

New Brain on-line



12

september 1988
uitgave van de
NewBrain-gebruikersgroep



newbraindag verhuist naar leiden - pag. 3

newbrain on-line is een uitgave van de nederlandse newbrain-gebruikers-groep. de bladen zijn licht gelijmd, zodat ze eenvoudig los te maken en 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 dan ook bij de inzender.

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

door middel van de aanmeldingskaart achterop het omslag kunt u zich abonneren newbrain on-line. abonnementen kunnen alleen beëindigd worden door schriftelijke opzegging bij de gebruikersgroep.

reeds verschenen nummers kunt u bestellen door overmaking van f 10,- per nummer op girorekening 2505800 ten name van hcc newbrain-gebruikersgroep, amsterdam



kopij

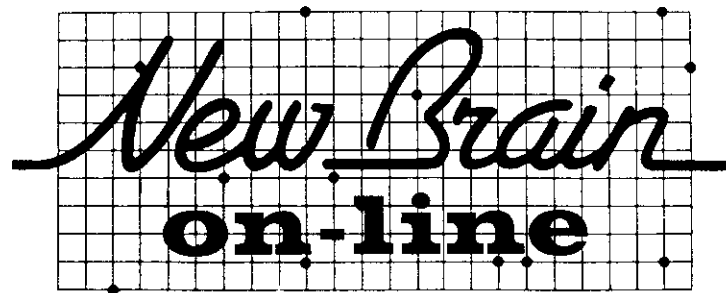
newbrain on-line 13 zal (bij voldoende kopij!) verschijnen op de newbrain-dag op 8 april 1989. sluitingsdatum voor de kopij is 1 maart 1989. maar stuur uw kopij zo spoedig mogelijk op: het werkt veel prettiger, als niet alles op het laatst gehaast gedaan moet worden. ons adres is:

newbrain-gebruikersgroep, postbus 4494, 1009 AL amsterdam

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: wordperfect-, texy-, wordstar-, wp- en alle mogelijke ascii-bestanden zijn bruikbaar. de cassette of diskette krijgt u natuurlijk terug, 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

Memo feat



TEN GELEIDE

tot ons grote genoegen is newbrain on-line 12 verschenen, en bovendien van voor tot achter gevuld met artikelen over de newbrain. er ligt zelfs nog enige kopij te wachten voor het volgende nummer. maar daar kan uw stukje ook nog bij. graag insturen voor 1 maart 1989

in het gevarieerde aanbod van artikelen in deze on-line treft u ook een van de klassieke teksten uit de newbrain-literatuur aan: 'ode aan sint pancras' van gordon crosse

tot slot nog een mededeling voor onze abonnees: wilt u om de een of andere reden uw abonnement beëindigen, dan moet u dat de gebruikersgroep rechtstreeks laten weten. als u alleen maar van de hcc-lijst verdwijnt, weten wij niet, of u on-line nog op prijs stelt!

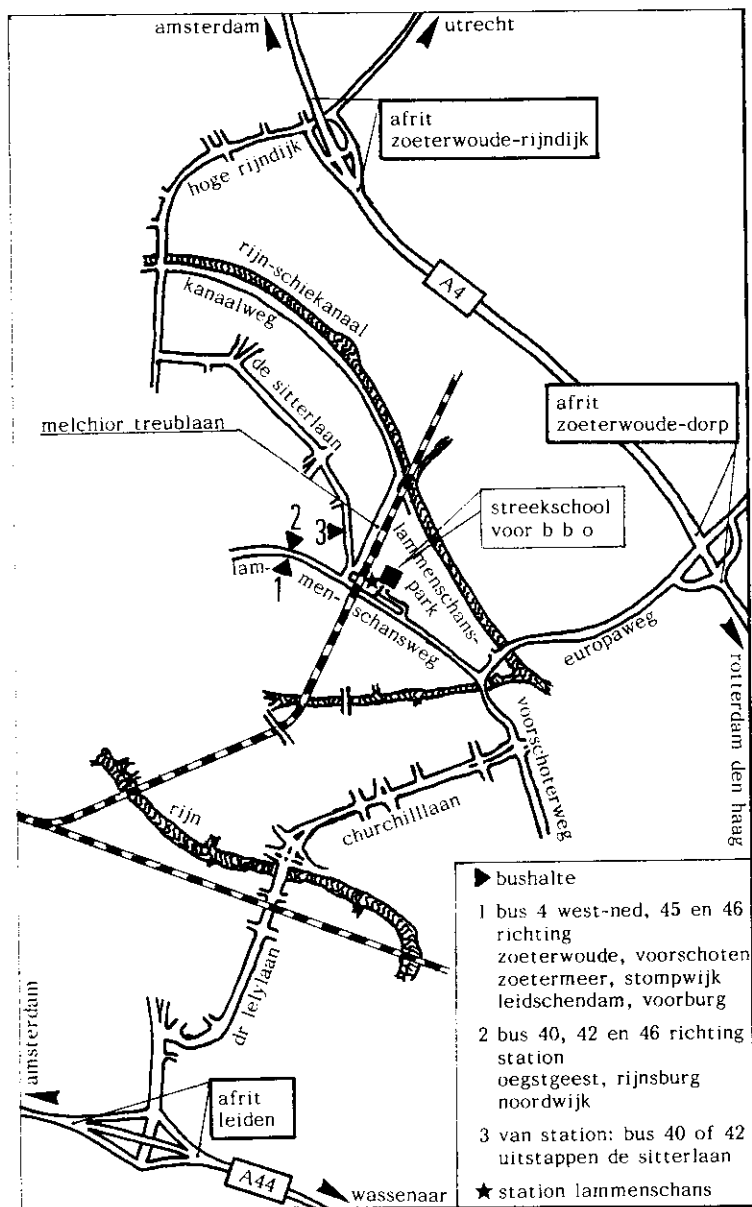
de redactie



het bestellen van software

neem het bestelformulier achterin deze on-line, en vul daarop in, welke software u wenst te ontvangen (niet alleen de opgesomde delen zijn leverbaar, maar ook alle volumes van de 'programmatheek' van de cp/m-gebruikersgroep!). kruis het gewenste formaat aan: cassette of diskette van 200k, 400k ss, 400k ds of 800k voor de newbrain, of vul het proton-formaat in. stuur de bon samen met een girobetaalkaart of betaal- of eurocheque (vergeet niet het nummer van giro- of betaalpas in te vullen) op naar bovenstaand adres.

het verschuldigde bedrag kan ook worden voldaan door overschrijving op girorekening 2505800 ten name van hcc newbrain-gebruikersgroep te amsterdam. ook in dat geval het bestelformulier opsturen.



newbraindag in leiden



de gebruikersgroep verlaat 'de bron'. op de komende newbraindagen zal ze te gast zijn in een gezellige ruimte in de streekschool voor beroepsbegeleidend onderwijs in leiden. de school is zowel met het openbaar vervoer als met de auto gemakkelijk te bereiken. automobilisten moeten er wel aan denken op de lammenschansweg tijdig de ventweg op te gaan, maar dat is ook te zien op de plattegrond, die op de pagina hiernaast afgebeeld is. op het terrein van het scholencomplex is met grote nummers duidelijk aangegeven, waar de sbbo is.

u kunt dus vast de newbraindagen van volgend jaar noteren:

zaterdag 8 april 1989
zaterdag 7 oktober 1989
s b b o
lammenschanspark 1
2321 JK leiden

NewBrain- gebruikersgroep postbus 4494 1009 AL amsterdam

voorzitter: m s vreedenburg, (02159) 11068

secretaris-penningmeester: menno stevens, (020) 924137

postrekening 2505800 tnv hcc newbrain-gebruikersgroep, amsterdam

de newbrain-gebruikersgroep is een onderdeel van de

HOBBY COMPUTER CLUB
inschrijvingsnummer kvk leiden: V445230

LANDELIJKE NEWBRAINDAGEN

coördinatie: m s vreedenburg, (02159) 11068

komende newbraindagen: zaterdag 8 april 1989 en zaterdag 7 oktober 1989

in de sbbo, lammenschanspark 1, 2321 JK Leiden

NEWBRAIN ON-LINE

redactie: menno stevens, (020) 924137

sluitingdatum voor de kopij voor on-line 13 is 1 maart 1989

zie de binnenkant van het omslag voor het inzenden van kopij

SOFTWAREBIBLIOTHEEK

bibliothecharis: bas boetekees, hans van hoek, w a van hoek, (020) 948430

onderwijsbibliothecharis: maarten floor, (02963) 4374

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

uitlevering op diskette: m s vreedenburg, (02159) 11068

COÖRDINATOREN

VAN DE WERKGROEPEN

basicode: pieter van dijk, (01751) 12367

hardware: henk dekker, (076) 713390

onderwijs: maarten floor, (02963) 4374

proton: jan wubben, (010) 4557698

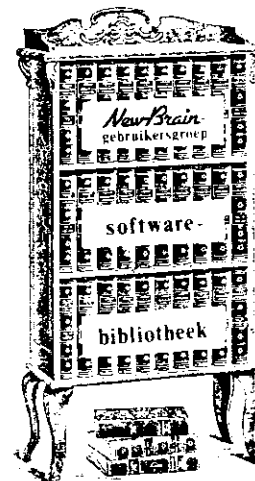
CP/M-VRAAGBAAK

henk blik, (03240) 37932



henk blik, (03240) 37932
bas boetekees
henk dekker, (076) 713390
pieter van dijk, (01751) 12367
maarten floor, (02963) 4374
jos hermans
hans van hoek (020) 940851
w a van hoek, (020) 948430
wim luijt, (01100) 28197
rob maris, (055) 424485
guus von morgen, (02158) 3381
jan wubben, (010) 4557698

nieuw team



'De functie van softwarebibliothecharis is vacant' stond er in NewBrain On-line 11 (april 1988). Het nummer (11) was wel vreemd en de maand deed er ook al geen goed aan, maar een oplossing was toch gewenst. Thans is het zo, dat er drie leden van de gebruikersgroep zich inzetten om te proberen deze functie te bemannen, al is het wel zo, dat ieder zijn eigen kwaliteiten heeft en dat wij nog niet de ervaring en de feeling van de vorige bibliothecharis bezitten. Maar als de gebruikersgroep het nog een poosje uithoudt, dan komt dit ook wel weer in orde. Het driemanschap bestaat uit: Bas Boetekees, Hans van Hoek (het jongste lid van de gebruikersgroep in jaren van lidmaatschap) en zijn vader. We hopen, dat we tot tevredenheid van de NewBrain-getrouwen ons werk zullen verrichten.

De algemene trend volgend heeft de gebruikersgroep de prijzen van de softwarebibliotheek verlaagd. De nieuwe prijzen zijn:

- elke cassette of diskette uit de softwarebibliotheek van de newbrain-gebruikersgroep: f 10,00
 - elk volume uit de programmatheek van de cp/m-gebruikersgroep op 400k-, 800k- of proton-formaat eveneens f 10,00; op 200k-formaat f 15,00
 - de catalogusdiskette met het volledige overzicht van de voorradige volumes uit de programmatheek van de cp/m-gebruikersgroep: f 5,00
- daarbij komt per postbestelling een vast bedrag van f 2,50 voor de verzending

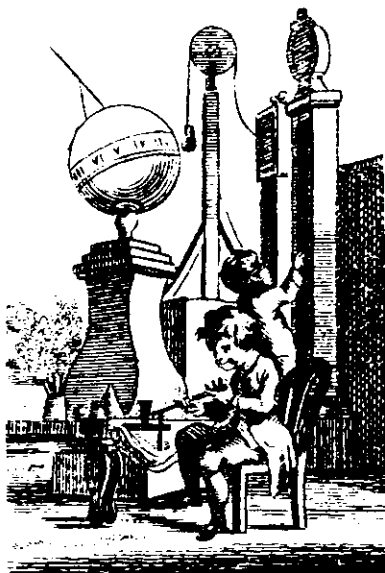


NIEUWE SOFTWARE

bestelnummer: DIVERS-6 (cassette)

INVADER.BAS en .BIN, spel in machinetaal van bjork hammer (voor 32k)
PACMAN.BAS, .BIN .HLP, spel in machinetaal van jan peter dijkstra (32k)
REACTIE.BAS, meting van reactietijd (in een iets ingewikkelder situatie dan in REACTEST uit SPEL-7), door ger. van laere (zie ook pag. 15)

NIEUW



verder nog een aantal machine-taalroutines (o.a. nautische / geografische; voor 32k) van de hand van arnold niessen, die in een programma kunnen worden opgeroepen ter verrichting van een aantal berekeningen:
CASSINI, MOLODENS, VEREFF-A: wiskundige programma's
WORTELFO berekent wortels van een vierkantsvergelijking
DATUMCON datumconversie
GEO>UTM, UTM>GEO, conversie van eenheden naar een ander stelsel
UTM<>RD conversie coördinaten
HIRES-IN laat scherm van kleur wisselen
MENU een programma om een menu op het scherm te maken
POSITIE drukt cursorpositie in line-display af (alleen model AD)
SPATIEPO haalt overbodige spaties uit string
UPC<>LWC omzetten kleine letters in hoofdletters en omgekeerd

bestelnummer: HULP-3 (cassette)

de nieuwe schermdriver voor de unexpanded newbrain van jan peter dijkstra, met aparte status- en berichtgebieden op het scherm, en een mogelijkheid voor gestructureerde invoer. een knap stuk werk! zie ook het artikel op pag. 39. met beschrijving DRIVER.DOC en demo FVDEMO.BAS

bestelnummer: N227 (diskette)

INVADER, PACMAN en REACTIE van cassette DIVERS-6 (zie boven)
de nieuwe beeldschermdriver voor de 32k van jan peter dijkstra (zie bij cassette HULP-3)

PLOTGEN.BAS, een door ger. van laere verbeterde versie van het programma van GRAFI-1 / N205 (tekening op het grafische scherm wordt in plot-opdrachten opgeslagen), met uitvoerige uitleg PLOTUITL.001 t/m .004
BIGLETT.BAS, .TXT en BIGMESS.DAT om tekst in grote letters op het scherm te maken, samenvoeging van het programma van t a morris (N505) en een van ger. van laere)

UNPAGE.BAS
TURING.BAS

NIEUW

bestelnummers: N228 en N229

deze nummers bevatten samen de volledige ROM van de disc controller, gecompileerd door wim luijt, op dezelfde manier opgezet als het rom-boek van de kale newbrain. veel code staat op een adres, waar het niet gebruikt wordt. daarom is het 'actieve' adres toegevoegd, en aangegeven, waarop de code betrekking heeft (N = 32k o.s.; P = 96k o.s.; O = beide; I = alleen bij initialisatie; C = 64k cp/m; c = 32k cp/m)

bestelnummer: N230 (diskette)

NBA, de nieuwe assembler van frans horstink voor de expanded newbrain. met editor, toelichting en source codes! zie ook pag. 35
de nautische / geografische en andere machinetaalroutines van cassette DIVERS-6 (cassini, datumcon, geo>utm, upc<>lwc, utm<>geo, utm<>rd, hires-in, menu, molodens, positie, spatiepo, vereff-a, wortelfo)

bestelnummer: N313 (diskette)

FMAP4DB.COM, FMAP4DB.DOC, UCAT4DB.COM: door wim luijt aangepaste versie van FMAP4.COM etc (schijf N311), zodat output geschikt is voor toepassing in dBASE II

FAST21.COM: verbeteringetje van FAST2 (zie pag. 68)

FASTPLUS.COM, .DOC: gordon crosse's uitbreiding van FAST met softkeys (functietoetsen). de betekenis van de toetsen kan interactief gewijzigd worden met EDITFAST.COM. ED81.HLP bevat besturingscodes hiervoor. zie ook pag. 71

FEXIT+.COM: EXIT te gebruiken bij FASTPLUS

C.FKY

CONFIG4.PAS, FKV200.COM, FKV400DS.COM, FKV400SS.COM, FKV800.COM en NBSETUP2.COM: serie programma's om een of meer van uw drives op

een ander formaat in te stellen

DEV0.COM, .ZSM: programma om device 0 onder cp/m te gebruiken. met source code
IVV.COM

bestelnummer: N401 (diskette)

de technical notes, die bij de disc controller en de eim horen. aan sommigen zijn ze (geheel of gedeeltelijk) bij aanschaf van de apparatuur verstrekt (in fotokopie), aan andere niet. op de schijf staan:

technical note 18: production of user defined characters; en 101 tot en met 114: cp/m utilities, cp/m system, discio system, configure program, kbbfio, sseio, paged memory operating system, system page description, basic extensions with peripherals, acpio and acvio, centronics printer device driver, z80 port device driver, serial memory device driver, analog device driver; en een lijst van de verschenen technical notes

bestelnummer: N402 (diskette)

verzamelde informatie over de disc controller en de eim: artikelen uit open stream, waarvan we toestemming tot publicatie gekregen hebben. voor een deel is er informatie gelijkend op de technical notes, maar ze zijn vaak uitgebreider en minder technisch. de teksten zijn in het engels. behandeld worden:

basic extensions with peripherals, cp/m system documentation, cp/m utility documentation, configuration program documentation, reading a foreign disc, newbrain discio devices 12 to 15, technical specification acia devices 16 to 17: serial printer & comms, device 18: screen editor, device 19: buffered keyboard, device 20: serial memory, device 21: centronics printer, device 33: stand alone graphics, ode to st. pancras, newbrain paged memory operating system, expansion interface & disc controller error codes, system page description

bestelnummer: BWK-1 (cassette) en N231 (diskette)

programmatuur voor de buitenwereldkaart:

ADSOFT.PUB, .DEM, .COM: a/d-omzetting (oscilloscoop)

STAPMOT.BAS (stappenmotor), LOOP.LBAS (looplicht), VERKEER.BAS (verkeerlichten bij een kruising)

BWKAART.DOC: de volledige documentatie bij de buitenwereldkaart

NIEUW

WÁAROM WERKEN SOMMIGE PROGRAMMA'S NIET ONDER FAST?

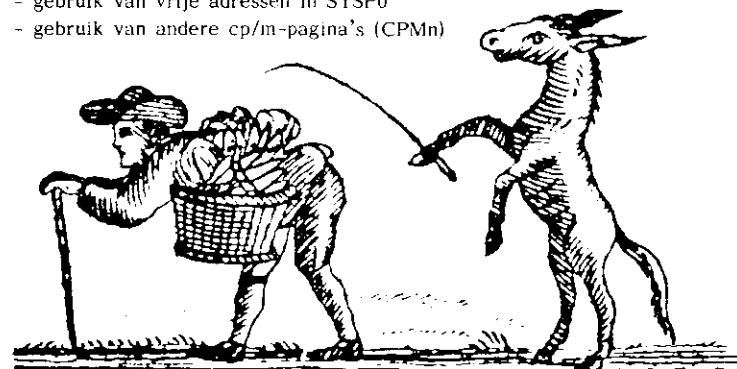
Er zijn verschillende programma's die onder cp/m op een 96k NewBrain werken, maar zodra met FAST gewerkt wordt, gaat het mis. Ik heb de softwarebibliotheek globaal doorgenomen en opvallend is, dat het meestal programma's betreft, die in pascal zijn geschreven. Ze bevatten dan procedures in pascal of z80 om routines (ZCALLs) uit het paged memory operating system van de 96k NewBrain in pascal (onder cp/m) te gebruiken. Voorbeelden hiervan zijn: PIOS.INC etc. voor Turbo-Pascal (N309), procedures voor Turbo-Pascal (N503)* en HiSoft-Pascal (N507) door Gordon Crosse. Ook andere programma's werken plotseling niet meer: DU2.COM, PROMPT.COM en SDMP.COM.

Verschillende gebruikers (zelf heb ik er geen ervaring mee) hebben mij gevraagd naar de (mogelijke) oorzaak hiervan en een oplossing. Om de oorzaak hiervan te vinden zal ik eerst de veranderingen in cp/m onder FAST vermelden.

Wijzigingen in het bios zijn de nieuwe CONST-, CONOUT- en INTRPT-routines en de gewijzigde adressen van: CONIN, CONOUT, LIST, PUNCH, READER, LISTST en CPMZCALL. De system page (SYSP0) bevat een deel van een nieuwe driver voor device 18 (SSEIO) en is geheel gevuld.

Verder bevat FAST2.COM nog een kleine bug, die mogelijk van invloed kan zijn (zie pagina 68 in deze On-line). Stream 0 en 1 (scherm en toetsenbord) veranderen niet. Mogelijke oorzaken kunnen dus zijn:

- gebruik van absolute bios-adressen
- gebruik van vrije adressen in SYSP0
- gebruik van andere cp/m-pagina's (CPMn)



*) de betreffende programma's in N309 staan ook in N503; ik heb ze daarom niet apart bekeken

Welke worden gebruikt? Uit bestudering van de pascalprocedures bleek, dat het gebruik van CPM7 voorkomt in SYSPAGE.PAS, zodat bovengenoemde bug een rol kan spelen. Verder wordt er een adres 1F00h gebruikt, dat op SYSP0 staat, maar het wordt uitsluitend met een offset gebruikt en kan daarom niet de oorzaak zijn. Alle procedures gebruiken, direct of indirect, CPMZCALL. Het directe gebruik van andere bios-routines kon niet worden vastgesteld. Op grond van deze opsomming is het meest waarschijnlijke, dat het falen onder FAST te wijten is aan het gebruik van CPMZCALLs, hoewel de andere oorzaken niet geheel uit te sluiten zijn. De oorzaak van het foutieve gebruik van CPMZCALL ligt waarschijnlijk in het boek van Philip Crookes (The NewBrain Files). Op pagina 66 vermeldt hij, dat een CPMZCALL gebruikt kan worden met:

```
CPMZCALL EQU 0F4FBh
...
CALL CPMZCALL
DEFB code
```

Het adres F4FBh leidt hij waarschijnlijk af uit 'zijn' bios-tabel op pagina 67. Deze tabel is een stukje listing uit de ROM, maar het is in de 96k cp/m actief vanaf adres F400h! In 32k cp/m staat de tabel vanaf 7000h en komt een verwijzing naar CPMZCALL zelfs niet voor.

