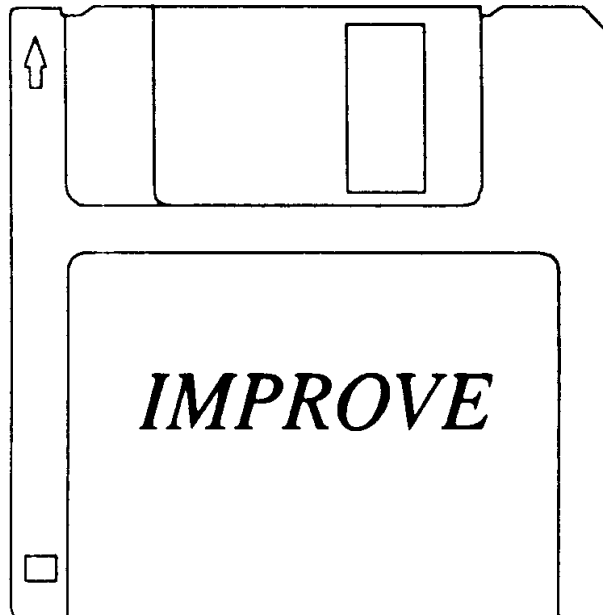


MST

Improve

MST

Copyright 1992 by
MSX Software Team



*Original and scanned by Bifi
Converted to PDF by HansO, 2002*



IMPROVE

IMPROVE version 1.0

(C) MSX Software Team 1992
Written by Ramon van der Ninkel

The ultimate floppy disk and hard disk speed improvement utility for MSX

— PRESS ANY KEY —

Inhoudsopgave

Een woord van de programmeur	5
Snel versus traag	6
Directories sorteren	8
Clusters verplaatsen	9
Sneller, maar minder veilig	10
Opstarten	11
Opties	13
Automatisch beginnen	14
In vergelijking	15
Deel 2	
Over FAT's en clusters	16
Clusters en Sectoren	17
De File Allocation Table	18
De Media Identificatie	19
De Directory	20
Rekenen aan clusters	22



IMPROVE

IMPROVE version 1.0

(C) MSX Software Team 1992
Written by Ramon van der Winkel

The ultimate floppy disk and hard disk speed improvement utility for MSX

— PRESS ANY KEY —

Een woord van de programmeur

Ik ben trots op dit nieuwe MST produkt, waar ik al sinds november 1991 mee bezig ben. Op dat moment kocht ik mijn harddisk om samen met mijn Turbo-R nog snellere en grotere programma's te ontwikkelen. Ik wist van de disk optimizers van de MS-DOS computers en wilde graag eenzelfde programma voor mijn MSX hebben, dus wat doe je dan als ras-echte MSX programmeur? Precies, je schrijft het zelf! De eerste werkende versie was er redelijk snel, maar zolang er bugs in zitten kan je het niet maken je produkt de markt op te brengen. Daarom zijn er eerst een aantal slachtoffers gezocht die hun geliefde data beschikbaar wilden stellen als testbank voor Improve. In de beginsituatie is er behoorlijk wat verloren gegaan moet ik toegeven en heb ik regelmatig boze testers aan de lijn gehad. Maar het had nut, want het produkt dat u heeft aangeschaft is gegarandeerd operationeel bug vrij!

Als programmeur zie ik graag dat mijn produkten wijd verspreid gebruikt worden, maar dan wel graag legaal. Er is veel tijd in gestoken en daarom verzoek ik u dan ook niet mee te werken aan het illegaal verspreiden van Improve. Dat houdt mijn motivatie hoog en zorgt ervoor dat u binnenkort nog meer MST produkten kunt verwachten. Als u het niet voor mij wilt doen, doe het dan voor de MSX in het algemeen!

Als laatste zou ik graag een aantal mensen willen bedanken voor hun bijdrage aan de ontwikkeling van Improve, namelijk:

Amoud-Jan van Tuin, voor het ontdekken van onmogelijke bugs.

Robbert Wethmar, voor zijn commentaar op de handleiding en hulp bij de lay-out ervan.

Ries Vriend, voor de discussies over de algoritmen.

Bob, zodat ik de buitenlucht af en toe te zien krijg.

Veel plezier met Improve,

Ramon van der Winkel

April 1992

Snel versus traag

Op een diskette worden regelmatig bestanden geschreven, verwijderd en soms zelfs verplaatst. Hierbij komen de nieuwe bestanden op of over de oude bestanden en vullen alle lege stukken op de diskette. Als er meerdere vrije stukken zijn, dan komt het bestand over al die stukken heen te staan.

Nadeel is dat bij het inlezen van het bestand de kop van de diskdrive steeds naar een andere positie op de disk verplaatst moet worden om ook daar weer een beetje data van het bestand in te lezen. Dit is niet bevorderlijk voor de tijd die nodig is om het bestand in te lezen.

Ideaal zou het zijn als ieder bestand in opeenvolgende stukken op de diskette staat; het kan dan ingeladen worden zonder dat de kop van de drive het heen-en-weer krijgt en de laadtijd zou er zo kort mogelijk door zijn.

Onder MSX-DOS2 is het mogelijk sub-directories aan te maken en er een aantal programma's in te verzamelen. In een sub-directory met bijvoorbeeld de naam UTILS zouden alle utilities gezet kunnen worden. Door een zoek-pad op die betreffende directory te zetten kunnen de programma's die zich daar bevinden altijd opgestart worden, ongeacht de huidige directory waarin het commando ingetypt is.

