

Hoofdstuk 5

Geheugen optimaliseren

In dit hoofdstuk

- > Termen, typen en wetenswaardigheden
 - > Het geheugenarsenaal van DOS 6
 - > Instellingen van het geheugen controleren met MEM
 - > De grootste geheugenwinst
 - > Van RAM naar UMB
 - > Hoe DBLSPACE hoog te laden
 - > Hoe alle basisgeheugen te laden, het extended geheugen te maximaliseren en het expanded geheugen te beheren
 - > Geheugenconflicten oplossen
-
-

Foto's en herinneringen

Als kleine jongen was ik een padvinder. Ik herinner mij dat ik foto's nam van het jaarlijkse kamp. Een vader van één van de andere jongens maakte ook foto's. Toen ze waren ontwikkeld, bleken de foto's van mij goed, terwijl die van de vader uitstekend

waren. Zijn reactie was dat alles makkelijk is als je het eenmaal snapt.

Als je weet wat de functie van de sluitertijd is en waar het diafragma voor dient, ben je al een eind op weg.

Hetzelfde geldt eigenlijk voor het geheugen van een computer. De eerste keer dat het wordt uitgelegd, lijkt het wel geheimtaal. Een ingewijde heeft het over DRAM's, SIMM's, RAM, ROM, UMB, HMA en andere NAF's (nietszeggende afkortingen). Maar als u eenmaal het principe doorhebt en weet waarvoor de termen dienen, zult u zich afvragen waarom het allemaal zo moeilijk leek.

Er is nog één overeenkomst tussen fotografie en het computergeheugen. De meeste mensen kunnen een foto maken, maar vraag ze niet hoe het toestel werkt en wat je er allemaal mee kunt doen. Zo weten ook de meeste computergebruikers wel iets van het geheugen af, maar niet voldoende om dit geheugen te optimaliseren. Uw systeem kan veel meer als het geheugen optimaal wordt gebruikt.

In dit hoofdstuk leert u de basisprincipes van het geheugen: wat het is, wat het doet en waarom u voor bepaalde bewerkingen heel veel geheugen nodig hebt. Bovendien zult u worden ingewijd in de wereld van het geavanceerde geheugengebruik, zodat u uw systeem optimaal zult kunnen optimaliseren.

De vraag naar geheugen

Alle programma's hebben geheugen nodig om te kunnen draaien. Een besturingssysteem kan niet zonder, een eenvoudige kalender vraagt erom, een diskcache heeft het nodig, voor de aansturing van een muis is geheugen nodig. U ziet dat elke opdracht en bewerking een beroep doen op dat geheugen. Hoe groter het programma, des te meer geheugen is er voor nodig.

Elke computer heeft een geheugen van 640K, dat het *conventionele geheugen* of het *basisgeheugen* wordt genoemd. Veel toepassingen kunnen alleen gebruik maken van dit geheugen, en doen geen beroep op ander geïnstalleerd geheugen, zelfs al hebt u een intern geheugen van 16MB. Het probleem is dat

het conventionele geheugen al is teruggelopen naar 550K tegen de tijd dat de startprogramma's worden geladen, zoals het besturingssysteem (COMMAND.COM), de stuurprogramma's voor de muis en het netwerk en een diskcache. (In DOS 6 zijn er verschillende manieren om deze programma's buiten dit geheugen van 640K te starten, zodat er zo weinig mogelijk van het conventionele geheugen wordt verbruikt.)

Een algemene misvatting is dat hoe meer geheugen er wordt geïnstalleerd, des te meer programma's kunnen worden gestart. Ongetwijfeld werkt een groter intern geheugen comfortabeler, maar u kunt er niet van uitgaan dat al dit geheugen door uw programma's kan worden benut. Niet alle geheugen is gelijk en geheugen kan worden aangepast om op verschillende manieren te werken. Op systemen vanaf een 286-computer kan het geheugen worden ingesteld als *expanded* en *extended*. Sommige programma's maken alleen gebruik van het expanded geheugen; andere daarentegen werken alleen met het extended geheugen en er zijn er ook die beide geheugentypen kunnen aanspreken. Lotus 1-2-3 versie 2.4 bijvoorbeeld werkt met het expanded geheugen, terwijl Lotus 1-2-3 versie 3.4 met het extended geheugen werkt.

Er zijn verschillende technieken waardoor u elk stukje geheugen kunt benutten. Voordat deze technieken worden besproken, krijgt u eerst een algemeen overzicht.

Termen, typen en wetenswaardigheden

Wanneer u al op de hoogte bent van geheugenhardware, geheugenadressen en de voornaamste typen geheugen (basis, hogere, hoog, expanded en extended), kunt u dit gedeelte overslaan en verder gaan bij *Geheugen optimaliseren met DOS 6*. Zo niet, dan kunt u beter de volgende paragrafen goed doorlezen, omdat er belangrijke informatie in staat over de principes van het geheugen waarmee u uw systeem zo optimaal mogelijk kunt aanpassen.

Om wegwijs te raken in het doolhof van het geheugen staan we even stil bij de geschiedenis van de PC.

Sinds de introductie van de eerste PC's ruim tien jaar geleden, schreeuwen gebruikers (en programma's) om meer geheugen. Het

geheugentekort werd opgevangen door het plaatsen van steeds meer geheugenchips in de computer, maar al snel werd de grens van 640K bereikt. De oorspronkelijke 8088/8086 computers, zoals de IBM PC en de IBM PC-XT, konden maximaal 640K aan lees-/schrijfgeheugen hebben.

In hun streven om de grens van 640K te doorbreken gingen producenten (zoals Lotus, Intel en Microsoft) samenwerken. Tegelijkertijd werden nieuwe chips op de markt gebracht die maximaal 16MB geheugen konden adresseren. Standaarden verdwenen. De geheugentechnieken van de XT-computers konden niet worden toegepast op de AT-computers of werden zelfs overbodig. Bij de computers uit de 386-klasse kwamen nieuwe opties voor het geheugen beschikbaar.

Dankzij het ontbreken van een standaard is er nu een groot aanbod van allerlei termen en technieken.

ROM en RAM

De twee klassieke geheugens zijn *read-only memory* (ROM) en *random-access memory* (RAM).

Wanneer u de computer aanzet, gaat de PC door een speciaal initialisatieproces. U hebt misschien wel eens de controle van het geheugen op het scherm gezien, of de stations horen draaien voordat DOS werd geladen. Deze handelingen liggen opgeslagen in de ROM. Zoals de naam al aangeeft, wordt dit geheugen wel benaderd, maar nooit gewijzigd. De instructies worden tijdens de fabricage van de chips 'ingebakken' en zijn niet te verwijderen.

Daar tegenover staat de RAM die in het begin leeg is: er staan geen gegevens in die een betekenis hebben. In deze RAM worden tijdelijk programma's en gegevens opgeslagen. RAM is de levensbron van computerprogramma's.

U kunt de ROM vergelijken met een boek waarin tekst staat. Deze tekst is vele malen te lezen maar absoluut niet te veranderen. De RAM is meer een schoolbord, waarop gegevens worden geschreven, veranderd en verwijderd.

Als computergebruiker wilt u RAM zo efficiënt mogelijk gebruiken. De instructies in de ROM zijn onmisbaar voor de computer. Zonder deze gegevens is de computer dood. Het is niet van belang dat u

weet hoe de ROM werkt; u kunt uw aandacht beter richten op de RAM. Er zijn zelfs utility's die ongebruikte ROM-adressen converteren en programma's laten geloven dat ROM RAM is. Hierover verderop meer.

ROM-BIOS

Behalve de ROM die de startcyclus bestuurt, is er ook de ROM-BIOS, die staat voor *read-only memory - basis input/output system*. Met ROM-BIOS worden op een laag niveau taken uitgevoerd, zoals het ophalen van een sector met gegevens of het sturen van gegevens naar een COM-poort. ROM-BIOS is daarom sterk afhankelijk van de hardware.

Wanneer u een oud systeem uitbreidt met nieuwe apparatuur, bijvoorbeeld een nieuwe vaste schijf aan een Compaq Deskpro 286, zult u waarschijnlijk ook de chips voor de ROM-BIOS moeten vervangen. Deze chips zijn niet op de geheugenkaart gesoldeerd. U kunt de oude chips er eenvoudig afhalen en de nieuwe plaatsen.



In het verborgen bestand IO.SYS in de hoofddirectory van de startschijf staan ook programmacorrecties, updates en extensies voor de ROM-BIOS. Met dit bestand kunnen op een makkelijke manier de mogelijkheden van de BIOS van de PC worden uitgebreid zonder dat de chips voor de ROM-BIOS hoeven te worden vervangen.

Toen de eerste IBM PC's op de markt kwamen, waren deze uitgerust met de ROM-BIOS code die was ontwikkeld door IBM en waar een patent op stond. Als gevolg van de stroom niet-IBM PC's, ook wel *clones* of *PC compatibelen* genoemd, begon een aantal leveranciers hun eigen versie te ontwikkelen van de ROM-BIOS. Enkele van de bekendste van deze bedrijven zijn American Megatrends, Award en Phoenix Technologies.

Bij elke ROM-BIOS staat de naam van de producent, een versienummer en de datum waarop de BIOS is geschreven. Met de utility MSD.EXE van DOS 6 kunt u bekijken welk type ROM-BIOS er op uw PC is geïnstalleerd.

Geheugen-hardware

Alle geheugenchips waarmee afzonderlijke gegevensbits worden opgeslagen, werken volgens hetzelfde basisprincipe. Elke gegevensbit is of een 0 of een 1. De aanwezigheid van een kleine elektrische lading geeft aan of de bit *aanstaat* en dientengevolge de waarde 1 heeft. Als de bit geen lading heeft, is de waarde 0. Een *byte* bestaat uit acht bits en een enkel teken wordt met één byte opgeslagen.

Gedurende de afgelopen tien jaar is de roep om geheugen sterk toegenomen. De techniek probeert aan deze vraag te voldoen door een groot aantal verschillende geheugenchips op de markt te brengen. In de volgende paragrafen wordt een overzicht gegeven van de chips die de markt overheersen.

De DRAM-chip

De meest toegepaste geheugenchip in PC-, XT- en AT-computers is de *dynamic random access memory* (DRAM) chip. De DRAM-chip heeft zestien metalen pinnen. Deze chip wordt geïnstalleerd door

de pinnen in de juiste gaten te steken of te solderen op de moederkaart of de geheugenuitbreidingskaart.

De opslagcapaciteit van DRAM-chips varieert van ongeveer 16K bits tot aan 4096K bits en meer. De 16K-bits chips werden in de originele IBM PC's geplaatst, maar tegenwoordig is de meest gebruikte chip de 1024K-bits chip. (De geheugengrootte wordt uitgedrukt in bits en niet in bytes.) Elke DRAM-chip slaat één bit van een byte op. Wanneer er gegevens uit het geheugen worden opgevraagd, worden er tegelijkertijd acht chips benaderd; elke chip levert één achtste deel van de gegevens.)

Het DRAM-geheugen wordt geplaatst in bankjes van negen chips. Acht van deze chips slaan één bit van elke byte op; de negende bit slaat een kruisreferentie of pariteitswaarde op. Deze waarde zorgt ervoor dat de chips foutloos functioneren. (Zie voor een gedetailleerde uitleg de kanttekening *De werking van de pariteitscontrole*.)

Zoals in alle geheugenchips worden gegevens als een elektrische lading van ongeveer vijf volt in een DRAM-chip opgeslagen. Na verloop van tijd neemt deze lading in de DRAM-chip af en verliest de chip het geheugen, vandaar de naam *dynamic* RAM. Om

ervoor te zorgen dat de gegevens accuraat worden bijgehouden, moet de lading om de paar duizendsten van een seconde worden opgefrist. Dit kan vertragend werken. Bij het opslaan van gegevens in de chip kan er worden gewacht op dit opfrissen van de lading.

De SRAM-chip

Voor Dynamic RAM-chips is er niet veel stroom nodig, maar de lading moet regelmatig worden opgefrist. De *static* RAM-chip, of SRAM, heeft meer stroom nodig maar hoeft niet te worden opgefrist. SRAM-chips zijn 'stroomvreter' en duur, maar wel erg snel. De computer hoeft niet te wachten op het opfrissen van de elektrische lading.

In het vorige hoofdstuk hebt u kunnen lezen dat door diskcaches de computer sneller kan worden, doordat het geheugen veel sneller is te benaderen dan de schijf. Zoals u ook weet is het ene geheugentype weer sneller dan het andere; SRAM-chips zijn veel sneller dan DRAM-chips. In geavanceerde systemen worden SRAM-chips soms als een geheugencache gebruikt. Zij slaan geheugengegevens op bijna dezelfde manier op als een diskcache schijfgegevens opslaat. Met de

SRAM-cache worden geheugengegevens opgeslagen die regelmatig worden benaderd of pas nog zijn benaderd. Wanneer er om gegevens wordt gevraagd, wordt eerst in de SRAM-cache gezocht. Indien de gegevens in de cache staan, worden deze zeer snel door de SRAM-chips aan het systeem doorgegeven.



Systemen die SRAM-chips als cache gebruiken, worden gewoonlijk aangeduid als systemen met *cachegeheugen*. Dergelijke systemen presteren veel beter dan dezelfde systemen zonder cachegeheugen. Alhoewel systemen met cachegeheugen duurder zijn, is de aanschaf ervan het overwegen waard omdat een dergelijk systeem veel sneller is. (In feite is de snelheidsverbetering door zelfs een klein geheugen zo opmerkelijk, dat Intel een interne geheugencache van 8K in de 486-chip heeft gebouwd.) Hoe groter de cache, des te beter. Het cachegeheugen dat aan de systeemkaart tussen het geheugen en de computer is toegevoegd, is tussen 64K en 256K groot.

Tot nu toe kan er nog geen cachegeheugen aan een PC worden toegevoegd. Het cachegeheugen wordt tijdens de fabricage geïnstalleerd. Als u een snel systeem wilt, zult u er één moeten aanschaffen dat

gebruik maakt van cachegeheugen, omdat u het naderhand niet kunt installeren.

SIMM's

DRAM-chips voldeden prima in de begintijd van de PC, toen geheugen gelijk stond aan kaviaar en gebruikers slechts mondjesmaat het geheugen konden uitbreiden: 256K op de ene en 384K op de andere machine. Tegenwoordig wordt geheugen in blokken van 4MB of meer toegevoegd. Dat zijn heel wat chips.

Aangezien snelle systemen steeds meer binnen het bereik van de gebruiker komen, worden DRAM-geheugenchips door de fabrikant op kleine printplaatjes geïnstalleerd. Op de moederkaart worden speciale voetjes gemaakt voor deze kleine geheugenkaarten. Deze chips-op-kaart worden *single in-line memory modules* genoemd, afgekort als SIMM's. Op het printplaatje kunnen gewoonlijk 4 tot 9 DRAM-chips worden gesoldeerd met een opslagcapaciteit van 1 tot 8MB. De SIMM's worden ongeveer geïnstalleerd als de normale uitbreidingskaarten in een PC, zoals netwerkkaarten en interne modems. Hiermee behoort het buigen van de pinnen tot het verleden.

De CMOS-chip

Alle 286-computers of hoger hebben een speciale geheugenchip, de CMOS-chip. CMOS betekent *complementary metal-oxide semiconductor*. De CMOS-chip is een speciale geheugenchip die de lading met heel weinig stroom op peil kan houden. Alhoewel er verscheidene CMOS-chips in de hedendaagse PC worden geïnstalleerd, wordt één speciale chip altijd beschouwd als *de* CMOS-chip.

In de CMOS-chip staan de algemene gegevens over het systeem, zoals de datum en de tijd, de configuratie van het diskettestation, het nummer en type van de geïnstalleerde vaste schijf, enzovoorts. Wanneer u nieuwe apparatuur installeert of de systeemconfiguratie wijzigt, kunt u deze informatie met het SETUP-programma bijwerken.

