Installatie van DOS65

Overtuig u ervan dat de hardware in orde is. Als de hardware aan de eisen voldoet, zoals beschreven in de hardware beschrijving, dan is de installatie vrij simpel. De meegeleverde Eprom's worden in de sockets gestoken. De diskette is enkelzijdig geformatteerd voor 40 of voor 80 tracks (afhankelijk van uw systeem). Nadat het systeem is aangezet wordt de diskette in drive 0 gestoken. Als er een reset wordt gegeven komt het systeem in I/O65. Nu moet de bootstrap geactiveerd worden door een B (b) in te typen. De DOS wordt nu opgestart. Hieronder volgen een aantal procedures die gevolgd moeten worden voor de verschillende drive types en of men over 1 of 2 drives beschikt. Als u gelijk al met de handen in het haar zit en niet direkt het juiste commando in de manual kunt vinden gebruik dan de HELP utility. Typ het commando HELP in, eventueel gevolgd door het commando waarover u informatie wilt. Ook alleen HELP kan. Dan verschijnt op het beeldscherm een overzicht van alle geldige commando's. Op dat moment kunt u een keuze maken uit die lijst. Door alleen een <return> in te typen komt men weer terug op DOS command niveau.

Algemeen

- Copieer de systeemdiskette eerst naar een nieuwe diskette. Denk aan het BOOTLINK commando bij het copieren van systeemschijven. Ook mag een systeemschijf niet Extra Density geformatteerd zijn. Typ eventuuel in op DOS command niveau, dat is na de $ prompt : HELP COPY <return>
- Start het systeem op vanaf de zojuist gecopieerde diskette en roep het configuratie programma op. Met behulp van dat programma wordt een nieuwe DOS geactiveerd als u snelle disk drives hebt. Ook is dit programma voor het installeren van programmatuur voor de real time clock kaart van Elektuur op adres $E120. Het configuratie programma wordt opgestart met CONFIG. Beantwoord de vragen over de drive steprate, de frequentie van het kristal op de real time clock kaart en of u een 6545 dan wel een 6845 crt controller op de VDU kaart hebt. Als alle vragen beantwoordt zijn wordt een nieuwe DOS gemaakt en worden de juiste real time clock programma's geactiveerd. Geef een reset en start het systeem opnieuw op.
- Het maakt niet uit of u een 40 of 80 tracks systeem hebt. Alleen de opslag capaciteit is anders. Mensen die een 80 tracks systeem hebben kunnen eventueel wel 40 tracks diskettes aanmaken, schrijven, lezen, dus gebruiken. Omgekeerd is dat niet mogelijk. De DOS ziet zelf of er een 40 of 80 tracks diskette in de drive zit. In geval van 40 tracks worden er twee stapjes gegeven om naar een volgende track te komen. Men hoeft daar dus geen hardare trucs voor uit te halen. Als er op een 80 tracks systeem een 40 tracks diskette wordt aangemaakt kunnen er echter wel eens problemen ontstaan op dat 40 tracks systeem. Deze gebruiker kan leesproblemen hebben. Die ontstaan doordat een 40 tracks systeem de informatie 'harder' op schijf zet en dus ook een 'harder' signaal terug wenst te lezen. Meestal echter gaat het wel goed. We wilden deze mogelijkheid voor de gebruikers open houden.
- Lees verder bij een van de volgende punten.

1 drive, enkelzijdig.

- Het systeem is al klaar voor de gebruiker. Er moet echter rekening worden gehouden met de volgende zaken:
- Formatteer de schijven die u gebruikt alleen enkelzijdig. Double of extra density kan wel als uw schijven dat toelaten.
- Gebruik bij diverse utilities de optie dat er gewacht moet worden na aanroep van de utility, dit om u de gelegenheid te geven schijven te kunnen wisselen.

2 drives, enkelzijdig.

- Geef voordat u begint met een sessie het commando :
  ASN U=1:
  Vanaf dat moment is de systeemschijf drive 0 en de gebruikerschijf drive 1. Het is aan te raden dit commando in de LOGIN.COM file te zetten.
  Let echter op het volgende :
- Formatteer de schijven die u gebruikt alleen enkelzijdig. Double of extra density kan wel als uw schijven dat toelaten.

1 drive, dubbelzijdig.

- Het systeem is al klaar voor de gebruiker. Er moet echter rekening worden gehouden met de volgende zaken:
- Gebruik bij diverse utilities de optie dat er gewacht moet worden na aanroep van de utility, dit om u de gelegenheid te geven schijven te kunnen wisselen.

2 drives, dubbelzijdig.

- Geef voordat u begint met een sessie het commando :
  ASN U=1:
  Vanaf dat moment is de systeemschijf drive 0 en de gebruikerschijf drive 1. Het is aan te raden dit commando in de LOGIN.COM file te zetten.
- Formatteer nieuwe schijven zoals u zelf wilt. Als een schijf naar een gebruiker moet die slechts enkelzijdige drives heeft, dan moet daar met het formatteren van die schijf rekening mee gehouden worden.

INHOUD

Inhoudsopgave..... pag. 1

HOOFDSTUK 1 : Specificaties en eigenschappen DOS65 ..... pag. 4
Hard- en software eisen ..... pag. 4

HOOFDSTUK 2 : DOS65 - Termen, definities, en andere uitleg ... pag. 5
Default ..... pag. 5
ESCAPE en CONTROL ..... pag. 5
Toggle ..... pag. 5
Controlcodes ..... pag. 5
File ..... pag. 5
Achtervoegsels (extensions) ..... pag. 5
Opties en parameters ..... pag. 6
DOS65 en disk-drives ..... pag. 6
Input/Output-redirect, indirekte files ..... pag. 7
Drive- en directory specificatie ..... pag. 11
Foutmeldingen DOS65 ..... pag. 12

HOOFDSTUK 3 : DOS65 Permanente kommando's ..... pag. 13
< ..... pag. 13
<T ..... pag. 13
> ..... pag. 13
>> ..... pag. 14
>T ..... pag. 14
>P ..... pag. 14
<> ..... pag. 14
@ ..... pag. 14
CHMODE ..... pag. 15
DIR ..... pag. 15
GO ..... pag. 15
LCMD ..... pag. 15
LOAD ..... pag. 16
RUN ..... pag. 16
SAVE ..... pag. 16
SEE ..... pag. 17
STAT ..... pag. 17
SYNC ..... pag. 17

HOOFDSTUK 4 : DOS65 Utilities ..... pag. 18
>D ..... pag. 18
APPEND ..... pag. 18
AS ..... pag. 18
ASN ..... pag. 18
BOOTLINK ..... pag. 19
CAT ..... pag. 19
CLEAR ..... pag. 19
COPY ..... pag. 19
CREATE ..... pag. 20
DDOCTOR ..... pag. 20
DELETE ..... pag. 21
DPTIME ..... pag. 21
DUMP ..... pag. 21
ECHO ..... pag. 22
EDITOR ..... pag. 22
FORMAT ..... pag. 22

HELP ..... pag. 24
LIST ..... pag. 24
MAP ..... pag. 25
MEMFILL ..... pag. 25
MEMMOVE ..... pag. 25
MONITOR ..... pag. 25
MTOAS ..... pag. 26
OLOAD ..... pag. 26
PLIST ..... pag. 26
PRINT ..... pag. 27
RENAME ..... pag. 28
RS232 ..... pag. 28
SDIR ..... pag. 28
SETBEGIN ..... pag. 29
SETDRIVES ..... pag. 29
SETMODE ..... pag. 29
SETPROMPT ..... pag. 30
SETRTC ..... pag. 30
SETUP ..... pag. 30
TIME ..... pag. 30
TYPE ..... pag. 30
UNEXPAND ..... pag. 31
VFORMAT ..... pag. 31

HOOFDSTUK 5 : Filenamen en speciale karakters ..... pag. 32

DOS65 is een universeel disk operating system voor 65(C)02 computers.

DOS65 is ontwikkeld door: A.B.M. Brouwer
Nieuwkoopseweg 44
2641 PB Pijnacker

Deze handleiding is geschreven door:
A.B.M. Brouwer

en

A.S. Hankel
Willem Kloosstraat 32
7606 BB Almelo
tel. 05490-51151

Voor DOS65, voor de voor DOS65 ontwikkelde utilities, alsmede voor de bij DOS65 behorende handleidingen geldt:

Copyright: KIM Gebruikersclub Nederland 1986.

HOOFDSTUK 1 : Specificaties en eigenschappen DOS65 2.01:

- Geschikt voor:
  * 5 1/4 inch disk-drives (40/80 tracks, single/double sided, single/ double density).
  * hard disk.
  * virtual disk.
- Maximale geformatteerde schijf-kapaciteit: 719 Kbyte.
  (5 1/4 floppy disk, 80 tracks, double sided, extra density)
- Ondersteunt: FULL SCREEN EDITOR (ED)
  ASSEMBLER
  BASIC
- Uitgebreide input/output redirect mogelijkheden.
- Zeer gebruikersvriendelijk.
- Aan te passen voor vrijwel elk 6502 systeem.
- Zeer efficient schijfruimte gebruik.
- Volledige dokumentatie beschikbaar.
- Overdrachtsnelheid ca. 7 Kbyte/seconde.
- Eenvoudige hardware.