Als het zoek-pad meerdere directories bevat, dan wordt bij ieder programma dat opgestart wordt en niet in de huidige directory staat, het hele zoek-pad afgewerkt. Iedere directory moet dan opnieuw ingeladen worden en doorzocht worden, op zoek naar het gewenste programma. Als de directories zich verspreid over de diskette bevinden, dan kan dat nogal wat drive-kop bewegingen vereisen en zal dat de totale zoektijd zeker niet positief beïnvloeden.

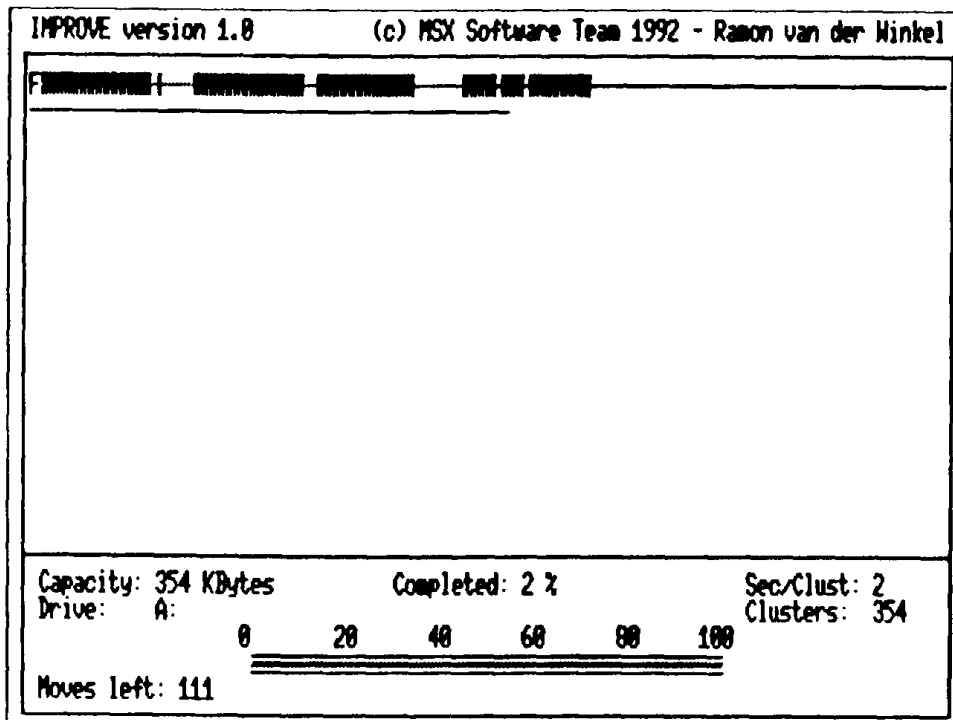
Als een sub-directory registratie niet vooraan in de directory staat, dan worden alle voorgaande registraties ook iedere keer weer vergeleken als het operating systeem op zoek is naar de sub-directory. Ook dit gaat ten koste van de zoektijd, relatief gezien niet zoveel, maar toch.

Ideaal zou het zijn als alle directories dicht bij elkaar staan, aan het begin van de diskette bijvoorbeeld, want daar staat de root-directory en die moet daar staan. Alle sub-directory registraties zouden bovenaan in het directory-overzicht moeten staan, want dan zijn ze snel te vinden en wordt er niets onnodig vergeleken.

Om de hierboven beschreven ideale situatie te creëren is Improve ontwikkeld. Het programma kan een bijna lege, maar ook een goed gevulde

diskette reorganiseren en werkt onder MSX-DOS1 en 2 en natuurlijk ook op harddisks.

Het volgende stuk gaat in op de techniek die Improve gebruikt. Vooral de term 'cluster' moet een bekend begrip zijn om alles te begrijpen. Het hoofdstuk 'Over FAT's en clusters' legt precies uit wat ze zijn en waar ze voor nodig zijn, lees eventueel eerst dat deel!



Improve aan het werk op een enkelzijdige floppy



Directories sorteren

Als Improve gestart wordt, dan worden eerste de root directory en alle onderliggende sub-directories gesorteerd volgens de opgegeven instellingen. De directory registraties worden vooraan gezet, de bestanden eventueel op naam gesorteerd. Als er gedelete of lege registraties in de sub-directory staan, dan worden deze verwijderd.

Een sub-directory is niets anders dan een bestand, maar dan met het directory-attribuu. Een sub-directory bestaat dan ook gewoon uit een aantal clusters die in de FAT met elkaar verbonden zijn. Het is daarom ook mogelijk veel meer bestanden in een sub-directory te zetten dan er in de root passen. Als er geen vrije registratieplaats meer is, dan wordt er gewoon weer een cluster aangeplakt en is er weer ruimte.

Als Improve een aantal lege en gedelete entries weghaalt, dan is het mogelijk dat zo'n cluster weer vrij komt. Improve kan dan het cluster weer vrijgeven, zodat er weer data in gezet kan worden. Als er bijvoorbeeld 500 bestanden in een directory gezet worden en de eerste 490 worden weggehaald, dan moet MSXDOS voor de laatste 10 bestanden alle 490 gedelete entries doorlopen voordat ze gevonden worden. Improve zet die 10 bestanden vooraan in de directory en verwijdert alle gedelete registraties en geeft eventueel een aantal clusters vrij.

