

Computer Club Masano Schaffen

Waterstraat 38 B-3290 Schaffen



# CCMS

# XPLORER

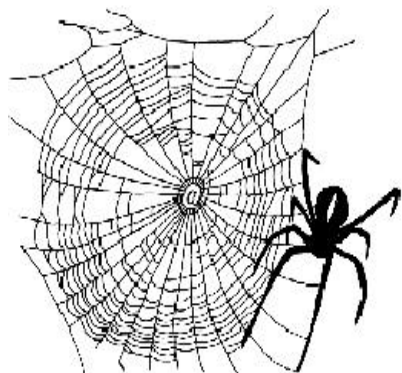
Elke zondag  
Van 10.00 tot 12.30

[info@ccms.be](mailto:info@ccms.be)  
<http://ccms.catsanddogs.com>

Officieel tijdschrift van CCMS, vereniging voor iedereen die computer als hobby heeft.



Losse nummers: € 1



## Korte inhoud

- 1 Windows – Een overzicht (deel 7 vervolg)
- 2 Activiteitenkalender 2007

## Het Bestuur

Voorzitter	Dominique Haesevoets	<a href="mailto:info@ccms.be">info@ccms.be</a>
Secretaris	Robert De Vroe	<a href="mailto:robert@ccms.be">robert@ccms.be</a>
Cultureel afgevaardigde	Michel Gielens	<a href="mailto:michel@ccms.be">michel@ccms.be</a>
Erevoorzitter	Geert Conard	<a href="mailto:geert@ccms.be">geert@ccms.be</a>

### Redactie CCMS-Xplorer

Hoofdredacteur & V.U.	Dominique Haesevoets
Eindredacteur	Michel Gielens
Lay-out	D. Haesevoets
Drukwerk	Tech-Nick
Redactie	M. Gielens, D. Haesevoets

De redactie is bereikbaar via 013/32.88.08 of via E-mail op [redactie@ccms.be](mailto:redactie@ccms.be)

Volgende deadlines CCMS-Xplorer: 16 sep, 14 oct, 11 nov, 16 december

### Werken mee aan dit boekje

Tech-Nick Studio (Drukwerk) [www.tech-nick.be](http://www.tech-nick.be)

### CCMS, Waterstraat 38, 3290 Schaffen

Elke zondag voormiddag van 10:00 tot 12:30

Homepage: [www.ccms.be](http://www.ccms.be)

Algemeen e-mail adres: [info@ccms.be](mailto:info@ccms.be)

Bankrekening CCMS: Fortis 001-297.43.62-30

Het CCMS-lidmaatschap kost € 15 per kalenderjaar. Word je pas lid na 1 augustus betaal je nog € 7,50. Je ontvangt alle CCMS-Xplorer nummers van die periode.

Andere leden van hetzelfde gezin betalen slechts € 10, maar dan ontvangt men wel slechts 1 clubblad.

CCMS-Xplorer is een uitgave van CCMS, Computer Club Masano Schaffen. Verantwoordelijke uitgever: Dominique Haesevoets [redactie@ccms.be](mailto:redactie@ccms.be). CCMS-Xplorer bevat links naar websites die de redactie interessant vond. Deze zijn zuiver informatief en de uitgever is niet verantwoordelijk voor de inhoud van de sites of het gebruik ervan. Geen enkele tekst van CCMS-Xplorer, geheel of zelfs gedeeltelijk, mag overgenomen worden zonder schriftelijke toestemming van de verantwoordelijke uitgever. De uitgever is niet verantwoordelijk voor de inhoud van de advertenties.

## Woord van de voorzitter

Hallo Computeraars,

de grote vakantie loopt alweer op zijn einde. Veel zomer hebben we niet gehad, maar ik hoop dat jullie toch wat van de vakantie hebben kunnen genieten. Persoonlijk ben ik door m 'n werk nog niet aan verlof toe gekomen, maar wat nog niet was, moet nog komen.

