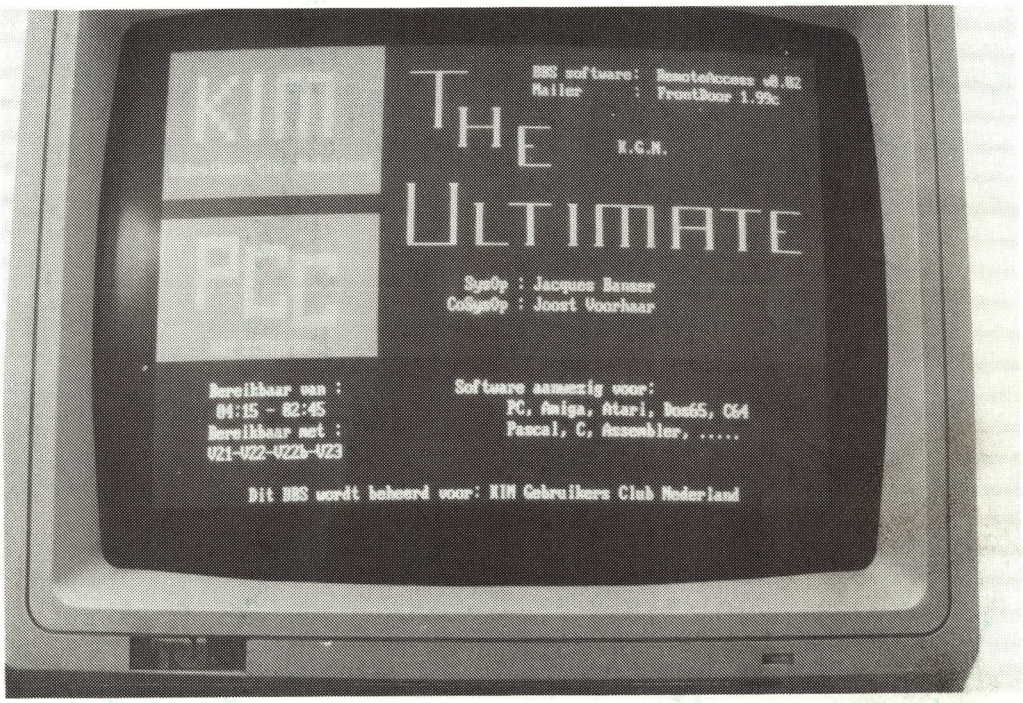
De µP Kenner

Veertiende jaargang nr. 4
Oktober 0x7C6
0x44



In dit nummer o.a.:

Nieuwe PC-FIX
Spreadsheet in PASCAL
Datacommunicatie deel 4
Het UNIX operating system
Keyboardbuffer voor DOS65

www.dottyflowers.com

Inhoud

De μP Kenner
Nummer 0x44, oktober 0x7C6
Verschijnt 5 maal per jaar
Oplage: 250 stuks
Druk: FEBO Offset, Enschede

De redactie:
Gert van Opbroek
Bram de Bruine
Antoine Megens
Nico de Vries
Joost Voorhaar

Eindredactie:
Gert van Opbroek

Vormgeving:
Joost Voorhaar
Nico de Vries

Redactieadres:
Gert van Opbroek,
Bateweg 60
2481 AN Woubrugge

De μP Kenner nummer 0x45 verschijnt op 22 december 0x7C6.

Copijsluitingsdatum voor nummer 0x45 is vastgesteld op 8 december 0x7C6.

Vereniging
Uitnodiging voor de clubbijeenkomst 5
Agenda algemene ledenvergadering 6
Van de bestuurstafel 49
Informatie 50

Algemeen
Redactioneel 4
De computerdokter 11

Datacommunicatie
Methoden en technieken voor datacommunicatie (deel 4) ... 44

Systemen
Een printerbuffer en een toetsenbordbuffer voor DOS65 ... 12
De IBM-PC en z'n klonen (Deel 10) 22

Talen/Software
Het standaard Operating System "UNIX" (deel 1) 7
To Share Or Not To Share, That's The Question 14
Het programmeren van de 8088 in de IBM (Deel 2) 17
Nu beschikbaar: versie 3.10 van PCFIX 27
REKEN, een spreadsheet(je) in Pascal voor DOS-65 29

De μP Kenner is het huisorgaan van de KIM gebruikersclub Nederland en wordt bij verschijnen gratis toegezonden aan alle leden van de club. De μP Kenner verschijnt vijf maal per jaar, in principe op de derde zaterdag van de maanden februari, april, augustus, oktober en december.

Kopij voor het blad dient bij voorkeur van de leden afkomstig te zijn. Deze kopij kan in papier-, maar liever in machine-leesbare vorm opgestuurd worden aan het redactieadres. Kopij kan ook op het Bulletin Board van de vereniging gepost worden in de redactie area.
Nadere informatie kan bij het redactieadres of via het bulletin board opgevraagd worden.

De redactie houdt zich het recht voor kopij zonder voorafgaand bericht niet of slechts gedeeltelijk te plaatsen of te wijzigen. Geplaatste artikelen blijven het eigendom van de auteur en mogen niet zonder diens voorafgaande schriftelijke toestemming door derden geplubiceerd worden, in welke vorm dan ook.
De redactie noch het bestuur kan verantwoordelijk gesteld worden voor toepassing(en) van de geplaatste kopij.

www.dottyflowers.com

Redactioneel

Nu de dagen korter beginnen te worden en de wintertijd ingevoerd is, breekt de tijd weer aan dat de barbecue naar zolder gaat en de computer en eventueel de soldeerbout van tevoorschijn gehaald worden. Dit is ook de tijd van de grote gebeurtenissen op het gebied van de hobbycomputer. In de eerste plaats zijn dat natuurlijk de HCC-dagen. Hoewel een aantal mensen binnen de club niet zo HCC-minded is, zijn de HCC-dagen toch wel gebeurtenis waar bijna iedereen die met computers bezig is naar uitkijkt. Zoals in μP Kenner 67 aangegeven is, is onze club ook weer officieel op de HCC-dagen vertegenwoordigd met een eigen stand. Weliswaar is de stand niet zo groot als vroeger, dat valt namelijk niet meer te betalen, maar we staan er in ieder geval weer. Dus als u op 30 november of 1 december de HCC-dagen bezoekt, dan wordt u van harte uitgenodigd even bij ons langs te komen.

Een tweede hoogtepunt is de algemene ledenvergadering. Deze zal op zaterdag 17 november in Almelo gehouden worden. Uiteraard treft u hiervoor in deze μP Kenner een uitnodiging aan. Op de agenda staat onder andere de toekomst van de club. Dit betekent niet dat het bestuur van plan is de club op te gaan heffen, ik mag wel zeggen integendeel. Het laatste jaar zijn namelijk enkele plannen boven komen drijven en we zijn zo'n twee maanden geleden voorzichtig begonnen die plannen een beetje uit te werken. Welnu hierdoor zijn we in een enorme stroomversnelling terecht gekomen. Met name het voorzichtig contact zoeken met de MINIX gebruikersgroep heeft enorme gevolgen gehad. Stonden we twee maanden geleden nog op het standpunt dat zelfbouw van hardware eigenlijk niet meer zinvol is, gisteren hebben we op de bestuursvergadering uitgebreid gesproken over de ontwikkeling van een 68000 of 68010 processorkaart die dan samen met goedkope (Taiwan) hardware uit de PC-wereld een low cost MINIX systeem moet vormen. Zoals het er vannacht om twee uur naar uitzag is het zeer waarschijnlijk dat dit project daadwerkelijk opgestart zal gaan worden.

Om dit project op te starten hebben we wel wat hulp van de leden nodig. Een 68010 processorkaart met een kloksnelheid van 16 MHz of iets dergelijks ontwerp je namelijk niet zomaar, zeker niet als op de processorkaart ook nog pakweg 2 MB aan RAM zit, een boot-ROM en minimaal één seriële poort. Ook het schrijven van de driver software voor de aansturing van de randapparatuur is iets wat gedaan moet worden. Voor het aanpassen van MINIX krijgen we dan hulp van buiten. De invulling en opbouw van dit systeem zal de komende tijd vorm beginnen te krijgen waarna we op de ledenvergadering hierover eens willen brainstormen.

Het grote voordeel van MINIX is het feit dat er een gigantische bibliotheek van (vrije!) software beschikbaar is. Tekstverwerkers, hogere programmeertalen, utilities... eigenlijk alles schijnt er te zijn. Binnenkort zal deze software ook via onze club en ons bulletin board ter beschikking komen (2 MB aan nieuwe software per dag!).

Vooruitlopend op de vergadering wil ik nu al vast een oproep plaatsen voor hulp bij het hele MINIX gebeuren. Uiteraard hebben we leden nodig die zich bezig willen houden met de verspreiding van de software. Verder hebben we ook hulp nodig bij de ontwikkeling van de eigen hardware voor het KGN-MINIX systeem en tenslotte hebben we hulp nodig bij de ontwikkeling van de systeem-afhankelijke software (drivers etc.). Kortom, heeft u tijd en voelt u er voor een bijdrage te leveren, laat mij dat dan even weten.

Hoewel ik helemaal lyrisch ben over het KGN-MINIX systeem wil ik niet de schijn wekken dat dit systeem de enige manier is om met MINIX te werken. In tegendeel, MINIX is van de plank beschikbaar voor de PC-wereld, voor Atari ST, voor Amiga en nog veel meer systemen. Het is niet de bedoeling om MINIX naar DOS-65 te porteren. Voor een dergelijk operating systeem is een 6502 waarschijnlijk net even te krap.

Hoewel ik bijna uitsluitend over MINIX geschreven heb, gaan we uiteraard ook verder met de andere operating systemen. We gaan verder met DOS-65, MS-DOS etc. als U maar kopij instuurt. Met name kopij voor andere systemen dan MS-DOS (Amiga, Atari, Octopus) kunnen we zeer goed gebruiken.

Tenslotte wens ik u, zoals altijd, veel plezier met uw computerhobby en hopelijk tot ziens in Almelo op 17 november.

Gert van Opbroek.

www.dottyflowers.com

Uitnodiging voor de clubbijeenkomst

Datum: 17 november 1990
Locatie: Wijkcentrum 't Veurbrook
Jan Tooropstraat 27
7606 Almelo
Telefoon: 05490 - 10353

Entree: gratis
Thema: Algemene ledenvergadering

Routebeschrijving

Auto

Vanuit het westen en het zuiden (A1/A35):

1. Aan het einde van de snelweg rechtsaf. Bij het eerstvolgende kruispunt met verkeerslichten linksaf, richting Wierden/ Zwolle. Bij de eerstvolgende verkeerslichten rechtdoor. Bij de volgende verkeerslichten (links BP tankstation en Opel garage Kamp) gaat u rechtsaf.

2. U rijdt nu op de Windmolenbroeksweg. Doorrijden tot over de brug, dan de eerste straat rechts. Dit is de W. van Konijnenburgstraat. Na plm. 50 meter rechtsaf. Dit is de Tooropstraat. Met de bocht mee naar links. Na plm. 50 meter aan de rechterkant: 't Veurbrook.

Vanuit het noorden (via de N 36):
1. Bij de stoplichten rechtsaf, richting streekziekenhuis. U bevindt zich nu op de rondweg om Almelo. Deze weg blijven volgen tot u het BP tankstation ziet bij dit kruispunt linksaf. Zie verder punt 2.

Openbaar vervoer

Vanaf NS-station Almelo met de stadsbus naar de wijk Molenbroek. Uitstappen bij de halte Windmolenbroeksweg. Schuin tegenover de bushalte staat een wegwijzer, daarop staat ook 't Veurbrook vermeld.

Programma

9:30	Zaal open met koffie
10:15	Opening
10:30	Algemene ledenvergadering, zie agenda
12:00	Forum en markt
12:30	Lunch, consumpties tegen betaling

Aansluitend het informele gedeelte met de mogelijkheid om andermans systemen te bewonderen en Public Domain software uit te wisselen. U en uw systeem zijn uiteraard van harte welkom.

17:00 Sluiting

Attentie

Het is ten strengste verboden illegale kopieën van software te verspreiden. Aan personen die deze regel overtreden zal de verdere toegang tot de bijeenkomst ontzegd worden. Breng verder alleen software mee die u legaal in uw bezit heeft. Het bestuur aanvaardt geen enkele aansprakelijkheid voor de gevolgen van het in bezit hebben van illegale software.



www.dottyflowers.com

Agenda algemene ledenvergadering

Datum: 17 november 1990
Plaats: zie uitnodiging clubbijeenkomst

1. Opening ledenvergadering.
2. Notulen ledenvergadering d.d. 20 januari 1990.
3. Conceptbegroting 1991.
4. Verkiezing kascontrole-commissie 1991.
5. Verkiezing bestuursleden; de volgende bestuursleden treden af en zijn herkiesbaar:
   Jacques Banser (penningmeester)
   Jan Derksen
   Gert van Opbroek
   Tegenkandidaten dienen uiterlijk voor aanvang van de vergadering, schriftelijk, bij de voorzitter aangemeld te worden. Deze voordracht dient vergezeld te gaan van de handtekening van tenminste twee leden.
6. De toekomst van de KGN. Bij dit agendapunt wil het bestuur van gedachten wisselen over de mogelijke activiteiten die vanuit de club opgestart zouden kunnen worden.
7. Rondvraag.
8. Sluiting.

Vragenrubriek

In uitgave 65 werden enkele vragen gesteld over een in uitgave 64 gepubliceerd programma. Het betrof de syntax van enkele BASIC-commando's, en wel de SCREEN statements in de regels 960 (drie parameters) en regel 2530 (die er maar één heeft). Verder was er een vraag over het LINE-statement in regel 2640. Wat doet LOCATE 7? En wat doet COLOR 7,1? De kleuren 4 en 5 stellen rood en groen voor, maar welke is nu wat?

In het vermelde programma is gebruik gemaakt van GW-BASIC. Het commando SCREEN p,q,r,s geeft voor p = 0 textmode,q = 0 zwart/wit. De beide andere parameters zijn in dit programma overbodig (hebben te maken met de paginakeuze). Line(X1,Y1-X2,Y2) trekt een lijn tussen de beide punten, -line(X3,Y3) trekt een lijn vanaf het vorige punt {hier(X2,Y2)} tot (X3,Y3). De cijfers achter de komma zijn de kleurkeuzes 4 = rood en 5 = groen. Om met complementaire kleuren te kunnen werken moet de achtergrond wit zijn, de groene kleur door rood bekeken verliest dan de kleur en is dan zwart evenzo rood door een groen glas bekeken. De code color 7,1 geeft een witte achtergrond (7) en een palet (kleurverzameling = 1), dat het beste past bij de kleuren van de bril. Het palet 1 is experimenteel vastgesteld, eventueel kan de kleur enigszins worden bijgeregeld door contrast en intensiteit te varieren. Locate a,b tenslotte geeft de plaats op het beeldscherm waar de tekst geplaatst moet worden (b ontbreekt geeft b = 0). Ik hoop hiermee de vragen voldoende beantwoord te hebben.

Ad Oerlemans

Pascal

Heeft de ontwerper al een versie gemaakt waar de foutjes uit gehaald zijn ? Dat betreft o.a.:

- dispose werkt niet.
- conditionele expressie's werken niet altijd goed.

Comprimeren

Voor de MS-DOS machines bestaan programma's die files comprimeren. Vooral handig bij het overseinen naar een BBS of een kennis, de bekende bla-bla.ZIP. Bestaat zoiets ook voor DOS65? Niet iedereen heeft naast de DOS65 machine een PC staan, of wel? Zo niet, dan moet ik alles overseinen in de gewone ASCII code. Anders hebben de DOS65 gebruikers er niets aan. Of wil iemand eens een compressie programma maken?

Editor TABS

Met !edtab kunnen de tab-stops naar keuze ingesteld worden en dus precies zoals nodig voor een bepaalde tabel. Als echter later die text verder bewerkt moet worden dan is de tab-instelling verloren en moet die opnieuw uitgezocht worden. Kan een tab-instelling op disk bewaard worden?

Print TABS

Als een text gemaakt is met de editor waarbij de tabs niet op de gewone plaats staan, kan dat dan goed op papier gezet worden? Dan moet print dus de tab-instelling weten zoals die er tijdens het werk met de editor uit gezien heeft.

www.dottyflowers.com

Het standaard Operating System "UNIX" (deel 1)

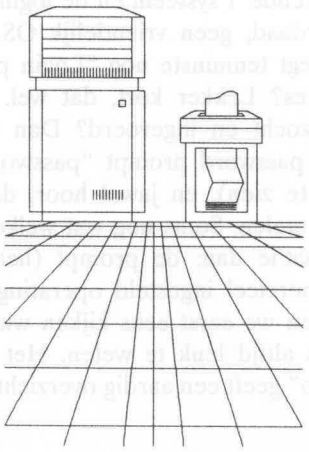
Het begin

UNIX is een oud operating system. Veel zaken in het operating system hebben een historische achtergrond die omwille van de compatibiliteit voortgezet wordt tot in de meest nieuwe UNIX versies. In mijn artikel over MINIX in de μP Kenner nummer 67 beschreef ik reeds in korte trekken de geschiedenis van UNIX. Het is niet mijn bedoeling dit zelfde verhaal nógmaals te houden, maar een beetje historie is nooit weg.

We gaan terug tot in de vijftiger jaren. De computers uit die tijd waren weliswaar programmeerbaar, maar ideaal werken was het niet toendertijd. De programmeur maakte zijn programma, zette dat om in ponskaarten en deponeerde die in een bak in de computer kamer. De computer operator stopte die kaarten dan in de computer, het ding werd aan de gang gezet... en genereerde in 90% van de gevallen een syntax error. Typefoutje, meneer!

In het begin van de zestiger jaren ontwierpen onderzoekers van het Dartmouth College en M.I.T. een timesharing system. Verschillende gebruikers konden nu gelijktijdig inloggen op dezelfde computer vanaf hun eigen terminals. Dit eerste timesharing systeem werd "CTSS" gedoopt; een acroniem van "Compatible Time Sharing System". Het ding was min of meer compatible met de systemen zoals die tot dan toe gebruikt werden bij het M.I.T. Het succes van CTSS was groter dan de ontwerpers ooit hadden durven hopen. Men besloot het systeem nog meer uit te buiten door het over te zetten op één der grootste computers beschikbaar: de GE 645. Het resultaat, MULTICS (MULtiplexed Information and Computing Service), is de grootvader van UNIX, VMS, PRIMOS en nog een hele hoop andere multitasking operating systems. Het MULTICS project was een product van de samenwerking tussen M.I.T., General Electric en AT&T Bell Labs. General Electric werd verkocht aan Honeywell, de firma die in samenwerking met het M.I.T. de uiteindelijke MULTICS versie maakte.

Eén van de ontwikkelaars van MULTICS bij Bell Labs, Ken Thompson, zat zich inmiddels te bezinnen op een nieuw project. En op een goede dag vond'ie een oude PDP-7 die niemand meer gebruikte. Hij besloot een beperkte MULTICS-versie over te zetten naar de PDP-7. Het zaakje werkte als een trein, maar ondersteunde slechts één gebruiker tegelijkertijd. Eén van de werknemers van Bell Labs, Brian Kernighan, doopte het systeem bij wijze van grap "UNICS" (UNiplexed Information and Computing System).



UNICS trok de aandacht van de managers van Bell Labs. Ze gaven Thompson een PDP-11 en de opdracht er "even" een echt Operating System van te maken. Dennis Ritchie kreeg na deze implementatie de opdracht samen met Thompson een hogere programmeertaal te ontwikkelen opdat het systeem portable werd naar andere computers. De taal die zij ontwikkelden heette in het begin "B", later "C". UNICS werd omgeschreven naar C en kreeg en passant ook nog even de nieuwe naam "UNIX".

Het systeem evolueerde uit naar wat tegenwoordig als de UNIX standaard wordt beschouwd. In deze reeks artikelen zal ik proberen een rode draad te leggen langs de valkuilen en over de smalle paden van UNIX. Wie onderweg valt, kan altijd nog een goed UNIX boek kopen...

De UNIX mentaliteit

In de beginjaren van UNIX waren er alleen langzame en lawaaierige teletype papierterminals. Het UNIX operating system is daar redelijk op afgesteld; korte, cryptische opdrachten en foutmeldingen karakteriseren de UNIX mentaliteit. Eén der leden van de KGN brengt dit aardig onder woorden in zijn lijfspreuk: "UNIX is not a user friendly operating system; it merely requires a friendly user". UNIX doet wat gevraagd wordt: klant is koning. Het OS vraagt niet of de user wil wat'ie zei; "niet zeuren"! Geen nieuws is goed nieuws. Een desastreuze remove opdracht ("rm *") vraagt helemaal niets, laat niets zien, maar intussen... Wie eenmaal gewend is aan het twee maal controleren van het ingetypte statement alvorens de opdracht af te sluiten met een krachtige neergaande beweging van de op de Enter-toets liggende vingers (ahum...) leert vanzelf UNIX te waarderen. En stuurt MS-Windows en GEM pardoes de bitjeshemel in.

www.dottyflowers.com

Een terminal sessie

Nemen we eens plaats achter een terminal en zetten 't ding aan. "Piep"! Hmm... een schermpje met wat tekst betreffende 't systeem en de login-prompt "login:". Inderdaad, geen vriendelijk OS, dat UNIX. PRIMOS zegt tenminste nog "Login please:". Nou ja, who cares? Lekker kort, dat wel. Gebruikersnaam opgezocht en ingevoerd? Dan verschijnt er meestal de password prompt "password:". Tikkerde-tik (nix te zien), en jawel hoor, daar begint 't zaakje te reutelen. Soms nog wat welkomst rommel en daar staat'ie dan: de prompt (het dollarteken "$"). Commercieel ingesteld operating system, dat UNIX! Laten we eerst eens kijken wie er allemaal online is; is altijd leuk te weten. Het UNIX-commando "who" geeft een aardig overzichtje van wie er ingelogd is:

```
$ who
USER     LINE     TIME        ACTIVE      PID
root     tty0     11:45:55    00:05:47    15
joost    tty2     11:50:47    00:00:55    36
```

Welwel, ik ben niet de enige op het systeem. Was ook niet te verwachten... meestal is de system operator ook wel ingelogd. De system account van de sysop heeft in UNIX bijna per definitie de naam "root". Let op de kleine letters; UNIX is case-sensitive! Eens zien. Ik ben zelf ingelogd op de terminal met de naam "tty2". De sysop is ingelogd op "tty0", de console. Sja, ik ben begonnen om 11:50:21.