Een kleine batterij in de computer levert de stroom voor deze CMOS-chip. Zo kan de computer de datum en tijd onthouden, zelfs als u de stroom uitschakelt. De levensduur van deze batterij is ongeveer twee jaar.

De snelheid van het geheugen

Sommige geheugentypen zijn sneller dan andere. De snelheid van een geheugenchip wordt uitgedrukt in *nanoseconden*. Een nanoseconde is één miljardste van een seconde.

De snelheid wordt uitgedrukt als de toegangstijd tot de chip.

Hoe kleiner de toegangstijd, des te sneller is de chip. Toegangstijden tot het geheugen kunnen liggen tussen 140NS, aan het begin van het spectrum, tot 60NS of beter aan het eind.

Het is niet altijd verstandig het snelste geheugen te installeren. Boodschappen doen in een Ferrari roept ook vraagtekens op. Hetzelfde geldt voor geheugenchips. Het is geldverspilling om geheugenchips te kopen die te snel zijn voor de computer. Kijk in de handleiding van de computer of vraag advies aan de leverancier voordat u extra geheugenchips gaat aanschaffen.

De snelheid van de computer, de *kloksnelheid*, wordt uitgedrukt in *megahertz* (MHz). De oorspronkelijke PC AT van IBM liep op 6MHz;

tegenwoordig zijn er zelfs huishoudens die een 66MHz hebben staan. De kloksnelheid regelt de snelheid waarmee de computer de opdrachten verwerkt. Hoe hoger de kloksnelheid, des te sneller worden de taken uitgevoerd. De snelheid van de geheugenchip heeft echter geen gelijk tred gehouden met de kloksnelheid van de computer.

Wanneer er gegevens worden opgevraagd uit het geheugen of in het geheugen worden geplaatst, wordt er gewacht op een reactie van de geheugenchips. Het geheugen kan gewoon traag zijn, of de opfriscyclus is de schuldige (wanneer een DRAM-chip wordt opgefrist, worden er geen gegevens opgeslagen). Het gemiddelde aantal *klokcycli* (of *tikken*) die een computer moet wachten terwijl het geheugen zijn werk doet, wordt de *wachtstand* genoemd. De optimale computer heeft geen wachtstanden, hetgeen betekent dat de computer nooit hoeft te wachten op een antwoord van de geheugenchips.

Nadat de gegevens uit een bepaalde geheugenplaats zijn benaderd, moet de chip zichzelf voorbereiden op de volgende vraag. De tijd die wordt besteed aan de voorbereiding, wordt de *precharge time* genoemd.

Enkele jaren geleden zagen computerontwerpers zich geconfronteerd met een dilemma: zelfs de snelste DRAM-chips konden de snelste computers niet bijhouden, en SRAM-chips waren te duur en verbruikten te veel stroom. In plaats van een snellere geheugenchip te ontwerpen werd dit probleem opgelost door de architectuur van de geheugenopslag te veranderen ten behoeve van een hogere snelheid. Het idee van *geheugeninterleaving* werd geïntroduceerd.

Interleaved geheugen wordt georganiseerd op het principe dat het geheugen meestal in aaneengesloten blokken wordt benaderd. Stel dat een computer gegevens benadert in geheugenlokatie 10000 en vervolgens onmiddellijk daarna de gegevens in locatie 10001. Om geen vertraging op te lopen door de tijdvertraging, wordt het geheugen verdeeld in twee grote blokken: even geheugenlokaties en oneven geheugenlokaties. Met interleaved geheugen benadert het systeem één chip om gegevens op te halen uit adres 10000 en een andere chip om gegevens op te halen uit adres 10001. Op deze manier is er geen tijdvertraging.

Het blijkt dat ontwerpers bijna alles hebben gehaald uit de geheugenchips die op DRAM zijn gebaseerd.

Gebruikers die zitten te wachten op meer geheugen, hopen op de opvolger van de DRAM-chips. Alleen door een totaal andere technologie zullen geheugenchips de computer kunnen bijhouden.

Geheugen adresseren

Het proces van het optimaliseren van het geheugen houdt ook in dat het geheugenbeheer moet weten welke geheugenlocaties moeten worden opgenomen bij het laden van bepaalde programma's. Elke geheugenlocatie heeft een adres, net zoals elk huis een adres heeft. Om het geheugenbeheer te kunnen doorgronden zult u moeten begrijpen hoe dit geheugen wordt geadresseerd.

Hexadecimale getallen

De meeste mensen kennen het tientallig of decimale stelsel, dat tien getallen heeft, van 0 tot en met 9. Stel dat de hand van een mens acht vingers zou tellen, dan zouden wij misschien met het zestientallig stelsel werken.

Het adresseren van het geheugen is gebaseerd op dit zestientallig stelsel, dat bekend staat als

hexadecimaal of *hex*. In dit stelsel vertegenwoordigt elk teken één van de zestien waarden.

In talen die van het Latijn zijn afgeleid, worden met bepaalde tekens getallen aangegeven: een cirkel stel een 0 voor, een verticale lijn een 1, een 2 is een gebogen lijn met een horizontale lijn aan de onderkant, enzovoorts. Hetzelfde geldt voor hexadecimale getallen, maar er is een probleem met zes getallen die groter zijn dan negen. Door welke symbolen zullen deze getallen worden voorgesteld? Wiskundigen hebben hiervoor een saaie oplossing gevonden.

Bij hexadecimale getallen worden de letters A tot en met F gebruikt voor de getallen van 10 tot en met 15. De zestien getallen van het hexadecimale stelsel zijn als volgt: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E en F.

In het decimale stelsel worden getallen groter dan 9 voorgesteld door twee of meer cijfers. Het decimale stelsel kent tientallen, honderdtallen, duizendtallen, enzovoorts. Het getal 625 bestaat uit een 6 die zeshonderd waard is, een 2 die twintig waard is en een 5 die 5 waard is. De plaats is dus bepalend voor de waarde van een cijfer. Wij zijn hier zo mee

vertrouwd dat wij er nooit over nadenken. Totdat er getallen moeten worden opgelezen in het Engels. Dan raken we soms in de knoei.

Bij het hexadecimale stelsel wordt hetzelfde schema aangehouden. De getallen worden vergroot in blokken van 16: 16, 256, 4096, enzovoorts.

Het hexadecimale getal 235 betekent twee 256-tallen, drie 16-tallen en vijf eenheden. Omgerekend is dit getal in het decimale stelsel 565. ($2 \times 256 = 512$; $3 \times 16 = 48$; $5 \times 1 = 5$. Bij elkaar opgeteld 565.)

<i>Decimaal</i>	<i>hexadecimaal</i>
3	3
9	9
10	A
13	D
17	11
55	37

Hoofdstuk 5: Geheugen optimaliseren

256 100

266 10A

Zonder letters is het moeilijk, zo niet onmogelijk decimale van hexadecimale getallen te onderscheiden. Er zijn echter enkele manieren om te bepalen of een getal een hexadecimaal getal is. Eén manier is om de tekens *0x* of *\$* voor het getal te plaatsen. Een andere manier is om de kleine letter *h* achter het getal te plaatsen. Het hexadecimale equivalent van het decimale getal 55 kan worden geschreven als *0x37*, *\$37* of *37h*.

De meeste programmeurs zijn vertrouwd met het hexadecimale stelsel. Ik ken echter een melkboer die zijn produkten kan optellen in hexadecimaal. U hoeft echter alleen het principe van dit stelsel te begrijpen voor het beheren van het geheugen. Ik ben tussen twee haakjes 24 jaar oud in hexadecimaal.

Segmenten en offsets

In de oorspronkelijke IBM PC's zat de 8088-microprocessor van Intel. Kort nadat de IBM PC was gelanceerd, won de 8086-processor aan populariteit. Deze werd geïnstalleerd in sommige

compatibele IBM PC's en ook in de IBM PS/2 model 25 en model 30.

Bij de 8088 werd een 8-bits bus gebruikt, terwijl de 8086 gebruik maakte van een 16-bits gegevensbus (over bussen verderop meer). Voor de rest waren deze beide chips volkomen gelijk. Zij vormen de grondslag voor de gehele computerfamilie. Alle nieuwe microprocessoren van Intel, ook de 80486, werken als snelle 8086-ers.

Bij de 8086-chip wordt een set *registers* toegepast voor het beheren en manipuleren van gegevens. U wilt bijvoorbeeld twee hele getallen vermenigvuldigen. Elk getal wordt in een ander register geplaatst en de processor krijgt de opdracht deze getallen met elkaar te vermenigvuldigen. Elk register is 16 bits groot.

Met de registers worden ook gegevens benaderd die in het geheugen liggen opgeslagen. De grootste waarde van een 16-bits getal kan 64K zijn, dus een enkel register kan aan 64K geheugenadressen benaderen. Deze waarde is eigenlijk 2 tot de 16e macht, of 65.536 om precies te zijn. Zelfs de vroegere computertechnici zagen in dat een PC met een geheugen van 64K wel een erg magere

computer zou worden. Daarom werd besloten met twee registers geheugenadressen op te slaan. Zij ontwikkelden een systeem waarin een geheugen van 1.048.576 bytes (1 megabyte) kon worden benaderd. Geen enkel programma zou toch ooit meer geheugen hoeven te benaderen?

Zo werkt het systeem. Elke byte in het geheugen heeft een uniek adres, zoals elk kantoor in een kantorencomplex een uniek adres heeft. Een geheugenadres van een computer bestaat uit twee delen, het *segment* en de *offset*. Dit zijn de verdieping en het kantoornummer in het kantorencomplex.

Het 1MB grote geheugen wordt onderverdeeld in grote blokken, die *segmenten* worden genoemd. Elk segment begint 16 bytes onder het vorige segment. Dit is hetzelfde als het bepalen van de verdieping waarop het kantoor moet komen. De *offset* geeft aan hoeveel bytes er liggen tussen het begin van het segment en de exacte lokatie van het geheugen. Dit is hetzelfde als het bepalen van het nummer van het kantoor.

Om een geheugenadres op te geven moet u zowel het segment als de offset specificeren. Normaal

gesproken geeft u het segment en de offset als volgt op: *segment:offset*. De segment- en offsetwaarden worden door een dubbele punt gescheiden. Het segment wordt in het ene 16-bits register opgeslagen en de offset in het andere.

Waarden worden weergegeven in hexadecimale getallen en de getallen worden gewoonlijk met nullen tot vier cijfers aangevuld. Het geheugenadres in bijvoorbeeld het segment B800, offset 0 wordt geschreven als *B800:0000*, en het geheugenadres in segment 40, offset 17 wordt geschreven als *0040:0017*. (Alhoewel de getallen in het hexadecimale getalstelsel worden weergegeven, worden de symbolen 0x, h of \$ vaak niet toegevoegd, omdat geheugenadressen altijd als hexadecimale getallen worden weergegeven.)

Sommige geheugenbeheerstaken worden uitgevoerd op een geheugenbereik in plaats van een enkele lokatie. Wanneer u twee getallen van vier cijfers ziet die gescheiden worden door een minteken (-), is het adres eigenlijk een geheugenbereik. De waarde voor het minteken is het eerste segment in het bereik en de twee waarde geeft het laatste segment in het bereik aan, zoals *beginsegment-eindsegment*.

Wanneer u nooit zult programmeren, vraagt u zich misschien af waarom u moet weten hoe het principe van het geheugenadres in de computer werkt. Het antwoord is dat u deze geheugenadressen kunt tegenkomen tijdens het installeren van stuurprogramma's voor hardware of het toevoegen van opdrachten aan CONFIG.SYS voor het nauwkeurig instellen van het geheugenbeheer. De volgende regel is gehaald uit mijn CONFIG.SYS:

```
device=c:\dos\emm386.exe ram
win=dd00-dfff win=da00-dcff
```

De WIN-opdrachten aan het einde van de regel zijn de instructies voor het geheugenbeheer EMM387 om het gespecificeerde geheugengebied te reserveren voor Windows. Verderop in dit hoofdstuk staat meer over EMM386 en de opties.

Geheugenadressen met de segment- en offsetnotatie kunnen liggen tussen 0000:0000 en FFFF:FFFF. (Het grootste hexadecimale getal is F.)

De voornaamste geheugentypen

De 8086-microprocessors maken gebruik van een 20-bits adressenschema dat een maximaal

adresseerbaar geheugen oplevert van 1MB. Toen Intel de 80286 (of de 286) introduceerde, kwamen er nieuwe mogelijkheden voor het geheugen bij. Er werd ook voor gezorgd dat de 286 als een snelle 8086-chip zou kunnen functioneren. Dat moest ook wel, anders was er een rel uitgebroken als was gebleken dat software wel onder de 8086 maar niet onder de 80286 zou draaien. Men schreef enkele nieuwe programma's die gebruik maakten van het geheugenschema dat in de 80286 was geïntroduceerd. Gebruikers van 8086-systemen konden niet met deze programma's werken. U kent vast wel software die alleen werkt op een 286 of hogere machine, Lotus 1-2-3 versie 3 bijvoorbeeld.

Gebruikers waren niet tevreden, zelfs niet met de betere geheugenmogelijkheden in de 286. De 386 werd geïntroduceerd met nog meer geheugenfuncties. En eens te meer moest de chip achterwaarts compatibel zijn, deze keer met de 286 en 8086. Software werd gemaakt voor de 386 en de uitgebreide geheugenmogelijkheden. Voorbeelden van dergelijke programma's zijn Windows 3.1 (in enhanced mode), Paradox 386 en zelfs enkele functies in DOS zijn voorbeelden van dergelijke programma's.

Geheugen komt in allerlei vormen, en voor sommige programma's moet dit geheugen op een bepaalde manier worden geconfigureerd. In het voorafgaande hebt u over de verschillende geheugentypen gelezen die voor elk platform beschikbaar zijn.

Hoe een systeem te controleren

De mate waarin u een systeem kunt verbeteren, hangt af van de grootte van het geïnstalleerde geheugen en de microprocessor. Met MSD kunt u de exacte informatie over het systeem en het geheugen opvragen.



Zelfs al werkt u met een vorige versie van DOS, dan nog hebt u het programma MSD zonder dat u dit waarschijnlijk weet. Het wordt geleverd bij Windows 3.1 (In Microsoft Word 5.5 zaten de eerste probeersels van MSD.) In DOS 6 (of DOS 5) kunt u MSD op de vaste schijf opzoeken met de volgende opdracht:

```
C:\> dir \msd.exe /s
```

U start MSD zonder opties als volgt op:

```
C:\> msd
```

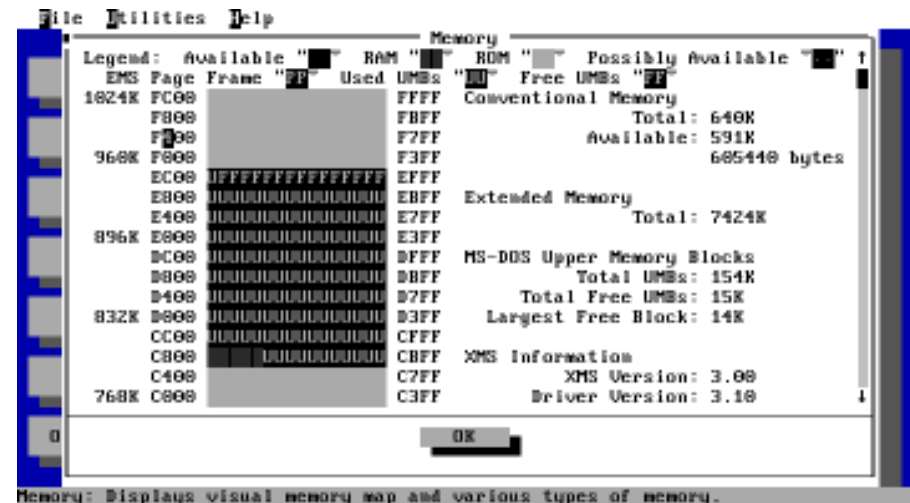
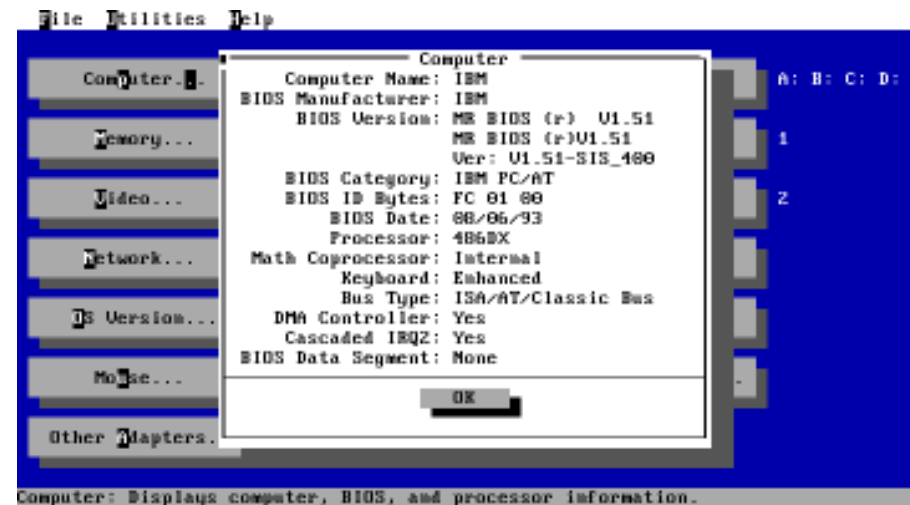
Nadat MSD in het systeem heeft rondgekeken, wordt u gevraagd een toets in te drukken waarna het hoofdmenu verschijnt.