Zoals jullie kunnen zien in de kalender beginnen we deze maand met Power Point voor Linux (de gratis versie dus). Robert is de man die zich zal bezig houden met deze cursus. Daarna staat dan Excel op het programma, of, indien er vraag naar andere zaken zou zijn, doen we dat uiteraard. Je merkt het dus: we hebben niet stil gezeten tijdens de verlofperiode.

Je vakantiekaartje blijft uiteraard nog steeds welkom, komt het niet op onze cover, dan hangen we het beslist uit op de club.

Groetjes en nog veel plezier,

Dominique

## Windows – Een overzicht deel 7 (vervolg)

### MIPS is een RISC computer

De Reduced Instruction Set Computer (RISC) is een processor architectuur waarbij heel de instructieset dezelfde tijd nodig heeft om uitgevoerd te worden. Daaruit is de MIPS architectuur gegroeid.

De Reduced Instruction Set Computer (kortweg: RISC) is een processor die werkt met een kleine set simpele basisinstructies. Dit in tegenstelling tot de CISC-processors (Complex Instruction Set Computer), die werken met een relatief omvangrijke set complexe instructies.

De RISC-processor heeft minder schakelingen en is daardoor eenvoudiger te maken en verder te miniaturiseren. Dit wordt bereikt door de kleinere instructieset en door het ontbreken van microcode. In feite wordt de microcode naar het gewone geheugen buiten de processor verplaatst.

Het feit dat de RISC-processor voor dezelfde opdracht meer instructies moet uitvoeren dan een CISC-processor, kan gecompenseerd worden door een hogere verwerkingssnelheid. In de praktijk ontlopen de twee ontwerpprincipes elkaar niet veel in verwerkingssnelheid. Het belangrijkste verschil is het energieverbruik: RISC processoren zijn zuiniger.

Aanvankelijk, na de uitvinding van RISC, werd wel gedacht dat CISC geen lang leven meer beschoren zou zijn, maar doordat de processoren complexer zijn blijken er ook meer ontwerpstrucs mogelijk. De ontwerp- en bouwkosten zijn hierdoor wel hoger, maar dat is bij zeer grote oplages overkomelijk.

De x86-lijn van Intel en AMD heeft een CISC-architectuur. Andere leveranciers zoals Apple, IBM en Sun Microsystems passen ook RISC-processoren toe in hun computers. Voorbeelden hiervan zijn de PowerPC van IBM, Sun's UltraSPARC en de AVR core van ATMEL. Een processor waarin het principe zeer zuiver toegepast wordt is de ARM van ARM Ltd.

### MIPS geheugen

De bedoeling van het geheugen is om groepen bits op te slaan en die door te zenden wanneer ze worden opgevraagd. De meeste huidige computers slaan informatie op in meervouden van 8 bits (of 1 byte). Elke byte krijgt ook een numeriek adres toegewezen. De MIPS processor heeft een 32-bit-adresgrootte, beginnend van 0x00000000 tot 0xFFFFFFFF. In het geheugen kan je zowel programma-instructies als data opslagen. Dus één van de functies van een besturingssysteem is dat het stukken geheugen kan toewijzen aan instructies of data van elk proces. Het geheugen kan zowel als Little Endian en als Big Endian gebruikt worden. Little Endian is wanneer je de minst beduidende bit eerst in het geheugen plaatst. Big Endian is wanneer je de meest beduidende bit eerst in het geheugen plaatst.

### MIPS instructieset

MIPS maakt gebruik van 32 verschillende registers. Hieronder een overzicht van de verschillende registers.

