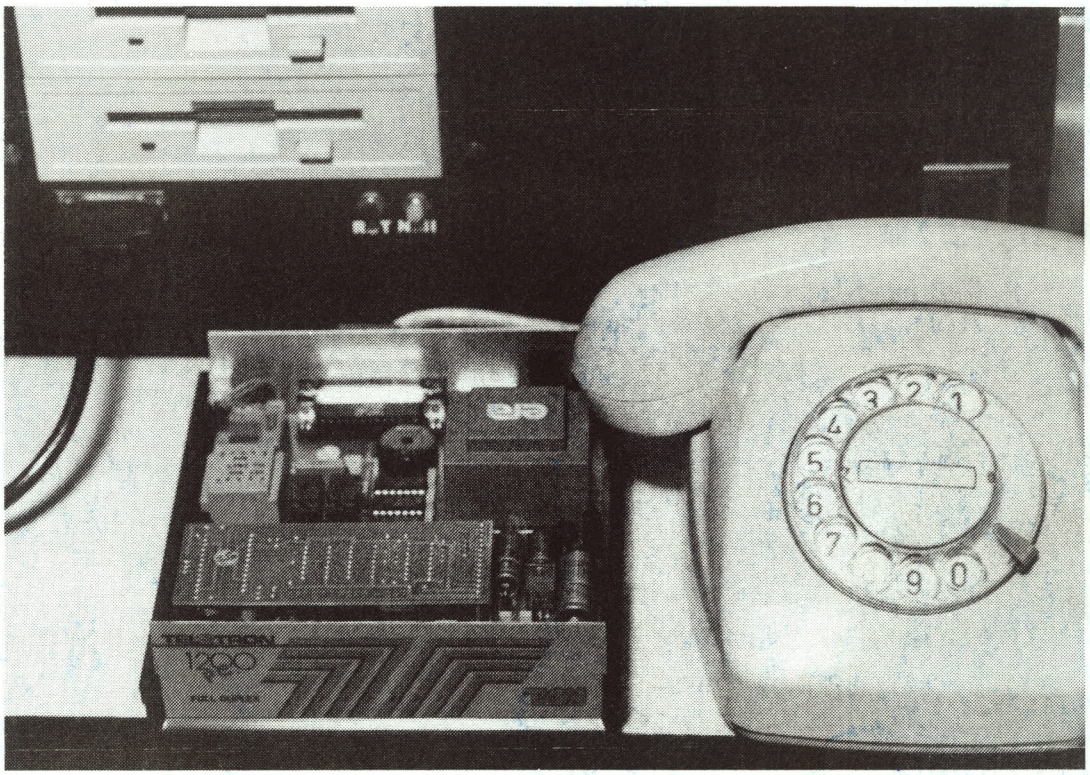
De µP Kenner

Vijftiende jaargang nr. 3
Augustus 1991
72



In dit nummer o.a.:

Datacom deel 8
Versienummers
KGN-68k project
Listige inpakhulpen
Nieuw PC/XT KGN BIOS

Inhoudsopgave

De μP Kenner

Nummer 72, augustus 1991
Verschijnt 6 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 73 verschijnt op 19 oktober 1991.

Kopijsluitingsdatum voor nummer 73 is vastgesteld op 5 oktober 1991.

Algemeen

Vereniging

Talen/Software

Systemen

Hardware

Datacommunicatie

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 op 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.

Redactioneel

Hallo allemaal. Het was met het weertje wel deze zomer... Dat is ook de reden dat μP Kenner 72 van augustus 1991 (veel) te laat uitkomt. Mijn excuses daarvoor en gelukkig valt de bijeenkomst in september zo laat dat u de uitnodiging zeker nog op tijd zult hebben.

In de vorige aflevering heb ik min of meer beloofd dat de volgende aflevering een special over Minix, Unix en KGN-68k zou zijn. In overleg met bestuur en werkgroep hebben we echter besloten deze special pas uit te brengen als we echt iets over KGN-68k kunnen vertellen. Tot nu toe is het hele systeem nog steeds een papieren tijger omdat we net gisteren het definitieve (?) schema hebben vastgesteld. We beginnen nu aan het printontwerp zodat u nog even geduld moet hebben voordat we het systeem kant en klaar hebben. Over de status van het project kunt u uiteraard weer lezen in de bijdrage van de werkgroep. Eén en ander houdt niet in dat de special niet zal verschijnen. Op het moment dat er een werkende KGN-68k gepresenteerd kan worden, zal deze presentatie gepaard gaan met een special over dit board en de bijbehorende software. Kortom, wat in het vat zit verzuurt niet.

Een tweede zaak die ons wat tegen gezeten heeft is het feit dat we gebrek aan kopij beginnen te krijgen. Ik begrijp wel dat de zomermaanden nu niet uitnodigen tot het schrijven van artikelen en/of software. Ik probeer daarom altijd een klein buffertje op te bouwen voor het augustusnummer. Welnu, dat buffertje is nu dus weer leeg en voor de volgende μP Kenner kunnen we wel weer wat bijdragen gebruiken.

Tja, en wat is er zoal loos in de club? Eigenlijk een heleboel. De werkgroep KGN-68k heeft enkele inzinkingen meegemaakt maar is nu toch weer vol goede moed bezig. Zo vol goede moed zelfs dat het hele project een beetje in een stroomversnelling lijkt te komen. Verder willen we deze winter ook fanatiek bezig met DOS-65 3.0. Er komen enkele ideeën binnen over het hoe en het wat en vanuit het bestuur heeft Nico aangegeven dat hij daar de komende maanden, samen met anderen, uitgebreid mee bezig wil gaan.

Ook heeft het bestuur besloten weer deel te nemen aan de HCC-dagen in november. Wat we daar precies gaan doen hangt sterk af van de status van het KGN-68k project. Zeker is in ieder geval dat we uitgebreid aandacht aan Minix zullen besteden omdat dat toch zo'n beetje ons "eigen" systeem in de komende jaren zou moeten gaan worden. Verder zal DOS-65 natuurlijk ook niet ontbreken. Hebt u nog ideeën voor een presentatie op de HCC-dagen, laat ze ons dan weten. De stand moet tenslotte de club vertegenwoordigen en in de club gebeurt wat de leden willen!

Tenslotte hebben we in november een ledenvergadering. Dan zullen er zeker enkele verschuivingen in het bestuur gaan plaatsvinden. Nico heeft namelijk aangegeven dat hij het bestuur wil gaan verlaten. Verder wil ondergetekende stoppen met de redactie. Het is namelijk haast niet meer te combineren met mijn werk en met de taken die ik als goed echtgenoot en vader heb en verder vind ik ook dat er zo nu en dan wat vers bloed nodig is. Ik ben niet direct van plan het bestuur te verlaten en ben ook zeker voornemens een regelmatige correspondent van de μP Kenner te blijven. Hoewel er binnen het bestuur al wel over een opvolger gesproken is, staan we uiteraard ook open voor suggesties van de leden. Als u beschikt over een vlotte pen (PC met WordPerfect mag uiteraard ook), overweeg dan eens de redactie van de μP Kenner te gaan doen. Binnen de club heeft de redactie een onafhankelijke status en is alleen achteraf verantwoording aan het bestuur en de leden verschuldigd. Een redacteur heeft dus een grote (pers-)vrijheid en mag zelf bepalen hoe het blad er uitziet en wat de inhoud van het blad is.

Hoewel het blad wat aan de late kant is, denk ik toch dat de inhoud weer veel mensen aan zal spreken. We hebben met een aantal mensen toch weer getracht er een nummer met een behoorlijk niveau van te maken en met een inhoud "Voor Elck Wat Wilsch". Ik hoop dat u, als lezer, ook die mening bent toegedaan en wens u uiteraard weer veel plezier met uw computerhobby toe en hopelijk tot ziens in Haarlem op 21 september.

Gert van Opbroek

Uitnodiging voor de clubbijeenkomst

Datum: 21 september 1991
Locatie: Wijkcentrum "De Ringvaart"
Floris van Adrichemlaan 98
2035 VD Haarlem
Telefoon: 023-363856
Entree: Fl. 10,--

Routebeschrijving

AUTO:

Komende uit de richting Utrecht, Amersfoort of Rotterdam:
Afslag Haarlem-Zuid; tweede stoplicht links; bij de tweede kruising met stoplichten links; Floris van Adrichemlaan.

Komende uit de richting Alkmaar:
afslag Haarlem-Zuid; verder zie boven.

OPENBAAR VERVOER:

Vanaf het station Haarlem met buslijn 7, 71, 72 of 77; halte Floris van Adrichemlaan.

Programma:

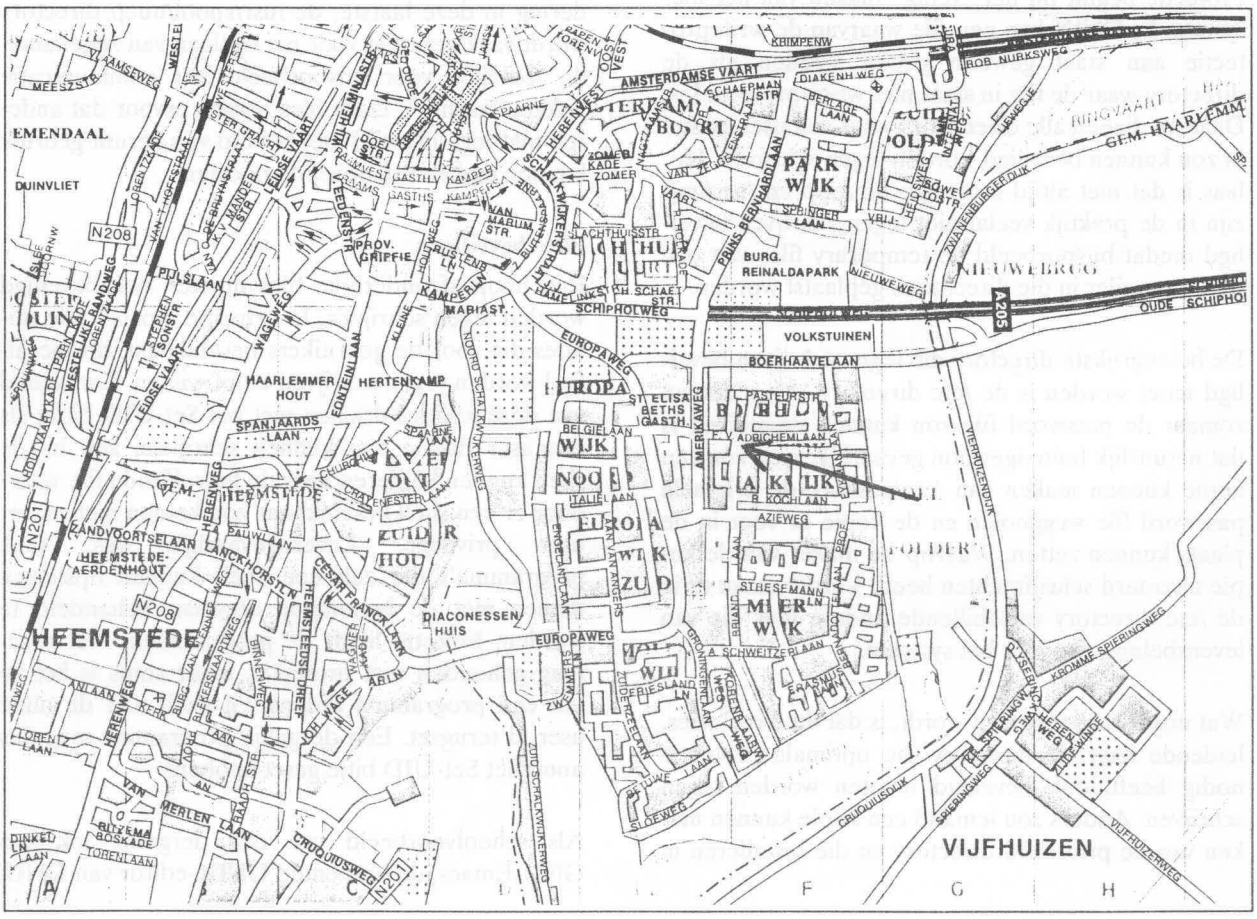
9:30 Zaal open met koffie
10:15 Opening door de voorzitter Nico de Vries.
10:30 Voordracht door Ruud Uphoff:
"Kwaliteitsbeheersing in systeemsoftware"
11:30 Forum en markt
12:00 Lunchpauze, Koffie en broodjes op eigen kosten te verkrijgen.

Aansluitend het informele gedeelte bedoeld om kennis, ervaring, Public Domain alsmede eigen ontwikkelde software uit te wisselen met uw medeleden. **BRENG DAAROM OOK UW EIGEN SYSTEEM MEE!** (En vergeet de snoeren niet...). Gedurende dit informele gedeelte zullen er enkele Minix-systemen aanwezig zijn waarop de mogelijkheden van dit operating systeem getoond zullen worden. Ik heb horen fluisteren dat er zelfs een netwerk van Minix-systemen opgebouwd zal gaan worden.

17:00 Sluiting

Let op

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.



Het standaard operating system UNIX (deel 5)

Langzamerhand zijn we aangeland bij het echte werk: systeembeheer op een UNIX systeem. Het beheren van een UNIX systeem is vrij simpel zolang er slechts één persoon op werkt zoals op de moderne (grafisch georiënteerde) workstations. Beheren we een systeem waarop meerdere users, al dan niet tegelijkertijd, kunnen inloggen, dan krijgen we te maken met uitgebreide beveiligingssystemen en protectie mogelijkheden. Als het UNIX systeem gevoelige informatie bevat en dan ook nog eens voorzien is van de nodige modem aansluitingen, dan is het veilig maken en houden van het systeem zoniet de belangrijkste dan toch wel één der belangrijkste taken van de systeembeheerder.

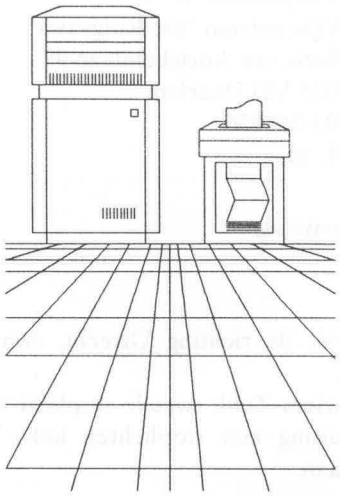
Naast deze veiligheidsproblemen is het voor de sysop ook belangrijk dat hij een werkbare omgeving schept voor zichzelf. Op een eenvoudig systeem is het nog wel te doen om "even snel" een user toe te voegen in het systeem; naarmate het systeem complexer wordt zal deze taak ingewikkelder worden totdat het systeem dermate ingewikkeld in elkaar steekt dat alleen met behulp van flexibele en "intelligente" shellscripts of gespecialiseerde programmatuur deze taak nog op verantwoorde wijze uitgevoerd kan worden.

Directory protectie

Protectie begint bij het "veilig" maken van het filesystem. In UNIX kan een file waarvan de writeprotectie aan staat gewoon gewist worden als de directory waar de file in staat niet write-protected is. Daarom dienen alle directories waar een user kwaad in zou kunnen beveiligd worden tegen schrijven. Helaas is dat niet altijd mogelijk; de /tmp en /usr/tmp zijn in de praktijk veelal niet tegen schrijven beveiligd omdat bijvoorbeeld de temporary files van o.a. de C-compiler in die directories geplaatst worden.

De belangrijkste directory die tegen schrijven beveiligd moet worden is de /etc directory. Als iedereen zomaar de password file zou kunnen weggooien is dat natuurlijk buitengewoon gevaarlijk; men zou een kopie kunnen maken van /etc/passwd, de originele password file weggooien en de kopie er voor in de plaats kunnen zetten. Waarop de maker van de kopie uiteraard schrijfrechten heeft. Verder staan er in de /etc directory verschillende andere files die van levensbelang zijn voor het systeem.

Wat nogal eens vergeten wordt, is dat de directories, leidende naar de directory die optimale protectie nodig heeft, ook beveiligd moeten worden tegen schrijven. Anders zou iemand een kopie kunnen maken van de protected directory en die installeren in



de plaats van de originele. Het path na zo'n directory begint in de root; de root directory moet dan ook beveiligd worden tegen schrijven. Hoe logisch dit ook klinkt, in veel systemen heeft "men" dit over het hoofd gezien. Een kopie van de /etc directory is dan zo gemaakt. De /etc directory renamen, en de kopie (waarop de hacker schrijfrechten heeft!) kan dan in de plaats van /etc gezet worden.

Andere belangrijke directories in de standaard UNIX-structuur zijn o.a. /dev, /bin, /usr, /usr/bin en vrijwel de volledige /usr/spool tak. Er is één uitzondering in deze laatste; de /usr/spool/uucp directory wordt vaak gebruikt voor het opslaan van zogenaamde lockfiles voor bijvoorbeeld het communicatie pakket "kermit". Deze files zorgen ervoor dat anderen niet tegelijkertijd met behulp van kermit gebruik kunnen maken van dezelfde poorten.

Files beveiligen

Een hoop afzonderlijke files moeten ook beveiligd worden tegen schrijven. In principe moeten alle utilities die voor de gebruikers beschikbaar zijn beveiligd worden tegen schrijven en/of wissen. Een aantal van deze utilities draaien met het Set-UID bitje; als een user dáárin zou kunnen schrijven, zou hij de shell kunnen kopieren naar de betreffende file waardoor er een shell beschikbaar zou komen met superuser privilege. Levensgevaarlijk dus. Ook programma's die zelf een shell kunnen opstarten dienen met de benodigde argwaan behandeld te worden. Meestal heeft de programmeur wel rekening gehouden met protectie, maar soms is het zo dat een programma tijdens z'n shell niet de oude user id terugzet. Een dergelijk programma mag dus **nooit** het Set-UID bitje geset hebben!

Als schoolvoorbeeld van een dergelijk lek kan GNU-Emacs, de beroemde UNIX-editor van GNU,

genoemd worden. Met behulp van GNU-Emacs kan niet alleen aan tekstverwerking gedaan worden, er kan in de uitgebreide versie onder andere ook post vandaan gestuurd worden. In een bepaalde versie van GNU-Emacs was het mogelijk een postbestand naar de directory van een ander te sturen. Helaas, ook naar beschermde directories! In het boek "het koekoeksei" van Clifford Stoll wordt beschreven hoe een kraker hiervan gebruik maakt door de clockdeamon aan te passen. De clockdeamon (cron) is een programma dat resident in het geheugen van een UNIX machine zit. Eéns in de zoveel tijd start het een ander programma op, genaamd atrun. De functie van atrun is, alle jobs die van te voren met behulp van het at-commando klaargezet zijn op de juiste tijdstippen te laten uitvoeren. Atrun draait met volledig system privilege, en dus zou het mogelijk zijn atrun een shell te laten opstarten met volledige systeem-privileges. Het gat in GNU-Emacs doet denken aan een koekoek die zijn eieren laat uitbroeden door andere vogels in andere nesten. Vandaar ook de titel van het boek, maar dat terzijde.