Hard- en software eisen:

- DOS65 floppy-controller kaart. De print hiervoor is leverbaar via de KIM Gebruikersclub Nederland.
- Minimaal 56 Kbyte RAM geheugen.
- Terminal eisen: 80 koloms, cursorbesturing.
  (Elekterminal is dus niet bruikbaar.)
- HAVISOFT IO-programma I065 2.01 voor implementatie op de ELEKTUUR CPU/VDU konfiguratie.

HOOFDSTUK 2 : DOS65 - Termen, definities, en andere uitleg.

Default:

Default betekent uitgangswaarde, uitgangspositie. Bijvoorbeeld: wanneer een variabele default op 2 (twee) staat, betekent dat, dat de gebruiker wel een andere waarde op mag geven, maar er wordt uitgegaan van de waarde 2.

ESCAPE en CONTROL:

In DOS65 wordt de ESCAPE-toets gebruikt, om aan de computer duidelijk te maken, dat er na de ESCAPE een kommando volgt. Wanneer er in de manual bijvoorbeeld gesproken wordt van ESC A, wordt bedoeld, dat eerst de ESCAPE, en daarna de A wordt getypt.
Wordt er echter in de handleiding gesproken van CTL A (control A), dan wordt het GELIJKTIJDIG indrukken van zowel de control-toets als de A bedoeld.

Toggle:

Het begrip toggle houdt in, dat met een toets een zekere funktie aan en uit, of wel/niet aktief geschakeld kan worden. Wanneer met bijvoorbeeld toets A getoggled kan worden, dan houdt dat in dat 1 keer de A de funktie aanzet, nog een keer de A de funktie weer uitzet enz.

Controlcodes:

Controlcodes zijn ascii-codes, met een waarde tussen $00 en $1F. Deze controlcodes hebben meestal een besturingsfunktie voor randapparatuur, dus inklusief terminals (denk aan cursorbesturing).
Enkele controlcodes: BS (backspace, $08), LF (linefeed, $0A).

File:

Een file is, simpelweg gezegd, een hoeveelheid computergegevens, met een 'naamkaartje' eraan, die op een schijf (diskette) staat. Deze gegevens kunnen van de schijf naar het geheugen van de computer getransporteerd worden, en uiteraard ook omgekeerd, van het geheugen van de computer naar de schijf. In een file staan dus gegevens. Die gegevens kunnen van allerlei vorm zijn. Bijvoorbeeld een BASIC-programma, een FORTH-screen, ASCII-gegevens, maar ook machinetaalprogramma's en zelfs groepen kommando's. Aan een file 'hangt' altijd een naamkaartje, beter gezegd een filenaam. Elke file, die op een schijf staat, heeft dus zijn eigen naam. Met behulp van speciale kommando's, die in deze handleiding ter sprake komen, kunnen we zien welke files er op een schijf staan, wat voor type ze zijn, hoe lang ze zijn (dus: hoeveel gegevens ze bevatten) en nog heel wat meer.

Achtervoegsels (extensions):

Elke file heeft een eigen, unieke naam. Daarnaast zou het gemakkelijk zijn, om aan de naam van de file te kunnen zien wat voor soort file het is. Daartoe bestaat het begrip achtervoegsel, of extension. Dit houdt in dat de komplete naam bestaat uit de eigenlijke naam, met daarachter een soort indikator. Op een schijf staat dus niet alleen de file met de naam NAAM, maar de naam NAAM.XXX ; .XXX staat hierbij voor het achtervoegsel. Voorgesteld wordt, de volgende achtervoegsels te gebruiken:
.TXT Assembler source file (Moser, Micro-Ade).

.MAC Assembler source file (AS).
.DOC Dokumentatie file (correspondentie, teksten enz.)
.LST List file (AS, Moser, Micro-Ade, BASIC enz.)
.BIN Binary file (code).
Een file, die bijvoorbeeld een brief is, zou op schijf gezet kunnen worden met de naam BRIEF.DOC ; het gedeelte voor de punt is het aanduidende gedeelte, het gedeelte achter de punt de indikator voor het soort file.
In principe is iedere gebruiker vrij in de keuze wel of geen, en welke achtervoegsels te kiezen. Niettemin verdient het aanbeveling toch zoveel mogelijk voornoemde methode aan te houden, was het maar uit oogpunt van uniformiteit.
Wanneer er onder DOS65 een aan de DOS aangepaste versie van Micro-Soft BASIC draait, bestaan er nog een tweetal achtervoegsels, die NIET door de gebruiker te beinvloeden zijn. Dit zijn .BAS en .VAR
.BAS staat voor een BASIC programma (in BASIC formaat, dus 'tokenized', d.w.z. dat een BASIC sleutelwoord wordt afgekort tot een (1) byte.)
Wordt een BASIC programma op schijf gezet, door middel van het commando SAVE "NAAM" , dan komt op de schijf de file NAAM.BAS te staan.
.VAR staat voor door BASIC gegenereerde data, die naar schijf wordt geschreven. Een .VAR file wordt wel in ascii vorm naar schijf geschreven, en kan dan ook eventueel m.b.v. de editor gewijzigd worden.
Tenslotte zijn er ook nog file-namen zonder achtervoegsel. Dit zijn externe DOS-kommando's (utilities), of kommando files.

Opties en parameters:

De gebruiker kommuniceert met zijn computersysteem door er opdrachten aan te geven in de vorm van kommando's. Gebruikelijk is dus het intypen van de naam van het kommando gevolgd door een return, dus KOMMANDO <cr> (Het teken <cr> betekent in deze handleiding het indrukken van de RETURN toets).
DOS65 biedt de gebruiker de mogelijkheid, om aan bepaalde kommando's extra gegevens mee te geven in de vorm van opties en parameters. Meegegeven opties beinvloeden in een bepaalde mate het effect van het kommando, afhankelijk van het kommando en de beschikbare opties.
Parameters zijn nummerieke waarden, (getallen), die door het kommando als een soort begrenzing worden gezien. Bij DOS65 worden parameters bijv. gebruikt om regelnummers op te geven.
Opties worden altijd voorafgegaan door een - (het min-teken).
Parameters worden altijd voorafgegaan door een + (het plus-teken).
De vorm van intypen bij gebruik van een kommando met opties en parameters wordt dus: het kommando, een spatie, een -, de optieletter(s), een spatie, een +, eerste parameter, een komma, tweede parameter, en tenslotte een return. Bijvoorbeeld: LIST -CE +10,20 . LIST is het kommando. Met -CE geven we aan gebruik te willen maken van de opties C en E. Met +10,20 geven we als parameters resp. 10 en 20 op.

In deze handleiding wordt (bij de beschrijving van de kommando's) de mogelijkheid van opties en/of parameters aangegeven door ze tussen een '[' en een ']' te plaatsen. Bijvoorbeeld: LIST [-CELMN +m,n] betekent dat het kommando LIST als opties de C, E, L, M en de N kent, en dat het mogelijk is twee parameters 'm' en 'n' op te geven.

DOS65 en disk-drives:

DOS65 laat aan de gebruiker de keuze over van het aantal (1 of 2) en het type (40 of 80 tracks, single- of doublesided, single- of double density) drives.

aankondiging. In het normale geval verschijnt nu de prompt van BASIC, ter indikatie dat de gebruiker kan beginnen met zijn invoer. Nu er echter input redirect wordt gepleegd, worden de gegevens van schijf gehaald, uit de file FILE dus. Wanneer de hele file FILE ingelezen is, wordt de gesloten, en wordt de input redirect beeindigd.

Bij input redirect in de vorm van een kommando file luidt de syntax:
KOMMANDO <cr>. KOMMANDO is de naam van een file, waarin een aantal kommando's staan. Het is noodzakelijk dat KOMMANDO als kommando aangeroepen kan worden. Dit is alleen toegestaan, als in de filemode de 'C'-mode aanstaat (SETMODE -C, zie de beschrijving van het kommando SETMODE).
Een (overigens weinig zinvol) voorbeeld:

```
CAT
DIR
SDIR
```

Achtereenvolgens worden CAT, DIR en SDIR uitgevoerd. Daarna wordt de de kommando file weer gesloten. Kommando files kunnen tot twee niveaus diep gebruikt worden. Het is dus mogelijk om in de eerste kommando file een tweede kommando file aan te roepen. Ook is het mogelijk om aan kommando files parameters mee te geven. In dat geval luidt de syntax:
KOMMANDO [ARGUMENT-1] [ARGUMENT-2]
KOMMANDO staat hierbij voor de kommando file, ARGUMENT-1 voor het eerste argument, ARGUMENT-2 voor het tweede argument.
Om de onderstaande voorbeelden enigszins overzichtelijk en begrijpelijk te houden wordt
KOMMANDO ARGUMENT-1 [ARGUMENT-2] vertaalt naar

```
AAA      BBB         CCC
```

Voor de duidelijkheid:
AAA staat voor de kommando file
BBB staat voor ARGUMENT-1
CCC staat voor ARGUMENT-2

Belangrijke gegevens ten aanzien van indirekte files met parameters zijn:
- argumenten worden gescheiden door spaties.
- meer dan twee argumenten kunnen niet worden opgegeven.
- de totale lengte van de twee argumenten is maximaal 24 karakters.
- definitie van argumenten vindt plaats :
  * door de start van het eerste niveau, of
  * met het laatste kommando van het eerste niveau.
- essentieel in indirekte files zijn de karakters '&', en de daarop volgende karakters '1' of '2'.
  Belangrijk: het karakter '&' staat voor het opgegeven ARGUMENT. De karakters '1' en '2' duiden ARGUMENT-1 resp. ARGUMENT-2 aan !
- Kommentaar in kommando file's dient te worden voorafgegaan door het karakter ';'

Waarschuwing: Als U gaat experimenteren met de nu volgende voorbeelden, doe dat dan met schijven waarop geen belangrijke informatie staat. Maak eventueel eerst een kopie van Uw systeemschijf, en gebruik een lege user-schijf ! Een fout is snel gemaakt, en zou vervelende gevolgen kunnen hebben !!

