

## Hoofdstuk 6

---

# Weergave en toetsenbord regelen

---

---

*In dit hoofdstuk*

- > Een kleurrijker DOS
  - > Technieken voor ANSI.SYS
  - > Schermweergave verbeteren
  - > Hoe het toetsenbord aan te passen
  - > Alternatieven voor ANSI.SYS
- 
- 

## Wij willen een kleurentelevisie!

In de zeventiger jaren zetten mijn broer en ik mijn vader onder druk. Elke avond voor het slapen gaan zeurden wij hem aan zijn hoofd voor een kleurentelevisie. Uiteindelijk stemde hij toe en kocht hij er een. De ontwikkelaars van DOS kregen blijkbaar een zwart-wit opvoeding, want laten wij eerlijk zijn: DOS is niet echt een kleurenpracht.

Wanneer u zit te wachten op een kleurenversie van DOS, zult u misschien verbaasd zijn te lezen dat u DOS al in kleur hebt. Met het stuurprogramma ANSI.SYS kunt u heel wat kleur en fleur aan uw DOS-sessie toevoegen. Naast de ondersteuning voor kleur kan ANSI.SYS de cursor een andere plaats geven en het toetsenbord aanpassen. U kunt bijvoorbeeld een serie toetsen aan functietoetsen toewijzen. Ondanks dat ANSI.SYS krachtig is, kost het maar 5K geheugen.

## Technieken voor ANSI.SYS

Om met ANSI.SYS te kunnen werken zult u het eerst moeten installeren in CONFIG.SYS. U kunt dit stuurprogramma zowel met DEVICE als met DEVICEHIGH laden.

ANSI.SYS ondersteunt twee schakelopties, die beide betrekking hebben op het uitgebreide (101 toetsen) toetsenbord. Op dergelijke toetsenborden kunt u met meerdere toetsen bepaalde bewerkingen uitvoeren. Er zijn bijvoorbeeld twee toetsen voor het plusteken en twee toetsen voor PgUp. Door de schakeloptie /X kan ANSI.SYS deze toetsen

onafhankelijk behandelen. Met deze schakeloptie kunt u de functie van de toets wijzigen, zodat elke toets zijn eigen, specifieke functie heeft.

Wanneer u een niet-standaard uitgebreid toetsenbord hebt en het opnieuw toewijzen van toetsen verloopt niet zonder problemen, kunt u met de schakeloptie /K opgeven dat het toetsenbord als een toetsenbord met 84 toetsen moet worden beschouwd. De schakelopties /X en /K kunnen niet tegelijkertijd worden toegepast.



Wanneer u /K opgeeft in SWITCHES in CONFIG.SYS, laad dan ANSI.SYS met de schakeloptie /K. Gebruikt u de schakeloptie /K bij SWITCHES, dan moet u dit ook doen bij ANSI.SYS.

De volgende opdracht in CONFIG.SYS laadt ANSI.SYS in het hogere geheugen en specificeert afzonderlijke toetsondersteuning op uitgebreide toetsenborden:

```
devicehigh=c:\dos\ansi.sys /x
```

## Instructies voor ANSI.SYS

Wanneer u de kleuren of de plaats van de cursor wilt veranderen of de toetsen aan andere functies wilt toewijzen, zult u dit via ANSI.SYS moeten doen. U laat ANSI.SYS speciale escapecodes sturen naar het standaard uitvoerapparaat (het scherm).

De escapecodes beginnen alle met twee specifieke tekens: het Esc-teken, dat een ASCII-waarde heeft van 27, en het teken van het linker vierkante haakje ([), dat een ASCII-waarde heeft van 019. De tekens die achter deze specifieke tekens worden geplaatst, zijn afhankelijk van de taak die u wilt uitvoeren; de tekenreeks 33m bijvoorbeeld kleurt tekst bruin.

Door ANSI.SYS worden tekens onderschept die naar het standaard uitvoerapparaat (het scherm) worden gestuurd, maar doet zelf niets totdat het de specifieke tekens ontvangt. Wanneer deze tekens worden ontvangen, komt ANSI.SYS tot leven en reageert op de instructies die volgen.

De DOS-opdrachten ECHO, PROMPT en TYPE sturen alle tekens naar het standaard uitvoerapparaat, zodat u de specifieke tekens er mee

naar ANSI.SYS kunt zenden. U dient echter wel met enkele punten rekening te houden:

Stel dat u de voorgrondkleur in bruin wilt veranderen. U denkt misschien dat de volgende opdracht hiervoor voldoende is:

**Echo Esc[33m**



In de voorbeelden stelt Esc de Esc-toets voor (ASCII-code 27); voer de letterlijke tekens *E*, *s* en *c* niet in.