Het gat in GNU-Emacs doet denken aan een koekoek die zijn eieren laat uitbroeden door andere vogels in andere nesten.

Standaard utilities die in Set-UID mode moeten lopen zijn o.a. mail en passwd. Veelal is login ook zo'n proces. In het algemeen geldt dat er geen programmatuur moet zijn waarvan root de eigenaar is en die door iedereen geschreven mag worden.

Devices

Helaas, ook in mijn systeem zaten ernstige security-lekken. Zo waren de devices voor een deel leesbaar door iedereen. Zo ook de harddisk. Een user had met behulp van een "raw" leesactie op het device en de nodige kennis van i-nodes en dat soort zaken alles kunnen lezen wat'ie maar wilde. Public write-access op een raw device is natuurlijk helemáál uit den boze. Dit betekent wel dat bepaalde utilities die raw access moeten hebben op een beveiligd device (bijv. df) in SUID of SGID mode moeten lopen.

In principe zou er ook geen read-access moeten zijn op het geheugen van de computer. Als een user kan lezen van /dev/mem of van /dev/kmem (respectievelijk user- en kernel memory!), dan zou hij toegang kunnen krijgen tot informatie die hij helemaal niet màg zien. Het opvragen van process-status (ps) is dan ook alweer zo'n taak die SUID mode benodigd.

Ook terminals behoren tot een kwetsbare groep. Bekend is natuurlijk de truuk met "het paard van Troje"; een proces wordt achtergelaten op een terminal dat een login: simuleert. Een user die op die terminal probeert in te loggen typt zijn naam en password in, die door het programma weggeschreven worden in een file. Het programma zegt nog snel even dat de user een foutieve id of password heeft ingevoerd en verdwijnt, de terminal in het echte login-proces achterlatend. De user zal denken zijn password verkeerd ingevoerd te hebben en probeert het nogmaals; nu met (hopelijk) meer succes.

Dit soort dingen zijn verschrikkelijk moeilijk te voorkomen. Een algemene remedie tegen dit soort truuks is niet in een artikeltje te geven; het verschilt per systeem. Een tip voor superusers: de eerste maal express een foutief password invullen bij de login prompt zal heel wat hackers tot wanhoop drijven...

Een laatste opmerking met betrekking tot devices: floppy disks kunnen razend gevaarlijk zijn! Een normale user mag onder geen beding toegang hebben tot de mount- en unmount commando's. Het zou erg eenvoudig zijn een flop elders te prepareren door er een shell op te zetten waarvan user 0 (de root dus!) de eigenaar is en waarvan het SUID bitje staat. Als een user een dergelijke flop zou kunnen mounten, dan heeft'ie dus een shell ter beschikking met SUID status! Sommige oude systemen lieten het ook nog eens toe directories te mounten waar de user niet voldoende access op had, bijvoorbeeld op de /etc of /bin directory. In het algemeen: mounten en unmounten mag **alleen** door de root.

Gebruikers

De meeste UNIX systemen hadden best veilig kunnen zijn, zolang er maar geen gebruikers rondliepen. Gebruikers hebben de nare eigenschap niet geïnteresseerd te zijn in veiligheidsmaatregelen. Een login prompt én een password... dat is voor een hoop gebruikers al een vervelende drempel om van het systeem gebruik te maken. Het liefst zetten ze hun terminal aan om dan onmiddelijk in de applicatie die ze nodig hadden aan de gang te kunnen. Passwords onthouden? Nou ja, je kunt natuurlijk je eigen naam gebruiken (al dan niet omgekeerd), of de naam van je hondje. Of die van je vrouw of maïtresse... Sommige systemen genereren zelf eens in de xx maanden een password en delen de gebruiker mee dat hij een nieuw password krijgt. Combinaties die niet uit te spreken zijn, zijn ook moeilijk te onthouden. Men schrijft ze op de onderkant van keyboards of op een werkblad op het bureau. De enige remedie tegen dit soort zaken, is de gebruikers te doordrin-

gen van de veiligheids maatregelen voor ze toegang tot het systeem krijgen.

Het gebruik van superuser access moet zo veel mogelijk voorkomen worden. Superuser access levert dusdanige kracht op, dat een "slip of the finger" al fataal kan zijn. Grotere systemen worden meestal niet door één persoon maar door meerdere personen onderhouden. Laten we deze groep personen even aanduiden als de "system staff". Leden van de system staff hoeven niet per definitie superuser access te hebben als ze bezig gaan met het installeren van nieuwe utilities. Als de /bin en /usr/bin directories beide read/write zijn voor de groep "bin", dan kunnen deze gebruikers met behulp van het newgrp commando write access verkrijgen op deze directories. Hetzelfde gaat op voor user-maintenance als de /etc/passwd file read/write is voor de bin-groep.

Hoe belangrijk ook; veel systemen draaien zonder dat er redelijke backups van bijgehouden worden. UNIX heeft voor het backuppen verschillende commando's die "raw" naar een device kunnen schrijven (bijvoorbeeld de tape-archiver "tar"). Van een goed beheerd systeem zijn er altijd minimaal twee backups: één van de meest recente backup-sessie en één van de sessie daarvoor. De oudste van de twee wordt steeds gebruikt om de nieuwe backup op te draaien. Mocht er dan iets fout gaan tijdens het backuppen, dan is er nog geen man overboord. Backups dienen uiteraard zodanig weggestopt te worden dat niemand er zo maar bij kan...

Booten en afsluiten

Een UNIX systeem wordt in het algemeen zo weinig mogelijk uit gezet. Het kan af en toe echter nodig zijn! Als we het gemiddelde UNIX systeem zo maar zondermeer uitzetten, dan ontstaan er de nodige problemen. Users die nog ingelogd zijn zien hun werk richting energiecentrale verdwijnen en opeens valt hun terminal stil. In een beetje multitasking systeem wordt uit perfomance overwegingen veel gebufferd. Als die buffers niet geflushed worden ontstaat er dus een incongruentie tussen filesystem en de physieke disks; een "inconsistent filesystem" is het gevolg. In mijn systeem heb ik een speciaal shutdown-script dat ruim van tevoren messages verstuurt aan users alvorens ze uit te loggen en hun processen te killen. De buffering wordt er in uitgezet en de disks worden gesynchroniseerd met het filesystem. Eventuele gemounte disks worden nog snel even ontkoppeld van het systeem en de terminals gaan allemaal down. De juiste volgorde van dit proces is:

- Het aanmaken van de /etc/nologin file (als die er is, kan niemand meer inloggen)
- Het versturen van messages naar de users op 10, 5 en 2 minuten vóór het systeem ze uitlogt (zodat de user dat zelf op tijd kan doen)
- Eventueel nog ingelogde users automatisch uitloggen met een message dat ze er door het systeem uitgegooid zijn ("*** Forced logout from system")
- Alle user processen afkappen
- Background tasks stoppen
- Uitzetten van buffering en buffers flushen
- Unmounten van devices
- Resterende background tasks stoppen
- Synchroniseren van filesystem

De terminals worden ontkoppeld zodra er geen user meer op ingelogd is.

Booten is meestal een tijdvretende bezigheid in een UNIX systeem. Afgezien van het werkelijke bootproces worden de volgende taken (meestal in de /etc/rc task) uitgevoerd:

- Zet de path environment vars
- Leeg de accounting files met mounted filesystems (/etc/mtab) en ingelogde gebruikers (/etc/utmp)
- Mount de verschillende directories indien nodig
- Leeg de temporary directories (/tmp, /usr/tmpen /usr/spool/lpd/tmp)
- Unlock printerspooler
- Start shell-accounting indien nodig
- Start (disk-) buffers en clockdeamon (/etc/update en /etc/cron)
- Unlock de terminals
- Unlock de inlog procedures (/etc/nologin)

Vaak wordt ook nog het complete filesystem gecheckt in het begin van /etc/rc met behulp van het fsck commando.

Meerdere terminals

Een systeem kan pas echt multiuser genoemd worden als het meerdere terminals ondersteunt. De manier waarop de terminal configuratie in elkaar steeks kan per systeem verschillen. Daar binnen de KGN het MINIX operating system gebruikt wordt, zal ik uitgaan van een UNIX V7-systeem, waarmee MINIX compatible is. Hier en daar kan onderstaande afwijken van die V7-standaard ter wille van de MINIX-systemen in de club.

Allereerst de files die we nodig hebben om de configuratie aan het systeem duidelijk te maken. Dat zijn (allemaal in de /etc directory): ttys, ttytype en termcap.

We beginnen bij de file "ttys". Hierin staat voor iedere terminal een regeltje. Iedere regel begin met

een 0, 1 of 2. Staat er een 0, dan is de terminal "disabled" en wordt door het systeem genegeerd. Een 1 duidt aan dat er een standaard login-proces op opgestart moet worden. Een 2 duidt aan dat er een getty proces voor die betreffende terminal gedraaid moet worden. Getty is een proces dat de terminal lijn in de gaten houdt en de terminal parameters op de juiste manier instelt. Voor een direct-wired terminal is dit niet nodig; er kan direct een login op gestart worden. Het tweede character geeft de baudrate van het device aan. Voor terminals waar een getty opgestart wordt heeft deze letter in beginsel geen betekenis. Tenslotte volgt er een specificatie van de terminal line, in MINIX het cijfer van de device name (dus een 0 voor console c.q. tty0, een 1 voor tty1 etc.). In "standaard" UNIX (bestaat dat?) heeft de ttys file een iets andere betekenis. Het eerste character geeft informatie over het proces (0 = geen proces, 1 = actief). Het tweede character geeft óf een baudrate, óf een letter welke aangeeft welk proces er opgestart moet worden (bijvoorbeeld /etc/getty, maar ook dialers of andere processen kunnen opgestart worden!). Daarna volgt dan de volledige naam van het device. Terug naar MINIX. Bij mij draait er naast de console nog een VT101-terminal en een modem. Het modem zit op /dev/tty1, de terminal op /dev/tty2. De ttys-file heeft de volgende inhoud:

```
100
0d1
2f2
```

Inhoud van "/etc/ttys"

De eerste regel geeft aan dat het om de console gaat. De console heeft geen baudrate, vandaar de 0 op de tweede positie. De console wordt ook wel aangeduid als /dev/tty0, line 0 dus.

De tweede regel beschrijft het modem. Er mag niemand zomaar inloggen op het modem; er wordt geen proces op opgestart. Standaard staat 't ding op 2400 baud, 8 databits en geen parity (de "d"). Het modem hangt aan lijn 1, /dev/tty1.

De derde regel is de terminal op tty2. Hoewel 't ding met een null-modem aan de PC verbonden is, wordt er toch een getty voor opgestart. De reden hiervoor zal ik hieronder verduidelijken. Het ding staat standaard op 9600 baud, 8 databits en geen parity (mode "f").

Als een terminal eenmaal kenbaar is gemaakt aan het systeem dient het systeem nog af te weten van de preciese parameters van de terminal. Een VT100 heeft immers andere stuurcodes dan een televideo TV950 terminal. Daartoe dienen de andere twee files: /etc/ttytype en /etc/termcap. Mijn ttytype file heeft de volgende inhoud:

```
minix tty0
vt100 tty1
vt100 tty2
```

Inhoud van "/etc/ttytype"

Iedere regel geeft van één der terminals een identifier. De console is van het type "minix console"; de andere twee lijnen hebben als type "vt100". Deze identifiers verwijzen naar de terminal definities zoals die gegeven zijn in /etc/termcap. De /etc/termcap file geeft voor iedere mogelijke terminal de codes die van de terminal komen en welke codes naar de terminal gestuurd moeten worden om bepaalde effecten te bereiken. Daarnaast wordt wat aanvullende informatie opgegeven zoals het aantal regels en characters per regel van het betreffende terminal type.

Terug naar getty. Voor het getty proces zijn ook nog een aantal andere files nodig. In MINIX is de file "/etc/gettydefs" nodig; de file /etc/issue kan ook nog voorkomen. In de gettydefs file staat voor iedere mogelijke instelling een regel. Deze regel heeft de vorm:

<label>#<initial flags>#<final flags>#<login prompt>#<next label>

Het eerste label is een simpele identifier. Het tweede veld staat vol met de zogenaamde "initial flags". Als getty opstart worden de terminal parameters die hier genoemd worden gebruikt. Getty gaat vervolgens staan wachten op bijvoorbeeld een carrier of een ander levensteken op de lijn waarvoor'ie is opgestart. Hij stuurt dan de tekstfile /etc/issue naar de terminal (als die tenminste aanwezig is!) en de loginprompt zoals genoemd in het vierde veld. De gebruiker moet zich nu identificeren. Na het inlezen van de naam. De terminal wordt nu volgens de parameters in het derde veld ingesteld en getty stopt. Login neemt de verdere afhandeling over. De regels in de gettyfile eindigen allemaal met de label waarmee de volgende poging ter herkenning van de terminal gewaagd wordt als deze poging mislukt.

Voor MINIX wordt als standaard terminal altijd "minix" aangenomen. Om te voorkomen dat het operating system denkt dat het te maken heeft met een minix-compatible terminal als de gebruiker inlogt op de VT100, wordt het terminal type expliciet

in de .profile van de gebruiker geset. In standaard UNIX kan dit met het commando: . tset. In MINIX wordt dit: eval tset.

Optimalisatie

Tenslotte nog wat over systeemonderhoud ten behoeve van de totale perfomance van het systeem. Door het regelmatig weggooien en/of toevoegen van files aan het filesystem, raken directories en files geheel gefragmenteerd over de disk. Dit is uiteraard nadelig bij het inlezen van files; diskaccess loopt behoorlijk terug in snelheid als de kop veel heen en weer staat te "zagen". In MS/PC-DOS kan een (Hard-) disk gemakkelijk geoptimaliseerd worden (lees: gedefragmenteerd en gesorteerd) met behulp van allerlei disk optimizing programma's. Voor UNIX is er nog geen utility die dat kan. IJverige systeembeheerders richten daarom eens in de zoveel tijd het filesystem helemaal opnieuw in.

Op grote UNIX systemen wordt er veel meer accounting gedaan dan alleen het registreren van user-logins en diskmounts. Er kunnen dan allerlei leuke en interessante statistieken bekeken worden met behulp van het iostat commando. Verder is er nog /etc/accton, een programma dat shell-accounting bijhoudt. Met behulp van het programma "sa" kan bekeken worden hoevaak een bepaald proces opgestart is, hoeveel tijd het in beslag nam en hoeveel CPU-tijd het proces gebruikte. Wordt een bepaald programma erg veel opgestart, dan verdient het aanbeveling het betreffende programma snel opvraagbaar te maken. In sommige UNIX-systemen is daar een speciaal bitje voor gereserveerd in de file attributes: het "sticky" bitje. Een file met het sticky bitje geset wordt zo veel mogelijk in het geheugen gehouden.

Niet alleen het setten van een sticky-bitje kan de zaken behoorlijk versnellen; ook de volgorde waarin bepaalde programma's op disk staan kan veel uitmaken. Dat wil zeggen: in non-Berkeley systemen. Bij het ontwerpen van BSD UNIX werd het optimalisatie probleem opgelost door gebruik te maken van hashtables. Op andere systemen wordt dat vaak niet gedaan, waardoor de inode informatie niet echt snel toegankelijk is. Deze inodes staan aan het begin van de disk; als ze niet in het geheugen vastgezet kunnen worden (hardbuffered), dan moet de kop van de disk steeds naar het begin van de disk om de inode informatie op te halen. De meest gebruikte files worden dan zo dicht mogelijk aan het begin van de disk neergezet zodat de koppen de kleinste afstand af hoeven te leggen.

Als laatste noem ik nog de verschillende monitor programma's. Met een monitor programma kan je heel aardig de statistieken van de (disk-) access in beeld krijgen in de vorm van staafdiagrammetjes. Dit lijkt soms best aardig; helaas zijn de huidige monitor programma's zo belastend voor de machine dat de optimalisatie die extra bereikt kan worden geheel teniet gedaan wordt door de activity monitor zelf. De statistieken kunnen trouwens ook door de monitor beïnvloed worden; vooral als er relatief weinig andere processen draaien!

In sommige UNIX-systemen is daar een speciaal bitje voor gereserveerd in de file attributes: het "sticky" bitje.

Wel, dit was dan de laatste aflevering in deze serie over UNIX. Ik hoop met deze serie wat meer informatie gegeven te hebben dan in het gemiddelde boek terug te vinden is. Met name over systeembeheer is er weinig informatie te vinden. Ik heb persoonlijk veel gehad aan de documenten die in de verschillende literatuur lijstjes opgesomd zijn. Ik hoop dat u, de lezer, eveneens profijt trekt van deze serie. Tot een volgend keer!

Literatuur:

1: Eric Foxley, UNIX For Super-users, Addison-Weshley; International Computer Science Series (ISBN 0-201-14228-7)

Joost Voorhaar

't Was eruit voor ik er erg in had

Weet u nu, wat het tegengestelde is van U-NIX? In de handel is dat: Wij alles...

Layouter

Nieuwe versie van het KGN PC/XT-BIOS

Bestellers van het KGN PC/XT BIOS weten het allang: een beter BIOS is er niet. Toch wel. Het was niet mooi genoeg. Er kon nog wat bij. Het heeft menig zweetdruppeltje gekost, maar het is gelukt: het KGN PC/XT-BIOS herkent nu ook een MM58167 batterijklok.

Hoezo, nokvol?

De wens voor ondersteuning van een dergelijke batterijklok was er al bijna twee jaar. Verzoekje van Joost Voorhaar die toen nog op een XT zat te hengsten. Hoe dat moest was ook al bekend: voor het booten tick-tijd uitrekenen en goed zetten, en calls toevoegen voor minimaal INT 1Ah, AH = 2, AH = 3, AH = 4 en AH = 5. Zo is het snel en simpel neergeschreven. Bestellers van de BIOS sourcelisting wisten het al: het BIOS was nokkie, nokkievol. Als je ging tellen, bleven er misschien zo'n 30 bytes over. Nu kun je in assembler met 30 bytes nog wel wat beginnen maar dan moeten ze wel allemaal bij elkaar staan. En dat was niet zo...

Het heeft menig zweetdruppeltje gekost, maar het is gelukt: het KGN PC/XT-BIOS herkent nu ook een MM58167 batterijklok.

Eerst maar eens de routines voor de klok gemaakt, en in een extensie-ROM gegoten. Uitgetest, en zo klein mogelijk gemaakt. Resultaat: ROMTIMER.ASM. Nodig binnen het BIOS: minstens 100 bytes, als het even kan nog een paar meer. Sourcelisting uitgedraaid. Speuren naar mogelijke besparingen in bytes. Dat was al 10 keer gebeurd om de overige pret van het BIOS erin te krijgen. Maar zoals een bekend gezegde luidt: de aanhouder wint. In het INT09/IRQ1 gedeelte van het BIOS bleek nog gesnoeid te kunnen worden. Resultaat: 50 bytes winst. Ook in de POST kon nog wat ruimte worden gevonden. En in de NMI/pariteitfout afhandeling. Tot mijn grote verbazing was zelfs in het floppy disk gedeelte nog plaats te winnen.