Voorbeelden:
Er wordt een kommando file AAA gekreeerd die er als volgt uit ziet:

```
; dit is file AAA
TYPE &1
; einde file AAA
```

De eerste en derde regel zijn kommentaar, ze worden voorafgegaan door het karakter ';'. Maar wat doet nu regel 2 ? Welnu, deze kommando file AAA wordt geexecuteerd door op te geven:
AAA BBB <cr>
Dit gebeurt er nu:

```
TYPE BBB        oftewel file BBB wordt geTYPEd
```

De plaats van het karakter '&' in de kommando regel TYPE &1 wordt ingenomen door 'BBB'. De '1' duidt op het EERSTE argument. Zou in plaats van
AAA BBB <cr> zijn opgegeven:
AAA TEST <cr) dan gebeurt er:

```
TYPE TEST       oftewel file TEST wordt geTYPEd.
```

Door het opgeven van de systeem gebruiker van BBB resp. TEST wordt dus het verloop, of beter gezegd de executie van het kommando AAA bepaald. Het opgeven van BBB of TEST is dan ook de parameter-overdracht, die plaats vindt.
Dan nu een wat ingewikkelder voorbeeld. Het vorige voorbeeld had maar een (1) parameter. Het volgende voorbeeld gebruikt twee parameters:

```
; dit is file AAA
TYPE &1
TYPE &2
TYPE &1.mac t.&1.mac
TYPE &1 &2
TYPE &BBB
; einde file AAA
```

We gebruiken nu twee parameters, dus executie van de file AAA vindt plaats door:
AAA BBB CCC
Er gebeurt nu het volgende:

```
TYPE BBB
TYPE CCC
TYPE BBB.mac t.BBB.mac
TYPE BBB CCC
TYPE BBB
```

Snapt U er niets meer van ? We zetten de zaken even naast elkaar. Links het regelnummer, in het midden de desbetreffende regel van de file AAA en rechts datgeen er gebeurt:

```
1               ; dit is file AAA
2               TYPE &1                         TYPE BBB
3               TYPE &2                         TYPE CCC
4               TYPE &1.mac t.&1.mac            TYPE BBB.mac t.BBB.mac
5               TYPE &1 &2                      TYPE BBB CCC
6               TYPE &BBB                       TYPE BBB
7               ; einde file AAA
```

Regel 1: invoer: '; dit is file AAA', uitvoer: niets, dit is kommentaar, want het wordt voorafgegaan door een ';'.
Regel 2: invoer TYPE &1, uitvoer TYPE BBB, klopt, want '&1 staat voor de eerste parameter, en dat is BBB.
Regel 3: invoer TYPE &2, uitvoer TYPE CCC, klopt, want '&2 staat voor de tweede parameter, en dat is CCC.

Regel 4: invoer TYPE &1.mac t.&1.mac, uitvoer TYPE BBB.mac t.BBB.mac. In een (1) regel worden nu twee file's geTYPEd. Dat mag, want ze zijn immers gescheiden door een spatie. In deze regel gebeuren twee opvallende zaken:
- Na beide '&'en staat een '1', dit slaat dus op de eerste parameter, en die bevat immers 'BBB'.
- In het tweede gedeelte van deze regel staat 't.&1.mac'. De uitvoer is 't.BBB.Mac' De '1' in het tweede gedeelte slaat op de eerste parameter (BBB). De uitvoer bestaat dan ook uit een 'optelsom' van: (t.)+(BBB)+(.mac) tesamen t.BBB.mac.

Regel 5: invoer TYPE &1 &2, uitvoer TYPE BBB CCC, klopt, want '&1' en '&2' staan voor resp. de eerste en de tweede parameter, en dat zijn weer resp. BBB en CCC.

Regel 6: invoer TYPE &BBB, uitvoer TYPE BBB. In aansluiting op het voorgaande lijken invoer en uitvoer logies, MAAR: de invoer TYPE &BBB heeft geen '1' of '2' na het '&' karakter. Omdat '1' en '2' ontbreken, betekent dit, dat altijd BBB geTYPEd wordt, zonder dat er naar de opgeven parameters wordt gekeken !

Regel 7: invoer: '; einde file AAA', uitvoer: niets, ook dit is kommentaar, want het wordt voorafgegaan door een ';'

Dit voorbeeld ging uit van een kommando file met opgave van twee parameters, dus:
AAA BBB CCC
Wordt er slechts een (1) parameter opgegeven, dus:
AAA BBB
dan gebeurt er dit:

```
TYPE BBB
TYPE
```

Na de tweede TYPE wordt er een foutmelding gegeven. Immers, de tweede parameter ontbreekt. Aansluitend wordt de file AAA gesloten, en komt men terug op standaard DOS65 niveau.

Bij de voorgaande voorbeelden was de file AAA de file op het eerste niveau. Het is echter ook mogelijk om deze file in een andere kommando file aan te roepen. Voorbeeld:

```
; dit is file QQQ
AAA &1 &2
; einde file QQQ
```

Executie vindt nu plaats door: QQQ BBB CCC
De executie verloopt als volgt:

```
AAA BBB CCC
TYPE BBB
TYPE CCC
TYPE BBB.mac t.BBB.mac
TYPE BBB CCC
TYPE BBB
```

Tenslotte nog een praktisch voorbeeld: Wanneer we een AS source file TEST.MAC assembleren, levert dit een binaire file TEST.BIN op. Om deze file, aanwezig op de user drive, om te zetten naar een kommando op de system drive, moeten de volgende handelingen verricht worden:
- de file TEST.BIN moet als kommando worden gedefinieerd,
- kopieren van TEST.BIN (user drive) naar TEST (system drive),
- DELETEen van de TEST.BIN.

Deze handelingen kunnen in een universele kommando file 'makecom' worden gezet:

```
; file makecom
;
; make from .bin file a command, delete .bin file
;
setmode -c U:&1.bin
copy U:&1.bin S:&1
delete -y U:&1.bin
; end makecom
```

makecom TEST <cr> doet het volgende:

```
setmode -c U:TEST.bin
copy U:TEST.bin S:TEST
delete -y U:TEST.bin
```

Opmerking 1: Het aantal files, dat gelijktijdig open mag zijn, is maximaal zes ! Met een indirektie op twee niveaus, twee spooler files en een assembler die meerdere files tegelijk opent, is dit maximum snel bereikt.
Opmerking 2: Om onderscheid te maken tussen binaire kommando's en indirekte kommando's, wordt aanbevolen om de laatste als files met namen in lower case op de systeemschijf te zetten. Dus 'TEST' in plaats van 'test' en 'makecom' in plaats van 'MAKECOM'.

Output redirect:
Uitvoer van programma's gaat standaard naar het beeldscherm. Bij output redirect gaat deze uitvoer niet meer naar het beeldscherm, maar naar een file. Bij output redirect luidt de syntax: > FILE KOMMANDO <cr>.
Na afloop van KOMMANDO wordt vanzelf FILE gesloten.
Een voorbeeld: > FILE DIR. Het kommando DIR wordt geexecuteerd, maar de uitvoer ervan gaat niet naar het beeldscherm, maar naar de file FILE.

Drive- en directory specificatie:

Drivespecificatie:
DOS65 kent de volgende manieren om drives te specificeren:

```
0:      drive 0
1:      drive 1
2:      drive 2
S:      systeemdrive
U:      userdrive
W:      workdrive
```

Een drivespecificatie dient te worden afgesloten met het karakter ':'.
Directoryspecificatie:
DOS65 kent per schijf acht directories, namelijk de hoofd-directory, en zeven sub-directories. De directories worden aangeduid met het karakter '@' voor de hoofd-directory, en met de karakter 'A' tot en met 'G' voor de sub-directories. Een sub-directory wordt automatisch gekreeerd, wanneer er in wordt geschreven. Een sub-directory wordt automatisch verwijderd, wanneer er geen files (meer) in staan. Een directory-specificatie dient te worden afgesloten met het karakter '/'
Voorbeelden:

```
SAVE A/TEST             schrijf file TEST naar user drive directory A.
LOAD 1:B/TEST           laad file TEST van drive 1 directory B.
COPY 0:C/* 1:A/         kopieer alle files van drive 0 directory C naar
                        drive 1 directory A.
```

Zie voor verdere uitleg en voorbeelden de nieuwe utility ASN.

