

KERKLOH, TORNSDORF,
ZOLLER

HET AMIGA-DOS HANDBOEK



DATA BECKER
NEDERLANDS*

HET AMIGA-DOS HANDBOEK

THE UNIVERSITY OF CHICAGO

PHYSICS DEPARTMENT

PHYSICS 354

**KERKLOH, TORNSDORF,
ZOLLER**

HET AMIGA-DOS HANDBOEK

**DATA BECKER
NEDERLANDS***

Oorspronkelijke titel: Das grosse Buch zu Amiga-DOS
Copyright © 1987 Data Becker GmbH, Düsseldorf
Voor de Nederlandse vertaling © 1988 A.W. Bruna Uitgevers BV, Utrecht/Antwerpen
Vertaling: LINE UP tekstproducties

ISBN 90 229 3478 0
D/1988/0939/35
NUGI 854

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Kerkloh

Het Amiga-DOS handboek / Kerkloh, Tornsdorf, Zoller ; [vert. uit het Duits: Line up tekstproducties]. - Utrecht [etc.] : Bruna. - Ill. - (Data Becker)
Vert. van: Das grosse Buch zu Amiga-DOS. - Düsseldorf : Data Becker, 1987. - Uitg. van Data Becker Nederlands. - Met reg.
ISBN 90-229-3478-0
SISO 525.1 UDC 681.3.066 NUGI 854
Trefw.: Amiga-DOS (operating system).

De uitgever heeft ernaar gestreefd ervoor te zorgen dat de informatie in dit boek correct is. Op het gebied van de behandelde onderwerpen vinden echter snelle ontwikkelingen plaats. Data Becker Nederlands stelt zich daarom niet aansprakelijk voor de juistheid van de informatie op het moment van publikatie.

Niets uit deze uitgave mag worden veelevoudigd, in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door middel van fotokopieën, opnamen, of op welke wijze ook, zonder voorafgaande schriftelijke toestemming van de uitgever.

Boeken en programma's van Data Becker Nederlands worden in de handel gebracht door:

A.W. Bruna Uitgevers BV
Postbus 8411, 3503 RK Utrecht
en
A.W. Bruna & Zoon NV
Cornelis Schutstraat 28, 2100 Deurne-Antwerpen

Inhoud

1	Inleiding	9
1.1	De taak van het DOS	9
1.2	Workbench of CLI	9
1.3	Vorbereidingen	10
1.4	De CLI meldt zich	11
1.5	De eerste handelingen	11
1.6	De directory-structuur	12
1.7	De Help-functie	14
1.8	De CLI verlaten	15
2	De CLI-instructies	16
2.1	Instructies voor diskettes en bestandsbeheer	16
2.1.1	FORMAT	16
2.1.2	DIR	17
2.1.3	CD	18
2.1.4	MAKEDIR	19
2.1.5	DELETE	19
2.1.6	COPY	20
2.1.7	RENAME	21
2.1.8	LIST	23
2.1.9	DISKCOPY	25
2.1.10	RELABEL	26
2.1.11	INFO	26
2.1.12	INSTALL	27
2.1.13	TYPE	28
2.1.14	JOIN	29
2.1.15	SEARCH	29
2.1.16	SORT	30
2.1.17	PROTECT	30
2.1.18	FILENOTE	31
2.1.19	SETDATE	31
2.1.20	DISKDOCTOR	32
2.1.21	DISKCHANGE	32
2.2	Systeminstructies in de CLI	33
2.2.1	NEWCLI	33
2.2.2	ENDCLI	34
2.2.3	RUN	34
2.2.4	STATUS	35
2.2.5	CHANGTASKPRI	35
2.2.6	BREAK	35
2.2.7	PATH	36
2.2.8	ASSIGN	37
2.2.9	ADDBUFFERS	38
2.2.10	WHY	38
2.2.11	FAULT	39

2.2.12	DATE	39
2.2.13	SETCLOCK	40
2.2.14	PROMPT	40
2.2.15	STACK	41
2.2.16	BINDDRIVERS	41
2.2.17	MOUNT	41
2.3	Instructies voor batchbestanden	42
2.3.1	EXECUTE	42
2.3.2	ECHO	44
2.3.3	FAILAT	44
2.3.4	QUIT	45
2.3.5	IF/ELSE/ENDIF	45
2.3.6	ASK	47
2.3.7	SKIP en LAB	48
2.3.8	WAIT	49
2.4	De editors ED en EDIT	50
2.4.1	Overzichtelijke tekstverwerking met ED	51
2.4.2	Flexibele tekstbewerking met EDIT	51
3	Tips en trucs voor de CLI	57
3.1	Invoer en uitvoer in de CLI	57
3.2	Jokers	57
3.3	Onderbrekingen in de CLI	58
3.4	De standaard-devices van de Amiga	60
3.4.1	Floppydrive, DF	60
3.4.2	De RAM-disk	60
3.4.3	Het parallelle device	61
3.4.4	Het seriële device	62
3.4.5	Het printer-device	63
3.4.6	Het CON-device	63
3.4.7	Het RAW-device	64
3.5	De RAM-disk en de CLI	65
3.6	Printen vanuit de CLI	67
3.6.1	Bestanden printen met COPY	68
3.6.2	Algemene opmerkingen over omleiden van de uitvoer	68
3.6.3	Stuurtekens voor een printer	69
3.7	Toepassingen van het CON-device	70
3.8	Gebruik van de seriële interface	73
4	Batch processing met de Amiga	77
4.1	Inleiding	77
4.1.1	Wat is een batchbestand?	77
4.1.2	De structuur van batchbestanden	77
4.1.3	Batchbestanden oproepen	78
4.1.4	Een eenvoudig voorbeeld	78
4.2	Wijzigingen in de startup-sequence	79
4.3	Een bijzonder batchbestand voor de printer	84
4.4	Zelf batch-instructies maken	86

5	AmigaDOS en multitasking	89
5.1	Multitasking	89
5.2	Multitasking met CLI en Workbench	90
5.3	Multitasking met NewCLI	92
5.4	Multitasking met Run	93
5.5	Effectief gebruik van de CLI	95
5.6	ChangeTaskPri	97
5.7	Punten ter overweging	99
6	AmigaDOS intern	101
6.1	Programma's, segmenten en BPTR-pointers	101
6.2	Het programma CLI	102
6.3	De interne DOS-bibliotheek	104
6.4	Voorbeeld: RUN	106
7	Nieuwe CLI-instructies	109
7.1	Programmeren in C	110
7.2	TASK	113
7.3	TASKPRI	115
7.4	TASKSTOP	118
7.5	DELAY	119
7.6	BASIC-laders voor de CLI-instructies	123
7.6.1	BASIC-lader voor TASK	124
7.6.2	BASIC-lader voor TASKPRI	131
7.6.3	BASIC-lader voor TASKSTOP	140
7.6.4	BASIC-lader voor DELAY	150
8	Quick Reference	160
8.1	ED	160
8.2	EDIT	161
8.3	CLI-instructies	163
8.3.1	CLI-instructies in functiegroepen	163
8.3.2	Alfabetische lijst van CLI-instructies	173
9	Workbench versie 1.3	176
9.1	De CLI-instructies	177
9.2	Nieuwe handlers en devices	192
	Trefwoordenregister	196

1. The first part of the document discusses the importance of maintaining accurate records of all transactions. It emphasizes that proper record-keeping is essential for the integrity of the financial system and for the ability to detect and prevent fraud. The text also notes that clear and concise reporting is necessary for effective communication between different levels of management.

2. The second part of the document focuses on the role of internal controls in ensuring the reliability of financial information. It describes how a well-designed system of internal controls can help to minimize the risk of errors and misstatements. The text also discusses the importance of regular audits and reviews to ensure that the internal control system is effective and up-to-date.

3. The third part of the document addresses the issue of transparency and accountability in financial reporting. It argues that providing clear and accessible information to stakeholders is crucial for building trust and confidence in the organization. The text also discusses the importance of disclosing any potential risks and uncertainties that may affect the financial performance of the organization.

4. The fourth part of the document discusses the role of technology in improving financial reporting and internal controls. It highlights how the use of advanced software and systems can help to automate many of the manual processes involved in financial reporting, thereby reducing the risk of errors and increasing the efficiency of the reporting process. The text also discusses the importance of ensuring that any technology used is secure and reliable.

5. The fifth part of the document discusses the importance of ongoing training and education for staff involved in financial reporting and internal controls. It argues that regular training is essential to ensure that staff are up-to-date on the latest developments in financial reporting and internal controls. The text also discusses the importance of providing staff with the necessary resources and support to enable them to perform their duties effectively.

6. The sixth part of the document discusses the role of external audits in providing an independent assessment of the organization's financial reporting and internal controls. It argues that external audits are essential for providing stakeholders with confidence in the organization's financial information. The text also discusses the importance of ensuring that external audits are conducted in a fair and objective manner.

7. The seventh part of the document discusses the importance of maintaining a strong ethical culture within the organization. It argues that a strong ethical culture is essential for ensuring that all staff act in a fair and honest manner. The text also discusses the importance of providing staff with clear guidance on the organization's ethical standards and the consequences of any breaches.

1 Inleiding

In de eerste paragrafen van dit hoofdstuk willen we u zo snel mogelijk op weg helpen met de CLI en zullen we u dus zo weinig mogelijk lastig vallen met theorie. In deze inleiding vindt u voldoende informatie om met de CLI aan de slag te kunnen. In hoofdstuk 6 vindt u een grote hoeveelheid achtergrondinformatie over de CLI.

1.1 De taak van het DOS

Om zinvol over de CLI te kunnen spreken, is het nodig een paar begrippen van het Disk Operating System (DOS) toe te lichten. DOS is een sterk schijfgeoriënteerd besturingssysteem. Een besturingssysteem is een verzameling programma's die nodig is voor de werking van de computer. Het besturingssysteem is geen toepassingsprogramma zoals een tekstverwerker of een elektronische kaartenbak. De routines en procedures van het besturingssysteem vormen het fundament waarop u een programma bouwt. Het besturingssysteem beheert het geheugen, vertaalt de signalen van en naar de hardware (toetsenbord, scherm, printer, enzovoorts) en coördineert die activiteiten. Er zijn ook kant en klare functies beschikbaar. U hoeft bijvoorbeeld zelf geen procedure te bedenken die berekent welke geheugengedeeltes nog vrij zijn. Als u een programma laadt, krijgt dat automatisch een voldoende groot gedeelte toegewezen.

Een kenmerk van het besturingssysteem DOS is dat een groot aantal instructies die de computer kan uitvoeren, niet vast zijn geïntegreerd in DOS. (Bij veel home computers herkent het besturingssysteem sommige instructies rechtstreeks en voert ze meteen uit: LOAD, SAVE, RUN, enzovoorts). De instructies van AmigaDOS zijn programma's op schijf die gebruik maken van de systeemfuncties. De naam van een instructie fungeert als oproep van zo'n programmaatje vanaf een diskdrive (floppy disk, harddisk of RAM-disk). Deze werkwijze heeft een aantal voordelen boven geïntegreerde instructies:

1. Een instructie neemt slechts geheugen in beslag zolang ze in werking is. Daarna wordt ze meteen gewist.
2. Het is mogelijk een instructie later te vervangen door een verbeterde versie.
3. Het aantal instructies is niet onderhevig aan beperkingen. Uitbreiding van DOS met extra instructies blijft mogelijk.

De nadelen komen vooral tot uiting bij kleine configuraties met weinig geheugen en maar één diskdrive. Het steeds laden van de instructies kost tijd en veelvuldig diskettes wisselen gaat irriteren. Deze nadelen verdwijnen bij gebruik van een harddisk of RAM-disk.

1.2 Workbench of CLI

U moet de DOS-instructies op een of andere manier doorgeven aan de computer. In de Workbench maakt u daarvoor gebruik van menu's en de muis. Formatteren van een diskette gaat als volgt: u activeert met de muis de gewenste diskette, selecteert het menu 'Disk' en kiest daar de optie 'Initialize'. De instructie die daar bijhoort, wordt geladen van de Workbench-diskette en vervolgens uitgevoerd. Goed beschouwd is de Workbench een programma dat van diskette wordt geladen en het gebruikersniveau weergeeft met plaatjes.

De Command Line Interface (CLI) doet dat op een soberder manier. Dit hulpmiddel is een 'doorgeefluik' voor instructieregels van u naar de computer. U activeert een opdracht hier niet met de muis, maar voert die in via het toetsenbord binnen een eenvoudig venster op het scherm. De muis gebruikt u alleen om eventueel de afmetingen van dat CLI-venster te veranderen.

De CLI lijkt een stap terug vergeleken met het bedieningsgemak van de Workbench. Als u de Amiga alleen af en toe aanzet voor tekstverwerking, hebt u de CLI zelfs helemaal niet nodig. Dat wordt anders zodra u het beginnersniveau ontstijgt. Sommige taken zijn in de Workbench-omgeving slechts langs lange omwegen of zelfs helemaal niet te verrichten. Veranderen van de 'startup-sequence' is bijvoorbeeld uitsluitend mogelijk via de CLI. Deze opstartprocedure die op de Workbench-diskette staat, wordt elke keer dat u de Amiga start afgehandeld. Lang niet alle bestanden die op de Workbench-diskette staan, krijgt u te zien in het venster van de Workbench. Met de CLI hebt u wel toegang achter de coulissen van het Amiga-toneel.

1.3 Voorbereidingen

Maak in ieder geval een veiligheidskopie om dagelijks mee te werken voordat u met de CLI gaat werken. Al doende zult u merken dat u een of meer programma's niet nodig hebt en dat het op dat moment handiger is die te wissen. Later verandert u misschien weer van gedachten en dan hebt u de originele diskette nog om de ontbrekende programma's toch weer te kopiëren. Beschadiging van de diskette en onbedoeld wissen is ook nooit helemaal uit te sluiten.

Vanuit de Workbench is het heel eenvoudig een kopie te maken. Voor de volledigheid vatten we de procedure puntsgewijs samen.

1. Schuif - voorzover dat niet al eerder is gebeurd - het schuifje voor schrijfbeveiliging van de Workbench-diskette naar buiten. Dit verhindert onbedoeld beschrijven van de originele Workbench-diskette. Het schuifje heeft dezelfde functie als de lipjes achterop een muziek-cassette.
2. Doe de Workbench-diskette in de ingebouwde diskdrive 'df0:' en start het systeem door gelijktijdig de Ctrl-toets en de twee toetsen links en rechts van de spatiebalk in te drukken. Zodra u de toetsen loslaat begint de computer programma's te laden.
3. Na enkele ogenblikken verschijnt de bekende afbeelding van de Workbench. De Workbench-diskette wordt gesymboliseerd door een tekeningetje (icon) rechtsboven. Activeer dat symbool met een klik van de linker muisknop en kies in het menu 'Workbench' de optie 'Duplicate'.
4. De procedure vraagt u de FROM-diskette (brondiskette) in de diskdrive te doen. Dat is in dit geval niet nodig omdat die al in de drive zit.
5. Tijdens het kopiëren zult u een paar maal de bron- en de doeldiskette moeten verwisselen. Vergeet niet het schrijfbeveiligingsschuifje van de doeldiskette te sluiten.
6. Ter onderscheiding van de originele diskette krijgt de nieuwe diskette behalve zijn naam het voorvoegsel 'Copy of'. Met behulp van 'Rename' kunt u die toevoeging eventueel weer weghalen. In dat geval kan DOS het origineel en de kopie toch nog van elkaar onderscheiden aan de hand van de datum en de tijd. (Deze gegevens worden altijd met de actuele waarden genoteerd.)

1.4 De CLI meldt zich

Start de Amiga opnieuw, maar nu met de kopie van de Workbench. (Als u dat achterwege laat, vraagt de Amiga steeds naar de originele diskette. Berg die diskette veilig op.) Kies in het menu 'Disk' de optie 'System'. In de archiefla die nu opengaat behoort de CLI te zitten. Als dat klopt, kunt u de volgende twee alinea's overslaan.

De makers van de Amiga hebben een beveiliging aangebracht. Als die is ingeschakeld, krijgt u het CLI-symbool niet te zien en kunt u de CLI dus niet starten. Dat is natuurlijk niet de bedoeling.

Sluit het 'System'-venster en activeer daarna het programma 'Preferences'. Na enkele ogenblikken verschijnt er een groot venster. Naast veel andere dingen staat daarin links een kader met de naam CLI. Activeer met de muis het direct daarnaast staande veld 'On' en daarna het veld 'Save' rechts in het venster. De nieuwe configuratie wordt dan opgeslagen op de diskette. De volgende keer dat u de archiefla 'System' opent, is ook bij u de CLI-icon zichtbaar.

Laad en start de CLI met een dubbele klik met de muis op de CLI-icon. Er verschijnt een venster met de naam 'New CLI Window'.

De geopende archiefladen op de achtergrond mogen nu dicht: ze zijn niet meer nodig. Het CLI-venster heeft bijna alle eigenschappen van de andere vensters van de Amiga. U kunt de plaats en de grootte veranderen; de gadgets voor voor- en achtergrond zijn er ook. Alleen het afsluitsymbool in de linker bovenhoek ontbreekt. In plaats daarvan is er een instructie die u via het toetsenbord in moet voeren (zie paragraaf 1.8).

Binnen het CLI-venster staat alleen de DOS-prompt. Die bestaat uit een of meer tekens die melden dat de computer klaar is nieuwe opdrachten op te nemen en uit te voeren die afkomstig zijn van het toetsenbord. Bij de CLI bestaat de prompt uit het cijfer '1' en een punthaak (groter dan-teken). Daarnaast wacht de cursor op invoer.

1.5 De eerste handelingen

Sluit al uw opdrachten in de CLI af met een druk op Enter of Return (de toets met de gebogen pijl). Deze twee toetsen zijn geheel gelijkwaardig. In het vervolg noemen we alleen nog maar de Return-toets.

Een losse Return verplaatst de prompt naar de volgende regel. De vier cursortoetsen hebben in het CLI-venster geen effect. Het is dan ook niet mogelijk een hoger in het venster staande opdracht te herhalen met behulp van een Return. Elke opdracht moet u steeds opnieuw volledig invoeren. Binnen de actuele regel kunt u alleen van rechts naar links een eenmaal getypte tekst verwijderen met de Backspace-toets (links naast de Del-toets). Wis de actuele regel in zijn geheel door gelijktijdig indrukken van de toetsencombinatie Ctrl-X (de prompt blijft daarbij staan).

Alleen instructies die in het CLI-woordenboek staan, kunnen worden uitgevoerd. De opdracht

inhoudsopgave

leidt tot de reactie

Unknown command inhoudsopgave

In paragraaf 1.1 bleek al dat de instructies van de CLI niet in de CLI zelf staan, maar dat het eigenlijk namen zijn van programma's die na oproep worden gestart. Het bijzondere van de CLI is dat u aan de lijst met DOS-instructies eigen namen kunt toevoegen. De CLI zoekt op de diskdrive het gelijknamige programma, laadt dat en start het in het werkgeheugen van de computer.

Het heeft weinig zin hier een lange lijst af te drukken met de standaardinstructies die op de Workbench-diskette staan. De instructie DIR (directory) is een eenvoudige, maar belangrijke instructie. Ze levert de inhoudsopgave van elk type diskdrive. Het maakt niet uit of u een instructie typt met kleine letters, hoofdletters of als mengsel van die twee. Geef vooraf het CLI-venster de maximale afmeting want het resultaat van de functie vraagt nogal wat ruimte. Voer de opdracht in:

```
dir <Return>
```

Op het scherm verschijnt na korte tijd de inhoudsopgave van de diskette in diskdrive 0 (Workbench-diskette). Dat dat niet erg snel gebeurt, is een gevolg van de manier waarop de boekingen op de diskette worden geadmineistreerd. Als u het venster van de CLI niet groot genoeg hebt gemaakt, kan het voorkomen dat het al vol is terwijl nog niet alle boekingen zijn uitgevoerd. De uitvoer gaat door, ook al bereikt een regel de bovenrand van het venster. Onderbreek de uitvoer door een willekeurige toets in te drukken; door Backspace komt de uitvoer weer op gang.

1.6 De directory-structuur

In de lijst met namen die het resultaat is van de instructie DIR, komt u er vast en zeker een aantal tegen die u kent van de Workbench. Die bestandsnamen staan tweemaal in de lijst: eenmaal apart en een keer met de extensie (het achtervoegsel) '.info'. Omgekeerd treft u een bestand zonder die extensie niet aan in de Workbench. Een .info-bestand bevat onder andere de gegevens over de icon (het plaatje dat het bestand in de Workbench symboliseert). In de CLI hebt u deze bestanden niet nodig. Het programma 'Preferences' werkt probleemloos zonder het .info-bestand.

Alle bestanden die in de Workbench zijn afgebeeld met een archiefla, zijn in de directorylijst voorzien van het commentaar 'dir'. In de Workbench en de CLI maakt AmigaDOS gebruik van dezelfde opbouw van de administratie voor de boekingen in de inhoudsopgave. Ook in de CLI krijgt u niet in een keer alle bestanden te zien: in eerste instantie levert de instructie DIR alleen de hoofddirectory.

De directories zijn geordend in een boomstructuur met een stam (de hoofddirectory of root directory), takken (subdirectories, onderverdelingen) en bladeren (de bestanden). Er zijn theoretisch geen grenzen aan de hoeveelheid vertakkingen en het aantal vertakkningsniveaus (de nestingsdiepte).

In de Workbench komt u steeds op een volgend vertakkningsniveau door de archiefla te activeren met twee klikken met de muis. In de CLI moet u aan de instructie een pad toevoegen vanaf de hoofddirectory via een aantal directories naar de gewenste subdirectory. Bijvoorbeeld levert de opdracht 'dir system' de inhoudsopgave van de ordner 'System' op de Workbench-diskette. In dat resultaat staat geen boeking met het commentaar 'dir' en dus stelt elke naam een bestand voor: u bent aangekomen aan het einde van de reeks vertakkingen.

Probeer als tweede voorbeeld wat er schuil gaat achter de naam 'Devs' in de directory van de Workbench. Dit bestand heeft geen info-bestand en is in de Workbench dus niet herkenbaar

aan een archiefla. De opdracht 'dir devs' heeft als resultaat:

```
keymaps (dir)
printers (dir)
clipboards (dir)
clipboard.device      MountList
narrator.device       parallel.device
printer.device        serial.device
system-configuration
```

U ziet dat er nog drie vertakkingen zijn. Voor de weergave van de inhoud van een van die ordners verlengt u de omschrijving van het pad. Schrijf na een schuine streep de naam van de gewenste ordner:

```
dir devs/printers
```

Verwar de schuine streep niet met de backslash ('\') die onder de toets F10 zit. Het resultaat ziet er ongeveer als volgt uit:

```
Alphacom_Alphapro_101    Brother_HR-15XL
CBM_MPS1000              Diablo_630
Diablo_Advantage_D25    Diablo_C-150
Epson                    Epson_JX-80
generic                  HP_LaserJet
HP_LaserJet_PLUS        ImagewriterII
Okidata_292              Okidata_92
Okimate_20               Qume_LetterPro_20
```

Het programma 'Preferences' haalt uit deze directory de gegevens voor de aansturing van de gekozen printer. Er zijn verder geen directories meer. Dat klopt met de graad van nesting van de ordners in de Workbench. Als u de lijst met printers naar hun type verdeelt over twee directories, wordt het pad een schakel langer (bijvoorbeeld: dir devs/printers/matrix).

Een volledig pad vermeldt aan het begin het nummer van de diskdrive waarin de diskette zit. In de uitgangstoestand gebruikt de computer altijd drive df0:. U hoeft die dan niet te vermelden. Als u twee of meer drives gebruikt, typt u de bedoelde drive zonder spatie links voor het pad. Roep de hoofddirectory van de diskette in de tweede drive op met 'dir df1:'. Gebruik bij een harddisk de aanduiding 'dh0:'. Bijvoorbeeld:

```
dir dh0:tekst/brieven/rekeningen
```

In het algemeen is er een probleem, als u maar één diskdrive hebt. (Een oplossing geven we in hoofdstuk 3.) Voor het opvragen van een directory moet de CLI de benodigde routine laden. Die zal niet op een willekeurige diskette staan zodat er meteen een requester verschijnt die vraagt om de Workbench-diskette. De meeste CLI-instructies staan in de ordner 'C' van die diskette. Bekijk ze met de opdracht 'dir c'. Een paar instructies die meestal wel tot de CLI worden gerekend, maar naar hun interne structuur afwijken van de echte CLI-instructies, hebt u al gezien na de opdracht 'dir system'. Voor de voorbeelden in de volgende twee paragrafen hebt u maar één diskdrive nodig.

1.7 De Help-functie

In de routines van elke CLI-instructie is een Help-functie opgenomen. Om de syntaxis van een instructie op te zoeken kunt u de 'Quick Reference' in hoofdstuk 8 van dit boek erbij pakken. Maar meestal zult u voldoende informatie kunnen putten uit de formaatregel die de CLI-instructie zelf levert als u hem oproept, gevolgd door een spatie en een vraagteken:

```
dir ?  
dir,OPT/K:
```

De onderdelen van een instructie worden in de formaatregel gescheiden door komma's die u niet mag typen als u de opdracht gebruikt. Het voorbeeld bestaat uit 'dir' en 'OPT/K'. Een onderdeel van een instructie bevat soms, voorafgegaan door een schuine streep, een besturingsteken. In het voorbeeld is dat een van de drie mogelijkheden (K).

A (Argument)

Op de plaats van het onderdeel van de instructie typt u een gegeven. Zonder dat argument kan de opdracht niet worden uitgevoerd.

K (Key, trefwoord)

Typ de naam van het onderdeel van de instructie (OPT) plus een parameter. De namen van de toegestane parameters en het effect dat ze hebben verschillen per CLI-instructie (zie paragraaf 2.1.2 voor de opties van de instructie DIR).

S (Switch, omschakelfunctie)

Dit besturingsteken meldt dat de naam van het onderdeel van de instructie al voldoende is om een zeker effect te veroorzaken; een parameter is overbodig.

Als een onderdeel van een instructie geen van de drie besturingstekens bevat, blijkt de waarde van een parameter uit positie in de invoerregel. Bijvoorbeeld het onderdeel 'dir' van de instructie DIR:

```
dir df0:c
```

Sommige onderdelen van een instructie hebben geen naam: de plaats tussen de scheidingskomma's blijft onbenut. De formaatregel van de instructie DEL ziet er bijvoorbeeld als volgt uit:

```
,, , , , , , , , ALL/S, Q=QUIET/S:
```

Omdat er geen besturingstekens tussen de eerste negen komma's staan, is invoer van een waarde voor de eerste negen onderdelen niet verplicht. In dit voorbeeld ziet u een CLI-instructie die voor een onderdeel verschillende namen accepteert. De toegestane versies van de invoer zijn in de formaatregel vermeld, gekoppeld door een is gelijk-teken (Q = QUIET).

U bent vrij in de volgorde van de onderdelen van een instructie die een naam hebben. De instructie COPY omvat onder andere de onderdelen 'from' en 'to/A'. De twee onderstaande opdrachten zijn geheel gelijkwaardig:

```
copy FROM df0:brieven TO df1:teksten
```


copy TO df1:teksten FROM df0:brieven

(Binnen de CLI maakt het niet uit of u kleine letters of hoofdletters gebruikt.)

Als u de Help-functie gebruikt, verschijnt na uitvoer van een formaatregel niet de DOS-prompt. De cursor blijft staan achter de dubbele punt. Typ daar de waarde van de gevraagde parameter. Het is dus niet nodig de gehele opdracht opnieuw te typen. Gebruik in de CLI bij twijfel over parameters altijd eerst de instructie met een vraagteken.

1.8 De CLI verlaten

De instructie ENDCLI van de CLI heeft hetzelfde effect als activering van de afsluitings-gadget van de Workbench. In beide gevallen verdwijnt het venster.

2 De CLI-instructies

We bespreken de CLI-instructies niet in alfabetische volgorde maar rekening houdend met een toenemende moeilijkheidsgraad en een afnemend belang. Bovendien staan ze geordend naar een aantal gebieden van toepassing: diskdrives en bestanden (paragraaf 2.1), instructies die het systeem beïnvloeden (paragraaf 2.2), instructies voor batchbestanden (paragraaf 2.3) en ten slotte een paragraaf over de editors ED en EDIT. Dit zijn eigenlijk geen instructies maar programma's.

Voer in ieder geval de gegeven voorbeelden uit. Om te voorkomen dat uw resultaten afwijken van de uitwerking in dit boek, moet u gebruik maken van een complete kopie van de Workbench-diskette (paragraaf 1.3).

2.1 Instructies voor diskettes en bestandsbeheer

De meeste instructies in deze paragraaf bent u al onder een andere naam in de Workbench tegengekomen.

2.1.1 FORMAT

Voordat u een nieuwe diskette kunt gebruiken, moet die een voorbereidende bewerking ondergaan (formatteren). In de Workbench heet dat 'Initialize', in de CLI luidt de instructie **FORMAT**. De bewerking is noodzakelijk om later gegevens op een systematische manier op de diskette te kunnen schrijven en lezen. DOS merkt direct als een diskette niet is geformatteerd. In de Workbench heet zo'n diskette 'bad'.

Vermeld bij de instructie **FORMAT** in welke drive de diskette moet worden geformatteerd en welke naam ze moet krijgen. De volledige syntaxis luidt:

```
format DRIVE <disk> NAME <naam> [NOICONS]
```

Voor een diskette die de naam 'Spelen' krijgt na formattering in de standaarddrive:

```
format DRIVE df0: NAME Spelen [NOICONS]
```

De naam mag een lengte hebben van 30 tekens. Maak daar een spaarzaam gebruik van als u wilt voorkomen dat een lange naam in de Workbench een andere naam overlapt. Een naam mag geen spatie bevatten. Dat geldt trouwens overal in de CLI. Spaties zijn wel toegestaan als het hele onderdeel van de instructie tussen aanhalingstekens is geschreven:

```
dir "df0: Mijn verhalen"
```

De toevoeging **NOICONS** (geen icons) voorkomt dat de gegevens voor de prullebak die de Workbench nodig heeft, op de diskette komen te staan. Onder de CLI is dat overbodig.

De woorden **DRIVE** en **NAME** zijn noodzakelijk. Als u volstaat met de vermelding van de drive en de nieuwe naam, krijgt u meteen de juiste syntaxis voorgeschoteld. Na een correcte invoer krijgt u in het CLI-venster de boodschap:

Insert disk to be initialized in drive df0: and press Return

Alle gegevens voor het formatteren zijn geladen. Ook als u maar één diskette hebt, mag u nu de Workbench-diskette uit de drive halen en deze vervangen door de nieuwe diskette. Denk nog even goed na of de gegevens op een gebruikte diskette werkelijk mogen worden gewist. Druk bij twijfel tegelijk op de toetsen 'Ctrl' en 'C'. Dat onderbreekt altijd een onderhanden opdracht. De reactie in het CLI-venster is: '*** Break'. Kijk voor de zekerheid met de instructie DIR welke bestanden u op het punt stond te wissen.

Als alle in orde is en u hebt na de opdracht tot formatteren meteen op de Return-toets gedrukt, kunt in het CLI-venster volgen welke cilinder actief is. Elke cilinder bestaat uit twee concentrische cirkels met een breedte van ongeveer 0,5 mm aan weerszijden van de diskette. Elk spoor is onderverdeeld in 11 sectoren die elk 512 bytes kunnen bevatten. Een diskette bevat 80 cilinders (160 sporen). De opslagcapaciteit bedraagt derhalve 880 Kilobyte ($80 \cdot 2 \cdot 11 \cdot 512 / 1024$ byte).

De instructie FORMAT geldt alleen voor de 3,5 inch-diskettes die de Amiga gebruikt. Voor andere formaten (harddisk, 5,25 inch floppies) gelden andere instructies.

Als u een diskette volledig kopieert met de instructie DISKCOPY, hoeft u de doeldiskette niet vooraf te formatteren. Dat kan dus ook naar een nog niet eerder gebruikte diskette (DISKCOPY omvat FORMAT).

2.1.2 DIR

In paragraaf 1.6 hebt u al met behulp van de instructie DIR gezien volgens welke structuur AmigaDOS bestanden behandelt op een diskette of harddisk. De volledige syntaxis van de instructie 'dir' luidt:

```
dir DIR,OPT/K
```

Het onderdeel 'DIR' moet de volledige omschrijving bevatten van het pad naar de gewenste directory.

Vermeld eerst het aan te spreken apparaat (df1:, ram:, jh0:) als dat niet het actuele apparaat is. In de uitgangstoestand geldt df0:. DOS herkent een vermelding als aanduiding van een apparaat aan de aansluitende dubbele punt.

U wilt bijvoorbeeld de inhoudsopgave van 'Brieven' in de ordner 'Teksten' op de diskette in drive 1:

```
dir df1:Teksten/Brieven
```

Dieper geneste directories voegt u toe met een schuine streep (/) voor elke volgende laag. Als u geen directory opgeeft, geldt de actuele directory. Welke dat is, hangt af van wat u eerder definiëerde met de instructie CD (zie paragraaf 2.1.3).

'opt a' als uitbreiding van de opdracht, veroorzaakt iets nieuws. De uitvoer wordt veel uitgebreider (a is de afkorting van 'all': geheel, alles). Niet alleen de hoofddirectory maar ook alle subdirectories verschijnen op het scherm. Dat is erg handig als u vanwege een groot aantal onderverdelingen een zeker bestand niet kunt vinden. Als u deze optie gebruikt bij een harddisk, loopt het CLI-venster al vlug vol met gegevens en raakt u alsnog het spoor bijster. Druk snel op een willekeurige toets om de uitvoer te onderbreken. Zodra u het ingetoetste teken wist met

een Backspace, komt de uitvoer weer op gang.

'opt i' geeft u invloed op de uitvoer van de inhoudsopgave. Na elke naam van een bestand of een directory stopt de uitvoer en volgt een vraagteken. Het verdere verloop kiest u via het toetsenbord. Na een Return gaat de uitvoer (interactief) verder. Door invoer van DEL kunt u een van de vermelde bestanden wissen. Het is niet mogelijk een ordner te wissen als hij nog een of meer bestanden en/of subdirectories bevat. Als u dat desondanks probeert, verschijnt de boodschap 'Error code 216' en gaat de routine verder naar de volgende boeking.

Als de gegeven naam betrekking heeft op een ordner (te zien aan de toevoeging 'dir'), maakt u deze actueel door de invoer van 'E'. De uitvoer van de inhoudsopgave gaat verder met die directory. Op elk gewenst moment komt u na invoer van 'B' terug op het eerstvolgende hogere niveau van de inhoudsopgave. Als er geen hoger niveau meer is, wordt de opdracht DIR afgebroken.

Als de gegeven naam betrekking heeft op een tekstbestand, voert 'T' de inhoud van dat bestand uit in het CLI-venster (afkorting van Type). Deze optie heeft alleen zin bij echte ASCII-bestanden. Als u T gebruikt bij een programma of CLI-routine, loopt het venster vol met een ratjetoe van vreemde tekens. De toetsencombinatie Ctrl-C is de beste remedie tegen deze uitvoer. Als de uitvoer dan nog niet normaal is, brengt Ctrl-O de Amiga terug in de normale tekenset.

Als u een vraagteken invoert, levert de Amiga een overzicht van alle op dat moment geldige invoer voor de CLI. Een verkeerde keuze veroorzaakt de reactie 'Invalid response - try again:?' . U kunt het nog eens proberen.

Op elk gewenst moment kunt u de interactieve modus van DIR verlaten met 'Q' (Quit).

'opt ai' is de combinatie van 'opt a' met 'opt i'. Het spreekt voor zich dat u met deze optie alle bestanden interactief te zien krijgt.

2.1.3 CD

De instructie CD maakt een gegeven directory actueel (change directory, ga naar andere ordner). Tot nu toe was de hoofd-inhoudsopgave de actuele ordner. Dat blijft zo als u bij een instructie een volledig pad vermeldt: alle aanduidingen zijn gezien vanuit de hoofddirectory. Met CD verandert u het perspectief. Na die opdracht werkt u vanuit de vermelde directory.

De instructie CD zonder meer levert de actuele directory. Meteen na inschakeling van de computer zal dat de diskette zijn waarmee hij is opgestart, bijvoorbeeld: 'A500 WB1.2 N'. Geef dan de opdracht 'cd system' en daarna de opdracht 'cd'. De uitvoer zal dan zijn:

```
A500 WB1.2 N:System
```

Alle CLI-instructies zijn in het vervolg van toepassing op deze directory. DIR geeft dus niet meer de inhoud van de hoofddirectory. Laat goed tot u doordringen wat het verschil is tussen onderstaande twee manieren om de inhoud van de directory 'System' uit te voeren.

eerste mogelijkheid: `dir df0:System (Return)`

tweede mogelijkheid: `cd df0:System (Return)`
`dir (Return)`

Na de tweede variant hebt u geen toegang meer tot de hoofddirectory. Dat is niet voorgoed on-

mogelijk. Met de opdracht 'cd /' wordt de directory van één niveau hoger actueel gemaakt. Gebruik zonodig twee of meer schuine strepen om hogerop te komen. Met 'CD : ' komt u meteen terug in de hoofddirectory. U kunt ook de syntaxis 'CD df0:' gebruiken (afhankelijk van het aangesloten apparaat). Voor een beter begrip van het subtiele verschil tussen de laatste twee instructies moeten we iets dieper ingaan op de interne systeemprocedures.

AmigaDOS zoekt in het algemeen niet in een bepaalde drive naar een gegeven diskette, maar behandelt alle diskdrives gelijkwaardig. Als u in een pad de naam van een diskette hebt opgenomen, maakt het voor het systeem niet uit in welke drive die diskette zit. Als de diskette in geen enkele drive zit, spoort een requester u aan de gezochte diskette in een willekeurige drive te plaatsen.

Vaak is het noodzakelijk of handiger het systeem met name te kennen te geven welke drive moet worden aangesproken.

Neem aan dat u de hoofddirectory van de diskette met de naam 'A500 WB1.2 N' in de drive 0 wilt verklaren tot actuele directory. Dat is mogelijk met de opdracht: CD "A500 WB1.2 N:". De Amiga zoekt in alle drives en vindt die directory in df0:. Hetzelfde effect heeft de opdracht: CD df0:. Dat veronderstelt dat de gewenste inhoudsopgave (de gewenste diskette) werkelijk in drive 0 zit.

AmigaDOS slaat de naam van de diskette in een drive op en herkent in het vervolg deze diskette aan zijn naam (Workbench). Hij mag later in een andere drive worden gestopt. De opdracht 'cd :' geldt voor de actuele ordner, het maakt niet uit in welke drive. De opdracht 'cd df0:' maakt de hoofddirectory van drive 0 actueel, het maakt niet uit welke diskette daar in zit.

2.1.4 MAKEDIR

Het effect van de instructie MAKEDIR is vergelijkbaar met de procedure rond de ordner 'Empty' in de Workbench. U maakt een kopie van de lege ordner, geeft de nieuwe ordner een naam naar keuze en vult hem met toepasselijke bestanden. MAKEDIR maakt een nieuwe directory (ordner). De behoefte aan directories neemt sterk toe met de toename van de capaciteit van de opslagmedia. Een nieuwe directory voegt u toe één niveau lager dan een bestaande directory. Het pad naar de nieuwe directory wordt een stap langer dan een bestaand pad:

```
makedir df1:system/monitoren
```

Het is niet toegestaan een pad te verlengen met twee of meer stappen tegelijk.

2.1.5 DELETE

De instructie DELETE verwijdert een bestand of een directory van de vermelde diskdrive (ook een RAM-disk). Schrijf achter de instructie het volledige pad naar het object dat u wilt wissen:

```
delete df0:c/versie
```

Als 'versie' een directory is, kan die alleen worden gewist als hij geen bestanden of subdirectories meer bevat. Anders krijgt u foutmelding 216 (directory not empty). Met de toevoeging 'all' wist u de directory met inhoud en al. Vraag voor de zekerheid eerst de directory nog eens op.

Bij de omschrijving van het te wissen bestand zijn jokers toegestaan (* en ?, zie paragraaf

3.2). Als u tevreden bent met het resultaat van de ontwikkeling van een programma, kunt u alle testversies (test1 tot en met testx) in een keer wissen met de opdracht 'delete df0:test#?'. Denk ook hier nog eens na voordat u de Return-toets indrukt.

Behalve met jokers kunt u maximaal negen bestanden tegelijk wissen in één DELETE-opdracht. Scheid de paden met een spatie:

```
delete df0:utilities/notepad df1:system/say.info
```

U hoeft zich geen zorgen te maken of u wel negen paden op een regel kunt persen. Als het CLI-venster de maximale breedte heeft, springt de cursor drie regels naar beneden zodra u de grootst mogelijke invoer overschrijdt. Ontgrendel het toetsenbord daarna met tenminste één Backspace, want zelfs een Return wordt genegeerd.

Aan de instructie DELETE kunt u de optie 'Q' (of Quiet) toevoegen. Op die manier voorkomt u de vraag om een bevestiging van de opdracht als AmigaDOS zelf moet uitzoeken welke bestanden moeten worden gewist (bij toepassing van 'all' of van jokers):

```
delete df0:utilities all quiet
```

2.1.6 COPY

De instructie COPY is een van de belangrijkste en meest flexibele instructies voor bewerkingen met bestanden in de CLI. Hij kan bestanden of zelfs hele directories kopiëren naar een apparaat dat gegevens kan opnemen. Dat kan ook binnen een apparaat. De syntaxis van de instructie luidt:

```
FROM, TO/A, ALL/S, QUIET/S
```

Op de plaats van FROM schrijft u het pad naar het bron-object; deze vermelding is niet verplicht omdat zijn syntaxis geen '/A' bevat. Bij ontbreken van gegevens voor FROM geldt de actuele directory.

Op de plaats van TO schrijft u het pad naar het doel-object van het kopiëren. Wat u daar mag vermelden, hangt af van de aard van het bron-object.

FROM verwijst naar één bestand

Als u één bestand kopieert, mag het doelpad wijzen naar een willekeurige directory van het apparaat of naar een ander apparaat. Als het doelapparaat een diskdrive is, wordt het bestand in de uitgekozen directory opgeslagen onder dezelfde naam.

Voorbeeld 1:

```
copy df0:c/type ram:c
```

Het bestand 'type' wordt met dezelfde naam opgeslagen in de directory 'C' van de RAM-disk. Deze directory moet daar al bestaan. Maak hem zonodig eerst aan met de instructie MAKE-DIR.

Pas op: Als er in de doel-directory al een bestand bestaat met de naam 'type', wordt daar zonder meer overheen geschreven. Vertrouw niet op de waarschuwing 'File Exists' die sommige andere computers geven. Deze opmerking geldt voor alle conflicten tussen namen van bestanden.

Voorbeeld 2:

```
copy df0:s/startup-sequence prt:
```

Deze opdracht voert het tekstbestand 'startup-sequence' uit via de printer.

Als de kopie een andere naam krijgt dan het origineel, moet u die toevoegen aan het pad naar het doelobject. De bovengenoemde waarschuwing geldt ook hier. Als in het doelobject een subdirectory aanwezig is met een naam die gelijk is aan de beoogde nieuwe naam van het gekopieerde bestand, dan wordt de kopie daar opgeslagen onder zijn oude naam. Bij de invoer wordt immers geen onderscheid gemaakt tussen namen voor bestanden en directories. Voorbeeld:

```
copy df0:c/makedir ram:md
```

De routine voor de instructie MAKEDIR uit de directory C in drive 0 wordt met de naam 'md' gekopieerd naar de RAM-disk. Als daar al een directory bestaat met de naam 'md', krijgt de kopie dezelfde naam als het origineel (makedir).

FROM verwijst naar een subdirectory

Als het bron-object een subdirectory is, moet het doelobject ook een subdirectory zijn omdat de bewerking meer dan één bestand kopieert. In dit geval mag het doelobject geen printer zijn. De instructie COPY is niet geschikt om in een keer een aantal bestanden te laten uitvoeren via een printer.

Als het bron-object subdirectories bevat, worden die overgeslagen bij het kopiëren. Het is mogelijk die subdirectories met hun inhoud wel mee te kopiëren, als u de toevoeging 'all' gebruikt. In het doelobject worden dan automatisch de benodigde subdirectories aangemaakt.

```
copy df0: dh0:spelen all
```

Deze opdracht kopieert alle bestanden op de diskette in drive 0 naar de harddisk in een reeds bestaande directory met de naam 'spelen'.

Voor exacte kopieën van een diskette naar een andere diskette is de instructie DISKCOPY in de meeste gevallen een mogelijkheid die het voordeel heeft dat de doeldiskette meteen wordt geformatteerd. De instructie COPY heeft het voordeel dat een eventueel over de sectoren verspreid bronbestand netjes achter elkaar op de doeldiskette komt te staan. De schrijf- en leeskop van de diskdrive hoeft bij een zo ontstane kopie minder bewegingen te maken dan wanneer de sectoren met gegevens her en der verspreid liggen.

2.1.7 RENAME

De instructie RENAME verandert de naam van een bestand; de oude en de nieuwe naam zijn de verplichte argumenten (/A) van de instructie. De syntaxisregel luidt:

```
rename FROM/A,TO=AS/A
```

Vermeld bij zowel de oude als de nieuwe naam het volledige pad. In het eenvoudigste geval zijn die gelijk:

```
rename teksten/mijntekst teksten/opstel
```

RENAME is toepasbaar op bestanden en directories. Net als bij COPY vragen deze twee gevallen een iets verschillende behandeling.

FROM verwijst naar één bestand

In het eenvoudigste geval werkt RENAME op een enkelvoudig bestand. Bij de nieuwe naam mag u zowel het pad als de naam veranderen:

```
rename df0:system/format df0:c/formatteren
```

Het nieuwe pad moet leiden naar een al bestaande directory. Als de CLI de opgegeven directory niet kan vinden, krijgt u een foutmelding:

```
Can't rename df0:system/format as df0:c/formatteren
```

FROM verwijst naar een subdirectory

Als u alleen de naam van een subdirectory wilt veranderen, is de situatie gelijk aan die bij een enkelvoudig bestand:

```
rename df0:demos df0:demobestanden
```

Als u de naam van een pad verandert, verplaatst u een bestand of directory naar een andere plaats in de directory-structuur. Het volgende voorbeeld verplaatst de directory 'spelen' met alle spelprogramma's in zijn geheel naar uw algemene directory met BASIC-programma's:

```
rename df0:SPELEN to df0:BASIC/PROGRAMMAS/SPELEN
```

Merk op dat de verhuisde directory meteen een naam krijgt. Ook als u de oude naam wilt handhaven, moet u hem vermelden als nieuwe naam. Het onderdeel 'to' of 'as' kunt u ook weglaten.

Een fout in het besturingsysteem veroorzaakt problemen als u de instructie RENAME toepast op de RAM-disk. Het behoort onmogelijk te zijn dat tweemaal dezelfde naam in één directory voorkomt. De instructie MAKEDIR bijvoorbeeld levert een foutmelding: 'xxx already exists' (boeking xxx bestaat al). Maar zie wat er gebeurt als u de volgende regels invoert.


```
makedir ram:test
makedir ram:test/sub
rename ram:test/sub ram:test
dir ram:
```

In de hoofddirectory van de RAM-disk staat nu tweemaal een subdirectory met de naam 'test'. Het probleem is snel verholpen door een van de twee met RENAME een andere naam te geven. De boeking die het besturingssysteem het eerste tegenkomt, verandert van naam.

Met de instructie RENAME kunt u niet een directory verplaatsen van de ene drive naar een andere. Niet toegestaan is bijvoorbeeld:

```
rename df0:c/type ram:type
```

Dat is wel mogelijk met de instructie COPY.

2.1.8 LIST

De instructie LIST geeft een inhoudsopgave die meer informatie bevat dan het resultaat van DIR. Voer de instructie LIST in met het gewenste pad:

```
list df0:
```

Bij de Workbench-diskette ziet het resultaat er ongeveer als volgt uit:

```
Directory "df0:" on Sunday 18-Oct-87
Expansion.info      346 rwed 02-Mar-87 23:29:10
Trashcan           Dir rwed 02-Mar-87 23:29:10
.info              82  rwed 09-Jun-87 12:43:46
c                  Dir rwed 02-Apr-87 09:44:21
Clock.info         338 rwed 02-Mar-87 23:31:49
Demos              Dir rwed 15-Apr-87 07:57:18
Clock              19668 rwed 02-Mar-87 23:32:20
System             Dir rwed 06-Apr-87 15:11:06
l                  Dir rwed 02-Mar-87 23:33:26
devs               Dir rwed 02-Mar-87 23:35:49
s                  Dir rwed 16-Apr-87 08:43:34
t                  Dir rwed 06-Apr-87 15:08:45
Preferences.info   418 rwed 02-Mar-87 23:36:00
Preferences        58136 rwed 02-Mar-87 23:36:20
Demos.info         346 rwed 02-Mar-87 23:36:22
fonts              Dir rwed 02-Mar-87 23:37:32
libs               Dir rwed 02-Mar-87 23:37:59
Empty              Dir rwed 09-Jun-87 12:46:57
Utilities.info     346 rwed 02-Mar-87 23:38:08
Disk.info          306 rwed 02-Apr-87 09:05:36
System.info        346 rwed 02-Mar-87 23:38:13
Empty.info         346 rwed 02-Mar-87 23:38:16
```

```

Trashcan.info          430 rwed 02-Mar-87 23:38:17
Utilities              Dir rwed 15-Apr-87 07:57:45
Expansion              Dir rwed 15-Apr-87 07:35:10
12 files - 13 directorys - 196 blocks used

```

Blijkbaar hebt u de vermelde directory opgevraagd op uw vrije zondag, aannemende dat u met 'Preferences' de juiste datum hebt ingesteld of dat u een Amiga hebt met doorlopende klok met batterijvoeding. In tegenstelling tot de instructie DIR krijgt u een ongesorteerde lijst met bestanden en directories (aangeduid met 'Dir') in de hoofddirectory. Achter een enkelvoudig bestand staat zijn grootte uitgedrukt in bytes.

De vermelde datum en tijd zijn die van de laatste handeling met het bestand.

De letters in de combinatie 'rwed' melden de status van een bestand:

r	(READ)	lezen toegestaan
w	(WRITE)	beschrijven toegestaan
e	(EXECUTE)	programma mag worden uitgevoerd
d	(DELETE)	boeking mag worden gewist

In het voorbeeld is bij alle bestanden alles toegestaan. Als een letter ontbreekt, is de bijbehorende activiteit niet toegestaan. Een bestand is beschermd tegen wissen als er 'rwe' achter staat.

Met de instructie PROTECT (paragraaf 2.1.17) kunt u de status van een bestand veranderen.

Let op: De flags r, w en e zijn op dit moment nog niet geïmplementeerd. AmigaDOS trekt zich niets aan van wat daar staat. Deze activiteiten zijn dus steeds mogelijk.

Ten slotte vermeldt de lijst het aantal bestanden, het aantal directories en het aantal daardoor in beslag genomen blokken (2 blokken = 1 Kbyte).

De syntaxisregel van LIST laat zien wat deze instructie nog meer kan. Roep die regel op met 'LIST?'.

```
DIR, P=PAT/K, KEYS/S, DATES/S, NODATES/S, TO/K, S/K, SINCE/K, UPTO/K, QUICK/S
```

Meestal zult u geen of maar een paar van al deze opties nodig hebben.

<i>DIR</i>	Geef een pad op als u niet de actuele directory wilt zien (bijvoorbeeld: list ram:c).
<i>PAT</i>	(pattern, patroon) Vermeld hier naar welke boekingen moet worden gezocht. Jokers komen daarbij goed van pas (list pat a#? levert een lijst van alle bestanden waarvan de naam begint met een 'a').
<i>KEYS</i>	Deze optie levert bij elke boeking het nummer van het startblok.
<i>DATES</i>	Deze optie is bedoeld om de weergave van de datum in te schakelen. Dit onderdeel van de syntaxis kunt u net zo goed weglaten, want de datum wordt altijd vermeld. (Niet geïmplementeerd.)
<i>NODATES</i>	In tegenstelling tot DATES is deze optie wel geïmplementeerd. Ze onderdrukt de weergave van de data.
<i>TO</i>	Vermeld hier een bestand of een device als u de uitvoer ergens anders heen wilt sturen (bijvoorbeeld 'prt:' voor de printer).

- S** (subname, onderdeel van een naam) Definieert een zoekpatroon dat geldt vanaf een gegeven positie in de namen. In paragraaf 3.2 leggen we uit dat de vermelding '#?subname#?' samen met de optie hetzelfde effect heeft.
- SINCE** Met deze optie levert LIST een lijst van alle boekingen na de opgegeven datum. Voor die datum geldt het Engelse formaat: DD-MMM-YY (schrijf voor maart 'mar' in plaats van 'mrt').
- UPTO** Analooq aan SINCE. De opgegeven datum is de bovengrens van de gewenste termijn.
U kunt ook een combinatie van SINCE en UPTO gebruiken. Bovendien kent de Amiga de begrippen yesterday en today:

```
list since 09-jun-87 upto yesterday
```

- QUICK** De lijst bestaat alleen nog uit de namen van de boekingen. Het enige voordeel in vergelijking met de instructie DIR is dat u de door de opgegeven boekingen gebruikte ruimte te zien krijgt.

Een voorbeeld:

```
list df1:klantenbestand #?man nodates to prt: since 10-sep-86 upto today
```

Van de subdirectory 'klantenbestand' van de diskette in drive 1 worden via de printer alleen de namen van die bestanden uitgevoerd die eindigen op 'man'. Daarbij geldt bovendien de beperking dat ze niet voor 10 september 1986 mogen zijn opgeslagen.

2.1.9 DISKCOPY

De instructie DISKCOPY in de CLI komt overeen met het menupunt 'Duplicate' in de Workbench. In tegenstelling tot de CLI-instructie COPY zijn de bestanden op de kopie even ordelijk of rommelig op de diskette geschreven als bij het origineel. Op een paar uitzonderingen na staat alles op dezelfde plaats. Het is ook niet nodig de diskette vooraf te formatteren met FORMAT. Als u een gebruikte diskette neemt om de kopie op te maken, bent u daarvan onherroepelijk alle gegevens kwijt.

De syntaxis van de instructie luidt:

```
diskcopy [FROM] <drive> TO <disk> [NAME <naam>]
```

Als u met één diskdrive een duplicaat van een diskette wilt maken luidt de opdracht:

```
diskcopy df0: to df0:
```

(TO is verplicht, FROM mag u weglaten.)

Als u ervan overtuigd bent dat u de gegevens op de doeldiskette niet meer nodig hebt, drukt u na het inleggen van de doeldiskette op de Return-toets. Gebruik bij twijfel de toetsencombinatie Ctrl-C. Bij de volgende Return wordt de routine van DISKCOPY verlaten en krijgt u de de volgende boodschap:

***** BREAK**
Disk Copy Abandoned.
Remember to insert original disk

Disk Copy Terminated

Met de doeldiskette is nog niets gebeurd. De CLI is zo vriendelijk u eraan te herinneren dat u het origineel terug moet zetten om verder te kunnen. De boodschap 'Disk Copy Terminated' is in dit geval eigenlijk overbodig, omdat het geen zin heeft te melden dat een karwei klaar is als er niet aan is begonnen.

Evenals in de Workbench krijgt u in extreme situaties een waarschuwende melding over het aantal malen dat u de diskette zult moeten wisselen. Het is niet te hopen dat u de volgende regel vaak zult zien:

The Disk Copy will take 40 swaps.

Als u een duplicaat maakt van een Workbench-diskette zonder tekstverwerker en het tekenprogramma voor 4096 kleuren, blijft het aantal wisselingen binnen redelijke grenzen. Een Amiga 500 in de standaarduitvoering (1/2 Mbyte) maakt zo'n kopie met driemaal wisselen.

We raden u aan een extra diskdrive boven aan uw boodschappenlijstje te zetten. Dat voorkomt niet alleen bij de instructie DISKCOPY zenuwslopend wachten.

De kopie verschilt op één of twee punten van het origineel. In de eerste plaats is het tijdstip van aanmaken anders. Datum en tijd krijgen op de kopie altijd de actuele waarden. In de tweede plaats kunt u zelf een verschil aanbrengen door de kopie een andere naam te geven. Het laatste stuk van de syntaxis biedt daartoe de gelegenheid: [name < naam >]. Bijvoorbeeld:

```
diskcopy drive df0: to drive df1: name "Workbench 1.2"
```

2.1.10 RELABEL

De instructie RELABEL geeft een nieuwe naam aan een diskette die al een naam heeft. De volgende opdracht geeft een diskette die in drive 1 zit de naam 'spelen':

```
relabel df1: spelen
```

Vergeet niet een spatie te zetten achter de aanduiding van de diskdrive. Als de naam zelf spaties bevat, schrijft u hem in zijn geheel tussen dubbele aanhalingstekens: "A500 WB1.2". De naam mag 30 tekens lang zijn, maar op de Workbench is een lange naam hinderlijk.

2.1.11 INFO

De instructie INFO heeft geen parameters. Op grond van zijn overeenkomst met andere systeemfuncties hoort deze instructie eigenlijk thuis in paragraaf 2.2. Omdat de instructie betrekking heeft op diskdrives, vindt u de beschrijving hier.

Het resultaat is bijvoorbeeld:

Mounted disks:

Unit	Size	Used	Free	Full	Errs	Status	Name
DF0:	880K	1645	113	93%	0	Read	A500 WB1.2
DF1:	880K	534	1224	30%	0	Read/Write	Textomat

Volumes available:

A500 WB1.2 [Mounted]

Textomat [Mounted]

In het eerste gedeelte vindt u gegevens over de aangesloten (mounted) diskdrives. Hun capaciteit hebben we al eens eerder berekend (880 Kbyte).

Onder 'Used' en 'Free' staan de aantallen gebruikte en vrije blokken (2 blokken = 1 Kbyte). Bij een percentage voor de bezettingsgraad kunt u zich misschien meer voorstellen. Op de diskettes van het voorbeeld zijn blijkbaar geen blokken met fouten aanwezig. Dat gegeven is vooral van belang voor bezitters van een harddisk. Als het aantal blokken met fouten toeneemt, wordt het hoog tijd te gaan sparen voor een nieuwe harddisk.

De 'Status' van de drive meldt of er op de diskette in die drive mag worden geschreven of niet. De diskette in drive 0 mag dus alleen worden gelezen.

In de laatste kolom staat de naam van de diskette die in de drive zit.

Onder de kop 'Volumes available' staat geschreven welke diskettes u kunt gebruiken zonder wisselingen.

2.1.12 INSTALL

Bij een koude start is de Amiga heel kieskeurig bij de opstartdiskettes. Normaal gesproken reageert de Amiga alleen op een Workbench-diskette met een laadprocedure. De instructie **INSTALL** maakt een diskette zelfladend (bootable). De syntaxis van de instructie luidt:

```
install [drive] {df0:|df1:|df3:}
```

De opdracht 'install df0:' maakt van de diskette in drive 0 een systeemdiskette. De diskette moet vooraf zijn geformatteerd. (Het is nog niet mogelijk het systeem op te starten vanaf een harddisk.) Als u de computer opstart met een systeemdiskette, komt het besturingssysteem al snel in de CLI. Allerlei zaken die bij opstarten met een Workbench-diskette automatisch gebeuren, blijven nu achterwege.

Ten eerste is de schermmodus is niet aangepast aan een monitor met tachtig tekens per regel. Activeer daartoe in de Workbench een van de twee velden in het 'Preferences'-venster. Het programma slaat de gekozen modus op in het bestand 'System-configuration' in de directory 'Devs'. Maak met **MAKEDIR** ook zo'n directory op de nieuwe diskette en kopieer dat bestand naar die diskette. Na een Reset met de nieuwe diskette werkt het scherm in de modus met 80 tekens.

Nederlanders zijn dan nog relatief goed af, want voor de meeste talen moet ook nog een afwijkende tekenset worden geïnstalleerd. De directory 'Keymaps' bevat voor diverse talen een aangepaste tekenset. Het programma 'SetMap' in de directory 'System' brengt het systeem in een modus die past bij het gebruikte toetsenbord. De opdracht 'system/setmap ...' regelt dat.

Dergelijke opdrachten kunt u met een editor of eenvoudige tekstverwerker opslaan in een bestand met de naam 'startup-sequence' (bijvoorbeeld met **ED** in de directory **C**). Als dat be-

stand in een directory met de naam 'S' staat, zal AmigaDOS die reeks opdrachten automatisch afhandelen na een Reset. Een bestand met CLI-opdrachten heet een batchbestand (zie hoofdstuk 4). Voor een goed werkende minimumconfiguratie hebt u nog een paar routines van CLI-instructies nodig. Kopieer die ook naar de directory C op de nieuwe diskette, omdat AmigaDOS alleen zoekt naar een directory met die naam.

2.1.13 TYPE

Bij de interactieve weergave van een directory brengt de CLI een zuiver tekstbestand op het scherm (dir opt i). De instructie TYPE doet dat ook. De volgende opdracht brengt het bestand 'mountlist' in de directory 'Devs' op het scherm:

```
type df0:devs/mountlist
```

De uitvoer van de tekst stopt niet als het scherm vol is. Op dat moment is het indrukken van een willekeurige toets voldoende om de uitvoer te onderbreken. Elke andere toets brengt de uitvoer weer op gang, maar alleen de Backspace-toets heeft het voordeel dat na afloop weer de gewone DOS-prompt verschijnt ('1>').

Met een uitbreiding van de instructie kunt u de uitvoer omleiden naar een ander apparaat ('prt:') of een bestand:

```
type df0:devs/mountlist t/mijntekst
```

Het bestand 'mijntekst' bevat na afloop datgene wat anders op het scherm zou zijn verschenen. Deze opdracht is equivalent met:

```
type >t/mijntekst df0:devs/mountlist
```

Het teken '>' biedt altijd de mogelijkheid de uitvoer van een CLI-opdracht om te leiden.

Met de optie 'opt n' kunt u de regels van de uitvoer voorzien van een doorlopende nummering. Dat is bijvoorbeeld handig bij een als ASCII-bestand opgeslagen BASIC-programma. Met de regelnummers is het voor iemand anders makkelijker het programma over te typen.

De optie 'opt h' is van nut voor krakers. De instructie TYPE geeft alleen een correcte uitvoer als de tekst geen besturingstekens bevat. Dat is bijvoorbeeld bij de routine van een instructie van AmigaDOS wel het geval. De opdracht 'type c/type' veroorzaakt zo op het oog alleen een hoop rommel op het scherm. Toevoeging van 'opt h' aan deze opdracht levert een ordelijke tabel. Het onderstaande geeft een gedeelte van het CLI-venster weer:

```
0000: 000003F3 00000000 00000002 00000000 .....
0010: 00000001 0000004F 000001C4 000003E9 .....0.....
0020: 0000004F 286A0164 700C4E95 2401223C ...0(j.dp.N.$."<
0030: 00000095 49FAFFEE 286CFFFC 2F0C2F02 ....I...(1.././.
```

Helemaal rechts staan de leesbare tekens van de routine. Elke punt staat op de plaats van een niet-afdrukbaar teken (afgezien van een paar echte punten).

In de eerste kolom worden de tekens hexadecimaal genummerd en in het midden staat de inhoud van het bestand, steeds in regels van vier longwords. Elk longword bestaat op zijn beurt

uit vier bytes. Een regel bevat dus zestien bytes. Zie ook de nummering van de bytes aan de linkerkant en het aantal tekens rechts.

De letter 'T' in de laatste regel staat op de 52^e positie ($3 \cdot 16 + 4$). De ASCII-code van de letter 'T' is \$49, dat is decimaal 73 ($4 \cdot 16 + 9$). (Het teken '\$' kenmerkt het erachter staande getal als hexadecimaal getal.)

2.1.14 JOIN

De instructie JOIN koppelt maximaal vijftien bestanden tot een nieuw bestand. De syntaxis van de instructie is:

```
join,,,,,,,,,,,,,AS/A/K
```

De vijftien komma's reserveren ruimte voor de bronbestanden. Het woord AS is verplicht (vanwege 'K'). Vermeld ('/A') achter AS het pad naar het doelbestand. Op een eenvoudige manier simuleert u met JOIN de basisfunctie van de instructie TYPE. Het teken '*' leidt de weergave van de tekst van de startup-sequence om naar het scherm:

```
join df0:s/startup-sequence as *
```

In de reeks bronbestanden mag meer dan één keer hetzelfde bestand voorkomen. De koppelmogelijkheid van JOIN is handig als u een aantal bestanden achter elkaar wilt afdrukken met een printer. De instructie COPY accepteert jokers, maar dan krijgt u alle bestanden die aan de omschrijving voldoen. Met JOIN kunt u expliciet opgeven welke bestanden van de reeks tekstX u wel en welke u niet wilt afdrukken:

```
join tekst1 tekst3 tekst4 tekst10 as prt:
```

Als u programmeert in een compiler-taal, gebeurt het wel eens dat het werkgeheugen van de teksteditor niet groot genoeg is om het programma in zijn geheel te bevatten. Maak dan het programma in modulen en koppel die met de instructie JOIN tot één bestand voor de compiler.

2.1.15 SEARCH

Met de instructie SEARCH doorzoekt u bestanden op een gegeven reeks tekens. Bij aantreffen van die tekens bestaat de uitvoer uit de naam van het bestand gevolgd door het nummer en de tekst van de regel die die tekens bevat. De syntaxis van de instructie luidt:

```
search FROM,SEARCH/A,ALL/S
```

Schrijf op de plaats van FROM het volledige pad naar een directory of een bestand. Bij afwezigheid van deze vermelding geldt de actuele directory. Er staat dan tweemaal 'search' achter elkaar:

```
search search "eerste helft"
```

De aanhalingstekens zijn verplicht als de zoekstring spaties bevat. De routine maakt bij wijze van uitzondering onderscheid tussen hoofdletters en kleine letters. Met de optie ALL doorzoekt de routine ook de subdirectories van de vermelde directory.

De instructie SEARCH accepteert jokers:

```
search ???/#?t hallo
```

Deze opdracht zoekt naar het woord 'hallo' in alle bestanden waarvan de naam eindigt op een 't' en die staan in een directory met een naam van drie willekeurige letters.

Net als bij de andere CLI-instructies breekt de toetsencombinatie Ctrl-C de zoekprocedure af. Als de opdracht betrekking heeft op een hele directory, zoekt de routine na de toetsencombinatie Ctrl-D direct verder in het volgende bestand.

Als een bestand niet alleen tekst bevat, verschijnt de melding 'Line x truncated' (Regel x afgekapt). Evenals bij de instructie TYPE ziet de uitvoer er dan soms erg vreemd uit.

De instructie SEARCH is bijvoorbeeld handig om in een programma in de taal C uit te vinden in welk include-bestand een gegeven structuur voorkomt.

2.1.16 SORT

De instructie SORT, die zuivere tekstbestanden sorteert, vindt bij lange na niet zoveel toepassing als ze verdient. De syntaxis luidt:

```
sort FROM/A, TO/A, COLSTART/K
```

- FROM** Definieert het pad naar het te sorteren bestand (geen directory!). Hier moet een geldige naam staan ('/A').
- TO** Hier definieert u een pad naar een doelbestand of de naam van een device dat het gesorteerde bestand opneemt, want het bronbestand blijft ongewijzigd. Schrijf een sterretje ("*") als u het scherm kiest of 'prt:' voor de printer.
- COLSTART** Vermeld hier vanaf welke kolom (column) de regels moeten worden gesorteerd. Neem aan dat het bestand een lijst is met namen waarin tien posities zijn gereserveerd voor de voornamen. Als u voor COLSTART de waarde 10 invoert, wordt het bestand gesorteerd volgens de kolommen met een hoger nummer (achternamen). Als u hier niets vermeldt, sorteert de routine vanaf het begin van de regels.

2.1.17 PROTECT

Met de instructie PROTECT zet u de statusflags (rwd) van een bestand (zie paragraaf 2.1.8). Hoewel de instructie PROTECT alle vier de flags kan zetten, houdt AmigaDOS alleen rekening met de flag 'd' (delete, wissen). Deze flag is als het ware een softwarematig beschermings-schuijfe tegen beschrijving voor elk bestand afzonderlijk.

U kunt ook een directory op deze manier beschermen, maar dat is overbodig:

```
protect dfl:brieven
```


Zolang de directory niet helemaal leeg is, is het onmogelijk hem te wissen (Error 216). De bescherming werkt niet als u met de opdracht 'delete ... all' de inhoud van die directory wilt wissen. De instructie verhindert in dit voorbeeld alleen maar dat een lege directory per ongeluk wordt gewist. De instructie heeft wel zin voor de afzonderlijke bestanden in de directory:

```
protect df1:brieven/uitnodiging
```

Het statuswoord bestaat nu uit vier mintekens: '----'. Dat betekent dat het bestand in theorie helemaal niet meer kan worden gebruikt, maar in werkelijkheid wordt alleen een poging tot wissen verijdeld (foutcode 222). Verwijder de bescherming met de opdracht:

```
protect df1:brieven/uitnodiging rwed
```

Het bestand heeft zijn oude status terug en kan weer worden gewist.