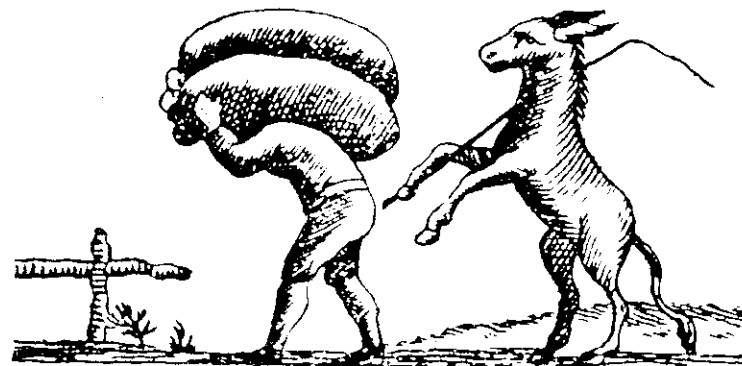
Hoewel de programma's wel zullen werken, is het gebruik van absolute adressen in het bios om bios-routines door een programma te laten aanroepen strijdig met de grondslagen van cp/m. Een bios-routine moet worden aangeroepen via de bios-sprongtabel. Mocht er een nieuw bios gemaakt worden (FAST is als zodanig te beschouwen), dan mogen de routines op andere adressen beginnen. De bios-sprongtabel staat echter altijd op dezelfde plaats, namelijk aan het begin van het bios, en in een vaste volgorde. Zolang een nieuw bios nu op hetzelfde adres begint (F400h voor de 96k NewBrain) zullen programma's geschreven voor toepassing onder een vroegere bios-versie blijven werken met een directe CALL naar de sprongtabel!

Wat is er tegen te doen? Aanpassing van FAST aan het gebruikte CPMZCALL adres wordt elders besproken. De oplossing is mogelijk, maar verdient niet de voorkeur. Het call-adres van CPMZCALL in de niet goed werkende programma's moet dus worden aangepast en veranderd in F43Ah.

Ik heb de programma's in de delen N307, N503 en N507 bekeken, nadat ik dit stukje had geschreven. Toen bleek, dat na de bovenstaande wijzigingen een aantal programma's nog steeds niet goed of maar half werkt. De

oorzaak hiervan heb ik niet gevonden, omdat ik bepaalde programma's niet kan compileren bij gebrek aan de juiste compiler. Enkele programma's (onder andere SYSTEST) vragen om invoer via een stream en blijven dan hangen, mogelijk is dit de oorzaak. Als mogelijke oorzaken van het falen kunnen nog worden toegevoegd:

- FAST blokkeert sommige streams: 0 (device 18), 1 (device 19) en 4 (device 3), en devices (0)
- het paged operating system vindt het onder FAST niet aangenaam om een pagina in een ander slot te gebruiken, als waarin het 'normaal' thuis hoort



Helaas heb ik dit niet meer kunnen ontrafelen. Mocht ik meer aan de weet komen, dan hoort u er van. Voorlopig kan ik alleen als advies geven om invoer van het toetsenbord via CONIN en uitvoer naar het scherm via CONOUT te doen, in Turbo-Pascal bijvoorbeeld met BIOS(2) en BIOS(3). Let op in Turbo-Pascal: in cp/m is 'warm boot' = bios-call 1; in Turbo Pascal is 'warm boot' = BIOS(0).

Hieronder volgt een (onvolledig) overzicht van de programma's in de softwarebibliotheek, die getest zijn, en van de regels, adressen of posities, die zijn veranderd (in de .COM files moet op het desbetreffende adres de FB in 3A worden veranderd):

N307:	SDMP	regel 32
	SDMP.COM	adres 16Bh
	PROMPT	regel 4
	PROMPT.COM	adres 10Ah

beide programma's werken nog niet onder FAST
voor GDEMO.COM en GDEMO2.COM zie N309 of N503

N309 of N503: GRAPH.INC	regels 40, 45
PIO.INC	regels 32, 56, 79, 109, 123, 145
SYS.INC	regels 81, 87
SYSPAGE.COM	adressen 2362h, 236Eh
MPAGE.COM	adres 16Bh
NBRFP.BLK	positie 279
SYSPAGE.BLK	positie 347
PIOS.COM	adressen 1FF5h, 201Dh, 2053h, 4258h, 4264h
GDEMO.COM	adressen 5166Dh, 58F4h, 590Ah, 5955h, 59E5h, 5A0Ah, 5A4Dh, 5B52h, 5B59h
GDEMO2.COM	adressen 1FF5h, 2042h, 2078h, 2145h, 2161h, 2195h, 22ADh, 22B7h

Opmerkelijk is, dat GTEST en GDEMO2 waarschijnlijk door dezelfde auteur zijn gemaakt, er verschillende (?) compilers zijn gebruikt, en ze allebei na wijziging op dezelfde plaats in het programma een manco vertonen, dat niet aan de hand van de source codes kan worden verklaard.

N506:	ZCALLCDE.ASM	regel 5
	ZCALLCDE.EXT	adres 128h
	SYSLIB.PAS	regel 284
	GTEST.COM	adres 3C7h; lijkt veel op GDEMO2.COM
	FPTTEST.COM	adres 543h
	ZTEST.COM	adres 36Eh

SYSTEST.COM en ZTEST.COM werken na wijziging nog niet onder FAST

Wim Luijt



TURBO-PASCAL EN 32K NEWBRAIN

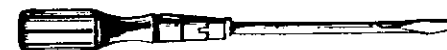
Hoewel dit niet het onderwerp van het artikel betreft, heb ik nog een verzoek aan de gebruikers, die in Turbo-Pascal programmeren. Bij het compileren naar een .COM-bestand bepaalt de compiler automatisch het eindadres. Bij de 96k NewBrain is dit Dxxxh. Als er geen pointervariabelen gebruikt worden, wordt een stuk van het geheugen niet gebruikt. Turbo-Pascal is te groot voor de 32k NewBrain, zodat gebruikers zonder EIM het niet kunnen gebruiken. Als je nu als eindadres 6200h opgeeft en je het programma zonder overflow kunt compileren, dan kunnen deze gecompileerde programma's ook onder cp/m op 32k NewBrain gebruikt worden.



TOETSEN VERSLETEN?

Bij een veel gebruikte NewBrain willen de letters wel eens van de toetsen verdwijnen. Dat is erg lastig. Geef er daarom vóór die tijd een likje blanke nagellak op.

Nog erger is het, als een toets niet meer reageert. In dat geval kan de boel door een enigszins technisch iemand worden opengeschroefd, waarbij blijkt, dat vervangen moeilijk is; maar er is een uitweg. De toets-schakelaars zitten samen in een aluminium plaat gemonteerd. Vanuit ieder montagegat lopen drie sleufjes in het aluminium. Zet nu een klein schroe-



vedraaiertje in een van de sleufjes bij de weigerende toets en wrik lichtjes tegen de schakelaar, totdat hij het weer doet. Ik heb dit aan enige toetsen uitgevoerd en de eerste zo gerapareerde toets heeft zo al langer gewerkt dan vóór die reparatie. Zo doet de letter L het al weer maanden uit de kunst. De geheel dode P had ik binnen tien minuten weer aan de praat. Veel gemakkelijker dan het 'nuodlike' vernieuwen.

P. Kramer



ZCALLS



De ZCALLs ZBLCKIN en ZBLCKOUT hebben een ernstige tekortkoming, welke mij veel tijd en ergenis gekost heeft. In het Software Technical Manual staat, dat BC de bloklengte bevat (Technical Note 10). Elders staat, dat BC kleiner moet zijn dan 256 (Appendix D)! Ook aan HL (blokbeginadres) worden eisen gesteld: groter dan 8191 en kleiner dan 40960.



Ik heb het in twee gerenommeerde Z80-boeken (Zaks en Hutty) niet kunnen vinden, maar de displacement bij geïndexeerde adressering (IX + d) en (IY + d) loopt van -128 tot en met +127.

Frans Horstink

quickfill

QUICKFIL.BAS EEN LEERZAAM SPEL

Quickfil.bas is een in spelvorm gegoten testprogramma. De bedoeling van het ontwerp was, een test te ontwerpen waarmee de volgende aspecten konden worden aangetoond:

- 1 ogen zien, maar nemen niet steeds alles even duidelijk waar
- 2 beslissingen van mensen kunnen worden beïnvloed door omstandigheden
- 3 de fysieke - en psychische - toestand van een mens beïnvloedt de door deze te nemen beslissingen

HET OOG ALS BASIS VOOR DE WAARNEMING

Het feit dat u dit artikel leest, bewijst dat u tot de zienden behoort. Maar wist u, dat er verschillende manieren van zien zijn? Welnu, afgezien van het antwoord op deze vraag, mogelijk zult u na het verder lezen in het vervolg bewuster zien.

Zonder al te veel in academische spraak te vervallen zijn er enige opmerkingen over het oog te maken:

¶ We bezitten er (hopelijk nog) twee van.

¶ Overdag zien we beter dan 's nachts.

¶ In het oog zit een vlies dat de lichtstralen, die via de ooglenzen binnenkomen, registreert en door middel van elektrische signalen naar de hersenen stuurt (het netvlies).

¶ Het netvlies heeft een zogenaamde blinde vlek. Die vlek is aan te tonen door op een blank vel papier een tweetal vette stippen op een afstand van 10 centimeter van elkaar te plaatsen. Klaar? Houd nu het



papier op een afstand van ongeveer 35 centimeter voor het gezicht en knijp een oog dicht. Kijk naar een van de stippen en constateer, of u de andere stip nog tegelijk waarneemt. Zo ja, kijk dan naar de andere stip en merk op, dat u nu de eerste niet ziet. Is dat zo, dan heeft u de aanwezigheid van de blinde vlek bij uzelf aangetoond, maar kijk toch even naar de andere stip om op te merken, dat de vorige nu wel waar genomen wordt. De blinde vlek bevindt zich bij het ene oog links, bij het andere oog rechts, zodat ze bij het kijken gewoonlijk niet samenvallen. Hebt u geen van beide stippen goed gezien? De oogarts kan u mogelijk helpen.



¶ U hebt door de vorige proef tevens aangetoond, dat u gericht (direct) kijkend naar de ene stip, de andere wel kunt waarnemen, zij het dan dat deze iets vager is. De stip die u scherp en duidelijk zag, lag in uw directe gezichtsveld. De stip die u zijdelings wel waarneemt maar iets vager, ligt in het perifere gezichtsveld.

¶ De hoek van het directe gezichtsveld bedraagt slechts 1,5 graad, zodat een directe waarneming van iets waarvan u de aanwezigheid niet bewust was, min of meer op de toevalligheid berust, dat het zich in uw directe gezichtsveld bevond. Dit deel van het gezichtsveld wordt het foveale deel genoemd. U merkt dat er toch wat academisch aandoende woorden in de tekst sluipen. U mag ze wat mij betreft weer vergeten. Noodgedwongen moet ik er echter wel enige noemen. Er zijn moeilijk andere woorden voor te vinden.

¶ Een oog heeft tijd (en soms veel tijd) nodig om aan een veranderde lichtsituatie te wennen. Voor een totale gewenning is een tijd van twee uren normaal. U zult begrijpen, dat de meeste dingen dus wel gedeeltelijk aan ons voorbij moeten gaan: ze gunnen u de tijd niet ze 100 % waar te nemen. Een ieder kent het onaangename niets zien, nadat bij donker het licht uit gaat. Na enige tijd neemt u toch meestal wel weer wat waar. Even wennen aan het licht zegt men dan wel eens.

¶ Er is tijd nodig om het oog te verdraaien om de blik te verleggen. Dat houdt in, dat hetgeen zich in het perifere gezichtsveld afspeelt maar toch onze aandacht vraagt, pas enige tijd later direct kan worden bekeken. Als voorbeeld hiervan noem ik de situatie, dat u met een persoon staat te praten, terwijl zijdelings iemand naar u toe komt. Vaak zult u de blik moeten verleggen om echt te zien, wie er aan komt.

¶ De verandering van de blikrichting (de zogenaamde saccade) kost u tijd. Normaal gesproken ervaart u dat niet als hinderlijk. Toch kost het u steeds minimaal circa 0,27 seconde (= 270 milliseconde). Voor de rekkenaars: de totale blikverdraaiing is in tijd te benaderen door tweemaal de oogverdraaiingshoek + 30 te nemen. De uitkomst is in milliseconden. Ogen kunnen (soms tijdelijk en zeer kort) verblind raken. Met name als er een plotselinge toename van de lichtsterkte optreedt. Denk aan het naderen van een tegenligger in het verkeer bij donker of het plotseling opdoemen van een laagstaande zon.

¶ Ogen kunnen vermoeid raken. Dit houdt voornamelijk in, dat de spieren die de ooglenzen (bolling) in de juiste stand brengen, vermoeid geraken. Het scherpzien wordt dan bemoeilijkt.



¶ U heeft, als u twee goede ogen bezit, steeds een oog dat overheerst (leidt). Door de volgende proef kunt u erachter komen welk oog uw leidende oog is. Veelal is dat ook het bestziende oog. Kijk maar eens met geopende ogen naar een smal voorwerp dat zich op enige afstand (bij voorbeeld op 2 meter) bevindt. Plaats een duim omhoog gestoken en met gestrekte arm tussen het voorwerp en uw ogen, zodanig dat het voorwerp voor het gezicht gedeeltelijk of geheel door de duim aan het zicht wordt onttrokken. Knijp nu een oog dicht of houdt de andere hand ervoor. Als uw duim nu het voorwerp aan het zicht blijft onttrekken, dan heeft u uw leidende oog open. Verplaatste uw duim zich voor het gezicht, terwijl u hem toch op dezelfde plaats hield, dan kneep u uw leidende oog juist dicht of bedekte het. Er is over ogen nog veel meer te vermelden, maar de belangrijkste dingen (voor het doel dat ik voor ogen heb) zijn zo wel vermeld.

TUSSEN WAARNEMEN EN REAGEREN

Athankelijk van de situatie die zich voordoet, zal uw reactie verschillend zijn. Neem eens een verkeerssituatie waarbij u als fietser plotseling geconfronteerd wordt met een zeer nabij recht op u afkomende tegenligger (auto). Of neem een verkeerssituatie waarbij u met een auto rijdt met een snelheid van 50 km/h over een weg en er op een afstand van 100 m een voetganger voor u dwars de rijbaan oversteekt. Mogelijk constateert u bij uzelf enig verschil in reactie (liever de eerste situatie niet doelbewust uitproberen). De genoemde situaties behoren tot de minder gecompliceerde. De volgende situatie zal u mogelijk als ingewikkelder overkomen.

U rijdt binnen de bebouwde kom door een straat. Aan beide zijden van de straat staan auto's stil. Er blijft net voldoende ruimte over om in twee richtingen personenauto's te laten rijden. Plotseling rent een kind tussen de rechts staande auto's de rijbaan op en wel op geringe afstand voor uw voertuig. Op het moment dat u voor het kind naar links uitwijkt, wordt het portier van een links staande auto wijd open gedaan, waardoor een voor u tegemoet komende auto in uw richting uitwijkt. U begrijpt dat voor het oplossen van al deze problemen een aantal reacties snel op elkaar gevraagd wordt, waarbij de reeds genomen beslissingen steeds moeten worden bijgesteld. Een meer gecompliceerde situatie dus.

De twee eerste situaties behoren mogelijk tot een ervaring die u reeds eerder opdeed. De laatste dient zich mogelijk voor het eerst aan.

U zult begrijpen dat voor verschillende situaties ook een verschillende reactietijd zal gelden. De reactietijd bestaat in het algemeen uit de volgende perioden:

- 1 de waarnemingsperiode (waarover in het vorige gedeelte iets werd vermeld)
- 2 de bewustwordingsperiode (wat is er aan de hand?)
- 3 de beslissingsperiode (wat zal ik eraan doen?)
- 4 de omschakelingsperiode (de vorige handeling beïnvloeden)
- 5 de aanspreekperiode (de nieuwe handeling uitvoeren)

Ten aanzien van QUICKFIL.BAS kan worden gesteld, dat van al het vorige de volgende aspecten in meer of mindere mate tijdens het uitvoeren van het programma aan de orde komen. Bewust of onbewust zult u ze beleven of automatisch gebruiken.

a. Het directe gezichtsveld.



verloopt. De aandacht (concentratie) is niet optimaal meer en de ogen geven mogelijk reeds een iets ander gevoel.

- e. De waarnemingsperiode. Er wordt u namelijk gevraagd erop te letten of er een teken knippert.
- f. De bewustwordingsperiode. Als u heeft waargenomen, dat er een teken knippert, moet u zich bewust worden, waar dit knippert, een foute of een goede plaats?
- g. De beslissingsperiode. Hebt u eenmaal onder e. en f. de bewustwording doorgemaakt, zult u moeten beslissen, of u een toets wel of juist niet indrukt om het knipperende teken in een ander teken te veranderen.
- h. De omschakelperiode. De vinger wordt stil (maar gespannen) boven de bedoelde toets gehouden. Is de beslissing onder g. gevallen en luidt deze 'drukken', dan moet de spierspanning van stilhouden eerst worden beëindigd.

b. Het perifere gezichtsveld. Uw monitor staat normaal gesproken op een zodanige afstand, dat u het beeldscherm geheel waarneemt, maar slechts een deel direct. Het grootste deel van het beeldscherm bevindt zich in het perifere gebied.

c. De saccade. Door het zojuist vermelde bij a. en b. bent u genoodzaakt de blik te verleggen door de ogen te verdraaien. U zult immers boven aan het scherm kijkend bij een tekstpagina de onderste regels niet kunnen lezen. Die oogverdraaiing kost u tijd.

d. Vermoeidheid. Na het achter elkaar uitvoeren van een aantal spelen ontdekt u, dat het niet zo gladjes meer

- i. De aanspreekperiode. De nieuwe handeling 'drukken' wordt uitgevoerd. De spieren die voor drukken kunnen zorgen worden geactiveerd. De handeling wordt uitgevoerd.

De gehele situatie in het spel is zo ontworpen, dat een enkele fout in een spel gemaakt uw promotie (mogelijk tijdelijk) in de weg staat. U zult een spelniveau terug moeten en daar bewijzen, dat u weer hogerop kunt. De (psychische) druk die hiervanuit gaat, telt mee in enkele van de opgesomde principes en beïnvloedt het geheel.

DE REALISATIE VAN HET PROGRAMMA

Teneinde gemakkelijk een speelraster met 100 vakjes (10 * 10) te ontwerpen werd gekozen voor karakterset 2, welke bij de start van de machine reeds is ingeschakeld (via het toetsenbord control/w-b of via het programma PUT 23, 66). Het beeldscherm is donker, de tekens licht. Het speelraster is in DATA-regels in het programma geplaatst. Dit is niet de zuinigste manier om met het geheugen van de machine om te gaan, maar wel duidelijk voor iemand die de listing wil bestuderen. Het speelraster wordt opgebouwd door middel van een FOR-NEXT-lus met daarin een READ-opdracht gevolgd door een PRINT-opdracht. Tijdens de uitleg wordt het raster op soortgelijke wijze voorzien van een voorbeeld. Voor de speler (of degene die dit programma meer wetenschappelijk wenst te gebruiken) werd een keuzemogelijkheid voor de te bedienen toets gegeven. In vele programma's miste ik eerder deze optie. Kennelijk is niet iedere programmeur zich ervan bewust, dat er bijna net zoveel linkshandige mensen zijn als rechtshandige.

Door middel van een positiegenerator (RND-getallen) wordt bepaald, waar in het speelveld een reken wordt aangeboden. Door het gebruik van de opdracht RANDOMIZE zijn de gegenereerde getallen optimaal gevarieerd. Aangezien het programma, ondanks het veelal wat tragere basic, wat te snel zou verlopen, werden wachtlussen geplaatst. Dit gaf als bijkomend voordeel, dat met een wat trager spel uitvoering kan worden begonnen, terwijl het tempo bij verder spelen (als beloning voor goede prestatie) kan worden verhoogd.

Door het aantal programmahandelingen tijdens het spel zo beperkt mogelijk te houden kon zelfs een zo grote snelheid worden bereikt, dat de meest geoefende speler naar mijn mening ook in de meest optimale situatie het allersnelste spelniveau (niveau 1) nimmer zal kunnen halen. De mens is te beperkt voorzien van refer-mogelijkheden om dit niveau tot

een goed einde te brengen. Overigens werkt het uitschakelen van het linedisplay bij het NewBrainmodel AD zeer positief op de printspeed, zodat dit in het programma tot een van de eerste handelingen behoorde.

Door het vergelijken van de goede oplossing van het spel (in de DATA-regels vastgelegd) met de situatie op het beeldscherm wordt het aantal fouten bepaald. Dit is een pure stringvergelijking, waarbij karakter voor karakter wordt vergeleken. Zodoende kan per regel het aantal fouten worden vermeld, is in alle regels van het speelraster geen enkele fout gemaakt? Dan wordt het spelniveau verhoogd. De aandachtstrekker in het speelraster gaat sneller knipperen en de tijd waarin de beslissing aan de machine kan worden meegedeeld, wordt korter. Tijdens het spel kan niet worden gestopt. Dit is bereikt door de opdracht ON BREAK GOTO te koppelen aan de opdracht RESUME. Omwille van de testwaarde van dit spel is geen optie voor eigen invoer van het aanvangsspelniveau ingevoegd. Het spel dient men steeds vanaf het begin te spelen en daarbij te zien, hoe ver men komt. Teneinde de huiselijke omstandigheden (als het drinken van koffie of thee) niet al te veel door het spel te laten verdringen, werd gekozen voor de mogelijkheid van pauze tussen de verschillende spelen in.