Na een heel weekeinde van passen en meten zat het erin. Ik geloof het nog steeds niet, maar het is zo. Netto ruimte in het BIOS nu: minder dan 10 bytes, her en der verspreid.

Wat is er nu veranderd?

Het BIOS ondersteunt thans een MM58167 Real Time Clock op I/O adres 0340h-035Fh. Een dergelijke RTC wordt meestal bediend met het programma TIMER.COM of CLOCK.COM. Deze RTC komt het meest voor in PC/XT's, en zat ook in de club XT's. Kijk bij twijfel in de documentatie van uw multi I/O kaart. Zit een dergelijke RTC in uw machine, dan heeft het zin een nieuw BIOS te proberen, en TIMER/CLOCK.COM uit uw AUTOEXEC.BAT te slopen. De RTC wordt dan gewoon gelijk gezet door TIME en DATE te gebruiken. Dit laatste werkt alleen met MS/PC-DOS 3.30 en hoger.

Het BIOS test niet langer op een RTC op adres 0240h, om de simpele reden dat de extra code om de RTC op twee verschillende adresen te ondersteunen in niet in ROM past. Om die reden wordt het adres van de RTC ook niet meer in de equipment list vermeld bij het starten.

Er is nog een tweede verbetering doorgevoerd: er wordt getest of er een ander floppy BIOS actief is. In dat geval wordt de drivetest om te bepalen welke drives 720k zijn, overgeslagen. Een dergelijke situatie kan voorkomen als u een speciale floppy card in uw machine heeft, die ook 1.2Mbyte en/of 1.44Mbyte floppy drives ondersteunt.

Voor de rest is het BIOS geheel hetzelfde gebleven.

Hoe kom ik eraan?

Bezitters van het bestaande BIOS kunnen de EPROM die ze nu hebben naar ondergetekende sturen. Gelieve postzegels voor retourporto bij te voegen. Ze ontvangen dan na enige tijd de nieuwste versie. Voor het geval u het KGN PC/XT BIOS nog niet bezit geldt de klassieke regeling: u kunt het BIOS bestellen bij de penningmeester door fl. 25,- over te maken op Postbank rekening no. 3757649 onder vermelding van KGN PC/XT BIOS. U krijgt de EPROM dan per omgaande thuisgestuurd.

Nico de Vries

Paper Disk, listings compact weergegeven

Enkele jaren geleden las ik in het (duitse) tijdschrift mc een artikeltje van iemand die het afdrukken van listings maar verspilling van ruimte vond en daar een zeer creatieve oplossing voor bedacht had. Hij zette zijn printer in grafische mode en stuurde de listing vervolgens naar de printer. Op deze manier kreeg je een "listing" waarbij voor elke letter (byte) slechts 8 dots op papier gebruikt werden, een reductie van zo'n 85%.

Uiteraard werd deze "oplossing" als geintje afgedaan, je had bijvoorbeeld geen mogelijkheden de listing weer te vertalen naar iets dat door mens of computer leesbaar was.

In mei van dit jaar verscheen het juninummer van mc met daarin een nieuwe manier voor het uitwisselen van informatie die toch op dat oude idee voortborduurde; de zogenaamde paper disk. Met behulp van een paper disk kan men met behulp van druktechnieken voor computer leesbare informatie verspreiden. In dit artikel wil ik enkele aspecten van dit nieuwe medium behandelen.

Voor de volledigheid: Paper Disk is in opdracht van mc ontwikkeld door Rolf-Dieter Klein en een patent hiervoor is aangevraagd. Er bestaan ook geen plannen in de μP Kenner listings op deze manier af te gaan drukken (maar wat niet is kan uiteraard nog komen).

Methoden voor uitwisseling van informatie

Computertijdschriften, waaronder ook de μP Kenner, hebben het probleem dat een tijdschrift eigenlijk niet geschikt is voor het verspreiden van software. Je kunt listings af gaan drukken in het blad met als nadeel dat je je moet beperken tot de echt kleine programma's. Er gaan ongeveer 60 regels op een pagina en met een omvang van 50 pagina's (μP Kenner) is tien pagina's voor één listing eigenlijk al te veel. Nu wil het geval dat nuttige programma's vaak een omvang hebben van enkele honderden tot zelfs wel één- of tweeduizend regels zodat die programma's zeker niet in de vorm van een listing afgedrukt kunnen worden. Een tweede nadeel van het afdrukken van listings is het feit dat de listing eerst nog door de lezer ingetikt moet worden alvorens hij (of zij) het programma kan gaan gebruiken.

Deze problemen komen ook binnen de KGN regelmatig naar voren, het ene lid wil graag veel listings in de μP Kenner (tot zelfs de complete source van DOS-65), het andere lid vindt dit zonde van het papier. De redactie probeert een koers te varen waarbij ongeveer 20 tot 30% van het blad gevuld is met listings en de rest met andere informatie. Dit beperkt de lengte van programma's die geplaatst kunnen worden tot ongeveer 600 regels.

Een alternatief die we ook wel hanteren is het verspreiden van software via andere manieren. In de μP Kenner verschijnt dan een inleidend verhaal over de toepassingen van het programma en de leden kunnen het programma op flop bestellen of downloaden van het bulletin board The Ultimate. Uiteraard heeft deze methode ook z'n nadelen. Je kunt niet meteen na het lezen van het artikel aan de slag met het programma maar moet het eerst ophalen van het BBS (in gesprek @!#@$#@!&!!) of een flop bestellen. Een tweede nadeel is het feit dat als je het programma niet direct nodig hebt, je meestal niet na 3 jaar het programma nog even van het BBS kunt halen.

Kortom, het beste zou zijn dat programma's ook in het blad staan en dan het liefste op zo'n manier dat de computer ze nog kan lezen. Een alternatief zou zijn dat bij elke uitgave van de μP Kenner een flop meegestuurd werd met daarop alle software maar ja, er bestaat helaas nog steeds geen universeel disketteformaat. Bovendien, wie gaat die 200 floppies elke twee maand kopieren?

Mogelijkheden

Er zijn een aantal manieren om informatie op papier in te lezen in de computer. In de eerste plaats is dat via bar- of streepjescode. De bekendste toepassing van barcode is de EAN-code die tegenwoordig op zo'n beetje alle artikelen staat afgedrukt die in supermarkten worden verkocht. In deze code staat een getal gecodeerd die ook onder de barcode staat afgedrukt (interpretatie). Met behulp van dit getal kunnen de kassa's inlezen welke artikelen de klant gekocht heeft en de kassabon opstellen. Op dezelfde manier is het ook mogelijk teksten in barcode af te gaan drukken.

Barcode wordt nauwelijks toegepast om programma's in te lezen. Er is namelijk relatief veel ruimte nodig om de informatie af te drukken (kijk maar eens naar de verhouding barcode/tekst in de codering van een melkpak) zodat er voor een listing nog veel meer ruimte nodig is. De enige toepassing van programmatuur in barcode die ik ken waren de HP-rekenmachines die met behulp van een leespen programma's in barcode in konden lezen.

De tweede manier is het afdrukken van een listing met een zodanig lettertype dat een speciaal computerprogramma in staat is deze listing met een leespen of scanner in te lezen. Hiervoor is een speciaal

lettertype gedefinieerd: OCR die onder andere gebruikt wordt op overschrijvingsformulieren van de postgiro. Het blijft echter zo dat er nog steeds veel ruimte nodig is voor het afdrukken van een listing. Groot voordeel is wel dat iedereen die kan lezen ook de listing kan lezen, zonder dat hij de beschikking hoeft te hebben over een computer voorzien van inlees-apparatuur.

De derde manier lijkt eigenlijk het meest op het voorstel van enkele jaren geleden. Een printer kan een bepaald aantal punten (dots) per inch afdrukken. Gebruik elke punt om aan te geven of een bit een 1 of een 0 is. Met de beschikbare huis tuin en keuken laserprinters kun je op die manier al 300 bits per inch (25,4 mm) kwijt en ook voor de grafische industrie is dit totaal geen probleem. Met behulp van een scanner wordt vervolgens de informatie ingelezen in de computer waarna een programma er weer listings van maakt. Het is uiteraard niet verstandig het onderste uit de kan te gaan halen, ook als je slechts 50 bits per inch af zou drukken kom je al tot een enorme informatie-dichtheid. Volgens mc gaat er op één pagina A4 30 kilobyte ongekomprimeerde tekst. Door de listings eerst te komprimeren, gaat er in totaal bijna 100 kilobyte op een pagina; dat is ongeveer 50 keer zoveel als een listing in tekst. Kortom, het enige probleem dat voor de lezer overblijft is de investering in een scanner.

Paper Disk nader bekeken

In figuur 1 is een voorbeeld afgedrukt van een blok uit de paper disk. Op een pagina A4 staan maximaal 21 van dergelijke blokken. Elk blok bevat 1,7 kB aan nuttige informatie waarbij er een groot aantal voorzorgen getroffen zijn om te controleren of de informatie goed ingelezen wordt en eventueel geconstateerde fouten te corrigeren. Ongeveer een kwart van de ruimte wordt gebruikt voor crc's (cyclic redundancy checks) en stuur-informatie. Zo dienen de balken aan de linker, rechter en bovenkant van het blok voor het nauwkeurig indelen van het blok in de afzonderlijke punten zodat het programma precies weet waar een punt kan staan.

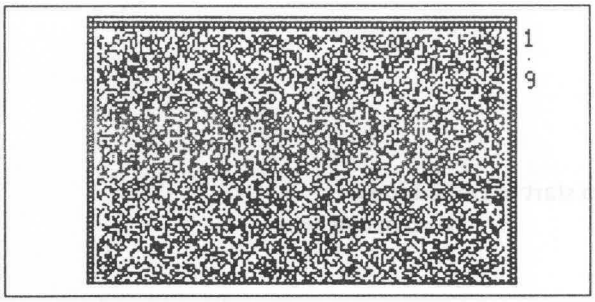


Fig. 1: voorbeeld van een blok uit een paperdisk

Met behulp van een vlakbed- of handscanner worden de blokken één voor één ingelezen in de computer en op de (harde) schijf opgeslagen in een bepaald grafisch formaat (PCX-formaat). Vervolgens wordt deze grafische informatie door een inleesprogramma geconverteerd naar een archive, vergelijkbaar met een gezipte of gearcte file. Tenslotte kan het archive uitgepakt worden en krijgen we de oorspronkelijke informatie terug.

Vooral bij het gebruik van een handscanner zijn er een aantal zaken die fout kunnen gaan. De scanner moet gelijkmatig over het papier bewogen worden. Verder moet de scanner zo goed mogelijk haaks op de randen van het blok staan. Het inleesprogramma is, met behulp van de synchronisatiebalken, in staat een groot deel van dergelijke afwijkingen te corrigeren. Om deze reden heeft een vlakbed-scanner de voorkeur boven een handscanner.

Benodigdheden

Hoewel, bij mijn weten, op dit moment alleen mc gebruik maakt van deze techniek, lijkt het mij niet onwaarschijnlijk dat meer tijdschriften in de toekomst informatie via Paper Disk gaan verspreiden. Het lijkt mij zelfs niet onmogelijk dat ook in de µP Kenner listings via deze techniek zullen worden uitgewisseld, zeker als het Minix-project inderdaad die grote vlucht gaat nemen. Franzis-Verlag GmbH geeft ook aan dat ze die mogelijkheden open houdt en ieder die Paper Disk wil gaan gebruiken kan contact met hen opnemen.

Om Paper Disk te kunnen gebruiken, heb je absoluut een scanner nodig. Dit kan zowel een handscanner of een vlakbed-scanner zijn waarbij voor de handscanner een resolutie van 400 dpi (dots per inch) wenselijk is en voor een vlakbed-scanner een resolutie van 300 dpi. Verder heb je een inleesprogramma nodig die de grafische informatie in het zogenaamde PCX-formaat vastlegt of een programma die het gescande naar een dergelijk formaat om kan zetten. Tenslotte nog een inleesprogramma dat momenteel alleen nog voor MS-DOS beschikbaar is. Dit programma (MCREADER) kan voor DM 9,90 + DM 1,70 (5 1/4 ") voor verzending en verpakking bij de uitgever besteld worden waarbij dan meteen het benodigde de-archive programma ARJ op de flop staat.

Gert van Opbroek

Literatuur

1: Peter Cechowitz: Platzsparende Programm-Listings, mc 4 1986 blz. 83

2: Rolf-Dieter Klein: Futter fÜr Scanner, mc 6 1991 blz. 140

RAMTEST voor DOS65

In een grijs verleden stond er in de μP Kenner een RAMtestprogramma, geschreven voor de Octopus65. Nu heeft een DOS65 machine geen RAMtest: het is zelfs zo, dat IO65 ervan uitgaat, dat het RAM functioneert! Daarom onderstaand programma. Kunt u uw RAM ook eens testen.

Frank Vandekerkhove

```
; file          RAMTEST.mac
; program       RAMTEST
; function      test specific RAM range
; usage         RAMTEST
; by            Frank Vandekerkhove
;               Sint-Michielsstraat 4
;               B-2789 Verrebroek-Beveren
;               + 32 3 773.27.94
; date          20 aug. 1989
; ref.          Elector april 1982
;               De 6502 kenner nr. 50: RAM test for octo65 by M. Lachaert

              lib     dvar.mac

              org     $200

main          jmp     ramt
              fcc     $c8,$c5,$cc,$d0
              fcc     $4c,$4c,$4c
hlpinfo       fcc     '\rUtility to test a RAM area.'
              fcc     '\rSyntax: RAMTEST start,end (address in hex)'
              fcc     '\rNo options'
              fcc     '\rAbbr.: RAMT',0

ramt          ldx     #$00                    hex parms without +
              jsr     spar                    get parameters
              fcc     t1,t2,0                 two parms
              bcc     1.f
              jmp     ermes                   print error and back to caller

1             jsr     print                   clear and tittle
              fcc     '\f\Ei RAMTEST from $',0
              lda     t1+1
              jsr     hexout
              lda     t1
              jsr     hexout
              jsr     print
              fcc     ' to $',0
              lda     t2+1
              jsr     hexout
              lda     t2
              jsr     hexout
              jsr     print
              fcc     ' \r\r Up \En',0    test from start to end address
              jsr     zero                        zero in test area
```

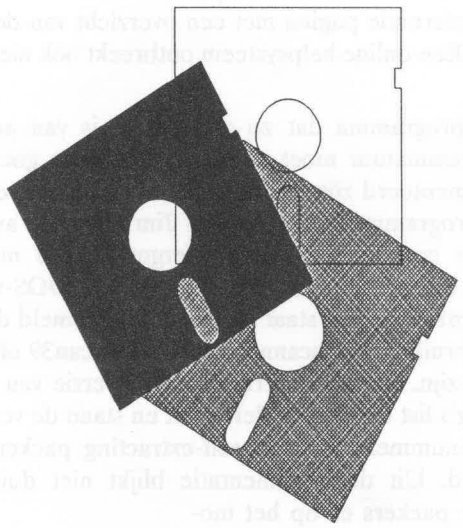
Fig. 1: sourcetekst van RAMTEST.MAC

```
                jsr     setpnt                  set pnt equal to start address

2               jsr     walk                    test each bit of a cell
                bne     4.f
                lda     #$ff                    test pattern for double addressing
                sta     [t3],y
                jsr     incpnt                  inc the actual address
                bcs     2.b                     next cell to test bits

                jsr     print
                fcc     '\r\Ei Down \En',0 test from end to start
                jsr     zero
                ldx     t2
                stx     t3
                ldx     t2+1
                stx     t3+1

3               jsr     walk
                bne     4.f
                lda     #$ff
                sta     [t3],y
                jsr     decpnt
                bcs     3.b

                jsr     print
                fcc     '\r\Ei Succesfull test. \En',0

exit            jsr     crlf
                jmp     crlf

4               jsr     print
                fcc     '\rError at $',0
                lda     t3+1
                jsr     hexout
                lda     t3
                jsr     hexout
                jsr     print
                fcc     '\rRestart/Continue/Quit ? (R/C*/Q) ',0
                jsr     in
                jsr     loupch
                cmp     #'Q                     quit ?
                beq     exit
                cmp     #'R                     restart ?
                beq     5.f
                jsr     incpnt                  inc. pointer to next cell
                ldy     t3                      restart test at this address
                lda     t3+1
                sty     t1
                sta     t1+1
5               jmp     1.b
;
;*** SUBROUTINES ***
;
zero            jsr     print
                fcc     '\r$00 in area.',0
                jsr     setpnt                  fill test area with $00
                ldy     #$00
```

```
1           lda     #$00
            sta     [t3],y
            jsr     incpnt
            bcs     1.b
            jsr     print
            fcc     '\rTest each bit of a cell.',0
            rts
;
setpnt      lda     t1                  copy start address (t1) to pointer (t3)
            sta     t3
            lda     t1+1
            sta     t3+1
            rts
;
incpnt      inc     t3                  inc. pointer
            bne     1.f
            inc     t3+1
1           sec                         check pnt = end address
            lda     t2
            sbc     t3
            lda     t2+1
            sbc     t3+1
            rts
;
decpnt      sec                         dec. pointer
            lda     t3
            sbc     #$01                dec. one
            sta     t3
            lda     t3+1
            sbc     #$00
            sta     t3+1
            sec                         check pnt = start address
            lda     t3
            sbc     t1
            lda     t3+1
            sbc     t1+1
            rts
;
walk        lda     #$01                init pattern
            sta     t4
            ldy     #$00
            lda     [t3],y              is $00 still present ?
            bne     2.f
            ldx     #$08                walk counter
1           sta     [t3],y              store pattern in memory
            cmp     [t3],y              pattern still present ?
            bne     2.f
            asl     t4                  walking bits
            dex
            bne     1.b                 test each bit in cell
2           rts
;
            end     main
```

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

Zomertijd is komkommertijd. Ook, of beter: juist in de public domain en shareware hoek. Daarom deze keer een blik op een aantal utilities die eigenlijk al heel lang op mijn harddisk rondzwerven en die ik dagelijks gebruik zonder er bij stil te staan.



Dat er veel verschillende packers, zippers en arcers zijn, zal zo langzamerhand niets nieuws meer zijn. Een heel handige utility voor het werken met allerlei soorten archives is het programma "shez". Waar de naam vandaan komt valt niet te achterhalen; in de documentatie wordt overal gesproken over shez als "the compression companion". In principe is shez een heel eenvoudig programma. Je kunt er mee door de complete directory structuur van de (hard-) disk wandelen en alle archives die je op je weg tegenkomt uit en te na bekijken, converteren en manipuleren. Het programma is grotendeels menugestuurd, maar voor vrijwel alle commando's is er wel een hotkey. Shez steunt geheel op andere programmatuur. Zo is voor iedere compressie methode de bijbehorende unpacker nodig. Wil men files in een archive kunnen bekijken en/of aanpassen, dan is ook een externe editor en een list-programma noodzakelijk. Tenslotte biedt shez ook de mogelijkheid om in archives te scannen voor virussen. Daartoe heeft men dan uiteraard wel weer een losse virusscanner nodig.