Het is niet zo eenvoudig als het lijkt. Wanneer u op Esc drukt, denkt DOS dat u de gedeeltelijk ingevoerde opdracht wilt annuleren, en dus wordt de opdracht gewist voordat u een kans krijgt de tekens [33m in te voeren. In veel situaties is het ingedrukt houden van de Alt-toets en het invoeren van de ASCII-waarde (drie cijfers) een andere manier om een speciale toets in te voeren. Wanneer u deze techniek op de opdrachtregel gebruikt, gaat DOS ervan uit dat u de Esc-toets direct hebt ingedrukt. De opdracht wordt geannuleerd.

Dit probleem is als volgt op te lossen.

### Met batchbestanden werken

Een manier om het aangehaalde probleem te omzeilen is een batchbestand te maken waarin de opdracht ECHO staat. De volgende opdracht maakt de tekst bruin:

```
@echo off
echo ←[33m
```



Door deze benadering doet zich een ander probleem voor. Wanneer u de Esc-toets in EDIT indrukt, gebeurt er niets. Hetzelfde geldt voor de Alt-toets met drie cijfers. Door middel van een eenvoudige truc kunt u dit verhelpen. In EDIT kunt u een speciaal teken invoeren door op Ctrl-P te drukken, vervolgens Alt ingedrukt te houden en een ASCII-code (2 of 3 cijfers) in te voeren. Het invoeren van Esc is zelfs makkelijker: druk op Ctrl-P en vervolgens op Esc.

Het Esc-teken wordt in EDIT als een pijl naar links (←) weergegeven.

Nadat u een batchbestand hebt gemaakt met daarin de ANSI.SYS escapetekens, kunt u het bestand starten en ANSI.SYS reageert. Wanneer u de

bovenstaande opdracht hebt ingevoerd, wordt de tekst bruin weergegeven.



Eén van de speciale escapetekens van ANSI.SYS is Esc[2J, dat het scherm schoonmaakt en de cursor naar de basisplaats terugbrengt. (De opdracht Esc[K wist de regel te beginnen bij de cursor.)

Zie hoofdstuk 10 voor meer informatie over batchbestanden.

### De opdracht TYPE

In plaats van met de opdracht ECHO in batchbestanden, kunt u de opdrachten ook in platte tekst invoeren en met TYPE de opdrachten naar het scherm schrijven. U kunt bijvoorbeeld met EDIT een tekstbestand van één regel maken, BROWN.COL, waarin de volgende tekens staan:

```
←[ 33m
```

Door vervolgens de volgende opdracht in te voeren kunt u de escapetekens naar ANSI.SYS omleiden:

```
type brown.col
```

### De opdracht PROMPT

Zoals EDIT en TYPE stuurt PROMPT tekens naar het scherm, zodat u door middel van deze opdracht met ANSI.SYS kunt communiceren. Eén van de speciale metatekens die door PROMPT worden ondersteund, is \$e dat het Esc-teken weergeeft.

Zie de informatie over PROMPT in deel 4 *Alle DOS-opdrachten* voor een volledige lijst met alle metatekens die door PROMPT worden ondersteund.

Met het eerder besproken voorbeeld kunt u de tekst bruin maken door de volgende opdracht in te voeren:

```
prompt $e[33m
```

Alhoewel de opdrachten foutloos door ANSI.SYS worden onderschept, heeft deze benadering een vervelend nadeel: de systeemprompt wordt veranderd, gewoonlijk in een nulprompt (zonder teken). Na met PROMPT één of meer instructies naar ANSI.SYS te hebben gestuurd, moet u PROMPT opnieuw oproepen om de DOS-prompt terug te halen. het volgende batchbestand,

ANSIPRMT.BAT maakt het leven er een stuk makkelijker op. Het slaat de oude prompt in een batchbestand op en herstelt deze nadat de ANSI.SYS-instructies zijn uitgevoerd.

```
@echo off
set|find "PROMPT" > oldprmt.bat
:LOOP
if "%1"==" " goto stop
prompt $e[%1
off
shift
goto loop
:STOP
call oldprmt
del oldprmt.bat
```

Maak u geen zorgen als enkele opdrachten in dit batchbestand abacadabra voor u zijn; alles wordt uitgelegd in hoofdstuk 10.