Als alle directories gesorteerd zijn is de eerste stap voltooid, nu moeten alleen de clusters van alle bestanden nog achter elkaar geplaatst worden.

Clusters verplaatsen

Tijdens het sorteren van de directories onthoudt Improve bij welke clusters van welke bestanden zijn. Na het sorteren wordt bepaald welke clusters waar moeten komen te staan, zodat alle clusters van alle bestanden achter elkaar komen te staan en de directory clusters vooraan op de disk terecht komen.

Improve weet nu precies hoeveel clusters er nog verplaatst moeten worden en het verplaatsen kan beginnen. Bij het verplaatsen van de clusters wordt maximaal één cluster tegelijk verplaatst en op zijn nieuwe positie weggeschreven. Het algoritme dat hiervoor gebruikt wordt zoekt steeds naar het eerste vrije cluster op de diskette, bekijkt welk ander cluster daar moet komen te staan, leest die in en schrijft hem weer weg op de nieuwe positie. De oude positie van het cluster wordt weer vrijgegeven en er is altijd weer een nieuw vrij cluster.

Voorwaarde om te kunnen beginnen is dus minimaal één vrij cluster, als er geen vrij cluster meer is geeft Improve een foutmelding en kan het verplaatsen niet beginnen.

Nadat een cluster verplaatst is zijn er twee mogelijkheden: Het cluster zit in een cluster chain, dan hoeft alleen de voorganger opgezocht te worden en de verwijzing aangepast. In het andere geval is het cluster de eerste van de chain en dan moet de directory registraties aangepast worden. In sommige gevallen moeten meerdere registratie aangepast worden, namelijk als een cluster verplaatst wordt van een directory met daaronder meerdere sub-directories. Alle ".." registraties in die onderliggende sub-directories moeten dan aangepast worden.

Het algoritme van het zoeken naar vrije clusters houdt natuurlijk op als er geen cluster meer op een van de vrije clusters gezet hoeft te worden. Er kunnen dan echter nog wel clusters zijn die nog niet verplaatst zijn, namelijk de onderling te verwisselen clusters.

Omdat Improve veilig wil werken wordt het eerste cluster dat nog verplaatst moet tijdelijk verplaatst maar een vrij cluster dat altijd verderop op de diskette ligt. Daarna doet het standaard algoritme weer zijn werk en verplaatst weer een hoeveelheid clusters. Aan het einde wordt het tijdelijk verplaatste cluster ook op zijn plaats gezet.

Nadat alle clusters op de juiste plaats zijn gezet is Improve klaar en wordt dit gemeld. Tussendoor afbreken geeft ook geen problemen, de volgende keer staan de al verplaatste clusters natuurlijk nog steeds op de nieuwe posities en gaat Improve verder waar hij gebleven was.

Sneller, maar minder veilig

Iedere verplaatsing van een cluster wordt in de FAT bijgehouden en meteen naar disk geschreven. Dit voorkomt dat er data verminkt wordt of verloren gaat, als bijvoorbeeld de stroom uitvalt. Om de FAT weg te kunnen schrijven moet de kop echter naar het begin van de diskette en dit moet na iedere verplaatsing van een cluster, dit kost tijd en dat kan sneller.

Hiervoor is de optie 'Decrease FAT writes'. Als deze aan wordt gezet, dan schrijft Improve de FAT pas weg als alle clusters verplaatst zijn, of als er tussendoor afgebroken wordt. Hierdoor hoeft de kop niet bij iedere verplaatsing terug naar het begin van de diskette en dat spaart veel tijd, maar zoals al gezegd: het is minder veilig.

FAT controle

Om de veiligheid te verhogen controleert Improve, voordat er begonnen wordt met het verplaatsen van de clusters, eerst of de FAT wel in orde is. Mochten er 'cross-linked clusters' (twee verwijzingen naar één cluster) of 'lost clusters' (zijn wel bezet maar wordt niet naar verwezen) aanwezig zijn, dan wordt daar melding van gemaakt en blijven alle clusters op hun plaats.

Mocht Improve een fout opgemerkt hebben in de structuur van de disk, dan kan met de DOS-utility CHKDSK de fout opgelost worden. Dit precieze werkje is niet in Improve opgenomen omdat deze fout in de meeste gevallen niet zal optreden en het veel programma code in beslag neemt. Deze ruimte is hard nodig voor andere zaken. Improve detecteert slechts.

Opstarten

Nadat Improve is ingeladen verschijnt er een introductie scherm met een versie vermelding. Na een druk op een willekeurige toets komt het werkscherm van Improve in beeld met in het midden het hoofdmenu.

Het belangrijkste deel van het scherm van Improve is het grote gat in het midden, waar nu tijdelijk het hoofdmenu huist. Daarin kan de verplaatsing van de clusters visueel worden gevolgd: voor ieder cluster op de diskette is er een stukje gereserveerd. Als de cluster data bevat, dan staat er een streepje, als de cluster leeg is een puntje. De felheid van het streepje of het puntje geeft aan of de cluster nog verplaatst moet worden of niet.