Installatie

De installatie van shez is simpel. Er is een eenvoudig configuratie programma genaamd "shezcfg" waarmee je allerlei default instellingen als kleuren, ondersteunde packers en externe programma's kunt definieren. Tijdens de installatie wordt onmiddelijk duidelijk dat shez slechts een beperkt aantal packers ondersteunt. De zes meest gebruikte packers zijn gelukkig allemaal bekend: zip, arc, lharc, arj, zoo en pak. Wat minder bekende packers (UNIX compress en tar, hyper, warp, dwc, lbr, ...) worden niet ondersteund door shez en het is niet mogelijk om nieuwe arcers toe te voegen aan het geheel. Dat betekent wel dat er iedere keer een update van shez uitkomt zodra een nieuwe packer populair wordt. Verder kan alleen gebruik gemaakt worden van de virusscanner van McAffee's scan. Liefhebbers van andere viruskillers zullen het dus zonder hun favoriete antibioticum moeten stellen. Gelukkig is het wel mogelijk om willekeurige editors en listers te gebruiken (hoewel in de documentatie alleen Vern Buerg's "list" genoemd wordt, zie verderop in dit artikel). Een eenvoudige doch doeltreffende installatie procedure dus. Het was echter meer in de stijl van shez zelf geweest als de installatie procedure ook de mogelijkheid biedt om slechts één item te selecteren. Nu moet je je bijvoorbeeld iedere keer door alle "advanced options" heen worstelen om de gevoeligheid van de muis aan te passen aan je eigen wensen.

Dat er veel verschillende zackers, pippers en arcers zijn, zal zo langzamerhand niets nieuws meer zijn.

Het dagelijks gebruik

Shez is door zijn eenvoudige opzet makkelijk te gebruiken. De menu's zijn redelijk doordacht en je kunt er zo mee overweg. De "standaard" toetsen zoals Escape en Enter werken geheel zoals je van ze mag verwachten. Lanzamerhand beginnen ook steeds meer programma's de toetscombinaties ALT-X en ALT-Q te ondersteunen. Helaas, hier gaat shez jammerlijk de mist in. ALT-Q wordt ondersteund, maar de voor de hand liggende combinatie ALT-X is CTRL-X geworden.

De muis blijft bij shez ook niet werkeloos. Je kunt het ding naar believen gebruiken of uitschakelen. Helaas blijkt ook hier weer de starheid van het programma: je kunt geen andere functies onder de muistoetsen verstoppen dan die, die de auteur aan zijn grijze massa heeft laten ontspruiten. De gevoeligheid van de muis is gelukkig wél in te stellen.

Documentatie

The compression companion is goed gedocumenteerd. Het programma is echter zodanig in elkaar gezet dat de documentatie nauwelijks tot niet ge-

raadpleegd hoeft te worden. De docs zijn overzichtelijk gerangschikt en een redelijke index ontbreekt evenmin. Naast de 87 kilobytes in de .DOC-file, is er een referentie pagina met een overzicht van de hotkeys. Een online helpsysteem ontbreekt ook niet.

Een programma dat zo afhankelijk is van andere programmatuur moet uiteraard bijzonder goed gedocumenteerd zijn wat betreft bruikbare versies van die programmatuur. Hier laat Jim Derr, de auteur, echter een steekje vallen. Nergens in de manual staat informatie over de te gebruiken DOS-versie bijvoorbeeld. Wel staat uitdrukkelijk vermeld dat de te gebruiken virusscanner McAffee's scan39 of later moet zijn. Verder is vermeld welke versie van Vern Buerg's list de muis ondersteunt en staan de vereiste versienummers voor de self-extracting packers genoemd. Uit de documentatie blijkt niet duidelijk welke packers er op het moment van de verschijning van shez allemaal beschikbaar waren.

List

Van de programma's die door shez benodigd zijn is alleen Vern Buerg's list nog niet besproken in deze rubriek. De laatste versie op het moment van schrijven is versie 7.5i. De laatst beschikbare versie op dit moment is echter versie 7.5f. De 7.5i (injectie?) heb ik helaas niet geheel compleet en kan dan ook niet verder beschikbaar gesteld worden.

Het vervult de meest simpele functies van een shell; je kunt er mee door directory structures wandelen, files copiëren, verhuizen, weggooien of uitvoeren.

In principe is list een eenvoudig programma; je kunt er teksten mee bekijken. Maar daar blijft het niet bij. Door de vele nuttige uitbreidingen is list een bijzonder krachtige utility geworden. Het vervult de meest simpele functies van een shell; je kunt er mee door directory structures wandelen en naar believen files copiëren, verhuizen, weggooien of uitvoeren. Verder kun je de file bekijken of een externe editor op het ding loslaten. Dat bekijken gaat ook al met superlatieven gepaard. Een binary file? Geen probleem; je kunt de zaak zo in hex-mode zetten! Een bestandje van een tekstverwerker die bijvoorbeeld het achtste bit misbruikt om carriage returns van end of lines te onderscheiden? No problem; gewoon in 7-bits mode verdergaan! Verder kan het ding zoeken, printen, stukken files uitknippen (en printen) en dat alles kan ook nog in windows.

Vanaf list versie 7.2 wordt ook de muis ondersteund. Mijn ervaring is dat de muis in dit soort applicaties alleen maar onhandig is, maar dat is een persoonlijke mening.

De standaard distributie van list bevat een viertal verschillende versies. De kleinste versie (lists) heeft geen help mogelijkheden, kan files aan tot ca. 600 kByte en heeft geen shell to DOS mogelijkheden. Deze small versie heeft echter slechts 30 kilobyte vrij RAM nodig om te kunnen draaien. De reguliere versie (listr) gebruikt ca. 80 kilobyte RAM. Daarmee kan hij files aan met een maximale grote van maar liefst 16 megabyte. Dan is er de plus-versie. Deze heeft meer helpschermen beschikbaar en heeft wat file management mogelijkheden. Tevens biedt de plus-versie een telephone dialer. Tenslotte is er nog de enhanced versie (liste), maar die is alleen bedoeld voor de commerciële markt en zit niet in de shareware distributie. Wel wordt in die shareware distributie een versie geleverd die via ANSI codes en standard output werkt (drlist).

Configuratie

De configuratie van list is merkwaardig. Configuratie en/of installatie programma's zitten er niet bij. Een groot aantal functies kan echter ingesteld worden als je met het programma bezig bent en de executable kan dan vervolgens "gecloned" worden. Een aantal zaken is niet op die manier in te stellen, maar daar heeft Vern Buerg debug-patches voor meegeleverd. De documentatie beschrijft alle belangrijke locaties waar defaultwaarden ingesteld kunnen worden. Het geheel komt zo slordig over. Een installatie programma'tje dat vrijwel hetzelfde doet als de patches lijkt me zo in elkaar gedraaid en is een stuk zekerder.

Utilities

De plusversie biedt de mogelijkheid om de inhoud van gearchiveerde bestanden te bekijken. Daartoe wordt gebruik gemaakt van een extern programma "fv", dat bij de standaard distributie zit. Tevens is er voorzien in een ont-ARC programma dat compatible is met SEA's ARC. Naast deze utilities heeft Vern Buerg nog een hele stapel ander min of meer handige utilities geschreven. Zie de mailer die bij list is ingesloten.

Documentatie

De documentatie van list is goed verzorgd. Ruim 100 kilobyte aan informatie met een goede index en een redelijke inhoudsopgave. Jammer is wel dat drlist

een eigen docfile heeft gekregen. De documentatie van fv en arce is wat magertjes, maar zo te zien staat alles genoemd dat van belang zou kunnen zijn in samenwerking met list.

Conclusie

Ofschoon ik een fervent gebruiker van 4DOS ben, maak ik vaak gebruik van Vern Buerg's list utility omdat het ding een stuk meer kan dan de ingebouwde list optie van 4DOS. Shez is vooral handig als u veel documentatie opgeslagen heeft op harddisk. Het programma functioneert goed en is behoorlijk stabiel gebleken. Shez vereenvoudigt het werken met archives aanzienlijk zonder dat je de indruk krijgt met window-achtig speelgoed bezig gehouden te worden.

Joost Voorhaar

Besproken produkt	: Shez (the compression companion), versie 6.1
Categorie	: Systeem utilities
Registratie	: $30,00
Auteur/Leverancier	: James Derr / California Software Design
Verkrijgbaarheid	: The Ultimate, arcers alle systemen Shez61.Zip, ca. 140 kB.

Minimale systeemeisen

Niet gespecificeerd

Algemene beoordeling

Documentatie	: In orde. Engelstalig
Online help	: Redelijk, standaard help schermen
Gebruikersinterface	: Combinatie van keyboard en muis, pulldown menus
Muisondersteuning	: Redelijk. Muisfuncties niet definieerbaar

Positief

Eenvoudige bediening
Geen exceptionele eisen aan apparatuur

Negatief

Externe programmatuur nodig
Vrij star in configuratie

Eindbeoordeling

Stabiliteit	: 8
Bruikbaarheid	: 8
Totaalresultaat	: 8

'n Knal assembler

Wordperfect is een mooi pakket. Geinstalleerd tegenwoordig meer dan een megabyte. Soms kun je in een handvol bytes ook wel wat. Bijvoorbeeld de CapsLock aanzetten. Tik dit eens in DEBUG:

```
-A 100
XXXX:0100     MOV AX,40 <enter>
XXXX:0103     MOV DS,AX <enter>
XXXX:0105     OR Byte Ptr [0017],40 <enter>
XXXX:010A     MOV AX,4C00 <enter>
XXXX:010C     INT 21 <enter>
XXXX:010F     <enter>
-NCAPSLOCK.COM <enter>
-R CX <enter>
0000: F <enter>
-W <enter>
Writing 000F bytes
-Q <enter>
```

Nu heeft u een kolossaal programma dat CapsLock activeert. Handig voor in de AUTOEXEC.BAT. Op een AT worden zelfs de LEDs goed aangepast.

Nico de Vries

Besproken produkt	: List, versie 7.5f
Categorie	: Systeem utilities
Registratie	: $30,00
Auteur/leverancier	: Vern Buerg
Verkrijgbaarheid	: The Ultimate, msdos utilities 2 List75f.Zip, ca. 100 kB.

Minimale systeemeisen

IBM PC/XT compatible computer
64 kB vrij RAM, meer bij gebruik van verschillende opties
PC/MS-DOS 2.0 of later

Algemene beoordeling

Documentatie	: Uitgebreid, Engelstalig
Online help	: Helpschermen voor verschillende situaties
Gebruikersinterface	: Keyboard, muis. Klein beetje windowing
Muisondersteuning	: De muis kan gebruikt worden met list

Positief

Verschillende uitvoeringen "to suit your needs"
Eenvoudig maar krachtig
Snel

Negatief

Opmerkelijke configuratie procedures

Eindbeoordeling

Stabiliteit	: 8
Bruikbaarheid	: 8
Totaalresultaat	: 8

Enkele matrix-bewerkingen in C

Het onderstaande programma bevat een aantal functies waarmee enkele basis-bewerkingen op matrices uitgevoerd kunnen worden. Verder is een voorbeeld van het gebruik van deze functies nader uitgewerkt. In dit voorbeeld wordt een zogenaamd stelsel van vergelijkingen opgelost.

Voor de bepaling van de inverse van een matrix, wordt gebruik gemaakt van een zogenaamde Gauss-eliminatie. De beschrijving en het bewijs van dit rekenschema valt buiten dit artikel.

Gert van Opbroek

```
#include <stdio.h>
#include <math.h>

#define FALSE 0
#define TRUE 1

#define onder_grens 1e-300

extern *calloc();

/*****************************************************************************
MATDET:   Bereken de determinant van een matrix
Aanroep: double *matrix
         extern double matdet();
         int orde;
MATDET(matrix,orde);
Parameters:  matrix: pointer op de te inverteren matrix
             orde:   orde van de matrix
Returns:     Determinant
Bron: GEVOP 05-09-87 (C-version)
*****************************************************************************/

double MATDET(matrix,orde)

double *matrix;
int   orde;

{
    int R,R1,K,K1;
    double determinant;
    double deel_produkt;

    determinant =  0.0;
    K1  = -1;

    for (R1=0;R1<orde;R1++) {
        K1++;
        deel_produkt = 1.0;

        for (R=0,K=K1;R<orde;R++,K++) {
            if (K >= orde) K = 0;
            deel_produkt *= *(matrix + R * orde + K);
        }
        determinant += deel_produkt;

        deel_produkt = 1.0;
```

```
        for (R = orde - 1,K = K1;R > = 0;R--,K + +) {
            if (K > = orde) K = 0;
            deel_produkt * = *(matrix + R * orde + K);
        }
        determinant - = deel_produkt;
    }
    return determinant;
}

/*************************************************************************
MATINV: Inverteert matrices
Aanroep: double *matrix, *inverse;
         int orde;
MATINV(matrix,inverse,orde);
Parameters:  matrix: pointer op de te inverteren matrix
             inverse: pointer op de matrix waarin de inverse komt
             orde: orde van de matrix
Returns:     0: singuliere matrix, of onvoldoende geheugen
             1: reguliere matrix
Bron: GEVOP 25-08-87 (C-version)
 *************************************************************************/

int MATINV(matrix,inverse,orde)

double *matrix,*inverse;
int orde;

{
    int I,R,K;
    int *MAX,*PIVOT_SET;
    double C;
    double *Lmatrix;
    double *Rmatrix;

    /* Get scratch space */

    MAX = (int *) calloc(orde,sizeof(int));
    PIVOT_SET = (int *)  calloc(orde,sizeof(int));
    Lmatrix = (double *)calloc(orde * orde,sizeof(double));
    Rmatrix = (double *)calloc(orde * orde,sizeof(double));

    if  (MAX = = NULL ||
        PIVOT_SET = = NULL ||
        Lmatrix = = NULL ||
        Rmatrix = = NULL ) {
            /* Foutafhandeling bij onvoldoende vrij geheugen */

            if (MAX ! = NULL) free(MAX);
            if (PIVOT_SET ! = NULL) free(PIVOT_SET);
            if (Lmatrix ! = NULL) free(Lmatrix);
            if (Rmatrix ! = NULL) free(Rmatrix);
            return 0;  /* Signaleer de foutconditie */
        }

        /* Initialisering
        - Pivot_set leegmaken
```

```
    - matrix kopieeren naar Rmatrix
    - in Lmatrix komt de eenheidsmatrix */

    for (R=0;R<orde;R++) {
    *(PIVOT_SET + R) = FALSE; /* PIVOT_SET = leeg */
    for (K=0;K<orde;K++) {
        *(Rmatrix + R * orde + K) = *(matrix + R * orde + K);
        if (R==K) *(Lmatrix + R * orde + K) = 1.0;
        else *(Lmatrix + R * orde + K) = 0.0;
    }
}

/* Pivoting */

for (K=0;K<orde;K++) {
    I = 0;
    while (*(PIVOT_SET + I) == TRUE) I++;
    *(MAX + K) = I;

    for (R=1;R<orde;R++)
        if (*(PIVOT_SET + R) == FALSE &&
            fabs(*(Rmatrix + R * orde + K)) >=
            fabs(*(Rmatrix + *(MAX + K) * orde + K))) *(MAX + K) = R;

            *(PIVOT_SET + *(MAX + K)) = TRUE;


    if (fabs(*(Rmatrix + *(MAX + K) * orde + K)) <= onder_grens) {
        /* Singuliere matrix */

        free(MAX);
        free(PIVOT_SET);
        free(Lmatrix);
        free(Rmatrix);
        return 0;
    }

    for (R=0;R<orde;R++)
        if (*(PIVOT_SET + R) == FALSE) {
            C = *(Rmatrix + R * orde + K) / *(Rmatrix + *(MAX + K) * orde + K);

            for (I=K;I<orde;I++)
                *(Rmatrix + R * orde + I) -= C * *(Rmatrix + *(MAX + K) * orde + I);

            for (I=0;I<orde;I++)
                *(Lmatrix  + R * orde + I) -= C * *(Lmatrix  + *(MAX + K) * orde + I);
        }
    }

    /* Berekening van de inverse  */

    for (K=0;K<orde;K++)
        for (R=orde-1;R>=0;R--)
            *(inverse + R * orde + K) = *(Lmatrix + *(MAX + R) * orde + K);

    for (K=0;K<orde;K++)
        for (R=orde-1;R>=0;R--) {
```

```
            for (I=orde-1;I>R;I--)
                *(inverse + R * orde + K) -= *(Rmatrix + *(MAX + R) * orde + I) *
                        *(inverse + I * orde + K);

            *(inverse + R * orde + K) /= *(Rmatrix + *(MAX + R) * orde + R);
        }

    free(MAX);
    free(PIVOT_SET);
    free(Lmatrix);
    free(Rmatrix);
    return 1;
}

/*************************************************************************
MATMUL: Vermenigvuldigt matrices en vectoren
Aanroep: double *matrix_1, *matrix_2, *produkt;
         double *MATINV();
         int R_1, K_1, K_2;
MATMUL(matrix_1,matrix_2,produkt,R_1,R_2,K_2);
Parameters:  matrix_1,matrix_2
             pointers op resp. de linker en de rechter matrix in de vermenigvuldiging.
produkt:     pointer op de matrix waarin het resultaat moet komen
             R_1: Aantal rijen in matrix_1
             K_1: Aantal kolommen in matrix_1
             K_2: Aantal kolommen in matrix_2
Returns:     Altijd TRUE
Bron: GEVOP  25-08-87 (C-version)
 *************************************************************************/

double *MATMUL(matrix_1,matrix_2,produkt,R_1,K_1,K_2)

double *matrix_1, *matrix_2, *produkt;
int R_1, K_1, K_2;

{
   int I,R,K;

   for (R=0; R<R_1; R++)
      for (K=0; K<K_2; K++)
         *(produkt + R * K_2 + K) = 0.0;

   for (R=0; R<R_1; R++)
      for (K=0; K<K_2; K++)
         for (I=0; I < K_1; I++) *(produkt + R * K_2 + K) += *(matrix_1 + R * K_1 + I) *
            *(matrix_2 + I * K_2 + K);
   return TRUE;
}

main()

/*************************************************************************

 Voorbeeld programma

 *************************************************************************/
```

```
{
    int i,j,orde;
    double matrix [10*10];
    double inverse[10*10];
    double vector [10];
    double result [10];

    printf("\nGeef de orde van de matrix : ");
    do
        scanf("%d",&orde);
    while (orde < 1 || orde > 10);

    printf("\n\nInvoeren van de matrix ..... \n\n");

    for (i=0;i<orde;i++)
        for (j=0;j<orde;j++) {
        printf("\nGeef element %2d,%2d : ",i+1,j+1);
            scanf ("%lf",(matrix + i * orde + j));
        }

    for (i=0;i<orde;i++) {
        for (j=0;j<orde;j++) printf("%10f ",*(matrix + i * orde + j));
        printf("\n");
    }

    printf("\nDe determinant van de matrix = %f \n\n",MATDET(matrix,orde));
    printf("\nDe inverse van de ingevoerde matrix is : \n\n");

    if (MATINV(matrix,inverse,orde) == FALSE)
        printf("De matrix is te groot of singulier\n");
    else {
        for (i=0;i<orde;i++) {
        for (j=0;j<orde;j++) printf("%10f ",*(inverse + i * orde + j));
            printf("\n");
    }
        printf("\n")
        printf("\nInvoeren van een vector met lengte %2d ..... \n\n",orde);

        for (i=0;i<orde;i++) {
            printf("\nGeef element %d : ",i+1);
            scanf ("%lf",&vector[i]);
        }

        MATMUL(inverse,vector,result,orde,orde,1);
        printf("De resultaat-vector is : \n\n");
        for (i=0;i<orde;i++) printf("%10f\n",result[i]);
    }
```

Fig. 1: matrixbewerkingen in C

In het volgende voorbeeld wordt een stelsel van vergelijkingen opgelost. Het volgende stelsel wordt opgelost:

A + B + C = 11

2A + 2B - 2C = -2

A - B - C = -9

Gevraagd A, B en C.