Wanneer u de standaard systeemprompt kent, kunt u alles vereenvoudigen door de instellingen van de standaardprompt aan het einde van de ANSI.SYS-opdrachten toe te voegen. Als u gewoonlijk de prompt \$p\$g gebruikt dat de huidige directory als de prompt weergeeft, kunt u met één

## Hoofdstuk 6: Weergave en toetsenbord regelen

opdracht bruin als kleur instellen en de oorspronkelijke prompt weer terughalen.

```
prompt $e[33m$p$g
```

### Meerdere escapetekens sturen

U hoeft niet elke ANSI.SYS-opdracht op een nieuwe regel te plaatsen. Er kunnen meerdere instructies op één regel worden ingevoerd als elke instructie door de standaardreeks Esc[ wordt voorafgegaan.

Nadat de tekst in bruin is veranderd, kunt u met de volgende opdracht 2J het scherm schoonmaken:

```
@echo off
echo ←[33m
echo ←[2
```

Deze twee opdrachten zijn ook tot één opdracht terug te brengen:

```
@echo off
echo ←[33m←[2J
```



Bij de ANSI.SYS-opdrachten wordt onderscheid gemaakt tussen hoofdletters en kleine letters. Als u 2J invoert als 2j, wordt de opdracht niet uitgevoerd, dat wil zeggen wordt het scherm niet schoongemaakt.

## Bepalen of ANSI.SYS is geladen

Als ANSI.SYS niet is geïnstalleerd, reageert DOS niet bepaald galant op al deze escapetekens. Alhoewel er geen ongewenste dingen gebeuren, worden de onbegrijpelijke escapetekens naar het scherm geschreven, hetgeen er niet erg professioneel uitziet. Een geavanceerd scherm dat met ANSI.SYS kleuren verandert, de cursor herplaatst en toetsen opnieuw definieert, ziet er afschuwelijk uit als ANSI.SYS niet is geïnstalleerd. Wanneer u een batchbestand maakt dat gebruik maakt van ANSI.SYS, moet u eerst controleren of ANSI.SYS wel is geladen.



Bij oudere versies van DOS moest er met DEBUG of een andere programmeertaal een programma worden geschreven om te controleren of ANSI.SYS was geïnstalleerd in het geheugen. Dankzij de schakeloptie MEM /M is dit niet langer nodig.

DOS geeft aan waar ANSI.SYS in het geheugen is geïnstalleerd als u de volgende opdracht ingeeft:

```
mem /m ansi
```

Wanneer ANSI.SYS niet is geïnstalleerd, ziet u het volgende bericht:

```
ansi is not currently in memory
```

Met FIND kunt u de uitvoer "not currently" opzoeken en vervolgens het ERRORLEVEL van FIND testen om te bepalen of ANSI.SYS is geïnstalleerd.

Het volgende batchbestand, ANSIOK.BAT, laat een bericht zien dat aangeeft of ANSI.SYS is geïnstalleerd:

```
@echo off
mem /m ansi|find "not currently" /i > nul
if errorlevel 1 goto loaded
:NOTLOADED
echo ANSI.SYS staat niet in het geheugen
goto quit
:LOADED
echo ANSI.SYS is geladen
rem Voeg uw ANSI-opdrachten hier in
:QUIT
```

U kunt tegelijk met MEM /M en FIND testen of een bepaald programma, zoals Share, Windows of DOSKEY, in het geheugen is geladen. Dit wordt uitvoerig besproken in hoofdstuk 10.

Het bestand ANSIOK.BAT is uit te breiden door het toevoegen van een test waarmee wordt bepaald of ANSI.SYS is geladen:

```
@echo off
mem /m ansi|find "not currently" /i > nul
if errorlevel 1 goto loaded
:NOTLOADED
Echo ANSI.SYS staat niet in het geheugen
goto quit
:LOADED
:LOOP
if "%1"==" " goto quit
echo ←[%1
shift
goto loop
:QUIT
```

## Het scherm opfleuren

U hebt in het voorafgaande gedeelte een aantal manieren geleerd om de tekst bruin weer te geven.

## Hoofdstuk 6: Weergave en toetsenbord regelen

U zult ongetwijfeld verheugd zijn om te vernemen dat u nog veel meer kunt doen. U kunt de kleuren van de achtergrond en voorgrond instellen, tekens op iedere plaats op het scherm schrijven en de tekst vet en omgekeerd weergeven en laten knipperen.

### Voorgrond- en achtergrondkleuren

De kleur van de voor- en achtergrond kan worden ingesteld met de opdrachtset m. De syntaxis van deze opdracht is als volgt:

```
Esc[ codem
```

De variabele *code* vervangt u door een getal van twee cijfers voor de gewenste kleur. In tabel 6-1 staan alle codes voor de kleuren.