Degene die erachter wil komen, welke de kortst mogelijke reactietijd voor hem- of haarzelf is, verwijs ik gaarne naar de programma's die ik voor dat doel ontwierp, te weten:

1. REACTEST.BAS (reactietijdmeter in een simpele situatie)
2. REACTIE.BAS (reactietijdmeter in een iets ingewikkelder situatie)

Veel plezier en meer bewustwording van eigen (on)mogelijkheden toegewenst.

Ger. van Laere

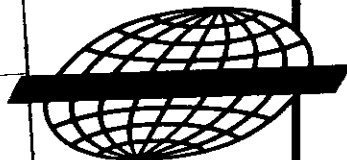


REACTIE.BAS is in de softwarebibliotheek opgenomen op bestelnummer DIVERS-6 (cassette) en N227 (diskette), zie ook pag. 5.
REACTEST.BAS en QUICKFIL.BAS op bestelnummer SPEL-7 (cassette) en N226 (diskette)

buitenwereldkaart

De Buitenwereldkaart kan gebruikt worden in alle mogelijke NewBrain-systeemconfiguraties en bevat de volgende basiscomponenten:

PIO parallelle I/O-poort met 16 lijnen
CTC counter/timer chip
SIO dubbele seriële-poortchip (RS-232)



De buitenwereldpoorten:

- 2 x V24-poort volgens NewBrain-connectoraansluiting
- 1 x modem-/diversenpoort met enkele CTC-aansluitingen en DTR, DCD
- 1 x PIO-poort met alle PIO-poort aansluitingen
- 1 x expansiepoort: gebufferde databus met enkele I/O-selectlijnen
- 2 x voedingsaansluiting: voor doorlussen NewBrain-voeding

Toepassingen zijn onder andere:

modemcommunicatie zonder flikkerende beelden (o. a. Fido)
diverse RS-232-communicatie
A/D-omzetting (oscilloscoop)
D/A-omzetting (analoge besturing)
pulsteller (lopende band)
Centronics-printeraansturing
modelbaanbesturing
procescontrole

NewBrain
gebruikersgroep
postbus 4494

1009 AL amsterdam

Verdere interfaces:

A/D-omzetter: komt op expansiedatabus, zodat PIO vrij blijft. Kosten ongeveer f 40,-

RELBOX: 16-kanaals output-interface voor galvanische scheiding van computer en stuur-eenheden (belastbaar 8 Ampère bij 220 Volt). Kostprijs f 275,- inclusief relais, kast en kabels

OPTOBOX: tweemaal 8-kanaals input-interface voor galvanische scheiding van computer en aanstuur-eenheden. Geschikt voor statische opnemers, reedcontacten etcetera. Kostprijs f 175,- inclusief kast en verbindingkabel

nog enkele exemplaren beschikbaar
geheel gemonteerd, zó in te pluggen, voor maar f 150,-
(inclusief software en een handleiding van 50 pagina's)
neem contact op met maarten floor, telefoon (02963) 4374

msdos

COMMUNICATIE MET EEN MSDOS-MACHINE

Er zijn zo van die omstandigheden, dat het handig is als twee computers met elkaar kunnen communiceren. In mijn geval was dat de aanwezigheid van een NewBrain zonder cassetterecorder, diskdrive en expansion-unit, alsmede een msdos-machine. In deze situatie is de vraag gerezen, of het mogelijk was om programma's van de NewBrain op te slaan op, en te laden van de schijf-eenheid van de msdos-machine.

Omdat de comms-poort de enige in- / uitgang is op de NewBrain, bleef als tegenpool niet veel anders dan de seriële in- / uitgang op de msdos-machine over. Dit is dan de tweede seriële poort geworden (een 25-pens RS-232-uitgang, aangezien de eerste (een 9-pens RS-232) reeds in gebruik is voor een printer. Dit leidde tot de volgende fysieke verbindingen:

RS-232 (25-pens)	RS-232 (9-pens)	NewBrain (6-pens)
2 ← TXD (transmit data) →	3 ← RX →	1
4 → RTS (ready to send) →	7 → busy/CTS →	2
3 → RXD (receive data) →	2 → TX →	4
5 → CTS (clear to send) →	8 → RTS →	5
7 → CD (carrier detect) →	5 → SG →	6
8 → CD (carrier detect) →	1 ←	
6 → DSR (data set ready) →	6 →	
20 ← DTR (data terminal ready) →	4 ←	

Er is in de 25-pens connector een doorverbinding noodzakelijk tussen de pennen 6, 8 en 20, omdat de drie signalen niet zijn voorzien in de aansluiting van de NewBrain.

De 9-pens NewBrain-connector hangt af van de 25-pens NewBrain-connector. Achterop de NewBrain-connector is een aansluiting van de comms-poort zo:

2	4	6
1	5	

het dataformaat van de comms-poort (device 9) is nauwelijks beïnvloedbaar; alleen de zend- en ontvangsttransmissiesnelheden zijn nog instelbaar. Het dataformaat is daar ingesteld op acht databits, twee stopbits en geen pariteitsopt. Toen enig probleem met pascal- en msdos-device drivers niet tot succes leidde, heb ik mij gewend tot gwbasic om daarmee het juiste dataformaat op de COM2-poort te kunnen installeren.

Het datatransmissieprotocol tenslotte werd bepaald door de door mij gewenste mogelijkheden. Ten aanzien van de NewBrain wilde ik mij beperken tot een simpel SAVE-, resp. LOAD-commando. Omdat opslaan resp. laden van een programma via device 9 niet gecombineerd kan worden met het opgeven van een filenaam, blijkt het effect van deze commando's als volgt te zijn:

SAVE: de NewBrain kijkt, of de ontvangende poort geopend is, en dumpst de data. De datastream wordt afgesloten met een control-Z

LOAD: de NewBrain leest alle binnenkomende data, totdat een control-Z wordt gedetecteerd

Omdat in deze situatie de NewBrain geen signalen produceert, waarmee hij aangeeft, wat er moet gaan gebeuren, is er voor de msdos-machine geen signaal, waaruit hij kan opmaken, wat hij moet gaan doen. Hetgeen betekent, dat ik dat zelf moet vertellen. Het volgende gwbasic-programma verzorgt daarom de volgende taken:

- het vertonen van een menu met de keuzen: L(oad), S(ave), C(hange program name) en Q(uit)
- na keuze L: het openen van de seriële poort (met alle benodigde parameters); het openen van de schijfteenheid; en het dumpen van de data, zodra de ontvangende poort geactiveerd is
- na keuze S: het openen van de schijfteenheid; het initialiseren van een buffer; het openen van de seriële poort; het ontvangen van aankomende data (totdat control-Z ontvangen wordt) en wegschrijven in een buffer (dit omdat de interne buffer van de seriële poort te klein is om het verschil in de verwerkingssnelheden van seriële poort en schijfteenheid op te vangen); en het wegschrijven van de data in de buffer naar de schijfteenheid
- na keuze C: het veranderen van de filename
- na keuze Q: het beëindigen van het programma en terugkeer naar msdos

Met de ingestelde parameters wacht de msdos-machine ongeveer een halve minuut, voordat er melding wordt gemaakt, dat de zendende / ontvangende machine zich niet heeft klaargemaakt. De NewBrain reageert evenzo.

!Het opstarten van de communicatie is zeer eenvoudig: bij de NewBrain moet device 9 geopend zijn c.q. worden; op de msdos-machine is het

commando 'gwbasic nbrainio.bas' voldoende, wanneer tenminste het communicatieprogramma 'nbrainio.bas' is genoemd.

Tenslotte volgt hier een listing van het programma (in gwbasic dus!).

Hans van Hoek



```

1 REM Dit programma verzorgt het wegschrijven naar, en het lezen van
2 REM schijf voor een NewBrain via een msdos-machine.
3 REM Gebruiksmogelijkheden: msdos-schijven te gebruiken voor
4 REM NewBrain-programma's
5 REM de communicatie tussen beide computers verloopt via de seriële
6 REM com2-poort op 9600 baud, geen pariteit, 2 stopbits, 8 databits.
7 REM Op de NewBrain moeten de programma's naamloos geSAVED,
8 REM resp. geLOAD worden naar resp. van de comms-uitgang.
9 REM De programmaam wordt in dit transmissieprogramma opgegeven.
10 A$ = "TEMP.TXT" REM default filenaam; wordt bij starten getoond
20 B$ = ""
30 WHILE B$ <> "q" AND B$ <> "Q"
40 GOSUB 70
50 WEND REM einde while-loop
55 SYSTEM REM verlaat basic
60 END
70 REM menu; spreekt verder voor zich
80 CLS
90 LOCATE 8, 10 REM zet de cursor op de juiste plaats
100 PRINT "file in use: ", A$
110 LOCATE 10, 10
120 INPUT "enter Load, Save, Change file or Quit ", B$
130 IF (B$ = "c") OR (B$ = "C") THEN 140 ELSE 160
140 LOCATE 12, 10
150 INPUT "enter new filename: ", A$
160 IF (B$ = "l") OR (B$ = "L") THEN GOSUB 300
170 IF (B$ = "s") OR (B$ = "S") THEN GOSUB 500
180 REM terugkeer van l of s
190 RETURN REM naar hoofdloop
300 REM sub laden
310 LOCATE 14, 10
320 PRINT "LADEN"
330 OPEN "com2: 9600, n, 8, 2, cs30000, ds30000, bin" FOR OUTPUT AS

```

#2

```
331 REM com2 is de tweede, com1 is de eerste seriële poort
332 REM cs / ds geven de wachttijd in milliseconden, voordat er een
333 REM foutmelding gegenereerd wordt
334 REM bin is de soort file (binary, en niet ascii)
335 REM poort te openen voor output resp. voor input
336 REM #2 is de "streamnaam"
340 OPEN "i", #1, A$ REM het openen van de schijf eenheid voor input
350 WHILE NOT EOF(1) REM zolang einde file niet is gedetecteerd
360 IN$ = INPUT$(1, #1)
370 PRINT #2, IN$;
380 PRINT IN$;
390 WEND
400 REM PRINT #2, 4
410 CLOSE #1 : CLOSE #2
420 RETURN 180
500 REM sub saven
510 LOCATE 14, 10
520 PRINT "SAVEN"
530 OPEN "o", #2, A$
540 DIM TEMP$(255, 1) REM declaratie buffer. Deze is zo groot, dat
541 REM de NewBrain zich er al in kan verslikken
550 FOR I = 0 TO 255 : FOR J = 0 TO 1 : TEMP$(I, J) = "" : NEXT
: NEXT REM initialisatie
560 I = 0 : J = 0
570 OPEN "com2: 9600, n, 8, 2, cs30000, ds30000, bin" FOR INPUT AS
#1
580 WHILE NOT EOF(1)
590 IN$ = INPUT$(LOC(1), #1) REM leest de in de 'poortbuffer' aanwe-
zige karakters
600 IF (LEN (TEMP$(I, J)) + LEN (IN$)) > 255 THEN I = I + 1 REM
maximale stringlengte is 255
610 IF I > 255 THEN I = 0 : J = J + 1
620 TEMP$(I, J) = TEMP$(I, J) + IN$ REM verzamel alle ingelezen
data in de buffer
630 PRINT TEMP$(I, J)
640 WEND
650 FOR L = 0 TO J : FOR K = 0 TO I : PRINT #2, TEMP$(K, L);
: NEXT : NEXT
651 REM schrijf de inhoud van de buffer naar de schijf
660 CLOSE #1 : CLOSE #2
670 ERASE TEMP$
680 RETURN 180
```



assembler

MIJN WEG TOT ASSEMBLER

"Wa 't it wit, dy kin it net
En wa 't it kin, dy wit it net."

Met andere woorden, wie eenmaal goed op de hoogte is van een zaak, die heeft geen weet meer van de moeilijkheden van een beginner. En dat geldt zeker voor het werken met assembler. Daarom is iemand die net de eerste fase van dat werk heeft afgesloten, waarschijnlijk de beste leidsman voor de beginner. Aangezien ik pas met veel moeite mijn eerste grote assembler routine aan de gang heb gekregen, trek ik nu de stoute schoenen aan om daarover te schrijven.

GROTE TIJDWINST

Het loont tegenwoordig meestal niet meer de moeite om zelf een tekstverwerker te maken. Als men speciale eisen stelt, wordt dat echter anders. Een van mijn speciale eisen was het kunnen verwerken van zeer veel aparte leettertekens voor gebruik in taalkundige teksten. Als computer gebruik ik daarom de NewBrain A, nog steeds een van de zeer weinige die naast een uitgebreide Westeuropese tekenset ook nog het complete Griekse alfabet op het scherm kunnen brengen. De printer is een Epson RX-80 F/T, die ook heel wat kan.

Het probleem is echter, dat een heleboel tekens op de NewBrain en de Epson verschillende ASCII-codes hebben. Er moet dus worden vertaald en dat blijkt met basic een zeer tijdrovende geschiedenis te zijn. Het afdrucken van een blad A4 van 54 regels à 78 kolommen duurde daardoor een half uur, wat zelfs voor een amateur te veel is. Ik ben daarom met de moed der wanhoop met assembler aan de slag gegaan en na meer dan een jaar worstelen met voetangels en klemmen kan ik eindelijk een bladzijde afdrucken in 3 minuten, waarvan dan nog de helft printertijd is. De tekstverwerker is onder de naam LEKSISKRIFT opgenomen in de softwarebibliotheek van de NewBrain-gebruikersgroep (alleen op cassette; bestelnummer DIVERS-5).

DE ASSEMBLER

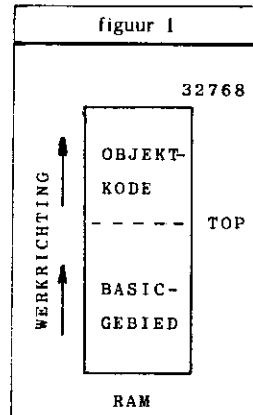
De eerste zorg is de aanschaf van een assemblerprogramma. Die zijn vrij duur en je koopt een kat in de zak. Ik heb dan ook getreuzeld tot in genoemde softwarebibliotheek het assemblerprogramma Z80BRAIN van F. Horstink verscheen. Dat heeft goed voldaan, mede omdat er een goed voorbeeld bij was gegeven. Bovendien heeft de auteur mij nog een keer over een hindernis geholpen, waarvoor hier mijn hartelijke dank.

Om te zien wat zo 'n assembler doet, moeten we eerst even recapituleren hoe het computergeheugen is opgebouwd en wel speciaal het vrij toegankelijke gedeelte, de RAM (random access memory). De RAM bestaat (bij de standaard NewBrain) uit 32768 ($\approx 2^{15}$) genummerde geheugenplaatsen, die ieder een getal (byte) tussen 0 en 255 ($= 2^8 - 1$) kunnen bevatten (zie figuur 1 met onderaan geheugenplaats 1 en bovenaan geheugenplaats 32768).

Als we nu het assemblerprogramma laden, dan vult dit het basicgebied aan de onderkant. Aan de bovenkant blijft dus wat ruimte onbezet, waar we straks het resultaat van ons werk, de object code, gaan neerzetten. Om ervoor te zorgen, dat de basic niet per ongeluk in dat gebied terechtkomt, moeten we het reserveren, wat in de NewBrain met de instructie RESERVE gaat. Na RESERVE 1000 blijkt de in basic opvraagbare TOP te zijn verkleind van 32768 tot 31768.

CENTRALE PROCESSORUNIT

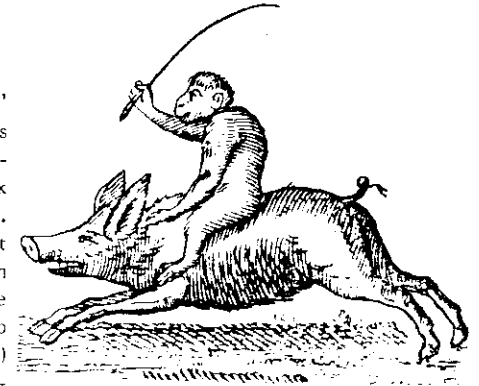
Het geheugen wordt bestuurd door de centrale processorunit (CPU, hier de bekende Z80), die ook alle verdere bewerkingen kan uitvoeren. De CPU bevat een aantal registers; dat zijn als het ware stukjes geheugen waarin met de bytes kan worden gemanipuleerd: optellen, aftrekken, schuiven enzovoort. Die registers worden aangeduid met letters: A, B, C, D, E, H, L enzovoort, en hebben ieder hun specifieke mogelijkheden. Het meest flexibele register is A, de zogenaamde accumulator. Deze registers kunnen één byte (maximaal 255) bevatten. De combinaties BC, DE en HL kunnen echter wel twee bytes bevatten (dus getallen tot 65535). Hoe dat allemaal werkt, hoe de registers moeten worden aangestuurd en wat wel en wat niet mag vind je allemaal in processorhandboeken. Ik heb met grote tevredenheid gebruik gemaakt van: Rodney Zaks, Programmeren van de Z80, ISBN 3-88745-100-7 (620 bladzijden!)



Na het starten van de assembler kan de source code worden ingetikt. Dat is een reeks van instructies, zoals die uit het bovengenoemde handboek kunnen worden afgeleid. Van de grote hoeveelheid daar beschreven instructies heb ik er echter maar een tiental nodig gehad. Is het intikken beëindigd met END, dan laat men de assembler assembleren, dat wil zeggen, dat de source code wordt omgezet in de object code, het uiteindelijke machinetaalprogramma. Daarbij verschijnt op het scherm een tekst zoals gegeven in figuur 2. Het gaat in dit geval om een afgerond stuk van mijn programma, dat in het volgende zal worden beschreven.

DE SOURCE CODE

De kolommen 'instructie' (met inbegrip van de labels NIJ; enzovoort) en 'commentaar' geven de oorspronkelijk ingetikte source code weer. Het begint in regel 1 met ORG 28993; dat is een speciale instructie van de assembler, die aangeeft op welke geheugenplaats (adres) de object code moet beginnen. Die plaats wordt bepaald door de maximaal te verwachten lengte van de object code; in dit geval zijn daarvoor dus $32768 - 28993 = 3775$ bytes beschikbaar. De assembler zorgt er voor dat die ruimte gereserveerd wordt.



Bij mijn programma was het de bedoeling, dat de letters één voor één van scherm worden gepakt, daarna zonnig worden omgezet (omgecodeerd) en tenslotte worden gedrukt. De tekst op het scherm is te vinden in het schermgeheugen. Dat is een stukje van de RAM en het begint in dit geval op geheugenplaats 644 (uit te vinden met PEEK). Daarom wordt in regel 2 van figuur 2 dit getal in het registerpaar HL gezet. De assembler bepaalt de (hexadecimale) Z80-code als 21 84 02 (dat is decimaal 33 132 2). Hex 21 is volgens het handboek de code voor 'LD dd, nn' met dd = HL, terwijl $132 + 2 * 256$ het getal 644 vertegenwoordigt. De hele instructie kost dus drie bytes, zodat de volgende op adres 28996 kan beginnen.

Die volgende instructie (regel 3) laadt het aantal van het scherm af te drukken regels (hier slechts 6, omdat het een proef betreft) in register C. Omdat het getal 6 in een byte past, zijn voor deze instructie maar twee bytes nodig. In regel 4 wordt de inhoud van C vermeerderd (INCRement) met 1 (reden zie beneden). Daarna wordt in regel 5 de regellengte (80

tekens) in register B geladen en in regel 6 de geheugenplaats waar HL naar wijst (dat is de eerste plaats van het scherm) in register A. Regel 7 laadt het getal 9 in register E, wat in de NewBrain de betekenis heeft van het instellen van 'stream' 9 voor het printen (dat is te vinden in het NewBrain Software Manual). In regels 8, 9 en 10 worden daarna de inhoud van A, HL en BC (dat is: B en C) op vrije geheugenplaatsen boven in de RAM geladen. Dit is voor de zekerheid, omdat bij de meeste bewerkingen de inhoud van een of meer registers verloren gaat. In regel 11 wordt de inhoud van de accumulator A (in dit geval het eerste teken van het scherm) vergeleken (ComPare) met CHR\$ (148). Dat teken gebruik ik als loze spatie, zodat er in dat geval niets behoeft te worden geprint. Is het teken op het scherm inderdaad gelijk aan CHR\$ (148), dit is het verschil is nul (Zero), dan springt het programma van regel 12 naar label STAP (regel 18). Daar worden de oorspronkelijke waarden weer in HL en BC geladen en wordt in regel 20 HL met 1 vermeerderd (en wijst dus naar de volgende plaats op het scherm!). DJNZ WER betekent, dat B met 1 wordt verminderd en dat, zolang B niet nul (NZ) is, naar WER wordt teruggestapt, waarna opnieuw wordt begonnen met de volgende plaats op het scherm.

AFDrukKEN

Blijkt in regels 11 en 12 het teken op het scherm ongelijk aan CHR\$ (148) te zijn, dan springt het programma in regel 13 naar de omcodeer-routine in regel 34. Daar wordt getest, of het schermteken overeenkomt met de Griekse bèta (CHR\$ (162) op de NewBrain), die ik wil afdrukken als de Duitse 'lange' s (CHR\$ (126) op de printer). Is het inderdaad bèta, dan wordt 126 in A geladen en wordt in regel 37 teruggesprongen naar adres 29022 (regel 16). Daarvoor had in dit geval ook de label PRINT kunnen worden gebruikt (maar zie beneden). In regel 16 en volgende geeft RST 20h samen met de specifieke NewBrain-instructie DEFB 30h de print-opdracht, zodat het teken wordt geprint. Daarna wordt via STAP de volgende schermplaats gekozen.

