Deel III
Geheimen om DOS te automatiseren

Hoofdstuk 9

Met DOSKEY aan het werk

In dit hoofdstuk

- DOSKEY installeren
- DOSKEY opnieuw installeren
- DOSKEY vanaf de opdrachtregel
- Macro's met DOSKEY
- DOSKEY-macro's vergelijken met batchbestanden
- Mijn favoriete trucs met DOSKEY-macro's

DOSKEY installeren

DOSKEY is een veelzijdig stuk gereedschap waarmee een groot aantal functies kan worden uitgevoerd. De drie belangrijkste daarvan zijn het opzoeken van eerder uitgevoerde opdrachten, bewerkingen op de opdrachtregel (geen backspace meer) en een eenvoudige, maar handige utility voor macro's.

DOSKEY neemt genoegen met minder dan 4K geheugen en zelfs dit kan hoog worden geladen. Iedere gebruiker van DOS zou met DOSKEY moeten werken, en niet alleen omdat het zo weinig geheugen inneemt maar omdat het programma zoveel biedt.

Het installeren van DOSKEY kan zelfs mijn grootmoeder. Het enige wat u hoeft te doen, is achter de DOS-prompt de opdracht DOSKEY in te voeren (zonder schakelopties). DOSKEY wordt als resident programma geladen. Beter is het als DOSKEY aan AUTOEXEC.BAT wordt toegevoegd, zodat het altijd wordt geladen.

Modus Insert instellen

De bewerkmodus van de opdrachtregel is ingesteld op overschrijven. U kunt ook instellen dat de tekens bij het typen worden tussengevoegd. Dit doet u met de volgende opdracht:

```
doskey /insert
```

Met de toets Ins (of Insert) activeert u de overschrijfstand weer. Wanneer u echter naderhand

een opdracht invoert, is de invoegstand weer actief. U kunt de standaard veranderen als u DOSKEY opnieuw installeert met de schakeloptie /INSERT of /OVERSTRIKE. Deze schakelopties moeten in zijn geheel worden ingevoerd; er zijn geen afkortingen voor. Om de overschrijfstand standaard in te stellen moet u DOSKEY als volgt invoeren:

```
doskey /overstrike
```

Deze opdracht verandert niets aan de andere instellingen van DOSKEY. De al ingevoerde opdrachten worden niet uit het geheugen gewist, en ook worden er geen macro's verwijderd.

De buffergrootte instellen

DOSKEY maakt gebruik van een buffer om de ingevoerde opdrachten te onthouden en gedefinieerde macro's op te slaan. Standaard worden hiervoor 512 bytes gereserveerd. Met de schakeloptie /BUFSIZE kan de grootte van de buffer worden veranderd. De volgende opdracht installeert DOSKEY, activeert de invoegmodus en maakt een buffer van 1024 bytes:

```
doskey /insert /bufsize=1024
```

De kleinste buffergrootte is 256 bytes; de grootte hangt van u af. 512, 1024, 2048 of meer bytes is mogelijk, maar 2048 bytes is meer dan voldoende.

Wanneer u eerder ingevoerde opdrachten niet meer kunt ophalen (door op de pijl-omhoog te drukken) is de buffer waarschijnlijk vol. Als u probeert een nieuwe DOSKEY-macro toe te wijzen (zie verderop in dit hoofdstuk), verschijnt het volgende bericht:

```
Insufficient memory to store macro.
Use the DOSKEY command with the
/BUFSIZE switch to increase
available memory.
```

Na dit bericht zult u geneigd zijn de buffergrootte te veranderen met de opdracht DOSKEY /BUFSIZE=1024, maar zo eenvoudig gaat dit natuurlijk niet. Als reactie hierop verschijnt het volgende bericht:

```
Cannot change BUFSIZE
```

U zult DOSKEY opnieuw moeten installeren.

