

In dit nummer o.a.:

UNIX deel 6  
PC/XT RTC support  
Sorteren in PASCAL  
SCSI, EDI, MFM, RLL: Uitleg  
Shareware: Grafische hulpjes

## Inhoudsopgave

### De $\mu$ P Kenner

Nummer 73, oktober 1991  
 Verschijnt 6 maal per jaar  
 Oplage: 250 stuks  
 Druk: FEBO Offset, Enschede

### De redactie:

Gert van Opbroek  
 Geert Stappers  
 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  $\mu$ P Kenner nummer 74 verschijnt op  
 21 december 1991.

Kopijsluitingsdatum voor nummer 74 is  
 vastgesteld op 7 december 1991.

### Algemeen

Redactioneel .....	4
Schijnwereld .....	6
HCC-dagen 1991 .....	12
Te koop .....	29

### Vereniging

Uitnodiging voor de clubbijeenkomst .....	5
Agenda voor de algemene ledenvergadering .....	6
Conceptbegroting voor 1992 .....	7
Van de bestuurstafel .....	45

### Talen/Software

Het standaard operating system UNIX (deel 6) .....	8
Real Time Clock support voor uw PC/XT .....	13
To Share Or Not To Share, That's The Question .....	30
Een extern sorteerprogramma in Pascal .....	33

### Hardware

ST506, MFM, RLL, ESDI, IDE, SCSI: Enige uitleg .....	21
De KGN-68k computer .....	37

### Datacommunicatie

Methoden en technieken voor datacommunicatie (Deel 9) ..	41
----------------------------------------------------------	----

De  $\mu$ P Kenner is het huisorgaan van de KIM gebruikersclub Nederland en wordt bij verschijnen gratis toegezonden aan alle leden van de club. De  $\mu$ P 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.

De redactie houdt zich het recht voor kopij zonder voorafgaand bericht niet of slechts gedeeltelijk te plaatsen of te wijzigen. Geplaatste artikelen blijven het eigendom van de auteur en mogen niet zonder diens voorafgaande schriftelijke toestemming door derden gepubliceerd worden, in welke vorm dan ook.

De redactie noch het bestuur kan verantwoordelijk gesteld worden voor toepassing(en) van de geplaatste kopij.

## Redactioneel

Het is zover, dit is de laatste keer dat ik het redactioneel schrijf. Na de ledenvergadering begin ik met het overdragen van het redactiewerk naar de nieuwe redacteur die dan ook voor het volgende nummer het redactioneel wel zal gaan schrijven. Wie dat zal zijn wil ik nog niet bekend maken omdat de redactie-secretaris een bestuurslid is en de man maakt nog geen deel uit van het bestuur. Na de bestuursverkiezing op de komende ledenvergadering verdeelt het bestuur intern de functies en taken en daar zal besloten worden wie de nieuwe redacteur zal worden.

Ik blijf zelf nog minimaal een jaar bestuurslid en wil mijn activiteiten wat gaan verleggen naar KGN68k en MINIX. Verder zal ik ook zeker nog het nodige aan kopij blijven produceren want dat is iets wat ik zo langzamerhand erg leuk ben gaan vinden.

Dat laatste brengt me er toe eens terug te gaan kijken over de periode waarin ik verantwoordelijk was voor de redactie. Ik ben in deze functie begonnen nadat er enkele strubbelingen tussen bestuur en mijn voorganger geweest waren. Dat was bij nummer 51/52 van de  $\mu$ P Kenner en is dus nu zo'n vier jaar geleden. Nummer 52 en 53 waren echte crisesnummers. Ze werden op één enkele zondag door het voltallige bestuur "even" in elkaar gezet. Dat beviel toch niet echt en omdat ik eigenlijk nog geen taak binnen het bestuur had en het eigenlijk ook wel leuk vond, ben ik redacteur geworden.

Mijn eerste nummer was uitgave 54. In die tijd werd het blad nog met schaar en lijmpot in elkaar gezet en ik had veel te veel lijm gebruikt waardoor de regels over het papier golfd. Gelukkig voor de lezers heb ik ook die techniek goed onder de knie gekregen. Bij het aftreden van mijn voorganger zijn we overgegaan van de bekende gele omslag naar een omslag in vierkleurendruk, wel fraai maar eigenlijk veel te duur. Twee jaar geleden zijn we afgestapt van het knip- en plakwerk en begonnen met Desktop Publishing met behulp van een beroemd professioneel DTP-pakket. Dat was meteen na nummer 63, een uitgave die de KGN in eigen beheer gedrukt heeft en waar, helaas, veel bij mis is gegaan.

Hoewel het blad er nu, door de nieuwe vormgeving, beter uitziet, zijn toch de kosten aanmerkelijk lager,

ondermeer doordat de omslag niet meer in vierkleurendruk is. Dat is dus weer prettig voor de clubkas want het blad is de belangrijkste kostenpost en wat je daarop kunt uitsparen, zonder aan de kwaliteit te tornen, is mooi meegenomen.

Ik heb in de afgelopen vier jaar samengewerkt met diverse mensen. Uiteraard kan ik niet iedereen persoonlijk bedanken voor de prettige samenwerking; daarvoor er gewoon te veel mensen die "iets" voor de redactie gedaan hebben. Drie mensen wil ik echter toch met name noemen omdat ze veel werk verzet hebben en nog steeds verzetten om het blad bij de leden te krijgen. In de eerste plaats is dat Joost Voorhaar die de geestelijke vader van de vorm van het blad is. Hij heeft niet alleen het blad vormgegeven maar ook een aantal keren de layout verzorgt. Verder is hij degene geweest die ervoor gezorgd heeft dat de redactie (en de club) eens na ging denken over een beleid. De tweede man die ik wil bedanken is onze voorzitter Nico de Vries die zelf ook voor een behoorlijke kopij-stroom zorgt en al enige tijd de layout van het blad verzorgt. Tenslotte de derde man die veel, maar dan ongemerkt, voor het blad doet. Dat is onze penningmeester Jacques Banser met zijn hele familie. Hij onderhoudt het contact met de drukker, zorgt ervoor dat de bladen in de enveloppen komen zorgt voor de verzending etc. Kortom Jacques is een duizendpoot die een hoop klusjes doet die voorheen door de drukker gedaan werden waardoor de rekening van de drukker aanmerkelijk lager kan zijn.

Goed, na mijn memoires ook nog even over de bijzondere maand november. Op de zestiende houdt de KGN haar gebruikelijke jaarvergadering. Ik hoop dat u weer in grote getale naar Almelo komt om het bestuur op het juiste spoor te houden of weer te zetten. Verder zijn er op 22 en 23 november de hcc-dagen waar de KGN ook dit jaar weer met een eigen stand vertegenwoordigd zal zijn. Mocht u niet op bijeenkomsten komen, dan kunt u bij de stand van de KGN toch even kennis komen maken met een aantal bestuursleden. Ik zal er zeker zijn.

Verder de groeten en een hele goede toekomst met de nieuwe redacteur.

*Gert van Opbroek*

## Uitnodiging voor de clubbijeenkomst

Datum: 16 november 1991  
 Locatie: Wijkcentrum 't Veurbroek  
 Jan Tooropstraat 27  
 7606 Almelo  
 Telefoon: 05490 - 10353  
 Entree: gratis  
 Thema: Algemene ledenvergadering

### Met openbaar vervoer:

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

### Routebeschrijving

#### Per auto, vanuit het westen en het zuiden (A1/A35):

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

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

#### Per auto, vanuit het noorden (via de N 36):

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

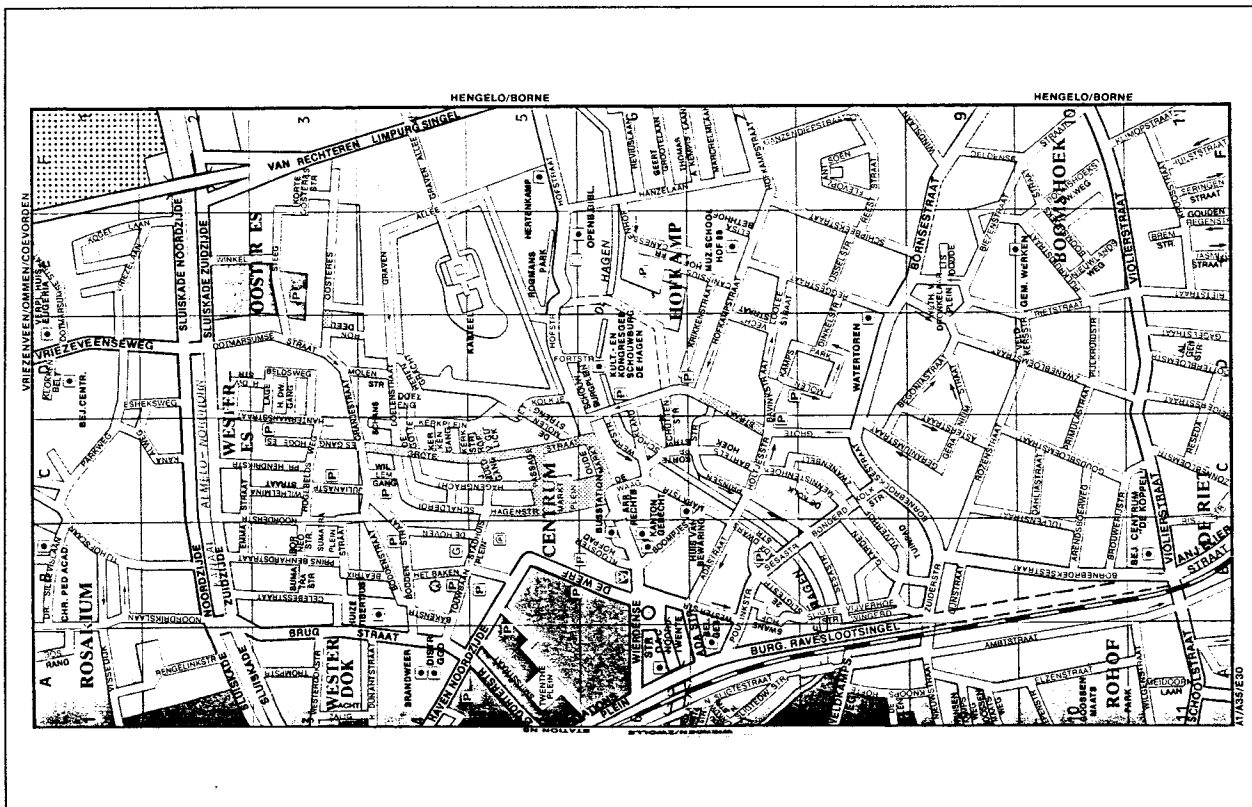
### Programma:

09:30 Zaal open met koffie  
 10:15 Opening  
 10:30 Algemene ledenvergadering, zie agenda.  
 11:45 Stand van zaken werkgroep KGN-68k  
 12:15 Forum en markt  
 12:30 Lunch, consumpties tegen betaling  
 Aansluitend het informele gedeelte met de mogelijkheid om andermans systemen te bewonderen en Public Domain software uit te wisselen. U en uw systeem zijn uiteraard van harte welkom.

17:00 Sluiting

### 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.



## Agenda voor de algemene ledenvergadering:

- Datum: 16 november 1991  
Plaats: zie uitnodiging clubbijeenkomst
- 1: Opening ledenvergadering.
  - 2: Notulen ledenvergadering d.d. 19 januari 1991.
  - 3: Conceptbegroting 1992 (zie elders in dit blad).
  - 4: Verkiezing kascontrole-commissie 1992.
  - 5: Verkiezing bestuursleden; de volgende bestuursleden treden per 1 januari af en zijn niet herkiesbaar:  
Mick Agterberg,  
Ton Smits en  
Nico de Vries

- Het bestuur heeft Joost Voorhaar bereid gevonden zich kandidaat te stellen voor een bestuursfunctie.
- Andere kandidaten dienen uiterlijk voor aanvang van de vergadering, schriftelijk, bij de voorzitter aangemeld te worden. Deze voordracht dient vergezeld te gaan van de handtekening van tenminste twee leden.
- 6: Het beleid van de club in 1992 en daarna. Tijdens dit agendapunt wil het bestuur van de KGN met de leden van gedachten wisselen over de activiteiten die de club zou kunnen ontplooiën.
  - 7: Rondvraag
  - 8: Sluiting

---

## Schijnwereld

Iedereen heeft wel eens gedacht over wat ons in de toekomst staat te wachten. Computers worden kleiner en krachtiger, waar zal dat allemaal toe leiden? Wat zal de opvolger worden van de huidige PC? Zal de huisrobot zich in de toekomst laten gelden? Zullen we straks drie dimensionale televisie hebben?

Het antwoord ligt dichterbij in de toekomst dan u wellicht denkt. In Delft is men bezig met een schijnwereld project. Met behulp van een computer wordt een irreële wereld gebrouwd. Via een helm, waarvan het vizier bestaat uit een tweetal schermplaatjes, worden de beelden van de gecreëerde schijnwereld overgebracht naar het netvlies van een proefpersoon. In die helm zitten ook een tweetal luidsprekertjes waarmee het mogelijk wordt het audiovisuele beeld compleet te maken. Tenslotte heeft men een terugkoppelingshandschoen ontwikkeld waarmee druk op verschillende delen van de hand gesimuleerd wordt. De handschoen zit vol met luchtkussentjes die opgeblazen worden op het moment dat er een object op die plaats je hand zou raken. Als je dus in je schijnwereld een vaas ziet, kun je die ook voelen als je hem beetpakt. Men is inmiddels al zo ver dat de proefpersoon in zijn schijnwereld een bal vangt die door een andere persoon toegeworpen wordt.

De mogelijkheden die met een ver doorgevoerde versie van dit systeem ontstaan zijn legio. Nemen we eens aan dat een drukpak ontwikkeld wordt, compleet met luchtsimulaties en temperatuur elementen. Je zou dan een complete schijnwereld kunnen fabriceren waarin je de meest maffe dingen zou kunnen doen. Plant je de proefpersoon dan in een waterbassin, dan kun je helemaal ver gaan. Een duik van de Eiffeltoren of een botsing met een trein? Je overleeft 't! Door muren lopen? Best...

Een dergelijk project kan eenvoudig uitgroeien tot de opvolger van de televisie. Een compleet audiovisueel stereobeeld dat real-time reageert op al je bewegingen... wat je daar allemaal wel niet mee kan doen! De gemiddelde thriller wordt een wel heel spannende gewaarwording, om nog maar te zwijgen over de gemiddelde "pikant" getinte film! Misschien vormt deze "simworld" machine ook nog wel een aardige oplossing voor het alcohol- en drugsprobleem. Geef de junk een uurtje in simworld en hij krijgt de trip van z'n leven!

Voorlopig zijn dergelijke machines natuurlijk nog véél te duur. Maar ja, wie weet wat de toekomst brengt?

*Joost Voorhaar*

Enschede, 10 Okt 1991

## CONCEPT BEGROTING 1992

BATEN :	
Contributie: 170 * f 55,00	f 9350,00
Losse uP Kenners	- 100,00
Bijeenkomsten	- 1000,00
Reclame	- 250,00
Dos 65	- 100,00
Projecten	- 2500,00
BBS-donateur/point	- 500,00
	-----
Totaal	f 13800,00
LASTEN :	
µP Kenner drukken/verzenden: 5 * f 1000,00	f 5000,00
Afschrijving inventaris	- 1000,00
Bulletin Board The Ultimate	- 1500,00
Bestuurs kosten	- 500,00
Reclame - ledenwervingsactie	- 1000,00
Projecten	- 2500,00
Bijeenkomsten	- 1000,00
Op pijl brengen voorraad	- 3000,00
Onvoorzien	- 1000,00
	-----
Totaal	f 16500,00

De begroting is dit jaar gemaakt met de gedachte dat volgend jaar het zelfde aantal leden zal worden behaald als in 1991 gezien het feit dat we een zeer aantrekkelijk project aan het opbouwen zijn en doordat dit jaar het aantal leden met 50 is gestegen en wij ervan uitgaan dat dit zich nu heeft gestabiliseerd of misschien wel voortzet.

Ook gaan we als bestuur er van uit dat een contributie verhoging van f 5,00 reëel is gezien het feit dat dit al zeer lange tijd niet gebeurd is en de kosten die diverse activiteiten met zich meebrengen wel omhoog zijn gegaan. Gezien het feit dat we op dit moment enigszins aan het interen zijn op onze reserve is het naar ons inzien wel verantwoord deze verhoging door te voeren.

- Het blad is het afgelopen jaar goed ontvangen bij de leden. Ook een intensieve 'begeleiding' bij de drukker heeft ertoe bijgedragen dat het blad zo is geworden zoals we het graag zien. Daarom gaan we op de zelfde voet door als het afgelopen jaar.
- De verwachte opbrengsten voor een advertentie in ons, toch wel perfecte, blad hebben we voor volgend jaar weer op f 250,00 gesteld. In de hoop dat dit volgend jaar wel gerealiseerd kan worden.....
- De losse verkoop van µP Kenners ligt zogoed als plat. Het komt nog sporadisch voor dat men een los nummer koopt. Dit gebeurt dan meestal op aanvraag via het BBS, hoewel de frequentie niet van betekenis is. Misschien dat we daar nog iets aan kunnen doen tijdens de komende HCC dagen, zoals het afgelopen jaar het geval was (alhoewel we er toen natuurlijk ook een hoop weg hebben gegeven)
- Het KGN68K project is nog niet in het stadium gekomen zoals we verwacht hadden. Maar het bestuur heeft goede hoop dat dit wel in het volgende boekjaar gerealiseerd kan worden. De bedragen zijn dan ook -bijna- identiek aan afgelopen jaar. Daarentegen heeft het project ook nog geen geld gekost (buiten enkele kleine declaraties). Ook voor DOS65 geldt voorgenoemde.

Uw penningmeester,

*Jacques Banser.*

## Het standaard operating system UNIX (deel 6)

Hoewel ik vorige keer schreef dat de artikelenreeks over UNIX beëindigd was, volgt hier nog een aflevering. Er bereikten mij namelijk een aantal vragen over het programmeren in de taal C die uiteindelijk neerkwamen op UNIX-specifieke zaken. Vandaar dus een aflevering 6, waarin ik zal trachten een aantal UNIX-specifieke zaken voor de (systeem-) programmeur toe te lichten.

### Standards

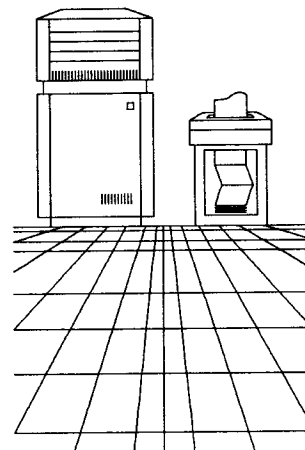
Er zijn in de loop der tijd een aantal verschillende "standaard UNIX" versies ontstaan. De belangrijkste standards zijn wel AT&T's system V en BSD, de UNIX variant van "the university of California at Berkeley". Beide stromingen zijn helaas behoorlijk verschillend. In deze serie ben ik voornamelijk uitgegaan van de AT&T system V variant, ook al omdat Minix op system V gebaseerd is.

Waar een standaard is zijn instituten die die standaard omschrijven. Zo zijn er bijvoorbeeld de standaardisatie instituten "ANSI" en "IEEE". ANSI houdt zich niet echt bezig met UNIX, maar wel met C. IEEE houdt zich zowel met C als met UNIX bezig. De richtlijnen waar ik hier van uit zal gaan zijn een mengeling van ANSI- en IEEE-POSIX standards; ze zijn gebaseerd op AT&T's system V release 4.