Nou zou ik toch wel eens willen weten wat de sysop aan het doen is. Ook daarvoor heeft UNIX een aardig commando: ps ("process status"). Het intypen van ps geeft als resultaat:

```
$ ps
PID    TTY    TIME    CMD
36     t2     0:08    -
62     t2     0:00    ps
```

Hmmm... dat zijn alleen mijn processen. Mijn copie van de command interpreter (het streepje) en ps zelf. Geeft het nummer dat het process toegekend is, de terminal waar het process vanaf gestart is, de tijd die het zaakje geconsumeerd heeft en de naam. Eens zien of we er wat meer te zien kunnen krijgen:

```
$ ps -a
PID    TTY    TIME    CMD
15     co     0:07    -
36     t2     0:09    -
68     t2     0:01    ps -a
```

Hmmm... er is slechts één process van de root. Zijn command-interpretor (process nummer 15). Of hij is er niet (maar zijn terminal, de console, staat wel ingelogd!) of hij is bezig met het intaffen van een commando. Nog meer informatie krijgen we te zien als we de opdracht "ps -lax" zouden uitvoeren. UNIX biedt de mogelijkheid "vlaggetjes" te combineren door ze zonder seperators achter één minnetje te groeperen. Wel zo aardig. Met het genoemde commando krijgen we een lange lijst met uitgebreide omschrijving. De "x" geeft aan dat alle processen getoond moeten worden; de "l" geeft aan dat meer informatie van ieder proces getoond moet worden. Een aantal processen die vrijwel altijd zullen draaien in een UNIX systeem zijn o.a. "kernel", "swapper", "init", "update", "cron" en vaak ook nog wel ergens "getty". Deze processen komen verderop in de serie aan de orde. Eéntje wil ik hier nog wel even nader toelichten. De kernel. De kernel is het eigenlijke operating system. Het verzorgt de shell (de command interpretor) en de hardware. Als een process gegevens wil schrijven of lezen, dient het daartoe een verzoek in bij de kernel. De kernel zorgt dan voor de verdere afhandeling. Hetzelfde gaat op voor geheugen aanvragen en andere systeem specifieke zaken. Op de keper beschouwd is de kernel-task vaak een verzameling van taken. De memory manager, de file system server en andere processen zijn eigenlijk aparte processen. In veel UNIX systemen zijn ze echter allemaal opgenomen in de kernel.

Nu heeft iedere user meestal wel een eigen directory. Eens kijken wat er allemaal in mijn huidige directory (de current directory) staat. De UNIX opdracht daarvoor is "ls":

```
$ ls
Makefile
a.out
crc32.c
crc32.h
minix.his
unzip.c
unzip.1
```

Hmmm... alweer, nogal zuinig met informatie. Ik wil meer zien! Oke...

```
$ ls -l
total 95
-rw-rw-rw-  1 joost   189 Sep  3  1990 Makefile
-rwxr-x---  1 joost 43136 Sep  4  1990 a.out
-rw-rw-rw-  1 joost  8246 Sep  3  1990 crc32.c
-rw-rw-rw-  1 joost   150 Sep  4  1990 crc32.h
-rw-r--r--  1 joost 11132 Aug 12  1990 minix.his
-rw-rw-rw-  1 joost 22289 Sep  3  1990 unzip.c
-rw-rw-rw-  1 joost  2864 Sep  3  1990 unzip.1
```

Help! Wat betekent dat nu allemaal weer!? Laten we links in de kolom beginnen: de vlaggetjes. UNIX maakt gebruik van een redelijk slim protectie systeem. Iedere user heeft een eigen nummer ('n UID). Verder zijn alle users opgedeeld in groepen met ie-

www.dottyflowers.com

der hun eigen Group ID (GID). De vlaggetjes in de linker kolom zijn de attributen van de file. Het allereerste vlaggetje is leeg. Hier kunnen allerlei letters staan; een "d" geeft bijvoorbeeld aan dat het om een directory entry gaat. De overige negen vlaggetjes zijn opgedeeld in groepen van drie. De eerste groep geeft aan wat de eigenaar van de file allemaal met de file mag doen; de tweede groep wat users uit dezelfde groep als de eigenaar allemaal mogen en de derde geeft aan wat de rest van de wereld mag. Mogelijke vlaggetjes in deze groepen zijn steeds "rwx"; "Read", "Write" en "eXecutable". In het overzichtje mag iedereen bijvoorbeeld de file "unzip.doc" lezen en schrijven. De file "a.out" mag gelezen, geschreven en uitgevoerd worden door de eigenaar. Anderen uit de zelfde groep mogen alleen lezen en uitvoeren. Alle andere users (met uitzondering van de root!) mogen helemaal niets met de file. Let op: schrijven betekent **niet** weggooien! Wil iemand iets weggooien, dan moet hij write access hebben op de directory waar de betreffende file in staat!

Nog meer informatie is te krijgen als bij ls ook de g- en a vlaggetjes geset worden (zie figuur 1). Het "g" vlaggetje geeft aan dat ook de group getoond moet worden ("user"). Het vlaggetje "a" geeft aan dat we ook files die beginnen met een punt willen zien. Standaard worden files die beginnen met een punt niet getoond door UNIX. Daardoor zien we meestal alleen de relevante informatie. De eigen directory (".") en parent directory ("..") zijn niet van belang om te zien. De file ".profile" is een file die automatisch uitgevoerd wordt bij het inloggen van de user. Vergelijkbaar met de "Autoexec.Bat" file onder MS-DOS.

Als je dit in de praktijk uitvoert, dan gaat er natuurlijk altijd ergens wel eens wat mis. Of je wilt wat afbreken, of... UNIX heeft daarvoor verschillende tekens. Die tekens zijn op te vragen met het stty commando stty:

```
$ stty
-tabs  -nl echo
kill = @
erase = ^H
int = DEL
quit = ^\
speed = 9600
bits = 8
```

De termen "kill", "erase", "int" en "quit" geven aan welke characters gebruikt worden. Ze zijn ook anders in te stellen; DEL is niet echt handig als int signal. He, een nieuwe term! UNIX maakt gebruik van een signalling systeem waarin tasks verschillende signalen naar andere tasks kunnen sturen. De belangrijkste signals zijn:

1	Hangup
2	Interrupt
3 *	Quit
4 *	Illegal instruction
5 *	Trace trap
6 *	IOT instruction
7 *	EMT instruction
8 *	Floating point exception
9	Kill (cannot be caught or ignored)
10 *	Bus error
11 *	Segmentation violation
12 *	Bad argument to system call
13	Write on a pipe with no one to read on it
14	Alarm clock
15	Software termination (from kill(1))

De signals uit deze lijst die gemerkt zijn met een asterisk produceren een core dump. Dat is een file waarin de huidige systeem situatie beschreven wordt die bruikbaar is voor debugging doeleinden.

```
$ ls -lga
total 101
drwxr-x---   2 joost     user          224   Sep  17  11:51  .
drwxr-xr-x   7 bin       root          128   Sep  13  14:53  ..
-rw-r--r--   1 joost     user           15   Sep  14  21:04  .exrc
-rw-rw-rw-   1 joost     user           25   Sep  17  11:50  .history
-rw-r--r--   1 joost     user           34   Sep  14  21:05  .kermrc
-rwxr--r--   1 joost     user          280   Sep  12  12:17  .profile
-rw-rw-rw-   1 joost     user          189   Sep   3  10:02  Makefile
-rwxr-x---   1 joost     user        43136   Sep   4  09:03  a.out
-rw-rw-rw-   1 joost     user         8246   Sep   3  10:02  crc32.c
-rw-rw-rw-   1 joost     user          150   Sep   4  09:01  crc32.h
-rw-r--r--   1 joost     user        11132   Aug  12  23:07  minix.his
-rw-rw-rw-   1 joost     user        22289   Sep   3  10:00  unzip.c
-rw-rw-rw-   1 joost     user         2864   Sep   3  10:01  unzip.1
```

Fig. 1: uitvoer van het commando "ls -lga"

www.dottyflowers.com

Zoals uit de terminal settings (output van "stty") bleek, kunnen we een interrupt signal sturen naar het lopende proces door middel van de DEL-toets. Een signal kan ook aan een process gestuurd worden door middel van het kill-commando. Voor het afbreken van de shell (uitloggen!) kan bijvoorbeeld een kill signaal aan de shell gestuurd worden. In de terminal sessie zoals hierboven omschreven had mijn shell process nummer 36. Ik kan uitloggen door middel van de opdracht "kill -9 36". Hiermee stuur ik dus signaal nummer 9 naar process 36.

En fraaiere manier om uit te loggen is door de invoer af te sluiten. De shell leest en verwerkt commando's vanaf standard input tot hij een End Of File character tegenkomt. Het EOF character van UNIX is ^D. Als de user dus ^D intoetst op een lege regel, dan wordt de invoer van de terminal beëindigd en stopt de shell.

Volgende keer

In de volgende aflevering maken we een tour door het file system en bekijken een aantal eenvoudige shell-scripts (batch-jobs).

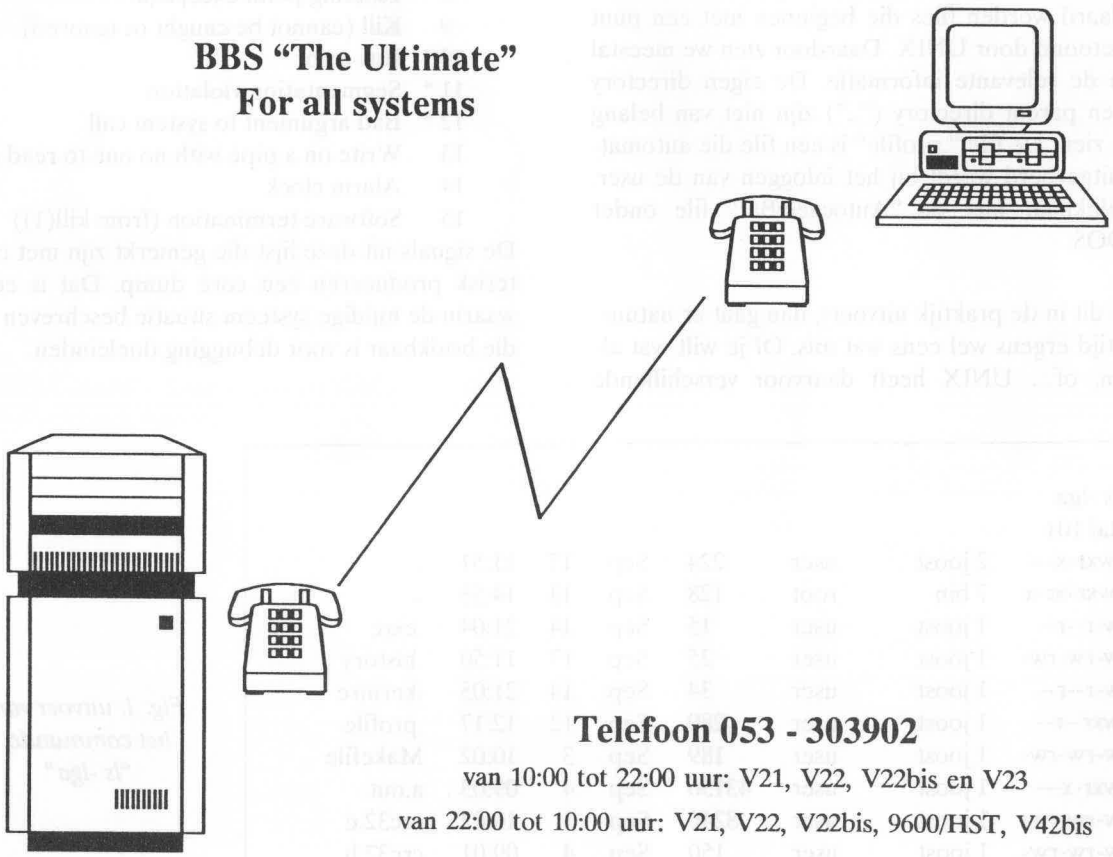
Literatuur:

1: Dr. Ir. K.L. Boon & Ir. A.R.Th. Pelsmaeker, UNIX compact Gids, Addison-Wesley; OMIKRON (ISBN 90-6789-020-0)

2: G.J.M. Austen, UNIX: Het standaard operating system, Academic Service (ISBN 90-6233-217-X)

Joost Voorhaar

BBS "The Ultimate"
For all systems



Telefoon 053 - 303902

van 10:00 tot 22:00 uur: V21, V22, V22bis en V23

van 22:00 tot 10:00 uur: V21, V22, V22bis, 9600/HST, V42bis

www.dottyflowers.com

De computerdokter

Dit jaar begon de herfst meteen met volle overtuiging aan zijn werk. De eerste de beste dag al haalde ik tot drie keer toe een nat pak. Om mij heen manifesteerden zich de eerste verkoudheden; truien, shawls en warme sokken worden door alle moeders van Nederland uit de kast gehaald en onderzocht op bruikbaarheid.

Niet alleen de mens heeft last van de koude, computertjes hebben dat blijkbaar ook. Deze week heb ik voor het eerst het verwoestende werk van een virus van dichtbij meegemaakt. De eigenaar van het besmette systeem belde mij in zak en as op: "Hij doet 't niet meer!". Een bekende winkel in computerspelletjes (die trouwens zijn naam veranderd heeft om van dat spelletjes-imago af te komen) verdiende die dag weer het één en ander aan backup-diskettes. Want natuurlijk had de treurende, eens zo trotse computereigenaar nog nooit backups gemaakt...

De telefoon stond trouwens de hele week al roodgloeiend. Een vriendin belde in paniek op dat haar computersysteem het nu helemaal niet meer deed! Zij heeft al een paar dagen problemen met de diskdrives, en nu dit weer. Dat was om een uur of twee in de middag, dus ik wilde eerste even ontbijten. In de tussentijd keek zij nog even verder en belde prompt vijf minuten later weer: "Hij doet 't weer... er zit een schakelaar op de achterkant die ik per ongeluk uit had gezet...". Dat heb je met die vreemde Europese ontwerpen waar een contact-sleutel aan de voorkant van zo'n apparaat blijkbaar niet genoeg is.

Een paar dagen later belde een vriend die op het moment aan 't afstuderen is bij een grote machinefabriek. Daar hadden ze een computer met een harddisk erin die 't niet meer deed. Wat bleek? De plaatselijke "expert" zou wel even opnieuw formatteren. Low-level, wel te verstaan. "Neen, het was een gewone disk. Ja, 20 megabyte, 4 koppen. Openschroeven? Waarvoor dan? Nou ja, da's goed, ik kijk wel even. Heb je hier wat aan: '782c, 2h'? Ja, dat staat op de disk... Nee, natuurlijk niet, gewoon, 16 sectors per track. Huh? RLL? Deze disk? Nee, natuurlijk niet! Controller? Uhhh.... heb je hier iets aan: WD27X. Toch RLL? Weet je dat helemaal zeker? Nou ja, in dat geval... ja? 26 of 27 cylinders? 2 koppen? Zekers weten? Sja, zal dan wel..."

In zo'n weekje blijkt pas echt hoe moeilijk het is om "even" per telefoon hulp te bieden aan de vertwijfelde computereigenaar. Wel eens geprobeerd iemand een computer configuratie aan te laten passen per telefoon? Ik wel:

"Ja, en dan typ je nu in: 'type autoexec.bat'"
"Hoe zeg je? Kun je dat even spellen?"
"Nou, typ nou in: tee-iegrek-pee-ee. En dan een spatie, gevolgd door aa-uu-tee-oo-ee-iks-ee-see. En dan een punt en bee-aa-tee."
"Ja, heb ik gedaan. En wat nu?"
"Wat staat er nou op het beeldscherm?"
"Nou, gewoon, wat ik net in heb getikt"
"Huh? Heb je wel op return gedrukt?"
"Uhhh... nee, moest dat dan?"
"Dachut wel ja..."
"Hmm... return? Kan ik hier niet vinden hoor!"
"Ja, of een toets met 'enter' erop. Soms staat er ook alleen maar een pijltje op met een hoek d'r in"
"Oooohhh, die! Zeg dat dan!"
"En, wat zie je nu?"
"Uhh... bet kommant or fajlneem".
"Aaarrrrgh !@#$%^& !!!"*
Tijdens zo'n gesprek rijst bij mij altijd onmiddellijk de vraag of ze nou echt nog nooit een handleiding gelezen hebben. Ik moet me dan altijd gigantisch inhouden: "Een handleiding, zo'n boekje met letters... een stapel faxen? Zoiets ja, maar dan anders. Die zat er wél bij? Hoezo, onleesbaar? En dat had je in de winkel al gezien? Nou ja, mijn Japans is ook niet zo goed, maar mijn leverancier nam tenminste nog de moeite de vertaalde handleiding even te kopiëren toen ik 'm daar om vroeg...".

In Apeldoorn is er iemand ooit begonnen met een soort computer-advies bureau c.q. algemene computer werkplaats. Hij noemde zichzelf "computerdokter" (en haalde er prompt de plaatselijke krant mee). Ik ben er één keer geweest... Nou, de gemiddelde dokter heeft volgens mij een beter instrumentarium om computers te genezen! In ieder geval heeft mijn huisarts nog een boekenplank met een grote medische encyclopedie erop en het telefoonnummer van de apotheek...

En moet ik hier nou ooit mijn vak van maken? Een goede vriend van mij vergeleek computers ooit met auto's. Hij is nu part-time verkoper in die computerzaak waar ze nog steeds struikelen over de spelletjeskopers. "Er is één verschil met het verkopen van een auto", zei hij: "Als iemand een auto koopt heeft'ie in ieder geval meestal nog wel een rijbewijs...". Als u mij dus ooit nog eens belt met een vraagje over computers... zorgt u er dan in ieder geval even voor uw ziekenfonds kaart bij de hand te hebben?

Uw huisarts, C. Ompudok.
(Behandeling alleen op afspraak)

www.dottyflowers.com

Een printerbuffer en een toetsenbordbuffer voor DOS65

Al tijden lang maak ik gebruik van mijn goeie ouwe printer, een Diablo hitype II met eigengemaakte interface die ik vlak na het uitkomen van de Junior computer gekocht heb. Dat ding leek onverwoestbaar maar een paar maanden terug was het opeens zover, er kwamen nog wat tekens uit maar het leek in het geheel niet meer op de tekst die er in ging. Van een kompleet vel A4 bleef ongeveer een halve regel haakjes sluiten over en verder moest ik het maar zelf invullen. Als je je printer tamelijk vaak gebruikt, bijvoorbeeld ook als typemachine, is dat een regelrechte ramp. Ik organiseerde een andere printer (te leen voor een paar dagen) en ging vrolijk weer aan het werk. Hier kwam echter helemaal niets op papier, wat ik ook probeerde. Na wat zoekwerk bleek dat de Centronics printer de parallel signalen zo sterk belastte dat de 6522 in de DOS65 (=octopus) gewoon geen fatsoenlijk signaal meer afgaf. Een oplossing bleek niet zo moeilijk en is eventueel best in te bouwen in de computer: twee IC's 74LS367 tussen de uitgangen van de 6522 (ook de strobe) en de printer ingangen. Met als resultaat dat elke parallel printer voortaan te gebruiken is. In plaats van de 74LS367 kan natuurlijk ook een andere driver gebruikt worden, zoals de 74LS244.

Een tweede probleem doet zich af en toe voor als ik enthousiast verder tik terwijl mijn computer aan het assembleren is of iets anders doet dat veel interrupt afhandeling vergt. Voor elke toets die ik indruk moet de 6502 een interrupt afhandelen. Als de disk nogal intensief gebruikt wordt gaan deze interrupts voor (zo hoort het ook) maar tijdens het afhandelen van een diskinterrupt kan ik best twee toetsen ingetikt hebben (daar hoef je ook weer niet zo snel voor te tikken). De eerste toets wordt vergeten want voordat de keyboard interrupt wordt afgehandeld staat de tweede toets alweer klaar. Nu zijn er verschillende oplossingen mogelijk. Ten eerste zou ik niet meer zo ongeduldig kunnen zijn en rustig kunnen wachten tot de computer er weer klaar voor is, of in ieder geval zo langzaam kunnen typen dat alle interrupts er door komen. Dit begon me mateloos te irriteren. Een tweede mogelijkheid is om er wat aan te doen. Dat leek me wel wat. De mogelijkheid om dan maar op een andere computer verder te gaan heb ik niet serieus overwogen omdat ik toch nog steeds zeer tevreden ben met DOS65. Nu zijn er van die mooie kleine geheugen IC's die FIFO's genoemd worden (AMD 2841 of 3341). Deze 2841 bestaat uit 64 x 4 bits, voor het verwerken van bytes heb je er dus twee nodig. Het leuke van deze mini geheugenchips is dat je vier bits klaarzet en als je dan een strobepuls geeft dan begint er een hele operatie in de fifo waardoor die vier bits door het hele geheugen heenvallen en op de laatste positie terecht komen, bij de uitgang. Verder geeft de uitgang een signaal af om te kennen te geven dat er informatie klaar staat. Als je nu weer data aanvoert en een strobe puls geeft, zal dit tweede byte doorstomen naar de voorlaatste plaats en zo voort, tot alle 64 plaatsen gevuld zijn. Als je een acknowledge geeft aan de uitgang om aan te geven dat je een 'nibble' hebt opgepakt, dan schuift de hele zaak een plaats verder en zal er weer een strobe signaal komen, mits er tenminste nog een nibble aanwezig is. Dit IC leek me ideaal om mijn probleem op te lossen, temeer omdat er helemaal geen clock of zo nodig is, het IC regelt alles zelf. Het enige probleem was nog dat DOS65 geen acknowledge stuurt naar het toetsenbord, maar dit bleek helemaal geen probleem te zijn. Door het veranderen van twee bytes in de eprom (I/O65) wordt de 6522 gevraagd om een automatische handshake te doen. Als je die handshake niet gebruikt heeft I/O65 er ook geen last van dus je kunt altijd terug zonder wisselen van eproms. De volgende lokaties in eprom zijn bij mij aangepast (I/O65 2.02 of 2.01):
F0DFwas 0E, wordt 0A en F0E1was FE, wordt FA.

De schakeling is zo ontworpen dat je het toetsenbord kunt lostrekken en via de schakeling weer op dezelfde manier kunt aansluiten. Je zou het geheel in een klein kastje tussen toetsenbord en computer kunnen hangen hoewel het mooier is om het in de computer in te bouwen. Mochten er problemen zijn met de verkrijgbaarheid van de AMD 2841 laat het dan even weten. Succes,

Ernst Elderenbosch.