Het volgende batchbestand geeft de text in cyaan weer op een blauwe achtergrond

```
@echo off
echo ←[36m
echo ←[44m
```

Meerdere m-opdrachten kunnen in één instructie worden samengebracht waarvoor u de volgende syntaxis moet hanteren:

```
Esc[ code;code;...m
```

Door de volgende regel uit een batchbestand wordt de tekst zwart weergegeven op een groene achtergrond:

Tabel 6-1 Schermcodes ANSI.SYS		
<i>Kleur</i>	<i>Voorggrondcode</i>	<i>Achtergrondcode</i>
Zwart	30	40
Rood	31	41
Groen	32	42
Bruin	33	43
Blauw	34	44
Magenta	35	45
Cyaan	36	46
Lichtgrijs	37	47

```
echo ←[30;42m
```

## Schermattributen instellen

De opdracht m van ANSI.SYS heeft nog een aantal codes van één cijfer voor het veranderen van de tekstattribuut van tekst. In tabel 6.2 staan de geldige codes. De meest gebruikte attribuutcode is 1, waarmee de saaie kleuren in heldere kleuren worden veranderd. Bruin wordt geel, lichtgrijs wordt helder wit, enzovoorts.

Het volgende batchbestand, ANSIATTR.BAT, laat elk attribuuteffect zien:

```
@echo off
echo ←[2J
echo ←[0m Alle attributen uit
echo ←[0;1m Helderheid
echo ←[0;4m Onderstreept (alleen op
                    monochroom schermen)
echo ←[0;5m Knipperend
echo ←[0;7m Omgekeerde weergave
echo ←[0;8m Verborgen of onzichtbaar
echo ←[1;33m
```





De attribuut 8m verbergt tekst en de cursor. Als 8m de laatste opdracht in een batchbestand is, worden de systeemprompt en de cursor onzichtbaar als het batchbestand is afgelopen. Dit kan in bepaalde omstandigheden een handigheidje zijn, bijvoorbeeld als u bepaalde zaken van een programma of batchbestand niet wilt laten zien. Het bovenstaande batchbestand stelt echter als laatste de tekstkleur in op geel (lichtbruin).

Tabel 6-2 Codes schermattribuut ANSI.SYS	
Code	Attribuut
0	Alle attributen uit
1	Helder
4	Onderstreept (alleen op monochroom monitoren)
5	Knipperend
7	Omgekeerde weergave
8	Verborgen of onzichtbaar

## Plaats van de cursor bepalen

In elke programmeertaal voor de computer zit een mogelijkheid om tekst in een bepaalde kleur ergens op het scherm weer te geven. Normaal gesproken wordt de cursor op de regel geplaatst die volgt op een ECHO-opdracht, waardoor tekst op regels boven de cursor kan worden geschreven zonder dat het scherm eerst wordt schoongemaakt. Dit is alleen mogelijk als ANSI.SYS is geïnstalleerd.

De opdracht H uit ANSI.SYS ondersteunt de plaatsbepaling van de cursor op het scherm. Nadat de cursor ergens is geplaatst, wordt tekst die u invoert met de opdracht ECHO, weergegeven bij de cursorpositie.

De syntaxis van de opdracht H is als volgt:

```
Esc[ row; colH
```

De variabele *row* is het nummer van de rij of regel, te beginnen met 1 boven aan het scherm en eindigend met 25, 43 of 50 onder aan het scherm. De parameter *col* is de kolom of de x-coördinaat, te beginnen met 1 links van het scherm en eindigend met 40 of 80 rechts van het scherm.

Het volgende batchbestand schrijft tekst rechtsboven aan het scherm:

```
@echo off
echo ←[1;59HSouthern Translations
```

Het enige probleem met dit batchbestand is, dat de cursor boven aan het scherm blijft staan. Nadat het batchbestand is afgelopen, staat de DOS-prompt boven aan het scherm ergens te midden van oude opdrachten. Niet echt fraai.

Twee opdrachten van ANSI.SYS kunnen dit probleem verhelpen. De opdracht s slaat de positie van de cursor op; de opdracht u zet de cursor weer terug op de plaats die de cursor innam toen de laatste opdracht s werd uitgevoerd.

Het volgende batchbestand WRITEAT.BAT geeft een algemene routine waarmee tekst op elke plaats op het scherm kan worden geschreven, en brengt daarna de cursor weer terug naar de vorige positie.

```
@echo off
if "%1"==" " goto quit
echo ←[s
echo ←[%1;%2H%3 %4 %5 %6 %7 %8 %9
echo ←[u
:QUIT
```

Naast de opdrachten voor het plaatsen, opslaan en terughalen kan met ANSI.SYS ook de verplaatsing van de cursor in rijen en kolommen worden geregeld, uitgaande van de huidige cursorpositie. In tabel 6.3 staat een overzicht van alle cursoropdrachten van ANSI.SYS.

