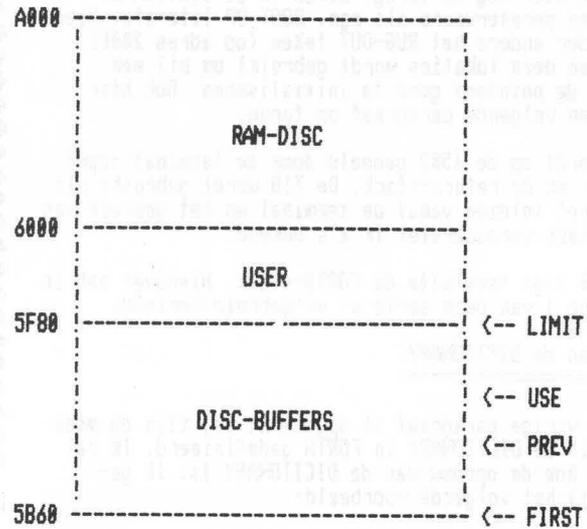
Vlak onder de RAM-DISC ligt het USER-gebied. Hierin liggen een aantal zgn. user variabelen. Deze worden gedefinieerd met het woord "USER". Voorbeelden hiervan zijn de variabelen TIB, FENCE, CONTEXT, VOC-LINK, Dit zijn meestal variabelen in FORTH zelf gebruikt worden. Een aantal van deze variabelen zullen we in deze aflevering nog tegen komen.

Direct onder het USER-gebied liggen de DISC-BUFFERS. Deze buffers vormen het interface tussen informatie op disc of in de RAM-DISC en FORTH. Als we met behulp van de woorden BUFFER of BLOCK het startadres van een DISC-blok opvragen dan krijgen we altijd een adres in de DISC-BUFFERS. Als het opgevraagde blok niet in dit gebied aanwezig is, dan zorgt FORTH ervoor dat het betreffende blok van disc gehaald wordt. Door het woord UPDATE kan aangegeven worden dat er iets in het blok gewijzigd is waarna FORTH ervoor zorgt dat het blok teruggeschreven wordt als het betreffende gebied in de DISC-BUFFERS voor een ander blok nodig is of wanneer de opdracht SAVE-BUFFERS of FLUSH gegeven wordt.

De bovengenoemde gebieden hebben een vaste positie in het geheugen, dit in tegenstelling tot de TEXT BUFFER, naar zijn pointer ook wel PAD genoemd. Dit gebied begint altijd op een vaste afstand van het eind van de DICTIONARY. De TEXT BUFFER wordt o.a. gebruikt voor het formateren van getallen.

Tussen de TEXT BUFFER en de DICTIONARY liggen 68 bytes. In de FORTH documentatie word dit de "word" buffer genoemd. Het is mij nog niet duidelijk waar dit gebiedje voor gebruikt wordt.

Gaan we nog lager in het geheugen, dan komen we in de DICTIONARY. Om dit gebied draait het in het gebruik van de programmeertaal FORTH. In de DICTIONARY staat hoe de FORTH-woorden uitgevoerd moeten worden. Meestal zijn deze



Figuur 1: Geheugen indeling van fig-FORTH voor de JUNIOR.

definities weer in de taal FORTH zelf. In de volgende paragraaf kom ik hier nog op terug. Boven in de DICTIONARY zijn 22 bytes gereserveerd als zgn. BOOT-UP literals. Hierin staat onder andere het RUB-OUT teken (op adres 200E). De inhoud van deze lokaties wordt gebruikt om bij een koude start de pointers goed te initialiseren. Ook hier kom ik in een volgende paragraaf op terug.

Pagina 01 wordt op de 6502 gedeeld door de terminal-input-buffer (TIB) en de return-stack. De TIB wordt gebruikt als buffer bij het inlezen vanaf de terminal en het gebruik van de return-stack veronderstel ik als bekend.

In pagina 00 ligt tenslotte de FORTH-stack. Hierover heb ik in aflevering 1 van deze serie al uitgebreid verteld.