Als een streepje fel oplicht bevat de cluster data en hoeft deze niet meer verplaatst te worden, een donker streepje moet nog wel verplaatst worden. Een donker puntje is nu nog leeg, maar er komt nog data in te staan en een fel puntje is en blijft leeg.

Als alle clusters op de juiste posities gezet zijn, dan is het begin van het scherm dus gevuld met felle streepjes en het einde met felle puntjes. Het komt echter af en toe voor dat diskettes bad sectors hebben, deze komen ook op het scherm te staan als felle streepjes en zullen niet verplaatst worden.

De FAT is van het 12-bits type en er gaan dan ook maar 4086 cluster registraties in. De streepjes zijn zo opgebouwd dat voor iedere cluster registraties één streepje of puntje op het scherm past. Als een diskette minder clusters bevat, dan wordt alleen het eerste deel van het scherm gebruikt. Daar waar de puntjes ophouden zijn geen cluster registraties meer.

Onder aan het scherm bevindt zich nog een informatieblok. In het midden pronkt een balk om aan te geven hoeveel procent van de clusters al op de juiste positie staat. Boven de balk staat het percentage als getal aangegeven. In de linker onderhoek staat het aantal clusters dat nog verplaatst moet worden.

Links naast de balk staat de huidige diskdrive die aangestuurd wordt en de opslagcapaciteit van de diskette die zich daarin bevindt. Bij een harddisk is het de opslagcapaciteit van de partitie.

Aan de linker kant staat informatie over de verdeling van opslag capaciteit in clusters. Het aantal sectoren per cluster en de het totale aantal clusters zijn hiervoor de enige belangrijke gegevens.



Vanuit het hoofdmenu kunnen de volgende mogelijkheden gestart worden:

- (S) Start improve
- (D) Drive select
- (O) Options menu
- (Q) Quit

Main Menu
Start improve
Drive select
Options
Quit to dos

De eerste optie 'Start improve' start het functionele deel van Improve. Als eerste worden de gegevens van de drive bepaald, de directories gesorteerd en bepaald welke clusters waar moeten komen te staan. Als dit klaar is kan het verplaatsen van de clusters beginnen, de werking is hiervoor al beschreven. Op ieder moment kan op ESCape gedrukt worden om het proces af te breken. Improve vraagt dan om een bevestiging, schrijft eventueel eerst nog even de FAT naar disk en keert daarna terug naar het hoofdmenu.

Bij het opstarten bepaald Improve de huidige drive en neemt deze als drive die bewerkt moet worden. Met de optie 'Drive select' is het echter mogelijk een andere drive te kiezen, Improve geeft hiervoor een menuutje waarin de voor het systeem bekende drives te zien zijn. Met de letters A tot en met H kan een andere drive gekozen worden, deze wordt in de linker onderhoek in het informatieblok getoond. Eventueel kan met ESCape teruggekeerd worden naar het hoofdmenu zonder dat er een nieuwe drive gekozen is. Het is ook mogelijk om een andere drive op te geven bij het opstarten van Improve. Op de volgende manier wordt Improve opgestart van de A-drive, maar neemt als werkdrive dan de C-drive.

A>IMPROVE C:

In het 'Options menu' kunnen een aantal instellingen gewijzigd worden, deze worden hieronder behandeld.

Met de laatste optie, Quit, kan Improve verlaten worden.

Opties

De bewerkingen die het programma uitvoert kunnen een beetje worden aangepast. Hiervoor zijn vijftal instellingen, die met een optie bij het opstarten veranderd kunnen worden. Binnen het programma kunnen de instellingen ook veranderd worden vanuit het 'Options menu'. Om bijvoorbeeld het sorteren van bestanden (S) aan te zetten kan Improve op de volgende manier worden opgestart:

```
A>IMPROVE /S
```

Vanuit het Options menu zijn de volgende vijf instellingen aan te passen:

- (S) Sort files
- (D) Directories first
- (T) Truncate directories
- (C) Sequence Clusters
- (F) Decrease FAT writes

Option switches	
Sort files:	No
Directories first:	Yes
Truncate directories:	Yes
Sequence Clusters:	Yes
Decrease FAT writes:	No

De optie 'Sort files' geeft aan of de bestanden in een directory gesorteerd moeten worden. Standaard staat deze optie op 'No', maar als dit gewijzigd is in 'Yes' dan worden alle bestanden in oplopende alfabetische volgorde gesorteerd.

Om het bereiken van zoekpaden sneller te maken kunnen de sub-directory registraties vooraan in de directory gezet worden. Standaard gebeurt dit al, maar door de optie 'Directories first' op 'No' te zetten blijven de sub-directories op hun oorspronkelijke plaats staan.

Als er één of meerdere gewiste-file registraties in een directory staan dan worden ze door Improve verwijderd. Als er daardoor een cluster vrij komt, dan kan Improve deze weer vrijgeven zodat ze door andere programma's gebruikt kunnen worden. Standaard staat deze optie aan, maar door 'Truncate directories' op 'No' te zetten worden deze clusters niet vrijgegeven, maar blijven ze onderdeel van de sub-directory.

Improve kan ook gebruikt worden om alleen de directories te sorteren. Dit kan door 'Sequence Clusters' op 'No' te zetten. Na het sorteren van de directory stop Improve; alle clusters blijven op hun huidige positie staan, ook de directory clusters.