Naam	Nummer	Gebruik
\$zero	\$0	Blijft steeds 0
\$at	\$1	Voorbehouden voor de assembler
\$v0-\$v1	\$2-\$3	Geeft de resultaten van functies
\$a0-\$a3	\$4-\$7	Functie argumenten
\$t0-\$t7	\$8-\$15	Tijdelijke data
\$s0-\$s7	\$16-\$23	Opgeslagen tijdelijke data
\$t8-\$t9	\$24-\$25	Tijdelijke data
\$k0-\$k1	\$26-\$27	Voorbehouden voor de kernel van het besturingssysteem
\$gp	\$28	Globale pointer
\$sp	\$29	Stack pointer
\$fp	\$30	Frame pointer
\$ra	\$31	Terugkeeradres

MIPS instructies zijn 32 bit lang, zelfs bij de 64-bits processors. Dit zorgt voor een betere compatibiliteit met de 32-bit instructieset. Er is ook een mogelijkheid om via een extensie de processor een 16-bit instructieset te laten gebruiken. Dit wordt gebruikt wanneer de codelengte belangrijker is dan de performantie.

De MIPS instructieset architectuur is achterwaarts compatibel, dit wil zeggen dat een MIPS IV processor nog steeds een MIPS I code kan uitvoeren.

De MIPS is een load-store machine. De enige toegelaten geheugenoperaties zijn het laden van waarden in één van de registers of het opslaan van data vanuit een register in het geheugen.

Er zijn drie categorieën van instructies:

- Laad en opslag
  - Laad data van het geheugen in een register, of sla een registerinhoud op in het geheugen.
  - Vb.: lw \$t0, num1 #laad woord van num1 in register \$t0
- Wiskundige en logische bewerkingen
  - Dit laat u toe om wiskundige en logische bewerkingen uit te voeren.
  - Vb.: add \$t0, \$t3, \$t4 # \$t0 = \$t3 + \$t4
- Jump en branch
  - Wordt gebruikt om naar verschillende plaatsen in het programma te gaan.
  - Vb.: bij het gebruik van een IF-instructie.

## MIPS instructie formaten

MIPS heeft drie instructieformaten.

Type	-31-	formaat (bits)	-0-
R	opcode (6 bits)	rs (5 bits) rt (5 bits) rd (5 bits) shamt (5 bits) funct (6 bits)	
I	opcode (6 bits)	rs (5 bits) rt (5 bits) immediate (16 bits)	
J	opcode (6 bits)	address (26 bits)	

## Pseudo-Ops

Hieronder zijn enkele "pseudo-ops" of pseudo-operaties beschreven. Ze zullen het gedrag van de assembler mee bepalen. Gewone opcodes zijn processor gebonden terwijl pseudo-ops specifiek bij een bepaalde assembler horen omdat ze door de assembler zelf worden uitgevoerd.

Pseudo-op	Beschrijving
.2byte expressie1 [ , expressie2 ] ... [ , expressieN]	Verkleint de expressies in de komma-gescheiden lijst (ook wel comma-separated list) naar 16-bits waarden en assembleert de waarden in overervende waarden. De expressies moeten absoluut zijn of, als de labels in dezelfde sectie zijn gedefinieerd, moeten ze gedefinieerd zijn in een verschil van labels(label1-label2). Deze richtlijn kan optioneel de vorm "expressie 1 [:expressie2] hebben. Wat betekent dat expressie 2 de waarde van expressie 1 twee keer kopieert. Deze richtlijn heeft geen automatische schikking (alleen maar 64-bit en N32)
.data	Zegt de assembler om alle volgende data bij de data sectie te zetten.
.end [proc_name]	Duidt het einde van een procedure aan. Gebruik deze verwijzing als je informatie wilt maken voor de debugger.

## Adresseer methoden

Er zijn drie belangrijke manieren om naar data te verwijzen. Hieronder kan je zien hoe MIPS dit implementeert in zijn instructieformaten.

### Direct

De bronoperatoren kunnen constanten zijn. Een constante waarde wordt dan ook direct in machinetaalinstructies geëncodeerd. Hierdoor is het adres onmiddellijk te bereiken nadat de instructie is gedecodeerd, zonder dat het adres uit het geheugen gehaald moet worden.