DOSKEY opnieuw installeren

De installatie van DOSKEY kan niet ongedaan worden gemaakt, maar u kunt DOSKEY wel opnieuw installeren door nieuwe schakelopties op te geven. Hiervoor geeft u de opdracht DOSKEY in, gevolgd door de schakeloptie /REINSTALL en eventueel andere schakelopties zoals /INSERT en /BUFSIZE=1024:

```
doskey /reinstall /insert
/bufsize=1024
```

Door een hernieuwde installatie van DOSKEY worden alle eerder uitgevoerde opdrachten en macro's verwijderd. Verderop in dit hoofdstuk wordt in de paragraaf *Een macroconstructie* een batchbestand beschreven waarmee de macro's opnieuw worden gemaakt.

DOSKEY en de opdrachtregel

Het grootste deel van de gebruikers neemt zijn toevlucht tot DOSKEY om twee redenen: voor het bewerken van de opdrachtregel en om eerder ingevoerde opdrachten terug te halen. Zelfs al zou DOSKEY nooit voor iets anders worden gebruikt, dan nog is het een waardevol programma.

De opdrachtregel bewerken

Soms geven de eenvoudigste dingen in het leven de meeste voldoening. Dit geldt ook voor DOSKEY. Wanneer u een opdracht invoert, gaat de cursor met de pijl-links een plaats naar links, maar het teken wordt niet verwijderd zoals dat in DOS wel gebeurt. Met andere woorden, de linkerpijl gedraagt zich niet als de toets Backspace, maar als de pijl-links in het programma EDIT. Behalve deze functie biedt DOSKEY nog meer mogelijkheden die in onderstaande tabel worden samengevat.

Tabel 9-1 De bewerktoetsen in DOSKEY

Toets	Doel
Pijl links	Verplaatst de cursor één positie naar links
Ctrl-Pijl links	Verplaatst de cursor naar de eerste letter van het woord links
Pijl rechts	Verplaatst de cursor één positie naar rechts

Ctrl-Pijl rechts	Verplaatst de cursor naar de eerste letter van het woord rechts
Home	Verplaatst de cursor naar het begin van de opdrachtregel
Ctrl-Home	Verwijdert alle tekens links van de cursor en verplaatst de resterende tekens naar het begin van de opdrachtregel
End	Verplaatst de cursor naar het einde van de opdrachtregel
Ctrl-End	Verwijdert het teken onder de cursor en alle tekens rechts van deze positie
Del	Verwijdert het teken onder de cursor
Backspace	Verwijdert het teken links van de cursor
Esc	Verwijdert de volledige opdrachtregel
Ins	Schakelt tussen invoegstand en overschrijfstand

Opdrachten onthouden

Nadat DOSKEY is geladen, wordt elke opdracht die wordt uitgevoerd, opgeslagen in een opdrachtbuffer. Hierin staan alle oude opdrachten. Wanneer de buffer vol is, worden de oudste opdrachten verwijderd als er nieuwe aan worden toegevoegd.

Met de pijl-omhoog wordt de laatst ingevoerde opdracht weer op de opdrachtregel geplaatst. Nogmaals op deze toets drukken laat de voorlaatste opdracht verschijnen. Zo kunt u alle eerder ingevoerde opdrachten oproepen. Met de pijl-omlaag bladert u in omgekeerde richting door de lijst. Wanneer u de gewenste opdracht ziet, hoeft u alleen op Enter te drukken om de opdracht uit te voeren. De opdracht kunt u natuurlijk ook wijzigen.

Met PgUp haalt u in één keer de oudste opdracht op.

Mijn vier favoriete (vergeten) DOSKEY-opdrachten

De vijf meest gebruikte toetsen in DOSKEY zijn de pijltoetsen en de Ins-toets. Er zijn nog andere toetsen die de moeite waard zijn en die uw produktiviteit verhogen:

Opdrachten op nummer selecteren

Met F7 wordt een lijst afgebeeld met alle opdrachten die eerder zijn uitgevoerd. De oudste opdracht is opdracht 1. Wanneer de lijst met opdrachten te lang is voor het scherm, wordt door DOS automatisch het doorrollen van het scherm gestopt met MORE.

Wanneer u op F9 drukt, verschijnt de regel `Line number:` Hierachter kunt u een nummer van een opdracht invoeren en vervolgens met Enter de geselecteerde opdracht starten.