Foutmeldingen DOS65:

Tijdens het gebruik van DOS65 kunnen er situaties ontstaan, die tot een foutmelding (error) leiden. Deze situaties zijn:
- de systeemgebruiker geeft een onjuist en/of onuitvoerbaar kommando,
- DOS65 signaleert een foutmelding vanuit het systeem.

In vrijwel alle gevallen zal DOS65 reageren met een beknopte, doch volledige foutmelding in de zin van 'read error', 'option error' o.i.d. Onder zekere omstandigheden zal DOS65 echter slechts een foutmelding geven in de vorm van 'Error n', waarbij 'n' staat voor het foutnummer. Om te kunnen achterhalen wat er fout ging, volgt hieronder een overzicht van de meest voorkomende DOS65 foutmeldingen.

1	memory full	ontstaat bij een LOAD, indien file te groot voor beschikbaar geheugen.
2	disk full	ontstaat bij WRITE, als schijf vol is.
10	file specification	filenaam langer dan 14 karakters.
11	device/directory select	onjuist device of directory opgave.
12	no filename	filenaam ontbreekt, terwijl deze nodig was.
13	file not found	file niet gevonden in de momemtele directory.
15	option	verkeerde opgave van optie(s).
16	hex/dec data	foutieve opgave parameters.
25	delete protected	status is delete protected (RW-).
26	write protected	file status is write protected (R-D).
27	read protected	file status is read protected (-WD).
28	permission denied	file is geen kommando (RWD-).
29	incorrect format	ontstaat bijv. bij LIST, TYPE of SEE van niet-ASCII file.
41	existing file	opgegeven file bestaat reeds (mogelijk bij RENAME).
42	not the same drive	ontstaat o.a. bij onjuiste drivespec bij RENAME.
80	read error	ontstaat bij defekte schijf.
90	write error	ontstaat bij defekte schijf.
C0	drive not ready	ontstaat bij ontbreken drive-ready (geen diskette en/of drive-vergrendeling).
D0	disk protected	ontstaat bij poging om op disk-protected schijf te schrijven.

HOOFDSTUK 3 : DOS65 permanente kommando's.

Kommando's en opties mogen zowel in hoofdletters als in kleine letters worden opgegeven. De letters, die noodzakelijk zijn voor de uitvoering van een kommando, zijn bij de beschrijving ervan steeds in hoofdletters aangegeven. LC doet dus hetzelfde als LCMD (LCmd), LO hetzelfde als LOAD (LOad).
De volgorde van opgave bij opties is niet belangrijk, -WDR doet hetzelfde als -RWD. Parameters, zoals gebruikt bij bijvoorbeeld het SAVE kommando (BG,EN,ST), moeten gescheiden worden door een komma.

Als de systeemgebruiker een kommando geeft aan DOS65, wordt er door DOS65 eerst gekeken, of het een permanent kommando is, dus een hierna beschreven kommando. Wordt dit kommando niet gevonden, dan gaat DOS65 kijken, of het kommando als utility op de systeemschijf staat. De kommando's die daar beschikbaar zijn, worden beschreven in hoofstuk 4.

< FILE COMMAND

Haal de invoer voor COMMAND uit de file FILE. Hiermee is het mogelijk om bijv. COMMAND te initialiseren vanuit FILE.

<T COMMAND

Reset input redirect. Dus: geef het toetsenbord weer vrij. In wezen is dit een 'overbodig' kommando. Normaliter vindt reset input redirect immers automatisch plaats (zie de beschrijving van het voorgaande kommando) Het wordt dan ook alleen bij bijzondere toepassingen gebruikt.

> FILE [-TF] COMMAND

Stuurt de uitvoer van het kommando COMMAND naar de file FILE. Met dit kommando kan dus uitvoer bewaard worden om later nog eens te bekijken, of te bewerken, bijv. met de editor. Als FILE nog niet bestond, wordt deze gekreeerd. Als FILE wel bestond, wordt de inhoud ervan overschreven, tenzij ze write-protected was. Opletten dus. Bovendien moet FILE, als deze al bestond, wel van het ascii type zijn.
Opties:

-T Stuur de uitvoer ook naar het beeldscherm. Dit is handig om het verloop van COMMAND te blijven volgen.
Immers, zonder de -T optie gaat de uitvoer alleen naar de file FILE, en is er dus niets meer op het beeldscherm te zien.

-F De uitvoer gaat, voordat het naar de file FILE geschreven wordt, eerst door een 'filter'. Dit filter houdt control en grafische karakters tegen. Deze hebben we namelijk niet nodig, en zorgen hooguit voor ongemak c.q. problemen.

De syntax van dit kommando met gebruik van opties wordt

> -T FILE COMMAND	-T optie ingeschakeld.
> -F FILE COMMAND	-F optie ingeschakeld.
> -TF FILE COMMAND	beide opties ingeschakeld, in de praktijk wordt dit het

meest gebruikt.

>> [-TF] FILE COMMAND

Werking gelijk als > FILE COMMAND. De uitvoer wordt echter 'achter' de inhoud van FILE gezet. Een append-funktie dus.
Opties: gelijk aan > FILE COMMAND.

>T COMMAND

Reset output redirect. Dus: uitvoer weer naar beeldscherm.
Ook dit is een 'overbodig' kommando. (Zie beschrijving <T)

>P COMMAND

Dit is een bijzondere vorm van output redirect. De uitvoer gaat nu niet naar schijf, maar naar de printer.
Optie:
-T Uitvoer ook naar het beeldscherm.

<> COMMAND

Dit kommando laat de input/output redirect aktief na afloop van COMMAND, in plaats van dat toetsenbord/beeldscherm weer worden vrijgegeven, zoals bij 'normale' input of output redirect. Het kan gebruikt worden om meerdere resultaten in een file op te slaan (output) of als invoer om kommando's uit de file te halen (input).
Voorbeeld:
< INPUT <> DIR voert het kommando DIR uit, en haalt het volgende kommando uit de file INPUT. Dit gebeurt om het kommando DIR geen invoer nodig heeft, er wordt dus ook niets uit de file INPUT gelezen. Zonder de toevoeging <> zou, na uitvoering van DIR, de input-redirect gestopt worden, zodat er niets met de gegevens in de file INPUT gedaan zou worden.
N.B. Toepassingen van dit kommando zijn beperkt. Bovendien dient het met zorg gebruikt te worden.

@ FILE [parma parmb]

Executeer FILE als kommando file. Laat VOOR executie van de in file aanwezige kommando's de aktuele kommando regel zien.
(dit onderscheid
'@ FILE <cr>' van
'FILE' <cr>').
Opties:
parma parameter &1
parmb parameter &2
(Zie hoofdstuk 2, onderdeel indirekte files).
Voorbeeld:
Veronderstel file TEST
ECHO Dit is file test.
@ TEST <cr> geeft als resultaat:
ECHO Dit is file test.

```
Dit is file test.
```

CHMODe -RWD FILE

Dit kommando verandert de mode van de file FILE, afhankelijk van de opgegeven opties.
Voorbeelden:

CHMODE -R FILE	Zet file-mode readable, de file kan gelezen worden.
CHMODE -W FILE	Zet file-mode writeable, er kan in de file geschreven worden.
CHMODE -D FILE	Zet file-mode deleteable, de file kan gewist worden.
CHMODE FILE	Met deze file kan niets meer worden gedaan, omdat ze geen mode (meer) heeft. Ze is noch readable, noch writeable, noch deleteable.

Kombinaties zijn ook mogelijk, dus -RW of -WD of -RWD.
Er bestaat een uitgebreidere change-mode, te weten de utility SETMODE (zie verderop).

Dir [-/]

Laat de directories van een schijf zien, voorafgegaan door de naam van de schijf, en afgesloten met vermelding van het aantal vrije blocks.
(Elk block is 256 bytes)
Optie:

-/ Laat ook de eventueel aanwezige sub-directories zien.

Voorbeelden:

DIR	Geef directory van de gebruikersdrive.
DIR 0:	Geef directory van drive 0.
DIR -/	Geef alle aanwezige directories van de gebruikersdrive.
DIR -/ S:	Geef alle aanwezige directories van de systeemschijf.

Go XXXX

Voer een (sub)routine uit, die begint op adres XXXX (XXXX is hexadecimaal). Als de routine eindigt met een RTS ($60, return from subroutine) komt men weer terug op DOS niveau.