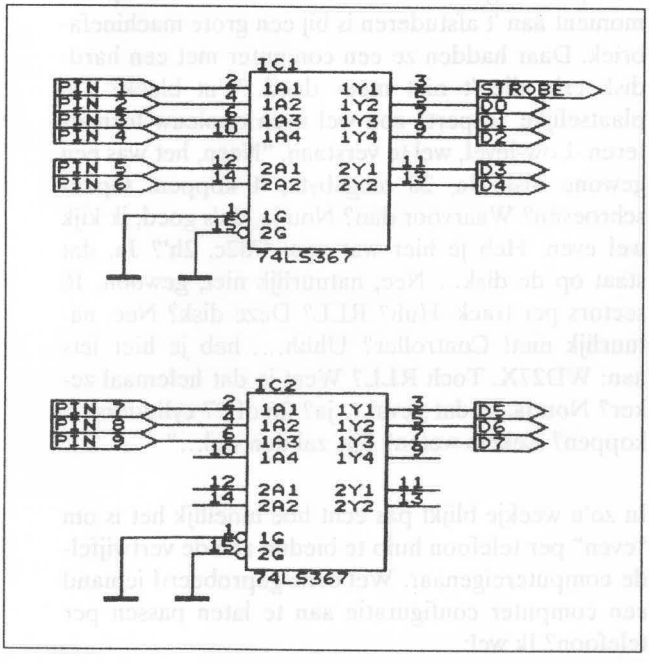


Fig. 1: printerbuffer

www.dottyflowers.com

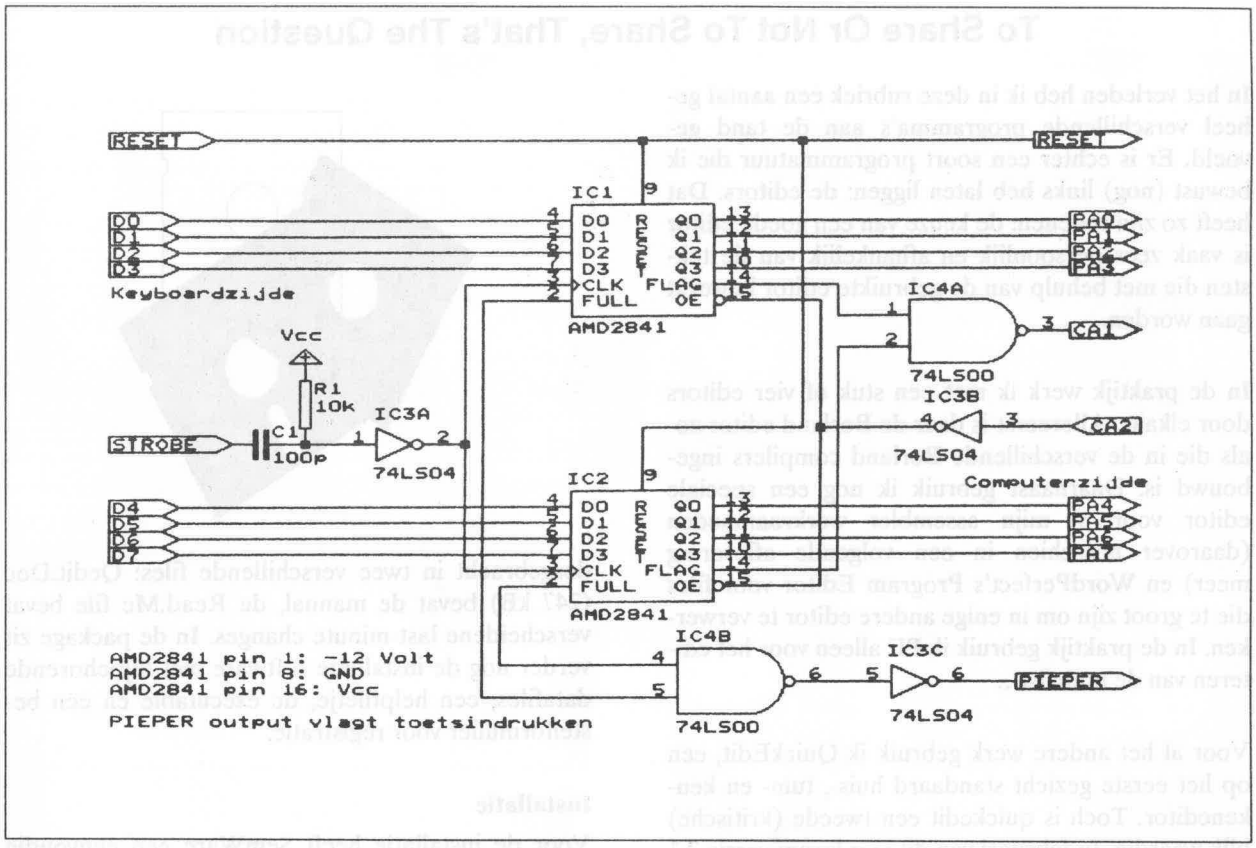
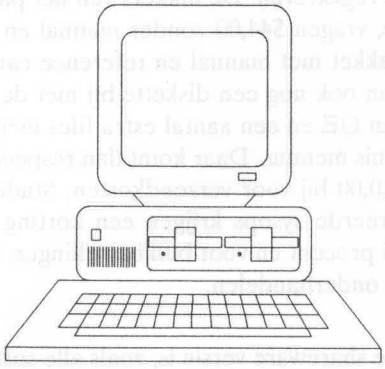


Fig. 2: keyboardbuffer voor DOS65

(Advertentie)

Speciale aanbieding: De KGN club-PC

Een **PC/XT compatible computer** standaard met:
4.77 of 10 MHz clocksnelheid, hard- en software omschakelbaar
640 kbyte RAM
MDA/Hercules compatibele displaykaart met printerpoort
Multi I/O kaart met: Seriële poort
Parallel printer poort
Game poort
Batterij-klok
Floppy disk controller
Plaats voor 2e seriële poort
KGN XT-BIOS met 720k floppy support (natuurlijk!)
1 360k floppy disk drive
8 I/O slots
Klapkast in klassiek PC/XT design, met plaats voor nog 3 drives
150 Watt voeding met stille ventilator
Enhanced toetsenbord, 101 toetsen, XT/AT omschakelbaar



PRIJS: *f* 600,00 per stuk, inclusief B.T.W. (alleen voor leden)

Garantie: 3 maanden op arbeid en onderdelen, geldt niet voor vervoerskosten
Bestellen: maak *f* 600,00 over op Postbank rekening 3757649 o.v.v. KGN-XT en telefoonnummer voor afspraak aflevering.

www.dottyflowers.com

To Share Or Not To Share, That's The Question

In het verleden heb ik in deze rubriek een aantal geheel verschillende programma's aan de tand gevoeld. Er is echter een soort programmatuur die ik bewust (nog) links heb laten liggen: de editors. Dat heeft zo zijn redenen: de keuze van een goede editor is vaak zeer persoonlijk en afhankelijk van de teksten die met behulp van de gebruikte editor bewerkt gaan worden.

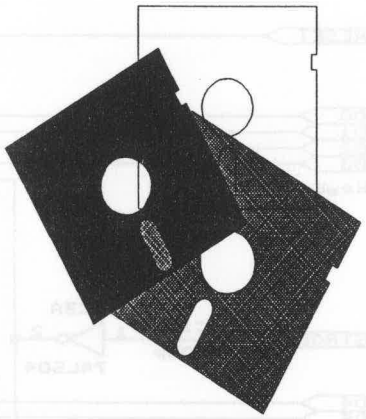
In de praktijk werk ik met een stuk of vier editors door elkaar. Allereerst is daar de Borland editor zoals die in de verschillende Borland compilers ingebouwd is. Daarnaast gebruik ik nog een speciale editor voor al mijn assembler werkzaamheden (daarover misschien in een volgende aflevering meer) en WordPerfect's Program Editor voor files die te groot zijn om in enige andere editor te verwerken. In de praktijk gebruik ik PE alleen voor het editeren van de nodelist...

Voor al het andere werk gebruik ik QuickEdit, een op het eerste gezicht standaard huis-, tuin- en keukeneditor. Toch is quickedit een tweede (kritische) blik waardig. In februari van dit jaar kwam versie 2.1 uit, hetgeen me een aardige aanleiding leek om het ding op de testbank aan een nader onderzoek te onderwerpen. Daar ik een gewone standaard PC/XT heb, kan ik alleen over de MS-DOS versie spreken. Er is namelijk ook een versie voor OS/2. Een aardige bijkomstigheid is dat er ook een TSR-versie van schijnt te zijn.

De software

QuickEdit wordt geleverd als shareware produkt. Je mag de editor dertig dagen uitproberen. Bevalt het ding, dan wordt vriendelijk doch dringend verzocht te registreren. De makers van het pakketje, SemWare, vragen $44,00 zonder manual en $54,95 voor het pakket met manual en reference card. Daar krijg je dan ook nog een diskette bij met de nieuwste versie van QE en een aantal extra files met o.a. macro's en muis menuus. Daar komt dan respectievelijk $3,00 of $10,00 bij voor verzendkosten. Studenten en geregistreerde sysops krijgen een korting van maar liefst 35 procent en voor bulkbestellingen is met SemWare te onderhandelen.

De shareware versie is, zoals alle software die ik hier bespreek, ook verkrijgbaar op het bulletin board van de vereniging. QuickEdit is in één file ondergebracht, QEdit21.Zip, 127 kilobytes. Op 2400 baud is dat ruim 9 minuten downloaden, hetgeen dus hard meevalt. In het pakket zit voldoende documentatie om een lege avond zonder moeder de vrouw en T.V. door te kunnen brengen. Deze documentatie is on-



dergebracht in twee verschillende files: Qedit.Doc (247 kB) bevat de manual, de Read.Me file bevat verscheidene last-minute changes. In de package zit verder nog de installatie software met bijbehorende datafiles, een helpfiletje, de executable en een bestelformulier voor registratie.

Installatie

Voor de installatie heeft SemWare een eenvoudig stukje programmatuur geschreven. Een eenvoudig menuutje met een beperkt aantal mogelijkheden opent de poort tot een gigantische hoeveelheid voorkeurinstellingen. Het loont de moeite ze allemaal even door te lopen en te bult vragen zo goed mogelijk te beantwoorden. Gelukkig heeft iedere vraag een default antwoord dat meestal correct is. De auteurs hebben de editor de ietwat ongelukkige naam "Q" gegeven; ik heb 'm dan ook "Ed" gedoopt. Een detail, dat naar mijn idee heel aardig weergeeft dat men bij SemWare nadenkt over hun produkt alvorens het uit te leveren, is de vraag in het installatie programma: "Enter program name to config, <CR> for Q.Exe:". Dit soort vragen worden ook gesteld voor de namen van de configuratie files; beetje overbodig naar mijn idee, maar toch...

Tijdens het werken met QuickEdit heeft men alleen maar de executable nodig. De volledige configuratie wordt, inclusief helpfile, in de executable ondergebracht. De diskruimte die hierdoor nodig is beslaat net 47 kByte.

In de standaard configuratie is de editor WordPerfect compatible. Wil men een andere editor nabootsen, dan kost dat nog redelijk wat werk. De geregistreerde versie komt met een aantal andere configuraties en op een aantal bulletin boards heb ik ook al andere complete configuratie files zien rondzwerven. Keuze te over dus. Wat heel erg aardig is van QuickEdit, zijn de popup/dropdown menu's. Zodra men <Esc> taft, verschijnt er een menu-

www.dottyflowers.com

balkje in de top van het scherm. Daarmee kan gekozen worden uit "File", "Window", "Block", "Search", "Print", "Macro", "Editing", "Other" en "Quit". Onder al deze rubriek-keuzes hangt weer een submenu met de verschillende individuele functies. De meest gebruikte functies zijn ook bereikbaar via de gebruikelijke short-cuts. Vaak spring je met zo'n "short-cut" onmiddelijk naar het menu met het menubalkje op de juiste plaats. Soms geeft QuickEdit ook alleen maar een message window zoals met ALT-X: "Save changes?" met de opties "Yes" en "No". Een keuzebalkje dat standaard op "Yes" staat vervangt hier het aloude "File not saved. Save changes [Y/n] ? ".

De editor is eng snel. Het verplaatsen van de top van een document naar het einde neemt in een hoop editors nog behoorlijk wat tijd in beslag. Dat valt met QuickEdit alleszins mee; je staat binnen mum van tijd aan het andere einde van je document. Het laden van een documentje neemt wel wat meer tijd in beslag. Een seconde of twee á drie per 1024 regeltjes. Nou ja, wat heet veel. Probeer het zelfde maar eens met de Program Editor van WordPerfect...
De maximale grootte van te editeren files wordt slechts beperkt door de hoeveelheid beschikbare geheugenruimte onder DOS. De editor heeft ongeveer 130 kB RAM nodig om te kunnen werken. De overblijvende geheugenruimte kan ten volle benut worden voor de te editeren file. Gaat men over de maximale document grootte heen, dan protesteert 't ding, maar gaat wel in edit mode. Is een document niet in zijn geheel geladen, dan kan het ook niet zondermeer weggeschreven worden: "Cannot save partial files", protesteert QuickEdit. De hele file als blok markeren en dan met "block save" wegschrijven gaat wel. Maar ja, uiteraard niet de hele file...

De één zoekt heel andere dingen in een editor dan een ander.

Documentatie

Over de grootte van de documentatie kan ik kort zijn: 247 kB is groot. De uitgebreide documentatie is voorzien van een goede index. Ik heb nooit de moeite genomen de volledige documentatie door te nemen; wie gewend is aan WordStar compatible editors kan vrijwel direct aan de slag met QuickEdit. Alle menu's staan in de documentatie omschreven zodat niets onduidelijk hoeft te blijven. De online-help is een beetje magertjes; één schermpje met de belangrijkste toetscombinaties is alles.

Features

Wil men meerdere files editeren die met een wildcard specificatie omschreven kunnen worden, dan blijkt onmiddelijk een ander voordeel van QuickEdit. De editor accepteert wildcard specificaties op de commandline, hetgeen bijzonder handig kan zijn. Alle gespecificeerde files worden in een soort ring opgenomen waar men naar believen door heen kan lopen.

Sommigen vinden het belangrijk dat een editor meerdere windows tegelijkertijd op het scherm kan hebben. QuickEdit kan maar liefst 8 windows tegelijkertijd open hebben.

De editor beschikt tevens over een bijzonder uitgebreid macro systeem. In de macro's kan gebruik gemaakt worden van externe programmatuur, zodat het heel goed mogelijk is een soort "integrated environment" à la Borland te maken met behulp van een macrootje.

Naast deze eigenschappen heeft QuickEdit nog een heel scala aan extraatjes die je in de praktijk bijzonder leert waarderen (zeker als je er tervergeefs naar zoekt in andermans favoriete editor). Zo kan QuickEdit kolommen markeren, lijntjes trekken, zoeken naar "matching bracket pairs" en heeft mogelijkheden voor het automatisch instellen van TABs voor verschillende file extensies. Filenamen worden automatisch aangevuld met van te voren gedefinieerde extensies als ze niet opgegeven worden. Dat wil zeggen: bij het zoeken naar files. Een nieuwe file krijgt geen standaard extensie, hetgeen nog wel eens voor verwarring kan zorgen.

Conclusie

Het kiezen van een goede editors is persoonlijke zaak. De één zoekt heel andere dingen in een editor dan een ander. Een editor moet daarom bovenal flexibel zijn. QuickEdit voldoet uitermate goed aan deze eis. Voor werk in een programmeeromgeving heeft QuickEdit een aantal bijzondere eigenschappen die, mits men ze goed gebruikt, het ding bijzonder waardevol maken. De snelheid en comfort zijn beide van een dusdanig hoog gehalte dat commerciële editors wel héél veel meer te bieden moeten hebben om ze interessant te maken. De prijs is wat aan de hoge kant; ruim honderd gulden is niet niks. Aan de andere kant krijg je voor dat geld wel een goede manual, muisondersteuning, extra keyboard definities en technical support. Een commerciële editor met een dergelijke prijs/prestatie verhouding zal moeilijk te vinden zijn.

Joost Voorhaar

www.dottyflowers.com

Besproken produkt	: QuickEdit 2.1 (advanced version)
Catogerie	: Editors
Registratiekosten	: Maximaal US $64.96 (zie tekst)
Auteur/Leverancier	: Semware, Georgia, USA.
Verkrijgbaarheid	: The Ultimate, MS-DOS Utilities area De file heet "QEDIT21.ZIP" en is ruim 124 kByte groot

Minimale systeemeisen

Standaard PC (Of PS/2, zie tekst)
128 kByte vrije RAM. QEdit gebruikt meer indien beschikbaar
PC/MS-DOS 2.00 of hoger
1 360 kByte diskdrive
Beeldscherm: monochrome of kleur, minimaal 80 kolommen. Qedit ondersteunt EGA/43 en VGA/50 modes. Maximaal 200 kolommen/100 regels per scherm (maar ja, wie heeft een scherm dat meer aankan?)

Algemene beoordeling

Documentatie	: Uitvoerig, Engelstalig
Online help	: Beperkt. Eén schermpke
Gebruikersinterface	: Keyboard, gedeeltelijk window gestuurd
Muisondersteuning	: Alleen in de geregistreerde versie

Positief

Compact
Snel
Wildcard ondersteuning
Meerdere windows
Speciale features voor programmeeromgeving
Externe programma's uit macro's aan te roepen

Negatief

Geen onbeperkte document grootte
Ietwat aan de dure kant

Eindbeoordeling

Stabiliteit	: 9
Bruikbaarheid	: 9
Totaalresultaat	: 9

Ik heb interesse in de KGN en wil

☐ Lid worden van de KGN ☐ Meer informatie over de KGN

Naam : _____

Adres : _____

Postcode en woonplaats : _____

Datum : _____ Handtekening : _____

Dit strookje kunt u ingevuld opsturen aan het secretariaat van: KIM Gebruikersclub Nederland
Davidvosstraat 29
1063 HV Amsterdam

www.dottyflowers.com

Het programmeren van de 8088 in de IBM (Deel 2)

In de vorige aflevering hebben we in theorie de registers en de diverse adresseermethoden van de 8088 leren kennen, zonder al te veel details. We zullen nu de layout van een typisch 8088 programma onder MS-DOS leren kennen in zijn meest algemene vorm. Figuur 1 laat een envoudig programma onder MS-DOS zien dat niets anders moet doen dan alleen de tekst "Dit is ons eerste .EXE programma" afdrukken. De kreet ".EXE" slaat daarbij op de structuur van het programma zoals die tegenwoordig gebruikelijk is. De excuteerbare files zijn in de inhoudsopgave van een MS-DOS schijf herkenbaar aan de toevoeging EXE. De oudere structuur die compatible is met CP/M bestaat ook nog en het gebruik er van kan bepaalde voordelen hebben. Die komt in een latere aflevering aan bod.

De programma kop

Op de eerste regels kunnen we het programma een naam geven, aanwijzingen voor de assembler opnemen voor het genereren van een listing zoals paginabreedte, aantal regels en een titel boven elke pagina. Vervolgens kunnen we enige regels commentaar opnemen teneinde het doel van het programma de naam van de auteur ezv. vast te leggen. De kop is een optie en in geen enkel opzicht verplicht. Hij kan echter een belangrijke documenterende functie vervullen. De naam, in dit geval "FIGUUR1", dient een uniek symbool te zijn. Met PAGE geven we hier aan dat we, als we MASM opdracht geven een listing af te drukken, graag 53 regels per pagina en maximaal 128 tekens per regel willen. Met TITLE specificeren we een titelstring welke in een listing boven elke pagina zal worden afgedrukt.

Definitie van macro's en constanten

MASM heeft een zeer flexibele macrofaciliteit. Over gebruik en misbruik van macro's is een boek te schrijven. Het valt buiten het bestek van deze serie om uitvoerig op deze faciliteit in te gaan. Diverse typen constanten kunnen in MASM worden gedeclareerd met EQU. Voorbeelden zien we in het programma. Symbolen als CR en LF lezen prettiger dan 13 en 10. Daarom worden ze hier gedeclareerd als constanten. De constante DISPHAND is een handle van DOS. Het begrip "handle" komt later in deze aflevering aan de orde

Het codesegment

We kiezen voor het codesegment de relevante naam CODE en geven met het SEGMENT directive aan dat hier een segment met die naam begint. Verderop in het programma komen we de naam CODE weer tegen maar nu wordt met ENDS aangegeven dat dit het einde is van het codesegment.

Het datasegment

Ook hiervoor kiezen we een relevante naam: DATA. Dit segment bevat de enige data die dit voorbeeld gebruikt. Dat is de constante string MESSG die we gaan afdrukken. Een string is een aantal opeenvolgende bytes welke op een zeker adres in het geheugen van de computer staan. Een string dient dus informatie omtrent zijn lengte te hebben. De hier gekozen structuur is die welke ook in Pascal wordt toegepast: De lengte staat in het eerste byte en wordt altijd gedefinieerd als een expressie, zodat het wijzigen van de tekst mogelijk is zonder het lengtebyte aan te moeten passen. De assembler vult zo de juiste waarde voor de lengte in.

Het stacksegment

De programmeur moet onder MS-DOS een stacksegment definieren. In het voorbeeld is een riante stack van 256 woorden oftewel 512 bytes gereserveerd. Bij de declaratie van een segment kan men daaraan diverse attributen toekennen. Het stacksegment, in ons geval PGMSTACK, moet altijd het attribute STACK hebben. Een programma kan meerdere stacksegmenten hebben, doch slechts een daarvan dient van dit attribuut te zijn voorzien en wel die stack die bij het opstarten van het programma effectief zal zijn. Het mag ex-6502 programmeurs opvallen dat de 8088 riante stack mogelijkheden heeft vergeleken bij de 256 bytes waar de 6502 het mee moet doen.

De DUP clausule

De betekenis van het directive DB voor "DataByte" is herkenbaar uit andere assemblers voor diverse processoren die ook wel .BYT of iets dergelijks gebruiken. MASM gebruikt dit directive ook om een aantal bytes te reserveren. Met de DUP clausule wordt aangegeven dat de waarde achter DB niet de initiele inhoud voor een byte bevat, maar een aantal te reserveren bytes aangeeft. Achter de DUP clausule wordt alsnog de initiële inhoud voor alle bytes opgegeven. Indien die inhoud ongedefinieerd is gebruiken we een vraagteken.

Het ASSUME directive

Het eerste dat we tegenkomen in het codesegment is het ASSUME directive. Met CS: bedoelen we daarin niet het register CS maar de afkorting voor CodeSegment. We dragen MASM hierme op om tijdens het assembleren er vanuit te gaan dat CODE het codesegment is, DATA het datasegment en PGMSTACK de stack. ASSUME genereert dus geen enkele code en initialiseert ook geen segmentregisters. Indien in het programma aan welk item dan ook wordt gerefereerd en het segment waarin

www.dottyflowers.com

dat item is gedefinieerd is niet bekend aan de assembler, volgt een foutmelding.

Labels en andere symbolen

In de meeste traditionele assemblers heten alle gedefinieerde symbolen "label" doch MASM maakt onderscheid tussen verschillende typen. Van welk type een bepaald symbool is wordt bepaald door aan dat symbool een of meer attributen toe te kennen of door het commando waarmee een symbool wordt gedefinieerd. We leerden de programmanaam en de constante reeds kennen. Constanten gedefinieerd met het EQU directive vallen uiteen in twee subtypen: NUMBER en TEXT. Alle door ons gedeclareerde constanten zijn van het type NUMBER. Een speciale vorm daarvan treffen we aan in het datasegment: ENDMSG. Het dollarteken geeft daarbij aan dat de waarde van ENDMSG gelijk is aan de offset waarop de assembler hem aan treft. De naam van de af te drukken string MESSG is ook geen label maar een variabele. MASM kent het datatype "constante" niet maar wel de typen byte, word, double word en nog enkele anderen. MESSG is dus geen label maar een "byte". Labels treffen we alleen aan in het codesegment. Een label hangt altijd aan een instructie. Een label is dan ook een volledig adres of een offset in het codesegment, waarheen een sprongopdracht kan verwijzen. Labels kunnen namelijk van twee basistypen zijn: FAR of NEAR. Er zijn verschillende manieren om een label te definieren. De meest eenvoudige methode is de gekozen naam voor een instructie zetten. Het onderscheid tussen een label en een instructie wordt in MASM aangegeven door een dubbele punt aansluitend aan het label te zetten. Een dergelijk label heeft het attribuut NEAR. Dat wil zeggen dat het label niet geschikt is om vanuit een ander segment naar te verwijzen. "Ongeschikt" is hier niet hetzelfde als onmogelijk. Het kan wel, maar is geen goede programmeertechniek die vraagt om problemen. De betiteling "local" die voor dit label type wel wordt gebruikt is dus onjuist. In het voorbeeld wordt slechts een keer van een dergelijk label gebruik gemaakt: PRDONE.