2.1.18 FILENOTE

Met FILENOTE kunt u een korte notitie toevoegen aan een bestand (een versienummer of een toelichting op de aard van het bestand). Die wordt niet opgeslagen in het bestand, maar gekoppeld aan de vermelding in de directory. De notitie mag maximaal 80 tekens (één regel) lang zijn. Na de instructie LIST verschijnen ook de regels met notities op het scherm. Ze worden ter onderscheiding van de overige uitvoer aangeduid door een dubbele punt aan het begin van de regel.

Neem bijvoorbeeld aan dat u steeds vergeet wat de bedoeling is van de routine FILENOTE. Voeg daarom een geheugensteuntje toe:

```
filenote c/filenote "N.B. Hiermee kan ik een commentaar van 80 tekens bij  
elk bestand schrijven."
```

De aanhalingstekens zijn verplicht als de tekst spaties bevat. Het resultaat van de opdracht LIST op het bestand 'c/filenote' is:

```
c/filenote                700 rwed 02-Mar-87 23:30:19  
: N.B. Hiermee kan ik een commentaar van 80 tekens bij elk bestand  
  schrijven."
```

Als u een bestand kopieert, gebeurt dat zonder het bijbehorende commentaar. Een eventueel reeds bestaand doelbestand behoudt ook zijn oude commentaar.

2.1.19 SETDATE

Als u een Amiga hebt zonder een klok met batterijvoeding, moet u de datum en de tijd iedere keer instellen met het programma 'Preferences'. Vergeet u dat of vindt u dat lastig, dan kunt u ook achteraf de tijd en datum van aanmaak van een bestand wijzigen met de instructie SETDATE. De syntaxis luidt:

`setdate FILE/A,DATE/A,TIME`

Het spreekt vanzelf dat invoer van een bestandsnaam en het pad daarheen verplicht is ('FILE/A'). Vermeld de datum volgens hetzelfde formaat als de instructie LIST vereist (DD-
MMM-YY). Als de oude datum maximaal een week verschilt van de actuele datum, mag u de dag van de week schrijven (in het Engels):

`setdate teksten/brief saturday`

De routine rekent zelf de datum uit; hetzelfde geldt voor 'yesterday'. Dat woord komt ook te voorschijn als u een inhoudsopgave opvraagt met LIST. Als u een datum opgeeft die later valt dan de actuele datum in 'Preferences', herkent AmigaDOS de tegenstrijdigheid en vermeldt de inhoudsopgave 'Future'. Dat blijft zo totdat de werkelijke datum de opgeslagen datum heeft ingehaald. Vanaf dat moment ziet u de genoteerde datum.

Als u geen tijd opgeeft, schrijft de routine automatisch 0 uur bij het bestand. Data voor 2 januari 1978 accepteert de instructie niet. In de inhoudsopgave staan dan twee lege velden voor de datum en de tijd.

2.1.20 DISKDOCTOR

Het besturingssysteem geeft een boodschap via een requester als er iets mis is met de 'Disk Structure' van de actuele diskette. U wordt meteen doorverwezen naar het programma 'Disk-Doctor' waarmee u kunt proberen orde te scheppen in de uit zijn verband gerukte verzameling sectoren van allerlei bestanden. Roep het programma op met:

`diskdoctor drive df0:`

Het programma vraagt u de beschadigde diskette in de vermelde drive te stoppen en Return te geven. In de daarop volgende minuten mag u er het beste van hopen. Op het scherm kunt u volgen in welk spoor het programma aan het werk is en welke 'Hard Error Tracks' het daarbij ontdekt (onherstelbare sporen). In de tweede ronde krijgt u een inhoudsopgave van de bestanden die achtereenvolgens op de diskette staan. Het is best mogelijk dat daar ook (gedeelten van) bestanden tussen staan die u al lang geleden hebt gewist. Met de DiskDoctor kunt u gewiste bestanden weer boven water halen, mits er later niet voor een ander bestand overheen is geschreven.

Als de DiskDoctor een fout vindt, meldt hij welke 'Keys' (blokken) onleesbaar zijn en bij welke bestanden die horen. U krijgt dan de gelegenheid zo'n bestand te wissen (operatie geslaagd, patiënt overleden). Ten slotte krijgt u altijd het advies de geredde bestanden te kopiëren naar een andere diskette en het origineel te formatteren. Jammer genoeg gebeurt dat ook als de onderzochte diskette helemaal in orde is.

Als u na bovengenoemde boodschap de diskette te vroeg uit de drive haalt, kunt u meteen weer de dokter roepen. Er is een grote kans dat er tijdens de lange uitlooperperiode van de diskette ergens iets mis gaat.

2.1.21 DISKCHANGE

De instructie DISKCHANGE is alleen van belang voor bezitters van een diskdrive voor 5,25"-diskettes. Zulke diskdrives melden niet automatisch aan AmigaDOS dat er een diskettewisseling heeft plaatsgevonden. U moet dat steeds zelf met de opdracht 'diskchange *device-naam*' aan AmigaDOS doorgeven. *Device-naam* is de naam waarmee u die diskdrive aanspreekt. Pas na die melding kunt u de nieuwe diskette gebruiken.

2.2 Systeeminstructies in de CLI

In de volgende paragrafen komen de instructies voor het systeem aan de orde. In deze groep bespreken we ook de instructies voor de CLI zelf.

2.2.1 NEWCLI

Met de instructie NEWCLI komen de mogelijkheden van de Amiga tot multitasking binnen uw bereik. Multitasking wil zeggen dat de computer met een aantal taken praktisch tegelijk bezig is. In feite wikkelt de processor in een razendsnelle afwisseling zeer kleine stukjes van de onderhanden opdrachten af.

Met de instructie NEWCLI voegt u een opdracht toe aan die reeks taken. Op het scherm verschijnt een nieuw venster met daarin de mededeling bij welke taak het hoort (bijvoorbeeld: NewCLI task 2). Ga rustig door met de opening van nog meer vensters.

Zelf kunt u maar in één venster tegelijk werken, maar de Amiga gaat ondertussen verder met een eerder in een ander venster gegeven opdracht. Neem aan dat u in het oorspronkelijke CLI-venster de volgende opdracht hebt gegeven: 'format drive df0: name leeg'. Activeer dan met de muis een ander CLI-venster en vraag daar bijvoorbeeld de inhoudsopgave van de diskette in de andere diskdrive. Het spreekt vanzelf dat de tijd die de computer nodig heeft voor de afzonderlijke opdrachten, op deze manier merkbaar toeneemt. Het rendement van de computer wordt echter groter, omdat hij alle wachtperiodes opvult met andere taken (zie verder hoofdstuk 5).

De instructie NEWCLI heeft de afmetingen van het nieuwe venster als parameters:

```
newcli con:50/70/250/150/Amiga
```

Deze opdracht produceert een CLI-venster met de naam 'Amiga', dat 250 pixels breed is en 150 hoog en waarvan de linker bovenhoek zich bevindt in het punt (50,70). Zo'n uitgebreide invoer is niet de handigste manier om een CLI-venster vorm en plaats te geven. Dat gaat gemakkelijker met de muis: een CLI-venster heeft alle gadgets (behalve die voor afsluiten). Een opdracht als in het voorbeeld is wel zinvol in de startup-sequence.

Bij afwezigheid van argumenten voor de grootte beslaat een nieuw CLI-venster de hele breedte van het scherm en bijna de halve hoogte.

De instructie NEWCLI is uitbreidbaar met 'FROM' en de naam van een batchbestand. De opdrachten in dat bestand worden automatisch afgewikkeld bij opening van dat venster. Het batchbestand moet staan in de directory die behoort bij het virtuele device 'S:'. (Vraag de naam van die directory op met de instructie ASSIGN.) Een voorbeeld:

newcli from maak_kopie

U kunt pas zelf aan het werk in het nieuwe venster als het batchbestand 'maak_kopie' is uitgevoerd. Gewoonlijk staat het bestand in de directory 'S' van de standaarddrive van het systeem. Voor meer informatie over batchbestanden verwijzen we naar hoofdstuk 4.

2.2.2 ENDCLI

De instructie ENDCLI sluit een venster af dat werd geopend vanuit de Workbench of in een ander CLI-venster met NEWCLI. De instructie ENDCLI heeft alleen invloed op het actuele venster. Het is niet mogelijk vanuit een venster een ander venster te sluiten. Als u een venster sluit waarin nog een taak loopt die u met de instructie RUN hebt gestart, is er via dat venster alleen nog eenrichtingsverkeer mogelijk. U kunt er niet meer in werken, maar het venster blijft bestaan voor uitvoer van de onderhanden taak. Zodra die is voltooid, verdwijnt het. (De instructie RUN bespreken we in de volgende paragraaf.)

Pas op met het allerlaatste venster. Als u daar de instructie ENDCLI geeft, zaagt u de poten onder uw eigen stoel weg. U kunt AmigaDOS dan niet meer instrueren een ander programma te starten.

2.2.3 RUN

Behalve multitasking met behulp van meer dan één CLI-venster is dat ook mogelijk binnen één CLI-venster. Elk programma, dus ook CLI-instructies, kan draaien als een *achtergrondtaak*. Een programma wordt uitgevoerd in die modus als u het in een CLI-venster start met de instructie RUN. Onderwijl kunt u in dat venster iets anders doen. Uitvoer van de RUN-taak verschijnt in het CLI-venster, tenzij u dat anders regelt. Voorbeeld:

```
run c/join brief1 brief2 brief3 to prt:  
dir ram:
```

Terwijl de printer op de achtergrond bezig is de opgegeven brieven af te drukken, kunt u de inhoud van de RAM-disk bekijken.

Dit resultaat is ook mogelijk zonder de instructie JOIN. Daarvoor hoeft u niet viermaal de volledige opdracht 'run type briefx to prt:' te herhalen. Het teken '+' koppelt de bestanden:

```
run type Brief1 to prt: +  
type Brief2 to prt: +  
type Brief3 to prt:
```

De hele reeks opdrachten komt op gang als achtergrondtaak zodra u na de laatste regel een Return geeft. Merk op dat de laatste regel geen '+' bevat.

2.2.4 STATUS

Op elk moment levert de instructie STATUS een overzicht van de lopende processen (taken, tasks) in de CLI. Als u de opdracht geeft zonder parameters of met vermelding van 'CLI' of 'all', krijgt u een lijst met namen van de actuele taken. Voorbeeld:

```
Task 1: Loaded as command: status
Task 2: Loaded as command: textomat
```

De CLI zelf is natuurlijk bezig met de instructie STATUS. Het tekstverwerkingsprogramma Textomat is gestart vanuit de CLI.

U mag de gevraagde informatie ook beperken tot één taak. Vermeld dan het nummer van de taak achter de instructie: status 2. De instructie levert met de optie 'tcb' nog meer informatie over de taken. Toegepast op het hierboven gegeven voorbeeld, is het resultaat:

```
Task 1: stk 1600, gv 150, pri 0
Task 2: stk 3200, gv 150, pri 0
```

De gegevens achter een taak betekenen het volgende:

stk De grootte van de processor-stack voor deze taak.
gv Lengte van de 'Global Vector Table'.
pri De prioriteit van de taak. Deze waarde mag liggen tussen -128 en +127.

De laatste twee gegevens komen uitgebreider aan de orde in respectievelijk de paragrafen 5.6 en 6.3.

Met de optie 'full' levert de instructie STATUS alle gegevens in een keer. Voor het eerder gebruikte voorbeeld is het resultaat:

```
Task 1: stk 1600, gv 150, pri 0 Loaded as command: status
Task 2: stk 3200, gv 150, pri 0 Loaded as command: textomat
```

2.2.5 CHANGETASKPRI

De instructie CHANGETASKPRI bespreken we uitgebreid in hoofdstuk 5. Hier volstaan we met een korte beschrijving.

In de Amiga zijn niet alle taken even belangrijk. Het gewicht van een taak heet de prioriteit. Alleen van de actuele taak in de CLI kunt u de prioriteit veranderen met de instructie CHANGETASKPRI. Geef de actuele taak de prioriteit +5 met de opdracht:

```
changetaskpri 5
```

De opdracht 'status full' heeft op het eerder gegeven voorbeeld als resultaat:

```
Task 1: stk 1600, gv 150, pri 5 Loaded as command: status
Task 2: stk 3200, gv 150, pri 0 Loaded as command: textomat
```

Als u een waarde opgeeft buiten het toegestane gebied, krijgt u als antwoord:

```
Priority out of range (-128 to +127)
```

2.2.6 BREAK

De toetsencombinatie Ctrl-C breekt de uitvoering van een onderhanden CLI-instructie af. Vanuit een ander venster kunt u dat ook softwarematig veroorzaken met de instructie BREAK.

Neem aan dat u in venster 1 een complete inhoudsopgave hebt opgevraagd met 'dir opt a'. Onderbreek de uitvoer vanuit venster 2 met de opdracht 'break 1'. (In dit geval bereikt u hetzelfde door eerst venster 1 te activeren en daar Ctrl-C te geven.)

De instructie BREAK is ook bruikbaar om een taak te onderbreken die is gestart met RUN. Een taak die op de achtergrond loopt is niet meer vatbaar voor de toetsencombinatie Ctrl-C. Het is niet erg elegant om uitvoer via de printer te onderbreken door de printer uit te zetten. De opdracht 'break 2' brengt de printer tot zwijgen, aangenomen dat het in de opdracht opgegeven nummer gelijk is aan het nummer dat door het systeem aan die taak is toegekend.

Als u de instructie BREAK geeft zonder parameters, heeft ze hetzelfde effect als Ctrl-C. De toetsencombinaties van Ctrl met een van de letters D tot en met F zijn equivalent met die letter als optie van de instructie BREAK. Voorbeeld:

```
break 3 d
```

Neem aan dat de opdracht 'search df0: all' taaknummer 3 heeft gekregen. Na invoer van de opdracht 'break 3 d' gaat de routine meteen verder met zoeken in het volgende bestand. Het effect van Ctrl met een van de toetsen D tot en met F hangt af van de betrokken CLI-instructie. In de meeste gevallen gebeurt er helemaal niets.

U kunt ook grof geschut gebruiken en alle besproken toetsencombinaties tegelijk versturen naar een taak:

```
break 3 all
```

2.2.7 PATH

De instructie PATH zonder parameters (of met vermelding van 'show') laat de Amiga een lijst uitvoeren, die er bijvoorbeeld als volgt uitziet:

```
Current directory  
RAM:c  
A500 WB 1.2:System  
C:
```

Deze lijst geeft aan in welke directories en in welke volgorde de Amiga zoekt naar een bestand. In paragraaf 2.1 bespraken we al dat AmigaDOS eerst zoekt in de actuele directory (current directory). Welke dat is, kunt u opvragen met de instructie CD (zonder argument).

Als AmigaDOS hier niets vindt, worden achtereenvolgens de directory 'RAM:C' en de directory 'System' van de Workbench-diskette doorzocht. Ten slotte komt het device C aan de

beurt. Dat is geen echt apparaat, maar een pseudo-device dat de Amiga hanteert om CLI-instructies in de juiste directory te kunnen vinden. (De functie van zulke devices bespreken we bij de instructie ASSIGN.)

De instructie PATH kan een pad toevoegen aan of verwijderen uit de directory-structuur. Bijvoorbeeld:

```
path utilities add
```

Na deze opdracht hoeft u niet meer de opdracht 'utilities/calculator' te geven als u de rekenmachine in de directory 'Utilities' wilt gebruiken. 'Calculator' is voldoende. De lijst met paden is door bovengenoemde opdracht uitgebreid met de boeking:

```
A500 WB 1.2:Utilities
```

AmigaDOS zoekt automatisch in alle opgegeven paden.

Nieuwe paden worden altijd geactiveerd vóór de installatie van het device C:. Als u alle of enkele DOS-routines naar een directory op de RAM-disk kopieert en een pad opgeeft naar die directory, spaart u daarmee veel tijd en diskettewisselingen uit. Dat geldt vooral als u maar één diskdrive hebt. Pas als het besturingssysteem een DOS-routine niet kan vinden via de opgegeven paden, krijgt u het verzoek de Workbench-diskette in de drive te doen. In paragraaf 3.5 vindt u nog een paar opmerkingen over dit onderwerp. U kunt maximaal tien nieuwe paden tegelijk aanleggen in één opdracht. Het woord 'add' sluit zo'n lijst af. Als u een lijst met paden afsluit met 'reset', worden alle paden gewist behalve de in de lijst genoemde (maximaal 10). Na de opdracht 'path reset' zijn alle paden gewist behalve dat naar de actuele directory en naar het device C. Laatst genoemd pad moet altijd in de lijst staan.

2.2.8 ASSIGN

De mogelijkheden van de instructie ASSIGN worden het snelst duidelijk als u het resultaat bekijkt van de opdracht 'assign' of 'assign list':

Volumes:

RAM Disk [Mounted]

A500 WB 1.2 [Mounted]

Directories:

S	A500 WB 1.2:s
L	A500 WB 1.2:l
C	A500 WB 1.2:c
FONTS	A500 WB 1.2:fonts
DEVS	A500 WB 1.2:devs
LIBS	A500 WB 1.2:libs
SYS	A500 WB 1.2:

Devices:

DF0 DF1 PRT PAR SER

RAW CON RAM

Onder het kopje 'Volumes' staan de namen van de diskettes die AmigaDOS kent. De aanduiding 'Mounted' wil zeggen dat de diskette in de diskdrive zit. Voor de RAM-disk moet u dat niet al te letterlijk opvatten.

Onder 'Directorys' staan links de pseudo-devices. Die met de naam 'C' kent u al uit de vorige paragraaf. In de lijst staan de volledige paden die het besturingssysteem toekent aan de werkelijk bestaande directories voor elk van deze devices. De naam van de devices kunt u opvatten als een synoniem voor het pad dat ernaast is vermeld. Het pseudo-device 'C' is dus een pad naar de directory 'C' op de Workbench-diskette. De naam van een device hoeft niet gelijk te zijn aan de naam van zijn directory. Een programma dat het device 'Fonts:' gebruikt, mag u ook verwijzen naar een directory met een zelfgekozen naam, bijvoorbeeld 'tekensets'.

De instructie ASSIGN verwijst een device via een pad naar een directory. De syntaxis van de instructie is:

```
assign NAME,DIR,LIST/S
```

NAME	De naam van een device, inclusief de dubbele punt die dient als herkenningsteken voor AmigaDOS.
DIR	Als hier een volledig pad is vermeld, wordt aan de boeking 'dir' het opgegeven pad toegerekend. Bij gebrek aan een pad wordt de boeking 'dir' uit de lijst gewist.
LIST	Deze toevoeging veroorzaakt na een wijziging in de toekenningen de uitvoer van de actuele lijst. Als u niets hebt veranderd, hoeft u de optie 'list' niet te vermelden bij de instructie LIST.

Ten slotte staan onder de kop 'Devices' de namen van de apparaten die u vanuit de CLI kunt aanspreken. In paragraaf 3.5 vindt u meer toepassingen van de instructie ASSIGN.

2.2.9 ADDBUFFERS

Bij het werken met de CLI zal het u zijn opgevallen dat de routines voor de instructies bij hun eerste oproep moeten worden geladen vanaf de diskette, maar dat ze bij herhaalde toepassing direct ter beschikking zijn. Bij een diskdrive hoort een aantal buffergeheugens waarin opgeroepen programma's worden opgeslagen voordat ze verder worden verwerkt. Als een programma helemaal in de buffer past, blijft het daar beschikbaar en hoeft het niet te worden geladen bij een volgende oproep. Dat maakt de snelheid veel hoger.

Met de instructie ADDBUFFERS kent u een grotere buffer toe aan een diskdrive. De opdracht 'addbuffers df0: 11' koppelt aan drive 0 een extra werkgeheugen van 11 blokken (1 blok = 512 bytes). Dat is voldoende om een van de 160 sporen van een diskette volledig te laden in het geheugen.

De ruimte in de buffers gaat uiteraard ten koste van het systeemgeheugen. Bovendien zijn er per blok 32 bytes nodig voor de administratie van dat blok. Het is (nog) niet mogelijk aangeemaakte buffers softwarematig te wissen. U kunt ze alleen kwijtraken door de Amiga opnieuw op te starten.

2.2.10 WHY

Met de instructie **WHY** vraagt u op waarom de Amiga een gegeven opdracht niet kon uitvoeren. Neem aan dat u de startup-sequence wilt bekijken. Dat kan met de volgende opdracht:

```
type s/startup-sequentie
```

Het besturingssysteem reageert daarop met:

```
Can't open s/startup-sequentie
```

Als u met **WHY** opvraagt wat er verkeerd is, is de reactie:

```
Last command failed because Error code 205
```

Vraag door met de opdracht 'Fault 205'. Het antwoord is 'Object not found'. Dat is logisch, gezien de verkeerd ingevoerde naam van het van het gezochte bestand.

2.2.11 FAULT

De instructie **FAULT** levert de leesbare tekst die behoort bij een gegeven foutnummer. Lang niet bij alle foutnummers is een toelichtende tekst beschikbaar. In zulke gevallen herhaalt het besturingssysteem het foutnummer. Voorbeelden:

- invoer: fault 10
uitvoer: Fault 10: Error 10
- invoer: fault 120
uitvoer: Fault 120: Argument line invalid or too long

2.2.12 DATE

Met de instructie **DATE** kunt u op elk moment, buiten het programma 'Preferences' om, de tijd en de datum opvragen of veranderen. De syntaxis van de instructie is:

```
TIME,DATE,TO=VER/K
```

Schrijf op de plaats van **TIME** de tijd volgens het formaat HH:MM:SS (uren, minuten, seconden). U mag de seconden ook weglaten (HH:MM).

Schrijf op de positie van **DATE** een datum volgens het formaat DD-MMM-YY (dag, maand, jaar). Als de datum minder dan een week ligt voor de actuele datum, kunt u op deze plaats de weekdag invoeren. De Amiga kent het begrip 'yesterday' om de datum een dag terug te zetten.

Met de optie 'TO' (of 'VER') wordt de uitvoer omgeleid naar een bestand. Als u de instructie geeft zonder parameters, bestaat de uitvoer op het scherm uit de actuele dag, datum en tijd:

```
Wednesday 21-Oct-87 10:17:48
```

De instructie DATE is niet bruikbaar om uit te rekenen op welke dag u bent geboren. De eeuwigdurende kalender begint pas te tellen op 2 januari 1978. Op 1 januari 1978 reageert het besturingssysteem met 'unset' (niet ingesteld) en eerdere data worden bestempeld als ongeldig (invalid).

2.2.13 SETCLOCK

De instructie SETCLOCK is alleen van belang voor eigenaren van een Amiga met een klok met accuvoeding. Deze klok en de systeemklok lopen onafhankelijk van elkaar. De instructie SETCLOCK geeft de tijd van de ene klok door aan de andere. De opdracht 'setclock opt load' zet de systeemklok gelijk met de permanente klok. Het omgekeerde bereikt u met de opdracht 'setclock opt save'. Geef zo nodig eerst de systeemklok een nieuwe tijd met behulp van de instructie DATE.

Meestal is de opdracht de real-time clock uit te lezen opgenomen in de startup-sequence op een boot-diskette. Op de originele Workbench-diskette staat daar de regel 'setclock > nil: opt load'. De uitvoer van de instructie wordt omgeleid naar het device 'nil:'. Omdat de uitvoer verdwijnt in dit pseudo-device, ziet u er niets van op het scherm. De instructie SETCLOCK geeft namelijk de volgende reactie als er geen real-time clock is of als de accu leeg is:

Internal clock not functioning

Om te voorkomen dat u er steeds aan wordt herinnerd dat u geen permanente klok hebt, wordt deze boodschap met de hierboven gegeven regel weggewerkt. De hele procedure duurt ongeveer zes seconden en heeft dus geen enkel resultaat. U kunt die regel zonder meer uit de startup-sequence verwijderen.

2.2.14 PROMPT

In DOS is de prompt een teken(combinatie) die meldt dat de computer klaarstaat om nieuwe invoer te verwerken. Bij de Amiga bestaat hij standaard uit een punthaak en het nummer van de CLI waarin hij staat (bijvoorbeeld: 1 >).

De instructie PROMPT zet een ingevoerde tekst in de plaats van de oude prompt. Als de tekst van de prompt spaties bevat moet hij in zijn geheel tussen dubbele aanhalingstekens staan:

prompt "Wat kan ik voor u doen?"

Zonder argumenten levert de instructie PROMPT alleen de punthaak. Als u in de tekst voor de prompt de uitdrukking '%n' opneemt, voert de CLI daar het nummer van het actuele venster uit:

prompt "Dit is venster %n "

Voor het derde CLI-venster is de prompt:

Dit is venster 3

De standaard-prompt krijgt u terug met de opdracht:

```
prompt %n>
```

2.2.15 STACK

Elke CLI-taak reserveert een eigen geheugengedeelte voor de opslag van AmigaDOS-routines. Dat geheugen wordt 'stack' (stapelgeheugen) genoemd. Standaard heeft het een grootte van 4000 bytes per CLI. Met de instructie STACK kunt u dat veranderen in een willekeurige, gehele waarde groter dan 1600. Wees niet te zuinig met ruimte voor de stack. Het systeem gaat hangen als er niet voldoende ruimte is voor een routine. Dat gebeurt zonder Guru-meditatie, zodat er geen ontsnapping mogelijk is. Met name de instructie DIR is gevoelig op dit punt. Doe het volgende experiment alleen als er niets belangrijks meer in het werkgeheugen staat.

```
stack 1600  
dir opt a
```

De instructie SORT heeft dezelfde moeilijkheid. De benodigde ruimte is in dit geval afhankelijk van de grootte van het te sorteren bestand. Hier zijn geen algemeen geldige richtlijnen te geven.

Houd er rekening mee dat een taak een even grote stack krijgt toegewezen als de CLI van waaruit die taak is gestart. Dat kost zeer veel geheugen als u begonnen bent met een vergrote stack. De instructie STATUS zonder argumenten levert de actuele grootte van het stapelgeheugen.

2.2.16 BINDDRIVERS

De instructies BINDDRIVERS en MOUNT zijn alleen van belang voor bezitters van minder gebruikelijke randapparatuur voor de Amiga (harddisk, plotter, enzovoorts). Deze instructies vindt u in de startup-sequence van een boot-diskette. De instructie BINDDRIVERS installeert de drivers (stuurprogramma's) in de directory 'Expansion'. Bij aankoop van een randapparaat krijgt u de benodigde driver(s) meegeleverd. Als u zulke apparatuur niet bezit, kunt u de bijbehorende procedure zonder meer schrappen uit de startup-sequence. De directory 'Expansion' is in dat geval ook overbodig. Een en ander verkort de opstartprocedure met een paar seconden.

2.2.17 MOUNT

Het is niet voldoende de stuurprogramma's voor de randapparatuur op te nemen in het systeem met de instructie BINDDRIVERS. Als u met de instructie ASSIGN opvraagt of de 5,25"-diskdrive (df2:) in de lijst met devices staat na toevoeging van zijn driver, blijkt dat niet het geval. De instructie MOUNT meldt het systeem dat een randapparaat beschikbaar is. In het bestand 'MountList' in de subdirectory 'Devs' staan de gegevens over de geïnstalleerde apparaten. De fabrikant van het randapparaat behoort de software te leveren om dit alles te regelen.

Achter de instructie MOUNT hoeft u alleen de naam van het nieuwe apparaat te schrijven:

mount res0:

Na deze opdracht staat de naam van het apparaat 'Res:' in de lijst die wordt uitgevoerd na de instructie ASSIGN.

2.3 Instructies voor batchbestanden

In deze paragraaf bespreken we de instructies die voornamelijk worden gebruikt in batchbestanden (stapelbestanden). Het zijn zuivere tekstbestanden die bestaan uit een willekeurig aantal CLI-opdrachten. De reeks opdrachten in het bestand wordt uitgevoerd na de instructie EXECUTE. Hoofdstuk 4 is helemaal gewijd aan dit onderwerp.

2.3.1 EXECUTE

Een batchbestand kan net als elk ander tekstbestand worden aangemaakt en veranderd met een tekstverwerkingsprogramma of editor, bijvoorbeeld ED (zie paragraaf 2.4). Een batchbestand is zelf geen programma. Het is dus niet mogelijk het te starten door invoer van zijn naam zonder meer. Dat veroorzaakt foutmelding nummer 121: File is not an object module (bestand is geen uitvoerbaar programma). U hebt daarvoor de instructie EXECUTE nodig. Neem aan dat het batchbestand 'afdrukken' de volgende twee regels bevat:

```
type teksten/brief to prt:
date
```

Invoer van 'execute afdrukken' heeft hetzelfde effect als invoer van die twee regels via de CLI. Na het afdrukken van de brief in de directory 'teksten' verschijnen datum en tijd op het scherm.

Aan de instructie EXECUTE mag u naast de verplichte naam van het uit te voeren bestand argumenten toevoegen. Die worden overgedragen aan het batchbestand met variabelen om deze waarden op te slaan.

In het hierboven gegeven voorbeeld mag een variabele staan op de plaats van het met name vermelde bestand:

```
.key name
type <name> to prt:
date
```

Roep het bestand 'afdrukken' op met:

```
execute afdrukken teksten/brief
```

Bij de toepassing van variabelen in batchbestanden gelden een paar regels:

1. De instructie '.key' declareert de variabelen. Dit moet de eerste opdracht van het batchbestand zijn.
2. Als er meer dan één variabele voorkomt in het batchbestand, declareert u alle namen achter elkaar, gescheiden door komma's. De instructie '.key' mag maar eenmaal in het batch-

bestand staan. Anders krijgt u de foutmelding 'Execute: More than one K directive'. Een correct voorbeeld:

```
.key bestandsnaam,doelapparaat  
copy <bestandsnaam> to <doelapparaat>
```

3. Uit de voorbeelden blijkt dat de variabele tussen punthaken (< en >) wordt vervangen door de gegeven waarde die u opgeeft in de EXECUTE-opdracht.

Er zijn meer instructies voor batchbestanden die beginnen met een punt.

.def Geeft een variabele een waarde voor het geval dat die niet uitdrukkelijk is vermeld in de EXECUTE-opdracht. Dit heet een verstekwaarde (default). Deze optie hoeft niet per se aan het begin van het batchbestand te staan. Bijvoorbeeld:

```
.key bestand,apparaatnaam  
.def apparaatnaam prt:  
type <bestand> to <apparaatnaam>
```

Als u in de EXECUTE-opdracht geen apparaatnaam vermeldt, gaat de uitvoer van het gegeven bestand automatisch naar de printer. Speciaal voor het geval van het bovengenoemde voorbeeld bestaat er een eenvoudiger procedure. Schrijf tussen de punthaken achter de variabele een \$-teken plus de tekst die in de plaats van de variabele komt als de variabele in de EXECUTE-opdracht geen waarde heeft gekregen. Bijvoorbeeld:

```
.key bestand,apparaatnaam  
type bestand to <apparaatnaam$prt:>
```

In het voorbeeld gaat de uitvoer bij verstek van 'apparaatnaam' naar de printer (de verstekwaarde is 'prt:').

.dol Vervangt het dollarteken aan het begin van een default-tekst door een willekeurig ander teken. Bijvoorbeeld:

```
.dol #
```

Hierdoor verandert de regel met '.def' in het voorbeeld in:

```
type bestand to <apparaatnaam#prt:>
```

.bra Analooq aan '.dol', maar nu ter vervanging van de linker punthaak (<).

.ket Als '.bra', maar nu voor de rechter punthaak (>). (De namen van de twee opties vormen in het Engels samen een ezelsbruggetje: bracket, haakje.)

Een punt gevolgd door minstens een spatie dient als kenmerk van een commentaarregel (vergelijkbaar met REM in BASIC).

.dot Vervangt de punt voor de opties door een ander teken.

We raden het gebruik van de hierboven besproken veranderingen van stuurtekens af. Zulke veranderingen vormen geen bijdrage aan de duidelijkheid.

2.3.2 ECHO

Met de instructie ECHO kunt u een willekeurige reeks tekens uitvoeren via een apparaat naar keuze, meestal het scherm:

```
echo "Wakker worden, de zon schijnt al uren!"
```

Leid de uitvoer, net als bij andere instructies, om naar een ander apparaat met een punthaak:

```
echo >prt: "Wordt het niet eens tijd voor een nieuw lint?"
```

De aanhalingstekens zijn alleen nodig als er spaties voorkomen in de tekst.

De instructie ECHO heeft een optie '*n' die de uitvoer voortzet op de volgende regel. Bijvoorbeeld:

```
echo "Pas op,*n      afstapje!"
```

Uitvoer op het scherm:

```
Pas op,  
      afstapje!
```

Gebruik '**n' als u de tekencombinatie '*n' letterlijk in uw tekst wilt opnemen.

2.3.3 FAILAT

Een CLI-instructie levert een foutnummer als er iets is misgegaan bij de uitvoering van een opdracht. In de CLI hoort bij de meeste nummers een leesbare tekst. Foutnummer 216 treedt bijvoorbeeld op als u probeert een directory te wissen terwijl hij nog boekingen bevat.

Het besturingssysteem breekt de uitvoering van een batchbestand af zodra er een fout optreedt met een nummer groter of gelijk aan 10. Het systeem keert dan terug naar het programma één niveau hoger, bijvoorbeeld de CLI. Vraag de actuele waarde van de foutgrens op met de instructie FAILAT. Soms is het wenselijk dat de behandeling van een batchbestand al stopt als er sprake is van een 'Warning' (waarschuwing; foutnummer kleiner dan 10).

Compilers maken meestal een onderscheid tussen waarschuwingen en echte fouten. Waarschuwingen mag u negeren, maar het zijn signalen die optreden bij een slechte programmeerstijl. Als u voor een batchbestand de foutdrempel verlaagt tot één, stopt de uitvoering ervan bij elke twijfelachtige opdracht. Op die manier is het onmogelijk dat volgende procedures worden opgeroepen (assembleren, linken).

Verander de foutdrempel met de instructie FAILAT gevolgd door de laagste waarde waarbij het batchbestand mag worden onderbroken. Deze waarde geldt alleen voor de duur van dit batchbestand. Vervolgens komt de drempel automatisch weer op 10 te liggen.

Als u in een CLI-venster een nieuwe foutdrempel definieert, geldt die ook in een in dat venster opgeroepen batchbestand. Verandert dat batchbestand vervolgens op zijn beurt de foutdrempel, dan geldt na terugkeer naar de CLI in ieder geval weer de waarde 10. Als in het batchbestand de functie FAILAT niet voorkomt, of alleen om de actuele waarde op te vragen, dan blijft de nieuwe waarde gelden. In dochter-CLI's geldt automatisch dezelfde foutdrempel

als in de oproepende CLI. De dochter-CLI's kunt u daarna elk een eigen foutdrempel toekennen.

2.3.4 QUIT

Met de instructie QUIT breekt u een batchbestand op een willekeurige plaats af. Aan het einde van een batchbestand is QUIT niet verplicht. Als u de reden van het afbreken van een batchbestand door een fout wilt doorgeven aan het oproepende niveau gebruikt u de instructie QUIT gevolgd door het nummer van die fout. Als het oproepende niveau de CLI is, verschijnt, mits het foutnummer 'xx' groter is dan 10, de melding:

```
quit failed returncode xx
```

2.3.5 IF/ELSE/ENDIF

De drie instructies IF, ELSE en ENDIF behandelen we in één keer, omdat ELSE en ENDIF alleen mogen voorkomen in combinatie met IF. Deze instructies maken het mogelijk voorwaarden te stellen aan de uitvoering van een gedeelte van een batchbestand. In het eenvoudigste geval hebt u alleen IF en ENDIF nodig:

```
if exists teksten/brief
type teksten/brief to prt:
endif
echo "Heb ik de brief nu wel of niet geprint?"
...
...
```

In dit voorbeeld wordt de TYPE-opdracht alleen uitgevoerd als het bestand 'brief' in de directory 'teksten' werkelijk bestaat. Als dat niet zo is, gaat het besturingssysteem verder met de opdracht die volgt op de instructie ENDIF. De voorwaarde EXISTS laat het besturingssysteem op de diskdrive en de RAM-disk zoeken naar het vermelde bestand.

Naast EXISTS bestaan er nog een vijftal voorwaarden om achter IF te vermelden.

EQ (equal, gelijk)

De voorwaarde EQ vergelijkt twee teksten. Bijvoorbeeld:

```
if "Dit is de tekst" eq "Dit is de tekst"
echo "Ja, de twee teksten zijn gelijk."
endif
```

Het heeft natuurlijk weinig zin twee gelijke constanten te vergelijken zoals hierboven. De instructie EQ is alleen zinvol bij de vergelijking van variabelen.

Voorbeeld 1:

```
.key invoer
```

```

if <invoer> eq brief
echo "U hebt het woord 'brief' ingevoerd."
endif

```

Voorbeeld 2:

```

.key invoer
if <invoer> eq ""
echo "U hebt niets ingevoerd."
endif

```

Let op het verschil tussen de variabele (invoer), de waarde van de variabele ('invoer' tussen punthaken) en de tekst ('brief' of ""). Bij de vergelijking van tekst maakt de routine EQ geen onderscheid tussen kleine letters en hoofdletters. Het resultaat van 'IF Brief EQ brief' is dus 'waar'.

FAIL

Met de combinatie IF-FAIL kunt u controleren of de laatst uitgevoerde opdracht een fout heeft veroorzaakt met een nummer groter dan of gelijk aan 20. Deze test vindt alleen plaats als u voor de opdracht met de fout de foutdrempel hebt gedefinieerd op een waarde groter dan twintig. Anders zou de afhandeling van het batchbestand al zijn afgebroken voor de opdracht IF-FAIL aan bod komt.

ERROR

ERROR is analoog aan FAIL, maar de foutdrempel ligt bij ERROR altijd bij foutnummer 10.

WARN

Bij IF-WARN ligt de foutdrempel op vijf. U hoeft dus niet eerst met FAILAT de drempel te verhogen boven de standaarddrempel (10). Laat u niet in de war brengen door de aanduidingen WARN en ERROR. Als een fout met nummer 255 wordt afgevangen met IF-WARN is het geen waarschuwing, maar wel degelijk een ernstige fout. Vang dus eerst fouten met hoge nummers af als u de foutdrempel hebt verhoogd met FAILAT.

NOT

De toevoeging NOT verandert een voorwaarde in haar tegendeel. Voorbeeld:

```

.key tekst
if not exists <tekst>
echo "Dit bestand is er niet."
endif
if exists <tekst>
echo "Aan de slag."
type <tekst> to prt:
endif

```

De oproep van het batchbestand moet de naam van een willekeurig bestand bevatten. Deze wordt opgeslagen in de variabele 'tekst'. Het eerste deel test of het opgegeven bestand helemaal niet bestaat. In dat geval wordt de tekst achter ECHO uitgevoerd.

Het tweede deel van het batchbestand behandelt het andere geval: het bestand is wel aanwezig. Dit voorbeeld is uiteraard alleen van toepassing op tekstbestanden.

De negatieve en de positieve test kunnen worden gecombineerd met de instructie ELSE. We passen ELSE toe op het voorbeeld hierboven:

```
.key tekst
if not exists <tekst>
  echo "Dit bestand is er niet."
else
  echo "Aan de slag."
  type <tekst> to prt:
endif
```

Dit batchbestand heeft hetzelfde resultaat als het vorige, maar nu eenvoudiger en sneller (het scheidt het laden van een routine).

Beide blokken opdrachten in de constructie IF-ELSE-ENDIF mogen bestaan uit een willekeurig aantal opdrachten. Sluit ze af met respectievelijk ELSE en ENDIF. In elk blok kunt u een willekeurig aantal IF-opdrachten in elkaar schuiven (nesten). Een ELSE- of ENDIF-instructie heeft altijd betrekking op de dichtstbijzijnde voorgaande IF.

Het onderstaande voorbeeld illustreert de nesting van voorwaarden. Het batchbestand onderzoekt hoeveel parameters de instructie EXECUTE heeft meegekregen (maximaal drie). De schrijfwijze met inspringingen maakt het makkelijker de structuur van het batchbestand te doorzien.

```
.key tekst1,tekst2,tekst3

if not <tekst1> eq ""
  if not <tekst2> eq ""
    if not <tekst3> eq ""
      echo "Alle drie parameters hebben een waarde."
    else
      echo "De derde waarde ontbreekt."
    endif
  else
    echo "De tweede en de derde waarde ontbreken."
  endif
else
  echo "U hebt niets ingevoerd."
endif
```

2.3.6 ASK

Merkwaardig genoeg wordt de instructie ASK in bijna geen enkele publikatie besproken. ASK mag los worden gebruikt of in combinatie met een tekst (een vraag) die op het scherm verschijnt als de instructie wordt uitgevoerd. De Amiga wacht op uw reactie op die tekst. U kunt daarbij alleen maar kiezen uit 'Yes', 'No', 'Y' en 'N' gevolgd door een Return. Elk ander antwoord leidt tot een herhaling van de vraag. De computer interpreteert uw antwoord en rea-

geert met foutcode 5 als uw antwoord bevestigend was. Verlaag eerst de foutdrempel om dat mogelijk te maken.

```
failat 5
ask "Moet ik meteen ophouden met mijn taak? (y/n)"
echo "Dan ga ik maar weer eens verder."
...
```

Alleen na invoer van 'Yes' springt het besturingssysteem terug naar de CLI.

Deze oplossing is niet erg fraai. Het zou mooier zijn als de CLI, al naar gelang uw antwoord, zou springen naar een van twee bijpassende verschillende routines. Dat is mogelijk met de opdracht IF WARN-ELSE (zie paragraaf 2.3.5). Dat heeft bovendien het voordeel dat U de foutdrempel niet hoeft te verlagen tot een waarde kleiner dan 6. Bijvoorbeeld:

```
ask "Kent u de instructie ASK? (y/n)"
if warn
  echo "Prima, ga door."
else
  echo "Verlaag de foutdrempel naar 5."
endif
```

2.3.7 SKIP en LAB

Zodra de CLI bij de uitvoering van een batchbestand de instructie SKIP tegenkomt, zoekt hij in de tekst van het bestand naar het woord LAB (label, etiket) en voert het bestand vanaf die plaats verder uit. Als achter de instructie SKIP een naam staat, zoekt de routine naar die naam.

Voorbeeld:

```
ask "Weet u hoe u SKIP en LAB moet gebruiken? (y/n)"
if warn
  skip markering
endif
echo "Dat is niet zo erg."
quit

lab markering
echo "Niet te vaak gebruiken."
```

De instructie SKIP vertoont enige overeenkomst met de instructie GOTO (uit BASIC of C) en JMP (uit Assembler). Een belangrijk verschil is dat u met SKIP niet kunt terugspringen in de programmatekst.

Sprongen maken een programma onoverzichtelijk. Wees er dus zuinig mee. Bovenstaand programmaatje is eigenlijk een slecht voorbeeld, want het programma springt uit een IF-THEN-constructie. Helaas is het met de instructie SKIP meestal niet mogelijk dat netter te doen. In bijna alle gevallen is een IF-THEN-ELSE-constructie een betere keus.

2.3.8 WAIT

Soms zit de computer zichzelf in de weg met zijn grote snelheid. Dit probleem treedt bijvoorbeeld op als de Amiga tegelijk twee taken met dezelfde diskdrive moet uitvoeren (geldt niet voor de RAM-disk). Als u in een CLI-venster met DIR de inhoudsopgave vraagt van een diskette in drive 0 en vanuit een ander CLI-venster de opdracht 'list df0:', dan worden deze taken parallel uitgevoerd. Het resultaat is dat dat meer tijd vraagt dan wanneer u de twee opdrachten na elkaar invoert in hetzelfde venster. De verdeling van de rekentijd van de processor over de twee taken (multitasking) heeft tot gevolg dat de schrijf- en leeskop van de diskdrive steeds heen en weer wordt gestuurd om afwisselend een paar sporen te lezen voor de ene en de andere taak.

Een extremer voorbeeld betreft een startup-sequence die twee programma's tegelijk laadt. Het 'verknipte' gedrag is het duidelijkst als de gegevens van programma 1 in de buurt van spoor 1 liggen en die van programma 2 in de buurt van spoor 79. De schrijf- en leeskop maakt voortdurend sprongen over een afstand van bijna 80 sporen.

Voeg in zo'n geval pauzes toe met de instructie WAIT. Gebruik in het laatste geval een pauze die lang genoeg is om programma 1 helemaal te laden.

Het argument van WAIT is de lengte van de pauze in seconden of minuten. De verstekwaarde is één seconde. U kunt ook een tijdstip opgeven waarop de Amiga verder moet gaan (systeemtijd). Een paar voorbeelden:

wait	wacht 1 seconde
wait 5	wacht 5 seconden
wait 5 sec	wacht 5 seconden
wait min	wacht 1 minuut
wait 5 min	wacht 5 minuten
wait until 14:30	wacht tot 14:30 uur

U kunt de instructie WAIT net als elke andere instructie op een willekeurig moment afbreken met de toetsencombinatie Ctrl-C.

In paragraaf 3.7 staat een voorbeeld van een CLI-wekker die gebruik maakt van de instructie WAIT.

2.4 De editors ED en EDIT

ED en EDIT passen niet goed in de indeling die we in dit hoofdstuk gebruiken. Hun omvang geeft al aan dat ze een nogal afwijkende taak hebben. Het zijn editors, beide ongeveer 19 Kbyte groot.

Het programma EDIT is een regeeditor (line editor) die maximaal één regel tegelijk kan veranderen. Met EDIT is het dus niet mogelijk in een venster veranderingen aan te brengen die zich uitstrekken over meer dan één regel. De CLI werkt ook volgens dit principe: alleen in de actuele regel kunt u opdrachten invoeren.

Het programma ED werkt venstergeoriënteerd (window editor). Het programma laadt de hele tekst in het werkgeheugen van de Amiga en geeft van die gegevens, afhankelijk van de grootte van het venster, een gedeelte weer op het scherm. Met de vier pijltjestoetsen zet u de schrijfpunctie (de cursor) op een willekeurige plaats in het venster. Als u probeert onder of boven uit het venster te gaan, wordt het zichtbare gedeelte van de tekst automatisch verschoven

(gescrolld). Als de cursor de linker- of de rechterrاند van het venster raakt, en de tekst omvat meer kolommen dan de breedte van het ED-venster, schakelt ED automatisch over naar de beledende helft van de tekst.

Het toepassingsgebied van de twee editors beslaat een breed terrein, van batchbestanden en broncodes voor een compiler tot tekstverwerking. Vanwege zijn bedieningsgemak heeft ED meestal de voorkeur boven EDIT. Vooral overzicht over de tekst als geheel is zeer belangrijk bij het programmeren. Paragraaf 2.4.1 is gewijd aan ED.

EDIT heeft als voornaamste voordeel dat hij weinig beslag legt op het geheugen voor de geladen tekst (één regel). Bij een tekstbestand dat zo groot is dat het niet in het geheugen past, bent u wel gedwongen EDIT te gebruiken. Met EDIT kunt u een paar bronbestanden tegelijk openen, daaruit naar keuze een regel citeren en deze na bewerking opnemen in een doelbestand. EDIT is zo flexibel dat over deze editor alleen al een heel boek is te schrijven. In paragraaf 2.4.2 komen daarom alleen een paar basisfuncties van EDIT aan de orde.

2.4.1 Overzichtelijke tekstverwerking met ED

Roep de bestandseditor op met ED en daarachter (een pad en) het te bewerken bestand. Bijvoorbeeld:

```
ed teksten/prog
```

We onderscheiden twee gevallen:

1. Het bestand is aanwezig in de opgegeven directory.
ED probeert het opgegeven bestand in het geheugen te laden. Als dat is gelukt, verschijnt het venster van ED met de tekst of een gedeelte daarvan, afhankelijk van de omvang van beide. Een mislukking kan te wijten zijn aan een tekort aan vrij geheugen: 'Unable to open window x'. Het bestand kan ook een directory zijn of niet-afdrukbare tekens bevatten: 'X is a directory and cannot be edited' respectievelijk 'File contains binary'.
2. Er is geen bestand met de opgegeven naam in de vermelde directory.
Op het scherm verschijnt alleen een leeg venster, aangenomen dat er voldoende systeemgeheugen beschikbaar is. ED maakt niet automatisch een bestand aan met de opgegeven naam.

Met de optie SIZE kunt u de grootte van het werkgeheugen van ED definiëren, uitgedrukt in bytes:

```
ed teksten/prog size 60000
```

Zonder deze vermelding hanteert ED een verstekwaarde van 40 000 bytes, tenzij het te laden bestand groter is.

Definieer zo nodig een andere grootte en plaats van het venster. In het venster werken de pijltjestoetsen en de toetsen Backspace en Delete op de bekende manier. Onder ED heeft de muis geen functie.

U kunt ED op twee manieren een opdracht geven:

1. Direct, door een combinatie van de Ctrl-toets en een andere toets. De instructie die daar

bijhoort, wordt meteen uitgevoerd en u kunt direct verder gaan met redigeren.

2. Via de opdrachtenregel die aan de onderrand van het venster verschijnt, zodra u de Esc-toets indrukt. Die regel herkent u aan het teken '*' in de eerste kolom. Zolang het sterretje op het scherm staat, is ED in de opdrachtenmodus of indirecte modus. U kunt dan tijdelijk geen veranderingen in de tekst aanbrengen. Pas na volledige invoer van een opdracht en de afhandeling ervan of door een 'loze' Return verlaat ED de indirecte modus. In de opdrachtenregel mogen meerdere opdrachten staan, mits ze van elkaar worden gescheiden door puntkomma's.

In paragraaf 8.1 vindt u een compleet overzicht van de instructies van ED. Hieronder beschrijven we de meest voorkomende; u zult daar meestal voldoende aan hebben.

Instructies in de directe modus

<i>Ctrl-A</i>	Onder de actuele regel verschijnt een blanco regel.
<i>Ctrl-B</i>	De actuele regel wordt helemaal gewist.
<i>Ctrl-G</i>	Herhaalt de laatst ingevoerde Esc-opdracht. Deze instructie is met name handig bij zoeken en vervangen.
<i>Ctrl-Y</i>	De actuele regel wordt vanaf de cursorpositie gewist.

Instructies voor de indirecte modus

Ga met behulp van de Esc-toets naar de indirecte modus, type de instructie en geef Return. We geven de instructies in volgorde van afnemend gebruik.

<i>X (exit)</i>	Slaat de tekst op en verlaat daarna ED. Bovendien wordt in de subdirectory 'T' een ongewijzigde versie van de tekst opgeslagen onder de naam 'ed-back-up'.
<i>Q (quit)</i>	Verlaat ED zonder dat er iets in welke vorm dan ook wordt opgeslagen. ED vraagt u vooraf om een bevestiging van de opdracht.
<i>SA (save as)</i>	Slaat de tekst op zonder ED te verlaten. Geef de tekst zo nodig een nieuwe naam of de naam van een bestand dat al bestaat. In het laatste geval bent u de oude inhoud van het bestand onherroepelijk kwijt. U kunt de gegevens ook meteen naar een randapparaat versturen. Bijvoorbeeld: SA "prt:".
<i>J (join)</i>	Koppelt twee regels. Dit is een snelle manier om regels door te laten lopen als u ze per ongeluk hebt gescheiden met een Return. Zet de cursor aan het einde van de tekst van de eerste regel.
<i>BS (block start)</i>	Markeert het begin van een blok voor blokoperaties. De regel waarin de cursor staat, geldt als bovenste regel van het blok.
<i>BE (block end)</i>	Definieert de regel met de cursor als laatste regel van het blok.
<i>IB (insert block)</i>	Plaats een kopie van het met BS en BE gemarkeerde blok op de positie van de cursor.
<i>DB (delete block)</i>	Wist het met BS en BE gemarkeerde blok helemaal.

2.4.2 Flexibele tekstbewerking met EDIT

Als u met EDIT werkt, kunt u het beste (tijdelijk) alles vergeten wat er is gezegd over ED. EDIT werkt volgens een heel ander principe:

In de buffer van EDIT staat altijd maar een klein aantal regels en u kunt die alleen maar na

elkaar redigeren. Zodra u de laatste regel in de buffer hebt bewerkt, laadt EDIT automatisch een nieuwe regel uit het bestand en schrijft hij de eerste regel in de buffer in het doelbestand. U kunt terug in de tekst zover als de voorraad regels in de buffer toelaat. Als een regel uit de buffer is verdwenen en is weggeschreven naar het doelbestand, kunt u er met EDIT niets meer aan veranderen.

Als u EDIT verlaat, wordt eerst de inhoud van de buffer helemaal in het doelbestand geschreven. Doe dat ook met een ongeredigeerde rest van het doelbestand om te voorkomen dat u gegevens kwijtraakt. (Het is ook mogelijk EDIT te verlaten zonder dat er lees- en/of schrijfoperaties plaatsvinden.)

In EDIT kunt u tijdens het werken met een bestand andere bestanden openen en lezen. Deze bestanden worden, net als het eerste bestand, regel voor regel geredigeerd vanaf het begin. EDIT onthoudt waar u bent gebleven als u een geopend bestand tijdelijk verlaat om een ander bestand te redigeren.

EDIT kan zijn redigeeropdrachten ophalen uit een daarvoor aangemaakt bestand (een *macro*, zie het einde van deze paragraaf).

Parameters van EDIT

De lengte van de beschrijving van de syntaxis van EDIT suggereert al iets over de veelzijdigheid van het programma. De syntaxis van de oproep luidt:

```
edit FROM/A, TO, WITH/K, VER/K, OPT/K
```

De vijf onderdelen van de syntaxis reserveren de volgende parameters:

- | | |
|-----------------|--|
| <i>FROM</i> | De vermelding van de naam van het te redigeren bestand is verplicht: /A. Dat bestand moet al bestaan; EDIT kan geen nieuw bestand aanmaken. |
| <i>TO</i> | Schrijf hier de naam van het doelbestand voor de geredigeerde tekst. Bij ontbreken van deze naam doet EDIT het volgende:
In de subdirectory 'T' van de actuele diskdrive maakt EDIT een hulpbestand aan waarin de geredigeerde tekst wordt opgeslagen. Bij het verlaten van EDIT, krijgt het hulpbestand de volledige padnaam van het bronbestand die u hebt opgegeven voor de boeking FROM. Het oorspronkelijke bronbestand gaat verder door het leven als 'edit-backup', totdat er overheen wordt geschreven aan het einde van de volgende sessie met EDIT. |
| <i>WITH</i> | Schrijf achter WITH de naam van de macro als EDIT zijn opdrachten uit een bestand moet halen. |
| <i>VER</i> | Alle uitvoer van EDIT gaat naar een apparaat naar keuze in plaats van naar het scherm. Door de opdracht 'ver tekstbestand' gaat de uitvoer van EDIT naar een bestand met de naam 'tekstbestand'. De opdracht 'ver con:10/10/300/100/vervenster' stelt u in staat in het venster met de naam 'ver-venster' te zien wat EDIT in zo'n bestand opslaat. |
| <i>OPT P</i> | Definieert het maximale aantal regels in de buffer. Bijvoorbeeld: opt p100. (De verstekwaarde is 40 regels.) Een groot aantal regels is gemakkelijk, maar kost meer systeemgeheugen. |
| <i>OPT W</i> | Definieert een andere maximale lengte van de regels. (De verstekwaarde is 120 tekens.) Voorbeeld: opt w81. |
| <i>OPT PxWy</i> | Een combinatie van OPT P en OPT W. |

EDIT starten

Meestal is het voldoende om de editor op te roepen met 'edit <bestandsnaam>'. EDIT slaat de eerste 40 regels van dat bestand op in een hulpbestand.

Als dat klaar is, verschijnt de prompt van EDIT. Voer achter de dubbele punt uw opdracht in. Voordat u kunt beginnen te redigeren, moet u opgeven in welke regel u iets wilt veranderen. Intern heeft EDIT alle regels van de brontekst doorlopend genummerd. Meteen na de start is de eerste regel van de brontekst de actuele regel. Helaas verschijnt de inhoud van die regel niet automatisch op het scherm. Er zijn verschillende opdrachten mogelijk om een andere regel actueel te maken:

- n (next) Sprong naar de volgende regel.
- p (previous) Sprong naar de vorige regel.
- mx (move) Sprong naar de regel met nummer *x*.

Voer achtereenvolgens *N* en *P* in voor de weergave van de inhoud van regel 1. Als u ze scheidt met een puntkomma, kunt u meerdere opdrachten in een keer invoeren. EDIT maakt geen onderscheid tussen kleine letters en hoofdletters.

Met behulp van *P* of *M* kunt u zover in de tekst terug als de inhoud van de buffer toelaat. Regels die al zijn opgeslagen in het doelbestand, kunt u niet meer terugroepen. Zodra u een regel activeert met een nummer groter dan 40, laadt EDIT zoveel regels als nodig is om de gekozen regel in de buffer op te slaan. Als u daarna probeert terug te keren naar regel 1, krijgt u de foutmelding 'Line number 1 too small'.

Een getal voor de instructies *P* en *N* definieert het aantal malen dat die instructie moet worden uitgevoerd. De opdracht '10n' laat EDIT tien regels verder gaan in de tekst.

Met de instructie *F* (find) kunt u EDIT laten zoeken naar een regel waarin een gegeven string voorkomt. Schrijf de gezochte string tussen twee gelijke, overigens willekeurige tekens. Bijvoorbeeld:

f ?trefwoord?

Te beginnen in de actuele regel zoekt EDIT naar het woord 'trefwoord'.

Als achter *F* niets is vermeld, geldt de vorige zoektekst. Dat scheelt veel typewerk als u in een tekst meer keren dezelfde zoektekst wilt gebruiken. Ga voordat u verder zoekt met de instructie *N* naar de volgende regel. Als u dat vergeet, vindt EDIT de zoektekst in de actuele regel.

Tekst wijzigen

Als u de goede regel hebt gevonden, kunt u daar niet rechtstreeks iets in veranderen. Kies een geschikte instructie van EDIT om die verandering te laten uitvoeren. Hieronder passen we de meest gebruikte instructies van EDIT toe op het zinnetje 'Een witte olifant is zeer zeldzaam'.

- e* (exchange) Vervangt een gegeven string door een andere string.
Invoer: e/zeer/niet/
Nu staat er: Een witte olifant is niet zeldzaam.

of:
Invoer: e/niet //
Nu staat er: Een witte olifant is zeldzaam.

a (after) Voegt achter de gegeven string de vermelde tekst toe.
Invoer: a/Een /oude /
Nu staat er: Een oude witte olifant is zeldzaam.

b (before) Voegt een tekst toe voor de gegeven string.
Invoer: b/zeldzaam/bijzonder /
Nu staat er: Een oude witte olifant is bijzonder zeldzaam.

d (delete) De actuele regel wordt helemaal gewist. De volgende regels worden niet opnieuw genummerd. Behalve de actuele regel kunt u ook een of meer andere regels wissen. Schrijf achter *D* een of twee getallen (in het laatste geval wist u een heel blok). Voorbeeld:

d 10 100

Deze opdracht wist de regels 10 tot en met 100.

i (insert) De instructie *I* brengt EDIT in de invoegmodus. De regels die u daarna invoert, voegt EDIT toe voor de actuele regel. Sluit de invoer van nieuwe regels af met de instructie *Z*. De regels na de actuele regel schuiven op. De originele tekst is bijvoorbeeld:

20. "Bij gebrek aan werk"
21. "slaat de verving toe."

Neem aan dat regel 21 de actuele regel is, en voer de volgende regels in:

i
heeft menigeen weinig te besteden.
Het meeste amusement is niet gratis, dus
z

Het resultaat ziet er als volgt uit:

20. "Bij gebrek aan werk"
21. "heeft menigeen weinig te besteden."
22. "Het meeste amusement is niet gratis, dus"
23. "slaat de verving toe."

Werken met meer dan één bestand

Op ieder moment kunt u, naast het bij de start van EDIT geopende bestand, een of meer andere bronbestanden openen met de opdracht:

from .bestandsnaam.

Na deze invoer worden nieuwe regels alleen nog uit het bestand 'bestandsnaam' gelezen. Ook in het nieuw geopende bestand begint EDIT de invoer met de eerste regel van de tekst.

Zonder argument achter FROM springt u terug naar het bij de start van EDIT opgegeven bronbestand. Kanalen naar andere bestanden blijven bestaan en EDIT onthoudt van alle aanvullende bestanden wat daar de laatste geactiveerde regel was.

Als u uit een aanvullend bestand een regel hebt gelezen, kunt u die regel opnieuw laden als u dat bestand eerst hebt afgesloten (close file) en het vervolgens opnieuw hebt geopend. Daarvoor gelden respectievelijk de opdrachten:

cf .bestandsnaam.

en

from .bestandsnaam.

Macro's

EDIT kan zijn opdrachten niet alleen krijgen via het toetsenbord, maar ook uit een opdrachtbestand (macro). Schrijf bij de start van EDIT de naam van die macro achter de optie WITH. U kunt ook later de naam van de macro invoeren achter de instructie C. Het voorbeeld geeft een idee van het gemak dat een macro u kan bieden. Neem aan dat u een macro hebt gemaakt met de naam 'macro':

```
i
*****
* Programma           :                               *
* Auteur              :                               *
* Datum               :                               *
* Taal                : C                             *
* Assembler/compiler  : Aztec C68/am-c V3.4          *
*****
z
```

Met de opdracht:

c .macro.

voegt u alle regels tussen de instructies I en Z toe vóór de actuele regel van het bronbestand.

De instructie C is eigenlijk geen routine, maar een sprongopdracht voor EDIT. De editor gaat verder met het bestand 'macro' na de instructie I en springt naar aanleiding van de instructie Z terug naar de actuele regel van EDIT. Zodra het einde van 'macro' is bereikt of als daar de instructie Q staat, gaat de besturing terug naar een laag die één niveau hoger ligt. Dat hoeft niet per se het toetsenbord te zijn, want macro's mogen worden genest (in elkaar geschoven). Een geneste macro bevat dus zelf weer de instructie 'c'.

Als u eenmaal de slag te pakken hebt, kunt u met macro's in EDIT sneller werken dan met ED.

EDIT verlaten

De standaardmanier om EDIT te verlaten maakt gebruik van de instructie *W*. Eerst maakt EDIT nog een kopie van zijn buffer en de rest van het bronbestand in het doelbestand. Bovendien wordt een kopie van het oorspronkelijke bronbestand geboekt onder de naam 'edit-backup' in de subdirectory 'T'. Als u bij de oproep van EDIT geen naam hebt vermeld voor het doelbestand, kent EDIT de naam van het bronbestand toe aan het hulpbestand in de subdirectory 'T'. Dat heeft immers gefungeerd als tijdelijk doelbestand. Ten slotte worden de nog openstaande kanalen naar aanvullende bestanden en macro's afgesloten.

Het is ook mogelijk EDIT te verlaten via de instructie *STOP*. EDIT maakt dan geen kopieën. Het doelbestand is dus niet compleet. De consequentie daarvan is ook dat het hulpbestand geen nieuwe naam krijgt en dat het bronbestand zijn oude naam behoudt. Als u op deze manier EDIT verlaat, hebt u uw werk voor niets gedaan en blijft alleen het bronbestand over (ongewijzigd en met dezelfde naam).

In paragraaf 8.2 vindt u een korte beschrijving van de hierboven behandelde instructies van EDIT. Voor de volledigheid vermelden we daar bovendien de resterende, minder vaak gebruikte instructies.

3 Tips en trucs voor de CLI

In dit hoofdstuk komen problemen aan de orde die iedereen die met de CLI werkt, vroeger of later zal tegenkomen. Daartussen vindt u een groot aantal wetenswaardigheden over de CLI.

3.1 Invoer en uitvoer in de CLI

Uitvoer van de CLI blokkeert u tijdelijk door een willekeurige toets in te drukken. Bij een lange directory maakt u daar ongetwijfeld gebruik van. De uitvoer komt weer op gang na een druk op Return, Enter of Backspace. De reden dat juist deze toetsen die werking hebben is, dat de uitvoer pas verder kan gaan als de toetsenbordbuffer leeg is. Dat is het geval als u het ingetoste teken aan het systeem doorgeeft met behulp van Return of Enter of als u het wist met Backspace.

Een onderbreking in de uitvoer biedt u dus in feite de gelegenheid een nieuwe opdracht in te voeren terwijl de vorige nog niet is afgehandeld. Het eerste teken van de invoer onderbreekt de uitvoer van de actuele opdracht. Er is dus geen sprake van multitasking: de opdrachten worden na elkaar uitgevoerd en AmigaDOS blijft zonder pauze actief.

Niet elke reeks opdrachten is geschikt voor deze werkwijze. Bij de onderstaande opdrachten is het verstandiger de invoer van de tweede opdracht niet te doen tijdens de uitvoering van de eerste:

```
copy tekst to kopie
delete tekst
```

Het bestand 'tekst' wordt gewist, ook als het maken van de kopie mislukt (bijvoorbeeld omdat de diskette vol is). Dan hebt u noch een origineel, noch een kopie. Aangenomen dat er nog niet in de blokken is geschreven die door 'tekst' werden gebruikt, kan DiskDoctor uitkomst bieden. (DiskDoctor is een normale CLI-instructie.) Het is in ieder geval de moeite waard het te proberen.

3.2 Jokers

De vorige generaties Commodore computers (bijvoorbeeld: PET, VC20, C64) gebruiken de tekens '*' en '?' als jokers bij de invoer van namen. Jokers besparen veel typewerk bij de invoer van opdrachten. In plaats van de reeks namen Test1, Test2 tot en met Testn is het voldoende de aanduiding 'Test*' te vermelden achter de instructie SCRATCH als u die bestanden van een diskette wilt wissen. Gebruik 'Test?' als u alleen bestanden wilt wissen waarvan de naam uit vijf tekens bestaat ('Test1a' blijft buiten schot). U mag ook combinaties van jokers gebruiken:

```
????Versie*
```

In dit geval behoren tot de doelgroep al die bestanden waarvan de naam vanaf de vijfde positie de letterreeks 'Versie' bevat. De volgende voorbeelden voldoen aan die eis:

- TestVersie1

- GeenVersie
- SlotVersieVandaag

De Amiga kent ook jokers. Het vraagteken heeft dezelfde functie als in de hierboven gegeven voorbeelden. In plaats van het sterretje gebruikt de Amiga een hekje (het teken '#', ook wel 'matje' geheten). Het hekje is flexibeler dan het sterretje. In het Angelsaksische taalgebied is dit teken de afkorting van het woord 'number'. Er hoort dus (ten minste) een getal achter. Aan de opdracht 'dir Test#3a' voldoet een naam die eindigt op 'aaa'. De combinatie '#?' vervangt de '*' van oudere Commodore computers. Nieuw is dat u achter die combinatie kunt aangeven aan welke eis het einde van de naam moet voldoen (bijvoorbeeld: #?.info). Hieronder ziet u een lijst met bestandsnamen en daarna een paar namen met jokers. Aan u de taak uit te zoeken welke namen worden uitgevoerd na de zeven DIR-opdrachten.

katje.1	1.hond
katje.2	2.hond
katjesgespin	hondehokken
zwerfkatten	kaffer_hond

1. katje.?
2. k#?
3. ???????????
4. #?
5. honde?????
6. #?hond
7. ?????????#2t#?

De goede oplossingen zijn:

voor 1:	katje.1, katje.2
voor 2:	katje.1, katje.2, katjesgespin, kaffer_hond
voor 3:	hondehokken, zwerfkatten
voor 4:	alle bestanden
voor 5:	geen enkel bestand
voor 6:	1.hond, 2.hond, kaffer_hond
voor 7:	zwerfkatten

Bij de laatste omschrijving vraagt u zich waarschijnlijk af of jokers werkelijk voordelen bieden.

3.3 Onderbrekingen in de CLI

Onderbreking met behulp van het toetsenbord

De Amiga heeft op zijn toetsenbord geen Run/Stop-toets zoals bijvoorbeeld de C64. Als u een CLI-routine onderbreekt, geeft de CLI de besturing terug aan het programma dat die routine opriep. Dat is meestal de CLI zelf, omdat daar de CLI-routine werd opgeroepen met de instructie met dezelfde naam. Als bevestiging van de onderbreking verschijnt in het CLI-venster de melding '*** Break'. De drie sterretjes betekenen dat de onderbreking van het type 'C' is.