**Tabel 6-3 ANSI.SYS-cursorbesturingscodes**

Escapereeks	Omschrijving	Handeling
Esc[rij;kolomH	Absoluut	Verplaatst de cursor naar de opgegeven rij en kolom. Wanneer er slechts een waarde is opgegeven zonder puntkomma, wordt ervan uitgegaan dat een kolom is opgegeven en dat de rij de eerste rij is.
Esc[s	Opslaan	Slaat de actieve cursorpositie op.

Esc[u	Herstellen	Zet de cursor terug op de plaats die werd ingenomen voordat de laatste opdracht s (save) werd uitgevoerd.
Esc[posA	Omhoog	Verplaatst de cursor een aantal rijen omhoog aangegeven door de variabele <i>pos</i> . De cursor stopt bij de eerste rij.
Esc[posB	Omlaag	Verplaatst de cursor een aantal rijen omlaag. De cursor stopt bij de laatste rij.
Esc[posC	Naar rechts	Verplaatst de cursor <i>pos</i> kolommen naar rechts. De cursor stopt bij de laatste kolom.
Esc[posD	Naar links	Verplaatst de cursor <i>pos</i> kolommen naar links. De cursor stopt bij de eerste kolom
<b>Opmerking:</b> De opdracht f heeft dezelfde betekenis en syntaxis als H voor het verplaatsen van de cursor naar een bepaalde lokatie.		

## Aantrekkelijke prompts maken

Met de escapecodes van ANSI.SYS kunt u de systeemprompt een kleur geven. Ik acht me als inwoner van Houston verplicht de Texas Lone Star vlag als prompt te gebruiken.

De volgende prompt-opdracht is volgens mij een must in Texas:

```
@echo off
PROMPT $e[0;44m*$e[41m$b$e[40;36m $p$g $e[1;33;40m
```

De Californische beer is niet zo eenvoudig!

De liefhebbers van Bill the Cat zullen de volgende prompt zeker kunnen waarderen:

```
@echo off
prompt $e[6C/$b$_ \'.0'$_ $q(____)$q$_$e[1;5;31mU$e[0;37m$
_I EK!PFT!$_$P$G
```

## Het aantal schermregels veranderen

Steeds meer computers worden uitgerust met EGA- en VGA-monitoren. Daarom heeft Microsoft vanaf DOS 5 ANSI.SYS uitgebreid, omdat deze

monitoren meer regels per scherm kunnen weergeven, namelijk 43 en 50 regels per scherm.

De computer kan zo worden ingesteld dat DOS in Condensed Mode kan draaien. Start hiervoor MODE, vooropgesteld dat ANSI.SYS is geïnstalleerd. De syntaxis voor de opdracht MODE voor het veranderen van het aantal schermregels is als volgt:

```
mode con: lines:=xx
```

De parameter *xx* stelt het aantal regels per scherm voor en kan 25, 43 of op VGA-systemen 50 zijn.

Het volgende batchbestand SET43.BAT stelt de schermweergave op 43 regels in:

```
@echo off
mem /m ansi|find "Not currently" /i > /nul
if errorlevel 1 goto loaded
:NOTLOADED
echo ANSI.SYS staat niet in het geheugen
goto quit
:LOADED
mode con lines=43
:QUIT
```

U kunt met de opdracht MODE het aantal tekens tussen 40 en 80 per regel instellen. Zie deel 4 *Alle DOS-opdrachten* voor nadere bijzonderheden.

De vreemde eend in de bijt is =7h, waarmee de regeloverbrenging wordt geactiveerd. Deze opdracht is de tegenhanger van =7l (kleine letter L) waarmee deze optie wordt uitgezet.

Wanneer de regeloverbrenging niet actief is, wordt tekst die van het scherm afloopt, niet weergegeven. De tekst wordt aan het einde van de regel automatisch op de volgende regel geplaatst als deze optie wel is geselecteerd. Het volgende batchbestand LINEWRAP.BAT geeft een voorbeeld van deze opdrachten:

```
@echo off
echo ←[1;37;42m←[2J
echo ←[=7l
echo ←[2;70HDit past niet meer
echo ←[=7h
echo ←[5;70HDit past niet meer
echo ←20;1H
```