3. Opbouw van de DICTIONARY.

Zoals in de vorige paragraaf al opgemerkt is, zijn de meeste woorden in de DICTIONARY in FORTH gedefinieerd. Ik zal beschrijven hoe de opbouw van de DICTIONARY is. Ik gebruik hierbij het volgende voorbeeld:

```
: TEST 12 15 + ;
: DRUK-HALLO 10 0 DO CR ." HALLO" LOOP ;
```

Als we nu in de DICTIONARY kijken, dan zien we dat deze woorden opgeslagen zijn zoals in figuur 2 is weergegeven.

Als eerste staan in de definitie twee bytes. In het low byte staat de lengte van de naam, in het high byte staan enkele status bits. Dit zij o.a. het smudge-bit dat aangeeft dat de definitie geldig is en het immediate bit dat aangeeft dat bij compilatie het woord toch direct uitgevoerd moet worden. Achter deze twee bytes staat de naam van het woord. Het betreffende gebied in de definitie heet NAME FIELD en de bijbehorende pointer heet name field pointer NFA. Meteen na het name field komt het link field met de link field pointer LFA. Dit is de verbinding met het woord dat lager in de DICTIONARY ligt. Het link field verwijst naar het name field van het lager gelegen woord.

Na het link field komt het code field met de code field pointer CFA. Dit is de verwijzing naar de zgn. inner interpreter die de volgende verwijzingen interpreteert en uitvoert.

Hierna begint het parameter field met de parameter field pointer PFA. Hierin staan verwijzingen naar de code fields van de woorden waaruit het woord opgebouwd is.

Opvallend is dat hoewel een groot deel van de woorden waaruit een definitie bestaat rechtstreeks in het parameter field terechtkomen er toch enkele woorden zijn die ongezet worden in andere woorden. Dit zijn onder andere de woorden die een lus definiëren. Dit is gedaan om de snelheid op te voeren. De woorden DO en LOOP zijn dan ook als IMMEDIATE gedefinieerd.

In de DICTIONARY vormen alle woorden een keten die start bij het laatst gedefieerde woord en eindigt bij het eerste woord. Als FORTH nu bij het INTERPRETERen van invoer een woord zoekt, start hij bij het laatst gedefinieerde woord en loopt de keten af terug totdat hij het woord gevonden heeft. Tijdens het uitvoeren van de woorden hoeven de woorden waaruit de definitie bestaat echter niet meer opgezocht te worden. Om deze reden is FORTH aanmerkelijk sneller dan een taal als bijvoorbeeld BASIC die altijd geïnterpreteerd wordt.

4. VOCABULARYs.

In FORTH bestaat de mogelijkheid zelf zgn. VOCABULARYs te definiëren. Een VOCABULARY bevat een aantal woorden die

44F9	.BYTE	\$84,	'TES',D4			<-- NFA
44FE	.WORD	44E3				<-- LFA
4500	.WORD	25D6				<-- CFA
4502	.WORD	202A		LIT		<-- PFA
4504	.WORD	000C		12		
4506	.WORD	202A		LIT		
4508	.WORD	000F		15		
450A	.WORD	2499		+		
450C	.WORD	375D		:		
450E	.WORD	240C		;S		
4510	.BYTE	\$8A,	'DRUK-HALL',CF			<-- NFA
451C	.WORD	44F9				<-- LFA
451E	.WORD	25D6				<-- CFA
4520	.WORD	202A		LIT		<-- PFA
4522	.WORD	000A		10		
4524	.WORD	265D		0		
4526	.WORD	41E4		<DO>		
4528	.WORD	22F4		CR		
452A	.WORD	2AF6		<.">		
452C	.BYTE	05,	'HALLO'			
4532	.WORD	423C		<LOOP>		
4534	.WORD	240C		;S		

Figuur 2: Wijze waarop het voorbeeld in de DICTIONARY opgeslagen is. (FORTH-79).

min of meer bij elkaar horen. Voorbeelden van dergelijke VOCABULARYs zijn EDITOR en ASSEMBLER. De basis-VOCABULARY heet FORTH.

Een VOCABULARY wordt gedefinieerd met de opdracht:

VOCABULARY naam

Om aan een bestaande VOCABULARY nieuwe woorden toe te voegen moet de opdracht:

naam DEFINITIONS gegeven worden.

De gedefinieerde woorden komen beschikbaar door het invoeren van de naam van de VOCABULARY.