De onderbrekingen met Ctrl-D tot en met Ctrl-F zijn iets minder rigoreus en ze hebben dan ook minder sterren. Of de laatst genoemde instructies effect hebben, hangt af van de instructie die ze (proberen) te onderbreken. Voor de nu geïmplementeerde instructies is alleen Ctrl-D van belang (bij SEARCH en EXECUTE).

De instructie SEARCH reageert net als alle andere instructies op Ctrl-C. Het besturingssysteem verlaat de zoekprocedure. Met Ctrl-D verlaat de routine alleen het onderhanden bestand en gaat verder met het volgende bestand. Ctrl-D is dus minder krachtig dan Ctrl-C.

Bij de instructie EXECUTE is de zaak iets ingewikkelder. Deze instructie start de uitvoering van een reeks CLI-opdrachten in een batchbestand.

Ook bij de uitvoering van een CLI-opdracht in een batchbestand is Ctrl-C krachtiger dan Ctrl-D. Ctrl-C breekt de actuele CLI-opdracht meteen af en geeft de besturing terug aan het niveau van EXECUTE. Het hangt af van de onderbroken routine, wat er verder gebeurt. Als die routine vanwege de onderbreking een error-code naar EXECUTE levert (foutnummer ≥ 10), verlaat het besturingssysteem meteen het batchbestand met een passende melding. (Voorbeeld: Ctrl-C tijdens een SEARCH-opdracht.) Als de routine zonder foutmelding terugkeert naar het niveau van EXECUTE, wordt het batchbestand verder gewoon afgehandeld. (Voorbeeld: Ctrl-C tijdens een DIR-opdracht.)

Tijdens de instructie EXECUTE werkt Ctrl-D iets anders dan bij de andere instructies. Deze toetsencombinatie verlaat het batchbestand na beëindiging van de onderhanden routine. De zaak wordt ingewikkeld als de actieve routine zelf al gevoelig is voor Ctrl-D (dat geldt bijvoorbeeld voor SEARCH). Na de melding van zo'n routine is de kous af en komt de volgende opdracht van het batchbestand aan de beurt. Het besturingssysteem gaat niet terug naar EXECUTE. Er volgt alleen maar een sprong uit het batchbestand als dat geen andere opdrachten meer bevat.

Onderbreking door een opdracht

Zelfs een opdracht die geen invoervenster heeft, kunt u beëindigen. Nadat een taak op gang is gebracht met de instructie RUN, kunt u vanuit het oorspronkelijke CLI-venster geen invoer meer doorgeven aan die taak. Het is een niet-interactief proces. In het CLI-venster verschijnt alleen uitvoer van het proces. Een eenvoudig voorbeeld:

```
run dir df0:
```

Na de uitvoer van het nummer dat aan de nieuwe taak is toegekend (bijvoorbeeld: [CLI 2]), verschijnt de inhoudsopgave van de diskette in het actuele CLI-venster. De opdracht reageert niet meer op Ctrl-C.

Er is niettemin een manier om een niet-interactief proces te verlaten. De CLI-instructie BREAK drukt als het ware softwarematig op de Ctrl-C-toets van een taak. De instructie heeft als argumenten het nummer van de taak waar de opdracht heen moet en het type van de onderbreking (c-f). Als u in plaats van het type 'all' opgeeft, gelden alle beschikbare typen. Om de hierboven gedefinieerde taak te onderbreken luidt de opdracht:

```
break 2 c
```

Aarzel niet deze opdracht in te voeren in het venster waarin de uitvoer verschijnt. Zodra u het eerste teken hebt ingetoetst, stopt de uitvoer van de inhoudsopgave tot u Return geeft.

De BREAK-opdracht mag ook vanuit een CLI-venster een opdracht in een ander CLI-venster onderbreken. In dit geval is het nummer van de taak gelijk aan het nummer van het CLI-venster waarin die taak wordt uitgevoerd.

3.4 De standaard-devices van de Amiga

Een device is een apparaat waarmee de computer gegevens kan uitwisselen (bijvoorbeeld een diskdrive). Het gegevenstransport mag eenrichtingsverkeer zijn (printer, muis).

Als u in de CLI de instructie ASSIGN geeft, verschijnt op het scherm een lijst met de beschikbare apparaten. De standaard-devices van elke Amiga zijn:

```
DF0 RAM PAR SER PRT CON RAW
```

De CLI onderscheidt de namen van devices van die van directories of bestanden met behulp van de dubbele punt achter de naam van een device.

3.4.1 Floppydrive, DF

Alle namen van devices die beginnen met DF horen bij een drive voor floppy disks. In totaal kan de Amiga maximaal vier van zulke diskdrives aansturen (DF0 tot en met DF3). Deze code staat net zo vaak in de ASSIGN-lijst als er diskdrives zijn aangesloten. In de basisconfiguratie met één diskdrive gelden alle opdrachten in de CLI automatisch voor de diskdrive met nummer 0, tenzij u expliciet een andere drive definieert. In 2.1 gaan we uitvoerig in op de haken en ogen aan directories in verband met de instructies LIST en CD.

3.4.2 De RAM-disk

Het device met de naam RAM simuleert een diskdrive in het werkgeheugen van de Amiga. RAM is een afkorting van Random Access Memory, een type geheugen voor gebruik naar keuze (lezen en schrijven). In bijna alle situaties gedraagt een RAM-disk zich als een gewone diskdrive. De veel hogere snelheid is het grootste voordeel in vergelijking met een echte drive.

Een RAM-disk heeft ook een paar nadelen. De inhoud van een RAM-disk verdwijnt als de stroom uitvalt of als u een reset uitvoert. U loopt altijd het risico dat ergens een stop doorslaat en dat u de gegevens op de RAM-disk kwijtraakt. Als de computer gaat hangen, is een reset nodig. Het signaal daarvoor is de melding:

```
Guru-Meditation
```

Verklein de gevolgen van die risico's en maak van belangrijke gegevens op de RAM-disk zeer regelmatig een kopie op een echte diskette.

De ruimte die een RAM-disk in het geheugen inneemt, vormt soms een doorslaggevend onderdeel. Een RAM-disk is als het ware van elastiek. Hoe meer gegevens u opslaat, hoe meer ruimte hij van het werkgeheugen afsnoept. Dat is geen probleem als u uw systeemgeheugen hebt uitgebreid. De RAM-disk geeft zelf geen informatie over de hoeveelheid gegevens (het antwoord is altijd: 100 procent bezet). Dat is een gevolg van de opzet van het device.

Werken met de RAM-disk veronderstelt dat hij al is aangemaakt. Het is voldoende de inhoud op te vragen met 'dir ram:'. Deze opdracht staat meestal in de startup-sequence zodat u de RAM-disk meteen kunt gebruiken. Het is niet zo makkelijk een eenmaal aangemaakte RAM-disk weer kwijt te raken. (De instructie DELETERAMDISK bestaat niet.) In paragraaf 3.5 vindt u voorbeelden hoe u met de combinatie van één diskdrive en de RAM-disk het aantal diskwisselingen drastisch kunt verminderen.

De onderstaande instructies mogen wel worden gebruikt met een fysieke diskdrive, maar niet met de RAM-disk. Als u dat toch doet, hebben ze het hieronder vermelde effect. Een uitgebreidere bespreking vindt u in hoofdstuk 2.

AddBuffers Veroorzaakt ongeacht de hoeveelheid beschikbaar geheugen de volgende foutmelding:

Warning
Insufficient memory for buffers

DiskCopy Het is verspilling van ruimte aan een RAM-disk nog een buffer te koppelen. Een buffer als wachtkamer voor de RAM-disk vertraagt bovendien de snelheid. Bestanden en directories op de RAM-disk kunt u alleen stuk voor stuk kopiëren met de instructie COPY. Alle bestanden in een keer met behulp van DISKCOPY is niet mogelijk.

Format De RAM-disk hoeft niet te worden geformatteerd. Als u het toch probeert, krijgt u de volgende foutmelding:

system/format failed -
not a disk

Relabel Dit duurt even, maar de Amiga merkt het wel. Het is niet mogelijk de RAM-disk een naam te geven. Een poging daartoe leidt tot de volgende foutmelding:

Attempt to rename disk failed

Install Het heeft geen zin de RAM-disk te voorzien van boot-bestanden. Op het moment van booten of bij een reset bestaat de RAM-disk immers nog niet. De instructie INSTALL werkt dan ook niet met de RAM-disk.

3.4.3 Het parallele device

Met behulp van de device-naam PAR kunt u een parallele interface van het type Centronics aanspreken. Dat gaat via de aansluiting die op de behuizing van de Amiga de naam *Parallel Port* draagt. In tegenstelling tot eerder behandelde devices moet in dit geval het apparaat nog worden aangesloten. Het apparaat heet parallel omdat alle acht bits van een byte tegelijk worden verstuurd. (Een device dat de bits na elkaar verstuurt bespreken we in de volgende paragraaf.)

De interface is niet gereserveerd voor een bepaald apparaat. Een analoog/digitaal-converter vertaalt de doorlopende (analoge) signalen van spraak, muziek of meetresultaten naar een formaat dat de computer kan hanteren (digitaal). De stroom gegevens kan ook de andere kant

uit gaan (bijvoorbeeld naar een printer). De gegevens gaan van de computer via de interface naar de printer. In paragraaf 3.4.5 bespreken we waarom de computer voor een printer met een parallelle interface meestal een speciale interface gebruikt (prt:).

De snelheid van de transmissie van de gegevens hangt alleen af van de snelheid waarmee het apparaat die gegevens kan verwerken. Behalve de acht datalijnen heeft een Centronics-interface een paar lijnen voor stuursignalen. Die transporteren de gegevens van de *handshake*. Dat stuurprotocol verstuurt bijvoorbeeld van het ontvangende apparaat aan het zendende apparaat de boodschap dat het eerste klaar is om nieuwe gegevens te verwerken. Op die manier bent u verzekerd van de optimale transmissiesnelheid.

3.4.4 Het seriële device

De seriële interface van het type RS232C heet *SER*. De aansluiting daarvan op Amiga, de *Serial Port*, is bruikbaar voor uiteenlopende toepassingen (modem, MIDI, enzovoorts). De naam van dit type interface suggereert al dat de bits van een databyte na elkaar (serieel) worden verstuurd en niet zoals bij PAR: tegelijk (parallel). Per transmissierichting is er dan ook maar één stuurlijn nodig.

De interface heeft een modus waarin maar zeven bits per byte worden verstuurd. De vrijgekomen bit wordt dan gebruikt voor controle op de transmissie (pariteitsbit). U kiest uit een even of een oneven pariteit van de interface. De zendende interface zet de achtste bit zó dat de gezette bits van de verzonden byte een even of een oneven aantal vormen (Even Parity, Odd Parity). Als de ontvanger is geïnstalleerd met dezelfde pariteit, is er een zekere controle mogelijk op fouten in de overdracht van de gegevens.

Een startbit en twee stopbits completeren elke verstuurde eenheid gegevens. Ze dienen als merktekens voor de synchronisatie. De transmissiesnelheid van het zendende en het ontvangende apparaat moet binnen nauwe grenzen gelijk zijn. Deze snelheid wordt uitgedrukt in baud, een eenheid genoemd naar de Franse uitvinder Baudot (1 baud = 1 bit/seconde).

Voor de RS232-interface zijn drie methoden in gebruik om een correcte gegevensoverdracht te garanderen (*handshake*):

xON/xOFF-protocol

Het *xON/xOFF*-protocol is bedoeld voor bidirectionele interfaces. Volgens dit protocol verstuurt een ontvangend apparaat via de retourlijn een speciaal stuurteken (*xOFF*), zodra het geen nieuwe gegevens meer kan verwerken. Het zendende apparaat wacht met transmissie totdat via die lijn het stuurteken 'xON' binnenkomt. Voor dit protocol zijn maar drie lijnen nodig (twee signaallijnen en één lijn voor massa). Men spreekt daarom wel van een 3-draads *handshake*. Het besturingssysteem interpreteert onder dit protocol automatisch de stuurtekens.

Aparte stuurlijnen

Vanwege de extra stuurlijnen lijkt dit protocol een beetje op dat van de *handshake* van een parallelle interface. De lijnen RTS/CTS en DSR/DTR transporteren de gegevens over de toestand in de zender en de ontvanger. Dit protocol heeft geen lijn nodig voor terugmeldingen. Omdat de stuurtekens niet worden verzonden via de tamelijk trage datalijn, werkt deze methode iets sneller dan het *xON/xOFF*-protocol.

Veiligheidsreserves

Als u er absoluut zeker van bent dat de ontvanger de binnenkomende bytes sneller kan verwer-

ken dan de zender ze verstuurt, hebt u geen handshake nodig. Stuurlijnen zijn dan overbodig en voor de gegevensoverdracht hebt u genoeg aan twee draden. Met een wachtlus aan de kant van de zender verschaft u de ontvanger gelegenheid de gegevens zonder kop-staart botsingen te verwerken. Deze methode is geschikt als u af en toe op een simpele manier gegevens wilt versturen van de ene computer naar een andere. Voor regelmatig gebruik is ze niet effectief genoeg. De wachtlopen veroorzaken een te grote verspilling van tijd.

Definieer voordat u de eerste keer de seriële interface gebruikt de hierboven besproken parameters met het programma 'Preferences'. Kies daar in het veld 'Buffer Size' ook de grootte van de ontvangstbuffer. Deze buffer dient als veiligheid voor de opvang van gegevens als het leesprogramma even niet in staat is de binnenkomende gegevens te verwerken. De buffer loopt over als het leesprogramma te traag is en een handshake ontbreekt. De overgestroomde gegevens bent u kwijt.

Van de vele parameters die het besturingssysteem van de Amiga kent voor een seriële interface, is helaas maar een klein aantal te definiëren via 'Preferences'.

Als u de gegevens die via de seriële poort binnenkomen, naar het scherm stuurt, wordt eerst de RS232-invoerbuffer gevuld (minstens 512 bytes). Voer daartoe in de CLI bijvoorbeeld in:

```
copy ser: to con:10/10/300/100/uitvoer
```

Pas als de buffer overloopt, verschijnen de eerste gegevens in het CON-venster. De gegevens die in de buffer staan als de laatste gegevens daar zijn binnengekomen, blijven daar hangen. U krijgt ze pas te zien als er weer een flink groot 'gegevenspakket' binnenkomt. Een stuursignaal voor het einde van het bestand wordt blijkbaar niet ondersteund: een misser van Commodore.

De communicatie tussen twee computers via de RS232-interface beschrijven we in paragraaf 3.8.

3.4.5 Het printer-device

De Amiga heeft een aparte interface voor uitvoer naar een printer: het printer-device. Als u in een programma een printer wilt aansturen, hoeft u zich niet af te vragen of de printer is aangesloten op de seriële of de parallele interface. In het programma gebruikt u het besturingsformaat van de Amiga. Bij oproep van het device 'prt', gelden de gegevens die zijn opgeslagen in 'Preferences'. Kies daar een voor uw printer geschikte driver. Die vertaalt het besturingsformaat van de Amiga in codes die de printer begrijpt.

In de praktijk blijken de kant en klare drivers een bron van ergernis te zijn. Helaas is er geen slim programma in de trant van Preference dat de mogelijkheid biedt voor uw printer een driver op maat te maken.

3.4.6 Het CON-device

Het toetsenbord en het beeldscherm zijn apparaten voor in- en uitvoer die zich niet principieel onderscheiden van andere devices. U kunt ze dus op dezelfde manier aanspreken. Er zijn aparte vensters voor invoer en uitvoer. De oproep van het console-device heeft het volgende formaat:

```
con:x/y/breedte/hoogte/naam
```

De argumenten achter 'CON:' betekenen:

- x/y Coördinaten van de linker bovenhoek van het venster.
- breedte Breedte van het venster uitgedrukt in pixels.
- hoogte Hoogte van het venster uitgedrukt in pixels.
- naam Willekeurige naam. Deze verschijnt boven in het venster.

Voorbeeld 1:

```
dir >con:10/10/300/100/testvenster
```

De weergave van de directory verschijnt in een venster met de vermelde afmeting en naam. Meteen na afloop van de opdracht verdwijnt het venster.

Voorbeeld 2:

```
copy con:10/10/300/100/invoer con:10/150/300/100/uitvoer
```

Op het scherm ontstaan tegelijk twee CON-vensters. Wat u in het bovenste venster invoert, verschijnt na de Return ook in het onderste venster. Beide vensters verdwijnen door de toetscombinatie Ctrl-\ (Ctrl-backslash).

3.4.7 Het RAW-device

Het RAW-device vertoont veel overeenkomst met het console-device. U spreekt het ook op een analoge manier aan. Het verschil blijkt als iets invoert in een venster van RAW. De tekens worden niet zichtbaar. Maak nu twee vensters:

```
copy raw:10/10/300/100/invoer con:10/150/300/100/uitvoer
```

Als u in het RAW-venster (bovenaan) zo maar wat tekens typt, verschijnen ze meteen ook in het onderste venster (Return is niet nodig). Door een Return gaat de cursor naar het begin van de regel.

Alle stuurtekens worden ook doorgestuurd (cursor, Delete en Backspace, enzovoorts). De ontvanger (hier het CON-device) moet ze zelf correct interpreteren. In bovenstaand voorbeeld werken alleen de cursortoetsen op de gewone manier. Verlaat de opdracht met Ctrl-C.

Als u geen uitvoer ziet, hebt u waarschijnlijk eerst in het onderste venster iets ingevoerd. Dat onderdrukt, net als bij een CLI-venster, de uitvoer van andere gegevens. Het probleem is opgelost als u in het onderste venster een Return geeft.

3.5 De RAM-disk en de CLI

Vooraf bezitters van een Amiga met maar één diskdrive kunnen veel profijt hebben van een RAM-disk. Maar ook als u meer diskdrives hebt, is het de moeite waarde een RAM-disk te installeren.

De eerste CLI-frustratie loopt u waarschijnlijk op als u de Workbench-diskette vervangt

door een andere diskette om daarvan met behulp van DIR de inhoud op te vragen. Dat komt u te staan op een requester met het verzoek de Workbench-diskette in een drive naar keuze (!) te stoppen. Als u dat verzoek niet opvolgt en het veld 'Cancel' activeert, reageert de Amiga met de mededeling dat de instructie DIR niet bestaat. Als u dan alsnog de Workbench-diskette in de drive stopt, krijgt u de inhoud van de Workbench op het scherm. De opdracht 'dir df0:' heeft hetzelfde resultaat.

In hoofdstuk 1 werd al duidelijk dat een CLI-instructie een klein programma is dat moet worden geladen als u die instructie voor de eerste keer gebruikt. Gewoonlijk staan die routines op de diskette waarmee u het systeem hebt geboot. Op de Workbench-diskette staan de instructies in de directory 'C'. Het besturingssysteem van de Amiga slaat bij de start op welke diskette de systeemdiskette is en in welke directory van die diskette de CLI-instructies staan. (De Amiga kan die diskette altijd aanspreken met de naam 'SYS:')

Als u een CLI-instructie invoert, wil het besturingssysteem de routine van die instructie laden en het vraagt u zo nodig de systeemdiskette te plaatsen. In het hierboven gegeven voorbeeld is de fout gemaakt in de DIR-opdracht niet de goede naam te vermelden van de diskette waarvan u de inhoud wilt zien. 'df0:' is alleen maar de aanduiding van de diskdrive. AmigaDOS voert de DIR-instructie dus uit met de diskette die in drive 0 zit (Workbench).

Met de opdracht 'dir spelen:' verloopt alles naar wens. AmigaDOS vraagt eerst de Workbench-diskette om de routine DIR te laden, vervolgens de diskette met de naam 'spelen' en voert ten slotte de inhoud daarvan uit.

Gelukkig is er een eenvoudiger manier. Doe de Workbench-diskette in de drive en voer de volgende drie regels in:

```
makedir ram:c
copy from df0:c/dir to ram:c
path ram:c add
```

In het vervolg kunt u van een willekeurige diskette met 'dir df0:' de inhoud opvragen. De routine DIR wordt vliegensvlug geladen vanaf de RAM-disk. De opdrachten hebben het volgende effect:

- Regel 1: Maakt de directory 'C' aan op de RAM-disk.
- Regel 2: Kopieert de instructie DIR van de diskette (Workbench) naar die directory. Voeg zo nodig soortgelijke regels toe voor andere CLI-instructies.
- Regel 3: Deelt aan AmigaDOS mee dat het eerst naar de gevraagde routine langs het vermelde pad moet zoeken (directory 'C' van de RAM-disk) alvorens om de Workbench-diskette te vragen.

Behalve met PATH kunt u AmigaDOS op nog een manier opdragen CLI-instructies op te halen uit de RAM-disk. Met de instructie ASSIGN kunt u vooraf nieuwe paden definiëren naar directories met CLI-instructies. Bekijk de lijst die de Amiga uitvoert als u de instructie ASSIGN invoert zonder meer. Onder het kopje 'Directories:' staat onder andere de boeking 'C'. Rechts daarnaast staat een pad dat is toegekend aan de code 'C' (bijvoorbeeld: A500 WB1.2:c). AmigaDOS zoekt altijd in de daar vermelde directory naar de CLI-instructies. Bij het booten van de computer komt daar automatisch de directory C van de systeemdiskette te staan. Verander dat met de instructie ASSIGN:

```
makedir RAM:c
```

assign C: RAM:c

In het vervolg kent uw Amiga alleen de instructies die zich bevinden op de RAM-disk in de directory 'C'. Kopieer dus meteen na de MAKEDIR-opdracht alle CLI-instructies naar die directory. U houdt de beschikking over de CLI-instructies op de systeemdiskette in de drive als u een compleet pad daarheen definieert:

```
sys:c/assign c: sys:c
```

Deze opdracht maakt de toewijzing naar de RAM-disk ongedaan. (Anders kunt u geen routines van de CLI-instructies kopiëren.) Met de opdracht 'Sys:' spreekt u automatisch de systeemdiskette aan. Zo nodig verschijnt er een requester met het verzoek die diskette in de drive te doen.

De volgende regels kopiëren de meest gebruikte CLI-instructies naar de RAM-disk:

```
makedir ram:c ;(voor de zekerheid!)
copy c/assign ram:c
copy c/dir ram:c
copy c/cd ram:c
copy c/copy ram:c
copy c/delete ram:c
copy c/makedir ram:c
copy c/rename ram:c
copy c/newcli ram:c
copy c/endcli ram:c
copy c/run ram:c
copy c/list ram:c
assign c: ram:c
```

We zullen u niet verhinderen meer CLI-instructies naar de RAM-disk te kopiëren, maar het is verspilling van ruimte ook instructies als SETCLOCK of DATE mee te nemen. Als u een Amiga hebt met 'maar' 500 Kbyte, loopt u het risico in grote ruimteproblemen te komen.

Als het een keer voorkomt dat u een instructie nodig hebt die niet op RAM-disk staat, hebt u de keus uit een algemene aanpak en met ASSIGN terugschakelen naar de systeemdiskette (assign c: sys:c), of een specifieke aanpak en onder vermelding van een complete padnaam de ontbrekende instructie oproepen (bijvoorbeeld: sys:c/date 29-oct-87 11:30:35).

In hoofdstuk 4 vindt u een groot aantal tips waarmee u de invoer van bovenstaande lijst opdrachten kunt automatiseren.

Kopiëren met behulp van de RAM-disk

Het is geen kunst met twee diskdrives een bestand of zelfs een hele directory te kopiëren:

```
copy df0:utilities/notepad df1:hulpstukken
```

Met één drive kunt u de methode gebruiken waarin u de namen van de diskettes vermeldt in plaats van de drive(s). Bijvoorbeeld:

copy from textomat:brieven/peter to teksten:brieven

AmigaDOS vraagt automatisch afwisselend naar de diskettes met de opgegeven namen en let niet meer op de nummers van de drive(s).

Deze methode heeft twee grote nadelen. U moet weten hoe de twee diskettes heten, en zelfs voor een bestand van maar een paar bytes moet u minstens zeven keer van diskette wisselen. Voor serieuze toepassingen kunt u die methode dus wel vergeten.

Het werkt veel prettiger als u de RAM-disk gebruikt als tijdelijk onderkomen van de bestanden die u wilt kopiëren. Het is niet moeilijk bijvoorbeeld een directory met de naam 'C' te kopiëren naar een andere diskette:

- Doe de diskette met de bronbestanden in de drive.
- Maak met COPY een kopie van de bestanden naar de RAM-disk (copy df0:c ram:).
- Leg de doeldiskette in de drive.
- Maak een directory op de doeldiskette (mkdir df0:c).
- Kopieer de bestanden van de RAM-disk naar de doeldiskette (copy ram: df0:c).

Voorwaarde voor het slagen van deze procedure is dat de gebruikte instructies naar de RAM-disk zijn gekopieerd en dat het systeem daarvan op de hoogte is gebracht met ASSIGN of PATH.

Wees zuinig met geheugenruimte en wis na afloop de op de RAM-disk geparkeerde bestanden:

```
delete ram:c all
```

Deze opdracht wist de hele directory 'C', die heeft gediend als tijdelijke opslag van de bestanden op de RAM-disk.

3.6 Printen vanuit de CLI

Met de Workbench kunt u niet veel beginnen als u wilt printen. We willen weliswaar niet beweren dat u wel direct probleemloos kunt printen via de CLI (we weten wel beter), maar voor veel problemen vindt u oplossingen in de volgende paragrafen.

3.6.1 Bestanden printen met COPY

De instructie COPY kopieert in de meest ruime betekenis van het woord. Voor de volledigheid geven we nog eens de syntaxis:

```
copy from,to/a,all/s,quiet/s
```

FROM en TO vermelden de bron en het doel van het proces. De optie 'ALL' spreekt alle bestanden van de bron directory aan. In de QUIET-modus verschijnen er geen meldingen over de procedure op het scherm. Elk device dat gegevens kan leveren of opnemen, kan in de instructie

COPY dienen als bron respectievelijk als doel. (Een printer als bron is natuurlijk onmogelijk.)

Vraag met de instructie ASSIGN op welke apparaten zijn geïnstalleerd op uw Amiga. In de uitvoer staat bijvoorbeeld onder het kopje 'Devices':

```
DF0 DF1 PRT PAR SER RAW CON RAM
```

Uw printer staat daar vermeld als 'PRT'.

Geef voor een snelle kopie van een tekstbestand de opdracht 'copy bestandsnaam to prt:'.

Het formaat van de afdruk hangt af van de waarden van de parameters die zijn opgeslagen in het programma 'Preferences'. De invloed van de Workbench doet zich dus nog gelden in de CLI: PRT is serieel als dat zo is gedefinieerd in 'Preferences'. In een programma dat gegevens naar de printer stuurt, hoeft u zich geen zorgen te maken over het type interface, regelbreedte of papierlengte. De gekozen driver interpreteert het bestand en voegt automatisch de stuurtekens voor de printer toe.

Met de niet-specifieke drivers PAR en SER kunt u dat automatisme omzeilen voor speciale toepassingen. In paragraaf 3.6.3 gaan we daar nader op in.

3.6.2 Algemene opmerkingen over omleiden van de uitvoer

Na lezing van de vorige paragraaf denkt u misschien dat het nodig is de uitvoer van een programma eerst in een bestand te schrijven als u die uitvoer op de printer wilt hebben (met nog een tussenstap in de vorm van een COPY-opdracht van bestand naar printer). Zo ingewikkeld is het echter niet.

Alle uitvoer van AmigaDOS-instructies kunt u direct vanaf de instructieregel omleiden naar een willekeurig apparaat. Uit praktische overwegingen geldt bij verstek het scherm. Het is geen enkel probleem met behulp van '>SER:' achter de instructie DIR de inhoudsopgave van uw Workbench-diskette via een modem naar Tokio te versturen. Geef een omleiding aan met een punthaak naar rechts. Van alle opties van een instructie staat dit omleidingssymbool altijd vooraan. Meteen daarachter staat het doel-device. Een paar voorbeelden:

- | | |
|-----------------------------------|---|
| dir >prt: | Voert de inhoud van de actuele directory uit via de printer. |
| dir >prt: opt a | Voert de inhoud van alle directories van de actuele drive uit via de printer. |
| dir >prt: ram: opt a | Voert de inhoud van alle directories van de RAM-disk uit via de printer. |
| echo >prt: "Dag Sinterklaas" | De tekst 'Dag Sinterklaas' wordt geprint. |
| type >prt: df0:s/startup-sequence | De startup-sequence wordt geprint. |

3.6.3 Stuurtekens voor een printer

Heel veel opties van uw printer vindt u niet terug in het programma 'Preferences' op de Workbench-diskette. Het lijkt wel of double strike en andere tekensets niet bestaan voor de Amiga. Onder AmigaBASIC is er geen probleem (zie 'Printeropties' in hoofdstuk 5 van het handboek bij AmigaBASIC). De functie CHR\$ waarmee BASIC stuurtekens naar de randapparatuur stuurt, heeft echter geen equivalent in de CLI.

De problemen met stuurtekens ontstaan doordat ze niet met één toets zijn in te voeren via het toetsenbord. Enkele stuurtekens kunt u geven door gelijktijdig indrukken van Ctrl en een andere toets. De combinatie Ctrl-O levert de stuurcode '15' op ('O' is de vijftiende letter van het alfabet). Ctrl-O brengt Epson-printers in de condensed-modus. Op analoge manier kunt u terugrekenen welke toetsencombinatie nodig is voor een andere stuurcode. Het enige nadeel van deze manier van invoer is, dat er niets van is te zien op het scherm. Dat doet er weliswaar niet toe, want de uitvoer is toch bestemd voor de printer (zie de vorige paragraaf).

De omleiding naar de printer kunt u het beste toepassen op de instructie ECHO. Die is speciaal bedoeld voor schermuitvoer van tekst. Een voorbeeld:

```
echo >prt: [Ctrl-0]
```

De rechte haken betekenen dat niet de tekst, maar de vermelde toetsencombinatie moet worden ingevoerd. De spatie voor de linker haak voorkomt dat de tekst wordt opgevat als een directory in de printer. Tot nader order drukt de printer verder alles af in de modus condensed (herroepen met [Ctrl-T]).

In het hierboven beschreven systeem vormt de combinatie Ctrl-C een probleem, omdat die in de CLI een instructie beëindigt. Het is dus niet mogelijk die toetsencombinatie te versturen vanuit de CLI.

Met een truc laten we de computer die combinatie ergens anders ophalen. Maak onder AmigaBASIC op de diskette een bestandje dat alleen een Ctrl-C bevat in de vorm van de opdracht CHR\$(3) (immers: A = 1, B = 2, C = 3, enzovoorts). Het volgende programmaatje doet dat:

```
OPEN "df0:CTRL-C" FOR OUTPUT AS #1
PRINT#1,CHR$(3);
CLOSE #1
```

Verstuur een Ctrl-C met de opdracht:

```
COPY FROM df0:CTRL-C TO PRT:
```

of met:

```
TYPE >prt: df0:CTRL-C
```

Een groot aantal stuuropdrachten zijn equivalent met een 'Escape-sequentie'. Dat zijn reeksen stuurtekens die worden voorafgegaan door een Esc-teken. De tekens van zo'n reeks moeten na elkaar naar de printer worden verstuurd. Afhankelijk van de aard van de opdracht volgt op Esc minstens een combinatie van Ctrl met een ander teken. Het teken Esc is wel op het toetsenbord te vinden (helemaal boven links). Als voorbeeld geven we een echte Epson-opdracht. In

AmigaBASIC ziet hij er als volgt uit:

```
PRINT#1,CHR$(27);"R";CHR$(9);
```

CHR\$(9) activeert de Noorse tekenset. Het equivalent in de CLI is:

```
echo >prt: [ESC]R[Ctrl-I]
```

Achter de dubbele punt hoort een spatie; voor en achter de 'R' mag er juist geen staan.

Ook hier duiken problemen op. De Amerikaanse tekenset activeert u bijvoorbeeld in AmigaBasic met CHR\$(0). Het alfabet heeft echter geen letter met het nummer nul. Invoer via het toetsenbord als een combinatie met Ctrl is dus niet mogelijk. De oplossing is ook hier een bestandje dat de benodigde tekenreeks bevat:

```
OPEN "df0:USA-set" FOR OUTPUT AS #1
PRINT#1,CHR$(27);"R";CHR$(0);
CLOSE #1
```

Start de computer, maak onder AmigaBASIC bovenstaand bestand en kopieer het naar de CLI-werkdiskette. De volgende invoer activeert de nieuwe tekenset:

```
COPY USA-set prt:
```

of:

```
TYPE >prt: df0:USA-set
```

3.7 Toepassingen van het CON-device

Deze paragraaf geeft maar een paar voorbeelden van de vele mogelijkheden met het console-device. Hopelijk stimuleren ze u tot verder experimenteren.

Aparte vensters voor uitvoer van de CLI

Alle uitvoer die door CLI-opdrachten wordt veroorzaakt in het actuele venster, kunt u omleiden naar een ander apparaat. In sommige gevallen is het beter die uitvoer niet naar het CLI-venster te sturen, maar naar een eigen venster.

Neem aan dat u de inhoud van een diskette wilt laten staan terwijl u met de CLI verder werkt. Het is niet nodig daarvoor een nieuw CLI-venster te openen met NEWCLI. De volgende opdracht stuurt de inhoudsopgave van de diskette in drive 0 naar een eigen venster:

```
run dir >con:10/10/400/100/Inhoud df0:
```

U kunt de uitvoer in het venster 'Inhoud' op ieder moment onderbreken. Activeer het met de muis en druk op een willekeurige toets. Door een Return gaat de uitvoer verder. Na afloop van de opdracht verdwijnt het venster automatisch.

De printer als schrijfmachine

U hebt maar een simpele CLI-opdracht nodig om uw printer te veranderen in een luxe schrijfmachine:

```
copy * to prt:
```

Het sterretje verwijst naar het actuele CLI-venster als bron. De invoer gaat pas naar de printer na een Return. Daarvoor hebt u dus alle gelegenheid met de Backspace-toets nog correcties aan te brengen in de regel. Zoals altijd wist de instructie Ctrl-X de hele regel in de CLI. De instructie Ctrl-\ brengt u terug naar de CLI-prompt.

Tekstbestanden maken

Buiten een editor of tekstverwerker om maakt u met de volgende opdracht direct een tekstbestand:

```
copy * to df0:tekst
```

Het verschil met het vorige voorbeeld is slechts dat de tekens niet naar de printer gaan, maar naar de diskdrive. Het bestand 'tekst' wordt op een diskette opgeslagen. Kleine bestanden die geen blokoperaties vereisen, zijn op deze manier snel te maken. (Laden van de nogal omvangrijke programma's ED of EDIT is niet nodig.) Met de instructie Ctrl-\ beëindigt u ook deze opdracht.

Tekstbestanden uitbreiden

Een kleine uitbreiding voor of achter een tekstbestand is mogelijk met behulp van de instructie JOIN. Waarschijnlijk schrijft u eerst een werkend programma en daarna pas de koptekst met een toelichting. Dat gaat als volgt:

```
join con:10/10/400/100/invoer df0:tekst as df0:teksttotaal
```

De instructie JOIN wordt meestal geïntroduceerd met voorbeelden die diskdrives als bron en doel gebruiken. De syntaxis laat echter ook andere devices toe.

Na invoer van de opdracht verschijnt een venster met de vermelde afmetingen, positie en naam. Voer daar de nieuwe tekst in. Met de instructie Ctrl-\ sluit u nu niet alleen de invoer af, maar de JOIN-opdracht slaat het gecombineerde resultaat ook op onder de opgegeven naam.

Voor een aanvulling aan het eind van een bestand, hoeft u alleen de eerste twee parameters te verwisselen:

```
join df0:tekst con:10/10/400/100/invoer as df0:teksttotaal
```

Een CLI-wekker

Op de Workbench-diskette staat een klok waarmee u een alarmtijdstip kunt programmeren. Die klok heeft een paar nadelen:

- Het kost nogal wat werk om hem op te zoeken.
- Hij gebruikt veel ruimte in het geheugen.
- Hij is alleen geschikt voor absolute alarmtijden.
- U kunt maar één tijd tegelijk programmeren.

In de CLI hebt u die klok niet nodig. De volgende twee regels demonstreren het principe:

```
run wait 10 min +           <Return>
echo "De koffie is nu wel klaar." <Return>
```

De uitvoer verschijnt in het CLI-venster waar u de opdracht geeft. Zorg ervoor dat die CLI niet wordt afgedekt door andere vensters, of laat alle geactiveerde schermen even opflitsen met Ctrl-G aan het eind van de boodschap. Op het moment dat u Ctrl-G invoert ziet u een flits, maar de instructie is niet leesbaar. Ze staat desondanks wel in de tekst. Met de instructie WAIT kunt u ook een tijdstip voor een alarm programmeren (zie paragraaf 2.3.8).

Een CLI-klok is een apart proces (taak). In theorie kunt u net zo veel klokken laten lopen als er taken vrij zijn (maximaal 20). Zelfs in die extreme situatie merkt u in de CLI nauwelijks iets van de afname van de snelheid van de Amiga.

ASCII-codes

Bij alle cijfers, letters en andere tekens van het toetsenbord hoort een ASCII-code. Als u in het vervolg zo'n code wilt weten, hoeft u die niet meer op te zoeken in een boek. Voer via de CLI de volgende opdracht in:

```
type con:300/10/150/50/Converter to * opt h
```

Rechtsboven op het scherm verschijnt een klein venster. Typ daar de tekens waarvan u de ASCII-code nodig hebt. De invoer moet altijd uit zestien tekens bestaan. Het is het handigste dat te doen met de Return-toets. Neem aan dat u de codes van de toetsen 'S' en 'J' wilt weten. Na zestien aanslagen verschijnt het resultaat automatisch in in hexadecimale vorm in het CLI-venster (u hoeft dus niet tot veertien te tellen). Omdat de 'Converter' hexadecimaal werkt, bevat de opdracht de optie 'opt h'. De gezochte ASCII-codes blijken respectievelijk \$73 en \$6A te zijn. De overige posities zijn opgevuld met de code van de Return-toets (\$0A). U verlaat de omrekenfunctie met de instructie Ctrl-\
>

3.8 Gebruik van de seriële interface

De mogelijkheid van multitasking met de Amiga zijn algemeen bekend. Het begrip 'multi-user' gaat minder vaak over tafel. Men bedoelt daarmee dat een aantal terminals gekoppeld zijn aan een centrale, grote computer. Een terminal is soms alleen maar een schermstation zonder

rekencapaciteit. Het geeft de codes van het toetsenbord rechtstreeks door naar de centrale computer en geeft de uitvoer daarvan weer op het scherm. Wij hebben geëxperimenteerd met een Amiga als centrale eenheid, omdat die een aantal taken tegelijk onderhanden kan hebben.

Elke kleine computer met een geschikte interface kan de rol van terminal vervullen (VC20, C64, enzovoorts). Omdat de Amiga maar een seriële interface kan adresseren, is het zonder uitgebreide hardware en software niet mogelijk meer dan één terminal aan te sluiten.

We hebben een Amiga 500 in de standaardconfiguratie gebruikt als terminal en een Amiga 2000 met harddisk en twee diskdrives als centrale eenheid. De verbindingkabel bestaat uit een 7-aderige kabel met twee 25-polige sub-D stekkers. We hebben hem zelf gesoldeerd omdat zo'n kabel niet te koop is. Seriële datatransmissie is niet erg gevoelig voor storende invloeden zodat de lengte er niet erg op aankomt. De configuratie van de pennen is als volgt:

stekker 1 pen	stekker 2 pen	functie
2	3	TXD->RXD (zender->ontvanger)
3	2	RXD<-TXD (ontvanger<-zender)
4	5	RTS->CTS (handshake ene richting)
5	4	CTS<-RTS (handshake andere richting)
6	20	DTR->DSR (controle)
20	6	DSR<-DTR (controle)
7	7	GND (massa)

Waarschuwing: Schakel beide computers uit voordat u de verbinding maakt. Als u dat nalaat, vernielt u de 8520-chips en de daaraan gekoppelde RS232-drivers.

De configuratie is helemaal symmetrisch. Het maakt dus niet uit welke stekker u met welke computer verbindt. Let er bij aankoop van de stekkers op of hun schroeven wel in de Amiga passen. Goedkoop is meestal duurkoop. Stekkers zoals die aan de monitorkabel passen goed en zijn ook geschikt om een Atari computer aan te sluiten als terminal.

De kabel ondersteunt zowel het 3-draads protocol (xON, xOFF) als de RTS/CTS-handshake. Voor de eerste pogingen in de CLI hebben we met 'Preferences' de parameters als volgt gekozen en opgeslagen:

Baud Rate	9600
Buffer Size	512
Read Bits	8
Write Bits	8
Stop Bits	2
Parity	None
Handshaking	RTS/CTS

Op de Extras-diskette die bij de Amiga wordt geleverd, staat een terminalprogramma. Als u dat op beide computers start, verschijnt op de eerste regel de melding 'Modem not ready'. Dat ligt waarschijnlijk aan het ontbreken van een signaal op pen 8 (Carrier Detect). Dat signaal is actief als de ene modem een draaggolf van de ander ontvangt. Die piepton ontbreekt bij het experiment.

Bij de tweede poging gaven we de Amiga 2000 via de CLI de opdracht:

```
run copy ser: to con:10/10/300/100/Ontvangst
```

Alle ontvangen gegevens worden omgeleid naar het opgegeven venster. Met de Amiga 500 stuurden we de actuele inhoudsopgave naar de seriële interface:

```
dir >ser:
```

Dat lukt niet meteen. Pas na een paar herhalingen is de invoerbuffer vol en worden de gegevens door de Amiga 2000 verder verwerkt. Het laatste gedeelte van de gegevens blijft in de invoerbuffer hangen omdat de interface de code voor het einde van het bestand niet herkent. Deze fout in de seriële interface bespraken we al in paragraaf 3.4.4. Het resultaat bewijst dat de transmissie als zodanig is gelukt. Aan de fout in de interface is op dit moment weinig te verhalen.

Vanuit de centrale eenheid gezien (Amiga 2000), is een multi-user configuratie op te vatten als multitasking. Definieer voor de terminal een taak met de instructie NEWCLI. Eén optie van deze instructie opent een CON-venster. Via dat venster geeft u opdrachten en via dat venster verschijnt ook de uitvoer van de taak. De truc is eenvoudig: voer de NEWCLI-opdracht helemaal uit via de seriële interface. Dat gaat als volgt:

```
run newcli ser:
```

Als het probleem met de invoerbuffer is opgelost, kan de Amiga 2000 via de RS232-interface een terminal besturen.

Het is helaas niet eenvoudig vanuit de CLI een echte terminal te simuleren. In plaats daarvan hebben we daarom een klein batchbestand van de A500 verstuurd naar de A2000. Als dat bestand lang genoeg is, komen de eerste opdrachten ervan wel door de invoerbuffer. Omdat het batchbestand de opdracht 'dir jh0:' bevatte, kwam de harddisk op gang. Uit de uitvoer van de instructie STATUS bleek dat de opdracht ook werkelijk werd uitgevoerd. Een ervaren programmeur in de taal C zal er ongetwijfeld in slagen de dwarsliggende interface te temmen en een terminalprogrammaatje te schrijven. De eerste stappen zijn in ieder geval nu gezet.

Voor de zelfgemaakte kabel hebben we nog een toepassing in petto. Roep de Workbench eens op met de volgende uitbreiding:

```
loadwb -debug;
```

Na de start van de Workbench ziet het scherm er normaal uit. Sleep met de muis de menubalk over de rechterraand van het scherm. Op een gegeven moment verschijnt een menu met twee items: Debug en Flushlibs. Vanuit een tweede computer kunt u de 'ROM-Wack' oproepen bij een blokkade (Task Held) of een Guru-meditatie. (ROM-Wack is een soort diagnoseprogramma.)

Het BASIC-programma 'Communic' in de directory Basic Demos van de Extras-diskette is al voldoende als terminalprogramma. Druk op de rechter muisknop als de computer gaat hangen (Press right button to cancel/debug). Daarna kunt u met de aangesloten computer met het terminalprogramma het geheugen van de hangende computer onderzoeken. Het menupunt 'Debug' activeert 'ROM-Wack' ook als de computer niet hangt. Het menupunt 'Flushlibs' heeft meestal geen zichtbaar effect. De functie ervan hebben we niet kunnen achterhalen.

Ten slotte bespreken we een toepassing van de 'ROM-Wack'.

Neem aan dat computer A en computer B met elkaar zijn verbonden via de RS232-kabel. Onderstaande opdrachten brengt computer B in de toestand 'Task Held':

```
mount res0:  
dir >res0:
```

Het device RES0: staat meestal voor demonstraties in het bestand 'MountList' in de directory 'Devs'. Als u dit device echt installeert in de Amiga en er uitvoer heen stuurt, komt de computer gewoonlijk in de toestand 'Task Held'. Als u dan op de rechter muisknop drukt, krijgt u een Guru-meditatie. Na de volgende druk op deze toets verschijnt op de terminalcomputer de volgende uitvoer:

```
rom-wack  
PC: FE66E8 SR: 0015 USP: 022B32 SSP: 07FFF2 XCPT: 0003 TASK: 0221A0  
DR: 0000FFFF 0000FFFF 0001C815 00000001 00000000 00022394 000221FC 00000019  
AR: 000231B5 ABABABAB 00023090 00022394 00FF4834 00005D4A 000045B8  
SF: 2271 ABAB ABDD 2269 0015 00FE 66E8 0000 0000 0000 0676 00FC 07DC 00FC 07DE
```

Uit deze gegevens in assembler kunt u afleiden wat er aan de hand is. Het programma 'Wack' heeft instructies waarmee u in de hangende computer veranderingen kunt aanbrengen in die gegevens. Vraag op de terminalcomputer met het vraagteken een lijst van de instructies van de 'ROM-Wack':

```
alter boot clear fill find go ig limit list regs reset resume set show user
```

De belangrijkste zijn:

<i>alter</i>	Verandert een gedeelte van het geheugen.
<i>boot</i>	Reboot de hangende computer.
<i>clear</i>	Wist een gedeelte van het geheugen.
<i>fill</i>	Vult een gedeelte van het geheugen.
<i>find</i>	Doorzoekt een geheugengedeelte op een gegeven hex-waarde.
<i>go</i>	Start een programma.
<i>regs</i>	Voert de inhoud van de registers uit.
<i>reset</i>	Heeft hetzelfde effect als 'boot'.
<i>resume</i>	Activeert de Workbench weer, als 'debug' daar werd gestart.

Voer een nieuw adres in als u het adresbereik wilt veranderen (zonder enige opdracht).

We hebben eens onderzocht hoe de structuur van de taak eruit ziet die in het voorbeeld begint op het adres \$0221A0 (zie bij de boekingen van de registers achter TASK). De invoer hoeft dus alleen maar te zijn:

```
0221a0
```

Dat levert het volgende antwoord:

```
0221A0 0000 0810 0000 080C 0D0A 0002 2BB6 0002 ....^H^P....^H^L^M^J..^B +....^B
```

Met behulp van het boek 'Amiga Intern' hebben we het adres gevonden van de pointer naar de naam van de hangende taak. Het is het adres 022BB6, dat ook in bovenstaande regel voorkomt. Op dat adres staat:

022BB6 5245 5330 0000 0000 0000 0002 3338 0000 R E S 0..... 3 8....

We vinden de door onszelf opzettelijk gemaakte fout (task RES0) die de Amiga fataal is geworden. Het programma 'ROM-Wack' is natuurlijk pas echt interessant als een programma om onduidelijke redenen steeds de computer laat hangen. In de toestand 'Task Held' zou u het geheugen van de Amiga kunnen onderzoeken met een monitorprogramma dat u start als tweede CLI-proces.

Het nuttige programma ROM-Wack bewijst dat de RS232-interface ook probleemloos kan functioneren. De transmissie bij 9600 baud werkt foutloos.

4 Batch processing met de Amiga

Alle nu leverbare DOS-computers kunnen stapelbestanden (batch files) verwerken. Ook de Amiga's (en AmigaDOS) ondersteunen deze techniek. Dit hoofdstuk behandelt de achtergronden en de praktijk van batchbestanden.

4.1 Inleiding

In de volgende paragrafen ziet u wat een batchbestand is en hoe de Amiga het uitvoert. Leuke en nuttige toepassingen vindt u in paragraaf 4.2 en volgende.

4.1.1 Wat is een batchbestand?

De CLI-instructies vormen samen een (beperkte) taal. De instructie ECHO lijkt op PRINT uit de taal BASIC. CLI-instructies kunnen worden ingevoerd in de directe modus. Net als in het BASIC-venster van AmigaBASIC wordt een opdracht niet als nieuwe regel toegevoegd aan het programma, maar na de Return meteen uitgevoerd. Het is echter juist een belangrijk kenmerk van een programmeertaal dat opdrachten kunnen worden opgeslagen en dat ze pas na een daartoe strekkende instructie achter elkaar worden afgehandeld.

Bij de Amiga worden reeksen CLI-opdrachten opgeslagen in stapelbestanden (batchbestanden). Omdat dat zuivere tekstbestanden zijn, kunt u ze met iedere willekeurige editor of tekstverwerker aanmaken en redigeren (Notepad, de CLI-editors ED en EDIT, enzovoorts). Het komt er slechts op aan dat het resultaat uit gewone ASCII-tekens bestaat. Sla het bestand op onder een naam naar keuze.

4.1.2 De structuur van batchbestanden

Eenvoudige batchbestanden bestaan uit CLI-opdrachten zoals u die gebruikt in de directe modus in een CLI-venster. Op elke regel mag maar een opdracht staan, maar u mag er wel commentaar aan toevoegen. Commentaar begint met een puntkomma en mag niet langer zijn dan de actuele regel. Bijvoorbeeld:

```
copy test#? testprograms ; kopieert alle testversies
delete test#? ; naar een speciale directory en wist
; ze vervolgens uit de hoofddirectory.
```

Voor batchbestanden bestaan meer instructies dan voor een CLI-venster. Sommige daarvan hebben geen zin in een venster of zijn daar zelfs verboden. Die instructies zijn:

```
echo
failat
quit
if/else/endif
skip/lab
```

ask
wait

Zonder de instructies IF, ELSE en ENDIF zou een batchbestand altijd lineair van boven naar beneden worden uitgevoerd. Met deze drie instructies kan het besturingssysteem reageren op een gegeven situatie en aan de hand daarvan het programma op een andere manier doorlopen (zie paragraaf 2.3).

4.1.3 Batchbestanden oproepen

De instructie EXECUTE roept een batchbestand op en start de uitvoering ervan (zie paragraaf 2.3). In het eenvoudigste geval staat achter de instructie EXECUTE alleen de naam van het uit te voeren bestand:

```
execute df0:mybatch
```

Het besturingssysteem voert all CLI-opdrachten in het bestand 'mybatch' achtereenvolgens uit. Pas daarna is de CLI klaar om nieuwe opdrachten uit te voeren.

Het biedt dus vaak voordeel een batchbestand uit te voeren als aparte taak op de achtergrond. Creëer zo'n taak met de instructie RUN:

```
run execute df0:mybatch
```

De CLI blijft beschikbaar voor andere opdrachten. Uitvoer vanuit de taak op de achtergrond verschijnt wel in het CLI-venster. (Een niet-interactieve taak op de achtergrond krijgt niet automatisch een eigen venster.)

4.1.4 Een eenvoudig voorbeeld

De editor ED biedt voldoende mogelijkheden om snel een kleine tekst te schrijven en is dus geknipt voor batchbestanden (zie paragraaf 2.4.1). Een andere editor of tekstverwerker mag natuurlijk ook.

Neem aan dat u een batchbestand wilt hebben dat de namen van de specifieke batch-instructies weergeeft op het scherm. Het bestand krijgt de naam 'Instructies'. Open eerst een CLI-venster en roep ED op met de naam van het nieuwe bestand:

```
ed instructies
```

Na korte tijd verschijnt ED met een leeg invoervenster. Omdat er nog geen bestand is met de opgegeven naam, staat links onderaan de melding 'Creating new file'. Voer de gewenste instructies in:

```
echo "execute"  
echo "echo"  
echo "failat"  
echo "quit"
```



```
echo "if/then/else"  
echo "skip/lab"  
echo "ask"  
echo "wait"
```

Na de toetsen 'Esc' en 'X' slaat het besturingssysteem het bestand op onder de vermelde naam en verlaat het ED.

De Amiga voert het batchbestand op ieder gewenst moment uit na de opdracht 'execute instructies'. In het CLI-venster komt te staan:

```
execute  
echo  
failat  
quit  
if/then/else  
skip/lab  
ask  
wait
```

4.2 Wijzigingen in de startup-sequence

De meeste eigenaars van een Amiga zijn zich er niet van bewust dat ze hun computer na het overhalen van de netschakelaar op weg helpen met een batchbestand op de Workbench-diskette. Maak voor de zekerheid eerst een kopie van de Workbench-diskette en berg het origineel weg op een veilige plaats. Als we het hebben over de Workbench-diskette bedoelen we in het vervolg altijd een kopie.

Open een CLI-venster en stop de Workbench-diskette in drive 0. Uit de uitvoer van de opdracht 'dir df0:s' blijkt dat in de subdirectory 'S' een bestand staat met de naam 'startup-sequence'. Maak het CLI-venster zo groot mogelijk en geef de opdracht 'type df0:s/startup-sequence'. De diskdrive draait even en op het scherm verschijnt de inhoud van het bestand. Houd na de Return meteen de muis gereed om te voorkomen dat de tekst doorloopt. Zolang u de rechterknop van de muis ingedrukt houdt, scrollt de tekst niet verder. Hieronder staat de tekst afgedrukt van de startup-sequence van de Amiga 2000. Die van de Amiga 500 is aanzienlijk korter:

```
BindDrivers  
echo "A2000 Workbench 1.2 V33.56 27-APR-87*N"  
Assign WB: sys:  
IF EXISTS sys:s/JHO  
  echo "Initializing Janus ...*N"  
  Wait 25  
  Echo "Mounting Janus Hard Disk*N"  
  DJMount  
  IF EXISTS JHO:  
    Assign WB: JHO:  
    Echo "Transferring control to JHO: *N"  
  ELSE
```

```

    Assign WB: sys:
  EndIF
EndIF
IF EXISTS sys:s/DHO
  Assign WB: DHO:
  Echo "Transferring control to DHO: *N"
EndIF
AddBuffers WB: 30
CD WB:
IF EXISTS WB:c
  Assign c: WB:c
EndIF
IF EXISTS WB:s
  Assign s: WB:s
EndIF
IF EXISTS WB:t
  Assign t: WB:t
EndIF
IF EXISTS WB:l
  Assign l: WB:l
EndIF
IF EXISTS WB:libs
  Assign libs: WB:libs
EndIF
IF EXISTS WB:devs
  Assign devs: WB:devs
EndIF
IF EXISTS WB:fonts
  Assign fonts: WB:fonts
EndIF
Assign sys: WB:
IF EXISTS WB:System
  Assign System: WB:System
  Path add System
EndIF
IF EXISTS WB:Utilities
  Assign Utilities: WB:Utilities
  Path add Utilities
EndIf
IF EXISTS WB:PC
  Path add WB:PC
EndIf
Assign WB:
Dir RAM:
Path RAM: add
failat 30
SetClock opt load
LoadWb

```

```
IF Exists sys:s/Startup-HD
  Execute sys:s/Startup-HD
EndIF
endcli > nil:
```

Uit het feit dat de tekst alleen CLI-instructies bevat, kunt u opmaken dat het een batchbestand is. Het is een speciaal type, want het wordt automatisch uitgevoerd bij iedere start van de Amiga. Met behulp van dit bestand geeft u de Amiga opdracht bij welke activiteiten tijdens de start hij zich moet melden en welke opties of modi hij moet kiezen.

(Denk eraan dat u door een typefout een startup-sequence onbruikbaar maakt als daardoor een onbegrijpelijke opdracht ontstaat. Gebruik dus een kopie van de Workbench.)

In het eerste deel van de startup-sequence staan opdrachten voor de installatie van de harddisk (dh0) en/of de XT/AT-kaart (jh0). Die zijn alleen van belang voor eigenaren van een Amiga 2000. Elke overbodige opdracht die u schrapt, spaart tijd. Dat geldt het sterkst voor de Amiga 500. Hieronder gaan we die vereenvoudigingen aanbrengen. Van het eerste deel van het bestand nemen we alleen de opdracht 'Assign WB: SYS:' over. De tekst op de tweede regel is niet essentieel; die laten we ook weg. De twee grote IF-ENDIF blokken hebben betrekking op de harddisk en zijn dus overbodig. De eerstvolgende opdrachten die wel van belang zijn, zijn 'AddBuffers WB: 30-' en 'CD WB:'. In paragraaf 8.3 kunt u nazoeken wat het effect is van de diverse opdrachten.

Elk van de dan volgende ASSIGN-instructies is gevat in een IF-ENDIF blok. Met deze controlevragen onderzoekt de Amiga hoe zijn eigen configuratie er uitziet. Als u de configuratie kent, zijn de vragen overbodig. Vraag in de CLI met de opdracht 'dir df0:' op en kijk welke van de volgende directories op de Workbench-diskette staan: C, S, T, L, Libs, Devs, Fonts, System en Utilities. Het maakt niet uit of er behalve die directories ook .info-bestanden van die bestanden in de lijst staan.

Wis in de startup-sequence de IF-regels voor en de ENDIF-regels achter de ASSIGN-opdracht voor de wel aanwezige directories. Van het middelste deel van de startup-sequence blijven de volgende opdrachten over:

```
Assign c: WB:c
Assign s: WB:s
Assign t: WB:t
Assign l: WB:l
Assign libs: WB:libs
Assign devs: WB:devs
Assign fonts: WB:fonts
Assign sys: WB:
Assign System: WB:System
Path add System
Assign Utilities: WB:Utilities
Path add Utilities
```

Als u geen XT/AT-kaart hebt, heeft het geen zin te zoeken naar de directory 'PC'. Het besturingssysteem hoeft dus geen pad aan te leggen naar die directory. Dat maakt de opdracht 'path add WB:PC' overbodig. Als u de PC-directory en zijn .info-bestand wist, wint u nog wat ruimte op de Workbench. De directories Sidecar en Expansion plus hun .info-bestanden nemen op de Workbench-diskette opslagruimte in zonder dat ze gebruikt worden. Mocht u later een hard-

disk of een XT/AT-kaart willen toevoegen, dan kunt u de benodigde bestanden overnemen van de originele Workbench-diskette.

De opdracht 'Assign WB:' nemen we wel op in de nieuwe startup-sequence.

Beslis zelf of u de opdracht 'dir ram:' wilt overnemen. Als u hem weglaat, staat er geen diskette-icon van de RAM-disk op de Workbench na afloop van de startup-sequence. Die icon verschijnt wel meteen op het scherm als u de regel laat staan. Als u geen RAM-disk aanmaakt, hoeft er ook geen pad heen te leiden. Schrap dus de regel opdracht 'Path RAM: ADD'. Als het u niet interesseert hoe laat u de Amiga inschakelt, mag de regel 'SetClock opt Load' ook verdwijnen. Daarna laadt de opdracht 'LoadWB' de eigenlijke gebruikers-interface (de Workbench). Die kunt u niet missen.

Dan volgt een test of in de directory 'S' het bestand 'Startup-HD' aanwezig is. Vraag die directory nog eens op als u niet meer weet of dat het geval is (dir df0:s). Er staat wel een boeking 'No-hd'. Nu weet u genoeg: er is geen bestand 'Startup-HD'. Het IF-ENDIF blok in kwestie met inhoud is overbodig.

De laatste regel sluit het CLI-venster na afloop van de startup-sequence en laat het verdwijnen (EndCLI > NIL:).

Het resultaat van alle wijzigingen ziet er als volgt uit:

```
Assign WB: sys:
Addbuffers WB: 30
CD WB:
Assign c: WB:c
Assign s: WB:s
Assign t: WB:t
Assign l: WB:l
Assign libs: WB:libs
Assign devs: WB:devs
Assign fonts: WB:fonts
Assign sys: WB:
Assign Utilities: WB:Utilities
Path add Utilities
Assign System: WB:System
Path add System
Assign WB:
LoadWb
endcli > nil:
```

De nieuwe startup-sequence bevat alle belangrijke opdrachten. Omdat er geen controleblokken meer in staan, moet u er wel zeker van zijn dat de aangesproken directories altijd op de Workbench-diskette staan. Als niet aan deze voorwaarde is voldaan, kan de Amiga niet starten. Opstarten met een 'uitegeklede' startup-sequence gaat ongeveer vijftig procent sneller dan met de oude startup-sequence.

Als u een Amiga 500 hebt, ziet u dat in zijn startup-sequence geen ASSIGN-opdrachten staan. Dat mag u zo laten; u hoeft die opdrachten niet toe te voegen.

Het is niet nodig de nieuwe startup-sequence helemaal vanaf het begin te typen. Begin met de oude versie en wijzig deze. Dat gaat het makkelijkste met de editor ED, die op de Workbench-diskette staat. Die procedure bestaat uit de volgende stappen:

- Laat de kopie van de Workbench-diskette in drive 0 zitten.
- Geef de opdracht 'ED df0:S/startup-sequence'.
- Zet de cursor met de pijltjestoetsen ergens in een regel die u niet in de nieuwe startup-sequence wilt hebben. Druk tegelijk op de toetsen 'Ctrl' en 'B'. De hele regel verdwijnt.
- Behandel de andere regels die u wilt wissen op dezelfde manier.
- Geef de opdracht tot opslaan van de nieuwe startup-sequence met achtereenvolgens de toetsen 'Esc', 'X' en 'Return'.

Een en ander resulteert in een snelle versie van de startup-sequence.

U hebt echter niet alleen te maken met snelheid, maar ook met bedieningsgemak. Daarom zullen we de startup-sequence nu weer een paar regels langer maken. Hij wordt dus wat trager, maar geeft u in ruil daarvoor meer informatie en bovendien mogelijkheden tot ingrijpen.

We gaan uit van de verkorte versie van de startup-sequence die we zojuist hebben gemaakt. Dat is een versie voor de Amiga 2000, maar de startup-sequence voor de Amiga 500 is analoog (zie de opmerking over de ASSIGN-opdrachten hierboven).

In de nieuwe versie maken we gebruik van de instructie ASK. Kijk zo nodig in paragraaf 2.3.5 na hoe die te gebruiken. Vraag voor de zekerheid de directory 'C' op. Er blijken versies van de Workbench in omloop te zijn, die de instructie ASK niet bevatten. Kopieer die routine bij uw leverancier of een andere Amiga-gebruiker. In het vervolg van dit boek gaan we ervan uit dat u beschikt over de instructie ASK.

Het is vervelend dat u na de startup-sequence in de Workbench nog een hele reeks icons moet activeren voordat u een bruikbaar CLI-venster ter beschikking hebt. Bij de simpelste oplossing van dit probleem zorgen we ervoor dat het besturingssysteem het CLI-venster niet afsluit en dat het niet terugkeert naar de Workbench. Verwijder daartoe de opdracht 'EndCLI > NIL:'.

Een vriendelijker methode laat u kiezen. Voeg de volgende regels toe vóór de laatste regel van de startup-sequence:

```
ask "Wilt u een CLI-venster open laten? (y/n) "
if warn
  ask "Wilt u een groot (y) of een klein (n) CLI-venster? "
  if warn
    newcli "con:0/0/550/200/Startup-CLI"
  else
    newcli "con:0/0/160/30/Startup-CLI"
  endif
endif
```

Sla de nieuwe versie op en druk dan tegelijk op de twee Amiga-toetsen en de Ctrl-toets (warme start).

De Amiga doorloopt de nieuwe startup-sequence, maar stelt vlak voor het einde de vraag: 'Wilt u een CLI-venster open laten? (y/n)'. De letters Y/N suggereren uw mogelijke antwoorden. U mag ook het woord 'yes' of 'no' gebruiken. Bij het antwoord 'no' gebeurt er niets nieuws. Het CLI-venster gaat dicht en u moet verder de muis gebruiken. Na het antwoord 'y(es)' volgt er nog een vraag naar de gewenste grootte van het CLI-venster. Er verschijnt in ieder geval een venster met de naam 'Startup-CLI'.

Het antwoord 'yes' na de tweede vraag leidt tot een groot venster. Bij 'no' blijft het klein, te klein om er iets mee te kunnen beginnen. Het venster heeft als enige functie als reserve CLI-

venster paraat te staan. Als u het nodig hebt, hoeft u het venster alleen maar te vergroten om verder te kunnen werken met de CLI.

De hierboven besproken toevoeging is alleen maar bedoeld om u op een idee te brengen. Het principe is overal toepasbaar.

Als u alleen met de CLI werkt, hebt u de Workbench en de icons niet nodig. Met de instructie ASK hebt u op ieder moment de keus:

```
ask "Workbench laden? (y/n) "  
if warn  
    loadwb  
endif
```

Vervang de opdracht 'LoadWB' door de bovenstaande vier regels.

De twee voorbeelden geven maar een fractie van de mogelijkheden die er zijn de startup-sequence te kneden naar uw eigen wensen. In paragraaf 4.4 staat een batchbestand dat goed past in een startup-sequence.

4.3 Een bijzonder batchbestand voor de printer

In deze paragraaf construeren we een batchbestand met opdrachten die het uiterlijk van de uitvoer via de printer wijzigen. Om te beginnen besteden we aandacht aan de stuurtekens die de printer nodig heeft voor een wisseling van type en vorm van de letters.

Een stuurteken voor de printer bestaat uit een Esc-teken gevolgd door een of meer andere tekens (een escape-sequentie). Een escape-sequentie wordt naar de printer verstuurd met de opdracht:

```
copy * to prt:
```

Na invoer van deze opdracht en een Return gaat alle volgende invoer rechtstreeks naar de printer. De escape-sequenties kunt u zo op de gewenste plaats invoeren via het toetsenbord. Binnen een batchbestand is het niet mogelijk via het toetsenbord stuurtekens te versturen. We brengen de stuurtekens daarom onder in verschillende batchbestandjes.

Neem aan dat uitvoer wilt hebben in de NLQ-modus (Near Letter Quality, de bijna-briefkwaliteit van matrixprinters). Geef de opdracht:

```
copy * to nlq
```

De diskdrive draait even en maakt het bestand 'NLQ' aan. Alle toetsenbord-invoer gaat rechtstreeks naar dit bestand. Dat heeft als consequentie dat er geen uitvoer meer naar het scherm is en dat u verder blind typt. Ga zeer zorgvuldig te werk; er mogen geen fouten in het bestand 'NLQ' staan. Type de volgende tekst ('Esc' is de speciale toets, niet de lettercombinatie E-s-c):

```
Esc[2"z
```

Na de Return wordt dit opgeslagen in het bestand 'NLQ'. Verlaat de COPY-opdracht met de toetsencombinatie Ctrl-\. In de CLI verschijnt de normale prompt. Voor VET (bold), CURSIEF (italics) en RESET gaat u analoog te werk. Geef de batchbestandjes een duidelijke naam.

Zoek de bijbehorende escape-sequenties op in het handboek bij de Amiga.

Als u een escape-sequentie invoert naar een bestand, verandert soms de weergave op het scherm. Maak het bestand RESET als laatste aan, zodat het scherm terugkeert naar de normale modus. In de rest van deze paragraaf gaan we ervan uit dat de vier bestanden NLQ, VET, CURSIEF en RESET op uw diskette staan. Zorg ervoor dat ze niet in een subdirectory staan.

Hieronder staat het batchbestand dat we in de kop van de paragraaf bedoelden. Typ het in een editor en sla het op onder de naam 'printer'.

```
.key bestandsnaam
if exists <bestandsnaam>
  ask "Wilt u het bestand printen met NLQ? (y/n) "
  if warn
    copy nlq to prt:
  else
    echo "O.K., geen NLQ."
  endif
  ask "Wilt u het bestand vet printen? (y/n) "
  if warn
    copy vet to prt:
  else
    echo "O.K., niet vet."
  endif
  ask "Wilt u het bestand cursief printen? (y/n) "
  if warn
    copy cursief to prt:
  else
    echo "O.K., niet cursief."
  endif
  copy <bestandsnaam> to prt:
  copy reset to prt:
  echo "Klaar."
else
  echo "Het bestand <bestandsnaam> is niet aanwezig."
endif
```

Start dit batchbestand met de opdracht:

```
execute printer s/startup-sequence
```

Daarna krijgt u achtereenvolgens de vragen voorgelegd of het bestand moet worden afgedrukt in NLQ, met vette letters of in cursieve letters. Beantwoord een vraag alleen met 'y' als uw printer dat lettertype ondersteunt.

Kort na beantwoording van de derde vraag begint de printer met het gekozen lettertype het bestand 'startup-sequence' uit de directory 'S' af te drukken. Op de plaats van de startup-sequence kunt u de naam van een willekeurig ander bestand zetten. Vergeet niet het pad te vermelden als het gezochte bestand in een subdirectory staat. In het voorbeeld zijn maar drie keuzen gebruikt. Breid dat naar believen uit met andere mogelijkheden van uw printer. Uiteraard hoort daar voor elke printermodus een batchbestandje bij.

