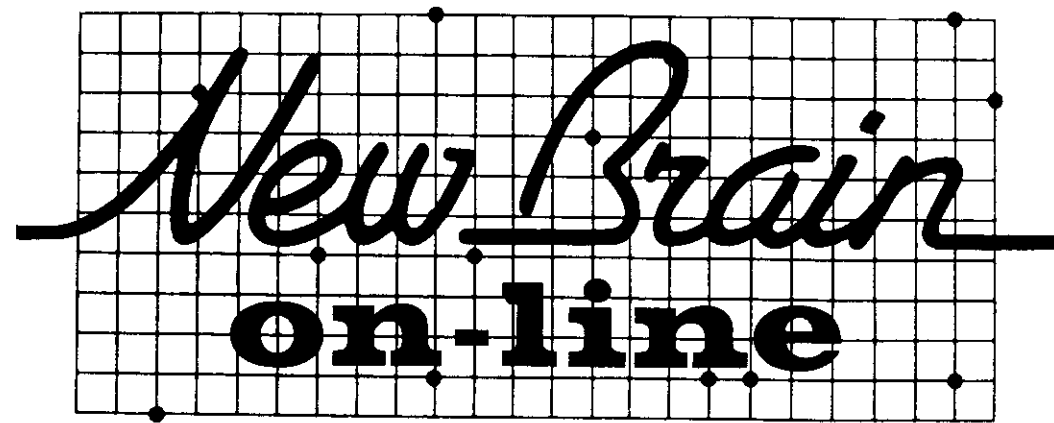


frankeren
als briefkaart

NewBrain-
gebruikersgroep
postbus 4494
1009 AL amsterdam

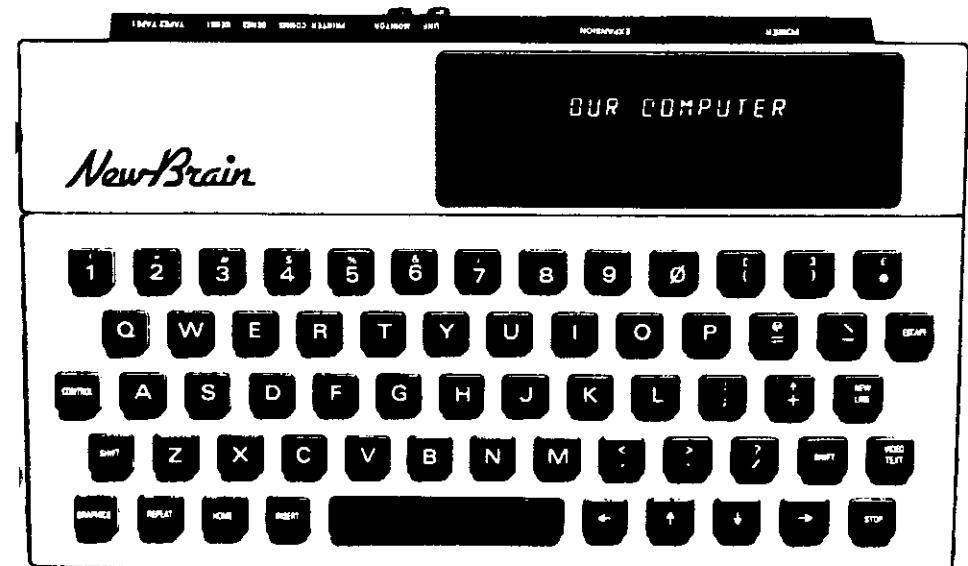
frankeren
als briefkaart

NewBrain-
gebruikersgroep
postbus 4494
1009 AL amsterdam



uitgave van de
NewBrain-gebruikersgroep
juni 1986

7



newbrain on-line is een officiële uitgave van de nederlandse newbrain-gebruikersgroep. de bladen zijn licht gelijmd zodat ze eenvoudig los te maken en daarna in een ringband op te bergen zijn. de artikelen kunnen dan bij voorbeeld op onderwerp geordend worden. de perforatie is afgestemd op de ringband met hcc-opdruk, die bij de hcc te koop is.

geplaatste artikelen mogen alleen voor niet-commerciële doeleinden, onder bronvermelding, overgenomen worden.

het is voor de redactie nu eenmaal onmogelijk om alle ingezonden artikelen en programma's op originaliteit te controleren. de aansprakelijkheid voor de ingezonden stukken ligt derhalve dan ook bij de inzender.

voor informatie omtrent commerciële advertenties of advertenties van leden, kunt u contact opnemen met de redactie.

door de aanmeldingskaart achterop het omslag op te sturen kunt u zich verzekeren van de toezending van nieuw verschenen nummers van on-line



kopij opsturen naar het adres van de redactie:

newbrain-gebruikersgroep
postbus 4494
1009 AL amsterdam

newbrain on-line 8 zal verschijnen op de newbraindag op 18 oktober 1986. sluitingsdatum voor de kopij is 28 augustus 1986. maar stuur uw kopij zo spoedig mogelijk op: het werkt veel prettiger, als niet alles op het laatst gehaast gedaan moet worden.

het is voor de redactie verreweg het makkelijkst de kopij aangeleverd te krijgen op cassette of diskette. dat bespaart de fouten bij het overtypen. vermeld bij een diskette wel het formaat. de tekst mag op alle gangbare manieren zijn opgeslagen: txy-, wordstar-, wp- en alle mogelijke ascii-bestanden zijn bruikbaar. de cassette of diskette wordt natuurlijk teruggezonden.

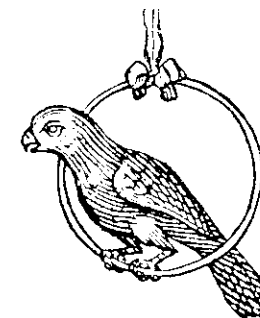
zeker als in een artikel een programma-listing van enige omvang voorkomt, is een cassette of diskette onontbeerlijk.

dit alles mag u natuurlijk niet beletten om kopij in te sturen. de redactie ontvangt liever kopij, waar ze wat meer werk mee heeft, dan helemaal geen kopij

menno fecit

New Brain on-line

TEN GELEIDE



je zou kunnen denken, dat we na zo vele jaren de newbrain inmiddels wel kennen. maar neen, er worden nog steeds nieuwe dingen gevonden en bedacht. de wondere wereld van de z80-machinetaal wordt meer en meer bezocht; de aansluiting met basic is gemakkelijk, en het operating system van de newbrain is direct beschikbaar via de zealls.

dit nummer van newbrain on-line is daar getuige van, en ook voor de toekomst belooft het wat. nu moet u niet denken, dat alleen spitsvondige ontdekkingen een artikel in on-line waard zijn. ook de ervaringen van beginnende newbrain-gebruikers (die zijn er steeds opnieuw!) zullen wij hier graag afdrucken

wim luijt is aangetreden als softwarebibliothecaris. dat heeft niet alleen een serie artikelen voor deze on-line opgeleverd, maar ook vijf nieuwe cassettes in de bibliotheek. een daarvan (ONLINE-7) bevat de programma's uit dit krantje

tot slot herinneren wij u aan de oproep uit on-line 6 om vragen over de newbrain, basic, newbrain-basic, cp/m enzovoort op te sturen, die dan in het komende nummer van newbrain on-line behandeld zullen worden. als u uw vragen snel instuurt, kunnen we nog op tijd deskundigen vinden om ze te beantwoorden

maar vermaakt u zich intussen met uw newbrain en deze newbrain on-line

de redactie

newbrain on-line 7

NewBrain-gebruikersgroep postbus 4494 1009 AL amsterdam

voorzitter	m s vreedenburg	(02159) 11068
secretaris	menno stevens	(020) 924137
penningmeester	jan de vries	(030) 511243
		/19.00 - 20.00 uur/

postrekening 2505800 tnv hcc newbrain-gebruikersgroep, utrecht

de newbrain-gebruikersgroep is een onderdeel van de

hcc HOBBY COMPUTER CLUB
inschrijvingsnummer kvk leiden: V445230

van de voorzitter

Rond de afgelopen jaarwisseling heeft Bert Hovius het voorzitterschap van de NewBrain-gebruikersgroep beëindigd. Het was voor hem niet meer mogelijk om voor de gebruikersgroep voldoende tijd vrij te maken. Voor zijn werk is hij veel van huis. Tijdens de NewBraindag in januari is een klein beetje afscheid genomen van Bert. Dat wil zeggen wel een afscheid als voorzitter, maar geen afscheid als actief lid van de gebruikersgroep. Ook hier nogmaals: Bert, veel dank voor je inzet voor de gebruikersgroep, eerst als penningmeester en later als voorzitter.

Er gebeurt meer in een vereniging. Een aantal leden is aangetrokken door de sterke opkomst van de PC. Voor een deel is men toch lid gebleven van

de NewBrain-gebruikersgroep. In de HCC kan men immers van maximaal zes gebruikersgroepen lid zijn, mits dit bij de nieuwe opgave wordt aangegeven. Deze leden steunen de gebruikersgroep in het ledental. Een geheel andere ontwikkeling is het instellen van de 'werkgroep Proton'. Een gevolg van een technische ontwikkeling: NewBrain-gebruikers vonden een goedkoop alternatief voor de floppy disk controller en expansion interface in het Proton-compact board met de standaard-NewBrain als toetsenbord. Het goedwerkende toetsenbord van de NewBrain en het snelle Proton-compact board geven een zeer snelle CP/M-combinatie. Om voor de Proton-leden onderlinge contacten eenvoudig te houden is de vorm van een werkgroep gekozen. Dit brengt een versterking met 40 leden mee.

De NewBrain-gebruikersgroep beschikt op dit moment over ruim 180 volumes van de public domain CP/M-bibliotheek. Tegen vergoeding van te maken kosten is deze software voor alle leden van de NewBrain-gebruikersgroep (en dus ook de Proton-gebruikers) beschikbaar.

De NewBrain-basic-softwarebibliotheek is eens danig onder handen genomen. Wim Luijt heeft er voor gezorgd, dat in de cassettesoftware zo duidelijk mogelijke aanwijzingen staan voor bij voorbeeld NewBrain met 32- of 96K. Hopelijk heeft u nu geen problemen met het op de juiste wijze inladen van de software. Aan de softwarebibliotheek zijn verschillende



nieuwe delen toegevoegd. Een paar bladzijden verder in deze NewBrain On-line staan de aanwinsten beschreven.

Hardware-nieuws is er ook te melden. Rob Maris en Theo Kamperman verzorgen ieder een eenheid die met de buitenwereld kan werken. Theo brengt een RS232-eenheid voor de unexpanded NewBrain. Hiermee is onder andere de toegang tot het Fido-net mogelijk.

Rob heeft een prototype van een eenheid voor RS232- en Centronics-aanpassing met extra de mogelijkheid voor het programmeren van EPROM's, inclusief de benodigde software. Er staat nog meer op stapel.

U ziet het, er is weer volop te beleven in de NewBrain-gebruikersgroep.

M. S. Vreedenburg, voorzitter



landelijke newbraindagen

coördinatie: m s vreedenburg, (02159) 11068

de volgende newbraindag is op zaterdag 18 oktober 1986
in technische school 'de bron' utrecht. de leden van de gebruikersgroep
ontvangen tevoren een uitnodiging met het programma

newbrain on-line

redactie: menno stevens, (020) 924137

sluitingsdatum voor de kopij voor on-line 8 is 28 augustus 1986

zie de binnenkant van het omslag voor het inzenden van kopij

softwarebibliotheek

bibliothecaris: wim luijt, (01100) 28197 /wo-za, ook overdag/

uitlevering op cassette: guus von morgen, (02158) 3381

uitlevering op diskette: jos hermans, (013) 552530 /ma-vr, 19.00 - 21.00 h/

zie bladzijde 6 voor de bestelprocedure en bladzijde 17 voor de nieuwe
cassettes. achterin deze on-line is een bestelformulier afgedrukt

cp/m

bibliothecaris: jos hermans, (013) 552530 /ma-vr, 19.00 - 21.00 h/

vraagbaak: henk blik, (02159) 42689 /di-, do-avond/

zie pag 6 voor de bestelprocedure en 20 voor een overzicht van specifieke
newbrain-cp/m-software. achter in deze on-line staat een bestelformulier

onderwijsbibliotheek en werkgroep onderwijs

pieter van dijk, (01751) 12367

maarten floor, (02963) 4374

zie bladzijde 6 voor de bestelprocedure en bladzijde 23 voor schijf 6

werkgroep proton

coördinator: jan wubben, (010) 4557698

bemiddeling verkoop tweedehands apparatuur

jan de vries, (030) 511243 /19.00 - 21.00 h/

internationale contacten

secretaris buitenland: rob van albeda, (020) 931191

werkgroep hardware

rob maris, (055) 424485

henk dekker, (076) 713390

henk blik, (02159) 42689 /di-, do-avond/

contactpersonen regionale bijeenkomsten

groningen: evert drijver, (050) 778415

deventer: r e van gelden, (05720) 3016

utrecht: m s vreedenburg, (02159) 11068

zeeland: wim luijt, (01100) 28197 /wo-za, ook overdag/

brabant en limburg: jos hermans, (013) 552530 /ma-vr, 19.00 - 21.00 h/



bibliotheken

de BESTELPROCEDURE is dezelfde voor de drie bibliotheken: vul uw keuze van de software in op de bon achterin deze on-line, en kruis het gewenste formaat aan: (cassette of) diskette van 200k, 400k ss, 400k ds of 800k. stuur de bon samen met een girobetaalkaart of betaal- of eurocheque (vergeet niet het nummer van giro- of betaalpas in te vullen) op naar:

newbrain-gebruikersgroep
postbus 4494
1009 AL amsterdam

het verschuldigde bedrag kan ook voldaan worden door overmaking ten laste van uw post- of bankrekening naar de nieuwe girorekening van de gebruikersgroep:

postrekening 2505800
t.n.v. hcc newbrain-gebruikersgroep, utrecht

PRIJSLIJST			
prijs per deel	cassette	disk 200k	disk 400k ss/ds, 800 k
softwarebibliotheek	f 15,00	f 15,00	f 15,00
onderwijsbibliotheek	f 25,00	f 25,00	f 25,00
cp/m-bibliotheek *)		f 25,00	f 15,00
*) per 10 delen in één bestelling 1 deel gratis			

indien u cheques bijsluit, en uw pasnummer op cheques en bestelformulier vermeldt, kan direct bij ontvangst van uw bestelling tot verzending worden overgegaan, als de gewenste cassettes / diskettes voorradig zijn. anders wordt gewacht, totdat de betaling binnengekomen is.



nieuws

NIEUWS VAN DE SOFTWAREBIBLIOTHEEK

De kwaliteit van de programma's, die aan de softwarebibliotheek worden aangeboden neemt nog steeds toe. Om ook alle aangeboden programma's op cassette en diskette te kunnen blijven verspreiden, moeten soms gedeelten van het programma in hulpbestanden worden ondergebracht. Dit zal op een zo systematisch mogelijke manier gebeuren. Hoe dat eruit gaat zien en welke hulpbestanden kunnen voorkomen, wordt in dit artikel uitgelegd.

Voortaan zal dikwijls met zogenaamde overlays worden gewerkt. Een overlay is een stuk (basic-) programma, dat bij een ander programma hoort. Door de overlay te MERGEN met het bijbehorende programma krijgt men een nieuw programma. Het zijn dus NOOIT zelfstandige programma's.

De cassettes en diskettes met programma's worden met de uiterste zorg geproduceerd. Toch komen er nog regelmatig klachten van gebruikers, die problemen hebben met het LOADen van programma's uit de softwarebibliotheek, speciaal van cassette. Als het niet lukt om de programma's te laden, wordt het meestal aan de kwaliteit van de cassettes geweten. Bij het opsporen van de fout blijkt echter dikwijls, dat die bij een slechte cassetterecorder of bij de gebruiker ligt. Zeker, er kunnen fouten worden gemaakt bij de productie van cassettes en diskettes, en de softwarebibliotheek is altijd bereid om gemaakte fouten te herstellen, maar u kunt uzelf waarschijnlijk veel ergernis en frustratie besparen, als u de volgende adviezen opvolgt. Elders in dit nummer worden de fouten behandeld, die bij de in- en uitvoer van een cassetterecorder kunnen optreden.

Hoe goed een cassetterecorder ook is afgesteld, de beste weergave krijgt men toch van een cassette, die met DEZELFDE recorder is opgenomen. Mocht u regelmatig problemen hebben met het LOADen van programma's uit de softwarebibliotheek, dan adviseer ik u de programma's naar een nieuwe, blanco cassette over te zetten en de ontvangen cassette geheel teruggespoeld als zogenaamde BACK-UP te bewaren.

Om productietechnische redenen werden de basic-programma's in blokken van 5k op de cassette gezet samen met het advies om ze met LOAD "*20" te laden.

Voortaan is het ook mogelijk om een basic-programma vanaf cassette te laten met de programmanaam. Deze moet echter wel worden voorafgegaan door '*20:'. Bij voorbeeld: om het programma PROTO.BAS te kunnen laden hebt u nu de keuze tussen de opdrachten:

- LOAD "*20:PROTO.BAS", en uw NewBrain zoekt net zolang, totdat hij op de cassette PROTO.BAS tegenkomt. De naam van het programma moet in HOOFDLETTERS.
- LOAD "*20", en uw NewBrain laadt het eerste programma, dat hij op de cassette tegenkomt. In dat geval moet u dus eerst de cassette zover spoelen, dat PROTO.BAS als eerstvolgend programma staat.

Alle .BAS-programma's uit de softwarebibliotheek, die vroeger verkocht zijn, kunnen alleen worden geladen met: LOAD "*20"

U bent natuurlijk geheel vrij om uw eigen kopie met een andere blok-grootte te SAVE en met een andere naam, zodat u daarna weer kunt laden, zoals u gewend bent.



Nieuwe cassettes van de softwarebibliotheek kunt u herkennen aan het eerste programma op de cassette. Dit programma heeft als naam de titel van de cassette en is een uitgebreide inhoudsopgave met

- 1 de titel van de opgenomen programma's en bestanden
- 2 de blok-grootte, als die anders is dan 1k (*4)
- 3 geschikt voor expanded, c.q. unexpanded NewBrain
- 4 eventuele aanwijzingen

Met LOAD en RUN wordt deze inhoud in de vorm van een listing op het scherm afgedrukt. Met LIST #pr kunt u hem ook op een printer afdrucken.

Er wordt naar gestreefd om ALLE programma's op ALLE modellen NewBrain (unexpanded of 32k, expanded of 96k, met cassettereclorder, met diskdrive(s), met of zonder VF-display (AD of A)) te laten draaien. Dit is niet altijd mogelijk. Oorzaken hiervoor zijn onder andere

Behalve het deel ONLINE-7 met de programma's uit On-line 7 zijn er nog 4 nieuwe delen verschenen, namelijk:

- NBFILES: de listings uit het boek van Philip Crookes, The NewBrain Files. Van de auteur ontvingen we de toestemming om alle listings en de drie programma's in de softwarebibliotheek op te nemen
- SPEL-6: voor ontspanning naast het serieuze gebruik van de NewBrain
- HULP-2: een aantal programma's speciaal voor de programmeurs in Z80-machinecode, en utility's voor verschillende printers
- DIVERS-2: onder andere tekenfilms maken op de NewBrain, en een zeer uitzonderlijk stratego-achtig spel voor grote computers nu ook op een 32k-NewBrain!

Verder zijn er nog 4 delen in bewerking:

- een aantal programma's over besluitvorming ('management decision')
- een subroutinelibrary met wiskunderoutines
- een aantal programma's speciaal voor disksystemen, onder andere programma's om onder BASIC een aantal directorybewerkingen te doen. In dit deel is nog plaats voor meer programma's, dus zend je programma in, dat diskdrives gebruikt
- een deel voor het programmeren in Z80-machinetaal, ook op een 32k-NewBrain met recorder, onder andere met een editor, een assembler (nog niet zeker) en een aantal sourcecodebestanden. Ook hierin is nog plaats voor sourceprogramma's. Als je dus specifieke Z80-routines voor de NewBrain hebt, stuur ze in.

Wat verder er verder nog uitgegeven kan worden hangt af van jullie inzendingen. Het adres is bekend: softwarebibliotheek NewBrain-gebruikersgroep, Postbus 4494, 1009 AL Amsterdam.