Om dit stelsel op te lossen, zetten we de coëfficiënten van A, B en C in een matrix. Deze matrix wordt geïnverteerd waarna deze geïnverteerde matrix met de vector van het resultaat van de vergelijkingen verminigvuldigd wordt. Het stelsel vergelijkingen wordt dus als volgt herschreven:

$$\begin{Bmatrix} 1 & 1 & 1 \\ 2 & 2 & -1 \\ 1 & -1 & -1 \end{Bmatrix} * \begin{Bmatrix} A \\ B \\ C \end{Bmatrix} = \begin{Bmatrix} 11 \\ -2 \\ -9 \end{Bmatrix}$$

Dit stelsel vergelijkingen is opgelost door middel van het bovenstaande programma:

Geef de orde van de matrix : 3

Invoeren van de matrix .....

Geef element 1, 1 : 1

Geef element 1, 2 : 1

Geef element 1, 3 : 1

Geef element 2, 1 : 2

Geef element 2, 2 : 2

Geef element 2, 3 : -2

Geef element 3, 1 : 1

Geef element 3, 2 : 1

Geef element 3, 3 : -1

1.000000	1.000000	1.000000
2.000000	2.000000	-2.000000
1.000000	-1.000000	-1.000000

De determinant van de matrix = -8.000000

De inverse van de ingevoerde matrix is :

0.500000	0.000000	0.500000
0.000000	0.250000	-0.500000
0.500000	-0.250000	0.000000

Invoeren van een vector met lengte 3 .....

Geef element 1 : 11

Geef element 2 : -2

Geef element 3 : -9

De resultaat-vector is :

1.000000
4.000000
6.000000

Als resultaat krijgen we dus:

$$\begin{Bmatrix} 1 & 1 & 1 \\ 2 & 2 & -1 \\ 1 & -1 & -1 \end{Bmatrix} * \begin{Bmatrix} 1 \\ 4 \\ 6 \end{Bmatrix} = \begin{Bmatrix} 11 \\ -2 \\ -9 \end{Bmatrix}$$

Nb. Om een dergelijk stelsel vergelijkingen op te lossen, is het niet strikt noodzakelijk de hele matrix te inverteren. Met behulp van de techniek die ook in MATINV gebruikt is, kan het stelsel ook in één keer opgelost worden. Omdat het hier echter een voorbeeld van het gebruik van meer algemene routines betreft, is gekozen voor de genoemde wijze van oplossen.

Het programmeren van de 8088 in de IBM (Deel 6)

De I/O van de 8088

Het is niet eenvoudig om een programmavoorbeeld te bedenken dat het gebruik van de I/O poorten in een IBM demonstreert. Vrijwel alles wat op dit gebied moet gebeuren wordt door DOS of toch tenminste door het BIOS verzorgd. Een ongelukkige uitzondering hierop vormt de ondersteuning van de seriele poorten. BIOS interrupt 14h die dit alles zou moeten verzorgen is volmaakt ongebufferd en daarom ongeschikt voor asynchrone communicatie. En helaas doet DOS niets anders dan deze interrupt aanroepen. Alle bestaande communicatie software stuurt dan ook de poorten direct aan of laat dit doen door een speciale driver. We zullen hier een eenvoudige methode laten zien die zeker niet volmaakt is. De ondersteuning van de poorten COM3 en COM4 blijft hier buiten beschouwing en ook andere belangrijke zaken zoals de XON/XOFF handshake moeten hier buiten beschouwing worden gelaten. Het programma geeft u een indruk hoe u een modem kunt besturen en u kunt ermee "inloggen" op een BBS. en daar rondkijken, uiteraard zonder files te kunnen ophalen.

Het hoofdprogramma "Terminal"

Eerst volgt het gebruikelijke initialiseren van het DS register en het teruggeven van overtollig geheugen aan DOS. Om niet in herhalingen te vervallen is het inlezen van parameters hier weggelaten. De parameters zoals het nummer van de communicatiepoort en de transmissiesnelheid zijn als geinitialiseerde variabelen gedefinieerd. Ook voor de transmissiemethode is uitgegaan van steeds meer gebruikelijke 1N8 oftewel: één stopbit, geen pariteit en 8 databits. Vervolgens wordt het scherm gewist en de naam van het programma afgedrukt. Daarna wordt de communicatiepoort geinitialiseerd, waarna het programma niets anders doet dan tekens van het toetsenbord naar het modem sturen welke deze op het scherm zal reflecteren. Daarbij zal het modem commando's herkennen en uitvoeren en indien aldus een verbinding met een ander systeem tot stand komt zal de gegevensstroom die dat systeem naar ons terugstuurt op het scherm verschijnen. Als we de toets ALT-X drukken wordt het programma afgebroken en verschijnt de DOS prompt weer.

BIOS interrupt 14h die dit alles zou moeten verzorgen is volmaakt ongebufferd en daarom ongeschikt voor asynchrone communicatie.

De poorten van de UART

Poort COM1 heeft als basisadres 03F8h. Voor poort COM2 is dat 02F8h. Denk eraan dat het hier gaat om POORT adressen en niet om adressen in het geheugen van de PC. De 8088 heeft twee instructies IN en OUT waarmee deze poorten kunnen worden benaderd. Bij gebruik daarvan krijgt de adresbus een andere functie en stuurt andere hardware aan. Zie voor details de serie "De IBM PC en zijn klonen" van Nico de Vries. Te beginnen op het basisadres hebben we toegang tot de diverse registers. We gebruiken hier behalve het basisadres (hier genoemd DATA) alleen:

IER	Interrupt Enable Register	Label: IRQEN
LCR	Line Control Register	LCTRL
MCR	Modem Control Register	MCTRL
LSR	Line Status Register	STAT

Zie voor de adressen de tabel in het programma.

Initialiseren van de poort (INITPORT)

De UART kan transmissiesnelheden tot 115200 bps aan. Dat komt overeen met een constante van 1 in de latches. We gaan bij het initialiseren uit van een initiele waarde van 384 voor 300 bps. Komt deze waarde niet overeen met de gewenste waarde, dan verdubbelen we de transmissiesnelheid en halveren de constante totdat de gewenste snelheid is gevonden. Op deze manier kan het programma 300, 600, 1200, 2400, 4800, 9600 bps aan. Voor de hogere snelheden is het zeker ongeschikt. De gevonden waarde moet in de latches opgeslagen worden. Daartoe wordt de UART in de commando modus gezet (Line Ctrl Register) waardoor op het basisadres de constante kan worden ingegeven. Met de instructie OUT kan een een byte of een word naar resp. een poort of een PAAR poorten worden gezonden. Aan het register AL of AX herkent de assembler of een 8 bits dan wel een 16 bits instructie wordt gebruikt. Het poortadres kan als constante (word) worden opgegeven of in DX worden geladen. Na het opgeven van de transmissiesnelheid wordt de commando modus weer uitgeschakeld en wordt het modem geinitialiseerd. Zie voor details de datasheet van de 8250.

Na het initialiseren van de poort op de gewenste transmissiesnelheid moet een interrupt-service rou-

tine aan de poort gehangen. Het juiste interruptnummer komt uit de tabel. Gebruik hiervoor altijd eerst DOS functie 35H om het adres van de originele routine te achterhalen en op te slaan. Dit moet bij het beëindigen van het programma uiteraard worden teruggezet. Gebruik DOS functie 25H om het adres van een eigen routine in de vector te zetten. DOS zorgt er voor dat dit veilig gebeurt. Hierna moet de 8259 nog worden geprogrammeerd om de interrupt van de poort toe te laten tot de processor. Her zien we een voorbeeld he met de IN instructie een poort wordt gelezen, een bit wordt veranderd en met de OUT instructie het byte weer wordt teruggeschreven. In dit geval is het poortadres een constante.

De interrupt service routine

Als data van de communicatiepoort wordt ontvangen is deze beschikbaar op het basisadres en veroorzaakt een interrupt. Hierdoor wordt het databyte verwerkt op het moment dat dat nodig is, zonder dat het programma voortdurend de poort zelf moet bewaren. Het ontbreken van deze faciliteit is er de oorzaak van de het gebruik van INT 14H of de DOS functies voor COM1 en COM2 alleen bruikbaar zijn als het programma al zijn aandacht richt op die poorten zelf. De routine OURIRQ gaat voorbij aan de mogelijkheid dat een interrupt niet afkomstig is van de gewenste poort. COM1 en COM3 gebruiken namelijk dezelfde interruptvector. Idem voor COM2 en COM4. Hierdoor is het onmogelijk om in dit voorbeeld COM1 en COM3 gelijktijdig te gebruiken. Veel software dateert overigens van voor de introductie van de twee nieuwe poorten dit bovendien pas in DOS 4.01 ondersteund werden. De constructie waarbij deze interruptvectoren door twee poorten worden gedeeld is dus, laten we maar zeggen "enigszins ongelukkig" gekozen. Als een interrupt optreedt is de inhoud can het DS register volkomen onbekend en moet dus worden bewaard, waarna DS naar ons eigen datasegment moet verwijzen. Behalve DS worden ook alle andere registers waarvan we gebruik maken bewaard. Het enige wat de interrupt service routine doet, is het databyte van de poort lezen en dit in een ringbuffer zetten waarna een EOI signaal aan de 8259 word gestuurd zodat deze een eventuele volgende interrupt kan gaan afhandelen.

De constructie waarbij deze interruptvectoren door twee poorten worden gedeeld is dus, laten we maar zeggen "enigszins ongelukkig" gekozen.

Data lezen (READBUF)

Als een of meer data bytes van het modem werden ontvangen, zal de leespointer een waarde hebben die ongelijk is aan de schrijfpointer. In dat geval wordt door deze routine het byte aangewezen door de lees pointer opgehaald en de pointer opgehoogd.

Data schrijven (SENDCOM)

Een data byte wordt rechtstreeks naar de poort geschreven. Voorwaarde is dat dit gebeurt als de poort op dat moment niet in gebruik is voor uitvoer. Dit is te zien aan bit 5 in het statusregister. Er moet dus worden gewacht tot dit bit is gezet.

Ruud Uphoff

Fig. 1: sourcetekst van TERMINAL.ASM

```
                NAME    Terminal

                ;Eerst onze luiheid weer bevredigen

OFS             EQU     OFFSET
BPT             EQU     BYTE PTR
WPT             EQU     WORD PTR

JMPS            MACRO TARGET
                JMP     SHORT TARGET
                ENDM
```

```
                ;Poortnummers van de interruptcontroller 8259A

PORT20H         EQU     20H             ;Commando register
PORT21H         EQU     21H             ;Masker register


                ;En de omvang van het buffer

BUFSIZE         EQU     1024            ;1Kb kan er wel mee door


                ;We gebruiken het .COM model, dus alles in een segment

CODE            SEGMENT

                ASSUME                  CS:CODE, DS:CODE, SS:CODE

                ORG     0100H

START:          JMP     TERMINAL

                ;Het ringbuffer met pointers

BUFFER          DB      BUFSIZE DUP (?) ;Ringbuffer
BUFPR           DW      0                               ;Lees pointer
BUFPW           DW      0                               ;Schrijf pointer

                ;We gebruiken COM2. Normaal zal dit nummer middels een
                ;parameter binnen komen. Hetzelfde geldt voor de
                ;transmissie snelheid.

COMPORT         DW      2                               ;1 voor COM1  2 voor COM2
SPEED           DW      2400                            ;2400 bps

                ;Poortadrestabel voor resp port 1 en port 2

DATA            DW      03F8H, 02F8H    ;Data registers
IRQEN           DW      03F9H, 02F9H    ;IRQ enable registers
LCTRL           DW      03FBH, 02FBH    ;Line control register
MCTRL           DW      03FCH, 02FCH    ;Modem control registers
STAT            DW      03FDH, 02FDH    ;Status registers
INTNUM          DB      0CH, 0BH        ;Interrupt nummers
MASKS           DB      10H, 08H        ;Te gebruiken maskers


                ;Overige variabelen

ORGVECTO        DW      (?)             ;De originele IRQ vector  (Offset)
ORGVECTS        DW      (?)             ;en segment

CURDATA         DW      (?)             ;Actuele adres van het data register
CURSTAT         DW      (?)             ;Actueel adres van het status register
```

```
                ;Mededeling bij het opstarten

HALLO           DB      'Eenvoudig modem terminal programma',13,10
                DB      '     Uit de µP-Kenner',13,10
                DB      10,'$'


                ;Interrupt 10h rekent soms grondig af met de hieronder
                ;veilig gestelde registers. Maar op deze manier hebben we
                ;daar geen omkijken meer naar:

INTR10          PROC    NEAR
                PUSH    DI
                PUSH    SI
                PUSH    BP
                PUSH    ES
                INT     10H
                POP     ES
                POP     BP
                POP     SI
                POP     DI
                RET
INTR10          ENDP


                ;Onze interrupt routine voor gebufferde ontvangst.

OURIRQ          PROC    NEAR
                PUSH    DS                          ;Registers opslaan
                PUSH    AX
                PUSH    BX
                PUSH    DX
                PUSH    CS              ;DS initialiseren !
                POP     DS
                MOV     DX,CURDATA      ;Poortnummer van data register in DX
                IN      AL,DX           ;en databyte lezen
                MOV     BX,BUFPW        ;Volgende vrije plaats in BX halen
                MOV     BUFFER[BX],AL   ;en byte daar in het buffer zetten
                INC     BX              ;Volgende vrije plaats in het buffer
                CMP     BX,BUFSIZE      ;maar indien einde buffer
                JB      NORING
                XOR     BX,BX           ;dan terug zetten op nul: Ringbuffer
NORING:         MOV     BUFPW,BX        ;Nieuwe bufferplaats nu opslaan
                MOV     AL,20H          ;Nu nog EOI naar de 8259 sturen
                OUT     PORT20H,AL
                POP     DX              ;Registers terug zetten
                POP     BX
                POP     AX
                POP     DS
                IRET                    ;en niet per ongeluk RET gebruiken!
OURIRQ          ENDP


                ;De COM poort initialiseren

INITPORT        PROC    NEAR
                MOV     BX,300          ;Initiele transmissie snelheid
```

```
                MOV     AX,180H         ;en tijdconstante voor 300 baud

TRYSPD:         CMP     BX,SPEED        ;Is dit de juiste snelheid
                JZ      SETSPEED        ;Zoja, dan gebruiken
                SHL     BX,1            ;anders snelheid verdubbelen
                SHR     AX,1            ;tijdconstante halveren
                JMPS    TRYSPD          ;en opnieuw testen

SETSPEED:       CLI                     ;Geen interrupts tijdens initialiseren
                PUSH    AX              ;Gevonden tijdconstante bewaren
                MOV     BX,COMPORT      ;Haal het poortnummer
                DEC     BX              ;Offset 0 voor poort 1 (Byte offset)
                MOV     DI,BX
                SHL     DI,1            ;Converteren naar word offset in DI

                MOV     DX,LCTRL[DI]    ;Poortadres van Line Ctrl reg. halen
                MOV     AL,83H          ;en de commando modus zetten
                OUT     DX,AL
                POP     AX              ;Tijdconstante ophalen

                MOV     DX,DATA[DI]     ;en nu het woord in een keer naar
                OUT     DX,AX           ; twee poorten schrijven

                MOV     DX,LCTRL[DI]    ;Commando mode uitschakelen
                MOV     AL,07H
                OUT     DX,AL

                MOV     DX,MCTRL[DI]    ;DTR initialiseren
                MOV     AL,0BH
                OUT     DX,AL

                MOV     AL,INTNUM[BX]   ;Het juiste interrupt nummer nemen
                PUSH    BX              ;en even onthouden
                PUSH    AX
                MOV     AH,35H          ;Die interrupt vector aan DOS opvragen
                INT     21H             ; in ES:BX
                MOV     ORGVECTO,BX     ; en opslaan
                MOV     ORGVECTS,ES
                POP     AX              ;Het nummer weer in AL
                MOV     DX,OFS OURIRQ   ;DS:DX naar onze routine laten wijzen
                MOV     AH,25H          ;en DOS de vector laten veranderen
                INT     21H
                POP     BX              ;De index weer terug halen

                MOV     AH,MASKS[BX]    ;Juiste masker ophalen
                NOT     AH              ;Bits inverteren
                IN      AL,PORT21H      ;en IRQ controller initialiseren
                AND     AL,AH
                OUT     PORT21H,AL

                MOV     DX,IRQEN[DI]    ;Interrupts doorlaten naar CPU
                MOV     AL,1
                OUT     DX,AL
                STI                     ;En weer interrups toestaan

                MOV     AX,DATA[DI]     ;Haal actueel adres van data register
                MOV     CURDATA,AX      ;en sla het op voor gebruik
```

```
                MOV     AX,STAT[DI]     ;Idem voor het statusregister
                MOV     CURSTAT,AX

                RET
INITPORT        ENDP


                ;Teken afdrukken op het scherm. Via DOS (Dus: ANSI.SYS)

WRITECH         PROC    NEAR
                PUSH    AX              ;Gwoon via DOS functie 02H
                MOV     DL,AL
                MOV     AH,02H
                INT     21H
                POP     AX
                RET
WRITECH         ENDP


                ;Teken ophalen uit het buffer

READBUF         PROC    NEAR
                MOV     BX,BUFPR        ;Lees pointer in BX
                MOV     AL,BUFFER[BX]   ;en teken aldaar in AL halen
                INC     BX              ;Lees pointer verhogen
                CMP     BX,BUFSIZE      ;maar indien einde buffer
                JB      NORNG           ;terug op 0 zetten
                XOR     BX,BX
NORNG:          MOV     BUFPR,BX        ;Nieuwe lees pointer opslaan
                RET
READBUF         ENDP


                ;Teken verzenden

SENDCOM         PROC    NEAR
                MOV     AH,AL           ;Teken bewaren in AH
                MOV     DX,CURSTAT      ;Poortadres van statusregister in DX
WAITS:          IN      AL,DX           ;en wachten tot "klaar voor ontvangst"
                AND     AL,20H
                JZ      WAITS
                MOV     AL,AH           ;teken weer in AL
                MOV     DX,CURDATA      ;Poortadres van het dataregister in DX
                OUT     DX,AL           ;en teken verzenden
                RET
SENDCOM         ENDP


                ;----------------------------------------
                ;       Het hoofdprogramma
                ;----------------------------------------


TERMINAL        PROC    NEAR

                ;Eerst de stack poiter initialiseren....
```

```
        CLI                         ;Geen interupts! anders....
        MOV     AX,CS               ;SS en DS is niet nodig
        MOV     SS,AX               ;maar wel handig bij debuggen
        MOV     DS,AX               ;van een .COM als een .EXE

        MOV     BX,OFS ENDSTACK                 ;Dit moet dus wel!
        MOV     SP,BX
        STI
        PUSH    AX                  ;Alleen maar om SP met 2 te verlagen

        ;En nu het overtollig geheugen teruggeven aan DOS

        MOV     CL,4                ;Einde programma DIV 16
        SHR     BX,CL
        INC     BX                  ;plus eem paragraaf ter afronding
        MOV     AH,49H              ;is wat we willen overhouden.
        INT     21H

        ;Scherm wissen en netjes "hallo" zeggen

        MOV     AX,0600h            ;Functie "scroll up". 0 regels = wissen
        MOV     BH,07H              ;In gewoon zwart/wit
        XOR     CX,CX               ;links boven is 0,0
        MOV     DX,184FH            ;rechts onder is 79,24
        CALL    INTR10
        XOR     DX,DX               ;Corsor naar positie 0,0
        XOR     BH,BH               ;op pagina 0
        MOV     AH,02H
        CALL    INTR10
        MOV     DX,OFS HALLO        ;De "hallo" string afdrukken
        MOV     AH,09H
        INT     21H

        CALL    INITPORT            ;Comport initialiseren
        PUSH    BX                  ;Die index in BX bewaren


        ;Daar gaat ie dan

TERMLOOP: MOV   BX,BUFPR            ;Lees de buffer pointer
        CMP     BX,BUFPW            ;Teken in buffer?
        JZ      TRYKEY
        CALL    READBUF             ;dan ophalen
        CALL    WRITECH             ;en afdrukken
        JMPS    TERMLOOP            ;en opnieuw proberen

TRYKEY: MOV     AH,01H              ;Teken van het toetsenbord?
        INT     16H
        JZ      TERMLOOP            ;Ook niet? Opnieuw proberen dan

        XOR     AH,AH               ;Zoja, dan lezen we het
        INT     16H
        CMP     AX,2D00H            ;Code voor Alt-X ?
        JZ      EXIT                ;Dan einde van dit avontuur
        CALL    SENDCOM             ;Anders teken naar het modem sturen
        JMPS    TRYKEY              ;en alles begint opnieuw
```

```
                ;Terug naar DOS, dus de zaak netjes achterlaten

EXIT:           POP     BX              ;Wie wat bewaard heeft wat
                IN      AL,PORT21H      ;Interupt controller gedag zeggen
                OR      AL,MASKS[BX]
                OUT   PORT21H,AL

                PUSH    DS              ;Interrupt vector terug zetten
                MOV     AL,INTNUM[BX]
                MOV     AH,25H
                MOV     DX,ORGVECTO
                MOV     DS,ORGVECTS
                INT     21H
                POP     DS
                MOV     AX,4C00H        ;En terug naar DOS
                INT     21H

TERMINAL        ENDP


                ;De stack

                EVEN                    ;Vooral een 8086 is hier blij mee!

STACKSPACE      DW      100H DUP (0)
ENDSTACK        EQU     $

CODE            ENDS

                END     START
```