LCmd

Springt naar de laatst in het geheugen gehaalde utility. Het is dus een manier, om die utility aan te roepen, zonder dat deze eerst weer van schijf moet worden gelezen. Toepassingen met name bij de COPY utility, bij het kopieren van meerdere schijven.
Stel, dat er twee schijven geheel gekopieerd moeten worden. De eerste wordt dan gekopieerd met:

```
COPY -W 0:* 1:          (Funktie * : zie hoofdstuk 5)
```

Omdat de COPY utility na het kopieren van de eerste schijf nog steeds in het geheugen staat, is het nu niet noodzakelijk deze voor de tweede keer in te lezen. Er kan dan ook volstaan worden met:

LCMD 0:* 1:
Wel dient er op gelet te worden, dat er niets wordt gewijzigd in het geheugen, waar de utility staat. Anders hangt het systeem na LCMD. Gebruik het dan ook alleen direct of vlak na het binnenhalen van de utility.

LOad FILE BG,EN

Laadt de file FILE in het geheugen vanaf beginadres BG tot en met eindadres EN. BG en EN zijn hexadecimaal. Zonder opgave van BG,EN wordt het beginadres uit de file gehaald. Als de file geen beginadres bevat, (0), dan volgt een foutmelding en stopt LOAD. Het maximale eindadres, in theorie, is $FFFF. Als dit wordt overschreden, dan komt er ook een foutmelding en stopt LOAD eveneens. In de praktijk zal het maximale eindadres lager zijn. Het bovenste gedeelte van het geheugen bij een 6502 systeem bestaat immers altijd uit ROM.
Dit kommando houdt rekening met het gegeven, dat FILE binaire code (machinetaal) kan bevatten. In dat geval staat het beginadres en de lengte in de file. Wordt er een beginadres opgegeven, dan wordt het beginadres in de file genegeerd, en de file ingelezen vanaf het door de gebruiker opgegeven beginadres BG.
Bij het opgeven van BG en/of EN moet worden opgepast dat er niet over DOS65 of het werkgeheugen van DOS65 wordt geschreven. Bij het inlezen van grote ascii files is het aan te raden om een eindadres op te geven dat lager is dan het beginadres van DOS65. Bij de gebruikelijke implementatie is dit beginadres $AA00.

Run FILE

Laadt een file van de opgegeven drive en executeert deze uit. De file moet wel executable zijn (SETMODE -C) Zo niet, dan wordt de file niet ingelezen, en dus ook niet geexecuteerd.
Voorbeelden:

RUN TEST	Laadt en executeer de file TEST van de gebruikersdrive.
RUN 0:TEST	Laadt en executeer de file TEST van drive 0.
RUN S:TEST	Laadt en executeer de file TEST van de systeem drive.

SAve [-AB] FILE BG,EN,ST

Schrijft een geheugengebied vanaf BG tot en met EN onder de naam FILE naar schijf. Opgave van BG en EN zijn noodzakelijk om het geheugengebied af te bakenen. Indien het geheugengebied een programma is, waarvan het daadwerkelijke startadres anders is dan het beginadres BG, dan kan dit startadres worden opgegeven bij ST.
BG,EN en ST zijn hexadecimaal.
Opgelet: als de file FILE al bestaat, en niet write-protected is, dan wordt deze overschreven.
Opties:

-A	Voegt toe aan de reeds bestaande file, een append funktie dus.

-B Schrijf weg onder binair formaat. Dit houdt in, dat , startadres, beginadres en lengte ook in de file worden geschreven.

SEE FILE

Laat de ascii file FILE zien. Dit is een 'uitgeklede' versie van de LIST utility.

STAT

Dit kommando laat zien welke files er geopend zijn, c.q. open staan. Bij 'normaal' gebruik van DOS65 worden alle eventuele openstaande, of geopende files door DOS65 vanzelf weer gesloten. Een toepassing van STAT is dan ook 'veiligheidshalve' even kijken of er nog files openstaan voor dat het systeem wordt uitgeschakeld. (Bijv. bij de utility PRINT, zie hoofdstuk 4)

SYNC

Dit kommando sluit alle openstaande files.
Optie:
-N Laat de file, die geprint wordt door de PRINT utility geopend.
Speciale toepassing: het voortijdig afbreken van het PRINT kommando.

HOOFDSTUK 4 : DOS65 Utilities.

>D COMMAND

Output redirect naar een 'dummy' routine. De output van COMMAND gaat naar het niets. Na afloop van het commando gaat de output weer naar het beeldscherm.
(In de praktijk wordt deze utility niet vaak gebruikt. Toepassingen zijn te vinden in bijv. het uittesten van programma's, zonder dat we in de uitvoer geinteresseerd zijn.)

APPend [-AQW] FILE1 FILEn FILE

Plaats FILE1 tot en met FILEn in FILE.
Opties:

-A Voeg toe aan FILE.
-Q Vraagt of de gevonden file(s) moet/moeten worden geplaatst in FILE.
-W Wacht na start zodat de systeemgebruiker schijven kan wisselen.

Voorbeelden:

APPEND FILE1 FILE2	plaats FILE1 in FILE2.
APPEND -A FILE3 FILE2	voeg FILE3 toe aan FILE2.
APPEND FILE1 FILE2 FILE3 FILE	plaats de files FILE1, FILE2 en FILE3 in FILE.

Wordt alleen APPEND opgegeven, dus zonder opties en FILE(S) dan worden de mogelijkheden van APPEND op het scherm gezet.

AS [OPTIES] FILE

Assembleer de source file FILE. FILE moet de extensie '.mac' hebben. De door AS gegenereerde code wordt default weggeschreven in de file FILE.bin.
Zie handleiding AS.

ASN drive=drivespecificatie, directoryspecificatie

Definieer de system en/of user en/of work drive en directory('s).
De karakters '=', ':' en '/' zijn optioneel. (Zie onderstaande voorbeelden)
Zonder opgave van de directory geldt de drive specificatie voor de home directory.
Zonder opgave van de drive geldt de directory specificatie voor de user drive.
Wordt alleen ASN opgegeven, dus zonder parameters, dan wordt de op dat moment aanwezige specificatie gegeven.
Voorbeelden:

ASN S=0/@ U=1	system drive 0, user drive 1
ASN S0 1	system drive 0, user drive 1
ASN 1	user drive 1
ASN W1:A/	work drive 1 directory A
ASN 1B	user drive 1 directory B
ASN B	user drive directory B
ASN	geen parameters, geeft (in dit voorbeeld) als resultaat: S=0: U=1: W=1:G/

Aanbeveling voor systemen met 2 drives:
ASN S=0: U=1: W=1:G/
Deze definitie kan dan worden opgenomen in de file LOGIN.COM. Voor deze definitie geldt:
system drive 0
user drive 1 home directory
work drive 1 directory G

BOOTLINK FILE

Vanwege de nieuwe schijforganisatie van DOS65 2.01 is het absoluut noodzakelijk deze utility te gebruiken na het kopieeren van de file BOOT, dus in het geval dat de de systeemschijf gekopieeerd wordt (aanmaken back-up bijv.).
Wanneer de systeemschijf in drive 0 gekopieeerd wordt naar drive 1, dan moet de volgende procedure worden gevolgt:

COPY 0:* 1:	Kopieeer alle files van drive 0 naar drive 1.
BOOTLINK 1:BOOT	Executeer BOOTLINK op de file 1:BOOT

BOOTLINK doet het volgende: Er wordt op een bepaalde plaats op de schijf een pointer gezet. Deze pointer is van primair belang, wanneer BOOT, oftewel DOS65 2.01 geladen wordt vanuit I065 middels B (b).
Zonder executie van BOOTLINK zal de in I065 aanwezige bootstrap-loader er niet in slagen om de file BOOT te herkennen, en dus in te lezen, hetgeen dan ook zal resulteren in een BOOT READ ERROR.

CAt FILE

List de directory met informatie over de files. De informatie bestaat uit de mode van de file, de plaats in de directory, de filenaam, het beginadres, de lengte en tijd en datum wanneer de file op schijf is gezet dan wel gewijzigd is. Een en ander wordt voorafgegaan door de naam van de schijf, en afgesloten met vermelding van het aantal vrije blocks.
(Elk block is 256 bytes)
Voorbeelden:

CAT	laat alle files op de gebruikersdrive zien.
CAT 0:	laat alle files op drive 0 zien.
CAT *.TXT	laat alle files op de gebruikersdrive zien, die eindigen op .TXT
CAT 1:B/	laat alle files zien van drive 1, directory B.
CAT U:A/SET*	laat alle files zien op de gebruikersdrive, directory A, die beginnen met SET.

Opmerking :
De funktie van de ster (*) wordt in hoofdstuk 5 nader verklaard.

CLear

Wist het scherm.

COpy [-PQW] FILE1 FILE2

Kopieer FILE1 naar FILE2. Als FILE2 niet wordt opgegeven wordt dezelfde naam genomen als FILE1. In dit geval moet de kopie van FILE1 wel op een andere schijf komen dan FILE1 zelf. Anders zou FILE1 twee keer op dezelfde schijf staan (zie voorbeelden).
FILE1 is doorgaans een (1) naam, maar het mogen er ook meer zijn, mits ze gescheiden zijn door een komma, en op dezelfde schijf staan.
Opties:

-P Print geen file list tijdens het kopieren.
-Q Vraag of de gevonden file gekopieerd moet worden.
-W Wacht na de start en elke keer als er een andere schijf in de drive moet.