Tabel 6-4 Besturingscodes schermmodi ANSI.SYS	
Escape-tekens	Beschrijving
Esc[=0h	Tekst - monochroom, 40 kolommen bij 25 regels
Esc[=1h	Tekst - kleur, 40 kolommen bij 25 regels
Esc[=2h	Tekst - monochroom, 80 kolommen bij 25 regels
Esc[=3h	Tekst - kleur, 80 kolommen bij 25 regels
Esc[=4h	Grafisch - 4 kleuren, 320 bij 200 pixels
Esc[=5h	Grafisch - monochroom, 320 bij 200 pixels
Esc[=6h	Grafisch - monochroom, 640 bij 200 pixels
Esc[=7h	Activeert regeloverbrenging
Esc[=13h	Grafisch - kleur, 320 bij 200 pixels
Esc[=14h	Grafisch - 16 kleuren, 640 bij 200 pixels
Esc[=15h	Grafisch - monochroom, 640 bij 350 pixels
Esc[=16h	Grafisch - 16 kleuren, 640 bij 350 pixels

Esc[=17h	Grafisch - monochroom, 640 bij 480 pixels
Esc[=18h	Grafisch - 16 kleuren, 640 bij 480 pixels
Esc[=19h	Grafisch - 256 kleuren, 320 bij 200 pixels

Als regeloverbrenging niet actief is, gebeurt er iets vreemds. De tekens die niet meer op het scherm passen, worden allemaal op de meest rechtse positie geschreven. Het laatste teken van de zin Dit past nr, de letter *r*, is eigenlijk de laatste letter van de hele zin, dus de *r* van *meer*.

## Toetsenbord aanpassen

Eén van de meest waardevolle functies van ANSI.SYS is de mogelijkheid toetsen opnieuw te definiëren en opdrachten (of reeksen) aan toetsen toe te wijzen. U kunt bijvoorbeeld met ANSI.SYS de toetsen voor de dubbele punt en puntkomma omwisselen, de opdracht HELP aan de toets F1 toewijzen, de opdracht EDIT aan F2, enzovoorts.

## Toetscodes

De sleutel tot het kunnen aanpassen van het toetsenbord met ANSI.SYS zijn de scancodes voor elke toets. Elke toetsaanslag die u invoert, wordt voorgesteld door een *scancode* van een of twee cijfers. De hoofdletter J bijvoorbeeld heeft een scancode (één byte) met de waarde van 74; de toets F10 heeft een scancode (twee bytes) van 0 en 69. Aan de meeste speciale of niet-alfanumerieke toetsen worden scancodes van twee bytes toegekend, met 0 als de waarde van de eerste byte

Wanneer u met ANSI.SYS het toetsenbord beheert, moet u de toets opgeven die u wilt herdefiniëren of wijzigen. Als u codes van twee bytes opgeeft, moet u elke byte door een puntkomma scheiden. U zult opmerken dat toetscombinaties soms dezelfde scancodes hebben als bepaalde, enkele toetsen. Ctrl-H is gelijk aan de toets Backspace. Voer een opdracht in bij de DOS-prompt en druk op Ctrl-H. Het teken links van de cursor wordt verwijderd, alsof u op de toets Backspace hebt gedrukt. Bepaalde toetscombinaties worden niet ondersteund, bijvoorbeeld de toetscombinatie Shift-Enter.

Tabel 6-5		ANSI.SYS-toetsenbordcodes		
Toets	Normaal	Met Shift	Met Ctrl	Met Alt
A	97	65	1	0;30
B	98	66	2	0;48
C	99	67	3	0;46
D	100	68	4	0;32
E	101	69	5	0;18
F	102	70	6	0;33
G	103	71	7	0;34
H	104	72	8	0;35
I	105	73	9	0;23
J	106	74	10	0;36
K	107	75	11	0;37
L	108	76	12	0;38
M	109	77	13	0;50

N	110	78	14	0;49
O	111	79	15	0;24
P	112	80	16	0;25
Q	113	81	17	0;16
R	114	82	18	0;19
S	115	83	19	0;31
T	116	84	20	0;20
U	117	85	21	0;22
V	118	86	22	0;47
W	119	87	23	0;17
X	120	88	24	0;45
Y	121	89	25	0;21
Z	122	90	26	0;44
1	49	33	nvt	0;120
2	50	64	0	0;121
3	51	35	nvt	0;122

4	52	36	nvt	0;123
5	53	37	nvt	0;124
6	54	94	30	0;125
7	55	38	nvt	0;126
8	56	42	nvt	0;126
9	57	40	nvt	0;127
0	48	41	nvt	0;129
-	45	95	31	0;130
=	61	43	nvt	0;131
[	91	123	27	0;26
]	93	125	29	0;27
spatie	92	124	28	0;43
;	59	58	nvt	0;39
'	39	34	nvt	0;40
komma	44	60	nvt	0;51
.	46	62	nvt	0;52