Bij het verplaatsen van een cluster worden iedere keer de FAT opnieuw weggeschreven. Deze behoorlijke tijd consumptie kan uitgezet worden door 'Decrease FAT writes' op 'Yes' te zetten. De FAT wordt dan niet meer naar disk geschreven totdat alle clusters op hun nieuwe positie staan.

Automatisch beginnen

Om het ook mogelijk te maken Improve vanuit bijvoorbeeld een batchfile op te laten starten en automatisch alle partities van een harddisk te Improven, is het mogelijk Improve automatisch te laten beginnen.

Hiervoor is de commandline switch /A beschikbaar. Als deze meegegeven wordt, dan wordt het intro scherm overgeslagen en vanuit het hoofdmenu automatisch de optie 'Start improve' gekozen. Als de hele disk behandeld is, dan wordt Improve automatisch weer verlaten. Met de volgende instructies in een batchfile neemt Improve de A: en C: drive onder handen, waarbij de files van de A-drive gesorteerd worden en de Decrease FAT writes optie bij de C: drive aan staat:

```
IMPROVE A: /S /A  
IMPROVE C: /F /A
```

De commandline switches komen overeen met de keuzes die in het 'Options menu' gemaakt kunnen worden.



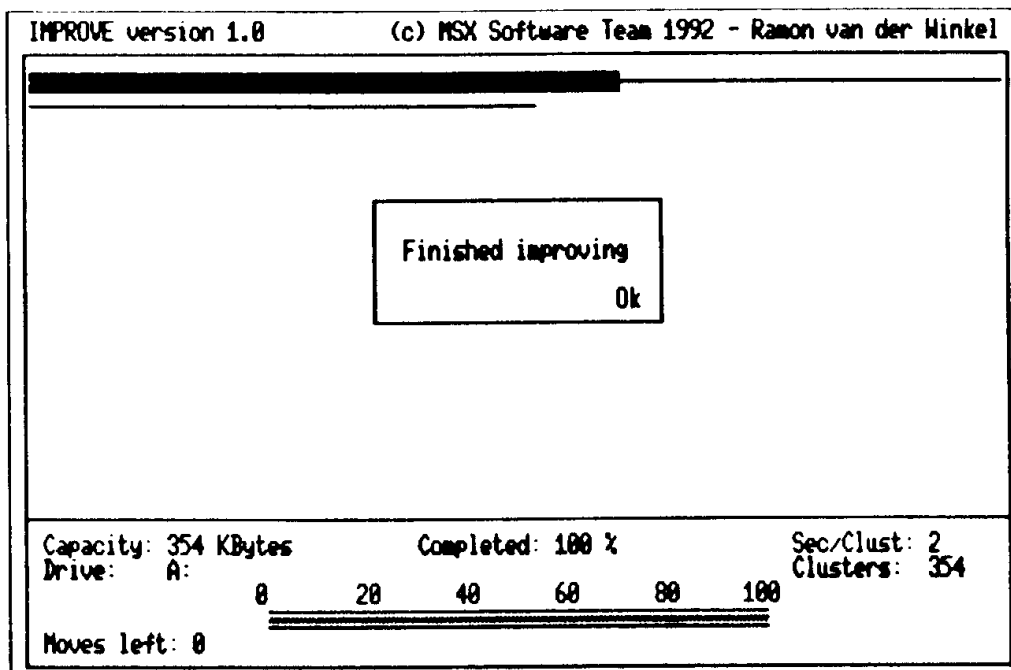
In vergelijk

De kenners van de MS-DOS versies van Improve, bijvoorbeeld Speed Disk van Norton, kunnen met de volgende uitleg de ware kracht van Improve naar boven halen.

De algoritmen van al die PC versies zijn gebaseerd op het inlezen van meerdere clusters op de te verbeteren positie van de disk en deze in de volgende lege clusters schrijven. Daarna wordt het gemaakte gat weer gevuld en weer een blok bewerkt etc. Tussen het verwerken van een blok wordt iedere keer de FAT geupdate. Het updaten van de FAT gebeurt daar dus minder vaak, maar kan bij Improve helemaal uitgezet worden.

Door het verplaatsen in blokken worden clusters regelmatig te vaak verplaatst, iets waar Improve geen last van heeft want die weet precies welk cluster waar moet komen te staan en kan ze in één keer op de juiste plaats zetten.

De schermuitvoer van de PC versies bestaat altijd uit 1 karakter voor meerdere clusters. Bij Improve wordt ieder cluster afzonderlijk op het scherm geplaatst, waardoor beter te zien is welke clusters nog verplaatst moeten worden.



Over FAT's en clusters

Heel veel gegevens op een diskje proppen is geen probleem, maar om later ook nog te weten waar wat stond moet er een administratie bijgehouden worden. Probleem is natuurlijk dat die administratie zo weinig mogelijk van de opslagcapaciteit in beslag moet nemen en dus vrij efficiënt moet zijn.

Als eerste is de diskette verdeeld in logische blokjes van 512 bytes, de zogenaamde sectoren. Iedere sector kan gevuld worden met willekeurige data. Een groot deel de sectoren wordt dan ook gebruikt om de data van de bestanden in op te slaan. Een aantal sectoren echter is gereserveerd voor de administratie.