Voorbeelden:

COPY 0:FILE1 1:	kopieer FILE1 van drive 0 naar drive 1.
COPY 0:FILE1,FILE2 1:	kopieer FILE1 en FILE2 van drive 0 naar drive 1.
COPY FILE1 FILE2	kopieer FILE1 naar FILE2 op de gebruikersdrive.
COPY 0:* 1:	kopieer alle files van drive 0 naar drive 1.
COPY -W 0:* 1:	idem als vorige voorbeeld, maar wacht met daadwerkelijk kopieren, zodat er eerst schijven gewisseld kunnen worden.
COPY 1:*.TXT 0:	kopieer alle files van drive 1 naar drive 0, die eindigen op .TXT
COPY 0:A/* 1:D/	kopieer alle files van drive 0, directory A naar drive 1, directory D.

Wordt alleen COPY opgegeven, dus zonder opties en FILE dan worden de mogelijkheden van COPY op het scherm gezet.
Opmerking :
De funktie van de ster (*) wordt in hoofdstuk 5 nader verklaard.

CREate FILE

Maakt een ascii file aan met de naam FILE. Als FILE reeds bestond, dan wordt ze overschreven. De prompt verandert in '&'. Alle ingetypte karakters worden weggeschreven in de file FILE. Met <del> wordt het laatste karakter gewist.
Indien aan het begin van een nieuwe regel control D (CTL D ,$04) getikt wordt, gevolgd door <cr>, dan wordt FILE gesloten en wordt CREATE verlaten, dus terugkeer naar DOS65.

DDoctor

Met deze utilitie is het mogelijk om afzonderlijke sektoren van schijf te lezen. De ingelezen sektor wordt in een buffer geplaatst. Het is mogelijk om deze buffer te verplaatsen naar een tweede buffer, en vice-versa. Op deze wijze kan dan een sektor verplaatst of gekopieerd worden. De inhoud van de buffer kan op twee manieren

gewijzigd worden, namelijk per nibble en per (ascii)-karakter. DDOCTOR heeft twee mogelijkheden om de sektoren de adresseren: de eerste is 'logical', d.w.z. dat de sektoren worden genummerd volgens de op de schijf aanwezige interleave factor (zie ook de toelichting bij het kommando FORMAT). De tweede is 'fysical'. In dit geval komt het sektornummer overeen met de fysieke plaats in de aktuele track. (dus 1, 2, 3, 4 enz.) De default adresseermode is 'logical'. Een gewijzigde sektor kan worden teruggeschreven naar de schijf. Ook is het mogelijk om te zoeken naar een hex- of ascii-string zowel binnen een sektor, als een track, alsmede de gehele schijf. Waarschuwing: Gebruik deze utilitie met zorg! Onzorgvuldig of onervaren gebruik kan gemakkelijk leiden tot onleesbare schijven of files. Met deze utilitie is het namelijk mogelijk om toegang te verkrijgen tot de interne DOS65 organisatie, bijv. de directory. Experimenteer daarom in eerste instantie met 'waardeloos' materiaal.

DELete [-YW] FILE

Verwijder FILE van schijf.
Opties:

-W Wacht na de start om een andere schijf in de drive te kunnen plaatsen.

-Y Verwijder zonder te vragen of de file echt wel verwijderd moet worden.

Voorbeelden:

DELETE FILE	verwijder FILE van de gebruikersdrive.
DELETE 0:FILE	verwijder FILE van drive 0.
DELETE -Y FILE	verwijder FILE van de gebruikersdrive zonder nader te vragen.
DELETE -Y 1:*	verwijder alle files van drive 1 , zonder nader te vragen. (de schijf wordt gewist)

DPTime [s]

Stelt display-tijd in. Om inbranden van het beeldscherm te voorkomen, zal, als geen enkele toets wordt gebruikt na de ingestelde tijd het beeld doven. Deze tijd wordt in seconden opgegeven. 's' is minimaal 0 en maximaal 65535. Het gebruik van een willekeurige toets re-aktiveert het scherm. DPTIME <return> laat de laatst ingestelde tijd zien. De default waarde voor de display-tijd is 1800 seconden (30 minuten).
Voorbeelden:

DPTIME <return)	geeft als resultaat:
1800 seconds	
DPTIME 60 <return>	stelt display-tijd op 60 seconden

DUmp [-CEM] FILE

Laat FILE als hexdump zien.
Opties:

-C Maak eerst scherm schoon, alvorens te dumpen.

-E Gebruik een 4 Kbyte buffer. Bij gebruik van de-

ze optie wordt eerst 4 Kbyte ingelezen, voordat ze gedumpt wordt.

-M Stop na een aantal regels. Om de rest van de file te zien gelden de volgende toetsen :

<cr> de volgende regel.
D de volgende 10 regels.
<sp> de volgende 20 regels.
C CLS en de volgende 23 regels.
CTL C stopt het kommando.

ECHo [-N] TEKST

Stuurt TEKST naar het beeldscherm. ECHO ABC zet dus ABC op het beeldscherm. De filosofie achter dit kommando is dat 'het beeldscherm' een (intelligente) terminal kan zijn, die met ECHO in een bepaalde mode kan worden gezet.
Een speciale toepassing is >P ECHO TEXT. In dit geval wordt TEXT naar de printer gestuurd, om deze in een bepaalde mode te zetten (denk aan zoveel lines per inch, zoveel karakters per inch, printmode enz.)
TEXT mag alle karakters bevatten, inklusief ESCAPE ($1B) en andere, voor de printer of terminal gebruikelijke besturingskarakters.
Optie:
-N Geen <cr> ($0D) en <lf> ($0A) verzenden na TEXT.
Voorbeeld:
Stel dat een printer in de 'elite'-mode kan worden gezet door ESC B op te geven. Het kommando luidt dan:
>P ECHO -N <esc> B

EDitor [FILE]

Start de full screen editor/wordprocessor. Indien FILE wordt opgegeven, dan wordt deze binnengelezen (mits FILE bestaat). FILE is dan de Edit file.
Zie handleiding ED.

FORMat

Formatteert een schijf voor DOS65. FORMAT stelt de volgende vragen:
Drive (0, 1* or 2)?
Hier wordt opgegeven in welke drive de te formatteren schijf is geplaatst.
40(Y) or 80(*) tracks, 2 steps/track(2)?
Hier wordt opgegeven welke type drive gebruikt wordt, 40 of 80 tracks. Het is mogelijk om te formatteren met 2 steps/track. Dit houdt in dat er met een 80 tracks drive een 40 tracks schijf wordt geformatteerd.
Single(Y), Double(*) or Extra(X) Density?
Hier wordt de density opgegeven. Opgelet: extra density mag niet worden gebruikt voor schijven, die bedoeld zijn om als systeemschijf te gebruiken !
Single(Y) or Double(*) sided?
Hier wordt single- of double-sided opgegeven. Dit is afhankelijk van het soort drive.
Interleave factor (2, 3*, 4 or 5)?
Hier wordt de interleave factor opgegeven.
Als bij de voorgaande vragen alleen <return> wordt ge-

kopieerd,
- enzovoorts, tot en met sektor 8.

Het begrip interleave factor heeft direkt te maken met de term 'wachten op' in dit voorbeeld. Wanneer we als interleave factor '1' kiezen, dan gebeurt het volgende:
De sektoren worden genummerd als volgt:
1 2 3 4 5 6 7 8
Herhalen we bovenstaand voorbeeld, dan gebeurt er in werkelijkheid dit:
- sektor nummer 1 wordt ingelezen, en naar de juiste plaats gekopieerd
- maar: tijdens het kopieren is sektor 2 de leeskop van de drive al gepasseerd. DOS65 moet dan wachten tot sektor 2 weer voorbij komt, dus bijna een hele omwenteling van de schijf. Om een (1) track (8 sektoren) in te lezen zou DOS65 dan de tijd van 8 omwentelingen nodig hebben. Deze tijd valt aanmerkelijk te bekorten door een andere interleave factor te kiezen.

Wordt de interleave factor op '3' gezet, dan worden de sektoren als volgt genummerd:
1 4 7 2 5 8 3
Dus: tussen de sektoren 1 en 2 zitten de sektoren 4 en 7, tussen de sektoren 2 en 3 zitten de sektoren 5 en 8 enz.
Wordt bovenstaand voorbeeld nu herhaald, dan blijkt dat DOS65 geen hele omwenteling hoeft te wachten, tot de volgende sektor voorbij komt. Nu gebeurt het volgende:
- sektor nummer 1 wordt ingelezen, en naar de juiste plaats gekopieerd. De schijf is intussen wel doorgedraaid, en staat bijv. ergens boven de sektor met nummer 7.
- DOS65 gaat nu zoeken naar de volgende sektor, namelijk sektor nummer 2, en wat blijkt: na sektor nummer 7 komt sektor nummer 2 ! Er hoeft nu dan ook niet een hele omwenteling gewacht te worden.