Procedures

Subroutines worden in MASM, analoog aan hogere programmeertalen "procedures" genoemd. De instructie waarmee een procedure wordt aangeroepen is de CALL en een procedure wordt afgesloten met een RET instructie. Van CALL bestaan twee versies FAR en NEAR. De bedoeling is nu dat met het FAR model alleen procedures in een ander segment (Inter segment) en met het NEAR model alleen procedures in hetzelfde segment (Intra segment) worden aangeroepen. De CALL FAR zet namelijk een compleet terugkeeradres op de stack dat bestaat uit de offset en het segment, dus de registers CS:IP. De CALL NEAR zet alleen een offset op de stack, dus de waarde van IP. Dat betekent dat ook de RET instructie waarmee de procedure eindigt de juiste actie moet verrichten. De programmeur moet er dus voor zorgen dat de procedure met het juiste type RET wordt afgesloten. Om fouten te voorkomen kent MASM een directive dat een label van het type NEAR of FAR definieert en speciaal bedoeld is om procedures te benoemen. In het voorbeeldprogramma zijn twee procedures van het type NEAR aanwezig. START en PRSTR. Het gebruik van het FAR model komt in een latere aflevering aan de orde. Een procedure wordt afgesloten door de naam te herhalen gevolgd door ENDP. Declaratie van een procedure op deze manier is niet verplicht. Men kan een CALL uitvoeren naar elk label. Belangrijk is echter dat alle RET- instructies binnen de procedure automatisch van het type zijn dat als attribuut aan de procedurenaam hangt. Denk er om dat ENDP geen RET assembleert!

De near CALL instructie

De CALL instructie van de 8088 volgens het NEAR model zet de offset naar de volgende instructie op de stack en zet een nieuwe offset in IP welke wordt verkregen uit de operand. Navolgende adresseermethoden zijn beschikbaar:

- Direct. De operand is het label van de aan te roepen procedure.
  Voorbeeld: CALL MYPROC
- Indirect. De operand is een variable (woord) waarin de offset naar de aan te roepen procedure staat. Dit woord mag worden gespecificeerd op alle mogelijke manieren die we eerder hebben leren kennen als "mod". Denk er om dat de operand niet de offset naar de procedure in het codesegment oplevert, maar de offset naar een woord in het codesegment. In dat woord staat de offset naar de procedure in het codesegment!
  Voorbeelden: CALL [MYVECT]
  CALL ROUTINES[BX]
  CALL [BP + SI]
- Register indirect. De operand staat in een register
  Voorbeeld: CALL AX

In dit geval wordt het woord in het register opgevat als de offset naar de procedure in het codesegment. Nogmaals ter vergelijking:

```
MOV AX,5
MOV BX,7
CALL AX     ;Voert de procedure op CS:5 uit
CALL BX     ;Voert de procedure op CS:7 uit
CALL [AX]   ;Foutmelding. AX is geen basis-
              of indexregister
CALL [BX]   ;Voert de procedure op het adres
              in DS:7 uit
```


www.dottyflowers.com

Invoer en uitvoer via DOS

Alle invoer en uitvoer wordt door DOS gedaan via files. Elke file heeft een naam. Voordat een datastroom van of naar een file kan plaats vinden dient deze file eerst geopend te worden, waarbij de naam dient te worden gespecificeerd. Verloopt dit openen foutloos, dan is de naam onbelangrijk geworden omdat DOS aan elke open file een uniek nummer toekent. Zo'n nummer heet een "handle". Het openen en sluiten van files komt in een latere aflevering aan bod. DOS opent zelf bij het opstarten reeds een vijftal files waaraan de handles 0..4 worden toegekend. Deze handles kunnen we zonder meer direct gebruiken.

0. Standaard invoer (Normaliter het toetsenbord)
1. Standaard uitvoer (Normaliter het beeldsherm)
2. Standaard uitvoer voor foutmeldingen (Beeldscherm)
3. Standaard seriele invoer/uitvoer (AUX b.v. modem)
4. Standaard printer (PRN = LPT1)

In het voorbeeldprogramma gebruiken we de handle 1 naar het beeldscherm. Het verschil met handle 2 is, dat handle 2 niet kan worden omgeleid. Als straks het programma FIGUUR1.EXE wordt uitgevoerd, heeft dus het DOS commando FIGUUR1 >PRN niet tot gevolg dat de string op de printer wordt afgedrukt. Moet dat wel mogelijk zijn, dan dienen we de constante DISPHAND de waarde 1 te geven.

Standaard handelingen

In een EXE programma zullen we altijd weer een aantal "verplichte figuren" moeten opvoeren. Die hebben te maken met taken die DOS verricht bij het laden en opstarten van een programma:

- DOS maakt een PSP (Program Segment Prefix) aan van 256 bytes. Hierin worden een aantal belangrijke gegevens opgeslagen. We komen dan ook in latere afleveringen nog dikwijls op dit PSP terug.
- DOS initialiseert de stack pointer SS:SP en begint de het programma uit te voeren op het adres dat we hebben gespecificeerd op de laatste regel van de source achter het END directive. DOS initialiseert niet de segmentregisters ES en DS. Bij de start van het programma bevatten die het segment van het PSP. Het is dus noodzakelijk DS te initialiseren. Vergeet men dit, dan werkt het programma niet en er kunnen de vreemdste dingen gebeuren. Meestal wordt u uitgenodigd uw systeem opnieuw op te starten. De eerste twee instructies verzogen de initialisering van DS. Het woord SEG is een assemblerfunctie die de beginparagraaf van het segment van een label of variabele oplevert. We moeten DS initialiseren via AX omdat de adresseermethode "seg,imm" niet bestaat. MOV DS,SEG DATA zou een foutmelding opleveren.
- DOS kent een hoeveelheid geheugen aan het programma toe. Meestal is dat het hele geheugen. Het is een goede gewoonte om overtollig geheugen terug te geven aan DOS. Daartoe dient functie 4AH van de "DOS function manager" DOS stelt de progrmmeur een groot aantal procedures ter beschikking die kunnen worden aangeroepen door het functienummer in AH te zetten en een INT 21H uit te voeren. De voorwaarden waaronder een degelijke aanroep mag plaats vinden is voor elke functie anders en deze informatie behoort deel uit te maken van de documentatie waarover de programmeur beschikt. Functie 4AH vereist dat de paragraaf van het PSP in segmentregister ES staat en dat het nieuwe aantal geheugenparagrafen dat aan het programma zal toebehoeren in BX staat. We moeten dus de omvang van ons programma berekenen in paragrafen. Dit wordt gevonden door de eerste geheugenparagraaf die niet meer bij het programma behoort te verminderen met de paragraaf waarop het PSP staat. Deze laatste staat dus reeds in segmentregister ES. Men ziet nu welke speciale functie het lege segment ZZZZZZZZ heeft. Het levert ons de beginparagraaf van het vrije geheugen die we dus in BX zetten en waarvan we de waarde in ES moeten aftrekken. Dit laatste gaat weer niet rechtstreeks omdat de SUB instructie de addressermethode "reg,seg" niet toestaat. De SUB instructie is een aftrekking die het carrybit (CF) negeert maar wel een geldige carry oplevert. Aan alle voorwaarden is nu voldaan om INT 21H aan te kunnen roepen. De naam ZZZZZZZZ voor het vrije geheugen segment is niet zomaar gekozen. Men kan MASM opdragen de segmenten alfabetisch te rangschikken. Dit segment moet natuurlijk wel altijd het laatste blijven!

De eigenlijke actie

Om de string af te drukken gebruiken we een eigen procedure die geschikt is om elke willekeurige string af te drukken op het "Standard Output Device" Voorwaarde is dat de procedure wordt aangeroepen met de offset naar de string in register SI. In de instructie MOV SI,OFFSET MESSG komen we de tegenhanger van de SEG-operator tegen: OFFSET. OFFSET levert het offsetgedeelte van een adres. De adresseermethode is hier weer immediate. De procedure PRSTRSI wordt nu aangeroepen en bij terugkeer daarvan is ons programma afgelopen en dus moet het op een correcte manier worden afgesloten zodat de DOS prompt weer op het scherm verschijnt.

www.dottyflowers.com

Het afsluiten van een programma

MS-DOS kent vele methoden om een programma af te sluiten. De enige juiste manier voor nieuw ontworpen software is het gebruik van DOS-functie 4CH. Deze wordt aangeroepen met een terugkeercode in register AL (0 = Geen fout). Verder behoeft de programmeur zich nergens zorgen over te maken. Overigens is deze functie pas beschikbaar vanaf DOS versie 3.XX. In een programma als dit is het dan ook noodzakelijk bij aanvang op de juiste DOS versie te controleren. Ik heb dat achterwege gelaten om het eerste voorbeeld niet te ingewikkeld te maken. In een volgend voorbeeld zal het wel worden gedemonstreerd.

De procedure PRSTRSI

Een aantal bytes wordt naar een file geschreven met functie 40H. Daartoe moet de handle in BX staan, DS:DX het volledig adres van het eerste byte bevatten en het aantal te verzenden bytes in CX staan. We praten over bytes, niet over karakters. Hoe deze bytes eventueel moeten worden geinterpreteerd is een zaak voor het ontvangende apparaat. In dit geval is dat het beeldscherm die deze bytes als karakters (ASCII) zal beschouwen. Om DS:DX het adres te laten bevatten hoeven we aan het segment niets te doen. Dat staat reeds in DS. De offset in SI wordt naar DX gebracht waarna DX met een wordt vermeerderd omdat het niet de bedoeling is dat de lengte in byte 0 als data wordt verstuurd. De lengte van de string kan niet zonder meer in CX worden geladen, maar wel in CL. U ziet hier een toepassing van SI als indexregister. Na het laden van de lengte in CL moet het hogere byte van CX, dat is dus CH, worden gewist. De voor de hand liggende methode zou zijn: MOV CH,0. De gekozen methode is echter een standaardmethode om registers te wissen. De exclusive OR van twee gelijke waarden is altijd 0 en een "reg,reg" operatie is veel sneller dan eem "reg.imm" Vervolgens komen we hier een voorbeeld tegen van de speciale functie van register CX. De instructie JCXZ is een conditionele spronginstructie (branch) welke uitsluitend zal worden uitgevoerd als de inhoud van register CX nul is. Er worden dus geen vlaggen in het statusregister getest. Het zal duidelijk zijn waarom deze instructie is opgenomen. We hebben geen behoefte aan het afdrukken van een lege string. Aan conditionele sprongen heeft de 8088 heel wat aan boord. Ex 6502 programmeurs zullen zich hier helemaal thuis voelen want het bereik is beperkt tot +127 en -128 bytes. Aangezien echter de meeste 8088 instructies veel meer geugenruimte vergen als de 6502 programmeur gewend is, moet worden gevreesd dat de assembler u in het begin regelmatig zal confronteren met een foutmelding: "Relative JMP out of range".

Na het voorgaande moet het u mogelijk zijn dit eenvoudige programma tot in de details te begrijpen. Over een groot aantal zaken is echter nog heel veel niet aan de orde gekomen. Gedeeltelijk komen we daarop nog terug. In een groot aantal andere gevallen zal bestuderen van uw documentatie uitkomst brengen.

Ruud Uphoff

```
                NAME    FIGUUR1
                PAGE    53,128
                TITLE   'Voorbeeld van de layout van een EXE programma'
COMMENT         #
                Het is een goede gewoonte om in deze ruimte nadere byzonder-
                heden over het programma te vermelden. Let er op hoe COMMENT
                hier een karakter definieert waarachter commentaar staat van
                onbeperkte omvang tot het wordt afgesloten met datzelfde
                karakter.
                #
                ;---------- CONSTANTEN ----------
DISPHAND        EQU     1               ;MS-DOS handle naar Standard Output Device
CR              EQU     13              ;ASCII voor Carriage Return
LF              EQU     10              ;ASCII voor Line Feed

                ;---------- CODESEGMENT ----------
CODE            SEGMENT
                ASSUME      CS:CODE, DS:DATA, SS:PGMSTACK
START           PROC    NEAR
                ;START is het hoofdprogramma
                MOV     AX,SEG DATA     ;Paragraaf van datasegment in AX (imm)
```


www.dottyflowers.com

```
                MOV     DS,AX                   ;en daarmee DS initialiseren
                MOV     BX,SEG ZZZZZZZZ         ;Beginparagraaf van vrij geheugen in BX
                MOV     AX,ES                   ;Haal PSP paragraaf uit ES naar AX
                SUB     BX,AX                   ;Bereken programma omvang in paragrafen
                MOV     AH,4AH                  ;DOS functie: Geheugen omvang wijzigen
                INT     21H                     ;Claim BX paragrafen op paragraaf ES
                MOV     SI,OFFSET MESSG         ;Offset naar de string in SI (imm)
                CALL    PRSTRSI                 ;Schrijf de string van DS:SI
                MOV     AH,4CH                  ;DOS functiecode "terminate program"
                XOR     AL,AL                   ;Exitcode:0 = "foutloos"
                INT     21H                     ;DOS functiemanager stopt executie
START           ENDP

PRSTRSI         PROC    NEAR
                ;PRSTRSI drukt de string op adres DS:SIaf op het
                ;Standard Output Device.
                MOV     DX,SI                   ;Voor DOS moet DS:DX destring aanwijzen
                INC     DX                      ;dus DX voorbij het lengte byte
                MOV     CL,[SI]                 ;Het aantal te schrijven bytes in CL
                XOR     CH,CH                   ;CH wissen
                JCXZ    PRDONE                  ;Voor 0 bytes doen we niets
                MOV     BX,DISPHAND             ;File handle in BX
                MOV     AH,40H                  ;DOS functie: Schrijf CX bytes vanaf
                                                ;adres DS:DX naar handle BX
                INT     21H                     ;Aanroep van de DOS functie manager
PRDONE:         RET
PRSTRSI         ENDP

CODE            ENDS

;---------------- DATASEGMENT ------------------------------
DATA            SEGMENT
MESSG           DB      ENDMSG-MESSG-1
                DB      'Dit is ons eerste .EXE programma',CR,LF
ENDMSG          EQU     $                       ;$ betekent "deze offset"
DATA            ENDS

                ;---------------- STACKSEGMENT ----------------------------
PGMSTACK        SEGMENT STACK
                DW      100H    DUP     (?)
PGMSTACK        ENDS

                ;---------------- VRIJ GEHEUGEN ---------------------------
ZZZZZZZZ        SEGMENT                         ;Dummy segment: Vrij geheugen
ZZZZZZZZ        ENDS                            ;Altijdde naam ZZZZZZZZ: gaat nooit fout
                END     START
```

Fig. 1: voorbeeld van een MASM source van een .EXE programma

www.dottyflowers.com

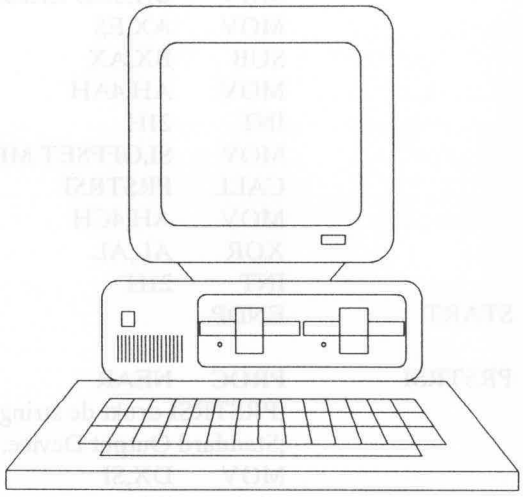
De IBM-PC en z'n klonen (Deel 10)

Toen in 1981 de PC uitkwam was dat wereldnieuws: IBM ging zich (eindelijk) met de microcomputerwereld bemoeien na een stilte van ruim 5 jaar. Gedurende die 5 jaar hadden insiders eigenlijk zitten wachten wat IBM ging doen met dat nieuwe verschijnsel, de microprocessor. Iedereen lette op IBM, die immers ook een trendsetter was geweest in de grotere systemen door de jaren heen. Vandaag de dag weten we allemaal dat IBM inderdaad ook trendsetter is geweest voor de personal computer. Na de PC kwam de PC/XT, een logisch vervolg met wat meer geheugen op het moederbord en de mogelijkheid een harde schijf toe te voegen. De snelheid bleef echter gelijk: een op 4.77 MHz lopende 8088 CPU. En er was al vraag naar meer en beter. Daar werd in eerste instantie alleen op ingespeeld door de (Oosterse) klonenbouwers door de clockfrequentie te verhogen. Eerst naar 6.67 MHz en later naar 8, 10 en tegenwoordig zelfs 12 MHz. Toch bleef iedereen kijken naar IBM hoe de volgende stap eruit moest gaan zien. Men moest wachten tot eind 1984. Toen pas kwam IBM met wat vandaag de dag DE de facto standaard in PC's is: de PC/AT.

De PC/AT

Stond PC/XT voor Personal Computer/eXTended, in PC/AT kreeg de T een nieuwe betekenis: AT staat voor Advanced Technology. Die kreet dekt de lading heel goed, als men de IBM PC/XT als vergelijking aanhoudt. Zo gaat er op het moederbord niet 256 kbyte maar 512 kbyte RAM. De processor loopt harder: 6 MHz, 1 wait state. Diezelfde processor is trouwens van een ander type, namelijk de 16-bit 80286. Dat bij elkaar maakt, dat de oer-PC/AT ongeveer 6 keer zo snel is als de oer-PC/XT. Er was nog meer nieuws. Een nieuwe floppy standaard. Nou ja, standaard.... IBM had namelijk de oude 8-inch compatibele 5.25 inch floppy disk drive van zolder gehaald. Zo'n drive is een vreemde eend in de bijt wat hij draait niet op 300 tpm zoals alle andere 5.25 inch floppy drives maar op 360 tpm. Dat laat een hogere schrijfdichtheid toe: 15 sectoren per track. Om de standaard nog krommer te maken week IBM ook nog af met het aantal tracks dat steeg van 77 (8-inch standaard) naar 80 tracks. Er gaat dan 1.2 Mbyte op 1 schijfje. Die rare 1.2 Mbyte drive is ondanks alles toch uitgegroeid tot standaard: iedere kloon en concurrent heeft hem namelijk.

Die vergroting van 77 naar 80 tracks had een reden. Het was nu met een truc mogelijk de 360 k floppies van de PC(/XT) in de nieuwe drive te lezen. Die nieuwe drive was dus opwaarts compatibel met de drives in de PC/XT. Dat was de rest van de PC/AT trouwens ook.



De 80286 CPU

Om die compatibiliteit met de PC(/XT) redelijk houden zit er aardig wat denkwerk in de AT. Het begint al met de CPU. De 80286 is eigenlijk een soort 16-bit super 8088. Hij verzet per clock cyclus meer werk en heeft de dubbele busbreedte. De 80286 kan in twee modes verkeren: real address mode en protected mode. In real address mode is de processor gewoon opwaarts compatibel met de 8088 en kan hij 1 Mbyte addresseren. Na een reset staat de 80286 altijd in real address mode. Met een instructie kan de processor in protected mode gebracht worden. In deze mode komen er 4 adreslijnen bij, zodat de totale hoeveelheid adresseerbaar geheugen stijgt naar 16 Mbyte. Met behulp van virtuele geheugen technieken is het overigens zo, dat de 80286 schijnbaar 4 Gbyte aankan in deze mode.

Vreemd genoeg is Intel vergeten een instructie in te bouwen, die de processor van protected mode weer terugbrengt naar real address mode. Dat kan dus alleen maar met een reset. Dat heeft een vreemd gevolg: een AT kan zichzelf resetten door middel van een I/O lijntje. Tijdens de POST doet het BIOS dat wel een keer of tien. De gebruiker merkt hier niets van, omdat er voor de reset een getal wordt weggeschreven in de nu wel op het moederbord aanwezige real time clock, dat na de reset weer wordt uitgelezen. Op die manier weet de POST dus, waar ze gebleven was. In IBM-termen heet een dergelijk iets een shutdown. Tijdens de POST komt de AT een aantal keren in protected mode, onder andere om deze mode testen, maar ook om eventueel geheugen boven de 1 Mbyte-grens te testen. Op het moment dat het DOS geboot moet gaan worden, geschiedt de laatste shutdown, zodat de AT altijd in real address mode staat als het DOS wordt binnengehaald.

www.dottyflowers.com

In real address mode is de 80286 opwaarts compatibel met de 8088 met betrekking tot z'n instructieset. De instructieset is groter, ten opzichte van die 8088. Sommige instructies doen echter andere dingen. Een voorbeeld hiervan zijn de instructies PUSH SP en POP SP die de stack pointer op de stack zetten en weer terughalen. De 80286 zet de waarde van SP voor de PUSH op de stack, de 8088 pushed de waarde van SP zoals die wordt nadat de PUSH is gedaan. Instinctmatig doet de 80286 het goed. Zo zijn er nog een paar verschilletjes, die overigens keurig zijn gedocumenteerd in de Technical Reference Manual van de PC/AT.

Mechanische compatibliteit

Ook op het mechanische vlak heeft IBM voor voortgang gekozen. In de PC/AT passen namelijk dezelfde adapterkaarten als in de PC/XT, althans mechanisch. Electrisch niet helemaal. De 80286 is immers een 16-bit processor. De 62 pinnen in de slots van de PC(/XT) waren echter al vol bezet, zodat de vier extra adreslijnen en de 8 extra datalijnen niet benut zouden kunnen worden op extra kaarten. En dat moest wel, vanwege het feit dat het moederbord gelimiteerd was tot 512 kbyte RAM en dus niet de volle 640 kbyte die DOS herkent toelaat, om nog maar te zwijgen van de volle 16 Mbyte die 80286 zelf aankan.

In de PC/AT passen dezelfde adapterkaarten als in de PC/XT, althans mechanisch.

IBM loste dit op door aan de slots een extra 36-polige connector toe te voegen in het verlengde van de al bestaande 62-polige connector. Op de 62-polige connector bleef de bestaande PC(/XT) bus gehandhaafd. Nu had IBM een aantal I/O kaarten in het programma die door hun bouwwijze toch niet in zo'n 16-bit slot met twee conector pasten, waaronder de CGA en MDA kaarten. Ook hier werd een oplossing voor gevonden: bij twee van de acht slots werd de 36-polige extra connector weggelaten. Deze slots werden dan ook 8-bit slots of XT-slots genoemd. Op deze manier kon je dus oude kaarten in de nieuwe machine gebruiken. Op de 62-polige connector zijn alle slots aan elkaar gelijk. Het is dus geen bezwaar om een 8-bit kaart in een 16-bit slot te gebruiken, mits het mechanisch past. Merk op dat de geschiedenis zich op dit punt herhaald heeft met de introductie van de EISA-bus voor 32-bit machines. Het enige verschil was, dat de EISA-bus geheel buiten IBM om tot stand is gekomen.

De byte swapper

Met de truc met de 8- en 16-bit slots ben je er niet. Want hoe komt de processor erachter of een geheugenlocatie die op de I/O slots zit nu 8- of 16-bit is? Of in andere woorden, mag een 16-bit woord in 1 keer 16-bit breed worden opgehaald of weggeschreven, of moet die actie worden gesplitst in twee 8-bit operaties op opeenvolgende addressen? Dit probleem geldt niet alleen voor geheugentransfers of voor de CPU. Het treedt ook op voor de DMA-controllers en voor I/O operaties, die bij een 80286 ook 16-bit kunnen zijn. Zolang je op het moederbord werkt weet je waar je aan toe bent, want het moederbord is voor iedere machine hetzelfde. Het wordt pas moeilijk op de I/O slots, want dat is het deel dat de gebruiker vult.