Voorbeeld:

- 1) Met de opdracht EDITOR komen de woorden van de editor beschikbaar en kunnen hierna gebruikt worden.
- 2) Met de opdracht EDITOR DEFINITIONS kunnen woorden aan de editor toegevoegd worden.
- 3) Terug naar de basis-VOCABULARY gaan we na geval 1 met de opdracht FORTH en na geval 2 met de opdrachten FORTH DEFINITIONS

Intern is een DICTIONARY als volgt gedefinieerd. In FORTH bestaan twee variabelen: CURRENT en CONTEXT.

CONTEXT wijst naar een veld waarin het adres opgeslagen wordt van het NAME FIELD van het laatste woord in de VOCABULARY waarin het zoeken van een woord begonnen wordt. CURRENT wijst naar een dergelijk veld van de VOCABULARY waaraan nieuw te definiëren woorden worden toegevoegd. De velden waar CURRENT en CONTEXT naar toe wijzen liggen vlak boven de definitie van de naam van het betreffende VOCABULARY. Bij het invoeren van de naam van de VOCABULARY wordt het adres van dit veld in CONTEXT geschreven, de aanroep van het woord DEFINITIONS copieert de inhoud van CONTEXT in CURRENT.

Als er aan een VOCABULARY nieuwe woorden worden toegevoegd, dan wordt de inhoud van het veld waar CURRENT naar toe wijst gewijzigd. In wezen kan het definiëren van een nieuwe VOCABULARY gezien worden als het maken van een vertakking in de keten waarmee de woorden in de DICTIONARY verbonden zijn. Op de vertakkingen worden de eindpunten van de nieuwe ketens bijgehouden. Om het verwijderen van meerdere VO-

VOCABULARYs in een keer mogelijk te maken, zijn de VOCABULARYs met elkaar verbonden. Het eindpunt van deze keten staat in VOC-LINK.

5. Een koude start.

Bij een koude start moeten de pointers HERE (einde DICTIONARY), VOC-LINK, CONTEXT, CURRENT en FENCE (adres van het eerste woord dat eventueel verwijderd mag worden) geïnitieerd worden. Hiervoor wordt gebruik gemaakt van de zgn. BOOT-UP literals. Deze liggen in een gebied van 22 byte te beginnen op adres \$200C.

Bij een koude start wordt eerst de pointer naar het laatste woord in de FORTH VOCABULARY in de definitie van FORTH geschreven. Dit adres staat op \$200C. Vervolgens worden de stack pointers SP, en RP en de statische pointers TIB en UP geïnitieerd. Deze pointers liggen ook in de BOOT-UP literals. Op adres \$201C ligt het adres waarmee FENCE geïnitieerd wordt en op adres \$201E ligt het startadres van HERE. Op adres \$2020 ligt het adres waarmee VOC-LINK geïnitieerd moet worden. Bij een koude (of warme) start worden CURRENT en CONTEXT altijd op de VOCABULARY FORTH gezet.

Bij een koude start kunnen we op de volgende manier problemen krijgen:

Als we in FORTH een VOCABULARY gelockt hebben dan moet er in de definitie van deze VOCABULARY een verwijzing staan naar het laatste woord in deze VOCABULARY. Als we voorafgaand aan deze koude start woorden aan deze VOCABULARY hebben toegevoegd die niet gelockt zijn, dan verdwijnen deze woorden uit de DICTIONARY maar niet uit de definitie van de VOCABULARY. Het gevolg hiervan is dat het gebruik van woorden uit de betreffende VOCABULARY vernietiging van FORTH tot gevolg heeft.

6. Het locken van woorden en VOCABULARYs.

Ik hoop dat uit de vorige paragraaf enigszins duidelijk geworden is wat er moet gebeuren om woorden en VOCABULARYs in FORTH vast te houden, te locken. Om alles volkomen duidelijk te maken, zal ik de procedure stap voor stap bespreken.

1) Het locken van woorden.