Opdrachten met behulp van tekens selecteren

Als u zich de eerste tekens van een opdracht herinnert, hoeft u niet door de hele lijst te bladeren. U hoeft alleen één of meer tekens in te voeren waarna u op F8 drukt. DOS zoekt in de lijst en plaatst de gewenste opdracht op de opdrachtregel als uw zoekcriterium overeenkomt met deze opdracht.

Twee of meer opdrachten tegelijk uitvoeren

Een weinig bekende mogelijkheid van DOSKEY is dat er meerdere opdrachten in één keer kunnen worden uitgevoerd. U typt de eerste opdracht op de opdrachtregel, vervolgens drukt u op Ctrl-T waarna u de tweede opdracht invoert. Zolang u niet meer dan 127 tekens invoert, kunt u opdrachten gescheiden door Ctrl-T opgeven. Met Enter worden alle opdrachten achter elkaar uitgevoerd.

De gehele lijst met opdrachten verwijderen

Met Alt-F7 verwijdert u alle eerder ingevoerde opdrachten uit de buffer.

DOSKEY-macro's

Alhoewel de macrofunctie van DOSKEY niet veel wordt gebruikt, is deze mogelijkheid toch handig voor het toekennen van één of meer opdrachten aan een enkele opdracht. De opdracht FORMAT kan worden toegewezen aan bijvoorbeeld de nieuwe opdracht QFA:

```
format a: /vtechnojock /q
```

Wanneer u QFA (Quick Format A) invoert, wordt door DOSKEY de volledige opdracht vervangen. Dit illustreert de mogelijkheid lange, ingewikkelde opdrachten terug te brengen tot een enkele eenvoudige opdracht. U definieert met de volgende syntaxis de verkorte opdrachtnaam en de opdrachtenset die door DOSKEY wordt uitgevoerd:

```
doskey macronaam=[tekst]
```

Macronaam is een willekeurige naam die u aan de macro geeft en *tekst* is een reeks van één of meer opdrachten. Wanneer u *macronaam* invoert, wordt *tekst* door DOSKEY uitgevoerd.

U maakt het voorafgaande voorbeeld van FORMAT met behulp van de volgende opdracht:

```
doskey qfa=format a: /v:technojock /q
```

U kunt elke naam voor een macro gebruiken, maar er mogen geen spaties in staan of een omleidingsteken, zoals <, > en |.

Een macro wordt net zo verwijderd als een omgevingswaarde wordt gewist. Om een macro te verwijderen start u DOSKEY en geeft u de macro op, gevolgd door het is-gelijk-teken (=). De voorbeeldmacro QFA verwijdert u als volgt:

```
doskey qfa=
```

Na elke start van de computer moet u de macro-opdrachten opnieuw installeren. Bij veel macro's is het handiger een batchbestand van de macro's te maken en dit bestand op te laten roepen door AUTOEXEC.BAT nadat DOSKEY is geïnstalleerd. Verderop in dit hoofdstuk wordt beschreven hoe u een batchbestand kunt maken van de actieve macro's.

DOSKEY kan alleen een volledige opdracht door een macro vervangen. U kunt met DOSKEY geen delen van opdrachten door een macro vervangen. Wanneer u regelmatig het bestand WIN.INI bewerkt, zou u het liefst WIN.INI door de letter W vervangen:

```
doskey w=win.ini
```

Wanneer u een opdracht invoert, wilt u dat DOSKEY de naam WIN.INI door W vervangt:

```
edit w
copy w \winbak
```

Dit klinkt leuk, maar het werkt zo niet! U kunt alleen een volledige opdracht door een macro vervangen. In bovenstaande voorbeelden is het resultaat dat u een bestand W bewerkt of kopieert.

Schakelopties voor de opdrachtregel

Optionele opdrachten of schakelopties kunnen aan batchbestanden worden doorgegeven met behulp van de identificatiecodes `%1` tot en met `%9`. Eenzelfde benadering wordt toegepast bij DOSKEY, behalve dat de identificatiecodes nu `$1` tot en met `$9` zijn (het teken $ wordt in plaats van % gebruikt).