Is het schermteken ongelijk aan CHR\$ (162), dan wordt via regel 35 (JP NZ = Jump indien verschil Not Zero) verder gestapt naar regel 38 en-voort, zodat achtereenvolgens A-umlaut, O-umlaut en U-umlaut worden getest. Een apart geval ontstaat in regel 50, als het schermteken gelijk is aan CHR\$ (205), dit is a met een kapje ofwel accent circonflexe. De printer drukt dan eerst a = CHR\$ (97) af (regel 52 en volgende). Dat gebeurt met de instructie CALL 29046, die werkt zoals GOSUB in basic. De subroutine staat in regels 29 tot en met 31; hij drukt het teken af zonder op het scherm verder te stappen en keert daarna door middel van

RETurn terug naar regel 54. Daarna wordt op dezelfde manier de control-code CHR\$ (8) 'afgedrukt', dat wil zeggen dat de printer een stap terug gaat. In regel 56 en volgende wordt dan het kapje (CHR\$ (94)) geprint. In dit laatste geval wordt niet met CALL gewerkt, omdat de letter klaar is en er dus wel weer verder moet worden gestapt op het scherm. In het geval van e-kap (regel 58) zou dezelfde procedure voor het kapje moeten worden gevolgd, maar kan ruimte worden bespaard door in regel 61 terug te springen naar de overeenkomstige procedure B205 van a-kap. Voldoet een teken van het scherm aan geen van de weergegeven gevallen, dan wordt het via regel 62 gewoon afgedrukt. Natuurlijk kan de omcodeer-routine veel langer zijn (mijn totale programma is ongeveer 20 maal zo lang!), maar in principe komt het allemaal op hetzelfde neer. Het blijft allemaal vrij eenvoudig, wat voor een beginner ideaal is. Iets ingewikkelder wordt het, als via de bit-image-mode van de printer zelf-ontwikkelde tekens moeten worden geprint. In feite komt het daarbij echter ook neer op een serie te 'printen' karakters en codes. Handig daarbij is de routine PR1 (regel 28 en volgende), die de vaak voorkomende ESCAPE-code 'print'.

Is het einde van de regel bereikt, dan is de inhoud van B gelijk aan nul en wordt van regel 21 naar 22 gegaan, waar het getal 48 in DE wordt geladen om dat in regel 23 bij HL op te tellen (direct optellen mag volgens het handboek niet). De optelling is nodig, omdat er tussen de laatste plaats op een schermregel en de eerste plaats op de volgende 48 ongebruikte geheugenplaatsen blijken te zitten, die moeten worden overgeslagen (uit te vinden met PEEK).



In regel 24 wordt C met 1 vermindert (er is immers een regel klaar) en in regel 25 wordt de inhoud van C in B gelezen, zodat met behulp van DJNZ (dat alleen op B werkt) in regel 26 kan worden gekeken, of C al nul is. Is dat het geval, dan is de opdracht (het printen van het scherm) afgewerkt en gaat het programma in regel 31 REToer naar basic. Hier wordt ook duidelijk, waarom in regel 4 bij C 1 is opgeteld. Anders zou namelijk al bij het begin van de zesde regel het programma naar basic zijn teruggegaan, daar DJNZ ook een vermindering met 1 inhoudt.

STARTEN

Er mist nog iets aan het verhaal, namelijk de start. Daarvoor wordt de Z80-code eerst in een DATA-regel geladen. Deze is weergegeven in figuur 3. Eerst wordt het startadres van het programma gegeven, samen met het aantal bytes. Daarop volgt de code, bij het printen weergegeven in decimale getallen. Deze DATA-waarden worden nu door middel van een basicroutine gePOKEt in het geRESERVEerde stuk geheugen, te beginnen met 33 = 21h op adres 28993. Om te starten wordt in basic de instructie CALL 28993 (of wat op hetzelfde neerkomt: CALL TOP) gegeven, waarna het programma gaat lopen.

Het bovenstaande geeft een indruk van de moeilijkheden. Het geeft aan, dat er processor-bepaalde en machine-bepaalde instructies zijn en waar die zijn te vinden. De werking van de processor is echter uitermate ingewikkeld en vaak blijken voor jou als beginner vanzelfsprekende routines niet te werken. De enige oplossing is om het dan maar op een andere manier te proberen.

Tot slot: geef het programma altijd gelegenheid naar basic terug te keren, anders loopt de boel vast en moet de computer worden uit- en aangeschakeld en de assembler opnieuw worden geladen. Om dat te voorkomen dient iedere subroutine van een RETURN te worden voorzien, evenals het programma als geheel. Bovendien is het verstandig om tijdens het programmeren hier en daar een routine aan te roepen die op de stoptoets reageert, vooral in loops.

Programma's die iets langer zijn dan het in figuur 2 afgedrukte, passen niet in een keer in de assembler, in feite omdat het geheugen van de NewBrain te klein is. In mijn geval kon dit worden opgelost door met de eerste helft van het programma steeds andere stukken omcodeerroutine uit te proberen. Die stukken worden dan steeds in andere DATA-regels gezet, die achter elkaar worden gePOKEt. Daarbij moet er op worden gelet, dat ze door middel van ORG op de juiste plaats achter elkaar worden gezet. Bovendien kan bij verwijzen naar andere stukken niet van labels gebruik worden gemaakt, daar de assembler dan niet weet, wat bedoeld wordt. Daarom moet met adressen worden gewerkt, zoals bijvoorbeeld in regel 37. Als de object code eenmaal werkt, dan is het assemblerprogramma niet meer nodig en kunnen na het POKEn ook de DATA-regels worden gEDELETE. Na verloop van tijd word je zo handig, dat je kleine veranderingen in de object code eenvoudig met POKE aanbrengt! Natuurlijk wel aan de hand van een listing als in figuur 2.

P. Kramer

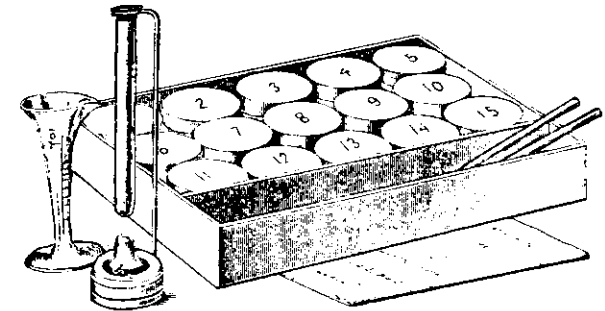
figuur 2: Z80-machinecodeprogramma geassembleerd op NewBrain

line	adres	z80-code	instructie	commentaar
1			ORG 28993	;SET OM FAN SKERM
2	7141	21 84 02	LD HL, #644	;IE SKERMADRES
3	7144	0E 06	LD C, #6	;TAL RIGELS
4	7146	0C	INC C	
5	7147	06 50	NIJ: LD B, #80	;RIGELLANGTE
6	7149	7E	WER: LD A, (HL)	;FAN SKERM
7	714A	1E 09	LD E, #9	;SET STREAM 9
8	714C	32 96 7F	LD (32662), A	;FOAR RESET
9	714F	22 9A 7F	LD (32666), HL	
10	7152	ED 43 9C 7F	LD (32668), BC	
11	7156	FE 94	CP #148	;R500
12	7158	CA 60 71	JP Z, STAP	
13	715B	C3 79 71	JP A162	;NEI OMSETRUTINE
14				;
15				;PRINT- EN STAPRUTINE
16	715E	E7	PRINT: RST \$20	;HANDBOOK 182
17	715F	30	DEFB \$30	
18	7160	2A 9A 7F	STAP: LD HL, (32666)	
19	7163	ED 4B 9C 7F	LD BC, (32668)	
20	7167	23	INC HL	;STAP OP SKERM
21	7168	10 DF	DJNZ WER	;FOLGJ. STAP
22	716A	11 30 00	LD DE, #48	;OERSLEINE POS.
23	716D	ED 5A	ADC HL, DE	;OERSLEINE POS.
24	716F	0D	DEC C	;RIGEL KLEAR
25	7170	41	LD B, C	;FOAR DJNZ
26	7171	10 D4	DJNZ NIJ	;NIJE RIGEL
27	7173	C9	RET	;WER NEI BASIC
28	7174	3E 1B	PR1: LD A, #27	;ESCAPE
29	7176	E7	PR2: RST \$20	;PRINTET SUNDER STAP
30	7177	30	DEFB \$30	
31	7178	C9	RET	
32				;
33				;**OMSETRUTINE**
34	7179	FE A2	A162: CP #162	;DUTSKE S
35	717B	C2 83 71	JP NZ, A229	
36	717E	3E 7E	LD A, #126	
37	7180	C3 5E 71	JP #29022	;PRINT
38	7183	FE E5	A229: CP #229	;A-UML.
39	7185	C2 8D 71	JP NZ, A232	

nba

SNELLE EN COMPLETE ASSEMBLER VOOR DE EXPANDED NEWBRAIN

In mijn verhaal in On-line 6 heb ik een verbeterde versie van Z80BRAIN (assembler in basic) beloofd. Het heeft enige tijd geduurd, maar het resultaat mag er zijn, vind ik. Zoals vaker het geval is, is het beter het bestaande overboord te zetten dan het bij te spijkeren. Ik stelde vrij hoge eisen aan de nieuwe versie. Het heeft derhalve dertien maanden geduurd (met ondertussen ook nog een cursus), voordat het perfect werkte.



Mijn plan is om een schaakprogramma te schrijven en ik nam me voor om het maken van de assembler - in assembly - als voorbereiding te beschouwen. Bovendien is een goede assembler daarbij noodzakelijk. Ik heb het resultaat NBA gedoopt. Het is een assembler voor de expanded NewBrain, omdat van bepaalde ZCALLs gebruik gemaakt wordt, die de kale NewBrain niet heeft. Dit euvel kan natuurlijk verholpen worden: of we kopen allen een EIM, of het programma wordt aangepast. Bij voldoende belangstelling kan het laatste uitgevoerd worden.

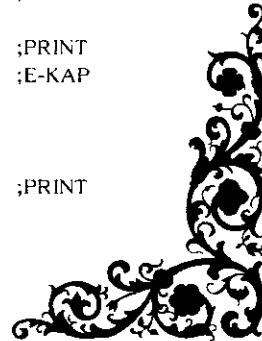
NBA is door de gebruiker aan te passen voor twee drives, cassetterecorder en printer omdat het openen van de devices in basic gebeurt. Vooraf nog een 'verontschuldiging': alle teksten die op het scherm om verschijnen zijn in Engels. Dit omdat voor bepaalde zaken geen korte Nederlandse omschrijving bestaat (default, editor etc.).

```

40 7188 3E 5B          LD A, #91
41 718A C3 5E 71      JP #29022 ;PRINT
42 718D FE E8        A232: CP #232 ;O-UML.
43 718F C2 97 71      JP NZ, A233
44 7192 3E 5C          LD A, #92
45 7194 C3 5E 71      JP #29022 ;PRINT
46 7197 FE E9        A233: CP #233 ;U-UML.
47 7199 C2 A1 71      JP NZ, A205
48 719C 3E 5D          LD A, #93
49 719E C3 5E 71      JP #29022 ;PRINT
50 71A1 FE CD        A205: CP #205 ;A-KAP
51 71A3 C2 B5 71      JP NZ, A210
52 71A6 3E 61          LD A, #97
53 71A8 CD 76 71      B205: CALL #29046 ;PR2
54 71AB 3E 08          LD A, #8
55 71AD CD 76 71      CALL #29046 ;PR2
56 71B0 3E 5E          LD A, #94
57 71B2 C3 5E 71      JP #29022 ;PRINT
58 71B5 FE D2        A210: CP #210 ;E-KAP
59 71B7 C2 BF 71      JP NZ, A167
60 71BA 3E 65          LD A, #101
61 71BC C3 A8 71      JP B205
62 71BF C3 5E 71      A167: JP 29022 ;PRINT
                        END

```

programlengte: 129 bytes
beginadres: 28993
eindadres: 29121



figuur 3: basic-listing

```

25000 DATA 28993, 129, 33, 132, 2, 14, 6, 12, 6, 80, 126, 30, 9,
50, 150, 127, 34, 154, 127, 237, 67, 156, 127, 254, 148, 202, 96,
113, 195, 121, 113, 231, 48, 42, 154, 127, 237, 75, 156, 127, 35,
16, 223, 17, 48, 0, 237, 90, 13, 65, 16, 212, 201, 62, 27, 231, 48,
201, 254, 162, 194, 131, 113, 62, 126, 195, 94, 113, 254, 229,
194, 141, 113, 62, 91, 195, 94, 113, 254, 232, 194, 151, 113, 62,
92, 195, 94, 113, 254, 233, 194, 161, 113, 62, 93, 195, 94, 113,
254, 205, 194, 181, 113, 62, 97, 205, 118, 113, 62, 8, 205, 118,
113, 62, 94, 195, 94, 113, 254, 210, 194, 191, 113, 62, 101, 195,
168, 113, 195, 94, 113

```

Nu dan de specificaties: het is een 2-pass assembler met een editor, welke beide via een menu aangeroepen worden. De assembler is in zijn geheel in machinetaal, en de editor waar het de geheugenbewerkingen betreft. Alles gebeurt in een oogwenk.

De source van NBA wordt meegeleverd. De code van de definitieve versie is aangemaakt met NBA: zoiets als de kip en het ei.

De assembler:

- ¶ verwerking van 70 tot 100 instructies per seconde
- ¶ alle Zilog-mnemonics met de pseudo-instructies: DEFM, DEFS, DEFB, DEFW, EQU, ORG en END. Speciaal voor de NewBrain: NSR <getal> = RST 32 + DEFB <getal> (NSR = NewBrain System Routine)
- ¶ de instructies hebben geen regelnummer
- ¶ diverse taistelsels en labels mogen vrij door elkaar gebruikt worden (decimaal positief of negatief, hexadecimaal, binair, ASCII)
- ¶ controle op:
 - de syntaxis van de instructie(delen) (op 12 punten)
 - de combinatie opcode-operand (op 29 punten)
 - de operand (op 69 punten)
- ¶ programmafouten worden gesignaleerd (meerdere ORG's, labeldefinitie niet, vaker of onjuist, relatieve sprongen). De foutmeldingen zijn in tekst en er wordt direct naar de foute regel in de editor gesprongen
- ¶ expressies en macro's zijn niet toegestaan

De editor:

- ¶ paginageoriënteerd met alle NewBrain controlcodes, plus:
 - bladeren door de sourcetekst per 22 regels
 - zoeken van regelnummers (GOTO)
 - het verwijderen van blokken (DELETE)
 - het zoeken van woorden in de source
- ¶ nieuwe regels worden direct op de syntaxis gecontroleerd
- ¶ overbodige spaties worden geaccepteerd en verwijderd. De source wordt dus zo compact mogelijk opgeslagen

Verdere mogelijkheden vanaf het menu:

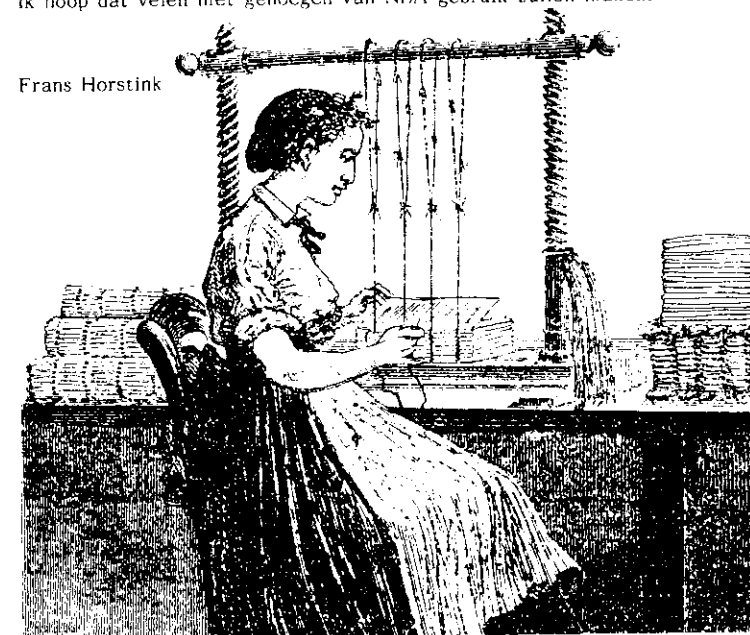
- ¶ programma's lezen en eventueel combineren (linken)
- ¶ source en code wegschrijven en eventueel verifiëren
- ¶ listing van de geassembleerde source (naar scherm of printer)
- ¶ opvragen van de .ASM-directory
- ¶ een labeldump: label met waarde/adres (naar scherm of printer)

Het werkgeheugen begint op 25000, zodat circa 24 k beschikbaar is voor de source, de labeltabel en de code. Hiermee kunnen ongeveer 1500 regels source verwerkt worden.

Op de schijf, waarop NBA bij de softwarebibliotheek van de gebruikersgroep te verkrijgen is (bestelnummer N230), staat:

- ¶ NBA.BAS, het assemblerprogramma
- ¶ NBA-ED.ASM en NBA-AS.ASM, de source van de machinetaalroutines
- ¶ NBA-ED.DAT en NBA-AS.DAT, de bijbehorende machinecode door NBA.-BAS te lezen
- ¶ Z80SET.ASM, de volledige z80-instructieset met enkele labeltoepassingen. Een controle op de juiste werking
- ¶ NBA-POK.BAS, een basicroutine om in uw programma's te MERGEN. Het leest de machinecode van schijf of tape
- ¶ NBA-POK.ASM, de source van de assemblyroutine in NBA-POK.BAS
- ¶ NBA-TOE.TXT, een toelichting op NBA (eigenlijk overbodig)
- ¶ DEV13.BAS om de toelichting te kunnen lezen of printen (van Wim Luijt)

Ik hoop dat velen met genoeg van NBA gebruik zullen maken.



MSDOS-schijven

Moeilijkheden met het omzetten van
(data)bestanden van CP/M naar MSDOS?

SPECIALE AANBIEDING
voor leden van de NewBrain-gebruikersgroep:

f 10,- per 360 kb MSDOS-schijf
(excl. drager en verzendkosten)
10 of meer schijven:
vraag prijsopgave

ook naar 3.5 inch, 1.2 en 1.44 Mb

Wim Luijt
Troelstrapad 22
4463 VM Goes
tel 01100-28197

videodriver

EEN NIEUWE BEELDSCHERMDRIVER VOOR DE
UNEXPANDED NEWBRAIN

dit artikel is een verkorte versie van de toelichting bij de driver
die in de softwarebibliotheek verschenen is op
cassette HULP-3 (diskette N227)

BESCHRIJVING VAN DE NIEUWE DRIVER

Deze nieuwe device driver kan worden gezien als een vervanging of als een uitbreiding van de normale screen editor. Bij het openen moet worden aangegeven van welke stream het eigen geheugen gebruikt gaat worden door de nieuwe driver. Deze driver werkt dus als een parasiet, net zoals de graphics driver dat doet. Indien het openen succesvol is, wordt het beeldschermgeheugen van de linked stream opgedeeld in drie gebieden, te weten de Status Area, de Display Area en de Message Area. Uitvoer naar en invoer van de driver resulteert dan ook in acties op de huidige area, terwijl de andere twee ongewijzigd blijven. De opdeling wordt echter alleen in het eigen geheugen van deze driver bijgehouden, zodat uitvoer rechtstreeks naar de linked stream precies doet wat het vroeger ook deed, alleen de plaats van de cursor kan zijn gewijzigd door in- of uitvoer van de nieuwe driver.

De driver kan zich in twee toestanden bevinden. Toestand 1 wordt de text mode genoemd en toestand 2 de field mode. Als de driver in text mode staat, vindt er op de uitvoer een zogenaamde word wrap plaats, zodat woorden niet aan de rechter kantlijn worden afgekapt. De invoer wordt beperkt tot het editen van één beeldschermregel. Continuation lines zijn dus niet meer mogelijk.

Als de driver zich in field mode bevindt, is gewone in- of uitvoer niet meer mogelijk. In plaats daarvan worden er speciale functies uitgevoerd, die bijvoorbeeld gestructureerde invoer mogelijk maken, zoals het updaten van een numeriek veld. Na een NewLine wordt dan ook gegarandeerd een numerieke string teruggegeven.



HET OPENEN VAN DE DRIVER

Het openen van de driver gebeurt met de volgende parameterstring:

[# strm] [S lines] [B lineno] [D bytes]

- # strm nummer van de stream van de videodriver waarvan het geheugen wordt 'overgenomen': de linked stream. Default is 0, de console.
- S lines het aantal beeldschermregels dat aan de Status Area wordt toegekend. Defaultwaarde is 1.
- B lineno het laatste regelnummer dat nog aan de Display Area wordt toegekend. Defaultwaarde is 20.
- D bytes het aantal geheugenlokatie's dat bij de Display Area wordt getrokken. Defaultwaarde is 1500. Dit wordt het geheugengebied waarin de velden worden opgeslagen.

Als de stream met succes is geopend, dan wordt het beeldscherm van de linked stream schoongemaakt en in de character mode gezet, die vergelijkbaar is met de Ctrl-WD mode van de normale videodriver. Dit houdt in, dat er een zwart beeld met witte letters ontstaat en dat de karaktercodes van 140 tot en met 255 de inverse zijn van 32 tot en met 127. Vervolgens wordt de Display Area geselecteerd voor toekomstige in- en uitvoer en de cursor wordt in deze area op de homepositie geplaatst. Als laatste wordt de autoclear mode aanzet.