Om te weten te komen welke processor er in het systeem zit, selecteert u de optie Computer door er met de muis op te klikken, of door P in te drukken en vervolgens de informatie op de regel Processor te bekijken. Voor het geheugen selecteert u de optie Memory. Zie afbeelding 5-1 voor de informatie die MSD weergeeft over de computer en het geheugen.

Basisgeheugen en hogere geheugen

De meeste gebruikers van DOS kennen de geheugengrens van 640K. Deze grens is de grootste hoeveelheid geheugen die door een programma kan worden gebruikt en is elke dag weer een bron van frustratie. U hebt echter kunnen lezen dat er een systeem is ontworpen waarmee 1MB geheugen kan worden geadresseerd. Waarom wordt er dan nog gesproken over een grens van 640K?

Alhoewel er 1MB adresseerbaar geheugen in een computer zit, is niet alles direct beschikbaar voor



Figuur 5-1: Specificaties over computer en geheugen door MSD

programma's onder DOS. Het gebied tussen 640K en 1MB wordt gereserveerd voor speciale adapterkaarten en de ROM-BIOS. Een deel van dit geheugen is bestemd voor de video display. Dit gereserveerde deel van het geheugen tussen 640K en 1MB staat bekend als het *hogere geheugen*.

Er wordt niet karig omgesprongen met het geheugenadres in het hogere geheugen, vooral wanneer u bedenkt hoe kostbaar het basisgeheugen is. In de meeste gevallen wordt een derde van het hogere geheugen helemaal niet gebruikt. Als u een 386 of hoger systeem hebt, kunt u dit geheugengebied voor het opslaan van DOS-programma's gebruiken. Hoe u dit doet, leest u verderop in dit hoofdstuk.



Gedurende de laatste paar jaar is er veel veranderd in de geheugentechnologie. Jammer genoeg geldt dit ook voor de terminologie. Sommige termen die in het begin van de tachtiger jaren werden gebezigd, hebben nu een totaal andere betekenis. Gelukkig schijnt de terminologie nu stabiel te zijn. De termen die in dit boek worden gebruikt, zijn afgeleid van deze terminologie.

Misschien wordt het iets makkelijker het schema van het geheugenbeheer te begrijpen, als u in termen van een magazijn denkt. Door middel van die analogie worden enkele belangrijke details duidelijk gemaakt:

- De DOS-opdracht COMMAND.COM gebruikt automatisch het eerste deel van het basisgeheugen (640K). Hierdoor blijft er minder dan 640K over voor de programma's.
- Stuurprogramma's, zoals voor de muis en het netwerk, en andere residente programma's maken ook aanspraak op dit basisgeheugen.
- Alhoewel 8086-computers 1024K geheugen kunnen adresseren, is er op de meeste computers niet zoveel RAM voor algemene doeleinden geïnstalleerd. De geheugenadressen in het hogere geheugen zijn gereserveerd voor het geheugen dat ergens anders is te vinden. Het geheugen voor ROM-BIOS is geplaatst in de ROM-chips en het videogeheugen op de videokaart. Die kunt u zich voorstellen als containers die in het magazijn liggen opgestapeld.

- Sommige gebieden van het hogere geheugen zijn niet bezet. Deze gebieden zijn beschikbaar voor extra apparatuur, maar in de meeste gevallen wordt dit deel van het geheugen ongebruikt gelaten.

Wanneer een programma wordt gestart, wordt het onder in het basisgeheugen geladen. De rest van het basisgeheugen is beschikbaar voor gegevens. Bij Lotus 1-2-3 zijn de gegevens het spreadsheet; bij WordPerfect zijn dit de tekstbestanden.

Als u een programma zoals Lotus 1-2-3 laadt, kan het meer dan 200K geheugen in beslag nemen, waardoor het geheugen voor de gegevens 300K of minder is.

Het expanded geheugen

Lotus 1-2-3 is het programma waardoor de PC bij bedrijven in Amerika gemeengoed werd. Maar al snel verlangde men grotere spreadsheets en computers hadden te weinig geheugen om het werk aan te kunnen.

Drie grote bedrijven in de computerwereld sloegen de handen ineen voor meer geheugen: Lotus

Development Corporation uit de spreadsheethoek, Microsoft Corporation uit de wereld van het besturingssysteem en Intel Corporation uit die van de hardware. Zij noemden hun oplossing het *expanded geheugen*.

Veel ruimte in het geheugen wordt ongebruikt gelaten. De voor de hand liggende oplossing was het hogere geheugen beter te gebruiken. Maar het hogere geheugen leverde slechts enkele honderdduizenden extra bytes op en het werd duidelijk dat de gebruikers spreadsheets wilden van 2MB of 3MB groot. Het antwoord hierop was *memory paging*.

Lotus, Intel en Microsoft ontwikkelden de *expanded memory specification* of EMS. Deze specificatie wordt vaak LIM-EMS genoemd, naar de drie grote krachten achter dit project (Lotus, Intel en Microsoft). Het principe achter LIM-EMS is in feite heel eenvoudig. Een gebied van 64K van het hogere geheugen wordt gereserveerd als expanded geheugen. Dit gebied, dat de *EMS-pagina-frame* wordt genoemd, is onderverdeeld in vier stukken van 16K elk. Deze vier stukken kunnen uit een voorraad van 32MB geheugen worden gehaald.

Een programma kan niet al het expanded geheugen in een keer benaderen. De gegevens moeten in stukken van 16K worden beheerd. Dit toewijzen van verschillende geheugenblokken in het pagina-frame wordt *bank switching* genoemd.

In het geheugenmagazijn wordt de voorraad expanded geheugen voorgesteld door opleggers van 16K. Dit geheugen wordt gescheiden van het 1MB magazijn en moet in kleine porties worden opgeslagen in het magazijn, voordat DOS het kan benaderen.

De vergelijking met de oplegger gaat enigszins mank, omdat stukken expanded geheugen van 16K niet naar het hogere geheugen kunnen worden vervoerd. Wanneer DOS probeert de gegevens in het expanded geheugengebied te manipuleren, wordt dit door het geheugenbeheer onderschept en worden de echte veranderingen in het echte geheugen aangebracht, dat ergens in het expanded geheugengebied van 32MB ligt.

De oplossing voor het expanded geheugen voor 8086-computers is een tussenoplossing. Er wordt inderdaad veel geheugen vrijgemaakt voor programma's, maar het bank switching is niet erg

efficiënt. Het benaderen van geheugen in porties van 16K kost tijd. Geheugen zou beter in grote, aaneengesloten blokken moeten worden benaderd.

Het extended geheugen

Toen Intel de 286-microprocessor ontwikkelde, besloten zij een nieuwe geheugenarchitectuur te implementeren waarmee grote stukken geheugen konden worden benaderd, zonder te worden belemmerd door de beperkingen van bank switching. Deze nieuwe architectuur werd *extended geheugen* genoemd. De PC AT was de eerste computer van IBM met de 286-microprocessor.

Het is een raadsel waarom men heeft gekozen voor de term extended die zo veel lijkt op expanded. Dit is een bron van verwarring. Men had het beter het 286-geheugen of het aanvullende geheugen kunnen noemen, maar ja, waarom makkelijk doen als het moeilijk kan?



Een ezelsbruggetje voor diegenen die deze twee geheugentypen door elkaar halen. De *p* van expanded komt in het alfabet voor de *t* van extended. Een ander bruggetje is dat het *extended* geheugen niet beschikbaar was op de PC *XT*.

Het extended geheugen zet de deur open naar het adresseerbare geheugen van 1MB. Intern wordt door de 286 met 24 bits het geheugen geadresseerd; de 8086 doet dit met 20 bits wat een maximale geheugencapaciteit van 16MB oplevert. Omdat de 286 achterwaarts compatibel is, kan deze processor zich gedragen als een 8086, zodat alle software die op de 8086 draait, ook op de 286 kan draaien.

De 286 heeft twee uitvoeringsmodi. Wanneer de 286 zich gedraagt als een 8086, is de *Real Mode* actief. (Deze modus had beter de *8086 Mode* kunnen heten.) Om het geheugen voorbij de 1MB te kunnen benaderen, en dat is in totaal 15MB, moet de microprocessor overschakelen naar *Protected Mode*. Het geheugen boven de 1MB is het *extended geheugen*.

In Protected Mode heeft het systeem toegang tot 16MB geheugen, vooropgesteld dat er zoveel geheugen is geïnstalleerd. Dat is een heel groot magazijn. Alle extended geheugen ligt opgeslagen in het magazijn en hoeft niet met vrachtwagens naar elders te worden vervoerd.

Programma's die in Protected Mode werken, hebben dus toegang tot 16MB geheugen, en dat is

heel wat meer dan die 1MB. Niet lang na de introductie van de 286 was de algemene tendens dat er veel software zou worden geschreven voor de Protected Mode, maar dat bleek niet zo te zijn.

Het probleem was eenvoudig. Elk programma dat speciaal wordt geschreven voor Protected Mode, draait niet op computers met een 8086-microprocessor, met andere woorden de oorspronkelijke PC's en XT's. Gedurende de eerste jaren na de introductie van de 286 werden er nog steeds veel meer 8086-ers verkocht. Deze markt was te groot om eraan voorbij te gaan. Toch werden er door enkele bedrijven produkten op de markt gebracht die uitsluitend voor Protected Mode waren geschreven. Het meest populaire programma daarvan was Lotus 1-2-3 versie 3. Maar Lotus verzekerde wel dat zij de versie voor de 8086 zouden blijven verbeteren.

DOS is een programma voor Real Mode en kan alleen die eerste 1MB van het geheugen benaderen. Een programma voor Protected Mode wordt vanuit DOS in Real Mode gestart. Het programma moet vervolgens de microprocessor omschakelen naar Protected Mode om het extended geheugen te kunnen benaderen. Wanneer het programma wordt

afgesloten, wordt de microprocessor teruggeschakeld naar Real Mode en wordt de besturing overgedragen aan DOS.

Waarom is DOS 6 een programma voor Real Mode? Als u denkt dat het maken van een besturingssysteem voor Protected Mode nut heeft, hebt u volkomen gelijk. Het beste voorbeeld hiervan is het besturingssysteem OS/2. Dit besturingssysteem is oorspronkelijk ontwikkeld door Microsoft, maar is nu in handen van IBM. Ondertussen heeft men wel de ondersteuning voor Protected Mode in Windows ingebouwd en zal dit ook gebeuren in komende versies van DOS.

Als een echt programma voor Real Mode heeft DOS niets te zoeken in Protected Mode. Hierdoor was Protected Mode een prooi voor een ieder en deed men wat men wilde. Er ontstond een wildgroei voor het extended geheugen. In 1988, vier jaar na de introductie van de 286-computers, staken de grote computerfabrikanten de hoofden bij elkaar en besloten een standaard te zetten voor het gebruik van het extended geheugen. De bedrijven die het meest betrokken waren bij het expanded geheugen (Lotus, Intel en Microsoft), kwamen met een nieuwe specificatie voor het extended geheugen.

Deze specificatie werd de *extended memory specification* of XMS genoemd.

De opvolger van de 286-microprocessor was de 80386, ook wel de 386 genoemd. De 386 kan alles wat de 286 ook kon, maar dan alleen een stuk sneller. De kloksnelheid van de 286 ligt tussen 6MHz en 12MHz, terwijl die van de 386 tussen de 16MHz en 33MHz en verder ligt.

Behalve een hogere snelheid biedt de 386 nog enkele, belangrijke voordelen:

- Een 386 heeft toegang tot 4096MB RAM, een slordige vier miljard bytes. Zelfs vandaag de dag is dit een hele GROTE spreadsheet.
- Een andere, belangrijke verbetering was dat in de 386 het extended geheugen zich met behulp van enkele stuurprogramma's ging gedragen als expanded geheugen. Zo konden gebruikers van 386-computers met veel geheugen dit geheugen aanspreken bij programma's die geschreven waren voor het expanded geheugen.
- Waarschijnlijk de meest in het oog springende nieuwe technologie in de 386 was de

mogelijkheid om de microprocessor in *Virtual 8086 Mode* te laten draaien. Dit houdt in dat het systeem met behulp van extended geheugen meerdere gebieden in het basisgeheugen van 640K kan emuleren. Bij elke sessie lijkt het voor de software alsof het in Real Mode draait. Producten zoals DESQview 386 en Windows 3.x in Enhanced Mode maken gretig gebruik van deze mogelijkheid, waardoor u meerdere DOS-sessies naast elkaar kunt draaien.

- De ongebruikte geheugenblokken in het hogere geheugengebied kunnen worden opgevuld met extended geheugen. DOS kan dit geheugen benaderen voor zichzelf, residente programma's en stuurprogramma's en zo kostbaar basisgeheugen vrijmaken voor toepassingssoftware.

Al deze opties hebben uiteindelijk de gebruikers in twee kampen verdeeld: degenen met een 386 en degenen zonder 386. Er worden steeds meer programma's ontwikkeld die alleen maar draaien op een 386 of hoger systeem.

Na de 386 werd de 486 door Intel geïntroduceerd. De 486 heeft een hogere kloksnelheid, een

ingebouwde coprocessor en talloze andere technologische voordelen. Wat betreft het geheugenbeheer is de 486 gelijk aan de 386.

Hoog geheugen (HMA)

Vanaf de 286-processoren kan het extended geheugen worden benaderd door de microprocessor over te schakelen naar Protected Mode. DOS kan echter niet draaien in Protected Mode, dus vinden de standaard DOS-programma's geen baat bij het extra geheugen.

Dankzij een vernuftige aanpassing in het ontwerp van de 386 kan bijna het gehele eerste deel van het extended geheugen ter grootte van 64K worden benaderd zonder te hoeven overschakelen naar Protected Mode. Dit betekent dat DOS in Real Mode bijna 64K aan extra geheugen kan benaderen op systemen met een 286-processor. 64K lijkt niet veel, maar het is wel 10 procent en die kan van pas komen.

De eerste 64K van het extended geheugen wordt het *high memory area*, of HMA genoemd. Quarterdeck was de eerste die tot dit gebied doordrong; vanaf versie 5.0 heeft DOS dit direct

ondersteund. Met het stuurprogramma HIMEM.SYS van DOS komt dit hoge geheugen beschikbaar voor DOS. Programma's als Lotus 1-2-3 en WordPerfect kunnen niet rechtstreeks gebruik maken van de HMA, maar u kunt de voornaamste programma's van DOS wel in deze HMA laden. Dit maakt zo veel mogelijk geheugen vrij van het basisgeheugen van 640K. Windows kan ook bij de HMA.