Ook dit probeem werd opgelost. Er was (gelukkig?) nog een pin over op de 62-polige bus. Als een kaart in een slot die pin naar ground trekt, is dat een teken dat het een 16-bit kaart is. Is het signaal hoog, dat is het 8-bit kaart. Blijkt het laatste het geval, dan treedt er een bijzonder stukje hardware op het moederbord in werking: de byte swapper. De byte swapper detecteert, dat er uitsluitend 8-bit transfers kunnen plaatsvinden en doet derhalve het volgende:

- Het genereert de adreslijn A0
- Het splitst een 16-bit transfer in twee 8-bit transfers

De adreslijn A0 ontbreekt bij 16-bit processors: er worden immers altijd 16-bits (twee bytes) tegelijk verwerkt. Daarom moet de byte swapper zelf die adreslijn erbij maken. De byte swapper genereert verder geheel zelfstandig, buiten de CPU om een tweede bus cycle na de eerste, waarbij A0 hoog gemaakt wordt en het upper byte op de lower byte databus verschijnt. Dit alles dus buiten de CPU om, die al die tijd gewoon in de wacht wordt gezet.

De byte swapper is er overigens de reden van, dat de PC/AT met minimaal 1 wait state werkt. Aan het begin van een bus cyclus zet de CPU het adres op de bus. Het moederbord en de kaarten in de slots starten nu hun adresdecodering. Pas als een kaart ontdekt heeft of hij geadresseerd wordt kan het signaal dat aangeeft of het een 8- of een 16-bit kaart is geactiveerd worden, en dat is dus ergens midden in de buscyclus. Op dat punt is door IBM dus een wait state ingevoerd, om dit mechanisme te laten werken. Latere kloon-machines omzeilden dit punt ten dele door het adres zo vroeg mogelijk op de bus te zetten. Op I/O kaarten is echter altijd tenminste die ene wait state extra nodig.

De byte swapper geeft derhalve al aanleiding voor een redelijk portie extra logica op het moederbord.

www.dottyflowers.com

Moederbord uitbreidingen

Met die byte swapper ben je er ook nog niet. Er zal ook 16-bit DMA mogelijk moeten zijn wil je tenminste ook op dit gebied je partijtje mee kunnen blazen. Er komt dus een DMA controller bij, want de 8-bit controller uit PC(/XT) is gewoon blijven zitten waar hij zit. Twee 8237 DMACs op het moederbord dus. Dus ook twee DMA pagina registers. Die pagina registers moeten nu echter 8 bit breed zijn, dus een 74(L)S670 voldoet niet meer: het wordt nu een 74LS612 memory mapper. Overigens is de truc met de DMA-controller om de DRAMs te refreshen in de PC/AT verlaten: het moederbord bevat nu een echte refresh-teller. Wel is de generatie van de refresh timing gelijk aan die in de PC(/XT), want de timer is ook gewoon overgenomen.

Ook in de PC/AT wordt kanaal nul van de timer gebruikt voor de systeemtijd, kanaal een voor de refresh en kanaal twee voor de luidspreker. Om de timing gelijk te houden en ook om de CGA kaart te kunnen gebruiken, is op het moederbord ook de 8284 clockgenerator van de PC/XT aanwezig, compleet met 14.318 MHz kristal.

De truc met de DMA-controller om de DRAMs te refreshen is in de PC/AT verlaten: het moederbord bevat nu een echte refresh-teller.

Een zelfde verhaal geldt eveneens voor de 8259 interrupt controller. De PC/AT heeft er twee, die in cascade geschakeld staan. De master controller is precies zo aangesloten als in de PC(/XT) met 1 verschil: op IRQ2 zit nu niet de I/O bus, maar de tweede 8259. Omdat IRQ2 in de PC(/XT) op de I/O bus voorkwam is op dezelfde pin van de bus nu IRQ9 van de tweede 8259 aangesloten. Het BIOS zorgt ervoor dat bij een IRQ9 er een INT 0Ah wordt uitgevoerd waardoor de oude software die IRQ2 gebruikt gewoon blijft werken.

De meeste I/O chips op het PC(/XT) moederbord vinden we dus ook weer in de AT. Er is echter ook een belangrijke uitzondering en dat is de toetsenbord interface.

Die toetsenbord interface is namelijk intelligent geworden. Was het in de PC(/XT) een 8255 en een schuifregister, in de PC/AT treffen we een 8042/8742 aan. De 8042 is een complete microcomputer met RAM, ROM en I/O poorten. Hij ziet er aan de buitenkant wat merkwaardig uit: er zit een normale bus interface voor I/O poorten aan. De hoofd-CPU praat dus met de keyboard controller (wat zo heet dat ding) alsof het I/O poorten zijn. De keyboard controller neemt twee I/O addressen in beslag waarvan er 1 compatibel is met de PC/XT: 60h en 64h.

Die twee adressen worden als volgt benut:

60h lezen: output buffer
60h schrijven: input buffer
64h lezen: keyboard controller status
64h schrijven: KB controller commando

Het lezen van adres 60h heeft hetzelfde effect als in de PC(/XT): data van het keyboard ophalen. Nieuw is het omgekeerde: schrijven naar poort 60h. Het byte dat daar wordt weggeschreven wordt naar het toetsenbord gestuurd. Dat kan, want zoals in het verhaal over het toetsenbord reeds is uitgelegd, kan een AT-toetsenbord ook commando's van de PC uitvoeren. Een voorbeeld is de besturing van de lock-LEDs, of het instellen van de repeat-rate.

Op poort 64h (lezen) vinden we het status register van de keyboard controller. Hierin zitten bits waaraan te zien is of de in- en/of output buffers leeg of gevuld zijn, of de keyboard controller bezig is met het verwerken van een commando (zie hieronder) en kan ook de status van de keylock schakelaar worden uitgelezen. Verder zijn er statusbits voor transmissie- en ontvang-time-outs.

Poort 64h (schrijven) is helemaal bijzonder: het is de commando poort. Door deze poort kunnen commando's aan de keyboard controller gegeven worden. Op die manier kan de controller geinitialiseerd worden, kan om een KB controller selftest gevraagd worden, en kan bijvoorbeeld de interface met toetsenbord getest worden op sluitingen en op open verbindingen. Verder is het mogelijk om de I/O poorten van de keyboard controller zelf te initialiseren. Aan een van die poorten hangt overigens de reset-lijn om de eerder genoemde shutdowns te kunnen doen.

Omdat de keyboard controller intelligent is, zijn er meer mogelijkheden dan in de PC(/XT). Ten eerste is aan het transmissie formaat met het keyboard een pariteitsbit toegevoegd. Verder kan de controller in het geval van pariteitsfouten ook aan het toetsenbord vragen om het laatste byte nog eens te versturen. Er is nog meer. De keyboard controller kan ook scancodes vertalen van de ene naar de andere set. Een AT-toetsenbord stuurt namelijk andere scan codes dan een PC(/XT) toetsenbord. De keyboard controller vertaalt deze codes naar de gebruikelijke XT-codes. Soms levert dat meer dan 1 scancode per toetsaanslag op. Een voorbeeld daarvan zijn de losse pijltjestoetsen in het enhanced toetsenbord.

www.dottyflowers.com

Het omgekeerde kan ook: een PC(/XT) toetsenbord aansluiten. In dat geval wordt de pariteitscontrole uitgezet, en blijft de scancode vertaling achterwege. Dit laatste feature ontbreekt trouwens in de meeste moderne keyboard controller BIOSsen in klonen. Al met al een behoorlijk complex ding die keyboard controller. Het keyboard BIOS in een AT is dan ook behoorlijk wat groter dan in een XT.

Er is nog een extra chip aanwezig op het moederbord: de eveneens al eerder genoemde CMOS real time clock, compleet met batterij. Die klok is van het type 146818, en bevat ook nog wat RAM (50 bytes). Behalve de tijd en datum wordt de klok ook gebruikt om de systeem configuratie te bewaren, compleet met een checksum. De POST controleert niet alleen het systeem, maar ook of de configuratiedata nog klopt. Is dat niet het geval, dat moet er speciaal programmaatje worden gedraaid, dat bij IBM op een schijfje staat: SETUP. SETUP zet de CMOS configuratiedata zoals de gebruiker dat opgeeft en rekent de checksum opnieuw uit. Nadat ook die opgeslagen wordt de machine koud gestart om de nieuwe data te effectueren. Tegenwoordig hebben alle kloon-BIOSsen een ingebouwde SETUP, vaak menu gestuurd.

IBM wilde uitwisselbaarheid met de bestaande PC's in twee richtingen hebben. De 1.2 Mb drive kan (helaas) ook 360 kb diskettes schrijven. De drive schrijft dan slechts een half spoor, en dat garandeert niet altijd dat een 40 track drive dat smalle spoor betrouwbaar kan lezen. Een verschijnsel waarmee iedere AT bezitter weleens mee geconfronteerd is, meestal op een zeer ongelegen moment.

Praktische uitvoering

Al met al is de AT heel wat uitgebreider dan een PC(/XT). Ofschoon dat best gekund had, is er vrijwel geen gebruik gemaakt van programmeerbare logica, en is de hele AT opgebouwd met F- en ALS-TTL. Op het moederbord komen dan ook meer dan 130 IC's voor. Die vereisen ook hun ruimte: het moederbord is dan ook gigantisch groot en ligt in de kast ten dele onder de harde schijf en zelfs onder de voeding. Al die IC's vreten natuurlijk een berg stroom, zodat ook de voeding heel wat meer uit de kluiten gewassen is: 200 Watt is standaard. De kast biedt verder plaats aan in totaal 5 slim line 5.25 inch drives. IBM monteerde in het begin uitsluitend harde schijven in volle hoogte zodat de configuratie dan beperkt wordt tot twee floppy drives, een winchester en eventueel een halve hoogte winchester (onder de twee floppy disk drives). De 1.2 Mbyte werd de standaard floppy drive op de AT. Als optie kon een tweede 1.2Mb of 360 kb drive worden besteld. Omdat er machines waren die maar 1 (1.2Mb) floppy drive hadden, en IBM uitwisselbaarheid met de bestaande PC's in twee richtingen wilde hebben, kunnen de 1.2 Mb drive (helaas) ook 360 kb diskettes schrijven. De drive schrijft dan slechts een half spoor, en dat garandeert niet altijd dat een 40 track drive dat smalle spoor betrouwbaar kan lezen. Een verschijnsel waarmee iedere AT bezitter weleens mee geconfronteerd is, meestal op een zeer ongelegen moment.

Net als bij de PC en de XT begonnen de klonenbouwers in eerste instantie met het compleet kopiëren van de IBM AT in ieder opzicht. Al spoedig verschenen de eerste snellere borden: zowel 6 MHz 0 wait state als 8 MHz 1 wait state bleek niet echt moeilijk te realiseren. Na enige tijd verschenen zelfs 10 MHz 0 wait state moederborden, nog steeds compleet in TTL opgebouwd. Dat waren de laatste TTL-borden. Want er was een nieuwe ontwikkeling gaande. Chipsets.

Chipsets

Die enorme hoeveelheid IC's op het moederbord had behalve voor de grootte van de voeding nog meer gevolgen. Allereerst financieel: Al die IC's moeten niet alleen betaald, maar ook verwerkt worden. Verder werd het moederbord zoals reeds vermeld behoorlijk groot. Zo groot dat de AT voorzien moest worden van een behoorlijk uit de kluiten gewassen kast. De Amerikaanse firma Chips & Technologies bedacht iets nieuws: alle domme TTL van het moederbord werd in een vijftal chips gebakken. Om problemen met Intel te voorkomen bleven de I/O chips (2x8259, 2x8237, 1x8254, en de 146818 klok) gewoon op het moederbord zitten. Verder werd in die chipset de decodering van het RAM wat uitgebreid: er kon tot maximaal 1 Mbyte op het moederbord.

Voor het eerst werd het mogelijk met deze chipset een AT compatibel moederbord te bouwen met de afmetingen van een XT moederbord. Omdat een

www.dottyflowers.com

AT-kast iets andere bevestigingsgaten heeft dan een XT-kast, moest het moederbord alleen iets langer worden. Dergelijke moederborden kregen de aandoenlijke naam "baby-AT". Ze zijn meestal voorzien van een dubbel stel bevestigingsgaten, waardoor baby-borden eventueel ook in XT-kasten gemonteerd kunnen worden.

In eerste instantie werd er enigszins sceptisch tegen de nieuwe ontwikkelingen aangekeken, te meer omdat de maximum clocksnelheid van de eerste versies beperkt was tot 8 MHz 1 wait state, terwijl er al TTL-moederborden waren die tot 10 MHz 0 wait state gingen. Met de introductie van de 10 MHz chipset, werd het tij gekeerd: de baby-AT had een vaste plaats veroverd.

Toen acties van IBM uitbleven om Chips & Technologies aan te pakken wegens diefstal van het AT-ontwerp verschenen er spoedig meer fabrikanten die een AT-chipset aanboden. Enige namen op dit gebied zijn: VLSI, G2, Suntac en Zymos. Deze laatste fabrikant slaagde er zelfs in, om vrijwel alles op het AT-moederbord behalve CPU, BIOS en RAM in slechts twee chips onder te brengen, inclusief de 146818 klok. Dat laatste was best knap, want dat leverde een combinatie van NMOS en CMOS in 1 chip op. De moederbordfabrikanten op hun beurt reageerden met baby-moederborden die bijvoorbeeld ook een floppy controller, een printerpoort en een tweetal seriële poorten aan boord hadden.

In 1988 introduceerde C & T een set van drie chips met bijzondere eigenschappen. De chipset kreeg de naam NEAT mee, een afkorting voor New Enhanced AT. In de NEAT set is zo ongeveer alles te programmeren, terwijl er ook aan zowat alles is gedacht.

Chips & Technologies bleef niet stil zitten. De eerste chipset werd uitgebreid met een zesde chip, die alle I/O chips in zich verenigde. Verder werd de maximum kloksnelheid gestaag groter terwijl er inmiddels extra schakelingen bedacht waren om de chipset met 0 wait states te laten werken. De standaard kloksnelheid voor deze chipset was 12.5 MHz, meer dan een verdubbeling van de oorspronkelijke IBM-snelheid. Nog was de koek niet op.

In 1988 introduceerde C & T een set van drie chips met bijzondere eigenschappen. De chipset kreeg de naam NEAT mee, een afkorting voor New Enhanced AT. De chipset was geheel uitgevoerd in CMOS en dus geschikt voor portable computers. Er waren een tweetal I/O adressen toegevoegd, waarmee iets bijzonders mogelijk was: de hardware opnieuw configureren. De ingestelde configuratie wordt bewaard met behulp van de toch al aanwezige batterij voor de klok en de setup. De maximum kloksnelheid was 20 MHz, terwijl er tot 8 Mbyte geheugen op het moederbord kan met de set. Dit laatste is mogelijk, omdat ook 1 Mbit DRAMs ondersteund worden.

In de NEAT set is zo ongeveer alles te programmeren, terwijl er ook aan zowat alles is gedacht. Zo kunnen de kloks voor de CPU, de I/O bus en de DMA controllers elk afzonderlijk worden gekozen. Hierdoor kan bijvoorbeeld de I/O bus op de originele IBM-AT snelheid bedreven worden, maar ook eventueel op de halve CPU kloksnelheid. Omdat de set zoveel RAM aankan is ook op dit gebied de variatie groot. Het simpelst is natuurlijk 640kbyte basisgeheugen, de rest boven de 1 Mbytegrens te leggen. Hiermee zijn we er nog niet: in het gebied van 640k tot 1M kan ook RAM zitten, onder de standaard daar aanwezige zaken als het EGA/VGA BIOS en het systeem-BIOS. Dat RAM is apart in te schakelen, en ook kunnen de andere ROMs in dat adresgebied worden uitgeschakeld. Op die manier ontstaat er dus een schaduw van die geheugengebieden, die het mogelijk maakt om bijvoorbeeld het (EGA/VGA) BIOS in snel 0 wait state RAM te draaien in plaats van in de van nature trage EPROMs. Behalve gewoon lineair geheugen ondersteunt de NEAT set ook hardware EMS-geheugen: een bankswitched geheugensysteem dat in een latere aflevering nog besproken zal worden. De BIOS fabrikanten reageerden met zeer fraaie setup-programma's in hun BIOSsen voor de NEAT set.

De volgende keer...

gaan we nog even verder over de AT. Onder meer over het BIOS, dat werd uitgebreid. Tot dan.

Nico de Vries

www.dottyflowers.com

Nu beschikbaar: versie 3.10 van PCFIX

Er is een verbeterde versie van het pakket PCFIX ter beschikking van de KGN. PCFIX is een pakket voor MS-DOS waarmee men de informatie op al dan niet harde schijven kan onderhouden. In het onderstaande stukje worden de verschillen met versie 1.00 aangegeven.

Verschillen tussen versie 3.00 en 1.00

Verschillende fouten in versie 1.00 zijn verholpen. De "critical error handler" is herzien. Versie 3.00 werkt nu ook op een AT of PS/2 systeem met DOS 4.0x. Als er een DOS functie actief is zal PC-FIX niet actief worden. Versie 3.00 kan verder ook met externe massageheugens werken.

De volgende fouten in versie 1.00 zijn verbeterd:

- Onder DOS 4.0x trad een onverwachte foutmelding op. Verder werkte versie 1.00 niet onder DOS 4.0x wanneer het geactiveerd werd vanaf de DOS prompt of vanuit de DOS SHELL.
- Versie 1.00 werkt niet op sommige 10 Mb harddisks omdat het een 16-bit FAT verwacht, zoals gedocumenteerd in de "DOS 3.3 Technical Reference". Dit bleek een fout in de IBM documentatie te zijn.
- Onder versie 1.00 kunnen de meest vreemde dingen gebeuren als een subdirectory precies 32 entries heeft, inclusief de gewiste files en de entries "." en "..".
- Versie 1.00 was niet in staat een sectornummer uit een 12-bit FAT te lezen dat op een sectorgrens stond. Dit kan een probleem zijn op een 10 Mb harde schijf.
- Het COMPARE commando meldde af toe verschillen in files die degelijk aan elkaar gelijk waren.
- Er zat een kleine bug in het SORT commando.
- Versie 1.00 werkte niet op media die een andere sectorlengte dan 512 byte gebruiken. Een voorbeeld hiervan kan een RAM-disk zijn. Versie 3.00 werkt correct met sectorgrootten van 128, 256 en 512 bytes. De zelden gebruikte sectorgrootte van 1024 bytes wordt niet ondersteund en veroorzaakt nu een foutmelding.

Versie 3.00 heeft verder een aantal uitbreidingen ondergaan t.o.v. versie 1.00. Het nieuwe commando "System Help" levert belangrijke informatie over DOS functies, BIOS interrupts en codes van het keyboard en karaktercodes. Verder kunnen naar eigen inzicht nog 3 extra hoofdstukken, ieder bestaand uit 20 helpschermen worden geïnstalleerd, zodat de gebruiker steeds de belangrijkste systeeminformatie bij de hand heeft zonder in handboeken te moeten bladeren. Deze informatie kan worden toegevoegd met behulp van de EDITHELP utility, dat ook deel uitmaakt van het PC-FIX pakket.

De "physical disk explorer", dat onder versie 1.00 een apart programma was is nu in versie 3.00 geïntegreerd. De residente teksteditor WPFIX is gewijzigd zodat nu alle source files van Turbo Pascal geaccepteerd worden: de maximum tekstbuffer is nu 64kb in plaats van de vroegere 40 kb.

Het nieuwe commando MOVE maakt het mogelijk files tussen verschillende directories en/of drives te verplaatsen.

WAARSCHUWING:

De maximum disk partitie die PCFIX aankan is 32 Mb. Verder ondersteunt PCFIX de 32-bit sectornummers die met DOS 4.0x werden geïntroduceerd (nog) niet. PCFIX test NIET of er 32-bit sectornummer gebruikt worden!

Systeem configuratie

PCFIX is niet hardware onafhankelijk: het werd ontworpen voor gebruik op de IBM PC/XT/AT en PS/2 modellen en werkelijk compatibele machines. In het bijzonder moeten het display-adresmechanisme en de BIOS interrupts voldoen aan de oorspronkelijke IBM specificaties. PCFIX spreekt direct de device drivers aan, zodat ook deze aan de PC-DOS specificatie moeten voldoen. De DOS versie moet 3.2 of hoger zijn.

- PCFIX is niet geschikt voor gebruik in netwerk- en multi tasking omgevingen!

LAATSTE NIEUWS

Er is een ernstige fout geconstateerd in versie 3.00: PSP/DTA management was onjuist. Dit kan resulteren in een onjuist display of zelfs zodanige overschrijvingen in het geheugen dat een hangend systeem ontstaat. Daarom: gebruik versie 3.00 beslist niet: er nu versie 3.10.

In deze versie is de FILER SORT uitgebreid: er is nu een keuzebalk te zien die met ^X en ^Y bewogen kan worden. DE file die op die manier werd gekozen kan vervolgens met PgDn en PgUp naar een andere plaats in de directory gebracht worden.

Bestellen

Leden van de KIM Gebruikersclub Nederland kunnen PCFIX bestellen door *f* 7,50 over te maken op giro 3757649 t.n.v. de KIM Gebruikersclub Nederland. S.v.p. even aangeven om welk programma het gaat. Indien u een 3.5 inch diskette wenst, dan ook dit s.v.p. even vermelden.

Ruud Uphoff

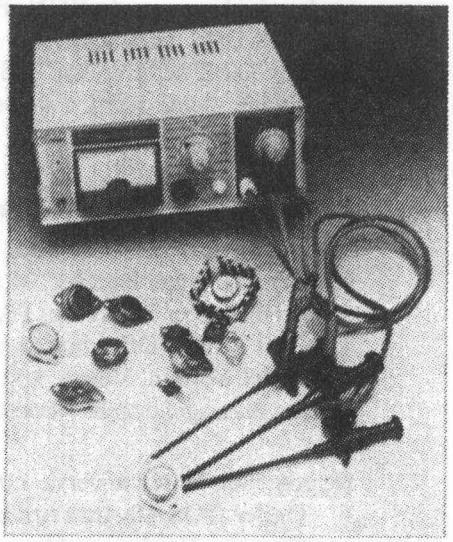
www.dottyflowers.com

HET NEUSJE VAN DE ZALM!

LET OP:

DIL ELEKTRONIKA IS WEGENS BALANS GESLOTEN VAN 31 DEC. T/M 4 JAN. ZATERDAG 5 JAN 1991 ZIJN WIJ DAN WEER MET-RAAD-EN-D(R)AAD VOOR U PARAAT!
WIJ WENSEN AL ONZE (POSTORDER) KLANTEN EEN PRETTIGE DECEMBER-MAAND!

Een nieuwe serie bouwpakketten van DIL naar aanleiding van beschrijvingen in het elektronika-vakblad ELEKTUUR.
Deze kits worden geleverd inkl. kast, front, knoppen, chassisdelen en evt. voeding; ekskl. 'kleinmateriaal' zoals draad, boutjes en soldeer.
Eerst de bouwbeschrijving lezen.....?
Bij elk pakket staat aangegeven in welke Elektuur u die kunt terugvinden: wij mogen geen foto-kopieën bijleveren van de uitgever.



LF-HF-SPEURNEUS

Dit veelzijdige meetapparaat herbergt onder één dak een signaalgenerator, een meetversterker, een millivoltmeter en een meeluisterversterker. Al met al een bijzonder praktische kombinatie dus om te meten, te testen en fouten te zoeken.