1.1) Ga eerst naar de standaard toestand met de opdracht:

```
FORTH DEFINITIONS DECIMAL
```

1.2) In de eerste plaats moet FORTH verteld worden dat het betreffende gebied van de DICTIONARY niet verwijderd mag worden. Hiervoor gebruiken we de opdracht:

```
HERE FENCE !
```

1.3) Vervolgens moet het begin van de FORTH-VOCABULARY vastgelegd worden met de opdracht:

```
LATEST 12 +ORIGIN !
```

(+ORIGIN telt het begin van het ram-gebied (\$2000) bij het getal op de stack op).

1.4) Verder moet het nieuwe eind van de DICTIONARY op \$201E geschreven worden:

```
HERE 30 +ORIGIN !
```

1.5) Ook moet de nieuwe waarde van FENCE op \$201C onthouden worden:

```
HERE 28 +ORIGIN !
```

2) Het locken van VOCABULARYs.

Als er behalve uitbreidingen van bestaande VOCABULARYs ook geheel nieuwe VOCABULARYs gelockt moeten worden dan moet behalve de stapen 1.1 t/m 1.5 ook de volgende stap uitgevoerd worden.

2.1) Om VOC-LINK op de juiste waarde te kunnen initialiseren moet ingetypt worden:

```
VOC-LINK @ 32 +ORIGIN !
```

in [1] staat vermeld:

```
' naam 6 + 32 +ORIGIN !
```

waarbij naam de naam van de nieuwste VOCABULARY is. Dit komt echter op hetzelfde neer.

Met de bovenstaande procedure worden alle woorden en VOCABULARYs die op dat moment in de DICTIONARY aanwezig zijn gelockt.

FORTH kan nu weggeschreven worden door te vragen:

```
HERE HEX .
```

en het gebied vanaf \$2000 tot het gevonden adres op cassette of disc te schrijven.

Opmerking:

In PE-FORTH van PROTON voor de SENIOR hoeft alleen stap 1.2 uitgevoerd te worden waarna FORTH bij de aanroep MDN zelf het gebruikte geheugengebied afdruckt.

7. Unlocken van woorden en VOCABULARYs.

Als we gelockte woorden willen verwijderen dan moet de in paragraaf 6 genoemde procedure in principe omgekeerd worden. Dit gaat dan als volgt:

1) Het unlocken van woorden:

1.1) ' naam NFA FENCE !

waarbij naam het eerste woord is dat uit de DICTIONARY verwijderd moet worden.

Op deze wijze wordt het mogelijk gemaakt om de keten vanaf het woord 'naam' te verwijderen.

1.2) FORGET naam

De woorden vanaf naam worden daadwerkelijk verwijderd.

1.3) Voer stap 1.1 t/m 1.5 uit paragraaf 6 uit.

2) Het verwijderen van hele VOCABULARYs.

2.1) Daar de initiële waarde van VOC-LINK ook onthouden moet worden moet stap 2.1 uit paragraaf 6 ook uitgevoerd worden.

9. Literatuur:

- 1) Fridus Jonkman: Korrektie;
De 6502 kenner nr.34 pag. 48.
- 2) Fridus Jonkman: Using FORTH with the 6502

** DE 6502 KENNERS ** — A CLUB FOR ALL 65xx USERS

Information about the club.

Most people are not well-informed about the first micro-computer club in The Netherlands. It was called The KIM Users Club The Netherlands. Today, the name of the club is not quite covering all the other members with a computer from the 65xx-family. So, the club appears also with the name DE 6502 KENNERS, and offers accommodation to all users of the 65xx-computers, like APPLE, COMMODORE, BBC, ACORN, ATARI, OSI, ITT 2020, AIM-65, CHE-1, CV-777, BASIS 108, PEARCOM, SYM-1, PET, VIC-20, Proton-computers, ORIC-1, ACC 1000, SYSTEM 65, PC-100, PALLAS, MINTA, FORMOSA, STARLIGHT ESTATE III, SBC 65/68, MCS 6502, KEMPAC System 4, Elektor JUNIOR, Elektor SAMSON65, KIM-1, etc., etc.

International

The club has members in The Netherlands, Belgium, Germany, France, Spain, Portugal, America, India, Zambia, etc. etc.

Purpose

The club has been established 29th of January 1977 with the purpose to promote the exchange of knowledge among the users of systems mentioned before.

Meetings

In The Netherlands the club organises meetings each third saturday of the odd months, except in July.