Een volgende macro maakt een VC-opdracht (voor Verify Copy) met toevoeging van een schakeloptie /V aan de opdracht COPY:

```
doskey vc=copy $1 $2 /v
```

Als u nu de opdracht `vc *.* a:` invoert, wordt door DOSKEY de opdracht `copy *.*. a: /v` uitgevoerd.

Behalve de negen parameters heeft DOSKEY nog een andere identificatiecode waarmee de gehele opdrachtregel kan worden benaderd, namelijk `$*`. Wanneer u `$*` specificeert in een opdracht, worden alle schakelopties en parameters, waaronder ook spaties, op de opdrachtregel vervangen. Dit komt vooral van pas als er meer dan negen parameters worden ingevoerd, of wanneer er spaties in de parameter(s) voorkomen, zoals de prompt in de opdracht CHOICE.

In het volgende voorbeeld wordt een macro DF (voor *def*rag) gemaakt, die de opdracht DEFRAG uitvoert met alle gespecificeerde opties, en die DEFRAG met behulp van de schakeloptie /SKIPHIGH in het conventionele geheugen laadt.

```
doskey df=defrag $* /skiphigh
```

Hopelijk dat in een volgende versie van DOS `%*` door batchbestanden wordt ondersteund.

Doorsluizen en omleiden met DOSKEY

Zie hoofdstuk 10 voor meer informatie over beide onderwerpen.

Om de symbolen voor deze bewerkingen in een macro op te nemen zult u speciale tekens moeten gebruiken die de echte omleidingstekens vertegenwoordigen. De volgende opdracht vermeldt de hoeveelheid vrije ruimte in de huidige directory:

```
doskey dsize=dir | find "file(s)"
```

De bedoeling van deze opdracht is om het resultaat van DIR door te sluizen naar FIND. De opdracht FIND zal elke regel weergeven waarop "file(s)" staat (de regel waarop staat hoeveel bytes in beslag worden genomen door alle bestanden).

Als u deze opdracht probeert uit te voeren, is het resultaat anders dan u verwacht. DOS onderschept het teken | en sluist de DOSKEY-opdracht onmiddellijk door naar FIND. Het gevolg is dat DOSKEY alleen de opdracht DIR bij DSIZE opslaat.

De oplossing ligt bij de reeks $b. Deze tekens worden door DOSKEY als het teken | opgevat, maar DOS negeert dit teken als u eerst de opdracht voor de macrodefinitie invoert. De volgende manier is de manier waarop de opdracht moet worden ingevoerd:

```
doskey dsize=dir $b find "file(s)"
```

Er zijn soortgelijke vervangingstekens voor elke omleidingsopdracht (zie tabel 9-2). De tekens kunnen zowel in hoofdletters als in kleine letters worden gebruikt.

Als u het dollarteken ($) in een opdracht wilt gebruiken, moet u twee dollartekens ($$) invoeren. Hierdoor wordt het teken volgend op het dollarteken niet als een vervangingsteken opgevat.

Tabel 9-2 Vervanging voor doorsluis- en omleidingstekens

Reeks	Betekenis
$b	Het equivalent van |, dat de uitvoer van de ene opdracht als de invoer naar een andere opdracht doorsluist.
$g	Het equivalent van >, dat de uitvoer naar een andere opdracht, ander apparaat of bestand omleidt.
$gg	Het equivalent van >, dat de uitvoer aan het einde van een bestand toevoegt.
$l	Het equivalent van <, dat de inhoud van een bestand als invoer naar een opdracht omleidt.

Meerdere opdrachten invoeren

Wanneer DOSKEY is geïnstalleerd, kunt u op de opdrachtregel meerdere opdrachten invoeren. U dient wel achter elke opdracht met Ctrl-T het teken ¶ te plaatsen. Bij meerdere opdrachten in een DOSKEY-macro wordt van hetzelfde principe uitgegaan. In plaats van Ctrl-T worden de vervangingstekens $t gebruikt. De volgende opdracht maakt een macro MCD waarmee een directory wordt gemaakt en de nieuwe directory wordt geselecteerd:

```
doskey mcd=md $1$tcd $1
```

Als u vervolgens de opdracht MCD ingeeft, gevolgd door een pad, wordt de opdracht MD gestart om de directory te maken, waarna met de opdracht CD deze nieuwe directory wordt geselecteerd.