/	47	63	nvt	0;53
'	96	126	nvt	0;41
Enter (numpad)	13	nvt	10	0;166
/ (numpad)	47	47	0;142	0;74
* (numpad)	42	0;144	0;78	nvt
- (numpad)	45	45	0;149	0;164
plus (numpad)	43	43	0;150	0;55
5 (numpad)	0;76	53	0;143	nvt
F1	0;59	0;84	0;94	0;104
F2	0;60	0;85	0;95	0;105
F3	0;61	0;86	0;96	0;106
F4	0;62	0;87	0;97	0;107
F5	0;63	0;88	0;98	0;108
F6	0;64	0;89	0;99	0;109
F7	0;65	0;90	0;100	0;110
F8	0;66	0;91	0;101	0;111

F9	0;67	0;92	0;102	0;112
F10	0;68	0;93	0;103	0;113
F11	0;133	0;135	0;137	0;139
F12	0;134	0;136	0;138	0;140
Home	0;71	55	0;119	nvt
Pijl omhoog	0;72	56	0;141	nvt
PgUp	0;73	57	0;132	nvt
Pijl links	0;75	52	0;115	nvt
Pijl rechts	0;77	54	0;116	nvt
End	0;79	49	0;117	nvt
Pijl omlaag	0;80	50	0;145	nvt
PgDn	0;81	51	0;118	nvt
Insert	0;82	48	0;146	nvt
Delete	0;83	46	0;147	nvt
Print Screen	nvt	nvt	0;114	nvt
Pause	nvt	nvt	0;0	nvt



Backspace	8	8	127	0
Enter	13	nvt	10	0
Tab	9	0;5	0;148	0;165
Home (grijs)	224;71	224;71	224;119	224;151
Pijl omhoog (grijs)	224;72	224;72	224;141	224;152
PgUp (grijs)	224;73	224;73	224;132	224;153
Pijl links (grijs)	224;75	224;75	224;115	224;155
Pijl rechts (grijs)	224;77	224;77	224;116	224;157
End (grijs)	224;79	224;79	224;117	224;159
Pijl omlaag (grijs)	224;80	224;80	224;145	224;154
PgDn (grijs)	224;81	224;81	224;118	224;161

Ins (grijs)	224;82	224;82	224;146	224;162
Del (grijs)	224;83	224;83	224;147	224;163

### De toetsen herdefiniëren

Met de opdracht `p` van ANSI.SYS kan de betekenis van een toets worden veranderd. De syntaxis van de opdracht `p` is als volgt:

```
Esc[keycode;newcodep
```

Vervang de variabele *keycode* door de scancode van de toets die u opnieuw wilt definiëren. De parameter *newcode* vertegenwoordigt de nieuwe code. (Andere toepassingen van de opdracht `p` worden in de volgende paragraaf besproken.)

Het volgende dient ter illustratie. U kunt de toetsen voor de puntkomma en dubbele punt omwisselen, zodat voor de dubbele punt de toets Shift niet meer hoeft te worden ingedrukt en voor de puntkomma wel. Met het volgende batchbestand

SWAPSEMI.BAT worden deze toetsen omgewisseld.

```
@echo off
echo ←[ 59;58p
echo ←[ 58;59p
```



Als de toets een standaard ASCII-toetsenbordteken is (dat wil zeggen geen toetscombinatie met Alt of Ctrl, een functietoets, enzovoorts), kunt u de toets als een reeks specificeren en niet als een scancode of ASCII-code. Een batchbestand wordt veel duidelijker als er reeksen worden gebruikt in plaats van scancodes. Het vorige batchbestand kan ook op de volgende manier worden gedefinieerd:

```
@echo off
echo ←[ ' ; ' ; ' : ' p
echo ←[ ' : ' ; ' ; ' p
```

Wanneer u met ANSI.SYS toetsen verandert, ziet u alleen het resultaat als u tekst achter de DOS-prompt invoert. De meeste programma's, ook EDIT, negeren de nieuwe toetstoewijzingen.

## Reeksen of opdrachten aan toetsen toewijzen

U kunt met de opdracht p meerdere tekens of toetsaanslagen aan een toets toewijzen. Hierbij is het mogelijk tekenreeksen en een ASCII-code te specificeren, mits items door een puntkomma worden gescheiden. Door de volgende opdracht wordt het woord *appel* aan de hoofdletter A toegewezen:

```
echo ←[ ' A ' ; ' Appel ' p
```

Wanneer u de hoofdletter A indrukt, wordt het woord appel naar de toetsenbordbuffer geschreven. U kunt beter geen standaardtoetsen opnieuw definiëren, zoals letters of getallen. Dit voorkomt een hoop ergernis. Stelt u zich voor dat in plaats van de hoofdletter A het woord Appel op het scherm verschijnt. Leuk voor Banana Split, maar verder ook niet.