The club delivers lectures of all kinds, about programming machine language, Basic, Forth, etc. etc.

Cassette Libr. The cassette library is free available for all members by ordering cassettes. Programs on cassette has been developed by members of the club, mostly published in our club magazin DE 6502 KENNER. All rights of the authors are reserved.

Paperware

The paperware service exists of providing source-listings and articles published in our club magazin DE 6502 KENNER. There are also listings in English.

Magazine

The magazin DE 6502 KENNER appears 5 times a year. The articles are written by members of the club from all countries, aided by the redaction. Articles of all kind, hardware as well as source-listings in assembly with comments, Basic, Forth, Comal, Focal, Logo, Elan, Pascal, Cobol, Fortran, C, etc. etc. Used assemblers: Micro-ADE, ASS/TED, Proton assembler, FATE, Big Mac, and so on. Each edition counts 48 pages at least. Some source-listings are too long to be published, like a 12K FATE. They will be available in the paperware-service. An independent jury awards prizes for some contributions.

Membership

Membership is open for all natural persons. Both for novice and advanced. Subscribe on Eurocheque HFL. 45,- and send to: Editorial Office DE 6502 KENNER
c/o Willem L. van Pelt
Jacob Jordaensstraat 15
2923 CK Krimpen a.d. IJssel
The Netherlands.
If not paying with Eurocheque: HFL. 52.50 !

Paperware-service

Micro-ADE 8K Assembler/Disassembler/Editor.
For use with any 6502 microcomputer.
Micro-ADE was developed by Peter Jermind of Micro-Ware Ltd Canada, (C) 1977. (C) 1982 by KIM Users Club.

FATE 12K Format Lister/Assembler/Tape-utilities/Editor.
For use with Elektor's JUNIOR-computer with PM + TM. Also for use with KIM-1 computer (separate I/O patches). We now try to make a FATE working on APPLE-computer. FATE was developed by our member Rob Banen, based on object information and published with permission of Proton Electr. Naarden, The Netherlands. (C) 1984. We recommend free memory spaces of at least 16K, better 32K

FATEC 12K Format Lister/Assembler/Tape-utilities/Editor.
Same FATE, but now for use with 65C02-processor.

ASM65C CrossAssembler for use with 65C02-processor.
Developed by Rob Banen for Elektor's JUNIOR-computer with Proton Disk Operating System (Senior).

CAT 8K two pass Club Assembler Texteditor for C-64.
Developed by our member Ruud H. Uhoff. Assembler with IF-THEN-ELSE, REPEAT-UNTIL, WHILE-DO, CASE-WHEN-OTHERWISE, MOVE, COPY, DELETE, RENUMBER, CHANGE, etc. With build-in structuremacro's and controlstructures. English manual not available yet.

FORTH-79 STANDARD
fig-FORTH 6502 Assembly Source Listing
fig-FORTH Instalation Manual with Glossary, Model, Editor
FORTH 79 STADARD Conversion

fig-FORTH for JUNIOR with Proton Disk Operating System. Developed by our member Gert van Oobroek.

FYS-FORTH for APPLE II
2nd FYS-FORTH VSN 0.3 Release of July 4th. 1984.

Disassembler for the Rockwel 65C02.
Developed by Rob Banen for JUNIOR with Proton DOS.

CFAST5 Controlling digital recorder 6000 baud.
Dutch version source-listing developed by our member Koen van Nieuwenhove for JUNIOR-computer with PM + TM.

JUNIOR on 5" Floppy.
Dutch manual and English source-listing developed by Koen van Nieuwenhove, with scheme, Byte-articles and other informations. Ask for prices.

Floppy disk on AIM-65
Article published in DE 6502 KENNER nr. 28 Oct. 1983. Dutch version, with scheme.
Author: D.A. Gerritsen.

Read basicode with your Microsoft SYM-1 Basic on Elektor's JUNIOR-computer with PM + TM, and Elektor's interface.

Read basicode with your Microsoft KB9-P Basic on Elektor's JUNIOR-computer with PM + TM, and Elektor's interface.

* Send your program listings to the editors office *
* c/o Willem L. van Pelt *
* Jacob Jordaensstraat 15 *
* 2923 CK Krimpen a.d. IJssel *
* The Netherlands *