Naast het normale gebeuren als het onderhouden van het filesysteem en het proces management biedt UNIX een aantal diensten aan in de vorm van zogenaamde systemcalls. Dat zijn functies die door het operating system geboden worden en waarvan de werkende code dus niet in een (C-) library omschreven hoeven te worden. Dat wil zeggen: meestal is er wel een aanpassing in de C-library van het exacte formaat van een systemcall naar één of andere standaard C-functie.

### Signals

UNIX is een multitasking operating system. Dat betekent, zoals ik al eerder heb aangegeven, dat er meerdere taken (schijnbaar) tegelijkertijd worden uitgevoerd. Het wordt natuurlijk pas echt leuk als al die taken ook nog met elkaar kunnen communiceren en invloed op elkaar kunnen gaan uitoefenen. Daarbij zul je uiteraard wel een gedegen ondersteuning moeten hebben van het operating system, anders zou het maar een zootje worden!



Om vanuit jouw proces met een ander proces te kunnen babbelen zul je de aandacht moeten hebben (of krijgen...) van dat andere proces. Daarvoor kun je twee manieren bedenken: óf je laat de ontvanger zijn taak keihard onderbreken (als een soort interrupt dus) óf je wacht tot de ontvanger ook echt luistert. De eerste mogelijkheid is slechts in beperkte mate in UNIX geïmplementeerd: je kunt een zogenaamde signal sturen naar een ander proces. Je hebt er absoluut geen controle op wat er vervolgens met jouw signal gebeurt en hoe het ding afgehandeld wordt in het ontvangende proces. UNIX biedt voor deze simpele communicatie methode een aantal systemcalls waarmee je signals kunt versturen, afvangen en negeren. De lijst van signals is gedefinieerd in een headerfile met de naam "signal.h" (zie figuur 1, volgende pagina).

De waarden die hierin gedefinieerd zijn worden over het algemeen wel aangehouden. Ik ken eigenlijk geen enkele UNIX implementatie waarin SIGKILL geen 9 is... Het aantal signals dat door het operating system ondersteund wordt verschilt van implementatie tot implementatie. Het komt ook voor dat sommige signals in bepaalde C-compilers een andere naam hebben gekregen. Dit is dan weliswaar in strijd met de POSIX-standaard, maar met een eenvoudige aanpassing in de include file is dit wel aan te passen.

In de headerfile staan ook de definities van twee belangrijke functies: signal() en raise(). Met behulp van de signal() systemcall kan de uitwerking van een signal gedefinieerd worden. Zo kan bijvoorbeeld bepaald worden dat een programma niet op de inter-

**Er zijn in de loop der  
tijd een aantal  
verschillende  
"standaard UNIX"  
versies ontstaan.**

```

extern void (*signal(int sig, void (*handler)(int)))(int);
extern int raise(int sig)

#define SIGHUP      1 /* Hangup */
#define SIGINT     2 /* Interrupt key */
#define SIGQUIT    3 /* Quit key */
#define SIGILL     4 /* Illegal instruction (not reset) */
#define SIGTRAP    5 /* Trace trap (not reset) */
#define SIGIOT     6 /* System error */
#define SIGEMT     7 /* System error */
#define SIGFPE     8 /* Floating point exception */
#define SIGKILL    9 /* Kill (cannot be caught or ignored) */
#define SIGBUS    10 /* Bus error */
#define SIGSEGV   11 /* Segmentation violation */
#define SIGSYS    12 /* Bad argument to system call */
#define SIGPIPE   13 /* Broken pipe error */
#define SIGALRM   14 /* Alarm clock expired */
#define SIGTERM   15 /* Termination signal from kill */
#define SIGUSR1   16 /* Reserved for user signal 1 */
#define SIGUSR2   17 /* Reserved for user signal 2 */
#define SIGCLD   18 /* Death of a child */
#define SIGPWR   19 /* Power-fail restart */

#define NSIG      20 /* Number of signals */

#define SIG_DFL (void (*)())0
#define SIG_IGN (void (*)())1

```

Fig. 1: de signal include file

rupt toets (standaard de Del-toets) moet reageren met behulp van de systemcall `signal(SIGINT, SIG_IGN)`. Om een bepaalde actie aan een signal te knopen kan volstaan worden meteen `signal()` systemcall met als tweede parameter het adres van de functie die als handler voor het signal moet gaan dienen.

Met behulp van `raise()` kan een signal naar het eigen proces gestuurd worden. Het enige argument is het signal nummer. De functie is vooral handig tijdens het debuggen. Een alarm bijvoorbeeld kan eenvoudig gesimuleerd worden met `raise(SIGALRM)`.

Veel interessanter is de `kill()` systemcall. Waar de definitie van `kill()` uithangt is in veel UNIX-en verschillend. In ieder geval is de syntaxis meestal: `kill(int pid, int sig)`. Met deze systemcall kunnen we signals naar andere processen sturen. De `pid`-parameter geeft aan welk proces; de `sig`-parameter geeft weer het signal aan. Met behulp van `pid` kunnen we ook een aantal processen aanduiden. De mogelijkheden voor de `pid`-parameter zijn als volgt:

- n Zendt het signal naar alle processen met een process-group-ID dat gelijk is aan `abs(n)`
- 1 Zendt het signal naar alle processen waarvan de werkelijke user-ID gelijk is aan de effectieve user-ID van het zendende proces. Als deze effectieve UID echter 0 is (root dus!) dan ontvangen alle processen dit signal.
- 0 Zendt het signal naar alle processen met dezelfde process-group-ID als het zendende proces.
- n Zendt een signal naar het proces met PID gelijk aan n.

Uiteraard waakt het operating system ervoor dat niet iedereen zomaar aan iedereen signals kan gaan sturen. Als "normale" user kan je bijvoorbeeld geen `SIGKILL` aan processen sturen van je buurman of (nog erger!) een `SIGTRAP` aan de kernel...

Signals worden vooral gebruikt door het operating system om bepaalde situaties invloed uit te laten oefenen op een proces. Als de gebruiker bijvoorbeeld de Del-toets intaft, stuurt de kernel signal nummer 2 naar het actieve proces dat op de bijbehorende terminal loopt.

## Pipes

Het zou handiger zijn als we meer informatie uit konden wisselen tussen twee processen. Gelukkig kan dat ook: tussen twee processen kan een zogenaamde pipe geslagen worden waarover data heen en weer gestuurd kan worden. Er zijn twee soorten pipes in UNIX: standaard pipes en zogenaamde na-

med pipes. Standaard pipes kunnen in feite alleen binnen één proces gebruikt worden. Met een paar standaard trucks kan men dit soort pipes gebruiken om de communicatie tot stand te brengen tussen een parent- en een child proces. Zie figuur 2 voor de C-code. De truuk werkt als volgt:

```

main()
{
    FILE *names; /* Input file met namen */
    FILE *sort; /* Output file piped naar sort */
    int fd[2]; /* File descriptors voor pipe() */
    int c;

    if ((names = fopen("names", "r")) == NULL)
        perror("names file"), exit(1);

    if ((pipe(fd)) /* Kunnen we een pipe creëren? */
        perror("pipe"), exit(2); /* Nee dus! */)

    if (fork() == 0) {
        /* Volgende code wordt alleen door de child uitgevoerd! */
        close(0); /* Gooi stdin dicht */
        dup(fd[0]); /* Knoop stdin aan de pipe */
        close(fd[0]); /* Pipe input hebben we dus niet meer nodig! */
        close(fd[1]); /* En we willen niet schrijven op de pipe... */
        execlp("sort", "sort", NULL);
        exit(3); /* Hier komen we alleen als sort niet kon draaien! */
    }

    /* Parent proces gaat hier verder... */

    close(fd[0]); /* We willen niet van de pipe lezen... */

    if ((sort = fdopen(fd[1], "w")) == NULL)
        /* Pipe output kon niet geopend worden! */
        perror("sort"), exit(4);

    while ((c = getc(names)) != EOF)
        putc(c, sort); /* Copieer data naar sort programma! */

    fclose(names);
    fclose(sort);
    return(0);
}

```

*Fig. 2: namen sorteren met behulp van pipes en sort*

Bij een `fork()` systemcall wordt een proces gekloond. Na `fork()` is er een parent proces en een child die er exact hetzelfde uitzien. Een paar uitzonderingen echter: ze hebben beide een eigen pid en het functionresultaat van de `fork()` call is voor de parent het pid van zijn child en voor de child is het resultaat altijd 0. Alle filehandles en variabelen zijn echter voor parent en child hetzelfde. Dat betekent dat alles dat op de pipe geschreven wordt bij beide processen uitkomt. De parent kan nu bijvoorbeeld de output van de pipe dichtgooien en het childproces kan de input van de pipe dichtstoppen. Alles wat dan door de parent geschreven wordt komt dan vanzelf bij het child uit. Halen we deze grap uit door standard in- en output aan een pipe te hangen en laten we dat volgen door een `exec()` van een ander programma door de child, dan hebben we dus in feite een ander proces gecreëerd waarmee we kunnen communiceren door middel van een pipe. Deze truuk wordt bijvoorbeeld in de meeste C-compilers uitgehaald om de verschillende compiler passes aan elkaar te knopen.

Doordat de pipe's filedescriptors altijd alleen beschikbaar zijn voor het eigen proces is deze standaard pipe niet te gebruiken voor processen die onafhankelijk van elkaar opgestart worden. Daar heeft men echter het verschijnsel "named pipe" voor bedacht. Een named pipe is een pipe met een naam die in het filesystem ergens voorkomt. Een named pipe wordt echter niet gecreëerd door de standaard `create()` calls, maar door `mknod()`. Hoe een named pipe precies opgezet wordt zal ik nu achterwege laten. Daarvoor verwijs ik graag naar de verschillende boeken die dieper op deze materie ingaan. Named

pipes gedragen zich na openen precies hetzelfde als normale pipes.

### Tenslotte

Al met al is het toch weer een redelijk lang verhaal geworden. En ik ben eigenlijk nog niet aan het eind van mijn stof over UNIX internals gekomen. Een volgend deel zou bijvoorbeeld kunnen gaan over spokende programmatuur (uhh... demons en zombies) en wellicht dat we ook nog wel wat tips uit de mouw kunnen schudden waar de gemiddelde UNIX gebruiker nog wel wat aan zou hebben. In dit opzicht vast één tip voor als u wilt gaan stoeien met zelfgeschreven programmatuur onder UNIX: kijk eerst eens de manual pages van SCCS door. SCCS is de afkorting van "Source Code Control System" en komt in allerlei varianten (SVC voor Minix bijvoorbeeld!) op vrijwel alle UNIX machines voor. Met behulp van deze programmatuur kun je uitstekend teruggrijpen op oudere versies wat handig is bij het bug-dead-created-a-new-one-syndroom.

Ga je met wat grotere programma's die wellicht uit meerdere modules bestaan aan de gang, dan is het de hoogste tijd om ook de manual pages van "make" eens te bestuderen...

### Literatuur

1. Brian W. Kernighan & Dennis M. Ritchie, *The C Programming Language* (second edition), Prentice Hall (ISBN 0-13-110362-8)
2. John J. Valley, *UNIX Programmer's Reference*, QUE; programming series (ISBN 0-88022-536-X)

*Joost Voorhaar*

## HCC-dagen 1991

Na een onderbreking van elke jaren, begint het weer traditie te worden. De KGN heeft ook dit jaar weer een standplaats gehuurd voor de HCC-dagen. In eerste instantie zijn we op de grootste beurs voor computer-hobbyisten aanwezig om ons te presenteren als een technische computerclub. Deze twee dagen zijn zeer geschikt om ons te tonen aan het grote computerpubliek en de naam KGN meer bekendheid te geven, dus vooral voor het algemeen belang van de club.

Voor leden zijn er enkele direct merkbare voordelen van de clubstand:

- Een meetingpoint in het geval dat men elkaar kwijtraakt.
- Een ontmoetingsplaats, om andere leden eens in levende lijve te ontmoeten.
- Een plaats om je gekochte spullen uit te proberen.

- Een plaats om spullen in tijdelijke bewaring te geven.
- Koffie, als de kan niet net leeg is gegaan.
- Standhouderskaarten bieden zelfs de mogelijkheid om een standbemanninglid even terug naar je auto te laten gaan om te kijken of je je lichten nog aan hebt staan. Alvast wat dingetjes wegbrengen kan ook. De club kan zelfs een eenvoudig transportmiddel voor de zware dozen ter beschikking stellen.

Heb je interesse om de stand te bemannen laat het ons (bestuur of redactie) dan weten. Er zijn al wat onderwerpen om op de beurs te laten zien, heb je zelf wat zaken die je wilt tonen, dan kan dat ook.

*Geert Stappers*

(advertentie)

# KOBRA

## PC SYSTEMS

Basis-systemen worden geleverd in de volgende configuratie:

Harddisk 40 Mb, 23 ms  
Floppy drive 5.25" HD of 3.5" HD  
1 Mb RAM  
2\*serieel, 1\*parallel en game aansluiting  
VGA-kaart 256 KbRAM  
Desktop behuizing  
AT-Keyboard  
VGA-Mono-Monitor

Systeem:	Prijs:	Opties:
AT-12MHz	2075	512Kb VGA + SVGA
AT-16MHz	2250	kleuren mon. 400
386SX-16MHz	2695	1 Mb VGA + SGVA
386DX-25MHz	3250	Mulisync mon. 750
386DX-33MHz	3575	Extra drive 200
486-33MHz	5290	125 Mb HD i.p.v. 750
		2 Mb RAM i.p.v. 200
		4 Mb RAM i.p.v. 550
		8 Mb RAM i.p.v. 1050

Enkele suplies:

Diskettes:

Wit 5.25" DD.....	7.50
Wit 5.25" HD .....	14.95
Wit 3.5" DD.....	14.95
Wit 3.5" HD .....	25.00

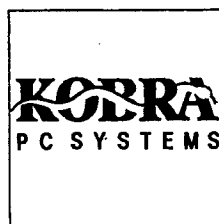
TDK 5.25" DD.....	17.95
TDK 5.25" HD .....	31.95
TDK 3.5" DD.....	31.95
TDK 3.5" HD .....	54.95

Diversen:

Muis: Tornado.....	79.00
Muismat .....	9.95
Diskettebak 80 stuks 5.25".....	17.95
Diskettebak 120 stuks 3.5".....	17.95

Alle prijzen zijn inclusief 18.5% BTW  
Op alle systemen krijgt u TWEE JAAR garantie!

KGN leden krijgen bij Kobra 10% korting!



K. Haaksbergerstraat 34  
7511 JS Enschede  
Tel: 053-309283  
FAX: 053-30926  
BBS: 053-309260  
&  
Wesselernering 3  
7544 JA Enschede.  
Tel: 053-772300

## Real Time Clock support voor uw PC/XT

Al een hele tijd geleden vroeg Joost Voorhaar aan mij: "Dat mooie BIOS van jou, ondersteunt dat ook een Real Time Clock?". Het antwoord was "Nee". De reden voor dat antwoord was er ook: het KGN-BIOS zat toen zo ongeveer ongeveer tot de nok toe vol. Dat kwam (en komt) door de toegevoegde ondersteuning voor de 720k 3.5 inch floppy drives, en het feit dat ik vind dat de gebruiker fatsoenlijke foutmeldingen in tekst moet krijgen en niet in cryptische getallen.

Joost vroeg of ik eens een keer wilde kijken of de Real Time Clock toch toegevoegd kon worden. Voordeel: na het booten van het DOS staat dan altijd de tijd en de datum goed, zonder een programmaatje als CLOCK.COM of TIMER.COM te moeten draaien. Nadeel: de RTC voor de XT is niet genormaliseerd, ofschoon zeer veel machines zijn uitgerust met een MM58167 RTC op de I/O adressen 0340h-035Fh. Er is ooit zo'n BIOS gemaakt. Toen slaagde ik er niet in om de RTC fatsoenlijk te ondersteunen en de toenmalige structuur van het BIOS onaangetast te laten. Een paar weken terug kwam de brainwave.

### Wat doet die socket daar?

Ik wierp een blik op het moederbord van mijn test-XT en zag iets vertrouwds: 5 EPROMs op rij. Cassette BASIC (4 stuks) en het BIOS zelf. Naast de BASIC ROMs zat (al jaren) een leeg voetje. Kan een 8k EPROM in. Adresbereik: 0F4000h-0F5FFFh. De brainwave: dat valt binnen de grenzen van de ROMscan van het BIOS!

Het moest dus mogelijk zijn de extra klokrouines in een ROMmetje te gieten dat door de ROMscan geactiveerd kon worden. Het resultaat ziet u hieronder: de listing van ROMTIMER.ASM.

### Globale werking

ROMTIMER ondersteunt de meest voorkomende RTC: de hierboven genoemde MM58167 op I/O adres 0340h en verder. Om de systeemtijd goed te krijgen direct na het booten moeten er twee dingen mogelijk zijn: de tick-tijd moet goed staan, en de datum moet opgevraagd kunnen worden met INT 1Ah, AH=03. De extra set bevat verder: haal RTC tijd (AH=02), zet RTC tijd (AH=04) en zet RTC datum (AH=05). De twee alarm calls (AH=06 en AH=07) worden niet ondersteund.

Bij aanroep van het ROM wordt eerst gekeken of de RTC inderdaad aanwezig is. Dit gebeurt in de routine CHECK\_RTC\_THERE. Deze routine kijkt of de teller voor 1000-sten seconden loopt, geeft in de carry aan of de RTC er is of niet. Wordt de klok niet aangetroffen, dan wordt het ROM onmiddellijk verlaten.

Wordt er wel een RTC aangetroffen, dan wordt INT 1Ah omgelegd naar de routines in het ROM, waarna de RTC-tijd wordt opgevraagd. Deze wordt omgerekend naar het aantal ticks (18.2 per seconde), waarna de systeemtijd gelijk wordt gezet.

"Dat mooie BIOS  
van jou,  
ondersteunt dat  
ook een Real Time  
Clock?"

De opstart van COMMAND.COM treft na het booten nu de correcte tijd in ticks aan. De datum wordt verkregen door INT 1Ah, AH=03 aan te roepen. De RTC kan eenvoudig worden gelijk gezet: Bij DOS 3.30 en hoger wordt ook de RTC gecorrigeerd als de tijd of datum met TIME of DATE wordt aangepast.

De sourcetekst is te vinden op het KGN-BBS "The Ultimate".