UITVOER NAAR DE DRIVER IN TEXT MODE

De uitvoer naar de driver valt uiteen in twee onderdelen, controlcodes en gewone tekst. Alvorens de karakters worden verwerkt, wordt het hoogste bit, bit 7, op nul gezet. Dit houdt in dat karaktercodes 128 tot en met 255 niet door de driver worden ondersteund. Bit 7 wordt echter wel intern gebruikt om de tekst eventueel in inverse op het beeldscherm te plaatsen. Uitvoer van een byte naar de driver heeft tot gevolg, dat eventueel nog gereedstaande karakters voor invoer vanuit de driver worden geschrapt. Uitvoer van tekst vindt plaats op woordbasis. Een woord bestaat uit een karaktergroep (ASCII-codes 33 tot en met 127) en een spatiegroep (ASCII-codes 0 tot en met 32). Ook wordt een woord begrensd door een vraag om invoer van de driver. Uitvoer van numerieke strings wordt eerst door een conversieroutine op de lokaties 121 en 122 gehaald, alvorens te worden afgedrukt. Lokatie 121 bevat keuzes voor de landcode (Verenigde Staten, Engeland, Duitsland of Nederland), plaatsing van het geldsymbool voor het getal, scheidingssymbool

om de drie cijfers, spaties voor en na het getal, en plaatsing van een nul voor de decimale punt bij getallen tussen -1 en 1. Lokatie 122 bevat de veldlengte (een string die korter is dan deze waarde, wordt uitgevuld met spaties) en aanwijzing voor deling of vermenigvuldiging met 1.000, 1.000.000 of 1.000.000.000.

NIEUWE CONTROLCODES IN TEXT MODE

- 3 selecteer een andere beeldschermarea. Als parameter moet het areanummer worden opgegeven: 0 = Display Area, 1 = Message Area en 2 = Status Area
- 4 het volgende byte, dat met een GET-opdracht wordt ingelezen, is het huidige areanummer
- 5 de volgende vier bytes op de invoer zijn de grenzen van de huidige area. Teruggelezen worden achtereenvolgens: x- en y-coördinaat van de linkerbovenhoek en x- en y-coördinaat van de rechteronderhoek
- 6 verander het regelnummer van de laatste regel van de Display Area. Als parameter wordt in een byte de nieuwe laatste regel opgegeven
- 7 de volgende twee teruggegeven bytes vormen het absolute adres van de cursor. Eerst wordt het lage en dan het hoge byte teruggegeven
- 14 en 15 inverse video aan en uit voor alle uitvoer
- 19 en 20 inverse video aan en uit voor uitvoer in deze area
- 16 definieer een window binnen de huidige area. Hieropvolgende uitvoer vindt dan alleen nog plaats binnen deze window, totdat de window weer wordt opgeheven. Zie code 5 voor de vier parameters
- 18 verwijder een eventueel gedefinieerde window. Deze actie wordt tevens gebruikt door de controlcodes 3, 6 en 16
- 23 ga over op field mode. De Message Area wordt hierbij gewist
- 27 de mogelijkheid van autoclear mode aan- en uitzetten (toggle). Als default is autoclear mogelijk

12, 21, 22 en 31 werken alleen binnen de area.

Voor alle acties die te maken hebben met de cursor, geldt dat alle cursorcoördinaten worden opgegeven als offset 1 ten opzichte van de homepositie binnen de huidige area. Alleen voor de codes 5 en 6 geldt, dat de homepositie van de Status Area als home wordt genomen (hier worden de coördinaten dus absoluut gezien).



FOUTMELDINGEN VAN DE DRIVER

- 90 de opgegeven linked stream heeft niet een waarde die tussen 0 en 255 ligt
- 95 de opgegeven string is niet numeriek
- 110 er is een syntactische fout in de parameterstring geconstateerd
- 113 de opgegeven linked stream bestaat wel, maar is niet gekoppeld aan een device 0 of 4
- 115 de linked stream is gesloten. Deze driver is dus niet meer toegankelijk
- 122 de opgegeven linked stream is niet geopend. Derhalve kan deze stream ook niet geopend worden
- 123 het opgegeven regelnummer is te groot voor de linked stream
- 124 er is in field mode om invoer van de driver gevraagd, terwijl er geen invoer meer gereed staat om opgevraagd te kunnen worden
- 125 er is een fout opgetreden in de window-coördinaten
- 126 het gedefinieerde veld is te groot om het op het scherm te plaatsen of het veld heeft de lengte 0
- 127 het opgegeven getal past in het gegeven formaat niet in het veld of er zit een fout in de opgegeven formaatcodes
- 128 de linked stream is dusdanig van grootte veranderd, dat in- of uitvoer naar deze driver zou resulteren in een systemcrash
- 129 er is niet voldoende geheugen in het eigen geheugen over om dit veld nog toe te voegen
- 160 velden in de veldadministratie mogen niet overlappen. Het veld wordt dus niet gedefinieerd
- 161 bij dit type veld is een lege string niet toegestaan
- 162 een veld van type 5 moet aan een veld van type 4 gekoppeld worden
- 163 het veld valt bij het printen op het scherm (gedeeltelijk) buiten beeld
- 164 het opgegeven veld- en indexnummer specificeren geen waarde in een waardenverzameling
- 165 voor een veld van type 5 zijn nog meer waarden nodig om de definitie volledig te maken (het gekoppelde veld van type 4 heeft er ook meer), of er is geen geheugen meer om nog een waarde aan een veld van type 5 toe te voegen. Enige remedie: stream sluiten en opnieuw openen met meer geheugen
- 166 de opgegeven veldenrange kan niet worden gedefinieerd



FIELD MODE

Voor gewone tekst geldt, dat deze wordt genegeerd, tenzij de driver in field mode met een controlcode is verteld, dat hij de geprinte tekst op een bepaalde manier moet verwerken. Uitvoer vindt ook niet meer op woordbasis plaats, maar er wordt zoveel tegelijk genomen, als maximaal tegelijk verwerkt kan worden. De rest wordt als overvullige ballast genegeerd. Er zijn onder andere controlcodes voor:

- het opvragen van het aantal gedefinieerde velden; de status, waarmee het editen afgesloten is; de onder- en bovengrens van de gedefinieerde range en de zichtbare range; het veldnummer van het laatst gewijzigde en van het laatst geselecteerde veld; de status van een veld.
- het definiëren van de status van de velden; van een of meer velden als de huidige range; een veld als het laatst geselecteerde of het laatst gewijzigde veld.
- het afdrukken onder op het scherm van een standaard of een meege-stuurde pauzetekst.
- het toevoegen van een veld in de administratie. Parameters zijn: type (tekst, boolean, numeriek, numerieke of tekstconstante, waardenverzameling, picturestring (nog niet volledig ondersteund)) en status (gewoon, invers, niet tonen, niet selecteren, niet wijzigen, wissen bij aanvang van editen) van het veld; x- en y-coördinaat van het veld; plus een aantal optionele, afhankelijk van het veldtype.
- het verwijderen van velden uit de administratie of van het beeldscherm.
- de overgang naar Ask-Text, Ask-Boolean en Ask-Real mode. Formaatcodes, prompts en defaultwaarden moeten meegegeven worden. Als alle parameters zijn verzameld, gaat de driver over op de bij dit veld behorende Edit mode.
- het plaatsen van de driver in een sub-mode, waarin van buitenaf nieuwe waarden aan de velden in de huidige range kunnen worden toegekend. Alleen sommige typen velden kunnen van een nieuwe waarde worden voorzien, mits zo'n veld niet is beschermd tegen wijzigingen. De omgekeerde actie is ook mogelijk. Hierbij worden de waarden juist aan de gebruiker teruggegeven.
- het door de gebruiker opnieuw laten intikken van de velden.
- het wissen van velden (alleen op het scherm en alleen sommige typen).

Bij de drie Ask-functies is het mogelijk om het editen in het veld te stoppen met de NewLine-, de Escape- of de Stoptoets. NewLine en Stop hebben als effect, dat de gewijzigde tekst in het veld klaar wordt gezet om teruggelezen te worden. Bij Escape worden de wijzigingen genegeerd en wordt de defaultstring onveranderd teruggelezen. Het is dus altijd mogelijk

een string terug te lezen, en u krijgt met de opdrachten

PUT #strm, 1 : GET #strm, X

in X de informatie met welke van de drie toetsen het editen is gestopt. Waar er in het voorgaande over een string wordt gesproken, wordt bedoeld een reeks karakters die worden afgesloten met een willekeurige controlcode. Het makkelijkst is om voor deze controlcode de NewLine te kiezen, zodat een simpele PRINT-opdracht voldoende is om de hele string aan de driver door te geven. Het is dus niet nodig om na zo'n PRINT-opdracht nog een PUT-opdracht te geven om de controlcode te genereren, tenzij de PRINT werd beëindigd met een puntkomma (;). In dit geval stuurt basic er immers geen NewLine achteraan.

Een Boolean veld wordt opgevat als een waardenverzameling met twee waarden. De ene representeert 'Nee' en de tweede 'Ja'. Voor de taal, waarin de woorden 'Ja' en 'Nee' worden afgedrukt, wordt gebruik gemaakt van de landcode in de geheugenlokatie 121. Het is mogelijk om de eerste letter van een waarde in de verzameling in te toetsen. De bijbehorende waarde wordt dan op het scherm geplaatst. Er wordt echter onderscheid gemaakt tussen hoofd- en kleine letters.

DE SELECT EN UPDATE SUBMODES

NewLine beëindigt de Select/Update submode. Stop ook, maar zet tevens de stopstatus. Escape doet hetzelfde als NewLine, maar zet tevens de escapestatus en bewaart het huidige veld niet als laatstgekozen veld. In Update submode schakelt Insert over naar de Edit-Field mode die bij het veldtype hoort. In deze Edit-modes gelden de daarvoor gestelde Edit-codes, op drie uitzonderingen na:

NewLine stopt editen en keert met de nieuwe waarde terug in deze submode. Stop bewaart ook de nieuwe waarde, maar keert niet meer terug in deze submode. Escape negeert de nieuwe waarde, zet de oude waarde terug in het veld en keert terug in deze submode.

Als u het eerste teken van een veld intikt, wordt de cursor op dat veld geplaatst. Hierbij wordt onderscheid gemaakt tussen hoofd- en kleine letters en het zoeken begint vanaf het huidige veld met wrap around. Dus als er twee velden zijn die met hetzelfde teken beginnen, wordt de cursor eerst op het eerste en de volgende keer op het andere veld gezet.

Jan Peter Dijkstra



eigen

AANVULLENDE INFORMATIE BIJ DE TECHNICAL NOTES OVER DE EIGEN GEHEUGENRUIMTE VAN DE DEVICES, SPECIAAL VOOR HET PROGRAMMEREN IN Z80 OP DE 96K NEWBRAIN

Device 0: TVIO
Device 3: LLIO
Device 4: TLIO
Omvang eigen geheugenruimte: 29 (device 3: 34; waarvoor de laatste vijf dienen, is niet bekend)

HEX offset	DEC	offset
00 AREANO	0	nummer van de video-area van de stream
01 TVMD1	1	TV mode
02 TVMD2	2	
03 ORIGDEP	3	kopie van DEP
04 DEP	4	getoonde hoogte in regels
05 LL	5	lengte van getoonde regel
06 EL	6	LL + extra lengte
07 FLAGS	7	vlaggen voor algemene doeleinden
bit0 TVMFLG		
bit1 ESCFLG		
bit2 YFLG		vlag: volgend byte is y-coördinaat
bit3 XYFLG		vlag: volgend byte is x-coördinaat
bit4 CSIFLG		
bit5 CNSFLG		vlag: device is console
bit6 IFLG		
bit7 SFTFLG		
08 EXFLGS	8	
bit3 NOSHOWFLG		vlag: geen schermuitlezing
bit4 CURSHOFLG		vlag: cursor tonen
bit5 SNDYFLG		vlag: geef y-coördinaat
bit6 SNDXYFLG		vlag: xy-coördinaat opgevraagd
bit7 SNDCCFLG		vlag: teken onder cursor opgevraagd
09 LN	9	regelnummer van het begin van de regel

0A	FRM	10	aantal regels tot de eerste getoonde
0B	WIN	11	offset tot begin van VF-venster
0C	INPB	12	pointers van input-string
0D	INPC	13	
0E	CHAR	14	tijdelijke opslag voor PEDIT
0F	DCHR	15	
10	S4STO	16	opslag van de inhoud van de slots
12	S5STO	18	
14	S6STO	20	
16	S7STO	22	
18	PARASIT	24	
19	CSTO	25	opslag van huidige positie
1A	BSTO	26	
1B	ESTO	27	
1C	DSTO	28	
1D	FLGSTO	29	gewone vlaggen. Dit is begin van video-RAM - 1
bit0	CRFLG		NewLine aangeslagen
bit1	HLDFLG		'Hold mode', prompt niet overschrijven
bit2	PGEFLG		'Page send' in uitvoering
bit3	OLFLG		eenregel VF-display of schermhoogte 1
bit4	INPFLG		editor in input-mode
bit5	LIFLG		VF-display ondersteund
bit6	CHRFLG		tekens te verzenden in input-mode
bit7	TVFLG		video-display ondersteund

Device 11: GRAPH, High resolution graphics
Device 33: GRAPH, zelfstandige High resolution graphics op EIM
Omvang eigen geheugenruimte: afhankelijk van de versie van het operating system

O.S. 1.4, 1.9 en 1.91

00	P1	0	eerste parameter
06	P2	6	tweede parameter
0C	P3	12	derde parameter
12	CP	18	kleur van de pen
14	CT	20	tijdelijke opslag van de kleur
16	MODEP	22	mode-parameter
18	CBACK	24	achtergrondkleur (1 byte)
19	SCWBYT	25	schermbreedte in bytes
1A	SCWTH	26	schermbreedte
1C	SCHT1	28	aangevraagde schermhoogte
1D	SCHT2	29	benodigde schermhoogte
1E	XR	30	X Range = floating-pointequivalent van SCWTH

24	YR	36	Y Range = floating-pointequivalent van SCHT
2A	SCSTART	42	begin van TV-RAM
2C	SCTOP	44	laatste adres van TV-RAM
2E	OSTRM	46	streamnummer van de gekoppelde driver
2F	MYSTRM	47	eigen streamnummer
30	JFLAGS	48	toont wanneer bepaalde lokaties van kracht zijn
bit0	CFLAG		vlag: CL van kracht
bit1	INVFLG		vlag: CL geïnverteerd
bit2	PFLAG		vlag: verschil x, y positief
bit3	FFLAG		vlag: start
bit4	DLXBIT		vlag: x-stap = 1
bit5	DLYBYT		vlag: y-stap = 1
bit5	FIFULL		vlag: te weinig geheugen voor FILL
bit6	DEGTOG		vlag: gebruiker wenst graden
bit7	NARROW		vlag: schermbreedte (alleen bij OPEN)
bit7	OFFSCR		vlag: punt buiten het scherm (DOLINE)
31	CMD	49	huidige opdracht
32	CMSTATE	50	type van volgend byte dat van het operating system verwacht wordt
33	POFFSET	51	offset van IX naar P1, P2 en P3
34	OSDEP	52	'DEPTH' voor andere stream (alleen bij OPEN)
35	??	53	
36	SPARE	54	twee overgebleven bytes
38	DLXL	56	
3A	XL	58	
3C	XBIT	60	
3D	XBYT	61	
3E	DLYL	62	
40	YL	64	
42	YBYT	66	
44	YINC	68	
46	SPARE2	70	twee overgebleven bytes
48	XP	72	x-coördinaat van pen
4E	YP	78	y-coördinaat van pen
54	THP	84	richting van pen
5A	XT	90	tijdelijke opslag van x
60	YT	96	tijdelijke opslag van y
66	THT	102	tijdelijke opslag van theta
6C	XSF	108	schaalfactor van x
72	YSF	114	schaalfactor van y
78	XOD	120	afstand van x tot oorsprong
7E	YOD	126	afstand van y tot oorsprong

De volgende offsets worden gebruikt door ARC en andere routines die

MOVETO aanroepen.

84	THA	132	vergroting van de hoek
8A	PHI2	138	halve hoek gegeven in P2
90	R	144	straal van arc
96	DA	150	koordelengte (ARC); tijdelijke variabele (AXES)
9C	DX1	156	tijdelijke omvangvariabele (AXES)
A2	DY1	162	tijdelijke omvangvariabele (AXES)
A8	DX2	168	tijdelijke omvangvariabele (AXES)
AE	DY2	174	tijdelijke omvangvariabele (AXES)

Variabelen voor MOVE, MOVETO en DOLINE. (voorzichtig) verplaatsen naar het gebied P1, P2, P3

B4	XA	180	beginpunt van DOLINE: x-coördinaat
BA	YA	186	beginpunt van DOLINE: y-coördinaat
C0	XB	192	eindpunt van DOLINE: x-coördinaat
C6	YB	198	eindpunt van DOLINE: y-coördinaat
CC	CL	204	location berekenen
D2	FL	210	tweede tijdelijke opslag
D7	naamloos	215	voor het niet geïmplementeerde FRAME
D9	naamloos	217	dito
DB	naamloos	219	???

De volgende offset-locaties (gebruikt door de TEXT-routine) zijn nieuwe definities van de gebieden P1 en P2

TXTA	0	dot-offset van LHS van eerste tekstkarakter
TXTB	2	dot-offset van top
TXTC	4	aantal getekende tekens
TXTOPR	5	mode voor tekens, gecodeerde operator van PC, MODEP
TXTBYTE	6	beginbyte voor teken in TV-RAM
TXTBIT	8	te gebruiken bits in eerste byte
TXTFILL	9	1 + te gebruiken in volgende byte

De volgende offset-locaties (gebruikt door de routine FILL) zijn nieuwe definities van P1 en P2

CFILL	0
FIRST	1
FIFTOP	3
FIFNXT	5
FICOUNT	7

Nieuwe definities van het gebied P1, P2, P3 voor ARC

FIRSTX	6	kan dus SBMOV gebruiken
FIRSTY	8	
LBYTE	9	
LMEMORY	10	
CTEMP	12	

Nieuwe definities van de hoge eigen geheugenruimte voor ARC

RX	PHI2	
RY	R	voorzichtig!
CTH	DX1	
STH	DY1	
CD	DX2	
SD	DY2	

totaal 220 bytes

Verschillen in O.S. 2.0:

tijdelijke vlaggen:

30	JFLAGS	48	toont wanneer bepaalde locaties van kracht zijn
bit0	FFLAG		vlag: start (ook gebruikt in ARC)
bit1	INVFLG		vlag: CL geïnverteerd
bit2	PF _{LAG}		vlag: verschil x, y positief
bit3	CFLAG		vlag: CL geldt (ook gebruikt in ARC)
bit4	DLXBIT		vlag: x-stap = 1
bit5	DLYBYT		vlag: y-stap = 1
bit6	FIFULL		vlag: te weinig geheugen voor FILL
bit7	NARROW		vlag: schermbreedte (alleen bij OPEN)
bit7	OFFSCR		vlag: punt buiten het scherm (DOLINE)
35	SPARE	53	drie overgebleven bytes
46	SPARE2	70	een overgebleven byte

semipermanente vlaggen:

48	UFLAGS	71	gebruikersvlaggen
bit0	AXETOG		vlag: geen getallen bij AXES, indien gezet
bit7	sDEGTG		vlag: gebruiker wenst graden
D7	CENX	215	
DD	CENY	221	

totaal 227 bytes

verschillen in EIM:

tijdelijke vlaggen:

30	JFLAGS	48	toont wanneer bepaalde locaties van kracht zijn
bit4	SEND _F		vlag: DUMP in uitvoering
verder als in O.S. 2.0			
34	TXTDATA	52	opslag voor teken voor uitvoer
34	LHEAD	TXTDATA	voor ARC
35	GAPSIZE	53	twee bytes omvang van FILL-buffer
37	OSDEP	55	'DEPTH' voor andere stream (alleen bij OPEN)
46	PGFLAGS	70	vlaggen nodig in paged system
bit0	FORTY		equivalent aan andere stream 40 (s) / 80 (l)
bit1	STANDALONE		niet parasiterend

bit2 EIGHTY tegendeel van FORTY
bit3 LOADTV alleen bij OPEN benodigd
semipermanente vlaggen:
47 UFLAGS 71 gebruikersvlaggen
bit0 AXETOG vlag: geen getallen bij AXES, indien gezet
bit1 SHOW vlag: aan het begin van de uitvoering van elke opdracht de video area tonen
bit2 NOXYERR vlag: error 116 is geen fout
bit7 DEGTOG vlag: gebruiker wenst graden
48 PAGELOC 72 twaalf bytes nodig voor implementatie in paged system
48 DMPCOL 72 een byte offset van LHS om als volgende te versturen
49 DMPROW 73 een byte rij-offset van bovenkant
4A TMODE1 74 opslag voor TV-mode (tijdelijk in parasiterende, permanent in zelfstandige device driver)
4B AREANO 75 nummer van video-area
4C BLANKSIZE 76 ongebruikte area in bytes
4E SCRSIZE 78 bruikbare area in bytes
D7 CENX 215
DD CENY 221
E3 XP 227 x-coördinaat van pen
E9 YP 233 y-coördinaat van pen
totaal 239 bytes

Device 5: KBWIO, toetsenbord-I/O-routine (wacht op toetsaanslag)
Device 6: KBIIO, onmiddellijk terugkerende toetsenbord-I/O-routine
Non-paged IOS
Omvang eigen geheugenruimte: 1 byte

00 naamloos 0 vergelijkbaar met KBMODE van device 0

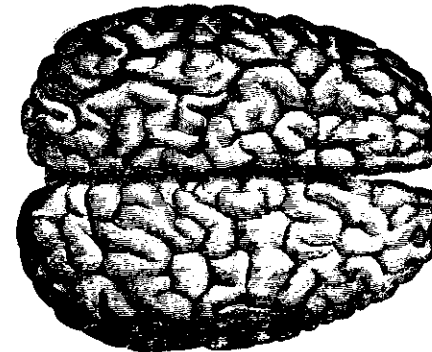
Mode	uitvoer naar KBWIO/KBIIOO	Control-toets voor editor
0	0 of 8	0 of *
1	1 of 9	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	-	8
9	-	9