Zie Elektuur december 1989.

Bestelkode: 890183-T **219.-** inkl. BTW

WISSELSPANNINGSVOEDING

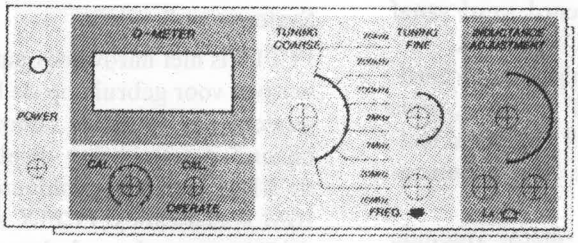
Een apparaat dat een regelbare wisselspanning levert van 0,5-30V bij een stroomsterkte van maximaal 1A. Zie Elektuur oktober 1990.

Bestelkode: 900104-T **229.-** inkl. BTW

Q-METER

Naast de zelfinduktie zijn ook de resonantiefrekwentie en de Q-faktor van een spoel van belang. De kwaliteitsfaktor-meter in deze kit voert de meting van de Q-faktor uit op de resonantiefrekwentie van een LC-seriekring. Deze testfrekwentie kan worden ingesteld tussen 70KHz en 70MHz; tevens is dit apparaat bruikbaar als HF-sinusgenerator. Uitlezing d.m.v. eksterne dig. multimeter of LCD-paneelmeter (niet bij pakketprijs inbegrepen!)

Zie Elektuur september 1990.



Bestelkode: 900031-T **215.-** inkl. BTW

FOCAL SUBWOOFER

10V516, zie beschrijving in Elektuur november/december '90.

NIEUW! Prijs luidspreker **329.-**

TRANSISTOR-HFE-METER

Dit instrument kan ook betrouwbaar de versterkingsfaktor meten van power-transistoren, waarbij gemeten wordt bij een grote collector-stroom. D.m.v. puls-testen kan de voeding klein blijven en behoeft "het slachtoffer" zelf niet gekoeld te worden.
Zie Elektuur september 1990.

Bestelkode: 900078-T **199.-** inkl. BTW

MILLI-OHM-METER

Het bepalen van doorgangs- en kontaktweerstanden van kabels, schakelaarkontakten e.d. is met een universeelmeter niet betrouwbaar mogelijk. Deze milli-Ohm-meter werkt volgens de 4-punts meetmethode en geeft overgangsweerstanden aan van 100 milli-Ohm tot 5 Ohm. Een must voor de audiofiel en de hardware-computerfreak.

Bestelkode: 910004-T **189.-** inkl. BTW

FUNKTIEGENERATOR LF-SWEEPGENERATOR

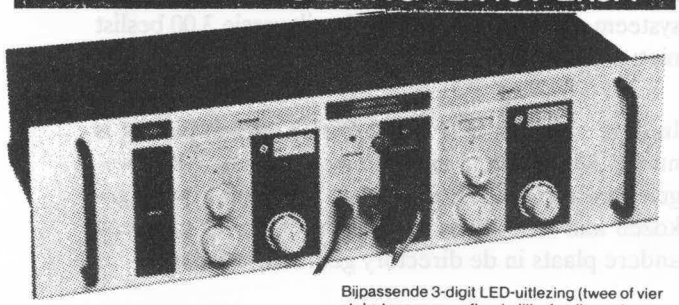
Een funktiegenerator is op de testbank feitelijk onmisbaar; dit ontwerp biedt echter tevens nog een LF-sweepfunktie en dat is in deze prijsklasse toch wel bijzonder.
Zie Elektuur mei 1990.

Bestelkode: 900040-T **209.-** inkl. BTW

NIEUWE ELEKTUUR BOUWPAKKETTEN

900020	9009*	**GITAARSTEMMER** ekskl. kast	89.50
900042	9010	**ELEKTRONISCHE BELASTINGSWEERSTAND** inkl. koelplaten en hoekprofiel	159.00
900081	9010	**BASIC-TELEFOONCENTRALE** inkl. relais en geprogr. eprom	325.00
900083	9009*	**DIMLICHT-AUTOMAAT** ekskl. kastje	45.00
900098	9010	**MONO EINDVERSTERKER** ekskl. kast en voeding	225.00
900098-MV	9010	**MONO VOEDING** voor 900098 ekskl. kast	217.50
900098-SV	9010	**STEREO VOEDING** voor 900098 ekskl. kast	275.00
900106	9011	**TERMOMETER PT-100** ekskl. opnemer	69.00
900111	9011	minutieuze **MD-VOORVERSTERKER** ekskl. kast	159.00
904039	9009	**INGANGSKEUZE-SCHAKELAAR** ekskl. relais	89.95

DUBBELE LABVOEDING 2x40V 2x5A



HET PARADEPAARDJE UIT DEZE SERIE

Twee geheel gescheiden helften, zodat ook parallel-schakelen (0-40V bij 10A!) en serieschakelen (0-80V bij 5A!) mogelijk wordt.

Uitstekende stabilisatie, weinig warmte-ontwikkeling, bestand tegen overbelasting en kortsluiting. Wordt geleverd eksklusief digitale uitlezing,

Zie Elektuur oktober 1990.

Bestelkode: 900082-T **749.-** inkl. BTW

Bijpassende 3-digit LED-uitlezing (twee of vier stuks toepassen, afhankelijk of u alleen stereospanning (?) of ook de stereo-stroom wilt kunnen "monitoren". **LET OP:** dit is een SMD-ontwerp, en dus misschien niet voor iedere doe-het-zelver geschikt.

Zie Elektuur november 1989.

Bestelkode: 890117 **79.50** inkl. BTW

ALTERNATIEF

Velleman meet-module: 3-digit LCD-paneelmeter.

Bestelkode: VK1000 **79.50** inkl. BTW

DIL elektronika

TELEFOON 010 - 4854213 / TELEFAX 010 - 4841150
JAN LIGTHARTSTRAAT 59-61, 3083 AL ROTTERDAM

*** partikulieren:**
Per brief met ingesloten EUROCHEQUE, GROENE BANKBETAALKAART of GIROBETAALKAART. (ondertekenen en pasnummer invullen) verzendkosten f 6,50 GEEN minimum orderbedrag

Door VOORUITBETALING op onze postgirorekening 649943 of ons bankrek. nr. 69.45.65.644 verzendkosten f 6,50 GEEN minimum orderbedrag

Per telefoon: levering geschiedt onder REMBOURS. Orders boven f 100,-: verzendkosten f 10,- Voor kleine orders: verzendkosten f 15,-

*** openingstijden en winkelverkoop:**
DINSDAG t/m VRIJDAG 9.00 - 17.30 uur
ZATERDAG: 9.00 - 16.00 uur.
GESLOTEN: op maandag en vrijdagavond.

*** levering volgens onze standaardleveringsvoorwaarden**

*** bedrijven/instellingen:**
Toezending per PTT of NPD na ontvangst van uw bestelbon of uw opgave per telefax.
Orders boven f 100,-: verzendkosten f 7,50
Voor kleinere orders: verzendkosten f 15,-

BALIEVERKOOP (voor levering 'op rekening' altijd een bestelbon of zakelijke legitimatie meenemen).

Na voorafgaande afspraak is maand-fakturering mogelijk voor diegenen die geregeld kleine aantallen komponenten nodig hebben.

AL ONZE PRIJZEN ZIJN INKL. BTW (tenzij anders vermeld).

*** voor België Elektro-8000 PVBA.**
Langestraat 108. B 8000 Brugge.
Tel. 050 - 341007 / Fax. 050 - 341168

www.dottyflowers.com

REKEN, een spreadsheet(je) in Pascal voor DOS-65

Inleiding

Het bij dit artikel behorende Pascal-programma is een eenvoudig spreadsheet, ontwikkeld voor DOS-65. Het doel van dit programma is het volgende: Speel er eens mee. Als er voldoende belangstelling is, kan wellicht een goede versie in assembler geschreven worden. Laat uw commentaar, kritiek, suggesties etc. eens horen.
(Noot van de redactie: Behalve rechtstreeks kunt u de auteur ook bereiken via het KIM-Club Bulletin Board "The Ultimate" en uiteraard ook via de redactie. In het laatste geval kan (een deel van) de discussie van wat goed en minder goed is mogelijk in de uP Kenner gevoerd worden).

Vragen

- Bestaat er documentatie over hoe men een goed spreadsheet programmeert? Er is namelijk een moeilijkheid met formules: het is mogelijk dat formules celverwijzingen bevatten naar andere formules. In dat geval moet er een strategie zijn om het gehele rekenblad te berekenen terwijl toch iedere cel maar 1x gedaan wordt.
- Hoe bergt men de informatie op op disk? Hierbij mag de file niet onnodig groot worden maar het spreadsheet moet wel na inlezen zonder veel rekenwerk te gebruiken zijn.
- Sommige bestaande spreadsheets hebben wel heel erg veel ingebouwde functies. Dat past nooit allemaal in 64 kB. Dan moet er gekozen worden voor een beperkt aantal functies die echt nodig zijn. Welke? Of moeten er meerdere versies gemaakt worden: financieel en technisch? Moderne pakketten voor de PC zijn vaak 3-dimensionaal, is dat alleen om indruk te maken of gebruikt men dat echt?
- Deze demo-versie is in Pascal geschreven zodat snel een aantal details kunnen worden uitgeprobeerd. Als bekend is hoe alles moet worden, dan is het beter om dit in assembler te programmeren. Voor dit prototype is Pascal echter beter omdat het schrijven van een programma veel sneller gaat en omdat veel beter te controleren is waar evt. fouten zitten. Compileren moet met de -E optie, i.v.m. de case..else. Als u veranderingen aanbrengt dan is het beter om te compileren met:

-T error PA -E reken

Hierdoor komen alle foutmeldingen in de file "error" waar ze later met behulp van de tekstverwerker te bekijken zijn. Nb: Dit programma werkt zo niet op een PC want Borland's Turbo Pascal is volgens de ANSI norm en de Pascal op het DOS65 systeem is volgens ISO.

Op mijn systeem, 6502 CMOS, 2 MHz, Teac diskdrives, duurt compileren en assembleren circa 5 minuten. Na het compileren is dit programma circa 23 kByte groot.

PAS OP: De DOS65 Pascal heeft een fout in de "dispose" waardoor dat niet werkt en soms zelfs het systeem geheel vastloopt. Er zit ook een fout bij het uitwerken van conditionele expressies waardoor bijv. het volgende niet goed gaat:

IF (a3) AND (a) THEN ...

Na erg veel zoeken ben ik er in geslaagd om voor beide problemen een oplossing te vinden maar die is moeilijk over te dragen. Heeft de ontwerper van Pascal deze fouten inmiddels hersteld? Eventueel kunt u alle "dispose" statements weglaten, maar wanneer u dan veel wijzigingen uitvoert, kan het geheugen vol raken.

Moderne pakketten voor de PC zijn vaak 3-dimensionaal, is dat alleen om indruk te maken of gebruikt men dat echt?

Coördinaten

Hoofdletters geven een absolute coördinaat aan, kleine letters een relatieve dus:

R2C3is de cel in rij 2 en kolom 3.

r2c3is de cel die 2 rijen en 3 kolommen verder staat t.o.v. de huidige cel.

r2c-1is de cel die 2 rijen verder staat in de vorige kolom.

Bijna overal waar absolute verwijzingen staan, mogen ook relatieve verwijzingen.

Gewoon typen:

appels is een text
46 is een integer
3.14 is een real

www.dottyflowers.com

Cursortoetsen

Deze kunt u gebruiken alsof u bij met gewoon typen bezig bent:
enter: ga naar kolom 1 op de volgende rij.
^Hcursor links
^Icursor rechts
^Jcursor neer
^Kcursor op

Commando's

Als u de ESCAPE-toets indrukt gevolgd door een ? dan krijgt u de onderstaande lijst commando's op uw scherm:

fformule invoer in de huidige cel
r reken het gehele spreadsheet door
g ga naar een bepaalde cel
e erase cel, row, colum, range, all
c copy cel naar cel of naar een range
q quit
s save rekenblad in bestand (nog niet gemaakt)
r read rekenblad van bestand (nog niet gemaakt)
(Hoe doe ik save en read m.b.v. deze Pascal?)
Rekenen: ESCAPE r

In het gehele rekenblad wordt iedere cel 1x berekend. Daarbij wordt in R1C1 begonnen, van links naar rechts en van boven naar beneden. Als uw formules gebruik maken van de resultaten van formules in andere cellen, dan gaat dat dus niet altijd in één keer goed.

Stel: in cel R1C1 staat R2C1*R3C1, in cel R2C1 staat sin(R4C1) en in cel R4C1 staat een getal. Dan zal na de eerste keer berekenen alleen R2C1 de juiste waarde hebben. Pas na de tweede keer is ook cel R1C1 juist.

Formules = ESCAPE f:

Het rekenblad accepteert in een formule iedere legale combinatie van:

integer-getallen
real-getallen
verwijzingen naar andere cellen
sin(E) = sinus van expressie, E cos(E) = cosinus van expressie E
sum(R) = som van range R
min(R) = minimum van range R
max(R) = maximum van range R
expressies met haakjes ()
^ * / + - = machtsverheffen, vermenigvuldigen, delen, optellen en aftrekken
pi = 3.141593

Bij de expressies is de prioriteit: eerst de expressie tussen haakjes, dan machtsverheffen, gevolgd door vermenigvuldigen en delen en als laatste optellen en aftrekken.
De som-functie verwacht een range van te sommeren cellen, dus bijvoorbeeld:
sum(R2..6C9) = sommeer in kolom 9 van rij 2 t/m rij 6
sum(R3C7..9) = sommeer in rij 3 van kolom 7 t/m kolom 9
sum(R2..6C7..9) = sommeer alle cellen van rij 2..6 in de kolommen 7..9
sum(R2C7..R6C9)is FOUT
sum(r1..7C5) = sommeer alle rijen van 1 t/m 7 (relatief t.o.v. de huidige cel) in kolom 5.

Iets dergelijks geldt ook voor min en max.

Relatieve verwijzingen:

r1c0 = de cel die onder de huidige cel ligt
r-1c0 = de cel die boven de huidige cel ligt
r0c1 = de cel die rechts naast de huidige ligt
r0c-1 = de cel die links naast de huidige ligt
r2c3 = de cel die 2 rijen lager en 3 kolommen verder ligt
r2C3 = de cel die 2 rijen lager ligt in kolom 3

Goto = ESCAPE g:
Na deze opdracht volgt de vraag naar welke cel.
Voorbeelden:
R2C7 = zet de cursor in cel R2C7
r0c5 = zet de cursor in dezelfde rij maar 5 cellen verder
r1C4 = zet de cursor 1 rij lager en in kolom 4

copy = ESCAPE c:
Hierna volgt de vraag: cel range. U kunt nu 1 cel opgeven, zowel absoluut als relatief, dan wordt de inhoud van de huidige cel gekopieerd naar de opgegeven cel. U kunt ook een range opgeven, bijvoorbeeld R6C3..7. In dat geval wordt de inhoud van de huidige cel gekopieerd naar de opgegeven cellen. Bij het kopiëren van een formule wordt deze in de nieuwe cel uitgerekend. Als daarbij een fout ontstaat, wordt de formule niet gekopieerd.

erase = ESCAPE e:
Na de opdracht volgt de vraag:
cel formule Row Column range all.
cel = de huidige cel wordt gewist
formule = in de huidige cel wordt alleen de formule gewist
Column = de huidige kolom wordt gewist
Row = de huidige row wordt gewist
range = geef een te wissen gebied op, bijv. R1..5C7
all = het gehele rekenblad wordt gewist

Noot van de redactie: Hoewel het programma vrij omvangrijk is, vond ik het toch zinvol dit programma

www.dottyflowers.com

te plaatsen. Uiteraard kan het zeer interessant zijn dit spreadsheet ook op andere computers te laten lopen. Zelf ben ik van plan het programma zeker naar Microsoft Pascal om te zetten en mogelijk ook nog naar een Pascal-versie voor een VAX/VMS-systeem en een Stratus-systeem onder VOS. Hebt u dit pakket op een ander systeem lopen, laat mij dat dan even weten. Mocht u het pakket in een andere taal omgezet hebben (C, Forth, Basic, assembler) dan ben ik uiteraard ook geïnteresseerd. (Gert van Opbroek).

H.Speksnijder

```
Program reken(input,output);

    { Demo Spreadsheet}
    { for Dos65 Pascal  }
    { 1990 07 17  }
    { H.Speksnijder }
    { Capelle a/d IJssel  }

const   rowmax=    12;   { number of rows                    }
        colmax=     7;   { number of colums                  }
        strlen=    15;   { max. length of a standard string  }
        breed=      9;   { breedte van een colom             }
        endmark=   '\';  { mark for end of string            }
        BS  =     127;   { cursor back space                 }
        LEFT=       8;   { cursor left                       }
        RIGHT=      9;   { cursor right (tab)                }
        DOWN=      10;   { cursor down                       }
        UP=        11;   { cursor up                         }
        TAB=        9;   { tabulator (=8 spaces)             }
        CRET=      13;   { cariage return                    }
        ESC=       27;   { escape                            }
        CLRSCR=    12;   { clear screen                      }
        CLREOL=    26;   { clear end of line                 }

type string = packed array [1..strlen] of char;
    valtyp= (et,it,rt,tt); { empty,integer,real,text }
    cel=record formule: ^string;
    case kind:valtyp of
        et: ( ev: char);    { empty }
        it: ( iv: real); { real }
        rt: ( rv: real); { real }
        tt: ( tp: ^string);    { string }
    end;

var key : char;    { for input from keyboard }
        keynr : integer;  { this is ord(key) }
        text : string;  { for input of a line }
        t : 1..strlen;  { as pointer in text[t] }
        ok: boolean;  { the result of a procedure }
        quit: boolean;
        row,col: integer; { cursor position }
        V: array[1..rowmax,1..colmax] of cel;{ sp. sheet }
    { key, keynr, text, t, ok, quit, row, col, V }
    { are global variables and are used by many }
    { procedures and functions, but they are never }
    { passed as parameter. }
```


www.dottyflowers.com

```
procedure readkey; { READ CHARACTER and also CR }
begin
    if eoln          { remark: ALWAYS read(key) after eoln }
    then begin read(key); key: = chr(CRET); end
    else read(key);
        keynr: = ord(key);
end;

procedure celprtyp(cr,cc:integer);    { PRINT CEL TYPE }
    begin with V[cr,cc] do
    case kind of
        et: write('empty ');
        it: write('integer');
        tt: write('text  ');
        rt: write('real  ');
    end;
end;

procedure celpr(cr,cc:integer);    { PRINT CEL VALUE }
var c:integer;
    begin with V[cr,cc] do
        case kind of
        et: for c: = 1 to breed do write(' ');
        tt: for c: = 1 to breed do write(tp ^ [c]);
        it: write(round(iv):breed);
        rt: begin if (abs(rv) < 10000)
            then write(rv:breed:2)
            else begin
                write('<!>');
                for c: = 3 to breed do write(' ');
            end;
        end;
    end;
end;

procedure shinit;    { INITIALIZE SHEET }
var i,j:integer;
begin
    write('clear all');
    for i: = 1 to rowmax do
        for j: = 1 to colmax do
            with V[i,j] do begin
                formule: = nil;
                kind: = et;
                end;
end;

procedure erasecel(re,ce: integer);    { ERASE CEL }
begin
    with V[re,ce] do begin
        if formule < > nil then dispose(formule);
        formule: = nil;
        if kind = tt then begin
            if tp < > nil then dispose(tp);
            end;
        kind: = et;
```


www.dottyflowers.com

```
    end;
end;

procedure getstring;{ READ STRING }
begin
    write('>');
    t:=1; readkey;
    while (t<strlen) AND (keynr>=32) AND (key<>endmark) do begin
        text[t]:=key; t:=t+1; write(key);
        readkey;
        while (keynr=BS) do begin
            if (t>1) then begin
                write(chr(LEFT),' ',chr(LEFT));
                t:=t-1;
                end;
            readkey;
            end;
    end;
    text[t]:=endmark; { mark end of string }
    ok:= (t>1);
    while (t<strlen) do begin t:=t+1; text[t]:=' ' end;
end;

procedure writecom( z:packed array[a..b:integer] of char);{ WRITE COMMAND }
begin
    write(chr(UP),chr(CRET),chr(TAB),chr(CLREOL));
    write(chr(UP),chr(CRET),chr(TAB),z,' ');
end;

procedure display(cr,cc:integer); { DISPLAY SHEET }
var r,c,x,y,i:integer;  { cr=cursor row, cc=cursor colum }
    {remark r,c is position in sps, x,y is position on display }
    procedure rcpr(ch:char;i:integer);
    begin
        write (ch);
        if i<10 then write(i:1)
        else write(i:2);
    end;
    begin
        r:=1; y:=1; write(chr(CLRSCR)); { clear screen }
        write('R  C');
        for c:=1 to colmax do begin
            write('    ',c:1);  { show colum number }
            for i:=6 to breed do write(' ');
        end;
    writeln; y:=y+1; x:=1;
    repeat begin
        c:=1;
        write(r:2,' '); { show row number }
        repeat begin
            if (r=cr) AND (c=cc)
            then write('[.....]')
            else celpr(r,c);
            c:=c+1; x:=x+8;
            end until (x>70)OR(c>colmax);
    writeln; x:=1; y:=y+1;
```


www.dottyflowers.com

```
    r:=r+1;
    end until (y>20) OR (r>rowmax);
    rcpr('R',cr); rcpr('C',cc);
    write('     ');
    if V[cr,cc].formule=nil
    then begin
        celprtyp(cr,cc); celpr(cr,cc)
    end
    else write('formule ',V[cr,cc].formule^);
end;

procedure getvalue(var getal:real; var inputyp:valtyp);
var i,sign: integer;
    power: real;
begin
    i:=t; power:=10; getal:=0; sign:= +1;
    if (text[i] = '-') then begin sign:= -1; i:=i+1 end;
    if (text[i] = '+') then i:=i+1;
    while text[i] in ['0'..'9'] do begin
            getal:= 10*getal + ord(text[i]) - ord('0');
            i:=i+1;
    end;
    if text[i]='.' then begin
        i:=i+1;
        while text[i] in ['0'..'9'] do begin
            getal:= getal + (ord(text[i])-ord('0'))/power;
            i:=i+1; power:= power*10;
        end;
    end;
    if power=10 then inputyp:= it else inputyp:= rt;
    if (getal=0) then inputyp:= et;
    if (i>t) AND (text[i]=endmark)
    then begin
        ok:=true; t:=i;
        end
    else ok:=false;
    getal:= sign * getal;
end;

function ival:integer;    { IVAL get integer value }
var getal:integer;
begin
    getal:=0; t:=t+1;
    while (text[t] in ['0'..'9']) do begin
        getal:= 10*getal + ord(text[t]) - ord('0');
        t:=t+1;
    end;
    ival:= getal;
end;
function sival:integer;   { SIVAL get signed integer }
begin
    case text[t+1] of
    '-': begin t:=t+1; sival:=0-ival end;
    '+': begin t:=t+1; sival:=ival    end
    else sival:=ival;
    end;
```