MIPS instructies vertalen directe constanten in de 16 minst beduidende bits van de directe instructie-layout. Indien de constante groter is dan 16 bits zullen er twee machine instructies worden gegenereerd. De meest beduidende cijfers worden dan geladen in het \$at register met de "load upper immediate" instructie.

## Register adressering

Door het gebruik van registers als bron en bestemming van data wordt de data sneller verwerkt dan bij het gebruik van geheugen. De meeste processoren hebben minstens één wiskundige instructieoperator nodig om register adressering te gebruiken. Instructies waarvan alle operators in registers zitten, zijn het snelst uit te voeren.

MIPS voorziet 32 registers en vraagt om gebruik te maken van deze registers om data te verwerken. Omdat je maar 5 bits ( $2^5 = 32$ ) nodig hebt om naar een register te verwijzen, kan je gemakkelijk naar 3 registers verwijzen in een 32-bit instructie.

## Geheugen adressering

Met 32-bit adressen kan een computer tot 4 GB geheugen toegang krijgen. Hierdoor kan je veel data opslaan, maar heb je ook meer tijd nodig om toegang te krijgen. In een MIPS instructie kan je maximum naar één geheugenadres verwijzen. Net zoals in de meeste supercomputers beperkt MIPS dit tot laad- en opslaginstructies.

## MIPS call- en returnfunctie

Als we een functie (een procedure of een subroutine) oproepen, dan zouden we graag na de functie terugkeren naar de plaats waar we in de programmacode gebleven waren. Om dit te kunnen doen moeten we iets vinden om die plaats te onthouden. Er moet een standaard overeengekomen worden waar de terugkeerplaats wordt opgeslagen. MIPS gebruikt register 31 om dit adres in op te slaan. De functies die worden gebruikt om een functie op te roepen en terug te keren naar de oorspronkelijke instructie zijn:

- jal - jump and link: Hetzelfde formaat als jump maar als verschil dat jal het terugkeeradres opslaat in register 31, genoemd \$ra of return address register.
- jr - jump register: In plaats van een vast doeladres te geven zoals bijvoorbeeld een functienaam, springt deze instructie naar de locatie die is opgeslagen in register 31.

Als een klein voorbeeldje staat hieronder een functie om een integer af te drukken en een andere die een eindelijk karakter afdruckt.

```
print_int:
    li $v0, 1
    syscall
    jr $ra
endline:
    la $a0, endl #define in .data as "\n"
    li $v0, 4
    syscall
    jr $ra
```

Amaai, niet gemakkelijk hé ;)

# Kernel

Een kernel is het meest centrale en meest onzichtbare onderdeel van een besturingssysteem, de kern die alle basisvoorzieningen verzorgt voor de andere delen van het besturingssysteem. De kernel omvat onder meer:

- een voorziening voor de afhandeling van onderbrekingssignalen (interrupt handler), die alle aanvragen en uitvoeringen van invoer/uitvoer-bewerkingen bijhoudt;
- een scheduler, een soort dienstregeling die bepaalt welk programma op welk moment uitgevoerd mag worden;
- een geheugenbeheerder, die aan programma's werkgeheugen toewijst en ervoor zorgt dat zij niet elkaars geheugenruimte kunnen beschadigen.

De diensten van de kernel kunnen door andere delen van het besturingssysteem en computerprogramma's worden gebruikt door middel van een verzameling programma-interfaces die meestal system calls oftewel systeemaanroepen genoemd worden.

## Microkernels, monolitische- en hybridkernels

Kernels worden onderverdeeld in drie categorieën: microkernels, monolitische kernels en hybride kernels. Ieder type kernel heeft een andere toepassing.