4.4 Zelf batch-instructies maken

In deze paragraaf geven we drie voorbeelden hoe u zelf een instructie kunt samenstellen. Het zijn eigenlijk geen instructies, maar opdrachten die worden uitgevoerd onder invloed van de instructie EXECUTE.

Daar is in de eerste plaats de instructie BACKUP. Met deze instructie kunt u van ieder bestand een backup-kopie maken. Het nieuwe bestand verschilt alleen van het origineel door zijn naam, die eindigt op de extensie '.bak'. Typ de volgende regels in een editor en sla het programma op onder de naam 'Backup'.

```
.key bestandsnaam /k
if exists <bestandsnaam>
    copy <bestandsnaam> to <bestandsnaam>.bak
else
    echo "Bestand niet te vinden."
endif
```

Een complete oproep van de instructie ziet er als volgt uit:

```
execute backup bestandsnaam
```

Het resultaat is een kopie die de naam 'bestandsnaam.bak' krijgt.

De tweede instructie heet 'Window'. Met deze instructie kunt u maximaal zes CLI-vensters openen met één opdracht. Typ onderstaande regels in een editor en sla de tekst op onder de naam 'Window'.

```
.key aantal
skip <aantal>
lab 6
newcli "con:0/0/319/59/Een_CLI"
lab 5
newcli "con:320/0/319/59/Een_CLI"
lab 4
newcli "con:0/60/319/59/Een_CLI"
lab 3
newcli "con:320/60/319/59/Een_CLI"
lab 2
newcli "con:0/120/319/59/Een_CLI"
lab 1
newcli "con:320/120/319/59/Een_CLI"
```

Roep de instructie op met de opdracht:

```
execute window n
```

Hierin is 'n' een van de getallen 1 tot en met 6. Een foutmelding kan het gevolg zijn van een gebrek aan geheugen of een ongeldige waarde voor n.

De nieuwe batch-instructies staan tot nu toe in de hoofddirectory van de diskette. Het is

gunstiger al deze batchbestanden te kopiëren naar een subdirectory. De subdirectory 'S' is daar geschikt voor. Het besturingssysteem zoekt daar automatisch een batchbestand dat hoort bij de instructie die is vermeld in de EXECUTE-opdracht.

In de subdirectory 'T' slaat de computer tijdens de uitvoering van een EXECUTE-opdracht allerlei bestanden op. De computer gebruikt die bestanden intern. U hoeft niet te schrikken als u na oproep van een batchbestand de melding krijgt 'Disk is write protected'. Het besturingssysteem kan dan geen subdirectory 'T' vinden; maak er zelf een aan. Als u met de RAM-disk werkt, treedt deze melding niet op. Soms ontstaat er plotseling als vanzelf een subdirectory 'T' op de RAM-disk. Daar is het besturingssysteem verantwoordelijk voor.

Ten slotte geven we een batchbestand dat een aantal instructies naar de RAM-disk kopieert. Daar kan het systeem erover beschikken met een verkorte naam. Het batchbestand ziet er als volgt uit:

```
if not exists RAM:d
  mkdir RAM:d
  copy sys:c/copy RAM:d/c
  copy sys:c/path RAM:d/p
  assign c: RAM:d
else
  copy sys:c/copy RAM:d/c
  c sys:c/path RAM:d/p
endif
p add sys:c
c sys:c:dir RAM:d/d
c sys:c:execute RAM:d/ex
c sys:c:delete RAM:d/del
c sys:c:type RAM:d/t
c sys:c:rename RAM:d/r
c sys:c:echo RAM:d/e
e "De verkorte instructies zijn nu beschikbaar."
```

Typ deze regels in een editor en sla het bestand op onder de naam 'RAMon'. Nadat u de computer het bestand heeft uitgevoerd ('execute ramon') kunt u diverse instructies oproepen met korte namen: c, p, d, ex, del, t, r en e. Welke instructies bij deze korte namen horen, kunt u afleiden uit de programmaregels.

Maak zelf meer instructies met korte namen. Let erop dat u een afkorting niet gebruikt voor meer dan één instructie. U mag dit batchbestand ook opnemen in de startup-sequence, zij het alleen aan het eind.

De tegenhanger van 'RAMon' is 'RAMoff'. Dit batchbestand geeft het geheugen vrij dat door de verkorte instructies in beslag werd genomen. Zie hier de tekst:

```
if exists RAM:t
  delete RAM:t all quiet
endif
cd ram:d
delete #? quiet
cd df0:
echo "De verkorte instructies zijn niet meer beschikbaar."
```

De opdracht 'execute ramoff' wist zowel de verkorte instructies als de subdirectory 'T' als die aanwezig is op de RAM-disk. Het vrijgekomen geheugen is nu weer beschikbaar voor ander gebruik.

De subdirectory 'D' wordt niet gewist door 'RAMoff'. De subdirectory is wel leeg; er staan geen instructies meer in. De Amiga zal echter bij een poging deze subdirectory te wissen een foutmelding geven die inhoudt dat 'D' niet mag worden gewist. Dat is een gevolg van het feit dat in het batchbestand 'RAMon' met de ASSIGN-opdracht het gebruik van een directory 'D' op de RAM-disk is geregeld.

Dit kleine nadeel valt in het niet bij het grote voordeel de keuze te hebben de originele of de verkorte CLI-instructies te gebruiken.

5 AmigaDOS en multitasking

De Amiga steekt op het gebied van zowel hardware als software met kop en schouders boven zijn concurrenten uit. De Amiga bevat standaard de supersnelle Blitter-chip voor operaties in het geheugen. Bij veel andere systemen kunt u kiezen uit maximaal 64 kleuren; de Amiga biedt 4096 kleuren tegelijk.

De software van de Amiga is nog krachtiger dan de hardware. Zowel op het terrein van venders als dat van multitasking kan de Amiga veel meer dan een willekeurige andere computer. Bovendien is de software van de Amiga ontworpen om toekomstige verbeteringen en uitbreidingen van de hardware te ondersteunen. Dat geldt bijvoorbeeld voor een vervanging van de 68000-processor door het snellere type 68020 en voor een uitbreiding van het geheugen tot een paar Mbyte. De graphics.library van de Amiga hoeft nauwelijks te worden uitgebreid als Commodore vliegensvlug een grafische kaart met een grotere resolutie, meer kleuren en opties op de markt brengt.

Het besturingssysteem van de Amiga loopt ver voor op de actuele hardware. In het besturingssysteem neemt AmigaDOS een bijzondere plaats in. AmigaDOS maakt echte multitasking mogelijk op een home computer. Het besturingssysteem biedt ruimte voor uitbreiding met veel functies die nodig zijn bij grotere mogelijkheden van de hardware. Het principe van device drivers is consequent toegepast, zodat alle door DOS ondersteunde apparaten met dezelfde routines kunnen worden aangesproken. De mogelijkheid tot omleiding van invoer en uitvoer is een belangrijke voorwaarde voor multitasking. AmigaDOS heeft veel meer mogelijkheden dan bijvoorbeeld MS-DOS.

Eigenlijk is AmigaDOS een te ruime jas voor een computer die niet alle mogelijkheden aankan. Multitasking op een Amiga 500 met 512 Kbyte intern geheugen is zoiets als met een Porsche op een zandweg rijden. AmigaDOS zou veel meer kunnen met een 68020-processor dan met de 68000. Het valt te hopen dat iedere eigenaar van een Amiga spoedig over al die mogelijkheden zal kunnen beschikken.

5.1 Multitasking

De meeste computergebruikers hebben er niet veel moeite mee een definitie van multitasking te geven. 'De computer verricht meerdere taken tegelijk'. Toch zitten de meesten te wachten tot een diskette is geformatteerd voor ze verder werken in één CLI. Of ze wachten met oproepen van de editor totdat de C-compiler klaar is met vertalen. Zo maken ze een marginaal gebruik van AmigaDOS.

Multitasking is iets heel natuurlijks. Mensen doen het voortdurend in hun dagelijkse doen en laten. U voert bijvoorbeeld de handelingen ter voorbereiding van een maaltijd niet na elkaar, maar zoveel mogelijk tegelijkertijd uit.

Bij veel computers is multitasking echter niet vanzelfsprekend. Multitasking vereist tamelijk veel planning en organisatie. Het is immers onmogelijk bijvoorbeeld een bestand te schrijven op een diskette die juist wordt geformatteerd. Opslaan van een bestand dat op dat moment wordt geprint levert ook een conflict op. In de volgende paragrafen gaan we daar nader op in.

Een multitasking computer verricht een aantal taken schijnbaar tegelijk. Het besturingssysteem knipt alle taken in kleine stukjes en handelt die om beurten zo af dat er geen conflicten ontstaan en zo dat ze ongeveer tegelijk klaar zijn. AmigaDOS probeert nooit twee bestanden tegelijk te printen. AmigaDOS weigert ook hardnekkig een C-programma op een diskette te

veranderen terwijl het op dat moment door een C-compiler wordt vertaald. De firma Metacomco heeft al deze problemen voorzien en voorkomt ze met geschikte routines. In paragraaf 5.6 bespreken we conflicten tussen taken en methoden om die te vermijden.

In de eerstvolgende paragrafen komen positieve ontwikkelingen aan de orde: makkelijker werken, tijd besparen en problemen oplossen.

5.2 Multitasking met CLI en Workbench

Hoewel u in de Workbench een aantal programma's tegelijk kunt starten, is die gebruikersinterface niet echt multitasking. Als u een diskette formatteert en tegelijk een programma probeert te starten, vindt u geen gehoor. De cursor verandert tijdens het formatteren in een wolkje en is nauwelijks in staat tot enige actie. In de CLI kunt u wel verschillende taken tegelijk laten lopen.

Als u een Amiga 500 met één diskdrive hebt, is het belangrijk dat u de meest gebruikte CLI-instructies op de RAM-disk hebt (zie paragraaf 4.2). Zo niet, dan wensen we u engelengeduld toe als Intuition een programma van de diskette probeert te laden terwijl AmigaDOS tegelijkertijd naar de routine van een instructie zoekt.

Als u een Amiga met twee diskdrives hebt, is het raadzaam de systeemdiskette met de CLI-instructies in de ene drive te stoppen en ze te starten vanuit de Workbench in de andere drive. Verander de startup-sequence als u werkt met een ongewijzigde kopie van de systeemdiskette. De originele startup-sequence heeft als laatste opdracht de afsluiting van de CLI:

```
endcli >nil:
```

Zelfs de melding over de sluiting verdwijnt in het niets. Doe de kopie van de Workbench-diskette in drive 0 en laad de startup-sequence met de opdracht:

```
ed df0:s/startup-sequence
```

Wis de ENDCLI-opdracht. Sla de nieuwe versie van de startup-sequence op met de toetsen 'Esc' en 'X'. Het grootste voordeel van de nieuwe configuratie is dat u steeds de CLI achter de hand hebt om problemen uit de wereld te helpen. Neem aan dat u een paar belangrijke bestanden op de RAM-disk hebt en dat een programma de Amiga laat hangen tijdens het laden ervan. Dat zal u regelmatig overkomen als het geheugen van uw Amiga niet is vergroot. Er rest slechts een reset ten koste van de bestanden op de RAM-disk.

Als u achter alle andere vensters een CLI-venster hebt staan, kunt u dat naar voren halen en daar een opdracht geven de bestanden te redden:

```
copy ram: df0: all
```

Zo'n CLI op de achtergrond biedt natuurlijk geen soelaas als uw Amiga het definitief opgeeft met de boodschap 'GURU Meditation'. Maar ook bij een tamelijk ongevaarlijke requester met de mededeling:

```
Task held - finish all disk activities .....
```

bent u meestal de pineut als u geen CLI-venster paraat hebt. Een piepklein CLI-venstertje op

de achtergrond kost niet veel geheugenruimte, maar voorkomt veel teleurstelling en ergernis.

De mogelijkheden van samenwerking van Workbench en CLI worden heel duidelijk als u een aantal bestanden wilt kopiëren. Neem aan dat u bezig bent in een editor en dat een vriend langs komt met de vraag een paar teksten te kopiëren naar zijn diskette. Volgens een methode maakt u het venster van de editor zo klein mogelijk en schuift u op de Workbench met de muis de gewenste bestanden naar de nieuwe diskette. Het is eenvoudiger het CLI-venster naar voren te halen en met COPY alle bestanden naar zijn diskette te kopiëren. Ondertussen kunt u gewoon doorwerken en uw alinea afmaken. Het is ook geen enkel probleem vooraf zijn diskette te formatteren:

```
format df0:
```

Zodra de cursor op de volgende regel staat, kunt u de volgende opdracht invoeren:

```
copy df1:teksten/##? df0:
```

U hoeft niet op het resultaat te wachten om uw nieuwe spelletjes te demonstreren.

Het voorgaande gaat mutatis mutandis ook op voor wissen van bestanden. Neem aan dat u een aparte directory hebt waarin u met BACKUP een veiligheidskopie maakt van al uw teksten (zie paragraaf 4.4). Als de diskette langzamerhand vol raakt en u hebt de .bak-bestanden niet meer nodig, kunt u op de Workbench de directory met backups openen, de bestanden allemaal activeren en ze wissen met de optie 'Disgard'. Al deze handelingen vragen nogal wat tijd en blokkeren de Workbench voor ander werk. Het is veel effectiever als u het CLI-venster naar voren haalt en daar de volgende opdracht geeft:

```
delete df0:backups/##?
```

Alle bestanden in de directory 'backups' worden gewist terwijl u verder kunt werken in de Workbench.

In het samenspel tussen Workbench en CLI heeft het standaardvenster van de CLI hinderlijke afmetingen en een plaats die slecht uitkomt. Dat heeft tot gevolg dat u voor uw werkzaamheden in de Workbench dat venster moet verkleinen en opzij schuiven. Dat kan met de muis, maar ook via de CLI. Open een venster met de gewenste eigenschappen en sluit het standaard CLI-venster. Een voorbeeld van het voor dat doel gewijzigde deel van de startup-sequence:

```
.....  
.....  
.....  
NEWCLI "CON:10/10/60/60/MijnCLI"  
ENDCLI  
.....  
.....  
.....
```

5.3 Multitasking met NewCLI

We gaan ervan uit dat u er nu aan gewend bent steeds een CLI-venster achter de hand te hebben en dat u belangrijke werkzaamheden ongemerkt door de CLI laat uitvoeren. Ook dan kunt u het rendement van de CLI nog flink opvoeren (!).

Denk eens aan de volgende situatie. U hebt vandaag vijf nieuwe teksten geschreven en ze opgeslagen in de directory 'df0:teksten'. Daar staan al veertig boekingen en u wilt de vijf nieuwe nog printen via de CLI. Vooraf wilt u de teksten nog eens bekijken zonder er veranderingen in aan te brengen. Doe dat met de opdracht:

```
type df0:teksten/tekstnaam
```

Voor afdrucken luidt de opdracht:

```
copy df0:teksten/tekstnaam prt:
```

Als u de naam niet meer paraat hebt, vraagt u eerst de inhoud van de directory op ('cd df0:teksten' en 'dir'). Aha, de naam van het bestand is 'Peter1.10.87'. Bekijk die tekst en druk hem af (zie boven). De printer is snel klaar, maar hoe heten de andere bestanden ook al weer? Met een notitieblaadje voorkomt u herhaling van zetten.

Een heel andere methode maakt gebruik van een extra CLI-venster. Schuif dat venster naar de bovenste helft van het scherm en geef daar de instructie DIR. (Stop zo nodig de uitvoer door een willekeurige toets in te drukken.) In het eerste CLI-venster is ruimte om de teksten op het scherm te brengen met de instructie TYPE. De namen van de bestanden scrollen niet meer over de bovenrand van het scherm.

Deze werkwijze is altijd voordelig als u verder wilt werken met de uitvoer in een CLI-venster (bestanden wissen of kopiëren). Verwijder het hulpvenster na afloop met de instructie ENDCLI.

In de hierboven geschetste situatie bestaat het werk uit drie verschillende taken. U spaart nog meer tijd als u die parallel uitvoert via drie CLI-vensters. Bekijken van de ene tekst gaat immers goed samen met afdrucken van een andere. Dat scheelt u een gedwongen koffiepauze bij lange bestanden.

Als u vaak met de CLI werkt in een niet al te ruim bemeten geheugen, raden we u aan altijd met een tweede CLI-venster te werken. Het argument is hetzelfde als bij de Workbench. Als het standaardvenster van de CLI bezet door een langdurige taak, of als het zich heeft opgehangen door een programma, dan kunt u verder werken met het andere CLI-venster. Als u in het enige CLI-venster per ongeluk de opdracht 'copy tekst prt:' geeft, moet u wachten of langs een lange omweg via de Workbench en een aantal vensters een tweede CLI-venster openen. Dat kunt u maar beter vooraf hebben gedaan.

Maak gebruik van de eigenschappen van AmigaDOS als u met extra CLI-vensters werkt. In een nieuw CLI-venster gelden dezelfde directories en devices voor invoer en uitvoer als in het vorige venster. AmigaDOS slaat die gegevens op. Bij de opening van een nieuw CLI-venster kunt u de afmeting en de positie meteen opgeven. Bijvoorbeeld:

```
newcli "con:10/10/60/60/MijnCLI"
```

Deze opdracht opent een minuscule CLI-venster in de linker bovenhoek. Typ deze lange opdracht in de editor ED en schrijf het bestand op de diskette onder de naam 'NC' (NewCLI).

Dat verkort de opdracht voor de opening van dit nieuwe venster aanzienlijk:

```
execute df0:nc
```

Als u de verkorte naam (E) van de instructie EXECUTE gebruikt en als df0: de actuele directory is, blijft er niet veel meer over van de lange opdracht:

```
e nc
```

5.4 Multitasking met Run

Er is een praktische grens aan het aantal vensters dat u met NEWCLI op het scherm kunt zetten. Meer dan vier vensters is zeer onoverzichtelijk. De Amiga kan weliswaar meer vensters aansturen, maar de afmeting van het scherm van de monitor beperkt de ruimte voor de vensters. Alle CLI-vensters maken deel uit van één Workbench-screen. Het werkt niet prettig voortdurend een ander venster naar voren te halen of naar de achtergrond te brengen. Bij een Amiga 500 (512 Kbyte) kost ieder venster bovendien relatief veel geheugen. Met een paar geopende vensters blijft er niet veel ruimte over voor andere taken.

De instructie RUN maakt het mogelijk een taak uit te voeren zonder daar een venster voor te hoeven openen. Deze instructie installeert een CLI die geen venster nodig heeft. Doe de Workbench-diskette met de CLI-instructies in df0. Voer de volgende regels in:

```
run copy df0:c/run ram:c  
dir df0:
```

Zodra u de eerste regel afsluit met de Return, reageert het besturingssysteem met de regel:

```
[cli 2]
```

en op de volgende regel de prompt: 1>. (Als er al andere CLI-processen lopen staan er andere getallen.) Probeer de tweede regel in te voeren terwijl de drive nog draait voor de eerste opdracht. De CLI zet onderwijl de inhoud van de diskette op het scherm. Dit voorbeeld heeft geen praktische betekenis, maar demonstreert alleen het idee achter RUN. RUN is geschikt voor alle taken die enige tijd vragen en geen uitvoer veroorzaken.

De eerste voorwaarde spreekt voor zich; de consequenties van de tweede eis zijn moeilijker te overzien. Voer meteen na elkaar de volgende regels in:

```
type df0:s/startup-sequence  
run dir df0:
```

Eerst verloopt alles normaal, maar dan verschijnen er in de lijst Startup-opdrachten stukken van de inhoud van df0. Omdat een met RUN gestarte CLI-opdracht geen eigen venster heeft, gaat de uitvoer naar het actuele CLI-venster.

Vooraf tijdovende CLI-instructies zonder uitvoer zijn geschikt voor multitasking met RUN (FORMAT, DISKCOPY, COPY). Een voorbeeld:

```
run copy tekst prt:
```

Terwijl de printer bezig is, kunt u de Amiga gebruiken voor ander werk. Denk eraan dat u bij deze opdracht de diskette niet uit de drive mag halen als u één diskdrive hebt. Als u de diskdrive nodig hebt, moet u 'tekst' kopiëren naar de RAM-disk en van daaruit laten printen.

RUN biedt veel voordelen in de situatie waarin u in de CLI een programma start en er een flinke periode mee gaat werken. Volgens de oude procedure opent u een venster met NEWCLI, gaat dan naar het nieuwe venster en roept het programma op met zijn naam. Het eerste venster blijft die hele periode onbenut op de achtergrond, maar kost wel ruimte in het geheugen. Het is beter het programma in de eerste CLI te starten met de opdracht:

```
run programma
```

De hierboven beschreven situatie doet zich vaak voor als u met een editor werkt. Neem aan dat u bezig bent een programma in de taal C te schrijven en dat u steeds de nieuwste versie wilt testen. Als u maar een CLI hebt en daar het programma rechtstreeks oproept, moet u voor elke test de editor verlaten en de compiler laden. Starten van de editor vanuit een tweede CLI legt beslag op een geheugengedeelte voor het venster. De oplossing is dus:

```
run ed programma.c
```

Sla een nieuwe versie op met de toetsen 'Esc', 's' en 'a'. De CLI blijft beschikbaar om de compiler op te roepen (met RUN). Zodra de compiler een fout signaleert, kunt u in de editor de regel met die fout verbeteren.

De aard van de instructie RUN houdt een beperking in van de programma's die u kunt starten. Het probleem met de uitvoer hebben we al besproken. Omdat voor invoer per se een venster nodig is, mag u RUN niet toepassen op programma's die (misschien) om invoer vragen. Zorg ervoor dat elk programma met invoer een eigen venster heeft. Als u twee zulke programma's start vanuit één venster, weet de Amiga niet waar die invoer heen moet.

De meeste CLI-opdrachten zullen na de invoer ervan geen gegevens meer vragen, maar u kunt ze wel afbreken met Ctrl-C of een andere Ctrl-functie. Als een CLI-opdracht loopt onder RUN, gaat een instructie Ctrl-C niet naar die opdracht maar naar de actuele CLI. Dat is een doodlopende weg.

Stel dat u een tekst van 30 Kbyte even snel wilt bekijken met de opdracht:

```
run type tekst
```

Het is geen probleem de tekst te stoppen met een willekeurige toets, maar als u de opdracht wilt afbreken met Ctrl-C, heeft dat geen effect. AmigaDOS bevat heeft voor dit geval de instructie BREAK. Ze simuleert een Ctrl-C voor een opdracht die is gestart met RUN. Start de weergave van de tekst vanuit het eerste venster met 'run type tekst'. Onderbreek die uitvoer vanuit het tweede venster met de opdracht:

```
break 1
```

Op deze plaats willen we alleen maar laten zien dat het ook mogelijk is een RUN-opdracht af te breken. De instructie BREAK wordt volledig besproken in paragraaf 2.2.6.

Ten slotte gaan we iets dieper in op RUN en NEWCLI. In paragraaf 7.2 introduceren we de zelfgemaakte CLI-instructie TASK. Die levert - in tegenstelling tot STATUS - niet alleen de lopende CLI-processen, maar ook alle taken van de Amiga. Elke NEWCLI-instructie genereert

een nieuwe taak met de naam 'New CLI'. Een RUN-opdracht genereert een taak met de naam 'Background CLI'. Een lijst van taken ziet er bijvoorbeeld als volgt uit:

CON	5
Initial CLI	0
input.device	20
File System	10
Workbench	1
trackdisk.device	5
CON	5
Background CLI	0

De boeking 'Background CLI' in de laatste regel betekent dat er onder RUN een opdracht loopt zonder eigen invoer- en uitvoervenster.

Nog een voorbeeld:

CON	5
New CLI	0
CON	5
Initial CLI	0
Workbench	1
File System	10
input.device	20
trackdisk.device	5
RAM	0
CON	5

In dit voorbeeld staat op de tweede regel 'New CLI'. Dat betekent dat het tweede CLI-venster net is geopend. Invoer vanaf het toetsenbord verloopt dus via dat venster.

5.5 Effectief gebruik van de CLI

In de paragrafen 5.2 tot en met 5.4 hebben we u een indruk gegeven van de mogelijkheden tot besparing van tijd en moeite die mogelijk zijn als u de kracht van AmigaDOS uitbuit.

Daar is in de eerste plaats de combinatie van de Workbench met de CLI voor tijdvreterende opdrachten (formatteren, kopiëren). Met de CLI is veel meer mogelijk dan met de Workbench, die vooral is ontworpen met het oog op het gebruiksgemak. Met de CLI kunt u ook bestanden zien die geen .info-bestand hebben. Dat geldt ook voor subdirectories zonder icon. Dat is van belang als u de hand hebt gelegd op een diskette waarop u interessante programma's vermoedt, terwijl die diskette volgens de Workbench leeg lijkt.

Bij intensiever gebruik van de CLI is het raadzaam een goede planning te maken. Bedenk of een opdracht uitvoer gaat leveren. Zo ja, dan mag u hem niet starten met RUN. Het is altijd verstandig een tweede CLI-venster op de achtergrond paraat te hebben. Als u per ongeluk een verkeerde, langdurige opdracht hebt gegeven en er is maar een CLI-venster, dan hebt u slechts de keus uit wachten of afbreken. Dat komt vaker voor dan u denkt.

Multitasking met de CLI is niet goed mogelijk zonder nadenken over de beschikbare devices. Dat geldt met name bij één diskdrive en 512 Kb RAM. Het heeft niet veel zin de Amiga op

te tuigen met drie opdrachten in verschillende CLI-vensters als u door die combinatie wordt gedegradeerd tot disk-jockey. Dat voorkomt u voor een groot gedeelte door de meest gebruikte CLI-opdrachten te kopiëren naar de RAM-disk. Daarmee verhindert u dat u tijdens het printen een diskette met een tekst uit de drive moet halen om een CLI-instructie te laden.

Het vergt enige organisatie om bestanden van verschillende diskettes te printen. Kopieer ze naar één diskette of gebruik de RAM-disk.

Als u een geheugenuitbreiding tot 1 Mbyte hebt, zijn er al veel minder problemen. Er is ruimte voor meer CLI-instructies en programma's in de RAM-disk en ook voor een aantal gegevensbestanden. Een andere praktische configuratie bestaat uit een kleine Amiga met twee diskdrives: één voor de Workbench-diskette (in df0:) en één voor de gegevens (in df1:). Het geheugen van de Amiga is dan bijna helemaal vrij voor programma's.

Bij de uitvoermedia kunt u niet elke willekeurige combinatie van opdrachten kiezen. Het heeft weinig zin twee verschillende bestanden tegelijk uit te voeren op het scherm, maar het kan wel (eenmaal 'run type' en eenmaal 'type' in de oorspronkelijke CLI). AmigaDOS verhindert simultane uitvoer van twee bestanden via een interface (printer, RS232). Maak dus één tekst af, print hem en schrijf onderwijl verder aan de tweede tekst.

Als u verwacht dat afdrukken veel tijd zal vergen, is het voordeliger het bestand inclusief stuurcodes naar een ander bestand te schrijven. Een goed tekstverwerkingsprogramma moet dat kunnen. Geef daarna vanuit de CLI opdracht het gecodeerde bestand te printen.

Bij een directory 'X' met veel boekingen heeft de instructie DIR nogal wat tijd nodig om alles op het scherm te krijgen. Als u die informatie niet meteen nodig hebt, stuurt u de uitvoer naar een bestand en bekijkt het resultaat op een geschikt moment met de instructie TYPE. Dat gaat veel sneller dan wachten op het resultaat. De opdrachten zijn:

```
run dir >ram:inhoud df0:x
```

en later:

```
type ram:inhoud
```

RUN is essentieel. Die instructie voorkomt dat u moet wachten op de afloop van de DIR-opdracht. En daar gaat het juist om.

Met batchbestanden maakt u nog beter gebruik van multitasking. De startup-sequence die in de loop van dit boek is gegroeid, maakt het u wel gemakkelijk, maar vraagt veel tijd (ongeveer twee minuten). Als u aan het begin ervan de instructie NEWCLI opneemt, kunt u in een nieuw CLI-venster meteen aan de slag. Op dat moment hebt u alleen de beschikking over de standaardtekenset. Als u een andere wilt hebben, hebt u aan het begin bijvoorbeeld voor een Duits toetsenbord de volgende opdracht nodig: 'df0:system/setmap df0:devs/keymaps/d'. In plaats daarvan kunt u de startup-sequence beginnen met een regel die een eigen startreeks uitvoert voor het nieuwe CLI-venster:

```
execute df0:s/MySequence
```

In dat bestand staan de opdrachten voor de tekenset en keuzes voor andere opties. Zo krijgt het extra CLI-venster de gewenste eigenschappen voordat de hele startup-sequence is doorlopen.

5.6 ChangeTaskPri

In de voorgaande paragrafen hebben we een groot aantal manieren besproken om taken zoveel mogelijk parallel uit te voeren en in elkaar te schuiven. Het besturingssysteem maakt steeds een keus uit stukjes van de diverse taken. Bij die keus gelden regels waardoor de belangen van de taken botsen.

De editor ED is tamelijk traag bij de weergave op het scherm. Als er bovendien op de achtergrond een taak loopt die veel rekentijd vraagt, zult u merken dat ED regelmatig eventjes is geblokkeerd. Soms is dat zeer hinderlijk.

Het besturingssysteem van de Amiga kent elke taak een prioriteit toe. Taken met een lage prioriteit komen minder vaak en later aan de beurt bij de processor. Als u met ED een tekst schrijft en u bent benieuwd naar het uiterlijk van een voorlopige versie, dan maakt het niet zo veel uit of de afdruk na een of twee minuten klaar is. Het is daarentegen erg hinderlijk als u niet verder kunt typen omdat de editor is geblokkeerd vanwege een lage prioriteit.

Met de instructie ChangeTaskPri kunt u ingrijpen in de standaardprioriteiten van CLI-opdrachten. De prioriteit mag een waarde hebben tussen -128 en +127.

Vraag de actuele gegevens op over de CLI-processen:

```
status full
```

Het resultaat voor een gegeven CLI is bijvoorbeeld:

```
Task 1: stk 1600, gv 150, pri 0 Loaded as command: status
```

De boeking 'pri 0' is de prioriteit van het actuele proces. Het is de verstekwaarde (default) bij de start van een CLI. Verander de prioriteit in '3':

```
changetaskpri 3
```

Het resultaat van de opdracht 'status full' is nu:

```
Task 1: stk 1600, gv 150, pri 3 Loaded as command: status
```

De CLI heeft zijn eigen prioriteit verhoogd tot 3. Dat houdt nog niet in dat alle taken met een lagere prioriteit moeten wachten tot een taak met prioriteit 3 volledig is voltooid. Dat is gemakkelijk na te gaan met een voorbeeld. Open een extra CLI met de instructie NEWCLI en verschuif de vensters zodanig dat het eerste venster de onderste helft van het scherm in beslag neemt en het tweede venster de bovenste helft. Geef in het onderste venster de opdracht 'changetaskpri -127' en in het bovenste venster 'changetaskpri 127'. Voor de vergelijking komt het erop aan de taken op hetzelfde moment te starten. Schrijf in beide vensters (zonder Return):

```
list df0:c
```

Het is de kunst de opdrachten zo snel mogelijk door te geven naar de taken. Dat gaat het best met de muis. Activeer het onderste venster, geef een Return, activeer het bovenste venster en geef nog een Return.

Het verschil in snelheid waarmee de directory 'C' in beide vensters verschijnt, is minder groot dan u waarschijnlijk verwachtte. De werkwijze van het besturingssysteem dempt het ef-

fect van het grote verschil in prioriteit. Zodra een taak iets vraagt van het besturingssysteem en wacht op het resultaat, verhuist die taak naar de wachtkamer (Waiting List). Hij kost in die periode geen rekentijd. Bovendien krijgt dan een taak met een veel lagere prioriteit de kans. Als een taak een spoor van een diskette laadt, gaat die taak in de wachtrij totdat het spoor helemaal is gelezen. Het volgende voorbeeld veroorzaakt een groter verschil. Verlaag van CLI 1 de prioriteit met de opdracht 'ChangeTaskPri 0' en genereer een proces met:

```
run ed ram:verschil
```

Geef in het lege venster van de editor ongeveer twintig keer een Return. Dat verschaft bewegingsruimte aan de cursor. Verschuif het CLI-venster zodanig dat u, behalve in de editor, daar ook iets kunt invoeren. Vraag in het CLI-venster de uitgebreide inhoud op van de directory 'X':

```
dir df0:x
```

Probeer de cursor te bewegen terwijl de inhoud op het scherm verschijnt. Hij volgt de positie van de muis heel slecht. De cursor blijft stilstaan als het besturingssysteem met een andere taak bezig is. Als u ondertussen de muis beweegt, gaat de cursor op het scherm met een sprong verder als de muisinvoer weer aan de beurt is.

Als het werk in het edit-venster het belangrijkste is, moeten de prioriteiten anders worden verdeeld. Verlaat de editor met Ctrl-X en geef de CLI de prioriteit 99 (ChangeTaskPri 99). Start de editor weer (run ed ram:verschil) en geef daarna het CLI-proces de prioriteit -99. Vraag de verdeling van de prioriteiten op met de opdracht 'status full'. Het resultaat is:

```
Task 1: stk 1600, gv 150, pri 157 Loaded as command: status
Task 2: stk 3200, gv 150, pri 99 Loaded as command: ed
```

Laat u niet in de war brengen door de vreemde waarde van de eerste prioriteit (157 = 256 - 99). De instructie STATUS voert alleen waarden uit tussen 0 en 255. (Het programma Task levert wel de juiste waarden; zie paragraaf 7.2).

Herhaal de hierboven beschreven test. Start in de CLI de uitvoer van de lange directory en beweeg met de muis de cursor in de editor. De beweging van de cursor is soepel, maar de weergave van de bestandsnamen stopt bijna helemaal als u in het venster van de editor een toets ingedrukt houdt. CLI 1 wacht op pauzes in ED.

Samenvatting

- Voor soepel werken met meer dan één proces is een goede verdeling van de prioriteiten nodig. Minder belangrijke taken duren iets langer, maar belangrijk werk verloopt bijna ongestoord.
- Verander de prioriteiten van de taken met de instructie CHANGETASKPRI. Hoewel waarden zijn toegestaan van -128 tot en met +127, is het meestal niet raadzaam waarden te gebruiken buiten het bereik van -5 tot en met +5 (zie paragraaf 5.7).
- Denk goed na voordat u de instructie CHANGETASKPRI gebruikt. U kunt alleen de prioriteit veranderen van het CLI-proces waarin u die opdracht geeft. De nieuwe prioriteit werkt ook door in alle dochterprocessen (bijvoorbeeld standaardinvoer en -uitvoer en de actuele directory). Daarom krijgen RUN-opdrachten de prioriteit van het venster waarin de opdracht werd gegeven.

- Controleer de verdeling van de prioriteiten met behulp van de opdracht 'status full'. Negatieve waarden worden uitgevoerd als een getal dat ontstaat door die waarde op te tellen bij 256. Als de vermelde waarde 255 is, is de werkelijke waarde dus -1.
- Met CHANGETASKPRI hebt u alleen invloed op CLI-processen, dus niet op taken die u start vanaf de Workbench. In paragraaf 7.3 presenteren we een nieuwe CLI-instructie (TaskPri). Die kan de prioriteit van alle soorten taken veranderen. Dat geeft nog meer invloed op de multitasking van de Amiga.

5.7 Punten ter overweging

Multitasking biedt veel voordelen, maar vereist een doordacht gebruik. In deze paragraaf bespreken we een aantal voetangels en klemmen.

Het volgende voorbeeld kan problemen veroorzaken:

```
cd df0:
run dir
run list
```

Twee processen in multitasking die beide de diskdrive gebruiken, kunnen de schrijf- en leeskop van de drive als een razende heen en weer sturen tussen ver van elkaar liggende sporen. Dat klinkt zeer onheilspellend. Voor de hardware is het niet zo'n probleem; het grootste nadeel is de verspilling van tijd. Soms duurt de uitvoering in multitasking langer dan wanneer de opdrachten na elkaar zouden zijn uitgevoerd. (Het voorbeeld is bovendien onzinnig, omdat de uitvoer door elkaar loopt in één venster.)

Dubbel gebruik van de diskdrive levert vaak problemen op. Eén opdracht leest gegevens op de diskette, terwijl een andere opdracht de diskette nodig heeft om een instructie te laden:

```
run copy df0:tekst prt:
cd df0:
```

Omdat er voor deze taken niet zo veel hoeft te worden gelezen, zal de schrijf- en leeskop minder lang heen en weer gaan. Deze combinatie van opdrachten kan eigenlijk alleen soepel worden afgehandeld met twee diskdrives, als de opdrachten elk een andere drive aanspreken. Het maakt niet uit of u daar een externe drive voor gebruikt of een RAM-disk.

In principe is het ook problematisch als twee opdrachten hetzelfde bestand nodig hebben. Dit geldt niet voor dubbel lezen. Een eenvoudig experiment bewijst dat. Open een extra CLI en voer met TYPE in beide vensters het bestand uit. Het is echter logisch dat het bij parallel beschrijven spaak loopt. Als de ene opdracht een bestand heeft geopend, weigert AmigaDOS daarom een andere opdracht de toegang tot dat bestand. Dat voorkomt het lezen van ongeldige gegevens. Zie het volgende experiment:

```
copy df0:s/startup-sequence ram:bestand
cd ram:
copy bestand bestand1
```

Probeer nu tegelijk in 'bestand1' te schrijven en ook in 'bestand1' te lezen:

```
run type >bestand1 bestand
```

type bestand1

Al vlug verschijnt melding 'Can't open bestand1'. Vraag de reden op met de instructie WHY. AmigaDOS meldt:

```
Last command failed because object in use
```

In de omgekeerde volgorde bestaat hetzelfde probleem. Als de ene opdracht een bestand leest, kan een andere opdracht dat bestand niet openen voor schrijven:

```
type >bestand1 bestand
```

De foutmelding is: 'CLI error: Unable to open redirection file'. Het voorbeeld is een beetje ver gezocht, maar zulke opdrachten komen voor in zelfgemaakte programma's en opdrachten. Als u een BASIC-programma schrijft dat een bestand opent voor schrijven en u breekt het programma af zonder dat bestand te sluiten, kan geen enkel ander proces dat bestand nog gebruiken. Gelukkig sluit AmigaBASIC automatisch alle geopende bestanden zodra u AmigaBASIC verlaat. Verlaat nooit een C-programma zonder expliciet alle bestanden af te sluiten.

Ten slotte komen we terug op de toekenning van een nieuwe prioriteit. Kies liever geen waarden kleiner dan -5 of groter dan +5. Als u een waarde kiest groter dan 5, krijgt de opdracht een hogere prioriteit dan het TrackDisk.Device dat het gebruik van de diskdrives regelt. Een programma met te hoge prioriteit kan de computer laten hangen. Onder multitasking mag een programma geen wachttijden genereren met lussen of onzinnige opdrachten. Voor pauzes bestaan speciale routines die tijdens het wachten andere processen aan de beurt laten komen. We illustreren dit probleem met twee voorbeelden.

Waarschuwing: Als u onderstaande voorbeelden uitvoert, hebt u de Amiga niet meer in de hand. Breng dus eerst alle bestanden en de inhoud van de RAM-disk in veiligheid.

1. Geef het actuele CLI-proces de prioriteit 50 en start AmigaBASIC met RUN. Activeer het linker venster en typ daar een tekst. Geef een Return. Waarschijnlijk is invoer via het toetsenbord dan niet meer mogelijk. Als u geluk hebt, kunt u de cursor nog wel verplaatsen met de muis. Activeer met de muis een paar vensters. Nu zit de muis ook vast. Omdat AmigaBASIC een hogere prioriteit heeft dan alle andere taken, óók alle systeemtaken, mogen die andere taken niet meer worden uitgevoerd. Maar AmigaBASIC wacht op de resultaten van de systeemroutines zonder gebruik te maken van de wachtroutines van het besturingssysteem. Het systeem is geblokkeerd, maar het hangt niet.
2. Het tweede voorbeeld maakt gebruik van de nieuwe CLI-instructie TASKPRI (zie paragraaf 7.3). Start het programma 'Lines' in de directory 'Demos'. Geef de opdracht:

```
taskpri lines 50
```

Activeer het venster Lines. Invoer via toetsenbord en muis blijken nu niet meer mogelijk.

Hopelijk verduidelijken deze voorbeelden voldoende hoe ingrijpend de wijziging van een prioriteit is. Als u geen waarden gebruikt die groter zijn dan 5 of kleiner dan -5, komen de onmisbare systeemtaken ook regelmatig aan de beurt.

6 AmigaDOS intern

Als u zelf programma's gaat schrijven, is het van belang te weten hoe de CLI en AmigaDOS in elkaar zitten. We hebben daarom wat speurwerk verricht naar de werking van de routines. We beschrijven de interne structuur van de CLI-instructies en de manier waarop die instructies gebruik maken van de 'Global Vector Table'. Deze vectoren zijn onder meer verantwoordelijk voor de boodschap 'gv 150' na de opdracht 'status full'.

Dit boek heeft niet de pretentie alle gegevens over de Amiga te bevatten. We gaan daarom niet diep in op de structuren van taken, message ports, enzovoorts. Daarvoor verwijzen we naar het boek 'Amiga Intern' van DATA BECKER NEDERLANDS*.

In de rest van dit hoofdstuk veronderstellen we een degelijke kennis van de taal C en van assembler. We hebben het hoofdstuk zo opgezet dat een onervaren Amiga-gebruiker in ieder geval enig idee krijgt van de hoofdzaken van de procedures.

6.1 Programma's, segmenten en BPTR-pointers

Omdat het besturingssysteem van de Amiga multitasking is, mogen de adressen waar programma's worden geladen, niet vaststaan. Anders zou een nieuw programma niet kunnen worden geladen, als op die plaats al een ander programma in het geheugen staat. Als AmigaDOS een programma laadt, zoekt het systeem daarom eerst een voldoende groot geheugengedeelte en past daarna de adressen aan. Men noemt dit herplaatsbare code (relocatable code).

U hoeft zelf niet te zorgen voor de aanpassing van de adressen, want de C-compiler en de assembler slaan zelf die onmisbare informatie op de diskette op. De schuivende adressen maken het wel heel lastig te doorzien hoe een voltooid programma in elkaar steekt. Elke keer dat u een programma laadt om het te onderzoeken met een monitor of een debugger, komt het op een andere plaats terecht. Aantekeningen van de ene en de andere keer kunt u dus niet samen interpreteren.

Het pakket PROFIMAT van DATA BECKER kan u verder helpen. Dit pakket bestaat uit een assembler, een debugger en een reassembler. Dit systeem heeft het voordeel dat u na het te onderzoeken programma in de debugger te hebben geladen met de reassembler een soort source-bestand van het programma kunt maken. Dat bestand is veel gemakkelijker te documenteren of uit te breiden. Meestal kan de assembler het 'source-bestand' zonder moeilijkheden weer in een werkend programma terugvertalen.

Segmenten

Doordat het besturingssysteem van de Amiga een aantal programma's tegelijk beheert, ontstaan er moeilijkheden met het vrije geheugen. De niet-gebruikte geheugengedeeltes zijn meestal flink verbrokkeld. Tussen de lege gedeeltes in staan gedeeltes waar zich programma's bevinden. Als in een vrij geheugen van 500 Kb het grootste vrije gedeelte 80 Kbyte is, zou het onmogelijk zijn een programma van 90 Kbyte te laden, hoewel er in totaal misschien nog wel genoeg ruimte is. Het besturingssysteem verdeelt een programma daarom in een aantal segmenten. AmigaDOS past de adressen in de onderdelen van het programma aan aan de positie van de andere segmenten. Met opdrachten aan de compiler of de assembler geeft u zelf aan hoe u het programma wilt indelen.

Voor elk geladen programma stelt AmigaDOS een lijst op van de koppeling van de segmen-

ten. Voor elk segment staan twee longwords (BPTR-pointers). Het eerste longword wijst naar het adres van het volgende segment, het tweede bevat de lengte van het segment. Met die segmentenlijst is het mogelijk alle gedeelten van het programma terug te vinden in het geheugen en die gedeelten stuk voor stuk met de reassembler terug te vertalen tot een source-programma.

BPTR-pointers

In C-programma's, include-bestanden en in de documentatie bij het besturingssysteem van de Amiga komt u steeds het begrip BPTR tegen. De BPTR-pointers zijn een gevolg van het feit dat AmigaDOS werd ontwikkeld in BCPL. In C wijst een pointer rechtstreeks naar de geheugenlokatie van het object. Een BPTR-pointer is een factor vier kleiner. Als een pointer wijst op een object vanaf geheugenlokatie 10000, wijst de BPTR naar 2500. Daarbij gaat geen informatie verloren, want BPTR-pointers wijzen altijd op objecten die alleen op even longword-adressen mogen staan. Als zo'n adres door vier wordt gedeeld en later weer met vier vermenigvuldigd, is het resultaat hetzelfde als het uitgangspunt.

De pointers naar het volgende segment in de segmentenlijst zijn BPTR-pointers. Omdat segmenten alleen op even longword-adressen beginnen, zijn de werkelijke beginadressen de waarden van de BPTR-pointers vermenigvuldigd met vier.

6.2 Het programma CLI

Meestal verstaat men onder het begrip CLI de gebruikersinterface inclusief de CLI-vensters en de CLI-routines in de subdirectory 'C'. Maar er is ook een programma met de naam CLI, een startprogramma voor de gebruikersinterface. Het programma staat noch in de directory 'System', noch in de hoofddirectory van de Workbench-diskette. Met dit programma kunt u het eerste CLI-venster openen als de startup-sequence het enige venster heeft gesloten met de instructie ENDCLI. In deze paragraaf bespreken we de structuur van dat programma en de overwegingen bij het ontwikkelen ervan.

Het programma CLI is in tegenstelling tot de meeste CLI-instructies geschreven in C. Uit de opbouw van het programma blijkt hoe weinig taken het heeft.

Opbouw van het programma CLI

Aan het begin van het programma CLI staan de gebruikelijke voorbereidingen zoals die in elk C-programma voorkomen. Er is een test of het programma werd gestart vanuit de Workbench of vanuit de CLI. In het laatste geval wordt de opdracht geïnterpreteerd en ontleed in de samenstellende elementen.

'Main()', de kern van het programma, test of er al een CLI-venster bestaat. Bij een negatief antwoord worden er 10000 bytes gereserveerd in het geheugen. De prompt wordt '%N>' en de directory 'SYS:' wordt de actuele directory.

Het programma opent altijd een venster met de DOS-functie 'Open()'. De naam ervan wordt opgenomen in de opdracht: 'Con:0/50/640/80/New CLI Window'.

De DOS-functie 'Execute' verwacht drie parameters: een opdrachtstring, de input handle en de output handle. Het programma zet daar als string een nul, als input handle het zojuist geopende venster en als output handle een nul. 'Execute' genereert zelf de CIS (Command Input Stream) en de COS (Command Output Stream) naar de geaccepteerde file handle, laadt de

CLI-opdracht 'c:run' en start die. Die opdracht gaat door met lezen en uitvoeren van opdrachten in het CON-venster met de bijbehorende weergave van de uitvoer totdat de invoer 'END-CL' optreedt. CIS en COS krijgen dan hun oorspronkelijke waarde terug en 'Execute' keert terug naar het besturingssysteem.

Het programma CLI sluit het venster, geeft eventueel gereserveerde geheugenruimte vrij en keert ook terug.

Het programma zelf heeft een opvallend beperkte taak. Het meeste werk wordt gedaan door de routine van de CLI-instructie RUN. Die is dus onmisbaar. Het bewijs van die stelling is eenvoudig. Maak in de RAM-disk een directory 'C' en kopieer alleen de instructie ASSIGN daarheen. Koppel 'RAM.C' aan 'C':

```
assign c: ram:c
```

Start vanaf de Workbench het programma CLI. Het CON-venster verschijnt, maar het is snel weer weg. Dat komt doordat 'C:RUN' onvindbaar is. Zorg dus dat deze opdracht nooit wordt gewist van de Workbench-diskette.

U weet nu voldoende van de werking van het programma CLI. Met die kennis kunt u zelf een CLI-programma schrijven. Dat gaat in een aantal stappen:

- Open een CON-venster.
- Roep de DOS-functie 'Execute' op met drie parameters:
 - pointer 0 voor de opdrachtstring in D1
 - pointer naar de input handle van het CON-venster in D2
 - pointer 0 voor de output handle in D3.
- Sluit het CON-venster.

De daarvoor onmisbare opdrachten in machinetaal zijn:

```
; ----- OpenCliWindow -----
OpenCliWindow:PEA $3ED ; modus: Old_File
PEA conwindow        ; "CON:0/50/640/80/New CLI Window
JSR call_Open        ; DOS-functie OPEN oproepen
MOVE.L D0,D2         ; file handle van OPEN is 'CON:'
ADDQ.L #8,A7         ; stack herstellen
BEQ LC2C47E          ; OPEN is mislukt -> geen Execute
CLR.L -(A7)          ; = output handle
MOVE.L D2,-(A7)      ; = input handle
PEA executestring    ; $C2B8C0 = 0
JSR call_Execute     ; laadt 'C:RUN'
                    ; levert CIS en COS
                    ; wist 'RUN', herstelt CIS en COS

RTS

conwindow: dc.b "CON:0/50/640/80/Mijn CLI-venster", 0
executestring: dc.b 0,0,0,0
```

Het programma is nog niet compleet. Het moet bijvoorbeeld vooraf de DOS-bibliotheek openen. Dit voorbeeld demonstreert wel duidelijk hoe eenvoudig het is zelf een CLI-programma te

schrijven. In het voorbeeld wordt de besturing helemaal gedelegeerd aan 'c:run'. Er zit niets anders op dan te wachten totdat RUN de instructie ENDCLI leest en terugkeert.

6.3 De interne DOS-bibliotheek

Als u een programma in C behandelt met een disassembler, vindt u aan het begin van dat programma een paar standaardopdrachten. Als u een programma in machinetaal schrijft, mag u die niet weglaten:

```
move.l  pointer_naar_bibliotheek,A6
jsr     offset(A6)
tst.l   D0
beq     fout
```

Na deze opdrachtenreeks is het mogelijk routines aan te roepen uit de bibliotheken (libraries) van het besturingssysteem. In een CLI-instructie zoals RUN zult u die reeks opdrachten zelden aantreffen. In plaats daarvan ziet u steeds de onderstaande opdrachten:

```
lc188e8:move.l D1,D6 ; aantal Global Vectors opslaan (=$95)
addi.l  #$32,D1    ; tel $32 op voor AllocMem (=$C7)
suba.l  A0,A0      ; zet A0 op 0
move.l  $74(A2),A4 ; = Allocate Memory (MEMF_PUBLIC)
                          ; (lengte=D1), Return BPTR(D1)

moveq   #$C,D0
jsr     (A5)        ; AllocateMem
tst.l   D1          ; geheugen gereserveerd voor vectortabel?
beq.l   LC189DE    ; nee -> einde
add.l   #$32,D1    ; ja, verder
```

Dit is een klein stukje van de CLI-instructie RUN. Let op een drietal opdrachten:

```
move.l  offset(A2),A4
jsr     (A5)
tst.l   D1
```

In het programma RUN vindt u weliswaar diverse offsets, maar steeds wordt via de opdracht 'jsr(A5)' dezelfde routine opgeroepen. Er is blijkbaar een interne bibliotheek. Het basisadres van die bibliotheek moet in register A2 staan; in A4 wordt het adres van de routine opgeslagen. Na nog wat voorbereidend werk roept de subroutine in A5 de routine op. We kunnen uit deze procedure twee conclusies trekken:

- CLI-routines zijn kort en compact, in ieder geval korter dan vergelijkbare C-programma's.
- CLI-routines moeten meer informatie laden dan een C-programma, bijvoorbeeld het adres van de interne bibliotheek. De Workbench levert dat adres niet. Het is dus niet mogelijk een CLI-opdracht te starten vanaf de Workbench. Als u aan de routine RUN een icon geeft en hem daarna activeert, verschijnt er een Guru die zelfs met een debugger niet meer onder controle is te krijgen.

In paragraaf 6.4 gaan we dieper in op de manier waarop CLI-instructies werken met de Global Vector Table. Nu kijken we alleen naar het feit dat de meeste CLI-instructies deze tabel eerst kopiëren naar een vrij geheugengedeelte en daar een aantal specifieke routines toevoegen. Elke CLI-instructie heeft dus een andere tabel. De standaardtabel bevat \$96 boekingen (\$00 tot en met \$95). Zie het commentaar bij de eerste regel van het hierboven afgedrukte programmageedeelte.

Als u met de opdracht 'status full' de gegevens over een CLI-proces opvraagt, vindt u daarbij de boeking 'gv'. Meestal heeft die de waarde 150 (het decimale equivalent van \$96). De variabele 'gv' bevat de lengte van de Global Vector Table van een proces. U hebt niet veel aan dat gegeven; wij hebben nog nergens een verwijzing gevonden naar dat getal. De tabel heet 'global' omdat ze beschikbaar is voor alle CLI-processen. Het afgedrukte gedeelte van de routine RUN omvat de voorbereiding voor het kopiëren van de tabel. In dat gedeelte wordt het benodigde geheugen gereserveerd.

De CLI-instructies halen bij hun oproep nog veel meer gegevens uit de registers van de processor. Dat bespaart veel rekenwerk als u een belangrijke variabele wilt gebruiken. De registers bevatten de volgende gegevens:

D0	aantal tekens voor de parameter in de opdrachtregel
D2	grootte van de stack van het programma
A0	pointer naar een teken van de parameter
A2	pointer naar de interne DOS-bibliotheek
A5	pointer naar de routine voor de oproep van functies
A6	pointer naar de terugsprongroutine.

Als u de instructie RUN zonder parameters oproept, is het resultaat als volgt (de DOS-functie 'Execute' doet dat ook):

```

;-----Voorbeeld van waarden in de registers -----
; D0 = 00000001 -> alleen de naam RUN
; D1 = 00C5AA7C
; D2 = 00000FA0      = 4000 = progr.-stack bij oproep in CLI
; D3 = 00000FA8
; D4 = 00000001 -> lengte van de parameterstring
; D5 = 0000003E
; D6 = 00309107
; D7 = 00C5A794

; A0 = 00C27411 -> pointer: 1e parameter achter 'RUN' (010A)
; A2 = 00C04CA0 -> pointer naar de DOS-bibliotheek
; A3 = 00C6F52C
; A4 = 00C6BB48
; A5 = 00FF44B4
; A6 = 00FF44A8 -> routine die de oproep van de functie afsluit
; A7 = 00C721C4 -> terugsprongadres

```

De manier waarop de parameterstring wordt opgeslagen, wijkt af van de conventie in C een string af te sluiten met een nul. Hier bevat de eerste byte de lengte van de string, en daarachter staan de tekens.

In het voorbeeld luidt de string:

\$01,\$0A

Dat is één teken: een Return. Pointers naar strings die op die manier worden opgeslagen, hebben de soortnaam BSTR. De waarde van zulke pointers moet u voor gebruik met vier vermenigvuldigen.

Er is alle reden de Global Vector Table te gebruiken bij de constructie van eigen C-opdrachten, mits u zich houdt aan de oproepconventies van de beschikbare CLI-instructies. Het is zeer onwaarschijnlijk dat Commodore iets gaat veranderen in de tabel, want dan zou geen van de meegeleverde CLI-instructies meer functioneren.

6.4 Voorbeeld: RUN

Aan de hand van RUN zullen we nagaan hoe de DOS-instructies in elkaar zitten. De meeste aandacht besteden we aan de manier waarop eigen vectoren worden toegevoegd aan de kopie van de Global Vector Table (GVT). De GVT bevat zelf kant en klare routines. Dat maakt het makkelijker andere CLI-instructies te analyseren en trucjes af te kijken als u zelf programmeert.

De meeste CLI-opdrachten beslaan twee segmenten (zie paragraaf 6.1). In het eerste segment staan de voorbereidingen voor de afhandeling van het programma zelf:

- Een lus onderzoekt na elkaar alle segmenten van het programma om de benodigde lengte van de Global Vector Table (GVT) te kunnen berekenen. Het laatste longword in de segmenten definieert die grootte. De verstekwaarde is \$95 = 149, omdat in de tabel die in het besturingssysteem aanwezig is, 150 vectoren staan. Als de test een hogere waarde oplevert, wordt de grootte van de GVT aangepast.
- Dan volgt de berekening van de benodigde geheugenruimte voor de GVT. Voor andere gegevens worden nog \$32 longwords extra gereserveerd met 'AllocMem'.
- Een kopie van de GVT in het ROM gaat naar het vrije geheugen. Dat gebeurt met een interne routine uit de DOS-bibliotheek:

```
move.l $70(A2),A4 ; = 'Fill in a Global Vector Table'  
moveq #$C,D0  
jsr (A5) ; voer de routine uit
```

Die routine wordt tweemaal opgeroepen, omdat de GVT op twee verschillende plaatsen staat. Daarna worden uit de CLI-structuur van de actuele taak 15 longwords gekopieerd naar het vrije geheugen. Deze structuur staat in de GVT vanaf \$218. In de nieuwe tabel komt hij ook op die positie te staan. Voor het programma in het tweede segment wordt de routine nog eens opgeroepen. De routine 'Fill in a Global Vector Table' verwacht in D1 een BPTR naar de segmentenlijst van het programma en levert als resultaat in D0 -1, als alles in orde is. Anders is het antwoord 0.

De structuur van de segmenten heeft een vast patroon. Het laatste longword van een segment bevat de maximale lengte van de tabel (het hoogst mogelijke nummer van de vectoren). Vóór dat longword staat de offset van de toe te voegen vector, gerekend vanaf het begin van het segment. Daar weer voor staat het nummer van de vector. Een offset van nul

beëindigt de procedure. Als voorbeeld geven we het einde van het tweede segment van RUN:

```
$00000000 $00000001 $00000024 $00000096
einde      vectornr. offset    max.grootte van de tabel
```

In dit geval voegt de routine de vector met het nummer 1 toe. Het adres van de subroutine bij die vector is 'segment2 + \$24'. Het getal \$00000000 als offset vormt het einde van dit segment. Omdat er geen andere segmenten zijn, is de routine 'Fill in a Global Vector Table' daarmee klaar.

- Vervolgens wordt over vector 1 het adres geschreven van een routine in het eerste segment die het geheugen met de nieuwe vectortabel weer vrijgeeft en de CLI-instructie beëindigt. Dat adres kon de kopieerroutine natuurlijk niet zelf in de tabel opnemen. Ten slotte volgt de oproep van vector 0. Die wijst naar het begin van het programma in het tweede segment. Op die manier gaat het besturingssysteem verder met de opdrachten in het programma. De programmacode van het tweede segment begint niet op nul (de nieuwe vector is dus ook niet nul). Het programma begint op \$24, omdat daarvoor de naam van de CLI-instructie staat. Het begin ziet er als volgt uit:

```
;-----
; RUN Segment 2
Offset: $00
L00C1A610:DC.B $30,$39      ; "09"
DC.B $11,$52,$55,$4E      ; ".RUN"
DC.B $20,$20,$20,$20
DC.B $20,$20,$20,$20
DC.B $20,$20,$20,$20
DC.B $20,$20,$00,$00      ;$20,$20
DC.B $07,$73,$74,$61      ; ".sta"
DC.B $72,$74              ; "rt"

Offset: $24                ; begin van de code van vector 0
```

De belangrijkste routine van vector 0 is heel kort, net als het CLI-programma. Ze begint met voorbereidingen zoals de oproepen van 'Input()', 'Output()' en 'AllocMem()'. Het essentiële gedeelte is:

```
LC1A860:MOVE.L $3C(A1),$68(A1)
LEA $3F4(A4),A3
MOVE.L A3,D4              ;haal pointer naar naam van de TASK NODE
LSR.L #2,D4              ;wijzig BPTR: TASK-NODE naam naar pointer
MOVE.L $34(A1),D3        ;haal prioriteit van het proces
MOVE.L #$320,D2          ;grootte van de stack in longwords
MOVE.L $20(A1),D1        ;BPTR op segmentenlijst van het proces
MOVEQ #$58,D0
MOVE.L $84(A2),A4        ; 'Create a Process'
;                        D1 = BPTR to SegList array for process
;                        D2 = Stack for process in long word
```

```

;          D3 = Priority of Process
;          D4 = BPTR to task node name
JSR (A5)
MOVE.L D1,$1C(A1) ; pointer naar het proces 'Message Port'
TST.L D1          ; proces gegenereerd?
BNE.S LC1A892    ; nee -> 'Can't Create Background Task'
BRA.L LC1A9C0    ; ja -> verder, geheugen vrijmaken, einde

```

Het komt erop neer dat de CLI-instructie RUN een nieuwe taak genereert met de naam 'Background Task'. Die naam kunt u definiëren met de instructie TASK (zie paragraaf 7.2).

Hopelijk blijkt uit de gegevens in dit hoofdstuk hoe krachtig AmigaDOS is en hoe u DOS-routines kunt toepassen in eigen programma's. Als u weet hoe instructies zoals ASSIGN en ADDBUFFERS werken, kunt u betere instructies samenstellen en het werk met AmigaDOS en de CLI zodoende makkelijker en produktiever maken.

7 Nieuwe CLI-instructies

Een van de sterke punten van het besturingssysteem van de Amiga is dat het bijna onbeperkt kan worden uitgebreid. U mag nieuwe bibliotheken toevoegen en met extra devices nieuwe apparaten aansturen. Omdat de CLI-instructies niet ontoegankelijk zijn verstrengeld in het besturingssysteem, maar als kleine programma's op de Workbench-diskette staan, kunt u altijd instructies toevoegen. De CLI-instructies staan in de directory 'C' van de Workbench-diskette. AmigaDOS zoekt altijd als laatste in de directory met de naam 'C:'. Bij de start van de computer wordt aan deze directory de archiefla 'C' van de Workbench-diskette toegekend. Langs deze omweg is elke CLI-instructie te vinden.

Nergens wordt het aantal instructies bijgehouden of beperkt. U mag uw favoriete CLI-instructie zo vaak onder andere namen kopiëren naar de directory 'C' als u wilt (of althans totdat de requester 'Disk Full' op het scherm verschijnt). Het is handig veel gebruikte instructies nog eens op te nemen onder een makkelijke korte naam: X voor EXECUTE, FC voor FILECOPY, enzovoorts. Behalve nieuwe instructies mag u elk willekeurig programma opnemen in de directory 'C'. Omdat een instructie een klein programma is, is er in theorie niets tegen het programma Clock naar 'C' te kopiëren. Als u de instructie LIST geeft om de inhoud op te vragen, blijkt uit de grote hoeveelheid gebruikt geheugen dat Clock een heel ander soort programma is dan de CLI-instructies. De meest gebruikte zijn opvallend kort en kunnen dus snel worden geladen in het werkgeheugen. Een BASIC-compiler is daarom niet erg geschikt om instructies te maken.

Een programmacode die net zo compact is als die van de originele CLI-instructies, kunt u met de Amiga alleen maken met een assembler. De Amiga ondersteunt de brontaal van de instructies niet (BCPL). De daaruit ontwikkelde taal C is echter een goed alternatief. (Het besturingssysteem is voor een groot gedeelte geprogrammeerd in C.) Het is niet mogelijk in C de kenmerkende programmastructuur van een CLI-instructie na te maken. Houd bij de manier van programmeren goed rekening met de aard van de CLI-omgeving. Die heeft ten minste de volgende eigenschappen:

1. Alle CLI-instructies zijn 'niet-interactief'. Dat betekent dat ze na hun start geen nieuwe gegevens meer nodig hebben. Parameters (voor zover aanwezig) geeft u meteen bij de oproep een waarde. Ze staan in de vorm van een lijst meteen achter de instructie (bijvoorbeeld: 'copy teksten ordner').
2. De instructies sturen hun uitvoer normaal gesproken naar het CLI-venster van waaruit ze werden opgeroepen. Bij een gewone oproep verschijnt er nooit een apart venster voor uitvoer.
3. De echte CLI-instructies bevatten allemaal een Help-functie. Activeer die met een vraagteken. Als u verkeerde parameters opgeeft bij de functie, volgt daar meestal een reactie op. Voorbeelden:

a) Help-functie:

invoer: date ?
uitvoer: TIME, DATE, TO=VER/K:

b) zinvolle foutmelding:

invoer: date vandaag
uitvoer: ***Bad args
- use DD-MMM-YY or <dayname> or yesterday etc.
to set date, HH:MM:SS OR HH:MM to set time

7.1 Programmeren in C

Als een nieuwe instructie goed is geprogrammeerd in C, kan een buitenstaander die niet onderscheiden van een standaardinstructie. In C geeft u de parameters ook op in de vorm van een lijst; C kent bovendien het omleidingsteken (>).

Zelfs als de nieuwe instructie geen gegevens nodig heeft om uitgevoerd te kunnen worden, moet het systeem de invoerregel interpreteren. Als iemand in zo'n geval wel iets achter de instructie schrijft, geeft dat blijk van onvoldoende kennis van de bedoeling van de instructie. De uitvoering ervan kan dan beter achterwege blijven.

De Amiga interpreteert een invoerregel in C in een aantal stappen. De ingevoerde gegevens worden in de vorm van twee parameters overgedragen aan het centrale gedeelte van de instructie ('Main'). De eerste parameter heet meestal 'ArgC' (Argument Counter), is van het type 'int' en bevat het aantal overgedragen argumenten. De naam van het programma telt daarbij ook mee. 'ArgC' heeft dus minstens de waarde één. De tweede parameter heet meestal 'ArgV' (Argument Vector). Hij moet worden gedeclareerd als een veld met pointers naar het gegevenstype 'char'. De elementen van dit veld zijn pointers naar de parameters van de invoerregel. Elke parameter eindigt op een nul.

Deze voorbereidingen worden getroffen door een gedeelte van het programma dat de linker bij het samenstellen van de programmacode altijd toevoegt. Dit is dus geen taak van de CLI, zoals u misschien dacht op grond van de naam 'Command Line Interface'.

De eerste regels van 'Main' zijn meestal:

```
main(argc, argv)
int argc;
char *argv[];
{
...

```

Omdat 'ArgC' en 'ArgV' gegevens zijn van buiten het programmagedeelte 'Main', moeten ze buiten de accolades worden gedeclareerd als parameters van het juiste type.

Hieronder staat een uitgewerkt voorbeeld van de programmastructuur die voldoet aan de eisen van de CLI. We hebben alle C-programma's vertaald met de compiler van Aztek. Een andere compiler werkt echter net zo goed.

```
/* Programma: interpretatie */

main(argc, argv)
int argc;
char *argv[];
{
    int i;
    printf (" Aantal: %d \n", argc);
    for ( i = 0; i < argc; i++)
    {
        printf (" Nr.: %d, argument: %s \n",i , argv[i]);
    }
}

```


Roep dit programma vanuit de CLI op met de opdracht:

```
interpretatie een nogmaals alweer parameter
```

De uitvoer wordt:

```
Aantal: 5
```

```
Nr.: 0 , Argument: interpretatie
Nr.: 1 , Argument: een
Nr.: 2 , Argument: nogmaals
Nr.: 3 , Argument: alweer
Nr.: 4 , Argument: parameter
```

Een spatie in de invoer geldt als scheidingsteken tussen de parameters. Als u vanuit de CLI een tekststring met spaties wilt doorgeven aan een C-programma, komt die in brokken over. Met een kleine routine in het programma kunt u ze weer samenvoegen. Het is echter veel eenvoudiger de tekststring tussen dubbele aanhalingstekens te zetten. De oproep:

```
interpretatie "een nogmaals alweer parameter"
```

heeft als resultaat:

```
Aantal: 2
```

```
Nr.: 0 , Argument: interpretatie
Nr.: 1 , Argument: een nogmaals alweer parameter
```

Alles verloopt net zoals bij de standaardinstructies van de CLI. Ook de tekens '>' en '<' gelden op de bekende manier als symbolen voor de omleidingsfuncties. Probeer dat uit met een testje:

```
interpretatie >df0:bestand parameter
```

De uitvoer gaat naar het gewenste bestand en niet naar het scherm in de trant van:

```
Aantal: 2
```

```
Nr.: 0 , Argument: >df0:bestand
Nr.: 1 , Argument: parameter
```

Het lukt echter nog niet de invoer om te leiden met behulp van het teken '<'. Dat ligt aan het programmagedeelte in C dat de invoerregel interpreteert. Het gaat niet uit van het actuele invoerapparaat, maar altijd van het toetsenbord.