Al deze administratie sectoren bevinden zich vooraan op de diskette. Hierdoor zijn ze snel te bereiken en kan de capaciteit van de diskette zich strekken voor zo ver als de fysieke opslag dit toe staat. Op een 3,5 inch dubbelzijdige floppy disk kan 720 kilobyte en op een harddisk partitie zelfs 32 Megabyte aan data. De administratie staat dus altijd aan het begin.

De eerste sector is wel een hele speciale. Hierin staat alle belangrijke informatie over de diskette, samen met het opstartprogramma dat nodig is om MSX-DOS1 of 2 in te laden, bijvoorbeeld als de computer gereset wordt met een diskette in de drive.

Na deze boot-sector volgt de administratie van alle sectoren op de diskette: de File Allocation Table of kortweg FAT. Afhankelijk van de capaciteit van de diskette staat hier per (groepje) sector(en) of ze in gebruik zijn. Dit is een behoorlijk geavanceerde methode waar verderop meer over uitgelegd wordt.

Het laatste stuk van de administratie volgt achter deze FAT en is de root directory. Hierin staan alle namen van de bestanden, datum en tijd waarop ze aangemaakt zijn, grootte, attributen en een verwijzing naar het eerste blokje in de FAT waarin het bestand begint. Onder MSX-DOS1 is dit enige directory die gebruikt kan worden, MSX-DOS2 echter kan extra directories aanmaken, de sub-directories.

De administratie valt dus nog wel mee. Er is die speciale boot-sector waarin de capaciteit gegevens van de diskette staan, daarna de FAT voor de administratie van alle sectoren en als laatste de directory om de namen in op te slaan.



Clusters en Sectoren

Omdat het te veel administratieruimte – en dus diskruimte – in beslag neemt is het onmogelijk om voor iedere sector bij te houden of deze gebruikt wordt of niet. Om de administratie klein te houden is de diskette verdeelt in groepjes sectoren. Er kunnen maximaal 4086 groepjes gemaakt worden, de zogenaamde clusters, die allemaal een eigen registratie in de FAT hebben. Niet iedere diskette heeft een FAT waarin 4086 registraties passen, vaak is deze kleiner en wordt alleen op een harddisk een FAT gebruikt met zoveel cluster registraties.

Voor iedere bestand wordt in de directory opgeslagen in welke cluster bestand begint. Het cluster staat voor een groepje sectoren en door om te rekenen welke sectoren dat zijn kan de data ingelezen worden. Een bestand bestaat echter meestal meer dan één sector en meestal ook wel meer dan één cluster. Daarom staat in die cluster registratie in de FAT of het cluster een opvolger heeft of niet. In dat volgende cluster gaat het bestand dan verder en verder en verder, tot het einde.

Iedere cluster registratie moet dus een getal kunnen bevatten om aan te geven wat het volgende cluster is. In 12 bits kan een getal tussen de 0 en 4091 opgeslagen worden. Er zijn 4086 clusters maximaal, dus overige niet gebruikte getallen kunnen dan ingezet worden om aan te geven dat er geen opvolgend cluster meer is, of dat het cluster helemaal niet in gebruikt is, want ook dat moet bijgehouden worden.

Als er 12 bits nodig zijn om één cluster registratie te maken, dan zijn er voor 4086 cluster registraties dus 49032 bits nodig, dat zijn bijna 12 sectoren. De grootst mogelijke FAT bestaat dus 12 sectoren.

Om te bepalen hoeveel sectoren er in n cluster kunnen, moeten we weten hoeveel sectoren er op de diskette passen. Op een 720 kb diskette kunnen 1440 sectoren van 512 bytes. Bij een FAT van maximaal 3 sectoren kunnen er maximaal 1024 clusters aangemaakt worden. Omdat het aantal sectoren per cluster altijd een 2-macht moet zijn (1, 2, 4, 8, 16, 32 etc.) kan er met clusters van 2 sectoren gewerkt worden. Dit wordt in de boot-sector opgeslagen.

Als er een bestandje van 1 byte aangemaakt wordt, dan wordt er dus altijd 1 kb aan opslag ruimte gebruikt. Een beetje zonde, maar het kan nu eenmaal niet anders. Voor een harddisk met een partitie van 32 Mb moeten 65536 sectoren geregistreerd worden, hiervoor zijn clusters van 16 sectoren per cluster nodig, als je dan een bestandje aan maakt van 1 byte wordt er in een keer 8 kb diskruimte gebruikt, waarvan het grootste deel leeg blijft. Het aantal sectoren per cluster moet dus goed afgewogen worden.

De File Allocation Table

In de FAT staan dus de cluster registraties. Als de registratie 0 is, dan is het cluster niet in gebruik, vrij dus. Als MSX-DOS bepaalt hoeveel bytes er nog vrij zijn op disk, dan worden het aantal vrije cluster registraties geteld en vermenigvuldigd met het aantal sectoren per clusters en bytes per sector. Bij 450 vrije clusters op een disk met 2 sectoren per cluster zijn er dus $450 * 2 * 512 = 460800$ bytes vrij.

Registraties tussen de 1 en 4086 zijn cluster verwijzingen naar het opvolgende cluster. Meerder clusters die zo aan elkaar 'geregen' zijn heten een chain, een ketting dus. Om het einde van een cluster aan te geven wordt een waarde vanaf 4088 in de registratie gezet, meestal 4095. Het enige nog overgebleven getal is 4087. Deze wordt in de registratie gezet als een of meerdere sectoren in dat cluster beschadigd zijn en dus niet meer bruikbaar zijn. Deze registratie heet een bad-sector registratie.