Paged IOS en Non-paged 1.9 Frans
Device 5: NKBWIO, toetsenbord-I/O-routine (wacht op toetsaanslag)
Device 6: NKBIIIO, onmiddellijk terugkerende toetsenbord-I/O-routine
Omvang eigen geheugenruimte: 3 bytes

00 naamloos 0 als Non-pged
01 naamloos 1 geen functie (om KLOOK compatibel te houden met device 19)
02 naamloos 2 opslag voor accent als O.S. 1.9 Frans aangesloten is; geen functie als O.S 1.4 1.91 aangesloten is

Device 7: PORTIO, Z80 user port IO in PIOS
Omvang eigen geheugenruimte: 2 bytes

00 naamloos 0 lage poortadres
01 naamloos 1 hoge poortadres
om te OPENen in BASIC: OPEN #e, d, [parm]
parm: "[Hx][Ly]"; x en y 0 - 255,
indien L is gebruikt, wordt d vervangen door y



Device 12: BDISCIO, Binaire disk-I/O
Device 13: TDISCIO, Tekst-I/O van disk
Device 14: RDISCIO, Random I/O van disk
Device 15: SDISCIO, Systeem-I/O van disk
Omvang eigen geheugenruimte: 41 + 128 bytes

00 FCB 0 het CP/M file control block
24 COUNT 36 positie van byteteller
25 COUNT2 37
26 COUNT3 38

27 COUNT4 39
 28 FLAGS 40 enkele vlaggen
 bit0 BINARY
 bit1 OUTPUT
 bit2 INPUT
 bit3 BWRITE
 bit4 CONTROL
 bit5 BUFON
 29 DMA 41 128 bytes buffer

Het File Control Block

DR	F1	...	F8	T1	T2	T3	EX	S1	S2	RC	D0	...	Dn	CR	R0	R1	R2
00	01		08	09	0A	0B	0C	0D	0E	0F	10		1F	20	21	22	23

DR drive-aanduiding (0 - 16)
 0: defaultdrive gebruiken voor het bestand
 1: auto disk select drive A
 2: auto disk select drive B
 16: auto disk select drive P
 F1 ... F8 bevatten de bestandsnaam in ASCII-hoofdletters, topbit = 0
 T1 ... T3 bevatten het bestandstype in ASCII-hoofdletters, topbit = 0
 T1', T2' en T3' geven het topbit van deze posities aan, T1' = 1 => Read only file, T2' = 1 => System file, niet in DIR-lijst
 EX bevat het huidige extentnummer, normaal op 00 gezet door de gebruiker, maar in het 0 - 31 tijdens bestands-I/O
 S1 gereserveerd voor intern gebruik door het systeem
 S2 gereserveerd voor intern gebruik door het systeem. Op nul gezet bij aanroep om te openen, te creëren en te zoeken
 RC recordteller per extent 'EX', krijgt waarde van 0 127
 D0 ... Dn door CP/M ingevuld, gereserveerd voor gebruik door het systeem
 CR huidige record om te lezen of te schrijven in een sequentieel operatie, normaal 0 gemaakt door de gebruiker
 R0 ... R2 optioneel random recordnummer in het bereik 0 65535, met overloop naar R2. R0 en R1 vormen een 16-bits waarde met R0 als lage byte en R1 als hoge byte

Device 16: ACLPIO, RS-232C seriële I/O (printer)
Device 17: ACVIO, RS-232C seriële I/O (modem)
Omvang eigen geheugenruimte: 14 bytes

00 FLAGS 0
 bit0 INPUT

bit1 WAIT
 bit2 FLAGHS
 bit3 DUPLEX
 bit4 INTMODE buffer
 bit5 LP
 bit6 NOTACIA ongebruikt?
 bit7 HALFDX
 01 LPLEN 1
 02 LPHEAD 2
 03 ACIA 3 GBS Network Module
 04 CONTROLPORT 4 dito
 06 BUFSIZE 6
 07 ACWT 7
 CWORD ACWT een tegelijk nodig
 TEMP CWORD de code voor data, stop en pariteit om CWORD te maken na het parsen
 COUNT2 ACWT + 1 dito
 09 STREAM 9
 0A TBR 10
 TXBRP TBR
 0C RBR 12
 RXBRP RBR
 0D EXFLAGS RBR+1 vlaggen gebruikt na berekening van RBR
 bit0 ADDLF 0 alleen bij ACLPIO, om meer dan 1 LF na CR te onderdrukken

Device18: SSEIO, Eenvoudige Screen Editor
Omvang eigen geheugenruimte: 7 bytes

00 AREANO 0 nummer van de video-area van de stream
 01 TVMD1 1
 02 FLAGS 2
 bit0 TVMFLG
 bit1 ESCFLG
 bit2 YFLG
 bit3 XYFLG
 bit4 CURSHOFLG vlag: cursor tonen
 bit5 NESC
 bit6 IFLG
 03 CHAR 3
 04 CSTO 4 opslag van huidige positie
 05 BSTO 5
 06 TEMP 6 tijdelijk voor SETXY-adres

Device 19: KBBFIO, Gebufferd Toetsenbord
Omvang eigen geheugenruimte: 4 bytes

00 KEYMODE 0
01 FLAG 1 tekens in buffer
02 KLOOKMD 2 door KLOOK gebruikt voor opslag van accent
(alleen in O.S. 1.9 Frans, cf. device 5 / 6)
03 ESCHAR 3 om 0 te onderscheiden van CHAR = 0

Device 20: SSMD, Serieel Geheugendevise
Omvang eigen geheugenruimte: 5 bytes

00 MMSDIR 0 objectnummer van de directory
02 ACPTR 2 access pointer
04 FLAGS 4 vlagbyte
bit0 OUTPUT
bit1 INPUT
bit2 SAVED
bit3 DIRECT

Device 21: CENTR, Centronics Driver
Omvang eigen geheugenruimte: 5 bytes

00 FLAGS 0
bit0 LINE0 regellengte = 0
bit1 WAIT
bit2 INTMODE gebufferd
bit3 ADDLF
01 BUFSIZE 1
02 PORT 2 alleen bij GBS Network Module (cf ACIA, device
16 / 17)
03 LL 3
04 PHPOS 4

Device 22: APIO, Analoge Poort-I/O
Omvang eigen geheugenruimte: 0 bytes

Device 23: SNDOUT, Expansion Box Sound Driver
Omvang eigen geheugenruimte: 0 bytes

samenstelling: Wim Luijt
(vertaling: menno stevens)



ode

ODE AAN SINT PANCRAS

Vorige herfst heb ik, daartoe aangezet door het gebrek aan informatie over de Expansion Interface Module, ten lange leste een gelofte gebroken en ben ik ervoor gaan zitten om de beginselen van z80-assembler te leren. Gewapend met de door Open Stream gepubliceerde bios-listings heb ik een poging ondernomen om de 'Berlijnse muur' neer te halen, die er leek te staan tussen de hogere talen, die ik onder cp/m gebruik (pascal, forth en C), en het operating system, waar ik tamelijk vertrouwd mee was, toen ik met NewBrain-basic werkte. Ik wilde voornamelijk graphics gebruiken in pascal-programma's, en tot op zekere hoogte toegang hebben tot de stream-architectuur van de machine zonder dat hele vervelende CONFIGUR-programma te moeten doorworstelen. Daarvoor had ik enig begrip nodig van: (a) het mechanisme van het verwisselen van pagina's, en (b) het paged input / output system (PIOS). Het volgende artikel is een kort verslag van een aantal dingen, die ik geleerd en gedaan heb.

Terugkijkend realiseer ik me, dat de beschikbare documentatie niet zozeer onvolledig is, als wel te spartaans, en voorbeelden en behulpzame metaforen veracht. Liever dan deze ondeugd van systeemprogrammeurs te volgen, zal ik proberen specifiek en bloemrijk te zijn. Mijn programmeerpogingen in elk geval zijn altijd het gevolg geweest van praktische behoefte en niet van theoretische belangstelling, dus mijn kennis is beetje bij beetje verworven. Ik zal moeten aannemen, dat de lezer bekend is - misschien zelfs beter bekend dan ik - met de hoofdzaken van het operating system, zoals dat te zien is van de basic-kant van de muur. Ik houd me hier bijna alleen bezig met cp/m en het interface.

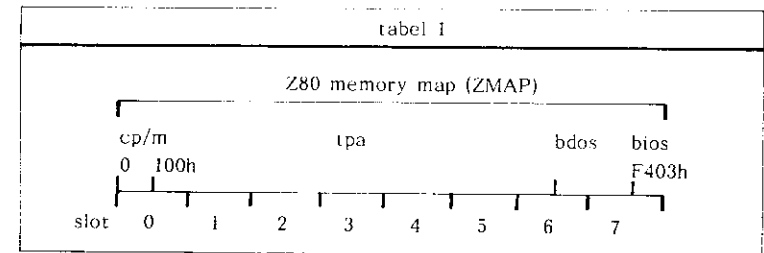
De 'system page' is het uitgangspunt; 8 k met tabellen, adressen, routines, vlaggen en pointers in het 96k systeem, die met gemak gePEEKt en gePOKEt kunnen worden door basic of een machinetaalroutine die niet onder cp/m draait. De eerste schrik is te ontdekken, dat het weg is onder cp/m! Cp/m heeft vanzelfsprekend de eerste 100h bytes voor eigen gebruik nodig, en daarom is de system page . . . ja waar? (en nu we het daar toch over hebben, waar is het schermgeheugen; hoe werkt de editor, wanneer alle 64 k anderszins gebruikt wordt . . .). Je hebt niet erg veel hulp van het antwoord, dat dat allemaal per pagina in de ruimte van 64 k gezet wordt, als het nodig is (tabel 1 geeft een beeld hiervan).

De Expansion module bevat 64 k - dit is het geheugen, dat zichtbaar is voor cp/m-programma's - en de originele 32 k van de machine wordt beschouwd als speciale video-RAM; in theorie in elk geval. Merk op, dat niets van de adresruimte van de z80 door de ROM gebruikt wordt, of door andere code die typisch voor het operating system is. Een paar absoluut noodzakelijke dingen worden in het bios-gebied bewaard, en de RST-locaties onderin, zoals dat door cp/m geregeld is. De system page wordt gekopieerd naar een van de vier pagina's van 8 k van de kale NewBrain (en pikt zo wat video-RAM in). De disk- en andere in- en uitvoer beslaan het grootste deel van een volgende (weer een weg); dan zijn er tabellen ter waarde van 8 k voor memory management, device drivers enzovoort (nu zitten we op de g-snaar) en device 18 gebruikt de eerste 3200 bytes van de laatste pagina voor het schermgeheugen (geen wonder, dat de graphics nooit meer worden dan een token in de 96k!).

Ik vrees, dat je niet om machinetaal heen kunt, maar de benodigde routines zijn erg kort, en als ze eenmaal klaar zijn, kunnen ze worden ingebouwd in functies in elke hogere cp/m-taal, die machinecode ondersteunt. Ik heb deze functies werkend gemaakt in C/80, Turbo-Pascal, xForth, en in BBC-basic (z80). Forth is speciaal aan te bevelen ter verkenning!

ENKELE ESSENTIËLE BEGRIPPEN

De z80 kan, zoals we weten, 64 k aan geheugenplaatsen adresseren - dit zal ik de ZMAP noemen. Het is belangrijk eraan te denken, dat dit logische adressen zijn, die onder hardwarebesturing (die ik met genoeg als een black box zal beschouwen) verwijzen naar verschillende fysieke geheugenplaatsen (namelijk chips). De ZMAP is verdeeld in 8 slots (0 tot en met 7), die beginnen met de adressen 0000h, 2000h, 4000h, 6000h, en zo voort tot E000h voor SLOT7. Het fysieke geheugen, dat het slot bezet, is een pagina, en iedere pagina heeft een uniek nummer. Je kunt als het ware een chip op het board aanwijzen en zeggen: 'Dit is pagina 125'. Er kunnen echter, wat de z80 betreft, maar 65536 adressen logisch onderscheiden worden, dus het is niet voldoende om te zeggen, dat een bepaald byte op adres A234h staat. Wat we wel kunnen zeggen, is dat een bepaald byte, dat op pagina 110 staat, op 234h bytes van het beginadres van het slot staat (dus op A234h), als die pagina in slot 5 zit. Deze combinatie van adres, slot en pagina, die met drie bytes aangegeven wordt, wordt het echte adres genoemd (real address), en dat definieert als enige een bepaald byte. De combinatie van slotnummer en paginanummer bij elkaar als een geheel van twee bytes (ik zal dat in het vervolg het



SPN noemen) komt des te meer voor, en het is noodzakelijk, dat u dat begrijpt. Als u DDT gebruikt om het geheugen vanaf F8AEh te dumpen, ziet u 16 bytes, die het SPN bevatten van de acht pagina's die op dat moment door cp/m gebruikt worden. Deze locaties heten CPM0 tot en met CPM7 in de listings (cf. tabel 2). Als u ook kijkt op lokatie F8C9h, ziet u nog een paar: het paginanummer van de system page (op dat moment in geen enkel slot), SYSP0 genaamd.

Technisch gesproken kan het paginanummer overlopen in de bits 1 tot en met 4 van het high byte. Het slotnummer is [slot * 20h], dat wil zeggen, het beslaat het bovenste 'nibble' van het hoge byte. Het vijfde bit is 1, als de pagina ROM is, en 0, als het RAM is. Deze opslagplaatsen voor de slots dupliceren de acht locaties op de system page van 8Bh tot 9Ah, die S0 tot en met S7 heten. Neen, die kunt u nu niet bekijken, omdat we de system page niet kunnen zien. Maar u zou ze vanuit basic kunnen PEEKen om te zien, hoe ze veranderd zijn. U kunt ook de SPNs van de ROMs nagaan (opgeslagen in B0 tot en met B5 op de system page). Voor serieuze studie van de system page raad ik u aan mijn programma SYSPAGE.COM(*) te pakken te krijgen, dat elk stuk van de pagina in DDT-formaat kan dumpen.

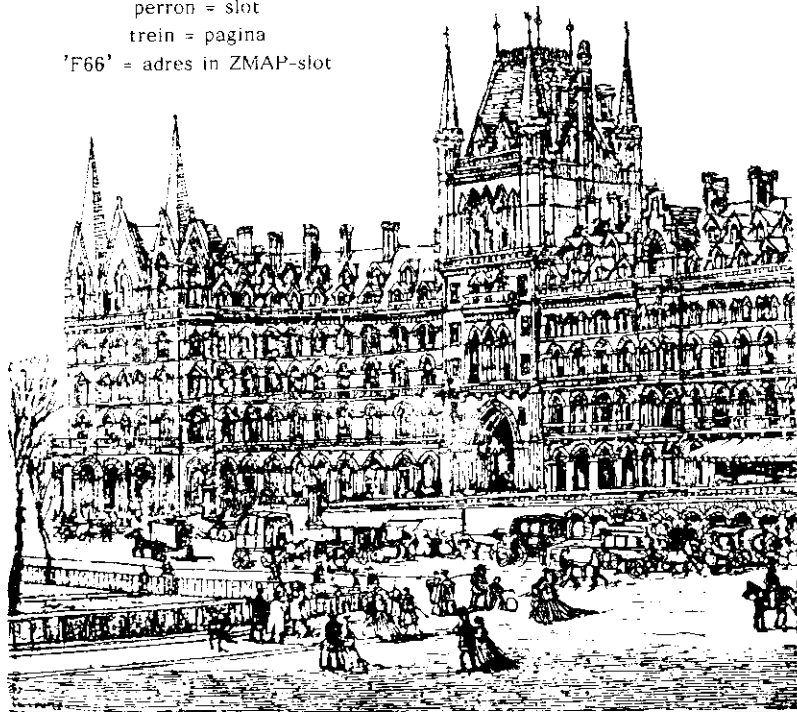
tabel 2		
naam	adres	waarde
cpm0	F8AE	61
	F8AF	00
cpm1	F8B0	68
	F8B1	20
cpm2	F8B2	67
	F8B3	40
cpm3	F8B4	66
	F8B5	60
sysp0	F8C9	60
	F8CA	00
SYSP0 heeft hetzelfde slot als CPM0, maar een andere pagina		

*) verkrijgbaar op bestelnummer N309 in de softwarebibliotheek (red.)

De softwareschakeling om een pagina in een slot te zetten gebruikt het SPN door het naar poort 2 te schrijven via de instructie OUT (C), <register>. C bevat de poort (2), B moet het slot bevatten en <register> de pagina.

Laten we eens even stoppen en een geschikte metafoor voor dit geheel zoeken. Ik heb het nuttig gevonden om het te beschouwen als een spoorwegstation. Stel u voor, dat u een gereserveerde plaats hebt: wagon F, plaats 66. Uw trein is die van 10.30 uur naar Rugby, en die moet vertrekken van perron 5. U vindt perron 5, daar staat de trein, en daar is plaats F66. Misschien merkt u pas, nadat u een andere passagier op plaats F66 aantreft, dat u in de verkeerde trein zit. Het is duidelijk: uw adres (perron 5, plaats F66) was niet voldoende. U had vergeten te controleren, of het wel de trein van 10.30 uur naar Rugby was (dat wil zeggen, de 'logische' reis). Ruw vertaald:

station = 64 k ZMAP
perron = slot
trein = pagina
'F66' = adres in ZMAP-slot

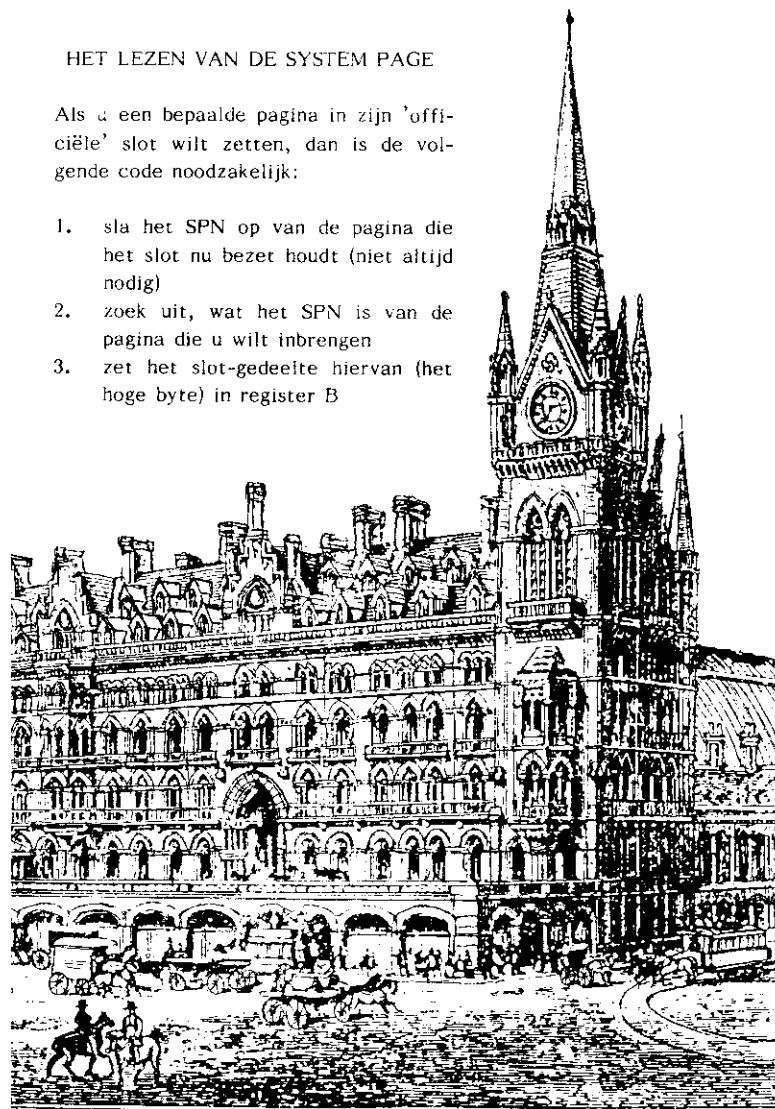


U hebt dus <10.30/Rugby + perron 5> nodig als SPN (Sint-Pancras-Nummer?) en voor het 'echte adres' is ook F66 nodig.

HET LEZEN VAN DE SYSTEM PAGE

Als u een bepaalde pagina in zijn 'officiële' slot wilt zetten, dan is de volgende code noodzakelijk:

1. sla het SPN op van de pagina die het slot nu bezet houdt (niet altijd nodig)
2. zoek uit, wat het SPN is van de pagina die u wilt inbrengen
3. zet het slot-gedeelte hiervan (het hoge byte) in register B



4. schrijf het paginagedeelte (het lage byte) naar het PAGEREGISTER (dat wil zeggen naar poort 2), dat staat in register C. Interrupts worden hiervoor in het algemeen onmogelijk gemaakt
5. doe wat u wilt doen met de pagina, die nu in het slot zit
6. zet de vorige bewoner van het slot terug (stap 2 tot 4)

```

SYSIN: LD HL, (SYSP0)
      LD B, H
      LD C, 2
      DI
      OUT (C), L
      EI

```

Klaar is kees! Alle lokaties van de system page zijn nu op de juiste adressen. U zou een byte kunnen lezen, op deze manier:

```
LD A, (000Fh) ;bijvoorbeeld het teken onder de cursor
```

of een tabel kopiëren (bijvoorbeeld die van de device drivers):

```

LD HL, TABLE;begin van de tabel op de system page
LD DE, STORE;waar u hem wilt opslaan
LD BC, BYTES;aantal te kopiëren bytes
LDIR

```

Zet de cp/m-pagina voor slot 0 (waarvan het SPN opgeslagen is in CPM0 op adres 0F8AEh) terug:

```

SYSOUT: LD HL, (CPM0)
      LD B, H
      LD C, 2
      DI
      OUT (C), L
      EI

```



Alles is weer zoals het was - en uw opgehaalde teken staat in A.