- het gebruik van devices, die niet op ieder model aanwezig zijn (device 12 tot 33)
- gebruik van het schermgeheugen
- operating system calls in machinetaalprogramma's
- gebrek aan geheugenruimte

Indien een beperking bekend is, wordt die duidelijk aangegeven in de inhoudsopgave en in het programma.

Programma's, die een of twee diskdrives vereisen, zullen bij voldoende aanbod voortaan op aparte diskettes worden uitgegeven. Slechts bij uitzondering zullen deze programma's op cassette verkrijgbaar zijn.

Programma's, die in- en uitvoer van cassette of diskette vereisen, zullen, indien mogelijk, worden aangepast door de redactie, zodat ze voor beide media geschikt zijn. Is dit onmogelijk, bij voorbeeld omdat het karakter van het programma te veel zou worden veranderd, of omdat het programma oprachten bevat, die wel met cassetterecorder maar niet met diskdrive (of omgekeerd) mogelijk zijn, dan zal zowel een cassetteversie als een disketteversie worden uitgegeven. Hiervoor is een gebruikersvriendelijke vorm gevonden, die zowel op cassette als op diskette komt te staan.

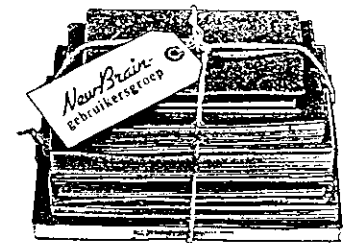
Toelichting. Het programma XXX.BAS is de cassetteversie. Tevens staat op diskette én cassette XXX.DSK. Bij gebruikers, die met diskdrives werken, zal bij het RUNnen van het programma automatisch de overlay (XXX.DSK) worden geMERGED, waardoor het programma wordt aangepast aan diskdrives. Het programma gaat dan normaal verder. Als zij dat wensen, dan kunnen zij een aan diskdrive aangepaste versie maken door XXX.BAS te MERGEN met XXX.DSK en deze te SAVFen met een zelfgekozen naam. Dit programma zal dan sneller LOADen en starten. Gebruikers met een cassetterecorder behoeven bij een eventuele aanschaf van diskdrives niet opnieuw de diskette met deze programma's aan te schaffen, maar kunnen na overzetting van cassette op diskette (bij voorbeeld met een van de vier kopieerprogramma's uit de softwarebibliotheek) deze blijven gebruiken. Ook het formaat van de bestanden is in beide versies gelijk, zodat die ook zonder meer kunnen worden overgezet.

Programma's in zogenaamde machinetaal worden ook zoveel mogelijk als een basic-programma behandeld. De machinecoderoutine wordt hierbij als DATA-regel geladen en met READ en POKE op zijn plaats gezet. Soms kan het starten van een programma hierdoor nodeloos lang duren. Om dit te versnellen, of als er andere dringende redenen zijn, zullen er ook programma's komen in de vorm van een DATA-bestand. Deze bestanden zijn door de extensie .DAT gekenmerkt. Ze kunnen niet apart geLOAD worden. U start het basic-programma van dezelfde naam, en dat zorgt ervoor, dat de DATA gelezen worden. Op de cassette is de volgorde YYY.BAS -YYY.DAT. Zorgt u ervoor, dat deze volgorde bij het kopiëren niet verandert, anders gaat het mis.

Ieder programma is voorzien van een handleiding of instructies. De handleidingen worden ook steeds groter. Opname van de handleidingen in de programma's zelf heeft als nadeel, dat men ook steeds de handleiding moet laden, wat extra tijd vergt en de kans op inleesfouten vergroot. Instructies of een handleiding kunnen op drie verschillende wijzen zijn toegevoegd:

- 1 de instructies vormen een onderdeel van het programma en worden tijdens het RUNnen van het programma getoond
- 2 de handleiding is als REMs aan het programma toegevoegd. Korte instructies of een verwijzing staan dan in de regels 5 - 9. Ze zijn te lezen door het programma te LISTen, eventueel op een printer
- 3 bij een lange handleiding staat de handleiding in een apart bestand met .HAN als extension. De handleiding staat dan in het formaat van device 13, en is onder cp/m te lezen met TYPE of met cassetterecorder met het programma DEV13.BAS

De inzenders van programma's zijn geheel vrij in de wijze, waarop ze de handleiding etc. bij hun programma's toevoegen. Bij het gereed maken zullen de programma's, indien nodig, worden aangepast. Inzenders van software, die een diskdrive bezitten, wordt daarom verzocht om een lange handleiding in het formaat van device 13 in te zenden. Degenen, die geen diskdrive bezitten, wordt verzocht om de handleiding als REMs in een AF-ZONDERLIJK programma in te zenden (vergeet de regelnummers niet!).



Door het groter worden van de programma's gaat niet alleen het LOADen langer duren, maar komen er ook problemen met de beschikbare geheugenruimte. Een van de meest effectieve manieren om een programma niet nodeloos lang te maken, is om de REMs weg te laten. Om zo'n programma toch leesbaar te houden voor gebruikers, die het programma willen bestuderen, zullen de REMs in een afzonderlijke overlay worden verstrekt. Deze overlays krijgen .REM als extension. Na het MERGEN staan de REMs weer op hun normale plaats en kan het programma bestudeerd worden. Voor het normaal RUNnen van het programma zijn deze REMs niet nodig.

Boven wordt geadviseerd om de ontvangen cassette zo snel mogelijk te kopiëren. Bestanden met de extension .BAS, .OVL, .DSK en .REM kunnen eenvoudig gekopieerd worden met LOAD "*"20" en daarna SAVE of SAVE "*"20", of met COPYCOPY.BAS. Andere bestanden kunnen gekopieerd worden met COPYCOPY.BAS, of als u meer dan één recorder hebt, met een van de andere kopieerprogramma's uit de softwarebibliotheek. COPYCOPY.BAS staat, indien nodig, op iedere cassette. Diskdrivebezitters kunnen deze bestanden kopiëren onder cp/m met FCOPY.COM, dat op iedere diskette komt te staan.

Nieuwe delen uit de softwarebibliotheek kunnen dus bestanden met de volgende extensions bevatten:

- .BAS basic-programma's
- .REM afzonderlijke REMs bij het gelijknamige programma
- .DSK overlay om van een cassetteversie automatisch een diskversie te maken
- .COR overlay om het gelijknamige basic-programma te verbeteren, zie pagina SS in deze On-line
- .OVL anderssoortige overlay, die het bijbehorende basic-programma verandert of aanvult
- .DAT machinetaalprogramma, dat geladen wordt door het gelijknamige basic-programma, of DATA-bestand
- .ASM sourcecodelisting van een machinetaalprogramma. Leesbaar als files van het formaat van device 13, of met DEV13.BAS
- .HAN handleiding, leesbaar als bij .ASM
- .TXT ander tekstbestand, leesbaar als bij .ASM

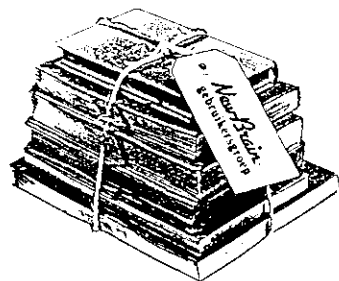
Het is dus iets ingewikkelder geworden, maar met bovenstaande uitleg hoop ik, dat de veranderingen ook voor de onervaren gebruiker doorzichtig zijn. Door deze veranderingen worden de mogelijkheden om nog krachtiger programma's uit te geven enorm vergroot. Laat dit u echter niet weerhouden om ook eenvoudige programma's in te sturen. Bedenk, dat er alleen nieuwe delen kunnen blijven verschijnen, als er software wordt ingezonden.

Als beloning voor een opgenomen programma ontvangt de inzender, afhankelijk van de originaliteit en de grootte van het programma, een tot vijf volumes uit de softwarebibliotheek (dat mogen ook cp/m-volumes zijn).

De redactie ontvangt graag niet alleen nieuwe programma's, maar als u in een bestaand programma een BUG ontdekt, of als u andere klachten hebt, hoort zij dit ook graag. De softwarebibliotheek is er DOOR de gebruikers en VOOR de gebruikers!

Wim Luijt

telefoon (01100) 28197
woensdag tot en met zaterdag
(ook overdag)



correctie

HERZIENE UITGAVEN VAN DE SOFTWAREBIBLIOTHEEK



Op de cassette HULP-1 stonden een aantal programma's, die niet foutloos werkten of door een of andere oorzaak van slechte kwaliteit waren.

Deze cassette is kort geleden geheel herzien en de desbetreffende programma's zijn verbeterd. Gebruikers, die de cassette in het verleden hebben gekocht, kunnen deze programma's zelf wijzigen door de onderstaande regels toe te voegen of te overschrijven. Of ze kunnen de overlayprogramma's van de cassette ONLINE-7 MERGEN en de programma's dan opnieuw SAVEN.

De overlays met de bijbehorende programma's zijn:

MONITOR.COR voor MONITOR.BAS

```

100 FOR i = 1 TO 255 : CLOSE #i : NEXT i : OPEN #1, 0, 1, "5" : PUT
31 : LINPUT ("Heb je NewBrain Model A (j/n) ") h$: IF INSTR ("JJ",
h$) ? #1, CHR$ (13); "10110 open #vf, 0, 2, ""1"" : put #vf, 23, 1" :
PUT #1, 27, 4, 3 : MERGE #1
5600 REM
9999 DELETE -9999
10000 ON BREAK GOTO 60000 : ON ERROR GOTO 60010
10100 FOR i = 1 TO 255 : CLOSE #i : NEXT i : kb = 12 : OPEN #kb, 6 : vf
= 2 : PUT #kb, 1
10101 REM
10110 OPEN #vf, 3, "16"
10120 sn = 0 : OPEN #sn, 0, "20" : PUT #vf, 6 : tp = 50 : FOR i = 1 TO 11 :
n1$ = n1$ + CHR$ (0) : NEXT i : b1$ = CHR$ (28)
30000 hx$ = "0123456789ABCDEF" : e = 0 : dc = 0 : pp = LEN (h$) : cp = -1
31000 hx$ = "0123456789ABCDEF"

```

Gebruiksaanwijzing

MONITOR.BAS is een zeer eenvoudig programma, waarmee je de inhoud van de geheugenadressen van de NewBrain kunt:

- lezen
- veranderen
- op cassette SAVEn
- eventueel machinecodeprogramma's schrijven

Het programma is niet geschikt voor systemen met diskdrive.

Wanneer het scherm leeg is (of met een witte achtergrond bij een NewBrain model A), zijn slechts de volgende eentoetsopdrachten mogelijk:

0 t/m 9 en A t/m F: hexdecimaal getal
 cursor omhoog: ga naar het volgende adres
 cursor omlaag: ga naar het vorige adres
 control/A: spring naar het in te typen adres
 control/C: maak een nieuwe file ('create')
 control/O: uitvoer van de gecreëerde file naar cassette ('output')
 control/F: zoek een op te geven file ('find')
 control/I: invoer van de opgegeven file van cassette

LET OP! Tijdens de uitvoering van de laatste vier opdrachten is het adres in het line display niet gelijk aan het begin- of eindadres.
 Er kunnen alleen hexadecimale getallen ingevoerd worden.



Het programma houdt geen rekening met de overgang 7FFFh-A000h bij de unexpanded NewBrain zonder diskdrive.

BASICLEE.COR voor BASICLEE.BAS

Van dit Basiccode2-leesprogramma zijn verschillende versies verspreid. Sommige kunnen zelfs helemaal niet werken! Wanhoop dus niet, als het tot nu toe niet gelukt is om een programma te vertalen. Een overlay, die op alle versies werkt, is niet te maken. De onderstaande overlay zal met de meeste een bruikbaar programma opleveren. Mocht het daarna nog niet werken, neem dan contact op met de softwarebibliotheek. Stuur een listing of de originele cassette op!

Na de toevoeging van deze overlay moet u de subroutines (BASICSUB.BAS) nog toevoegen.

```

9 DELETE 30000- : DELETE -360
1060 RESERVE TOP - en

```

BASICSUB.COR voor BASICSUB.BAS

```

305 IF sr < 0 THEN sr$ = "-" + sr$ : RET

```



COPY-D/C.COR voor COPY-D/C.BAS

```

65005 OPEN #0, 0 : POKE 29, 13 : RESERVE 22000 : PUT 23, 68, 6
65038 REM
65041 OPENOUT #12, 1, S$ (I)
65049 REM

```

Met deze overlay kunt u van diskette naar cassette kopiëren.

65005 OPEN #0, 0 : POKE 29, 13 : RESERVE 22000 : PUT 23, 68, 6
 65038 GOSUB 65056
 65041 OPENOUT #12, 12, S\$ (I)
 65049 REM

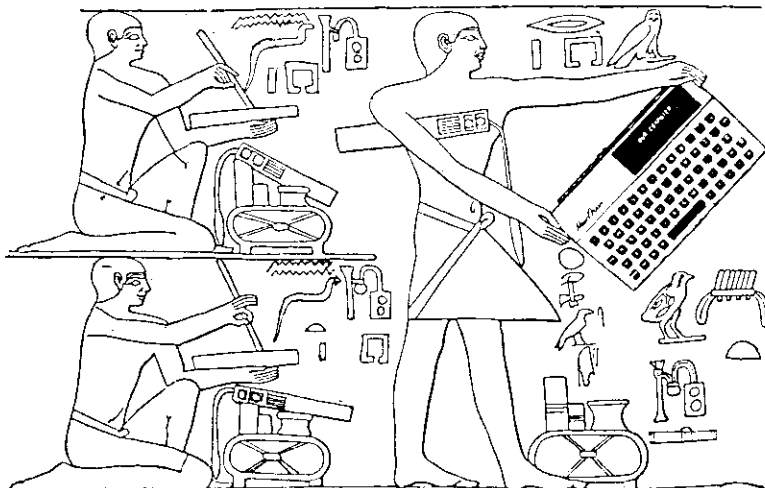
Als u maar één diskdrive hebt, kunt u met deze overlay afzonderlijke files van de ene disk naar de andere disk overzetten! Dus geen ongemak meer met SCOPY.COM

COPY-D/C.BAS en beide overlays werken alleen op een 32k-NewBrain met diskdrives.

Hebt u ook voor andere programma's uit de softwarebibliotheek kleine wijzigingen, die op deze wijze kunnen worden verspreid, aarzel dan niet om ze op te sturen aan de softwarebibliotheek.

verkrijgbaar op cassette
 zie pagina 18

Wim Luijt



catalogus softwarebibliotheek

de softwarebibliotheek van de newbrain-gebruikersgroep is dank zij de inspanningen van onze nieuwe bibliothecaris uitgebreid met vijf nieuwe delen, als aanvulling op de catalogus, zoals die in newbrain on-line 6 is gepubliceerd, vermelden we hier de inhoud van de nieuwe delen. de volledige catalogus is te verkrijgen door f 1,10 in postzegels op te sturen naar

newbrain-gebruikersgroep
 postbus 4494
 1009 AL amsterdam

bestelnummer: DIVERS-2

- DEV13.BAS leest device-13-bestanden vanaf cassette (.HAN-bestanden)
- MOTOR-2T.BAS laat op grafische schermen een 2-taktmotor (vast)lopen
- MOOI.BAS, mooie plaatjes op een grafisch scherm
- ANIMATOR.BAS, tekenfilmprogramma: teken op een standaardscherm beelden van 80 x 72 pixels en wissel die daarna snel om bewegende beelden te krijgen, bijbehorende documentatie: ANIMATOR.REM, ANIMATOR.HAN, ANIMATOR.DSK
- AMPHIBION, strategiespel voor grote computers, bindt de strijd aan tegen de vijand; meer dan 1000 ronden; verschillende moeilijkheidsniveaus, inclusief AMPHTEXT.BAS (de menugestuurde handleiding); AMPHKAAR.BAS (maak eerst een kaart van amphibion); AMPHPEL.DSK (maakt een diskversie van AMPHPEL.BAS)

bestelnummer: HULP-2

- DEV13.BAS leest device-13-bestanden vanaf cassette (.HAN-bestanden)
- NEWMON.BAS, monitorprogramma voor de 32k-newbrain, onmisbare hulp voor z80-programmeurs, onder andere relocate, move, find; uitvoer naar printer of scherm. inclusief NEWMON.HAN en NEWMON.DAT (z80)

- CONT-MON.BAS, continu-monitor: bekijk voortdurend het inwendige van de 32k-Newbrain en de z80-registers. documentatie: CONT-MON.HAN (z80)
- Z80-DIS.BAS, z80-disassembler voor de newbrain. Z80INDEX.BAS maakt met behulp van Z80-DIS.BAS een index van z80-codes
- GREXDUMP.BAS, grafische dump voor 96k newbrain op onder andere Epson en Star; eenvoudig aan te passen aan andere printers (z80)
- SEIKOSHA.BAS, nu alle newbrain-tekens op een Seikosha gp-250 printer
- TAPEDISK.BAS zet 10 bestanden van cassette op disk en nummert ze
- UNPAGEF.BAS, snelle UNPAGE.BAS voor expanded systemen, ZONDER terugkeer naar 96k (z80)
- COPYCOPY.BAS, kopieerprogramma met één recorder voor .HAN-bestanden

bestelnummer: NBFILES

deze cassette bevat alle listings en de programma's
 PROGTRAN.NBF, RENUMBER.NBF en VITADATA.NBF uit:
 Philip Crookes, the newbrain files
 Bromley (Kent), Vitagraph Ltd, 1985, isbn 0-948152-00-1
 en is uitgegeven met instemming van de auteur

bestelnummer: ONLINE-7

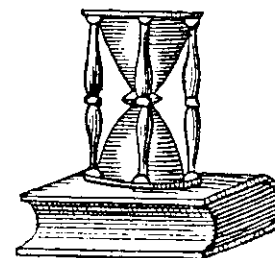
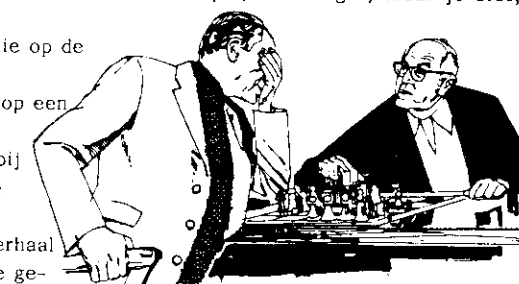
de programma's uit on-line 7 op cassette:

- TERMINAL.BAS, Theo Kampermans terminalprogramma voor het modem-interface van TradeCom (pagina 47)
- FP-DCONV.BAS zet decimale getallen om in het floating-pointformaat van de newbrain en omgekeerd (Wim Luijt; pagina 57)
- STERRENBEELDEN tekent naar keuze 45 sterrebeelden met hun helderheid
- de programma's van de prijsvraag (pagina 25):
 STER.BAS tekent de sterrenhemel op ieder tijdstip en op iedere plaats
 STERSNEL.OVL voor snellere uitvoering van STER.BAS, andere OVL's niet gelijktijdig mogelijk op 32k-newbrain
 STERMERC.OVL is als STER.BAS, maar in Mercatorprojectie
 STERHALL.OVL tekent de komeet Halley in STER.BAS, helaas zonder staart
 STERDAT.DAT, databestand bij STER.BAS; STER.REM, remarks bij STER.DAT en de inhoud van STER.DAT
- SSCC.BAS, Rob Maris' symbolic source code converter (pagina 35) met diskoverlay SSCC.DSK, testbestand TEST.BSS, en SSCCEDIT.BAS, een

- eenvoudige editor voor recordergebruikers
- LDCODE.OVL, Frans van der Veldens snellader voor machinecode-programma's (pagina 59)
- TEXY2MGI.OVL, M. Ploemens aanvulling op TradeComs 'texy2' om deelbestanden toe te voegen (pagina 51)
- correcties voor de programma's BASICLEE, BASICSUB, COPY-D/C, COPY-D/D en MONITOR van cassette HULP-1 (pagina 13)
- COPYCOPY.BAS voor het kopiëren van STERDAT.DAT en TEST.BSS

bestelnummer: SPEL-6

- PLUTO.BAS, raak de Pluto's
- ZEESLAG.BAS, het bekende plonsspel met echte speelvelden
- RALLY.BAS, dodenrace. (z80) alleen voor unexpanded newbrain
- NICOMA.BAS, rekenkundig spel. de newbrain raadt met 3 vragen elk getal onder de 100
- LEVEN1.BAS, een basic-versie van het spel; wat trager, maar je ziet, wat er gebeurt
- JAPLOG.BAS, variatie op de torens van Hanoi
- YAHTZEE1.BAS; nu op een grafisch scherm
- BRUTUS.BAS geeft bij het schaken aan, hoe wit kan winnen (z80)
- GEMRAADK.BAS: herhaal thuis of in de klas de gemeenteraadsverkiezingen



cp/m-bibliotheek catalogus

de cp/m-bibliotheek van de newbrain-gebruikersgroep heeft zo'n 180 'volumes' in voorraad uit de programmatheek van de hcc cp/m-gebruikersgroep. jos hermans heeft de inhoudsopgaven (met het commentaar) van al die delen bij elkaar gezet op een

catalogus-diskette

deze catalogus-diskette is tegen de speciale prijs van f 10,00 (voor elk formaat!) te bestellen, zie pagina 6 voor de bestelprocedure. hieronder treft u de catalogus aan van de delen met specifieke newbrain-cp/m-software

NBGG-1

- vreemde formaten
- INV, inverse karakters onder cp/m-highlight
- TOETS, gebruik van de newbrain-screeditor onder cp/m
- PTR schakelt de automatische linefeed na carriage return uit onder cp/m
- FAST, dit programma maakt de expanded newbrain twee keer (!) zo snel met de beeldschermuitvoer onder cp/m

INV, TOETS en PTR zijn in on-line 4 beschreven; FAST in on-line 5
deze 4 programma's werken alleen op een expanded newbrain

NBGG-2 en NBGG-3

geïnstalleerde ZCPR2-files (o.a. zelfstartende ZCPR2); 64k vereist!