Een overzicht van de mogelijke waarden van een cluster registratie:

0	Cluster is vrij
1..4086	Volgende cluster in de chain
4087	Bad sector
4088..4095	Einde van een chain

De FAT is dus een verzameling van allemaal getallen die allemaal iets betekenen. Omdat een getal maximaal 12 bits beslaat en het zonde zou zijn dit in twee bytes (16 bits) op te slaan en iedere keer 4 bits ongebruikt te laten, worden er twee registraties op een speciale manier in 3 bytes gezet. De registratie van de even nummers staan in de 12 minst significantie bits van de drie bytes. De oneven registraties staan in de 12 meest significante bits. Bij de volgende reeks cluster registraties, beginnend met bijvoorbeeld het even nummer 200, horen de volgende reeks bytes (alles hexadecimaal):

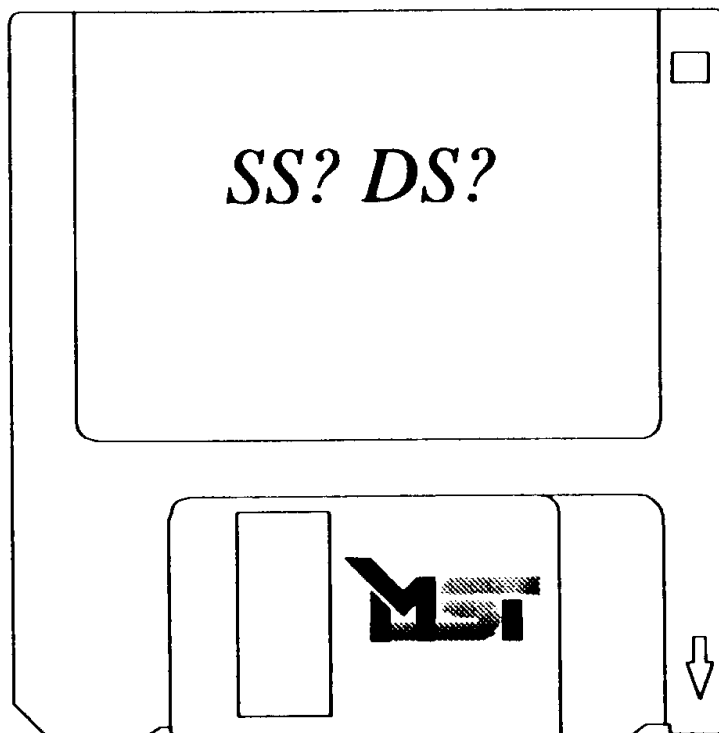
cluster registraties:	201 203 FFF 123 ABC DEF
in bytes:	01 32 20 FF 3F 12 BC FA DE

Naast de FAT kan er ook nog een backup van de FAT aanwezig zijn. Deze neemt evenveel sectoren als de gewone FAT in beslag en volgt meteen achter de gewone FAT, voor de directory dus. MSX-DOS2 gebruikt deze FAT om de oude chains van de gewiste bestanden in op te slaan, zodat UNDEL ze weer terug kan halen. Als er een nieuw bestand op disk wordt aangemaakt, dan wordt de backup FAT echter overschreven met de originele FAT en is het niet meer mogelijk de oude chains automatisch te herstellen.

De Media Identificatie

De eerste byte van de FAT is niet gereserveerd voor een cluster registratie, maar voor het Media ID byte. Deze byte geeft aan wat voor type diskette het is, maar het verpest wel de eerste mogelijk registratie. De tweede registratie wordt ook niet gebruikt, zodat netjes op de derde byte van de FAT de registratie van cluster nummer 2 kan beginnen. Het komt goed uit dat cluster 0 niet bestaat, want als er 0 in de registratie staat is het cluster vrij.

Het Media Identificatie byte wordt gebruikt om aan te geven wat voor type diskette het is, waarvan er op de MSX maar twee van de standaard lijst van 8 worden gebruikt. Het Media ID F8 (hexadecimaal) staat voor een enkelzijdige diskette en F9 voor een dubbelzijdige. MSX-DOS1 gebruikt dit Media ID om te bepalen hoeveel sectoren, sectoren per cluster en clusters er op de disk passen. MSX-DOS2 kijkt echter altijd in de boot sector, omdat bijvoorbeeld een harddisk met een niet gedefinieerde Media ID byte van F0 een volledig zelf in te stellen aantal clusters etc. kan bevatten.