Andere geheugentermen

In de volgende paragrafen staan korte beschrijvingen van andere geheugentermen die u in handleidingen of bladen kunt tegenkomen.

Geheugenbeheerders

Geheugenbeheerders zijn utility's die de geheugenbehoeften van programma's regelen. Zij zijn er onder andere verantwoordelijk voor hoe het geïnstalleerde geheugen wordt geconfigureerd, dat wil zeggen of het expanded of extended geheugen is, waar de software wordt geladen en hoe het geheugen aan programma's wordt toegewezen als er verzoeken voor binnen komen.

Programma's voor geheugenbeheer bestaan gewoonlijk uit meerdere programma's. Het ene programma is bijvoorbeeld verantwoordelijk voor de toewijzing van het expanded en het extended geheugen en een ander programma voor het beheer van de HMA. Geheugenbeheerders kunnen ook vaak alle geïnstalleerde software analyseren en de bestanden AUTOEXEC.BAT en CONFIG.SYS aanpassen voor een optimaal gebruik van het geheugen.

Twee van de populairste programma's op dit gebied zijn Expanded Memory Manager van Quarterdeck en 386MAX van Qualitas. Veel functies uit deze programma's zijn nu beschikbaar in DOS 6.

DOS extenders

Het schrijven van programma's voor Protected Mode is zeer ingewikkeld. Een aantal bedrijven brengt bibliotheken met routines voor Protected Mode op de markt die de programmeurs heel wat hoofdbrekens besparen. Deze bibliotheken zorgen voor een laag tussen het programma en DOS en staan bekend als de *DOS extenders*. De programmeur leent een routine uit de bibliotheek

en het DOS-verlengstuk zorgt voor de uitvoering ervan naar het besturingssysteem of de hardware.

Twee van de bekende bibliotheken zijn DOS-Extender van Phar Lap Software en DOS/16M van Rational Systems. Als gewone DOS-gebruiker zult u weinig interesse hebben voor deze programma's, maar zo nu en dan komt u een foutbericht tegen dat wordt gegenereerd door een programma dat met behulp van één van deze bibliotheken is geschreven.

Er is een serie programmeerstandaarden ontwikkeld die de *DOS Protected Mode Interface*, of DPMI, worden genoemd. De meeste programma's in Protected Mode komen overeen met DPMI versie 0.9. Door deze standaarden is het mogelijk dat verschillende programma's samen van de Protected Mode gebruik kunnen maken. Zo kunnen Windows 3.1 en Lotus 1-2-3 versie 3 tegelijkertijd draaien, dankzij deze DPMI-standaard.

Een andere programmastandaard die u zou kunnen tegenkomen, is de *Virtual Control Program Interface*, of de VCPI. Hiermee wordt een standaard gezet voor het opnieuw converteren van expanded geheugen naar extended geheugen, als het expanded

geheugen in werkelijkheid extended geheugen is dat als expanded geheugen is geconfigureerd.

Upper memory blocks (Hogere geheugenblokken)

Het hogere geheugen wordt gereserveerd voor het videogeheugen, ROM-BIOS en ander geheugen voor de hardware, maar veel van dit geheugen blijft ongebruikt. Vanaf de 386 kan het extended geheugen in deze ongebruikte gebieden worden ingedeeld en kunnen er stuurprogramma's en andere residente programma's in worden opgeslagen. De gebieden met extended geheugen die in dit ongebruikte hogere geheugen worden ingedeeld, zijn de *upper memory blocks* of de UMB's.

Shadow RAM (Schaduw RAM)

Over het algemeen is ROM veel trager dan RAM. Wanneer DOS een station moet benaderen, naar het videogeheugen moet schrijven of andere hardware gaat benaderen, moet er een code worden uitgevoerd in de ROM-BIOS. Om deze bewerkingen sneller te laten verlopen hebben sommige computers een systeem dat de *shadow RAM* heet. Wanneer de computer wordt gestart,

wordt er een kopie van de ROM-BIOS overgebracht naar de RAM, die een stuk sneller is dan de ROM. Wanneer DOS probeert de ROM-BIOS te benaderen, wordt DOS 'gedwongen' de kopie van ROM in het extended geheugen te lezen.

Stealth- en BIOS-compressie

In de strijd om de meeste RAM hebben sommige grote producenten van geheugenprogramma's ingenieuze technieken ontwikkeld om de laatste druppel uit de RAM te persen.

QEMM-386 van Quarterdeck Office Systems maakt gebruik van een speciale schakeltechniek, de *Stealth*. Stealth verbergt de ROM-BIOS, video-ROM en zelfs de schijf-ROM, en vult de leeggekomen adressen op met UMB's. Deze benadering is sterk afhankelijk van de hardware en werkt niet op alle systemen.

386MAX van Qualitas herkent ongebruikte gebieden van de BIOS en converteert deze naar UMB's. Deze techniek wordt BIOS-compressie genoemd en kan een extra 32K aan geheugen opleveren op systemen van Compaq en IBM.

Virtueel geheugen

Het virtuele geheugen is een truc om meer geheugen te benaderen dan er in werkelijkheid is geïnstalleerd. Wanneer al het aanwezige geheugen wordt bezet en er wordt door een programma om meer geheugen gevraagd, kan een virtueel geheugenbeheerder tijdelijk een deel van de gegevens die in het geheugen zijn opgeslagen, op schijf plaatsen. Op deze manier wordt dat geheugengebied voor het desbetreffende programma vrijgemaakt. Wanneer de eigenaar van het geheugen op schijf een beroep doet op dat geheugen, moeten de gegevens weer vanaf de schijf naar het geheugen worden overgeplaatst.

Het virtuele geheugen is veel langzamer dan de RAM omdat het op schijf is opgeslagen. Wanneer u echter geheugen te kort komt, is langzaam beter dan helemaal niets.

Windows is het bekendste voorbeeld van een programma dat virtueel geheugen ondersteunt. Het beruchte wisselbestand doet dienst als een tijdelijke opslag voor het virtuele geheugen.

Optimalisering van het geheugen met DOS 6

Na de theorie is het tijd voor de praktijk. In de volgende paragrafen wordt uitgelegd hoe met de opdrachten HIMEM.SYS, EMM386.EXE en MEMMAKER het geheugen is te optimaliseren.

Het geheugenarsenaal van DOS 6

In DOS 6 zitten enkele eersteklas utility's voor het beheren van het geheugen. In tabel 5-1 vindt u een kort overzicht van elke utility voor het geheugenbeheer dat in DOS 6 aanwezig is.

Tabel 5-1 Utility's voor geheugenbeheer	
<i>DOS-utility</i>	<i>Doel</i>
HIMEM.SYS	De beheerder voor het extended geheugen, onder andere voor het toewijzen van het extended geheugen. Met HIMEM.SYS kunt u DOS <i>hoog</i> laden (in de HMA).

EMM386.EXE	Simuleert expanded geheugen met behulp van het extended geheugen en staat toe dat extended geheugen kan worden gebruikt als UMB's voor het laden van stuurprogramma's en programma's.
DOS=HIGH	Geeft instructie aan DOS zichzelf hoog te laden.
DOS=UMB	Geeft DOS instructies de programma's in de UMB's te beheren.
MEM.EXE	Geeft de status van het geïnstalleerde geheugen weer, waaronder de lokatie van programma's in het geheugen en hoeveel geheugen er wordt gebruikt door programma's.
MDS.EXE	Stelt een diagnose over de hardware, BIOS en software.
DEVICEHIGH	Laadt stuurprogramma's in de UMB's.
LOADHIGH	Laadt programma's in de UMB's.
MEMMAKER	Optimaliseert het gebruik van het systeemgeheugen en wijzigt de bestanden CONFIG.SYS en AUTOEXEC.BAT voor een beter gebruik van het beschikbare geheugen.

Hoeveel geheugen u kunt gebruiken, hangt af van de microprocessor, het type en de hoeveelheid van het geïnstalleerde geheugen. Als geheugensteuntje: met MSD.EXE kan het type microprocessor en het geïnstalleerde geheugen worden bepaald (zie hiervoor de paragraaf *Hoe het systeem te controleren* eerder in dit hoofdstuk.)

Plankgas met de 80386

Als u een 386 of snellere machine hebt, kunt u alle geheugenprogramma's in DOS 6 benutten.

Met deze programma's kan het basisgeheugen zo groot mogelijk worden gemaakt door ongebruikte hogere geheugen op te vullen met extended geheugen (UMB's), waarin stuurprogramma's en residente programma's (TSR's) zijn te laden. In Real Mode benadert u de eerste 64K van het extended geheugen als hoog geheugen (HMA) en laadt u vervolgens DOS in HMA. Op systemen met een 386-microprocessor of hoger wordt door EMM386 met extended geheugen expanded geheugen geëmuleerd.

Opties voor de 286

Veel mogelijkheden voor het beheren van het geheugen benutten de hardware die in de 386-microprocessor is geïntroduceerd. Het voornaamste beheersprogramma, EMM386, draait alleen op systemen met een 386-microprocessor of hoger, zoals de naam al aangeeft. Zonder EMM386 is er geen ondersteuning voor UMB's. De voornaamste functie van MEMMAKER is om het gebruik van UMB's te optimaliseren, en ook dit programma draait alleen op systemen met een 386-microprocessor of hoger.

De systemen met een 286-microprocessor, zoals de AT's van IBM en compatibele machines, ondersteunen wel het extended geheugen en de HMA. U kunt daarom met HIMEM.SYS de HMA beheren en DOS hoog laden. Maar dat is dan ook alles.

Slecht nieuws voor bezitters van een 8086

Wanneer u nog met een 8086 werkt, zoals een originele PC of een XT van IBM, hebt u pech gehad. U kunt geen UMB's maken en u hoeft de HMA niet te benaderen omdat uw computer het

extended geheugen niet kan benaderen. Met andere woorden, niet één van de geheugenutility's in DOS 6 werkt op een 8086.

Alternatieven voor bezitters van een 8086 of 286

Alhoewel DOS 6 de bezitters van een 8086 en 286 tekort doet wat het geheugen aangaat, zijn er toch producten van andere producenten waarmee wel het geheugen is te verbeteren. Deze programma's leveren veel van de mogelijkheden voor UMB's door gebruik te maken van een bepaalde chipset van de 80286, zoals NEAT AT/386 en LEAP CHIPset van Chips and Technologies. Veel van dergelijke programma's kunnen ook gebruik maken van de expanded geheugenhardware van LIM-EMS 4.0. Twee van de populairste programma's zijn QRAM van Quaterdeck Office Systems en MOVE'EM van Qualitas. Zowel QRAM als MOVE'EM brengen het vermogen van de UMB's binnen het bereik van deze oudere systemen.

De geheugeninstellingen bekijken met MEM

Het nauwkeurig afstellen van de geheugenconfiguratie van uw systeem is een zaak

van vallen en opstaan. U bekijkt het beschikbare geheugen, verandert enkele instellingen, herstart het systeem en ziet of het vrije geheugen groter is geworden.

Met MEM van DOS 6 kunt u het geheugenverbruik van uw computer in kaart brengen. Wat er door MEM wordt weergegeven, hangt af van de schakelopties die u hebt gebruikt. Wanneer u MEM zonder schakelopties start, zal er een kort overzicht worden weergegeven, iets in de trant van het onderstaande:

Memory Type	Total =	Used +	Free
-----	-----	-----	-----
Conventional	640K	116K	524K
Upper	155K	80K	75K
Adapter RAM/ROM	384K	384K	0K
Extended (XMS)	7013K	7013K	0K
-----	-----	-----	-----
Total memory	8192K	7593K	599K
 Total under 1 MB	 795K	 196K	 599K
 Largest executable program size			524K (536272 bytes)
Largest free upper memory block			64K (65536 bytes)
MS-DOS is resident in the high memory area.			

Bij de opdracht MEM zijn de volgende schakelopties beschikbaar:

- /C Laat zien hoeveel basisgeheugen en hoger geheugen wordt gebruikt door elk actief programma en stuurprogramma, en geeft een overzicht van het algemene geheugenverbruik. De lange vorm van deze schakeloptie is /CLASSIFY.
- /D Laat gedetailleerde informatie zien over elk actief programma en stuurprogramma, waaronder de grootte van het item, het segmentadres en het type geheugen. De lange vorm van deze schakeloptie is /DEBUG.
- /F Laat de gebieden in het basisgeheugen en het hogere geheugen zien die vrij zijn. De lange vorm van deze schakeloptie is /FREE.
- /M Geeft complete informatie over het geheugen voor een bepaald programma. U moet de naam van het programma of stuurprogramma opgeven, voorafgegaan door de schakeloptie /M. De lange vorm van deze schakeloptie is /MODULE.
- /P Laat MEM wachten als het scherm vol is, met andere woorden door deze schakeloptie wordt voorkomen dat de informatie van het scherm

rolt. De lange vorm van deze schakeloptie is /PAUSE.



MEM in DOS 5 ondersteunt niet dezelfde schakelopties als de versie in DOS 6. In DOS 5 is de schakeloptie /P de afkorting van /PROGRAM, waarmee MEM gedetailleerde informatie weergeeft over elk programma dat in het geheugen is geladen. Bij DOS 5 wordt als alternatief voor de schakeloptie /P het pijlpijnteken |MORE gebruikt. De schakelopties /C en /D worden wel ondersteund, maar over het algemeen geeft MEM in DOS 6 meer informatie dan zijn tegenhanger in DOS 5.

Zie deel 4 *Alle DOS-opdrachten* voor een volledige beschrijving van elke schakeloptie bij MEM.

Voor het nauwkeurig afregelen van een computergeheugen worden voornamelijk de schakelopties /C en /D gebruikt. Wanneer een programma geheugenproblemen kent, kan met /D gedetailleerde informatie worden opgevraagd. De volgende lijst is gegenereerd met de opdracht MEM en de schakeloptie /C:

Modules using memory below 1 MB:

Name	Total		=	Conventional		+	Upper Memory	
-----	-----	-----		-----	-----		-----	-----
MSDOS	19709	(19K)		19709	(19K)		0	(0K)
HIMEM	1168	(1K)		1168	(1K)		0	(0K)
EMM386	3120	(3K)		3120	(3K)		0	(0K)
NCACHE2	28928	(28K)		28928	(28K)		0	(0K)
SMARTCAN	10288	(10K)		10288	(10K)		0	(0K)
KEYB	6224	(6K)		6224	(6K)		0	(0K)
COMMAND	2912	(3K)		2912	(3K)		0	(0K)
DOSSHELL	4704	(5K)		4704	(5K)		0	(0K)
DOSSWAP	35072	(34K)		35072	(34K)		0	(0K)
COMMAND	3504	(3K)		3504	(3K)		0	(0K)
COMMAND	3152	(3K)		3152	(3K)		0	(0K)
SETVER	816	(1K)		0	(0K)		816	(1K)
NAV&	8144	(8K)		0	(0K)		8144	(8K)
GMOUSE	14960	(15K)		0	(0K)		14960	(15K)
DBLSPACE	44384	(43K)		0	(0K)		44384	(43K)
SHARE	13888	(14K)		0	(0K)		13888	(14K)
Free	613056	(599K)		536368	(524K)		76688	(75K)

Memory Summary:

Type of Memory	Total		=	Used		+	Free	
-----	-----			-----			-----	
Conventional	655360	(640K)		118992	(116K)		536368	(524K)
Upper	158880	(155K)		82192	(80K)		76688	(75K)
Adapter RAM/ROM	393216	(384K)		393216	(384K)		0	(0K)
Extended (XMS)	7181152	(7013K)		7181152	(7013K)		0	(0K)
-----	-----			-----			-----	
Total memory	8388608	(8192K)		7775552	(7593K)		613056	(599K)
Total under 1 MB	814240	(795K)		201184	(196K)		613056	(599K)
Largest executable program size				536272	(524K)			
Largest free upper memory block				65536	(64K)			
MS-DOS is resident in the high memory area.								