NBGG-4

- MODEM7, aangepast voor de newbrain (incl. SET- en DOC-file)
- PIOS en SYSPAGE, programma's om onder cp/m devices te sluiten en opnieuw te definiëren en te openen, etc. (incl. PASCAL-file)
- PROMPT, maak onder cp/m gebruik van het line display van newbrain model AD (incl. ASM-file)
- GRDEMO en GRDEMO2, maak onder cp/m gebruik van high resolution (incl. PASCAL-files);
- SDMP, screendump onder cp/m
- DISKEDIT, programma om direct op disk te schrijven, lezen, veranderen; sectoren, tracks etc. (incl. PASCAL-file), werkt ook op unexpanded newbrain; de andere programma's van deze schijf alleen met expansion

nbgg-5

FASTPD.MAC
WSPATCH.MAC
FEXIT.COM

source van henk blik's fast
source van de wordstar-aanpassing uit newbrain on-line 6
nieuwe versie van EXIT.COM, die nu zowel unexpanded
als expanded, en dan zonder of met fast, werkt

FEXIT.ASM
FPTR.COM

de source van FEXIT.COM
PTR.COM, waarin in een kleine wijziging is aangebracht,
zodat het ook onder fast werkt, op een unexpanded
systeem stopt het programma met een mededeling

FPTR.ASM
PS.COM
BTTYPE.COM
QD.COM
PPM.COM
SEIKOSHA.COM

de source van FPTR.COM
full-screen-menu het instellen van de printer
list newbrain-basicprogramma's op het scherm
vervanging van diverse functies van PIP en TYPE
eenvoudige macro-preprocessor voor assembler
downloadt een aantal tekens van de 2e tekenset van de
newbrain naar een seikosha GP-250X

UNERA.COM
XDIR.COM
WC.COM
NSWP.COM

'tovert' per ongeluk gewiste files terug
uitgebreide versie van het DIR-commando
telt tekens, woorden en regels in een ASCII-tekstfile
nieuwste versie van SWEEP, deze versie kan nu ook
SQUEEZE en natuurlijk UNSQUEEZE

DIP.COM
MS2CPM.COM

leest 15 vreemde diskformaten op een expanded systeem
zet ascii-bestanden van een IBM-PC over naar expanded
newbrain

CONFIG.BAS

CONFIGUR.COM in een basic-uitvoering, configuratie
van uw newbrain-cp/m-systeem eenvoudiger dan ooit
sorteert de directory van uw diskette op alfabet
reproduceert uitgebreide diskettegegevens van uw sys-
teem op het scherm

SAP.COM
XDPB.COM

UNLOAD.COM

maakt van een COM-file weer een HEX-file, zodat het
mogelijk wordt om toch een COM-file 'over te PIPpen'
multi-file-transfer-programma voor twee diskdrives, dit

MFT.COM

programma kopieert niet domweg file voor file, maar
kopieert meteen meer files als het geheugen dit toelaat.
ideaal om kleine programma's te kopiëren

MFT1.COM

idem dito systemen met 1 diskdrive



full-screen-menu voor het instellen van een printer. selectie met de spatiebalk. het menu en de bijbehorende control codes staan in een .DTA-bestand, dat met elke tekstverwerker voor elke printer gemaakt kan worden. aan te roepen met

A> ps fx80 <CR>

het FX80-bestand is soms een beetje vreemd: wie wil nu alle opties buiten een programma instellen; maar het hele menuschema moest gevuld worden! het FX80A-bestand is identiek, behalve het eerste byte: dit bepaalt, of inverse video gebruikt kan worden. (turbo pascal)

LIST elk newbrain-basic-bestand, die met SAVF opgeslagen zijn, zijn een beetje onbegrijpelijk, omdat de regels in omgekeerde volgorde verschijnen! beeldt ook elk ander bestand af, dat geen linefeeds en geen tokens (ASCII 128 - 186) bevat. het programma breekt af bij de eerste de beste ↑ D (ASCII 4), en niet pas aan het einde van het bestand. dit programma heeft zijn nut bij het bekijken van alle public domain rommel, die over de noordzee komt aanwaaien, en de basic-procedure met LOAD en LIST (zonder pauzes!) kost me te veel moeite. zo hoeft u nooit uit uw comfortabele cp/m-stoel. (z80)

vervanging voor diverse functies van PIP en TYPE, vooral voor gebruik binnen programma's als wordstar. alle devices zijn beschikbaar voor in- en uitvoer; paginering; echo naar het scherm; meerdere bestanden in een opdracht met ? en *; tekst toevoegen aan een bestand, bij voorbeeld:

qd <file >lst: -e	bestand afdrukken met echo.
qd -m file1 file2 file3 >lst:	drie bestanden afdrukken
qd -m *.asm -p list	alle asm-bestanden met paginering
qd -a file	tekst van het toetsenbord aan het einde van het bestand toevoegen
qd >file	direct in een bestand typen
qd <CR>	toelichting (in het engels)

qd <file >pun: naar een modem is nog niet getest, maar het zou moeten werken, als de baudrates enz. goed ingesteld zijn met PIOS. (C/80)

n.b. de < en de > geven de richting van in- en uitvoer aan - geen spatie tussen zo'n teken en de bestandsnaam. de optie -m gebruikt het niet. dat is tevens de vlugste manier om een enkel bestand te bekijken.

onderwijsbibliotheek

de zesde schijf (en cassette) met onderwijssoftware is verschenen. hieronder worden alle programma's, die erop staan, kort omschreven. voor aanvullende inlichtingen kunt u zich wenden tot pieter van dijk, tel. (01750) 12367, en maarten floor, tel. (02963) 4374

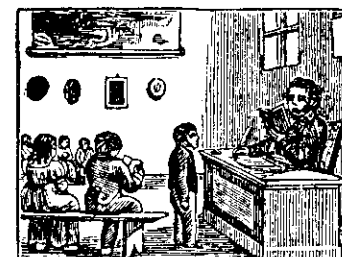
schijf 6

132. AARDR1: het overhoren van de plaatsnamen aan de hand van 11 provinciekaarten

133. WISK2: de computer tekent een kubus, kegel, cilinder of piramide. indien gewenst kan er daarna een doorsnede getekend worden met een vlak

134. TRAF01: naar keuze wordt het vermogen van de kern berekend of het aantal en de doorsnede van de wikkelingen. de resultaten kunnen geprint worden. werkt niet op een expanded system

135. BENOEM4: het programma kent 34 zinnen, die benoemd moeten worden. er is een optie om zinnen toe te voegen of te verwijderen. het programma werkt niet op een expanded system.



136. ZENER: het berekenen van een zenerdiode aan de hand van een schema

137. AARDR2: aan de hand van een kaart worden de plaatsnamen van de provincie noord-holland overhoord

138. ALFABET: probeer zo snel mogelijk het alfabet te typen

139. REKEN6: het helpen van kinderen bij het automatiseren van sommen, de sommen kunnen worden aangepast aan het niveau van de leerling, bijv. tafels tot 6 of sommen tot 20

140. REKEN7: oefenen van aftreksommen met behulp van de abacus, de leerling kan zelf sommen opgeven of dit door de computer laten doen (op 9 niveaus)

141. NED1: zinnen moeten zelf of d.m.v. een cassette recorder worden voorgelezen, de computer kent dezelfde zinnen, de leerling kan dan de zinnen natypen, waarna ze door de computer gecontroleerd worden

142. NED2: oefening voor kinderen om woordbeelden vast te houden en weer te geven, er zijn woorden opgenomen uit de methode 'woordspel' voor de 5e groep, de woorden staan in DATA-statements en kunnen dus snel gewijzigd worden, de woorden worden gedurende een korte tijd op het scherm getoond, waarna de leerling ze uit het hoofd moet natypen, de tijd kan gekozen worden tussen 1 en 10 seconden

143. BODEMD: determinatietabel voor een aantal bodemdieren

144. AARDR3: aan de hand van een kaart worden de plaatsnamen van de provincie zuid-holland overhoord

145. KROMMEN2: het tekenen van de grafieken van 4 functies:

1. $x = \cos(t + \pi / k)$, $y = \sin(2 * t)$
2. $x = t * \log(t)$, $y = 2 + \log(t)$
3. $x = \log(\text{abs}(t)) - 1 / t$, $y = 4 * t$
4. $x = 2 * t / (t * t - 4)$, $y = \text{sqr}(t) - 3$

de functies kunnen gewijzigd worden

146. TCOPY2: kopieerprogramma van tape naar tape, verbeterde versie van het programma 'tapecopy' (no 16)

147. GRAFIE11: het tekenen van grafieken van functies, er zijn 8 functies voorgeprogrammeerd, die desgewenst gewijzigd kunnen worden, verbeterde versie van het programma 'grafiek1' (no 2)

148. INTEGR2: het tekenen van integraalkrommen m.b.v. kutta's simpsonsregel, verbeterde versie van het programma 'integr1' (no 5)

149. INTEGR3: het tekenen van integraalkrommen m.b.v. trapreg + predictor, de opties zijn hetzelfde als in de programma's 5 en 148

150. GRAFIE12: het tekenen van grafieken van functies, het berekenen van nulpunten volgens de methode van newton en het berekenen van oppervlakken

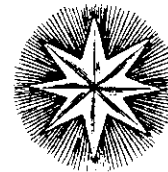


prijsvraag

deze keer presenteert de newbrain-gebruikersgroep een ander type prijsvraag dan beide vorige keren, de opdracht is nog wel programmeren in basic, maar nu wordt er niet een geheel zelfstandig programma verwacht, maar een module of een overlay, die iets extra's aan het hieronder beschreven sterrenkaartprogramma toevoegt, hiervoor is geen astronomische kennis vereist, het gaat om het programmeren in basic en het gebruik van de mogelijkheden van de newbrain, machinetaalprocedures zijn uiteraard toegestaan, een tweede noviteit is, dat elke inzender van een serieuze overlay in beloond wordt met een cassette of diskette uit de softwarebibliotheek, de winnaar kan er vijf krijgen.

uw oplossing wordt tot 1 augustus 1986 ingewacht op ons adres:

newbrain-gebruikersgroep
postbus 4494
1009 AL amsterdam



STER.BAS tekent op elk gewenst tijdstip op elke gewenste plaats op aarde de sterrenhemel, Als extra is een aantal overlays toegevoegd:

STERSNEL.OVL met een 'geheugen'-array voor de sterren,

STERMERC.OVL tekent de sterrenkaart in Mercatorprojectie (?),

STERHALL.OVL tekent ook de komeet Halley, helaas zonder staart.

Het oorspronkelijke programma is voor een Exidy Sorcerer geschreven en is via een programma voor een Commodore 64 omgezet in NewBrain-basic door N. Ballast uit Muiderberg, Dit programma was 17k groot en draaide alleen op de expanded NewBrain vanwege het nonchalante gebruik van het geheugen, Door mij is het bewerkt, waarbij allerlei truuks om geheugen te besparen zijn toegepast, Het gevolg is, dat het op alle NewBrains kan draaien en dat er na het tekenen van de kaart in een unexpanded NewBrain nog ruim 11k vrij geheugen is, Dit biedt de mogelijkheid om nog een aantal routines aan het programma toe te voegen, bij voorbeeld om de sterrenhemel te laten draaien, om de baan van de zon of van satellieten toe te voegen.

STER.BAS is door mijn persoonlijke stijl van programmeren misschien wat ondoorzichtig geworden. Voor programmeurs van aanvullende routines volgt daarom hier enige uitleg over de opbouw van STER.BAS.

regel 4: inleiding, is verder niet nodig en wordt daarom na gebruik gewist
regels 5 - 9: subroutines, die vaak worden gebruikt.

regel 7: subroutine die alleen bij het starten gebruikt wordt en daarna wordt gewist.

regels 10 - 13: inladen van de benodigde data en afhankelijk van de toegevoegde overlay; STERDAT.DAT bevat ook de data voor STERHALL.OVL

regels 15 - 130: interactieve invoer van de benodigde data.

regels 140 - 250: het hoofdprogramma; hier kunnen verwijzingen naar subroutines worden toegevoegd, zie bij voorbeeld 225 in STERHALL.OVL.

regels 300 - 320: omzetting van LMT (plaatselijke tijd) in LST (sterretijd) en datum in astronomische datum

regels 400 - 495: berekening van de positie van zon, planeten, maan en eventueel sterren. Niet-astronomen hoeven de routines niet te begrijpen om ze te kunnen gebruiken. Iedere routine levert een x- en een y-waarde op, die de positie van het desbetreffende hemellichaam op het grafische scherm bepalen. Met deze x- en y-waarden kunnen ook andere plot-opdrachten worden uitgevoerd.

Met STERMERC.OVL zijn de x- en de y-waarden de azimuth en de altitude, zonder deze overlay zijn ze coördinaten van het grafische scherm. De aanpassing hieraan vindt plaats in subroutine 6: $x = x * \cos(y)$

regels 500 - 510: plotten van de sterren. De berekende x- en y-waarden van de 237 sterren worden in tegenstelling tot die van de andere hemellichamen niet bewaard. Dit kost namelijk 2889 geheugenplaatsen. Daarom worden de x- en y-waarde per ster tijdens het plotten berekend. Er zijn echter mogelijkheden om de berekende en de uitgangswaarden niet als floating-pointgetal maar als integer op te slaan, waarbij naar schatting maar circa 550 geheugenplaatsen nodig zijn, inclusief de coderingsroutines. Wie neemt dit idee over (zie bij voorbeeld Z80-DIS.BAS)?

Met STERSNEL.OVL is er een extra array, ts (236, 1), waarin de sterreposities ten opzichte van de horizon kunnen worden bewaard. Dit array kost 2889 geheugenplaatsen.

regels 600 - 610: plotten van de zon

regels 700 - 710: plotten van de planeten

regels 800 - 810: plotten van de maan; het berekenen van de maanfase en de te plotten vorm van de maan is al gebeurd in regel 475.

regel 900: opzetten van het grafische scherm. De 'hemel' bestaat uit een gedeelte van 180 bij 90 graden, waarbij 0 = noord, 90 = oost, 180 = zuid en 270 = west.

Berekening van de te plotten x- en y-waarden:

In eerste instantie wordt als x-waarde de positie op de horizon (azimuth) uitgerekend en als y-waarde de positie boven de horizon (altitude). Beide worden in graden uitgedrukt. Daarna wordt gekeken, of de y-waarde tussen de 0 en 90 is, en zo ja, dan wordt subroutine 6 aangeroepen. In subroutine 6 wordt bepaald of de x-waarde valt binnen het gewenste stuk horizon, en zo ja, dan wordt x verminderd met het aantal graden, dat overeenkomt met de linkerzijde van de getekende horizon.

De benodigde data (efemeriden) zijn niet in het programma als DATA opgeslagen, maar staan in een afzonderlijk bestand, STERDAT.DAT. Het laden vanuit dit bestand kost enig geheugen, afhankelijk van de gebruikte configuratie.

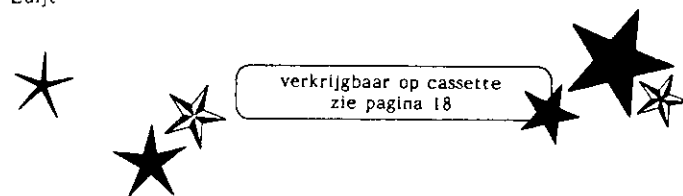
De kleinste configuratie is een unexpanded NewBrain met diskdrives, omdat bij het openen van een in- of uitvoerstream naar disk (device 12 - 15) altijd 4,7 k geheugen nodig is om het diskbesturingssysteem te laden. Bezitters van een NewBrain met cassette recorder kunnen dit vergelijken met een in- of uitvoerstream naar de recorder (device 1 of 2) met een buffer van minimaal 4700 bytes, dat wil zeggen, ze moeten een stream naar device 1 of 2 kunnen openen met OPEN #1, 1, "*19", als er in- of uitvoer moet plaats vinden, wil het programma ook op systemen met diskdrives kunnen werken.

Een ander heeft gevolgen voor array ts (237, 1), dat pas gedimensioneerd mag worden als de invoerstream geCLOSEd is.

De rekennauwkeurigheid van de NewBrain is beter dan die van de computers waarvoor STERDAT.DAT is opgesteld. De benodigde data zouden dus nog nauwkeuriger kunnen. Wie heeft ze?

Ik hoop, dat er veel gebruikers zijn die de handschoen opnemen en er aanvullende routines voor schrijven. De beste routine in de vorm van een overlay, die voor 1 augustus bij de redactie binnen is, ontvangt als prijs 5 delen uit de softwarebibliotheek.

Wim Luijt



BRAINSHEET

BRAINSHEET is een spreadsheet voor de 32k NewBrain. Het werkt echter op ALLE NewBrains. BRAINSHEET past zich bij het installeren aan de NewBrain, de printer en de monitor aan.

Een speciale routine zorgt ervoor, dat vastlopen en/of het verlies van data **UITGESLOTEN** is !

Capaciteit:

maximaal 26 kolommen; maximaal 99 rijen; maximaal aantal cellen met numerieke inhoud: ca. 800; cellen met een tekst verminderen het aantal cellen; op een 32k NewBrain is 22 kbyte geheugen voor informatie (data) beschikbaar !

Bewerkingen:

bewaren van een spreadsheet op cassette of disk; afdrukken van een spreadsheet; bewaren van een afdruk op cassette of disk; kolommen of rijen blinderen; calculator; enz.

Berekeningen:

optellen, aftrekken, vermenigvuldigen, delen, machtsverheffen, som, 3 soorten gemiddelden, standaard afwijking, ALLE berekeningen in NewBrain-BASIC (ook in formules).

PRIJS: voor leden NewBrain-gg f 30,- inclusief uitgebreide handleiding.
Bestellen door overmaking van f 30,- op giro 581510 t.n.v.
W.M.Luijt, Troelstrapad 22, 4463 VM Goes
onder vermelding van BRAINSHEET cass. of disk met gewenste formaat.

TOETSDETECTIE

Voor een simulatieprogramma zoek ik een routine, waarbij continu wordt gedetecteerd, of er al dan niet een toets is ingedrukt. Ik zal het toelichten: stel, er is een stip zichtbaar op het scherm. Zolang ik op toets 'A' druk, beweegt de stip naar links. Laat ik de toets

los, dan staat de stip stil. Zodra ik toets 'B' indruk, gaat de stip naar rechts en blijft naar rechts gaan, totdat ik de toets loslaat. Denk niet, dat dat zo makkelijk is op te lossen. Met device 6 lukt het niet. Dat detecteert wel een toetsindruk, maar zendt direct na de toetsindruk weer nullen, totdat opnieuw een toets ingedrukt wordt.

Pieter W. M. van Dijk

SNELLER MET SAVF

Wie zich een disk-controller aangeschaft heeft, heeft daarmee de beschikking gekregen over de basic-opdracht SAVF. Het verschil met de opdracht SAVE is, dat bij SAVE de regels in oplopende volgorde van regelnummer opgeslagen worden, en bij SAVF juist in de omgekeerde volgorde, dus eerst het hoogste, en het laagste het laatst.