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
Postbus 1336
7500 BH Enschede

Voortgang KGN-68k

Kristalvorming

De scheikundigen onder ons kunnen wel vertellen onder welke omstandigheden kristalvorming optreedt. Bij de KGN-68k gebeurt het uitkristalliseren onder invloed van zomerweer en vakantie periodes en vooral door enthousiaste hobbyisten zoals Ad Brouwer en Willem Oldeman.

Opgesplitst

Bij schemaversie 0.4 kwam het probleem dat een full size AT-kaart te klein is voor het aantal componenten op dat moment in het ontwerp zat. De mogelijkheden zijn dan:

- Print Circuit Board groter maken
- Componenten weglaten
- Kleinere componenten

Als je ze gaat uitwerken:

Een grotere print ligt het meest voor de hand, want die is gewoon te klein! De kast die je op een of andere beurs op de kop getikt had, is dan weer te klein. Het los laten van deze standaard afmetingen was eigenlijk toch ook niet de bedoeling. Zitten er dan te veel componenten in het ontwerp? Nee, toch niet, wat er op papier stond is toch echt nodig voor het uiteindelijke systeem. Er zit trouwens wel veel programmeerbare logica op het te bouwen board. Deze chips waar de inhoud ZELF van te definieren is zitten behoorlijk vol, bovendien vervangen ze een berg logische poorten uit de 74xx00 serie.

Een grotere print ligt het meest voor de hand, want die is gewoon te klein!

Kleinere componenten is ook een goed idee. Waar mogelijk gebruik je SMD-componenten, maar het is een zelfbouwproject. Er zullen weinig KGN-leden zijn een SMD soldeeroven hebben en in de magnetron mag je geen metalen voorwerpen ontdooien/verwarmen/solderen.

Hoe zijn we toen verdergegaan? Er is besloten om toch het printplaatoppervlakte te vergroten door het ontwerp op te splitsen in twee PCB's. Maar waar moet dan de scheiding getrokken worden?

Twee printen

Het merendeel van de ingredienten van het "Brouw"sel bleken te bestaan uit het moeilijkste stuk, de interface naar de AT. Heel veel buffers en control logic die hiervoor nodig is. Om nu te zeggen dat de helft van het ontwerp uit de koppeling 68030/AT bestond is wat overdreven, maar het kwam een heel eind in de buurt. Om de scheiding op deze plaats te leggen had/heeft de volgende voordelen:

+ Moeilijker stuk is apart te ontwerpen
+ Testen/repareren van duidelijk afzonderlijke stukken is eenvoudiger

En natuurlijk de algemene voordelen:

+ Overzichterlijker ontwerp
+ Minder complexe print (=goedkoper)

Er wordt nu dus ontworpen een 68030 (schema & PCB) systeem en een interface 68030/ATbus (schema). De KGN-68k werkgroep blijkt dus ineens met twee projecten bezig te zijn. De hardware mensen in de werkgroep kunnen zich wel uitleven, maar de software mensen kunnen nog niet echt aan de slag.

Consequenties

Onder tijdsdruk werd besloten om de hardwaregroep zich te laten concentreren op het processor gedeelte van het ontwerp. De consequentie hiervan is dat de stand-alone 68030 dan wel een disk-interface moet hebben.

Minimaal was dit een floppy disk controller. Een harddisk op dezelfde print als de processor geeft een veel betere performance dan wanneer deze communicatie via allerlei bussen moet verlopen. Voor een disk-geïntegreerd systeem MINIX is dit eigenlijk een zeer interessante optie! En zo waren de gevolgen van het splitsen dat er twee diskcontrollers en bijbehorende componenten bijkwamen.

IDE/SCSI

Een andere afweging die er geweest is ging tussen IDE & SCSI controllers. In prijs, en zeker bij grotere disks, doen ze niet voor elkaar onder. Doorslaggevend was het feit dat SCSI-controller gemakkelijker aan een achtenzestig duizend te koppelen is, dan een IDE-controller want die gaat uit van ATbus signalen.

DMA/FIFO

Er grote componentenbesparing van het ontwerpgenie Ad Brouwer was om de DMA controller met alle bijbehorende chips te vervangen door een First In First Out register. De principiële werking van FIFO schakeling houden jullie van mij te goed. Voor de software bouwers betekent dit dat er geen DMA controller geïnitialiseerd hoeft te worden.

Motherboard

Er zou de vraag kunnen zijn waarom we geen motherboard met een 68030 maken? (Tijdens de bestuursvergadering van juni werd er ook wel "moederkoek" tegen een motherboard gezegd). Op dit Printed Circuit Board zou dan ook al meteen de complexe ATbus logica mee ontworpen moeten worden. Het ontwerp voordeel dat we nu hebben is dan geheel verdwenen. Bovendien is het duurder een grote print opnieuw aan te laten maken dan een veel kleinere. Een motherboard past trouwens ook niet op de bevestigingsplaatsen van de diverse AT-kasten, met een insteekkaart is dat probleem er niet.

AT als I/O-processor

Een andere vraag zou kunnen zijn waarom we geen gebruiken van de resource in een AT? De voordelen op het eerste gezicht zijn:

+ goedkoop
+ staat er toch al

Even vooraf, twee fysiek gescheiden systemen heeft zeker zijn charmes. Het ene moment werk je met die ene hobby computer, het andere moment met die kantoormachine die je overdag al op het werk/school zag. Het goedkope is inderdaad een sterk argument. Hoeveel je op de voeding bezuinigt is ons niet bekend want de stroomopname van de KGN-68k is ons op dit moment nog niet bekend. Mooi dat er toch al een AT staat, dan hoef je niet meer op zoek naar een terminal of een RS232 interface voor je Commodore 64 (zelfbouw f25,-). Maar goed laten we er van uitgaan de AT alle Input & Output operaties doet. In hardware betekent dit een insteekkaart met een stuk dual ported RAM. De harddisk zul je wel eerste op moeten ruimen om plaats te maken voor het extra operating system. Om MINIX te kunnen booten moet er eerst tegen de AT gezegd worden dat er een kaart is die wat opdrachten voor hem heeft, eigenlijk een eenmalige handeling maar toch. Als de KGN-68k wat van disk wil hebben, een veel voorkomende vraag van MINIX, dan wordt eerst een message naar de I/O gestuurd. Deze boodschap wordt dan gelezen en de data wordt naar het AT geheugen gehaald. Er gaat een bericht naar de KGN-68k dat er data bij de 80286 is en dat er met de overdracht kan worden begonnen. De 68030 kan deze gegevens dan weer verwerken totdat de Motorola chip weer gegevens nodig heeft van de disk die onder beheer staat van de Intel chip. Een RAM-disk in het lineaire geheugen van de 80286 is een hele performance-verbetering, maar toch moet de informatie, via de ATbus, nog steeds door de dual ported RAM. Heel aannemelijk is het dat er één of andere I/O kaart in het systeem zit. Deze kaart zal nooit rechtstreeks aangesproken kunnen worden door de "nul dertig". De I/O routines zul je met een Intel compiler/assembler moeten maken, maar waar heb je dan die Motorola compiler/assembler voor? En waar ben je dan mee bezig? Met je zelf gebouwde 68030 of toch met een 80286?

Software

Het eerste programma is er al. Zie hiervoor de programmalisting. Getest is het nog niet, want het is eigenlijk om de hardware te testen. Deze paar regels assembly code maken het de hardware ploeg mogelijk om de processor, oscillator, decodeerlogica, data acknowledge circuit en EPROM te controleren. De

```
list bt
Microware OS-9/68000 Resident Macro Assembler V1.9 91/07/26 00:17 Page 1
bt.a
BasicTest - Test the most basicparts
                        nam         BasicTest
                        ttl         Test the most basicparts

80000000    stack       equ         $80000000       inital stackpointer

                                    psect           bt,$8000,0,0,0,adres0

0000 8000   adres0      dc.l        stack
0004 0000               dc.l        myself          initial program counter

0008 60fe   myself      bra.s       myself          endless loop, a signal generator

0000000a                ends
Errors: 00000
Memory used: 25k
Elapsed time: 0 second(s)
```

voedingsspanning wordt uiteraard van te voren gecontroleerd. De volgende stap is dan een bestaande & werkende "monitor" die momenteel overgezet wordt naar onze hardware. De software ploeg is ondertussen ook volop bezig met het verzamelen van kennis, moed, tijd en energie voor de daadwerkelijk MINIX port.

Reëel

Tot zo ver de status van het KGN-68k project. Vind je dat we nu eindelijk reëel bezig zijn, laat het ons dan weten. Overigens, fanatieke hobbyisten mogen ook weleens tegenwind hebben. Ik bedoel maar alle suggesties zijn welkom.

Geert Stappers (04788-1279)

Versienummers

U kent dat wel denk ik. De run op de nieuwste versie. Dat gaat meestal ongeveer zo: "Heb jij versie 9.8xyz van WeurdPeurfuct al?" "Nee?" "Loop je zwaar achter joh!" "Hebben?"
Enzovoort. Vroeger was ik ook zo. Het hoogste goed stond gelijk met het hoogste versienummer. Tegenwoordig wordt dat heel wat minder. Waarom? Lees eens verder.

PCTools

Het begon met PCTools. Een pakket dat eigenlijk iedereen wel op zo'n machine heeft staan. Als ik het goed heb, is net versie 7.0 uit. Prachtig mooi. Compleet met Shell, grafische gebruikersinterface, muisbesturing, viewers waarmee je zelfs bestanden kunt bekijken waarvan het formaat nog uitgevonden moet worden, vurschrukkulluk uitgebreide utilities om de schijf te testen en te editen en nog veeeel meer. Als je het gaat installeren ben je doodleuk 2 Mbyte of meer schijfruimte kwijt. Dat is 10 procent van de capaciteit van mijn harde schijf. Ondergetekende werkt echter niet met PCTools V7.0. Ook niet met 6.0. Of met 4.11. Ook niet met versie 2.00. Wel met versie 1.10. Al zo'n 4 jaar oud. Doet alles wat ik nodig heb: sector editor, undelete, search utility, directories sorteren en rommelen aan de attribuutbits, de tijd en datum van files. Dit alles in 1 enkel programmaatje, dat compleet 84 kilobytjes in beslag neemt.

NORTON

Ander voorbeeld. Norton utilities. Vroeger een uiterst zinvol pakket, dat met alles erop en eraan op 1 360k floppy paste. Liet je de franje weg, dan was het minder dan 100k. Deed alles wat een normaal gezond denkend mens nodig had. Maar het moest anders. Norton Utilities 5.0 is ook een paar Megabijtjes groot. Neem nou bijvoorbeeld SI. Iedere idioot gebruikt dat programma om vast te stellen dat de eigen machine veel te langzaam, en die van de buurman stukken sneller is. SI geeft ook nog wat zinvolle informatie over hoeveel geheugen je hebt en nog wat van die pret. Gewoon handig. 1 schermpje vol met alles wat je nodig hebt. In NU 5.0 gaat dat anders. SI geeft nu zo'n 30 schermen vol info over je machine. Reuze interessant, maar ik wilde alleen maar weten of ik een EGA of VGA mode had ingeschakeld. Conclusie: Norton 4.00 was mooi genoeg.

WordPerfect

Terecht de standaard tekstverwerker geworden. De gebruikersinterface is reuze onhandig, want ik kan na 4 jaar werken met WP nog altijd niet zonder sjabloon op het toetsenbord. De rest van de wereld vindt dat kennelijk niet zo'n probleem. Versie 4.0 had een paar blunders. 4.1 was beter. 4.2 (Engels alstublieft, bij WP Europe kunnen ze namelijk wel tekstverwerken, maar absoluut niet vertalen) was inderdaad (Word)Perfect. Toen kwam 5.0. Zaten minstens zoveel fouten in als dat het kilobytes (pardon, Megabytes) groot was. 5.1 was een stuk beter. Alleen is het geen tekstverwerker meer. Maar het is ook geen DTP pakket. 4.2 was goed genoeg voor mij.

MS/PC-DOS

Laatste voorbeeld. Versie 3.30 heeft het lang volgehouden: bijna twee jaar. Was goed, en compact: past op anderhalve 360k flop. Eergisteren kennisgemaakt met MS-DOS 5.0. Biedt niets nieuws, maar is wel even 4 maal zo groot. Want ik gebruik geen shell. En een XT heeft geen extended memory, dus meer vrij voor de applicatie zit er ook niet in. Undelete had ik al in PCTools 1.10. Unformat heeft een normaal functionerend persoon niet nodig: "God straft onmiddellijk" luidt hier het devies. Verder werkt de helft van de mij zo vertrouwde utilities niet, want ze hebben weer een nieuw soort partities uitgevonden. CLS werkt op mijn systeem alleen met ANSI.SYS actief. En het DOS is duidelijk trager. Ergo: ik blijf wel met 3.30 draaien. Met 4DOS erbij.

Tendens

Er valt een tendens waar te nemen. Als het versienummer boven 4.00 komt, gaat er iets mis. Overkomt u dat nou ook, of ben ik een beetje gestoord aan het raken?

Nico de Vries (V1.0)

Methoden en technieken voor datacommunicatie (Deel 8)

Inleiding

Ik ben na de vorige aflevering op mijn vingers getikt door een collega clublid vanwege het feit dat ik dingen geschreven heb die (nog) niet mogelijk zijn. Ik heb het gehad over een PC-netwerk waarbij je vanaf je eigen knooppunt een programma op kunt starten op een ander knooppunt (de 80386 van je baas). Welnu met de nu bestaande PC-netwerken is dat nog niet mogelijk. Zelf werk ik met een netwerk van VAXen van de firma DEC (DECnet dus) en daar zijn dergelijke dingen eigenlijk de gewoonste zaak van de wereld. Maar ja, wat niet is kan nog komen en naar mijn stellige overtuiging zal het ook nog allemaal komen.

Verder wil ik langs deze weg ook Jan-Pascal van Best bedanken voor de informatie over Novell en IPX die hij mij gestuurd heeft. Ik ben er van overtuigd dat we er iets aan zullen hebben.