Het is wel mogelijk gegevens over te nemen vanuit andere apparaten. Dat gaat als volgt:

```
#include <stdio.h>
FILE * Input(); /* Declaratie van een externe functie */
```

```

main(argc, argv)
int argc;
char *argv[];
{
    int i;
    FILE *invoer; /* pointer naar structuurtype FILE */
    char buffer[100];
    long lengte;
    printf (" Aantal parameters: %d\n", argc);
    for ( i = 0; i < argc; i++)
        printf (" Argument: %s\n", argv[i]);
    invoer = Input();
    gelezen = Read (invoer, &buffer[0], 30L);
    buffer [lengte] = 0;
    printf (" Gelezen: %s\n", &buffer[0]);
}

```

Maak een bestandje:

```
echo >bestand "een twee drie"
```

Start het bovenstaande programma met de opdracht:

```
invoer: <bestand hallo
```

De uitvoer op het scherm is:

```

Aantal parameters: 2

Argument: invoer
Argument: hallo
Gelezen: een twee drie

```

De betekenis van 'ArgC' en 'ArgV' verandert niet. De omleiding '<bestand' telt niet mee als parameter (ArgC = 2). Na de uitvoer van de gewone parameters activeert de functie 'Input()' het standaardinvoerapparaat voor het oproepende programma (de CLI). De functie 'Input()' staat in de DOS-bibliotheek. De variabelen in het programma hebben de volgende betekenis:

```

invoer: pointer naar de standaardinvoer
buffer: buffer voor de gelezen tekens
lengte: aantal werkelijk gelezen tekens

```

De instructie '<' in een CLI-regel leidt de standaardinvoer inderdaad om vanuit het daar vermelde apparaat.

7.2 TASK

Met de instructie STATUS kunt u in de CLI opvragen wat de Amiga doet als hij niet volledig bezet is. De geleverde informatie is echter niet compleet, want de programma's die zijn gestart vanuit de Workbench, ontbreken. Dat geldt ook voor de interne activiteiten van de Amiga. Kennis hiervan is onmisbaar als u er invloed op wilt kunnen uitoefenen.

De door ons ontwikkelde instructie TASK geeft niet alleen een overzicht van alle actieve taken, maar vermeldt ook hun prioriteit. Dat geeft u een idee van het onderling belang. Met de nieuwe instructie TASKPRI (zie paragraaf 7.3) kunt u de prioriteiten wijzigen.

Het besturingssysteem van de Amiga werkt met relatieve adressen. Geen enkel adres ligt vast, alles mag overal staan. Zoals we zagen, bestaan alle programma's uit herplaatsbare code die op een willekeurig adres mag worden geladen. Het enige vaste adres is het startadres van *Execbase*, het startadres van de basistabel van het besturingssysteem van de Amiga. Dit adres staat altijd op geheugenlocatie \$000004. *Execbase* zelf staat bij een Amiga 500 met geheugenuitbreiding al op een andere plaats dan bij een Amiga zonder zo'n uitbreiding. In *Execbase* staat een groot aantal belangrijke pointers naar alle devices, bibliotheken, enzovoorts. De adressen van de lijsten die de taken administreren, staan daar ook. Een taak kan drie statussen hebben:

Running	De taak wordt op dit moment uitgevoerd.
Ready	De taak is klaar om te worden uitgevoerd.
Waiting	De taak wacht op een bepaalde gebeurtenis. Eerder wordt ze niet uitgevoerd.

Er kan maar een taak tegelijk de status 'Running' hebben, maar een onbeperkt aantal mag de status 'Ready' of 'Waiting' hebben. Zodra de gebeurtenis plaatsvindt waarop een taak in de Waiting-lijst wacht, zet het besturingssysteem die taak automatisch in de Ready-lijst. De structuur van die lijsten is niet eenvoudig. We kunnen er hier niet verder op ingaan. De instructie TASK levert alleen een opsomming van de taken met de status 'Waiting'. Dat is geen ernstige beperking; er is maar één taak met de status 'Running' en zelden meer dan één met de status 'Waiting'. Voor meer informatie verwijzen we naar het boek 'Amiga Intern'.

Dit is de listing van de CLI-instructie TASK:

```
#include <exec/types.h>
#include <exec/execbase.h>
#include <exec/tasks.h>
#include <exec/exec.h>
#include <exec/execname.h>
#include <exec/lists.h>
extern struct ExecBase *SysBase;
main()
{
    struct Task *task;
    char *names [20];
    int pri [20];
    int count,i;
    count = 0;
    Disable();
    for (task = (struct Task *)SysBase->TaskWait.lh_Head;
```

```

    task->tc_Node.ln_Succ;
/* einde als de pointer naar de volgende boeking nul is */
    task = (struct Task *)task->tc_Node.ln_Succ)
    {
        names[count] = task->tc_Node.ln_Name;
        pri[count++] = task->tc_Node.ln_Pri;
    }
    Enable();
    for (i = 0; i < count; i++)
    {
        printf ("%s",names[i]);
        printf ("\r\t\t\t%d\n",pri[i]);
    }
}

```

Beschrijving van het programma

Om te beginnen worden de benodigde header files aan de source gekoppeld. Daarna volgt de definitie van een pointer met de naam 'SysBase'; hij wijst naar ExecBase. Het centrale deel van het programma definieert de variabelen. De eerste is een pointer naar de Task-structuur. Deze structuren hoeft u niet zelf te maken; ze staan al in de task-lijsten. De inhoud van de Task-structuren verandert voortdurend, zodat er geen tijd is de namen uit te voeren naar het scherm. Ze worden daarom zo snel mogelijk opgeslagen in een veld. Terwijl dat gebeurt, blokkeert de functie 'Disable()' de task switching. Vanaf dat moment is het programma TASK de enige actieve taak en verandert er niets meer aan de lijsten. Dat geeft het programma de gelegenheid de lijsten te lezen. Dat gaat door totdat een nul het einde aanduidt:

```

    task->tc_Node.ln_Succ;
/* einde als de pointer naar de volgende boeking nul is */

```

Daarna mag de task switching weer aan met 'Enable()'. De uitgevoerde lijst is dus een momentopname. Ondertussen kunnen de namen in de lijst al weer zijn veranderd, want het programma slaat alleen de pointers naar de namen op. Als u met de kopieerfunctie 'STRNCPY' de namen zelf wilt laten opslaan, gebruikt het programma onnodig veel tijd en geheugenruimte. Het programma beperkt het aantal boekingen tot twintig, maar dat is meestal ruim voldoende.

Een voorbeeld van de uitvoer, aangenomen dat de instructie TASK is gegeven in het derde CLI-venster:

Background CLI	0
CON	5
CON	5
New CLI	0
New CLI	0
Workbench	1
File System	10
input.device	20
trackdisk.device	5

File System	10
trackdisk.device	5
CON	5

De taak met de naam 'Background CLI' is in dit geval verantwoordelijk voor het programma Textomat dat met RUN werd gestart vanuit de CLI. De drie taken met de naam CON verzorgen de invoer en de uitvoer via de vensters, inclusief die van Textomat. De taken 'New CLI' vinden hun ontstaan elk in een CLI-proces dat werd gestart met NewCLI. De Workbench is zelf ook een taak. Uit de overige gegevens valt af te leiden dat de configuratie twee diskdrives omvat. Er zijn twee taken met de naam 'File System' en twee 'Trackdisk.devices'.

7.3 TASKPRI

TASKPRI is ontwikkeld uit de instructie CHANGETASKPRI (zie paragraaf 5.6). CHANGETASKPRI kan alleen een prioriteit veranderen in de actuele CLI. TASKPRI verandert vanuit een CLI-venster desgewenst niet alleen de prioriteit van een willekeurig proces, maar van alle aanwezige taken. Dit programma is een beetje langer dan het voorgaande, omdat het uw invoer moet interpreteren (de naam en de nieuwe prioriteit van de taak).

```
#include <exec/types.h>
#include <exec/execbase.h>
#include <exec/tasks.h>
#include <exec/exec.h>
#include <exec/execname.h>
#include <exec/lists.h>
extern struct ExecBase *SysBase;
long FindTask();
int settaskpri();
main(argc, argv)
int argc;
char *argv[];
{
    struct Task *MijnTask;
    long pri;
    int resultaat;
    char oldpri;
    long task;
    pri = 0L;
    if (argc != 3)
    {
        printf(" tasknaam prioriteit\n");
        exit(FALSE);
    }
    resultaat = Value(argv[2], &pri);
    if (resultaat == FALSE)
    {
        printf (" -128 <= Taskpri <= 127, niet: %ld\n", pri);
```

```

        exit(FALSE);
    }
    task = FindTask (argv[1]);
    if (task == 0)
    {
        printf ("Task <%s> niet aanwezig\n", argv[1]);
        exit(FALSE);
    }
    oldpri = ( char ) SetTaskPri (task, pri);
    printf ("Oude prioriteit was %d, de nieuwe is %ld\n", oldpri, pri);
    exit (TRUE);
}
/* omrekening van getal van drie tekens naar een getal tussen 0 en 255 */
Value(string, waarde)
char *string;
long *waarde;
{
    int i;
    int aantteken;
    char teken;
    int eenh;
    int tiental;
    tiental = 1;
    aantteken = 0;
    if (*string == '-') /* negatief getal */
    {
        zehner = -1;    /* in negatief getal omzetten */
        string++;      /* minteken overslaan */
    }
    for (i = 1; i <= 4; i++) /* aantal tekens controleren */
    {
        if (*string == 0)
            break;
        string++;
        aantteken++;
    }
    if ((aantteken == 0) || (aantteken > 3)) /* maar drie cijfers */
        return (FALSE);
    for (i = aantteken; i > 0; i-- )      /* getal maken */
    {
        string--;
        teken = *string; /* string van achteren naar voren lezen */
        if (teken < '0' || teken > '9')
            return (FALSE);
        cijfer = teken - '0';
        *waarde = *waarde + cijfer * tiental;
        tiental = tiental * 10;
    }
}

```

```

    }
    if (*waarde < -128 || *waarde > 127)
        return (FALSE);
    return (TRUE);
}

```

Beschrijving van het programma

De eerste zeven regels van de listing zijn gelijk aan die van TASK. Daarna volgt de declaratie van de externe functie 'FindTask()' in de Exec-bibliotheek. Deze functie zoekt in de Task-lijsten naar een boeking met de opgegeven naam. De functie heeft als parameter een pointer naar de gezochte naam van de taak. Het resultaat van de functie is een pointer naar de Task-structuur als de opgegeven boeking bestaat. In tegenstelling tot veel andere functies van de Amiga maakt 'FindTask' wel verschil tussen kleine letters en hoofdletters.

Als er meer dan één taak is met dezelfde naam, geeft 'FindTask' alleen een pointer naar de eerste structuur. Alleen met een veel groter programma is dit probleem op te lossen.

De tweede externe functie is 'SetTaskPri()'. Deze functie geeft een nieuwe prioriteit aan een taak. De parameters zijn de nieuwe prioriteit (-128 tot en met +127) en de pointer naar de structuur die is geleverd door 'FindTask()'. De uitgevoerde tekst bevat ook de oude prioriteit van de taak.

Het deel 'Main' van TASKPRI maakt een consequent gebruik van de beschikbare functies in het besturingssysteem. Het is daarom makkelijker te begrijpen dan het overeenkomstige gedeelte van het programma TASK. De interpretatie van de ingevoerde parameters is geprogrammeerd volgens de voorschriften in de inleiding van dit hoofdstuk. We hebben er een controle op de syntaxis aan toegevoegd die bij ongeldige invoer een foutmelding geeft. Als niet het juiste aantal parameters is opgegeven ($\text{argc} \neq 3$), verlaat het programma het gedeelte 'Main' na uitvoer van het geheugensteuntje 'tasknaam prioriteit'. Deze melding verschijnt ook als u een vraagteken typt.

De functie 'Value()' geeft de parameters het formaat 'long'. 'SetTaskPri' kan niets beginnen met ASCII-waarden. De parameters van 'Value()' zijn een pointer naar de ingevoerde gegevens ($\text{argv}[2]$) en het adres van een variabele om de geconverteerde waarde op te slaan (&pri). Als de functie een ongeldige waarde aantreft, geeft 'Value()' als antwoord 'False' plus een melding over correcte waarden. De rest van 'Main' spreekt voor zichzelf. Als 'Findtask()' een waarde ongelijk aan nul levert, is de gezochte Task-structuur aanwezig. Dan wordt de nieuwe prioriteit geboekt en samen met de nieuwe uitgevoerd in het CLI-venster.

De noodzaak van de conversiefunctie 'Value()' is een gevolg van een typische eigenschap van C; we gaan er daarom niet verder op in. Ze zet hier alleen maar ASCII-invoer om in een getal tussen -128 en 127.

Gebruik van het programma

Oproep van de nieuwe CLI-functie is heel eenvoudig. Kies een naam uit de lijst met taken die verschijnt na de instructie TASK en definieer daarvoor een nieuwe prioriteit. Roep daarna TASKPRI op met deze gegevens. Schrijf de naam van de taak tussen aanhalingstekens als hij spaties bevat (bijvoorbeeld "File System"). Een paar voorbeelden mogen het effect van de instructie verduidelijken.

Vraag met DIR de inhoud op van Workbench-diskette in het CLI-venster. Verplaats ondertussen de muiswijzer over het scherm. De muis heeft geen last van de DIR-opdracht. Verander de prioriteit met 'TaskPri Input.device -1'. Het antwoord is: 'Oude prioriteit was 20, de nieuwe is -1'. Het lukt nu niet meer de muis soepel over het scherm te bewegen. Het input-device komt soms niet meer aan de beurt.

Een zinvoller voorbeeld heeft betrekking op een tekstverwerkingsprogramma. Neem aan dat u een brief wilt schrijven en tegelijk een lange tekst wilt afdrukken. Soms daalt de snelheid van het tekstverwerkingsprogramma zo sterk, dat er nauwelijks meer mee te werken valt. Zet de prioriteit van printer.device bijvoorbeeld op -5 en die van de tekstverwerker op +5. Het afdrukken duurt misschien een paar minuten langer, maar daar staat tegenover dat u nu ongestoord de brief kunt schrijven.

Iets dergelijks treedt op bij programmeren met editors en compilers zoals bij C. Als u tijdens het compileren wijzigingen aanbrengt in de source, reageert ED vaak zeer traag op de invoer. Verlaag daarom de prioriteit van de compiler.

7.4 TASKSTOP

Met TASKSTOP kunt u de actuele taak tijdelijk onderbreken met de linker Shift-toets. In een schuldbevuste bui denkt u misschien aan het stoppen van spelletjes als de chef langskomt, maar de instructie kan ook verhinderen dat anderen gebruik maken van uw computer terwijl u koffie haalt. Met de rechter Shift-toets ontgrendelt u de Amiga weer.

Er is maar een kort C-programma nodig om dit te bereiken:

```
#include <exec/types.h>
#include <exec/execbase.h>
#include <exec/tasks.h>
#include <exec/exec.h>
#include <exec/execname.h>
#include <exec/lists.h>
#include <functions.h>

#define SHIFTLINKS 0x3f
#define SHIFTRECHTS 0x3d
#define CONTROL 0x39

extern struct ExecBase *SysBase;
main()
{
    struct Task *task;
    char *toetsen;
    toetsen = 0xbfec01; /* adres van de bijzondere toetsen */
    printf ("\n Linker Shift = stop, rechter Shift = verder, Ctrl = einde\n ");
    task = FindTask(0L);
    if ( task == 0L)
    {
        printf (" Task niet te vinden\n");
        exit (0L);
    }
}
```



```

}
SetTaskPri (task, 127L);
do
{
if ( *toetsen == SHIFTLINKS )
{
do
{
i++;
} while((*toetsen != SHIFTRECHTS)&&>(*toetsen != CONTROL));
i++;
}
Delay(10L);
} while ( *toetsen != CONTROL );
}

```

Beschrijving van het programma

Na de koppeling aan de header files definieert het programma drie macro's. Ze krijgen elk een toets toegewezen. Het programma gebruikt ook de pointer naar Execbase. In het gedeelte 'Main' definieert het programma een pointer naar een structuur van het type Task. De pointer '*toetsen' wijst later naar het hardware-register waar de toetscodes te lezen zijn. Dat is niet zo'n fraaie methode, maar wel de eenvoudigste en snelste manier om te reageren op toetsinvoer. Op het adres '0xbfec01' is te lezen of een van de bijzondere toetsen werd ingedrukt. De pointer 'toetsen' wordt dus op dat adres gezet.

Na weergave van een melding over het gebruik van het programma haalt 'FindTask' de pointer naar de eigen Task-structuur naar de task pointer. Als de instructie (om welke reden dan ook) de structuur niet kan vinden, breekt het programma af met een melding. Als alles in orde is, krijgt de eigen taak de hoogst mogelijke prioriteit (+ 127).

In de buitenste lus vertraagt de DELAY-opdracht de uitvoering van het programma gedurende 10/50 seconde. In die tijd hebben andere taken de gelegenheid een plaats te veroveren op de Running-lijst. In die tijd heeft een druk op de Shift-toets ook geen effect. Soms reageert de computer iets vertraagd op de toetsen. Als het programma actief is en u drukt op de linker Shift-toets, komt het in een lus die niet kan worden onderbroken vanwege de zeer hoge prioriteit van het programma. Alleen de rechter Shift-toets of de Ctrl-toets haalt het programma uit de lus. Bij de Ctrl-toets geeft het programma de besturing terug en wordt TASKSTOP gewist.

De instructie TASKSTOP draait ook zonder problemen als achtergrondtaak gestart met RUN. Op die manier past ze goed in de startup-sequence van zelfladende speldiskettes.

7.5 DELAY

Uitstel is geen afstel. Dat geldt ook voor de instructie TASKSTOP. Als u een spel met de rechter Shift-toets voortzet na een blokkade via de linker Shift-toets, raast het spel meteen ongeremd verder. Met de instructie DELAY kunt u een programma een zelfgekozen snelheid geven. Het grote voordeel is dat het programma wel langzamer werkt, maar de reactie op de invoer niet.

De twee Shift-toetsen werken in dit programma niet absoluut zoals bij TASKSTOP, maar geleidelijk. Hoe langer u ze ingedrukt houdt, hoe sterker het effect. Bovendien past de instructie zich aan bij de prioriteit van het programma waarop ze werkt, ook als het programma die prioriteit tussentijds verandert. Anderzijds moet de prioriteit van de instructie zo laag mogelijk blijven om geen systeemtaken in de weg te zitten. Als de prioriteit groter is dan 20, heeft de muis daar last van. De instructie DELAY lost al deze problemen automatisch op.

De C-listing van DELAY:

```

/* Ctrl = einde, linker Shift = langzamer, rechter Shift = sneller */

#include <exec/types.h>
#include <exec/tasks.h>
#include <functions.h>
#define CONTROL 0x39
#define SHIFTLINKS 0x3f
#define SHIFTRIGHTS 0x3d
extern struct ExecBase *SysBase;
main(argc, argv)
int argc;
char *argv[];
{
    char *toets;
    struct Task *task;
    long waarde;
    long pri;
    int resultaat;
    long i;
    waarde = 1L;
    toets = 0xbfec01;
    pri = 0L;
    /* prioriteit starten met 1 */
    if ((argc != 2) && (argc != 3))
    {
        printf(" Vertraging [prioriteit] \n");
        exit(FALSE);
    }
    resultaat = Value(argv[1], &waarde);
    if(resultaat == FALSE)
    {
        printf (" 0 < wachttijd < 999, niet: %ld\n", waarde);
        exit(FALSE);
    }
    if (argc == 3)
    {
        resultaat = Value(argv[2], &pri);

        if( (resultaat == FALSE) || (pri < -127) || (pri > 127) )
        {
            printf (" -127 < prioriteit < 127, niet: %ld\n", pri);
        }
    }
}

```

```

        exit(FALSE);
    }
}
task = FindTask(0L);
if (task == 0L)
{
    printf ("Task niet te vinden\n");
    exit(0L);
}
SetTaskPri (task, pri);

do
{
    i = 0L;
    Delay( 5L );
    for ( i = 0L; i < waarde * 10L; i++ )
    {
        while ( (*toets == SHIFTLINKS ) && (waarde < 995L ) )
        {
            waarde++;
            waarde++;
            waarde++;
            waarde++;
            waarde++;
            Delay(1L);
        }
        while ( (*toets == SHIFTRIGHTS ) && (waarde > 5L ) )
        {
            waarde--;
            waarde--;
            waarde--;
            waarde--;
            waarde--;
            Delay(1L);
        }
    }
} while (*toets != CONTROL);
printf (" waarde = %ld\n", waarde);
}

```

/* Maak een getal van maximaal drie cijfers (0-999) */

```

Value(string, waarde)
char *string;
long *waarde;
{
    int i;

```

```

int aantteken;
char teken;
int cijfer;
int tiental;
tiental = 1;
aantteken = 0;
*waarde = 0L;
if (*string == '-')          /* negatief getal */
{
    tiental = -1; /* in negatief getal omzetten */
    string++;    /* minteken verwaarlozen */
}
for (i = 1; i <= 4; i++) /* aantal cijfers vaststellen */
{
    if (*string == 0)
        break;
    string++;
    aantteken++;
}
if ((aantteken == 0) || (aantteken > 3)) /* maar drie cijfers */
{
    printf (" Te veel cijfers %d\n", aantteken);
    return (FALSE);
}
for (i = aantteken; i > 0; i-- ) /* getal vormen */
{
    string--;
    teken = *string; /* String van achteren naar voren lezen */
    if (teken < '0' || teken > '9')
    {
        printf ("Verkeerd teken %c \n", teken );
        return (FALSE);
    }
    cijfer = teken - '0';
    *waarde = *waarde + cijfer * tiental;
    tiental = tiental * 10;
}
/* if (*waarde <= 999 || *waarde > 999)
    return (FALSE);
return (TRUE);
*/
}

```

Beschrijving van het programma

Het begin van het programma is gelijk aan dat van TASKSTOP. Bij de interpretatie van de invoerregel onderscheidt het programma drie gevallen:

1. U hebt niet twee of drie parameters meegegeven. Het programma breekt af onder uitvoer van de correcte oproep.
2. U hebt twee parameters meegegeven. De vertraging wordt omgerekend in een getal, maar de standaardprioriteit (0) verandert niet.
3. U hebt drie parameters meegegeven. De tweede parameter wordt ook omgerekend in een waarde, getest op grootte en toegekend aan de variabele 'pri'.

'FindTask' berekent een pointer naar de eigen Task-structuur en 'SetTaskPri' definieert de nieuwe prioriteit. Als u geen prioriteit hebt vermeld, heeft de taak altijd de prioriteit 0, onafhankelijk van het oproepende CLI-proces. De DO-WHILE lus doet het werk in deze instructie. Hij bestaat uit drie delen:

1. Oproep van de wachtfunctie DELAY. Die garandeert dat ook bij hoge prioriteit de systeemtaak genoeg tijd krijgt.
2. Een vertraging lus waarvan de duur afhangt van de vermelde vertragingwaarde. Gedurende die tijd zijn alle taken met een kleinere prioriteit geblokkeerd.
3. Een dubbel testblok voor de twee Shift-toetsen. Als u op een van de toetsen drukt, controleert het programma of de vertragingwaarde nog binnen het toegestane gebied ligt. Zo ja, dan verandert de vertragingwaarde. De opdracht DELAY(1L) voorkomt dat het oproepende programma door de Shift-toets meteen de op de minimale of de maximale snelheid verder gaat. De vijf regels 'waarde++' en 'waarde--' hebben een dubbele functie. Enerzijds zorgen ze ervoor dat de waarde snel genoeg verandert (de vertraging kan niet kleiner worden dan 1L). Anderzijds verhinderen ze dat het afgeremde programma na een druk op de rechter Shift-toets op volle snelheid doorloopt. Als het controleblok alleen zou bestaan uit een DELAY- opdracht, zou het doelprogramma bijna niet worden afgeremd. In de afgedrukte vorm moet de gegeven programmacode met een hoge prioriteit altijd worden afgewerkt. Dat levert altijd een kleine vertraging op. Probeer de instructie eens uit met een vertragingwaarde van 500. Elke druk op de Shift-toets veroorzaakt dan een duidelijke snelheidstoename van het doelprogramma.

7.6 BASIC-laders voor de CLI-instructies

We willen de nieuwe CLI-instructies ook bereikbaar maken voor mensen die geen C-compiler bezitten. In de volgende paragrafen vindt u voor elk van de vier hierboven behandelde instructies een BASIC-lader. De listings zijn zo lang dat u het overtypen het best over een paar mensen kunt verdelen.

De programma's berekenen bij hun uitvoering over elke regel een proefsom en leveren de regel met de fout als er iets niet klopt in de DATA-regels. Als er geen fout is opgetreden, staat na afloop de hele CLI-instructie als programmabestand op de RAM-disk. Kopieer dat bestand vervolgens naar de werkdiskette. Het .info-bestand hoeft u natuurlijk niet mee te kopiëren.

De kop van alle vier programma's is gelijk, met uitzondering van de eerste regel. Typ de kop dus eenmaal over en sla hem apart op (bijvoorbeeld onder de naam 'header'). Bij de invoer van de volgende instructie hoeft u dat gedeelte dan alleen maar te laden.

Er zijn een paar DATA-blokken die alleen uit nullen bestaan. Die hoeft u niet stuk voor stuk te typen. Gebruik de instructies 'Copy' en 'Paste' uit het BASIC-menu.

7.6.1 BASIC-lader voor TASK

```
OPEN "ram:Task" FOR OUTPUT AS #1

ON ERROR GOTO fout
PRINT "Even wachten alstublieft."

WHILE 1
  regel=regel+1
  som=0
  FOR i=1 TO 17
    READ Dat$
    CALL Dec(Dat$)
    som=som+waarde
    PRINT#1,CHR$(waarde);
  NEXT i

  READ proef$
  CALL Dec(proef$)

  IF waarde <> som-INT(som/256)*256 THEN
    PRINT "Fout in DATA-regel ";regel:STOP
  END IF
WEND

SUB Dec(waarde$) STATIC
  SHARED waarde
  waarde=0
  FOR j=1 TO LEN(waarde$)
    decwaarde=ASC(RIGHT$(waarde$,j))
    IF decwaarde > 57 THEN decwaarde=decwaarde-7
    waarde=waarde+(decwaarde-48)*16^(j-1)
  NEXT j
END SUB

fout:
IF ERR <> 4 THEN PRINT "Error-Code";ERR:STOP
PRINT "O.K."
PRINT "Kopieer het bestand van de RAM-disk naar"
PRINT "uw CLI-werkdiskette."

CLOSE #1
END

DATA 0, 0, 3,F3, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0,F9
DATA 0, 0, 2, 0, 0, 4,46, 0, 0, 0,A5, 0, 0, 0, 1, 0, 0,F2
DATA 3,E9, 0, 0, 4,46,4E,FA, 0,BA,4E,55,FF,80,42,6D,FF,8
```

DATA 82,4E,BA,10,6E,20,6C,82,56,2B,68, 1,A4,FF,FC,60,3C,3B
DATA 20,6D,FF,FC,30,2D,FF,82,48,CO,E5,80,43,ED,FF,AC,23,D1
DATA A8, 0, A, 8, 0,20,6D,FF,FC,10,28, 0, 9,48,80,32,2D,AA
DATA FF,82,52,6D,FF,82,48,C1,E3,81,43,ED,FF,84,33,80,18,AC
DATA 0,20,6D,FF,FC,2B,50,FF,FC,20,6D,FF,FC,4A,90,66,BC,82
DATA 4E,BA,10,22,42,6D,FF,80,60,38,30,2D,FF,80,48,CO,E5,C9
DATA 80,41,ED,FF,AC,2F,30, 8, 0,48,7A, 0,34,4E,BA, 4,A8,6A
DATA 50,4F,30,2D,FF,80,48,CO,E3,80,41,ED,FF,84,3F,30, 8,E
DATA 0,48,7A, 0,1D,4E,BA, 4,8E,5C,4F,52,6D,FF,80,30,2D,BF
DATA FF,80,B0,6D,FF,82,6D,BE,4E,5D,4E,75,25,73, 0, D, 9,64
DATA 9, 9,25,64, A, 0, 0,61,70,43,EC,82,52,45,EC,82,52,7E
DATA B5,C9,66, E,32,3C, 0,10,6B, 8,74, 0,22,C2,51,C9,FF,54
DATA FC,29,4F,82,5A,2C,78, 0, 4,29,4E,82,56,48,E7,80,80,76
DATA 8,2E, 0, 4, 1,29,67,10,4B,FA, 0, 8,4E,AE,FF,E2,60,65
DATA 6,42,A7,F3,5F,4E,73,43,FA, 0,20,4E,AE,FE,68,29,40,2A
DATA 82,5E,66, C,2E,3C, 0, 3,80, 7,4E,AE,FF,94,60, 4,4E,87
DATA BA, 0,1A,50,4F,4E,75,64,6F,73,2E,6C,69,62,72,61,72,26
DATA 79, 0,49,F9, 0, 0,7F,FE,4E,75,4E,55, 0, 0,2F, A,48,1F
DATA 79, 0, 1, 0, 0,30,2C,82,4E,C1,FC, 0, 6,2F, 0,4E,BA,A0
DATA F,20,50,4F,29,40,82,62,66,14,42,A7,48,79, 0, 1, 0,40
DATA 0,4E,BA, E,E4,50,4F,2E,6C,82,5A,4E,75,20,6C,82,62,42
DATA 42,68, 0, 4,20,6C,82,62,31,7C, 0, 1, 0,10,22,6C,82,EC
DATA 62,33,7C, 0, 1, 0, A,20,6C,82,5A,20,2C,82,5A,90,A8,E4
DATA 0, 4,50,80,29,40,82,66,20,6C,82,66,20,BC,4D,41,4E,51
DATA 58,42,A7,4E,BA, E,E4,58,4F,24,40,4A,AA, 0,AC,67,2E,7B
DATA 2F,2D, 0, C,2F,2D, 0, 8,2F, A,4E,BA, 0,B2,4F,EF, 0,FD
DATA C,39,7C, 0, 1,82,6A,20,6C,82,62, 0,68,80, 0, 0, 4,A
DATA 20,6C,82,62, 0,68,80, 0, 0, A,60,44,48,6A, 0,5C,4E,62
DATA BA, E,FE,58,4F,48,6A, 0,5C,4E,BA, E,BE,58,4F,29,40,5F
DATA 82,6C,20,6C,82,6C,4A,A8, 0,24,67,10,20,6C,82,6C,22,91
DATA 68, 0,24,2F,11,4E,BA, D,DE,58,4F,2F,2C,82,6C,2F, A,E8
DATA 4E,BA, 2,58,50,4F,29,6C,82,6C,82,70,4E,BA, D,DE,20,89
DATA 6C,82,62,20,80,4E,BA, D,FE,20,6C,82,62,21,40, 0, 6,DA
DATA 67,16,48,78, 3,ED,48,7A, 0,2C,4E,BA, D,DA,50,4F,20,C9
DATA 6C,82,62,21,40, 0, C,2F,2C,82,70,3F,2C,82,74,4E,BA,73
DATA FD,A6,5C,4F,42,67,4E,BA, C,22,54,4F,24,5F,4E,5D,4E,4C
DATA 75,2A, 0,4E,55, 0, 0,48,E7, C,30,24,6D, 0,10,20,6D,DB
DATA 0, 8,20,28, 0,AC,E5,80,28, 0,20,44,20,28, 0,10,E5,2A
DATA 80,26,40,10,13,48,80,48,CO,D0,AD, 0, C,54,80,39,40,AF
DATA 82,76,42,A7,30,2C,82,76,48,CO,2F, 0,4E,BA, D,BE,50,8F
DATA 4F,29,40,82,78,66, 8,4C,DF, C,30,4E,5D,4E,75,10,13,18
DATA 48,80,3F, 0,20,4B,52,88,2F, 8,2F,2C,82,78,4E,BA, 1,E1
DATA 44,4F,EF, 0, A,48,7A, 1,3A,10,13,48,80,48,CO,D0,AC,F8
DATA 82,78,2F, 0,4E,BA, 1,78,50,4F,3F,2D, 0, E,2F, A,2F,2B
DATA 2C,82,78,4E,BA, 1,44,4F,EF, 0, A,42,6C,82,74,26,6C,F1
DATA 82,78,24,4B,10,13,48,80,3A, 0,B0,7C, 0,20,67,18;BA,13
DATA 7C, 0, 9,67,12,BA,7C, 0, C,67, C,BA,7C, 0, D,67, 6,63
DATA BA,7C, 0, A,66, 4,52,8B,60,D8, C,13, 0,20,6D,7A, C,F1

DATA 13, 0,22,66,2E,52,8B,20,4B,52,8B,10,10,48,80,3A, 0,10
DATA 67,1E,20,4A,52,8A,10,85,BA,7C, 0,22,66,10, C,13, 0,4D
DATA 22,66, 4,52,8B,60, 6,42,2A,FF,FF,60, 2,60,D6,60,38,69
DATA 20,4B,52,8B,10,10,48,80,3A, 0,67,26,BA,7C, 0,20,67,B4
DATA 20,BA,7C, 0, 9,67,1A,BA,7C, 0, C,67,14,BA,7C, 0, D,E0
DATA 67, E,BA,7C, 0, A,67, 8,20,4A,52,8A,10,85,60,CE,20,4D
DATA 4A,52,8A,42,10,4A,45,66, 2,53,8B,52,6C,82,74,60, 0,61
DATA FF,5A,42,12,42,A7,30,2C,82,74,52,40,48,C0,E5,80,2F,16
DATA 0,4E,BA, C,A8,50,4F,29,40,82,70,66, 8,42,6C,82,74,C8
DATA 60, 0,FE,E4,7A, 0,26,6C,82,78,60,1E,30, 5,48,C0,E5,E8
DATA 80,20,6C,82,70,21,8B, 8, 0,2F, B,4E,BA, 5,A2,58,4F,42
DATA 52,40,48,C0,D7,C0,52,45,BA,6C,82,74,6D,DC,30, 5,48,AA
DATA C0,E5,80,20,6C,82,70,42,B0, 8, 0,60, 0,FE,A6,20, 0,C1
DATA 4C,EF, 3, 0, 0, 4,20, 8,32,2F, 0, C,60, 2,10,D9,57,79
DATA C9,FF,FC,67, 6,52,41,60, 2,42,18,51,C9,FF,FC,4E,75,58
DATA 30,3C,7F,FF,60, 4,30,2F, 0, C,20,6F, 0, 4,4A,18,66,14
DATA FC,53,48,22,6F, 0, 8,53,40,10,D9,57,C8,FF,FC,67, 2,2F
DATA 42,10,20,2F, 0, 4,4E,75,20,6F, 0, 4,20, 8,22,6F, 0,B4
DATA 8,10,D9,66,FC,4E,75,4E,55, 0, 0,48,E7, E,30,24,6D,B7
DATA 0, 8,42,A7,48,7A, 0,8E,4E,BA, C,34,50,4F,29,40,82,13
DATA 92,66, 8,4C,DF, C,70,4E,5D,4E,75,20,6D, 0, C,22,68,38
DATA 0,24,2F,29, 0, 4,4E,BA, C,64,58,4F,28, 0,67,52,48,C8
DATA 7A, 0,6D,20,44,2F,28, 0,36,4E,BA, C,36,50,4F,26,40,27
DATA 4A,80,67,34,48,78, 3,ED,2F, B,4E,BA, B,54,50,4F,2C,81
DATA 0,67,24,20, 6,E5,80,2A, 0,20,45,25,68, 0, 8, 0,A4,DE
DATA 25,46, 0,9C,48,78, 3,ED,48,7A, 0,38,4E,BA, B,30,50,44
DATA 4F,25,40, 0,A0,2F, 4,4E,BA, C, 2,58,4F,2F,2C,82,92,B3
DATA 4E,BA, B,56,58,4F,42,AC,82,92,60,80,69,63,6F,6E,2E,C9
DATA 6C,69,62,72,61,72,79, 0,57,49,4E,44,4F,57, 0,2A, 0,F7
DATA 4E,55, 0, 0,48,6D, 0, C,2F,2D, 0, 8,48,7A, 4,70,4E,4C
DATA BA, 0,98,4F,EF, 0, C,4E,5D,4E,75,4E,55, 0, 0,48,E7,DC
DATA 8,20,24,6D, 0, E, C,6D, 0, 4, 0,12,66, 8,20,6D, 0,51
DATA 8,28,10,60,1C,4A,6D, 0, C,6F, C,20,6D, 0, 8,70, 0,FF
DATA 30,10,28, 0,60, A,20,6D, 0, 8,30,10,48,C0,28, 0,42,19
DATA 6D, 0,12,4A,6D, 0, C,6C,10,44,6D, 0, C,4A,84,6C, 8,BD
DATA 44,84,3B,7C, 0, 1, 0,12,32,2D, 0, C,48,C1,20, 4,4E,78
DATA BA, 3,8E,41,EC,80, 2,53,8A,14,B0, 0, 0,32,2D, 0, C,6
DATA 48,C1,20, 4,4E,BA, 3,84,28, 0,66,DA,4A,6D, 0,12,67,54
DATA 6,53,8A,14,BC, 0,2D,20, A,4C,DF, 4,10,4E,5D,4E,75,B7
DATA 4E,55,FF,22,48,E7, 8,30,24,6D, 0, 8,26,6D, 0, C,42,A5
DATA 6D,FF,FA,2B,6D, 0,10,FF,FC,20,4B,52,8B,10,10,48,80,39
DATA 38, 0,67, 0, 2,EC,B8,7C, 0,25,66, 0, 2,CA,42,2D,FF,86
DATA 30,3B,7C, 0, 1,FF,F8,3B,7C, 0,20,FF,F6,3B,7C,27,10,99
DATA FF,F4,20,4B,52,8B,10,10,48,80,38, 0,B0,7C, 0,2D,66,1A
DATA E,42,6D,FF,F8,20,4B,52,8B,10,10,48,80,38, 0,B8,7C,50
DATA 0,30,66,10,3B,7C, 0,30,FF,F6,20,4B,52,8B,10,10,48,32
DATA 80,38, 0,B8,7C, 0,2A,66,18,20,6D,FF,FC,54,AD,FF,FC,18
DATA 3B,50,FF,F2,20,4B,52,8B,10,10,48,80,38, 0,60,32,42,B8

DATA 6D,FF,F2,60,1C,30,2D,FF,F2,C1,FC, 0, A,D0,44,90,7C,F
DATA 0,30,3B,40,FF,F2,20,4B,52,8B,10,10,48,80,38, 0,30,34
DATA 4,52,40,41,EC,80,14, 8,30, 0, 2, 0, 0,66,D4,B8,7C,FF
DATA 0,2E,66,5A,20,4B,52,8B,10,10,48,80,38, 0,B0,7C, 0,82
DATA 2A,66,18,20,6D,FF,FC,54,AD,FF,FC,3B,50,FF,F4,20,4B,15
DATA 52,8B,10,10,48,80,38, 0,60,32,42,6D,FF,F4,60,1C,30,DD
DATA 2D,FF,F4,C1,FC, 0, A,D0,44,90,7C, 0,30,3B,40,FF,F4,A5
DATA 20,4B,52,8B,10,10,48,80,38, 0,30, 4,52,40,41,EC,80,DB
DATA 14, 8,30, 0, 2, 0, 0,66,D4,3B,7C, 0, 2,FF,F0,B8,7C,64
DATA 0,6C,66,12,20,4B,52,8B,10,10,48,80,38, 0,3B,7C, 0,3
DATA 4,FF,F0,60,10,B8,7C, 0,68,66, A,20,4B,52,8B,10,10,D7
DATA 48,80,38, 0,30, 4,48,C0,60,78,3B,7C, 0, 8,FF,EE,60,20
DATA 16,3B,7C, 0, A,FF,EE,60, E,3B,7C, 0,10,FF,EE,60, 6,4C
DATA 3B,7C,FF,F6,FF,EE,3F,2D,FF,F0,48,6D,FF,30,3F,2D,FF,43
DATA EE,2F,2D,FF,FC,4E,BA,FD,E4,4F,EF, 0, C,2B,40,FF,EA,CC
DATA 30,2D,FF,F0,48,C0,D1,AD,FF,FC,60,5A,20,6D,FF,FC,58,67
DATA AD,FF,FC,2B,50,FF,EA,2F,2D,FF,EA,4E,BA, 2, C,58,4F,E
DATA 3B,40,FF,F0,60,4A,20,6D,FF,FC,54,AD,FF,FC,38,10,41,21
DATA ED,FF,2F,2B,48,FF,EA,10,84,60,28,90,BC, 0, 0, 0,63,42
DATA 67,E2,53,80,67,94,90,BC, 0, 0, 0, B,67, 0,FF,74,59,A1
DATA 80,67,B4,55,80,67, 0,FF,72,57,80,67, 0,FF,74,60,CC,25
DATA 41,ED,FF,30,91,ED,FF,EA,3B,48,FF,F0,30,2D,FF,F0,B0,32
DATA 6D,FF,F4,6F, 6,3B,6D,FF,F4,FF,F0,4A,6D,FF,F8,67,68,DC
DATA 20,6D,FF,EA, C,10, 0,2D,67, A,22,6D,FF,EA, C,11, 0,C5
DATA 2B,66,2E, C,6D, 0,30,FF,F6,66,26,53,6D,FF,F2,20,6D,27
DATA FF,EA,52,AD,FF,EA,10,10,48,80,3F, 0,4E,92,54,4F,B0,2B
DATA 7C,FF,FF,66, A,70,FF,4C,DF, C,10,4E,5D,4E,75,60,16,84
DATA 3F,2D,FF,F6,4E,92,54,4F,B0,7C,FF,FF,66, 4,70,FF,60,47
DATA E4,52,6D,FF,FA,30,2D,FF,F2,53,6D,FF,F2,B0,6D,FF,F0,A7
DATA 6E,DC,42,6D,FF,EE,60,20,20,6D,FF,EA,52,AD,FF,EA,10,D4
DATA 10,48,80,3F, 0,4E,92,54,4F,B0,7C,FF,FF,66, 4,70,FF,9D
DATA 60,B0,52,6D,FF,EE,20,6D,FF,EA,4A,10,67, A,30,2D,FF,59
DATA EE,B0,6D,FF,F4,6D,CE,30,2D,FF,EE,D1,6D,FF,FA,4A,6D,71
DATA FF,F8,66,28,60,18,3F,3C, 0,20,4E,92,54,4F,B0,7C,FF,46
DATA FF,66, 6,70,FF,60, 0,FF,78,52,6D,FF,FA,30,2D,FF,F2,B7
DATA 53,6D,FF,F2,B0,6D,FF,F0,6E,DA,60,16,3F, 4,4E,92,54,F2
DATA 4F,B0,7C,FF,FF,66, 6,70,FF,60, 0,FF,52,52,6D,FF,FA,BD
DATA 60, 0,FD, A,30,2D,FF,FA,60, 0,FF,42,48,E7,48, 0,42,17
DATA 84,4A,80,6A, 4,44,80,52,44,4A,81,6A, 6,44,81, A,44,64
DATA 0, 1,61,3E,4A,44,67, 2,44,80,4C,DF, 0,12,4A,80,4E,B0
DATA 75,48,E7,48, 0,42,84,4A,80,6A, 4,44,80,52,44,4A,81,F
DATA 6A, 2,44,81,61,1A,20, 1,60,D8,2F, 1,61,12,20, 1,22,EB
DATA 1F,4A,80,4E,75,2F, 1,61, 6,22,1F,4A,80,4E,75,48,E7,40
DATA 30, 0,48,41,4A,41,66,20,48,41,36, 1,34, 0,42,40,48,88
DATA 40,80,C3,22, 0,48,40,32, 2,82,C3,30, 1,42,41,48,41,E3
DATA 4C,DF, 0, C,4E,75,48,41,26, 1,22, 0,42,41,48,41,48,20
DATA 40,42,40,74, F,D0,80,D3,81,B6,81,62, 4,92,83,52,40,2D
DATA 51,CA,FF,F2,4C,DF, 0, C,4E,75,20,6F, 0, 4,20, 8,4A,B

DATA 18,66,FC,91,C0,20, 8,53,80,4E,75,4E,55, 0, 0,48,6C,E0
DATA 80,AC,3F,2D, 0, 8,4E,BA, 0, 8,5C,4F,4E,5D,4E,75,4E,17
DATA 55, 0, 0,2F, 4,38,2D, 0, 8,2F,2D, 0, A,3F, 4,4E,BA,A6
DATA 0,30,5C,4F,B8,7C, 0, A,66,24,20,6D, 0, A,10,28, 0,72
DATA C,48,80, 8, 0, 0, 7,67,14,3F,3C,FF,FF,2F,2D, 0, A,3D
DATA 4E,BA, 0,F6,5C,4F,28,1F,4E,5D,4E,75,60,F8,4E,55, 0,59
DATA 0,2F, A,24,6D, 0, A,20,52,B1,EA, 0, 4,65,18,30,2D,BF
DATA 0, 8,C0,7C, 0,FF,3F, 0,2F, A,4E,BA, 0,CA,5C,4F,24,5C
DATA 5F,4E,5D,4E,75,20,52,52,92,10,2D, 0, 9,10,80,48,80,C1
DATA C0,7C, 0,FF,60,E8,4E,55, 0, 0,2F, A,41,EC,80,96,24,C6
DATA 48,20,4A,D5,FC, 0, 0, 0,16,2F, 8,61,10,58,4F,41,EC,15
DATA 82,4E,B5,C8,65,EA,24,5F,4E,5D,4E,75,4E,55, 0, 0,48,78
DATA E7, 8,20,24,6D, 0, 8,78, 0,20, A,66, A,70,FF,4C,DF,54
DATA 4,10,4E,5D,4E,75,4A,2A, 0, C,67,52, 8,2A, 0, 2, 0,EF
DATA C,67, C,3F,3C,FF,FF,2F, A,61,54,5C,4F,38, 0,10,2A,3
DATA 0, D,48,80,3F, 0,4E,BA, 4,F2,54,4F,88,40, 8,2A, 0,AF
DATA 1, 0, C,67, A,2F,2A, 0, 8,4E,BA, 2,30,58,4F, 8,2A,F2
DATA 0, 5, 0, C,67,14,2F,2A, 0,12,4E,BA, 2,C2,58,4F,2F,99
DATA 2A, 0,12,4E,BA, 2,14,58,4F,42,92,42,AA, 0, 4,42,AA,B1
DATA 0, 8,42,2A, 0, C,30, 4,60,8E,4E,55,FF,FE,48,E7, 8,79
DATA 20,24,6D, 0, 8,41,FA,FF,44,29,48,82,7C, 8,2A, 0, 4,DC
DATA 0, C,67, A,70,FF,4C,DF, 4,10,4E,5D,4E,75, 8,2A, 0,CB
DATA 2, 0, C,67,30,20,12,90,AA, 0, 8,38, 0,3F, 4,2F,2A,ED
DATA 0, 8,10,2A, 0, D,48,80,3F, 0,4E,BA, 2,80,50,4F,B0,2F
DATA 44,67,10, 8,EA, 0, 4, 0, C,42,92,42,AA, 0, 4,70,FF,F0
DATA 60,C0, C,6D,FF,FF, 0, C,66,10, 8,AA, 0, 2, 0, C,42,1B
DATA 92,42,AA, 0, 4,70, 0,60,A8,4A,AA, 0, 8,66, 8,2F, A,9D
DATA 4E,BA, 0,9A,58,4F, C,6A, 0, 1, 0,10,66,2A,1B,6D, 0,E8
DATA D,FF,FF,3F,3C, 0, 1,48,6D,FF,FF,10,2A, 0, D,48,80,49
DATA 3F, 0,4E,BA, 2,22,50,4F,B0,7C, 0, 1,66,A0,30,2D, 0,9A
DATA C,60, 0,FF,6A,24,AA, 0, 8,30,2A, 0,10,48,C0,D0,AA,97
DATA 0, 8,25,40, 0, 4, 8,EA, 0, 2, 0, C,20,52,52,92,10,D7
DATA 2D, 0, D,10,80,48,80,C0,7C, 0,FF,60, 0,FF,3E,4E,55,D
DATA 0, 0,2F, A,41,EC,80,96,24,48,4A,2A, 0, C,67,18,D5,BC
DATA FC, 0, 0, 0,16,41,EC,82,4E,B5,C8,65, 8,70, 0,24,5F,EC
DATA 4E,5D,4E,75,60,E2,42,92,42,AA, 0, 4,42,AA, 0, 8,20,88
DATA A,60,EA,4E,55,FF,FC,2F, A,24,6D, 0, 8,3F,3C, 4, 0,43
DATA 4E,BA, 0,C0,54,4F,2B,40,FF,FC,66,18,35,7C, 0, 1, 0,1
DATA 10,20, A,D0,BC, 0, 0, 0, E,25,40, 0, 8,24,5F,4E,5D,6F
DATA 4E,75,35,7C, 4, 0, 0,10, 8,EA, 0, 1, 0, C,25,6D,FF,18
DATA FC, 0, 8,10,2A, 0, D,48,80,3F, 0,4E,BA, 0,E2,54,4F,DF
DATA 4A,40,67, 6, 0,2A, 0,80, 0, C,60,CE,4E,55, 0, 0,48,C6
DATA E7, 0,30,24,6C,82,52,60,14,26,52,20,2A, 0, 4,50,80,85
DATA 2F, 0,2F, A,4E,BA, 4,24,50,4F,24,4B,20, A,66,E8,42,60
DATA AC,82,52,4C,DF, C, 0,4E,5D,4E,75,4E,55, 0, 0,2F, A,1
DATA 41,FA,FF,C6,29,48,82,80,42,A7,20,2D, 0, 8,50,80,2F,B0
DATA 0,4E,BA, 3,C2,50,4F,24,40,4A,80,66, 8,70, 0,24,5F,FB
DATA 4E,5D,4E,75,24,AC,82,52,25,6D, 0, 8, 0, 4,29,4A,82,A5

DATA 52,20, A,50,80,60,E6,4E,55, 0, 0,70, 0,30,2D, 0, 8,A
DATA 2F, 0,61,B2,58,4F,4E,5D,4E,75,4E,55, 0, 0,48,E7, 0,29
DATA 30,97,CB,24,6C,82,52,60, E,20,6D, 0, 8,51,88,B1,CA,4D
DATA 67,12,26,4A,24,52,20, A,66,EE,70,FF,4C,DF, C, 0,4E,D1
DATA 5D,4E,75,20, B,67, 4,26,92,60, 4,29,52,82,52,20,2A,6B
DATA 0, 4,50,80,2F, 0,2F, A,4E,BA, 3,76,50,4F,70, 0,60,2C
DATA D8,4E,55, 0, 0,2F, A,30,2D, 0, 8,C1,FC, 0, 6,24,40,40
DATA D5,EC,82,62,4A,6D, 0, 8,6D, E,30,2D, 0, 8,B0,6C,82,E2
DATA 4E,6C, 4,4A,92,66, E,39,7C, 0, 2,82,84,70,FF,24,5F,BD
DATA 4E,5D,4E,75,30,2D, 0, 8,C1,FC, 0, 6,20,6C,82,62,2F,35
DATA 30, 8, 0,4E,BA, 2,9C,58,4F,4A,80,67, 4,70, 1,60, 2,8D
DATA 70, 0,60,D8,4E,55, 0, 0,2F,2D, 0, 8,4E,BA, 2,66,58,77
DATA 4F,4A,80,66, E,4E,BA, 2,70,39,40,82,84,70,FF,4E,5D,A0
DATA 4E,75,70, 0,60,F8,4E,55, 0, 0,48,E7, C,20,38,2D, 0,EE
DATA 8,4E,BA, 0,70,30, 4,C1,FC, 0, 6,24,40,D5,EC,82,62,80
DATA 4A,44,6D, A,B8,6C,82,4E,6C, 4,4A,92,66,10,39,7C, 0,70
DATA 2,82,84,70,FF,4C,DF, 4,30,4E,5D,4E,75,30,2A, 0, 4,A2
DATA C0,7C, 0, 3,66, A,39,7C, 0, 5,82,84,70,FF,60,E4,70,92
DATA 0,30,2D, 0, E,2F, 0,2F,2D, 0, A,2F,12,4E,BA, 2,2C,77
DATA 4F,EF, 0, C,2A, 0,B0,BC,FF,FF,FF,FF,66, C,4E,BA, 1,57
DATA F0,39,40,82,84,70,FF,60,B8,20, 5,60,B4,4E,55,FF,FC,CD
DATA 48,78,10, 0,42,A7,4E,BA, 2,A0,50,4F,2B,40,FF,FC, 8,70
DATA 0, 0, C,67,12,4A,6C,82,6A,66, 8,20,2D,FF,FC,4E,5D,88
DATA 4E,75,4E,BA, 0, 6,70, 0,60,F4,4E,55, 0, 0,48,78, 0,F8
DATA 4,48,7A, 0,1E,4E,BA, 1,C6,2F, 0,4E,BA, 1,C8,4F,EF,F1
DATA 0, C,3F,3C, 0, 1,4E,BA, 0, C,54,4F,4E,5D,4E,75,5E,B
DATA 43, A, 0,4E,55, 0, 0,4A,AC,82,7C,67, 6,20,6C,82,7C,DB
DATA 4E,90,3F,2D, 0, 8,4E,BA, 0, 8,54,4F,4E,5D,4E,75,4E,C1
DATA 55,FF,FC,2F, 4,30,2D, 0, 8,48,C0,2B,40,FF,FC,4A,AC,4C
DATA 82,62,67,28,78, 0,60, A,3F, 4,4E,BA, 0,D0,54,4F,52,65
DATA 44,B8,6C,82,4E,6D,F0,30,2C,82,4E,C1,FC, 0, 6,2F, 0,B3
DATA 2F,2C,82,62,4E,BA, 1,C0,50,4F,4A,AC,82,80,67, 6,20,2C
DATA 6C,82,80,4E,90,4A,AC,82,86,67, A,2F,2C,82,86,4E,BA,26
DATA 1,66,58,4F,4A,AC,82,8A,67, A,2F,2C,82,8A,4E,BA, 1,F1
DATA 56,58,4F,4A,AC,82,8E,67, A,2F,2C,82,8E,4E,BA, 1,46,2E
DATA 58,4F,2C,78, 0, 4, 8,2E, 0, 4, 1,29,67,14,2F, D,4B,B5
DATA FA, 0, A,4E,AE,FF,E2,2A,5F,60, 6,42,A7,F3,5F,4E,73,CC
DATA 4A,AC,82,6C,66,30,4A,AC,82,78,67,28,30,2C,82,76,48,95
DATA C0,2F, 0,2F,2C,82,78,4E,BA, 1,46,50,4F,30,2C,82,74,84
DATA 52,40,48,C0,E5,80,2F, 0,2F,2C,82,70,4E,BA, 1,30,50,4
DATA 4F,60, E,4E,BA, 1,20,2F,2C,82,6C,4E,BA, 1,4C,58,4F,2B
DATA 20,2D,FF,FC,2E,6C,82,5A,4E,75,28,1F,4E,5D,4E,75,4E,84
DATA 55, 0, 0,48,E7, E,20,38,2D, 0, 8,30, 4,C1,FC, 0, 6,16
DATA 24,40,D5,EC,82,62,4A,44,6D, A,B8,6C,82,4E,6C, 4,4A,BC
DATA 92,66,10,39,7C, 0, 2,82,84,70,FF,4C,DF, 4,70,4E,5D,7E
DATA 4E,75,30,2A, 0, 4,C0,7C,80, 0,66, 8,2F,12,4E,BA, 0,94
DATA A,58,4F,42,92,70, 0,60,E0,22,2F, 0, 4,2C,6C,82,5E,2
DATA 4E,EE,FF,DC,22,2F, 0, 4,2C,6C,82,5E,4E,EE,FF,82,22,C3

DATA 2F, 0, 4,2C,6C,82,5E,4E,EE,FF,B8,2C,6C,82,5E,4E,EE,52
DATA FF,CA,2C,6C,82,5E,4E,EE,FF,7C,22,2F, 0, 4,2C,6C,82,67
DATA 5E,4E,EE,FF,28,4C,EF, 0, 6, 0, 4,2C,6C,82,5E,4E,EE,BA
DATA FF,E2,2C,6C,82,5E,4E,EE,FF,C4,4C,EF, 0, E, 0, 4,2C,D1
DATA 6C,82,5E,4E,EE,FF,DO,48,E7, 1, 4,4C,EF,20,80, 0, C,72
DATA 2C,6C,82,56,4E,AE,FF,94,4C,DF,20,80,4E,75,4E,FA, 0,D5
DATA 2,22,6F, 0, 4,2C,6C,82,56,4E,EE,FE,62,4C,EF, 0, 3,E1
DATA 0, 4,2C,6C,82,56,4E,EE,FF,3A,2C,6C,82,56,4E,EE,FF,94
DATA 88,2C,6C,82,56,4E,EE,FF,82,22,6F, 0, 4,2C,6C,82,56,BA
DATA 4E,EE,FE,DA,2C,6C,82,56,4E,EE,FF,7C,22,6F, 0, 4,20,F0
DATA 2F, 0, 8,2C,6C,82,56,4E,EE,FF,2E,20,6F, 0, 4,2C,6C,3B
DATA 82,56,4E,EE,FE,8C,2C,6C,82,56,22,6F, 0, 4,20,2F, 0,F2
DATA 8,4E,EE,FD,D8,22,6F, 0, 4,2C,6C,82,56,4E,EE,FE,86,DE
DATA 4C,EF, 0, 3, 0, 4,2C,6C,82,56,4E,EE,FE,CE,20,6F, 0,49
DATA 4,2C,6C,82,56,4E,EE,FE,80,4C,EF, 3, 0, 0, 4,2C,6C,8
DATA 82,92,4E,EE,FF,A0,20,6F, 0, 4,2C,6C,82,92,4E,EE,FF,69
DATA A6,20,6F, 0, 4,2C,6C,82,92,4E,EE,FF,B2, 0, 0, 3,EC,C1
DATA 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1,30, 0, 0, 0, 0, 0,33
DATA 0, 3,F2, 0, 0, 3,EA, 0, 0, 0,94,30,31,32,33,34,35,A5
DATA 36,37,38,39,61,62,63,64,65,66, 0, 0, 0,20,20,20,20,B3
DATA 20,20,20,20,20,30,30,30,30,30,20,20,20,20,20,20,70
DATA 20,20,20,20,20,20,20,20,20,20,20,90,40,40,40,40,30
DATA 40,40,40,40,40,40,40,40,40,40, C, C, C, C, C, C, C,D4
DATA C, C, C,40,40,40,40,40,40,9, 9, 9, 9, 9, 9, 1,1B
DATA 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,11
DATA 1, 1,40,40,40,40,40,40, A, A, A, A, A, A, 2, 2, 2,C4
DATA 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,22
DATA 40,40,40,40,20, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,20
DATA 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,2
DATA 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0,0,3
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0,1,4
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0

```

DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
DATA 0, 0, 0, 0, 0, 14, 0, 0, 0, 0, 0, 3, F2, 0, 0, 3, EB, 0, F7
DATA 0, 0, 1, 0, 0, 3, F2,

```

7.6.2 BASIC-lader voor TASKPRI

```

OPEN "ram:TaskPri" FOR OUTPUT AS #1

ON ERROR GOTO fout
PRINT "Even wachten alstublieft."

WHILE 1
  regel=regel+1
  som=0
  FOR i=1 TO 17
    READ Dat$
    CALL Dec(Dat$)
    som=som+waarde
    PRINT#1,CHR$(waarde);
  NEXT i

  READ proef$
  CALL Dec(proef$)

  IF waarde <> som-INT(som/256)*256 THEN
    PRINT "Fout in DATA-regel ";regel:STOP
  END IF
WEND

SUB Dec(waarde$) STATIC
  SHARED waarde
  waarde=0
  FOR j=1 TO LEN(waarde$)
    decwaarde=ASC(RIGHT$(waarde$,j))
    IF decwaarde > 57 THEN decwaarde=decwaarde-7
    waarde=waarde+(decwaarde-48)*16^(j-1)
  NEXT j
END SUB

fout:
IF ERR <> 4 THEN PRINT "Error-Code";ERR:STOP
PRINT "O.K."
PRINT "Kopieer het bestand van de RAM-disk naar"

```

PRINT "uw CLI-werkdiskette."