*Nico de Vries*

(Zie voor de sourcetekst de volgende pagina).

```

PAGE      60,131
.LIST
;*****
;*
;* ROMTIMER
;* An extension BIOS to service a
;* 58167 battery backed up RTC.
;*
;* To create ROM image do:
;* MASM ROMTIMER;
;* LINK ROMTIMER;
;* EXE2BIN ROMTIMER.EXE ROMTIMER.BIN
;*
;* Adjust the byte checksum to zero by
;* changing the byte at 01FFh.
;*
;* Written by Nico de Vries after a
;* suggestion by Joost Voorhaar.
;*
;* This program is in the public domain
;* Please copy and distribute to BBS's.
;*

;*****
;*
;* Vector area.
;*
ABS0      SEGMENT AT 0
          ORG      1AH*4
TIME_PTR  LABEL    WORD
ABS0      ENDS

;*****
;*
;* System BIOS stack definition.
;*
STACK     SEGMENT AT 030h
          ASSUME   SS:STACK
          DW      128 DUP(0)
TOS       LABEL    WORD
STACK     ENDS

;*****
;*
;* ROM BIOS data area.
;*
DATA      SEGMENT AT 40h
          ORG      006Ch
TIMER_LOW DW      ?           ;time: low word count
TIMER_HIGH DW     ?           ;time: high word count
TIMER_OFL DB     ?           ;time: roll over flag
DATA      ENDS

RTC_ADDR  EQU      0340h      ;base address of clock

```

Fig. 1: sourcetekst van ROMTIMER.ASM

```

;*****
;*
;* Fill EPROM space with empty state.
;*
CODE      SEGMENT PAGE
          ORG      0
          ASSUME   CS:CODE,DS:ABS0,ES:ABS0,SS:STACK
          DB       2000h DUP (0FFh) ;fill EPROM with empties

          PAGE
          ORG      0
;*****
;*
;* ROMTIMER. An extension BIOS to ser-
;* vice a 58167 RTC.
;*
ROM_START DW      0AA55h ;ROM signature
          DB       2 ;size is 1k
;*****
;*
;* Check if RTC is present. If so, re-
;* vector INT 1Ah, retrieve the RTC
;* time and set the tick count ac-
;* cordingly.
;*
;* The 58167 RTC is assumed to be pre-
;* sent on I/O addresses 340h-34Fh.
;*
ROM_TIMER PROC    FAR
          XOR     AX,AX ;point ES
          MOV     ES,AX ;to vector area
          PUSH   ES ;do the same
          POP    DS ;to DS
          CALL   CHECK_RTC_THERE ;if RTC not there
          JC     NO_58167 ;exit with carry set
          CLI    ;no interrupts allowed
          MOV     Word Ptr TIME_PTR,Offset TIME_OF_DAY ;else
          MOV     Word Ptr TIME_PTR+2,CS ;revector INT 1Ah
          MOV     AH,2 ;else get time
          INT     1Ah ;of RTC
          JC     NO_58167 ;if error, exit
          MOV     AL,DH ;get seconds
          CALL   BCD_TO_BIN ;convert BCD to byte
          MOV     BL,18d ;get ticks per second
          MUL    BL ;get seconds ticks
          XCHG   CX,AX ;get BCD minutes (CL) in AL
          PUSH   AX ;save BCD hours (was CH)
          CALL   BCD_TO_BIN ;convert BCD minutes to byte
          MOV     BX,182d*6d ;ticks per minute
          MUL    BX ;get minute ticks
          ADD    CX,AX ;add to seconds ticks
          POP    AX ;restore BCD hours
          MOV     AL,AH ;get them in AL
          CALL   BCD_TO_BIN ;convert BCD to byte
          PUSH   AX ;save actual hours
          MOV     BL,7 ;10007h ticks per hour
          MUL    BL ;get hour ticks low

```

```

                ADD     AX,CX                ;add to minutes/seconds
                MOV     DX,AX                ;set ticks low
                POP     CX                    ;get hours (ticks hi word)
                ADC     CX,0                 ;add possible roll over
                CALL    DDS                    ;point DS to BIOS RAM
                ASSUME  CS:CODE,DS:DATA,ES:ABS0,SS:STACK
                MOV     TIMER_LOW,DX        ;set time low
                MOV     TIMER_HIGH,CX       ;and high
                MOV     TIMER_OFL,0         ;clear roll over flag
NO_58167:       STI                          ;allow interrupts again
                RET                          ;and exit ROM call
ROM_TIMER      ENDP

                ;*****
                ;*
                ;* Convert BCD in AL to binary in AL.
                ;*
BCD_TO_BIN     PROC     NEAR
                PUSH    CX                    ;save CX
                MOV     AH,AL                ;get BCD in AH also
                MOV     CL,4                 ;shift high
                SHR     AH,CL                ;into place
                AND     AL,0Fh               ;get low only in AL
                AAD                    ;convert to binary
                POP     CX                    ;restore CX
                RET                          ;and exit
BCD_TO_BIN     ENDP

                ;*****
                ;*
                ;* Check presence of RTC. If RTC there,
                ;* carry is clear.
                ;*
CHECK_RTC_THERE PROC     NEAR
                CLI                          ;no interrupts
                PUSH    AX                    ;save
                PUSH    CX
                PUSH    DX                    ;registers
                MOV     DX,RTC_ADDR          ;get address
                IN     AL,DX                 ;get 1000ths of seconds
                CMP     AL,90h               ;if over 90 (BCD)
                JA     NO_RTC                ;no RTC,
                MOV     AH,AL                ;save current value
                MOV     CX,100h              ;get time out
RUN_TEST:      JMP     $+2                    ;wait a bit
                IN     AL,DX                 ;read next value
                XOR     AL,AH                ;if any change (clear carry)
                JNE    RTC_THERE            ;then clock is there
                LOOP   RUN_TEST              ;else wait for change
NO_RTC:        STC                          ;if through, flag no clock
RTC_THERE:    POP     DX                    ;restore
                POP     CX
                POP     AX                    ;saved registers
                STI                          ;allow interrupts
                RET                          ;and exit
CHECK_RTC_THERE ENDP

                PAGE

```

```

;*****
;
;* Revector INT 1Ah: Time services.
;*
;* AH = 00: Read tick time.
;* Input registers: None.
;* Output registers:
;* (DX) = Ticks since midnight, low.
;* (CX) = Ticks since midnight, high.
;* (AL) = Roll over flag (1 = thru midn.)
;* Roll over flag is cleared.
;*
;* AH = 01: Set tick time.
;* Input registers:
;* (DX) = Ticks since midnight, low.
;* (CX) = Ticks since midnight, high.
;* Output registers:
;* None, however: Roll over flag is cleared.
;*
;* AH = 02: Read RTC time.
;* Input registers: None.
;* Output registers:
;* When RTC time is valid:
;* (CH) = Hours in BCD.
;* (CL) = Minutes in BCD.
;* (DH) = Seconds in BCD.
;* (DL) = 0
;* Carry is cleared.
;* When RTC time is invalid or no RTC: Carry set.
;*
;* AH = 03: Set RTC time.
;* Input registers:
;* (CH) = Hours in BCD.
;* (CL) = Minutes in BCD.
;* (DH) = Seconds in BCD.
;* Carry is cleared.
;* NOTE: Daylight savings is NOT supported!
;* Output registers:
;* None, however carry is cleared.
;* When no RTC there: Carry set.
;*
;* AH = 04: Read RTC date.
;* Input registers: None.
;* Output registers:
;* When RTC date is valid:
;* (DL) = Day in BCD.
;* (DH) = Month in BCD.
;* (CL) = Year in BCD.
;* (CH) = Century in BCD.
;* Carry is cleared.
;* NOTE: The century is not stored in
;* the RTC, but generated automatically
;* years 00-79 get century 20, years
;* 80-99 get century 19.
;* NOTE: Year is stored and handled as
;* a constant. The RTC does NOT advance
;* the year after midnight on december

```

```

;* 31st. This is not a bug in the BIOS,
;* but a shortcoming of the 58167 RTC.
;* When RTC date is invalid or no RTC: Carry set.
;*
;* AH = 05: Set RTC date.
;* Input registers:
;* (DL) = Day in BCD.
;* (DH) = Month in BCD.
;* (CL) = Year in BCD.
;* (CH) = Century in BCD (ignored).
;* Carry is cleared.
;* NOTE: Concerning century, see note above.
;* Output registers:
;* None, however carry is cleared.
;* When no RTC there: Carry set.
;*
TIME_OF_DAY PROC FAR
PUSH DS ;save DS
PUSH DI ;and DI
CALL DDS ;point DS to BIOS RAM
ASSUME CS:CODE,DS:DATA,ES:ABS0,SS:STACK
MOV AL,AH ;get function code
CMP AL,2 ;if tick function
JB CMD_OK ;go ahead
CALL CHECK_RTC_THERE ;check for clock
JC CMD_ERR ;if not there, exit
CMP AL,6 ;else allow RTC commands
CMC ;invert carry
JC CMD_ERR ;if not valid command, exit
CMD_OK: CBW ;else get function# in AX
SHL AX,1 ;times two
MOV DI,AX ;in DI
XOR AX,AX ;zero AX
CLI ;no interrupts
CALL CS:[DI + Offset TIME_ADDR] ;call function
STI ;allow interrupts
CMD_ERR: POP DI ;restore
POP DS ;saved registers
RET 2 ;and exit saving flags
TIME_OF_DAY ENDP
;*****
;*
;* INT 1Ah: Address table.
;*
TIME_ADDR DW Offset GET_TICKS ;AH = 00: get time
DW Offset SET_TICKS ;AH = 01: set time
DW Offset GET_RTC_TIME ;AH = 02: get RTC time
DW Offset SET_RTC_TIME ;AH = 03: set RTC time
DW Offset GET_RTC_DATE ;AH = 04: get RTC date
DW Offset SET_RTC_DATE ;AH = 05: set RTC date
;*****
;*
;* INT 1Ah, AH = 00: get time.
;*
GET_TICKS PROC NEAR
MOV DX,TIMER_LOW ;get time low
MOV CX,TIMER_HIGH ;and high

```

```

XCHG    AL,TIMER_OFL    ;zero roll over flag and get it
RET                                            ;then exit
GET_TICKS ENDP
;*****
;*
;* INT 1Ah, AH = 01: set time.
;*
SET_TICKS PROC    NEAR
MOV     TIMER_LOW,DX    ;set time low
MOV     TIMER_HIGH,CX   ;and high
MOV     TIMER_OFL,AL    ;clear roll over flag
RET                                           ;and exit ROM call
SET_TICKS ENDP
;*****
;*
;* INT 1Ah, AH = 02: get RTC time.
;*
GET_RTC_TIME PROC    NEAR
MOV     DX,RTC_ADDR + 4 ;get hours address
IN      AL,DX           ;get hours
CMP     AL,23h          ;if over 23
JA      RTC_ERR         ;exit
MOV     CH,AL           ;save hours
DEC     DX              ;point to minutes
IN      AL,DX           ;get them
CMP     AL,59h          ;if over 59
JA      RTC_ERR         ;exit now
MOV     CL,AL           ;save minutes
DEC     DX              ;point to seconds
IN      AL,DX           ;get them
CMP     AL,59h          ;if over 59
JA      RTC_ERR         ;exit now
MOV     DH,AL           ;save seconds
XOR     DL,DL           ;zero DL (clear carry)
RET                                           ;and exit
GET_RTC_TIME ENDP
;*****
;*
;* INT 1Ah, AH = 03: set RTC time.
;*
SET_RTC_TIME PROC    NEAR
PUSH    DX              ;save DX
MOV     DX,RTC_ADDR + 4 ;get address
MOV     AL,CH           ;get hours
OUT     DX,AL           ;set RTC
DEC     DX              ;point to minutes
MOV     AL,CL           ;get minutes
OUT     DX,AL           ;set RTC
POP     AX              ;restore old DX
DEC     DX              ;point to seconds
STORE_END: MOV     AL,AH    ;get seconds
OUT     DX,AL           ;set RTC
STC                                           ;set carry
RTC_ERR: CMC           ;according to state
RET                                           ;and exit
SET_RTC_TIME ENDP

```

```

;*****
;*
;* INT 1Ah, AH = 04: get RTC date.
;*
GET_RTC_DATE PROC NEAR
MOV DX,RTC_ADDR + 7 ;get address
IN AL,DX ;read month
CMP AL,12h ;if over 12
JA RTC_ERR ;exit with STC
MOV CH,AL ;save month
DEC DX ;point to date
IN AL,DX ;get date
CMP AL,31h ;if over 31
JA RTC_ERR ;exit
MOV CL,AL ;else save day
ADD DX,+3 ;point to year (latch 100ths)
IN AL,DX ;read year
CMP AL,99h ;if invalid, exit
JA RTC_ERR ;
MOV DL,AL ;save year
MOV DH,19h ;get century
CMP AL,80h ;if before '80
JAE CENT_19 ;
MOV DH,20h ;get new century
CENT_19: XCHG CX,DX ;swap registers
CLC ;clear carry
RET ;and exit
GET_RTC_DATE ENDP
;*****
;*
;* INT 1Ah, AH = 05: set RTC date.
;*
SET_RTC_DATE PROC NEAR
PUSH DX ;save day/month
MOV AL,CL ;get year
MOV DX,RTC_ADDR + 9 ;get address
OUT DX,AL ;write year
SUB DX,+3 ;point to date
POP AX ;get day/month back
OUT DX,AL ;set date
INC DX ;point to month
JMP STORE_END ;and exit writing RTC
SET_RTC_DATE ENDP
;*****
;*
;* Point DS to BIOS RAM.
;*
DDS PROC NEAR
PUSH AX ;save AX
MOV AX,Seg DATA ;get segment value
MOV DS,AX ;in DS
POP AX ;restore AX
RET ;and exit
DDS ENDP
CODE ENDS
END

```

## ST506, MFM, RLL, ESDI, IDE, SCSI: Enige uitleg

### Inleiding

In de kop van dit artikel staan enkele kreten die de meeste lezers wel eens gehoord zullen hebben. Aangezien ik denk dat heel veel mensen niet weten wat de verschillen precies zijn lijkt het mij een goede zaak daar eens wat over te gaan schrijven. De termen kom je allemaal tegen als je overweegt een harddisk voor je computer te kopen of een computer met harddisk aan wilt schaffen.

We gaan het in dit artikeltje hebben over harddisks. In het normale computer-jargon worden dit ook wel Winchester-schijven genoemd. Ik heb bij mijn literatuurscan ook nog gevonden waar deze naam voor het eerst gebruikt is in relatie tot harddisks. De naam Winchester was de interne codenaam bij IBM voor het project waarin het IBM 3340 harddisk-systeem met verwisselbare schijven ontwikkeld werd. Dit speelde zich zo rond 1973 af. Hoewel dit systeem verwisselbare schijven had (heeft?), heeft het toch een groot aantal kenmerken die de moderne harddisks ook hebben. Het kenmerkende van de 3340 in tegenstelling tot zijn voorgangers, is dat de koppen waarmee de informatie op de schijven geschreven en gelezen wordt onderdeel uitmaken van het verwisselbare deel. Een diskelement bestaat uit een vier magnetische schijven met in totaal 7 koppen in een afgesloten ruimte. Bij het inzetten van het schijvenpakket in de drive worden de koppen mechanisch gekoppeld aan de motor die de koppen naar binnen en buiten laat bewegen.

NB. In dit artikel komt een aantal keren de maat-aanduiding "voor". Dit is de gebruikelijke afkorting voor inch = 25,4 mm.

### Een klein beetje natuurkunde

De eigenlijke schijf van een harddisk bestaat uit een pakket van een aantal ronde (harde) platen die aan beide zijden zijn voorzien van magnetisch materiaal. Dit materiaal is vergelijkbaar met het magnetische materiaal dat gebruikt wordt voor floppy's, cassettebandjes, videobanden etc. dit materiaal heeft de eigenschap dat het zich gedraagt als een heleboel kleine magneetjes. In een magneetveld willen al deze magneetjes in dezelfde richting willen gaan staan zoals een kompasnaald in een magneetveld ook in een bepaalde richting gaat staan. In tegenstelling tot een kompasnaald die gemakkelijk kan draaien, kunnen de magneetjes in het materiaal niet zo gemakkelijk

draaien. Het draaien van de magneetjes (het "richten") gebeurt niet voor alle magneetjes tegelijk, afhankelijk van de sterkte van het magneetveld gaan een aantal magneetjes in de goede richting staan. Dit betekent dat als je bijvoorbeeld een seconde een zwak magneetveld aanlegt er weinig magneetjes draaien en leg je een seconde een veel sterker magneetveld aan, dan draaien er veel magneetjes. Het aantal magneetjes dat zich richt is dus afhankelijk van de sterkte van het magneetveld.

Als je enige tijd een magneetveld op het materiaal hebt laten inwerken, en je zet het magneetveld uit, dan blijven bijna alle magneetjes die in de richting van het magneetveld gedraaid zijn zo staan. Deze magneetjes zullen pas weer kunnen draaien als er een magneetveld in een andere richting aangelegd wordt. Dit betekent dat de kleine magneetjes nu zelf een (zwak) magneetveld geven die tegengesteld is aan de richting van het oorspronkelijke magneetveld. Samen met het effect uit de vorige alinea geeft dit het effect dat er na het aanleggen van een zwak magneetveld een zwakke magneet overblijft en na een sterk magneetveld een sterkere magneet. Het materiaal kan dus als het ware "onthouden" wat voor magneetveld er uitgeoefend is.

Het principe van het onthouden van de sterkte en de richting van een extern magneetveld wordt gebruikt bij video's, cassettebandjes, floppy's etc. Waarschijnlijk is dit de techniek die het meeste gebruikt wordt om informatie vast te leggen en te bewaren. Dat wordt gedaan door met behulp van een "kop" een magneetveld op te wekken. Deze kop is niet meer dan een elektromagneet bestaande uit een kern van een metaallegering en een spoel. Laat je nu een stroom door deze spoel lopen, dan zal er in de kern een magneetveld opgewekt worden. Op de plaats waar de spleet in de kern zit, komt dit magneetveld naar buiten. Met behulp van dit magneetveld wordt het magnetisch materiaal gemagnetiseerd. Laat je een wisselstroom door de kop lopen en laat je het magnetisch materiaal langs de kop bewegen, dan zal er in dat materiaal een wisselend magneetveld vastgelegd worden. Dit is schematisch getekend in figuur 1 (zie volgend pagina).

### Het draaien van de magneetjes (het "richten") gebeurt niet voor alle magneetjes tegelijk.

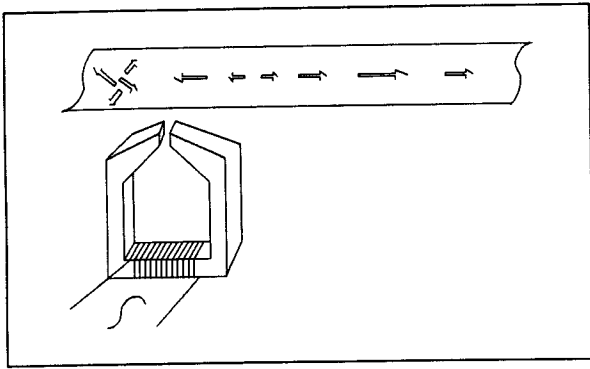


Fig. 1: magneetkop met magneetband

Het onthouden van de informatie is de ene helft van het verhaal. De tweede helft van het verhaal is het teruglezen van de informatie. Dit gebeurt door de band weer langs dezelfde kop te laten lopen. Het is namelijk zo dat als je door een spoel een wisselend magneetveld laat gaan, er in de spoel een stroompje gaat lopen. De sterkte van dit stroompje is afhankelijk van de sterkte van het wisselende magneetveld. Door de band weer langs de kop te laten lopen, wordt het magneetveld in de band opgepikt door de spleet die in de kern zit, wordt het veld dat in de band vastgelegd is opgepikt waarna er in het spoeltje een stroompje gaat lopen dat alleen wat zwakker is dan de stroom waarmee de band beschreven is maar er verder net zo uitziet.

Het laatste waar ik nog even op wil wijzen is het feit dat je alleen maar met wisselspanningen kunt werken. Je kunt wel een constant magneetveld in de band of op de schijf vastleggen, je kunt het echter niet teruglezen omdat er alleen een stroom in de spoel opgewekt wordt bij wisselende magneetvelden. Dit betekent dat je de stroom door de magneet met

een bepaalde minimale frequentie moet laten veranderen. Verder is er uiteraard ook een bepaalde maximale frequentie waarmee je kunt werken. Als je de stroom heel snel laat veranderen dan wordt er weinig materiaal gebruikt om één periode van deze wisselstroom vast te leggen en wordt de sterkte van het magneetveld bij het teruglezen te klein. De minimale en maximale frequentie die gebruikt kan worden is van een groot aantal factoren afhankelijk, maar vooral van de snelheid waarmee het magnetisch materiaal ten opzichte van de kop beweegt. Bij de lage snelheid van 4,75 cm/s waarmee cassettebandjes werken kun je het frequentiegebied van +/- 50 Hz tot zo'n 18.000 Hz bestrijken, voor videorecorders heb je veel hogere frequenties en daarmee een veel hogere relatieve snelheid van band ten opzichte van de kop. Bij een videorecorder wordt deze hoge snelheid bereikt door zowel de kop als de band te laten bewegen, maar aangezien we een computerclub en geen videoclub zijn lijkt het mij niet zinvol hier verder op in te gaan.