Het zal nu duidelijk zijn, dat de snelheid van de kommunikatie tussen de floppy-disk en de computer nauw verwant is aan de keuze van de interleave factor.
Het is deze keuze, die bepalend is voor de eigenschap van 'overdrachtsnelheid'. Een juiste keuze optimaliseert de snelheid, een foutieve keuze vertraagt deze. Gebleken is, dat de waarde 3 een goed kompromis is voor de meeste toepassingen. Wanneer men met andere waarden wil experimenteren, is men daar vrij in. De kompatabiliteit van DOS65 wordt er niet door beinvloed.

HElp [FILE]

Deze utilitie geeft (engelstalige) uitleg over de standaard kommando's en utilities van DOS65. Indien FILE wordt opgegeven, dan wordt alleen info gegeven over FILE, mits deze info aanwezig is.

LIst [-CELMN +m,n] FILE

Laat FILE zien. FILE moet ascii-gegevens bevatten, anders volgt een foutmelding.
Negatieve karakters worden geprint als '~' karakter.
DEL ($7F) en control karakters worden geprint als '^' karakter.

Een TAB karakter ($09) wordt vertaald naar (uitgevoerd als) 8 spaties.
Opties:

-C Maak eerst het scherm schoon.
-E Gebruik een 4 Kbyte buffer.
-L Print het regelnummer aan het begin van elke nieuwe regel.
-M Stop na een aantal regels. Om de rest van de file te zien gelden de volgende toetsen :
<cr> de volgende regel.
D de volgende 10 regels.
<sp> de volgende 20 regels.
C CLS en de volgende 23 regels.
CTL C stopt het kommando.
-N Converteer <cr> niet als <cr> <lf>.
+m,n Laat FILE zien vanaf regel m t/m n. De regels worden geteld met het <cr> ($0D) karakter.

Opmerking:
De optie -N wordt gebruikt bij printers die na ontvangst van <cr> automatisch een <lf> geven.

MAP FILE

Laat zien hoe een binaire file in het geheugen wordt geladen. Indien de file een startadres heeft, dan wordt dit als laatste gegeven.
Voorbeeld:

```
MAP S:EDITOR            geeft als resultaat:
2000-3FFF
0400-04FF
0600-06FF
0580-05FF
0700-07FF
2000                    (dit is het startadres)
```

MEMFill BEGIN,END,DATA

Vult het geheugen met het byte DATA vanaf adres BEGIN t/m END. Dit commando moet, gezien zijn werking, met zorg worden gebruikt. Immers, wordt er geheugen gevuld, waar DOS65 staat (doorgaans $A000-$BFFF), dan hangt het systeem.
Voorbeeld:

```
MEMFILL 2000,2FFF,AA    Vul het geheugen van $2000 t/m
                        t/m $2FFF met data $AA.
```

MEMMove BEGIN,END,DEST

Verhuisd een stuk geheugen vanaf BEGIN t/m END naar DEST e.v. Er wordt op de korrekte manier verhuisd, wat betekent dat er gekeken wordt waar er begonnen moet worden.
Voorbeeld:

```
MEMMOVE 2000,20FF,8000  Verhuisd het geheugenveld van
                        $2000 t/m $20FF naar $8000 en
                        hoger.
```

MONitor

Executeer het monitorprogramma MON65. Zie handleiding

MON65

MTOAS [-OTFLU] FILE

Converteer een source file van MOSER formaat naar AS formaat. Deze utilitie is geschreven voor systeemgebruikers die voorheen in MOSER schreven, en overstappen op AS. Vanwege de grotere mogelijkheden en assembleersnelheid van AS is het dan aan te bevelen de MOSER formaat source files te converteren naar AS formaat files. FILE moet de extensie .txt hebben. De door MTOAS gegenereerde file krijgt ook de naam FILE, maar met de extensie .mac
Deze extensie is noodzakelijk voor verwerking door AS.
Opties:

-O Geen output FILE.mac
-T Uitvoer naar beeldscherm
-F Geen output formattering
Label, opcode and operand veld conversie:
-L Naar kleine letters (lowercase, default)
-U Naar hoofdletters (uppercase)

OLOad FILE OFFSET

Laadt een binary file FILE in het geheugen, rekening houdend met de waarde van OFFSET. Deze waarde wordt opgeteld bij de in de file aanwezige load-adressen. Deze optelling gebeurt modulo 64K. Dit betekent dat de waarde OFFSET altijd een positief getal moet zijn.
Voorbeelden:

OLOAD FILE 1000	Laadt file $1000 hoger in het geheugen.
OLOAD FILE F000	Laadt file $1000 lager in het geheugen

PLIst [-Bn -Cn -E -F -Hn -Ln -Mn -N -On -Pn -Tn -U -W] FILE

List FILE geformatteerd op pagina basis. FILE moet ascii zijn. Hierbij wordt dan op elke pagina een header geprint die tijd, datum, filenaam en pagina nummer bevat.
Opties:

-Bn Breek na n karakters de kolom af en ga verder op de volgende regel.
-Cn Sla alle karakters over die na n karakters binnenkomen (geteld per kolom). De opties -B of -C zijn handig om te gebruiken bij het printen in twee kolommen.
-E Print geen header regel (tijd, datum etc).
-F Print form feed ($0C) i.p.v. lege staart regels
-Hn Print n lege regels op elke nieuwe pagina (default 3). Hiervan gaat er een (1) af als er een header wordt geprint.
-Ln Definieer een pagina grootte (default 72 regels). Het effectieve aantal regels dat geprint wordt per pagina wordt bepaald door de kop en staart regels van dit aantal af te trekken.
-Mn Print de file in twee kolommen. n is de positie die de tweede kolom krijgt. Als een regel van

REName [-W] FILE1 FILE2

Geef FILE1 de naam FILE2. FILE2 mag nog niet bestaan en moet dezelfde drive selekteren als FILE1.
Optie:
-W Wacht na start.
Voorbeelden:

RENAME OLD.TXT NEW.TXT	Verander de filenaam OLD.TXT in NEW.TXT op de gebruikersdrive.
RENAME -W 0:AAP 0:NOOT	Verander de filenaam AAP in NOOT. De file staat op de schijf in drive 0. Wacht na start, zodat de systeemgebruiker schijven kan wisselen.

RS232

Deze utilitie stelt de parameters in van de ACIA 6551. Zonder deze utilitie zou dat moeten plaatsvinden door bitjes in het kommando- en kontroleregister te (re)setten. Met behulp van RS232 kunnen de volgende parameters worden ingesteld: baudrate, woordlengte, pariteit, stopbits, receiver clock, receiver mode, receiver interrupt, DTR en transmit status.
Deze utilitie kent zes kommando's, te weten:

H	geeft een hulpmenu
CTL H	naar links in huidige parameterveld
CTL I	naar rechts in huidige parameterveld
CTL J	naar volgende parameterveld
CTL K	naar vorig parameterveld
Q	einde RS232 utilitie

Na het kommando Q wordt gevraagd of de aktuele status op schijf geschreven moet worden. Zoja, dan worden deze gegevens op de systeemschijf gezet onder de de naam RS232.DAT. Deze gevens kunnen dan in een latere sessie geladen worden, bijvoorbeeld via de file LOGIN.COM (LOAD S:RS232.DAT).

SDir [-IS]

Print een op alfabet gesorteerde directory.
Opties:
-I De niet-significante karakters van de filenaam worden ge-inverteerd op het beeldscherm gezet. (Zie ook hoofdstuk 5.)
-S Output in een (1) kolom
Voorbeelden:

SDIR	Geef een gesorteerde directory van de gebruikersdrive.
SDIR -I S:	Geef een gesorteerde directory van de systeemdrive. De schijf in de systeemdrive bevat alle utilities, waarvan het niet-significante deel van de naam van de utility ge-inverteerd op het beeldscherm wordt gezet. Hiermee is te zien, hoe de utility afgekort kan worden opgegeven.
SDIR -S	Geef een gesorteerde directory, maar

-W writeable, er kan in de file geschreven worden.
-D deleteable, de file kan verwijderd worden.
-P (nog) geen toepassing.
-C command, de file kan als kommando gebruikt worden.
-E de file is executable.
-CE geeft de file mode X, (nog) niet gebruikt.

Kombinaties zijn mogelijk, dus SETMODE -RW FILE maakt de file readable en writeable.
Er kunnen meerdere files tegelijk van mode worden veranderd, door ze gescheiden door komma's, of m.b.v. speciale karakters (zie hoofdstuk 5) op te geven.

SETPrompt abcd

Met dit kommando is het mogelijk om de standaard DOS65 prompt naar eigen keuze te wijzigen. Standaard is de prompt het teken '$'. Met behulp van SETPROMPT kan dit gewijzigd worden in een prompt van maximaal vier ASCII-karakters.
Voorbeelden:

SETPROMPT + <cr>	de prompt is nu een '+'
SETPROMPT CTL G$ <cr>	de prompt bestaat nu uit een 'beep' en een '$'

SETRtc [H,M,S [D,M,Y]]

Definieer de tijd en datum voor een op het systeem aangesloten real time clock. De datum hoeft niet te worden opgegeven als alleen de tijd moet worden veranderd.
Tijd en/of datum moeten decimaal en geheel worden opgegeven. Indien er geen argumenten worden opgegeven, print SETRTC de datum en de tijd. Indien SETRTC geexecuteerd wordt, zonder dat de real time clock op het systeem is aangesloten, zal er een foutmelding worden gegeven.