www.dottyflowers.com

```
end;

procedure celref(var a,b:integer; rabs,cabs:integer );{ CELREF }
begin                    { cel reference like R3C5 }
    case text[t] of        { or like r-2c0 or R4c-1 }
    'R':    begin
        a:=ival;
        case text[t] of
            'C': b:=ival;
            'c': b:=cabs+sival
        else ok:=false;
        end;
    end;
    'r': begin
        a:=rabs+sival;
        case text[t] of
            'C': b:=ival;
            'c': b:=cabs+sival
        else ok:=false;
        end;
    end
    else ok:=false;   { error not RyCx }
    end;
end;
procedure range(var rb,re,cb,ce: integer;{ RANGE }
        rabs,cabs:integer);
    procedure frange(var fbeg,fend:integer;
        abspos:integer; relativ:boolean);
    begin
        if relativ
        then fbeg:=sival+abspos
        else fbeg:=ival;
            if (text[t]='.') and (text[t+1]='.')
            then begin
                t:=t+1;
                    if relativ
                    then fend:=sival+abspos
                    else fend:=ival;
                end
            else fend:=fbeg;
        end;
    begin                    { body RANGE }
        if (text[t]<>'R') AND (text[t]<>'r')
        then ok:=false
        else begin
            frange(rb,re,rabs,(text[t]='r') );
            if (text[t]<>'C') AND (text[t]<>'c')
                then ok:=false
                else frange(cb,ce,cabs,(text[t]='c') );
    end;
end;

procedure arex(var final:real; rabs,cabs:integer);{ ARITHMETIC EXPRESSION }
{ remark: rabs,cabs= absolute row,col for relative reference }
        function E1:real; forward;  { EXPRESSIE NIVEAU 1 forward }
        function celval(r,c:integer):real;   { CELVAL + body }
```


www.dottyflowers.com

```
    begin
      if (r>0) AND (r<=rowmax) AND (c>0) AND (c<=colmax)
      then with V[r,c] do  { cel waarde bepalen }
        case kind of
          et: begin ok:=false; celval:=0 end;
          it: celval:= iv;
          rt: celval:= rv;
          tt: begin ok:=false; celval:=0 end;
        end
      else ok:=false;
    end;
function value:real;  { VALUE }
  var fr,fc:integer;
function rval:real;   { RVAL + body }
var getal,power:real;
begin
      getal:=0;
      while (text[t] in ['0'..'9']) do begin
            getal:= 10*getal + ord(text[t]) - ord('0');
            t:=t+1;
      end;
  if text[t]='.' then begin
      t:=t+1; power:=10;
      while (text[t] in ['0'..'9']) do begin
        getal:= getal + (ord(text[t])-ord('0'))/power;
        t:=t+1; power:= power*10;
      end;
  end;
rval:= getal;
end;
begin                  { body VALUE }
  if text[t] in ['0'..'9']
  then value:= rval    { it is a number }
  else case text[t] of
    'R' : begin
      celref(fr,fc,rabs,cabs);  { fr,fc=row/col from text }
      if ok then value:=celval(fr,fc); { get cel value V[fr,fc] }
    end;
    'r' : begin
      celref(fr,fc,rabs,cabs);  { fr,fc=row/col from text }
      if ok then value:=celval(fr,fc);
    end;
    'p' : begin          { it is pi=3.141593 }
      if text[t+1]='i'
      then begin
        value:=3.141593;
        t:=t+2;
      end
      else ok:=false;
    end;
    '(' : begin          { it is a (E1) expression }
    t:=t+1;
    value:= E1;
    if text[t]=')'
    then t:=t+1
    else ok:=false; { error no bracket }
```


www.dottyflowers.com

```
        end
        else ok: = false;
    end;
end;
function funoran(function f(a,b:real):real;ini:char):real;{ FUNORAN }
        var ra,rb,ca,cb,fi,fj:integer; { function on range }
    rtemp: real;
    begin
        range(ra,rb,ca,cb,rabs,cabs);
        if ini = 'z'
        then rtemp: = 0
        else rtemp: = celval(ra,ca);
        if ok then begin
            fi: = ra;
            while (fi < = rb) AND ok do begin
                fj: = ca;
                while (fj < = cb) AND ok do begin
                    rtemp: = f(rtemp,celval(fi,fj) );
                    fj: = fj + 1;
                end;
            fi: = fi + 1;
            end;   end;
funoran: = rtemp;
end;
function sum(a,b:real):real;     { function SUM }
begin
    sum: = a + b;
end;
function min(a,b:real):real;{ function MINIUM }
begin
    if a < b then min: = a else min: = b;
end;
function max(a,b:real):real;     { function MAXIMUM }
begin
    if a > b then max: = a else max: = b;
end;
function E5:real;{ EXPRESSIE NIVEAU 5 }
    var tfun:packed array[1..4] of char;
    p:integer;
    begin
        for p: = 0 to 3 do tfun[p + 1]: = text[t + p];
        t: = t + 4;
        if tfun = 'sin(' then E5: = sin(value);
        if tfun = 'cos(' then E5: = cos(value);
        if tfun = 'sum(' then E5: = funoran(sum,'z');
        if tfun = 'min(' then E5: = funoran(min,'m');
        if tfun = 'max(' then E5: = funoran(max,'m');
        if text[t] = ')'
        then t: = t + 1
        else ok: = false;
        end;
function E3:real;{ EXPRESSIE NIVEAU 3 }
    var first,second: real;    { doe ^ machtsverheffen }
    ttemp:integer;
        begin
            { writeln('E3  t:',t); }
```


www.dottyflowers.com

```
            ttemp:=t; first:= value;
            if not ok then begin
               t:=ttemp; ok:=true;
               first:=E5;
            end;
         while (text[t]='^') do begin
            t:=t+1;
            second:= value;
            first:= abs(first);
            first:= exp(second * ln(first));
         end;
      E3:= first
end;

function E2:real;{ EXPRESSIE NIVEAU 2 }
   var first: real;{ do * and / }
begin
   { writeln('E2  t:',t); }
   first:= E3;
   while (text[t]='*') OR (text[t]='/') do
      if text[t]= '*'
      then begin t:=t+1; first:= first * E3 end
      else begin t:=t+1; first:= first / E3 end;
   E2:= first
   end;
function E1;  { EXPRESSIE NIVEAU 1 }
   var first: real;{ do + and - }
begin
   { writeln('E1'); }
   first:= E2;
   while (text[t]='+') OR (text[t]='-') do begin
      if text[t]= '+'
      then begin t:=t+1; first:= first + E2 end
      else begin t:=t+1; first:= first - E2 end;
   end;
   E1:= first;
end;
begin                    { AREX body }
   ok:= true; t:=1;
   { remark E1..E4 have side effect on t and ok }
   final:= E1;
   if (text[t]<>endmark) then ok:= false;
   if NOT ok then begin
      write('calculation error in cel:R',rabs:1,'C',cabs:1);
      readkey;
   end;
end;
procedure formunew(rn,cn: integer); { NEW FORMULA }
var uitkomst: real;
      j: integer;
begin
   arex(uitkomst,rn,cn);
   if ok
   then begin { write('ok:',uitkomst:6:2);}
      j:=1;
      erasecel(rn,cn);
```


www.dottyflowers.com

```
        with V[rn,cn] do begin
            new(formule);
            formule ^ : = text;
            kind: = rt; rv: = uitkomst;
        end;
    end;
end;
procedure celchange( r,c:integer);    { CHANGE CEL }
var getal: real;
        inputyp: valtyp;
begin
    erasecel(r,c);
    t: = 1;
    getvalue(getal,inputyp);
    if ok
    then with V[r,c] do            { verwerk een getal }
        case inputyp of
            rt : begin
                kind: = rt;
                rv: = getal;
            end;
            it : begin
                kind: = it;
                iv: = getal;
            end;
        end
    else begin          { verwerk een text }
        t: = 1;
        while (t < strlen) AND (text[t] < > endmark) do t: = t + 1;
        text[t]: = ' ';
        with V[r,c] do begin
            kind: = tt; new(tp);
            tp ^ : = text;
        end;
    end;
end;
procedure celedit;   { CEL EDIT }
begin
    ok: = true;
    if V[row,col].formule = nil
    then begin getstring;{ read a line }
        if ok then celchange(row,col); { put line in cel }
        end
    else repeat readkey until keynr in [8,9,10,11,13,27];
end;
procedure erase;{ ERASE }
var i,j,rw,rw2,cw,cw2: integer;
begin
    write('erase: cel formule Row Colum range all');
    readkey;
    case key of
    'c' : erasecel(row,col);   { erase this cel }
    'f' : with V[row,col] do begin    { clear formule only }
        if formule < > nil
        then dispose(formule);
        formule: = nil;
```


www.dottyflowers.com

```
    end;
    'R' : for cw: = 1 to colmax do erasecel(row,cw);{ clear row }
    'C' : for rw: = 1 to rowmax do erasecel(rw,col);{ clear column }
    'r' : begin                          { clear range }
            writecom('range:'); getstring;
            if ok then begin
               range(rw,rw2,cw,cw2,row,col);
               for i: = rw to rw2 do
                  for j: = cw to cw2 do
                     erasecel(i,j);
            end;
          end;
    'a' : begin write('ALL ? y/n');  { clear all }
          readkey;
          if key = 'y' then begin
             for rw: = 1 to rowmax do
                for cw: = 1 to colmax do
                   erasecel(rw,cw);
             end;
          end
          else ;
    end;
end;
procedure copycel(rnew,cnew,row,col: integer);{ COPY CEL }
begin
    text: = V[row,col].formule ^;
    formunew(rnew,cnew);
end;
procedure copycelv(rnew,cnew,row,col: integer);{ COPY CEL VALUE }
begin
    erasecel(rnew,cnew);
    with V[rnew,cnew] do begin
       kind: = V[row,col].kind;
       case kind of
          it: iv: = V[row,col].iv;
          rt: rv: = V[row,col].rv;
          tt: tp ^ : = V[row,col].tp ^
       else ;
       end;
    end;
end;
procedure copy; { COPY }
var i,j,rnew,rnew2,cnew,cnew2: integer;
begin
    writecom('copy to: cel range');
    readkey;
    case key of
    'c': begin                  { copy to one cel }
       writecom('cel:'); getstring;
       if ok then begin
          t: = 1;
          celref(rnew,cnew,row,col);
          if ok AND ( (rnew < > row) OR (cnew < > col) )
          then if V[row,col].formule = nil
       then copycelv(rnew,cnew,row,col)
       else copycel(rnew,cnew,row,col);
```


www.dottyflowers.com

```
    end;
end;
    'r': begin              { copy to a range of cels }
        writecom('range:'); getstring;
        if ok
        then begin
            t:=1;
            range(rnew,rnew2,cnew,cnew2,row,col);
            if V[row,col].formule=nil
            then begin
                for i:=rnew to rnew2 do
                    for j:=cnew to cnew2 do
                    copycelv(i,j,row,col);
                end
            else begin
                for i:=rnew to rnew2 do
                    for j:=cnew to cnew2 do
                    copycel(i,j,row,col);
                end;
            end
        end
    else;
    end;
end;
procedure getform; { GET FORMULE }
begin
    write('formule:');
    getstring; { read a line }
    if ok then formunew(row,col); { put formule in cel }
    ok:=true;
end;
procedure calculate;    { CALCULATE }
var fr,fc: integer;
begin
    ok:= true; fr:=1;
    while ok AND (fr<=rowmax) do begin
        fc:=1;
        while ok AND (fc<=colmax) do begin
            with V[fr,fc] do
                if formule<>nil then begin
                text:=formule^;
                arex(rv,fr,fc);
            end;
            fc:= fc+1;
        end;
        fr:= fr+1;
    end;
    if NOT ok then write('calculation error in cel:R',fr:1,'C',fc:1);
    ok:=true;
    keynr:=CRET;
end;
procedure shedit;    { SHEET EDIT }
begin
    repeat begin
    display(row,col);
    celedit;
```


www.dottyflowers.com

```
    case keynr of
        LEFT : if col>1      then col:=col-1;
        RIGHT: if col<colmax then col:=col+1;
        DOWN   : if row<rowmax then row:=row+1;
        UP     : if row>1      then row:=row-1;
        CRET: begin col:=1; if row<rowmax then row:=row+1 end;
        ESC  :
        else ;
        end;
    end
        until keynr=ESC;
end;
procedure command;  { COMMAND }
var rnew,cnew:integer;
    procedure help; { HELP }
    begin
        writecom('?=more help');
        write(' formule');
        write(' rekenen');
        readkey;
        if key='?' then begin
            writecom(' >');
            write(' goto');
            write(' copy');
            write(' erase');
            write(' quit');
            readkey;
        end;
    end;
begin
    ok:=true;
    writecom('command');
    readkey;
    if key='?' then help;
    writecom('command');
    case key of
    'f' : getform;  { formule input }
    'g' : begin
        write('go to cel:');   getstring;
        if ok then begin
            t:=1;
            celref(rnew,cnew,row,col);
            if ok then begin
                row:=rnew;
                col:=cnew;
            end;
        end;
    end;
    'e' : erase;{ value, formule, Row, Colum, range, all }
    'c' : copy; { to cel, range }
    'r' : calculate;{ calculate sheet  }
    'q' : begin
        write('quit ? y/n');
        readkey;
        if key='y' then quit:=true;
            end
```


www.dottyflowers.com

```
        else ;
    end;
end;
{ MAIN MAIN MAIN MAIN MAIN MAIN MAIN MAIN MAIN MAIN MAIN MAIN }
begin
    shinit ; quit: = false; row: = 1; col: = 1;
    repeat begin
        shedit;
        command;
    end until quit;
    write(chr(CLRSCR));  { clear screen }
end.
```

(Advertentie)

KOBRA PC SYSTEMS

Aanbieding: diskettes (NoName)

- 5.25" DD ..... *f* 6,95
- 5.25" HD ..... *f* 14,95
- 3.5" DD ..... *f* 13,95
- 3.5" HD ..... *f* 27,95

Alle diskettes in doosjes van 10 stuks.

Motherboards:

- Motherboard AT 6/12 MHz..... *f* 399,-
- Motherboard DX-386 20 MHz..... *f* 1799,-
- Motherboard DX-386 33 MHz..... *f* 2450,-
- Motherboard 80486 25 MHz..... *f* 5200,-

Printerlinten

- Epson LX800 enz..... *f* 9,95
- Star LC10 ..... *f* 9,95

100% Garantie (ook op noname diskettes!)

Prijzen incl. BTW

Bestellingen:

Vooruitbetaling op NMB 66.53.97.186
of onder rembours (+ *f* 6,00)

KOBRA PC SYSTEMS
Postbus 40195
7504 RD Enschede
Tel.: 053 - 776101
Technische ondersteuning tussen 19:00 en 22:00 uur.

www.dottyflowers.com

Methoden en technieken voor datacommunicatie (deel 4)

Inleiding

Deze serie gaat over de manieren waarop computers in het algemeen en home dan wel hobbycomputers in het bijzonder met elkaar communiceren. In deel één hebben we het gehad over de manier waarop de informatie de computer binnenkomt of uitgaat. In deel twee ging het over de opbouw van het telefoonnet en in deel drie hebben we het gehad over modems.

In dit deel gaan we het hebben over een paar kruimels die bij de behandeling van de modems zijn blijven liggen. Verder gaan we het hebben over een vorm van standaardisatie die men heeft aangebracht als er over communicatie tussen computers gesproken wordt.

Modemkruimels

Eén van de kruimels die de vorige keer is blijven liggen is de zogenaamde Hayes commando set. Deze speelt in de communicatie met en tussen home en hobbycomputers een dermate belangrijke rol dat we er in een serie over datacommunicatie niet omheen kunnen. Ook in de μP Kenner is er over de Hayes commando's al vaker geschreven en de meeste mensen die een intelligent modem hebben zullen wel weten waar het allemaal om draait. Voor de beginners toch nog even een kleine uitleg.

Modems heb je in verschillende soorten. Je hebt zogenaamde domme modems die je door middel van schakelaars in de juiste toestand moet zetten. Dat betekent dat je eerst met de telefoon het telefoonnummer waarop een ander modem is aangesloten (053-303902!) moet kiezen en als je het andere modem hoort piepen zet je de spraak/data schakelaar om. Ook de instelling van het modem (baudrate etc.) doe je door middel van schakelaars.

Natuurlijk kun je ook een wat intelligenter modem maken. De instellingen van een dergelijk modem doe je niet met schakelaars maar met de computer. Via de verbinding tussen computer en modem kun je opdrachten naar het modem sturen zoals "Schakel over naar auto answer mode". Je zou zelfs een modem kunnen bouwen die zelf het gewenste telefoonnummer kiest en als hij een modem aan de andere kant van de lijn hoort ook overschakelt van spraak naar data. Vrijwel alle modems die nu op markt zijn kennen dergelijke kunstjes. Het verhaal gaat namelijk nog verder..... In plaats van dat elke modem-fabrikant zijn eigen commando's gebruikt, is er één, vrijwel overal geaccepteerde, standaard. Deze standaard is door de firma Hayes, uiteraard fabrikant van modems, ontwikkeld.

Het lijkt mij niet zinvol de Hayes-commando's te behandelen. Bij elk modem hoort een gebruiksaanwijzing en in deze gebruiksaanwijzing is ongetwijfeld de Hayes-commando set beschreven, tenzij het modem deze natuurlijk niet ondersteunt. Voor het verdere verhaal wil ik alleen even globaal aangeven wat allemaal mogelijk is met een modem met Hayes commando's.

Bij een Hayes (compatibel) modem kunnen alle instellingen vanaf de computer ingesteld worden. Vaak is het modem hierbij ook nog in staat zelf uit te zoeken met welke snelheid (bitrate) de computer de informatie stuurt. Verder kan ook het omzetten van "schakelaars" vanaf de computer gebeuren en de meeste modems kunnen in opdracht van de computer een telefoonnummer kiezen, meestal zowel in pulsmode als met behulp van de in computergestuurde telefooncentrales gebruikte toontjes. Vaak kan het modem ook detecteren dat de lijn "in gesprek" is zodat aan de computer verteld kan worden dat de aangevraagde verbinding niet tot stand kan komen. Een laatste feature die de modems kunnen hebben is het, in overleg met het modem aan de andere kant, vaststellen van de te gebruiken snelheid. Aan de computer wordt dan meegedeeld op welke snelheid gewerkt wordt zodat ook de computer zich op deze snelheid kan instellen.

Met behulp van een intelligent modem en een goed communicatiepakket (zelf gebruik ik op de PC Telix en op andere systemen een uitgebreide versie van Kermit) kun je op comfortabele wijze een modemverbinding tot stand brengen. Vaak kunnen communicatiepakketten, samen met een modem met Hayes commando's geheel zelfstandig verbinding maken met bijvoorbeeld een bulletin board. Aan het communicatieprogramma wordt dan opgegeven welk telefoonnummer gedraaid moet worden waarna de verbinding tot stand gebracht wordt. Is het gekozen nummer in gesprek, dan wacht het pakket even om vervolgens opnieuw een poging te wagen. Een pakket zoals Telix gaat nog een stapje verder. Met dit pakket kan men via een menu aangeven welke van de bekende bulletin boards gekozen moet worden waarna het pakket de verbinding tot stand brengt en alle instellingen goed zet. Telix gaat zelfs zover dat er een aantal standaard handelingen automatisch uitgevoerd kunnen worden. Telix verzorgt dan niet alleen de verbinding, hij vertelt het bulletin board ook wie er inlogt, wat het wachtwoord is etc. Ook in diverse andere pakketten, onder andere in de Kermit versie voor UNIX en aanverwante systemen, worden dergelijke LOGIN-scripts ondersteund.

www.dottyflowers.com

Een tweede kruimel die is blijven liggen is een iets uitgebreidere bespreking van de MNP klassen. Steeds meer modems krijgen tegenwoordig, behalve de term V22bis of iets dergelijks de toevoeging MNP4 of zelfs MNP5. In deze aflevering wil ik iets dieper op deze indeling ingaan.

Zoals de Hayes commando's een link leggen naar modems van de fabrikant Hayes legt MNP (Microcom Networking Protocol) een link naar de modemfabrikant Microcom. Bij microcom heeft men een aantal klassen van modemdefinities gespecificeerd. Deze klassen hebben als gemeenschappelijk kenmerk dat het modem in staat is fouten in de overdracht van informatie te detecteren en te corrigeren. Verder vindt er, bij de hogere MNP-klassen, in het modem een data-compressie plaats waarbij het ontvangende modem de expansie van de data voor zijn rekening neemt. Let wel, de aangesloten computersystemen merken hier niets van, alleen dat de effectieve snelheid van de verbinding sterk toeneemt.

MNP klasse 1 beschrijft binnen MNP de meest eenvoudige modems. Het gaat hier om byte-geörienteerde asynchrone half-duplex modems. Dat wil dus zeggen dat de informatie byte voor byte of letter voor letter overgestuurd wordt. Wel wordt er, door middel van het oversturen van extra informatie, voor gezorgd dat er een foutdetectie en -correctie mogelijk is. Modems uit MNP klasse 1 halen een effectiviteit van ongeveer 70%. Een V22bis modem (2400 bps) met MNP 1 haalt dus een effectieve snelheid van 1680 bps.

Bij MNP2 hebben we te maken met byte-geörienteerde asynchrone full-duplex modems. Deze hebben een effectiviteit van ongeveer 84%.

Een modem uit MNP klasse 3 werkt niet meer asynchroon maar synchroon. Dit heeft als voordeel dat niet voor elk byte een stopbit en een startbit meegestuurd behoeft te worden. Bij synchrone communicatie gebeurt dat maar één maal per blok. Voorwaarde is dan wel dat er een continue stroom van informatie is. Dit is echter meestal niet het geval. Om dit op te vangen, voegt het modem zelf extra bits, zogenaamde vulbits, toe. Verder is het modem bit-georienteerd en full-duplex. Een dergelijk modem heeft een effectiviteit van 108%.

Modems uit MNP klasse 4 kom je tegenwoordig vrij regelmatig op de markt tegen. Deze modems hebben twee extra mechanismen om de overdracht van gegevens te versnellen en te verbeteren. In de eerste plaats is er de Adaptive Packet Assembly. Dit mechanisme past de lengte van de overgestuurde informatieblokken aan de kwaliteit van de lijn. Bij een goede verbinding wordt de lengte van een blok 256 byte, bij een zeer slechte lijn wordt dit verminderd tot 32 byte. Dit heeft als voordeel dat er een optimum gezocht wordt tussen een lang blok (= hoge snelheid) en een kort blok (= weinig overtollige informatie sturen als een blok niet goed overkomt). Een dergelijk mechanisme wordt ook in het populaire protocol voor filetransfer ZMODEM gebruikt.

Het tweede mechanisme dat in MNP4 gebruikt wordt is de zogenaamde Data Phase Optimization. Gebleken is namelijk dat veel (stuur-) informatie in de datablokken vaak ongeveer hetzelfde blijft. Door middel van de optimalisering wordt deze overtollige informatie weggehaald. Een MNP4 modem haalt op deze manier een effectiviteit van ongeveer 120%.

Bij MNP5 wordt gebruik gemaakt van een real time datacompressie en -expansie mechanisme. Hierbij wordt door middel van compressietechnieken de informatie die verzonden moet worden samengeperst. Ontvangen informatie wordt uiteraard weer uitgerekt voordat deze naar de aangesloten computer gaat. Hoe een dergelijke compressie precies in zijn werk gaat, voert me nu te ver. Als voorbeeld kunt u nemen dat het voordeliger is het getal 10, gevolgd door een spatie te sturen in plaats van tien spaties achter elkaar. Iets dergelijks wordt er ook in het modem gedaan. De verhoging van de snelheid van een MNP5 modem is sterk afhankelijk van het type informatie dat overgestuurd wordt. Een tekst kan aanmerkelijk verder samengeperst worden dan code. Een file die al door een programma zoals PKZIP of PKARC gecomprimeerd is zal meestal niet kleiner worden. In een aantal gevallen kan de file zelfs langer worden. Door de bank genomen geeft een MNP5 modem een verhoging van het rendement tot 200%. Dit betekent dat met een V22bis modem met MNP5 een effectieve snelheid van 4800 bps bereikt kan worden.