### Koppen, cilinders, tracks en sectoren

Bij harde of floppy-schijven gaat het in principe net zo. Ook daar hebben we te maken met een kop die de informatie op het magnetisch materiaal schrijft en die de informatie terug kan lezen. In tegenstelling tot een cassettebandje zit het materiaal op één of meer ronde schijven. Toch is het principe niet anders dan bij een band. De schijf is namelijk opgebouwd uit een aantal concentrisch sporen. Een dergelijk spoor wordt een "track" genoemd. Als de computer nu een bepaalde track gelezen heeft en hij wil verdergaan op een andere track, dan wordt de kop in radiale (van of naar het middelpunt) richting verplaatst.

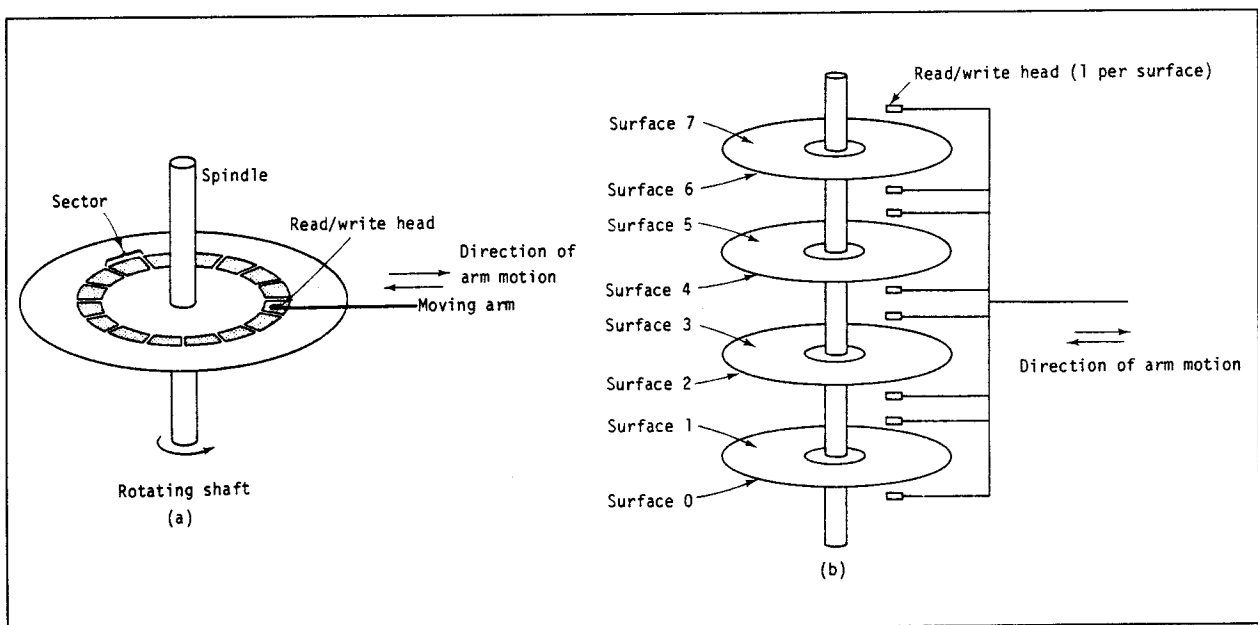


Fig. 2: schematische opbouw van een harddisk

Bij dubbelzijdige floppy's, kunnen we aan beide kanten van de floppy lezen en schrijven. Hiervoor zit er aan beide kanten van de floppy een kop. Bij harde schijven worden meestal meerdere schijven op één as gemonteerd waarbij elke schijf aan beide kanten gebruikt kan worden. Hier is het aantal koppen dus groter. Aangezien alle koppen recht boven elkaar gemonteerd zijn (zie figuur 2), kunnen we zonder de koppen te verplaatsen van elke schijfkant een track lezen. Een dergelijke eenheid wordt ook wel een cilinder genoemd.

Er zijn schijven waarbij niet alle schijven aan beide kanten voor dataopslag gebruikt worden. Deze schijven hebben dan een zogenaamde servo-kop. Op de bijbehorende schijf staat dan informatie waarmee de kop nauwkeurig op de cilinder gepositioneerd kan worden.

Om de informatie nog verder op te splitsen is elke track ook nog opgebouwd uit een aantal sectoren. Bij schijven is de sector de basiseenheid. Elke sector kan, afhankelijk van het type operating systeem, een bepaald aantal bytes bevatten. Voor MS-DOS bevat een sector effectief 512 byte. Deze 512 byte worden verder nog uitgebreid met een sector-adres en codes voor foutdetectie.

Tenslotte zijn in tabel 1 als voorbeeld enkele praktijkgevallen voor MS-DOS opgenomen.

Tabel 1: Enkele disk-types.

Type	Cil.	Kop	Sect.	Cap.	ca. prijs
3,5"	DD	80	2 9	720 kB	fl. 150,--
3,5"	HD	80	2 18	1,44 MB	fl. 200,--
5,25"	DD	40	2 9	360 kB	fl. 100,--
5,25"	HD	80	2 15	1,2 MB	fl. 100,--
ST-138		614	6 17	30 MB	fl. 450,--
ST_1126		1068	7 29	111 MB	fl. 1500,--

### Mechanische opbouw

Zoals opgemerkt, worden de koppen in radiale richting over de schijf verplaatst. Hiervoor zijn in principe twee systemen in omloop. In de eerste plaats is dit een systeem waarbij met behulp van een stappenmotor de kop stap voor stap naar binnen of naar buiten verplaatst kan worden. Door te tellen kan worden bijgehouden bij welke cilinder de kop is aangeland. Een tweede (moderner) systeem is de zogenaamde "Voice Coil". Dit is een positionering volgens het principe waarop een luidspreker werkt. Door meer stroom door een spoel te sturen gaat de kop naar binnen en door minder stroom door de spoel te sturen trekt een veer de kop naar buiten.

Het voordeel van een dergelijk systeem is dat bij spanningsuitval de koppen automatisch geheel naar buiten getrokken worden en worden vergrendeld, in tegenstelling tot het eerste principe waarbij de kop blijft op de plaats waar hij was. Over het algemeen bewegen de koppen bij een Voice Coil systeem sneller naar de gewenste positie als bij een stappenmotor. Verder kunnen bij de Voice Coil systemen de koppen m.b.v. een servo-kop zeer nauwkeurig op hun positie gezet worden omdat een voice coil in principe elke plaats op de schijf kan bereiken, in tegenstelling tot een stappenmotor die altijd per stap een vaste verplaatsing heeft.

De schijven hebben tegenwoordig meestal een gestandaardiseerde diameter. Deze diameter bepaalt de grootte van de drive en daarmee de ruimte die de drive inneemt in de kast. De standaardmaten voor de diameter van de schijf die je tegenwoordig tegenkomt zijn 3,5", 5,25" voor de buitenafmetingen van de drive komt daar dan nog een halve inch bij. Het aantal schijven en de verdere afmetingen van de onderdelen bepalen de hoogte van de drive. Een hoogte van 3,5" wordt "Volle hoogte" genoemd; de helft hiervan uiteraard "Halve hoogte". De standaard 5,25 inch floppy-drives zijn tegenwoordig halvehoogte. Behalve de standaard hele en halve hoogte komen ook andere afmetingen voor. Dit hoeft echter geen problemen bij het inbouwen te geven omdat er meestal wel inbouwsets verkrijgbaar zijn waarmee het hoogteverschil opgelost kan worden.

De schijven zelf zitten op een as die door de spindel-motor aangedreven wordt. Deze motor laat de schijven met een bepaald toerental draaien. Een voorbeeld van een toerental is 3600 omwentelingen per minuut voor 5,25" schijven. Dit toerental komt onder andere voor bij de kleinere harddisks van 20 MB. De koppen liggen, als de schijf stilstaat op de schijven. Draait de schijf op volle toeren, dan mogen de koppen de schijf niet raken omdat ze anders zouden kunnen beschadigen. Door de draaisnelheid, komt de lucht boven (en onder) de schijf in beweging, de schijf neemt als het ware de lucht mee bij het draaien. Door de vorm van de kop wordt de kop vervolgens opgetild zoals een bewegend vliegtuig opgetild wordt door de vorm van de vleugels. De kop "vliegt" dus als het ware boven de schijf. Deze afstand is, schrik niet, ongeveer 0.0005 mm. Een stofdeeltje is, vergeleken bij de afstand tussen kop en schijf, al een enorme afmeting en dat is de hoofdreden dat het schijvenpakket in een afgesloten capsule zit die alleen in zogenaamde stofvrije ruimtes geopend mag worden.



Fig. 3: opengewerkte harddisk

#### Modulatie-technieken transferrate etc.

In de tweede paragraaf is beschreven hoe een wisselspanning vastgelegd wordt in het magnetische materiaal van de schijf. Als je een wisselspanning kunt vastleggen, dan kun je uiteraard ook informatie vastleggen. Dit kun je op diverse manieren doen, vergelijkbaar met de diverse manieren die je hebt om informatie via een telefoonlijn over te brengen. In de praktijk wordt de informatie altijd serieel (dus bit voor bit) naar de schijf geschreven en wordt gebruikt gemaakt van een vorm van frequentie-modulatie. Hoe de modulaties precies werken is mij momenteel (nog) niet bekend. Wel dat er twee gangbare vormen zijn voor het beschrijven van harddisks. In de eerste plaats is dat dezelfde modulatie-techniek die voor double density floppy's gebruikt wordt, de Modified Frequency Modulation of MFM. Met behulp van deze modulatie-techniek, een draaisnelheid van 3600 toeren per minuut en een (standaard) overdrachtssnelheid van 5 MBit/s gaan er 17 sectoren van 512 kB op een track.

De tweede techniek werd mogelijk gemaakt doordat de kwaliteit van de gebruikte materialen toenam. Deze techniek wordt RLL van Run Length Limited genoemd. Het enige dat ik van deze techniek weet, is dat dezelfde schijven een 50% hogere capaciteit krijgen. De overdrachtssnelheid tussen controller en disk wordt namelijk verhoogd naar 7,5 MBit/s. In de praktijk zullen MFM schijven vaak gebruikt kunnen worden in combinatie met een RLL controller en

zodoende een 50% hogere capaciteit geven. Doe dit echter niet. De MFM schijven zijn namelijk niet berekend op de hogere schrijfdichtheden waardoor het kan zijn dat de gegevens niet onder alle omstandigheden teruggelezen kunnen worden. Het is ook niet verstandig het "even" te proberen. Ik ken een geval waarbij een dure MFM schijf van 150 MB geformatteerd werd met een RLL-controller en vervolgens zowel voor RLL als voor MFM onbruikbaar geworden was omdat kennelijk de koppen of het magnetische materiaal onherstelbaar beschadigd waren.

Het woord "transfer rate" is al even gevallen. Dit is de snelheid waarmee de informatie tussen de controller en de schijf uitgewisseld kan worden. Deze snelheid wordt opgegeven in MBit/s. De transfer rate bepaalt voor een deel de performance die een disk in een computer-systeem heeft. De performance is namelijk van een aantal factoren afhankelijk.

Stel, we willen een aantal sectoren (blokken) met informatie van de schijf halen. In de eerste plaats moeten we dan de koppen op de juiste track positioneren. Dat doen we door de koppen een aantal tracks naar binnen of naar buiten te laten bewegen. Bij het opstarten worden de koppen namelijk naar de eerste track (track 0) gestuurd en vervolgens wordt precies bijgehouden hoeveel stappen ze naar binnen en naar buiten gemaakt hebben. Voor het bewegen van een kop is enige tijd nodig, de step

time of ook wel aangeduid met de stepping rate, het aantal stappen per seconde die gemaakt mogen worden. Een tweede zaak waar we rekening met moeten houden, is het feit dat we gemiddeld één halve omwenteling moeten wachten voordat, na een kopverplaatsing, de gewenste sector bij de kop aangeland is. De tijd die hiervoor nodig is, is afhankelijk van de draaisnelheid en wordt wel avarage rotational delay genoemd. Bij 3600 toeren/ minuut bedraagt de rotational delay zo'n 8 milliseconden.

Nu is het niet zo dat de tijd om een kop te positioneren altijd het aantal stappen maal de step time is. Zowel bij stappen-motoren als bij voice coil mag men de stappen bij grotere verplaatsingen sneller toedienen dan bij kleine verplaatsingen. Bij voice coil wordt dit ook inderdaad gedaan, de harddisk telt het aantal steps dat uitgevoerd moet worden en plaatst vervolgens in één keer de kop op de juiste plaats. Om deze reden wordt de toegangstijd van een schijf meestal opgegeven als Average Seek Time of Average Access Time. De gemiddelde zoektijd (seek time) is de gemiddelde tijd die nodig is om de kop op een willekeurige track te plaatsen, de gemiddelde toegangstijd is de tijd die nodig is om de koppen aan het begin van een willekeurige sector op een willekeurige track te plaatsen en is de gemiddelde zoektijd plus de gemiddelde rotational delay.

Nadat de kop aangekomen is bij het begin van de gewenste sector, wordt de informatie van de schijf naar de controller gestuurd met de transfer rate. Aangezien een sector vervolgens naar het geheugen van de computer overgebracht moet worden, gaat ook hier weer enige tijd verloren en in deze tijd draait de schijf door. Over het algemeen kan de computer de controller nooit op tijd opdracht geven om de volgende sector van de schijf te halen. Om deze reden liggen de sectoren vaak niet in volgorde op de track. Bij een AT liggen de sectoren bijvoorbeeld in de volgorde 0, 9, 1, 10, 2, 11, 3, 12, 4, 13, 5, 14, 6, 15, 7, 16 en 8. In dit geval is de volgende sector altijd de tweede vanaf de huidige sector en spreekt men van een interleave 1:2. Bij een XT komen interleaves voor van 1:3 t/m 1:5, afhankelijk van de snelheid van de machine. Bij een interleave van 1:5 is de sectornummering: 0, 7, 14, 4, 11, 1, 8, 15, 5, 12, 2, 9, 16, 6, 13, 3, 10. De interleave wordt opgegeven bij het formateren van de harddisk en vervolgens op de schijf, in de sector-adressen verwerkt. Uiteraard is het zonde van de tijd om bijvoorbeeld eerst de sectoren 9 t/m 13 aan je neus voorbij te laten gaan als je sector 8 t/m 11 in wilt lezen. Je hebt dan in de praktijk anderhalve omwenteling nodig om alles in te lezen. Ook het feit dat de drive moet wachten omdat eerst de informatie van een sector naar het geheugen van de computer getransporteerd moet worden is zonde. Om deze reden zijn er tegenwoordig controllers op

de markt die zelf een complete track kunnen bufferen. Nadat de kop op een track aangekomen is, wordt gestart met lezen en wordt de hele track in het cache-geheugen van de controller opgeslagen. Is de gevraagde sector binnen, dan wordt deze doorgestuurd naar het geheugen van de computer. Wordt vervolgens om de volgende sector gevraagd, dan staat deze al in het cache-geheugen van de controller en kan dus meteen naar het werkgeheugen overgebracht worden. Deze controllers worden aangegeven door de term "Interleave 1:1". De invloed van een dergelijke ingreep bij een AT met MFM controller (5 MBit/s = 630 kB/s maximaal) blijkt uit het staatje in tabel 2.

Tabel 2: Transfer rate als functie van de interleave (MFM controller met ST 506 interface aangesloten op een AT).

Interleave	Transfer Rate	Opmerking
1:1	480 kB/s	Speciale controller
1:1	20 kB/s	Interleave te klein
1:2	240 kB/s	Optimale waarde
1:3	160 kB/s	

Deze paragraaf kunnen we afsluiten met vast te stellen dat MFM en RLL beide modulatie-technieken zijn waarbij RLL een transfer rate geeft die 50% hoger ligt dan bij MFM en daardoor ook bij dezelfde schijf-afmetingen 50% meer capaciteit oplevert. De hogere transfer rate zal echter over het algemeen weer voor een deel teniet gedaan worden door het feit dat er ook een hogere interleave nodig zal zijn, tenzij gebruik gemaakt wordt van een 1:1 controller.

### Interface-typen

Voor de verbinding tussen een computer en een harddisk worden een aantal interface-typen gebruikt. In de volgende paragrafen worden de interfaces die in de PC-wereld voorkomen behandeld.

#### 1) ST 506/412: De meest voorkomende.

Het (nu nog) meest voorkomende interface voor harddisks is het ST 506 (of ook wel ST 506/412) interface. Dit is eigenlijk niet meer dan een "opgevoerde" shugart interface zoals die voor de floppy's gebruikt wordt. Het interface bestaat uit een control kabel die van drive naar drive loopt en één of meer data kabels waarbij elke drive met een eigen datakabel op de controller is aangesloten. Aangezien in de datakabel ook het signaal "Drive selected" is opgenomen, kunnen de signalen "Drive select 1" en "Drive select 2" ook gebruikt worden om aan te geven welke kop gebruikt moet worden. Tezamen met de 3 "Head select" signalen, kunnen dan in totaal 16 kop-

pen aangestuurd worden. In tabel 3a en 3b zijn de signalen in het ST 506 interface opgesomd.

Tabel 3a: ST 506 interface, control kabel.

PIN	I/O	Signal
2	N/A	Reserved
4	O	- Head select 2
6	O	- Write gate
8	I	- Seek complete
10	I	- Track 0
12	I	- Write fault
14	O	- Head select 0
16	N/A	Reserved
18	O	- Head select 1
20	I	- Index
22	I	- Ready
24	O	- Step
26	O	- Drive select 1
28	O	- Drive select 2
30	N/A	Reserved
32	N/A	Reserved
34	O	- Direction in

All odd pins: ground.

Tabel 3b: ST 506 interface, data kabel.

PIN	I/O	Signal
1	O	Drive selected
13	O	+ MFM Write data
14	O	- MFM Write data
17	I	+ MFM Read data
18	I	- MFM Read data

2,4,6,8,11,12,15,18,19,20: Ground

3,5,7,9,10: Reserved

Opvallend in deze interface is het feit dat tussen diskdrive en controller (via de pennen 13/14 en 17/18) de gemoduleerde signalen uitgewisseld worden. Dit heeft als voordeel dat we bij gebruik van andere modulatie-technieken, bijvoorbeeld RLL, dezelfde interface en daarmee in principe dezelfde drives kunnen gebruiken. In de praktijk betekent dit dat MFM en RLL drives technisch op zowel MFM als RLL controllers aangesloten kunnen worden. Het is echter niet gegarandeerd dat het ook betrouwbaar zal werken, tenzij de fabrikant van de diskdrive deze mogelijkheid aangeeft.

Bij gebruik van een ST 506 interface is de vaste transfer rate tussen controller en drive 5 MBit/s (MFM) of 7,5 MBit/s (RLL).

## 2) ESDI: Een verbeterde ST 506

ESDI is de afkorting van Enhanced Small Devices Interface en is door IBM in de PC-wereld geïntro-

duceerd met de PS/2 serie. De uiterlijke kenmerken van ESDI zijn gelijk aan die van ST 506, alleen zijn de signalen die door de control en data kabels lopen anders. Dit betekent dat het aansluiten van een ESDI drive op een MFM controller (of andersom) het overlijden van óf de drive óf de controller of zelfs beide tot gevolg zal hebben. In tabel 4a en 4b zijn de signalen in de control en data kabel opgesomd.

Tabel 4a: ESDI interface, control kabel.