CLOSE #1

END

DATA 0, 0, 3,F3, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0,F9
DATA 0, 0, 2, 0, 0, 4,9E, 0, 0, 0,A5, 0, 0, 0, 1, 0, 0,4A
DATA 3,E9, 0, 0, 4,9E,4E,FA, 2,16,4E,55,FF,F0,42,AD,FF,6E
DATA F8, C,6D, 0, 3, 0, 8,67,12,48,7A, 0,AC,4E,BA, 6,6A,DB
DATA 58,4F,42,A7,4E,BA, F,C2,58,4F,48,6D,FF,F8,20,6D, 0,49
DATA A,2F,28, 0, 8,4E,BA, 1, A,50,4F,3B,40,FF,F6,4A,6D,42
DATA FF,F6,66,16,2F,2D,FF,F8,48,7A, 0,90,4E,BA, 6,38,50,AC
DATA 4F,42,A7,4E,BA, F,90,58,4F,20,6D, 0, A,2F,28, 0, 4,78
DATA 4E,BA,11,76,58,4F,2B,40,FF,F0,4A,AD,FF,F0,66,1A,20,16
DATA 6D, 0, A,2F,28, 0, 4,48,7A, 0,81,4E,BA, 6, 6,50,4F,C8
DATA 42,A7,4E,BA, F,5E,58,4F,2F,2D,FF,F8,2F,2D,FF,F0,4E,F1
DATA BA,11,A2,50,4F,1B,40,FF,F5,2F,2D,FF,F8,10,2D,FF,F5,DF
DATA 48,80,3F, 0,48,7A, 0,6D,4E,BA, 5,D6,4F,EF, 0, A,48,A9
DATA 78, 0, 1,4E,BA, F,2A,58,4F,4E,5D,4E,75,20,74,61,73,37
DATA 6B,6E,61,6D,65,20,20,70,72,69,6F,72,69,74,E4,74, A,B7
DATA 0,20,31,32,38,20,3C,3D,20,54,61,73,6B,70,72,69,20,72
DATA 3C,3D,20,31,32,37,20,6E,69,63,68,74,3A,20,25,6C,64,B8
DATA A, 0,54,61,73,6B,20,3C,25,73,3E,20,6E,69,63,68,74,5
DATA 20,67,65,66,75,6E,64,65,6E,20,21, A, 0,41,6C,74,65,3D
DATA 20,50,72,69,6F,72,69,74,E4,74,20,77,61,72,3A,20,25,4A
DATA 64,20,20,6E,65,75,65,20,69,73,74,20,25,6C,64, A, 0,E0
DATA 0,4E,55,FF,F6,3B,7C, 0, 1,FF,F6,42,6D,FF,FC,20,6D,7C
DATA 0, 8, C,10, 0,2D,66, A,3B,7C,FF,FF,FF,F6,52,AD, 0,6A
DATA 8,3B,7C, 0, 1,FF,FE,20,6D, 0, 8,4A,10,67,14,52,AD,26
DATA 0, 8,52,6D,FF,FC,52,6D,FF,FE, C,6D, 0, 4,FF,FE,6F,67
DATA E4,4A,6D,FF,FC,67, 8, C,6D, 0, 3,FF,FC,6F, 6,70, 0,61
DATA 4E,5D,4E,75,3B,6D,FF,FC,FF,FE,60,52,53,AD, 0, 8,20,E8
DATA 6D, 0, 8,1B,50,FF,FB, C,2D, 0,30,FF,FB,6D, 8, C,2D,EB
DATA 0,39,FF,FB,6F, 4,70, 0,60,D4,10,2D,FF,FB,48,80,90,D9
DATA 7C, 0,30,3B,40,FF,F8,20,6D, 0, C,30,2D,FF,F8,C1,ED,B9
DATA FF,F6,48,C0,22,6D, 0, C,D1,91,30,2D,FF,F6,C1,FC, 0,9
DATA A,3B,40,FF,F6,53,6D,FF,FE,4A,6D,FF,FE,6E,A8,20,6D,8E
DATA 0, C, C,90,FF,FF,FF,80,6D, C,22,6D, 0, C, C,91, 0,D6
DATA 0, 0,7F,6F, 4,70, 0,60,80,70, 1,60, 0,FF,7C,61,70,5F
DATA 43,EC,82,52,45,EC,82,52,B5,C9,66, E,32,3C, 0,10,6B,E3
DATA 8,74, 0,22,C2,51,C9,FF,FC,29,4F,82,56,2C,78, 0, 4,6D
DATA 29,4E,82,5A,48,E7,80,80, 8,2E, 0, 4, 1,29,67,10,4B,A8
DATA FA, 0, 8,4E,AE,FF,E2,60, 6,42,A7,F3,5F,4E,73,43,FA,7E
DATA 0,20,4E,AE,FE,68,29,40,82,5E,66, C,2E,3C, 0, 3,80,2A
DATA 7,4E,AE,FF,94,60, 4,4E,BA, 0,1A,50,4F,4E,75,64,6F,51
DATA 73,2E,6C,69,62,72,61,72,79, 0,49,F9, 0, 0,7F,FE,4E,A3
DATA 75,4E,55, 0, 0,2F, A,48,79, 0, 1, 0, 0,30,2C,82,4E,3F

DATA C1,FC, 0, 6,2F, 0,4E,BA, F,20,50,4F,29,40,82,62,66,7B
DATA 14,42,A7,48,79, 0, 1, 0, 0,4E,BA, E,E4,50,4F,2E,6C,F2
DATA 82,56,4E,75,20,6C,82,62,42,68, 0, 4,20,6C,82,62,31,5A
DATA 7C, 0, 1, 0,10,22,6C,82,62,33,7C, 0, 1, 0, A,20,6C,45
DATA 82,56,20,2C,82,56,90,A8, 0, 4,50,80,29,40,82,66,20,79
DATA 6C,82,66,20,BC,4D,41,4E,58,42,A7,4E,BA, E,D8,58,4F,E2
DATA 24,40,4A,AA, 0,AC,67,2E,2F,2D, 0, C,2F,2D, 0, 8,2F,94
DATA A,4E,BA, 0,B2,4F,EF, 0, C,39,7C, 0, 1,82,6A,20,6C,3C
DATA 82,62, 0,68,80, 0, 0, 4,20,6C,82,62, 0,68,80, 0, 0,28
DATA A,60,44,48,6A, 0,5C,4E,BA, F, 2,58,4F,48,6A, 0,5C,8A
DATA 4E,BA, E,B2,58,4F,29,40,82,6C,20,6C,82,6C,4A,A8, 0,32
DATA 24,67,10,20,6C,82,6C,22,68, 0,24,2F,11,4E,BA, D,DE,F6
DATA 58,4F,2F,2C,82,6C,2F, A,4E,BA, 2,58,50,4F,29,6C,82,41
DATA 6C,82,70,4E,BA, D,DE,20,6C,82,62,20,80,4E,BA, D,FE,74
DATA 20,6C,82,62,21,40, 0, 6,67,16,48,78, 3,ED,48,7A, 0,C6
DATA 2C,4E,BA, D,DA,50,4F,20,6C,82,62,21,40, 0, C,2F,2C,F2
DATA 82,70,3F,2C,82,74,4E,BA,FC,4A,5C,4F,42,67,4E,BA, C,9
DATA 22,54,4F,24,5F,4E,5D,4E,75,2A, 0,4E,55, 0, 0,48,E7,B2
DATA C,30,24,6D, 0,10,20,6D, 0, 8,20,28, 0,AC,E5,80,28,F3
DATA 0,20,44,20,28, 0,10,E5,80,26,40,10,13,48,80,48,C0,7A
DATA D0,AD, 0, C,54,80,39,40,82,76,42,A7,30,2C,82,76,48,53
DATA C0,2F, 0,4E,BA, D,BE,50,4F,29,40,82,78,66, 8,4C,DF,5D
DATA C,30,4E,5D,4E,75,10,13,48,80,3F, 0,20,4B,52,88,2F,48
DATA 8,2F,2C,82,78,4E,BA, 1,44,4F,EF, 0, A,48,7A, 1,3A,EF
DATA 10,13,48,80,48,C0,D0,AC,82,78,2F, 0,4E,BA, 1,78,50,69
DATA 4F,3F,2D, 0, E,2F, A,2F,2C,82,78,4E,BA, 1,44,4F,EF,E2
DATA 0, A,42,6C,82,74,26,6C,82,78,24,4B,10,13,48,80,3A,CE
DATA 0,B0,7C, 0,20,67,18,BA,7C, 0, 9,67,12,BA,7C, 0, C,C5
DATA 67, C,BA,7C, 0, D,67, 6,BA,7C, 0, A,66, 4,52,8B,60,A
DATA D8, C,13, 0,20,6D,7A, C,13, 0,22,66,2E,52,8B,20,4B,1B
DATA 52,8B,10,10,48,80,3A, 0,67,1E,20,4A,52,8A,10,85,BA,19
DATA 7C, 0,22,66,10, C,13, 0,22,66, 4,52,8B,60, 6,42,2A,6E
DATA FF,FF,60, 2,60,D6,60,38,20,4B,52,8B,10,10,48,80,3A,98
DATA 0,67,26,BA,7C, 0,20,67,20,BA,7C, 0, 9,67,1A,BA,7C,60
DATA 0, C,67,14,BA,7C, 0, D,67, E,BA,7C, 0, A,67, 8,20,E
DATA 4A,52,8A,10,85,60,CE,20,4A,52,8A,42,10,4A,45,66, 2,78
DATA 53,8B,52,6C,82,74,60, 0,FF,5A,42,12,42,A7,30,2C,82,66
DATA 74,52,40,48,C0,E5,80,2F, 0,4E,BA, C,A8,50,4F,29,40,66
DATA 82,70,66, 8,42,6C,82,74,60, 0,FE,E4,7A, 0,26,6C,82,D4
DATA 78,60,1E,30, 5,48,C0,E5,80,20,6C,82,70,21,8B, 8, 0,CA
DATA 2F, B,4E,BA, 5,A2,58,4F,52,40,48,C0,D7,C0,52,45,BA,12
DATA 6C,82,74,6D,DC,30, 5,48,C0,E5,80,20,6C,82,70,42,B0,BD
DATA 8, 0,60, 0,FE,A6,20, 0,4C,EF, 3, 0, 0, 4,20, 8,32,C8
DATA 2F, 0, C,60, 2,10,D9,57,C9,FF,FC,67, 6,52,41,60, 2,3
DATA 42,18,51,C9,FF,FC,4E,75,30,3C,7F,FF,60, 4,30,2F, 0,DF
DATA C,20,6F, 0, 4,4A,18,66,FC,53,48,22,6F, 0, 8,53,40,2A
DATA 10,D9,57,C8,FF,FC,67, 2,42,10,20,2F, 0, 4,4E,75,20,F4
DATA 6F, 0, 4,20, 8,22,6F, 0, 8,1C,D9,66,FC,4E,75,4E,55,E5

DATA 0, 0,48,E7, E,30,24,6D, 0, 8,42,A7,48,7A, 0,8E,4E,8D
DATA BA, C,28,50,4F,29,40,82,92,66, 8,4C,DF, C,70,4E,5D,CA
DATA 4E,75,20,6D, 0, C,22,68, 0,24,2F,29, 0, 4,4E,BA, C,7A
DATA 68,58,4F,28, 0,67,52,48,7A, 0,6D,20,44,2F,28, 0,36,10
DATA 4E,BA, C,3A,50,4F,26,40,4A,80,67,34,48,78, 3,ED,2F,97
DATA B,4E,BA, B,54,50,4F,2C, 0,67,24,20, 6,E5,80,2A, 0,7D
DATA 20,45,25,68, 0, 8, 0,A4,25,46, 0,9C,48,78, 3,ED,48,9D
DATA 7A, 0,38,4E,BA, B,30,50,4F,25,40, 0,A0,2F, 4,4E,BA,D4
DATA C, 6,58,4F,2F,2C,82,92,4E,BA, B,56,58,4F,42,AC,82,A8
DATA 92,60,80,69,63,6F,6E,2E,6C,69,62,72,61,72,79, 0,57,95
DATA 49,4E,44,4F,57, 0,2A, 0,4E,55, 0, 0,48,6D, 0, C,2F,3E
DATA 2D, 0, 8,48,7A, 4,70,4E,BA, 0,98,4F,EF, 0, C,4E,5D,0
DATA 4E,75,4E,55, 0, 0,48,E7, 8,20,24,6D, 0, E, C,6D, 0,D5
DATA 4, 0,12,66, 8,20,6D, 0, 8,28,10,60,1C,4A,6D, 0, C,90
DATA 6F, C,20,6D, 0, 8,70, 0,30,10,28, 0,60, A,20,6D, 0,DF
DATA 8,30,10,48,C0,28, 0,42,6D, 0,12,4A,6D, 0, C,6C,10,78
DATA 44,6D, 0, C,4A,84,6C, 8,44,84,3B,7C, 0, 1, 0,12,32,C3
DATA 2D, 0, C,48,C1,20, 4,4E,BA, 3,8E,41,EC,80, 2,53,8A,8B
DATA 14,B0, 0, 0,32,2D, 0, C,48,C1,20, 4,4E,BA, 3,84,28,13
DATA 0,66,DA,4A,6D, 0,12,67, 6,53,8A,14,BC, 0,2D,20, A,7A
DATA 4C,DF, 4,10,4E,5D,4E,75,4E,55,FF,22,48,E7, 8,30,24,FC
DATA 6D, 0, 8,26,6D, 0, C,42,6D,FF,FA,2B,6D, 0,10,FF,FC,5F
DATA 20,4B,52,8B,10,10,48,80,38, 0,67, 0, 2,EC,B8,7C, 0,F1
DATA 25,66, 0, 2,CA,42,2D,FF,30,3B,7C, 0, 1,FF,F8,3B,7C,5B
DATA 0,20,FF,F6,3B,7C,27,10,FF,F4,20,4B,52,8B,10,10,48,A6
DATA 80,38, 0,B0,7C, 0,2D,66, E,42,6D,FF,F8,20,4B,52,8B,73
DATA 10,10,48,80,38, 0,B8,7C, 0,30,66,10,3B,7C, 0,30,FF,E0
DATA F6,20,4B,52,8B,10,10,48,80,38, 0,B8,7C, 0,2A,66,18,3A
DATA 20,6D,FF,FC,54,AD,FF,FC,3B,50,FF,F2,20,4B,52,8B,10,58
DATA 10,48,80,38, 0,60,32,42,6D,FF,F2,60,1C,30,2D,FF,F2,C
DATA C1,FC, 0, A,D0,44,90,7C, 0,30,3B,40,FF,F2,20,4B,52,40
DATA 8B,10,10,48,80,38, 0,30, 4,52,40,41,EC,80,14, 8,30,6A
DATA 0, 2, 0, 0,66,D4,B8,7C, 0,2E,66,5A,20,4B,52,8B,10,B6
DATA 10,48,80,38, 0,B0,7C, 0,2A,66,18,20,6D,FF,FC,54,AD,6D
DATA FF,FC,3B,50,FF,F4,20,4B,52,8B,10,10,48,80,38, 0,60,41
DATA 32,42,6D,FF,F4,60,1C,30,2D,FF,F4,C1,FC, 0, A,D0,44,7B
DATA 90,7C, 0,30,3B,40,FF,F4,20,4B,52,8B,10,10,48,80,38,12
DATA 0,30, 4,52,40,41,EC,80,14, 8,30, 0, 2, 0, 0,66,D4,FB
DATA 3B,7C, 0, 2,FF,F0,B8,7C, 0,6C,66,12,20,4B,52,8B,10,18
DATA 10,48,80,38, 0,3B,7C, 0, 4,FF,F0,60,10,B8,7C, 0,68,C6
DATA 66, A,20,4B,52,8B,10,10,48,80,38, 0,30, 4,48,C0,60,74
DATA 78,3B,7C, 0, 8,FF,EE,60,16,3B,7C, 0, A,FF,EE,60, E,B6
DATA 3B,7C, 0,10,FF,EE,60, 6,3B,7C,FF,F6,FF,EE,3F,2D,FF,1E
DATA F0,48,6D,FF,30,3F,2D,FF,EE,2F,2D,FF,FC,4E,BA,FD,E4,6D
DATA 4F,EF, 0, C,2B,40,FF,EA,30,2D,FF,F0,48,C0,D1,AD,FF,6F
DATA FC,60,5A,20,6D,FF,FC,58,AD,FF,FC,2B,50,FF,EA,2F,2D,FE
DATA FF,EA,4E,BA, 2, C,58,4F,3B,40,FF,F0,60,4A,20,6D,FF,46
DATA FC,54,AD,FF,FC,38,10,41,ED,FF,2F,2B,48,FF,EA,10,84,8C

DATA 60,28,90,BC, 0, 0, 0,63,67,E2,53,80,67,94,90,BC, 0,9A
DATA 0, 0, B,67, 0,FF,74,59,80,67,B4,55,80,67, 0,FF,72,86
DATA 57,80,67, 0,FF,74,60,CC,41,ED,FF,30,91,ED,FF,EA,3B,DC
DATA 48,FF,F0,30,2D,FF,F0,B0,6D,FF,F4,6F, 6,3B,6D,FF,F4,A3
DATA FF,F0,4A,6D,FF,F8,67,68,20,6D,FF,EA, C,10, 0,2D,67,92
DATA A,22,6D,FF,EA, C,11, 0,2B,66,2E, C,6D, 0,30,FF,F6,FC
DATA 66,26,53,6D,FF,F2,20,6D,FF,EA,52,AD,FF,EA,10,10,48,3
DATA 80,3F, 0,4E,92,54,4F,B0,7C,FF,FF,66, A,70,FF,4C,DF,76
DATA C,10,4E,5D,4E,75,60,16,3F,2D,FF,F6,4E,92,54,4F,B0,94
DATA 7C,FF,FF,66, 4,70,FF,60,E4,52,6D,FF,FA,30,2D,FF,F2,9D
DATA 53,6D,FF,F2,B0,6D,FF,F0,6E,DC,42,6D,FF,EE,60,20,20,43
DATA 6D,FF,EA,52,AD,FF,EA,10,10,48,80,3F, 0,4E,92,54,4F,E8
DATA B0,7C,FF,FF,66, 4,70,FF,60,B0,52,6D,FF,EE,20,6D,FF,4B
DATA EA,4A,10,67, A,30,2D,FF,EE,B0,6D,FF,F4,6D,CE,30,2D,A7
DATA FF,EE,D1,6D,FF,FA,4A,6D,FF,F8,66,28,60,18,3F,3C, 0,53
DATA 20,4E,92,54,4F,B0,7C,FF,FF,66, 6,70,FF,60, 0,FF,78,7F
DATA 52,6D,FF,FA,30,2D,FF,F2,53,6D,FF,F2,B0,6D,FF,F0,6E,31
DATA DA,60,16,3F, 4,4E,92,54,4F,B0,7C,FF,FF,66, 6,70,FF,1B
DATA 60, 0,FF,52,52,6D,FF,FA,60, 0,FD, A,30,2D,FF,FA,60,86
DATA 0,FF,42,48,E7,48, 0,42,84,4A,80,6A, 4,44,80,52,44,10
DATA 4A,81,6A, 6,44,81, A,44, 0, 1,61,3E,4A,44,67, 2,44,29
DATA 80,4C,DF, 0,12,4A,80,4E,75,48,E7,48, 0,42,84,4A,80,51
DATA 6A, 4,44,80,52,44,4A,81,6A, 2,44,81,61,1A,20, 1,60,C0
DATA D8,2F, 1,61,12,20, 1,22,1F,4A,80,4E,75,2F, 1,61, 6,1
DATA 22,1F,4A,80,4E,75,48,E7,30, 0,48,41,4A,41,66,20,48,F
DATA 41,36, 1,34, 0,42,40,48,40,80,C3,22, 0,48,40,32, 2,D7
DATA 82,C3,30, 1,42,41,48,41,4C,DF, 0, C,4E,75,48,41,26,2B
DATA 1,22, 0,42,41,48,41,48,40,42,40,74, F,DO,80,D3,81,60
DATA B6,81,62, 4,92,83,52,40,51,CA,FF,F2,4C,DF, 0, C,4E,D5
DATA 75,20,6F, 0, 4,20, 8,4A,18,66,FC,91,CO,20, 8,53,80,40
DATA 4E,75,4E,55, 0, 0,48,6C,80,AC,3F,2D, 0, 8,4E,BA, 0,C2
DATA 8,5C,4F,4E,5D,4E,75,4E,55, 0, 0,2F, 4,38,2D, 0, 8,64
DATA 2F,2D, 0, A,3F, 4,4E,BA, 0,30,5C,4F,B8,7C, 0, A,66,30
DATA 24,20,6D, 0, A,10,28, 0, C,48,80, 8, 0, 0, 7,67,14,51
DATA 3F,3C,FF,FF,2F,2D, 0, A,4E,BA, 0,F6,5C,4F,28,1F,4E,1D
DATA 5D,4E,75,60,F8,4E,55, 0, 0,2F, A,24,6D, 0, A,20,52,61
DATA B1,EA, 0, 4,65,18,30,2D, 0, 8,CO,7C, 0,FF,3F, 0,2F,2A
DATA A,4E,BA, 0,CA,5C,4F,24,5F,4E,5D,4E,75,20,52,52,92,CE
DATA 10,2D, 0, 9,10,80,48,80,CO,7C, 0,FF,60,E8,4E,55, 0,C4
DATA 0,2F, A,41,EC,80,96,24,48,20,4A,D5,FC, 0, 0, 0,16,39
DATA 2F, 8,61,10,58,4F,41,EC,82,4E,B5,C8,65,EA,24,5F,4E,E9
DATA 5D,4E,75,4E,55, 0, 0,48,E7, 8,20,24,6D, 0, 8,78, 0,2B
DATA 20, A,66, A,70,FF,4C,DF, 4,10,4E,5D,4E,75,4A,2A, 0,2A
DATA C,67,52, 8,2A, 0, 2, 0, C,67, C,3F,3C,FF,FF,2F, A,2A
DATA 61,54,5C,4F,38, 0,10,2A, 0, D,48,80,3F, 0,4E,BA, 4,F2
DATA F2,54,4F,88,40, 8,2A, 0, 1, 0, C,67, A,2F,2A, 0, 8,6E
DATA 4E,BA, 2,30,58,4F, 8,2A, 0, 5, 0, C,67,14,2F,2A, 0,F8
DATA 12,4E,BA, 2,C2,58,4F,2F,2A, 0,12,4E,BA, 2,14,58,4F,B5

DATA 42,92,42,AA, 0, 4,42,AA, 0, 8,42,2A, 0, C,30, 4,60,C4
DATA 8E,4E,55,FF,FE,48,E7, 8,20,24,6D, 0, 8,41,FA,FF,44,9C
DATA 29,48,82,7C, 8,2A, 0, 4, 0, C,67, A,70,FF,4C,DF, 4,C0
DATA 10,4E,5D,4E,75, 8,2A, 0, 2, 0, C,67,30,20,12,90,AA,C1
DATA 0, 8,38, 0,3F, 4,2F,2A, 0, 8,10,2A, 0, D,48,80,3F,32
DATA 0,4E,BA, 2,80,50,4F,B0,44,67,10, 8,EA, 0, 4, 0, C,96
DATA 42,92,42,AA, 0, 4,70,FF,60,C0, C,6D,FF,FF, 0, C,66,3C
DATA 10, 8,AA, 0, 2, 0, C,42,92,42,AA, 0, 4,70, 0,60,A8,C
DATA 4A,AA, 0, 8,66, 8,2F, A,4E,BA, 0,9A,58,4F, C,6A, 0,62
DATA 1, 0,10,66,2A,1B,6D, 0, D,FF,FF,3F,3C, 0, 1,48,6D,65
DATA FF,FF,10,2A, 0, D,48,80,3F, 0,4E,BA, 2,22,50,4F,B0,C7
DATA 7C, 0, 1,66,A0,30,2D, 0, C,60, 0,FF,6A,24,AA, 0, 8,8B
DATA 30,2A, 0,10,48,C0,D0,AA, 0, 8,25,40, 0, 4, 8,EA, 0,4F
DATA 2, 0, C,20,52,52,92,10,2D, 0, D,10,80,48,80,C0,7C,42
DATA 0,FF,60, 0,FF,3E,4E,55, 0, 0,2F, A,41,EC,80,96,24,DF
DATA 48,4A,2A, 0, C,67,18,D5,FC, 0, 0, 0,16,41,EC,82,4E,2B
DATA B5,C8,65, 8,70, 0,24,5F,4E,5D,4E,75,60,E2,42,92,42,A3
DATA AA, 0, 4,42,AA, 0, 8,20, A,60,EA,4E,55,FF,FC,2F, A,ED
DATA 24,6D, 0, 8,3F,3C, 4, 0,4E,BA, 0,CO,54,4F,2B,40,FF,ED
DATA FC,66,18,35,7C, 0, 1, 0,10,20, A,D0,BC, 0, 0, 0, E,0
DATA 25,40, 0, 8,24,5F,4E,5D,4E,75,35,7C, 4, 0, 0,10, 8,2B
DATA EA, 0, 1, 0, C,25,6D,FF,FC, 0, 8,10,2A, 0, D,48,80,9B
DATA 3F, 0,4E,BA, 0,E2,54,4F,4A,40,67, 6, 0,2A, 0,80, 0,6D
DATA C,60,CE,4E,55, 0, 0,48,E7, 0,30,24,6C,82,52,60,14,14
DATA 26,52,20,2A, 0, 4,50,80,2F, 0,2F, A,4E,BA, 4,18,50,72
DATA 4F,24,4B,20, A,66,E8,42,AC,82,52,4C,DF, C, 0,4E,5D,DA
DATA 4E,75,4E,55, 0, 0,2F, A,41,FA,FF,C6,29,48,82,80,42,54
DATA A7,20,2D, 0, 8,50,80,2F, 0,4E,BA, 3,C2,50,4F,24,40,CB
DATA 4A,80,66, 8,70, 0,24,5F,4E,5D,4E,75,24,AC,82,52,25,62
DATA 6D, 0, 8, 0, 4,29,4A,82,52,20, A,50,80,60,E6,4E,55,A3
DATA 0, 0,70, 0,30,2D, 0, 8,2F, 0,61,B2,58,4F,4E,5D,4E,B7
DATA 75,4E,55, 0, 0,48,E7, 0,30,97,CB,24,6C,82,52,60, E,AB
DATA 20,6D, 0, 8,51,88,B1,CA,67,12,26,4A,24,52,20, A,66,D8
DATA EE,70,FF,4C,DF, C, 0,4E,5D,4E,75,20, B,67, 4,26,92,50
DATA 60, 4,29,52,82,52,20,2A, 0, 4,50,80,2F, 0,2F, A,4E,87
DATA BA, 3,6A,50,4F,70, 0,60,D8,4E,55, 0, 0,2F, A,30,2D,A7
DATA 0, 8,C1,FC, 0, 6,24,40,D5,EC,82,62,4A,6D, 0, 8,6D,0
DATA E,30,2D, 0, 8,B0,6C,82,4E,6C, 4,4A,92,66, E,39,7C,D4
DATA 0, 2,82,84,70,FF,24,5F,4E,5D,4E,75,30,2D, 0, 8,C1,8E
DATA FC, 0, 6,20,6C,82,62,2F,30, 8, 0,4E,BA, 2,9C,58,4F,26
DATA 4A,80,67, 4,70, 1,60, 2,70, 0,60,D8,4E,55, 0, 0,2F,82
DATA 2D, 0, 8,4E,BA, 2,66,58,4F,4A,80,66, E,4E,BA, 2,70,4
DATA 39,40,82,84,70,FF,4E,5D,4E,75,70, 0,60,F8,4E,55, 0,C7
DATA 0,48,E7, C,20,38,2D, 0, 8,4E,BA, 0,70,30, 4,C1,FC,31
DATA 0, 6,24,40,D5,EC,82,62,4A,44,6D, A,B8,6C,82,4E,6C,74
DATA 4,4A,92,66,10,39,7C, 0, 2,82,84,70,FF,4C,DF, 4,30,E1
DATA 4E,5D,4E,75,30,2A, 0, 4,CO,7C, 0, 3,66, A,39,7C, 0,30
DATA 5,82,84,70,FF,60,E4,70, 0,30,2D, 0, E,2F, 0,2F,2D,24

DATA 0, A, 2F, 12, 4E, BA, 2, 2C, 4F, EF, 0, C, 2A, 0, B0, BC, FF, 60
DATA FF, FF, FF, 66, C, 4E, BA, 1, F0, 39, 40, 82, 84, 70, FF, 60, B8, 6E
DATA 20, 5, 60, B4, 4E, 55, FF, FC, 48, 78, 10, 0, 42, A7, 4E, BA, 2, 9A
DATA 94, 50, 4F, 2B, 40, FF, FC, 8, 0, 0, C, 67, 12, 4A, 6C, 82, 6A, C8
DATA 66, 8, 20, 2D, FF, FC, 4E, 5D, 4E, 75, 4E, BA, 0, 6, 70, 0, 60, 2
DATA F4, 4E, 55, 0, 0, 48, 78, 0, 4, 48, 7A, 0, 1E, 4E, BA, 1, C6, A
DATA 2F, 0, 4E, BA, 1, C8, 4F, EF, 0, C, 3F, 3C, 0, 1, 4E, BA, 0, CE
DATA C, 54, 4F, 4E, 5D, 4E, 75, 5E, 43, A, 0, 4E, 55, 0, 0, 4A, AC, 61
DATA 82, 7C, 67, 6, 20, 6C, 82, 7C, 4E, 90, 3F, 2D, 0, 8, 4E, BA, 0, 4F
DATA 8, 54, 4F, 4E, 5D, 4E, 75, 4E, 55, FF, FC, 2F, 4, 30, 2D, 0, 8, 4F
DATA 48, C0, 2B, 40, FF, FC, 4A, AC, 82, 62, 67, 28, 78, 0, 60, A, 3F, F8
DATA 4, 4E, BA, 0, D0, 54, 4F, 52, 44, B8, 6C, 82, 4E, 6D, F0, 30, 2C, C2
DATA 82, 4E, C1, FC, 0, 6, 2F, 0, 2F, 2C, 82, 62, 4E, BA, 1, B4, 50, E
DATA 4F, 4A, AC, 82, 80, 67, 6, 20, 6C, 82, 80, 4E, 90, 4A, AC, 82, 86, 1E
DATA 67, A, 2F, 2C, 82, 86, 4E, BA, 1, 66, 58, 4F, 4A, AC, 82, 8A, 67, 53
DATA A, 2F, 2C, 82, 8A, 4E, BA, 1, 56, 58, 4F, 4A, AC, 82, 8E, 67, A, EE
DATA 2F, 2C, 82, 8E, 4E, BA, 1, 46, 58, 4F, 2C, 78, 0, 4, 8, 2E, 0, 3F
DATA 4, 1, 29, 67, 14, 2F, D, 4B, FA, 0, A, 4E, AE, FF, E2, 2A, 5F, 9A
DATA 60, 6, 42, A7, F3, 5F, 4E, 73, 4A, AC, 82, 6C, 66, 30, 4A, AC, 82, 54
DATA 78, 67, 28, 30, 2C, 82, 76, 48, C0, 2F, 0, 2F, 2C, 82, 78, 4E, BA, EF
DATA 1, 3A, 50, 4F, 30, 2C, 82, 74, 52, 40, 48, C0, E5, 80, 2F, 0, 2F, 89
DATA 2C, 82, 70, 4E, BA, 1, 24, 50, 4F, 60, E, 4E, BA, 1, 14, 2F, 2C, D0
DATA 82, 6C, 4E, BA, 1, 40, 58, 4F, 20, 2D, FF, FC, 2E, 6C, 82, 56, 4E, E6
DATA 75, 28, 1F, 4E, 5D, 4E, 75, 4E, 55, 0, 0, 48, E7, E, 20, 38, 2D, 8F
DATA 0, 8, 30, 4, C1, FC, 0, 6, 24, 40, D5, EC, 82, 62, 4A, 44, 6D, 3
DATA A, B8, 6C, 82, 4E, 6C, 4, 4A, 92, 66, 10, 39, 7C, 0, 2, 82, 84, 7D
DATA 70, FF, 4C, DF, 4, 70, 4E, 5D, 4E, 75, 30, 2A, 0, 4, C0, 7C, 80, 96
DATA 0, 66, 8, 2F, 12, 4E, BA, 0, A, 58, 4F, 42, 92, 70, 0, 60, E0, EC
DATA 22, 2F, 0, 4, 2C, 6C, 82, 5E, 4E, EE, FF, DC, 22, 2F, 0, 4, 2C, 65
DATA 6C, 82, 5E, 4E, EE, FF, 82, 22, 2F, 0, 4, 2C, 6C, 82, 5E, 4E, EE, 12
DATA FF, B8, 2C, 6C, 82, 5E, 4E, EE, FF, CA, 2C, 6C, 82, 5E, 4E, EE, FF, E7
DATA 7C, 22, 2F, 0, 4, 2C, 6C, 82, 5E, 4E, EE, FF, 28, 4C, EF, 0, 6, ED
DATA 0, 4, 2C, 6C, 82, 5E, 4E, EE, FF, E2, 2C, 6C, 82, 5E, 4E, EE, FF, 4C
DATA C4, 4C, EF, 0, E, 0, 4, 2C, 6C, 82, 5E, 4E, EE, FF, D0, 48, E7, C3
DATA 1, 4, 4C, EF, 20, 80, 0, C, 2C, 6C, 82, 5A, 4E, AE, FF, 94, 4C, 3B
DATA DF, 20, 80, 4E, 75, 4E, FA, 0, 2, 22, 6F, 0, 4, 2C, 6C, 82, 5A, 95
DATA 4E, EE, FE, 62, 4C, EF, 0, 3, 0, 4, 2C, 6C, 82, 5A, 4E, EE, FF, 8D
DATA 3A, 4E, FA, 0, 2, 22, 6F, 0, 4, 2C, 6C, 82, 5A, 4E, EE, FE, DA, A1
DATA 2C, 6C, 82, 5A, 4E, EE, FF, 7C, 22, 6F, 0, 4, 20, 2F, 0, 8, 2C, 43
DATA 6C, 82, 5A, 4E, EE, FF, 2E, 20, 6F, 0, 4, 2C, 6C, 82, 5A, 4E, EE, F4
DATA FE, 8C, 2C, 6C, 82, 5A, 22, 6F, 0, 4, 20, 2F, 0, 8, 4E, EE, FD, 23
DATA D8, 22, 6F, 0, 4, 2C, 6C, 82, 5A, 4E, EE, FE, 86, 4C, EF, 0, 3, DF
DATA 0, 4, 2C, 6C, 82, 5A, 4E, EE, FE, CE, 22, 6F, 0, 4, 20, 2F, 0, 64
DATA 8, 2C, 6C, 82, 5A, 4E, EE, FE, D4, 20, 6F, 0, 4, 2C, 6C, 82, 5A, 91
DATA 4E, EE, FE, 80, 4C, EF, 3, 0, 0, 4, 2C, 6C, 82, 92, 4E, EE, FF, E3
DATA A0, 20, 6F, 0, 4, 2C, 6C, 82, 92, 4E, EE, FF, A6, 20, 6F, 0, 4, 53
DATA 2C, 6C, 82, 92, 4E, EE, FF, B2, 0, 0, 3, EC, 0, 0, 0, 1, 0, 89
DATA 0, 0, 1, 0, 0, 2, 8C, 0, 0, 0, 0, 0, 0, 0, 0, 3, F0, 0, 0, 82

DATA 0, 2,5F,5F,48,30,5F,6F,72,67, 0, 0, 0, 0, 0, 0, 0, 0,DF
DATA 2,5F,6D,61,69,6E, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 2,C
DATA 5F,56,61,6C,75,65, 0, 0, 0, 0, 1,3E, 0, 0, 0, 2,2E,CB
DATA 62,65,67,69,6E, 0, 0, 0, 0, 2,18, 0, 0, 0, 2,5F,67,E7
DATA 65,74,61,34, 0, 0, 0, 0, 2,8A, 0, 0, 0, 2,5F,5F,6D,27
DATA 61,69,6E, 0, 0, 0, 0, 2,92, 0, 0, 0, 3,5F,5F,63,6C,5C
DATA 69,5F,70,61,72,73,65, 0, 0, 0, 3,CE, 0, 0, 0, 2,5F,15
DATA 73,74,72,6E,63,70,79, 0, 0, 5,74, 0, 0, 0, 2,5F,73,60
DATA 74,72,63,61,74, 0, 0, 0, 5,96, 0, 0, 0, 2,5F,73,74,1
DATA 72,6E,63,61,74, 0, 0, 5,9C, 0, 0, 0, 2,5F,73,74,72,73
DATA 63,70,79, 0, 0, 0, 5,C0, 0, 0, 0, 3,5F,5F,77,62,5F,A
DATA 70,61,72,73,65, 0, 0, 0, 0, 5,D0, 0, 0, 0, 2,5F,70,C1
DATA 72,69,6E,74,66, 0, 0, 0, 6,84, 0, 0, 0, 2,5F,66,6F,E3
DATA 72,6D,61,74, 0, 0, 0, 7,2E, 0, 0, 0, 2,2E,64,69,76,5C
DATA 73, 0, 0, 0, 0, 0, A,48, 0, 0, 0, 2,2E,6D,6F,64,73,A8
DATA 0, 0, 0, 0, 0, A,70, 0, 0, 0, 2,2E,6D,6F,64,75, 0,5F
DATA 0, 0, 0, 0, A,8A, 0, 0, 0, 2,2E,64,69,76,75, 0, 0,7C
DATA 0, 0, 0, A,96, 0, 0, 0, 2,5F,73,74,72,6C,65,6E, 0,99
DATA 0, 0, A,F0, 0, 0, 0, 2,5F,70,75,74,63,68,61,72, 0,52
DATA 0, B, 2, 0, 0, 0, 2,5F,61,70,75,74,63, 0, 0, 0, 0,8B
DATA B,18, 0, 0, 0, 2,5F,70,75,74,63, 0, 0, 0, 0, 0, B,4B
DATA 5A, 0, 0, 0, 2,5F,66,63,6C,6F,73,65, 0, 0, 0, B,BE,0
DATA 0, 0, 0, 2,5F,66,6C,73,68,5F, 0, 0, 0, 0, C,44, 0,BD
DATA 0, 0, 3,5F,6E,65,77,73,74,72,65,61,6D, 0, 0, 0, 0,38
DATA D,26, 0, 0, 0, 2,5F,67,65,74,62,75,66,66, 0, 0, D,84
DATA 5E, 0, 0, 0, 2,5F,6C,6D,61,6C,6C,6F,63, 0, 0, D,EE,9E
DATA 0, 0, 0, 2,5F,6D,61,6C,6C,6F,63, 0, 0, 0, E,2E, 0,15
DATA 0, 0, 2,5F,66,72,65,65, 0, 0, 0, 0, 0, E,42, 0, 0,53
DATA 0, 2,5F,69,73,61,74,74,79, 0, 0, 0, E,8E, 0, 0, 0,9B
DATA 2,5F,75,6E,6C,69,6E,6B, 0, 0, 0, E,E6, 0, 0, 0, 2,E8
DATA 5F,77,72,69,74,65, 0, 0, 0, 0, F, A, 0, 0, 0, 3,5F,5
DATA 43,68,6B,5F,41,62,6F,72,74, 0, 0, 0, 0, F,88, 0, 0,4
DATA 0, 2,5F,5F,61,62,6F,72,74, 0, 0, 0, F,B8, 0, 0, 0,9F
DATA 2,5F,65,78,69,74, 0, 0, 0, 0, 0, F,E4, 0, 0, 0, 2,10
DATA 5F,5F,65,78,69,74, 0, 0, 0, 0,10, 2, 0, 0, 0, 2,5F,EB
DATA 63,6C,6F,73,65, 0, 0, 0, 0,10,F0, 0, 0, 0, 2,5F,5F,D6
DATA 43,6C,6F,73,65, 0, 0, 0,11,3E, 0, 0, 0, 3,5F,5F,43,49
DATA 75,72,72,65,6E,74,44,69,72, 0, 0,11,4A, 0, 0, 0, 3,1D
DATA 5F,5F,44,65,6C,65,74,65,46,69,6C,65, 0, 0,11,56, 0,F8
DATA 0, 0, 2,5F,5F,49,6E,70,75,74, 0, 0, 0,11,62, 0, 0,43
DATA 0, 2,5F,5F,49,6F,45,72,72, 0, 0, 0,11,6A, 0, 0, 0,1C
DATA 4,5F,5F,49,73,49,6E,74,65,72,61,63,74,69,76,65, 0,FC
DATA 0, 0,11,72, 0, 0, 0, 2,5F,5F,4F,70,65,6E, 0, 0, 0,D5
DATA 0,11,7E, 0, 0, 0, 2,5F,5F,4F,75,74,70,75,74, 0, 0,E0
DATA 11,8C, 0, 0, 0, 2,5F,5F,57,72,69,74,65, 0, 0, 0,11,79
DATA 94, 0, 0, 0, 2,5F,41,6C,65,72,74, 0, 0, 0, 0,11,A2,A0
DATA 0, 0, 0, 4,5F,43,6C,6F,73,65,4C,69,62,72,61,72,79,2E
DATA 0, 0, 0, 0, 0,11,BA, 0, 0, 0, 4,5F,5F,43,6C,6F,73,1E


```

DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
DATA 0, 0, 0, 0, 0, 14, 0, 0, 0, 0, 3, F0, 0, 0, 0, 2, 5F, 68
DATA 5F, 48, 31, 5F, 6F, 72, 67, 0, 0, 0, 0, 0, 0, 2, 5F, 63, 43
DATA 74, 70, 5F, 0, 0, 0, 0, 0, 12, 0, 0, 0, 2, 5F, 43, 62, 5B
DATA 75, 66, 66, 73, 0, 0, 0, 0, 94, 0, 0, 0, 2, 5F, 5F, 6E, 75, EB
DATA 6D, 64, 65, 76, 0, 0, 2, 4C, 0, 0, 0, 2, 5F, 5F, 48, 31, 5F, 92
DATA 65, 6E, 64, 0, 0, 2, 50, 0, 0, 0, 2, 5F, 5F, 48, 32, 5F, 6F, 91
DATA 72, 67, 0, 0, 2, 50, 0, 0, 0, 2, 5F, 5F, 73, 61, 76, 73, 70, 18
DATA 0, 0, 0, 2, 54, 0, 0, 0, 2, 5F, 53, 79, 73, 42, 61, 73, 65, 71
DATA 0, 0, 2, 58, 0, 0, 0, 2, 5F, 44, 4F, 53, 42, 61, 73, 65, 0, 1C
DATA 0, 2, 5C, 0, 0, 0, 2, 5F, 5F, 64, 65, 76, 74, 61, 62, 0, 0, 94
DATA 2, 60, 0, 0, 0, 3, 5F, 5F, 73, 74, 6B, 62, 61, 73, 65, 0, 0, 10
DATA 0, 0, 0, 2, 64, 0, 0, 0, 4, 5F, 45, 6E, 61, 62, 6C, 65, 5F, 6F
DATA 41, 62, 6F, 72, 74, 0, 0, 0, 0, 0, 2, 68, 0, 0, 0, 3, 5F, C4
DATA 57, 42, 65, 6E, 63, 68, 4D, 73, 67, 0, 0, 0, 0, 2, 6A, 0, 0, CA
DATA 0, 2, 5F, 5F, 61, 72, 67, 76, 0, 0, 0, 0, 2, 6E, 0, 0, 0, E0
DATA 2, 5F, 5F, 61, 72, 67, 63, 0, 0, 0, 0, 2, 72, 0, 0, 0, 3, D4
DATA 5F, 5F, 61, 72, 67, 5F, 6C, 65, 6E, 0, 0, 0, 0, 0, 2, 74, 0, C
DATA 0, 0, 3, 5F, 5F, 61, 72, 67, 5F, 6C, 69, 6E, 0, 0, 0, 0, 0, 9D
DATA 2, 76, 0, 0, 0, 2, 5F, 63, 6C, 73, 5F, 0, 0, 0, 0, 0, 2, 7C
DATA 7A, 0, 0, 0, 2, 5F, 5F, 63, 6C, 6E, 0, 0, 0, 0, 0, 2, 7E, F7
DATA 0, 0, 0, 2, 5F, 65, 72, 72, 6E, 6F, 0, 0, 0, 0, 2, 82, 0, B
DATA 0, 0, 4, 5F, 4D, 61, 74, 68, 54, 72, 61, 6E, 73, 42, 61, 73, 65, 70
DATA 0, 0, 0, 0, 2, 84, 0, 0, 0, 3, 5F, 4D, 61, 74, 68, 42, 61, 15
DATA 73, 65, 0, 0, 0, 0, 0, 2, 88, 0, 0, 0, 5, 5F, 4D, 61, 74, E8
DATA 68, 49, 65, 65, 65, 44, 6F, 75, 62, 42, 61, 73, 42, 61, 73, 65, 0, FB
DATA 0, 2, 8C, 0, 0, 0, 3, 5F, 49, 63, 6F, 6E, 42, 61, 73, 65, 0, F4
DATA 0, 0, 0, 0, 2, 90, 0, 0, 0, 2, 5F, 5F, 48, 32, 5F, 65, 6E, FE
DATA 64, 0, 0, 2, 94, 0, 0, 0, 0, 0, 3, F2, 0, 0, 3, EB, DD
DATA 0, 0, 0, 1, 0, 0, 3, F2,

```

7.6.3 BASIC-lader voor TASKSTOP

```

OPEN "ram:TaskStop" FOR OUTPUT AS #1

ON ERROR GOTO fout
PRINT "Even wachten alstublieft."

WHILE 1
  regel=regel+1
  som=0
  FOR i=1 TO 17

```

```

READ Dat$
CALL Dec(Dat$)
som=som+waarde
PRINT#1,CHR$(waarde);
NEXT i

```

```

READ proef$
CALL Dec(proef$)

```

```

IF waarde <> som-INT(som/256)*256 THEN
  PRINT "Fout in DATA-regel ";regel:STOP
END IF
WEND

```

```

SUB Dec(waarde$) STATIC
  SHARED waarde
  waarde=0
  FOR j=1 TO LEN(waarde$)
    decwaarde=ASC(RIGHT$(waarde$,j))
    IF decwaarde > 57 THEN decwaarde=decwaarde-7
    waarde=waarde+(decwaarde-48)*16^(j-1)
  NEXT j
END SUB

```

```

fout:
IF ERR <> 4 THEN PRINT "Error-Code";ERR:STOP
PRINT "O.K."
PRINT "Kopieer het bestand van de RAM-disk naar"
PRINT "uw CLI-werkdiskette."

```

```

CLOSE #1
END

```

```

DATA 0, 0, 3,F3, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0,F9
DATA 0, 0, 2, 0, 0, 4,55, 0, 0, 0,A5, 0, 0, 0, 1, 0, 0,1
DATA 3,E9, 0, 0, 4,55,4E,FA, 0,E6,4E,55,FF,F6,42,6D,FF,B9
DATA F6,2B,7C, 0,BF,EC, 1,FF,F8,48,7A, 0,78,4E,BA, 5,3A,C1
DATA 58,4F,42,A7,4E,BA,10,92,58,4F,2B,40,FF,FC,4A,AD,FF,3D
DATA FC,66,12,48,7A, 0,98,4E,BA, 5,1E,58,4F,42,A7,4E,BA,91
DATA E,76,58,4F,48,78, 0,7F,2F,2D,FF,FC,4E,BA,10,C6,50,EF
DATA 4F,20,6D,FF,F8, C,10, 0,3F,66,1C,52,6D,FF,F6,20,6D,F1
DATA FF,F8, C,10, 0,3D,67, A,22,6D,FF,F8, C,11, 0,39,66,3
DATA E8,52,6D,FF,F6,48,78, 0, A,4E,BA, F,AA,58,4F,20,6D,5B
DATA FF,F8, C,10, 0,39,66,C6,4E,5D,4E,75, A,20,53,68,69,34
DATA 66,74,4C,69,6E,6B,73,20,3D,20,53,74,6F,70,2C,20,53,9D
DATA 68,69,66,74,52,65,63,68,74,73,20,3D,20,57,65,69,74,2A
DATA 65,72,2C,20,43,6F,6E,74,72,6F,6C,20,3D,20,45,6E,64,98

```

DATA 65, A,20, 0,20,49,63,68,20,66,69,6E,64,65,20,6D,65,DB
DATA 69,6E,65,6E,20,54,61,73,6B,20,6E,69,63,68,74, A, 0,9D
DATA 61,70,43,EC,82,52,45,EC,82,52,B5,C9,66, E,32,3C, 0,39
DATA 10,6B, 8,74, 0,22,C2,51,C9,FF,FC,29,4F,82,56,2C,78,E4
DATA 0, 4,29,4E,82,5A,48,E7,80,80, 8,2E, 0, 4, 1,29,67,51
DATA 10,4B,FA, 0, 8,4E,AE,FF,E2,60, 6,42,A7,F3,5F,4E,73,9C
DATA 43,FA, 0,20,4E,AE,FE,68,29,40,82,5E,66, C,2E,3C, 0,E4
DATA 3,80, 7,4E,AE,FF,94,60, 4,4E,BA, 0,1A,50,4F,4E,75,1
DATA 64,6F,73,2E,6C,69,62,72,61,72,79, 0,49,F9, 0, 0,7F,2A
DATA FE,4E,75,4E,55, 0, 0,2F, A,48,79, 0, 1, 0, 0,30,2C,BB
DATA 82,4E,C1,FC, 0, 6,2F, 0,4E,BA, F,2C,50,4F,29,40,82,8F
DATA 62,66,14,42,A7,48,79, 0, 1, 0, 0,4E,BA, E,FO,50,4F,2C
DATA 2E,6C,82,56,4E,75,20,6C,82,62,42,68, 0, 4,20,6C,82,61
DATA 62,31,7C, 0, 1, 0,10,22,6C,82,62,33,7C, 0, 1, 0, A,4C
DATA 20,6C,82,56,20,2C,82,56,90,A8, 0, 4,50,80,29,40,82,7F
DATA 66,20,6C,82,66,20,BC,4D,41,4E,58,42,A7,4E,BA, E,E4,CD
DATA 58,4F,24,40,4A,AA, 0,AC,67,2E,2F,2D, 0, C,2F,2D, 0,4
DATA 8,2F, A,4E,BA, 0,B2,4F,EF, 0, C,39,7C, 0, 1,82,6A,E7
DATA 20,6C,82,62, 0,68,80, 0, 0, 4,20,6C,82,62, 0,68,80,B4
DATA 0, 0, A,60,44,48,6A, 0,5C,4E,BA, F, E,58,4F,48,6A,3A
DATA 0,5C,4E,BA, E,BE,58,4F,29,40,82,6C,20,6C,82,6C,4A,F2
DATA A8, 0,24,67,10,20,6C,82,6C,22,68, 0,24,2F,11,4E,BA,B3
DATA D,DE,58,4F,2F,2C,82,6C,2F, A,4E,BA, 2,58,50,4F,29,3E
DATA 6C,82,6C,82,70,4E,BA, D,EA,20,6C,82,62,20,80,4E,BA,63
DATA E, A,20,6C,82,62,21,40, 0, 6,67,16,48,78, 3,ED,48,64
DATA 7A, 0,2C,4E,BA, D,E6,50,4F,20,6C,82,62,21,40, 0, C,1D
DATA 2F,2C,82,70,3F,2C,82,74,4E,BA,FD,7A,5C,4F,42,67,4E,CF
DATA BA, C,22,54,4F,24,5F,4E,5D,4E,75,2A, 0,4E,55, 0, 0,49
DATA 48,E7, C,30,24,6D, 0,10,20,6D, 0, 8,20,28, 0,AC,E5,7A
DATA 80,28, 0,20,44,20,28, 0,10,E5,80,26,40,10,13,48,80,1A
DATA 48,C0,D0,AD, 0, C,54,80,39,40,82,76,42,A7,30,2C,82,9D
DATA 76,48,C0,2F, 0,4E,BA, D,CA,50,4F,29,40,82,78,66, 8,FC
DATA 4C,DF, C,30,4E,5D,4E,75,10,13,48,80,3F, 0,20,4B,52,BC
DATA 88,2F, 8,2F,2C,82,78,4E,BA, 1,44,4F,EF, 0, A,48,7A,6B
DATA 1,3A,10,13,48,80,48,C0,D0,AC,82,78,2F, 0,4E,BA, 1,DC
DATA 78,50,4F,3F,2D, 0, E,2F, A,2F,2C,82,78,4E,BA, 1,44,6C
DATA 4F,EF, 0, A,42,6C,82,74,26,6C,82,78,24,4B,10,13,48,52
DATA 80,3A, 0,80,7C, 0,20,67,18,BA,7C, 0, 9,67,12,BA,7C,73
DATA 0, C,67, C,BA,7C, 0, D,67, 6,BA,7C, 0, A,66, 4,52,2B
DATA 8B,60,D8, C,13, 0,20,6D,7A, C,13, 0,22,66,2E,52,8B,9B
DATA 20,4B,52,8B,10,10,48,80,3A, 0,67,1E,20,4A,52,8A,10,45
DATA 85,BA,7C, 0,22,66,10, C,13, 0,22,66, 4,52,8B,60, 6,41
DATA 42,2A,FF,FF,60, 2,60,D6,60,38,20,4B,52,8B,10,10,48,4A
DATA 80,3A, 0,67,26,BA,7C, 0,20,67,20,BA,7C, 0, 9,67,1A,E4
DATA BA,7C, 0, C,67,14,BA,7C, 0, D,67, E,BA,7C, 0, A,67,1C
DATA 8,20,4A,52,8A,10,85,60,CE,20,4A,52,8A,42,10,4A,45,38
DATA 66, 2,53,8B,52,6C,82,74,60, 0,FF,5A,42,12,42,A7,30,20
DATA 2C,82,74,52,40,48,C0,E5,80,2F, 0,4E,BA, C,B4,50,4F,B7

DATA 29,40,82,70,66, 8,42,6C,82,74,60, 0,FE,E4,7A, 0,26,4F
DATA 6C,82,78,60,1E,30, 5,48,C0,E5,80,20,6C,82,70,21,8B,B0
DATA 8, 0,2F, B,4E,BA, 5,A2,58,4F,52,40,48,C0,D7,C0,52,1B
DATA 45,BA,6C,82,74,6D,DC,30, 5,48,C0,E5,80,20,6C,82,70,CA
DATA 42,B0, 8, 0,60, 0,FE,A6,20, 0,4C,EF, 3, 0, 0, 4,20,80
DATA 8,32,2F, 0, C,60, 2,10,D9,57,C9,FF,FC,67, 6,52,41,DB
DATA 60, 2,42,18,51,C9,FF,FC,4E,75,30,3C,7F,FF,60, 4,30,12
DATA 2F, 0, C,20,6F, 0, 4,4A,18,66,FC,53,48,22,6F, 0, 8,C6
DATA 53,40,10,D9,57,C8,FF,FC,67, 2,42,10,20,2F, 0, 4,4E,F2
DATA 75,20,6F, 0, 4,20, 8,22,6F, 0, 8,10,D9,66,FC,4E,75,D7
DATA 4E,55, 0, 0,48,E7, E,30,24,6D, 0, 8,42,A7,48,7A, 0,54
DATA 8E,4E,BA, C,34,50,4F,29,40,82,92,66, 8,4C,DF, C,70,7
DATA 4E,5D,4E,75,20,6D, 0, C,22,68, 0,24,2F,29, 0, 4,4E,5F
DATA BA, C,74,58,4F,28, 0,67,52,48,7A, 0,6D,20,44,2F,28,AC
DATA 0,36,4E,BA, C,46,50,4F,26,40,4A,80,67,34,48,78, 3,BD
DATA ED,2F, B,4E,BA, B,60,50,4F,2C, 0,67,24,20, 6,E5,80,7B
DATA 2A, 0,20,45,25,68, 0, 8, 0,A4,25,46, 0,9C,48,78, 3,92
DATA ED,48,7A, 0,38,4E,BA, B,3C,50,4F,25,40, 0,A0,2F, 4,D
DATA 4E,BA, C,12,58,4F,2F,2C,82,92,4E,BA, B,62,58,4F,42,9A
DATA AC,82,92,60,80,69,63,6F,6E,2E,6C,69,62,72,61,72,79,6C
DATA 0,57,49,4E,44,4F,57, 0,2A, 0,4E,55, 0, 0,48,6D, 0,5A
DATA C,2F,2D, 0, 8,48,7A, 4,70,4E,BA, 0,98,4F,EF, 0, C,90
DATA 4E,5D,4E,75,4E,55, 0, 0,48,E7, 8,20,24,6D, 0, E, C,13
DATA 6D, 0, 4, 0,12,66, 8,20,6D, 0, 8,28,10,60,1C,4A,6D,F1
DATA 0, C,6F, C,20,6D, 0, 8,70, 0,30,10,28, 0,60, A,20,7E
DATA 6D, 0, 8,30,10,48,C0,28, 0,42,6D, 0,12,4A,6D, 0, C,69
DATA 6C,10,44,6D, 0, C,4A,84,6C, 8,44,84,3B,7C, 0, 1, 0,FB
DATA 12,32,2D, 0, C,48,C1,20, 4,4E,BA, 3,8E,41,EC,80, 2,F2
DATA 53,8A,14,B0, 0, 0,32,2D, 0, C,48,C1,20, 4,4E,BA, 3,44
DATA 84,28, 0,66,DA,4A,6D, 0,12,67, 6,53,8A,14,BC, 0,2D,FC
DATA 20, A,4C,DF, 4,10,4E,5D,4E,75,4E,55,FF,22,48,E7, 8,D2
DATA 30,24,6D, 0, 8,26,6D, 0, C,42,6D,FF,FA,2B,6D, 0,10,B8
DATA FF,FC,20,4B,52,8B,10,10,48,80,38, 0,67, 0, 2,EC,B8,70
DATA 7C, 0,25,66, 0, 2,CA,42,2D,FF,30,3B,7C, 0, 1,FF,F8,20
DATA 3B,7C, 0,20,FF,F6,3B,7C,27,10,FF,F4,20,4B,52,8B,10,5
DATA 10,48,80,38, 0,B0,7C, 0,2D,66, E,42,6D,FF,F8,20,4B,EE
DATA 52,8B,10,10,48,80,38, 0,B8,7C, 0,30,66,10,3B,7C, 0,8E
DATA 30,FF,F6,20,4B,52,8B,10,10,48,80,38, 0,B8,7C, 0,2A,EB
DATA 66,18,20,6D,FF,FC,54,AD,FF,FC,3B,50,FF,F2,20,4B,52,3B
DATA 8B,10,10,48,80,38, 0,60,32,42,6D,FF,F2,60,1C,30,2D,B6
DATA FF,F2,C1,FC, 0, A,DO,44,90,7C, 0,30,3B,40,FF,F2,20,94
DATA 4B,52,8B,10,10,48,80,38, 0,30, 4,52,40,41,EC,80,14,CF
DATA 8,30, 0, 2, 0, 0,66,D4,B8,7C, 0,2E,66,5A,20,4B,52,53
DATA 8B,10,10,48,80,38, 0,B0,7C, 0,2A,66,18,20,6D,FF,FC,7
DATA 54,AD,FF,FC,3B,50,FF,F4,20,4B,52,8B,10,10,48,80,38,E2
DATA 0,60,32,42,6D,FF,F4,60,1C,30,2D,FF,F4,C1,FC, 0, A,C7
DATA DO,44,90,7C, 0,30,3B,40,FF,F4,20,4B,52,8B,10,10,48,6E
DATA 80,38, 0,30, 4,52,40,41,EC,80,14, 8,30, 0, 2, 0, 0,79

DATA 66,D4,3B,7C, 0, 2,FF,F0,B8,7C, 0,6C,66,12,20,4B,52,B7
DATA 8B,10,10,48,80,38, 0,3B,7C, 0, 4,FF,F0,60,10,B8,7C,F9
DATA 0,68,66, A,20,4B,52,8B,10,10,48,80,38, 0,30, 4,48,BC
DATA C0,60,78,3B,7C, 0, 8,FF,EE,60,16,3B,7C, 0, A,FF,EE,68
DATA 60, E,3B,7C, 0,10,FF,EE,60, 6,3B,7C,FF,F6,FF,EE,3F,60
DATA 2D,FF,F0,48,6D,FF,30,3F,2D,FF,EE,2F,2D,FF,FC,4E,BA,B8
DATA FD,E4,4F,EF, 0, C,2B,40,FF,EA,30,2D,FF,F0,48,C0,D1,A4
DATA AD,FF,FC,60,5A,20,6D,FF,FC,58,AD,FF,FC,2B,50,FF,EA,4E
DATA 2F,2D,FF,EA,4E,BA, 2, C,58,4F,3B,40,FF,F0,60,4A,20,36
DATA 6D,FF,FC,54,AD,FF,FC,38,10,41,ED,FF,2F,2B,48,FF,EA,64
DATA 10,84,60,28,90,BC, 0, 0, 0,63,67,E2,53,80,67,94,90,72
DATA BC, 0, 0, 0, B,67, 0,FF,74,59,80,67,B4,55,80,67, 0,D1
DATA FF,72,57,80,67, 0,FF,74,60,CC,41,ED,FF,30,91,ED,FF,28
DATA EA,3B,48,FF,F0,30,2D,FF,F0,B0,6D,FF,F4,6F, 6,3B,6D,D5
DATA FF,F4,FF,F0,4A,6D,FF,F8,67,68,20,6D,FF,EA, C,10, 0,F1
DATA 2D,67, A,22,6D,FF,EA, C,11, 0,2B,66,2E, C,6D, 0,30,9B
DATA FF,F6,66,26,53,6D,FF,F2,20,6D,FF,EA,52,AD,FF,EA,10,A0
DATA 10,48,80,3F, 0,4E,92,54,4F,B0,7C,FF,FF,66, A,70,FF,A3
DATA 4C,DF, C,10,4E,5D,4E,75,60,16,3F,2D,FF,F6,4E,92,54,C0
DATA 4F,B0,7C,FF,FF,66, 4,70,FF,60,E4,52,6D,FF,FA,30,2D,AB
DATA FF,F2,53,6D,FF,F2,B0,6D,FF,F0,6E,DC,42,6D,FF,EE,60,F4
DATA 20,20,6D,FF,EA,52,AD,FF,EA,10,10,48,80,3F, 0,4E,92,85
DATA 54,4F,B0,7C,FF,FF,66, 4,70,FF,60,B0,52,6D,FF,EE,20,82
DATA 6D,FF,EA,4A,10,67, A,30,2D,FF,EE,B0,6D,FF,F4,6D,CE,B6
DATA 30,2D,FF,EE,D1,6D,FF,FA,4A,6D,FF,F8,66,28,60,18,3F,74
DATA 3C, 0,20,4E,92,54,4F,B0,7C,FF,FF,66, 6,70,FF,60, 0,44
DATA FF,78,52,6D,FF,FA,30,2D,FF,F2,53,6D,FF,F2,B0,6D,FF,4A
DATA F0,6E,DA,60,16,3F, 4,4E,92,54,4F,B0,7C,FF,FF,66, 6,A
DATA 70,FF,60, 0,FF,52,52,6D,FF,FA,60, 0,FD, A,30,2D,FF,9B
DATA FA,60, 0,FF,42,48,E7,48, 0,42,84,4A,80,6A, 4,44,80,D4
DATA 52,44,4A,81,6A, 6,44,81, A,44, 0, 1,61,3E,4A,44,67,79
DATA 2,44,80,4C,DF, 0,12,4A,80,4E,75,48,E7,48, 0,42,84,CD
DATA 4A,80,6A, 4,44,80,52,44,4A,81,6A, 2,44,81,61,1A,20,29
DATA 1,60,D8,2F, 1,61,12,20, 1,22,1F,4A,80,4E,75,2F, 1,FB
DATA 61, 6,22,1F,4A,80,4E,75,48,E7,30, 0,48,41,4A,41,66,E
DATA 20,48,41,36, 1,34, 0,42,40,48,40,80,C3,22, 0,48,40,B
DATA 32, 2,82,C3,30, 1,42,41,48,41,4C,DF, 0, C,4E,75,48,F8
DATA 41,26, 1,22, 0,42,41,48,41,48,40,42,40,74, F,DO,80,73
DATA D3,81,B6,81,62, 4,92,83,52,40,51,CA,FF,F2,4C,DF, 0,CF
DATA C,4E,75,20,6F, 0, 4,20, 8,4A,18,66,FC,91,C0,20, 8,C7
DATA 53,80,4E,75,4E,55, 0, 0,48,6C,80,AC,3F,2D, 0, 8,4E,DB
DATA BA, 0, 8,5C,4F,4E,5D,4E,75,4E,55, 0, 0,2F, 4,38,2D,16
DATA 0, 8,2F,2D, 0, A,3F, 4,4E,BA, 0,30,5C,4F,B8,7C, 0,C8
DATA A,66,24,20,6D, 0, A,10,28, 0, C,48,80, 8, 0, 0, 7,46
DATA 67,14,3F,3C,FF,FF,2F,2D, 0, A,4E,BA, 0,F6,5C,4F,28,2B
DATA 1F,4E,5D,4E,75,60,F8,4E,55, 0, 0,2F, A,24,6D, 0, A,5C
DATA 20,52,B1,EA, 0, 4,65,18,30,2D, 0, 8,C0,7C, 0,FF,3F,6D
DATA 0,2F, A,4E,BA, 0,CA,5C,4F,24,5F,4E,5D,4E,75,20,52,19

DATA 52,92,10,2D, 0, 9,10,80,48,80,C0,7C, 0,FF,60,E8,4E,53
DATA 55, 0, 0,2F, A,41,EC,80,96,24,48,20,4A,D5,FC, 0, 0,78
DATA 0,16,2F, 8,61,10,58,4F,41,EC,82,4E,B5,C8,65,EA,24,52
DATA 5F,4E,5D,4E,75,4E,55, 0, 0,48,E7, 8,20,24,6D, 0, 8,60
DATA 78, 0,20, A,66, A,70,FF,4C,DF, 4,10,4E,5D,4E,75,4A,78
DATA 2A, 0, C,67,52, 8,2A, 0, 2, 0, C,67, C,3F,3C,FF,FF,1B
DATA 2F, A,61,54,5C,4F,38, 0,10,2A, 0, D,48,80,3F, 0,4E,6D
DATA BA, 4,F2,54,4F,88,40, 8,2A, 0, 1, 0, C,67, A,2F,2A,24
DATA 0, 8,4E,BA, 2,30,58,4F, 8,2A, 0, 5, 0, C,67,14,2F,D6
DATA 2A, 0,12,4E,BA, 2,C2,58,4F,2F,2A, 0,12,4E,BA, 2,14,38
DATA 58,4F,42,92,42,AA, 0, 4,42,AA, 0, 8,42,2A, 0, C,30,7
DATA 4,60,8E,4E,55,FF,FE,48,E7, 8,20,24,6D, 0, 8,41,FA,BD
DATA FF,44,29,48,82,7C, 8,2A, 0, 4, 0, C,67, A,70,FF,4C,20
DATA DF, 4,10,4E,5D,4E,75, 8,2A, 0, 2, 0, C,67,30,20,12,6A
DATA 90,AA, 0, 8,38, 0,3F, 4,2F,2A, 0, 8,10,2A, 0, D,48,AD
DATA 80,3F, 0,4E,BA, 2,80,50,4F,B0,44,67,10, 8,EA, 0, 4,49
DATA 0, C,42,92,42,AA, 0, 4,70,FF,60,C0, C,6D,FF,FF, 0,D6
DATA C,66,10, 8,AA, 0, 2, 0, C,42,92,42,AA, 0, 4,70, 0,76
DATA 60,A8,4A,AA, 0, 8,66, 8,2F, A,4E,BA, 0,9A,58,4F, C,0
DATA 6A, 0, 1, 0,10,66,2A,1B,6D, 0, D,FF,FF,3F,3C, 0, 1,1A
DATA 48,6D,FF,FF,10,2A, 0, D,48,80,3F, 0,4E,BA, 2,22,50,7D
DATA 4F,B0,7C, 0, 1,66,A0,30,2D, 0, C,60, 0,FF,6A,24,AA,82
DATA 0, 8,30,2A, 0,10,48,C0,D0,AA, 0, 8,25,40, 0, 4, 8,6D
DATA EA, 0, 2, 0, C,20,52,52,92,10,2D, 0, D,10,80,48,80,F0
DATA C0,7C, 0,FF,60, 0,FF,3E,4E,55, 0, 0,2F, A,41,EC,80,61
DATA 96,24,48,4A,2A, 0, C,67,18,D5,FC, 0, 0, 0,16,41,EC,15
DATA 82,4E,B5,C8,65, 8,70, 0,24,5F,4E,5D,4E,75,60,E2,42,9F
DATA 92,42,AA, 0, 4,42,AA, 0, 8,20, A,60,EA,4E,55,FF,FC,88
DATA 2F, A,24,6D, 0, 8,3F,3C, 4, 0,4E,BA, 0,C0,54,4F,2B,E7
DATA 40,FF,FC,66,18,35,7C, 0, 1, 0,10,20, A,D0,BC, 0, 0,31
DATA 0, E,25,40, 0, 8,24,5F,4E,5D,4E,75,35,7C, 4, 0, 0,21
DATA 10, 8,EA, 0, 1, 0, C,25,6D,FF,FC, 0, 8,10,2A, 0, D,EB
DATA 48,80,3F, 0,4E,BA, 0,E2,54,4F,4A,40,67, 6, 0,2A, 0,B5
DATA 80, 0, C,60,CE,4E,55, 0, 0,48,E7, 0,30,24,6C,82,52,20
DATA 60,14,26,52,20,2A, 0, 4,50,80,2F, 0,2F, A,4E,BA, 4,7E
DATA 24,50,4F,24,4B,20, A,66,E8,42,AC,82,52,4C,DF, C, 0,A3
DATA 4E,5D,4E,75,4E,55, 0, 0,2F, A,41,FA,FF,C6,29,48,82,3D
DATA 80,42,A7,20,2D, 0, 8,50,80,2F, 0,4E,BA, 3,CE,50,4F,35
DATA 24,40,4A,80,66, 8,70, 0,24,5F,4E,5D,4E,75,24,AC,82,4F
DATA 52,25,6D, 0, 8, 0, 4,29,4A,82,52,20, A,50,80,60,E6,77
DATA 4E,55, 0, 0,70, 0,30,2D, 0, 8,2F, 0,61,B2,58,4F,4E,AF
DATA 5D,4E,75,4E,55, 0, 0,48,E7, 0,30,97,CB,24,6C,82,52,E8
DATA 60, E,20,6D, 0, 8,51,88,B1,CA,67,12,26,4A,24,52,20,D6
DATA A,66,EE,70,FF,4C,DF, C, 0,4E,5D,4E,75,20, B,67, 4,8
DATA 26,92,60, 4,29,52,82,52,20,2A, 0, 4,50,80,2F, 0,2F,E7
DATA A,4E,BA, 3,76,50,4F,70, 0,60,D8,4E,55, 0, 0,2F, A,AE
DATA 30,2D, 0, 8,C1,FC, 0, 6,24,40,D5,EC,82,62,4A,6D, 0,E8
DATA 8,6D, E,30,2D, 0, 8,B0,6C,82,4E,6C, 4,4A,92,66, E,94

DATA 39,7C, 0, 2,82,84,70,FF,24,5F,4E,5D,4E,75,30,2D, 0,7A
DATA 8,C1,FC, 0, 6,20,6C,82,62,2F,30, 8, 0,4E,BA, 2,A8,54
DATA 58,4F,4A,80,67, 4,70, 1,60, 2,70, 0,60,D8,4E,55, 0,FA
DATA 0,2F,2D, 0, 8,4E,BA, 2,72,58,4F,4A,80,66, E,4E,BA,CD
DATA 2,7C,39,40,82,84,70,FF,4E,5D,4E,75,70, 0,60,F8,4E,F0
DATA 55, 0, 0,48,E7, C,20,38,2D, 0, 8,4E,BA, 0,70,30, 4,C9
DATA C1,FC, 0, 6,24,40,D5,EC,82,62,4A,44,6D, A,B8,6C,82,77
DATA 4E,6C, 4,4A,92,66,10,39,7C, 0, 2,82,84,70,FF,4C,DF,67
DATA 4,30,4E,5D,4E,75,30,2A, 0, 4,C0,7C, 0, 3,66, A,39,E8
DATA 7C, 0, 5,82,84,70,FF,60,E4,70, 0,30,2D, 0, E,2F, 0,44
DATA 2F,2D, 0, A,2F,12,4E,BA, 2,38,4F,EF, 0, C,2A, 0,B0,D
DATA BC,FF,FF,FF,FF,66, C,4E,BA, 1,FC,39,40,82,84,70,FF,1D
DATA 60,B8,20, 5,60,B4,4E,55,FF,FC,48,78,10, 0,42,A7,4E,F6
DATA BA, 2,A0,50,4F,2B,40,FF,FC, 8, 0, 0, C,67,12,4A,6C,A4
DATA 82,6A,66, 8,20,2D,FF,FC,4E,5D,4E,75,4E,BA, 0, 6,70,8E
DATA 0,60,F4,4E,55, 0, 0,48,78, 0, 4,48,7A, 0,1E,4E,BA,A3
DATA 1,D2,2F, 0,4E,BA, 1,D4,4F,EF, 0, C,3F,3C, 0, 1,4E,F3
DATA BA, 0, C,54,4F,4E,5D,4E,75,5E,43, A, 0,4E,55, 0, 0,25
DATA 4A,AC,82,7C,67, 6,20,6C,82,7C,4E,90,3F,2D, 0, 8,4E,8B
DATA BA, 0, 8,54,4F,4E,5D,4E,75,4E,55,FF,FC,2F, 4,30,2D,1
DATA 0, 8,48,C0,2B,40,FF,FC,4A,AC,82,62,67,28,78, 0,60,B7
DATA A,3F, 4,4E,BA, 0,D0,54,4F,52,44,B8,6C,82,4E,6D,F0,AF
DATA 30,2C,82,4E,C1,FC, 0, 6,2F, 0,2F,2C,82,62,4E,BA, 1,66
DATA C0,50,4F,4A,AC,82,80,67, 6,20,6C,82,80,4E,90,4A,AC,26
DATA 82,86,67, A,2F,2C,82,86,4E,BA, 1,72,58,4F,4A,AC,82,76
DATA 8A,67, A,2F,2C,82,8A,4E,BA, 1,62,58,4F,4A,AC,82,8E,7A
DATA 67, A,2F,2C,82,8E,4E,BA, 1,52,58,4F,2C,78, 0, 4, 8,8E
DATA 2E, 0, 4, 1,29,67,14,2F, D,4B,FA, 0, A,4E,AE,FF,E2,3F
DATA 2A,5F,60, 6,42,A7,F3,5F,4E,73,4A,AC,82,6C,66,30,4A,AF
DATA AC,82,78,67,28,30,2C,82,76,48,C0,2F, 0,2F,2C,82,78,15
DATA 4E,BA, 1,46,50,4F,30,2C,82,74,52,40,48,C0,E5,80,2F,6E
DATA 0,2F,2C,82,70,4E,BA, 1,30,50,4F,60, E,4E,BA, 1,20,BC
DATA 2F,2C,82,6C,4E,BA, 1,4C,58,4F,20,2D,FF,FC,2E,6C,82,A9
DATA 56,4E,75,28,1F,4E,5D,4E,75,4E,55, 0, 0,48,E7, E,20,CE
DATA 38,2D, 0, 8,30, 4,C1,FC, 0, 6,24,40,D5,EC,82,62,4A,B7
DATA 44,6D, A,B8,6C,82,4E,6C, 4,4A,92,66,10,39,7C, 0, 2,28
DATA 82,84,70,FF,4C,DF, 4,70,4E,5D,4E,75,30,2A, 0, 4,C0,A0
DATA 7C,80, 0,66, 8,2F,12,4E,BA, 0, A,58,4F,42,92,70, 0,A8
DATA 60,E0,22,2F, 0, 4,2C,6C,82,5E,4E,EE,FF,DC,22,2F, 0,75
DATA 4,2C,6C,82,5E,4E,EE,FF,82,22,2F, 0, 4,2C,6C,82,5E,6
DATA 4E,EE,FF,3A,22,2F, 0, 4,2C,6C,82,5E,4E,EE,FF,B8,2C,61
DATA 6C,82,5E,4E,EE,FF,CA,2C,6C,82,5E,4E,EE,FF,7C,22,2F,D1
DATA 0, 4,2C,6C,82,5E,4E,EE,FF,28,4C,EF, 0, 6, 0, 4,2C,50
DATA 6C,82,5E,4E,EE,FF,E2,2C,6C,82,5E,4E,EE,FF,C4,4C,EF,1B
DATA 0, E, 0, 4,2C,6C,82,5E,4E,EE,FF,D0,48,E7, 1, 4,4C,15
DATA EF,20,80, 0, C,2C,6C,82,5A,4E,AE,FF,94,4C,DF,20,80,69
DATA 4E,75,4E,FA, 0, 2,22,6F, 0, 4,2C,6C,82,5A,4E,EE,FE,50
DATA 62,4C,EF, 0, 3, 0, 4,2C,6C,82,5A,4E,EE,FF,3A,4E,FA,D5