### Monolitische kernel

Bij een monolitische kernel wordt de complete kernel in één geheugensegment geladen en vanuit dat segment uitgevoerd. Het voordeel hiervan is dat de kernel sneller te ontwikkelen en onderhouden is. Bovendien is de performance van dit type kernel vaak hoger. Een nadeel is de stabiliteit: wanneer een deel van de kernel crasht, crasht meestal de hele kernel en daarmee de hele computer. Een ander nadeel is de onmogelijkheid om (een deel van) de kernel te vervangen zonder de pc opnieuw te starten. Linux en FreeBSD zijn twee voorbeelden van monolitische kernelbesturingssystemen.

### Microkernel

Een microkernel gooit het helemaal over een andere boeg: de diensten die de kernel biedt worden zoveel mogelijk uitgesplitst in verschillende losstaande processen. Elk proces draait in zijn eigen beschermd gebied. De uitsplitsing van diensten bevordert de stabiliteit enorm: wanneer een deel van de kernel crasht draait de rest door. Het gecrashte onderdeel kan vervolgens simpelweg opnieuw worden gestart. Ook is het mogelijk een deel van de kernel te vervangen zonder de computer opnieuw te hoeven starten. Er kleeft echter ook een nadeel aan dit type kernel: complexiteit. Omdat alle kernel-diensten helemaal opzichzelf staan gaat er veel ontwikkeltijd zitten in de onderlinge communicatie. De performance ligt ook meestal lager dan bij een monolitische kernel, omdat er continu moet worden geschakeld tussen de diensten. Voorbeelden van microkernels zijn: QNX, Symbian OS en Machkernel.

### Hybride kernel

Een hybride kernel is een combinatie van bovengenoemde twee kernels die bepaalde eigenschappen van de twee werelden combineert. Hybride kernels zijn doorgaans

"gegroeide" microkernels, waarbij er meer functionaliteit in een microkernel is geplaatst. Windows NT, BeOS, Darwin (de kernel van Mac OS X) en DragonFly BSD zijn een paar voorbeelden van hybride kernel besturingsystemen.

Anderzijds werden door de komst van multi-processoren, hyper-thread en multi-core de klassieke (vooral Unix-achtige) kernels verplicht te evolueren naar een hybride kernel teneinde SMP (Symmetric Multi Processing) te ondersteunen. Dit was een van de grote veranderingen toen SUN van SUNOS overstapte naar Solaris rond 1993. De Linux kernels zijn slechts sinds een paar jaar in staat om op een efficiënte manier SMP systemen te draaien.

En ten slotte ook nog even over de FAT en NTFS

## File Allocation Table

De File Allocation Table (afgekort tot FAT) is het bestandssysteem dat ontwikkeld is voor MS-DOS en Windows. Het FAT-systeem wordt beschouwd als redelijk simpel en is daarom een populair formaat voor floppydisks; meer zelfs, FAT wordt ondersteund door bijna alle bestaande besturingsystemen voor de pc, daarom wordt FAT dikwijls gebruikt om gegevens te delen tussen verschillende besturingsystemen die op een zelfde pc staan (een multiboot environment). Het wordt ook vaak gebruikt op andere media zoals geheugenkaarten.

FAT is reeds vroeg ontwikkeld, de ontwerpers hebben het zo simpel en compact mogelijk gehouden voor de beperkte geheugens in die tijd. Daardoor heeft FAT verscheidene problemen.

- de eenvoudige bestandsstructuur leidt gemakkelijk tot fragmentatie, wat zorgt voor grote vertraging tijdens bestandsoperaties.
- FAT is niet goed bestand tegen systeemcrashes.
- de eerste versies van FAT lieten slechts bestandsnamen toe van maximaal 11 tekens (8 voor de bestandsnaam, 3 voor de extensie). Daarna werd door Microsoft VFAT ingevoerd, dat bestandsnamen tot een lengte van 255 ondersteunt.
- voor de invoering van clusters was de maximale partitiegrootte 32 Mbyte.
- de grote clustergrootte zorgt voor een inefficiënt gebruik van de beschikbare ruimte bij kleine bestanden.