De opdracht SAVF is niet alleen te gebruiken bij opslag op schijf, maar in alle gevallen, waar ook SAVE gebruikt kan worden. En bestanden, die met SAVF opgeslagen zijn, kunnen door alle NewBrains op dezelfde manier gelezen worden als bestanden, die met SAVE opgeslagen zijn (alleen VERIFY is niet meer mogelijk).

Het voordeel van de opdracht SAVF is, dat de NewBrain de binnenkomende programmaregels sneller kan tussenvoegen in zijn geheugen. Bij het oversturen van een programma van 7 kbytes van een NewBrain naar een andere via device 9 wordt de helft van de tijd bespaard: er zijn 22 seconden nodig in plaats van 42! En bij gebruik van device 9 is duidelijk te zien, op welke momenten de NewBrain aan het communiceren is (dan knippert het scherm) en op welke momenten de ontvangen regels tussengevoegd worden. Bij SAVF staat het scherm korter stil.

Rob Maris



Wanneer men in het bezit is van een NewBrain met alleen een diskdrive zonder de expansie-eenheid, zal het regelmatig voorkomen, dat het programma niet kan worden weggeschreven na het inladen van de opgenomen cassette. Dit komt, doordat de nu uitgezonden programma's (in hoofdzaak de beeldkrant) te veel geheugen in beslag nemen.

Een oplossing hiervoor is dan in het laad- en vertaalprogramma een routine in te bouwen, die een deel van dit vertaalprogramma DELETE. Een tweede verandering kan dan terzelfde tijd mee gebeuren. Dit is namelijk het vertaalde programma niet weg te schrijven op device 12, maar dit te doen op device 13. Dit doe ik, omdat dikwijls slecht opgenomen programma's niet meer ingeladen kunnen worden (regelnummer te groot krijgt men dan), en met het programma ook niet meer kan laten runnen.

Wanneer het echter is weggeschreven op device 13, kan men in cp/m met 'type' dit echter nog wel bekijken en uitzoeken waar de fout zich voordoet (en met een tekstverwerker verbeteren).

Zelf heb ik het leesprogramma afkomstig van disk HULP-1 op de volgende wijze aangepast: de regel IF dv = 1 OR dv = 2 . . . heb ik veranderd in:

```
IF FREE < 5000 THEN DELETE -1300 : DELETE 10000-
```

Dit maakt terug ongeveer 5k geheugen vrij, dat de diskcontroller nodig heeft om te kunnen wegschrijven.

Verder ook nog de volgende regel veranderd voor device 13:

```
OPENOUT #12, dv, a$ . . . in OPENOUT #12, 13, a$
```



en ge kunt het vertaalde programma altijd en snel bekijken met 'type'.

A. Moreels

ASSEMBLER Z80BRAIN

De assembler Z80BRAIN is wél feilloos (On-line 6, pag. 50), hij is alleen een beetje eigenwijs. Lees de handleiding goed door!

Frans Horstink

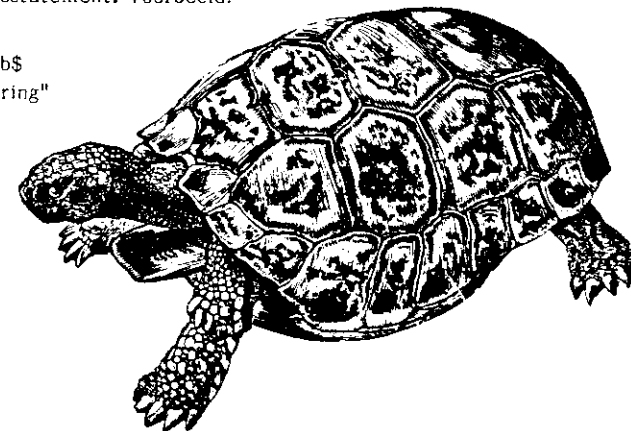
langzaam

LANGZAME STRINGBEWERKINGEN IN BASIC

De NewBrain is een redelijk snelle machine. Ware het niet, dat de stringbewerkingen op de NewBrain de sfeer aardig kunnen verpesten. Onderstaand wordt toegelicht, waar u op moet letten, indien u het onderste uit de kan wilt halen.

Iedere keer als een string gevuld wordt met een nieuwe string, wordt er behoorlijk met geheugenblokken geschoven in de NewBrain, omdat de lengte van de nieuwe string van te voren niet bekend is. Wanneer is er sprake van het vullen van een string met een nieuwe string? Bij het variabele-toekenningstatement. Voorbeeld:

```
a$ = b$  
b$ = "string"
```



Voor de duidelijkheid: het volgende dient onderscheiden te worden:

```
PRINT a$  
IF a$ = b$ . . .
```

Dit zijn geen toekenningen, maar bewerkingen die uitsluitend strings lezen. Er is dan ook geen sprake van verplaatsen van variabeleblokken. Het zijn dus gewone 'snelle' bewerkingen. Vooral wanneer er veel variabelen gebruikt worden, gaan toekenningbewerkingen vrij langzaam. Getalvariabelen hebben daar geen last van, omdat de lengte daarvan altijd vast

staat op 6 bytes. Het komt er dus op aan zo min mogelijk string-toekenningen te gebruiken.

Voorbeeld. Stringtoekenningen bij voorkeur buiten for-next-loops houden. Herhaald de simpele bewerking `a$ = "k"` uitvoeren kan de for-next-loop, ook een lange, flink vertragen. Ander voorbeeld.

```
d$ = a$ + b$ + LEFT$ (c$, 2) : ? d$ : ? #8, d$
```

Sneller gaat:

```
? a$ + b$ + LEFT$ (c$, 2) : ? #8, a$ + b$ + LEFT$ (c$, 2)
```

Noot: in de output is er verschil tussen het gebruik van `+` en `;` in het `?`-statement bij lengtes groter dan de schermbreedte.

Derde voorbeeld.

```
p$ = MID$ (STR$ (ASC (LEFT$ (a$, 1)) - 48), 2, 2) :  
? p$ + "-" + MID$ (STR$ (ASC (MID$ (a$, 2, 1)) - 48), 2, 2)
```

Sneller gaat:

```
? FNg$ (LEFT$ (a$, 1)) + "-" + FNg$ (MID$ (a$, 2, 1))  
DEF FNg$ (d$) = MID$ (STR$ (ASC (d$) - 48), 2, 2)
```

Ook in het laatste voorbeeld zit het verschil in de afwezigheid van een stringtoekenning in de goede oplossing. Jawel, er wordt wel een string toegekend, maar dat doet hij bij de FN-functie intern, in zijn klad-geheugen, en dat gaat niet gepaard met tijdrovende geheugenblokverschuivingen. Grappig is trouwens, dat aan de FN-functie de compilerwerking van NewBrain-BASIC zeer goed te zien is. Een uitgebreider versie van voorbeeld 3 (conversie van 3 bytes naar ondubbelzinnige datum van de vorm `? FNh$ (c$)` geeft "12-9 -85" etc.) kost ruim een SECONDE bij de eerste aanroep, maar een fractie (minder dan 50 ms) bij iedere volgende aanroep.

In ieder geval een pleidooi om, waar het even kan, FN-definities toe te passen. Extra voordeel is ook, dat de interne variabelen gerust dezelfde namen mogen hebben als variabelen in het programma, zonder dat het problemen geeft.



Rob Maris

32

basic

newbrain on-line 7

integer



INTEGER-BUG

Op de NewBraindag in januari j.l. attendeerde iemand me op een bug, die ik in het verleden ook al eens was tegengekomen, maar waar ik toen geen oplossing voor had gezocht.

De INTEGER van $10 * 3.2 = 32$. Probeer je dit op de NewBrain, dan krijg je als antwoord: 31.

Om de oorzaak van de bug op te sporen ben ik gaan onderzoeken, hoe deze fout kan ontstaan. Er zijn namelijk twee mogelijkheden:

1. in de vermenigvuldiging met de floating point van de NewBrain wordt een fout gemaakt;
2. de INTEGER-routine in het MATHS-pack is niet goed of houdt geen rekening met de eigenaardigheden van de floating point van de NewBrain.

Allereerst moest ik onderzoeken, hoe de floating-pointnotatie van de NewBrain is. De resultaten hiervan staan verderop in deze On-line. Toen ik wist, hoe gebroken getallen als floating point worden bewaard, kwam ik erachter, dat 32 anders wordt bewaard dan $10 * 3.2$. Bij handmatige uitvoering van de berekening bleek, dat

de vermenigvuldiging correct wordt uitgevoerd. Ik vond hetzelfde als de NewBrain, maar het resultaat is kleiner dan 32, namelijk $32 - (2 * -40)$.

Wordt dit resultaat door een Operating System Routine, waarschijnlijk ZOUT, omgezet in een string, bij voorbeeld voor het afdrucken op het scherm), dan wordt er kennelijk rekening gehouden met dit kleine verschil.

newbrain on-line 7

basic

33

De O.S.-routine ZINT doet het duidelijk niet. Van de analyse van deze routine ben ik door gebrek aan kennis over de rekenroutines in Z80-machinecode niet veel wijzer geworden.

Het antwoord op het probleem is dus, dat de O.S.-routine ZINT niet deugt! Mogen de ontwerpers van een NewBrain II (?) hier nota van nemen.

Voor de programmeurs in BASIC de volgende oplossing:

```
10 DEF FNin = INT (in * (128 * 256 + -5))
```

en in het programma:

```
XXXX in = YYY : in = FNin : REM in = INT (YYY)
```

Voor YYY kun je de gewenste berekening invullen.

Op deze wijze wordt de variabele, waarvan men de integer wil gebruiken, met 1.6 E-10 deel groter gemaakt. Houd hier rekening mee!

Als er iemand een elegantere oplossing weet, laat hij hem bekend maken in On-line. Ik heb deze methode getest en geen fouten gevonden. Dat wil zeggen, ik garandeer NIET, dat hij altijd de gewenste en goede uitkomst geeft. Hij maakt in ieder geval veel minder fouten dan de ongewijzigde methode!

Wim Luijt

SSCC

BASIC SYMBOLIC SOURCE CODE CONVERTER

De grote ellende van BASIC is, dat grote programma's praktisch onhanterbaar zijn en dus slecht onderhoudbaar. Bij de standaard-NewBrain is dan de ellende compleet, als ten behoeve van de ruimte voor de variabelen en eventueel het disksubstelsysteem alle REM-statements letterlijk het veld moeten ruimen.

Nu kan men overstappen op PASCAL of C. Uitstekend. Prima. Maar de ondersteuning van het NewBrain Operating System dan? Bij mijn weten is er nog geen PASCAL voor de NewBrain, die vergelijkbare ondersteuning biedt voor het I/O-systeem als BASIC, en laat staan voor de grafische schermen. Dat is erg jammer, aangezien een kale hogere programmeertaal met slechts kleine implantaties zeer goede resultaten kan opleveren, omdat bij het ontwerp van de NewBrain het I/O-systeem, het rekenkundig systeem en het grafische systeem als modules zijn opgebouwd, die

slechts hoeven te worden 'aangesloten' op een kale taal om iets moois te krijgen.

Het feit, dat de Z80 zeven werkregisters heeft, in combinatie met de RST-systeem-aanroepen, maakt de communicatie tussen taal en systeem wel erg gemakkelijk.

Helaas. Het heeft niet zo mogen zijn. Grundy failliet, Tradecom machteloos. Onvoldoende inspanning.

EEN ALTERNATIEF

Wat dacht u van een symbolische BASIC? In plaats van (regel)nummers gebruiken we namen, die associëren met de functie van een (regel)nummer. Ongelimiteerd gebruik van commentaar, en nog



meer voordelen die straks duidelijk worden. Assemblerprogrammeurs wordt direct duidelijk, wat de bedoeling is. Voor ieder ander wordt hieronder ook duidelijk gemaakt, wat de bedoeling is.

Het programma is oorspronkelijk gemaakt voor CP/M-systemen. De gang van zaken is als volgt. U tikt een programma in met de beste editor die u kent (bij voorbeeld PRGED of BASED *)), u haalt de programmatekst vervolgens door het hulpprogramma SSCC en ziedaar, u hebt dan een direct RUNbaar BASIC-programma met uiteraard regelnummers en dergelijke.

Ik zal nu de mogelijkheden en de 'syntax' behandelen en voorbeelden geven. Vervolgens geef ik nog wat voordelen aan.

Voor de tekstfile geldt het volgende.

Iedere tekst die tussen twee '~'-tekens staat (GRAPHICS/?), wordt beschouwd als label, dat wil zeggen, wordt vervangen door de string, die de werkelijke betekenis aanduidt. Bij voorbeeld regelnummers, maar een label kan ook een naam zijn, bij voorbeeld een variabelenaam. De tekst is verdeeld in twee velden. Een labelveld en een tekstveld. De scheiding tussen het label- en het tekstveld geschiedt met één TAB-karakter (ASCII 9). Een willekeurige string in het labelveld krijgt een nummer toegewezen (regelnummer).

Zelf betekenis geven aan labels kan ook. Bij voorbeeld:

```
label1 equ 100
count equ a
```

Op willekeurige plaatsen in de tekst is het mogelijk om andere regelnummering bij de RUN-file te dicteren. Bij voorbeeld:

```
org 2000
```

Het standaardbegin is bij 100. Verder is het mogelijk om lange regels te maken door een regel af te sluiten met het teken '~'. De volgende regel wordt dan hierachter gekoppeld, met dien verstande, dat het eerste karakter van de volgende regel niet wordt meegenomen. Dit is namelijk normaal gesproken het TAB-karakter, zodat het vervolg van de eerste regel niet in het labelveld komt.

We komen nu tevens op het formaat van de regels:

*) Voor recordergebruikers is een eenvoudige, maar speciaal aan SSCC aangepaste teksteditor (SSCCEDIT.BAS) in de softwarebibliotheek beschikbaar op cassette ONLINE-7

<string, label, optioneel> <TAB-karakter> <string, tekst, optioneel> <CR/LF>
of de lege regel: <CR/LF>

Alle CP/M-teksteditors voldoen aan dit formaat. Een goede editor is in staat om ook het benodigde TAB-karakter te genereren (aanbevolen: PRGED, verkrijgbaar bij (sorry) Maris electronics).

Tot slot - heel belangrijk - de commentaarmogelijkheden. Overal, waar de combinatie ';'/' voorkomt, wordt de rest van de regel door SSCC genegeerd. Gewone REM-statements, die in het programma blijven staan, blijven mogelijk.

Hieronder ziet u een voorbeeld, waarin alle mogelijkheden verwerkt zijn. Een programmatekst, en daarna het zogenaamde object.

source programma TEST.BSS (bss = basic source)

```
;/BASIC SOURCE FILE TEST FOR SSCC
;/nonsens programma
```

```
delay equ 120
command equ a
flag equ f ;/flag for such purpose
```

```
start FOR N = 1 TO ~delay~
      ? N ;/short delay loop
      NEXT N
```

```
login INPUT A$
      IF A$ = "ROB" THEN ~messrob~
      ? "JIJ BENT ROB NIET" : GOTO ~skipr~
messrob ? "HALLO, ROB, HOE GAAT HET MET JE?"
skipr input ("doe uw keuze nu: ") ~command~
      on ~command~ goto ~load~, ~save~, ~seek~, ~stop~
      goto ~start~
```

```
;/-----
;/here the several command routines
```

```
org 500
load ? "loading a file"
      goto ~start~
save ? "saving file" : goto ~start~
```



```

seek      if ~flag~ ret
          j = 1 : gosub ~error~ : goto ~start~
stop      end

```

```

;/-----
;/here just the subroutines
;/communication variables: j

```

```

error      on j goto ~err1~, ~err2~

```

```

err1       ? "dit is een fatale fout die tot gevolg heeft dat u in de \
            gevangenis belandt" : ret
err2       ? "mispoes" : ret

```

objectprogramma TEST.BSO (bso = basic object)

```

100 FOR N = 1 TO 120
110 ? N
120 NEXT N
130 INPUT A$
140 IF A$ = "ROB" THEN 160
150 ? "JIJ BENT ROB NIET" : GOTO 170
160 ? "HALLO, ROB, HOE GAAT HET MET JE?"
170 INPUT ("doe uw keuze nu: ") a
180 ON a GOTO 500, 520, 530, 550
190 GOTO 100
500 ? "loading a file"
510 GOTO 100
520 ? "saving file" : GOTO 100
530 IF f RET
540 j = 1 : GOSUB 560 : GOTO 100
550 END
560 ON j GOTO 570, 580
570 ? "dit is een fatale fout die tot gevolg
heeft dat u in de gevangenis belandt" : RET
580 ? "mispoes" : RET

```



Als u het verschil ziet, dan is het misschien wel even wennen, maar als u er mee bezig bent, wordt het juist plezierig werken. Groot voordeel is, dat u nu zeer goed kunt documenteren. Verder maakt het gebruik van labels duidelijk, welke regels aangeroepen kunnen worden vanuit andere regels, zodat het overzicht beter wordt. Met een goede editor is het ook mogelijk om het voorkomen van labels en andere strings op te sporen en naar de

gewenste plaats te springen in de programmatekst. Tot slot is het verplaatsen van blokken programma geen probleem bij de meeste editors.

Hieronder volgt een listing van het programma SSCC:

```

90 IF PEEK (19) = 12 THEN MERGE "SSCC.DSK" : DELETE 90
100 ON ERROR GOTO 900 : FOR i = 1 TO 255 : CLOSE #i : NEXT i
110 t$ = CHR$ (9) : s$ = CHR$ (126)
120 OPEN #0, 0, "I" : ? "BASIC symbolic source code converter (sscc)"
130 ? : LINPUT ("put recorder on play ") a$ : ? : INPUT ("source file
name: ") f$ : IF LEN (f$) THEN f$ = "." + f$ + ".BSS"
140 CLOSE #1 : OPEN #1, "*20" + f$ : a$ = FILES : IF RIGHT$(a$, 4) <>
".BSS" THEN 140
145 f$ = "*20:" + LEFT$ (a$, LEN (a$) - 4) : t1 = 16384 : t2 = t1 - 1 :
RESERVE TOP - t1
150 ? : ? "pass 1" : n = 90 : l$ = " "
160 LINPUT #1, a$ : GET #1, q
170 IF LEFT$ (a$, 1) = ";" OR LEN (a$) = 0 THEN 160
175 FOR i = 1 TO LEN (a$) : POKE t2 + i, ASC (MID$ (a$, i)) : NEXT i :
t2 = t2 + i : POKE t2, 13
180 p = INSTR (a$, t$ + "equ") : IF p THEN l$ = l$ + LEFT$ (a$, p - 1) +
" " + MID$ (a$, p + 5) + " " : ? LEFT$ (a$, p - 1), MID$ (a$, p + 5) :
GOTO 160
190 p = INSTR (a$, t$ + "org") : IF p THEN n = VAL (MID$ (a$, p + 5)) - 10
: GOTO 160
200 n = n + 10 : IF LEFT$ (a$, 1) = t$ THEN 230
210 p = INSTR (a$, t$) : IF p = 0 THEN p = LEN (a$)
220 l$ = l$ + LEFT$ (a$, p - 1) + STR$ (n) : ? LEFT$ (a$, p - 1), n
230 IF RIGHT$ (a$, 1) = CHR$ (92) THEN n = n - 10
240 GOTO 160
250 CLOSE #1 : ? : ? : ? "pass 2"
260 ? : ? : LINPUT ("put recorder on record ") a$ : OPENOUT #2, f$ +
".BSO" : n = 90
270 b$ = ""
280 a$ = ""
285 a = PEEK (t1) : t1 = t1 + 1 : IF a <> 13 THEN a$ = a$ + CHR$ (a) :
GOTO 285
290 IF LEN (b$) THEN a$ = LEFT$ (b$, LEN (b$) - 1) + MID$ (a$, 2)
300 IF RIGHT$ (a$, 1) = CHR$ (92) THEN b$ = a$ : GOTO 280
310 e = INSTR (a$, ";/") : IF e = 1 THEN 270
320 IF INSTR (a$, t$ + "equ") THEN 270
330 p = INSTR (a$, t$ + "org") : IF p THEN n = VAL (MID$ (a$, p + 5)) - 10
: GOTO 270

```

```

340 l = INSTR (a$, s$) : IF l = 0 THEN 410
350 ll = INSTR (a$, s$, l + 1) : IF l = 0 THEN ? "* missing "; s$; " error: "
: GOTO 410
360 i = INSTR (l$, " " + MID$ (a$, l + 1, ll - l - 1) + " ") + 1
370 IF i = 1 THEN : ? "* undefined label error: " : GOTO 410
380 il = INSTR (l$, " ", i + ll - 1)
390 a$ = LEFT$ (a$, l - i) + MID$ (l$, i + ll - 1, il - i - ll + 1) + MID$ (a$,
ll + 1)
400 GOTO 340
410 i = INSTR (a$, t$) : e = INSTR (a$, ";/") : IF e = 0 THEN e = LEN (a$)
+ 1
420 IF LEN (a$) THEN n = n + 10 : ? #2, STR$ (n) + MID$ (a$, i + 1, e - i -
1) : ? STR$ (n) + MID$ (a$, i + 1, e - i - 1)
430 IF t1 < t2 THEN 270
440 CLOSE #2 : END
900 IF ERRNO = 133 IF ERRLIN = 160 THEN RESUME 250
920 REPORT

```

SSCC.DSK (overlay voor diskettegebruikers)

```

130 ? : INPUT ("source file name: ") f$
140 OPEN #1, 12, f$ + ".bss"
145
175
260 OPENOUT #2, 12, 2, f$ + ".bso" : OPEN #1, 12, f$ + ".bss" : n = 90
280 LINPUT #1, a$ : GET #1, q
285
430 GOTO 270
440 CLOSE #1 : CLOSE #2 : END
900 IF ERRNO = 151 IF ERRLIN = 160 THEN RESUME 250
910 IF ERRNO = 151 IF ERRLIN = 280 THEN RESUME 440

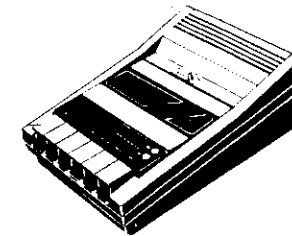
```

Gebruik in het kort: Bij de vraag 'filename:' invullen: naam van source tekst zonder achtervoegsel .BSS (dat voegt SSCC zelf toe). Resultaat: file met dezelfde naam, maar dan achtervoegsel .BSO, bevat RUNbare file. Bij LOADen komt aan het eind een ERROR 151, maar dat lost u op door opnieuw te SAVFen.