De MNP klassen lopen momenteel door tot MNP9. Modems met MNP6 en hoger zult u echter nauwelijks tegenkomen. MNP6 is bedoeld voor 9600 bps halfduplex volgens V29 (telefax). MNP7 is een verbetering van MNP5 waardoor bij V22bis maximaal 7200 bps gehaald kan worden. MNP8 is een combinatie van MNP6 en MNP7 en is weer speciaal voor V29 bedoeld.

Het klapstuk van de MNP klassen is MNP9. Deze norm gaat uit van V32 die zonder hulpmiddelen al 9600 bps haalt. Door middel van de datacompressie en -expansie van MNP7 en wat andere trucs is het mogelijk een rendement van 400% te halen. Dit houdt dus een effectieve snelheid van 38,4 kilobit per seconde in. Een dergelijk modem wordt momenteel alleen nog door Microcom (QX/V.32c) zelf op de markt gebracht.

www.dottyflowers.com

Behalve de door de CCITT gedefinieerde V-normen en de door Microcom vastgelegde MNP klassen zijn er uiteraard ook nog andere "standaards" van andere modem-fabrikanten. Een voorbeeld hiervan is de zogenaamde multi carrier techniek waarbij niet één of twee draaggolven in het frequentiespectrum gebruikt worden maar desnoods enkele honderden. De te verzenden data wordt dan parallel aan al deze draaggolven aangeboden. Voor de verdeling van de bits over de draaggolven wordt een afgesproken algorithme gebruikt. Met name het feit dat dit algorithme min of meer vrij gekozen kan worden maakt de methode voor de militairen in ons midden uitermate interessant. Het meest gebruikte voorbeeld van deze techniek wordt aangeduid met PEP (= Packetized Ensemble Protocol). Dit protocol is afkomstig van de firma Telebit, fabrikant van TrailBlazer modems. Dergelijke modems worden voornamelijk in de UNIX-wereld gebruikt. Ook in de datacompressie en -expansie zijn er afwijkende standaards. Zo ondersteund The Ultimate, ons eigen Bulletin Board de HST (High Speed Technology) van modemfabrikant USRobotics.

Het OSI model

Na de kruimels in de voorgaande paragraaf zijn we voorlopig wel uitgeschreven over modems. De rest van deze aflevering gaat over een model waarin datacommunicatie beschreven kan worden. Dit model wordt aangeduid met het OSI model waarbij OSI de afkorting is van Open Systems Interconnection. Dit model is door de ISO (International Organisation for Standardization) ontwikkeld. Het OSI-model is tegenwoordig één van de belangrijkste standaards voor de beschrijving van datacommunicatie. Dit model kent in principe 7 lagen genummerd van 1 t/m 7. Laag 0, het medium, is aan dit model toegevoegd. In de volgende paragrafen worden de diverse lagen stuk voor stuk besproken.

Laag 0: Medium

In het telefoonnet wordt gebruik gemaakt van koperdraad en tegenwoordig soms ook glasvezel. Ook een radiocommunicatie, al dan niet met behulp van een satelliet kan gezien worden als medium. Het medium is de eigenlijke draad tussen de beide systemen.

Laag 1: Physical Layer

De physical layer bevat de onderdelen die nodig zijn om de gegevens die in de computer aanwezig zijn naar buiten te krijgen. Verder bevat de laag de besturingssignalen om de verbindingen tot stand te brengen, te onderhouden en af te breken. Een voorbeeld van de physical layer is de V24 of RS232 interface van uw computer. Met behulp van de signalen op de aansluitingen kan uw computer zijn informatie naar buiten doorgeven en informatie van buiten ontvangen. Of modems tot de Physical Layer of tot het Medium behoren hangt een beetje af van het boek of artikel dat je leest. Voor mijn artikel komt het me het beste uit modems tot de Physical Layer te rekenen.

Laag 2: Datalink Layer

In de datalink layer zitten de onderdelen die het tot stand brengen, onderhouden en het afbreken van de fysische verbindingen (data links) verzorgen. Binnen deze laag wordt besloten of er synchrone of asynchrone communicatie plaatsvindt, of we met een byte georienteerd of met een bit georienteerd protocol werken, full duplex of half duplex etc. Soms is er in de datalink layer ook een protocol aanwezig die er voor zorgt dat de bitstroom die uitgezonden wordt ook precies zo door de ontvanger wordt ontvangen. Mocht dit niet zo zijn, dan zorgt de datalink layer er voor dat dit gedetecteerd wordt en dat het betreffende blok of teken nogmaals overgestuurd wordt.

Zoals de normale modems onderdeel zijn van de physical layer, zijn de in de vorige paragraaf beschreven MNP modems onderdeel van zowel de physical layer als de datalink layer. De onderdelen die zorgen voor de foutdetectie en -correctie en de online datacompressie en expansie is duidelijk een onderdeel van de datalink layer.

Laag 3: Network layer

De network layer zorgt er voor dat de communicatie langs de juiste communicatie-verbindingen loopt. Als we bijvoorbeeld meerdere seriële aansluitingen op de computer hebben die gelijktijdig in gebruik zijn, dan zorgt de network layer er voor de informatie over de juiste communicatie poort gaat. Ook is het mogelijk dat twee computersystemen met elkaar communiceren via een derde computersysteem. In figuur 1 is deze situatie getekend. Het de network layer van het tussenliggende systeem zorgt er dan voor dat de informatie die aan die van het ene systeem ontvangen wordt, doorgestuurd wordt naar het andere systeem en andersom. Deze routering naar de juiste data link wordt door de network layer verzorgt. In figuur 1 wordt de informatie in het middelste systeem dus door de physical en datalink layer doorgestuurd naar de network layer. De network layer weet wat er met de informatie moet gebeuren en stuurt het via de datalink en physical layer door naar het andere systeem. Uiteraard kunnen de communicatie parameters voor de verbinding tussen 1 en 2 verschillen van die tussen 2 en 3. Zo kan de verbinding tussen 1 en 2 bijvoorbeeld een verbinding met behulp van een telefoonlijn en asynchrone V22bis modems zijn terwijl de verbinding tussen 2 en 3 synchroon met glasvezel en gebeurt.

www.dottyflowers.com

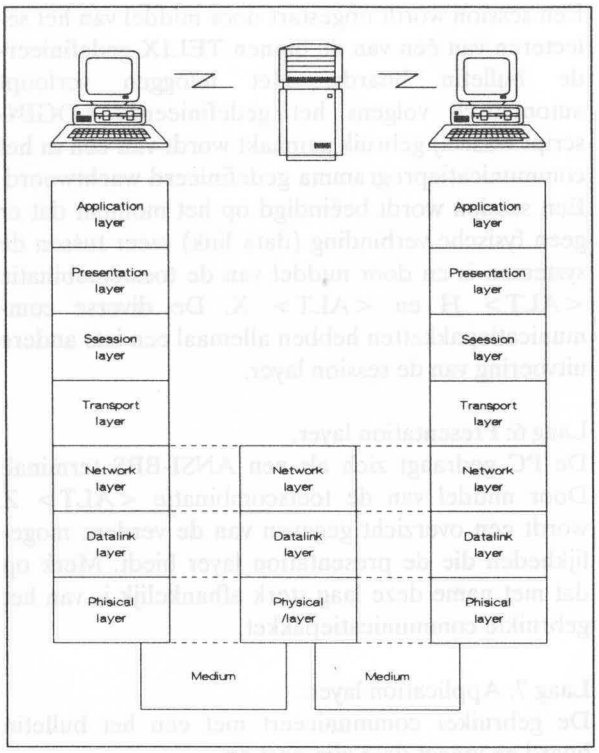


Fig. 1: communicatie via een tussenliggend systeem

Laag 4: Transport layer

De transport layer moet er voor zorgen dat de informatie op de juiste manier en ongestoord overgestuurd wordt. Hierbij maakt de transport layer gebruik van de mogelijkheden die hem in de onderliggende lagen ter beschikking staan. Verder moet de transport layer de informatie ontdaan van alle poes-pas en zonder fouten doorgeven aan de session layer. Een voorbeeld van wat er zoal in de transport layer gebeurt vinden we bij de protocollen voor de overdracht van files. Bij een protocol zoals bijvoorbeeld ZMODEM wordt de informatie opgedeeld in pakketjes. Bij elk pakketje wordt wat extra informatie meegestuurd zodat de andere kant kan zien of het pakketje goed overgekomen is en of de pakketjes in de juiste volgorde binnenkomen. Na ontvangst van een pakketje geeft de ontvanger door middel van een kort berichtje aan de afzender aan of alles correct verlopen is of dat er één of meer pakketjes opnieuw gestuurd moeten worden. Deze afhandeling vindt volledig in de transport layer plaats zodat de hogere lagen hier in het geheel niets van merken.

Laag 5: Session layer

De session layer brengt de verbinding tussen de zogenaamde eindgebruikers tot stand, onderhoudt en verbreekt ze. Dit zijn zogenaamde logische verbindingen. Uiteraard kan er alleen een logische verbinding zijn als er ook een fysieke verbinding is. De periode tussen het opbouwen en verbreken van een verbinding wordt een session genoemd. Als er voor de datacommunicatie betaald moet worden is de duur van een dergelijke session vaak een maatstaf voor de kosten. Een fraai voorbeeld van een dergelijke session is het werken met een bulletin board. Nadat de fysieke verbinding tot stand gebracht is, wordt er op het bulletin board een dergelijke session opgestart. Hierbij wordt als eerste de inlogger geïdentificeerd door middel van zijn naam en wachtwoord. Vervolgens krijgt de gebruiker, afhankelijk van zijn privileges, toegang tot de diverse onderdelen van het bulletin board systeem. Nadat de gebruiker kenbaar gemaakt heeft dat hij er mee op wil houden, worden zowel de logische als de fysieke verbinding verbroken. Wel wordt er door de session layer nog een aantal gegevens vastgelegd zodat de beheerder na kan hoeveel van de aanwezige faciliteiten gebruik gemaakt is.

Laag 6: Presentation layer

De taak van de presentation layer is de informatie die binnenkomt zodanig om te vormen dat de volgende laag er goed mee uit de voeten kan en anderzijds de informatie die afkomstig is van laag 7 om te zetten in informatie die door laag 5 verder getransporteerd kan worden. Bij iemand die met behulp van zijn computer informatie uit een database (bijvoorbeeld VIDITEL) opvraagt is het dialoogprogramma dat voor de opbouw van de schermen zorgt dus onderdeel van de presentation layer. De presentation layer verzorgt de presentatie van de overgestuurde gegevens. Als er gegevens de andere kant uitgestuurd worden, dan zorgt de presentation layer er voor dat er geen onnodige (layout) informatie meer aanwezig is.

Laag 7: Application layer

In de application layer zijn de uiteindelijke oorzaken en gebruikers van de informatie ondergebracht. Dit zijn de entiteiten (terminals, programma's, operators, printers etc., kortom "dingen") die iets met de informatie doen. Als u ingelogd bent bij een bulletin board en u bekijkt een overzicht van de aanwezige files, dan bent u zelf onderdeel van de application layer, u bent namelijk het ding (de entiteit) die iets met de informatie doet. Anderzijds kan het ook zijn dat er ergens een programma is dat de temperatuur in uw huis regelt. Opnemers bepalen de temperatuur en geven deze via datacommunicatie door aan het regelprogramma. Het regelprogramma genereert, op basis van onder andere de temperatuur, weer stuurinformatie voor uw verwarming. Het regelprogramma is dus onderdeel van de application layer omdat die iets met de temperatuurinformatie doet en de bron van de stuurinformatie is.

Samenvatting

In de voorgaande paragrafen is een manier gedefinieerd waarmee datacommunicatie op een éénduidige

www.dottyflowers.com

en consistente manier beschreven kan worden. Met behulp van het OSI model kan men een eenvoudige vorm van datacommunicatie tussen twee computers met behulp van een nulmodem kabel en seriële poorten beschrijven en zeer gecompliceerde Local Area en Wide Area Networks. Ter afsluiting een beschrijving van de situatie van iemand die communicatie heeft met een bulletin board en daar bijvoorbeeld zijn post bekijkt met het communicatiepakket Telix.

Laag 0: Medium.
Het medium bestaat uit het interlokale telefoonnet.

Laag 1: Physical layer.
Er wordt gebruik gemaakt van een V22 modem die middels een RS232 verbinding aangesloten is aan COM-poort 1 van een PC.

Laag 2: Datalink layer.
De communicatie geschiedt byte-georienteerd. De communicatie is asynchroon en full duplex met een snelheid van 1200 bps. Een frame bestaat uit één startbit, acht databits en één stopbit. Er wordt geen gebruik gemaakt van een pariteitbit.

Laag 3: Network layer.
De communicatie verloopt via COM-poort 1.

Laag 4: Transport layer.
Er wordt in de beschreven (connect) mode geen gebruik gemaakt van verdere protocollen. De informatie wordt zonder verdere behandeling in de transport layer doorgegeven. Bij filetransfer wordt standaard gebruik gemaakt van het protocol ZMODEM.

Laag 5: Session layer.
Een session wordt opgestart door middel van het selecteren van één van de binnen TELIX gedefinieerde bulletin boards. Het inloggen verloopt automatisch volgens het gedefinieerde LOGIN-script waarbij gebruik gemaakt wordt van een in het communicatieprogramma gedefinieerd wachtwoord. Een session wordt beëindigd op het moment dat er geen fysische verbinding (data link) meer tussen de systemen is en door middel van de toestcombinatie <ALT> H en <ALT> X. De diverse communicatiepakketten hebben allemaal een iets andere uitvoering van de session layer.

Laag 6: Presentation layer.
De PC gedraagt zich als een ANSI-BBS terminal. Door middel van de toetscombinatie <ALT> Z wordt een overzicht gegeven van de verdere mogelijkheden die de presentation layer biedt. Merk op dat met name deze laag sterk afhankelijk is van het gebruikte communicatiepakket.

Laag 7: Application layer.
De gebruiker communiceert met een het bulletin board en vraagt daar zijn post op.

Literatuur.

Dit artikel is gebaseerd op onder andere de volgende informatie:
1: Markus Aigner: Geschwindigheid ist keine Hexerei; mc 10/90.
2: Markus Aigner: Turbolader für High-Speed-Modems; mc 10/90.
3: P.C. den Heijer/R. Tolsma: Datacommunicatie (Kluwer PC boeken).
4: John H. Humphrey and Gary S. Smock: High-Speed Modems; Byte june 1988.
5: J. Voorhaar: To Share Or Not To Share, That's The Question (testrapport Telix) µP Kenner 66.
6: PBNA Poly Automatiserings Zakboekje.

(Advertentie)

720 kB floppy drive als drive D:?
Komt uw PC/XT niet uit de startblokken?
Alweer een compatibiliteitsprobleempje?

Koop eens een nieuw BIOS voor uw machine! Dat scheelt slapeloze nachten. Het KGN-BIOS dus. Speciaal en alleen voor leden.

Bestellen:

Maak *f* 25,00 over aan de KIM Gebruikersclub Nederland in Enschede, Postbank rekening no. 3757649 o.v.v. "KIM XT-BIOS".
U krijgt de EPROM dan zo snel mogelijk thuisgestuurd.

Drukker : hier graag de foto 'EPROMs'

www.dottyflowers.com

Van de bestuurstafel

Hijg!!! Ik heb de laatste week diep respect gekregen voor Joost Voorhaar. Zo'n blad opmaken is meer werk dan ik in eerste instantie dacht. Joost deed dat schijnbaar 'even' in een weekend, ik heb nu nog heel wat meer tijd nodig. Nu is mijn baas failliet gegaan, dus die tijd is er gelukkig wel. Joost is begonnen mij eerst maar een opstartcursus Ventura en elementaire opmaaktechnieken te geven, compleet met allerlei hints en aanwijzingen. In het begin ging het nogal stroef en langzaam, maar al werkend ontstaat er wat meer handigheid. Ik hoop dat het resultaat u tot dusver niet is tegengevallen. Uiteindelijk is het best leuk werk, dat maar een nadeel kent: als het blad in de bus valt, zitten er voor mij bijna geen verrassingen meer in.

Op de laatste bestuursvergadering zijn er weer wat zaakjes voorgevallen die de moeite van het vermelden waard zijn. Het eerste was, dat Mick Agterberg zich genoodzaakt ziet om het secretarisschap zal opgeven. Dit vanwege een opleiding tot rij-instructeur, die vrijwel al zijn vrije tijd gaat opeisen. Die opleiding gaat tot volgend voorjaar duren. Mick wil wel graag bestuurslid blijven, en daar is voldaan. De secretaristaken zijn inmiddels onder de andere bestuursleden verdeeld. We wensen Mick natuurlijk sterkte met zijn nieuwe initiatief.

De penningmeester kon trots melden, dat de club thans ongeveer 160 leden telt, en dat er meer dan aan het einde van vorig jaar: we groeien weer een beetje! Jacques had nog meer hoopvolle berichten: ondanks alle extra activiteiten (HCC-dagen, uitbreiding bulletinboard) teren we niet in op onze financiële reserves. Dit laatste is onder andere te danken aan de verkoop van de club-XT's, die een aardig zakcentje voor de clubkas opleveren. De XT's zijn nog niet op: dus als u een kennis heeft die een goedkope computer zocht om eens voorzichtig mee te beginnen: sla uw slag nu het nog kan. Al met al gaat het prima met de club, en zo hoort het ook.

Van MINIX-goeroe Fred van Kempen werd vernomen, dat de MINIX-gebruikers dringend zitten te springen om wat betere hardware, speciaal toegesneden op de eisen die MINIX aan de hardware stelt. Bovenaan het verlanglijstje staat een Motorola processor. Verder behoorlijk wat RAM, een boot-ROM en tenminste 1 seriële poort. Op de vergadering is er behoorlijk gebrainstormd hoe die hardware eruit moet gaan zien. Het huidige idee is als volgt:

Ontwikkel een kaart die op de PC/AT-bus past. Op die kaart komen dan een 68010 processor op 16 MHz, een 680x0 boot-ROM, tenminste een seriële poort, en zoveel RAM als zinvol en/of waarvoor ruimte is. Waarom een PC/AT-bus? Welnu, vanwege de kosten. De bijkomende hardware kan in zo'n geval namelijk heel goedkoop worden aangeschaft, omdat er inmiddels ook losse buskaarten voor de AT-bus zijn. Voedingen, kasten, hard- en floppy drives en meer van dergelijke zaken kunnen gewoon bij een klonen-verkoper voor een krats worden gekocht. Wordt er geen grote performance gevraagd, dan kunnen zelfs de diskcontrollers, seriële en parallelle poorten uit de klonenwereld benut worden. Een ander voordeel is ook nog, dat bestaande bezitters van een PC/AT zo kunnen instappen: in een AT mag een andere processor de bus overnemen. Iedere AT kan dus als basis dienen voor zo'n 68000-MINIX systeem.

Geert Stappers is inmiddels al druk doende allerlei mensen te contacteren die aan het project zouden kunnen meedenken en mee-ontwerpen. Heeft u plotseling ook ideeen gekregen, bel Geert dan eens.

Het bestuur is thans druk doende om de voorbereidingen voor de HCC-dagen te treffen. Als u soms een paar uurtjes in de kraam wilt staan kan dat natuurlijk. Hoe meer zielen, hoe meer vreugd. Overigens is het door een samenloop van omstandigheden zo, dat de derde zaterdag in november dit jaar heel vroeg valt, en dat de HCC-dagen juist heel laat gehouden worden. De bijeenkomst in Almelo hoefde daarom niet verschoven te worden. Er zit zelfs een week tussen die beide evenementen!

Hopelijk tot ziens in Almelo, of anders op een van de HCC-dagen.

Uw veurzitter,

Nico de Vries

www.dottyflowers.com

Informatie.

De μP Kenner (De microprocessor Kenner) is een uitgave van de KIM gebruikersclub Nederland. Deze vereniging is volledig onafhankelijk, is statutair opgericht op 22 juni 1978 en ingeschreven bij de Kamer van Koophandel en Fabrieken voor Hollands Noorderkwartier te Alkmaar, onder nummer 634305. Het gironummer van de vereniging is 3757649.

De doelstellingen van de vereniging zijn sinds 1 januari 1989 als volgt geformuleerd:

- Het vergaren en verspreiden van kennis over componenten van microcomputers, de microcomputers zelf en de bijbehorende systeemsoftware.
- Het stimuleren en ondersteunen van het gebruik van micro-computers in de meer technische toepassingen.

Om deze doelstellingen zo goed mogelijk in te vullen, wordt onder andere 5 maal per jaar de μP Kenner uitgegeven. Verder worden er door het bestuur per jaar 5 landelijke bijeenkomsten georganiseerd, beheert het bestuur een Bulletin Board en wordt er een softwarebibliotheek en een technisch forum voor diverse systemen in stand gehouden.

Landelijke bijeenkomsten:

Deze worden gehouden op bij voorkeur de derde zaterdag van de maanden januari, maart, mei, september en november. De exacte plaats en datum worden steeds in de μP Kenner bekend gemaakt in de rubriek Uitnodiging.

Bulletin Board:

Voor het uitwisselen van mededelingen, het stellen en beantwoorden van vragen en de verspreiding van software wordt er door de vereniging een Bulletin Board beschikbaar gesteld.

Het telefoonnummer is: 053-303902.

Software Bibliotheek en Technisch Forum:

Voor het beheer van de Software Bibliotheek en technischeondersteuning streeft het bestuur ernaar zgn. systeemcoordinators te benoemen. Van tijd tot tijd zal in de μP Kenner een overzicht gepubliceerd worden. Dit overzicht staat ook op het Bulletin Board.

Het Bestuur:

Het bestuur van de vereniging wordt gevormd door een dagelijks bestuur bestaande uit een voorzitter, een secretaris en een penningmeester en een viertal gewone leden.

Nico de Vries (voorzitter)
Mari Andriessenrade 49
2907 MA Capelle a/d IJssel
Telefoon 010-4517154

Jacques H.G.M. Banser (penningmeester)
Haaksbergerstraat 199
7513 EM Enschede
Telefoon 053-324137

Gert van Opbroek (Redactie μP Kenner)
Bateweg 60
2481 AN Woubrugge
Telefoon 01729-8636

Mick Agterberg
Davidvosstraat 29
1063 HV Amsterdam
Telefoon 020-131538

Jan D.J. Derksen
Ed Verkadestraat 9-1
7558 TH Hengelo
Telefoon 074-770970

Geert Stappers
Engelseweg 7
5825 BT Overloon
Telefoon 04788-1279

Ton Smits
De Meren 39
4731 WB Oudenbosch

Ereleden:

Naast het bestuur zijn er een aantal ereleden, die zich in het verleden bijzonder verdienstelijk voor de club hebben gemaakt:

Erevoorzitter:
Siep de Vries

Ereleden:
Mevr. H. de Vries-van der Winden
Anton Müller

www.dottyflowers.com

www.dottyflowers.com