Desondanks is FAT wijdverbreid. Dat komt doordat MS-DOS veel gebruikt werd als het hoofdbesturingsysteem voor de pc en MS-DOS maakt gebruik van FAT. Wegens zijn eenvoudige structuur is FAT gemakkelijk te implementeren.

### Aantal bits

Een eventueel getal achter de afkorting FAT zegt hoeveel bits er zijn gereserveerd voor de adressering van clusters.



Bestandssysteem	Max. aantal clusters	Wordt gebruikt bij
FAT12	212 = 4096	Diskettes, kleine harde schijven, memorycards
FAT16	216 = 65.536	Harde schijven in MS-DOS. MS-DOS t/m versie 7.00 ondersteunt geen FAT32.
FAT32	232 = 4.294.967.296	Harde schijven in Windows 95 OSR2 (MS-DOS 7.10) en hoger. Pas vanaf Windows 95 OSR2 wordt FAT32 ondersteund.

Zoals te zien valt, is er bij FAT32 niet langer een probleem met de opdeling in clusters. FAT16 is beperkt tot 65.536 clusters en een maximale grootte van 4 gigabyte. Dit heeft tot gevolg dat clusters minimaal 32 kilobyte (kB) groot zijn, waardoor zelfs een bestand van 1 enkele byte al 32 kB in beslag neemt. FAT32 kan in meer clusters opdelen en is ook niet meer gelimiteerd tot 4 gigabyte.

## FAT16

FAT16 (16-bits File Allocation Table) is een bestandssysteem dat gebruikt wordt door o.a. MS-DOS en Microsoft Windows.

Zoals de naam al suggereert werkt FAT met een tabel die op een vaste plek op de schijf opgeslagen is en waarin wordt bijgehouden welke stukken van de schijf door welke bestanden gebruikt worden. De tabel is maximaal 216 (65536) posities groot en bestaat uit 16-bits getallen. In het oorspronkelijke ontwerp kwam iedere positie in de tabel overeen met een enkele sector van 512 bytes, waarmee de grootte van een FAT16-partitie maximaal 32 megabyte kon bedragen. In MS-DOS versie 3.0 werd het mogelijk sectoren samen te voegen in zogenaamde clusters, waardoor harde schijven tot 2 GB gebruikt konden worden (onder Windows NT tot 4 GB).

### Gebruik

FAT16 wordt gebruikt door MS-DOS en Windows op harde schijven en USB-sticks tot 2 GB. Omdat harde schijven tegenwoordig veel groter zijn dan 2 GB, wordt daarvoor geen FAT16 meer gebruikt, maar wel FAT32 of NTFS.

### Hoe werkt het

Het FAT16-systeem bestaat uit 3 delen:

1. de FAT-tabel
2. de hoofddirectory
3. de subdirectory's

### FAT-tabel

De FAT-tabel volgt op de partitie onmiddellijk na de bootsector. Meestal staan er twee identieke FAT's vlak achter elkaar, zodat er bij beschadiging van een FAT altijd een reservekopie beschikbaar is.

De FAT-tabel bevat maximaal 65.536 (216) getallen van twee bytes elk. Elk getal komt overeen met een cluster op de harde schijf, en geeft met een gereserveerde code aan wat de status van de bijbehorende sector of cluster is. Voor een vers geformatteerde schijf is dit voornamelijk de code "leeg" of eventueel "bad".

Bij het aanmaken van een bestand wordt het nummer van de eerste positie bij de informatie over het bestand in de directory opgeslagen. Als het bestand groter is dan één sector of cluster zal in de aangewezen positie het nummer staan van de cluster waar het bestand verder gaat, daar staat weer het nummer van het derde stuk enzovoorts, totdat met een code aangegeven wordt dat het eind van het bestand bereikt is. Als een bestand verwijderd wordt worden alle posities weer als beschikbaar gemarkeerd.