PROBLEMEN

Dat stel al te eenvoudige routines kan tot twee soorten problemen leiden. Het ene heeft betrekking op wie wanneer in welk slot zit, en wie door wie gezien kan worden. Het andere is het probleem van het aanroepen van

een subroutine, die zorgvuldig de pagina kan wegzetten, waarvan u dacht, dat u ernaar zat te kijken. Eerst het probleem van het zichtbaar zijn.

a) Uw code (zoals hierboven) zat waarschijnlijk bij 0100h. Dit is in slot 0, dus het is verdwenen na het wegschrijven naar de poort. De eerste en voor de hand liggende oplossing is om de ORG van de code op 2000h te zetten, aan het begin van slot 1 (of waar dan ook, als het maar niet bdos en bios overschrijft). In veel situaties produceert dit een ongewenste lege ruimte van 8 k op de schijf voor een verder erg korte routine. Hieronder wordt een alternatieve oplossing gegeven (of u kunt de code relocaten bij het laden).

b) Als u, terwijl de system page in slot 0 zit, een bdos-functie wilt aanroepen (in het algemeen de beste, en eenvoudigste, manier om in- en uitvoer te doen), dan kan dat NIET - omdat lokatie 05h nu niet het bdos entry point is. Twee oplossingen:

1) gebruik een systemcall van het operating system (ZCALL) of doe een directe bios-call om de klus te klaren (het adres van de jump table en dergelijke zou ergens opgeslagen kunnen zijn); of

2) gebruik dezelfde oplossing, die ik hier voorstel voor probleem (a), namelijk de system page in een ander slot zetten. Dit biedt echter niet het hoofd aan het probleem van de subroutine - u moet dus enig geduld oefenen.

EEN PAGINA TIJDELIJK IN EEN ANDER SLOT ZETTEN (NIET HET 'OFFICIËLE' SLOT)

De truuk van het officiële slot (het slotgedeelte van het SPN) is, dat als er code op de pagina staat (zoals te verwachten is in ROM, en goed mogelijk in RAM - bijvoorbeeld diverse routines aan het einde van de system page), die dan in het algemeen op het verwachte adres moet staan - zelfs als het relocatable is, moet het met CALL aangeroepen worden. Als data lezen of data schrijven het enige is, dat u met deze pagina wilt doen, dan is dit een academisch probleem en kan de pagina net zo goed in elk ander slot gezet worden, dat niet in de knoop komt met ofwel de code, die het gebruikt, ofwel enige code verderop, die in hetzelfde slot gezet zou kunnen worden. Als u, bijvoorbeeld, van plan was een pagina in slot 4 te zetten en u laat in- of uitvoer doen, dat zult u zien, dat ZCALLs of wat dan ook ervanuit gaan, dat ze slot 4 gebruiken, en de data, waarvan u verwachtte ze te kunnen lezen of schrijven, zijn

niet meer aanwezig! Dit is een deel van het subroutineprobleem, dat ik eerder noemde. De documentatie, die in omloop is, is hier echt heel duidelijk over, en in de praktijk blijkt, dat alleen de slots 3 en 5 voor dit doel bruikbaar zijn. Dus, als u bijvoorbeeld parameters van de system page wilt lezen, of het schermgeheugen naar de printer wilt sturen, kunt u als volgt te werk gaan.

```
SYS2TEMP: LD HL, (SYSP0)      ;of zou de videopagina kunnen zijn
          LD B, TEMPSLOTNUM;slotnummer * 20h
                               ;b.v. voor slot 3 is het 60h

          LD C, 2
          enz.
```

Nu moet u, wanneer u wilt lezen of schrijven, een offset erbij optellen, die gelijk is aan het eerste adres van het slot, bijvoorbeeld TEMPSLOT-CODE EQU 6000h voor slot 3 of 0A000h voor slot 5. Het is het beste om of een variabele of een EQUate te gebruiken voor TEMP (zeg 60h voor slot 3). Dan is de offset altijd TEMP.SHL.8

```
LD HL, 000Fh + TEMPSLOTCODE
LD A, (HL)      ;gewenste byte
enz.
CALL PRINT      ;subroutine die bdos of ZOUTPUT
                ;gebruikt of iets dergelijks
```

Dan terugzetten:

```
LD HL, (TEMPSPN)
LD B, H
LD C, 2
enz.
```

Dit is allemaal tamelijk primitief gedoe, maar het is een simpele methode om op de verstopte system page terecht te komen - speciaal wanneer u aan het interessante werk toekomt, namelijk de vertaling hiervan in pascal- of C-functies of forth-woorden.

Blijft nog het tweede probleem van de subroutines, die uw werk ongedaan maken. Elke system call (inclusief de bdos-calls) voert zijn eigen huishouding, en zet de pagina waarvan het SPN in S0 . . . S7 staat, terug in het slot. Dus de routine hierboven is, zoals hij nu is, niet te gebruiken om meer dan een byte te printen; het eerste byte wordt gelezen van de pagina die tijdelijk in het slot zit, en als we terugkeren van PRINT,

vinden we, dat de oorspronkelijke pagina weer in het slot zit! Mijn eerste oplossing was, om de routine een byte per keer te laten doen, maar er is een betere manier: zeg tegen het operating system wat u aan het doen bent. Dat wil zeggen, als u de pagina in slot n verwisselt, pas dan de lokatie S_n op de system page aan, terwijl u de oorspronkelijke pagina opslaat (hetzij op de stack, hetzij in het geheugen). In het volgende voorbeeld gebeurt dat.

Zet in slot 3 een videopagina, waarvan het SPN staat op lokatie VPAGE. VIDEOTOP, een adres op de system page, bevat wat op het ogenblik de enige beschikbare videopagina is. Kopieer dat naar VPAGE met de routines SYSIN en SYSOUT (hiervoor behandeld). Het SPN van slot 3 is zes bytes na CPM0. De lokatie op de system page is (precies zo) zes bytes na S0. We nemen aan, dat de opslag voor SLOTSAVE op de pagina in slot 0 is, evenals de code.

```
LD HL, (CPM0 + 6);haal de inhoud van slot 3 op
LD (SLOTSAVE), HL;sla het op
LD HL, (SYSP0)    ;haal paginanr. uit SPN van system page
LD B, 060h        ;maar vervang slotnummer door slot 3
LD C, 2           ;en verwissel de pagina's
DI
OUT (C), L
LD HL, (06000h + VIDEOTOP);kopieer het video-SPN nu het
                        ;zichtbaar is
LD (VPAGE), HL     ;en sla het op voor later
LD H, 060h         ;vervang slotnummer door slot 3
LD (S0 + 6 + 06000h), HL ;zet dit in S3 nu de system page
                        ;er nog is
LD B, H           ;zet nu de videopagina in het slot
LD C, 2
OUT (C), L
EI
```

Doe nu naar hartelust bdos-calls of ZINPUT of wat dan ook, we komen altijd terug bij slot 3 met de videopagina erin. Als we klaar zijn, moeten we de oorspronkelijke pagina terugzetten in slot 3:

```
LD HL, (SYSP0)    ;haal SPN van de system page op
LD B, 060h        ;zet dat in slot 3
LD C, 2
DI
OUT (C), L
```

```
LD HL, (SLOTSAVE);haal het bewaarde SPN terug
LD (S0 + 6 + 06000h), HL ;zet het in S3
LD B, H ;en gebruik het om de oorspronkelijke pagina
LD C, 2 ;in het slot terug te zetten
OUT (C), L
EI
```

En we zijn weer thuis. Merk op, dat de system page (S0 tot en met S7) bijgehouden moet worden, niet cp/m. Er moeten dus altijd pagina's verwisseld worden om het bijhouden uit te voeren. Aangezien onze code in slot 0 zat, moesten we een tijdelijk slot gebruiken, maar dat zou niet nodig zijn, als de code in, bijvoorbeeld, slot 2 zat.

Het is veel beter om een EQUate te gebruiken voor het tijdelijke slot, in plaats van alsmaar 6, 060h, 06000h enzovoort. We kunnen drie samenhangende constanten definiëren als

```
TEMP EQU 3
TSLOT EQU TEMP * 20h
TSLOTCODE EQU (TSLOT.SHL.8)
```

ZCALLS ONDER CP/M

Ik heb gladjes gesproken over het aanroepen van systeemroutines vanuit uw code. Ik neem aan, dat u weet, hoe dit moet, als u een machinetaalroutine schrijft om bovenop basic te werken. Het gaat net zo recht-toe-rechtaan vanuit cp/m. Lokatie 0F4FBh*) bevat een routine (het bestuderen zeker waard, als je de bios-listings te pakken kunt krijgen**), genaamd CPMZCALL, die net zo gebruikt kan worden als RST 32 met normale ZCALLs, namelijk:

```
CALL CPMZCALL
DEFB ZOUTPUT;30h
```

Dit bewaart alle registers, die u wel zult gebruiken, en de parameters

*) het is gevaarlijk dit adres te gebruiken; zie het vervolg van dit artikel en ook wim luijts artikel op pagina 9 (red.)

**) verkrijgbaar als bestelnummer N228 in de softwarebibliotheek van de gebruikersgroep; zie ook pagina 5 (red.)



zijn precies gelijk aan die bij ZCALLs. Bijvoorbeeld een aanroep van OPENIN moet het device in A hebben, de stream in E, de poort in D, het adres van de parameterstring in HL en de lengte ervan in BC. Slechts een ding moet in gedachten gehouden worden, houd de parameterstring wel zichtbaar voor de device driver, die in een van de slots gezet gaat worden. In machinecode betekent dit het gebruik van hetzelfde slot, waarin ook de code staat. Maar wanneer zo'n routine ingebouwd is in een hogere functie, die misschien een opslag van variabelen boven DFFFh gebruikt, dan komt u in de problemen, omdat de parameterstring verloren gaat. In Turbo-Pascal bijvoorbeeld is het nodig om de compileroptie te gebruiken, die het eindadres van het programma verandert, en dat op DFFFh of minder vast te stellen. In C zult u in het algemeen de stack een beetje omlaag moeten schuiven, enzovoort. Een sterk punt van forth is, dat u weet, waar de woordenlijst is!

Vergeet niet, dat zowel de ROM met het maths-pack als de grafische ROM op deze manier toegankelijk is. Om een grafisch pakket te implementeren, zult u tenminste de routines nodig hebben, die floating-pointgetallen converteren naar binaire, en ook iets moeten weten van het floating-pointformaat van de NewBrain*). Helaas zijn de diverse grafi-

*) cf. het artikel van wim luijt in newbrain on-line 7, pag. 57, en de correctie in newbrain on-line 10, pag. 19 (red.)

sche routines inherent langzaam, en ik heb noch een pascal-, noch een forth-programma gevonden, dat sneller cirkels trekt dan basic!

Een laatste punt over de CPMZCALL: hoewel het absolute adres, dat ik genoemd heb ik de het algemeen in orde is, is het niet de goede manier om met de dingen om te gaan. Op de echte cp/m-manier kan men de bios-sprongtabel raadplegen (net als voor CONIN, SECTTRAN etc.). Grundy heeft de tabel met vier aanroepen uitgebreid; twee zijn tweebytes adressen van essentiële diskbuffers, dan volgt JP XDISC en tenslotte JP CPMZCALL. U kunt het adres dus met de volgende standaardprocedure vinden.

```
LD HL, (0001h)    ;adres van sprongtabel
LD DE, ZCPM       ;((16 * 3) + 4 + 3
ADD HL, DE
INC HL            ;de JP overslaan
LD E, (HL)        ;lage byte
INC HL
LD D, (HL)        ;hoge byte
EX DE, HL
LD (CPMZCALL + 1), HL
...
...
CPMZCALL EQU $
DEFB 0C3h        ;JP
DEFW 0           ;het zojuist gevonden adres
```

Nu kunt u met een gerust hart CPMZCALL aanroepen.

Het paging system en CPMZCALL maken, met elkaar, het u mogelijk om alles te doen, waarvan ik zei, dat ik het nodig had: graphics herinstalleren, baudrates veranderen, streams openen en sluiten enzovoort. Het is natuurlijk veel gemakkelijker al deze dingen op hoog niveau te doen, als u eenmaal een paar primitieven als SYSIN, SYSGETBYTE, SYSOUT, OPEN, CLOSE enzovoort geïnstalleerd hebt. Ik hoop een paar voorbeelden hiervan te geven in een volgend artikel. Dit is allemaal voor dat rare beest de expanded NewBrain. Misschien kan iemand anders het equivalent van deze observaties voor de 32k cp/m schrijven?

Gordon Crosse

eerder verschenen in Open Stream 4, juli 1985; ook op schijf N402 in de softwarebibliotheek. (vertaling: menno stevens)

bios

FAST

FAST is door Henk Blik geschreven en zorgt voor een snellere uitvoer naar het scherm onder cp/m. Het is uitgebreid beschreven in On-line 5 (pag. 87 - 91). Hier volgt alleen een korte opsomming van de onderdelen van FAST. Het bestaat uit:

- ¶ een nieuw schermdevice op SYSP0 (NewBrain system page) die, op kleine wijzigingen na, gelijk is aan device 18 (SSEIO). Het is niet te gebruiken onder het paged operating system, omdat het raamwerk voor een volledige device driver ontbreekt. Het device gebruikt zijn eigen keyboard translation table (ook op SYSP0). De code is niet aaneengesloten, zodat er nog enige ruimte op SYSP0 beschikbaar is (479 bytes). Het line display wordt niet gebruikt
- ¶ nieuwe bios-routines: CONOUT en CONST (ook CPMZCALL en PUNCH zijn veranderd maar de wijzigingen zijn niet noodzakelijk)
- ¶ de interruptroutines in het bios en op SYSP0 zijn veranderd
- ¶ de plaats in het bios van een aantal routines is gewijzigd, zodat de beginadressen van deze routines niet meer overeenkomen met de oude. 'Stoute' programma's, die bios-calls doen door middel van een CALL naar een absoluut adres in plaats van naar de bios-sprongtabel werken hierdoor niet meer, bijvoorbeeld DUTIL (wel 'public domain', niet in de softwarebibliotheek) en programma's die het grafische scherm onder cp/m gebruiken (zie pag. 9 in deze On-line). Ik gebruik ze slechts zelden of nooit, zodat ik het niet de moeite waard vind om ze aan te passen. Bovendien is het twijfelachtig, of deze programma's onder FAST sneller zullen worden



Er zijn dus nog enkele verbeteringen mogelijk, maar daarover straks meer.

FAST21

Bij het samenstellen van de ROM-listings van de disc controller bleek mij, dat bepaalde tabellen (CPM7 en ROMPAGE: F8BDh-F8BFh) in het bios anders zijn met FAST dan zonder FAST. De oorzaak hiervan is de nieuwe interruptroutine. Deze gebruikt SPSAVE (F8C5h) als stack, wanneer naar SYSP0 wordt gegaan. De nieuwe routine zet zes adressen op de stack in tegenstelling tot de oude, die er maar vier opzette. Ik ben nog geen programma's tegen gekomen, die de informatie in CPM7 of ROMPAGE gebruiken of waarvan het falen op deze bug berust, maar het lijkt mij gewenst om een correctie aan te brengen. Ik heb FAST2 daarom zodanig veranderd in FAST21 (FAST2.1 kan helaas niet als filenaam), dat een niet gebruikt stukje bios onder de WBOOT-routine als stack gaat fungeren. Het resultaat staat in FAST21.COM.

Voor degenen, die FAST2 direct willen veranderen, staat hiernaast, hoe het moet.

```
DDT FAST2.COM <ret>
DDT VERS 2.2
NEXT PC
2000 0100
-sIFE <ret>
01FE C5 76 <ret>
01FF F8 F5 <ret>
0200 31 <ret>
0201 C5 76 <ret>
0202 F8 F5 <ret>
0203 F5 . <ret>
-s257 <ret>
0257 BF 70 <ret>
0258 F8 F5 <ret>
0259 18 . <ret>
- <ctrl-C>

A> SAVE 31 FAST21.COM <ret>
```

ANTIFAST

Voor het falen van programma's onder FAST heb ik twee mogelijke, niet onderzochte oorzaken aangegeven. Enige proefnemingen en nader onderzoek van de uitgewerkte source codes van de 96k NewBrain hebben de oplossing naderbij gebracht. De juiste oorzaak kan per programma verschillen en is zonder zeer intensief speurwerk niet na te gaan.

De storing wordt vrijwel zeker veroorzaakt door het feit, dat FAST de zogenaamde stream-device-structuur van de NewBrain gedeeltelijk negeert. Van de vier onderdelen, OPEN, INPUT, OUTPUT en CLOSE, wijzigt FAST alleen INPUT en OUTPUT en maakt een nieuwe own memory area, die niet overeenkomt met de reeds bestaande. Wordt nu overgeschakeld naar een grafisch device of naar een ander scherm- of toetsenborddevice, dan kan het operating system de zaak niet meer goed beheersen met alle nare gevolgen. Er is een mogelijke oplossing, door FAST weer de streamstruc-

tuur te laten gebruiken, waarbij ervoor gezorgd wordt, dat alle benodigde onderdelen benaderd kunnen worden, zonder dat er van het vertragende page switching gebruik wordt gemaakt. Met andere woorden

- CONIN en CONOUT gaan deel uitmaken van de device drivers 18 resp. 19, maar blijven wel rechtstreeks aanroepbaar
- het schermgeheugen blijft ongewijzigd
- alle overig benodigde geheugen moet zich op SYS0 of CPM7 bevinden

Niet duidelijk voor mij is, of het operating system toelaat, dat aan de genoemde voorwaarden kan worden voldaan. Het grootste vraagteken is voor mij, of het mogelijk is om het own memory naar een van deze pages over te brengen.

Een eenvoudiger oplossing (ook op grond van onderstaande overwegingen) lijkt mij om een soort ANTIFAST te maken. ANTIFAST zou dan de oorspronkelijke toestand moeten herstellen, zonder FEXIT en het hoofdmenu te gebruiken.



GEBRUIK VAN
GRAFISCHE
DEVICES
(11 EN 33)
ONDER CP/M

De 32k RAM in de NewBrain zelf worden in de 96k NewBrain gebruikt voor het schermgeheugen (ook het grafische), de own memory area's en eventuele buffers voor de verschillende geopende streams. Onder cp/m zijn altijd vijf streams geopend, waardoor er het nodige geheugen in gebruik is. Dit heeft als gevolg, dat de maximale lengte van een grafisch scherm onder cp/m (en dus bijvoorbeeld ook pascal, modula, fortran enzovoort) kleiner zal zijn dan onder basic. Hoeveel kleiner, hangt af van de geopende streams en vooral van de buffers behorend bij device 16 (of 21), 17 en 20 (hoewel ik daar nog nooit een toepassing voor heb gezien onder cp/m).

Bij het starten van cp/m op de 96k NewBrain worden vijf streams geopend. Cp/m verwacht namelijk vier logische devices (CON:, RDR:, PUN: en LST:), die op de 96k NewBrain via de (logische) streams 0 tot en met 3 gekoppeld zijn aan de devices 19 en 18 (CON:), 17 (RDR: en PUN:) en 16 of 21 (LST:). Stream 4 (device 3) is niet noodzakelijk (en op model A zelfs overbodig). Dat er toch altijd vijf streams geopend worden komt omdat de configuratieparameters op de originele systeemschijven zo zijn en het programma om deze parameters te wijzigen (CONFIGUR.COM) ook van vijf streams uitgaat. Het is echter niet noodzakelijk. Cp/m werkt ook op minimaal twee streams voor CON:, namelijk device 18 en 19. Wil je nog eens 'ouderwets' werken, kies dan voor stream 1 device 16 (of 21): leve de teletype!

Degenen die minder streams willen openen, kunnen PIOS.COM gebruiken. Eventueel kan ik bij voldoende belangstelling CONFIG4.COM aanpassen.

FAST is een wijziging van het BIOS en is bedoeld om de console (CON:) te versnellen, wat vooral belangrijk is bij intensief interactief gebruik (tekstverwerking!). Een sorteeropdracht bij dBASE (tijdens het sorteren geen uitvoer naar het scherm) gaat met FAST niet waarneembaar sneller.

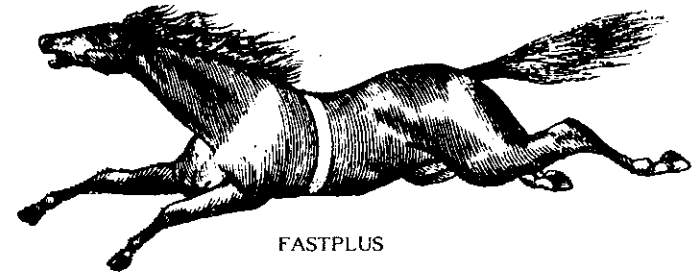
De traagheid van de grafische opdrachten (plot) onder cp/m is aan de traagheid van de grafische device drivers en niet aan het cp/m-bios te wijten. De oorzaken hiervan zijn:

- gebruik van floating-pointgetallen waar fixed point of integers volstaan
- intensief gebruik van ZCALLs
- intensief gebruik van maths-routines, speciaal de goniometrische
- device driver bevindt zich in hetzelfde slot als het grafische scherm, waardoor er veel page switching moet worden toegepast

Alle oorzaken gelden dus ook onder basic.

Verhoging van de snelheid van de grafische programma's kan dus niet verkregen worden door verandering van het bios, maar wel door verbetering van de device driver. Op het moment (eind juli 1988) heeft Marc Rouselle al een aantal snelle plot-routines werkend. Mogelijk kunnen deze op dezelfde wijze als FAST geïntegreerd worden tot een nieuwe device driver.

Het gebruik van FAST in combinatie met grafische programma's is, mede op grond van de kans, dat de programma's geheel of gedeeltelijk niet goed werken, nutteloos.