Een macromaker

U kunt de huidige macro's van DOSKEY bekijken met de schakeloptie /MACROS:

```
doskey /macros
```

DOSKEY laat een overzicht zien van alle geïnstalleerde macros:

```
DSIZE=dir $b find "file(s)"
?=help $*
MCD=md $1$tcd $1
QFA=format a: /v:technojock /q
VC=copy $1 $2 /v
DF=defrag $* /skiphigh
```

Na elke herstart van de computer of start van DOSKEY met de schakeloptie /REINSTALL zijn de actieve macro's gewist. Met de schakeloptie /MACROS kunt u de bestaande macro's opslaan, zodat u ze makkelijk opnieuw kunt installeren.

Door de uitvoer naar een batchbestand te sluizen hebt u het raamwerk van een macroconstructie. De volgende opdracht maakt het batchbestand MACROS.BAT:

```
doskey /macros > macros.bat
```

Het batchbestand moet worden aangepast om de DOSKEY-macros te kunnen laden. Elke regel moet beginnen met het woord `doskey` (gevolgd door een spatie). U kunt de wijzigingen zelf aanbrengen

of de computer het werk laten doen. Zij zijn tenslotte niet voor niets ontworpen.

DOSKEY-macro's versus batchbestanden

De macrofunctie lijkt in veel opzichten op batchbestanden. Bij beide technieken wordt een serie opdrachten bij elkaar gebracht die met een enkele opdracht worden uitgevoerd. Bovendien kunnen bij beide technieken parameters worden gebruikt.

In onderstaand overzicht worden de belangrijkste verschillen opgesomd tussen batchbestanden en macro's:

- In tegenstelling tot batchbestanden worden macro's in het geheugen opgeslagen en niet in een bestand. Hierdoor kunnen macro's overal worden gestart.

- Zelfs het kleinste batchbestand neemt minstens één cluster in beslag (gewoonlijk 4096 bytes). Macro's nemen geen schijfruimte in beslag.

- Macro's zijn veel sneller dan batchbestanden omdat zij in het geheugen worden geladen.

- Een macro kan een andere macro niet oproepen, terwijl een batchbestand wel een ander batchbestand kan starten.

- Een macro kan een batchbestand starten, maar een batchbestand kan geen macro-opdracht uitvoeren. DOSKEY kan alleen die opdrachten benaderen die rechtstreeks op de opdrachtregel worden ingevoerd.

- De uitvoer van macro-opdrachten kunt u niet op het scherm onderdrukken, zoals in batchbestand met `echo` wel mogelijk is. De gebruiker ziet welke opdrachten er worden uitgevoerd.

- DOSKEY ondersteunt de opdrachten IF en GOTO niet, dus een voorwaarde kan niet worden opgenomen.

- Macro's kunnen geen omgevingsvariabelen benaderen, zoals bijvoorbeeld het pad met de expressie `%path%`.

- Bij beide technieken kunnen negen parameters worden benaderd, maar macro's kunnen de volledige opdrachtregel benaderen met `$*`; hiervoor is geen equivalent bij batchbestanden.

- De opdrachtnamen van macro's kunnen gelijk zijn aan de interne opdrachten van DOS en aan programmabestanden (met een extensie EXE of COM). De macro-opdracht wordt altijd uitgevoerd voor de DOS-opdracht of het interne programma met dezelfde naam. Batchbestanden worden echter genegeerd als zij dezelfde naam hebben als een interne DOS-opdracht of een ander programma in de huidige directory.

- Een macro kan niet langer zijn dan 127 tekens, terwijl er wat dat betreft geen beperkingen zijn voor een batchbestand.

Mijn favoriete trucs met DOSKEY-macro's

DOSKEY-macro's kunnen erg handig zijn en de produktiviteit en veiligheid vergroten, als u de tijd neemt enkele goede macro's te schrijven. In de volgende paragrafen worden enkele van mijn favoriete macro's beschreven.