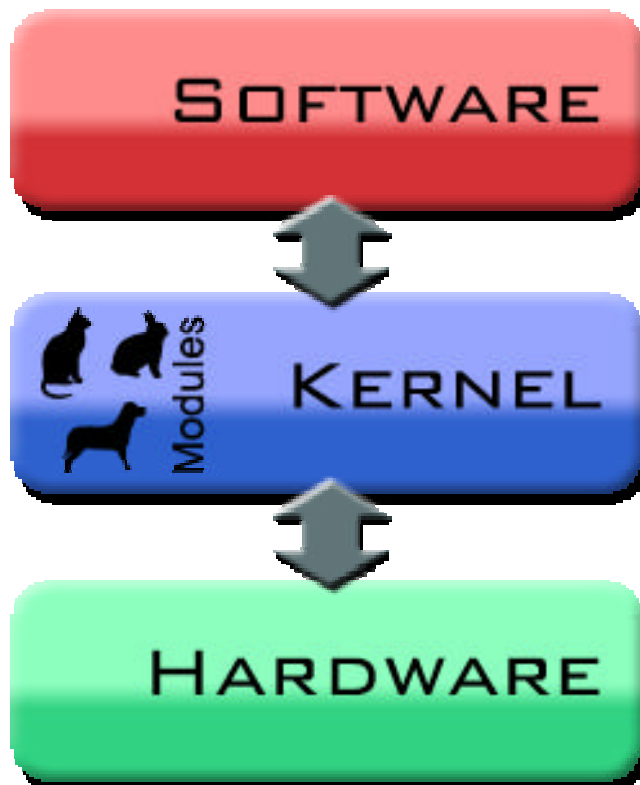
Vereenvoudigd voorbeeld, code 0 is leeg en 99 is "laatste sector". De schijf is eerst leeg.

Plaats	1	2	3	4	5	6	7	8	9
Waarde	0	0	0	0	0	0	0	0	0

Na het toevoegen van een bestand dat op cluster 1 begint en drie sectoren lang is:

Plaats	1	2	3	4	5	6	7	8	9
Waarde	2	3	99	0	0	0	0	0	0

Positie 1 bevat dus het getal "2", dat is het nummer van de volgende cluster waar het bestand verder gaat. Positie 2 bevat het getal "3", en positie 3 bevat de eindcode "99".



# Activiteitenkalender CCMS 2007

## Januari 2007

07-01 Geen club  
 14-01 Gewone clubmeeting **Cursus\_Tekstv.\***  
 21-01 Gewone clubmeeting  
 28-01 Gewone clubmeeting **Cursus\_Tekstv.\***

## Februari 2007

04-02 Gewone clubmeeting  
 11-02 Gewone clubmeeting **Cursus\_Tekstv.\***  
 18-02 Gewone clubmeeting  
 25-02 Gewone clubmeeting + **HMC Beurs Antwerpen**

## Maart 2007

04-03 Gewone clubmeeting  
 11-03 Gewone clubmeeting **Cursus\_Tekstv.\***  
 18-03 Gewone clubmeeting  
 25-03 Gewone clubmeeting **Cursus\_Tekstv.\***

## April 2007

01-04 Gewone clubmeeting  
 08-04 **PASEN Geen club**  
 15-04 Gewone clubmeeting **Cursus\_Tekstv.\***  
 22-04 Gewone clubmeeting  
 29-04 Gewone clubmeeting **Cursus\_Tekstv.\***

## Mei 2007

06-05 Gewone clubmeeting  
 13-05 Gewone clubmeeting **Cursus\_Tekstv.\***  
 20-05 Gewone clubmeeting  
 27-05 Gewone clubmeeting

## Juni 2007

03-06 Gewone clubmeeting  
 10-06 Gewone clubmeeting  
 17-06 Gewone clubmeeting  
 23-06 **MASANO BBQ**  
 24-06 Geen club