Gordon Crosse mag een van de beste NewBrain-programmeurs genoemd worden. Veel van de specifieke 96k NewBrain-programma's zijn van zijn hand. Enkele jaren terug heeft hij het programma SOFTKEYS geschreven, dat de 96k NewBrain van een aantal functietoetsen voorziet. SOFTKEYS is geen public domain, maar werd in de handel gebracht door GFC Microsystems. Zelf heb ik het niet, maar ik heb een demonstratie gezien door Philip Crookes op een van de gebruikersdagen. Als afscheid van de NewBrain heeft Gordon een public-domainversie gemaakt van SOFTKEYS en die gekoppeld aan FAST van Henk Blik. Dit programma is nu beschikbaar in de softwarebibliotheek als FASTPLUS (bestelnummer N313). Zelf heb ik hierin de bovengenoemde verbetering aangebracht. Verder behoort hierbij het programma EDITFAST om de betekenis van de toetsen interactief te kunnen wijzigen en een nieuwe FEXIT: FEXIT+.COM. Helaas werkt EDITFAST niet met FAST en FAST2.

Na analyse van de source code wil ik enige kanttekeningen plaatsen, omdat het programma van weinig nut voor mij is gebleken en het ook niet onder alle omstandigheden kan worden gebruikt.

In de opbouw van FAST is geen verandering aangebracht met uitzondering van de toevoeging van SK (gestripte, verbeterde versie van SOFTKEY) en een nieuwe CONIN in verband met het afdrukken van vaste strings. In SSEIO zijn nog drie nieuwe stuurcommando's aangebracht (cursor naar regel 25, gebruik van het geïnverteerde teken als cursor en een functie als PUT 21 : GET x).

Toegevoegd zijn een aantal functietoetsen, de mogelijkheid om van te voren gedefinieerde teksten (strings) met een toetsaanslag op het scherm te zetten, een stuk te markeren, gehele tekstschermen naar de printer te sturen of in een buffer op te slaan en een niet foutloos werkende 'cut & paste'. De laatste functies lijken erg veel op gelijke functies onder MSDOS.

Om FASTPLUS geen extra geheugen te laten gebruiken, wordt vrijwel al het 'vrije' geheugen buiten de TPA gebruikt. Gebruikt worden:

- ¶ de ruimte van niet meer gebruikte bios-routines (wat Henk Blik ook doet)
- ¶ ruimte die voor de administratie van de diskdrive beschikbaar is, maar niet gebruikt wordt, zolang er met maximaal twee 800k diskdrives in NewBrain-formaat gewerkt wordt. Gebruik van meer dan twee diskdrives of met Proton- of Joyce-formaat werken kan niet
- ¶ de diskbuffer wordt tot 512 bytes verminderd. Gevolg: geen Proton- of Osborne-formaat
- ¶ een vrij stuk boven de diskbuffer. Er zijn mij geen programma's bekend, die dit geheugen gebruiken, dus dit kan waarschijnlijk zonder bezwaar
- ¶ een door Henk Blik vrijgelaten stuk SYSP0
- ¶ als buffer om schermen op te slaan wordt video-RAM gebruikt. Als ik het goed heb, wordt dit geheugen normaal gebruikt om het eigen geheugen en de buffers van de geopende streams op te slaan: devices 3, 17, 16, 18 (ja nog steeds aanwezig, maar niet gebruikt) en 19 (werkloos)

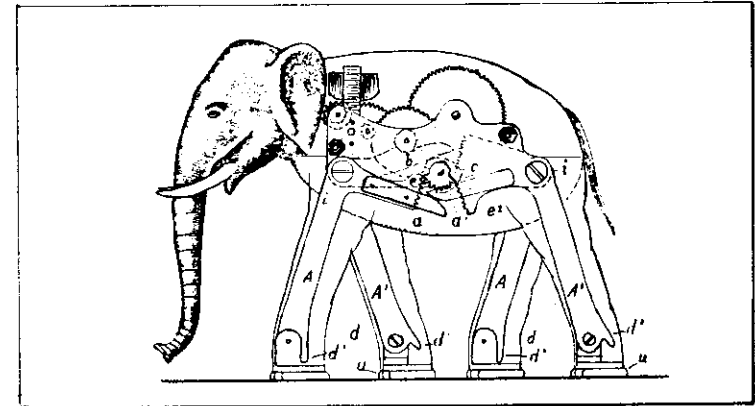
Mijn kennis van het paged memory operating system schiet te kort om te beoordelen of dit mag, maar gezien de problemen met 'cut & paste' vrees ik van niet.

In de bijbehorende documentatie en source code (en ook enkele andere programma's van zijn hand) vermeldt hij problemen met de opdracht PUT 22, x, y. Deze zijn te verklaren door de werking van het bdos. Als er voor $x = 9$ of $y = 9$ via bdos-call 2 of 9 een CHR(9) naar het scherm gestuurd wordt, maakt het bdos hier een aantal spaties van, die gedurende een call achtereenvolgens naar het scherm worden gestuurd (om dit te omzeilen is er in SSEIO PUT 27, 61, $y + 32$, $x + 32$).

Voor de meeste gebruikers is FASTPLUS dus goed te gebruiken, mits aan de volgende voorwaarden is voldaan:

- ¶ maximaal twee diskdrives in 800k NewBrain-formaat of kleiner*)
- ¶ devices mogen niet gebufferd zijn, dus in CONFIGUR device 16, 17, 21 met *0
- ¶ device 20 NIET gebruiken
- ¶ voorzichtig zijn met 'cut & paste'

*) voor WACD en ALVEC (zie On-line 10, pag. 60) zijn maar een beperkt aantal bytes beschikbaar: $2 \times 200k + 1 \times 800k$ kan ook nog. Voor andere combinaties kan ik het op verzoek uitrekenen.



BIOSPLUS

Bij FASTPLUS hoort het programma BIOSPLUS. Dit is helaas geen bestaand programma, maar een idee om het cp/m van de 96k NewBrain te verbeteren. Eerst enkele feiten om verwarring te voorkomen.

Het cp/m-bios van de NewBrain is nog in de periode van Grundy gemaakt en sindsdien niet gewijzigd of verbeterd. Het is op te vatten als een software interface tussen cp/m en het (paged memory) operating system van de NewBrain. Om het te gebruiken wordt het bij de 96k uit ROM geladen en in RAM gezet, terwijl het in de 32k grotendeels in ROM actief is. Het wordt dus niet door een 'bootstrap' vanaf schijf geladen. De andere delen van cp/m, bdos en ccp, worden bij iedere cold en warm boot vanaf schijf opnieuw geladen.

De ROMs in de EIM bevatten alleen routines behorend tot het paged operating system. De disc controller bevat twee ROMs. De eerste bevat het besturingsprogramma voor de Z80, die via de NEC 765 Floppy Disk Controller de diskdrive(s) bestuurt en de in- en uitvoer naar disk verzorgt. Deze ROM bevat het programma NDISC en zullen we ROM417 noemen. Er is een uitgebreidere versie van GFC Microsystems, waarvan er slechts enkele in Nederland zijn. De tweede ROM (ROM413) in de disc controller bevat het cp/m-bios voor de 32k en 96k NewBrain, de drivers voor de devices 12 tot en met 15, 18 en 19, en enige routines voor het starten. De devices 12 tot en met 15 vormen het disc operating system (DOS) van de NewBrain.

Dikwijls wordt dit DOS met 32k cp/m verward. Het DOS bevat de bios-diskroutines van 32k cp/m, aangevuld met eenvoudige routines genaamd CONST, CONIN en CONOUT. Ze zijn ongelijk aan de overeenkomstige routines in cp/m. CONST en CONIN, samen met de routine BDOS, zorgen ervoor, dat er een diskreset (vergelijkbaar met warm boot) kan worden uitgevoerd zonder de gehele NewBrain te resetten. CONOUT dient om een mededeling op het scherm (device 0) te kunnen zetten.

De routines van het DOS zijn voor de 32k en 96k vrijwel gelijk. Het cp/m-bios van de 32k is niet gelijk aan dat van de 96k NewBrain.

Voor zover ik weet, bevatten alle geproduceerde disc controllers dezelfde ROM-versie (ROM413: Issuc 2, ROM417: Issuc 1) en dus hetzelfde cp/m-bios. Er schijnen prerelease-versies en door GFC Microsystems verbeterde versies van ROM413 te bestaan, maar ik ga er van uit, dat deze niet in Nederland zijn verkocht.

Een andere uitzondering vormen misschien de unexpanded NewBrains, die voorzien zijn van CPM64, een produkt van Data Centre in Denemarken. Er zijn er een aantal in Nederland, maar ik heb geen informatie over de ROMs en RAMs en het bios van deze NewBrains.

BIOSPLUS gaat over het cp/m-bios van de 96k NewBrain en NDISC, samen kortweg bios genoemd. De belangrijkste verbetering van het bios vormt FAST van Henk Blik. Een andere verbetering is de toevoeging van functie-toetsen door Gordon Crosse (FASTPLUS). Beide verbeteringen zijn vrij van rechten voor niet-commercieel gebruik. XNDISC, een uitbreiding van NDISC, is niet vrij, maar de verschuldigde royalties zijn uiterst gering (mogelijk zelfs nihil).

Het bios bevat nog een aantal fouten: bijvoorbeeld VERIFY werkt niet, omdat er een verkeerd byte in NDISC bekeken wordt; het formatteren van dubbelzijdige schijven met Sideness = 1 of met 80 tracks per zijde kan problemen opleveren, omdat de disc controller vanuit ROM niet goed gestuurd wordt. Mogelijk zijn er nog ander fouten; weet je er, bel dan (01100) 28197 s.v.p.

Veel van de mogelijkheden van de NEC FDC 765 zoals single density, 8 inch disks, multisector en -track read en write, uitgebreide foutcontrole enzovoort om de diskin- en -uitvoer te versnellen worden niet gebruikt. Vrijwel alle informatie over het bios en de paged operating system is in de vorm van de source codes beschikbaar. De technische documentatie van de FDC 765 is beschikbaar.

FAST en FASTPLUS zouden eigenlijk standaard tot het bios moeten

behoren, waardoor ze na het starten niet van schijf behoeven te worden geladen. Bovendien zou de RAM dan efficiënter kunnen worden benut, liefst zonder dat er beperkingen aan het gebruik van diskdrives en buffers aan moeten worden verbonden.

De ccp wordt nu bij iedere cold en warm boot van drive A: geladen. Dit kost erg veel tijd. Als er in een van de pagina's nog vrije RAM te vinden is (scherm en ongebufferde devices gebruiken maar ongeveer 4k; er is 32k beschikbaar), dan zou de ccp vanuit een videopagina geladen kunnen worden. Bijkomend voordeel zou zijn, dat de CBOOT en vooral de WBOOT veel korter kunnen zijn, wat ruimte voor grotere bios-routines oplevert.

Veel gebruikte bios-routines (CONIN en CONOUT) zouden zonder (of met zo weinig mogelijk) dat er van pagina gewisseld moet worden, moeten kunnen. FAST is hierop gebaseerd en levert een aanzienlijke winst op. Als de door FAST gebruikte versie van SSEIO bereikbaar zou zijn, zonder dat er van pagina wordt verwisseld . . .

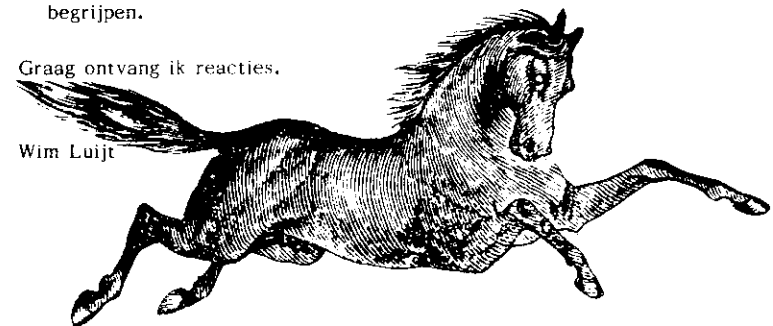
Minder gebruikte routines zouden dan naar andere pagina's kunnen verhuizen. Kortom er lijken nog mogelijkheden te zijn. Het bios is in RAM actief, wat als voordeel heeft, dat BIOSPLUS niet in ROM hoeft te staan, maar van schijf kan worden geladen.

Op grond van bovengenoemde punten zou ik graag het volgende horen:

- zijn er nog andere fouten in de bios
- zijn er suggesties voor andere verbeteringen
- is er belangstelling om deel te nemen in een eventuele werkgroep, zowel adviserend als 'programmerend', zodat de NewBrain nog beter kan worden dan hij al is. Je kunt hierdoor zowel cp/m als het voor microcomputers unieke paged memory operating system nog beter leren begrijpen.

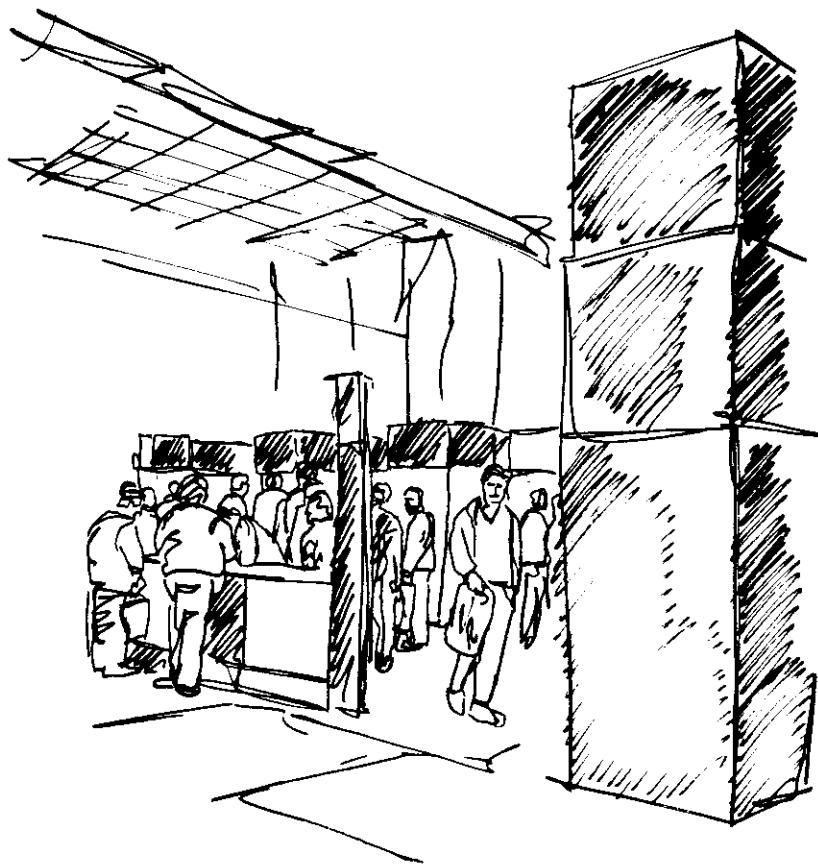
Graag ontvang ik reacties.

Wim Luijt



hcc-dagen 1988

25 en 26 november jaarbeurs utrecht



bezoek onze
kraam in de merwedeha!

**NewBrain-
gebruikersgroep**

schakelaar

HARDWARE-OMSCHAKELING 40/80 TRACKS TEAC FD-55F DRIVE

Met een kleine en simpele hardware-ingreep is het mogelijk om Teacs FD-55F drive omschakelbaar te maken voor 40 en 80 tracks. Het is dus nu heel goed mogelijk om zonder problemen 40 tracks te formatteren op een 80 tracks drive. Werk je, naast je NewBrain, nog met een IBM-PC (of kloon) dan is het helemaal makkelijk om je cp/m-diskettes op de PC te kunnen lezen en schrijven, maar ook om bij voorbeeld een backup te maken van je hard disk op de helft van je normale aantal diskettes.

	80 tracks	40 tracks
R16	10 Ohm	leeg
R14	leeg	10 Ohm



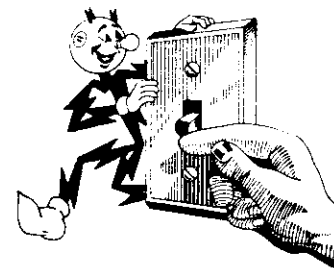
Door het weerstandje R16 (10 Ohm) uit de print te solderen en te vervangen door twee draadjes en door ook R14 te 'vullen' met twee draadjes, zijn we al een heel eind. Je verbindt deze vier draadjes (elk afzonderlijk!) aan de vier soldeerpunten aan de zijkanten van een 'dubbele om-en-om-schakelaar'.

Over de middencontacten van de schakelaar soldeer je het weerstandje van 10 Ohm. Staat de tuimelschakelaar naar boven dan heb je een 80 tracks drive en naar beneden een 40 tracks drive.

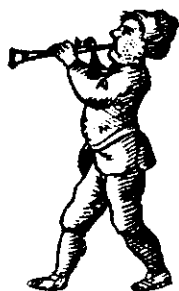
Er zijn kleine schakelaartjes te koop, die precies in de holle ruimte passen aan de voorzijde links onder van deze diskerteststations.

Het zou zelfs mogelijk zijn om op de 40 tracks versie (FD-55B) 80 tracks diskettes te lezen. Schrijven zou niet mogelijk zijn.

Jos Hermans



inhoud on-line 12



- 2 newbraindag in leiden
- 4 lijst van medewerkers
- 5 softwarebibliotheek: nieuw team
catalogus van de nieuwe software
- 9 waarom werken sommige programma's niet onder fast? /wim luijt/
- 12 turbo-pascal en 32k newbrain /wim luijt/
- 13 vraagjes en weetjes - vragen en weeten
toetsen versleten? /p kramer/
zcalls /frans horstink/
- 15 quickfil.bas een leerzaam spel /ger van laere/
- 23 communicatie met een msdos-machine /hans van hoek/
- 27 mijn weg tot assembler /p kramer/
- 35 nba: snelle en complete assembler voor de expanded newbrain /frans
horstink/
- 39 een nieuwe beeldschermdriver voor de unexpanded newbrain /jan peter
dijkstra/
- 45 aanvullende informatie bij de technical notes over de eigen geheugen-
ruimte van de devices, speciaal voor het programmeren in z80 op de
96k newbrain /wim luijt/
- 55 ode aan st pancras /gordon crosse/
- 67 bios: fast | fast21 | antfast | gebruik van grafische devices (11 en
33) onder cp/m | fastplus | biosplus /wim luijt/
- 77 hardware-omschakeling 40/80 tracks teac fd-55f drive /jos hermans/
- 79 bestelbiljet softwarebibliotheek

78

inhoud

newbrain on-line 12

NewBrain
gebruikersgroep
postbus 4494
1009 AL amsterdam

softwarebibliotheek

cassettesoftware	*) = ook verkrijgbaar op cassette; de andere van serie N... alleen op diskette		volumes uit de programmatheek van de cp/m- gebruikersgroep
— ASM-1	— N201	— N309	— ———
— BWK-1	— N202	— N310	— ———
— DIVERS-1	— N203	— N311	— ———
— DIVERS-2	— N204	— N312	— ———
— DIVERS-3	— N205	— N313	— ———
— DIVERS-4	— N206	— N401	— ———
— DIVERS-5	— N207	— N402	— ———
— DIVERS-6	— N208	— N501 *)	— ———
— EDUC-1	— N209	— N502 *)	— ———
— EDUC-2	— N210	— N503 *)	— ———
— EDUC-3	— N211	— N504 *)	— ———
— EDUC-4	— N212	— N505 *)	— ———
— FIN-1	— N213	— N506	— ———
— GRAFI-1	— N214	— N507	— ———
— GRAFI-2	— N215	— N508	— ———
— HULP-1	— N216	— N509 *)	— ———
— HULP-2	— N217	— N510 *)	— ———
— HULP-3	— N218	— N511 *)	— ———
— MATH-1	— N219	— N512	— ———
— MATH-2	— N220		— ———
— MATH-3	— N221		— ———
— MATH-4	— N222		— ———
— NBFILES	— N223		— ———
— ONLINE-5	— N224		— ———
— ONLINE-6	— N227		— ———
— ONLINE-7	— N228		— ———
— ONLINE-8	— N229		— ———
— ONLINE-9	— N230		— ———
— ONLINE-10	— N231		— ———
— SPEL-1	— N301		— ———
— SPEL-2	— N302		— ———
— SPEL-3	— N303		— ———
— SPEL-4	— N304		— ———
— SPEL-5	— N305		— ———
— SPEL-6	— N306		— ———
— SPEL-7	— N307		— ———
— UGV-1	— N308		— ———
— VIDITEL-1			— ———

per 10 delen
in één
bestelling
het 11e gratis

onderwijsbibliotheek

— SCHIJF-1*) — SCHIJF-6*)
— SCHIJF-2*) — SCHIJF-7*)
— SCHIJF-3*) — SCHIJF-8*)
— SCHIJF-4*) — SCHIJF-9*)
— SCHIJF-5*) — SCHIJF-10

PRIJZEN

alle software,
voor elk formaat, kost
f 10,- per bestelnummer,
behalve 200k-formaat uit de
programmatheek van de cp/m-
gebruikersgroep: dat kost
f 15,- per bestelnummer

plus bij bestelling per post
f 2,50 verzendkosten

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

formaat aankruisen:	_____ bestelnummers à f 10,00 = f _____
☐ cassette	_____ bestelnummers à f 15,00 = f _____
☐ diskette 200k	_____ bestelnummers à f 5,00 = f _____
☐ diskette 400k ss	verzendkosten f 2,50 f _____
☐ diskette 400k ds	
☐ diskette 800k	
☐ _____	T O T A A L f _____

bovenstaand totaalbedrag

☐ is overgemaakt d d _____

op postrek 2505800 tnv hcc newbrain-gebruikersgroep amsterdam

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



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