SETUp

Deze utilitie dient aan het begin van een sessie geexecuteerd te worden, bijvoorbeeld in de file LOGIN.COM
De funkties van setup zijn:
- het ondersteunen van de real time clock, en
- het opzetten van een 'welkomst-scherm'.

Indien SETUP geexecuteerd wordt, zonder dat de real time clock op het systeem is aangesloten, zal er een foutmelding worden gegeven.

TIme [H,M,S [D,M,Y]]

Definieer de tijd en datum. De datum hoeft niet te worden ingetypt als alleen de tijd moet worden veranderd.
Tijd en/of datum moeten decimaal en geheel worden opgegeven. Indien er geen argumenten worden opgegeven, print TIME de datum en de tijd.

TYpe [-CELMN] FILE

Laat een ascii file zien. De karakters worden letter-

lijk geprint (Dit onderscheidt TYPE van LIST).
Opties: gelijk aan LIST.

UNEXpand [-A] FILE

Vervang twee of meer spaties ($20) door een (of meer) tab ($09) karakter(s) indien de spaties aan het begin van de regel staan.
Er wordt van uitgegaan dat er een tab staat per acht kolommen.
Optie:
-A Zoek de gehele regel af.

VFORMat [-Y]

Formatteert virtual disk, mits aanwezig in het systeem. VFORMAT zoekt zelf uit, hoeveel geheugen er aanwezig is in de virtual disk. De virtual disk wordt gedefinieerd als drive 2:
Optie:
-Y Vraagt geen 'Are you shure (Y/N)'

HOOFDSTUK 5 : Filenamen en speciale karakters:

EEN WOORD VOORAF:
Dit hoofdstuk behandelt een aantal van de bijzondere mogelijkheden die in DOS65 aanwezig zijn. Voor iemand, die weet, wat hij of zij doet, kunnen deze mogelijkheden zeer waardevol zijn. De geboden mogelijkheden bieden bepaalde vormen, om nog aardiger met DOS65 te werken. De auteurs willen de systeemgebruiker graag waarschuwen voor ongewenste effekten van datgeen wat in dit hoofdstuk beschreven staat. Daarom:
- probeer de mogelijkheden NOOIT uit op belangrijke files, bijv. die files die op de systeemschijf staan.
- maak wat 'dummy' files aan, op een aparte schijf, en ga daarmee experimenteren. Als het daarmee fout gaat, is er nog niets aan de hand.

Enkele algemene opmerkingen:
- Maak regelmatig kopieen (backups) van belangrijke files of belangrijke schijven. Schijven zijn ook aan slijtage onderhevig ! Een omgevallen kop koffie kan veel informatie vernietigen !
- Bewaar backup-schijven nooit samen met 'werkschijven'. Bewaar ze op een veilige plaats.
- Vergeet niet, backup-schijven regelmatig te 'refreshen', door ze opnieuw te beschrijven.
- Leg geen schijven op Uw systeem of op Uw monitor. Het uitgestraalde magnetische veld kan vernietigend zijn.
- Verwijder altijd de schijven uit de drives voordat U Uw systeem uitschakelt.

Een file wordt/is uniek gedefinieerd door zijn naam.
De filenaam mag 14 karakters lang zijn, en mag bestaan uit alle karakters behalve:

<null>	(Null	$00)
<cr>	(Carriage return	$0D)
<sp>	(Spatie	$20)
,	(Komma	$2C)

Bovendien bestaan er een aantal speciale karakters, die van belang zijn tijdens het opgeven van de filenaam. De speciale karakters zijn:

% Definieer ALLE volgende karakters als geldig. Dit betekent dat de karakters $, %, :, \, ^ en ~, in de filenaam mogen worden opgenomen. Er kunnen nu ook negatieve karakters (bit 7=1) in de filenaam geplaatst worden. De speciale funktie van '?' en '*' is uitgeschakeld.
Voorbeeld:

```
CREATE %$$$$$$          Kreeert een file met de naam
                        $$$$$$
```

**** De werking lijkt enigszins op die van '%' , echter nu voor EEN (1) karakter (escape funktie).
Voorbeeld:

```
CREATE \$\%\%\$         Kreeert een file met de naam
                        $%%$
```

$ Definieer de karakters na $ als niet-significant. Dit is een handige mogelijkheid om de naam van een utility af te korten,

in de zin dat niet de gehele filenaam hoeft worden opgegeven.
Voorbeeld:
De systeemgebruiker heeft een utility geschreven met de naam NEWCOMMAND. Om deze utility te executeren moet NEWCOMMAND <cr> worden gegeven. Het is nu mogelijk om de filenaam zo te wijzigingen, dat bijv. alleen NEWC <cr> hoeft worden gegeven.
Dit gaat als volgt:
- De filenaam wordt gerenamed naar een andere, tijdelijke naam, die nog niet op schijf mag bestaan.
- De tijdelijke naam wordt, met toevoeging van $ terug gerenamed naar NEWCOMMAND.

De procedure in werkelijkheid:
- RENAME NEWCOMMAND DUMMY
- RENAME DUMMY NEWC$OMMAND

In de directory komt gewoon NEWCOMMAND te staan. NEWC <cr> is nu voldoende om het kommando te executeren. NEWCO, NEWCOMM enz. doen echter hetzelfde.
Opmerking 1:
Het renamen naar een tijdelijke file is nodig, omdat de procedure RENAME NEWCOMMAND NEWC$OMMAND niet wordt toegestaan door DOS65.
Opmerking 2:
Bovenstaande procedure heeft uitsluitend betrekking op files, die als utility gebruikt worden, dus als DOS65 kommando gegeven kunnen worden (mode van de file is C). Bij 'gewone' files, zoals bijv. ascii files dient altijd de hele naam opgegeven te worden.
Opmerking 3:
De systeemgebruiker dient er op te letten, dat er niet meerdere files op dezelfde schijf gelijkluidend worden afgekort. Dus niet een situatie kreeren dat er bijv. NEWC$COMMAND en NEWC$OMMANDO op schijf staat. De namen zijn wel verschillend, maar de afkortingen zijn gelijk. Wordt NEWC <cr> gegeven, dan neemt DOS65 de eerste die wordt tegengekomen in de directory.

De filenaam die opgegeven wordt, behoeft niet geheel overeen te komen met de gevonden filenaam. Er zijn drie zogenaamde 'wildcards' te weten:

? '?' staat voor de vergelijking met precies een (1) karakter. Dit betekent dat elk karakter op die plaats goed is.
Voorbeeld:
DIR TE?T Laat alle filenamen zien, waarvan de eerste twee letters TE zijn, de derde letter maakt niets uit, en de vierde letter een T is.

* het '*' karakter staat voor een vergelijking met nul (0) of meer karakters. Dit is handig om een filenaam als gedeelte op te geven en dan de rest als 'don't care' (niet van belang) te beschouwen. Hierbij kan ook een aantal karakters in b.v. het midden worden vergeleken.
Voorbeelden:
DIR *.TXT Laat alle filenamen zien, die eindigen op .TXT
DIR *COM* Laat alle filenamen zien, waar de letters COM in voorkomen.
DIR TEST* Laat alle filenamen zien, die beginnen met TEST

~ De funktie van dit karakter wordt gebruikt om bepaalde filenamen NIET te nemen.

Voorbeeld:
DIR *~.b Laat alle filenamen zien, MET UITZONDERING VAN die namen die eindigen op .b

Opmerking 1:
In bovenstaande voorbeelden werd steeds het DIR kommando gebruikt. De wildcards kunnen uiteraard bij alle kommando's worden gebruikt waar sprake is van een filenaam cq. filenamen.
Opmerking 2:
Van bovenstaande wildcards wordt in de praktijk de * het meest gebruikt. Zie bijv. COPY.
Opmerking 3:
Het is aan te bevelen de wildcard * niet te gebruiken bij een kreeer funktie (CREATE, SAVE enz.)
Veronderstel de aanwezigheid van een file met de naam TESTFILE, die belangrijke data bevat. Wordt nu een kommando gegeven dat luidt: SAVE TEST* 2000,2FFF dan wordt de file TESTFILE leeggemaakt (gewist), alvorens er in geschreven wordt. De originele inhoud van TESTFILE is daarmee verloren gegaan !

Een filenaam mag zowel in upper als in lowercase worden opgegeven. Als er een file bestaat met de naam 'TEST', dan is
LOAD TEST gelijk aan:
LOAD test
en
FORMAT doet hetzelfde als:
format

Maar:
Bij schrijven in een file wordt geen onderscheid gemaakt tussen upper en lower case.
Voorbeeld:
Veronderstel het kommando 'SAVE test'. Als de file 'test' nog niet bestond, dan wordt ze gekreeerd, met de filenaam 'test', in lower case dus. Bestond de filenaam 'test', dan wordt er ook geschreven onder deze filenaam. Bestond er echter een file 'TEST', in upper case dus, dan wordt er geen file 'test' gekreeerd, maar wordt er weggeschreven onder de filenaam 'TEST' !