Rob Maris

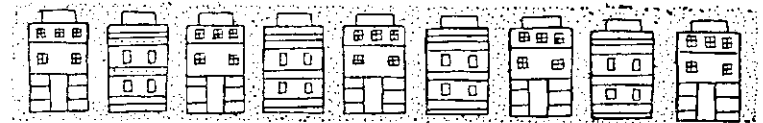
verkrijgbaar op cassette
zie pagina 18

cassette



HOE LEEST DE NEWBRAIN BESTANDEN VAN CASSETTE EN WELKE FOUTEN ZIET HIJ DAARBIJ

Bezitters van een NewBrain met cassetterecorder kunnen aardig gefrustreerd raken, als het inlezen van een bepaalde cassette niet lukt. Als er (nog) geen BACK-UP (kopie) gemaakt is, of er geen listing is, betekent dit opnieuw intypen van de data of het programma. Gebruikmakend van de schaarse informatie heb ik een methode ontwikkeld, die deze bufferen grotendeels kan redden. Om de methode met succes te kunnen toepassen is echter enige kennis van de in- en uitvoer van en naar cassette noodzakelijk. In dit artikel zal ik de theorie van de invoer van cassette behandelen, samen met de daarbij optredende fouten. In een volgend artikel zal de methode worden beschreven, samen met de bijbehorende programma's.



De Operating System (O.S.) Routines die databytes van en naar het geheugen van de NewBrain transporteren, maken geen onderscheid tussen programma's en 'echte' data, zoals getallen en teksten. Anders gezegd: de O.S.-inleesroutines kunnen aan de file op cassette niet zien, of het een data- of een programmaprofile is; pas de BASIC-opdrachten LOAD, SAVE, MERGE en VERIFY instrueren de NewBrain, dat de bytes een programma vormen, en daarom als zodanig moeten worden behandeld. In dit artikel worden met data dus aangeduid zowel data, die een programma vormen, als 'echte' data. Verder wordt alleen het transport van de cassette naar het

geheugen van de NewBrain behandeld, maar voor de uitvoer geldt in veel gevallen hetzelfde, maar dan in omgekeerde volgorde. Verder wordt alleen device 1 (cassetterecorder 1) behandeld, want voor device 2 (cassette-recorder 2) geldt hetzelfde.

Data, die van de cassette naar het geheugen moeten om voor BASIC beschikbaar te zijn, worden eerst in een buffer geplaatst. Deze buffer is niet hetzelfde als het zogenaamde Own Memory, maar vormt er wel een deel van. Tijdens het transport knippert het scherm en doven de 8 rechtse tekens op het VF-display op Model AD. Daarna worden de data vanuit de buffer op de plaatsen in het geheugen geplaatst, waar BASIC ze nodig heeft. Het scherm staat stil en het lijkt, of er niets gebeurt

De plaats in het geheugen van de buffer en het formaat zijn niet vast. De plaats in het geheugen kan onder BASIC moeilijk worden bepaald en heeft weinig praktische betekenis, omdat ze pas gemaakt wordt, als een stream geopend wordt. Zodra de stream naar de recorder weer gesloten wordt, zowel normaal als door een fout, wordt de buffer overschreven en gaan data in de buffer verloren onder BASIC.

Het formaat van de buffer kan wel worden bestuurd onder BASIC met de parameterstring in de OPEN-opdracht. Dit gebeurt door '*' gevolgd door een getal. Het getal geeft aan, hoe groot de buffer is in veelvouden van 256 bytes. '*20' geeft dus een buffer van $20 \times 256 = 5120$ bytes = 5 kbyte. Wordt geen '*<getal>' opgegeven, dan is de standaard (default) waarde 4, wat overeenkomt met een buffer van 1024 bytes = 1 kbyte. De voor- en nadelen van een ander bufferformaat dan standaard zullen verderop behandeld worden. Heeft het bestand ook een naam, dan volgt er na het getal een ':' met de bestandsnaam. Deze bestandsnaam wordt op het scherm getoond, als de opdracht OPEN #1, 1 met succes is volbracht. Deze bestandsnaam mag 255 tekens lang zijn. Door nu in de bestandsnaam allerlei stuurtekens, zoals CHR\$ (31), CHR\$ (13) etc. op te nemen kan een aardige lay-out worden verkregen bij het laden van een programma, bij voorbeeld:

```
c$ = CHR$ (13) : a$ = "*20:" + CHR$ (31) + "tekst 1" + c$ +  
"tekst 2" + c$ + "tekst 3" + c$ + "enz." : OPEN #1, 1, a$
```

Hierin kan een heleboel informatie gezet worden, zoals inhoud of gebruiksaanwijzing. Zorg er wel voor, dat LEN (a\$) < 255, anders gaat het mis.

Ook kan er tussen de ':' en de bestandsnaam nog een ')' worden toegevoegd. Bij het openen van een stream voor uitvoer (OPENOUT) heeft dit

BASIC kent ook de opdracht FILE\$, die nauwelijks vermeld wordt in het Handbook. Deze bevat de parameterstring van de laatst geopende invoerstream. Hij kan als volgt worden gebruikt:

```
OPEN #1, 1, "*20:bestandsnaam" : a$ = FILE$ : REM *20 niet nodig
```

Zolang stream 1 geopend is en er geen andere invoerstream geopend wordt, heeft FILE\$ de inhoud 'bestandsnaam' ZONDER '*20:'. Ook als de stream door een ERROR vroegtijdig gesloten wordt, gaat de inhoud van FILE\$ verloren. Zet a\$ = FILE\$ dus direct achter de OPEN-opdracht.

Voor diskdrivebezitters: FILE\$ werkt niet met devices 12 - 15.

geen zin. Alleen bij het openen van een invoerstream wordt de bestandsnaam NIET op het scherm getoond.

Hoe staat een file nu op cassette?

Een file op cassette bestaat uit een beginblok, Name Type Block genoemd in de documentatie van de NewBrain, en een of meer datablokken. Het eerste en laatste blok zal ik ter onderscheiding de kop (Engels 'header') en de staart (Engels 'trailer') noemen. Ieder blok bestaat uit de volgende bytes:

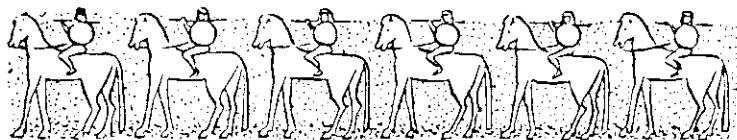
- 1 byte volgens sommige bronnen staat er hier een ongedefinieerd byte zonder betekenis
- 2 bytes lengte van het blok, d.w.z. het aantal databytes in een blok
- n bytes aantal databytes
- 1 byte volgnummer van het blok
- 2 bytes checksum, om te controleren, of het blok goed is ingelezen
- 9 bytes nullen, als buffer bij het stoppen / starten

Verder wordt ieder blok voorafgegaan en gevolgd door een pieptoon van circa 5 seconden.



De kop en de staart onderscheiden zich van de andere blokken, doordat in het volgnummerbyte bit 7 of bit 6 geset is, dat wil zeggen, blok 1 heeft als volgnummer altijd $2^7 + 1 = 129$, en bij het laatste volgnummer is altijd 64

opgeteld. Dit beperkt het aantal echte datablokken voor de NewBrain in theorie tot 63. De kop bevat behalve de parameterstring met het buffer-formaat en de bestandsnaam geen verdere data.



FOUTMELDINGEN BIJ HET INLEZEN

1. Het Handbook geeft de volgende fouten, die direct met de in- en uitvoer van cassette te maken hebben:

ERROR 130 - cassetteleesfout: hardwarefout (drop out), dat wil zeggen, tijdens het inlezen mist er een bit en het inlezen stopt onmiddellijk. Dit is bij mij de meest voorkomende fout. Hij is niet normaal te herstellen. Opnieuw proberen te laden met een hoger volume, maar meestal helpt dit niet. Met de door mij te publiceren methode kan de fout gedeeltelijk worden hersteld.

ERROR 131 - cassetteleesfout: in te lezen blok te groot voor de buffer (of hardwarefout). Makkelijk te herstellen: stream opnieuw openen met voldoende buffer, bijvoorbeeld met '*20:xxxx' enz.

ERROR 132 - checksum is verkeerd. Jammer, dat de ingelezen data toch verloren gaan. Beter was geweest, als deze waarschuwing was gegeven en de ingelezen data toch op de een of andere manier beschikbaar waren geweest. Afhankelijk van de oorzaak kan deze fout waarschijnlijk worden hersteld met mijn methode. Ik heb het helaas (of gelukkig?) nog niet kunnen testen. In theorie klopt het.

ERROR 133 - poging voorbij het einde van het cassettebestand te lezen (of hardwarefout). Deze fout is niet ernstig. Het betekent, dat er meer data van het bestand worden gevraagd dan er aanwezig zijn; of bij een programma, dat de normale afsluiting:

CHR\$ (13) + CHR\$ (4) + CHR\$ (13), kortweg 13 / 4 / 13

niet is gevonden, bij voorbeeld indien een programma niet naar cassette geSAVED was, maar geLIST. Normaal zorgt de opdracht SAVE ervoor, dat

een BASIC-programma wordt afgesloten met 13 / 4 / 13. De gebruiker hoeft zich hier geen zorgen over te maken. Beide BASICODE-vertaalprogramma's van de NewBrain gebruiken deze fout zelfs om het einde van het programma te vinden.

Een misvatting zou kunnen zijn, dat deze fout optreedt, als er een CHR\$ (4), 'end-of-file', ingelezen wordt. De fout wordt echter geactiveerd, als het nummerbyte van het ingelezen blok getest wordt op bit 6. De buffer is dan leeg en de ingelezen data zijn al elders in het geheugen voor BASIC beschikbaar. Bij het LOADen van een programma heeft deze fout dus GEEN betekenis. Het geLOADe programma kan normaal geRUNd worden.

Waarschijnlijk doordat CHR\$ (4) end-of-file genoemd wordt, is gesuggereerd, dat een file moet worden afgesloten door dit teken. Wil dit teken echter effectief zijn, dan moet een file onder BASIC byte voor byte worden ingelezen en getest. Een veel snellere methode met BASIC om een file in te lezen is om dit met LINPUT te doen. Er wordt dan een aantal bytes tegelijk ingelezen, totdat er een CHR\$ (13) wordt ingelezen. De ingelezen string kan daarbij zonder problemen groter zijn dan de buffer.

ERROR 134 - poging de blokken van een cassettebestand in verkeerde volgorde te lezen (of hardwarefout). Treedt op, als er een blok niet gelezen wordt, omdat het eerste byte (het ongedefinieerde byte?) niet gelezen wordt en het volgende blok wel gelezen wordt. Treedt dan echter pas op, als alle data in de buffer staan. Dit is met mijn methode gedeeltelijk te herstellen.



ERROR 135 - poging uitvoer te geven naar een cassettebestand, dat voor invoer geopend is; of andersom. Herstellen door het programma te corrigeren.

ERROR 136 - syntactische (= spellings-) fout in de parameterstring. Treedt alleen op in uitvoerstreams. Herstellen als ERROR 135.

Sommige van bovengenoemde fouten hebben als opmerking: of hardware-

fout. Wat er dan fout is, moeten we maar gissen, de technische documentatie vertelt er niets over. Verder zijn bepaalde fouten aan invoerstreams gekoppeld, zoals 130 tot en met 134. Toch kunnen deze fouten ook optreden bij uitvoerstreams! Misschien dat daarmee de hardwarefout bedoeld wordt, ik heb er geen idee van. Voor de hardwarefouten van 130 en 134 heb ik wel een idee, namelijk dat bij 130 het uitgangsvolume (de geluidssterkte) van de cassette recorder te laag is, en bij 134 andersom (134 wegens hardwarefout heb ik 3 jaar geleden voor het laatst gehad; misschien was het zelfs een ander nummer, ik weet het niet zeker).

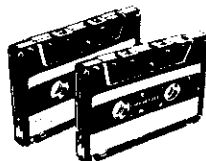
De NewBrain is een van de weinige computers, die met twee cassettepoorten zijn uitgerust. Naar mijn mening wordt hier te weinig gebruik van gemaakt, zeker nu er voor circa f 70,- al goede data-recorders verkrijgbaar zijn. Toepassingen zijn:
INPUT van recorder 1, OUTPUT naar recorder 2;
kopiëren en BACK-UPS maken van bestanden en programma's;
grote bestanden in gedeelte invoeren van recorder 1, bewerken in de NewBrain en het resultaat uitvoeren naar recorder 2. Zelf heb ik op die manier eens een bestand van meer dan 1000 records alfabetisch gesorteerd. Ook kon ik de bedrijfsboekhouding op deze wijze doen met slechts 32k.

2. Fouten, die ontstaan door een verkeerde BASIC-opdracht, bij voorbeeld ERROR 14, 30, 31, 54. Deze zijn gemakkelijk te herstellen door het programma te verbeteren.

ERROR 4 - het LOADen van een 'echte' datafile resulteert meestal in een ERROR 4. Hier is niets aan te doen, omdat het in te lezen bestand niet voldoet aan de eisen, die de LOAD-opdracht aan dat bestand stelt.

Tot zover de theorie achter de in- en uitvoer van en naar een recorder. Volgende keer hoop ik verder in te gaan op de methode om beschadigde bestanden te redden.

Wim Luijt



terminal

DE UNEXPANDED NEWBRAIN ALS TERMINAL

In On-line 2 heb ik geprobeerd uiteen te zetten, wat met 'Terminal Emulatie' beoogd wordt. Kort herhaald: de NewBrain als terminal te gebruiken om te kunnen 'inloggen' op andere (grotere) computers, al of niet via modems aan beide kanten. Dit is met een unexpanded machine moeilijk, omdat de 'communications port' niet interactief gebruikt kan worden. Dat is wel mogelijk met het 'expansion board', omdat daar een zogenaamde UART in zit. Toch is het mogelijk een standaard NewBrain uit te breiden om dit doel te bereiken!

UART AAN EEN UNEXPANDED NEWBRAIN

Er zijn verschillende ontwerpen gemaakt voor een UART-uitbreiding aan de standaard NewBrain. De enige die, voor zover ik weet, in productie gekomen zijn, zijn die van KUMA in Engeland en die van Tradecom. Levering door KUMA is moeilijk vanwege douane, en er wordt geen software bijgeleverd. Dat doet Tradecom wel met een interface, die zij de 'Velotype-interface' noemt. Ik heb er een gekocht (wel bijna f 300!), maar het blijkt perfect te werken voor ons doel. Deze interface geeft bovendien de mogelijkheid om ook 'zonder geflikker' een printer aan te sluiten (RS-232). Tradecom levert een voorbeeld bij, hoe software opgezet kan worden voor communicatiedoeleinden. Het blijkt echter, dat er een assembler-programma nodig is om dat echt goed te doen.

MODEMS

In On-line 2 noemde ik het EXAR-14412 IC als kandidaat om een goedkoop modem te maken. Ik heb van de fabriek nooit antwoord gekregen op de vragen, die naar aanleiding van de documentatie over die chip aan de orde

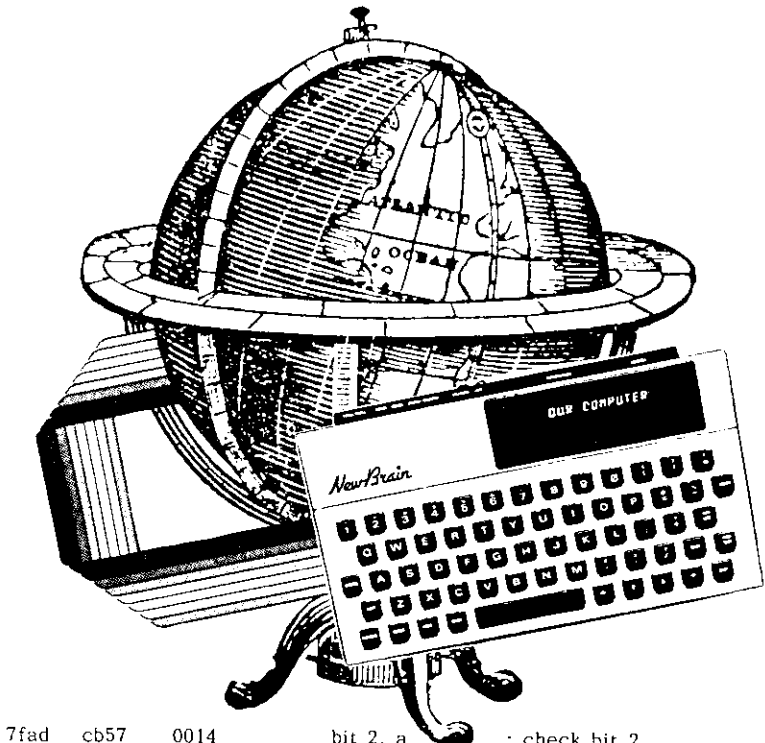
bleken te komen. Daar zaten heel wat fouten in. Wellicht heeft dat te maken met het feit, dat AMD intussen de veel meer geavanceerde AM7910 op de markt bracht. Ik ben diverse ontwerpen tegengekomen met dat IC. In Elektuur: de telektor, september 1984; in Wireless World: Multi-standard-Modem, mei-augustus 1984. Ook binnen onze gebruikersgroep leven diverse ideeën om zoiets zelf te bouwen. Ik heb op de HCC-dagen een Philips-modem gekocht, gebaseerd op ook weer diezelfde chip.

SOFTWARE

Hieronder zie je een listing van de software, zoals die nodig is om met de 'Velotype-interface' op een unexpanded NewBrain en een modem te kunnen communiceren met andere computers. Dit betreft alleen de standaard ASCII-code. Dat is voldoende om bijvoorbeeld met FIDO te communiceren. Of met andere computers. In het algemeen zal voor communicatie met andere computers een 'vertaaltabel' nodig zijn om alle 'control codes' (ASCII 0 - 31) op de goede manier om te zetten, zowel voor input als voor output. Ik ben bezig daarvoor wat uitgebreidere assembler routines op te zetten. Het doel ik daarbij om terminals van bijvoorbeeld Digital Equipment, Data General, IBM, Hewlett Packard te kunnen emuleren.