Netwerken bij MS-DOS en OS/2 computers

Voor de PC-compatible computers zijn er meerdere netwerksystemen beschikbaar. Voorbeelden hiervan zijn onder andere Novell NetWare en Microsoft OS/2 LAN-Manager. Met behulp van dit soort producten kunnen netwerken van twee of meer PC's worden opgebouwd. Binnen een dergelijk netwerk worden dan één of meer zogenaamde (file-) servers gedefinieerd waarop de gebruikers van het netwerk in kunnen loggen. Na het inloggen kunnen de gebruikers gebruik maken van de diensten van de server zoals bijvoorbeeld de harde schijf. Hiervoor kan de netwerkbeheerder per directory rechten aan gebruikers toekennen om files aan te kunnen maken, te kunnen openen voor lezen en/of schrijven en uit kunnen wissen. Kortom de gebruiker kan na het inloggen op de server gebruik maken van de informatie die op de server aanwezig is, mits de netwerkbeheerder dit goed vindt. Zo kan men dus op de server de nieuwste versie van het tekstverwerkingspakket houden die de gebruikers vanaf de server inlezen en op hun eigen systeem draaien. Verder kunnen de gebruikers hun informatie op de server opslaan zodat er slechts van één systeem geregeld (dagelijks) een backup gemaakt hoeft te worden. Meestal is het dan zo geregeld dat de netwerkbeheerder zorgt voor de backups van de server en de gebruiker voor die van zijn eigen systeem.

Behalve een harde schijf zijn er in een netwerk meestal ook één of meer centrale printers zodat alle gebruikers van het netwerk deze printer kunnen gebruiken voor het afdrukken van teksten. Ook kunnen er in het netwerk zogenaamde gateways naar andere netwerken van bijvoorbeeld mainframes worden opgenomen. Via een dergelijke gateway kan men verbinding zoeken met de onderdelen van dat andere netwerk. Tenslotte kunnen de gebruikers elkaar berichtjes toesturen. Dit is met name van belang als de netwerkbeheerder het netwerk even uit wil schakelen. Hij kan dan een bericht versturen in de trant van "Over 2 minuten gaat het netwerk even plat... Uitloggen s.v.p.".

Hij kan dan een bericht versturen in de trant van "Over 2 minuten gaat het netwerk even plat... Uitloggen s.v.p.".

Ontwikkelaars van software-pakketten kunnen uiteraard de faciliteiten die een netwerk biedt gebruiken in hun producten. Voorbeelden hiervan vindt je onder andere een aantal database pakketten die het zogenaamde client-server model ondersteunen. Hierin is de client uiteraard de gebruiker of, gezien vanuit het netwerk het programma dat op de PC van de gebruiker draait. Deze client geeft opdrachten (transacties) door aan de server die vervolgens deze transacties in de door hem beheerde database aanbrengt of een foutmelding geeft. Het standaard-voorbeeld van een dergelijke transactie komt helaas niet uit een PC-netwerk maar is zo algemeen van aard dat het zinvol is dit voorbeeld in de volgende paragraaf te behandelen.

Het client-server model van een geldautomaat

Een geldautomaat is één van de schoolvoorbeelden van een centrale database op een file-server (sorry, een database-server) en een groot aantal verspreide systemen, de geldautomaten, die als client optreden. De afhandeling loopt als volgt:

1. Een giropas-bezitter komt bij de geldautomaat en steekt zijn pas in de gleuf. De geldautomaat detecteert dit en vraagt de gebruiker zijn PIN-code (Persoonlijk Identificatie-Nummer) in te toetsen. Van de pas wordt bovendien het pasnummer gelezen.
2. De geldautomaat vraagt aan de gebruiker welk bedrag hij op wil nemen. Verder vraagt de automaat tegenwoordig of de gebruiker een zogenaamde "bon" wil.
3. Nadat alle gegevens gevraagd zijn, wordt er (één maal !) contact gezocht met de centrale

server. Deze krijgt van de geldautomaat het gelezen pasnummer, de PIN-code en het gevraagde bedrag.

4 De server krijgt de opdracht van een bepaalde geldautomaat en zal de volgende transactie starten:
   - Controleer pasnummer en PIN-code, bepaal het bijbehorende rekeningnummer. Breek bij fouten de transactie af met een foutmelding.
   - Maak melding van de transactie in de meelopende transactielogging.
   - Boek het gevraagde bedrag af van het saldo van het bepaalde rekeningnummer. Mocht dit saldo negatief worden breek dan de transactie met een foutmelding af.
   - Stuur het resultaat van de transactie (foutmelding of transactie-OK code) terug naar de geldautomaat.

5 De geldautomaat gaat vervolgens tot uitbetaling over, drukt eventueel een bonnetje af en wenst de gebruiker "Tot ziens".

De tijd tussen 3 en 5, de tijd die nodig is om het zoeken van contact met de server en het uitvoeren van de transactie op de server, kan vrij lang duren. Als er namelijk van tientallen geldautomaten gelijktijdig een transactie binnenkomt, dan heeft de server het even behoorlijk druk. Boze tongen hebben mij eens verteld dat de geldautomaat gedurende deze tijd een kunstmatig signaal genereert dat veel lijkt op het uittellen van geld (flap, flap, flap...). Dit zou gedaan zijn om de gebruiker de indruk te geven dat de machine bezig is de gegeven opdracht uit te voeren zodat hij niet in paniek raakt omdat er schijnbaar niets gebeurt. Of dit verhaal correct is weet ik niet. Misschien is er een clublid die dit kan bevestigen of ontkennen?

De meeste mensen tikken de juiste PIN-code in...

Het uitvoeren van transacties op de manier die bij de geldautomaat beschreven is komt steeds vaker voor. Systemen die op een dergelijke manier werken noemt men transactie verwerkende systemen of OLTP systemen van OnLine Transaction Processing. In het bovenstaande voorbeeld zitten bepaalde kenmerken die het systeem tot een transactie verwerkend systeem maken. In de eerste plaats is dat het feit dat de transactie meteen (Online) uitgevoerd wordt. Het saldo op de rekening wordt meteen bijgewerkt. In de tweede plaats wordt er in stap 4 een transactie gestart die uit een aantal onderdelen bestaat. De geldautomaat vraagt eerst alle gegevens aan de gebruiker die nodig zijn om de transactie uit te voeren en pas daarna worden de gegevens verwerkt. Een alternatief zou zijn:

- Laat de gebruiker zijn PIN intoetsen en lees het pasnummer van de kaart.
- Bepaal het rekeningnummer en controleer bij de centrale computer of de PIN klopt. NB. uit veiligheidsoverwegingen wordt de PIN niet lokaal in de geldautomaat gecontroleerd.
- Laat de gebruiker intoetsen hoeveel geld hij op wil nemen.
- Controleer of het saldo toereikend is, werk bovendien de logging bij en boek het bedrag van het saldo af.
- Geef de geldautomaat opdracht tot uitbetaling over te gaan.

De verschillen tussen de transactie verwerkende aanpak en de bovenstaande beschrijving lijken gering maar hebben verregaande consequenties. In de eerste plaats is er in het tweede geval twee keer een communicatie over het netwerk in elke richting terwijl dit in het eerste geval slechts één keer is. Het tweede verschil is echter veel fundamenteler. Bij de eerste transactie wordt er per definitie vanuit gegaan dat PIN-code en gevraagd bedrag correct zijn. In het tweede geval wordt er vanuit gegaan dat het wel fout zal zijn en wordt de volgende stap niet eerder gezet dan nadat de vorige gecontroleerd is. Aangezien het merendeel van de transacties wel goed zal zijn (de meeste mensen tikken de juiste PIN-code in...) is de eerste aanpak veel efficienter. Natuurlijk moet je dan wel de mogelijkheid hebben een transactie af te breken en alle gegevens van voor de start van de transactie terug kunnen zetten. Het is namelijk niet de bedoeling dat de transactielogging bijgewerkt wordt als de transactie afgebroken wordt omdat het saldo niet toereikend is. In een dergelijk geval wordt er een zogenaamde "roll back" uitgevoerd waarbij de gegevens teruggezet worden. Gaat de transactie wel goed, dan wordt er een zogenaamde "commit" uitgevoerd waarbij de wijzigingen definitief gemaakt worden.

Uiteraard kunnen het client-server model en transactie verwerking ook gebruikt worden op systemen die niet een netwerk hangen. Voor client-server heb je dan wel multitasking nodig waarbij het ene proces als client werkt en de andere als server. De moderne database pakketten hebben tegenwoordig allemaal mogelijkheden tot transactie verwerking. Wel is het zo dat netwerken optimale mogelijkheden bieden tot transactie verwerking en client-server. Een server in een netwerk heet tenslotte niet voor niets een server.

Netwerken opgebouwd uit lagen

Goed, we hebben het gehad over wat praktijkgevallen, laten we ons nu weer wat meer met de theorie van netwerken gaan bezighouden. In de vorige aflevering heb ik het al even gehad over de fysieke opbouw van een netwerk. Daarin is uitgelegd dat voor het netwerk een bepaalde structuur (ster, ring, bus etc.) gebruikt wordt. Voor de bekabeling tussen de knooppunten wordt gebruik gemaakt van coax-kabel, glasvezel of twisted pairs.

Bij PC-netwerken komen twee soorten veelvuldig voor. Dat is in de eerste plaats natuurlijk Ethernet. Een tweede netwerk is ARC-net. Ethernet bestaat er in twee vormen, één met een dikke kabel (thick Ethernet en één met een dunne kabel (thin Ethernet). De dikke kabel heeft meestal een gele kleur en lijkt het meeste op een tuinslang. De dunne ethernet kabel lijkt het meeste op een gewone coax-kabel uit een electronica werkplaats, heeft meestal een rode kleur en is voorzien van BNC-connectoren zoals die ook in electronische werkplaatsen aangetroffen wordt. De impedantie van een ethernet kabel is 50 ohm. Aftakkingen van een dikke ethernetkabel worden gemaakt door middel van een klem met daarin een precisie schroef. Bij indraaien raakt deze schroef precies de binnenader van de coax-kabel. Deze situatie is schematisch getekend in figuur 1.

Bij een dunne kabel wordt voor aftakkingen gebruik gemaakt van min of meer normale T-stukken met BNC-connectoren. In één netwerk kunnen thin en thick ethernet tot op zekere hoogte door elkaar gebruikt worden. De maximale afstand die met thin ethernet overbrugd kan worden ligt in de orde van zo'n 250 meter; met behulp van thick ethernet kunnen afstanden tot zo'n 1000 meter overbrugd worden. De snelheid van een ethernet-verbinding bedraagt 10 Megabit per seconde. Bij Ethernet wordt er gebruik gemaakt van Manchester encoding waarbij een 0 wordt gerepresenteerd door een spanning van -0.85 V gevolgd door een spanning van +0.85 V. Bij een 1 is dit net andersom. Staat er geen signaal op de kabel, dan is de spanning op de kabel constant 0 V.

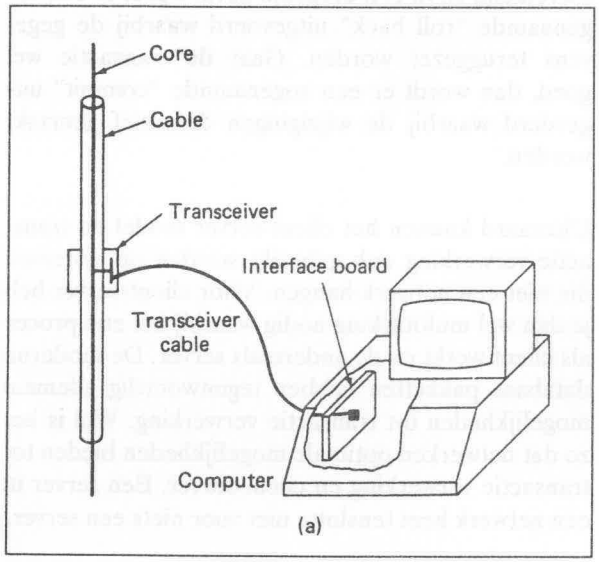


Fig. 1: aansluiting van een computer op een Ethernetkabel

Het tweede netwerk dat voor PC-netwerken in gebruik wordt, is het zogenaamde ARC-net. Dit netwerk doet het allemaal wat rustiger aan en heeft een maximale overdrachtssnelheid van 2,5 Megabit per seconde. Het netwerk zelf is opgebouwd met behulp van RG-62 A/U coaxkabel met een impedantie van 93 ohm en voorzien van normale BNC-connectoren. Het netwerk werkt met spanning van -5 V en +5 V. De topologie van het netwerk hangt af van de situatie. Er zijn netwerken die opgebouwd zijn in een bus-structuur waarbij er voor elk station een aftakking gemaakt is van een centrale kabel. Een tweede mogelijkheid is een stervormige opbouw van het netwerk waarbij een aantal kabels uitkomen op een zogenaamde "Hub" die deze kabels met elkaar verbindt. Het belangrijkste is echter dat alle kabels hetzelfde signaal voeren. De afstanden die overbrugd kunnen worden zijn zo'n 600 meter van knooppunt tot de Hub; totaal dus 1200 meter in een stervormig netwerk.

Als ik even het OSI-model uit aflevering 4 van deze serie in de herinnering roep, dan is de bovenstaande beschrijving die van level 0 (medium) en level 1 (physical layer) uit het OSI-model. Gaan we een stapje omhoog in het model, dan komen we bij de datalink layer. Hier zitten de eerste wezenlijke verschillen tussen ARC-net en Ethernet. ARC-net werkt namelijk met behulp van een token passing protocol terwijl Ethernet met behulp van CMSA/CD werkt. Deze twee protocollen zijn in vorige aflevering al heel even aan bod geweest. Ik denk echter dat het nu tijd is voor een iets uitgebreidere beschrijving.

Laten we eerst even Ethernet bij de kop pakken. Binnen Ethernet wordt de informatie overgestuurd in de vorm van packets met een opbouw zoals die in figuur 2 getekend is. Dit packet bestaat uit een kop van 64 bits waarvan de eerste 7 bytes als inhoud 101010 hebben. Dit genereert een 10 MHz blokgolf op de kabel en kan gebruikt worden voor synchronisatie. Het laatste byte van de kop bevat 101011 en is het scheidingsteken tussen de kop en het volgende veld. Het tweede veld heeft een lengte van 48 bits en bevat het netwerk-adres van de afzender. Evenzo het derde veld dat het netwerkadres van de geadresseerde bevat. Het derde veld bevat de lengte van het dataveld, opgegeven in bytes, in twee bytes. Vervol-

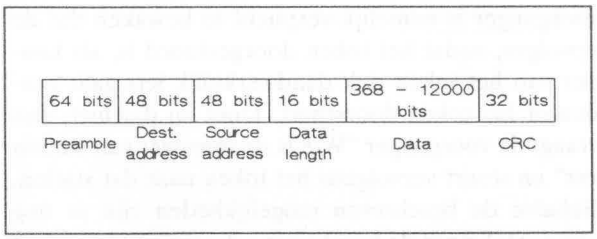


Fig. 2: opbouw van een Ethernetpacket

gens komt het dataveld dat een lengte heeft van 368 tot 12000 bits, afhankelijk van de lengte. Een datalengte van 0 bytes is toegestaan, om echter toch een minimale totale lengte van 64 bytes te hebben worden in gevallen waarin de lengte onvoldoende is maximaal 46 zogenaamde pad-tekens in het dataveld opgenomen. Aangezien de lengte wel op de juiste waarde ingesteld staat worden deze tekens door de ontvanger genegeerd. Het laatste veld bevat een 32 bits CRC om de inhoud van het packet te kunnen controleren.

Als een station een packet wil versturen, dan moet de lijn uiteraard vrij zijn. Het station gaat dan op de lijn kijken of er geen dataverkeer is (Carrier Sense). Is de lijn vrij, dan zet het station het packet op de lijn. Een ander station zou echter tot dezelfde conclusie gekomen kunnen zijn want de communicatie is tenslotte Multiple Access. Mocht dit het geval zijn, dan detecteert de zender dit ook omdat een zender altijd naar de respons van de lijn kijkt als hij een packet wegstuurt. Als een zender een dergelijke Collision Detecteert, dan wordt acuut het versturen van het packet afgebroken en wordt er een speciaal collision signaal op de lijn gezet zodat de andere, ontvangende stations weten dat er een collision heeft plaatsgevonden. Zouden nu beide stations opnieuw proberen te gaan zenden, dan krijgen we geheid weer een collision. Om dit te voorkomen schrijft de specificatie voor dat een station na een collision een random tijd moet wachten voordat er opnieuw een poging gedaan wordt om het packet te versturen. Deze techniek heet Carrier Sense Multiple Access/Collision Detect of afgekort CMSA/CD. Ethernet heeft enkele zeer belangrijke nadelen. In de eerste plaats is dat het feit dat bij toenemende belasting het aantal collisions sterk toe zal nemen en dat daardoor de efficiëntie van het netwerk sterk afneemt. Het tweede nadeel is het feit dat er geen maximale tijdsduur gespecificeerd kan worden voordat een zender zijn ei in de vorm van een packet kwijtgeraakt is. Bij elke botsing moet de zender zich een random periode koest houden en het is niet uitgesloten dat het netwerk daarna in rust is of dat er weer een collision optreedt. Een Ethernet netwerk kan dus niet gebruikt worden in situatie waarin een gegarandeerde maximale responstijd (real time systemen dus) vereist is.

Bij ARC-net wordt gebruik gemaakt van een zogenaamd Token Bus protocol. Ook dit protocol werkt met packets. Bij dit protocol wordt er gebruik gemaakt van een zogenaamd token die van station tot station doorgegeven wordt. Het station dat het token bezit mag informatie op het netwerk zetten. Het mechanisme werkt als volgt:

Als het netwerk opstart, dan mag het knooppunt met het hoogste adres in het netwerk beginnen met zenden. Heeft hij niets te zenden, dan stuurt hij een speciaal frame, het zogenaamde token, naar het station met het eerstvolgende lagere adres. Dit station krijgt dan dus het token en mag informatie op netwerk zetten. Dit proces gaat door totdat het knooppunt met het laagste adres het token ontvangt. Deze stuurt het token weer naar het station met het hoogste adres. Elk station hoort dus te weten wat zijn buurman in de logische ring is en waarheen hij het token dus moet sturen. In figuur 3 is een packet-formaat voor een token bus netwerk getekend. Dit packet-formaat wijkt iets af van het formaat dat in ARC-net gebruikt wordt doch de verdere afhandeling is vergelijkbaar.