Vraagteken-hulp

Deze macro is niet echt wereldschokkend, maar het is sneller een vraagteken in te voeren dan de opdracht HELP. De volgende macro laat zien wat daarvoor moet worden gedaan:

```
doskey ?=help $*
```

Wanneer u ? invoert, wordt de helpmodule van DOS gestart.

DOS-opdrachten vervangen, uitbreiden en uitschakelen

Een macro-opdracht van DOSKEY gaat voor alle andere opdrachten, zowel interne DOS-opdrachten als externe programma's. Dat betekent dat een macro-opdracht altijd wordt uitgevoerd. Hierdoor kunt u geselecteerde bestanden uitschakelen of extra opdrachtschakelopties verwerken.

De volgende macro-opdracht schakelt de opdracht FORMAT uit:

```
doskey format=Er kan niet worden
geformatteerd.
```

Wanneer u de opdracht FORMAT invoert, wordt door DOSKEY de opdracht `Er kan niet worden geformatteerd` uitgevoerd in plaats van de eigenlijke formatteringsopdracht. De opdracht wordt op het scherm weergegeven en DOS antwoordt met `bad command or file name`.

De volgende opdracht voegt de schakeloptie /P toe aan de opdracht DEL, zodat er toestemming moet worden gegeven als er een bestand wordt verwijderd:

```
doskey del=del $1 /p
```

Met behulp van eenzelfde techniek wordt door de volgende macro een opdracht gemaakt die COPY vervangt door XCOPY:

```
doskey copy=xcopy $*
```

De meeste macro's die aan bod zijn gekomen, zijn hulpmiddelen waarmee computers voor andere, minder bedreven gebruikers kunnen worden geconfigureerd. Een goede configuratie kan ze behoeden voor fouten met vervelende gevolgen. Het zal niet de eerste keer zijn dat een onervaren gebruiker de vaste schijf formatteert. Deze DOSKEY-macro's zijn vanuit AUTOEXEC.BAT te laden. De DOSKEY-opdracht zou u zelfs ook kunnen uitschakelen, zodat er niet kan worden gerommeld:

```
doskey doskey=Deze opdracht is niet
beschikbaar.
```

Het is natuurlijk wel duidelijk dat de beschreven technieken niet veel bescherming bieden, zeker niet tegen een ervaren DOS-gebruiker. Met Alt-F10 kunnen alle macro's worden gewist. Maar bij een beginnende computergebruiker biedt het voldoende bescherming.

Verandering van station onderscheppen

Mijn desktop computer heeft twee diskettestations, station A voor 1,2MB diskettes en station B voor 1,44MB diskettes. Ik ben geprogrammeerd op station B voor 3,5 inch diskettes. Wanneer ik met mijn notebook werk, raak ik steevast in de war.

Daar is station A voor 3,5 inch diskettes. Met een macro kunnen de stationsletters eenvoudig worden omgewisseld:

```
doskey b:=a:
```

Wanneer ik op mijn notebook b: invoer, wordt dit door DOSKEY onderschept en wordt de opdracht a: gegeven. Om te voorkomen dat ik steeds de verkeerde letter intyp, heb ik een macro geschreven die een geluid geeft als ik de verkeerde letter invoer:

```
doskey b:=echo ^G $ta:
```

U moet de reeks `^G` niet als twee aparte tekens invoeren, maar als Ctrl-G.

Deze macro produceert eerst een geluidssignaal en schakelt daarna over naar a:. Dit ging mij op den duur irriteren, omdat ik toch de verkeerde letter bleef intypen (het hondje van Pavlov). Het geluidssignaal is er dus uitgehaald.

U kunt hetzelfde principe toepassen bij een netwerkstation. De volgende macro controleert of station F bestaat. Als station F niet aanwezig is, wordt er een inlog-opdracht gegeven voordat er wordt overgeschakeld naar station F. In het andere geval wordt de opdracht F: gewoon uitgevoerd.

```
doskey f:=if not exist f: net
logon$tf:
```

Het concept om met IF en EXIST te controleren of een directory bestaat, wordt besproken in hoofdstuk 10.