Theo Kamperman
Schipperdreef 65
3972 VB Driebergen

adres	code	regel	label	instructie	commentaar
		0001		org 7f9ah	; start is 32666
7f9a	1e06	0002	keyin:	ld e, 6	; stream #6
7f9c	e7	0003		rst 20h	
7f9d	36	0004		defb 36h	; stop key
7f9e	dad77f	0005		jp c, stop	
7fa1	e7	0006		rst 20h	
7fa2	31	0007		defb 31h	; input byte
7fa3	fe00	0008		cp 0	
7fa5	cab77f	0009		jp z, rstat	; no byte entered
7fa8	47	0010		ld b, a	; store byte in b
7fa9	1e42	0011	wstat:	ld e, 42h	; stream #66
7fab	e7	0012		rst 20h	
7fac	31	0013		defb 31h	; read write status



7fad	cb57	0014		bit 2, a	; check bit 2
7faf	caa97f	0015		jp z, wstat	; wait for bit = 1
7fb2	1e40	0016	trans:	ld e, 40h	; stream #64
7fb4	78	0017		ld a, b	; get byte from b
7fb5	e7	0018		rst 20h	
7fb6	30	0019		defb 30h	; tx byte
7fb7	1e42	0020	rstat:	ld e, 42h	; stream #66
7fb9	3e10	0021		ld a, 10h	; a = 16
7fbb	e7	0022		rst 20h	
7fbc	30	0023		defb 30h	; set cts
7fbd	3e00	0024		ld a, 0	
7fbf	e7	0025		rst 20h	
7fc0	30	0026		defb 30h	; select reg 0
7fc1	e7	0027		rst 20h	
7fc2	31	0028		defb 31h	; read dart status
7fc3	cb47	0029		bit 0, a	; check bit 0
7fc5	ca9a7f	0030		jp z, keyin	; no byte received
7fc8	1e40	0031	read:	ld e, 40h	; stream #64
7fca	e7	0032		rst 20h	
7fcb	31	0033		defb 31h	; read rx byte

```

7fcc 1e00 0034 disp: ld e, 0 ; stream #0
7fce e7 0035 rst 20h
7fcf 30 0036 defb 30h ; input to screen
7fd0 3e06 0037 ld a, 6
7fd2 e7 0038 rst 20h
7fd3 30 0039 defb 30h ; cursor on
7fd4 c39a7f 0040 jp keyin
7fd7 c9 0041 stop: ret ; return to basic

```

```

100 REM algemeen communicatieprogramma voor 300 baud, geen parity,
    2 stopbits
110 REM initialisatie: gebaseerd op tradecom's 'veloctype-interface'
120 REM routine in assembler begint op locatie 32666. geschreven door
130 REM ir theo kamperman, schippersdreef 65, 3972 VB driebergen
140 REM assembler routine mag niet voor commerciële doeleinden
    gebruikt worden!
150 FOR x = 1 TO 255 : CLOSE #x : NEXT x : OPEN #0, 0, "124" : PUT
    23, 1
160 FOR n = 64 TO 70 : OPEN #n, 7, n : NEXT n
200 REM *** opzetten van CTC volgens tradecom's recept
220 PUT #68, 69, 104 : PUT #69, 69, 4 : PUT #70, 69, 4
300 REM *** opzetten DART volgens tradecom's recept
310 PUT #66, 24, 76, 76, 1, 0, 3, 183, 5, 106, 16
400 REM *** eigen assemblerprogramma voor snelle communicatie
410 OPEN #6, 6 : RESERVE 100
420 m = 32666
430 READ a : IF a = 999 THEN 450
440 POKE m, a : m = m + 1 : GOTO 430
450 PRINT "typ <NL> als de telefoonverbinding tot stand is " : INPUT ("")
    a$
460 CALL 32666 : END
500 DATA 30, 6, 231, 54, 218, 215, 127, 231, 49, 254, 0, 202, 183, 127
510 DATA 71, 30, 66, 231, 49, 203, 87, 202, 169, 127, 30, 64, 120, 231
520 DATA 48, 30, 66, 62, 16, 231, 48, 62, 0, 231, 48, 231, 49, 203, 71
530 DATA 202, 154, 127, 30, 64, 231, 49, 30, 0, 231, 48, 62, 6, 231
540 DATA 48, 195, 154, 127, 201, 999

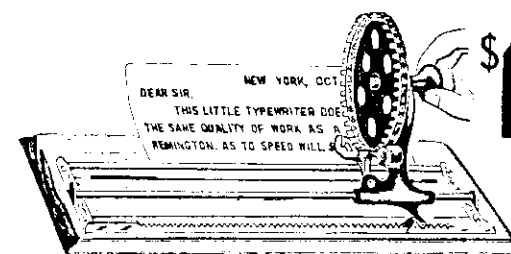
```

verkrijgbaar op cassette
zie pagina 18

texy2

MERGE TEKST MET 'TEXY2'

Het tekstverwerkingsprogramma 'TEXY2' van de firma Tradecom kent niet de mogelijkheid om een tekstfile, die is opgeslagen op cassette of diskette, toe te voegen aan het document, dat zich op een bepaald moment in het geheugen van de computer bevindt, zonder dat deze tekst verloren gaat. Met enkele wijzigingen en toevoegingen is het mogelijk om naar keuze een van de vijf pagina's van een opgeslagen tekstfile te MERGEN op pagina 5. Als er op dat moment tekst op pagina 5 staat, dan moet u die eerst kopiëren naar een andere pagina en verwijderen. Het programma antwoordt in het genoemde geval met KAN NIET.



Oude programmaregels:

```

1100 ON ERROR GOTO 31200 : ON BREAK GOTO 31100 : FOR i = 1 TO
15 : CLOSE #i : NEXT i : OPEN #0, 0 : OPEN #6, 6 : OPEN #5, 5 : GOSUB
8000 : ? "<control/w>d TEKST - TEKST - TEKST - TEKST" : PUT
22, 1, 7 : ? " 1 ... EDIT." : ? " 2 ... PRINTEN baudrate=" : bd$ : ? " 3 ...
LADEN van " : d$ : ? " 4 ... SAVE naar " : d$ : ? " 5 ... OPTIES." : ? " 6
... DEF. FUNKTIES" : ? " 7 ... EINDE"
1110 ...
1190 PUT 22, 3, 19, 7 : ? ... (deze regel is niet te listen door escape
shift/home na :REM)
1195 ...
1197 IF a < 1 OR a > 7 GOTO 1210
1200 ...
1210 ON a GOTO 100, 5000, 7000, 6000, 10000, 20000, 1300

```

Wijzigen in:

```
1100 ON ERROR GOTO 31200 : ON BREAK GOTO 31100 : FOR i = 1 TO
15 : CLOSE #1 : NEXT i : OPEN #0, 0 : OPEN #6, 6 : OPEN #5, 5 : GOSUB
8000 : PUT 23, 68 : ? " TEKST - TEKST - TEKST - TEKST - TEKST " :
PUT 22, 1, 7 : ? " 1 ... EDIT." : ? " 2 ... PRINTEN baudrate = "; bd$ : ? : ?
" 3 ... LADEN van "; d$ : ? " 4 ... SAVE naar "; d$ : ? : ? " 5 ... OPTIES." :
? " 6 ... DEF. FUNCTIES" : ? : ? " 7 ... MERGE van "; d$ : ? " 8 ...
EINDE"
1110 }ongewijzigd\
1190 PUT 22, 3, 20, 8 : ? "KIES CIJFER" : GET #5, a : a = a - 48
1195 }ongewijzigd\
1197 IF a < 1 OR a > 8 GOTO 1210
1200 }ongewijzigd\
1210 ON a GOTO 100, 5000, 7000, 6000, 10000, 20000, 7300, 1300
```

Toevoegen:

```
7200 IF t$(5) = "" RETURN
7210 GOTO 31200
7250 PUT 22, 1, 10, 30 : ? "Geef nummer pagina (1 t/m 5): "; : PUT 6 : GET
#5, i1 : PUT 7, i1 : IF i1 < 49 OR i1 > 53 GOTO 7200
7260 i1 = i1 - 48 : RETURN
7300 ? : ? "MERGE (naar pagina 5!)" : nn$ = nm$ : GOSUB 7100 : GOSUB
7200 : GOSUB 7250 : CLOSE #12 : OPENIN #12, dv, nm$ : INPUT #12, h :
INPUT #12, h : GET #12, h : LINPUT #12, h$ : FOR i = 0 TO 9 : LINPUT
#12, h$ : NEXT i : GET #12, h : FOR i = 1 TO i1 : LINPUT #12, t$(5) :
NEXT i : GET #12, i : nm$ = nn$ : h = 0 : h$ = "" : t = 5 : GOTO 1100
```

Regel 7300 kunt u verkrijgen door in regel 7000 (laden) de wijzigingen aan te brengen.

Gegevens die niet geladen moeten worden, worden in h of h\$ geplaatst. Kantlijn, gedefinieerde functies en dergelijke blijven dus ongewijzigd.

De te wijzigen en toe te voegen programmaregels staan onder de naam 'TXY2MG1.OVL' op de schijf of cassette. De aanwijzingen staan in de REM-regels.

verkrijgbaar op cassette
zie pagina 18

M. Ploemen

pascal

TIPS VOOR HISOFT-PASCAL

Door op toets B (van Basic) te drukken, komt de computer terug in Basic. HiSoft heeft helaas verzuimd aan te geven, hoe je daarna weer in Pascal komt. Er zijn twee manieren om in Pascal terug te keren, via de WARME start en via de KOUDE start.

Onder de WARME start wordt verstaan: terugkeer in Pascal, waarbij het programma behouden blijft. Deze start gebruik je bij voorbeeld, als je per ongeluk op de B hebt gedrukt. Je kunt het ook doen, als je tussendoor even iets wilt uitrekenen. Je drukt op toets B, de computer is in Basic. Je maakt je berekening en keert weer terug naar Pascal. Bij de KOUDE start is het Pascalprogramma verdwenen. Dit doe je alleen, als je helemaal opnieuw wilt beginnen.

voor de KOUDE start gebruik je CALL 6208

voor de WARME start gebruik je CALL 6213

Drie control-codes zijn in Pascal onbruikbaar. Het zijn de codes 8, 12 en 16. Deze codes gebruikt Pascal zelf voor de programmabesturing.

Een statement als WRITE ((CHR (22), CHR (8), CHR (12)) om de cursor op de 8e positie van regel 12 te plaatsen zal dus grandioos mislukken. Om deze bezwaren te ondervangen heb ik de procedure AT geschreven. Deze procedure wordt aangeroepen met 2 variabelen. Om de cursor op de 8e positie van regel 12 te zetten gebruik je AT (8, 12). Het volgende DEMO-programma toont het gebruik van de procedure AT.

```
PROGRAM DEMO; VAR A: INTEGER;
```

```
PROCEDURE AT (X, Y: INTEGER); VAR X1, Y1: INTEGER;
BEGIN
```

```
  X1 := 0; Y1 := 0;
```

```
  IF (X = 8) OR (X = 12) OR (X = 16) THEN X1 := 1;
```

```
  IF (Y = 8) OR (Y = 12) OR (Y = 16) THEN Y1 := 1;
```

```
  WRITE (CHR (22), CHR (X - X1), CHR (Y - Y1), CHR (X1 * 26),
        CHR (Y1 * 10))
```

```
END;
```

```

BEGIN
  PAGE;
  FOR A := 1 TO 23 DO
  BEGIN
    AT (A, A);
    WRITE (A)
  END
END.

```

HiSoft geeft in de 'Alteration Guide' (helaas niet meer verkrijgbaar), de volgende tips om de snelheid van het Pascalprogramma op te voeren:

1. FOR . . . DO-loops zijn veel sneller dan de overeenkomstige REPEAT of WHILE-loops.
2. Bewerkingen als opslaan en laden van variabelen, vergelijkingen, CASE . . . OF statements enz. zijn het snelst met niet-numerieke variabelen (CHARS en BOOLEANS). INTEGERS zijn langzamer, maar op hun beurt weer sneller dan REALS.
3. Programma's zijn sneller als de verschillende checks weggelaten worden. Dat wil zeggen, gebruik de compiler-opties O-, S-, I-, A- en C-. Zie sectie 3 van de handleiding.
4. CASE-statements zijn sneller dan de overeenkomstige IF-statements.
5. Voor REALS zijn de vergelijkingen '=' en '<>' sneller dan de andere vergelijkingen.
6. ARRAYS met een lage ondergrens (0 - 3) zijn sneller dan andere arrays. Een array met ondergrens 0 is weer sneller dan een met ondergrens 1 enz.
7. Lokale variabelen en parameters zijn sneller, als geïndexeerde adressering kan worden gebruikt, dat wil zeggen, als hun adressen liggen tussen (IX - 128) en (IX + 127). Zie appendix 3.2. van de handleiding.
8. Een functie is sneller dan de overeenkomstige procedure.
9. Het gebruik van globale variabelen is sneller dan lokale variabelen.
10. Een vermenigvuldiging is sneller dan de overeenkomstige deling.
11. PEEK en POKE zijn, in tegenstelling tot Basic, erg snel.
12. Bij bewerkingen (+, - of *) van een INTEGER getal (geen variabele), is het sneller om het getal op de 2e plaats te zetten. Dus $1 * 2$ is sneller dan $2 * 1$. Hieruit volgt dat $1 * 2$ ook sneller dan $1 + 1$.

HiSoft kent de mogelijkheid om een Pascalprogramma met het T-commando te save. Het programma wordt dan zo op de band gezet, dat het later te laden en te runnen is, zonder dat eerst de compiler geladen moet worden. Het programma kan geladen worden met dezelfde lader als waar-



mee de compiler geladen wordt.

Helaas werkt dit alleen met cassette, en niet vanaf een disk.

Bij de onderwijsbibliotheek zijn een aantal van dergelijke programma's binnengekomen, zodat ik ben gaan zoeken naar een methode om zulke programma's ook vanaf disk in te kunnen lezen.

Het probleem is, dat het laadprogramma in het onderste deel van het geheugen komt en dat dat, plus nog een deel van het Pascalprogramma zelf, wordt overschreven, zodra een diskstream wordt geopend. De oplossing die ik heb gevonden, bestaat eruit, dat de lader hoog in het geheugen wordt geplaatst en het Pascalprogramma zelf 5k hoger wordt geladen. Nadat de stream gesloten is, wordt het programma op zijn juiste plaats gezet. Op een expanded system werkt deze routine niet. Je zult dan eerst de machine 'unexpanded' moeten maken, bij voorbeeld met "U3.BAS".

Hoe is nu de volledige procedure?

Het Pascalprogramma wordt eerst op de normale manier met het T-commando naar cassette gesaved. Daarna wordt met een transferprogramma dit Pascalprogramma overgezet naar disk.

Vervolgens laad je onderstaand laadprogramma. Vergeet niet op regel 3 de

floating point

Ik heb met verschillende transferprogramma's geprobeerd om een T-file naar disk over te zetten, maar om een of andere reden lukt dat niet. Ik denk, dat het iets te maken heeft met buffergrootten. Met het onderstaande simpele transferprogramma lukt het probleemloos. Je moet alleen opletten, dat de goede namen op regel 2 staan ingevuld.

PROGRAMMA 'PASTRANS'

```
1 FOR i = 1 TO 255 : CLOSE #1 : NEXT i
2 OPENIN #1, 1, "*"6:NAAM . . ." : OPENOUT #2, 12, "NAAM . . ."
3 ON ERROR GOTO 5
4 GET #1, x : PUT #2, x : GOTO 4
5 IF ERRNO = 133 PUT #2, 4 : CLOSE #1 : CLOSE #2 : ? "Klaar" :
  END
6 PUT #2, 4 : CLOSE #1 : CLOSE #2 : ? "foutje . . ." : END
```

juiste naam van het Pascalprogramma te zetten.

Met RUN start je de lader. Automatisch wordt dan het Pascalprogramma geladen en gestart.

Wanneer het Pascalprogramma beëindigd is, wacht de computer, totdat op Newline is gedrukt. Dan wordt het scherm gewist en verschijnt het NewBrain-logo. Als je dat niet wilt, verander dan regel 10 in: CALL 6144 : END.

Het Pascalprogramma is iedere keer opnieuw te starten met RUN.

```
1 RESERVE TOP - 32673 - 65536 * (TOP < 32673 )
2 FOR I = 32673 TO 32762 : READ X : POKE I, X : NEXT I
3 OPEN #12, 12, "NAAM . . ."
4 CALL TOP
5 DATA 17, 12, 12, 6, 16, 231, 49, 254, 85, 216, 16, 249, 6, 8, 231,
  49, 216, 16, 251, 231, 49, 216, 50, 223, 127, 231, 49, 216, 61, 50,
  204, 127, 6, 103, 231, 49, 216, 16, 251, 33, 0, 44, 14, 0, 6, 128,
  231, 49, 216, 119, 35, 16, 249, 231, 54, 216, 231, 49, 13, 32, 239,
  6, 0, 231, 49, 216, 119, 35, 16, 249, 231, 52, 216, 17, 0, 24, 33,
  0, 44, 58, 204, 127, 71, 58, 223, 127, 79, 237, 176, 201
6 RESERVE 2 + 15 + TOP
7 DELETE -7
10 CALL 6144 : GET G$ : OPEN #0, 0 : CALL 49388
```

Pieter W. M. van Dijk

56

basic

newbrain on-line 7



Voor het onderzoeken van de INTeGer-bug van de NewBrain (zie elders in dit nummer) had ik een programma nodig om de floating-pointnotatie van de getallen te vinden. Een goed conversieprogramma bestond er niet, en dus heb ik op basis van de theorie van de floating-pointnotatie een conversieprogramma in BASIC geschreven. Dit werkte goed, zolang het 'normale' getallen betrof. Kwam ik in de buurt van het uiterste bereik van de NewBrain, dan liet dit programma het afweten. Voor de vraagstelling van de INTeGer-bug was dit voldoende. Om toch de uitersten te kunnen onderzoeken, heb ik aan het conversieprogramma een machinecode-routine toegevoegd, die de geheugenplaatsen, waar de (berekende) variabele A is opgeslagen, leest en op het scherm of de printer zet. Als invoer kunnen niet alleen gehele getallen, maar ook functies worden ingevoerd. Enkele voorbeelden zijn: 0, -0, 1e150, 1 / 256, 1 / 65536, 1e-150, 4 - 4, 4 * 4, PI, INT (10 * 3.2), enzovoort. Getallen groter dan 1e99 kunnen als volgt worden ingevoerd: 10000000000000000000e99.

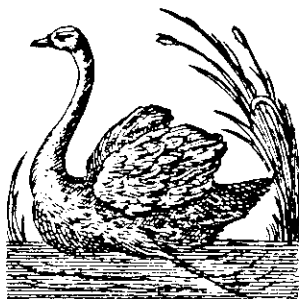
newbrain on-line 7

operating system

57

Enige theorie over de floating-pointnotatie en de praktische toepassing ervan bij populaire computers (inclusief de NewBrain) kun je vinden in: Maurice Hawes, Pointing the way, Personal Computer World (januari 1985), pp. 166 - 173.

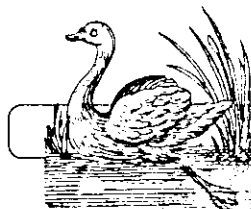
Verder kun je over de NewBrain informatie vinden in: Personal Computer World (juli 1984), p. 126.



Voor degenen die de theorie van floating-pointnotatie kennen:

- 1 de NewBrain gebruikt 6 bytes om een getal vast te leggen;
- 2 de NewBrain kent geen INTEGERS, dat wil zeggen numerieke variabelen, die in 1 of 2 bytes zijn vastgelegd;
- 3 byte 1 is de exponent en de bytes 2 tot en met 6 vormen de fractie;
- 4 het grondtal van de exponent is 16 (zie ook punt 6);
- 5 het grondtal van de fractiebytes is 256;
- 6 bit 0 van het exponentbyte geeft aan, of de exponent positief ($b_0 = 0$) of negatief is ($b_0 = 1$). Hierdoor neemt de exponent in stappen van 2 toe, waardoor het effectieve grondtal van de exponent 256 wordt;
- 7 bit 7 van het exponentbyte geeft aan, of het getal positief ($b_7 = 0$) of negatief is ($b_7 = 1$);
- 8 de nauwkeurigheid van de NewBrain is variabel en bedraagt meer dan 11 cijfers. Door het Operating System worden echter maar 10 cijfers gebruikt, dat wil zeggen, in- en uitvoer kan slechts met een nauwkeurigheid van 10 cijfers.