Het formaat wijkt op slechts enkele punten af van het Ethernet formaat. Ook hier zien we een destination veld, een source veld en een data veld. De netwerkadressen hebben een lengte van 6 bytes of van 2 bytes, afhankelijk van de situatie. Het dataveld heeft een maximale lengte van 65392 bits als er 6 byte adressen gebruikt worden; worden er 2 byte adressen gebruik, dan mogen de overige 8 bytes ook in het dataveld opgenomen worden. De lengte van een dataveld is dus ruim 5 maal zo lang als de maximale lengte in een Ethernet packet. De reden hiervan ligt in het feit dat bij Ethernet een station het netwerk niet te lang mag bezetten. Bij token bus mag elk station dat het token in zijn bezit heeft gedurende een bepaalde tijd packets versturen. Deze maxi-

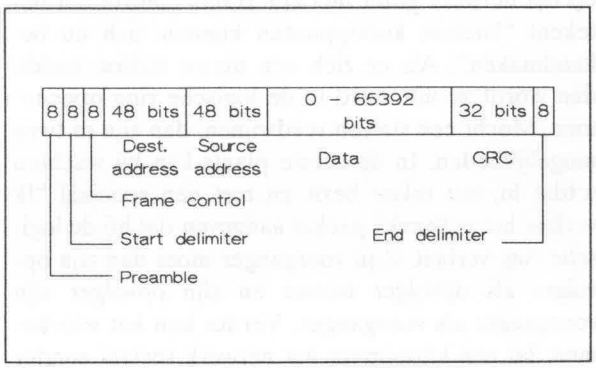


Fig. 3: packet-formaat in een Token Bus protocol

male tijd is een constante in het netwerk. Uiteraard dient een station ervoor te zorgen dat hij op tijd ophoud met zenden en dat hij ook geen packets verstuurt waarvoor de transmissietijd langer is dan de nog resterende tijd. In bepaalde gevallen kan het dus voorkomen dat een station de kans heeft zeer lange packets te sturen en waarom zouden we hem dan ook niet de gelegendheid daarvoor geven? Ook een 32 bit CRC is onderdeel van een packet. Bij ARC-net zijn er kleine verschillen ten opzichte van de getekende situatie die is afgeleid van de IEEE 802.4 standaard. Zo is bijvoorbeeld de CRC slechts 2 byte lang.

De eerste velden en het laatste veld zijn afwijkend van Ethernet. Het eerste veld is weer de preamble die gebruikt wordt voor het synchroniseren van zender en ontvanger. De lengte is echter slechts één byte. Het tweede veld is de zogenaamde "Start delimiter" en bevat analoge signalen die afwijkend zijn van de digitale 0 en 1. Hetzelfde geldt voor de "End delimiter" in het laatste veld. Op deze manier kan de ontvanger bepalen wat de lengte van een packet is en hoeft er geen lengte-veld meegestuurd te worden. Het derde veld in het packet is de zogenaamde "Frame control". In dit veld staat wat voor type packet het is. Zo betekent een inhoud 00001000 dat dit packet het token is. Er is uiteraard ook een code om aan te geven dat het een packet met informatie is en er is ook voorzien in een code waarmee de geadresseerde de zender kan waarschuwen dat een packet niet goed overkomt.

Bij Ethernet gaat het eigenlijk vanzelf als er een station bijkomt of verdwijnt. Bij Token Bus kan dat niet zomaar. Elk station moet namelijk minimaal weten wat zijn opvolger en voorganger zijn. Om nieuwe stations op te kunnen nemen, wordt er van tijd tot tijd door het station dat het token heeft een packet op het netwerk gezet met een frame control dat betekent "Nieuwe knooppunten kunnen zich nu bekendmaken". Als er zich een nieuw station meldt, dan wordt ze uiteraard in de logische ring opgenomen. Mocht een station verdwijnen, dan zijn er twee mogelijkheden. In de eerste plaats kan hij wachten totdat hij het token bezit en met een speciaal "Ik verlaat het netwerk" packet aangeven dat hij de logische ring verlaat. Zijn voorganger moet dan zijn opvolger als opvolger nemen en zijn opvolger zijn voorganger als voorganger. Verder kan het voorkomen dat een knooppunt het netwerk verlaat zonder dat te melden (spanningsuitval). Dit wordt gedetecteerd door de voorganger van dat knooppunt. De voorganger is namelijk verplicht te bewaken dat de opvolger, nadat het token doorgestuurd is, als houder van het token ook daadwerkelijk iets gaat zenden of het token doorstuurt. Doet hij dat niet, dan vraagt de voorganger "Wie is de opvolger van station xxx" en stuurt vervolgens het token naar dat station. Behalve de beschreven mogelijkheden zijn er nog een aantal maar de beschrijving daarvan voert me nu te ver. Mensen die precies willen weten hoe dat het allemaal in elkaar zit, moeten het maar eens nalezen in het boek van Tanenbaum.

Nieuwe knooppunten kunnen zich nu bekendmaken.

Goed, we hebben nu een netwerk en we zijn in staat op de één of andere manier packets over te sturen. Nu is het fraaie van OSI-netwerken dat het vanaf dit moment niet meer uitmaakt welk type netwerk gebruikt wordt. Laag 2 in het OSI-model, de datalink layer, biedt namelijk voor zowel Ethernet (IEEE 802.3) als voor Token Bus (IEEE 802.4) als voor de hier niet behandelde Token Ring (IEEE 802.5) precies dezelfde diensten aan de volgende laag, de netwerk layer. Het grappige is dan ook dat netwerken als Novell Netware met zowel ARC-net als Ethernet om kunnen gaan.

Afsluiting

In vorige paragraaf zijn we binnen de beschrijving van een netwerk gekomen tot in het midden van laag 2, de datalink layer. Deze datalink layer is namelijk opgebouwd uit twee subniveaus te weten de MAC-sublayer van Medium Access Control en de LLC-sublayer van Logical Link Control. De MAC-sublayer is in de vorige paragraaf behandeld, de LLC-sublayer is het startpunt in de volgende aflevering van deze serie. Na het afronden van de beschrijving van level 2 gaan we verder met level 3, de network layer en level 4 de transport layer. In die aflevering komt dan ook het beroemde protocol TCP/IP naar voren dat door een medeclublid eens de Kermit van de jaren 90 genoemd is.

Tot de volgende aflevering dus.

Gert van Opbroek

Literatuur

1: Andrew S. Tanenbaum: Computer Networks (Prentice-Hall)

2: H.M. Deitel: An introduction to Operating Systems (Addison Wesley)

3: Jürgen Siebert: ARC vernetzt; mc 10/90, 11/90, 12/90

Een verbeterd trace-programma voor DOS-65

TRACE.MAC

De tracer zoals gepubliceerd in de μP kenner in april 1989 (oorspronkelijk ontwerp: J. Rupert, Elektuur februari 1984) is een erg nuttig programma maar er blijven nog veel wensen over. Die wensen heeft Vandekerkhove zelf ook al genoemd.

Na het overtypen van zijn listing is eerst van een groot aantal variabelen de naam aangepast zodat het duidelijker is wat er gebeurt. Vervolgens is er voor gezorgd dat de tracer netjes na elke 15 regels stopt en wacht totdat er een toets ingedrukt wordt. Hierbij is het mogelijk om met een Q, q, E of e uit de tracer te gaan (er volgt in dat geval een jmp $C003, = de warme start van DOS-65).

Verder is er een disassembler ontworpen en verbonden met de tracer. De veranderingen aan de tracer om dit mogelijk te maken waren zeer gering, waarbij het programma nu ook rockwell cmos instructies ondersteunt.

Er zijn voor de gehele tracer/disassembler maar 2 bytes op pagina nul nodig. In de listing heten die bytes: pnt net als bij het programma van april 1989. Mocht het programma, dat door de tracer gevolgd wordt, ergens van dit adres gebruik gemaakt worden, dan volgt een foutmelding: error.

Het programma ondersteunt de DOS-65 help utility doch de tekst is heel erg beknopt gehouden om geheugenruimte te sparen.

Behalve de bovengenoemde aanpassingen is er verder een lijst van DOS-65 functies aan de tracer toegevoegd. De functies in het eerste deel van de lijst worden volledig uitgevoerd, met normale snelheid. De functies uit het tweede deel van die lijst worden volledig overgeslagen waarbij er op het scherm de boodschap SKIP verschijnt. Deze lijst ziet er als volgt uit:

```
;DOS65 BIOS behandelen.
;In deze tabel de adressen v.d. subroutines
;Van dostab tot skip worden de subroutines uitgevoerd
;vanaf skip worden ze overgeslagen.

dostab
        fdb  inecho     input and print character
        fdb  prch       print character
        fdb  crlf       print CR and LF
        fdb  prspace    print a space
        fdb  prhx       print a byte hexadecimal
        fdb  aschex     ascii to hex
        fdb  loupch     lower to upper
        fdb  sopt1      get option
        fdb  spar1      get parameter
        fdb  testkey    $F009 subr. test (with x=9)
                        if key depressed
        fdb  inch       input character
        fdb  ermes1     error messages
        fdb  prfile     print drive:dir and file

skip
        fdb  command    execute dos65-command
        fdb  sread      single read from file, X is
                        filenummer
        fdb  swrite     single write to file, X is
                        filenummer
        fdb  read       read file from disk, AY wijst
                        naar controlblok
        fdb  write      write data to disk, AY wijst
                        naar controlblok
        fdb  create     create a file, AY wijst naar
                        filenaam
        fdb  open       open a file, AY wijst naar
                        filenaam
        fdb  close      close a file, X is filenummer
        fdb  setfiln    set file name, AY wijst naar
                        filenaam
        fdb  curpos     direct cursor addressing
                        (x=col, y=row)
        fdb  ucurpos    bring the cursor where it
                        was before
;
```

LET OP: Als er in een programma JSR $C003 staat dan verwacht de tracer dus wel dat u iets intoetst.

Als er in een programma JSR $C02F staat dan doet DOS-65 een CR+LF en dus wordt het scherm wat rommelig.

Als er in een programma JSR $C000 staat dan wordt er een karakter op het scherm gezet. Echter als dat toevallig een clear screen is, dan ziet uw display er wel erg leeg uit, wat verwacht u anders?

Als er in een programma staat:

```
        lda  #$0D       accu:=return
        JSR  $C023      print een character
```

Dan wordt er een CR uitgevoerd door de subroutine op C023 en kunt u die JSR $C023 op het display NIET zien.

Met al deze uitbreidingen was het programma toch wel erg groot geworden en daarom is geprobeerd het weer wat korter te maken. Daarbij is alles gebruikt wat ik kon verzinnen. Bijvoorbeeld: in de lijs-

ten van de disassembler zijn zoveel mogelijk karakters of getallen in één byte "gestopt".

Dat scheelde ongeveer 256 bytes. Ook is waar mogelijk gebruik gemaakt van branch in plaats van jmp instructies. Het gebruik van de rcmos instructies scheelt circa 70 bytes. Zowel de nmos als de cmos versie is kleiner dan 2 kByte.

```
00..1F   1C0055551320888E4DF056651A20899E
20..3F   8C0055551320888E4DF066651A20999E
40..5F   1C0005551320888E4DF006651A10099E
60..7F   1C0055551320B88E4DF066651A10B99E
80..9F   4C0055551310888E4DF066751A10899E
A0..BF   3C3055551310888E4DF066751A1099AE
C0..DF   3C0055551310888E4DF006651A10099E
E0..FF   3C0055551310888E4DF006651A10099E
```

Fig. 1: volledige adresseertabel voor TRACE

Het is geen probleem de tracer op een ander adres te zetten, gewoon de instructie ORG in de file aanpassen en opnieuw assembleren.

Het is eenvoudig om te assembleren voor nmos of rcmos processor:

rcmos: as -W trace
cmos: as -WAcmos trace
nmos: as -WAnmos trace

TRATEST.MAC

Het programma tratest.mac is door mij gebruikt om alle mogelijkheden van trace te testen.

Afhandeling:

Print tratest.mac op papier.
Assembleer tratest.mac en trace.mac.
Lo tratest.bin om het in het geheugen te zetten.
trace.bin en geef als startadress op $1000.

Vergelijk dan het resultaat op het display met de listing van tratest.mac.

Suggesties

De gebruikte tabel voor het vaststellen van de addressing mode is ook anders te schrijven en ziet er dan zo uit als in figuur 1.
Daaruit blijkt dat er wel erg vaak dezelfde nummers in voorkomen. Het is dus mogelijk de laagste 5 bits van de opcode te gebruiken in een vereenvoudigde tabel, die in figuur 2 staat. Deze is te comprimeren tot 16 bytes. Er zijn dan 27 uitzonderingen. Zet die in een tabel, daarvoor zijn 27 + 14 bytes nodig. Er is dan een stukje extra software nodig dat de opcode in die extra tabel opzoekt. Geschatte lengte daarvan is 20 bytes, zodat er dan alles samen 128-(16 + 27 + 14 + 20) = 50 bytes te besparen valt.

De subroutine die de operand print maakt nu gebruik van erg veel vergelijkingen en jsr of jmp instructies. Maak een tabel waarin voor ieder element dat te printen is een bit voorkomt. Die tabel heeft een lengte van 16 bytes.

Dan kan een standaard routine m.b.v. de addressing mode die bit-tabel afwerken. Daarmede is nog circa 80 bytes te besparen. Dit is uitgeprobeerd maar het programma wordt dan toch minder leesbaar. Daarom is het in definitieve versie er niet ingebouwd.

Het lijkt me dat de lengte van de tracer in z'n huidige vorm geen echt probleem is. Door de veranderingen om het nog korter te maken wordt het programma misschien erg moeilijk te lezen of erg moeilijk te onderhouden. Een veel nuttiger uitbreiding lijkt mij om de lijst van subroutines van DOS-65 aan te passen zodat dat er meer routines ondersteund worden. Maar welke zijn dan nuttig? Het programma mag namelijk niet te groot worden. Ook interessant is misschien, als u aan een bijzonder groot project bezig bent, uw subroutines die veel gebruikt worden ook in de lijst te zetten.

Mijn systeem: Rockwell 65C02 op 2 MHz. Daarmee kon ik uiteraard wel trace assembleren t.b.v. de nmos versie. Het zou echter kunnen dat daarbij toch iets ontstaat dat niet geheel uitwisselbaar is zodat u misschien met een nmos systeem fouten tegenkomt die op mijn systeem niet optreden. Mocht dit zo zijn, dan wil ik dat uiteraard graag van u horen.

H. Speksnijder

Noot van de redactie

De programma's TRACE.MAC en TRATEST.MAC hebben een dermate grote omvang dat afdrukken in de µP Kenner niet mogelijk is. U kunt de programma's downloaden van het bulletin board The Ultimate of bestellen via de DOS-65 coördinator.

```
1C0055551320888E4DF006651A10099E
```

Fig. 2: gecomprimeerde adresseertabel

Van de bestuurstafel

Zomertijd. Gert van Opbroek heeft het dan onmiddellijk over komkommertijd. Zo gaat dat. Mooi weer. Geen zin om wat aan computers te doen. Buiten zijn en uitzakken is veel leuker. Vakantie houden is ook in trek en zo mogelijk nog populairder. Met enige regelmaat verdwijnen vertrouwde gezichten een tijdje uit de omgeving, om na enige tijd geheel verkleurd weer op te duiken. U heeft gelijk. Bij de KGN draaien de molens ook wat langzamer. De beloofde special is niet gekomen: geen kopij genoeg. De molens stonden de afgelopen maanden echter niet stil.

De MINIX-werkgroep bijvoorbeeld heeft heel wat werk verzet. Zoveel dat ze al bijna toe zijn aan het ontwerpen van een print voor de computer. Ze hebben ook een sponsor weten te vinden, zodat het kostenrisico wat gespreid kan worden. Verder heeft men rechtstreeks contact kunnen leggen met Prentice Hall, de uitgever van MINIX. Er komt derhalve schot in de MINIX-zaken.

Waar even geen schot in zat was het Bulletin Board. Daar sneuvelde een vette harde schijf, en zoals u weet zijn die dingen niet bepaald goedkoop. De Sysop zag zich dan ook genoodzaakt om bij het bestuur aan te kloppen voor toestemming om een nieuwe harde schijf voor het BBS te mogen kopen. Dat heeft het bestuur goedgevonden, zodat het BBS nu een brandnieuwe 200 Mbyte SCSI harde schijf heeft. Het bestuur vond dat een verantwoorde uitgave: via het BBS krijgen we regelmatig nieuwe leden binnen, en het BBS is een belangrijke verbindingsweg naar buiten, die dus ook P.R.-waarde heeft. Verder zijn we als club nog altijd financieel gezond, en dat is een goede zaak. Er wordt nu ook driftig gebeld op de tweede lijn van het BBS. De klachten over de slechte bereikbaarheid zijn ook minder.

Over Public Relations gesproken. Op de laatste bestuursvergadering is besloten dat de club ook dit jaar weer op de HCC-dagen aanwezig zal zijn op een stand. We moeten er vlug bij zijn, want het aantal beschikbare kramen voor zusterorganisaties is geringer dan ooit. Toch menen we daar te moeten staan, gezien de vele positieve reacties vorig jaar. Ook dit jaar komen we niet in conflict met onze eigen bijeenkomst in november.

Het ledental vertoont nog steeds een lichte groei (thans circa 160 leden). Een aantal mensen heeft nu echter al geroepen dat we een kleine maar gezellige club moeten blijven. Het bestuur denkt dat ze daar gelijk aan hebben. De club heeft tijden gekend met meer dan 400 leden. Dat leverde een gezonde hoeveelheid contributie op, maar tegelijkertijd waren er ook wat minder prettige zaken gaande. De belangrijkste daarvan was wel, dat het niveau van de club duidelijk naar beneden ging. Het signaal van de leden is dus duidelijk: liever klein met een goed niveau, dan andersom. Het bestuur zal erop blijven letten.

De volgende bijeenkomst zal in Haarlem zijn, de gebruikelijke locatie voor september. Op verzoek zal Ruud Uphoff zijn spreekbeurt over kwaliteitsbeheersing in software nog eens ten beste geven. Dit omdat het onderwerp nog altijd actueel is, en omdat de vorige keer de opkomst ronduit teleurstellend was. Beslist de moeite waard, u komt toch ook? Voor de bijeenkomst in november is het bestuur een poging aan het ondernemen een meer centraal gelegen locatie te vinden, zodat meer mensen een reden hebben om deel te nemen aan de ledenvergadering.

Op die ledenvergadering zal ondergetekende overigens het bestuur verlaten. De voornaamste redenen zijn de enorme drukte op het werk en het feit dat het tijd wordt plaats te maken voor vers bloed in het bestuur. Misschien is een bestuurstaak ook wel iets voor u. Het is in ieder geval altijd gezellig, en u bent niet alleen actief voor uzelf bezig, maar ook voor anderen. Ik heb het al die jaren leuk werk, en meestal ook zinvol bestede tijd gevonden. Mijn clubactiviteiten zullen niet tot nul gereduceerd worden. Het layoutwerk blijft. Misschien wordt DOS65 versie3 wel een uitdaging. Iedereen heeft het erover, maar niemand begint ermee. Dat zou jammer zijn. Nog 1 nummer te gaan...

Uw leidend voorwerp,

Nico de Vries

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 dediverse 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: 053-328506 of 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.

Correspondentie adres

Alle correspondentie betreffende verenigingszaken kan gestuurd worden aan:

KIM Gebruikersclub Nederland
Postbus 1336
7500 BH Enschede

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

Jan D.J. Derksen(secretaris)
Verfailleweg 434
1783 BP Den Helder
Telefoon 02230-46126

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

Ton Smits
De Meren 39
4731 WB Oudenbosch

Geert Stappers
Engelseweg 7
5825 BT Overloon
Telefoon 04788-1279

Mick Agterberg
Davidvosstraat 29
1063 HV Amsterdam
Telefoon 020-131538

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
Rinus Vleesch Dubois