De grootste geheugenwinst

Met enkele heel eenvoudige manieren kan op een 386 de hoeveelheid beschikbaar geheugen worden uitgebreid. Voordat u zich gaat bezighouden met het bij elkaar sprokkelen van 1K geheugen hier en 2K geheugen daar, kunt u zich beter bezighouden met de grote brokken.

Het geheugen van een computer wordt geconfigureerd door de startbestanden AUTOEXEC.BAT en CONFIG.SYS. Door enkele simpele aanpassingen in CONFIG.SYS kunt u een enorme geheugenwinst behalen.

DOS hoog laden

De eerste winst behaalt u door munt te slaan uit het vermogen van de HMA en DOS hoog te laden. Dit is de enige functie in DOS 6 die door gebruikers van een 286 kan worden toegepast. Om DOS hoog te laden moet u HIMEM.SYS installeren. Dan pas kunt u DOS instrueren zichzelf hoog te laden door het toevoegen van de regel DOS=HIGH in het bestand CONFIG.SYS. Als DOS hoog wordt geladen, worden de buffers ook hoog geladen.

In het volgende bestand CONFIG.SYS wordt deze techniek geïllustreerd:

```
FILES=50
BUFFERS=15
DEVICE=C:\WINDOWS\MOUSE.SYS /Y
DEVICE=C:\DOS\SMARTDRV.EXE
/DOUBLE_BUFFER
```

Wanneer een computer met dit bestand wordt gestart, geeft MEM aan dat er ongeveer 571.000 bytes van het basisgeheugen vrij is. Dat is niet veel gezien het feit dat alleen DOS, een diskcache en een stuurprogramma voor een muis zijn geladen. MEM laat als volgt zien hoe dit geheugen over deze programma's is verdeeld:

Name	Total	=	Conventional	+	Upper Memory
-----	-----		-----		-----
MSDOS	60093 (59K)		60093 (59K)		0 (0K)
MOUSE	17088 (17K)		17088 (17K)		0 (0K)
SMARTDRV	2480 (2K)		2480 (2K)		0 (0K)
COMMAND	5024 (5K)		5024 (5K)		0 (0K)
Free	570704 (557K)		570704 (557K)		0 (0K)

Nadat de CONFIG.SYS is aangepast om DOS hoog te laden, ziet dit bestand er zo uit:

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\WINDOWS\MOUSE.SYS /Y
```

```
DEVICE=C:\DOS\SMARTDRV.EXE
/DOUBLE_BUFFER
BUFFERS=15,0
FILES=60
DOS=HIGH
```

Door alleen de opdrachten HIMEM.SYS en DOS=HIGH toe te voegen, wordt het vrije geheugen bijna 616.000 bytes. Het geheugenverbruik wordt nu als volgt door MEM weergegeven:

Name	Total	=	Conventional	+	Upper Memory
-----	-----		-----		-----
MSDOS	15757 (15K)		15757 (15K)		0 (0K)
HIMEM	1200 (1K)		1200 (1K)		0 (0K)
MOUSE	17088 (17K)		17088 (17K)		0 (0K)
SMARTDRV	2480 (2K)		2480 (2K)		0 (0K)
COMMAND	2960 (3K)		2960 (3K)		0 (0K)
Free	615920 (601K)		615920 (601K)		0 (0K)

U ziet dat met een hoog geladen DOS er een restje van 15K in het basisgeheugen is achtergebleven.

UMB's gebruiken

Het hogere geheugen is ontworpen voor het videogeheugen, de BIOS en de apparatuur die wordt aangesloten, maar een groot deel van het hogere geheugen wordt niet gebruikt. De tweede grote winst behaalt u door de lege plekken in het

hogere geheugen op te vullen met extended geheugen. Dit wordt het maken van upper memory blocks, oftewel UMB's, genoemd.

U kunt met het programma EMM386.EXE UMB's maken door een opdracht EMM386 aan het bestand CONFIG.SYS toe te voegen. EMM386 voert veel functies uit; met de schakeloptie /RAM converteert EMM386 extended geheugen naar UMB's. Nadat EMM386 is geïnstalleerd, kunt u DOS instrueren de UMB's te beheren door het in het bestand CONFIG.SYS de opdracht DOS= te veranderen in DOS=HIGH,UMB.



Als u met een programma van een andere producent werkt en niet met EMM386, zult u gewoonlijk de opdracht DOS=UMB niet gebruiken. De geheugenbeheerder zal de UMB-ruimte apart beheren.



Wanneer u bestaande CONFIG.SYS-bestanden bekijkt, zult u twee verschillende opdrachten DOS=zien. De ene opdracht is DOS=HIGH en de andere is DOS=UMB. Deze twee afzonderlijke opdrachten hebben hetzelfde effect als de enkele opdracht DOS=HIGH,UMB. Deze twee aparte opdrachten worden vaak gemaakt door het

geheugen-optimaliseringsprogramma van DOS, MEMMAKER (dat verderop wordt besproken). MEMMAKER voegt een tweede opdracht DOS= toe als het component UMB ontbreekt.

Na het maken van de UMB's kunt u deze benutten door stuurprogramma's en programma's in de UMB's te laden. Programma's die in de UMB's zijn geladen, zijn *hoog geladen*. Met de DOS-opdrachten DEVICEHIGH en LOADHIGH worden stuurprogramma's en programma's respectievelijk hoog geladen. In de voorbeeldconfiguratie is MOUSE.SYS het enige geladen stuurprogramma. Dit kan worden geladen in de UMB's door de opdracht DEVICE te veranderen in DEVICEHIGH.

Het voorbeeldbestand CONFIG.SYS wordt als volgt bijgewerkt om volledig gebruik te kunnen maken van de UMB's:

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE RAM
DEVICE=C:\DOS\SMARTDRV.EXE
/DOUBLE_BUFFER
BUFFERS=15,0
FILES=50
DOS=HIGH,UMB
```

```
DEVICEHIGH=C:\WINDOWS\MOUSE.SYS /Y
```

Het effect van deze veranderingen op het vrije geheugen worden door MEM samengevat in onderstaand overzicht:

Name	Total	=	Conventional	+	Upper Memory
-----	-----		-----		-----
MSDOS	15757 (15K)		15757 (15K)		0 (0K)
HIMEM	1200 (1K)		1200 (1K)		0 (0K)
EMM386	3072 (3K)		3072 (3K)		0 (0K)
SMARTDRV	2480 (2K)		2480 (2K)		0 (0K)
COMMAND	2960 (3K)		2960 (3K)		0 (0K)
MOUSE	17120 (17K)		0 (0K)		17120 (17K)
Free	706080 (690K)		629920 (615K)		76160 (74K)

Memory Summary:

Type of Memory	Total	=	Used	+	Free
-----	-----		-----		-----
Conventional	655360 (640K)		25440 (25K)		629920 (615K)
Upper	93280 (91K)		17120 (17K)		76160 (74K)
Adapter RAM/ROM	299936 (293K)		299936 (293K)		0 (0K)
Extended (XMS)	15466496 (15104K)		589824 (576K)		14876672 (14528K)
-----	-----		-----		-----
Total memory	16515072 (16128K)		932320 (910K)		15582752 (15218K)

Zoals u ziet, eigent EMM386 zich 3K basisgeheugen toe, maar dit wordt dubbel en dwars teruggewonnen door het stuurprogramma van de muis in de UMB's te laden. En er komt een extra 74K UMB's beschikbaar voor andere

stuurprogramma's en programma's, zoals SHARE, netwerkshells, DOSKEY, enzovoorts.

Van te weinig RAM naar te weinig UMB

In de dagen van de 8086/80286 klaagde iedereen over het tekort aan RAM. Met DOS, residentie programma's zoals Sidekick en grote programma's was er nooit genoeg RAM. Tot de UMB's verschenen en het tekort aan RAM tot het verleden behoorde.

Als u naar een grotere woning verhuist, vraagt u zich misschien af wat u met al die ruimte moet. Dat duurt maar even, want voordat u het in de gaten hebt, wordt er weer geklaagd over te weinig ruimte. Hoe groter de woning, des te meer spullen. Ditzelfde geldt voor de UMB's. Nu klaagt men over te weinig UMB's: te veel programma's en te weinig UMB's.

Het gebruik van de beschikbare UMB's optimaliseren is makkelijker gezegd dan gedaan.

Ruimte in de UMB's bepalen

Het hogere geheugen is een mijnenveld in het geheugen: hier de ROM-BIOS, daar lege ruimte, ergens anders het videogeheugen en vervolgens weer een lege plaats. Om het nog ingewikkelder te maken is de organisatie van het hogere geheugen per systeem verschillend. Over het algemeen staat de ROM-BIOS in het bereik F000-FFFF, het bereik A000-BFFF wordt gebruikt voor het videogeheugen en C000-CFFF voor de BIOS van de video en de schijf.



EMM386 is verantwoordelijk voor het plaatsen van het extended geheugen in de ongebruikte gebieden van het hogere geheugen. Met de instructies I= en X= kunt u bepaalde gebieden van het hogere geheugen respectievelijk opnemen of uitsluiten.

Standaard ondersteunde EMM386 van DOS 5 het hogere geheugen alleen in het bereik C000-DFFF. Alleen met de schakeloptie I= konden de andere gebieden worden benaderd. In veel boeken en bladen werd de tip gegeven dat met de schakeloptie I=E000-EFFF een extra 64K van de UMB-ruimte kon worden gewonnen. In DOS 6 ondersteunt

EMM386 ook dit bereik E000-EFFF, zodat u het niet expliciet hoeft op te geven.

Laadvolgorde en lokatie besturen

Omdat UMB's in de ongebruikte delen van het hogere geheugen worden geplaatst, zijn ze gefragmenteerd omdat enkele geheugenblokken op het ene en andere blokken op een ander adres staan. Deze verschillende gebieden van de UMB's worden *regio's* genoemd.

De opdracht MEM geeft waardevolle informatie over de vrije UMB-ruimte. Hieronder staat een voorbeeld van de opdracht MEM /F, met één regio in de UMB:

Free Conventional Memory:

Segment	Total	
-----	-----	-----
00F1B	80	(0K)
01D02	240	(0K)
01D11	88608	(87K)
032B3	447680	(437K)
Total Free:	536608	(524K)

Free Upper Memory:

Region	Largest Free	Total Free	Total Size
1	65536 (64K)	76688 (75K)	158880 (155K)

Standaard wordt met de opdrachten `DEVICEHIGH` of `LOADHIGH` het programma in de grootste beschikbare regio geplaatst. Wanneer er eerst een klein programma wordt geladen, wordt de grootste regio gebruikt, zodat de overgebleven ruimte niet voldoende is om een tweede, groter programma te laden. Als dat het geval is, wordt het tweede programma in het basisgeheugen geladen. Wanneer u de programma's omgekeerd laadt, kunnen beide hoog worden geladen.

Bij `DEVICEHIGH` en `LOADHIGH` kunt u de schakeloptie gebruiken waarmee u kunt bepalen in welke regio een programma zal worden geladen.

Geheugentoe wijzing van UMB

Een effectief beheer van de UMB wordt nog ingewikkelder gemaakt door het feit dat voor het laden van bepaalde programma's meer geheugen nodig is, maar na de installatie juist minder. Het is

iets zoals een bedrijf waarvoor een heel gebouw moet worden neergezet, maar dat vervolgens maar voor de helft wordt gebruikt. Het omgekeerde is ook het geval: bepaalde programma's hebben meer ruimte nodig om te draaien dan om geladen te worden.

Met de schakeloptie `/L` van `DEVICEHIGH` en `LOADHIGH` kunt u bepalen hoeveel geheugen er minstens beschikbaar moet zijn in de regio om het programma hoog te laden. Dit is noodzakelijk voor programma's die met minder geheugen genoeg nemen bij het laden dan wanneer er met ze wordt gewerkt. In samenhang met de schakeloptie `/L` kunt u met `/S` de grootte van het programma tot een minimum verkleinen wanneer het is geladen.

DBLSPACE.BIN hoog laden

Met een speciale techniek kan het stuurprogramma voor de schijfcompressie in de UMB's worden geladen.

Het stuurprogramma `DBLSPACE.SYS` van DOS 6 heeft maar één doel in zijn leven: `DBLSPACE.BIN` naar het hogere geheugen verplaatsen. De opdracht

DBLSPACE.SYS moet worden toegevoegd aan het bestand CONFIG.SYS en kan zowel met een opdracht DEVICE= in het conventionele geheugen als met een opdracht DEVICEHIGH= in het hogere geheugen worden geladen.

```
DEVICEHIGH=C:\DOS\DBLSPACE.SYS /MOVE
```

De schakeloptie /MOVE is eigenlijk een commentaar. Schakelopties bij DBLSPACE.SYS worden genegeerd. De ontwikkelaars van DOS hebben de schakeloptie /MOVE toegevoegd ter verduidelijking van het doel van het stuurprogramma. Het is geen stuurprogramma voor de DoubleSpace-stations, maar een stuurprogramma voor het *verplaatsen* van het stuurprogramma in het hogere geheugen.



Tijdens het starten van het systeem wordt DBLSPACE.BIN eerder door IO.SYS geladen dan CONFIG.SYS. Het wordt aanvankelijk helemaal boven in het geheugen geladen, naast het geheugengebied waarin IO.SYS staat. Het moet worden geladen in het basisgeheugen, omdat zo vroeg in de startcyclus de UMB's nog niet beschikbaar zijn. Wanneer er in de CONFIG.SYS geen stuurprogramma DBLSPACE.SYS staat,

wordt DBLSPACE.BIN naar de onderkant van het basisgeheugen verplaatst nadat alle opdrachten in CONFIG.SYS zijn verwerkt.

MEMMAKER het werk laten doen

Een volledige optimalisering van de UMB's is een nauwgezet, ingewikkeld en arbeidsintensief proces. In feite kan dit veel beter door de computer worden uitgevoerd.

Gelukkig hebben de ontwikkelaars van DOS dit ook ingezien en daarvoor een nieuw programma ontworpen, MEMMAKER. Het enige wat dit programma doet, is het geheugenverbruik op uw computer analyseren en vervolgens de bestanden CONFIG.SYS en AUTOEXEC.BAT aanpassen voor een optimaal gebruik van het geheugen. De ontwikkelaars zijn erin geslaagd een uitstekend programma te maken.

MEMMAKER in actie

Hieronder staan de bestanden CONFIG.SYS en AUTOEXEC.BAT van een 386 die niet optimaal is geconfigureerd:

CONFIG.SYS:

```
FILES=50
BUFFERS=15
DEVICE=C:\WINDOWS\MOUSE.SYS /Y
DEVICE=C:\DOS\SETVER.EXE
LASTDRIVE=M
```

AUTOEXEC.BAT

```
@ECHO OFF
PATH
C:\DOS;C:\;C:\WINDOWS;C:\WINDOWS\NDW;C
:\BATFILES
PROMPT $P$G
DOSKEY
SET SAGEEDIT=C:\SPE
SET NU=C:\U\NORTON
SET TEMP=C:\TEMP
```

Het vrije geheugen is 567.520 bytes; MEM laat de volgende statistiek zien:

Memory Type	Total	= Used	+ Free
-----	-----	-----	-----
Conventional	640K	86K	554K
Upper	0K	0K	0K
Adapter RAM/ROM	384K	384K	0K
Extended (XMS)	15104K	15104K	0K
Expanded (EMS)	0K	0K	0K
-----	-----	-----	-----
Total memory	16128K	15574K	554K
Total under 1 MB	640K	86K	554K
Largest executable program size		554K	(567520 bytes)
Largest free upper memory block		0K	(0 bytes)
The high memory area is available.			