Wim Luijt



verkrijgbaar op cassette
zie pagina 18

Idcode



MACHINETAALLOADER VOOR MACHINETAAL

Machinecode, die als subroutine in BASIC is opgeslagen in een DATA-regel, moet aanvankelijk door een READ/POKE-lus ergens geladen worden (meestal boven TOP na RESERVE). De NewBrain doet dit, vind ik, ergerlijk langzaam, waarschijnlijk door omslachtig gedoe met floating-point-vertalingen en dergelijke.

In On-line is hierover al vaker geschreven (onder anderen door J. Koning met 'trickloader', On-line 3, pag. 48); kennelijk ben ik niet de enige ongeduldige. Een 'relocator' in machinetaal is als mogelijke oplossing genoemd. Zo'n machinetaal-relocator is aantrekkelijk vanwege de volgende eigenschappen:

- Nauwelijks uitbreiding van bestaand BASIC-programma (mits kort!)
- Oude manier van cassetteopslag blijft gehandhaafd, zodat bestaande BASIC-programma's met machinetaalroutine in DATA-regels eenvoudig aangepast kunnen worden.
- Snelle omzetting van decimale DATA naar machinetaalbytes op een vrij te kiezen stuk RAM (gewoonlijk boven TOP).

Een dergelijke dataloader, vanaf nu loader genoemd, wil ik hier presenteren met, voor de liefhebbers, een korte toelichting.

De 'READ DATA / POKE DATA'-lus kan DELETED worden.

MERGE de volgende passages:

```
90 REM laad loader op normale wijze.
100 DATA 86, 43, 94, 235, 231, 39, 221, 110, 8, 221, 102, 9, 14, 0, 126,
      254, 44, 40, 20, 254, 42, 200, 214, 48, 254, 10, 48, 16, 71, 121, 7, 79,
      7, 7, 129, 128, 79, 24, 5, 121, 18, 19, 14, 0, 35, 24, 223
```

```

110 rt = 180 : IF PEEK (51) > 127 THEN rt = PEEK (168) + 256 * PEEK
    (169) : POKE 168, (rt + 47) AND 255 : POKE 169, INT ((rt + 47) / 256)
120 RESTORE 100 : FOR i = rt TO rt + 47 : READ by : POKE i, by :
    NEXT i
...
150 CALL rt, 32000
160 DATA ... <machinecode in de vorm van decimale getallen> ...

```

Regel 100 bevat de loader, die in regel 120 op de normale manier gePOKEd wordt (duurt kort, slechts 47 bytes). In regel 110 heeft de computer gekeken, of hij expanded is, en zo ja, een geschikte plaats uitgezocht voor de loader *).

In regel 150 wordt deze loader geactiveerd, waardoor de DATA van regel 160 (het te laden machinetaalprogramma) vanaf locatie 32000 (boven TOP) geplaatst worden.

Het is essentieel voor de werking, dat de opdracht 'CALL rt, 32000' ONMIDDELIJK vooraf gaat aan de regel met machinecode-DATA. Dus in bovenstaand voorbeeld GEEN regels tussen 150 en 160. De reden hiervan zal uit het volgende blijken.

ACHTERGRONDEN MACHINETAALLOADER

We houden bovenstaande passage in BASIC nog steeds aan. Het Z80-register IX houdt tijdens het RUNnen van een BASIC-programma bij, welk regelnummer op ieder moment uitgevoerd wordt. IX doet dit door te wijzen naar de entry in de line number table behorende bij het regelnummer-invoering. Zo'n entry bestaat uit 6 bytes, te weten:

- 2 bytes linenummer (hex)
- 2 bytes pointer naar de sourcecode van die regel
- 2 bytes idem naar object code

Interessant voor ons doel zijn de middelste twee bytes, omdat deze direct aangeven waar de tekst ergens in het geheugen te vinden is.

Stel, regel 150 wordt geëxecuteerd, de CALL-opdracht dus. IX wijst dan naar de plaats in het geheugen, waar de tekst 'CALL ...' staat. Omdat in de line number table de entries van de regelnummers op volgorde van grootte staan (ongeacht de volgorde van intypen), weten we ZEKER, dat de

*) aan het einde van de system page (adressen 0 - 8191) heeft de expanded newbrain nogal wat vrije ruimte. op adres 168-169 staat, waar het eerste vrije byte te vinden is, en waar dus nieuwe routines toegevoegd kunnen worden, als de routine toegevoegd is, wordt het adres in 168-169 aangepast (red.)

tekst van regel 160 begint op het adres, dat wordt aangegeven door IX + 6, IX + 7. We moeten de middelste 2 bytes van deze entry hebben, dus IX + 8 en IX + 9 vormen een pointer naar de sourcecode van regel 160. Met andere woorden: we hebben het beginadres gevonden van onze te verplaatsen machinecode!

De machinetaalloader speelt hier op in door vanaf het adres, aangegeven door IX + 8, IX + 9, de decimale DATA in te lezen, om te zetten in bytes en deze tenslotte op de gewenste adressen in het boven TOP reseserveerde stuk RAM te plaatsen.

Interessant is, dat het keyword DATA nu ook bij voorbeeld REM mag zijn, immers we gebruiken geen RESTORE en READ.

Belangrijk is, dat de gehele te verplaatsen code in één DATA-regel is samengeperst. Dit moet dus eventueel bij bestaande programma's worden aangepast, bij voorbeeld met de hand door GRAPHICS / PIJL OMLAAG, of, als dit te gortig wordt, met behulp van een daartoe geschreven BASIC-routine *). Het laatste DATA-element wordt verondersteld een asterisk (*)



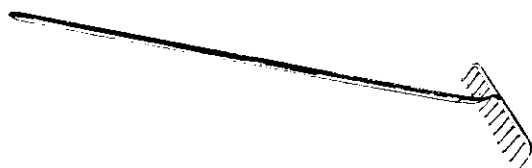
*) Een manier is: OPEN een tweede pagina, READ DATA, PRINT deze alle op de tweede pagina achter één regelnummer, MERGE de tweede pagina, DELETE de oorspronkelijke DATA.

te zijn. Hierop detecteert de loader het einde van de DATA.

De parameter bij de CALL-opdracht (bij voorbeeld 32000) MOET meegegeven worden. In foutendetectie is niet voorzien terwille van de compactheid. Bij mijn langste machinetaalprogramma ('Monitor' van J. Braga) werkt het perfect - meer dan 500 bytes - dus waarom zou het niet werken bij duizenden bytes?

Hopenlijk bewijst deze loader bij deze of gene zijn diensten.

Frans van der Velden



LD.CODE.ZSM

een routine om machinecode, decimaal opgeslagen in een DATA-regel, na binaïromzetting te verplaatsen naar een gebied waarvan het beginadres als parameter moet worden meegegeven.
(c) maart 1986, fr. van der velden

	org 180		;routine is vrij verplaatsbaar
			;locaties 121 t/m 255 zijn niet in
			;gebruik bij standaard-newbrain
00b4	56	ld d, (hl)	;adres van meegegeven parameter
00b5	2b	dec hl	;in
00b6	5e	ld e, (hl)	;registerpaar de
00b7	eb	ex de, hl	;nu in hl
00b8	e7	rst 20h	;roep operating system
00b9	27	defb 27h	;zet parameter binair om in de

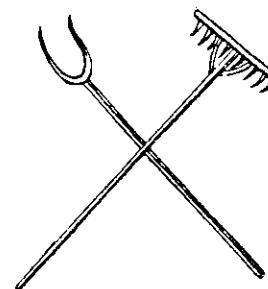
de wijst nu naar het door de gebruiker gegeven bestemmingsadres

00ba	dd6e08	ld l, (ix + 8)	;pointer naar source code
00bd	dd6609	ld h, (ix + 9)	;van data-regel in hl
00c0	0e00	ld c, 0	;reset c
00c2	7e	karak ld a, (hl)	;karakter in a
00c3	fe2c	cp ',',	;is dit een komma?
00c5	2814	jr z, numb	;ja, dan getal opbergen

00c7	fe2a	cp '*',	;is het een asterisk?
00c9	c8	ret z	;ja, dan terug naar basic
00ca	d630	sub '0'	;ontASCII het karakter
00cc	fe0a	cp 10	;is karakter numeriek?
00ce	3010	jr nc, next	;carry = 0: spatie of token

hier begint omzetting decimaal naar binair

00d0	47	digit	ld b, a	;digit naar b
00d1	79		ld a, c	;vorige digit naar a
00d2	07		rlca	;maal 2
00d3	4f		ld c, a	;naar c
00d4	07		rlca	;maal 4
00d5	07		rlca	;maal 8
00d6	81		add a, c	;maal 10
00d7	80		add a, b	;vorige digit * 10 + digit
00d8	4f		ld c, a	;naar c
00d9	1805		jr next	;haal volgend karakter op
00db	79	numb	ld a, c	;byte naar a
00dc	12		ld (de), a	;berg byte op
00dd	13		inc de	;hoog bestemmingspointer op
00de	0e00		ld c, 0	;reset byte
00e0	23	next	inc hl	;hoog bronpointer op
00e1	18df		jr karak	;lees-nieuw karakter in
00e3			end	



verkrijgbaar op cassette
zie pagina 18

Computerhobbyisten opgelet!

Speciale aanbieding:

-forse metalen dual-drive kasten 30x30x12,5cm met ventilator,
netentree met filter, aan/uit schakelaar, D25 en D15
connectors en diverse andere onderdelen.
als pakket F 275,-

-disk-drives voor deze kasten:

200k Teac ss/sd/40tr F 295,-
800k Mitsubishi ds/dd/80tr F 375,-

-gebruikte printerterminals v.a. F 500,-

-gebruikte Datamedia DT80 beeldschermterminals
(VT-100 compatible) F 150,-

-printers:

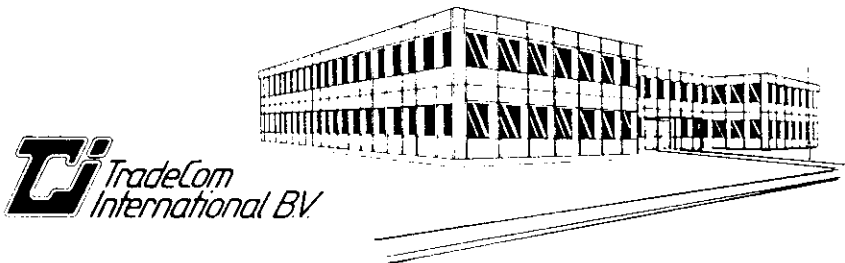
Riteman matrixprinter 120 cps F 800,-
Microline 82-A 120 tekens ps F 690,-
-modems MI1200/75 V-23 F 250,-
-netfilters FR22z (Schaffner) F 135,-

-voor de newbrain:

dual-drives 2x200k compleet F 1600,-
dual-drives 2x800k met schakelende
voeding (50W) en controller F 2550,-
deze kast is ook leverbaar met ingebouwde
expansion unit en bovendien kan men uit
deze kast de newbrain voeden.

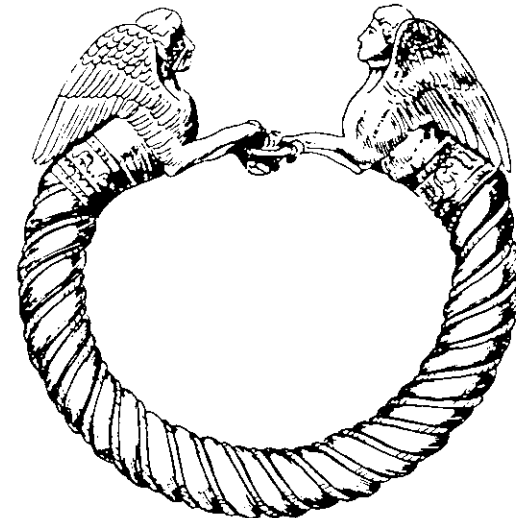
Prijzen zijn exclusief B.T.W., afgehaald magazijn:
Hoogewaard 209
Koudekerk a/d Rijn.
Tel: 01714-4300

Wijzigingen voorbehouden en zolang de
voorraad strekt.



call

DE CALL-PARAMETERLIST
OFWEL DE BASIC-MACHINETAALINTERFACE



In appendix 8 van het handboek wordt kort uitgelegd, hoe men constanten of basic-variabelen aan een machinetaalroutine kan meegeven. Hier zal een en ander iets concreter weergegeven worden.

Na het uitvoeren van de instructie

CALL adres, a, b(1), x\$, 12, y\$(14, 2)

zet je basic opzij en moet je de parameters zelf 'begeleiden'. Het HL-register wijst nu naar het laatste byte van een tabel, die de informatie bevat, waar de parameters te vinden zijn. In het B-register staat het aantal parameters, niet in C, zoals het handboek vermeldt.

Het lokaliseren van de parameters kan als volgt gedaan worden.

```

GETNUM: LD BC,(PARMLEN)
        XOR A
        CP B
        SCF
        RET Z
        DEC B
        LD (PARMLEN), BC
        LD HL, (PARMLIST)
        DEC HL
        LD A, (HL)
        DEC HL
        LD (PARMLIST), HL
        LD L, (HL)
        LD H, A
        XOR A
        RET

```

HL wijst naar getal in floating-pointnotatie of naar stringdescriptor

De eerste instructies van de aangeroepen routine (vanaf basic) moeten luiden:

```

INC HL
LD (PARMLIST), HL
LD (PARMLEN), BC

```

Verder moeten de adressen PARMLIST en PARMLEN gedefinieerd worden. Met CALL GETNUM of GETSTR worden de numerieke of stringparameters door HL aangewezen. Indien een van deze subroutines met carry-set returnt, dan is de parameter tabel leeg of doorlopen (ook bruikbaar als einde van een loop, indien er meerdere parameters opgehaald moeten worden). Let erop, dat bij het lokaliseren van een string het IY-register dezelfde waarde moet hebben, als bij het verlaten van basic. Met de parameteroverdracht en de restarts vervaagt de grens tussen basic en machinetaal en kunnen op eenvoudige wijze onder andere userfuncties gemaakt worden.

Frans Horstink



```

GETSTR: CALL GETNUM
        RET C
        LD E, (HL)
        INC HL
        LD D, (HL)
        INC HL
        LD C, (HL)
        INC HL
        LD B, (HL)
        LD L, (IY + 26)
        LD H, (IY + 27)
        ADD HL, DE
        XOR A
        RET

```

HL wijst naar begin string,
BC = stringlengte

cp/m

INLEIDING

CP/M staat voor control program microcomputers en is ontwikkeld door Digital Research. Het is een zogenaamd operating system (O.S.). Het operating system is het programma, dat altijd aanwezig moet zijn in een computer om ervoor te zorgen, dat de computer iets zinvol doet. Het operating sytem zorgt ervoor, dat de computer iets naar een beeldscherm kan sturen, een karakter van het toetsenbord kan binnenhalen, een bestand van disk of tape kan lezen en een bestand naar disk of tape kan schrijven. Kortom het operating system zorgt voor alle basisroutines van een computer.

OPBOUW VAN CP/M

CP/M is een O.S. voor microcomputers met een 8080-, 8085- of Z80-microprocessor en een of meer diskdrives. Het CP/M operating system bestaat uit drie delen, namelijk de console command processor (CCP), het basic disk operating system (BDOS), en het basic input / output system (BIOS).

DE DISK

Voordat we verder gaan met de uitleg van het operating system, eerst iets over de disk. Een disk bestaat uit een schijfje met daarop een magnetische laag, waarop we informatie kunnen schrijven. De schijven kunnen op twee manieren worden beschreven, in een dubbele (DD) en een enkele (SD) dichtheidsmode. Meestal worden de schijven tegenwoordig in de DD-mode gebruikt. Om een disk te gebruiken wordt hij onderverdeeld in tracks. Dit zijn concentrische cirkels, waarvan er zich 40 of 80 aan een van de kanten

of aan beide kanten bevinden. Deze tracks zijn weer onderverdeeld in fysische sectoren, meestal in 10 sectoren van 512 bytes of 5 sectoren van 1024 bytes. Dit verschilt per computer, maar er gaat bij de DD-mode altijd 5 kbyte op een track. Om te bepalen, wat de eerste sector van een track is, zit vlak bij het grote gat midden in de schijf een klein gaatje (index hole). De diskdrive stuurt hierdoor (infrarood) licht en meet het moment, waarop het gaatje langs komt.

De hier beschreven diskindeling is de meest gebruikte maar er zijn uitzonderingen (Apple 2).

HET BIOS

Het BIOS is het gedeelte van CP/M, dat het dichtst bij de computer staat, en verschilt ook per computer. Voor de gebruiker en de applicatieprogrammeur is ieder BIOS wat werking betreft gelijk; de verschillen zijn alleen intern en de eerste 15 startpunten van ieder BIOS zijn gelijk aan elkaar. Het BIOS zorgt voor de meest essentiële handelingen van de computer, zoals een bepaald aangevoerd karakter op het beeldscherm plaatsen, naar de printer of een communicatiepoort sturen, en een karakter van een toetsenbord of communicatiepoort lezen. Ook kunnen we het BIOS opdracht te geven om vanaf een door ons opgegeven plaats op de schijf een bepaald blok informatie te lezen, of om naar een bepaalde plaats op schijf een blok informatie te schrijven.

Het BIOS zorgt ook voor het binnenlezen van het BDOS en de CCP vanaf schijf (dit heet de warm boot). CCP, BDOS en BIOS staan altijd op de eerste tracks van de schijf (meestal nemen ze samen twee tracks in beslag). Het BIOS zelf wordt door een programma in ROM ingelezen vanaf schijf, of wordt gekopieerd vanuit ROM-geheugen naar RAM. Dit proces heet de cold boot. De tracks waarop het operating system staat, worden de sytem tracks genoemd.

HET BDOS

Het BDOS is voor iedere computer gelijk en zorgt hoofdzakelijk voor de administratie van de bestanden (files) op schijf. Voor het BDOS bestaat een file uit een verzameling van logische sectoren (CP/M-sectoren) van 128 bytes. 8, 16, 32, 64 of 128 CP/M-sectoren vormen samen een block. Het BDOS houdt op de directory (= inhoud-) tracks bij, welke blocks door een bepaalde file worden gebruikt. De directory tracks zijn de tracks die na de system track komen. De blocks liggen na elkaar op de schijf, waarbij block 0 het block is, dat begint op de eerste directory track.

DISK ACCESS

Als de gebruiker een file wil gebruiken, kan hij opgeven aan het BDOS, welke file hij wil gebruiken, en daarna kan hij aan het BDOS opgeven, welke sector hij wil lezen of schrijven. Dit kan steeds de volgende zijn (read sequential, write sequential), of een door de gebruiker aan te wijzen sector (random access). Voordat de gebruiker een sector gaat lezen of schrijven, moet hij aangeven, naar welk stuk geheugen de sector moet worden gekopieerd, of waarvan de te schrijven sector komt (set dma). Dit adres wordt direct doorgegeven aan het BIOS.



Het BDOS gebruikt de informatie, die in de directory tracks staat, om te berekenen, in welk block de gevraagde sector bestaat. Hieruit wordt berekend, op welke track de sector ligt, en welke 128 bytes sector van die track we willen hebben. Als het om een schrijfpdracht gaat voor een sector van een file die nog niet bestaat, wordt door het BDOS een nieuwe sector in gebruik genomen voor de file. Dit wordt ook weer bij gehouden in de directory. Deze informatie over track en sector wordt doorgegeven aan het BIOS. Het BIOS berekent uit de gevraagde logische sector (van 128 bytes), in welke fysische sector deze sector ligt. In het BIOS is een ruimte gereserveerd, waarin een fysische sector past (diskbuffer). Als de data uit de juiste sector (nog) niet in de diskbuffer aanwezig zijn, wordt de

informatie over de gevraagde track en fysische sector doorgegeven aan de floppy-diskcontroller (FDC). Dan krijgt de FDC opdracht om de fysische sector van schijf te lezen. De data uit deze sector komen dan in de diskbuffer.

Nu staat de gewenste fysische sector in de diskbuffer. Als er een leesopdracht was, worden de data van de gevraagde logische sector uit de diskbuffer gekopieerd naar de met 'set dma' gevraagde geheugenlokatie en zijn we klaar. Als er een schrijfoperatie was, worden de te schrijven data uit het geheugen gekopieerd naar de juiste plaats in de diskbuffer. De FDC krijgt dan opdracht dezelfde track en fysische sector, die hij zojuist heeft gelezen, weer te beschrijven. De data uit de diskbuffer worden dan weer op schijf gezet. Zo wordt dus van de 5 of 10 CP/M-sectoren, waaruit een fysische sector bestaat, een CP/M-sector met nieuwe data beschreven. In de volgende On-line(s) zal ik uitleggen, hoe de directory is opgebouwd, en hoe het berekenen van sectoren en tracks in zijn werk gaat.