## Juli 2007

01-07 Gewone clubmeeting  
 08-07 Gewone clubmeeting  
 15-07 Gewone clubmeeting  
 22-07 Geen club  
 29-07 Gewone clubmeeting

## Augustus 2007

05-08 Gewone clubmeeting  
 12-08 Gewone clubmeeting  
 19-08 Gewone clubmeeting  
 26-08 Gewone clubmeeting

## September 2007

02-09 Geen club  
 09-09 Geen club  
 16-09 Gewone clubmeeting + **Cursus**  
 23-09 Gewone clubmeeting + **Cursus**  
 "Presentatie" Impress of Powerpoint  
 30-09 Gewone clubmeeting

## October 2007

07-10 Gewone clubmeeting + **Cursus**  
 "Presentatie" Impress of Powerpoint  
 14-10 Gewone clubmeeting  
 21-10 Gewone clubmeeting  
 28-10 Gewone clubmeeting

## November 2007

04-11 Gewone clubmeeting  
 11-11 Geen club  
 18-11 Gewone clubmeeting  
 24-11 **5<sup>e</sup> Masano avondwandeling**  
 25-11 Gewone clubmeeting

## December 2007

02-12 Gewone clubmeeting  
 09-12 Gewone clubmeeting  
 16-12 Gewone clubmeeting  
 23-12 Gewone clubmeeting  
 30-12 Geen club

Deze kalender kan wekelijks aangepast worden

Hij staat op onze website

<http://www.ccms.be>

En natuurlijk in ons maandblad  
 CCMS Xplorer

E-mail bestuur: [info@ccms.be](mailto:info@ccms.be)

Deze uitgave is mede mogelijk dankzij onderstaande sponsors:

## Tech-Nick

Desktop PC's, Laptops en randapparatuur,  
kwaliteit aan scherpe prijzen.  
Webhosting, Webdesign en Printjobs

Erkend Norman Dealer  
Belgacom Partner  
Topcom Telecom Dealer

Wat niet in de lijst staat kan er steeds bijkomen. [info@tech-nick.com](mailto:info@tech-nick.com)  
Tel/Fax 013/32.88.08  
Of tijdens de wekelijkse bijeenkomsten van CCMS bij Dominique

Verzekeringsmakelaar BVBA Maes  
(c.d.v. nr. 11669)

### Agentschap Goffin-Bank

Kloosterbergstraat 33  
3290 Diest  
tel 013/33.40.73.

E mail [vbamaes@skynet.be](mailto:vbamaes@skynet.be)

## REMA

### Keukens

### Renovatie Woningen

0475/40 12 82



# Bookplus

BOEKHANDEL SAENEN

Schoonaerde 48 3290 Schaffen

tel:013/312365 fax :013/523606

e-mail:[kristel.saenen@skynet.be](mailto:kristel.saenen@skynet.be)

openingsuren :ma-vrij van 05.30u -12.30u en 13.00u-18.00u  
zat van 06.30u-17.00u zondag en maandagnamiddag gesloten



Een slimme zet

Marcel, Erna

[www.jet.be](http://www.jet.be)

Schoonaerde 70 – 3290 Schaffen  
Leuvensesteenweg 311, 3293 Kaggevinne

Sint-Hubertusplein 52  
3290 Schaffen  
Tel: 013 55 04 70  
[www.fortisbank.com](http://www.fortisbank.com)




Hasseltsestraat 27  
3290 Diest - 013 32 24 82

[www.optiekpunie.be](http://www.optiekpunie.be)

# De Kwaffeur Schaffen



Open: Ma. Di. Do. Vr. 16—23 u.  
Wo. 13:30—23 u.  
Za. Zo. 9:30—23 u.

Sporthal  
MASANO  
Waterstraat 38  
3290 Schaffen  
013/33.77.63