Nadat u MEMMAKER in Express Mode hebt gestart, zijn de configuratiebestanden als volgt gewijzigd:

CONFIG.SYS:

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE RAM
WIN=DD00-DFFF WIN=D800-DCFF
BUFFERS=15,0
FILES=50
DOS=UMB
LASTDRIVE=M
DEVICEHIGH /L:1,55168
=C:\WINDOWS\MOUSE.SYS /Y
DEVICEHIGH /L:1,12048
=C:\DOS\SETVER.EXE
```

AUTOEXEC.BAT:

```
@ECHO OFF
PATH
C:\DOS;C:\;C:\WINDOWS;C:\WINDOWS\NDW;C
:\BATFILES
PROMPT $P$G
LH /L:1,6400 DOSKEY
SET SAGEEDIT=C:\SPE
SET NU=C:\U\NORTON
SET TEMP=C:\TEMP
```

Het vrije geheugen wordt vergroot naar 583.216 bytes en MEM laat de volgende statistiek zien:

Memory Type	Total	=	Used	+	Free
Conventional	640K		70K		570K
Upper	59K		21K		38K
Adapter RAM/ROM	325K		325K		0K
Extended (XMS)	15104K		15104K		0K
Expanded (EMS)	0K		0K		0K
Total memory	16128K		15520K		608K
Total under 1 MB	699K		92K		607K

EMS is active
 Largest executable program size 570K (583216 bytes)
 Largest free upper memory block 38K (38464 bytes)
 The high memory area is available.

De voornaamste veranderingen in CONFIG.SYS zijn de opdrachten HIMEM.SYS en EMM386 die

zijn toegevoegd, DOS=UMB en de opdracht DEVICEHIGH waarmee MOUSE.SYS en SETVER.EXE in de hogere geheugenblokken worden geladen. De enige verandering in AUTOEXEC.BAT was dat door LOADHIGH (de afkorting LH) DOSKEY in de hogere geheugenblokken wordt geladen.

De schakeloptie /L wordt bij elke DEVICEHIGH en LOADHIGH gebruikt voor de besturing van de regio en het minimum geheugen dat voor elk programma nodig is.

Nu u weet hoe u de grootste geheugenwinst kunt behalen, zult u misschien hebben gemerkt dat er iets in CONFIG.SYS ontbreekt. HIMEM.SYS werd geïnstalleerd, maar DOS werd niet in het hoge geheugen geladen. Wanneer u de regel DOS=UMB zelf verandert in DOS=HIGH,UMB, wordt het vrije geheugen 631.648 bytes.

Memory Type	Total	= Used	+ Free
-----	-----	-----	-----
Conventional	640K	23K	617K
Upper	59K	21K	38K
Adapter RAM/ROM	325K	325K	0K
Extended (XMS)	15104K	544K	14560K
Expanded (EMS)	0K	0K	0K
-----	-----	-----	-----
Total memory	16128K	913K	15215K
Total under 1 MB	699K	44K	655K
EMS is active			
Largest executable program size		617K	(631648 bytes)
Largest free upper memory block		38K	(38464 bytes)
MS-DOS is resident in the high memory area.			

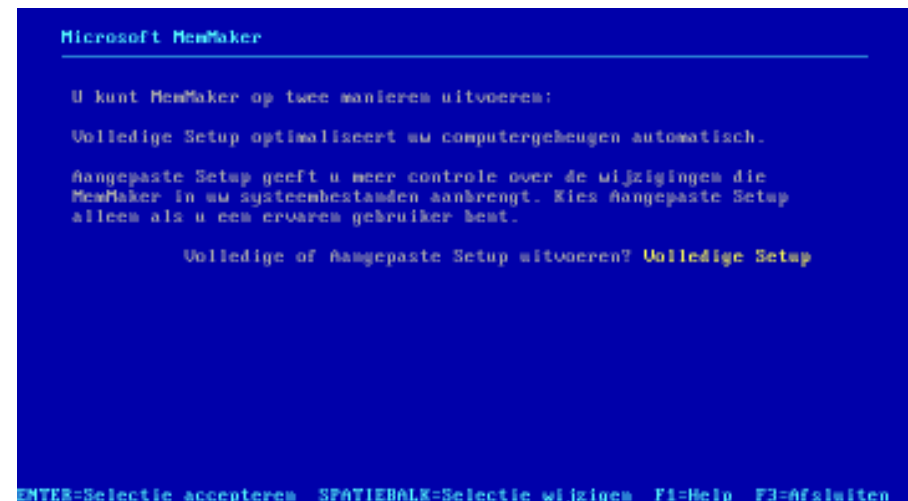
U zult zich afvragen waarom MEMMAKER een dergelijk belangrijke verbetering over het hoofd ziet. Wanneer DOS 6 voor de eerste keer wordt geïnstalleerd, wordt de opdracht DOS=HIGH aan het bestand CONFIG.SYS toegevoegd. De ontwikkelaars van DOS namen aan dat er een reden moet zijn waarom DOS=HIGH is verwijderd, en dus wordt deze opdracht niet door MEMMAKER toegevoegd als deze ontbreekt.

MEMMAKER uitvoeren

MEMMAKER behoort tot die categorie programma's die heel veel mogelijkheden hebben en toch zo makkelijk in gebruik zijn. Het

programma staat in de DOS-directory en wordt normaal gesproken gestart zonder schakelopties of andere opties. De enige toetsen die u dient te onthouden, zijn de spatiebalk (voor het schakelen tussen menu-opties), F3 om het programma af te sluiten en Enter om de selecties door te voeren.

Het eerste scherm in MEMMAKER is een welkomstscherf. Door op een toets te drukken verschijnt het hoofdscherf (zie afbeelding 5-2) waarin u moet kiezen tussen *Express Mode* of *Custom Mode*.



Figuur 5-2: Memmaker in Express of Custom Mode

MEMMAKER stelt de volgende vraag, ongeacht de modus die u hebt gekozen:

Do you use any program that need
expanded memory (EMS)?

Eén van de functies van EMM386 is het converteren van extended naar expanded geheugen. Zoals u zich misschien nog wel herinnert, gebruikt het expanded geheugen voor het pagineren een pagina-frame van 64K in het hogere geheugen. Wanneer u niet met programma's werkt die gebruik maken van EMS, kunt u net zo goed dit pagina-frame verwijderen en het vrijgekomen geheugen als UMB's benutten. Wanneer u deze vraag met No beantwoordt, wordt er een schakeloptie /NOEMS aan EMM386 toegevoegd.

Nadat deze verandering is doorgevoerd, kunnen bepaalde programma's trager werken dan normaal. Dit is een goede indicatie dat zij toch gebruik maakten van het expanded geheugen. Verwijder de schakeloptie /NOEMS uit het bestand CONFIG.SYS, herstart het systeem en controleer of de programma's weer op de oude snelheid werken. Als dat zo is, zou u MEMMAKER opnieuw kunnen

starten en de vraag over de EMS met YES beantwoorden.

Wanneer u Express Mode selecteert, gaat MEMMAKER gewoon verder met het werk. Met Custom Mode hebt u een vinger in de pap. Zie afbeelding 5-3 voor de beschikbare opties in Custom Mode. Met de cursortoetsen kunt u een andere selectie maken; met de spatiebalk stelt u de instelling in op Yes of No.



Figuur 5-3: Configuratie-opties in Custom Mode

Jammer genoeg wordt de bestaande instelling niet eerst door MEMMAKER bekeken voordat de opties worden weergegeven. Daarom kunnen de waarden van de configuratie-opties verschillen van de instellingen in het bestand CONFIG.SYS.

In hoofdstuk 7 van de DOS-handleiding wordt een goede beschrijving gegeven van elke optie. Twee opties verdienen echter speciale aandacht: Optimize upper memory for use with Windows en Use the monochrome region.

Het hogere geheugen optimaliseren voor Windows

Kort gezegd moet u deze optie selecteren als u de meeste tijd met Windows werkt en geen geheugen tekort komt wanneer u met DOS buiten Windows werkt. MEMMAKER zal één of meer schakelopties WIN= toevoegen aan de opdracht EMM386 in het bestand CONFIG.SYS, waarmee wordt aangegeven dat de UMB-ruimte is gereserveerd voor Windows.



Windows moet vertaalbuffers maken in de eerste megabyte geheugen en deze vertaalbuffers worden om redenen van snelheid uitgelijnd op een grens van 4K. Deze buffers kunnen echter in de UMB's

liggen. Zowel EMM386 als DOS slaan informatie, *arena's* genoemd, aan het begin en einde van elke UMB op. Door deze arena's worden het begin en einde van het vrije geheugen in de UMB *niet* uitgelijnd op de 4K. De eerste 4K en de laatste 4K van elke vrije ruimte in de UMB kunnen daarom niet door Windows worden benut voor vertaalbuffers.

Als met de schakeloptie X= een regio wordt buitengesloten, zal Windows 3.1 daar geen vertaalbuffers plaatsen. De informatie EMMIMPORT van EMM386 zorgt ervoor dat deze regio niet beschikbaar is. Door de schakeloptie WIN= wordt er niets naar de UMB's overgebracht en wordt aan Windows meegedeeld dat het adres kan worden gebruikt voor vertaalbuffers. Zo kan Windows de UMB-ruimte volledig benutten zonder dat er 8K verloren gaat. Het compromis is dat u dit gebied niet voor UMB-geheugen kunt gebruiken als er niet met Windows wordt gewerkt (in een standaard sessie van DOS).

De monochroom-regio

Het B-gebied van het hogere geheugen, dat wil zeggen het bereik van 0xB000 tot 0xBFFF, is

gereserveerd voor het videogeheugen. De eerste helft van dit bereik, 0xB000-0xB7FF, is gereserveerd voor monochroom systemen; de rest, 0xB800-0xBFFF voor kleurensystemen. Wanneer u met een kleurenscherm werkt, zult u waarschijnlijk het monochroom gebied met UMB's kunnen opvullen. Als u deze optie op Yes instelt, voegt MEMMAKER I=B000-B7FF toe aan de opdracht EMM386. Op systemen waarop Windows is geïnstalleerd, voegt MEMMAKER ook de regel DEVICE=MONOUMB2.386 toe aan de sectie [386enh] in het bestand SYSTEM.INI. Zonder deze aanpassing is er niet met Windows te werken.



Wanneer u met Windows werkt en u voegt zelf de opdracht I=B000-B7FF toe aan de opdracht EMM386, vergeet dan ook niet het bestand SYSTEM.INI aan te passen en de regel DEVICE=MONOUMB2.386 toe te voegen aan de sectie [386enh].

Een enkele keer kan MEMMAKER ook de configuratiebestanden wijzigen waardoor uw systeem vastloopt. Raadpleeg in een dergelijke situatie de paragraaf *Geheugenconflicten oplossen* verderop in dit hoofdstuk.



Het principe van MEMMAKER

In een eerder aan bod gekomen voorbeeld verrichtte MEMMAKER zeker geen wonderen. Als u echter een ingewikkelde configuratie hebt met een groot aantal verschillende stuurprogramma's en programma's die bij het starten van het systeem worden geladen, doet MEMMAKER dat wel degelijk. MEMMAKER kan duizenden configuraties evalueren voordat er een optimale configuratie wordt gekozen.

De meest gecompliceerde taak die MEMMAKER op zich neemt, is het bepalen van de ruimte die nodig is voor programma's en stuurprogramma's, zowel tijdens het laden als daarna. Hiervoor roept MEMMAKER de hulp in van twee speciale utility's, te weten CHKSTATE.SYS en SIZER.EXE.

Aan het begin van het bestand CONFIG.SYS wordt door MEMMAKER een nieuwe regel toegevoegd om het stuurprogramma CHKSTATE te laden. Als de opdrachten HIMEM.SYS, EMM386 en DOS=UMB ontbreken, worden ze door MEMMAKER toegevoegd. Ook wordt elke DEVICE-opdracht aangepast, zodat met

SIZER.EXE de stuurprogramma's worden geladen en gecontroleerd.

Hieronder staat hoe het bestand CONFIG.SYS, dat als voorbeeld dienst doet, wordt aangepast door MEMMAKER tijdens het optimaliseren:

CONFIG.SYS:

```
DEVICE=C:\DOS\CHKSTATE.SYS /S:SR1
/14277
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE
BUFFERS=15,0
FILES=50
DOS=UMB
LASTDRIVE=M
DEVICE=C:\DOS\SIZER.EXE /14277 /8
C:\WINDOWS\MOUSE.SYS /Y
DEVICE=C:\DOS\SIZER.EXE /14277 /9
C:\DOS\SETVER.EXE
```

Het bestand AUTOEXEC.BAT wordt ook door MEMMAKER aangepast. Zoals met de stuurprogramma's het geval is, worden ook alle opdrachten voor het starten van programma's, onder andere batchbestanden, gewijzigd en worden ze met USER.EXE geladen. Aan het einde van

AUTOEXEC.BAT wordt er een regel toegevoegd waarmee MEMMAKER wordt opgeroepen, zodat het optimaliseringsproces gewoon verder gaat.

AUTOEXEC.BAT:

```
@ECHO OFF
PATH
C:\DOS;C:\;C:\WINDOWS;C:\WINDOWS\NDW;C
:\BATFILES
PROMPT $P$G
C:\DOS\SIZER.EXE /14277 /3 DOSKEY
SET SAGEEDIT=C:\SPE
SET NU=C:\U\NORTON
SET TEMP=C:\TEMP
C:\DOS\MEMMAKER.EXE /SESSION:14277
```

Nadat de configuratiebestanden zijn gewijzigd, wordt het systeem door MEMMAKER herstart. Vervolgens gaan CHKSTATE.SYS en SIZER.EXE informatie verzamelen en bekijken als de programma's en stuurprogramma's worden geladen. Op basis van deze gegevens berekent MEMMAKER welke programma's er in het hogere geheugen zullen worden geladen, in welke volgorde en hoeveel geheugen daar voor nodig is.

Uitgaande van de berekeningen worden er verdere wijzigingen aan de configuratiebestanden aangebracht. De voornaamste verandering is de verwijdering van de opdrachten voor SIZER.EXE, die worden vervangen door opdrachten voor DEVICEHIGH en LOADHIGH met de desbetreffende schakelopties. Hieronder staan de configuratiebestanden zoals zij zijn na de tweede aanpassingsronde:

CONFIG.SYS:

```
DEVICE=C:\DOS\CHKSTATE.SYS /S:FR1
/27107
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE RAM
WIN=DD00-DFFF WIN=D800-DCFF
BUFFERS=15,0
FILES=50
DOS=UMB
LASTDRIVE=M
DEVICEHIGH /L:1,55168
=C:\WINDOWS\MOUSE.SYS /Y
DEVICEHIGH /L:1,12048
=C:\DOS\SETVER.EXE
```

AUTOEXEC.BAT:

```
@ECHO OFF
```

```
PATH
C:\DOS;C:\;C:\WINDOWS;C:\WINDOWS\NDW;C
:\BATFILES
PROMPT $P$G
LH L:1,6400 DOSKEY
SET SAGEEDIT=C:\SPE
SET NU=C:\U\NORTON
SET TEMP=C:\TEMP
C:\DOS\MEMMAKER.EXE /SESSION:27107
```

Het systeem wordt voor een tweede maal door MEMMAKER gestart en vraagt u te controleren of alles goed werkt. Als u hier bevestigend op antwoordt, wordt de opdracht CHKSTATE aan het begin van het bestand CONFIG.SYS verwijderd. U krijgt een overzicht van de geheugenwinst en keert vervolgens terug naar de DOS-prompt. Nu kunt u het nieuwe verworven geheugen gaan gebruiken. Als het systeem niet stabiel blijkt te zijn en u beantwoordt de vraag met No, kunnen de oorspronkelijke configuratiebestanden worden teruggehaald en wordt het systeem herstart. Zo wordt de vorige status van het systeem hersteld voordat MEMMAKER werd gestart.

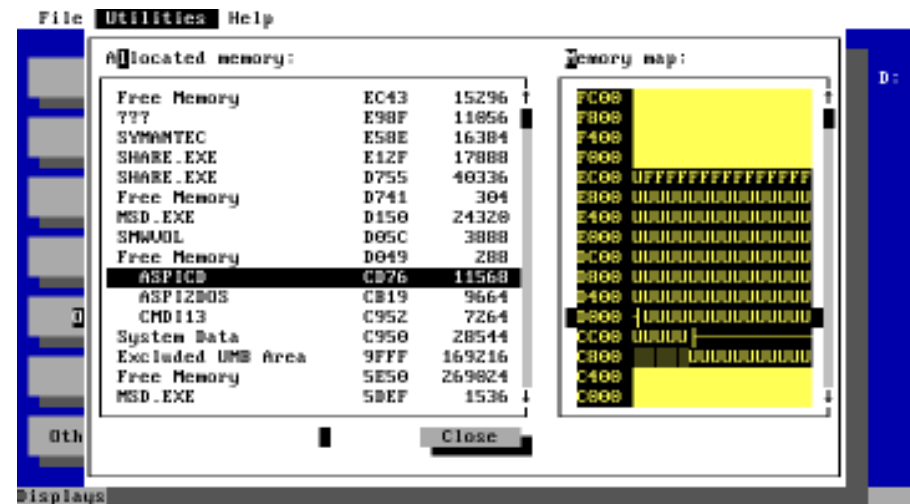
Achter de schermen wordt in de DOS-directory door MEMMAKER het bestand



MEMMAKER.STS gemaakt. Hierin staan alle analysegegevens die bij elkaar zijn gebracht door CHKSTATE en SIZER. Hiermee kunt u de bijzonderheden bekijken van elk programma en stuurprogramma zoals deze tijdens het starten van het systeem zijn.

CHKSTATE.SYS plaatst het overzicht boven aan het bestand, waarna SIZER de uitgebreide informatie over elke opdracht invoert.

Het diagnoseprogramma MSD.EXE van DOS 6 laat zien hoe het basisgeheugen en het hogere geheugen worden gebruikt, en waar de programma's en stuurprogramma's daadwerkelijk liggen opgeslagen in het geheugen. Start MSD en selecteer de optie Memory Block Display in het menu Utilities. Wanneer u door de lijst met geïnstalleerde programma's bladert, laat het blokdiagram zien waar het programma is opgeslagen. In afbeelding 5-5 wordt de lokatie van de device driver ASPICD aangegeven door |----| in het segment CC00.



Figuur 5-5: Welke programma's zitten er in het geheugen?

MEMMAKER een handje helpen

Alhoewel MEMMAKER een intelligent programma is, kunnen bepaalde alledaagse situaties een negatieve uitwerking hebben. In de volgende paragraaf wordt aangegeven wanneer u MEMMAKER een duwtje in de rug kunt geven.

Vermijd meerdere configuraties in CONFIG.SYS

MEMMAKER verliest aan kracht als er in het bestand CONFIG.SYS een menu met opties staat (zie voor een volledige beschrijving hoofdstuk 8).

Een CONFIG.SYS waarin meer dan één configuratie staat, kan meerdere ingangen hebben voor HIMEM.SYS, EMM386, enzovoorts. Wanneer MEMMAKER deze situatie tegenkomt, worden de eerste HIMEM.SYS en EMM386.EXE naar het begin van het bestand gekopieerd. Wanneer u opgeeft dat de schakelopties I (inclusief) en X (exclusief) voor EMM386 moeten worden gehandhaafd, worden de schakelopties van de eerste opdracht EMM386 gebruikt. Het vreemde is dat alle overige opdrachten HIMEM.SYS en EMM386.EXE in CONFIG.SYS inactief worden gemaakt met REM.

Soms zal het nodig zijn meerdere configuraties te laten optimaliseren door MEMMAKER. U kunt bijvoorbeeld een configuratie hebben waarmee geen netwerkstuurprogramma's worden geladen, een andere configuratie waarmee een stuurprogramma voor een scanner wordt geladen en misschien een derde waarmee alles wordt geladen.

MEMMAKER kan alle meervoudige opties configureren, maar dat gaat niet automatisch. De beste handelswijze is een reservekopie te maken van CONFIG.SYS en AUTOEXEC.BAT en vervolgens met uitgeklede versies de opdrachten uit te voeren voor een bepaalde configuratie of menu-optie. Nadat u dit hebt gedaan, voegt u de gewijzigde CONFIG.SYS en AUTOEXEC.BAT samen met de reservekopieën van deze bestanden, waardoor de oorspronkelijke opdrachten worden vervangen. Herhaal deze procedure voor elk scenario.



Als u een stuurprogramma of programma (zoals DOSKEY) in elke configuratie-optie hebt staan, plaats de programma-opdracht dan niet in de sectie [COMMON] van CONFIG.SYS. Plaats liever een kopie van de opdracht in elke configuratieset afzonderlijk. Denk er wel om dat de schakelopties DEVICEHIGH/LOADHIGH kunnen verschillen per configuratie. Alhoewel het programma gelijk is voor elke configuratie, kan de manier waarop het programma hoog wordt geladen, verschillen van de andere programma's die de UMB's delen.

Roep geen andere programma's vanuit batchbestanden op

Het is beter dat in het bestand AUTOEXEC.BAT geen batchbestanden worden opgeroepen die op hun beurt weer programma's starten. Laad bijvoorbeeld DOSKEY en SHARE niet vanuit een batchbestand dat zelf wordt opgeroepen vanuit AUTOEXEC.BAT. Door MEMMAKER worden alle programma's die vanuit het batchbestand worden geladen, als een eenheid geanalyseerd. Of alle programma's worden hoog geladen of geen enkel programma wordt hoog geladen. U zult eerst AUTOEXEC.BAT moeten wijzigen voordat u MEMMAKER start. Zorg er in dit voorbeeld voor dat DOSKEY en SHARE direct worden opgeroepen.

Roep geen programma's in IF-opdrachten op

Start geen programma vanuit een IF-opdracht in AUTOEXEC.BAT. MEMMAKER zal geen programma optimaliseren dat rechtstreeks vanuit een IF-opdracht wordt opgeroepen. In het volgende voorbeeld zal MEMMAKER het programma DOSKEY links laten liggen:

```
if "%anarkey%"==" " doskey
```

Met een GOTO-opdracht in een IF-opdracht zal MEMMAKER het geheugenverbruik van het programma optimaliseren. Dit voorbeeld kan als volgt worden aangepast:

```
if "%anarkey%"==" " goto loaddoskey
.....
:LOADDOSKEY
doskey
.....
```

Programma's niet laten analyseren en uitvoeren

Er kunnen programma's en stuurprogramma's zijn die u niet aan een analyse wilt laten onderwerpen door MEMMAKER. U wilt ze gewoon starten zonder de schakelopties en andere instellingen te wijzigen. Er kunnen ook programma's zijn die u niet wilt laten analyseren en uitvoeren, bijvoorbeeld een menu dat altijd aan het einde van AUTOEXEC.BAT wordt geopend. Het is niet nodig dat tijdens de analyses door MEMMAKER dit menu steeds wordt geopend.

Met het bestand MEMMAKER.INF in de DOS-directory kunt u aangeven welke

programma's en stuurprogramma's zullen worden geanalyseerd en uitgevoerd. Dit bestand is een ASCII-bestand en kan makkelijk in EDIT worden aangepast.

Een voorbeeld van dit bestand is:

```
;USED BY MEMMAKER
AUTOMENU
DBASE
DOSSHELL
EZMENU
LOTUS
MENU
NC
PARADOX
*PROTMAN
*SMARTDRV /DOUBLE_BUFFER
*SSWAP
*STRETCH
WORD
WP
WIN
123
```

Alle regels moeten helemaal links beginnen, zonder voorloopspaties. Regels met een puntkomma ervoor worden genegeerd. In deze lijst staat een aantal programma's en stuurprogramma's (de extensies

worden niet ingevoerd, dus geen EXE, COM, SYS, enzovoorts). Een programma waar een asterisk voor staat, wordt niet aangepast door MEMMAKER maar wel gestart. Alle andere programma's worden overgeslagen tijdens het optimaliseringsproces.

Zuinig met geheugen

De meeste gebruikers van DOS en Windows willen het basisgeheugen zo groot mogelijk maken. De eerste stap is om met MEMMAKER het systeem te analyseren en de belangrijke veranderingen door te voeren, zoals het starten van HIMEM.SYS en EMM386.EXE en het hoog laden van programma's. MEMMAKER pakt het in het groot aan, maar u kunt hier en daar nog wel een extra byte geheugen vinden.

Steeds meer programma's en stuurprogramma's worden zo ontworpen dat basisgeheugen wordt gespaard en er rechtstreeks gebruik wordt gemaakt van UMB's of expanded geheugen. De laatste versies van de diskcache PCKWIK worden automatisch in de UMB's geladen. Het stuurprogramma van de faxkaart, SATISFAX.SYS wordt grotendeels in het expanded geheugen

geladen. In dergelijke gevallen probeert MEMMAKER niet het programma hoog te laden, zelfs als er hoger geheugen vrij is, omdat waarschijnlijk het programma zonder hulp hoog wordt geladen. U kunt altijd met de opdracht MEM nagaan waar een programma wordt geladen en hoeveel geheugen er wordt gebruikt.

Het nauwkeurig afstellen van de geheugenconfiguratie van een systeem vraagt geduld, daar sommige veranderingen een systeem instabiel kunnen maken. In het geval dat een systeem vastslaat, zult u precies moeten nagaan wat u hebt gedaan en de "verbeteringen" ongedaan moeten maken. Op de volgende, eenvoudige wijze kunt u precies bijhouden wat u hebt gedaan:

- Wanneer u alleen de schakelopties van een opdracht verandert, wijzig ze dan niet in de originele regel. Zet voor de originele regel de opdracht REM en voeg een nieuwe regel toe waarop u deze wijzigingen aanbrengt. Bij de REM zou u ook nog de datum kunnen toevoegen, zodat u precies kunt nagaan wanneer u wat deed. Sommige problemen doen zich pas na enkele weken voor. Hier volgt

een voorbeeld van een gewijzigde opdracht EMM386:

```
REM 02/20/93 DEVICE=C:\DOS\EMM386.EXE RAM I=B000-B7FF NOEMS  
DEVICE=C:\DOS\EMM386.EXE RAM I=B000-B7FF NOEMS MIN=0
```

- Wanneer u enkele veranderingen aanbrengt, maak dan eerst een kopie van de configuratiebestanden.
- U zou de multiple configuration facility in DOS 6 (zie hoofdstuk 8) kunnen gebruiken. Maak een nieuwe menu-optie (bijvoorbeeld een optie *Experiment*) en sla de oude configuratie op in een apart configuratieblok.

In de volgende paragraaf worden een paar veranderingen bekeken die enkele bytes basisgeheugen extra opleveren. Breng deze wijzigingen pas aan als uw systeem zonder problemen draait na door MEMMAKER te zijn geoptimaliseerd. Bij veel wijzigingen tegelijk is het moeilijk vast te stellen waar het fout is gegaan.

Een kleinere omgevingsgrootte

Met de DOS-omgeving wordt het zoekpad, het uiterlijk van de DOS-prompt, de plaats van COMMAND.COM en alle andere gegevens over bepaalde programma's (bijvoorbeeld een tijdelijke directory voor Windows) opgeslagen. U kunt de instellingen van de huidige omgeving met de opdracht SET bekijken. Hieronder volgt een instelling van één van mijn systemen:

```
CONFIG=NORMAL
COMSPEC=C:\DOS\COMMAND.COM
PATH=C:\DOS;C:\;C:\WINDOWS;C:\WINDOWS\
NDW;C:\BATFILES
PROMPT=$P$G
SAGEEDIT=C:\SPE
NU=C:\U\NORTON
TEMP=C:\TEMP
```

Steeds meer programma's slaan in de omgeving informatie op. De standaardgrootte van deze omgeving is 256 bytes, maar dat is vaak te klein. Met de opdracht SHELL in CONFIG.SYS kan de omgevingsgrootte worden vergroot door het opgeven van de schakeloptie /E van COMMAND. Door de volgende opdracht wordt de omgeving vergroot tot 500 bytes:

```
SHELL=C:\DOS\COMMAND.COM C:\DOS\
/E:500 /P
```

De ruimte die door de omgeving wordt gebruikt, wordt gehaald uit het basisgeheugen. Voorkom verspilling door alleen de gevraagde ruimte aan de omgeving toe te wijzen.

Een makkelijke manier om te bepalen hoeveel geheugen er eigenlijk door de omgeving wordt gebruikt, is de uitvoer van de opdracht SET om te leiden naar een bestand. Een voorbeeld:

```
c:\> set > sizeof.set
```

U kunt vervolgens met DIR de grootte van het bestand aflezen.

Het bestand is enkele bytes groter dan de werkelijke grootte van de omgevingsgegevens, maar het is een vrij nauwkeurige methode om de grootte op te vragen.

Vergelijk de omgevingsgrootte met het geheugen dat is toegewezen door de opdracht SHELL. U kunt misschien enkele honderden bytes uitsparen door de hoeveelheid te verminderen die door

SHELL is toegewezen. De omgeving kan liggen tussen 160 en 32.768 bytes en wordt altijd afgerond naar het dichtstbijzijnde getal dat deelbaar is door 16.

Wanneer u het bericht `Out of Environment Space` ziet, hebt u de omgeving te klein gemaakt.

De opdracht FILES afstellen

De opdracht FILES in CONFIG.SYS bepaalt het maximum aantal bestanden dat tegelijkertijd in DOS kan zijn geopend (bijvoorbeeld FILES=25). Hoe meer bestanden er worden gespecificeerd, des te meer geheugen wordt er gebruikt. Elk extra bestand boven de standaardinstelling 8 kost een slordige 50 bytes. U moet daarom deze opdracht nauwkeurig omschrijven, omdat er anders geheugen wordt verspild. Dit kan behoorlijk oplopen; 20 bestanden teveel kost ongeveer 1000 bytes.

Er is geen duidelijke regel voor het bepalen van het aantal bestanden dat u nodig hebt. Het hangt af van de specifieke eisen van de software waarmee u werkt. Bij Windows is FILES=50 een normale instelling. Werkt u niet met Windows, maar wel

met een database, dan is FILES=30 voldoende. Bij alleen tekstverwerkers en spreadsheetprogramma's is FILES=20 waarschijnlijk genoeg.

Wanneer u een bericht *Out of file handles* ziet of iets wat daarop lijkt, zult u het aantal bestanden moeten verhogen.

De opdracht BUFFERS afstellen

Deze opdracht werd oorspronkelijk geïntroduceerd om bestandsverwerkingen sneller te laten verlopen. Zie het als een soort armeluis diskcache. U kunt tussen de 1 en 99 buffers toewijzen; elke buffer verlangt ongeveer 500 bytes. Wanneer u DOS hoog laadt, worden de BUFFERS ook hoog geladen.

Sinds de introductie is BUFFERS overvleugeld door echte diskcache-software. Als u nog met een 8086 werkt zonder diskcache, heeft het toewijzen van 20 buffers zin. Alhoewel deze opdracht een paar duizend bytes geheugen kost, wordt de schijf er sneller door. Werkt u echter met een 286 of hogere machine, maak dan gebruik van SMARTDRV van DOS of een diskcacheprogramma van een andere producent en verlaag de buffers tot 4. Bij een diskcache is het helemaal niet nodig buffers hoger

in te stellen. In feite kan een hogere instelling in conflict komen met de diskcache en de snelheid van uw systeem nadelig beïnvloeden.

De opdracht STACKS optimaliseren

Met de opdracht STACKS in CONFIG.SYS wordt er geheugen toegewezen om de hardware-interrupts te beheren. STACKS accepteert twee argumenten: het aantal stackkaders en de grootte van elk kader in bytes. Het volgende is hiervan een voorbeeld:

```
STACKS=9,256
```

Deze opdracht gebruikt meer dan 2000 bytes van het basisgeheugen. Bij niet alle systemen hoeft er geheugen worden toegewezen voor stacks en wordt er dus, indien dat wel het geval is, geheugen verspild. Als er geen opdracht STACKS wordt gespecificeerd, vult DOS standaard 9 stackkaders in van elk 128 bytes groot.

Probeer met STACKS=0,0 alle toewijzing van stackkaders te verwijderen. In de meeste gevallen zal uw systeem zonder probleem functioneren. Wanneer u foutberichten ziet, zoals Internal Stack Overflow, zult u de opdracht STACKS

weer moeten invoeren. Begin met een lage waarde, bijvoorbeeld 9,128. Als de problemen zich voor blijven doen, verhoog dan de instelling naar 9,256. Lost dit nog niets op, probeer dan 12,128, vervolgens 12,256 totdat het probleem is verholpen.

Een lagere toewijzing van FCBS

File Control Blocks (bestandsbesturingsblokken) zijn een overblijfsel uit de eerste versie van DOS. U hebt deze FCBS niet nodig, tenzij u nog met zeer oude programma's werkt. FILES is FCBS in een modern jasje.

Omdat DOS een verstekwaarde van 4 hanteert, is het zinvol om de opdracht fcbs=1 in CONFIG.SYS op te nemen.

Zuinig met LASTDRIVE

De opdracht LASTDRIVE wordt gebruikt in CONFIG.SYS voor extra logische stations, zoals netwerkstations en RAM-schijven. De opdracht LASTDRIVE=M bijvoorbeeld wijst voldoende geheugen toe aan elk station tot en met de letter M. Omdat elk station ongeveer 80 bytes verbruikt, moet u geen geheugen toewijzen aan stations die

nooit worden gebruikt. Zonder RAM-schijven of netwerkstations kunt u beter de opdracht LASTDRIVE vergeten.

Lager verbruik van MSCDEX-basisgeheugen

MSCDEX, het stuurprogramma voor CD-ROM's, gebruikt ongeveer tussen de 16K en 5K voor sectoren, afhankelijk van de configuratie. Wanneer uw systeem is geconfigureerd voor expanded geheugen, kunt u basisgeheugen (of hoger geheugen als MSCDEX hoog is geladen) uitsparen met de schakeloptie /E bij MSCDEX. Door de schakeloptie /E worden de buffers van CD-ROM door MSCDEX in het expanded geheugen geplaatst.

Bij de standaardinstellingen verbruikt MSCDEX 35K geheugen (hoog of basis), maar door de schakeloptie /E op te geven wordt dit geheugen teruggebracht naar 16K.

Zoveel mogelijk extended geheugen

Het voornaamste doel voor gebruikers van programma's onder DOS is het vergroten van het basisgeheugen. Voor gebruikers van Windows ligt

dit doel op een ander vlak, namelijk het *extended geheugen*. In Real Mode gebruikt Windows expanded geheugen, maar in Standard en Enhanced Modes put Windows uit het extended geheugen.

De beste manier om het beschikbare extended geheugen te vergroten is meer geheugen te installeren. U weet heel snel hoeveel extended geheugen er beschikbaar is, als u de opdracht MEM start. Doe dit niet vanuit Windows, omdat dan waarschijnlijk zal worden meegedeeld dat alle extended geheugen door Windows in gebruik is. Mocht u zien dat er geen extended geheugen beschikbaar is, plaats dan de opdrachten HIMEM.SYS en EMM386 in het bestand CONFIG.SYS.

De meeste besproken technieken voor het maximaliseren van het basisgeheugen hebben geen nadelige invloed op de snelheid van Windows. Feitelijk wordt het vrije geheugen verbeterd dat beschikbaar is voor programma's als Windows een DOS-programma start.

Sommige programma's reserveren extended geheugen voor zichzelf en verhinderen dat Windows dit gereserveerde geheugen kan benutten.

In het verleden vielen diskcaches en beheersprogramma's voor expanded geheugen in deze categorie, maar zowel SMARTDRV en EMM386 kunnen gezamenlijk met Windows het extended geheugen gebruiken. Als Windows bijvoorbeeld meer extended geheugen nodig heeft om een bewerking uit te voeren, wordt door SMARTDRV automatisch een gedeelte van 'zijn' extended geheugen aan Windows gegeven. Wanneer u met een programma werkt waarvoor te veel extended geheugen nodig is, zoals een diskcache, raadpleeg dan de handleiding van het desbetreffende programma en probeer het extended geheugen te verkleinen dat wordt gereserveerd.

Op een systeem waarop weinig extended geheugen is geïnstalleerd, ongeveer 2 of 4 megabytes, kunt u als volgt basisgeheugen omruilen voor meer extended geheugen:

- Als u met EMM386 werkt zonder de schakeloptie /NOEMS te hebben gespecificeerd, kan EMM386 extended geheugen converteren naar expanded geheugen. Standaard wordt er tijdens het laden door EMM386 automatisch wat extended geheugen geconverteerd naar expanded geheugen. Met de

schakeloptie min=0 springt EMM386 zuinig om met het extended geheugen en converteert het alleen naar expanded geheugen wanneer het voor een programma nodig is.

- Wanneer u de schakeloptie /NOEMS bij EMM386 gebruikt, kunt u extra extended geheugen besparen door de schakeloptie NOVCPI toe te voegen. Dit verwijdert de ondersteuning voor programma's die VCPI ondersteunen. De vuistregel is dat zonder EMS de ondersteuning voor VCPI niet nodig is.
- De HMA gebruikt 64K extended geheugen, en de UMB's worden gemaakt door extended geheugen in het hogere geheugen te plaatsen. Door de opdrachten EMM386 EN DOS=HIGH te deselecteren in CONFIG.SYS, maakt u dit geheugen vrij voor Windows. De hoeveelheid vrije geheugen wordt aanzienlijk minder, omdat DOS, stuurprogramma's en residente programma's in het basisgeheugen moeten blijven. Maar als u extended geheugen nodig hebt, is een ruil met het basisgeheugen het overwegen waard.

Expanded geheugen beheren

De manier waarop EMM386 expanded geheugen simuleert, is aanzienlijk verbeterd in DOS 6.



Wanneer uw computer hardware voor het expanded geheugen gebruikt, moet u met een stuurprogramma voor het expanded geheugen van een andere producent gaan werken, zoals EMM.SYS. DOS 6 heeft geen directe beheersprogramma's voor het expanded geheugen; DOS 6 kan alleen expanded geheugen met extended geheugen simuleren.

In vorige versies kon EMM386 met behulp van de schakeloptie /MEMORY een gedeelte van het extended geheugen reserveren om expanded geheugen te simuleren. Het zwakke punt in dit geheel was, dat het geheugen echt werd gereserveerd en niet voor het extended geheugen kon worden gebruikt.

EMM386.EXE van DOS 6 is een hele verbetering ten opzichte van zijn voorganger, omdat een 386 of hogere computer geheugen uit een gezamenlijke voorraad kan halen. Wanneer een programma expanded geheugen nodig heeft, wordt er geheugen

uit deze voorraad gehaald en aan het programma als expanded geheugen aangeboden. Zo kan ook extended geheugen uit deze voorraad worden gehaald als een programma daarom vraagt.

Expanded geheugen wordt rechtstreeks ondersteund, mits EMM386 is gespecificeerd zonder de schakeloptie /NOEMS.

De schakeloptie /MEMORY van EMM386 stelt de maximale hoeveelheid expanded geheugen in die wordt geleverd; de schakeloptie MIN= geeft de minimale hoeveelheid. Met vijf andere schakelopties wordt het pagina-frame van EMS beheerd: MX, FRAME=, /P, PN en H=. Zie voor deze schakelopties deel 4 *Alle DOS-opdrachten*.

Geheugenconflicten oplossen

"Mijn systeem hangt". Dit zou een mooie sticker voor op de auto kunnen zijn. Door te goochelen met configuraties en de volgorde waarin programma's worden geladen, zult u een keer in een situatie belanden dat uw systeem begint te protesteren. Een programma wil niet starten of, en dat is veel erger, het systeem is "dood". "Einde

oefening" klinkt ook leuk, maar mijn verwensingen liggen minder prettig in het gehoor.

In de volgende paragrafen worden enkele oplossingen aan de hand gedaan.

Door MEMMAKER te draaien hangt het systeem

Herstart eerst het systeem. Normaal gesproken ziet MEMMAKER dat er ergens een probleem is en herstelt de oude configuratiebestanden. Wanneer MEMMAKER voor de tweede keer hangt, herstart dan het systeem nog eens en druk op F5 zodra u het bericht `Starting MS-DOS . . .` ziet om schoon te herstarten. Ga naar de DOS-directory en voer de volgende opdracht in:

```
memmaker /undo
```

Door MEMMAKER worden de oude configuratiebestanden teruggehaald en het systeem herstart. Probeer MEMMAKER nog eens te draaien maar kies deze keer de optie Custom. Zet de optie Scan Upper Memory Aggressively? op No in en probeer het nog eens.

Door de nieuwe configuratie hangt het systeem

Wanneer het systeem in de DOS-prompt is geblokkeerd of steeds blijft hangen na enkele eenvoudige opdrachten, is er waarschijnlijk ergens in het geheugen een conflict. U zult wat speurwerk dienen te doen om de dader op te zoeken.

Wanneer het systeem alle dienst weigert nadat u de bestanden AUTOEXEC.BAT en CONFIG.SYS hebt aangepast, is het waarschijnlijk dat één van deze veranderingen de oorzaak is. Ga terug naar de oorspronkelijke configuratiebestanden en voer de veranderingen één voor één in. Herstart na elke verandering het systeem; u ziet vanzelf welke verandering het systeem laat hangen.

Als u niet meer weet welke veranderingen u de laatste tijd hebt aangebracht, moet u methodisch te werk gaan om het probleem te isoleren. Een tactiek is om de bestanden CONFIG.SYS en AUTOEXEC.BAT uit te kleden en één voor één de opdrachten in te voeren, totdat het probleem zich voordoet. Houdt u van een gokje, dan kunt u ook in de omgekeerde richting werken en opdrachten verwijderen. Bij het opstarten kunt u met F8 interactief de opdrachten selecteren die moeten

worden uitgevoerd. Vaak worden problemen veroorzaakt door een conflict in de UMB's. In plaats van stuurprogramma's te verwijderen kunt u ook DEVICEHIGH veranderen in DEVICE.

Wanneer u het programma niet kunt vinden dat het probleem veroorzaakt, kan EMM386 de boosdoener zijn. Door de schakelopties van EMM386 aan te passen moet u het probleem kunnen lokaliseren.

Begin de schakeloptie HIGHSCAN te verwijderen, als die erin staat. Door HIGHSCAN stelt EMM386 op een agressieve manier elke vrije ruimte in UMB vast. Dit kan op sommige systemen voor problemen zorgen, vooral de systemen met SCSI-stuurprogramma's, zoals ASP14DOS.

Veel ISA-bussystemen (industry standard architecture) die zijn uitgerust met SCSI-stuurprogramma's kunnen niet meer dan 16MB geheugen behandelen. Deze problemen ontstaan omdat de DMA (direct memory access), die wordt gebruikt door SCSI-stuurprogramma's, op ISA-systemen alleen 16MB RAM kan benaderen. De algemene oplossing is de CMOS te veranderen zodat alleen de eerste 16MB van het

geïnstalleerde geheugen wordt herkend, en de rest van dit geïnstalleerde geheugen wordt genegeerd. Zie ook hoofdstuk 7 voor een uitgebreide bespreking van ISA en DMA.

Wanneer HIGHSCAN het probleem niet is, probeer dan de schakeloptie NOMOVEXBDA, waardoor EMM386 het extended geheugengebied van de BIOS niet meer verplaatst. Verwijder de schakeloptie te zamen met eventuele inclusief-opdrachten (I=) als dit niet werkt. Mocht het wel werken, isoleer dan het probleemgebied door het aanpassen van de I= opdrachten totdat het probleem zich weer voordoet.

Het systeem kan ook hangen doordat EMM386 extended geheugen overbrengt naar een actief gebied van het hogere geheugen. Dit probleem kunt u op een snelle manier isoleren door al het hogere geheugen met de opdracht X=A000-FFFF uit te sluiten. Wanneer het probleem hiermee wordt verholpen, dient u een kleiner gebied uit te sluiten, bijvoorbeeld X=A000-DFFF. Maak het geheugengebied kleiner totdat u een bepaald geheugengebied hebt geïsoleerd. Met de schakeloptie X die daarvan het resultaat is, kunt u

EMM386 verhinderen dat gebied van het hogere geheugen te gebruiken.

Een systeem hangt door een programma

Sommige programma's houden er niet van in het hogere geheugen te worden geladen. Wanneer een programma een systeem laat hangen terwijl ermee wordt gewerkt, probeer het dan in het basisgeheugen te laden door de opdracht LOADHIGH (of LH) te verwijderen.

Maar als het programma niet hoog wordt geladen, kunt u proberen het probleem te isoleren zoals hiervoor is beschreven. Begin met een eenvoudige configuratie die voldoende is voor het programma, en voeg steeds meer opdrachten toe totdat het programma het systeem laat hangen.

Natuurlijk is het ook belangrijk dat u de handleiding van het programma raadpleegt, waarin misschien melding wordt gemaakt van conflicten die bekend zijn.

Een programma werkt veel trager

Dit kan zich voordoen als de schakeloptie NOEMS is toegevoegd aan EMM386. In een dergelijk geval versnelt het programma waarschijnlijk de bewerkingen met behulp van het expanded geheugen. Een systeem kan ook trager worden als er twee diskcaches zijn geladen.

Nogmaals, als u het probleem niet kunt vinden, begin dan weer met een uitgekleepte configuratie en voeg één voor één de opdrachten toe.

Het systeem reageert niet op Ctrl-Alt-Del

Voeg de schakeloptie ALTBOOT toe aan de opdracht EMM386 in het bestand CONFIG.SYS, wanneer het systeem niet herstart met Ctrl-Alt-Del. Voeg deze schakeloptie ook toe als het systeem vreemd reageert op de toetsen.

Samnevatting

Dit hoofdstuk is helemaal gewijd aan het computergeheugen, één van de hete hangijzers van de DOS-gebruiker. De volgende punten zijn hierbij aan de orde gekomen:

- > De normale geheugenchip in de computer is de DRAM-chip. Chips met een hoge capaciteit worden op compacte printplaten gesoldeerd en staan bekend als SIMM's. Krachtige SRAM-chips worden gebruikt voor geheugencaches, maar hun stroomverbruik is te hoog voor de algemene RAM.
- > Elke lokatie in het geheugen heeft een adres segment:offset. Op systemen met een 286-microprocessor of hoger wordt met het segment in een opzoektabel het fysieke adres bepaald. Door deze techniek wordt het maximale adresseerbare geheugen vergroot naar 16MB op een 286 en tot 4 gigabytes op 386 en hogere machines. (Op systemen met een 8086 is het grootste adresseerbare geheugen 1MB).
- > De eerste megabyte van het geheugen wordt verdeeld in basisgeheugen en het hogere geheugen. Extra geheugen is of expanded of extended geheugen.
- > Op 286 en hogere systemen kunt u met HIMEM.SYS de eerste 64K van het extended geheugen als *hoog* geheugen

laten fungeren. Het grootste deel van DOS kan in het hoge geheugen worden geladen.

- > Op 386 en hogere systemen kunnen lege gebieden in het hogere geheugen worden opgevuld met extended geheugen, waardoor UMB's worden gemaakt.
- > MEMMAKER analyseert een systeem en past de bestanden CONFIG.SYS en AUTOEXEC.BAT aan voor een optimaal gebruik van het systeemgeheugen.
- > U lost geheugenproblemen systematisch op door de configuratiebestanden te herzien totdat het probleem wordt gevonden.

In hoofdstuk 6 wordt besproken hoe u met de stuurprogramma's van DOS de weergave van het scherm verbetert en het toetsenbord aanpast.