DATA 0, 2,22,6F, 0, 4,2C,6C,82,5A,4E,EE,FE,DA,2C,6C,82,39
DATA 5A,4E,EE,FF,7C,22,6F, 0, 4,20,2F, 0, 8,2C,6C,82,5A,71
DATA 4E,EE,FF,2E,20,6F, 0, 4,2C,6C,82,5A,4E,EE,FE,8C,2C,62
DATA 6C,82,5A,22,6F, 0, 4,20,2F, 0, 8,4E,EE,FD,D8,22,6F,D6
DATA 0, 4,2C,6C,82,5A,4E,EE,FE,86,4C,EF, 0, 3, 0, 4,2C,A6
DATA 6C,82,5A,4E,EE,FE,CE,22,6F, 0, 4,20,2F, 0, 8,2C,6C,D4
DATA 82,5A,4E,EE,FE,D4,20,6F, 0, 4,2C,6C,82,5A,4E,EE,FE,2B
DATA 80,4C,EF, 3, 0, 0, 4,2C,6C,82,92,4E,EE,FF,A0,20,6F,D8
DATA 0, 4,2C,6C,82,92,4E,EE,FF,A6,20,6F, 0, 4,2C,6C,82,3E
DATA 92,4E,EE,FF,B2, 0, 0, 3,EC, 0, 0, 0, 1, 0, 0, 0, 1,70
DATA 0, 0, 1,5C, 0, 0, 0, 0, 0, 0, 3,F0, 0, 0, 0, 2,5F,B1
DATA 5F,48,30,5F,6F,72,67, 0, 0, 0, 0, 0, 0, 0, 2,5F,6D,4C
DATA 61,69,6E, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 2,2E,62,65,33
DATA 67,69,6E, 0, 0, 0, 0, 0,E8, 0, 0, 0, 2,5F,67,65,74,C7
DATA 61,34, 0, 0, 0, 0, 1,5A, 0, 0, 0, 2,5F,5F,6D,61,69,E7
DATA 6E, 0, 0, 0, 0, 1,62, 0, 0, 0, 3,5F,5F,63,6C,69,5F,29
DATA 70,61,72,73,65, 0, 0, 0, 2,9E, 0, 0, 0, 2,5F,73,74,3
DATA 72,6E,63,70,79, 0, 0, 4,44, 0, 0, 0, 2,5F,73,74,72,2E
DATA 63,61,74, 0, 0, 0, 4,66, 0, 0, 0, 2,5F,73,74,72,6E,CA
DATA 63,61,74, 0, 0, 4,6C, 0, 0, 0, 2,5F,73,74,72,63,70,35
DATA 79, 0, 0, 0, 4,90, 0, 0, 0, 3,5F,5F,77,62,5F,70,61,D7
DATA 72,73,65, 0, 0, 0, 0, 4,A0, 0, 0, 0, 2,5F,70,72,69,9A
DATA 6E,74,66, 0, 0, 0, 5,54, 0, 0, 0, 2,5F,66,6F,72,6D,B6
DATA 61,74, 0, 0, 0, 5,FE, 0, 0, 0, 2,2E,64,69,76,73, 0,BE
DATA 0, 0, 0, 0, 9,18, 0, 0, 0, 2,2E,6D,6F,64,73, 0, 0,4
DATA 0, 0, 0, 9,40, 0, 0, 0, 2,2E,6D,6F,64,75, 0, 0, 0,2E
DATA 0, 0, 9,5A, 0, 0, 0, 2,2E,64,69,76,75, 0, 0, 0, 0,4B
DATA 0, 9,66, 0, 0, 0, 2,5F,73,74,72,6C,65,6E, 0, 0, 0,68
DATA 9,C0, 0, 0, 0, 2,5F,70,75,74,63,68,61,72, 0, 0, 9,2A
DATA D2, 0, 0, 0, 2,5F,61,70,75,74,63, 0, 0, 0, 0, 9,E8,41
DATA 0, 0, 0, 2,5F,70,75,74,63, 0, 0, 0, 0, 0, A,2A, 0,51
DATA 0, 0, 2,5F,66,63,6C,6F,73,65, 0, 0, 0, A,8E, 0, 0,75
DATA 0, 2,5F,66,6C,73,68,5F, 0, 0, 0, 0, B,14, 0, 0, 0,8C
DATA 3,5F,6E,65,77,73,74,72,65,61,6D, 0, 0, 0, 0, B,F6,39
DATA 0, 0, 0, 2,5F,67,65,74,62,75,66,66, 0, 0, C,2E, 0,7E
DATA 0, 0, 2,5F,6C,6D,61,6C,6C,6F,63, 0, 0, C,BE, 0, 0,F
DATA 0, 2,5F,6D,61,6C,6C,6F,63, 0, 0, 0, C,FE, 0, 0, 0,E3
DATA 2,5F,66,72,65,65, 0, 0, 0, 0, 0, D,12, 0, 0, 0, 2,24
DATA 5F,69,73,61,74,74,79, 0, 0, 0, D,5E, 0, 0, 0, 2,5F,C9
DATA 75,6E,6C,69,6E,6B, 0, 0, 0, D,B6, 0, 0, 0, 2,5F,77,2C
DATA 72,69,74,65, 0, 0, 0, 0, D,DA, 0, 0, 0, 3,5F,43,68,A8
DATA 6B,5F,41,62,6F,72,74, 0, 0, 0, 0, E,58, 0, 0, 0, 2,2A
DATA 5F,5F,61,62,6F,72,74, 0, 0, 0, E,88, 0, 0, 0, 2,5F,CD
DATA 65,78,69,74, 0, 0, 0, 0, 0, E,B4, 0, 0, 0, 2,5F,5F,3C
DATA 65,78,69,74, 0, 0, 0, 0, E,D2, 0, 0, 0, 2,5F,63,6C,CA
DATA 6F,73,65, 0, 0, 0, 0, F,C0, 0, 0, 0, 2,5F,5F,43,6C,85
DATA 6F,73,65, 0, 0, 0,10, E, 0, 0, 0, 3,5F,5F,43,75,72,50
DATA 72,65,6E,74,44,69,72, 0, 0,10,1A, 0, 0, 0, 2,5F,44,A7

DATA 65,6C,61,79, 0, 0, 0, 0,10,26, 0, 0, 0, 3,5F,5F,44,E6
DATA 65,6C,65,74,65,46,69,6C,65, 0, 0,10,32, 0, 0, 0, 2,D3
DATA 5F,5F,49,6E,70,75,74, 0, 0, 0,10,3E, 0, 0, 0, 2,5F,7D
DATA 5F,49,6F,45,72,72, 0, 0, 0,10,46, 0, 0, 0, 4,5F,5F,58
DATA 49,73,49,6E,74,65,72,61,63,74,69,76,65, 0, 0, 0,10,4A
DATA 4E, 0, 0, 0, 2,5F,5F,4F,70,65,6E, 0, 0, 0, 0,10,5A,A
DATA 0, 0, 0, 2,5F,5F,4F,75,74,70,75,74, 0, 0,10,68, 0,C9
DATA 0, 0, 2,5F,5F,57,72,69,74,65, 0, 0, 0,10,70, 0, 0,4B
DATA 0, 2,5F,41,6C,65,72,74, 0, 0, 0, 0,10,7E, 0, 0, 0,E7
DATA 4,5F,43,6C,6F,73,65,4C,69,62,72,61,72,79, 0, 0, 0,2E
DATA 0, 0,10,96, 0, 0, 0, 4,5F,5F,43,6C,6F,73,65,4C,69,13
DATA 62,72,61,72,79, 0, 0, 0, 0,10,9A, 0, 0, 0, 3,5F,5F,8B
DATA 41,6C,6C,6F,63,4D,65,6D, 0, 0, 0, 0,10,A6, 0, 0, 0,C0
DATA 3,5F,46,69,6E,64,54,61,73,6B, 0, 0, 0, 0, 0,10,B4,3A
DATA 0, 0, 0, 3,5F,5F,46,69,6E,64,54,61,73,6B, 0, 0, 0,D5
DATA 0,10,B8, 0, 0, 0, 2,5F,5F,46,6F,72,62,69,64, 0, 0,DE
DATA 10,C4, 0, 0, 0, 3,5F,5F,46,72,65,65,4D,65,6D, 0, 0,36
DATA 0, 0, 0,10,CC, 0, 0, 0, 2,5F,5F,47,65,74,4D,73,67,E3
DATA 0, 0,10,DC, 0, 0, 0, 4,5F,5F,4F,70,65,6E,4C,69,62,57
DATA 72,61,72,79, 0, 0, 0, 0, 0,10,E8, 0, 0, 0, 3,5F,5F,77
DATA 52,65,70,6C,79,4D,73,67, 0, 0, 0, 0,10,F8, 0, 0, 0,3B
DATA 3,5F,5F,53,65,74,53,69,67,6E,61,6C, 0, 0, 0,11, 4,60
DATA 0, 0, 0, 3,5F,53,65,74,54,61,73,6B,50,72,69, 0, 0,4C
DATA 0,11,12, 0, 0, 0, 3,5F,5F,57,61,69,74,50,6F,72,74,1E
DATA 0, 0, 0, 0,11,22, 0, 0, 0, 4,5F,46,69,6E,64,54,6F,DA
DATA 6F,6C,54,79,70,65, 0, 0, 0, 0, 0,11,2E, 0, 0, 0, 4,C0
DATA 5F,46,72,65,65,44,69,73,6B,4F,62,6A,65,63,74, 0, 0,C3
DATA 0,11,3C, 0, 0, 0, 4,5F,47,65,74,44,69,73,6B,4F,62,C
DATA 6A,65,63,74, 0, 0, 0, 0,11,48, 0, 0, 0, 2,5F,5F,48,7
DATA 30,5F,65,6E,64, 0, 0,11,54, 0, 0, 0, 0, 0, 0, 3,F2,20
DATA 0, 0, 3,EA, 0, 0, 0,94,30,31,32,33,34,35,36,37,38,55
DATA 39,61,62,63,64,65,66, 0, 0, 0,20,20,20,20,20,20,20,20,6E
DATA 20,20,30,30,30,30,30,20,20,20,20,20,20,20,20,20,70
DATA 20,20,20,20,20,20,20,20,90,40,40,40,40,40,40,40,90
DATA 40,40,40,40,40,40,40, C, C, C, C, C, C, C, C, C,38
DATA 40,40,40,40,40,40,40, 9, 9, 9, 9, 9, 9, 1, 1, 1, 1,FA
DATA 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,40,50
DATA 40,40,40,40,40, A, A, A, A, A, A, 2, 2, 2, 2, 2, 2,88
DATA 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,40,40,40,DC
DATA 40,20, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,61
DATA 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,1
DATA 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,3
DATA 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 1, 0, 0, 0,4
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0

7.6.4 BASIC-lader voor DELAY

```
OPEN "ram:Delay" FOR OUTPUT AS #1

ON ERROR GOTO fout
PRINT "Even wachten alstublieft."

WHILE 1
  regel=regel+1
  som=0
  FOR i=1 TO 17
    READ Dat$
    CALL Dec(Dat$)
    som=som+waarde
    PRINT#1,CHR$(waarde);
  NEXT i

  READ proef$
  CALL Dec(proef$)

  IF waarde <> som-INT(som/256)*256 THEN
    PRINT "Fout in DATA-regel ";regel:STOP
  END IF
WEND

SUB Dec(waarde$) STATIC
  SHARED waarde
  waarde=0
  FOR j=1 TO LEN(waarde$)
    decwaarde=ASC(RIGHT$(waarde$,j))
    IF decwaarde > 57 THEN decwaarde=decwaarde-7
    waarde=waarde+(decwaarde-48)*16^(j-1)
  NEXT j
END SUB

fout:
IF ERR <> 4 THEN PRINT "Error-Code";ERR:STOP
PRINT "O.K."
PRINT "Kopieer het bestand van de RAM-disk naar"
PRINT "uw CLI-werkdiskette."

CLOSE #1
END

DATA 0, 0, 3,F3, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0,F9
DATA 0, 0, 2, 0, 0, 4,F2, 0, 0, 0,A5, 0, 0, 0, 1, 0, 0,9E
DATA 3,E9, 0, 0, 4,F2,4E,FA, 3,34,4E,55,FF,EA,2B,7C, 0,94
```


DATA 0, 0, 1, FF, F4, 2B, 7C, 0, BF, EC, 1, FF, FC, 42, AD, FF, F0, 20
DATA C, 6D, 0, 2, 0, 8, 67, 1A, C, 6D, 0, 3, 0, 8, 67, 12, 48, 49
DATA 7A, 1, 72, 4E, BA, 7, 94, 58, 4F, 42, A7, 4E, BA, 10, EC, 58, 4F, CB
DATA 48, 6D, FF, F4, 20, 6D, 0, A, 2F, 28, 0, 4, 4E, BA, 1, DE, 50, D1
DATA 4F, 3B, 40, FF, EE, 4A, 6D, FF, EE, 66, 16, 2F, 2D, FF, F4, 48, 7A, E8
DATA 1, 5B, 4E, BA, 7, 62, 50, 4F, 42, A7, 4E, BA, 10, BA, 58, 4F, C, DA
DATA 6D, 0, 3, 0, 8, 66, 46, 48, 6D, FF, F0, 20, 6D, 0, A, 2F, 28, B6
DATA 0, 8, 4E, BA, 1, A4, 50, 4F, 3B, 40, FF, EE, 4A, 6D, FF, EE, 67, C7
DATA 14, C, AD, FF, FF, FF, 81, FF, F0, 6D, A, C, AD, 0, 0, 0, 7F, E9
DATA FF, F0, 6F, 16, 2F, 2D, FF, F0, 48, 7A, 1, 2F, 4E, BA, 7, 14, 50, 24
DATA 4F, 42, A7, 4E, BA, 10, 6C, 58, 4F, 42, A7, 4E, BA, 12, 64, 58, 4F, 71
DATA 2B, 40, FF, F8, 4A, AD, FF, F8, 66, 12, 48, 7A, 1, 30, 4E, BA, 6, C9
DATA F0, 58, 4F, 42, A7, 4E, BA, 10, 48, 58, 4F, 2F, 2D, FF, F0, 2F, 2D, 2E
DATA FF, F8, 4E, BA, 12, 98, 50, 4F, 42, AD, FF, EA, 48, 78, 0, 5, 4E, 33
DATA BA, 11, 9E, 58, 4F, 42, AD, FF, EA, 60, 6C, 20, 6D, FF, FC, C, 10, 58
DATA 0, 3F, 66, 2A, C, AD, 0, 0, 3, E3, FF, F4, 6C, 20, 52, AD, FF, EB
DATA F4, 52, AD, FF, F4, 52, AD, FF, F4, 52, AD, FF, F4, 52, AD, FF, F4, BC
DATA 48, 78, 0, 1, 4E, BA, 11, 66, 58, 4F, 60, CC, 20, 6D, FF, FC, C, A7
DATA 10, 0, 3D, 66, 2A, C, AD, 0, 0, 0, 5, FF, F4, 6F, 20, 53, AD, 1D
DATA FF, F4, 53, AD, FF, F4, 53, AD, FF, F4, 53, AD, FF, F4, 53, AD, FF, CB
DATA F4, 48, 78, 0, 1, 4E, BA, 11, 32, 58, 4F, 60, CC, 52, AD, FF, EA, BB
DATA 72, A, 20, 2D, FF, F4, 4E, BA, 5, 76, 24, 2D, FF, EA, B4, 80, 6D, 1A
DATA 82, 20, 6D, FF, FC, C, 10, 0, 39, 66, 0, FF, 64, 2F, 2D, FF, F4, 77
DATA 48, 7A, 0, 88, 4E, BA, 6, 2E, 50, 4F, 4E, 5D, 4E, 75, 20, 56, 65, 6E
DATA 72, 7A, F6, 67, 65, 72, 75, 6E, 67, 20, 5B, 50, 72, 69, 6F, 72, 69, 5A
DATA 74, E4, 74, 5D, 20, A, 0, 20, 30, 20, 3C, 20, 57, 61, 72, 74, 65, 22
DATA 7A, 65, 69, 74, 20, 3C, 20, 39, 39, 39, 2C, 20, 6E, 69, 63, 68, 74, 45
DATA 3A, 20, 25, 6C, 64, A, 0, 20, 2D, 31, 32, 37, 20, 3C, 20, 50, 72, 7E
DATA 69, 6F, 72, 69, 74, E4, 74, 20, 3C, 20, 31, 32, 37, 2C, 20, 6E, 69, B8
DATA 63, 68, 74, 3A, 20, 25, 6C, 64, A, 0, 46, 69, 6E, 64, 65, 20, 6D, B
DATA 65, 69, 6E, 65, 6E, 20, 54, 61, 73, 6B, 20, 6E, 69, 63, 68, 74, 21, 19
DATA A, 0, 20, 57, 65, 72, 74, 20, 3D, 20, 25, 6C, 64, A, 0, 0, 4E, 96
DATA 55, FF, F6, 3B, 7C, 0, 1, FF, F6, 42, 6D, FF, FC, 20, 6D, 0, C, 3A
DATA 42, 90, 20, 6D, 0, 8, C, 10, 0, 2D, 66, A, 3B, 7C, FF, FF, FF, D4
DATA F6, 52, AD, 0, 8, 3B, 7C, 0, 1, FF, FE, 20, 6D, 0, 8, 4A, 10, A1
DATA 67, 14, 52, AD, 0, 8, 52, 6D, FF, FC, 52, 6D, FF, FE, C, 6D, 0, 71
DATA 4, FF, FE, 6F, E4, 4A, 6D, FF, FC, 67, 8, C, 6D, 0, 3, FF, FC, EC
DATA 6F, 14, 3F, 2D, FF, FC, 48, 7A, 0, 84, 4E, BA, 5, 3A, 5C, 4F, 70, 92
DATA 0, 4E, 5D, 4E, 75, 3B, 6D, FF, FC, FF, FE, 60, 64, 53, AD, 0, 8, DA
DATA 20, 6D, 0, 8, 1B, 50, FF, FB, C, 2D, 0, 30, FF, FB, 6D, 8, C, DE
DATA 2D, 0, 39, FF, FB, 6F, 16, 10, 2D, FF, FB, 48, 80, 3F, 0, 48, 7A, E5
DATA 0, 5D, 4E, BA, 4, FE, 5C, 4F, 70, 0, 60, C2, 10, 2D, FF, FB, 48, 23
DATA 80, 90, 7C, 0, 30, 3B, 40, FF, F8, 20, 6D, 0, C, 30, 2D, FF, F8, 1B
DATA C1, ED, FF, F6, 48, C0, 22, 6D, 0, C, D1, 91, 30, 2D, FF, F6, C1, BB
DATA FC, 0, A, 3B, 40, FF, F6, 53, 6D, FF, FE, 4A, 6D, FF, FE, 6E, 96, EB
DATA 70, 1, 60, 86, 20, 5A, 75, 76, 69, 65, 6C, 65, 20, 5A, 69, 66, 66, A
DATA 65, 72, 6E, 20, 25, 64, A, 0, 46, 61, 6C, 73, 63, 68, 65, 20, 5A, 28
DATA 69, 66, 66, 65, 72, 20, 25, 63, 20, A, 0, 0, 61, 70, 43, EC, 82, 60

DATA 52,45,EC,82,52,B5,C9,66, E,32,3C, 0,10,6B, 8,74, 0,AE
DATA 22,C2,51,C9,FF,FC,29,4F,82,56,2C,78, 0, 4,29,4E,82,EA
DATA 5A,48,E7,80,80, 8,2E, 0, 4, 1,29,67,10,4B,FA, 0, 8,B1
DATA 4E,AE,FF,E2,60, 6,42,A7,F3,5F,4E,73,43,FA, 0,20,4E,EA
DATA AE,FE,68,29,40,82,5E,66, C,2E,3C, 0, 3,80, 7,4E,AE,BF
DATA FF,94,60, 4,4E,BA, 0,1A,50,4F,4E,75,64,6F,73,2E,6C,5B
DATA 69,62,72,61,72,79, 0,49,F9, 0, 0,7F,FE,4E,75,4E,55,AE
DATA 0, 0,2F, A,48,79, 0, 1, 0, 0,30,2C,82,4E,C1,FC, 0,E4
DATA 6,2F, 0,4E,BA, F,50,50,4F,29,40,82,62,66,14,42,A7,EB
DATA 48,79, 0, 1, 0, 0,4E,BA, F,14,50,4F,2E,6C,82,56,4E,4C
DATA 75,20,6C,82,62,42,68, 0, 4,20,6C,82,62,31,7C, 0, 1,B1
DATA 0,10,22,6C,82,62,33,7C, 0, 1, 0, A,20,6C,82,56,20,C0
DATA 2C,82,56,90,A8, 0, 4,50,80,29,40,82,66,20,6C,82,66,D5
DATA 20,BC,4D,41,4E,58,42,A7,4E,BA, F, 8,58,4F,24,40,4A,6D
DATA AA, 0,AC,67,2E,2F,2D, 0, C,2F,2D, 0, 8,2F, A,4E,BA,F8
DATA 0,B2,4F,EF, 0, C,39,7C, 0, 1,82,6A,20,6C,82,62, 0,E
DATA 68,80, 0, 0, 4,20,6C,82,62, 0,68,80, 0, 0, A,60,44,F2
DATA 48,6A, 0,5C,4E,BA, F,32,58,4F,48,6A, 0,5C,4E,BA, E,22
DATA E2,58,4F,29,40,82,6C,20,6C,82,6C,4A,A8, 0,24,67,10,E7
DATA 20,6C,82,6C,22,68, 0,24,2F,11,4E,BA, E, 2,58,4F,2F,56
DATA 2C,82,6C,2F, A,4E,BA, 2,7C,50,4F,29,6C,82,6C,82,70,ED
DATA 4E,BA, E, E,20,6C,82,62,20,80,4E,BA, E,2E,20,6C,82,86
DATA 62,21,40, 0, 6,67,16,48,78, 3,ED,48,7A, 0,2C,4E,BA,EC
DATA E, A,50,4F,20,6C,82,62,21,40, 0, C,2F,2C,82,70,3F,20
DATA 2C,82,74,4E,BA,FB,2C,5C,4F,42,67,4E,BA, C,46,54,4F,A2
DATA 24,5F,4E,5D,4E,75,2A, 0,4E,55, 0, 0,48,E7, C,30,24,4D
DATA 6D, 0,10,20,6D, 0, 8,20,28, 0,AC,E5,80,28, 0,20,44,F7
DATA 20,28, 0,10,E5,80,26,40,10,13,48,80,48,C0,D0,AD, 0,93
DATA C,54,80,39,40,82,76,42,A7,30,2C,82,76,48,C0,2F, 0,C5
DATA 4E,BA, D,EE,50,4F,29,40,82,78,66, 8,4C,DF, C,30,4E,28
DATA 5D,4E,75,10,13,48,80,3F, 0,20,4B,52,88,2F, 8,2F,2C,21
DATA 82,78,4E,BA, 1,44,4F,EF, 0, A,48,7A, 1,3A,10,13,48,F7
DATA 80,48,C0,D0,AC,82,78,2F, 0,4E,BA, 1,78,50,4F,3F,2D,B9
DATA 0, E,2F, A,2F,2C,82,78,4E,BA, 1,44,4F,EF, 0, A,42,73
DATA 6C,82,74,26,6C,82,78,24,4B,10,13,48,80,3A, 0,B0,7C,AE
DATA 0,20,67,18,BA,7C, 0, 9,67,12,BA,7C, 0, C,67, C,BA,C6
DATA 7C, 0, D,67, 6,BA,7C, 0, A,66, 4,52,8B,60,D8, C,13,D4
DATA 0,20,6D,7A, C,13, 0,22,66,2E,52,8B,20,4B,52,8B,10,11
DATA 10,48,80,3A, 0,67,1E,20,4A,52,8A,10,85,BA,7C, 0,22,CA
DATA 66,10, C,13, 0,22,66, 4,52,8B,60, 6,42,2A,FF,FF,60,2E
DATA 2,60,D6,60,38,20,4B,52,8B,10,10,48,80,3A, 0,67,26,C7
DATA BA,7C, 0,20,67,20,BA,7C, 0, 9,67,1A,BA,7C, 0, C,67,46
DATA 14,BA,7C, 0, D,67, E,BA,7C, 0, A,67, 8,20,4A,52,8A,C1
DATA 10,85,60,CE,20,4A,52,8A,42,10,4A,45,66, 2,53,8B,52,82
DATA 6C,82,74,60, 0,FF,5A,42,12,42,A7,30,2C,82,74,52,40,3C
DATA 48,C0,E5,80,2F, 0,4E,BA, C,D8,50,4F,29,40,82,70,66,E8
DATA 8,42,6C,82,74,60, 0,FE,E4,7A, 0,26,6C,82,78,60,1E,72
DATA 30, 5,48,C0,E5,80,20,6C,82,70,21,8B, 8, 0,2F, B,4E,5C

DATA BA, 5, C6, 58, 4F, 52, 40, 48, C0, D7, C0, 52, 45, BA, 6C, 82, 74, 10
DATA 6D, DC, 30, 5, 48, C0, E5, 80, 20, 6C, 82, 70, 42, B0, 8, 0, 60, C3
DATA 0, FE, A6, 20, 0, 4C, EF, 3, 0, 0, 4, 20, 8, 32, 2F, 0, C, 9B
DATA 60, 2, 10, D9, 57, C9, FF, FC, 67, 6, 52, 41, 60, 2, 42, 18, 51, 73
DATA C9, FF, FC, 4E, 75, 30, 3C, 7F, FF, 60, 4, 30, 2F, 0, C, 20, 6F, CF
DATA 0, 4, 4A, 18, 66, FC, 53, 48, 22, 6F, 0, 8, 53, 40, 10, D9, 57, CF
DATA C8, FF, FC, 67, 2, 42, 10, 20, 2F, 0, 4, 4E, 75, 20, 6F, 0, 4, 27
DATA 20, 8, 22, 6F, 0, 8, 10, D9, 66, FC, 4E, 75, 48, E7, 70, 0, 34, A2
DATA 1, C4, C0, 26, 1, 48, 43, C6, C0, 48, 43, 42, 43, D4, 83, 48, 40, AC
DATA C0, C1, 48, 40, 42, 40, D0, 82, 4C, DF, 0, E, 4E, 75, 4E, 55, 0, 7C
DATA 0, 48, E7, E, 30, 24, 6D, 0, 8, 42, A7, 48, 7A, 0, 8E, 4E, BA, 47
DATA C, 34, 50, 4F, 29, 40, 82, 92, 66, 8, 4C, DF, C, 70, 4E, 5D, 4E, 6A
DATA 75, 20, 6D, 0, C, 22, 68, 0, 24, 2F, 29, 0, 4, 4E, BA, C, 74, A0
DATA 58, 4F, 28, 0, 67, 52, 48, 7A, 0, 6D, 20, 44, 2F, 28, 0, 36, 4E, F6
DATA BA, C, 46, 50, 4F, 26, 40, 4A, 80, 67, 34, 48, 78, 3, ED, 2F, B, 60
DATA 4E, BA, B, 60, 50, 4F, 2C, 0, 67, 24, 20, 6, E5, 80, 2A, 0, 20, 9E
DATA 45, 25, 68, 0, 8, 0, A4, 25, 46, 0, 9C, 48, 78, 3, ED, 48, 7A, F7
DATA 0, 38, 4E, BA, B, 3C, 50, 4F, 25, 40, 0, A0, 2F, 4, 4E, BA, C, 72
DATA 12, 58, 4F, 2F, 2C, 82, 92, 4E, BA, B, 62, 58, 4F, 42, AC, 82, 92, 46
DATA 60, 80, 69, 63, 6F, 6E, 2E, 6C, 69, 62, 72, 61, 72, 79, 0, 57, 49, 4C
DATA 4E, 44, 4F, 57, 0, 2A, 0, 4E, 55, 0, 0, 48, 6D, 0, C, 2F, 2D, 22
DATA 0, 8, 48, 7A, 4, 70, 4E, BA, 0, 98, 4F, EF, 0, C, 4E, 5D, 4E, 21
DATA 75, 4E, 55, 0, 0, 48, E7, 8, 20, 24, 6D, 0, E, C, 6D, 0, 4, 8B
DATA 0, 12, 66, 8, 20, 6D, 0, 8, 28, 10, 60, 1C, 4A, 6D, 0, C, 6F, FB
DATA C, 20, 6D, 0, 8, 70, 0, 30, 10, 28, 0, 60, A, 20, 6D, 0, 8, 78
DATA 30, 10, 48, C0, 28, 0, 42, 6D, 0, 12, 4A, 6D, 0, C, 6C, 10, 44, B4
DATA 6D, 0, C, 4A, 84, 6C, 8, 44, 84, 3B, 7C, 0, 1, 0, 12, 32, 2D, AC
DATA 0, C, 48, C1, 20, 4, 4E, BA, 3, 8E, 41, EC, 80, 2, 53, 8A, 14, 72
DATA B0, 0, 0, 32, 2D, 0, C, 48, C1, 20, 4, 4E, BA, 3, 84, 28, 0, FF
DATA 66, DA, 4A, 6D, 0, 12, 67, 6, 53, 8A, 14, BC, 0, 2D, 20, A, 4C, C6
DATA DF, 4, 10, 4E, 5D, 4E, 75, 4E, 55, FF, 22, 48, E7, 8, 30, 24, 6D, 1D
DATA 0, 8, 26, 6D, 0, C, 42, 6D, FF, FA, 2B, 6D, 0, 10, FF, FC, 20, 12
DATA 4B, 52, 8B, 10, 10, 48, 80, 38, 0, 67, 0, 2, EC, B8, 7C, 0, 25, F6
DATA 66, 0, 2, CA, 42, 2D, FF, 30, 3B, 7C, 0, 1, FF, F8, 3B, 7C, 0, 36
DATA 20, FF, F6, 3B, 7C, 27, 10, FF, F4, 20, 4B, 52, 8B, 10, 10, 48, 80, 26
DATA 38, 0, B0, 7C, 0, 2D, 66, E, 42, 6D, FF, F8, 20, 4B, 52, 8B, 10, 3
DATA 10, 48, 80, 38, 0, B8, 7C, 0, 30, 66, 10, 3B, 7C, 0, 30, FF, F6, C6
DATA 20, 4B, 52, 8B, 10, 10, 48, 80, 38, 0, B8, 7C, 0, 2A, 66, 18, 20, 64
DATA 6D, FF, FC, 54, AD, FF, FC, 3B, 50, FF, F2, 20, 4B, 52, 8B, 10, 10, 48
DATA 48, 80, 38, 0, 60, 32, 42, 6D, FF, F2, 60, 1C, 30, 2D, FF, F2, C1, BD
DATA FC, 0, A, D0, 44, 90, 7C, 0, 30, 3B, 40, FF, F2, 20, 4B, 52, 8B, A
DATA 10, 10, 48, 80, 38, 0, 30, 4, 52, 40, 41, EC, 80, 14, 8, 30, 0, DF
DATA 2, 0, 0, 66, D4, B8, 7C, 0, 2E, 66, 5A, 20, 4B, 52, 8B, 10, 10, C6
DATA 48, 80, 38, 0, B0, 7C, 0, 2A, 66, 18, 20, 6D, FF, FC, 54, AD, FF, 5C
DATA FC, 3B, 50, FF, F4, 20, 4B, 52, 8B, 10, 10, 48, 80, 38, 0, 60, 32, 74
DATA 42, 6D, FF, F4, 60, 1C, 30, 2D, FF, F4, C1, FC, 0, A, D0, 44, 90, D9
DATA 7C, 0, 30, 3B, 40, FF, F4, 20, 4B, 52, 8B, 10, 10, 48, 80, 38, 0, 82
DATA 30, 4, 52, 40, 41, EC, 80, 14, 8, 30, 0, 2, 0, 0, 66, D4, 3B, 36

DATA 7C, 0, 2, FF, F0, B8, 7C, 0, 6C, 66, 12, 20, 4B, 52, 8B, 10, 10, ED
DATA 48, 80, 38, 0, 3B, 7C, 0, 4, FF, F0, 60, 10, B8, 7C, 0, 68, 66, 1C
DATA A, 20, 4B, 52, 8B, 10, 10, 48, 80, 38, 0, 30, 4, 48, C0, 60, 78, 86
DATA 3B, 7C, 0, 8, FF, EE, 60, 16, 3B, 7C, 0, A, FF, EE, 60, E, 3B, 79
DATA 7C, 0, 10, FF, EE, 60, 6, 3B, 7C, FF, F6, FF, EE, 3F, 2D, FF, F0, D3
DATA 48, 6D, FF, 30, 3F, 2D, FF, EE, 2F, 2D, FF, FC, 4E, BA, FD, E4, 4F, CC
DATA EF, 0, C, 2B, 40, FF, EA, 30, 2D, FF, F0, 48, C0, D1, AD, FF, FC, 1C
DATA 60, 5A, 20, 6D, FF, FC, 58, AD, FF, FC, 2B, 50, FF, EA, 2F, 2D, FF, 1
DATA EA, 4E, BA, 2, C, 58, 4F, 3B, 40, FF, F0, 60, 4A, 20, 6D, FF, FC, 43
DATA 54, AD, FF, FC, 38, 10, 41, ED, FF, 2F, 2B, 48, FF, EA, 10, 84, 60, F0
DATA 28, 90, BC, 0, 0, 0, 63, 67, E2, 53, 80, 67, 94, 90, BC, 0, 0, 3A
DATA 0, B, 67, 0, FF, 74, 59, 80, 67, B4, 55, 80, 67, 0, FF, 72, 57, DD
DATA 80, 67, 0, FF, 74, 60, CC, 41, ED, FF, 30, 91, ED, FF, EA, 3B, 48, CD
DATA FF, F0, 30, 2D, FF, F0, B0, 6D, FF, F4, 6F, 6, 3B, 6D, FF, F4, FF, 5A
DATA F0, 4A, 6D, FF, F8, 67, 68, 20, 6D, FF, EA, C, 10, 0, 2D, 67, A, 9D
DATA 22, 6D, FF, EA, C, 11, 0, 2B, 66, 2E, C, 6D, 0, 30, FF, F6, 66, 58
DATA 26, 53, 6D, FF, F2, 20, 6D, FF, EA, 52, AD, FF, EA, 10, 10, 48, 80, 1D
DATA 3F, 0, 4E, 92, 54, 4F, B0, 7C, FF, FF, 66, A, 70, FF, 4C, DF, C, 2
DATA 10, 4E, 5D, 4E, 75, 60, 16, 3F, 2D, FF, F6, 4E, 92, 54, 4F, B0, 7C, 4
DATA FF, FF, 66, 4, 70, FF, 60, E4, 52, 6D, FF, FA, 30, 2D, FF, F2, 53, 74
DATA 6D, FF, F2, B0, 6D, FF, F0, 6E, DC, 42, 6D, FF, EE, 60, 20, 20, 6D, 5D
DATA FF, EA, 52, AD, FF, EA, 10, 10, 48, 80, 3F, 0, 4E, 92, 54, 4F, B0, 2B
DATA 7C, FF, FF, 66, 4, 70, FF, 60, B0, 52, 6D, FF, EE, 20, 6D, FF, EA, 85
DATA 4A, 10, 67, A, 30, 2D, FF, EE, B0, 6D, FF, F4, 6D, CE, 30, 2D, FF, BC
DATA EE, D1, 6D, FF, FA, 4A, 6D, FF, F8, 66, 28, 60, 18, 3F, 3C, 0, 20, 74
DATA 4E, 92, 54, 4F, B0, 7C, FF, FF, 66, 6, 70, FF, 60, 0, FF, 78, 52, B1
DATA 6D, FF, FA, 30, 2D, FF, F2, 53, 6D, FF, F2, B0, 6D, FF, F0, 6E, DA, B9
DATA 60, 16, 3F, 4, 4E, 92, 54, 4F, B0, 7C, FF, FF, 66, 6, 70, FF, 60, A1
DATA 0, FF, 52, 52, 6D, FF, FA, 60, 0, FD, A, 30, 2D, FF, FA, 60, 0, 26
DATA FF, 42, 48, E7, 48, 0, 42, 84, 4A, 80, 6A, 4, 44, 80, 52, 44, 4A, 5A
DATA 81, 6A, 6, 44, 81, A, 44, 0, 1, 61, 3E, 4A, 44, 67, 2, 44, 80, 5F
DATA 4C, DF, 0, 12, 4A, 80, 4E, 75, 48, E7, 48, 0, 42, 84, 4A, 80, 6A, 3B
DATA 4, 44, 80, 52, 44, 4A, 81, 6A, 2, 44, 81, 61, 1A, 20, 1, 60, D8, 2E
DATA 2F, 1, 61, 12, 20, 1, 22, 1F, 4A, 80, 4E, 75, 2F, 1, 61, 6, 22, 4B
DATA 1F, 4A, 80, 4E, 75, 48, E7, 30, 0, 48, 41, 4A, 41, 66, 20, 48, 41, 2E
DATA 36, 1, 34, 0, 42, 40, 48, 40, 80, C3, 22, 0, 48, 40, 32, 2, 82, 18
DATA C3, 30, 1, 42, 41, 48, 41, 4C, DF, 0, C, 4E, 75, 48, 41, 26, 1, AA
DATA 22, 0, 42, 41, 48, 41, 48, 40, 42, 40, 74, F, D0, 80, D3, 81, B6, 15
DATA 81, 62, 4, 92, 83, 52, 40, 51, CA, FF, F2, 4C, DF, 0, C, 4E, 75, 94
DATA 20, 6F, 0, 4, 20, 8, 4A, 18, 66, FC, 91, C0, 20, 8, 53, 80, 4E, 19
DATA 75, 4E, 55, 0, 0, 48, 6C, 80, AC, 3F, 2D, 0, 8, 4E, BA, 0, 8, 7C
DATA 5C, 4F, 4E, 5D, 4E, 75, 4E, 55, 0, 0, 2F, 4, 38, 2D, 0, 8, 2F, 8B
DATA 2D, 0, A, 3F, 4, 4E, BA, 0, 30, 5C, 4F, B8, 7C, 0, A, 66, 24, 25
DATA 20, 6D, 0, A, 10, 28, 0, C, 48, 80, 8, 0, 0, 7, 67, 14, 3F, 6C
DATA 3C, FF, FF, 2F, 2D, 0, A, 4E, BA, 0, F6, 5C, 4F, 28, 1F, 4E, 5D, 3B
DATA 4E, 75, 60, F8, 4E, 55, 0, 0, 2F, A, 24, 6D, 0, A, 20, 52, B1, B5
DATA EA, 0, 4, 65, 18, 30, 2D, 0, 8, C0, 7C, 0, FF, 3F, 0, 2F, A, 83
DATA 4E, BA, 0, CA, 5C, 4F, 24, 5F, 4E, 5D, 4E, 75, 20, 52, 52, 92, 10, D4

DATA 2D, 0, 9,10,80,48,80,C0,7C, 0,FF,60,E8,4E,55, 0, 0,B4
DATA 2F, A,41,EC,80,96,24,48,20,4A,D5,FC, 0, 0, 0,16,2F,68
DATA 8,61,10,58,4F,41,EC,82,4E,B5,C8,65,EA,24,5F,4E,5D,17
DATA 4E,75,4E,55, 0, 0,48,E7, 8,20,24,6D, 0, 8,78, 0,20,EE
DATA A,66, A,70,FF,4C,DF, 4,10,4E,5D,4E,75,4A,2A, 0, C,16
DATA 67,52, 8,2A, 0, 2, 0, C,67, C,3F,3C,FF,FF,2F, A,61,7F
DATA 54,5C,4F,38, 0,10,2A, 0, D,48,80,3F, 0,4E,BA, 4,F2,83
DATA 54,4F,88,40, 8,2A, 0, 1, 0, C,67, A,2F,2A, 0, 8,4E,CA
DATA BA, 2,30,58,4F, 8,2A, 0, 5, 0, C,67,14,2F,2A, 0,12,BC
DATA 4E,BA, 2,C2,58,4F,2F,2A, 0,12,4E,BA, 2,14,58,4F,42,E5
DATA 92,42,AA, 0, 4,42,AA, 0, 8,42,2A, 0, C,30, 4,60,8E,10
DATA 4E,55,FF,FE,48,E7, 8,20,24,6D, 0, 8,41,FA,FF,44,29,37
DATA 48,82,7C, 8,2A, 0, 4, 0, C,67, A,70,FF,4C,DF, 4,10,A7
DATA 4E,5D,4E,75, 8,2A, 0, 2, 0, C,67,30,20,12,90,AA, 0,B1
DATA 8,38, 0,3F, 4,2F,2A, 0, 8,10,2A, 0, D,48,80,3F, 0,32
DATA 4E,BA, 2,80,50,4F,B0,44,67,10, 8,EA, 0, 4, 0, C,42,D8
DATA 92,42,AA, 0, 4,70,FF,60,C0, C,6D,FF,FF, 0, C,66,10,A
DATA 8,AA, 0, 2, 0, C,42,92,42,AA, 0, 4,70, 0,60,A8,4A,46
DATA AA, 0, 8,66, 8,2F, A,4E,BA, 0,9A,58,4F, C,6A, 0, 1,19
DATA 0,10,66,2A,1B,6D, 0, D,FF,FF,3F,3C, 0, 1,48,6D,FF,63
DATA FF,10,2A, 0, D,48,80,3F, 0,4E,BA, 2,22,50,4F,B0,7C,44
DATA 0, 1,66,A0,30,2D, 0, C,60, 0,FF,6A,24,AA, 0, 8,30,3F
DATA 2A, 0,10,48,C0,D0,AA, 0, 8,25,40, 0, 4, 8,EA, 0, 2,21
DATA 0, C,20,52,52,92,10,2D, 0, D,10,80,48,80,C0,7C, 0,40
DATA FF,60, 0,FF,3E,4E,55, 0, 0,2F, A,41,EC,80,96,24,48,27
DATA 4A,2A, 0, C,67,18,D5,FC, 0, 0, 0,16,41,EC,82,4E,B5,98
DATA C8,65, 8,70, 0,24,5F,4E,5D,4E,75,60,E2,42,92,42,AA,98
DATA 0, 4,42,AA, 0, 8,20, A,60,EA,4E,55,FF,FC,2F, A,24,67
DATA 6D, 0, 8,3F,3C, 4, 0,4E,BA, 0,C0,54,4F,2B,40,FF,FC,C5
DATA 66,18,35,7C, 0, 1, 0,10,20, A,D0,BC, 0, 0, 0, E,25,29
DATA 40, 0, 8,24,5F,4E,5D,4E,75,35,7C, 4, 0, 0,10, 8,EA,F0
DATA 0, 1, 0, C,25,6D,FF,FC, 0, 8,10,2A, 0, D,48,80,3F,F0
DATA 0,4E,BA, 0,E2,54,4F,4A,40,67, 6, 0,2A, 0,80, 0, C,3A
DATA 60,CE,4E,55, 0, 0,48,E7, 0,30,24,6C,82,52,60,14,26,2E
DATA 52,20,2A, 0, 4,50,80,2F, 0,2F, A,4E,BA, 4,24,50,4F,A7
DATA 24,4B,20, A,66,E8,42,AC,82,52,4C,DF, C, 0,4E,5D,4E,D9
DATA 75,4E,55, 0, 0,2F, A,41,FA,FF,C6,29,48,82,80,42,A7,AD
DATA 20,2D, 0, 8,50,80,2F, 0,4E,BA, 3,CE,50,4F,24,40,4A,7A
DATA 80,66, 8,70, 0,24,5F,4E,5D,4E,75,24,AC,82,52,25,6D,85
DATA 0, 8, 0, 4,29,4A,82,52,20, A,50,80,60,E6,4E,55, 0,36
DATA 0,70, 0,30,2D, 0, 8,2F, 0,61,B2,58,4F,4E,5D,4E,75,2C
DATA 4E,55, 0, 0,48,E7, 0,30,97,CB,24,6C,82,52,60, E,20,56
DATA 6D, 0, 8,51,88,B1,CA,67,12,26,4A,24,52,20, A,66,EE,A6
DATA 70,FF,4C,DF, C, 0,4E,5D,4E,75,20, B,67, 4,26,92,60,C2
DATA 4,29,52,82,52,20,2A, 0, 4,50,80,2F, 0,2F, A,4E,BA,E1
DATA 3,76,50,4F,70, 0,60,D8,4E,55, 0, 0,2F, A,30,2D, 0,F9
DATA 8,C1,FC, 0, 6,24,40,D5,EC,82,62,4A,6D, 0, 8,6D, E,E
DATA 30,2D, 0, 8,B0,6C,82,4E,6C, 4,4A,92,66, E,39,7C, 0,C6

DATA 2,82,84,70,FF,24,5F,4E,5D,4E,75,30,2D, 0, 8,C1,FC,8A
DATA 0, 6,20,6C,82,62,2F,30, 8, 0,4E,BA, 2,A8,58,4F,4A,80
DATA 80,67, 4,70, 1,60, 2,70, 0,60,D8,4E,55, 0, 0,2F,2D,65
DATA 0, 8,4E,BA, 2,72,58,4F,4A,80,66, E,4E,BA, 2,7C,39,28
DATA 40,82,84,70,FF,4E,5D,4E,75,70, 0,60,F8,4E,55, 0, 0,8E
DATA 48,E7, C,20,38,2D, 0, 8,4E,BA, 0,70,30, 4,C1,FC, 0,31
DATA 6,24,40,D5,EC,82,62,4A,44,6D, A,B8,6C,82,4E,6C, 4,78
DATA 4A,92,66,10,39,7C, 0, 2,82,84,70,FF,4C,DF, 4,30,4E,2B
DATA 5D,4E,75,30,2A, 0, 4,CO,7C, 0, 3,66, A,39,7C, 0, 5,E7
DATA 82,84,70,FF,60,E4,70, 0,30,2D, 0, E,2F, 0,2F,2D, 0,1F
DATA A,2F,12,4E,BA, 2,38,4F,EF, 0, C,2A, 0,B0,BC,FF,FF,6B
DATA FF,FF,66, C,4E,BA, 1,FC,39,40,82,84,70,FF,60,B8,20,9B
DATA 5,60,B4,4E,55,FF,FC,48,78,10, 0,42,A7,4E,BA, 2,A0,1A
DATA 50,4F,2B,40,FF,FC, 8, 0, 0, C,67,12,4A,6C,82,6A,66,9A
DATA 8,20,2D,FF,FC,4E,5D,4E,75,4E,BA, 0, 6,70, 0,60,F4,90
DATA 4E,55, 0, 0,48,78, 0, 4,48,7A, 0,1E,4E,BA, 1,D2,2F,51
DATA 0,4E,BA, 1,D4,4F,EF, 0, C,3F,3C, 0, 1,4E,BA, 0, C,B7
DATA 54,4F,4E,5D,4E,75,5E,43, A, 0,4E,55, 0, 0,4A,AC,82,D7
DATA 7C,67, 6,20,6C,82,7C,4E,90,3F,2D, 0, 8,4E,BA, 0, 8,D5
DATA 54,4F,4E,5D,4E,75,4E,55,FF,FC,2F, 4,30,2D, 0, 8,48,8F
DATA C0,2B,40,FF,FC,4A,AC,82,62,67,28,78, 0,60, A,3F, 4,B4
DATA 4E,BA, 0,D0,54,4F,52,44,B8,6C,82,4E,6D,F0,30,2C,82,40
DATA 4E,C1,FC, 0, 6,2F, 0,2F,2C,82,62,4E,BA, 1,CO,50,4F,E7
DATA 4A,AC,82,80,67, 6,20,6C,82,80,4E,90,4A,AC,82,86,67,36
DATA A,2F,2C,82,86,4E,BA, 1,72,58,4F,4A,AC,82,8A,67, A,2
DATA 2F,2C,82,8A,4E,BA, 1,62,58,4F,4A,AC,82,8E,67, A,2F,1F
DATA 2C,82,8E,4E,BA, 1,52,58,4F,2C,78, 0, 4, 8,2E, 0, 4,20
DATA 1,29,67,14,2F, D,4B,FA, 0, A,4E,AE,FF,E2,2A,5F,60,F6
DATA 6,42,A7,F3,5F,4E,73,4A,AC,82,6C,66,30,4A,AC,82,78,6C
DATA 67,28,30,2C,82,76,48,CO,2F, 0,2F,2C,82,78,4E,BA, 1,78
DATA 46,50,4F,30,2C,82,74,52,40,48,CO,E5,80,2F, 0,2F,2C,CO
DATA 82,70,4E,BA, 1,30,50,4F,60, E,4E,BA, 1,20,2F,2C,82,3E
DATA 6C,4E,BA, 1,4C,58,4F,20,2D,FF,FC,2E,6C,82,56,4E,75,E5
DATA 28,1F,4E,5D,4E,75,4E,55, 0, 0,48,E7, E,20,38,2D, 0,1A
DATA 8,30, 4,C1,FC, 0, 6,24,40,D5,EC,82,62,4A,44,6D, A,D
DATA B8,6C,82,4E,6C, 4,4A,92,66,10,39,7C, 0, 2,82,84,70,E3
DATA FF,4C,DF, 4,70,4E,5D,4E,75,30,2A, 0, 4,CO,7C,80, 0,26
DATA 66, 8,2F,12,4E,BA, 0, A,58,4F,42,92,70, 0,60,E0,22,E
DATA 2F, 0, 4,2C,6C,82,5E,4E,EE,FF,DC,22,2F, 0, 4,2C,6C,AF
DATA 82,5E,4E,EE,FF,82,22,2F, 0, 4,2C,6C,82,5E,4E,EE,FF,A5
DATA 3A,22,2F, 0, 4,2C,6C,82,5E,4E,EE,FF,B8,2C,6C,82,5E,72
DATA 4E,EE,FF,CA,2C,6C,82,5E,4E,EE,FF,7C,22,2F, 0, 4,2C,B5
DATA 6C,82,5E,4E,EE,FF,28,4C,EF, 0, 6, 0, 4,2C,6C,82,5E,6C
DATA 4E,EE,FF,E2,2C,6C,82,5E,4E,EE,FF,C4,4C,EF, 0, E, 0,DD
DATA 4,2C,6C,82,5E,4E,EE,FF,DO,48,E7, 1, 4,4C,EF,20,80,96
DATA 0, C,2C,6C,82,5A,4E,AE,FF,94,4C,DF,20,80,4E,75,4E,EB
DATA FA, 0, 2,22,6F, 0, 4,2C,6C,82,5A,4E,EE,FE,62,4C,EF,DC
DATA 0, 3, 0, 4,2C,6C,82,5A,4E,EE,FF,3A,4E,FA, 0, 2,22,5C

DATA 6F, 0, 4,2C,6C,82,5A,4E,EE,FE,DA,2C,6C,82,5A,4E,EE,AB
DATA FF,7C,22,6F, 0, 4,20,2F, 0, 8,2C,6C,82,5A,4E,EE,FF,16
DATA 2E,20,6F, 0, 4,2C,6C,82,5A,4E,EE,FE,8C,2C,6C,82,5A,6F
DATA 22,6F, 0, 4,20,2F, 0, 8,4E,EE,FD,D8,22,6F, 0, 4,2C,BE
DATA 6C,82,5A,4E,EE,FE,86,4C,EF, 0, 3, 0, 4,2C,6C,82,5A,BE
DATA 4E,EE,FE,CE,22,6F, 0, 4,20,2F, 0, 8,2C,6C,82,5A,4E,B6
DATA EE,FE,D4,20,6F, 0, 4,2C,6C,82,5A,4E,EE,FE,80,4C,EF,BC
DATA 3, 0, 0, 4,2C,6C,82,92,4E,EE,FF,A0,20,6F, 0, 4,2C,4D
DATA 6C,82,92,4E,EE,FF,A6,20,6F, 0, 4,2C,6C,82,92,4E,EE,DC
DATA FF,B2, 0, 0, 0, 0, 3,EC, 0, 0, 0, 1, 0, 0, 0, 1, 0,A2
DATA 0, 3,AA, 0, 0, 0, 0, 0, 0, 3,F0, 0, 0, 0, 2,5F,5F,60
DATA 48,30,5F,6F,72,67, 0, 0, 0, 0, 0, 0, 0, 2,5F,6D,61,4E
DATA 69,6E, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 2,5F,56,61,6C,5F
DATA 75,65, 0, 0, 0, 0, 2,2A, 0, 0, 0, 2,2E,62,65,67,69,CD
DATA 6E, 0, 0, 0, 0, 3,36, 0, 0, 0, 2,5F,67,65,74,61,34,DD
DATA 0, 0, 0, 0, 3,A8, 0, 0, 0, 2,5F,5F,6D,61,69,6E, 0,10
DATA 0, 0, 0, 3,B0, 0, 0, 0, 3,5F,5F,63,6C,69,5F,70,61,DC
DATA 72,73,65, 0, 0, 0, 4,EC, 0, 0, 0, 2,5F,73,74,72,6E,62
DATA 63,70,79, 0, 0, 6,92, 0, 0, 0, 2,5F,73,74,72,63,61,62
DATA 74, 0, 0, 0, 6,B4, 0, 0, 0, 2,5F,73,74,72,6E,63,61,1A
DATA 74, 0, 0, 6,BA, 0, 0, 0, 2,5F,73,74,72,63,70,79, 0,3A
DATA 0, 0, 6,DE, 0, 0, 0, 2,2E,6D,75,6C,75, 0, 0, 0, 0,D7
DATA 0, 6,EE, 0, 0, 0, 3,5F,5F,77,62,5F,70,61,72,73,65,8
DATA 0, 0, 0, 0, 7,12, 0, 0, 0, 2,5F,70,72,69,6E,74,66,D
DATA 0, 0, 0, 7,C6, 0, 0, 0, 2,5F,66,6F,72,6D,61,74, 0,B7
DATA 0, 0, 8,70, 0, 0, 0, 2,2E,64,69,76,73, 0, 0, 0, 0,5E
DATA 0, B,8A, 0, 0, 0, 2,2E,6D,6F,64,73, 0, 0, 0, 0, 0,78
DATA B,B2, 0, 0, 0, 2,2E,6D,6F,64,75, 0, 0, 0, 0, 0, B,AD
DATA CC, 0, 0, 0, 2,2E,64,69,76,75, 0, 0, 0, 0, 0, B,D8,97
DATA 0, 0, 0, 2,5F,73,74,72,6C,65,6E, 0, 0, 0, C,32, 0,37
DATA 0, 0, 2,5F,70,75,74,63,68,61,72, 0, 0, C,44, 0, 0,A8
DATA 0, 2,5F,61,70,75,74,63, 0, 0, 0, 0, C,5A, 0, 0, 0,E4
DATA 2,5F,70,75,74,63, 0, 0, 0, 0, 0, C,9C, 0, 0, 0, 2,C7
DATA 5F,66,63,6C,6F,73,65, 0, 0, 0, D, 0, 0, 0, 0, 2,5F,49
DATA 66,6C,73,68,5F, 0, 0, 0, 0, D,86, 0, 0, 0, 3,5F,6E,6F
DATA 65,77,73,74,72,65,61,6D, 0, 0, 0, 0, E,68, 0, 0, 0,DE
DATA 2,5F,67,65,74,62,75,66,66, 0, 0, E,A0, 0, 0, 0, 2,F4
DATA 5F,6C,6D,61,6C,6C,6F,63, 0, 0, F,30, 0, 0, 0, 2,5F,E3
DATA 6D,61,6C,6C,6F,63, 0, 0, 0, F,70, 0, 0, 0, 2,5F,66,BE
DATA 72,65,65, 0, 0, 0, 0, F,84, 0, 0, 0, 2,5F,69,73,C
DATA 61,74,74,79, 0, 0, 0, F,DO, 0, 0, 0, 2,5F,75,6E,6C,51
DATA 69,6E,6B, 0, 0, 0,10,28, 0, 0, 0, 2,5F,77,72,69,74,A1
DATA 65, 0, 0, 0, 0,10,4C, 0, 0, 0, 3,5F,43,68,6B,5F,41,D9
DATA 62,6F,72,74, 0, 0, 0, 0,10,CA, 0, 0, 0, 2,5F,5F,61,B2
DATA 62,6F,72,74, 0, 0, 0,10,FA, 0, 0, 0, 2,5F,65,78,69,68
DATA 74, 0, 0, 0, 0, 0,11,26, 0, 0, 0, 2,5F,5F,65,78,69,B1
DATA 74, 0, 0, 0, 0,11,44, 0, 0, 0, 2,5F,63,6C,6F,73,65,40
DATA 0, 0, 0, 0,12,32, 0, 0, 0, 2,5F,5F,43,6C,6F,73,65,FA

DATA 0, 0, 0,12,80, 0, 0, 0, 3,5F,5F,43,75,72,72,65,6E,C2
DATA 74,44,69,72, 0, 0,12,8C, 0, 0, 0, 2,5F,44,65,6C,61,8
DATA 79, 0, 0, 0, 0,12,98, 0, 0, 0, 3,5F,5F,44,65,6C,65,5E
DATA 74,65,46,69,6C,65, 0, 0,12,A4, 0, 0, 0, 2,5F,5F,49,18
DATA 6E,70,75,74, 0, 0, 0,12,B0, 0, 0, 0, 2,5F,5F,49,6F,1
DATA 45,72,72, 0, 0, 0,12,B8, 0, 0, 0, 4,5F,5F,49,73,49,BA
DATA 6E,74,65,72,61,63,74,69,76,65, 0, 0, 0,12,C0, 0, 0,7
DATA 0, 2,5F,5F,4F,70,65,6E, 0, 0, 0, 0,12,CC, 0, 0, 0,30
DATA 2,5F,5F,4F,75,74,70,75,74, 0, 0,12,DA, 0, 0, 0, 2,3F
DATA 5F,5F,57,72,69,74,65, 0, 0, 0,12,E2, 0, 0, 0, 2,5F,1E
DATA 41,6C,65,72,74, 0, 0, 0, 0,12,FO, 0, 0, 0, 4,5F,43,A0
DATA 6C,6F,73,65,4C,69,62,72,61,72,79, 0, 0, 0, 0, 0,13,9B
DATA 8, 0, 0, 0, 4,5F,5F,43,6C,6F,73,65,4C,69,62,72,61,AA
DATA 72,79, 0, 0, 0, 0,13, C, 0, 0, 0, 3,5F,5F,41,6C,6C,E4
DATA 6F,63,4D,65,6D, 0, 0, 0, 0,13,18, 0, 0, 0, 3,5F,46,C4
DATA 69,6E,64,54,61,73,6B, 0, 0, 0, 0, 0,13,26, 0, 0, 0,7
DATA 3,5F,5F,46,69,6E,64,54,61,73,6B, 0, 0, 0, 0,13,2A,12
DATA 0, 0, 0, 2,5F,5F,46,6F,72,62,69,64, 0, 0,13,36, 0,5F
DATA 0, 0, 3,5F,5F,46,72,65,65,4D,65,6D, 0, 0, 0, 0, 0,62
DATA 13,3E, 0, 0, 0, 2,5F,5F,47,65,74,4D,73,67, 0, 0,13,6B
DATA 4E, 0, 0, 0, 4,5F,5F,4F,70,65,6E,4C,69,62,72,61,72,FE
DATA 79, 0, 0, 0, 0, 0,13,5A, 0, 0, 0, 3,5F,5F,52,65,70,CE
DATA 6C,79,4D,73,67, 0, 0, 0, 0,13,6A, 0, 0, 0, 3,5F,5F,4A
DATA 53,65,74,53,69,67,6E,61,6C, 0, 0, 0,13,76, 0, 0, 0,13
DATA 3,5F,53,65,74,54,61,73,6B,50,72,69, 0, 0, 0,13,84,E3
DATA 0, 0, 0, 3,5F,5F,57,61,69,74,50,6F,72,74, 0, 0, 0,FB
DATA 0,13,94, 0, 0, 0, 4,5F,46,69,6E,64,54,6F,6F,6C,54,7D
DATA 79,70,65, 0, 0, 0, 0, 0,13,A0, 0, 0, 0, 4,5F,46,72,1C
DATA 65,65,44,69,73,6B,4F,62,6A,65,63,74, 0, 0, 0,13,AE,6D
DATA 0, 0, 0, 4,5F,47,65,74,44,69,73,6B,4F,62,6A,65,63,F1
DATA 74, 0, 0, 0, 0,13,BA, 0, 0, 0, 2,5F,5F,48,30,5F,65,3D
DATA 6E,64, 0, 0,13,C8, 0, 0, 0, 0, 0, 0, 3,F2, 0, 0, 3,A5
DATA EA, 0, 0, 0,94,30,31,32,33,34,35,36,37,38,39,61,62,4E
DATA 63,64,65,66, 0, 0, 0,20,20,20,20,20,20,20,20,20,20,30,E2
DATA 30,30,30,30,20,20,20,20,20,20,20,20,20,20,20,20,60
DATA 20,20,20,20,20,90,40,40,40,40,40,40,40,40,40,FO
DATA 40,40,40,40, C, C, C, C, C, C, C, C, C, C,40,40,40,38
DATA 40,40,40,40, 9, 9, 9, 9, 9, 9, 1, 1, 1, 1, 1, 1, 1,3D
DATA 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,40,40,40,40,D
DATA 40,40, A, A, A, A, A, A, 2, 2, 2, 2, 2, 2, 2, 2,CE
DATA 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,40,40,40,40,20, 0,36
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,1
DATA 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,1
DATA 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,3
DATA 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,4
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0
DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0

8 Quick Reference

De eerste twee paragrafen van dit hoofdstuk geven een overzicht van de bediening van de editors ED en EDIT. De instructies zijn gegeven in tabelvorm om gemakkelijker als naslagwerk te kunnen dienen. In paragraaf 8.3 staan de CLI-instructies op een soortgelijke manier geordend.

8.1 ED

Bij de editor ED zijn er twee soorten instructies. De eerste soort wordt meteen uitgevoerd na indrukken van een bepaalde toetsencombinatie. Het tweede type kan alleen worden ingevoerd in de indirecte modus. Een paar functies komen in beide groepen voor.

De instructies in de directe modus bestaan grotendeels uit een combinatie van de Ctrl-toets met een andere toets. U schakelt om naar de indirecte modus met de Esc-toets. Als er in de linker onderhoek van het venster van de editor een sterretje staat, is de editor in de indirecte modus. Voer de instructie in en geef Return. Pas dan heeft de instructie effect.

Directe instructies zonder Ctrl

<i>Tab</i>	Verplaatst de cursor naar de volgende tabulatorstop.
<i>Del</i>	Wist het teken onder de cursor.
<i>Backspace</i>	Wist het teken links van de cursor.
<i>Return</i>	Tekst tussen de cursorpositie en het einde van de regel gaat naar de volgende regel.
<i>Esc</i>	Schakelt over naar de indirecte modus.

Directe instructies met Ctrl

<i>Ctrl-I</i>	Verplaatst de cursor naar de volgende tabulatorstop.
<i>Ctrl-U</i>	Verplaatst de cursor 12 regels naar boven.
<i>Ctrl-D</i>	Verplaatst de cursor 12 regels naar beneden.
<i>Ctrl-R</i>	Verplaatst de cursor naar het eind van het vorige woord.
<i>Ctrl-T</i>	Plaatst de cursor aan het begin van het volgende woord.
<i>Ctrl-J</i>	Verplaatst de cursor naar het begin of het einde van de regel.
<i>Ctrl-E</i>	Verplaatst de cursor naar het begin of het einde van het venster.
<i>Ctrl-H</i>	Wist het teken links van de cursor.
<i>Ctrl-O</i>	Wist een woord of alle spaties tot het volgende woord.
<i>Ctrl-Y</i>	Wist alles vanaf de cursorpositie tot het einde van de regel.
<i>Ctrl-B</i>	Wist de hele actuele regel.
<i>Ctrl-M</i>	Tekst tussen cursorpositie en einde van de regel gaat naar de volgende regel.
<i>Ctrl-A</i>	Voegt een regel tussen.
<i>Ctrl-G</i>	Herhaalt de laatste instructie uit de indirecte modus.
<i>Ctrl-[</i>	Schakelt over naar de indirecte modus.
<i>Ctrl-F</i>	Het teken onder de cursor verandert van een hoofdletter in een kleine letter of omgekeerd.

Instructies in de indirecte modus (met Esc)

De instructies staan vermeld in hoofdletters, de argumenten in kleine letters.

<i>M n</i>	Verplaatst de cursor naar regel n.
<i>CL</i>	Verplaatst de cursor één positie naar links.
<i>CR</i>	Verplaatst de cursor één positie naar rechts.
<i>CS</i>	Verplaatst de cursor naar het begin van de regel.
<i>CE</i>	Verplaatst de cursor naar het einde van de regel.
<i>P</i>	Verplaatst de cursor naar het begin van de vorige regel.
<i>N</i>	Verplaatst de cursor naar het begin van de volgende regel.
<i>T</i>	Verplaatst de cursor naar het begin van de tekst.
<i>B</i>	Verplaatst de cursor naar het einde van de tekst.
<i>BS</i>	Markeert de cursorpositie als begin van het blok.
<i>BE</i>	Markeert de cursorpositie als einde van het blok.
<i>SB</i>	Geeft het gemarkeerde blok weer in het venster.
<i>WB "bestand"</i>	Gemarkeerd blok wordt opgeslagen in 'bestand'.
<i>IB</i>	Gemarkeerd blok wordt gekopieerd naar de cursorpositie.
<i>DB</i>	Gemarkeerd blok wordt gewist.
<i>DC</i>	Teken onder de cursor wordt gewist.
<i>D</i>	Hele actuele regel wordt gewist.
<i>S</i>	Tekst tussen cursorpositie en einde regel gaat naar de volgende regel.
<i>J</i>	Koppelt de actuele regel aan de volgende regel.
<i>I "tekst"</i>	'Tekst' wordt tussengevoegd vóór de actuele regel.
<i>A "tekst"</i>	'Tekst' wordt toegevoegd na de actuele regel.
<i>IF "bestand"</i>	Bestand wordt tussengevoegd op de cursorpositie.
<i>E "tekst1" "tekst2"</i>	'Tekst1' wordt vervangen door 'tekst2'.
<i>EQ "tekst1" "tekst2"</i>	'Tekst1' wordt vervangen na toestemming vooraf.
<i>F "tekst"</i>	Zoekt vanaf cursorpositie naar 'tekst'.
<i>BF "tekst"</i>	Zoekt tot cursorpositie naar 'tekst'.
<i>LC</i>	Wel verschil tussen hoofdletters en kleine letters bij zoeken naar 'tekst'.
<i>UC</i>	Geen verschil tussen hoofdletters en kleine letters.
<i>X</i>	Beëindigt ED en slaat de tekst op.
<i>Q</i>	Beëindigt ED en slaat de tekst niet op.
<i>SA</i>	Slaat de tekst op zonder ED te beëindigen.
<i>RP instructie</i>	'Instructie' wordt herhaald tot er een fout optreedt.
<i>SH</i>	Geeft de actuele instellingen van de editor weer.
<i>U</i>	Maakt wijzigingen in de actuele regel ongedaan.
<i>SL n</i>	Linker kantlijn op positie n.
<i>SR n</i>	Rechter kantlijn op positie n.
<i>EX</i>	Negeert in de actuele regel de rechter kantlijn.

8.2 EDIT

In deze paragraaf geven we een korte samenvatting van de instructies van EDIT. De onderstaande lijst is niet bedoeld als handleiding. Het gebruik van EDIT bespreken we in paragraaf 2.4.2.

Sommige instructies geeft u argumenten mee. De schuine streep is het scheidingsteken tussen strings. Niet-verplichte argumenten staan tussen haakjes. In de syntaxis hebben we de volgende afkortingen gebruikt:

a,b	regelnummers (of . of *)
ig	instructiegroep
m,n	getallen
q	nadere aanduiding
zk	zoekstring
sw	omschakelwaarde (+ of -)
str1,str2	strings

Overzicht van de instructies

<	De cursor gaat één teken naar links.
>	De cursor gaat één teken naar rechts.
#	Het teken op de cursorpositie wordt gewist.
\$	Het teken op de cursorpositie wordt een kleine letter.
%	Het teken op de cursorpositie wordt een hoofdletter.
-	Het teken op de cursorpositie verandert in een spatie.
PA (q)/str1/	De cursor gaat naar de positie na de eventueel nader aangeduide string.
PB (q)/str1/	De cursor gaat naar positie voor de eventueel nader aangeduide string str1.
PR	De cursor gaat naar het begin van de regel.
M n	De cursor gaat naar regel n.
M +	De cursor gaat naar de laatste uit het bronbestand gelezen regel.
M -	De cursor gaat naar de eerste in de buffer aanwezige regel.
N	De cursor gaat naar de volgende regel.
P	De cursor gaat naar de vorige regel.
Rewind	Het bronbestand rolt terug (backscroll).
F ((q)/zk/)	Vooruit zoeken naar de regel die de eventueel nader aangeduide string zk bevat.
BF ((q)/zk/)	Terug zoeken naar de regel die de eventueel nader aangeduide string zk bevat.
DF ((q)/zk/)	Vooruit zoeken naar de regel die de eventueel nader aangeduide string zk bevat. Bovendien worden alle onderzochte regels gewist.
?	De actuele regel wordt gecontroleerd.
!	De actuele regel wordt gecontroleerd, inclusief niet afdrukbaar tekens.
T	Geeft het bronbestand weer tot aan het einde.
T n	Geeft de volgende n regels van het bronbestand weer.
strL n	Geeft de volgende n regels van het bronbestand weer met regelnummers.
strN	Geeft zoveel regels weer als er in de tekstbuffer passen.
strP	Geeft alle regels in de tekstbuffer weer en zet de cursor op het begin van de tekst.
V sw	Schakelt verificatie in (+) of uit (-).
A (q)/str1/str2/	De string str2 wordt tussengevoegd na de eventueel nader aangeduide string str1.
AP (q)/str1/str2/	De string str2 wordt tussengevoegd na de eventueel nader aangeduide string str1. De cursor komt achter str2 te staan.