PIN	I/O	Signal
2	O	- Head select 3
4	O	- Head select 2
6	O	- Write gate
8	I	- Config/Status data
10	I	- Transfer Ack
12	I	- Attention
14	O	- Head select 0
16	I	- Sector
18	O	- Head select 1
20	I	- Index
22	I	- Ready
24	O	- Transfer request
26	O	- Device select 1
28	O	- Device select 2
30	O	- Device select 3
32	O	- Read gate
34	O	- Command data

All odd pins: ground.

Tabel 4b: ESDI interface, data kabel.

PIN	I/O	Signal
1	I	- Device selected
2	I	- Sector
3	I	- Command complete
4	O	- Address mark enable
5	N/A	Reserved
6	O	Ground
7	O	Write clock
8	O	- Write clock
9	N/A	Reserved
10	I	Read/Reference clock
11	I	-Read/Reference clock
12	N/A	Ground
13	O	NRZ Write data
14	O	- NRZ Write data
15	N/A	Ground
16	N/A	Ground
17	I	NRZ Read data
18	I	- NRZ Read data
19	N/A	Ground
20	I	- Index

Ten opzichte van ST 506 is een belangrijk deel van de zaken die bij ST 506 geregeld worden verplaatst van de controller naar de drive. De drive kan namelijk een aantal commando's zelfstandig uitvoeren en de status terugmelden. Als voorbeeld het verplaatsen van de kop naar een bepaalde cilinder. Bij ST 506 moet de controller step- (stap-) pulsen aan de drive doorgeven, bij ESDI kan de controller de drive opdracht geven met de koppen naar een bepaalde cilinder te gaan of aan de disk vragen hoe zijn configuratie (aantal koppen, cilinders en sectoren) is. Opvallend is verder dat het selecteren van een device weer via de control kabel gaat, zoals ook bij floppy's gebruikelijk is en dat de drive zelf via de datakabel terugmeldt dat hij geselecteerd is.

Tenslotte is het type modulatie niet meer een zaak van drive en controller samen maar alleen van de drive. De controller biedt de gegevens serieel aan zonder dat ze zich zorgen maakt over de modulatievorm waarmee ze straks op de drive opgeslagen worden. Hierbij staat de afkorting NRZ voor No Return to Zero hetgeen wil zeggen dat alleen veranderingen ten opzichte van het voorgaande bit worden doorgegeven. De snelheid waarmee de informatie tussen disk en controller uitgewisseld wordt is 10 MBit/s. De werkelijke transfer rate komt daarmee op maximaal 10 MBit/s.

### 3) SCSI: Universeel en snel

SCSI (spreek uit: SKOEZIE) is afgeleid van Small Computer System Interface en wordt zo langzamerhand een echte standaard voor het uitwisselen van informatie tussen een willekeurig computersysteem en een willekeurig massa geheugen. De SCSI-standaard werd door de firma Shugart ontwikkeld als SASI-standaard (Shugart Associates System Interface).

Behalve Winchester schijven met SCSI interface zijn er ook tape streamers en CD-ROM's die dit interface gebruiken. SCSI komen we behalve bij de PC's ook tegen bij Atari, Amiga, DEC-machines, KGN-68k, Macintosh, kortom voor (bijna) elke machine is wel een SCSI-interface beschikbaar.

Standaard SCSI gaat uit van een 50-polige kabel van maximaal 6 meter, die, vergelijkbaar met de floppy's, van device naar device wordt doorgelust. Deze kabel wordt ook wel de SCSI-bus genoemd. In tabel 5 wordt een overzicht van de signalen in deze SCSI-bus gegeven. Hierbij wordt uitgegaan van de SCSI-standaard. Tegenwoordig bestaat er ook een geavanceerdere SCSI-2 standaard die in dit artikel buiten beschouwing wordt gelaten.

Tabel 5: Signalen in de SCSI-bus.

PIN	Signal	Pin	Signal
2	- DB(0)	28	Ground
4	- DB(1)	30	Ground
6	- DB(2)	32	- ATN (Attention)
8	- DB(3)	34	Ground
10	- DB(4)	36	- BSY (Busy)
12	- DB(5)	38	- ACK (Acknowledge)
14	- DB(6)	40	- RST (Reset)
16	- DB(7)	42	- MSG (Message)
18	- DB(P)	44	- SEL (Select)
20	Ground	46	- C/D (Control/Data)
22	Ground	48	- REQ (request)
24	Ground	50	- I/O
26	Terminator power		

All odd pins: Ground.

Opvallend van SCSI t.o.v. de voorgaande interfaces is het feit dat de data parallel via de pennen 2 t/m 18 over de bus gaan. Pen 18, DB(P), is een pariteitsbit waarmee gecontroleerd kan worden of de data correct zijn. De snelheid waarmee de informatie over de SCSI-bus gestuurd kan worden ligt, bij asynchrone communicatie op ongeveer 1,5 MB/s (bytes; geen bits !) en bij synchrone communicatie op ongeveer 3 MB/s.

Bij een SCSI-bus heeft elk device een eigen adres. Er kunnen maximaal 8 devices aan één SCSI-bus aangesloten worden die worden geadresseerd met behulp van de acht databits in de SCSI-bus. Deze devices zijn weer onder te verdelen in initiators en targets. Een initiator is een device dat de zaakjes regelt en opdrachten geeft aan de targets. Een target is een device dat opdrachten van een initiator krijgt en uitvoert. De standaard laat toe dat er meerdere initiators op een SCSI-bus aanwezig zijn doch in de praktijk zal er slechts één initiator zijn: de computer. De rest van de devices (drives, streamers...) zijn in dat geval targets. De communicatie over een SCSI-bus verloopt in fasen. Hiervan zijn er in totaal 11 gedefinieerd:

BUS FREE:	De bus is vrij en wordt niet gebruikt
ARBITRATION:	De bus wordt toegewezen aan één van de initiators
SELECTION:	De initiator selecteert een target
RESELECTION:	De target kiest een initiator
COMMAND:	De target vraagt aan de initiator een commando
DATA - IN, DATA - OUT:	Er worden gegevens overgestuurd van initiator naar target of andersom

STATUS:	De target geeft een statusmelding door aan de initiator
MESSAGE - IN, MESSAGE - OUT:	Er wordt een bericht van initiator naar target of andersom overgestuurd
DISCONNECT:	De target geeft de bus vrij omdat uitvoering van het commando enige tijd vergt. Na uitvoering van het commando selecteert de target de initiator d.m.v. een RESELECTION.

Voor communicatie met harddisks zijn er in de SCSI-norm een aantal commando's gedefinieerd. De belangrijkste zijn:

Test Unit Ready:	Controleer of het device gebruikt kan worden
Recalibrate:	Zet de koppen op cilinder 0
Request Sense:	Foutstatus inlezen
Format Unit:	Schijf formateren
Read Block Limits:	Grootte van de harddisk opvragen
Read:	Leesopdracht
Write:	Schrijfopdracht
Seek:	Positioneringsopdracht
Mode Select:	Harddisk-parameters instellen
Mode Sense:	Harddisk-parameters opvragen

Om bij een PC een SCSI-bus te kunnen gebruiken, moet er in de PC een zogenaamde SCSI host-adapter, vaak ook ten onrechte een SCSI-controller genoemd, opgenomen worden. Deze host-adapter zorgt ervoor dat de PC als initiator voor een SCSI-bus kan fungeren. De gebruikte harddisk heeft vervolgens ook nog een hoeveelheid intelligentie nodig om de SCSI commando's uit te kunnen voeren. Dat zou gedaan kunnen worden door een aparte controller die aan de ene kant een SCSI interface heeft en aan de andere kant bijvoorbeeld een ST 506 interface. Per SCSI-device mag men in totaal maximaal 8 devices op een dergelijke controller aansluiten zodat men met één host-adapter in totaal maximaal 56 drives aan kan sturen. Tegenwoordig worden de controllers bijna altijd met de drive samengebouwd en spreken we van een zogenaamde SCSI-drive; van de mogelijkheid meerdere drives op één controller aan te sluiten wordt in dat geval geen gebruik gemaakt. In totaal kan men dus 7 van dergelijke drives aansluiten op een PC met een host-adapter.

Een laatste opmerking over het fenomeen SCSI gaat over de adressering van de informatie-blokken. Standaard wordt de informatie op de schijf opgedeeld in blokken van 512 kB. Deze blokken worden gewoon genummerd van 0 t/m het hoogste bloknummer. De opdeling in cilinders, koppen en sectoren vindt dus geheel in de controller in de SCSI-drive plaats. Helaas moeten bij MS-DOS schijven opgedeeld worden in koppen, cilinders en sectoren. Daarom wordt in het BIOS op de host-adapter meestal een vast aantal koppen en sectoren per track gekozen. Het werkelijke aantal koppen en sectoren per track op de SCSI-drive kan in dat geval wel heel anders zijn.

Voor mensen die meer willen weten van SCSI in het algemeen, verwijs ik naar de artikelenreeks die in het kader van KGN68k gestart zal gaan worden. Mensen die meer willen weten van SCSI in combinatie met een PC worden verwezen naar de artikelenreeks in mc (literatuurlijst 5).

#### 4) AT-bus of EDI: Niet universeel maar wel goedkoop en tamelijk snel

De oorsprong van AT-bus (of ook wel EDI) drives ligt bij de laptop of notebook computers. In deze draagbare computers is over het algemeen maar weinig ruimte beschikbaar en daarom moeten de gebruikte onderdelen bij voorkeur ook zo klein mogelijk zijn. Bij een AT-bus drive zijn (net als bij de moderne SCSI-drives) de controller en de drive samengebouwd in één fysieke behuizing. Voor het aansluiten van de drive aan de PC wordt gebruik gemaakt van een interface-kaartje met daarop enkele buffers en de adresdecodering. Met behulp van een 40 polige kabel wordt de drive met het interface-boardje verbonden. Er zijn zelfs computers waarbij de 40 polige kabel meteen op het motherboard aangesloten kan worden. De 40-polige kabel voert een belangrijk deel van de signalen op de AT-bus en daar komt ook de naam van dit type drive vandaan.

Eén van de vele voordelen van AT-bus boven bijvoorbeeld ST 506 of ESDI is het feit dat de schijven uitgerust kunnen zijn met een foutcorrectie. Mochten er op de schijf sectoren aanwezig zijn die niet gebruikt kunnen worden (bad sectors), dan heeft de schijf een aantal reserve-sectoren die in plaats van een niet-buikbare sector gebruikt worden. In het algemeen heeft elke track een dergelijke reserve-sector. Bij ST 506 of ESDI-drives worden bij het formateren de tracks waarop zich een slechte sector bevindt handmatig ingevoerd en gemarkeerd als niet bruikbaar. Bij AT-bus kan de controller zelf constateren dat een sector slecht is en vervolgens een reserve sector activeren. Om deze reden mag men nooit een AT-bus drive opnieuw formateren omdat dan de informatie over de bad sectors verloren gaat.

Een tweede voordeel van AT-bus is de transfer rate die bij een AT-bus drive gehaald kan worden. Deze ligt zo rond de 1 MB/s en is dus ongeveer vergelijkbaar met ESDI.

AT-bus drives hebben echter ook een nadeel. Er kan in een AT-systeem slechts één AT-bus interfacekaart aanwezig zijn en op deze kaart kunnen slechts twee drives aangesloten worden. Van deze twee drives is de één de master en de andere de slave. Beide drives hebben een controller en deze twee controllers moeten onderling afspreken wie welke taken uitvoert. Om deze reden is het in de praktijk verstandig bij het aansluiten van twee drives te kiezen voor twee drives van dezelfde fabrikant. De reden hiervoor ligt in het feit dat het EDI-interface wel genormaliseerd is maar dat er momenteel nog wat kleine verschillen tussen de drives van de diverse fabrikanten bestaan zodat het voor kan komen dat de controllers in de drives elkaar niet begrijpen.

#### Afsluiting

Over harddisks is nog heel veel te schrijven. We zullen in de toekomst ook echt nog wel op deze materie terugkomen al was het alleen maar omdat KGN-68k een SCSI-interface gebruikt voor communicatie naar de harddisk. Ik wil dit artikel echter nu afsluiten met een anekdote uit de PC-wereld.

Enige tijd geleden vertelde iemand dat de steptijd van een track naar de volgende die hij met een testprogramma bepaalde veel hoger was dan die door de fabrikant van zijn grote, dure harddisk opgegeven werd. Bij het literatuur-onderzoek dat aan dit artikel vooraf ging heb ik ook het antwoord op deze vraag gevonden. Binnen MS-DOS zijn namelijk een aantal maxima gedefinieerd. Zo mag een harddisk maximaal 16 koppen bevatten en maximaal 1024 cilinders. Elke track mag maximaal 256 sectoren bevatten. Nu zijn er schijven met bijvoorbeeld 7 koppen, 1224 cilinders en 35 sectoren per track. Om deze schijven te kunnen gebruiken, is er een truck nodig. Eén truck die wel gebruikt wordt is de volgende: In plaats van 7 koppen en 1224 cilinders definieert men 14 koppen en 612 cilinders. de eerste 7 (logische) koppen lezen dan de oneven (fysieke) cilinders en de tweede 7 logische koppen lezen dan de even (fysieke) cilinders. Op deze manier bestaat een logische cilinder dus uit twee fysieke cilinders en moet de kop van de schijf twee stappen maken voor de volgende logische track en dat kost meer tijd. De omrekening van de logische koppen en cilinders naar de fysieke tracks en cilinders wordt uitgevoerd door de software op de controller. Vooral bij AT-bus drives komt dit voor die dan in een zogenaamde "translated mode" werken.

#### Literatuur:

Tot slot nog een overzicht van de literatuur die ik bij het schrijven van dit artikel gebruikt heb.

- 1) Rainer Hoffman: Wenn die neue Harddisk nicht lauft; mc 4 1987, blz. 36.
- 2) Herbert Cebulla: Festplatten - ein Wunderwerk der Technik; mc 7 1988, blz. 58
- 3) Herbert Cebulla: ESDI, eine neue Winchester Schnittstelle; mc 1 1989, blz. 42
- 4) Manfred Völkel: SCSI: Small Computer System Interface; mc 1 1989, blz. 56
- 5) Rolf-Dieter Klein, Tobias Thiel: SCSI-Adapter für Festplatten; mc 10 1989, blz. 50; mc 11 1989, blz. 122; mc 12 1989, blz. 88; mc 3 1990, blz. 124
- 6) Peter Pilger, Henrik Fisch: Rotierendes Quartet; mc 10 1990; blz. 84
- 7) Henrik Fisch: Platten ohne Macken; mc 10 1990; blz. 74
- 8) Helga M. Schmidt: Marktübersicht Dicke Speicher für kleine Computer; mc 10 1990; blz. 87
- 9) Henrik Fisch: Alles halb so wild; mc 10 1991; blz. 126

NB. Deze opsomming van literatuur is niet bedoeld als reclame voor het Duitse tijdschrift mc, uitgeven door Franzis Verlag. Het toeval wil nu eenmaal dat alle informatie in dit artikel uit de bovengenoemde artikelen in dit blad komt. Mochten er mensen zijn die deze artikelen in willen zien, dan kunnen ze contact met me opnemen.

*Gert van Opbroek*

### Te koop

#### Amiga 500 computer

bestaande uit:

- A500 met interne klok (A501)
- 20 MB harddisk (A590)
- externe floppy disk drive (Profex)
- kleurenmonitor (Ancona)
- 2 MB intern geheugen
- software en boeken

Alles in 1 koop : fl. 2000,-

Hans Kwabek  
Haarlemmerstraat 30  
2312 GA Leiden  
071-126813

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

Ik had gehoopt deze aflevering van "To Share..." te kunnen beginnen met de zinsnede "lang veracht, stil gezwegen...". 4DOS 4.0 zit er namelijk aan te komen, en volgens de auteur (Rex Conn) zit die nieuwe versie stampvol nieuwe handige uitbreidingen. Maar goed, het mocht dus niet zo zijn... Daarom deze keer een blik op een tweetal handige grafische utilities: "the Graphics Workshop" en "PrintGI", respectievelijk versie 6.1 en 1.18.

### Soorten tekeningen

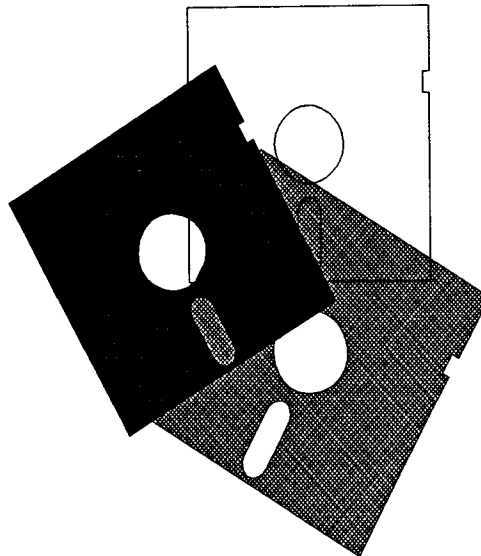
Electronische tekeningen zijn er in verschillende soorten. Allereerst zijn er de zogenaamde bit-images. In een bit-image representeert iedere eenheid in de datafile (meestal een bit of een byte) één pixel. Het nadeel van een bit-image is, dat de file een beperkte resolutie heeft. Ga je een bit-image uitvergrooten, dan worden de afzonderlijke pixels duidelijk zichtbaar.

De tegenhanger van de bit-image is de lijntekening. In een lijntekening wordt de tekening omschreven in termen van vectors of lijnen. Een lijn kan dan vastgelegd worden door een begin- en een eindpunt te definiëren. Daarnaast kan er ook nog informatie vastgelegd worden omtrent de lijn zelf. Gestippeld bijvoorbeeld, of met een breedte van een halve millimeter. Een lijntekening kan, in tegenstelling tot de bit-image, uitvergroot worden zonder verlies van kwaliteit.

### Lijntekeningen

PrintGI is in de eerste plaats bedoeld voor het printen van HPGL tekeningen op een standaard printer. HPGL is een acroniem van "Hewlett Packard Graphics Language", een taaltje dat HP gebruikt voor het aansturen van zijn plotters. Het programma valt uiteen in twee afzonderlijke delen. Er is een standaard UNIX-style deel dat een verschrikkelijk stel parameters heeft. Om het leven gemakkelijker te maken (daar had men immers de computer ooit voor bedoeld!) heeft de auteur er een shell omheen geschreven. Deze shell luistert naar de naam "PrintGI Menu Interface", oftewel "PMI". Via een menu kunnen de verschillende parameters ingesteld worden waarna PrintGI opgestart wordt met de cryptische vertalingen van de ingestelde parameters.

De output mogelijkheden van PrintGI zijn legio. Er zijn drivers voor Epson- en IBM compatible 9 pin printers, Epson-, IBM-, NEC-, en Toshiba compatible 24 pin printers, HP Laserjet, Deskjet, PaintJet, QuietJet en plotters, IBM ExecJet, Quietwriter 2/3, LaserPrinter, Canon LBP en BJ printers, Kodak Color 4, en PostScript printers. Verder kan HPGL



plots weergeven op de standaard schermen (CGA, EGA, VGA, EVGA en hercules) en kan er output gegenereerd worden in ZSoft PCX- en GEM IMG formaten. De laatste twee zijn bit image formaten die door een groot aantal programma's ondersteund worden.

PrintGI's documentatie bestaat uit zo'n 38 pagina's aan uitgebreide informatie. Niet alleen de programmatuur zelf komt aan de orde, ook de structuur van HPGL, ZSoft's PCX en GEM's IMG formaten worden nader toegelicht. De manual is echter vrijwel niet nodig; de menu interface is eenvoudig en logisch opgebouwd.

De printkwaliteit van PrintGI is verrassend goed. Op een klein 9-naalds printertje staat PrintGI behoorlijk lang te kachelen over een A4-tje, maar het resultaat is van voldoende kwaliteit om van een geprintte PCB layout een prototype print te maken.