De functietoetsen zijn in samenhang met de toetsen Alt en Ctrl hiervoor wel geschikt. Met het volgende batchbestand wordt de help-opdracht van DOS aan de functietoets F1 toegewezen:

```
@echo off
```

```
echo ←[0;59;'help';13p
```

De code 0;59 bepaalt de toets die wordt gewijzigd, in dit geval F1. Wanneer F1 wordt ingedrukt, worden de letters *help* ingevoerd, gevolgd door 13. Dit getal is de code voor de Enter-toets. Nadat de nieuwe functie aan de toets is toegewezen, wordt door het indrukken van F1 het woord *help* ingevoerd waarna er op Enter wordt gedrukt.

Zo verandert het volgende batchbestand toets F2, waarmee in het vervolg EDIT wordt gestart:

```
@echo off
echo ←[0;60;'edit';13p
```



Wijst niet meer dan 200 tekens toe aan een enkele toets. ANSI.SYS reserveert niet meer dan 200 bytes voor de toetsbuffer. Wanneer u meer dan 200 tekens toewijst, overschrijft het teveel aan tekens de code van COMMAND.COM. Dit is vervelend. Als dit gebeurt, hangt het systeem en wordt het herstart.



Een eenvoudig maar effectief middel tegen ongewenste blikken is het veranderen van de Enter-toets wanneer het systeem herstart. Het volgende is een deel van een bestand

AUTOEXEC.BAT en geeft de opdrachten waarmee de Enter-toets aan een andere toets wordt toegewezen:

```
echo ←[13;8p
echo ←[24;13p
echo ←[0;133;'prompt $e[13;13p$p$g';13;13p
echo ←[0;134;'prompt $e[13;8p$e[2J';13;13p
```

De eerste regel wijst de Backspace-functie aan de Enter-toets toe. Wanneer iemand op Enter drukt, wordt alleen het laatste teken op de opdrachtregel verwijderd. Op de tweede regel wordt de Enter-functie aan Ctrl-X toegewezen. Als u een opdracht wilt invoeren, moet u in dit geval op Ctrl-X drukken.

De derde regel wijst aan de toets F11 een prompt-opdracht toe waarmee de Enter-opdracht weer aan de Enter-toets wordt toegewezen. Nadat het systeem opnieuw is gestart, drukt u op F11 waarna alles weer bij het oude is. Aan de toets F12 wordt een prompt-opdracht toegewezen die de Enter-toets weer als de backspace instelt. Voordat u uw werkplek verlaat, hoeft u alleen maar op F12 te

drukken. De Enter-toets is aan een andere toets toegewezen, het scherm is schoongemaakt en de prompt is verdwenen. Druk op F11 als u terugkomt en u kunt gewoon weer verder. Dit is verre van waterdicht, maar schrikt nieuwsgierige collega's wel af.

---

---

*Samenvatting*

In dit hoofdstuk werden ideeën aangedragen voor het uitbreiden van de schermweergave en het toetsenbord met ANSI.SYS. De volgende punten zijn hierbij aan de orde gekomen:

- > U moet ANSI.SYS in het bestand CONFIG.SYS laden met behulp van de opdrachten DEVICE of DEVICEHIGH.
- > U stuurt met een speciale set van twee tekens, de ESC en [ tekens, opdrachten naar ANSI.SYS. De opdrachten worden via ECHO, PROMPT of TYPE doorgegeven.
- > Wanneer u batchbestanden schrijft voor ANSI.SYS, moet u eerst met de opdrachten MEM en FIND bepalen of ANSI.SYS in het geheugen is geladen.
- > ANSI.SYS heeft opdrachten voor het herplaatsen van de cursor en voor het opslaan en terughalen van de cursorpositie.

- > Met de opdracht p kunt u de toetsen opnieuw definiëren en opdrachten aan toetsen toewijzen. Ken niet meer dan 200 tekens toe aan een enkele toets.

Steeds meer apparaten worden geïntegreerd in computers waarbij de comptabiliteit een steeds grotere rol gaat spelen. In hoofdstuk 7 wordt er uitgebreid ingegaan op de manier waarop hardware gebruik maakt van IRQ's, DMA-kanalen en de bus om te integreren met het hoofdsysteem.

---

---