<i>B (q)/str1/str2/</i>	De string <i>str2</i> wordt tussengevoegd vóór de eventueel nader aangeduide string <i>str1</i> .
<i>BP (q)/str1/str2/</i>	De string <i>str2</i> wordt tussengevoegd vóór de eventueel nader aangeduide string <i>str1</i> . De cursor komt achter <i>str1</i> te staan.
<i>CL (lstr1/)</i>	De actuele regel, <i>str1</i> en de volgende regel worden gecombineerd tot één nieuwe regel.
<i>D</i>	Wist de actuele regel.
<i>DFA (q)/str1/</i>	Wist de actuele regel vanaf de eventueel nader aangeduide string <i>str1</i> .
<i>DFB (q)/str1/</i>	Wist de actuele regel, te beginnen met de eventueel nader aangeduide string <i>str1</i> .
<i>DTA (q)/str1/</i>	Wist de actuele regel vanaf het begin tot en met de eventueel nader aangeduide string <i>str1</i> .
<i>DTB (q)/str1/</i>	Wist de actuele regel vanaf het begin tot aan de eventueel nader aangeduide string <i>str1</i> .
<i>E (q)/str1/str2/</i>	De eventueel nader aangeduide string <i>str1</i> wordt vervangen door <i>str2</i> .
<i>EP</i>	De eventueel nader aangeduide string <i>str1</i> wordt vervangen door de string <i>str2</i> . De cursor komt achter <i>str2</i> te staan.
<i>I (a)</i>	Voegt tekst toe vanaf het toetsenbord totdat u 'z' geeft. De nieuwe tekst wordt geplaatst voor de actuele regel of voor regel <i>a</i> .
<i>I str1</i>	Voegt vóór de actuele regel de hele inhoud van het bestand <i>str1</i> toe.
<i>R (a(b))</i>	Wist de regels <i>a</i> tot en met <i>b</i> en voegt daarna tekst toe vanaf het toetsenbord.
<i>(a(b)) str1</i>	Wist de regels <i>a</i> tot en met <i>b</i> en voegt daarna de inhoud van het bestand <i>str1</i> toe.
<i>SA (q)/str1/</i>	De actuele regel wordt gesplitst na de eerste keer dat de eventueel nader aangeduide string <i>str1</i> optreedt.
<i>SB (q)/str1/</i>	De actuele regel wordt gesplitst vóór de eerste keer dat de eventueel nader aangeduide string <i>str1</i> optreedt.
<i>GA (q)/str1/str2/</i>	In alle actuele regels wordt de string <i>str2</i> tussengevoegd na de eventueel nader aangeduide string <i>str1</i> .
<i>GB (q)/str1/str2/</i>	In alle actuele regels wordt de string <i>str2</i> tussengevoegd vóór de eventueel nader aangeduide string <i>str1</i> .
<i>GE (q)/str1/str2/</i>	De eventueel nader aangeduide string <i>str1</i> wordt in alle actuele regels vervangen door de string <i>str2</i> .
<i>CG (n)</i>	Schakelt de globale operatie 'n' of alle globale operaties uit.
<i>DG (n)</i>	Schakelt de globale operatie 'n' of alle globale operaties tijdelijk uit.
<i>EG (n)</i>	Schakelt de globale operatie 'n' of alle globale operaties weer in.
<i>SHG</i>	Geeft informatie over alle actieve globale operaties.
<i>From</i>	Het oorspronkelijke 'From'-bestand wordt het actuele bronbestand.
<i>From .str1.</i>	Het bestand <i>str1</i> wordt het actuele bronbestand.
<i>To</i>	Het oorspronkelijke 'To'-bestand wordt het actuele doelbestand.
<i>To .str1.</i>	Het bestand <i>str1</i> wordt het actuele doelbestand.
<i>CF .str1.</i>	Sluit het bestand <i>str1</i> af.
,	Herhaalt de laatst gegeven instructie A, B of E.
= <i>n</i>	De actuele regel krijgt nummer <i>n</i> .
<i>C .str1.</i>	Leest instructies voor EDIT uit het bestand <i>str1</i> .
<i>H n</i>	Zet het volgende markeringspunt bij regel <i>n</i> . (Als <i>n</i> = *, stoppen en oude markeringspunt wissen.)
<i>Q</i>	Verlaat het instructie-niveau. Als dat het hoogste niveau is, wordt de rest van

	de tekst aan het bronbestand doorgegeven.
<i>SHD</i>	Geeft de opgeslagen informatie over de instructies weer.
<i>Stop</i>	Verlaat de editor EDIT.
<i>TR sw</i>	Houdt in het vervolg wel (+) of geen (-) rekening met spaties aan het einde.
<i>W</i>	De rest van het bronbestand gaat naar het doelbestand.
<i>Z str1</i>	Verandert het einde-teken voor invoeginstructies (verstekwaarde: '/').

8.3 CLI-instructies

In deze paragraaf komen alle CLI-instructies nog eens aan de orde, zij het niet zo uitvoerig als in hoofdstuk 2. In de eerste opsomming staan de instructies gesorteerd naar hun taak. We geven steeds eerst de complete syntaxis van de instructie, daarna een omschrijving van de functie en ten slotte een beschrijving van de opties.

De tweede lijst bevat alle CLI-instructies in alfabetische volgorde, elk voorzien van een korte omschrijving.

8.3.1 CLI-instructies in functiegroepen

FORMAT DRIVE drive NAME "naam"

Formateert een diskette en geeft er een naam aan.

<i>drive</i>	De drive waar de te formatteren diskette in zit.
<i>naam</i>	De geformatteerde diskette gaat "naam" heten.

DIR naam (OPT (A)(I)(AI)(D))

Geeft de inhoudsopgave van een diskette weer.

<i>naam</i>	Een drive of directory.
<i>OPT A</i>	Levert alle bestanden van de directory, inclusief de subdirectories daarvan en hun inhoud.
<i>OPT I</i>	Geeft de inhoudsopgave interactief weer. Na elk bestand kiest u uit een van de volgende instructies:
<i>?</i>	Weergave van de mogelijke instructies.
<i>B</i>	Verder met het volgende bestand.
<i>E</i>	Voert de inhoud van het bestand uit.
<i>Del</i>	Wist het bestand.
<i>Q</i>	Beëindigt de DIR-opdracht.
<i>OPT AI</i>	Combineert de opties A en I.
<i>OPT D</i>	Geeft alleen de directories weer.

CD naam

Schakelt om naar een andere drive of directory.

naam De nieuwe drive of directory.

MAKEDIR naam

Maakt een nieuwe directory aan.

naam De nieuwe directory.

DELETE naam (ALL)(QUIET)(Q)

Wist bestanden en/of directories.

naam Het te wissen bestand.
ALL Wist de directory inclusief de hele inhoud.
QUIET Geen wismeldingen op het scherm.
Q Zie QUIET.

COPY naam1 TO naam2 (ALL)(QUIET)

Maakt een kopie van een bestand of een directory.

naam1 Het bronbestand.
naam2 Het doelbestand.
ALL Kopieert een directory inclusief de inhoud.
QUIET Geen kopieermeldingen op het scherm.

RENAME naam1 TO naam2

Geeft een andere naam aan een bestand.

naam1 Het bestand dat een andere naam krijgt.
naam2 De nieuwe naam.

LIST (naam)(P patroon)(KEYS)(NODATES)(S tekst)(SINCE datum) (UPTO datum)(QUICK)

Levert een lijst met gegevens over bestanden.

naam De lijst heeft alleen betrekking op het bestand 'naam'.

<i>P patroon</i>	De lijst bevat gegevens over bestanden met namen die voldoen aan het vermelde patroon.
KEYS	Vermeldt van elke boeking het beginblok.
NODATES	Geen data in de uitvoer.
<i>S tekst</i>	Alleen gegevens over bestanden waarvan de naam de string 'tekst' bevat.
<i>SINCE datum</i>	De instructie werkt alleen met bestanden die zijn weggeschreven na de vermelde datum.
<i>UPTO datum</i>	De instructie werkt alleen met bestanden die zijn weggeschreven voor de vermelde datum.
QUICK	Geeft alleen de bestandsnamen weer.

DISKCOPY drive1 TO drive2 (NAME naam)

Maakt een kopie van een diskette.

<i>drive1</i>	De brondrive.
<i>drive2i</i>	De doeldrive.
<i>naam</i>	De naam die de kopie krijgt.

RELABEL drive naam

Geeft een diskette een andere naam.

<i>driveI</i>	De drive met de diskette waarvan u de naam wilt veranderen.
<i>naam</i>	De naam die de diskette krijgt.

INFO

Geeft informatie op het scherm over de aangesloten drives.

INSTALL drive

Maakt een systeemdiskette (bootable disk) van een lege diskette.

<i>drive</i>	De drive met de te installeren diskette.
--------------	--

TYPE naam1 (TO naam2)(OPT(N)(H))

Geeft de inhoud van een bestand weer.

<i>naam1</i>	Het bronbestand.
<i>naam2</i>	Het doelbestand waarheen 'naam1' wordt gekopieerd. Bij afwezigheid van een doelbestand gaat de uitvoer naar het actuele uitvoerkanaal.

TN De uitgevoerde regels worden voorzien van regelnummers.
TH De tekens worden uitgevoerd als hexadecimale getallen.

JOIN naam1 naam2 AS naam3

Voegt twee of meer bestanden samen tot één bestand.

naam1 Een bestand dat met andere bestanden moet worden samengevoegd.
naam2 Een ander bestand.
naam3 Het nieuwe bestand waarin de gecombineerde bestanden worden geschreven.

SEARCH naam SEARCH tekst (ALL)

Zoekt in bestanden naar de vermelde string.

naam Het bestand waarin moet worden gezocht.
tekst De string waarnaar moet worden gezocht.
ALL Zoekt ook in de subdirectories van 'naam'.

SORT naam1 TO naam2 (COLSTART n)

Sorteert een bestand op alfabet en slaat het op in een ander bestand.

naam1 Het bronbestand.
naam2 Het bestand waarin de gesorteerde tekst wordt geschreven.
COLSTART n n is de kolom van waaraf de opdracht sorteert.

PROTECT naam status

Definieert de manier waarop een bestand mag worden gebruikt.

naam Het bestand waarvoor de beschermingsstatus geldt.
status De bescherming. Maak een of meer keuzen uit:
R Het bestand mag worden gelezen.
W Het bestand mag worden beschreven.
D Het bestand mag worden gewist.
E Het bestand is uitvoerbaar.

FILENOTE naam COMMENT tekst

Voegt commentaar toe aan een bestand.

naam Het bestand dat een toelichting krijgt.
tekst Het commentaar bij het bestand.

SETDATE naam DATE datum (TIME tijd)

Voegt datum en tijd toe aan een bestand.

naam Het bestand waar datum en/of tijd bij worden geschreven.
datum De datum die bij het bestand moet worden geschreven.
tijd De tijd die bij het bestand moet worden geschreven.

DISKDOCTOR drive

Verhelpt zo nodig fouten op een diskette of in bestanden.

drive De drive die DiskDoctor moet controleren.

DISKCHANGE drive

Meldt AmigaDOS dat er een diskette is verwisseld.

drive De drive waarin van diskette is gewisseld.

NEWCLI (CON:x/y/breedte/hoogte/(tekst))

Opent een nieuw CLI-venster.

x De x-positie van de linker bovenhoek van het nieuwe venster.
y De y-positie van de linker bovenhoek van het nieuwe venster.
breedte De breedte van het venster in pixels.
hoogte De hoogte van het venster in pixels.
tekst De naam van het nieuwe venster.

ENDCLI

Sluit het CLI-venster.

RUN opdracht

Voert een opdracht uit op de achtergrond.

opdracht Een AmigaDOS-instructie die op de achtergrond wordt uitgevoerd.

STATUS (nummer)(FULL)(TCB)(CLI)

Levert informatie over CLI-processen (taken).

<i>nummer</i>	Het nummer van het proces waar u informatie over wilt hebben.
<i>FULL</i>	Combineert de opties TCB en CLI.
<i>TCB</i>	Levert informatie over prioriteit en stack-grootte van het proces.
<i>CLI</i>	Somt de actueel opgeroepen instructies van de processen op.

CHANGETASKPRI n

Kent een prioriteit toe aan een proces.

<i>n</i>	De nieuwe prioriteit.
----------	-----------------------

BREAK nummer (ALL)

Onderbreekt een proces.

<i>nummer</i>	Het te onderbreken proces.
<i>ALL</i>	Activeert alle onderbrekingsniveaus (C, D, E en F).

PATH (naam ADD)(SHOW)((naam) RESET)

Toont of verandert de paden naar de directories.

<i>naam ADD</i>	Er wordt een pad aangelegd naar de directory 'naam'.
<i>SHOW</i>	Geeft de actuele paden weer.
<i>naam RESET</i>	Alle paden worden gewist, behalve die naar 'C' en 'naam'.

ASSIGN ((naam1) naam2)(LIST)

Koppelt een logisch apparaat aan een directory.

<i>naam1</i>	Het logische apparaat.
<i>naam2</i>	De directory waaraan het logische apparaat wordt gekoppeld.
<i>LIST</i>	Levert een lijst van de actuele toewijzingen.

ADDBUFFERS drive n

Reserveert een geheugengedeelte voor een diskdrive.

<i>drive</i>	De drive waaraan de buffer wordt toegewezen.
--------------	--

n De grootte in blokken van de te reserveren buffer.

WHY

Levert informatie over de laatst opgetreden fout.

FAULT *n*

Levert de melding bij de gegeven foutcode.

n Een geldige foutcode.

DATE (datum)(tijd)(TO naam)

Invoer of uitvoer van datum en/of tijd.

datum De in te voeren datum.

tijd De in te voeren tijd.

naam Een bestand waarin de datum en/of de tijd moeten worden geschreven. Bij verstek geldt het actuele uitvoerkanaal.

SETCLOCK (OPT LOAD)(OPT SAVE)

Transport van datum en tijd van het systeem naar de permanente klok en omgekeerd.

OPT LOAD De tijd van de permanente klok wordt de systeemtijd.

OPT SAVE De permanente klok wordt gelijk gezet met de systeemtijd.

PROMPT tekst

Verandert de prompt van de CLI-processen.

tekst Het uiterlijk van de prompt. Hierin is '%*n*' het nummer van het proces.

STACK (*n*)

Verandert de grootte van de stack of levert de actuele grootte.

n De grootte van de stack in bytes.

BINDDRIVERS

Voegt een driver toe aan het systeem.

MOUNT naam

Installeert een device.

naam Een nieuw device.

EXECUTE naam (tekst)

Voert een batchbestand uit.

naam De naam van het batchbestand.

tekst De argumenten die worden meegegeven aan het batchbestand.

ECHO "tekst"

Geeft een tekst weer via het actuele uitvoerkanaal.

"tekst" De tekst die op het scherm moet komen.

FAILAT (n)

Definieert de foutdrempel voor afbreken of voert de actuele waarde uit.

n De waarde van de foutdrempel voor afbreken.

QUIT (n)

Onderbreekt een batchbestand bij de gegeven foutcode.

n Een foutcode.

**IF (NOT)(WARN)(ERROR)(FAIL)(tekst1 EQ tekst2)(EXISTS naam)
instructiel
(ELSE instructie2)
ENDIF**

Beslist of bepaalde instructies in een batchbestand wel of niet worden uitgevoerd.

NOT	Logische omkering van een voorwaarde.
WARN	Waar als de foutcode groter of gelijk is aan 5.
ERROR	Waar als de foutcode groter of gelijk is aan 10.
FAIL	Waar als de foutcode groter of gelijk is aan 20.
tekst1 EQ tekst2	Waar als 'tekst1' gelijk is aan 'tekst2'.
EXISTS naam	Waar als het bestand 'naam' bestaat.
instructie1	Deze CLI-instructie wordt uitgevoerd als is voldaan aan de voorwaarde.
ELSE instructie2	Deze CLI-instructie wordt eventueel uitgevoerd als niet is voldaan aan de voorwaarde.

SKIP tekst

Springt in een batchbestand naar een label.

tekst Bevat een string die dient als insprongpunt als hij is gedefinieerd als label.

LAB tekst

Definieert een string als insprongpunt.

tekst De string.

ASK "tekst"

Voert een ja/nee-vraag uit en wacht op antwoord. Het antwoord 'Y(es)' levert de foutcode 5, 'N(o)' geeft geen foutcode.

"tekst" De ja/nee-vraag die op het scherm moet verschijnen.

WAIT (n)(SEC)(MIN)(UNTIL tijd)

Brengt het systeem in de wachtmodus.

<i>n</i>	Het aantal eenheden van de wachtperiode.
<i>SEC</i>	Definieert de seconde als eenheid (verstekwaarde).
<i>MIN</i>	Definieert de minuut als eenheid.
<i>UNTIL tijd</i>	Wacht tot het vermelde tijdstip.

8.3.2 Alfabetische lijst van CLI-instructies

AddBuffers

Reserveert een hoeveelheid geheugen voor een diskdrive.

Ask

Voert een ja/nee-vraag uit en wacht op antwoord. 'Y(es)' levert de foutcode 5, 'N(o)' geeft geen foutcode.

Assign

Koppelt een directory aan een logisch apparaat.

BindDrivers

Voegt een driver toe aan het systeem.

Break

Onderbreekt een proces (taak).

CD

Maakt een andere drive of directory actueel.

ChangeTaskPri

Geeft een nieuwe prioriteit aan een proces.

Copy

Maakt een kopie van een bestand of directory.

Date

Invoer of uitvoer van datum en/of tijd.

Delete

Wist bestanden en/of directories.

Dir

Voert de inhoudsopgave van een diskette uit.

DiskChange

Meldt AmigaDOS dat er een diskette is verwisseld.

DiskCopy

Maakt een kopie van een diskette.

DiskDoctor

Herstelt zo nodig fouten op diskettes of in bestanden.

Echo

Voert een tekst uit via het actuele uitvoerkanaal.

EndCLI

Sluit het CLI-venster.

Execute

Voert een batchbestand uit.

Failat

Definieert de foutdrempel voor afbreken of voert de actuele waarde uit.

Fault

Voert tekst uit bij een foutcode.

FileNote

Schrijft commentaar bij een bestand.

If (Else) EndIf

Beslist of bepaalde instructies in een batchbestand wel of niet worden uitgevoerd.

Info

Levert informatie over de aangesloten diskdrives.

Install

Maakt een systeemdiskette van een lege diskette.

Join

Voegt twee of meer bestanden samen tot één nieuw bestand.

Lab

Definieert een insprongpunt.

List

Levert een lijst met informatie over bestanden.

MakeDir

Maakt een nieuwe directory aan.

Mount

Installeert een device.

NewCLI

Opent een nieuw CLI-venster.

Path

Toont of verandert het pad naar een directory.

Prompt

Verandert de CLI-prompt.

Protect

Definieert de toegankelijkheid van een bestand.

Quit

Onderbreekt een batchbestand bij optreden van een eventueel meegegeven foutcode.

ReLabel

Geeft een diskette een andere naam.

Rename

Geeft een bestand een andere naam.

Run

Voert een opdracht uit als achtergrondtaak.

Search

Zoekt in bestanden naar de opgegeven string.

SetClock

Geeft datum en tijd van het systeem door aan de permanente klok of omgekeerd.

SetDate

Schrijft datum en tijd bij een bestand.

Skip

Springt in een batchbestand naar een label.

Sort

Sorteert een bestand alfabetisch en slaat het resultaat op in een ander bestand.

Stack

Verandert de grootte van de stack of voert de actuele grootte uit.

Status

Leverd informatie over CLI-processen.

Type

Geeft de inhoud van een bestand weer.

Wait

Brengt het systeem in de wachtmodus.

Why

Leverd informatie over de laatst opgetreden fout.

9 Workbench versie 1.3

In dit hoofdstuk bespreken we de extra mogelijkheden en de nieuwe instructies van de Workbench versie 1.3. De Kickstart 1.3 is volledig compatibel met de versie 1.2, maar het is nu ook mogelijk te booten vanaf een harddisk of zelfs vanuit RAM.

De Workbench is daarentegen flink veranderd. Vooral op CLI-niveau zijn de verschillen goed merkbaar. De nieuwe versie is een aanzienlijke verbetering ten opzichte van de versie 1.2.

Waarschuwing:

Onderstaande informatie is gebaseerd op een gamma-versie van de Workbench. Naar alle waarschijnlijkheid verandert er niet veel meer voordat de definitieve versie verschijnt, maar er kunnen verschillen bestaan.

Voor de duidelijkheid geven we eerst een overzicht van de verschillen tussen de Workbench 1.2 en 1.3. Daarna volgt een uitgebreide toelichting op de instructies.

Fast Filing System

Het Fast Filing System bevat snellere routines voor lezen en schrijven met harddisks.

Printer

Versie 1.3 heeft volledig nieuwe printer- en devicedrivers. Die kunnen met een hogere snelheid grafiek printen en bovendien met een betere kwaliteit.

Mathematische bibliotheek (Math Library)

De routines in deze bibliotheek zijn sneller. Ze nemen automatisch waar of er een coprocessor 68881 aanwezig is. Zo ja, dan wordt deze meteen gebruikt.

Tekstuitvoer naar beeldscherm

Door middel van FastFonts verloopt de uitvoer naar het scherm ongeveer vier keer zo snel.

Reset-bestendige RAM-disk

Deze RAM-disk bewaart zijn inhoud tot u de stroom uitschakelt.

Adobe scherm-fonts

Times Roman, Helvetica en Courier zijn standaardopties (nog niet beschikbaar in de gamma-versie).

Een terminal bij de Amiga

De nieuwe handler AUX: maakt het mogelijk dat een ander de Amiga tegelijk met u gebruikt via een terminal.

Uitvoer van spraak

De nieuwe handler SPEAK: vertaalt tekst in spraak.

Uitvoer van een programma als invoer voor een ander programma

Door middel van de nieuwe handler PIPE: kunt u de uitvoer van het ene programma omleiden tot invoer van een ander.

Nieuwe Preferences

Preferences 1.3 ondersteunt de nieuwe printerdrivers en de real-time clock.

Nieuwe CLI-instructies

Lock, FixFonts, NewShell, More, CMD, ASK, Avail, GetEnv, SetEnv, RemRAD, Resident, SetAlert, Which, Xicon, FastMemFirst, NoFastMem, InitPrinter.

Krachtiger CLI-instructies

Default CLI-Startup, Shell-Startup files, Say, GraphicDump, Copy, List, Break, Status, ChangeTaskPri, DiskDoctor, Date, Echo, Skip, Execute, Mount, Install, Protect, SetDate, Search, Version, Assign, Info, Run, Dir, MicroEmacs.

9.1 Fast Filing System

Het Fast Filing System is een nieuwe manier om data van en naar een harde schijf te transporteren. We duiden het in het vervolg aan met FFS. Het is gebaseerd op het oude filing system (FS), maar het is nu volledig herschreven in Assembler. Bovendien is een groot aantal routines volledig nieuw opgezet met het oog op een zo groot mogelijke snelheid.

Het FFS gebruikt een iets andere manier van gegevensopslag dan het FS zodat er ongeveer 50 Kb per Mb méér op een harde schijf kan worden weggeschreven. Het is een nadeel van het nieuwe formaat dat u de DiskDoctor niet meer kunt gebruiken. Die gaat uit van het standaard Amiga-formaat.

Aangezien de Commodore diskcontroller van het type 2090 de eerste partitie van een harddisk als een normaal FS opvat, mag u deze partitie niet formatteren met FFS. Alle andere partities mag u wel met FFS gebruiken omdat ze met de instructie Mount worden geïnstalleerd. U moet deze partities vooraf opnieuw formatteren met de optie 'FFS'.

Doe dat als volgt:

- Maak een backup van uw harde schijf.
- Kopieer de nieuwe files l:FastFileSystem, c:Mount en c:Format naar uw bootdisk.
- Als uw harde schijf al naar tevredenheid is gepartitioneerd, voegt u aan elke partitie in de MountList de volgende regels toe:

```
GlobVec = -1
FileSystem = l:FastFileSystem
```

- Verander de waarden voor LowCyl en HiCyl niet. Als u dat toch doet, moet u de harddisk opnieuw partitioneren. Indien u de partities nieuwe namen geeft, moet u ze ook veranderen in de Startup-Sequence bij de Mount-opdracht.
- Reset de computer. Op het moment dat de partities worden geïnstalleerd met de Mount-opdracht, krijgt u de melding 'Not a DOS disk'. Cancel deze mededeling.
- Formateer de FFS-partities met de nieuwe instructie Format (de partitie heet fs1:):

```
Format drive fs1: name "FAST1" FFS
```

- Als u niet tevreden bent met de partitionering van de harde schijf is het nodig de drive opnieuw te installeren met de instructie Prep. Formateer daarna alle partities opnieuw.

Wijzig de MountList en maak lijsten aan voor de nieuwe FFS-partities. De waarden LowCyl en HighCyl hangen af van het aantal cilinders dat u definieert voor het mount-gedeelte met het normale FS.

Voorbeeld:

Neem aan dat u werkt met een harddisk ST506 20 Mb en dat u 880 Kb wilt reserveren voor de Workbench. Dat vraagt 26 cilinders op uw harde schijf.

Deel de harddisk als volgt in:

cilinders	0 - 1	gereserveerd (RES0:)
cilinders	2 - 28	eerste partitie, normaal FS (DH0:)
cilinders	29 - 611	partitie onder FFS

Reset de computer als u klaar bent met Prep. Neem achtereenvolgens de volgende maatregelen:

- Wijzig de MountList en voeg een lijst toe waarin de FFS partitie wordt gedeclareerd. De regels GlobVec en FileSystem zijn nodig voor elke FFS-partitie. Voorbeeld van zo'n lijst:

```
FAST:      Device = hddisk.device
           FileSystem = 1:FastFileSystem
           Unit = 1
           Flags = 0
           Surfaces = 4
           BlocksPerTrack = 17
           Reserved = 2
           Interleave = 0
           LowCyl = 29 ; HighCyl = 611
           Buffers = 20
           GlobVec = 1
           BufMemType = 1
           Mount = 1
           DosType = 0x444F5301
           Stacksiz = 4000
```

#

- Formateer uw eerste partitie (DH0:) voor het normale FS:

```
Format drive DH0: name AmigaHd
```

- Mount de FFS-partitie:

```
Mount FS1:
```

- Formateer de FFS-partitie:

```
Format drive FS1: name "FAST1" FFS
```

- Wijzig de Startup-Sequence en voeg de regel 'Mount FS1:' na de instructie 'BindDrivers'.

De CLI-instructies

Alle files krijgen onder Workbench 1.3 vier extra status-bits mee. Ze worden besproken bij de instructie Protect. U zult ze ook tegenkomen bij enkele andere instructies.

We hanteren dezelfde syntax als in de rest van het boek (/A,/K,/S). Verder gebruiken we de volgende tekens:

| is het of-teken,

[] betekent optioneel,

tussen < > staat een variabele.

ADDBUFFERS

syntax: AddBuffers "Drive/A, Buffers/A"

werking: Voegt cache-geheugenblokken toe aan een drive.

nieuw in 1.3: Maakt ook onder FFS kortere toegangstijden tot een harddisk mogelijk.

ASK

syntax: Ask "Prompt/A"

werking: Vraagt invoer van u tijdens uitvoeren van een tekstfile (execute).

nieuw in 1.3: Ask schrijft de prompt-string naar de actuele uitvoer-optie (meestal het scherm) en wacht dan op invoer. Geldige invoer is Y (yes (ja)), N (no (nee)) of Return (hetzelfde als nee). De statusvlag wordt op 5 gezet (waarschuwing) als de invoer 'ja' was en op 0 als u 'nee' invoerde.

ASSIGN

syntax: Assign "Naam, Dir, List/S, Exists/S"

werking: Kent een FS-device toe aan een logische naam.

nieuw in 1.3: Printen van meer dan drie letters werkt nu goed. Assign zoekt in de lijst naar een gegeven naam als u de optie Exists gebruikt samen met een naam. Assign levert als antwoord 0 als het zoekwoord bestaat. Anders is het antwoord 5 (waarschuwing).

AVAIL

syntax: Avail

werking: Levert de hoeveelheid beschikbaar geheugen.

nieuw in 1.3: Nieuwe instructie.

CLI

syntax: CLI

werking: Een nieuwe CLI starten vanaf de Workbench.

nieuw in 1.3: Het eerste wat een nieuwe CLI nu doet is uitvoeren van de instructies in het bestand S:CLI-Startup. Door middel van de Workbench-optie INFO geeft aan de CLI-icon de volgende tooltypes mee:

- WINDOW (vensterspecificatie). Voorbeelden:

```
CON:0/10/640/100/NewCLI  
NEWCON:0/10/640/100/AmigaShell
```

- STACK (definieert de startgrootte van de stack)

Als u de Shell-seg SHELL gebruikt, zal de CLI-icon een SHELL activeren in plaats van een CLI.

CLOCK

syntax: **CLOCK** [analog|digital|digital2] [= <x>, <y>, <breedte>, <hoogte>] [12hour|24hour] [seconds] [date]

werking: Geeft tijd en datum.

nieuw in 1.3: In de opdracht kunt u nu het type klok definiëren en bovendien de plaats van de klok op het scherm, en de grootte van de klok. In de de optie Digital2 opent u een menubalk met de tijd.

CLOCKPTR

syntax: **Clockptr**

werking: Verandert de muispijl in een digitale klok.

nieuw in 1.3: Nieuwe instructie, werkt alleen in het Workbench-venster. Als u de klok naar de linkerkant van het beeldscherm verschuift, verandert hij in de datum: linksboven op het scherm worden de seconden weergegeven. Ga terug naar de normale muispijl met Ctrl-C of met de instructie BREAK als u CLOCKPTR hebt geactiveerd als achtergrondprogramma. De instructie werkt alleen als de Workbench is geladen.

CMD

syntax: **CMD** [-s] [-m] [-n] <apparaatnaam> <filenaam> of
CMD <apparaatnaam> <filenaam> [opt s m n]
(dit hangt af van de definitieve 1.3-versie)

werking: Leidt een seriële of een parallele WRITE om naar een file.

nieuw in 1.3: Nieuwe instructie die printeruitvoer direct naar een bestand schrijft.

De volgende opties zijn mogelijk:

- s (SKIP) sla elke korte initiële schrijfactie over (bijvoorbeeld een reset wanneer een hardcopy van het scherm maakt).

- m geïnstalleerd voor meer dan een bestand tot Ctrl-C of Break.
- n schakelt NOTIFY in (geeft meldingen met de stand van zaken).

<apparaatnaam> is SERIAL (serieel) of PARALLEL (parallel).
 <filenaam> is de volledige naam van het bestand waarin de informatie wordt geschreven.

U mag CMD ook activeren vanuit de Workbench. De volgende parameters kunt u meegeven aan het 'Tool Type'-venster van de CMD-icon:

DEVICE	parallel of serial
FILE	naam van het uitvoerbestand
SKIP	TRUE (ja, overslaan) of FALSE (nee, niet overslaan)
MULTIPLE	TRUE (ja, installeren) of FALSE (nee)
NOTIFY	TRUE (ja, inschakelen) of FALSE (nee)

COPY

syntax: Copy "From,To/A, All/S, Quiet/S, Buf = Buffer/K, Clone/S, Date/S, Nopro/S, Com/S"

werking: Kopieert een bestand of een directory.

nieuw in 1.3: De optie Buf = reserveert het aantal buffers van 512 bytes, die worden gebruikt tijdens het kopiëren. Maak dat aantal niet te groot bij kopiëren naar een ramdisk.

De optie All kopieert een complete directory.

De tekens "" verwijzen naar de actuele directory, als u deze gebruikt als doel.

Voorbeeld:

copy rad:?? to "".

Normaal gesproken maakt de instructie COPY gebruik van de protection bits van het oorspronkelijke bestand. Dit kunt u omzeilen in de optie Nopro.

In de opties Date en Com worden respectievelijk datum en commentaar van het oorspronkelijke bestand meegenomen naar de kopie van het bestand.

In de optie Clone gaan alle attributen mee naar de kopie (datum, protection-bits en commentaar).

Als u kopieert naar een directory die nog niet bestaat, maakt Copy deze aan.

DATE

syntax: Date "Date, Time, To = Ver/K"

werking: Stelt datum en/of tijd in of geeft ze weer.

nieuw in 1.3: De instructie Date accepteert een 0 als begin van een getal of een getal van één cijfer.

DELETE

syntax: Delete " ,,,,,,,All/S, Quiet/S"
werking: Maximaal tien files of directories wissen.
nieuw in 1.3: Als u meer dan een file hebt vermeld, stopt Delete niet wanneer er in één een fout optreedt.

DIR

syntax: Dir "Dir, Opt/K, All/S, Dir/S, Inter/S"
werking: Geeft een gesorteerde lijst weer van de files in een directory.
nieuw in 1.3: Dir heeft nu synoniemen voor de opties A, D en I (All, Dirs en Inter). In de modus Inter (interactief) kunt u via de optie Command= bijna elke opdracht laten uitvoeren.
(Met de optie Dirs krijgt u alleen de directories te zien.)

DISKDOCTOR

syntax: DiskDoctor "Drive/A"
werking: Probeert een defecte diskette te herstellen.
nieuw in 1.3: Diskdoctor controleert of er genoeg geheugen aanwezig is om het programma uit te voeren. Diskdoctor kan worden gebruikt voor een FFS-partitie mits u in de MountList het juiste DosType hebt gedefinieerd om kenbaar te maken dat het om een FFS-partitie gaat. Gebruik DiskDoctor niet voor de FFS-partitie als u dat hebt nagelaten.

ECHO

syntax: Echo "Noline"
werking: Print een string.
nieuw in 1.3: Met de optie Noline, voert de instructie Echo geen nieuwe regel uit na het printen van de string.

EXECUTE

syntax: Execute < naam van file waarin de instructies staan > < parameters >
werking: Voert een bestand uit met instructies (meestal een tekstbestand).
nieuw in 1.3: De instructie Execute gebruikt de logische naam 'T:'mits deze is gedeclareerd met Assign. Zo niet, dan wordt de directory T: gebruikt.

Als u met SHELL werkt is de instructie Execute is zeer geschikt voor residente toepassingen omdat tekstbestanden de instructie Execute aanroepen zodra de Script-bit is gezet.

FASTMEMFIRST

syntax: Fastmemfirst
werking: Herschikt de geheugenindeling zo dat gewoon Autoconfig-RAM voor andere geheugenuitbreiding(en) komt te staan.

FF

syntax: FF -0 -n
werking: Versnelt tekstuitvoer naar het scherm.
nieuw in 1.3: FF (FastFonts) is een programma met twee opties:
-0 FF aanzetten
-n FF uitzetten

FIXFONTS

syntax: Fixfonts
werking: Werkt de .font-bestanden bij in de directory FONTS:.
nieuw in 1.3: Een nieuwe instructie. U kunt hem gebruiken als u nieuwe fonts hebt aange-
maakt of verwijderd in een van de subdirectories van de directory FONTS:.
Het programma corrigeert alle .font-bestanden om ze in overeenstemming te
brengen met de actuele inhoud van de subdirectories in de directory
FONTS:.

FORMAT

syntax: Format "Drive/A, Naam/A, Noicons/S, Quick/S, Ffs/S, Noffs/S"
werking: Formateert en initialiseert een diskette, harddisk of een RAM-disk.
nieuw in 1.3: Met de optie Quick maakt de instructie Format alleen het rootblock (met het
spoor), een bootblock (met het spoor) en de bitmap-blokken aan. Dat is van
nut voor herformatteren van een harde schijf.

De optie Ffs markeert het doelmedium voor gebruik met het Fast Filing System.
De optie Noffs markeert het doelmedium voor gebruik met het normale Filing System.
Met de laatste twee opties kunt u uw schijf anders formatteren dan staat aangegeven in uw
MountList. Doe dat niet. Gebruik in de MountList het juiste DosType bij een gegeven partitie.

GETENV

syntax: Getenv "Naam/A"
werking: Levert de waarde van een omgevingsvariabele.
nieuw in 1.3: Een nieuwe instructie. De waarden worden opgeslagen in de ENV:-handler.
Dat is nu nog de RAM-disk, maar in de toekomst wordt dit een echte hand-
ler.

GRAPHICDUMP

syntax: Graphicdump [Tiny|Small|Medium|Large|xpunten:ypunten]
werking: Maakt een screendump.
nieuw in 1.3: Na invoer van deze instructie volgt een pauze van 10 seconden om u de gelegenheid te geven het gewenste scherm op de voorgrond te zetten. (De instructie werkt alleen met het voorste scherm van Intuition.)

De volgende opties zijn mogelijk:

Tiny De afdrukbreedte is ongeveer een vierde van de maximale breedte van de printer. De hoogte wordt aangepast.
Small De afdrukbreedte is ongeveer de helft van de maximale breedte van de printer. De hoogte wordt aangepast.
Medium De afdrukbreedte is ongeveer driekwart van de maximale breedte van de printer. De hoogte wordt aangepast.
Large De afdrukbreedte is gelijk aan de maximale breedte van de printer. De hoogte wordt aangepast.
xpunten:ypunten De absolute afmetingen in pixels voor de x- en de y-richting. Ze moeten binnen de grenzen van de printer vallen.

INFO

syntax: Info "Device"
werking: Geeft informatie weer over alle file systems (RAD:, DF0:, enzovoorts).
nieuw in 1.3: De instructie Info past zich aan bij lange namen.
Met de naam 'device' kunt u informatie opvragen van een met name genoemd onderdeel (bijvoorbeeld: Info RAD:).

INITPRINTER

syntax: Initprinter
werking: Laat het printerdevice geforceerd de instellingen lezen van Preferences.
nieuw in 1.3: Nieuwe instructie.

INSTALL

syntax: Install "Drive/A, Noboot/S, Check/S"
werking: Installeert het bootblock van een geformatteerde diskette.
nieuw in 1.3: De instructie Install wist het geheugen voor het aanmaken van een bootblock.

De optie Noboot maakt het onmogelijk om met de diskette te booten, hoewel de het een DOS-diskette wordt.

Met de optie Check kunt u controleren of de bootcode in orde is. Dat helpt in de strijd tegen virussen.

JOIN

syntax: Join " ,,,,,,,,,,,,,,To = As/K"
werking: Voegt maximaal 15 bestanden samen tot een nieuw.
nieuw in 1.3: 'To' is nu een synoniem van 'As'.

LIST

syntax: List "Dir, P = Pat/K, Keys/S, Dates/S, Nodates/S, To/K, S/K, Since/K, Upto/K, Quick/S, Block/S, Files/S, Dirs/S, Lformat/S"
werking: Voer een lijst uit met informatie over de directory en de bestanden.
nieuw in 1.3: List vermeldt de vier nieuwe protection bits:
H (wordt nog niet gebruikt)
S (script = tekstfile)
P (pure)
A (archive)
(Zie ook de beschrijving van de instructie Protect.)

List werkt nu ook met een zoekpatroon (bijvoorbeeld: List c:f#?).

List quick blokkeert spaties na de filenaam.

Met de optie Block geeft List de grootte van de files weer in blokken in plaats van in bytes.

De optie Nohead onderdrukt de weergave van de header-informatie.

Met de optie Files krijgt u alleen de files, met Dirs alleen de (sub)directories.

Gebruik de optie Lformat om de uitvoer van de instructie List snel geschikt te maken voor script(tekst)files. Deze optie schakelt automatisch de opties Quick en Nohead in. Definieer een string wanneer u Lformat gebruikt (bijvoorbeeld: Lformat = "uitvoer formaat specificatie"). List past deze specificatie toe tijdens het printen.

Het pad en de filenaam mogen deel uitmaken van de string: voeg daartoe %S toe aan de string.

Met één %S voert List de filenaam uit. Bij twee %S-codes in de string voert List de filenaam tweemaal uit. Na driemaal %S worden het pad, de filenaam en nog een keer de filenaam uitgevoerd. Vier %S-codes resulteren in: pad, filenaam, pad en filenaam.

Voorbeelden:

```
List >ram:antiwis #? lformat="protect %S -d"
```

(Maakt een scriptfile, die alle bestanden in de directory beveiligd tegen wissen).

```
List >ram:backup #? lformat="copy %S ram:%S.BAK"
```

(Maakt een backup in de RAM-disk van alle bestanden in de actuele directory).

LOADWB

syntax: Loadwb
werking: Start de Workbench.
nieuw in 1.3: Deze instructie wordt normaal gesproken uitgevoerd bij het opstarten van

het systeem. Zet daartoe Loadwb in de Startup-Sequence.

De vermelde paden gaan automatisch mee naar elke CLI (of Shell) als die vanuit de Workbench worden geactiveerd.

LOCK

syntax: Lock "Drive/A, On/S, Off/S, Code"
werking: Zet de schrijfbeveiliging van een harde schijf onder FFS aan of uit.
nieuw in 1.3: Nieuwe instructie. Hiermee beveiligt u de harde schijf (of een partitie die werkt onder FFS) tegen beschrijven. U mag een als wachtwoord een code van vier letters meegeven als u de harddisk beveiligt. In dat geval moet u diezelfde code vermelden bij opheffing van de beveiliging (optie Off).

MORE

syntax: More <filenaam >
werking: Geeft ASCII-bestanden weer op het scherm.
nieuw in 1.3: Nieuwe instructie. U kunt More activeren vanuit de Workbench (klik op de tekstfile, dan Shift-dubbelklik op de icon van More), maar ook in de CLI met de hierboven vermelde syntax. Het helpmenu van More verschijnt als u de H-toets aanslaat.

Met More kunt u bestanden bekijken, maar ook devices zoals PIPE:. Voorbeelden:

- bekijk een bestand:

```
More a.doc
```

- bekijk de inhoud van PIPE: in hex-formaat:

```
run type > pipe:a file hex  
more <pipe:a
```

MOUNTLIST

syntax: geen
werking: MountList bevat de beschrijvingen van de aangesloten devices (apparaten, RAM-disks, enzovoorts).
nieuw in 1.3: (Hoewel MountList eigenlijk geen CLI-instructie is, volgt hier toch een beschrijving.)

Elke definitie begint met de naam van het device (gevolgd door een dubbele punt) en eindigt met het #-teken op een aparte regel. Commentaar kunt u invoegen op de manier van de taal C (begin met /* en eindig met */).

Plaats de keywords (eigenschappen) op een eigen regel of scheid ze door een puntkomma. Een

afwezig keyword krijgt de default. Dat kan een waarde zijn die u slecht past.
De volgende keywords zijn toegestaan:

Handler	specificeert een handler
FileSystem	specificeert een filesystem
Device	specificeert een device
Priority	prioriteit van het proces (5 is een goede waarde voor handlers, 10 voor filesystems)
Unit	unit-nummer van het device
Flags	vlaggen voor OpenDevice (normaal 0)
Surfaces	aantal kanten (diskette: 2)
BlocksPerTrack	aantal blokken per spoor
Reserved	aantal blokken voor het bootblock (altijd 2)
PreAlloc	niet in gebruik
Interleave	interleavefactor, varieert per device.
LowCyl	nummer van de startcilinder
HighCyl	nummer van de eindcilinder
Stacksize	hoeveelheid te reserveren stackruimte
Buffers	aantal initiële cache-buffers
BufMemType	soort geheugen (0,1 willekeurig; 2,3 Chip; 4,5 Fast)
Mount	indien bevestigend, wordt het device of de handler meteen geladen, in plaats van te wachten op een aanvraag.
MaxTransfer	maximaal aantal te verplaatsen blokken (Fast File System).
Mask	adresmasker, specificeert een geheugenblok dat door DMA kan worden gebruikt bij FFS.
GlobVec	specificeert een global vector (GV) voor het proces -1: geen global vector (C- en assembler-programma's) 0: plaatst een privé-GV. Als dit keyword niet is vermeld, geldt de 'shared global vector'.
Startup	een BPTR naar een BSTR wordt bij opstarten doorgegeven aan device/file-system/handler.
BootPri	definieert de prioriteit van het zelfstartende device. De waarde varieert van -127 tot 129 (volgens afspraak betekent '-127': niet bootable).
DosType	deze waarde definieert het file system. Voor FFS is de waarde: #X444F5301. Anders is de waarde #X444F5300 (of er wordt niets vermeld). In de toekomst komen er wellicht meer DosTypes; de twee genoemde gelden alleen voor de driver van de ST506.

NEWCLI

syntax:	Newcli "Vensterspecificatie, From (van filenaam)"
werking:	Start een nieuwe CLI met een eigen venster.
nieuw in 1.3:	NewCLI heeft een default startup-bestand: 'S:CLI-Startup'. In de optie From wordt dit bestand gebruikt.

NEWSHELL

syntax: NewShell "Vensterspecificatie, From (van filenaam)"
werking: Start een nieuwe interactieve Shell met een NewCon venster.
nieuw in 1.3: Nieuwe instructie. Start een interactieve shell die gebruik maakt van de NEWCON:-handler.

De Newcon:-handler moet zijn geïnstalleerd met Mount en het Shell-Seg moet resident zijn. (Doe dat in de Startup-Sequence).

Newshell gebruikt 'S:Shell-Startup' als startup-bestand. In de optie From wordt dit bestand gebruikt.

Als het Shell-Seg niet resident is, wordt een gewone CLI gestart.

Als de Newcon:-handler niet is geïnstalleerd met Mount, wordt een Con:-venster gebruikt.

NOFASTMEM

syntax: NoFastMem
werking: Zet het FAST-geheugen tijdelijk buiten werking.
nieuw in 1.3: Sommige programma's gebruiken het snelle geheugen op een verkeerde manier. Maak de instructie ongedaan met Ctrl-C of invoer van BREAK. Vanuit de Workbench kunt u de instructie uitschakelen door de icon nog een keer te activeren.

PREFERENCES

syntax: Preferences
werking: Wijzigt de standaardinstellingen van scherm en printer of geeft die weer.
nieuw in 1.3: Met deze instructie kunt u rechtstreeks de datum en de tijd van de real-time clock in de A500/A2000 definiëren. Bovendien zijn er meer mogelijkheden om de printer in te stellen voor weergave van grafiek.

PROMPT

syntax: Prompt "Prompt"
werking: Verandert de prompt-string van de CLI of de Shell.
nieuw in 1.3: Zet bijvoorbeeld de actuele directory als prompt op het scherm (in combinatie met de Shell).

PROTECT

syntax: Protect "File/A, Flags, Add/S, Sub/S"
werking: Wijzigt de Protection bits van een bestand.
nieuw in 1.3: Met Protect is het mogelijk de vier nieuwe Protection bits Hidden (H), Script (S), Pure (P) en Archive (A) te wijzigen. Met het instructie kunt u Protection

bits optellen of aftrekken van een file in plaats van ze alleen te zetten. Dit gebeurt met de opties Add en Sub. Ook de + en - vlaggen worden hiervoor gebruikt.

Met List worden de Protection bits weergegeven. Voorbeelden:

```
Protect c:ED +rw
Protect l:filenaam -h
```

REMRAD

syntax: Remrad
werking: Verwijdert het resetvaste ramdrive.device (rad:).
nieuw in 1.3: Nieuwe instructie. Omdat bleek dat rad: niet op de gewone manier kan worden verwijderd, is deze instructie ontwikkeld. Als u deze opdracht geeft, worden eerst alle files uit rad: verwijderd; daarna wordt hij zo klein mogelijk gemaakt. Bij de volgende reset verdwijnt het ramdrive.device definitief.

RESIDENT

syntax: Resident "Naam, File, Remove/S, Add/S, Replace/S,
Pure/S"
werking: Voegt residente instructies toe of verwijdert ze.
nieuw in 1.3: Nieuwe instructie. Vaak gebruikte instructies kunt u op voorraad houden in het geheugen van de computer (anders moeten ze steeds worden geladen van schijf). Dat levert tijdwinst op. Resident werkt alleen vanuit de Shell-Seg CLI.

Instructies mogen alleen resident worden als ze zowel opnieuw kunnen worden aangeroepen als opnieuw kunnen worden uitgevoerd. Zulke instructies hebben een gezette (P). De meeste instructies in de directory C: voldoen aan deze eis. U mag proberen van een instructie met een gewisse Pure-bit die bit te zetten met de optie Pure. Als dat niet lukt, merkt u dat zodra u die instructie nog een keer oproept (GURU).

Zonder opties levert de instructie Resident op het scherm een lijst met residente instructies.

RUN

syntax: Run <instructie >
werking: Start een proces op de achtergrond.
nieuw in 1.3: Het is nu mogelijk een taak op de achtergrond te starten en daarna het CLI-venster te sluiten van waaruit de taak is gestart. Zorg wel voor omleiding van de uitvoer van die taak.

Ter verhoging van de snelheid komen eerst de residente instructies aan de beurt; daarna die uit het pad.

SEARCH

syntax: Search "From, Search/A, All/S, Nonum/S, Quiet/S, Quick/S, File/S"
werking: Zoekt een gegeven tekststring in de vermelde bestanden.
nieuw in 1.3: Bij aantreffen van de tekst, is het antwoord 0, anders 5 (waarschuwing). Search stopt door Ctrl-C.

In de optie Nonum worden regelnummers onderdrukt.
De optie Quick zorgt voor een compactere weergave.
De optie Quiet onderdrukt de mededelingen (Search levert dan alleen 0 of 5).
Met de optie File kunt u zoeken naar een bestand met een gegeven naam in plaats van naar een naam in een bestand.

SETALERT

syntax: Setalert
werking: Een patch in versie 1.2 en 1.3 van Kickstart om een aantal routines te verbeteren.
nieuw in 1.3: Nieuwe instructie. Neem die op in de Startup-Sequence. U mag deze instructie maar eenmaal geven.

SETENV

syntax: Setenv "Naam/A, String"
werking: definieert een omgevingsvariabele.
nieuw in 1.3: Nieuwe instructie. Hiermee definieert of wist u de waarde van een omgevingsvariabele. De waarden worden opgeslagen in de ENV:-handler. Dat is nu nog de RAM-disk (of de harddisk). In de toekomst wordt dit een echte handler. Verwijder een waarde met de opdracht 'SetEnv Naam'.

SHELL.INFO

syntax: Geen
werking: Kent vanuit de Workbench een Newcon:-handler toe aan een Shell.
nieuw in 1.3: Shell.info is een project-icon die precies lijkt op de CLI-icon. U activeert hiermee een CLI met een Newcon:-venster. Wijzigen van een instructieregel is mogelijk en u kunt de vorige opdrachten kopiëren. (Zie bij NEWCON:-handler.) Met het Window-tooltype definieert u plaats en grootte van het venster. Kopieën van deze icon met andere eigenschappen activeren CLI's met verschillende grootten.

SHELL-SEG

syntax: Geen
werking: Verzorgt een uitgebreide CLI omgeving.
nieuw in 1.3: De Shell-Seg wordt niet direct uitgevoerd, maar resident opgestart via de CLI. Wanneer de Shell-Seg actief is, worden alle CLI's Shell-CLI's in plaats van normale CLI's. Normaal gesproken wordt de Shell-Seg aangeroepen in de startup-sequence.

Een Shell-CLI heeft meer mogelijkheden dan een gewone CLI:

- residente instructies (direct oproepbaar uit het geheugen).
- ALIAS

In een Shell mag u instructies een andere naam geven :

```
Alias <nieuwe naam> <oude naam>
```

Wanneer de Shell een Alias tegenkomt, wordt de oude naam op de plaats van de nieuwe gezet de rest van de opdrachtregel toegevoegd. Daarna wordt de opdracht uitgevoerd. Gebruik rechte haken om iets toe te voegen aan een alias. Voorbeeld:

```
Alias xcopy copy [] clone
```

De instructie Alias zonder meer, levert een lijst van alle vervangingen. Verwijder een Alias door vermelding van zijn naam:

```
Alias <nieuwe naam>
```

Elke Shell mag een eigen set vervangingen te hebben. Het is gebruikelijk alle vervangingen op te slaan in het bestand 's:Shell-Startup'. Dat wordt automatisch uitgevoerd zodra u een Shell start.

- De Shell vermeldt de directory in de prompt. Definieer zo'n prompt met de instructie Prompt. Voorbeelden:

```
Prompt "%N> " (alleen het CLI-nummer)
```

```
Prompt "%N.%S> " (CLI-nummer en actuele directory, gescheiden door een punt)
```

```
Prompt "%S.%N> " (actuele directory en CLI nummer, gescheiden door punt)
```

U mag de prompt ook definiëren in het bestand 's:Shell-startup'.

- De Shell ondersteunt de script(tekst)bit. Als deze bit is gezet bij een bestand, dan geldt dat bestand voor Execute als een tekstbestand. Dit heeft tot gevolg dat u alleen maar de naam van dat bestand hoeft in te voeren om de inhoud ervan te laten afhandelen door de instructie Execute. (Een script file is een ASCII-bestand met een reeks AmigaDOS-opdrachten.)

WHICH

syntax: Which "File/A"
werking: Zoekt naar de locatie van een bestand.
nieuw in 1.3: Nieuwe instructie. Zoekt een bestand binnen het vermelde pad.

XICON

syntax: Geen
werking: Activeert een script file vanaf de Workbench.
nieuw in 1.3: Nieuwe optie. Maak een script file, zet de script bit met de instructie Protect en koppel een icon aan dat bestand. Definieer de venstergrootte (WINDOW =) met de optie Info van de Workbench. Wijzig de default tool in 'c:Xicon'. Sla het veranderde .info-bestand op.

Een dubbele klik op de icon activeert de script file.

9.2 Nieuwe handlers en devices

AUX:-handler

syntax: Aux:
werking: Maakt ongebufferde seriële invoer en uitvoer mogelijk.
nieuw in 1.3: Nieuwe handler, die lijkt op de SER:-handler. Er zijn twee verschillen: de Aux:-handler moet eerst worden geïnstalleerd met de instructie Mount en de transmissie via AUX: gebruikt geen buffer. Voorbeeld:

```
Copy aux: to ram:testfile
```

ENV:-handler

syntax: Env:
werking: Slaat omgevingsvariabelen op.
nieuw in 1.3: Nieuwe handler. Momenteel wordt hiervoor nog de RAM-disk gebruikt, maar een echte handler is in de maak.

NEWCON:-handler

syntax: Newcon:
werking: Vervangt de CON:-handler.
nieuw in 1.3: Nieuwe handler die, vergeleken met CON:, de volgende extra mogelijkheden heeft:
Wijzigen in de opdrachtregel
De pijltoetsen naar links en rechts worden gebruikt om de cursor over de

instructieregel te bewegen. Een teken onder de cursor kunt u wijzigen. Shift-pijl-links of Ctrl-A zet de cursor op het begin van de regel; Shift-pijl-rechts of Ctrl-Z aan het einde van de regel.

Backspace en Del werken normaal.

Ctrl-X wist een hele regel, Ctrl-U wist de regel vanaf het begin tot aan de cursor en Ctrl-K vanaf de cursor tot aan het einde van de regel. Ctrl-W verplaatst de cursor naar de volgende tabulatorstop.

Regelgeheugen

Newcon: heeft een rondgaand geheugen van 2 Kb. Eerdere opdrachten kunt u terughalen met de toets pijl-omhoog; pijl-omlaag brengt u lager in de geheugenbuffer. Shift-pijl-omhoog of Ctrl-R zoekt in de buffer naar een string die klopt met een gedeelte wat u hebt ingevoerd. Shift-pijl-omlaag of Ctrl-B verplaatst u naar het einde van de buffer.

Control-codes zijn niet zichtbaar in Newcon:. In plaats daarvan ziet u een spatie. Ze staan er echter wel en u mag ze normaal gebruiken.

PIPE:-handler

syntax: Pipe: [Naam]

werking: Legt een communicatiekanaal aan tussen programma's.

nieuw in 1.3: Nieuwe handler. Hij wordt op dezelfde manier gebruikt als PAR:, SER: en PRT:. Pipe: moet u tevoren aanroepen met de instructie Mount. Bovendien moet de pipe zijn gedefinieerd in de MountList. Vermeld daar de volgende gegevens:

```
PIPE:           Handler = 1:Pipe-handler
                Stacksize = 5000
                Priority = 5
                GlobVec = -1
#
```

De Pipe:-handler is eigenlijk een I/O-mechanisme tussen programma's onderling. De Pipe: kan 4 Kb aan data opnemen voor het schrijfproces wordt geblokkeerd. Een ander proces kan deze data inlezen. De buffer heeft een zodanige structuur dat de data meteen bruikbaar zijn voor het lezende proces. Voorbeelden:

```
CLI venster 1  Copy hugefile pipe:a
CLI venster 2  Copy pipe:a pipe:b
CLI venster 3  Copy pipe:b pipe:c
CLI venster 4  Copy pipe:c pipe:d
CLI venster 5  Woordtelling pipe:d
```

RAMDRIVE.DEVICE

syntax: Rad:
werking: Resetbestendige RAM-disk
nieuw in 1.3: Nieuw device. RAD: is een RAM-disk die de data vasthoudt na een reset. Bovendien is het met Kickstart 1.3 mogelijk vanaf deze RAM-disk te booten zonder gebruik te maken van een Workbench-diskette.

Installeer het Ramdrive.device voor gebruik met de instructie Mount. Formateer RAD: voor u hem de eerste keer gebruikt. De grootte van RAD: kunt u definiëren met de parameters LowCyl en HighCyl in de MountList.

U kunt het ramdrive.device ook gebruiken onder FFS. Booten vanaf RAD: is dan niet mogelijk. Geef BootPri in de MountList de waarde -127.

Verwijder RAD: met de instructie RemRad.

Als er na een reset een aantal bestanden zijn verdwenen uit RAD:, kunt u proberen ze terug te halen met behulp van DiskDoctor.

SPEAK-HANDLER

syntax: Speak: OPT/K
werking: Uitvoer van spraak.
nieuw in 1.3: Nieuwe handler. Geeft uitvoer weer als spraak. Het is een handler zoals PAR:, SER: en PRT:, maar deze SPEAK: moet u eerst initialiseren met 'Mount Speak:'.

Vermeld achter het keyword 'opt' een of meer van de volgende opties van SPEAK:

P### pitch (65 < ### < 320)
S### snelheid (30 < ### < 400)
M mannelijke stem
F vrouwelijke stem
R robot stem
N natuurlijke stem
O0 niet toestaan van onderstaande opties in de invoertekst
O1 wel toestaan van onderstaande opties
A0 uitschakelen van fonetisch modus
A1 inschakelen van fonetische modus
D0 zinnen afbreken afhankelijk van interpunctie
D1 zinnen afbreken op grond van interpunctie, Return en LineFeed.

Voorbeeld:

Geef de volgende opdrachten om te luisteren naar een file:

```
1> Mount Speak:  
1> Copy s:startup-sequence to speak:  
1> Copy s:startup-sequence to speak: opt f s160
```

Voor de zekerheid willen nogmaals benadrukken dat bovenstaande gegevens zijn gebaseerd op een gamma-versie van de nieuwe Workbench 1.3. Deze bijna-definitieve versie zal maar heel weinig verschillen van de definitieve versie.

De in dit hoofdstuk besproken wijzigingen hebben niet alles overhoop gehaald, maar ze maken het werken met de Amiga zo veel gemakkelijker dat we mogen spreken van een flinke stap voorwaarts.

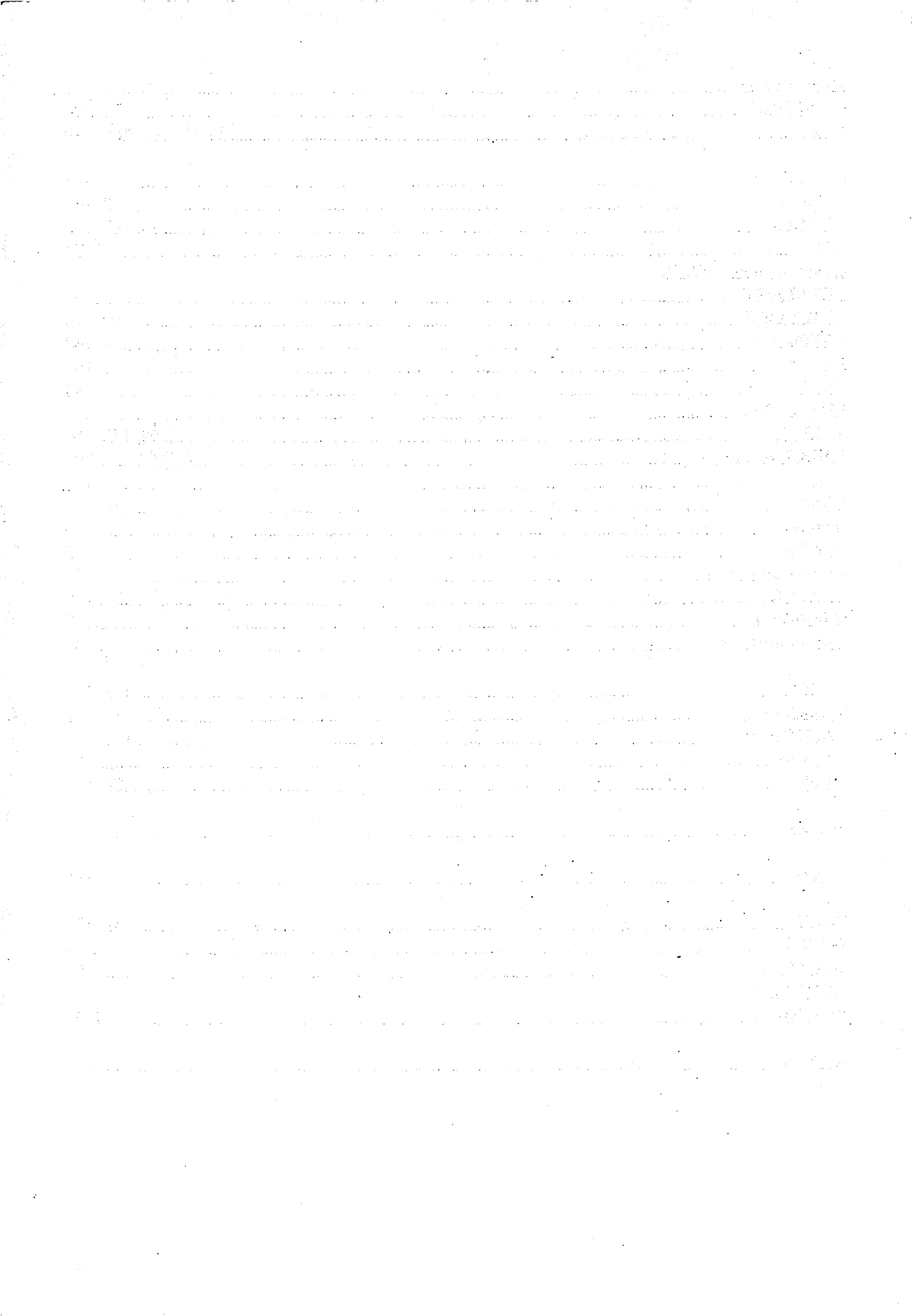
Trefwoordenlijst

.info	12
ADDBUFFERS	38, 61, 169, 179
ALIAS	191
archive	188
ASCII-codes	72
ASK	47, 179
ASSIGN	37, 169, 179
AUX:	192
AVAIL	179
BASIC-lader	123
batchbestanden	42, 77
- oproepen	78
batchverwerking	77
BINDDRIVERS	41, 171
boomstructuur	12
bootable	27
BPTR-pointer	101
BREAK	36, 59, 169
C	104, 110
CD	18, 165
CD /	19
Centronics	61
CHANGETASKPRI	35, 169
ChangeTaskPri	97
CLI	9, 16, 57, 90, 95, 179
-instructies	16, 109, 164, 173, 179
-venster	59
-wekker	72
CLOCK	180
CLOCKPTR	180
CMD	180
Command Line Interface, zie CLI	
CON	62, 70
console-device, zie CON	
COPY	20, 67, 165, 181
Ctrl-C	36, 59
Ctrl-D	59
Ctrl-F	59
Ctrl-\	64, 71
DATE	39, 170, 181
DELAY	119, 150

DELETE	19, 165
devices	60
DF	60
DIR	17, 23, 164, 182
DISKCHANGE	33, 168
DISKCOPY	25, 61, 166
DISKDOCTOR	32, 168, 182
DOS	9, 89
-bibliotheek	104
ECHO	44, 171, 182
ED	49, 160
EDIT	49, 160
ENDCLI	15, 34, 168
ENV:	192
EQ	45
ERROR	46
EXECUTE	42, 171, 182
FAIL	46
FAILAT	44, 171
FAST FILING SYSTEM	177
FASTMEMFIRST	183
FAULT	39, 170
FF	183
FILENOTE	31, 167
FIXFONTS	183
FORMAT	16, 61, 164, 183
formatteren	16
foutnummer	44
GETENV	183
GRAPHICDUMP	189
help-functie	14
hidden	188
IF/ELSE/ENDIF	45, 171
INFO	26, 166, 184
INITPRINTER	184
INSTALL	27, 61, 166, 184
invoer	57
JOIN	29, 167, 185
jokers	19, 30, 57
kopiëren	66

LAB	48, 172
LIST	23, 165, 185
LOADWB	185
LOCK	186
longword	28
MAKEDIR	19, 165
MORE	186
MOUNT	41, 171
mountlist	178, 186
MULTIPLE	181
multitasking	33, 72, 89
NEWCLI	33, 92, 168, 187
NEWCON:	192
NEWSHELL	188
NOFASTMEM	188
NOT	46
NOTIFY	181
omleiding uitvoer	28, 68
opdracht onderbreken	59
PAR	61
parallele device, zie PAR	
partitie	177
PATH	36, 169
PIPE: 193	
PREFERENCES	188
printen	67, 84
printer-device, zie PRT	
PROMPT	40, 170, 188, 191
PROTECT	24, 30, 167, 188
protection bits	188
PRT	62
pure	188
Quick reference	160
QUIT	45, 171
RAM-disk	60
RAMDRIVE.DEVICE	194
RAW	64
regelgeheugen	193
RELABEL	26, 61, 166
REMRAD	189
RENAME	22, 165
RESIDENT	189

Return-toets	11
ROM-Wack	74
RUN	34, 93, 104, 168, 189
screendump	184
script file	182, 188, 192
SEARCH	29, 167, 190
SER	62, 73
serieel device, zie SER	
SETALERT	190
SETCLOCK	40, 170
SETDATE	31, 168
SETENV	190
SHELL.INFO	190
SHELL-SEG	191
SKIP	48, 172, 180
SPEAK-handler	194
stack	180
SORT	30, 167
sorteren	30
STACK	41, 170
startup-sequence	79
STATUS	35, 169
stuurtekens	69
Systeeminstructies	33
TASK	113, 124
TASKPRI	115, 131
TASKSTOP	118, 140
tekstverwerking	50
TYPE	28, 166
uitvoer	57
virussen	184
WAIT	49, 172
WARN	46
WHICH	192
WHY 39, 170	
Workbench	9, 90
XICON	192



De Intuition van de Commodore Amiga maakt het de gebruiker zo gemakkelijk als maar mogelijk is. Wanneer u echter méér uit uw Amiga wilt halen, dan kunt u niet om de Command Line Interpreter heen. Via deze CLI kunt u rechtstreeks met Amiga-DOS werken. Pas dan ontdekt u de echte mogelijkheden van uw Amiga. Zo kunt u een diskette formatteren of een file printen, terwijl u met een muziekprogramma werkt of de directory van uw RAM-disk bekijkt. Dit boek helpt u de vele mogelijkheden te leren kennen.

Een greep uit de inhoud:

- uitgebreide beschrijving van alle CLI-commando's (ook Workbench-versie 1.3)
- heldere uitleg van de multitasking-mogelijkheden
- veel tips en trucs met betrekking tot Amiga-DOS
- opzet van batch-files en de mogelijkheden hiervan.

Een boek dat de echte Amiga-gebruiker niet kan en mag missen!