### Bit images

Het lijkt wel of iedereen die een tekenprogramma maakt, ook een eigen grafische standaard lanceert. Er worden zelfs nieuwe "standaard" formaten bedacht óm de standaard... De meeste direct beschikbare plaatjes zijn in de vorm van Compuserve's GIF-, Apple's MAC- ZSoft's PCX formaat. Graphic WorkShop is een programma dat files kan omzetten tussen allerlei verschillende formaten en daar in de tussentijd ook nog een veelvoud aan bewerkingen op los kan laten. De ondersteunde file formaten zijn weergegeven in tabel 1.

MAC	(MacPaint)
GEM/IMG	(Ventura)
PCX	(PC Paintbrush)
GIF	(CompuServe)
BMP	(Windows 3)
IFF/LBM	(Amiga)
TGA	(Truevision Targa)
MSP	(Microsoft Paint)
WPG	(WordPerfect)
PIC	(PC Paint/Pictor)
TIFF	
EXE	(Self displaying)
TXT	(Text files)
EPS	(PostScript)
CUT	(Halo)

*Tabel 1: door Graphic Workshop ondersteunde formaten*

De plaatjes kunnen tussentijds ook zichtbaar gemaakt worden op het scherm. Hier deed zich echter een probleem voor op de test-computer. Het plaatje verscheen zoals verwacht in beeld en kon naar believen heen- en weer geschoven worden. Wilden we terug naar het hoofdmenu, dan liep het hele zaakje hopeloos vast. Ditzelfde probleem deed zich niet voor op andere systemen, die echter allen voorzien waren van EGA- en VGA schermen. De test computer was de enige met een hercules scherm. Volgens de documentatie wordt hercules echter 100% ondersteund.

De conversies gaan wel erg traag, maar het conversieresultaat is goed. De kwaliteit van de afbeeldingen wil bij dit soort programma's nog wel eens wat te wensen overlaten, maar Graphic WorkShop faalt hier niet. De plaatjes werden zonder uitzondering zonder verlies van resolutie (zoals het ook hoort natuurlijk) foutloos geconverteerd.

Naast het converteren van teksten van het ene formaat naar het andere is het ook mogelijk plaatjes te printen. Ingebouwd zitten drivers voor PostScript- en LaserJet printers, maar via externe drivers kunnen ook de wat eenvoudiger matrix printertjes aangestuurd worden.

De documentatie van GWS beslaat zo'n 126 kByte aan diskruimte. De documentatie is echter niet opgedeeld in pagina's, maar alles is aaneengeplakt tot één groot, onoverzichtelijk geheel. Gelukkig maakt de eenvoudige bediening van het programma weer veel goed.

### Conclusie

Voor zij die regelmatig met grafische gegevens stoeien zijn PrintGl en Graphic WorkShop twee bijzonder aardige programma's. Beiden geven ze de mogelijkheid tot een pre-view en tot het uitprinten op een scala aan printers. PrintGl is wat doelformaat betreft wat soberder uitgevoerd dan Graphic WorkShop en als sourceformaat wordt alleen HPGL ondersteund. Daarentegen zal PrintGl in de wat meer professionele toepassingen zich meer thuisvoelen dan het wat hobbyistisch aandoende GWS.

*Joost Voorhaar*

Besproken produkt	: PrintGl, versie 1.18
Categorie	: utilities
Registratie	: \$40,00
Auteur/leverancier	: Ravitz Software Inc.
Verkrijgbaarheid	: The Ultimate, MS-DOS utilities PrtGl118.Zip, 94 kByte

### Minimale systeemeisen

Minimaal 96 kB RAM, worst case 200 kB

### Algemene beoordeling

Documentatie	: Okido! Engelstalig, dat wel...
Online help	: Njent, nix, noppes.
Gebruikersinterface	: Commandline en eenvoudige menu interface
Muisondersteuning	: Alleen in de menu interface

### Positief

Uitstekende resultaten  
Redelijk snel

**Negatief**

Ietwat onoverzichtelijk menu  
Geen scroll mogelijkheden in screen preview

**Eindbeoordeling**

Stabiliteit : 8  
Bruikbaarheid : 9  
Eindresultaat : 8.5

Besproken produkt : Graphic WorkShop, versie 6.1  
Categorie : Utilities  
Registratie : \$40,00  
Auteur/leverancier : Alchemy Mindworks Inc.  
Verkrijgbaarheid : The Ultimate, Viewers  
GrafWK61.Zip, ca. 344 kB.

**Minimale systeemeisen**

Niet gespecificeerd

**Algemeen**

Documentatie : zonder paginanummering, Engelstalig  
Online help : geen  
Gebruikersinterface : hotkeys, een beetje menu  
Muisondersteuning : nee

**Positief**

Ondersteunt veel formaten  
Geen kwaliteitsverlies tijdens conversie  
Plaatjes kunnen ook uitgeprint worden

**Negatief**

Loopt vast op de testmachine  
Slechte menu sturing

**Eindbeoordeling**

Stabiliteit : 6  
Bruikbaarheid : 7  
Totaal resultaat : 7

## Een extern sorteerprogramma in Pascal

Over sorteren kun je boeken vol schrijven, sterker nog er zijn al boeken over volgeschreven. Om ook een kleine bijdrage aan deze papierberg te leveren, heb ik een bestaand sorteerprogramma een beetje aangepast en ingetypt onder Turbo Pascal.

Van sorteren bestaan er twee varianten. De eerste variant is het sorteren van een reeks gegevens die in het geheugen van de computer staan. Dit legt meteen een beperking op aan de hoeveelheid gegevens die je kunt sorteren; ze moeten namelijk wel allemaal in het geheugen van de computer passen. De tweede variant sorteert gegevens die op een zogenaamde sequential file staan. In principe kunnen deze files zo groot zijn als de ruimte die je op devices voor massageheugen hebt. Dit mogen bijvoorbeeld tapes zijn.

De eerste variant noemt men intern sorteren en de algoritmen die daarvoor bestaan hebben fraai klinkende namen als "Bubble sort", "Shaker sort", "Shell sort", "Quick sort" "Heap sort" etc. De tweede variant noemt men extern sorteren en één van de bestaande algoritmen is in het bijgevoegde programma verwerkt. Dit programma werkt met in totaal 3 files, de file die gesorteerd moet worden en twee hulpfiles. Zouden die files op tape staan, dan moet je dus drie tapes gelijktijdig in je computer kunnen verwerken. De file die als invoer gebruikt wordt, wordt ook weer gebruikt voor de uitvoer. De oorspronkelijke informatie gaat dus verloren.

Het eigenlijke algoritme is vrij simpel. Je begint met het lezen van de invoerfile. Zolang de regels netjes in volgorde staan, kopieer je de regels naar file a. Kom je een regel tegen die eigenlijk voor de vorige regel zou moeten staan, dan kopieer je die regel naar b. Vervolgens blijf je kopiëren naar b net zolang totdat je weer een regel tegenkomt die weer voor de vorige regel zou moeten komen. Je gaat dan weer verder met a enzovoort tot je aan het einde van de invoerfile bent. In de tweede fase lees je de twee file a en b terug en maak je weer een file c. Je doet dit door steeds een regel van a te vergelijken met een regel van b en de kleinste naar c te kopiëren. Is één van de twee files leeg, dan kopieer je uiteraard de rest van de file die nog niet leeg is. Dit proces wordt herhaald totdat alle regels in volgorde staan.

Voor de echte snelheidsfanaten is dit programma niet geschikt. Het algoritme is nog vrij eenvoudig. Er bestaan algoritmen die veel sneller zijn. Die maken onder andere gebruik van meerdere hulp-files en combineren bovendien het extern sorteren met een intern sorteeralgoritme. Een volgende keer zal ik een dergelijk formule-1 programma laten zien.

Voor mensen die meer willen weten van sorteren of andere interessante algoritmen, verwijs ik naar het boek dat in de listing genoemd wordt. Ik heb dat boek in mijn bezit en mensen die het in willen zien kunnen contact met mij opnemen.

*Gert van Opbroek*

```
PROGRAM mergesort(input,output);

{ Dit programma sorteert een tekstfile d.m.v. het algoritme
  3-tape, 2 fase natural merge sort.

  De kern van het programma is overgenomen uit:
  Nicklaus Wirth: Algorithms + Data Structures = Programs
  uitgegeven door Prentice-Hall.

  Het programma is enigszins gemodificeerd en aangepast voor Turbo Pascal
  door G. van Opbroek in oktober 1991
}
TYPE  item = string;
      tape = TEXT;

VAR   c           : tape;           { file to sort           }
      eof_c       : BOOLEAN;       { end-of-file marker   }
```

*Fig. 1: sourcetekst van SORT.PAS*

```

filename      : STRING;           { filename to sort  }
s_start,s_length : INTEGER;       { sort key start,length }

```

```
PROCEDURE list (VAR f : tape);
```

```
{ Deze procedure toont een tekstfile. Hij wordt gebruikt om het verloop
van het sorteerproces te tonen.
}
```

```
VAR    x : item;
```

```
BEGIN
```

```
  reset(f);
  WHILE NOT eof(f) DO
  BEGIN
    readln (f,x); writeln(output, x)
  END;
  writeln;
END; { list }
```

```
PROCEDURE naturalmerge;
```

```
{ Deze procedure verzorgt het eigenlijke sorteren van de file die
aangeduid wordt met de globale variabele c.
}
```

```
VAR l : INTEGER;           { number of runs merged }
    eor: BOOLEAN;         { end-of-run indicator }
    a,b: tape;           { scratch files }
    eof_a,eof_b : BOOLEAN; { end-of-file markers }
```

```
FUNCTION compare (x,y : item) : BOOLEAN;
```

```
{ Deze functie bepaalt de volgorde na sortering. Aan de hand van
de globale variabelen s_start en s_length wordt bepaald welke
delen van de parameters x en y met elkaar vergeleken moeten worden.
}
```

```
VAR sortkeyx,sortkeyy : STRING;
```

```
BEGIN
```

```
  IF s_start > length(x)
  THEN sortkeyx := ""
  ELSE IF s_start + s_length > length(x) + 1
  THEN sortkeyx := copy(x,s_start,length(x) - s_start + 1)
  ELSE sortkeyx := copy(x,s_start,s_length);
```

```
  IF s_start > length(y)
  THEN sortkeyy := ""
  ELSE IF s_start + s_length > length(y) + 1
  THEN sortkeyy := copy(y,s_start,length(y) - s_start + 1)
  ELSE sortkeyy := copy(y,s_start,s_length);
```

```
  compare := sortkeyx > sortkeyy
```

```
END;
```

```
PROCEDURE copy (VAR x,y : tape; VAR buf : item; VAR eof_x : BOOLEAN);
```

```
{ Copieert een regel van x naar y en controleert of de volgende
regel ook gekopieerd mag worden. Omdat Turbo Pascal niet toestaat
```

in het file-buffer van een file te lezen zonder de file-pointer te verzetten, wordt gebruik gemaakt van een eigen file-buffer in buff.

```

}
VAR buf1 : item;

BEGIN
  IF NOT eof_x THEN writeln(y, buf); eof_x := eof(x);
  IF eof_x
  THEN eor := TRUE
  ELSE BEGIN
    readln(x,buf1); eor := compare(buf,buf1); buf := buf1;
  END;
END;

PROCEDURE copyrun (VAR x,y : tape; VAR buf : item; VAR eof_x : BOOLEAN);

  { Stuur copy aan een maakt op deze manier een zogenaamde kopieerrun.
  }
  BEGIN
    REPEAT copy(x,y,buf,eof_x); UNTIL eor
  END;

PROCEDURE distribute;

  { Verdeelt de regels in c zodanig over a en b dat de blokken met
  regels in de juiste volgorde groter worden.
  }
  VAR buf : item;

  BEGIN
    eof_c := eof(c);
    IF not eof_c THEN
      BEGIN
        readln(c,buf);
        REPEAT copyrun(c,a,buf,eof_c);
          IF NOT eof_c THEN copyrun(c,b,buf,eof_c)
        UNTIL eof_c
      END;
    END;

PROCEDURE mergerun(VAR buf1,buf2 : item);

  { Kopieert een blok vanuit de file met het kleinste startpunt
  naar c. Stop op het moment dat de regels niet meer in de juiste
  volgorde staan.
  }
  BEGIN
    REPEAT
      IF NOT compare(buf1,buf2) THEN
        BEGIN
          copy(a,c,buf1,eof_a);
          IF eor THEN copyrun(b,c,buf2,eof_b);
        END ELSE
        BEGIN
          copy(b,c,buf2,eof_b);
          IF eor THEN copyrun(a,c,buf1,eof_a)
        END;
    END;

```

```

        UNTIL eof
        END;

PROCEDURE merge;

    { Kopieert m.b.v. mergerun de blokken (runs) in a en b naar c
    }
    VAR buf1,buf2 : item;

    BEGIN
        eof_a := eof(a); eof_b := eof(b);
        readln(a,buf1); readln(b,buf2);
        WHILE NOT eof_a OR NOT eof_b DO
            BEGIN
                mergerun(buf1,buf2); l := l + 1;
            END;
        WHILE NOT eof_a DO
            BEGIN
                copyrun(a,c,buf1,eof_a); l := l + 1;
            END;
        WHILE NOT eof_b DO
            BEGIN
                copyrun(b,c,buf2,eof_b); l := l + 1;
            END;
        list(c);
    END;

BEGIN

    { Body van naturalmerge. Splitst en voegt weer samen totdat de
    file (c) in volgorde staat.
    }
    assign(a,'klad1.del'); assign(b,'klad2.del');
    REPEAT
        rewrite(a); rewrite(b); reset(c);    distribute;
        reset(a); reset(b); rewrite(c); l := 0; merge;
    UNTIL l = 1;
    erase(a); erase(b);
END;

BEGIN
    { Hoofdprogramma. Leest de filenaam in en het startpunt en de lengte
    van de sorteersleutel.
    }
    write('Sortfile: '); readln(filename);
    assign(c,filename);
    write('startposition of sort key      :'); readln(s_start);
    write('lenght of sort key              :'); readln(s_length);
    IF s_start <= 0 THEN s_start := 1;
    IF s_length <= 0 THEN s_length := 255;
    list(c); naturalmerge; list(c);
END.

```

## De KGN-68k computer

### Rapportage van de werkgroep KGN-68k

#### Hetzelfde

Eén van de opmerkingen die we van de leden gehoord hebben is de volgende: De rapportage in het blad is eigenlijk altijd hetzelfde. Het begint met een inleiding, vervolgens komt er een zeer globale beschrijving van de status, zonder IC-nummers etcetera, met als boodschap "wie meer wil weten moet er maar naar vragen", waarna afgesloten wordt met een oproep aan de leden. Deze keer zal het niet echt anders zijn. Ook nu beginnen we met een globale beschrijving en sluiten we af met een oproep aan de leden. Daar tussenin komt er echter toch wat meer specifieke informatie over wat het allemaal gaat worden. IC-nummers worden nog niet genoemd, maar dat scheelt niet veel meer.

#### Kathedralen

Weet je hoe bijvoorbeeld de Sint Jans kathedraal in Den Bosch gebouwd is? Wel, men begon met het bouwen zonder dat er duidelijke tekeningen waren. Op basis van ervaring wisten de vaklieden dat het zinvol was te beginnen met de fundering en de afwerking als laatste te doen, maar hoe en in welke volgorde dat zag men wel tijdens het bouwen. En zo ontstonden de prachtigste kerken, kathedralen paleizen en piramides.

Hetzelfde is gebeurd binnen de werkgroep KGN-68k. De specificatie die gebruikt werd was "Er moet een 68020 of -030 in" en "Het moet heel hard lopen". Verder heeft de projectgroep zich eigenlijk nauwelijks met het vastleggen van een specificatie beziggehouden en de creativiteit en vakmanschap het werk laten doen. Op deze manier is de projectgroep nu een jaar bezig en zo is er een definitief schema ontstaan. Vervolgens is men vol goede moed begonnen met het ontwerpen van een print. Dat gaf wel wat problemen, de printruimte op een full size AT-insteekkaart was namelijk wel wat krap om alle gewenste componenten op onder te brengen. Welnu, niet getreurd, je gooit het AT-bus interface eraf en in de plaats daarvan zet je er een Floppy- en SCSI-interface op en dan moet het kunnen. Inderdaad, het kon maar... er bleef na het plaatsen van de componenten te weinig ruimte over om alle sporen te leggen. Al weer niet getreurd, de creatieve oplossing bestond uit het omgooien van het mechanische concept waarin de computer ondergebracht werd op eeninsteekkaart. In plaats daarvan wordt de compu-

ter op de ruimte van een AT-motherboard geplaatst. Het AT-bus interface wordt niet op deze print ondergebracht maar op een Amiga-achtige manier via een bridge board en dat wordt de KGN-68k computer.

Er bleken ook nog problemen te zijn met het pakket waarmee de print ontworpen wordt maar ach, dan neem je een ander pakket en dan zijn ook die problemen weer over. Wel bleef als doelstelling overeind dat er een draaiend stuk hardware op de HCC-dagen zou zijn en daarvoor was het nodig dat het bestuur begin oktober toestemming zou geven een prototype te bouwen zonder dat bekend was hoeveel de onderdelen kosten, wat voor print het zou worden en erger: zonder voldoende nacontrole van schema en printontwerp. Wel, dat deed het bestuur dus mooi niet. De penningmeester was, terecht, niet bereid een blanco cheque ter beschikking te stellen en andere bestuursleden wilden toch op zijn minst een fatsoenlijke specificatie en begroting hebben. Kortom, de projectgroep werd door het bestuur op de vingers getikt en kreeg als opdracht eerst een fatsoenlijke specificatie en kostenraming te overleggen voordat er geld voor het bouwen van een prototype beschikbaar gesteld zou worden.

### De rapportage in het blad is eigenlijk altijd hetzelfde.

Welnu, dat is de status. Er bestaat een specificatie van de KGN-68k, inclusief een uitge-

breide schemabeschrijving en een kostenraming. Ondertussen wordt er hard gewerkt aan het layouten van de print en het aanpassen van een monitor en bijbehorend ontwikkel-systeem die door de firma Besamu te Hengelo ter beschikking gesteld zijn. Op basis van de specificatie gaat de projectgroep het bestuur opnieuw vragen geld voor het bouwen van een prototype ter beschikking te stellen waarna de bedachte hardware met behulp van de monitor en de ontwikkel-omgeving getest kan gaan worden. Parallel daaraan wordt er uiteraard gewerkt aan het porteren van het MINIX operating systeem naar de KGN-68k computer.

#### Systeembeschrijving

Aangezien er een specificatie van het systeem bestaat, kan de projectgroep nu ook aan de leden bekend maken wat het gaat worden en hoe het er uit komt te zien. De volgende paragrafen zijn een kort uittreksel uit de globale systeembeschrijving in de specificatie.

In het plan van aanpak werd als opdracht aan de projectgroep geformuleerd dat de hardware in de vorm van een insteekkaart voor de AT-bus ontwikkeld moest te worden. Deze insteekkaart moest samen kunnen werken met (goedkope) kaarten uit de AT-wereld waaronder bijvoorbeeld een gecombineerde floppy- harddisk controller, een herculeskaart etc. Hierbij werd in eerste instantie uitgegaan van een concept waarin de processor van de AT stilgezet, danwel verwijderd zou worden. Het zou ook mogelijk zijn een busprint te gebruiken met daarop een passieve AT-bus.