DE CCP

De CCP is het gedeelte van CP/M, waar we als gebruiker direct iets van zien. Het is namelijk het deel van CP/M, dat zorgt voor de 'A>' en waaraan we direct opdrachten kunnen geven. Als de CCP de opdracht herkent, voert hij hem direct uit. Dit geldt voor opdrachten zoals DIR (= inhoud van schijf tonen) en TYPE filenaam.ext (= geef op het scherm de inhoud van een bestand weer). Als de CCP een opdracht niet herkent, dan gaat de CCP kijken, of er een bestand '<opdracht>.com' op schijf staat. Zo nee, dan drukt de CCP de opdracht af met een vraagteken erachter om aan te duiden, dat hij de opdracht niet kent en er ook geen bestand is met de naam '<opdracht>.com'. Als er wel een bestand is met de naam '<opdracht>.com', dan neemt de CCP aan, dat dit bestand een programma is, leest het van de schijf en plaatst het begin van het bestand op adres 256 in het geheugen. Als het bestand helemaal gelezen is, roept de CCP het programma aan, beginnende op adres 256. Als het geen programma was, hangt de computer nu waarschijnlijk. Het programma loopt nu en kan gebruik maken van het BDOS en het BIOS. De CCP is nu niet meer van belang en het geheugengedeelte, waar de CCP staat, mag worden gebruikt door het programma. Als het programma klaar is, kan het teruggaan naar de CCP, als die niet vernietigd is, of naar adres 0, vanwaar het BIOS wordt aangeroepen, dat zorgt voor een warm boot.

Henk Blik



wp(demo?)

in dit nummer van newbrain on-line de derde aflevering van het WP-feuilleton. deze keer worden methoden beschreven om te redden, wat er te redden valt, als u ontdekt, dat u terug bent in de command mode van cp/m.

soms is het eigen schuld (het is maar al te gemakkelijk de Q aan te slaan, voordat de tekst op schijf staat), maar als de toetsenbordbuffer of het geheugen te vol wordt, raakt WP de kluts kwijt, en dan kan het niets beters verzinnen dan een warm boot. ook cp/m kan de oorzaak zijn: na het inzetten van een schijf moet <control/c> gegeven worden. in WP gebeurt dit met de opdracht ?a of ?b



WPREDDER

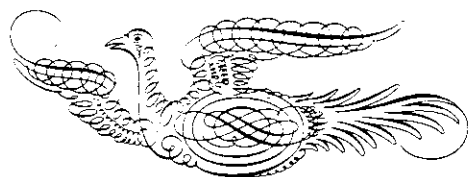
als het programma WP gestart wordt, voert het eerst allerlei initialisaties uit, voordat het voor het eerst met 'Command?' om een opdracht vraagt. dan is het op adres 225 aangeland, waar het ook steeds naar terugkeert, als het klaar is met het uitvoeren van een opdracht. het zou prettig zijn, als je dat adres ook rechtstreeks kunt bereiken, als je tegen je zin uit het programma geraakt bent. daarvoor heb ik de volgende methode bedacht

u maakt (eenmalig), voordat er ook maar een vuiltje aan de lucht is, een overlay van het begin van het programma, zoals dat eruit ziet tijdens de uitvoering, met als bijzonderheid, dat het programma meteen naar adres 225 springt. de procedure is als volgt

- 1 start WP
- 2 ga er weer uit (met Q Q)
- 3 SAVE 2 WPREDDER.COM
- 4 laad DDT WPREDDER.COM
- 5 geef de opdracht tot assembleren op adres 100: -a100
- 6 vul in JMP 225 (of JP 225 bij ZSID)
- 7 verlaat het programma: ↑C
- 8 weer SAVE 2 WPREDDER.COM

deze overlay met de naam WPREDDER.COM laat u op de schijf staan, waar ook WP.COM op staat. als u nu weer eens uit het programma WP gegooit bent (of per ongeluk de opdracht 'Q Q' gegeven hebt), laadt u wpredder en bent u terug in het programma, als u er niet uit geweest bent: de oude tekst is er nog

menno stevens



WPDUMP - WPHERST

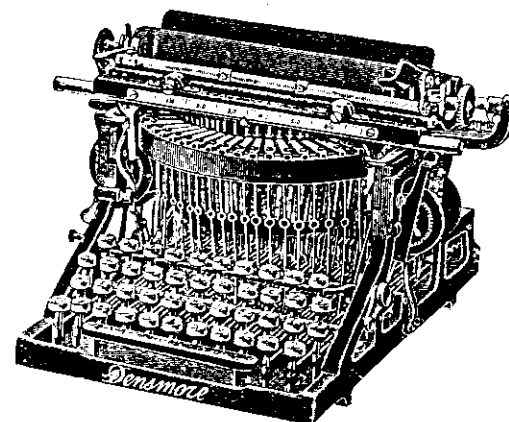
Bij het werken met WP in combinatie met (een aangepaste versie van) FAST.COM overkomt het me nog wel eens, dat ik uit WP gegooit word, voordat de tekst op schijf staat. Door een fout van mij lukt het me soms niet naar schijf te schrijven en word ik ook uit WP gegooit. WP kan bij voorbeeld niet schrijven op een schijf, waar nog geen file op staat. Ik zal maar niet neerschrijven, wat ik in het verleden in zo'n geval zei. Nu heb ik een methode gevonden, waarmee de tekst nog gered kan worden, mits de NewBrain niet op tilt is gegaan.

- 1 als de prompt A> (of B>) van cp/m verschijnt, type:
A> SAVE 100 WPDUMP en geef NewLine
- 2 wacht tot de prompt verschijnt; de tekst staat nu weer op schijf onder de naam WPDUMP
- 3 start DDT WPDUMP of ZSID WPDUMP
- 4 let op, welke waarde onder NEXT komt te staan. Stel dat dit XXXX is.
- 5 geeft de opdracht:
-m37CA XXXX 100 (bij ZSID komt de # als prompt)
- 6 zoek met de opdracht D, wat het laatste adres is, dat nog ingetypte tekst bevat. Stel dat dit YYYZ is
- 7 voer nu de volgende opdrachten uit:
-SYYYZ NewLine
-?? 0 NewLine
-?? 1A NewLine
-?? . NewLine

- 8 deel YYYZ door 100h. Denk erom, dat dit een hexadecimaal getal is. Neem dus YY en maak daar een decimaal getal van (DDD)
- 9 geeft Control/C; de prompt A> verschijnt weer. Type nu:
SAVE DDD WPHERST NewLine
- 10 begin weer met WP WPHERST

Deze methode werkt waarschijnlijk ook bij andere tekstverwerkers, alleen zullen de getallen dan moeten worden aangepast.

Wim Luijt



PRINTEN VIA COMMS-UITGANG

cp/m-programma's sturen in het algemeen de tekst, die met een printer afgedrukt moet worden, naar het list-device. maar als je je printer om de een of andere reden aan de comms-poort (device 17) hebt hangen, wil je liever, dat de data daarheen gestuurd worden. daartoe zijn dan twee mogelijkheden: of de cp/m-configuratie veranderen, of het programma zo maken, dat gegevens voor de printer naar het punch-device gaan. bij WP is het heel gemakkelijk te wijzigen: verander op adres 2ac6 de 05 in een 04

NewBrain-
gebruikersgroep
cp/m-bibliotheek

catalogus- diskette

de cp/m-bibliotheek van de newbrain-gebruikersgroep is nu gemakkelijk toegankelijk gemaakt: de inhoudsopgaven (met de commentaarregels) van de rond 180 voorradige en direct leverbare volumes uit de programmatheek van de hcc cp/m-gebruikersgroep zijn op diskette bij elkaar gezet.

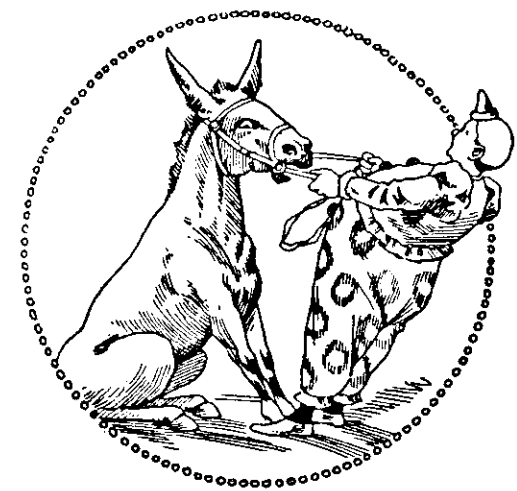
u laat de hele catalogus zo over het scherm rollen; en u kunt er alle zoekprogramma's en tekstverwerkers op los laten, die u hebt.

speciale prijs f 10,-

de catalogus-diskette is verkrijgbaar voor de speciale prijs van f 10,- (voor elk formaat!), en te bestellen op de gebruikelijke manier (zie pagina 6 van dit nummer van newbrain on-line).

NewBrain-
gebruikersgroep
postbus 4494
1009 AL amsterdam

fast + exit



bij de beschrijving van zijn programma FAST (in on-line 5, pp. 87 -91) noemt henk blik het een klein probleempje, dat je onder FAST niet meer van cp/m naar basic kunt gaan. maar wie zijn newbrain niet wil degraderen tot een ordinaire cp/m-machine, heeft er wel meer last van!

henk blik heeft nu het programma EXIT.COM zo aangepast, dat het eerst FAST buiten werking stelt en dan cp/m verlaat. de nieuwe versie, FEXIT.COM genaamd, werkt op expanded én op unexpanded systemen, met én zonder FAST. ze is verkrijgbaar bij de cp/m-bibliotheek van de gebruikersgroep op volume nbgg-5. voor de doe-het-zelvers volgt hier de source code

		0001	org 100h
0100	3a0700	0002	ld a, (7)
0103	07	0003	rlca ;test if paged!
0104	3804	0004	jr c, paged
0106	f3	0005	di
0107	c300e0	0006	jp 0e000h
		0007	paged:
		0008	cpm0 equ 0061h
010a	110020	0009	ld de, 2000h
010d	212401	0010	ld hl, rout

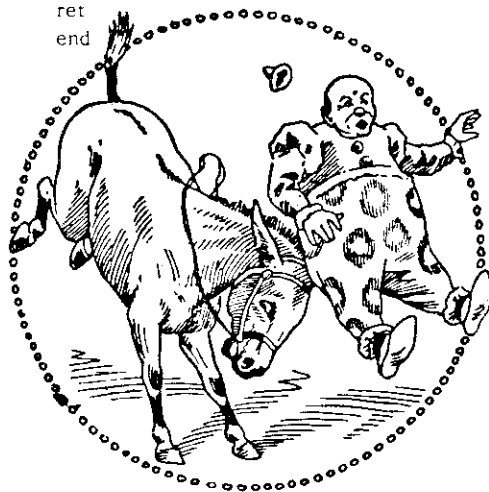
```

0110 012000 0011      ld bc, 20h
0113 edb0  0012      ldir
0115 cd0020 0013      call 2000h
0118 2a0100 0014      ld hl, (0001)
011b 113700 0015      ld de, 3ah - 3
011e 19 0016          add hl, de
011f cd2301 0017      call callhl
0122 4a 0018          defb 4ah ;code for zexit
0123 e9 0019 callhl:  jp (hl)
                        0020 rout: ;set irq vector back to
                                ;newbrain int routine
                                ;s0, pagereg

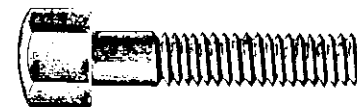
0124 010200 0021      ld bc, 0002 ;s0, pagereg
0127 2e60 0022      ld l, 60h
0129 f3 0023          di
012a ed69 0024          out (c), l ;system page 0
012c 210217 0025      ld hl, 1702h ;newbrain interrupt routine
012f 223900 0026      ld (39h), hl
0132 210619 0027      ld hl, 1906h ;end of nb sytem routines
                                ;in sysp0

0135 22a800 0028      ld (00a8h), hl
0138 216100 0029      ld hl, cpm0
013b 44 0030          ld b, h
013c ed69 0031          out (c), l
013e fb 0032          ei
013f c9 0033          ret
                        0034 end

```



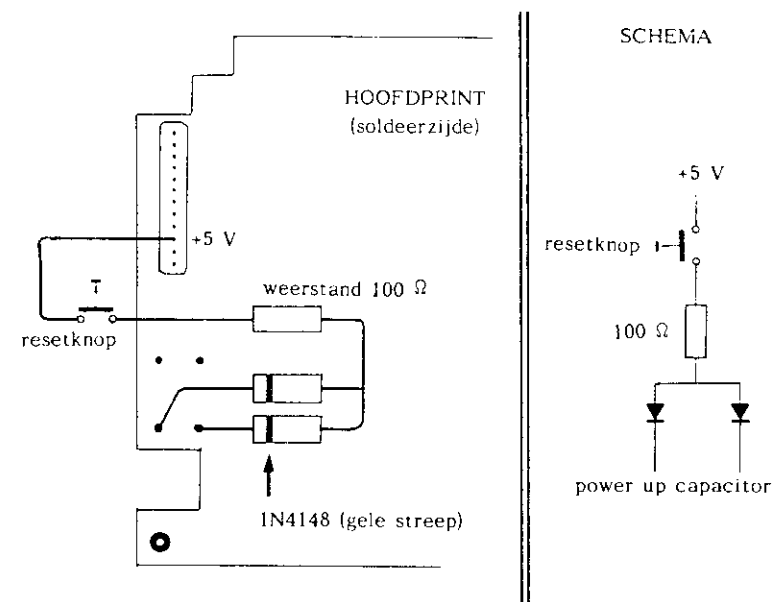
resetknop



POWER UP RESETKNOP

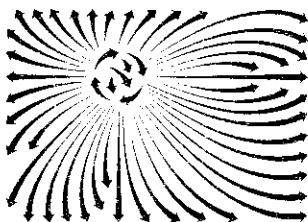
Hieronder volgt een tekening van opbouw voor een resetknop voor de NewBrain. Het geheel is simpel en gedraagt zich exact als ware de NewBrain opnieuw ingeschakeld.

Rob Maris



inhoud on-line 7

- 2 van de voorzitter /m s vreedenburg/
- 4 lijst van medewerkers
- 6 bestelprocedure softwarebibliotheken
- 7 nieuws van de softwarebibliotheek /wim luijt/
- 13 herziene uitgaven van de softwarebibliotheek /wim luijt/
- 17 catalogus van de nieuwe cassettes in de softwarebibliotheek
- 20 cp/m-bibliotheek: catalogus nbgg-serie
- 23 onderwijsbibliotheek: schijf 6
- 25 prijsvraag
- 29 vraagjes en weetjes - vragen en weeten
- roetsdetectie /pieter w m van dijk/
- sneller met savf /rob maris/
- gebruik van basicode /a moreels/
- assembler z80brain /frans horstink/
- 31 langzame stringbewerkingen in basic /rob maris/
- 33 integer-bug /wim luijt/
- 35 basic symbolic source code converter /rob maris/
- 41 cassette: hoe leest de newbrain bestanden van cassette en welke fouten
ziet hij daarbij /wim luijt/
- 47 de unexpanded newbrain als terminal /theo kamperman/
- 51 merge tekst met 'taxy2' /m ploemen/
- 53 tips voor hisoft-pascal /pieter w m van dijk/
- 57 floating point /wim luijt/
- 59 machinetaalloader voor machinetaal /frans van der velden/
- 65 de call-parameterlist, ofwel de basic-machinetaalinterface /frans
horstink/
- 67 cp/m {deel 1} /henk blik/
- 71 wpredder /menno stevens/ | wpdump-wpherst /wim luijt/
- 75 fast + exit
- 77 resetknop /rob maris/
- 79 bestelformulieren softwarebibliotheken



78

inhoud

newbrain on-line 7

NewBrain-
gebruikersgroep
postbus 4494
1009 AL amsterdam

softwarebibliotheek

BESTELFORMULIER

naam

adres

postcode en woonplaats

telefoon

hcc-lidmaatschapsnummer

bestelt bij deze de aan ommezijde aangegeven software

het totaalbedrag ad f

☐ is overgemaakt d d

op postrek 2505800 tnv hcc newbrain-gebruikersgroep utrecht

afkomstig van bank-/postrek nr

tnv te

☐ wordt voldaan met bijgesloten girobetaalkaart,
betaal- of eurocheque

nummer giropas of betaalpas

handtekening

NewBrain-
gebruikersgroep
postbus 4494
1009 AL amsterdam

cp/m-bibliotheek

BESTELFORMULIER

naam

adres

postcode en woonplaats

telefoon

hcc-lidmaatschapsnummer

bestelt bij deze de aan ommezijde aangegeven volumes

het totaalbedrag ad f

☐ is overgemaakt d d

op postrek 2505800 tnv hcc newbrain-gebruikersgroep utrecht

afkomstig van bank-/postrek nr

tnv te

☐ wordt voldaan met bijgesloten girobetaalkaart,
betaal- of eurocheque

nummer giropas of betaalpas

handtekening

NewBrain
gebruikersgroep

naam _____

straat _____

postcode en woonplaats _____

telefoon _____

computerconfiguratie _____

- ☐ geeft zich op als lid van de HCC en de NewBrain-gebruikersgroep
- ☐ is al lid van de HCC, lidmaatschapsnummer _____
en geeft zich op als lid van de NewBrain-gebruikersgroep
- ☐ geeft zich op voor regelmatige toezending van NEWBRAIN ON-LINE
beëindiging van dit abonnement geschiedt door schriftelijke opzegging

datum _____

handtekening _____

AANMELDINGSKAART



NewBrain
gebruikersgroep

naam _____

straat _____

postcode en woonplaats _____

telefoon _____

computerconfiguratie _____

- ☐ geeft zich op als lid van de HCC en de NewBrain-gebruikersgroep
- ☐ is al lid van de HCC, lidmaatschapsnummer _____
en geeft zich op als lid van de NewBrain-gebruikersgroep
- ☐ geeft zich op voor regelmatige toezending van NEWBRAIN ON-LINE
beëindiging van dit abonnement geschiedt door schriftelijke opzegging

datum _____

handtekening _____

AANMELDINGSKAART

NewBrain
gebruikersgroep
postbus 4494
1009 AL Amsterdam

**cp/m-
bibliotheek**

formaat aankruisen:

- ☐ diskette 200k
☐ diskette 400k ss
☐ diskette 400k ds
☐ diskette 800k

NBGG-1 ☐ catalogus-diskette f 10,-

NBGG-2 ☐ andere volumes uit de
cp/m-bibliotheek

NBGG-3

NBGG-4

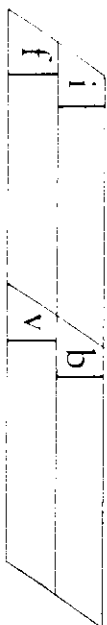
NBGG-5

prijs per volume: 400k ss, 400k ds of 800 k: f 15,00 (1 diskette)
200k f 25,00 (2 diskettes)
per 10 volumes in één bestelling een volume gratis

prijs per volume _____

besteld aantal _____ x

totaalbedrag _____



NewBrain
gebruikersgroep
postbus 4494
1009 AL Amsterdam

**software-
bibliotheek**

formaat aankruisen:

- ☐ cassette
☐ diskette 200k
☐ diskette 400k ss
☐ diskette 400k ds
☐ diskette 800k

☐ DIVERS-1 ☐ HULP-2 ☐ SPEL-2 ☐ onderwijs-
bibliotheek

☐ DIVERS-2 ☐ MATH-1 ☐ SPEL-3 ☐ SCHIJF 1

☐ EDUC-1 ☐ NBFILES ☐ SPEL-4 ☐ SCHIJF 2

☐ EDUC-2 ☐ ONLINE-5 ☐ SPEL-5 ☐ SCHIJF 3

☐ FIN-1 ☐ ONLINE-6 ☐ SPEL-6 ☐ SCHIJF 4

☐ GRAFI-1 ☐ ONLINE-7 ☐ VIDITEL-1 ☐ SCHIJF 5

☐ HULP-1 ☐ SPEL-1 ☐ _____ ☐ SCHIJF 6

prijs per cassette / diskette f 15,00 f 25,00

prijs per cassette / diskette _____

besteld aantal _____ x

totaalbedrag _____