De Directory

Het laatste deel van de diskette administratie beslaat de directory. Zoals al eerder gezegd staat hier alleen de root-directory, sub-directories komen alleen onder MSX-DOS2 voor en worden verderop besproken. De root directory bestaat uit een maximum aantal sectoren, afhankelijk van het ingestelde maximum aantal directory registraties. Standaard gaan er echter 112 registraties in. Ieder registratie beslaat 32 bytes, dus zijn er zeven sectoren nodig om dit allemaal op te slaan. Op een harddisk bijvoorbeeld, kunnen er maximaal 254 registraties in de root-directory en zijn er 16 sectoren nodig. Iedere directory registratie beslaat 32 bytes en is op de volgende manier ingedeeld:

byte	inhoud
0-7	Naam
8-10	Extensie
11	Attributen
12-21	Gereserveerd
22-25	Creatie tijd en datum
26,27	Eerste clusternummer
28-31	Bestandsgrootte

De naam en de extensie bevatten de naam van het bestand. Bijvoorbeeld COMMAND.COM. Merk op dat de "." in de naam niet opgeslagen wordt, deze komt er altijd al tussen, dus is het niet zinnig deze op te slaan. De naam en extensie bevatten de label naam van de disk als het label attribuut gezet is. Dit attribuut byte, dat volgt op de extensie, bevat een zestal bits waarmee het bestand een speciale status gegeven kan worden. De volgende tabel geeft de mogelijkheden:

bit	functie
0	Read-only
1	Hidden
2	System
3	Volume label
4	Sub-directory
5	Archive

De eerste drie bits kunnen het bestand beschermen. Als het 'Read-only' bit gezet is kan er niet meer over het bestand heen geschreven worden. Als een bestand het 'Hidden bit' gezet heeft, dan wordt deze niet meer getoond in het directory overzicht. Het 'System bit' kan gezet worden voor systeem bestanden, bijvoorbeeld MSXDOS2.SYS. Systeem bestanden zijn ook niet zichtbaar in de directory lijst.

Het 'Volume' bit wordt gezet als deze registratie de labelnaam van de diskette is. Zo'n registratie komt alleen in de root-directory voor. Als dit bit gezet is bevatten de naam en extensie de labelnaam en dus niet een bestandsnaam. De label naam wordt onder MSX-DOS2 bovenaan de directory lijst gezet. Onder MSX-DOS1 is deze niet zichtbaar.

Een sub-directory bestaat ook uit een chain van clusters en staat net als een file geregistreerd in een directory. Als het 'Sub-directory' bit gezet is komt er achter de naam het woordje <dir> te staan en kan de directory onder MSX-DOS2 gebruikt worden.

Het 'Archive' bit tenslotte wordt gezet als de inhoud van een bestand geschreven wordt. Een programma dat een backup maakt van bijvoorbeeld een harddisk kan dan alleen die bestanden kopiëren waarvan het 'Archive' bit gezet is, om daarna meteen het bit weer te wissen. Op die manier kunnen alle gewijzigde bestanden eenvoudig gebackupt worden.

Na het attribuut byte volgen 10 bytes die gereserveerd zijn, althans dat waren ze onder MSX-DOS1. Onder MSX-DOS2 wordt het eerste byte gebruikt om de eerste letter van een verwijderde registratie op te slaan. Als een bestand gewist wordt, dan wordt de cluster chain vrijgegeven in de FAT en in de directory registratie de eerste letter van de naam vervangen door een E5 (hexadecimaal). Onder MSX-DOS1 ging de eerste letter verloren, maar onder MSX-DOS2 kan deze dus eenvoudig van de eerste byte van het gereserveerde blok opgehaald worden. Op die manier kan UNDEL dus zijn werk volledig automatisch doen en hoeft nooit de eerste letter van de naam ingegeven te worden.

De volgende vier bytes bevatten de creatiedatum en -tijd van het bestand. Dit is niet opgeslagen in de vorm DD-MM-YY en HH:MM:SS, maar in een gecompriëerde vorm. Voor de tijd is dat in de vorm HHHHHMMM MMMSSSSS, waarbij het laatste byte als eerste opgeslagen is. De letters H,M en S geven aan welke bits gebruikt zijn voor de uren, minuten en seconden. Merk op dat de seconden het aantal seconden gedeeld door twee is. Voor de datum is er eenzelfde opslag methode: YYYYYYYM MMMDDDDD, waarbij Y,M en D voor Year, Month en Day staan. Het laatste byte is weer als eerste opgeslagen.

Na deze bit-reeks om de leeftijd van het bestand mee aan te duiden, volgt het eerste clusternummer van de cluster chain waarin het bestand is opgeslagen. Vanuit de directory entry kan het eerste cluster nummer in de FAT opgezocht worden, waarna in de FAT het volgende cluster gevonden kan worden, als er meerder in de chain zitten.

De laatste vier bytes van de directory registratie zijn de lengte van het bestand in een 32 bits getal. Het minst significante byte staat eerst.

Rekenen aan clusters

De eerste sector achter alle administratie bevat de eerste data. Het eerste cluster slaat dus op die sector. In veel boeken worden vreemde formules aangegeven waarmee het cluster nummer omgerekend kan worden naar het sectornummer van de eerste sector die bij dat cluster hoort, maar die formules zijn meestal alleen van toepassing op de standaard Media ID's F8 en F9.

De juiste algemene formule om een clusternummer om te rekenen naar een sectornummer is echter de volgende:

$(\text{clusternummer} - 2) * \text{sectoren per cluster} + \text{eerste data sector nummer}$

Het clusternummer wordt met twee verminderd omdat de eerste twee registraties in de FAT niet gebruikt worden en het eerste cluster dus nummer 2 heeft. Daarna wordt het getal vermenigvuldigd met het aantal sectoren per cluster om van het groepjes idee naar sector nivo te komen. Als laatste wordt het sectornummer van de eerste datasector erbij opgeteld, omdat daar de data pas begint en dit nummer niet altijd hetzelfde is.