Nadat de projectgroep van start gegaan was, werd onderzocht of het mogelijk was de KGN-68k samen te laten werken met een draaiende AT-processor. De twee processoren zouden dan beide gebruik kunnen maken van de op de AT aangesloten periferie. Het bleek echter dat het niet mogelijk was de in de AT aanwezige DMA-controller opdracht te geven de bus vrij te maken zodat een DMA-controller op de KGN-68k de beschikking over de (AT) bus kon krijgen. Dit betekende dat er, binnen de randvoorwaarden in de opdracht, twee systemen ontworpen zouden moeten worden, één systeem dat zou kunnen werken in een AT met daarin een werkende DMA-controller en een tweede systeem dat zelf een DMA-controller heeft om te kunnen werken in een systeem gebaseerd op een passieve AT-bus.

Een tweede probleem dat al spoedig opdoemde was de beperking van de printruimte. Een insteekkaart bleek al snel veel te klein te zijn om alle benodigde componenten op te plaatsen en bovendien bleek dat het aantal lagen in de print de vier zeker zou gaan overschrijden.

Omdat vooral de prijs van SCSI-harddisks het afgelopen jaar sterk gedaald is, werd het mogelijk het systeem op te bouwen als volledige stand-alone computer. In feite komt het er op neer dat het toevoegen van een Floppy- en SCSI-interface relatief gezien nauwelijks geld kosten maar de computer veel veelzijdiger maken. Bovendien gaf dit de projectgroep de mogelijkheid eerst het computersysteem zonder het interface naar de AT-bus te bouwen en pas in tweede instantie het, vrij gecompliceerde, AT-bus interface. Het idee daarbij was, de KGN-68k computer op een full size AT-insteekkaart te plaatsen zonder dat de connector voor de AT-bus aangesloten wordt. Deze insteekkaart zou een tweede busconnector hebben waarmee hij kon communiceren met een tweede insteekkaart. Deze tweede insteekkaart vormde het interface naar de AT-bus.

Bij het ontwerpen van de print bleek echter dat ook voor dit concept de printruimte van een insteekkaart nog onvoldoende was. Bovendien hadden een aantal

leden van de projectgroep bedenkingen tegen het mechanische concept van een insteekkaart met aan de bovenkant nog een busconnector voor communicatie met een tweede insteekkaart.

De projectgroep heeft daarom gekozen voor een concept dat ook bij de Amiga 2000 gebruikt is. De KGN-68k computer wordt opgebouwd op een print ter grootte van een full size AT-motherboard. Dit motherboard zal zodanig geconstrueerd worden dat het past in de bestaande kasten voor full size AT-motherboards. Op dit board wordt de volledige stand-alone versie van de computer opgebouwd, inclusief SCSI-controller, floppy-controller, een DU-ART, geheugen etc. doch exclusief een interface naar de AT-bus. Wel zal er op de print ruimte zijn voor in totaal acht AT-busconnectoren die onderling verbonden zijn. Deze connectoren vormen, bij uitbreiding, een passieve AT-bus.

In het verlengde van de connectoren voor de AT-bus worden extra 96-polige DIN 41612 connectoren geplaatst die een, voor de KGN-68k apart gespecificeerde, bus met signalen van de Motorola processor vormen. Op deze manier kan men uitbreidingskaarten op de AT-bus aansluiten en uitbreidingskaarten op de Motorola-bus. Minstens één van de kaarten zal gebruik maken van beide aansluitingen. Dit is de kaart met het interface tussen de KGN-68k en de AT-bus. Met behulp van deze interfacekaart (bridge board) zal de KGN-68k in staat zijn samen te werken met uitbreidingskaarten uit de AT-wereld. Op de interface kaart zal bovendien een aansluiting voor een toetsenbord komen zodat men een console-device krijgt bestaande uit een AT-videokaart en een los toetsenbord. Wil men in één systeem zowel de beschikking hebben over een KGN-68k als een AT, dan kan men een processorkaart voor een passieve AT-bus gebruiken. Uiteraard moet de interfacekaart hier dan wel op voorbereid zijn.

De mechanische opbouw van een dergelijk systeem is getekend in figuur 1.

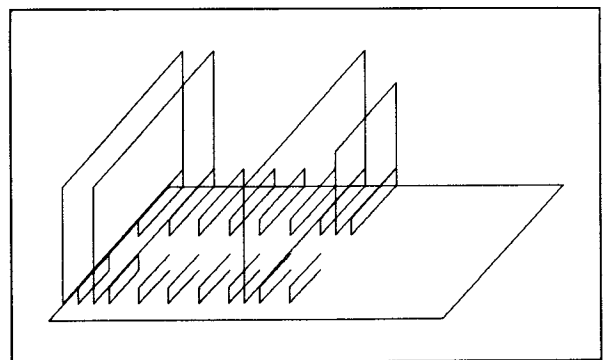


Fig. 1: mechanische opbouw van de KGN-68k computer

## Hardware

De KGN-68k computer zal in eerste instantie gebouwd worden als volledig stand-alone systeem. Het hart van dit systeem zal bestaan uit een Motorola mc68030 processor geklokt met een frequentie van minimaal 16 MHz.

In het systeem zit minimaal 4 MB geheugen, de standaard-hoeveelheid bestaat uit 4 1 MB SIPP-modules. Door toevoeging nog eens 4 extra SIPP-modules wordt naast een vergroting van het geheugen ook een geringe verbetering van de performance verkregen. Verder bestaat de mogelijkheid de 1 MB modules te vervangen door 4 MB modules waardoor de totale geheugen-capaciteit op het motherboard 32 MB wordt. Een eventuele verdere uitbreiding van het geheugen kan ondergebracht worden op aparte uitbreidingskaarten.

Behalve het hoofdgeheugen beschikt de KGN-68k ook nog over 32 kB SRAM met battery backup en 64 kB EPROM. Deze geheugen-onderdelen worden 8 bits breed uitgevoerd. Het SRAM is bedoeld voor het vastleggen van een aantal SET $\mu$ P parameters, de EPROM bevat het monitor-programma, de bootstrap en de BIOS.

Het systeem krijgt de beschikking over een bi-directionele centronics-poort voor het aansluiten van de systeem-printer. Doordat de poort vanuit de software ook als input-poort te configureren is, kan ze ook voor andere doeleinden gebruikt worden.

De gebruiker communiceert met de KGN-68k via één of twee RS-232C poorten waarop een terminal of ander systeem met terminal-emulatie is aangesloten. Na uitbreiding komt er een mogelijkheid rechtstreeks een video-kaart en een toetsenbord op de computer aan te sluiten.

Het massa-geheugen van de computer zal bestaan uit een 3,5" HD floppy-disk drive en één of meer SCSI harddisks. Het floppy-interface ondersteunt, behalve het HD-formaat, ook het DD en SD formaat. Dit betekent dat alle gangbare 3,5" en 5,25" floppy drives ondersteund worden met uitzondering van de 1,2 MB 5,25" HD-floppy drive. Op het floppy-interface kunnen tot maximaal 3 floppy-drives aangesloten kunnen worden. Het SCSI-interface bestaat uit een SCSI-host adapter waarop maximaal 7 andere SCSI devices aangesloten kunnen worden. Het SCSI-interface is vanuit de software als initiator en als target te definiëren. De Floppy- en SCSI-interfaces communiceren naar de drives via een soort pseudo DMA in de vorm van een 2 kB FIFO-buffer.

In het systeem wordt een real time clock opgenomen. Deze verzorgt ook de battery backup van de aanwezige SRAM.

Behalve de bovengenoemde onderdelen komen er voor de computer een aantal opties. Deze staan in de volgende alinea's beschreven.

De computer wordt voorbereid op het plaatsen van een mc68881 of mc68882 Floating Point Unit. Een dergelijke FPU wordt als coprocessor van de mc68030 CPU in het systeem opgenomen.

Een tweede optie is het AT-bus interface waarmee het mogelijk wordt in het systeem uitbreidingskaarten die ook in XT's en AT's gebruikt worden op te nemen.

Tegelijk met de tweede optie krijgt de KGN-68k ook een keyboard en muis-interface om, in combinatie met een video-kaart, een intern console-device te vormen.

Aangezien de computer rijkelijk voorzien wordt van uitbreidingsconnectoren, zijn er nog zeer veel uitbreidingskaarten denkbaar. Het is de bedoeling dergelijke uitbreidingskaarten, bij voldoende belangstelling, binnen de KGN te gaan ontwikkelen.

### Monitor, BIOS en bootstrap

Voor de KGN-68k zal een monitor-programma ontwikkeld worden waarmee men op eenvoudige wijze de hardware van het systeem kan testen. Deze monitor heeft de mogelijkheid te communiceren met een ontwikkel-omgeving die onder MS-DOS op een PC draait. De volledige beschrijving van deze monitor is in de specificatie van de KGN-68k computer opgenomen.

Behalve het monitor-programma zullen ook de basis-functies voor de input en output in de EPROM ondergebracht worden. Om dit mogelijk te maken, wordt er een standaard-interface naar het operating system gedefinieerd.

Om een operating system van schijf te kunnen laden, wordt in de EPROM een bootstrap-programma opgenomen. Dit programma zal in staat zijn naar keuze een operating system van floppy of een operating system van de harddisk in te lezen.

### Operating system

Voor het systeem zal de KGN een port maken van het operating system MINIX naar de KGN-68k computer. Dit operating system zal door de KGN volledig ondersteund worden. Uitgegaan wordt van

de versie voor de Atari ST waarbij met name bij het memory management zo optimaal mogelijk gebruik gemaakt zal worden van de in de mc68030 aanwezige faciliteiten in de vorm van caching en een MMU.

Behalve het MINIX operating system zal ook de GNU C-compiler voor het systeem beschikbaar komen en door de KGN ondersteund gaan worden.

#### Input van leden

De projectgroep heeft tot nu toe een behoorlijke input van de leden gehad. Twee mensen die contact met de projectgroep opgenomen hebben zijn meteen bij de projectgroep ingelijfd en hebben een behoorlijke inbreng in het geheel gehad. Hoewel er al een behoorlijk aantal mensen in de projectgroep werken, vinden we toch dat we op enkele punten nog wel wat versterking kunnen gebruiken.

In de eerste plaats zou het prettig zijn als de projectgroep versterkt zou kunnen worden met een echte MINIX-goeroe. Het porteren van MINIX is een be-

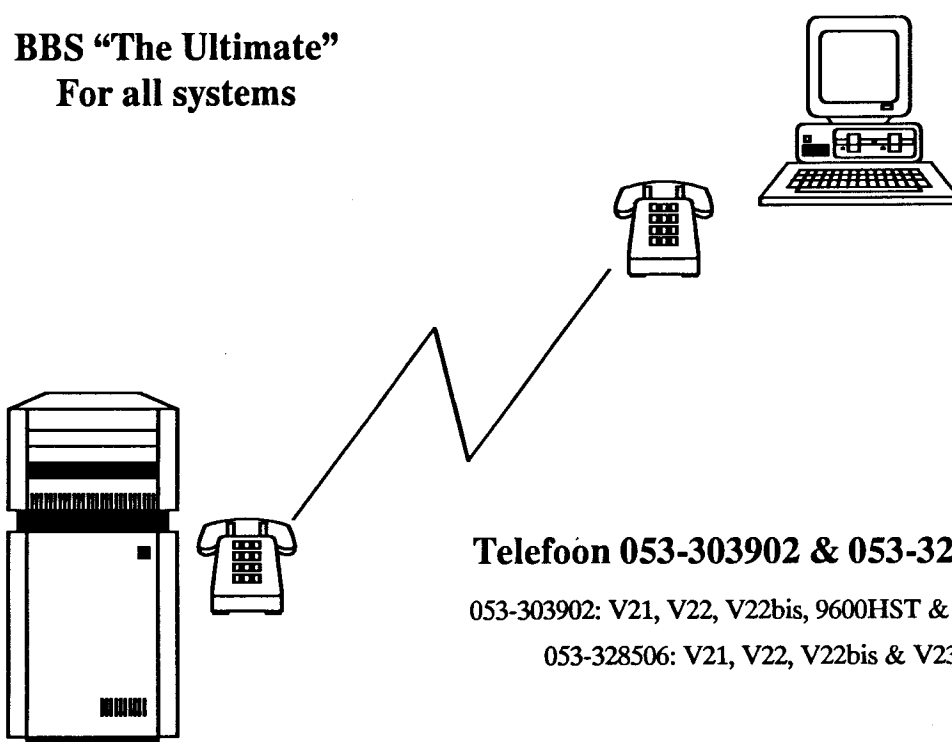
hoorlijke klus en binnen de projectgroep is wel wat MINIX-kennis aanwezig maar een echte goeroe ontbreekt nog. Hetzelfde geldt eigenlijk voor de hele software-tak. Het merendeel van de leden van de projectgroep zijn hardware-deskundigen en een uitbreiding met een echte (assembler of C) softwerker zou zeer welkom zijn.

In de tweede plaats is een uitbreiding op het gebied van micro-controllers wenselijk. Het is de bedoeling dat het bridge board met het businterface ook een aansluiting voor een keyboard en een muis krijgt. Voor deze sub-systemen denken we een micro-controller in te zetten en mensen die ervaring met deze onderdelen hebben worden verzocht te reageren.

Tot zover de rapportage van de projectgroep. Je ziet, de trein is in beweging en nu maar zorgen dat de vaart er in blijft.

*Geert Stappers*

**BBS "The Ultimate"**  
**For all systems**



**Telefoon 053-303902 & 053-328506**  
053-303902: V21, V22, V22bis, 9600HST & V42bis  
053-328506: V21, V22, V22bis & V23

## Methoden en technieken voor datacommunicatie (Deel 9)

### Inleiding

Mede naar aanleiding van een reactie van Hugo van der Kooij wil ik het in deze aflevering van de datacom-serie gaan hebben over het koppelen van netwerken. Hierbij kun je denken aan het koppelen van meerdere LAN's binnen een bedrijf tot een groter netwerk maar bijvoorbeeld ook aan het koppelen van een computernetwerk in Amsterdam aan een computernetwerk in Timboektoe. Om vooral deze koppelingen over grote afstand te kunnen beschrijven, gaan we ons ook even heel vluchtig bezighouden met het zogenaamde X.25 netwerk.

### Reactie op de vorige aflevering

Naar aanleiding van de vorige aflevering over datacommunicatie kreeg ik via het bulletin board The Ultimate de volgende reactie van Hugo van der Kooij.

Gert,

Naar aanleiding van je stuk over datacommunicatie in de  $\mu$ P Kenner nr 72 de volgende reactie.

In je verhaal noem je twee soorten die veel voorkomen, namelijk EtherNet en ARCnet. Maar je lijkt die Token-Ring netwerken over het hoofd te zien. Terwijl toch een fiks deel van de markt wordt ingenomen door TKR. Verder ken ik naast dun (10base2) en dik (10base5) Ethernet ook nog Fiber-Optic (glasvezel; 10baseF) en UTP (unshielded twisted pair; 10baseT). Deze twee gaan in snel tempo dun en dik Ethernet vervangen. Ook de genoemde kabel lengtes bij dun en dik zijn veel te lang. 10base2 staat voor 10 Mb, baseband (recht toe recht aan modulatie) 2(00) meter. En volgens de specs is dit officieel slechts 185 meter. 10base5 mag maximaal 500 meter zijn en de aansluitingen mogen alleen daar op de kabel worden gemaakt waar de kabel is gemarkeerd met een ring.

Verder miste ik de uitleg WAAROM gelet moet worden op een collision als al gekeken is of niet iemand anders die kabel gebruikt. De verklaring zit in de traagheid van elektrische signalen. In een standaard ethernet kabel is dat niet meer dan 0,6 maal de lichtsnelheid. Een signaal kan dus al op het andere eind van de kabel staan maar nog niet gezien worden omdat dat wel 0,5 tot 1 microseconde kan duren en in die tijd kunnen al 5 tot 10 bit op de kabel zijn gezet.

In de praktijk blijkt bij een goed netwerk ontwerp de responsetijd van Ethernet erg mee te vallen. Een belasting tot 20 a 25 procent is nog geen probleem. Om zoiets voor elkaar te krijgen heb je toch al gauw een paar HEEL actieve SUN-Sparc stations nodig, of iets van die orde.

Mocht je nog vragen hebben over netwerken en datacommunicatie, ik zit al dik twee jaar in de markt en mag me enigszins deskundig noemen. Succes verder met je serie en tot horens/ziens/modems.

CU, Hugo.

### Antwoord

Ik vind het altijd prettig reacties op mijn artikelen te krijgen. Helaas gebeurt dat naar mijn mening veel te weinig. Ik wil Hugo dan ook hartelijk bedanken voor de aanvullingen op het vorige artikel.

Nu even over de inhoud van zijn opmerkingen. Hij geeft zelf al aan dat hij een beetje een specialist is op het gebied van netwerken. Uit zijn opmerkingen valt dat ook duidelijk af te lezen. Ik ben absoluut geen specialist in de hele netwerk-materie. Via mijn werk heb ik momenteel een klein beetje te maken met een netwerk waarop enkele VAX mini-computers van de firma DEC zijn aangesloten. Verder interesseer ik mij in deze materie en tracht ik mijn kennis te vergroten door het lezen van de artikelen die ik over netwerken tegenkom en het boek Computers

Networks van Tanenbaum. Maar dat geeft allemaal niet, via deze artikelenreeks komt er toch een beetje discussie los waardoor de kennis bij iedereen die actief of passief aan deze discussie deelneemt aangescherpt wordt.

Hugo noemt in zijn reactie een vrij populaire Token Ring netwerken. Die heb ik inderdaad over het hoofd gezien. Ik bezit een beetje informatie over Novell en over Arcnet en ben me er zeker van bewust dat de wereld van de PC-netwerken niet bij deze twee ophoudt, in tegendeel, er komen in rap tempo steeds weer nieuwe netwerkproducten op de markt. Verder moet ik eerlijk toegeven dat de PC in het algemeen mij totaal niet interesseert met als hoofdreden het feit dat er, naar mijn persoonlijke mening, een verkeerde processor in zit; ik gebruik de PC zelf

### Via deze artikelenreeks komt er toch een beetje discussie los .

alleen maar als een soort veredelde schrijfmachine, houdt me beroepshalve bezig met VAXen en uit hobby met computers die zijn gebaseerd op de Motorola 68000 familie.

De lengtes die Hugo noemt voor de Ethernetkabels kloppen. In Computers Networks op pagina 143 staat inderdaad dat de maximale lengte van een (dikke) Ethernet-kabel 500 meter is. Verder staat er op pagina 142 dat aftakkingen van een dergelijke Ethernet-kabel alleen gemaakt mogen worden op de plaatsen waar de (gele) kabel gemarkeerd is. Deze plaatsen liggen 2,5 meter uit elkaar.

Wil je een groter netwerk aanleggen dan de specificatie toestaat, dan is dat wel degelijk mogelijk, je moet dan in het netwerk een ding opnemen dat zich gedraagt als een versterker. Een dergelijk apparaat noemt men een repeater. In de specificatie van Ethernet wordt aangegeven dat er tussen een ontvanger en een afzender maximaal 4 repeaters mogen zitten en dat de onderlinge afstand, gemeten langs de kabel, maximaal 2500 meter mag bedragen. Wel mag je, binnen de specificatie, allerlei vertakkingen in het netwerk maken zolang je er maar voor zorgt dat er nooit meer dan vier repeaters tussen twee stations liggen en dat de kabellengte tussen twee stations nooit meer wordt dan 2500 meter.