Meerdere bestandsnamen ondersteunen

Enkele van de meest gebruikte opdrachten ondersteunen slechts één bestandsmasker (Del). Door een FOR-lus in een macro kunt u dergelijke opdrachten geschikt maken voor meerdere bestandsmaskers, zoals het volgende voorbeeld laat zien:

```
doskey delall=for %n in ($*) do del
%n
```

Wanneer u meerdere bestandsnamen of -maskers opgeeft bij de opdracht DELALL, wordt de opdracht DEL voor elke afzonderlijke parameter opgeroepen. Om alle bestanden met de extensie

DEL, BAK, TMP en $$$ te verwijderen voert u de volgende opdracht in:

```
dellal *.del *.bak *.tmp *.$$$
```

De volgende macro maakt gebruikt van dezelfde techniek om meerdere bestandsmaskers te kunnen kopiëren met COPY:

```
doskey copyto=for %f in ($2 $3 $4 $5
$6 $7 $8 $9) do copy %f $1
```

De opdracht COPYTO accepteert de doeldirectory als het eerste argument; de resterende argumenten zijn de bronbestandsmaskers. Met een dergelijke opdracht kopieert u een groep bestanden naar station a:

```
copyto a: *.doc *.tif *.txt
\wks\*.wk3
```

Het zoekpad uitbreiden

Lange paden zijn vervelend; de omgeving wordt in beslag genomen en de snelheid wordt lager omdat DOS in het pad van elke directory naar een opdracht moet zoeken. Wanneer u een directory alleen in het pad hebt staan om een bepaalde opdracht vanuit een andere directory te kunnen starten, moet u toch de volgende mogelijkheid eens bekijken. U kunt een macro schrijven waarmee een opdracht overal kan worden gestart:

```
doskey cmfiler=c:\utils\cm\cmfiler
```

Als u de opdracht CMFILER invoert, wordt het programma geladen vanuit de directory c:\utils\cm. Hierbij maakt het niet uit welk station en welke directory er actief zijn.

Nog enkele snelle macro's als toetje

De volgende macro's zijn voorbeelden van macro-opdrachten die bestaan uit één teken.

Alleen met de backslash \ kunt u naar de hoofddirectory. De volgend macro maakt dit mogelijk:

```
doskey \=cd\
```

Vaak moet er iets in de bestanden CONFIG.SYS en AUTOEXEC.BAT worden veranderd. Met een

enkele letter wordt EDIT gestart met het desbetreffende bestand geopend:

```
doskey c=edit c:\config.sys
doskey a=edit c:\autoexec.bat
```

Samenvatting

In dit hoofdstuk is de veelzijdigheid van DOSKEY belicht. Hierbij zijn de volgende punten aan de orde gekomen:

- Behalve basisbewerkingen kunnen eerder ingevoerde opdrachten met de pijl-omhoog en de pijl-omlaag worden opgehaald.
- Met F7 wordt er een lijst met eerder ingevoerde opdrachten opgeroepen en met F9 kan een 'oude' opdracht worden gestart.
- Er kunnen meerdere opdrachten achter elkaar worden ingegeven, zolang de opdrachten worden gescheiden door Ctrl-T.
- Eén of meer opdrachten kunnen worden opgeslagen als een DOSKEY-macro en worden uitgevoerd door het invoeren van de macronaam. Een DOSKEY-macro kan ook bestaande opdrachten vervangen of uitschakelen.
- Macro's kunnen door middel van de expressies $1 tot $9 opdrachtparameters benaderen. De speciale tekenreeks $* vertegenwoordigt de volledige parameterreeks op de opdrachtregel.
- DOSKEY slaat de ingevoerde opdrachten en macro's op in een buffer. Deze buffer is 512 bytes groot, maar kan met de schakeloptie /BUFSIZE worden veranderd.
- DOSKEY-macro's zijn sneller dan batchbestanden en nemen geen schijfruimte in beslag. Batchbestanden kunnen echter meer opdrachten uitvoeren, hebben toegang tot omgevingsvariabelen en ondersteunen voorwaardelijke vertakkingen.

In menig opzicht gaat een batchbestand verder dan DOSKEY. In hoofdstuk 10 worden geavanceerde technieken met betrekking tot batchbestanden besproken.