Met behulp van die afstand van 2500 meter kan ik ook uitleggen waarom je, ondanks het feit dat een station dat wil gaan zenden eerst controleert of de lijn vrij is, je toch nog een collision kunt krijgen. Hugo geeft aan dat de snelheid waarmee een signaal zich over de kabel voortplant 60% van de lichtsnelheid is. Bij een lichtsnelheid van 300.000 m/s duurt het ruim 10 micro seconden voordat het signaal van de ene kant van het netwerk getransporteerd is naar de andere kant. Hierbij komt nog de tijd die de repeaters nodig hebben. Bij een snelheid van 10 MBit/s heeft een zender dus al ruim 100 bits verstuurd voordat een station aan de andere kant dat überhaupt op de lijn kan detecteren. Al die tijd denkt het station ten onrechte dat de lijn vrij is en zou dus zelf ook kunnen besluiten te gaan zenden, hetgeen onherroepelijk tot een collision zal leiden. Bij kortere afstanden is het natuurlijk veel gunstiger omdat het "tijdgat" dan veel kleiner is; maar het kan toch altijd nog voorkomen dat een station net in de tijd die nodig is om de

kop van het signaal over de kabel te transporteren besluit te gaan zenden.

Tenslotte nog een opmerking over uitvoeringen van Ethernet met behulp van glasvezel of twisted pairs. Strikt genomen zijn dit geen Ethernet-netwerken omdat de Ethernet-specificatie het gebruik van een coax-kabel voorschrijft. De oorspronkelijk, door Xerox in Paolo Alto ontwikkelde Ethernet-specificatie is later als IEEE 802.3 CMA/CD netwerk gespecificeerd en gestandaardiseerd. Deze specificatie en de naam Ethernet worden tegenwoordig vaak onderling verwisseld. Onder de IEEE 802.3-specificatie vallen een hele familie CMA/CD netwerken met snelheden van 1 tot 10 MBit/s en draaiend met behulp van diverse media, zoals coax-kabel, twisted pairs en glasvezel.

#### X.25, een soort internationaal telefoonnet voor computers

Zoals met behulp van de telefoon en de diverse telefoonnetwerken mensen verspreid over de hele wereld met elkaar kunnen communiceren, zijn er nationale en internationale, openbare netwerken waarlangs computers over de hele wereld met elkaar kunnen communiceren. Evenals een lokaal netwerk kunnen ook de openbare netwerken beschreven worden volgens het OSI 7 laags model. Er zijn voor deze netwerken ook specificaties opgesteld; in dit geval de CCITT (International Consultative Committee for Telephony and Telephony), de overkoepelende standaardisatie organisatie van de PTT-bedrijven. Deze specificaties zijn bekend geworden als de "X.25" aanbevelingen. De X.25 specificatie is ook door de ISO overgenomen en is de standaard voor openbare computernetwerken. Binnen het OSI model beschrijft X.25 de onderste drie lagen. Dat wil zeggen dat een X.25-netwerk ervoor kan zorgen dat packets met informatie van computer A naar computer B verstuurd worden en andersom. Wat er met die informatie binnen computer A en B gebeurt is een zaak van de hogere lagen in het OSI-model en wordt niet in de X.25-specificatie vastgelegd.

Omdat je wereldwijd veel meer knooppunten in een netwerk kunt hebben dan in een lokaal netwerk, is er voor X.25 een codering voor de stations vastgelegd die veel overeenkomsten vertoont met een telefoonnummer. Een stations-adres bevat maximaal 14 deci-

**Bij een snelheid van 10 MBit/s heeft een zender dus al ruim 100 bits verstuurd voordat een station aan de andere kant dat überhaupt op de lijn kan detecteren.**

male digits. Hiervan zijn de eerste drie het zogenaamde landnummer, gevolgd door een net(werk)nummer. De laatste 10 digits zijn bestemd voor het abonneenummer. Mocht een groot land meer dan 10 netwerknummers nodig hebben, dan worden er extra landnummers gebruikt. Zo zijn voor de Verenigde Staten de landnummers 301 t/m 329 gereserveerd en voor Canada 302 t/m 307. Nederland heeft landcode 204.

Een voorbeeld van een openbaar netwerk dat volgens X.25 werkt is het nederlandse netwerk Data-netwerk Nr. 1 (afgekort DN 1) dat sinds 1981 beschikbaar is en dat (volgens mijn enigszins verouderde gegevens) verbindingen kan verzorgen met snelheden van 2400, 4800, 9600 en 48000 bits per seconde. Binnen DN 1 kunnen zowel vaste verbindingen als geschakelde verbindingen gedefinieerd worden.

### Repeaters, bridges en routers

Na het uitstapje over openbare netwerken, gaan we ons weer in hoofdzaak bezighouden met de lokale netwerken of LAN's. We gaan in deze paragraaf eens kijken welke doosjes je tegenkomt als je meerdere netwerken gaat koppelen.

In de inleiding is de term "repeater" al even gevallen. Dit is niet meer dan een versterker die de informatie op de netwerkkabel versterkt. Een dergelijke repeater werkt dan ook bit voor bit en zal verder

niets met de inhoud van de informatie doen. Met behulp van dergelijke repeaters wordt het signaal op de kabel als het ware weer wat opgepept. In repeater koppelt twee netwerken via de eerste laag, de physical layer, in het OSI-model, zoals getekend is in figuur 1.

Bij een bridge worden twee netwerken met elkaar gekoppeld. Deze twee netwerken hoeven zelfs niet van hetzelfde type te zijn, zoals bijvoorbeeld in figuur 1 getekend is. Wel is het een vereiste dat de network layer voor beide netwerken gelijk is zodat de informatie hoe je van A naar B komt in beide netwerken op dezelfde manier in het packet verwerkt wordt. De packets die via de bridge van het ene netwerk naar het andere moeten, worden ingelezen in de bridge, ontdaan van de informatie die speciaal voor de data link layer (laag 2) toegevoegd is en omgezet in een bericht dat over het andere netwerk vervoerd kan worden. Uiteraard moeten de overige lagen (3 t/m 7) in de beide netwerken wel gelijk zijn. Als we even weer de figuren 2 en 3 uit de vorige aflevering er bij nemen (figuur 2 en 3), dan zien we dat de source en destinations adressen bij Ethernet en Token Bus beide 48 bits zijn. De lengte van de data is kan bij Token Bus wat groter zijn maar dat is ook geen echt probleem, je splitst bij de overgang van Token Bus naar Ethernet het packet gewoon op. De rest van de informatie (preamble, delimiters en de data length) worden in een bridge tussen de twee typen netwerken er af gehaald of er bij gemaakt.

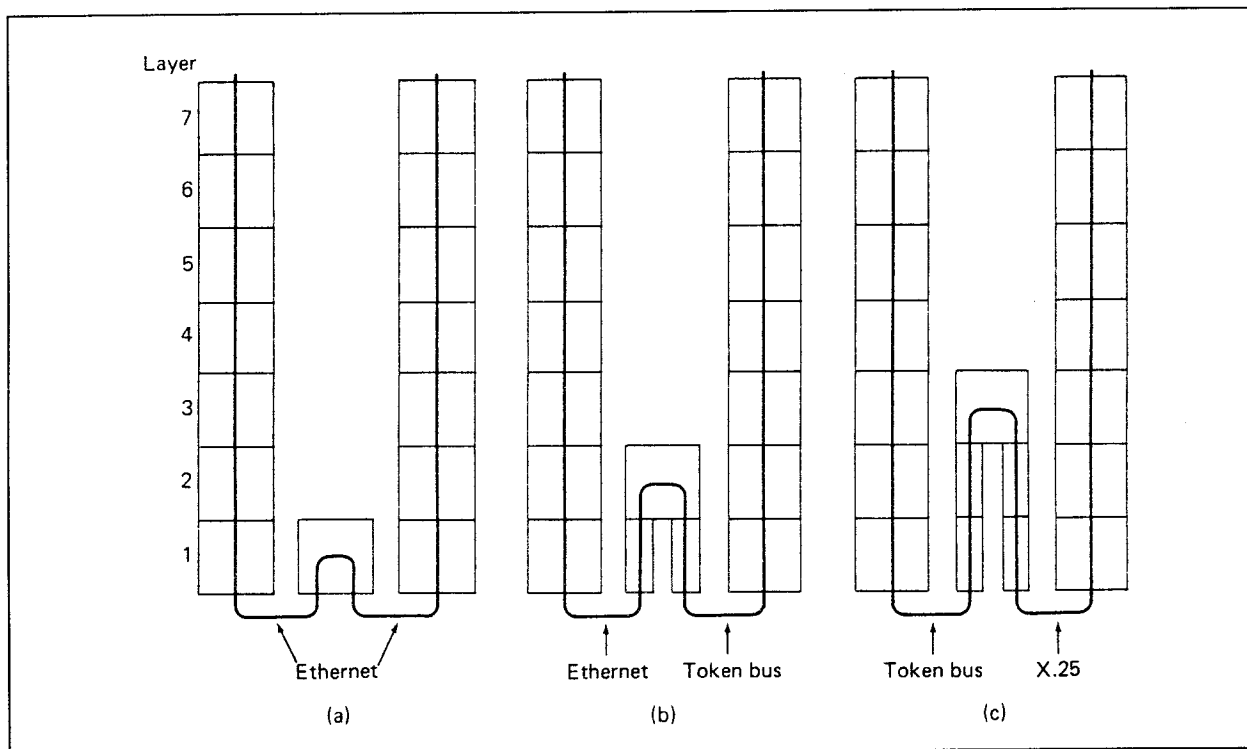


Fig. 1: een repeater (a), een bridge (b) en een router (c)

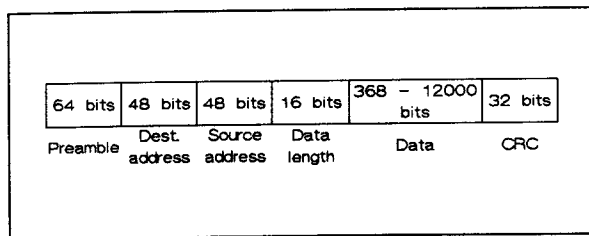


Fig. 2: opbouw van een Ethernetpacket

Gaan we nog een niveau hoger, dan spreken we over een router. Ook deze apparaten lezen een bericht in maar doen er nog wat meer mee. Bij een router hoeft ook de netwerk layer (laag 3) niet meer gelijk te zijn. Dat betekent dat een router in staat is de adressen van afzender en geadresseerde in het ene netwerk om te zetten in adressen in het andere netwerk. Als voorbeeld is in figuur 1 een koppeling gelegd tussen een Ethernet netwerk en een X.25 netwerk. Binnen het X.25 netwerk worden de stations aangesproken via een codering die veel lijkt op de codering in het openbare telefoonnetwerk. Een compleet nummer kan tot maximaal 14 decimale digits bevatten. Nu kun je in de 48 bits die je in Ethernet voor een adres hebt wel alle X.25-nummers coderen maar echt zinvol is dat niet. Je kunt veel beter de paar stations waarmee je over het netwerk wilt communiceren een adres binnen je eigen Ethernet netwerk geven en vervolgens in de router een tabel opnemen welk X.25-adres bij welk intern adres hoort. De router kan dan netjes de informatie in een X.25 packets omzetten naar Ethernet packets en andersom.

Nu zijn er zo langzamerhand al heel veel zogenaamde OSI-netwerken. Dit zijn de netwerken die volgens het OSI 7 laags model opgebouwd zijn. Helaas is dit nog lang niet voor alle netwerken het geval. Eén van de veel voorkomende afwijkingen is het SNA-netwerk (Systems Network Architecture) van IBM. Dit netwerk heeft een iets andere opbouw dan een OSI netwerk en om deze reden kunnen OSI netwerken niet m.b.v. routers of bridges gekoppeld worden aan een SNA netwerk. Om een koppeling tussen een OSI netwerk en een SNA netwerk tot stand te brengen moet je als het ware alle 7 lagen in

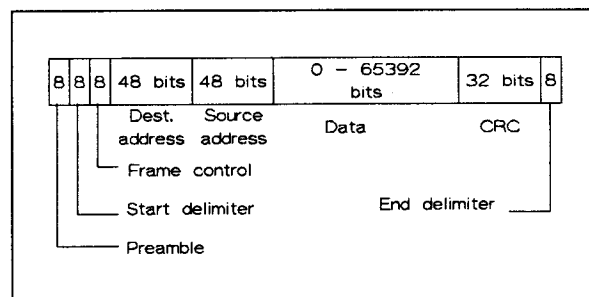


Fig. 3: packet-formaat in een Token Bus

het OSI netwerk omzetten naar de lagen in het SNA netwerk. Dit wordt gedaan m.b.v. een zogenaamde Gateway.

Samenvattend:

- Een repeater werkt binnen één en hetzelfde netwerk.
- Een bridge koppelt twee netwerken van hetzelfde soort of twee netwerken waarvan alleen de physical layer en de data link layer verschillen.
- Een router koppelt twee netwerken waarvan, behalve laag 1 en 2 ook laag 3 verschilt. De overige lagen (4 t/m 7) zijn echter wel gelijk.
- Een gateway koppelt twee netwerken waartussen meer verschillen zijn dan in de onderste drie lagen van het OSI model.

Tenslotte nog even een leuk gevolg van standaard 802 netwerken (Ethernet 802.3, Token Bus 802.4 en Token Ring 802.5). Deze netwerken zijn op laag niveau verschillend. Op hoog niveau zijn ze echter precies hetzelfde. De verschillen tussen de netwerken worden midden in laag 2 (de data link layer) teniet gedaan. De onderste helft van de data link layer, de MAC (Medium Access Control) sublayer is voor de drie netwerken verschillend, de bovenste helft de LLC (Logical Link Control) sublayer is echter voor de drie netwerken hetzelfde en daarmee automatisch de specificatie van de lagen 3 t/m 7.

Als gevolg hiervan kunnen de diverse netwerk-pakketten met al deze netwerken omgaan. Meestal worden er voor de netwerk-pakketten zogenaamde netwerk-drivers meegeleverd die het interface naar bijvoorbeeld Ethernet, ARCnet of TKR vormen. Een tweede gevolg is dat de drie 802-netwerken altijd met behulp van bridges aan elkaar gekoppeld kunnen worden. Dit blijkt ook al uit het voorbeeld dat in de vorige paragraaf gegeven is.

Afsluiting

Ik heb me in deze aflevering een beetje laten leiden door de reactie van Hugo van der Kooij. In de volgende aflevering zal ik trachten de draad van de serie weer op te pakken en in te gaan op ondermeer TCP/IP, het netwerk-protocol dat steeds meer in de belangstelling komt te staan.

Verder houd ik mij uiteraard aanbevolen voor open aanmerkingen per telefoon, fax, bulletin board, brief, koffievsite of welk menselijk communicatienetwerk u ook maar kiest.

Gert van Oproek

## Van de bestuurstafel

Hi... Whe're ye from? Holland? Do they still wear wooden shoes and ride bikes? No? Pity... Tja, daar zit je dan. Net terug van een zeer geslaagde vakantie. Doorgebracht in het land van de ongekenne mogelijkheden. Aan de overkant van de Atlantische Oceaan dus. Voor mijn vakantieadres moet je nog verder: aan de kust van de Pacific. Daar was het schitterend weer: 35 graden (Celsius, daar denken ze nog altijd in Fahrenheit) zodat ik het in het pietepouterige Hollandse maar knap koud vind. Het was daar dus opperbest uit te houden (met de airco aan, dat wel). Al die warmte zorgde overigens wel voor een nijpend watertekort, terwijl alles in de natuur kurk- en kurkdroog is geworden. Gevolg: spontane branden in bermen en bossen. In Berkeley (UNIX/universiteitsstad) en Oakland (voorstad van San Francisco) weten ze daar inmiddels alles van...

Wat dit alles met onze club te maken heeft? Niet zoveel denk ik. Maar je moet toch wat om een inleiding te hebben. Want van de afgelopen twee maanden ben ik bijna vijf weken foetsie geweest. Zodat er bijna geen stof tot typen is. Uiteraard is alles wel verder zijn gang gegaan. In Haarlem was de opkomst matig naar ik heb begrepen. Ik was er ook niet: ik had Ruud Uphoff z'n verhaal al eens meegeemaakt, en er moesten nog koffers gepakt worden en zo. Voor diegenen die er niet waren: U heeft het verhaal van Ruud niet gemist, want hij was helaas ziek. Het vervangende verhaal was echter minstens zo interessant. Vroegere club-hyper-activering Erwin Visschedijk deed een boekje open over zijn modelbaanproject. Ik kan u verzekeren: dat hyper kan bij Erwin nog altijd niet van het woord actief worden gescheiden, want de gehele zolder in Wierden wordt zo ongeveer opgeslokt door een gigantische modelspoorbaan die wordt bevolkt door tenminste tien NS-treinen en minstens zoveel computers. De laatste keer dat ik bij Erwin op visite was werkte er al het en ander en het geheel belooft een zeer indrukwekkend en knap project te worden. Misschien horen we er in de toekomst meer van in de uP-Kenner.

In diezelfde vakantietijd is er ook een bestuursvergadering gehouden. Daar kan ik dus bijna niets van vertellen, want toen was ik in het land van de McDonalds hamburgers (over 95 billion served). Bij de KGN-68k groep is men druk bezig met het printontwerp voor het prototype van het bord, op het BBS blijkt de nieuwe harde schijf het opperbest te doen, volgens de penningmeester zijn we financieel gezond

en zelfs de redactie heeft nauwelijks moeite om dit nummer vol te krijgen. Dit nummer komt wel te laat. Die stomme layouter gaat ook veel te lang op vakantie...

Over de volgende bijeenkomst is ook het een en ander te melden: dat is de gebruikelijke jaarlijkse algemene ledenvergadering. Ook gebruikelijk is de plaats: Almelo. Voor sommigen misschien een beetje ver rijden, maar het is er altijd gezellig. Vergaderen is misschien minder interessant dan computerknoeien, maar het een noodzakelijk kwaad. Bovendien heeft u de gelegenheid als lid om het bestuur op de vingers te tikken over zaken waar u het mogelijk niet mee eens was. Sterker nog: u krijgt de kans om zich actief met het reilen en zeilen van de club bezig te houden! Er treden 3 bestuursleden af. Dat houdt in dat er plaats is voor drie nieuwe. Er ontstaan zelfs drie vacatures, want geen der aftredende figuren stelt zich om uiteenlopende redenen herkiesbaar. Joost Voorhaar heeft zich kandidaat gesteld. Het zou leuk zijn eens meer kandidaten te hebben dan openstaande bestuursplekken.

U kunt zich kandidaat stellen door van twee clubleden een handtekening te vragen en u hiermee te schriftelijk te melden bij ondergetekende. Voor de bijeenkomst is het leukst, maar het mag zelfs op de bijeenkomst, voor de stemming. Probeer het ook eens. De minimumduur is 1 jaar, al hopen we dat het gemiddelde bestuurslid het wat langer naar zijn zin heeft in het bestuur. Ik kan uit ervaring melden, dat het geen moeite kost om 8 jaar vol te maken...

Tenslotte een mopje, voortbordurend op de opening. Zoals u misschien weet hebben ze in Los Angeles zeer vaak smog. Ik ving het volgende gesprekje op:

- A: "Where are you from?"  
 B: "Los Angeles"  
 A: "You must have a tough time here in Las Vegas!"  
 B: "Why should I?"  
 A: "...Breathing air you can't see..."

See ya in Almelo,

*Nico de Vries*

## Informatie

De  $\mu$ P 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  $\mu$ P Kenner uitgegeven. Verder worden er door het bestuur per jaar 5 landelijke bijeenkomsten georganiseerd, beheert het bestuur een Bulletin Board en wordt er een softwarebibliotheek en een technisch forum voor diverse systemen in stand gehouden.

## Landelijke bijeenkomsten:

Deze worden gehouden op bij voorkeur de derde zaterdag van de maanden januari, maart, mei, september en november. De exacte plaats en datum worden steeds in de  $\mu$ P 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 technische ondersteuning streeft het bestuur ernaar zgn. systeemcoördinatoren te benoemen. Van tijd tot tijd zal in de  $\mu$ P 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)  
C.P. Soeteliefstraat 41  
1785 CC Den Helder  
Telefoon 02230-35002

Gert van Opbroek (redactie  $\mu$ P 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