3.1 Computers en getallen

Of het nu gaat om de uitvoering van opdrachten, of om het opslaan van gegevens, computers werken alleen maar met getallen. Een opdracht voor de microprocessor, die het hart van je computer is, is altijd een combinatie van getallen. Een soort telefoonnummer. De ene combinatie telt op de andere combinatie trekt af, weer een andere combinatie leest iets uit het geheugen enzovoorts.

Ook de opslag van gegevens vindt plaats in de vorm van getallen, ook als het om lettertekens gaat zoals bij tekstverwerking. Lettertekens staan in de zogenaamde ASCII-tabel. Lettertekens hebben daarin een bepaalde volgorde. Het nummer in de tabel is de waarde van het letterteken.

De ASCII-tabel is als bijlage in dit boek opgenomen. In deze tabel is bijvoorbeeld de letter ’A’ opgenomen onder nummer 65. De kleine letter ’a’ heeft nummer 97. Voor een computer is de hoofdletter ’A’ dus absoluut iets anders dan de kleine letter ’a’.

Al die getallen, die een computer beheerd, als hij bezig is met het uitvoeren van een programma staan niet genoteerd in het tientallig stelsel, waar wij gewoonlijk in rekenen, maar in het tweetallig stelsel. Dit stelsel wordt ook wel het binaire stelsel genoemd.

Om een computer echt te kunnen programmeren moet de programmeur weten op welke manier een computer omgaat met getallen. Het kan zijn dat je nu niet zo erg de noodzaak inziet om te leren rekenen in het binaire stelsel. Bij het verder werken in dit boek zal echter blijken dat deze kennis onontbeerlijk is.

3.2 Het tientallig stelsel

Om te leren werken in het binaire stelsel is het nodig dat we ons realiseren op welke manier een getal in het tientallig stelsel in elkaar zit. In het tientallig stelsel beschikken we over tien cijfersymbolen: de cijfers 0 tot en met 9. Als we een getal, bijvoorbeeld 165, bekijken dan zie je dat de waarde van het cijfer bepaald wordt door de positie die het inneemt in het geschreven getal. Het getal 167 is als volgt opgebouwd:

7	×	1	=	7
6	×	10	=	60
1	×	100	=	100
totaal				167

Een decimaal getal begint dus aan de rechterkant met eenheden, de volgende positie zijn de tientallen dan volgen de honderdtallen, de duizendtallen enzovoorts. Je ziet dus dat de waarde vanaf links telkens met tien vermenigvuldigd wordt:

7 × 1	=	7
6 × 1 × 10	=	60
1 × 1 × 10 × 10	=	100
Totaal		167

Je kunt ook zeggen:

7 × 10 tot de macht 0	=	7
6 × 10 tot de macht 1	=	60
1 × 10 tot de macht 2	=	100

Totaal	167

Mensen hebben het tientallig stelsel waarschijnlijk ooit ontwikkeld omdat ze nu eenmaal de beschikking hebben over tien vingers. Voor computers is het tientallig stelsel niet zo handig. Zoals ik al eerder aangaf werkt de computer met een tweetallig stelsel. Dit heeft een technische reden. Als je naar het elektronische geheugen van een computer kijkt dan zou je dit kunnen zien als een enorme hoeveelheid schakelaartjes die ’aan’ of ’uit’ staan. ’Aan’ of ’uit’ zijn maar twee mogelijkheden. Als je ’uit’ als een 0 ziet en ’aan’ als 1, dan is de automatische conclusie dat een computer slechts de beschikking heeft over twee cijfersymbolen namelijk 0 en 1. Daarom werkt een computer niet volgens het tientallig, maar volgens het tweetallig stelsel oftewel binair.

3.3 Het binaire stelsel

In principe werkt het tweetallig stelsel op precies dezelfde wijze als het tientallig stelsel. Het enige verschil is dat hier niet grondgetal 10 als uitgangspunt genomen wordt maar het grondgetal 2. Evenals in het tientallig stelsel wordt de waarde van een cijfer bepaald door de positie die het inneemt in het geschreven getal.

Laten we het binaire getal 10100111 eens onder de loep nemen. Net als bij de ontleding van het decimale getal beginnen we aan de rechterkant van het getal.

	1	×	1	=	1	
	1	×	2	=	2	
	1	×	4	=	4	
	0	×	8	=	0	
	0	×	16	=	0	
	1	×	32	=	32	
	0	×	64	=	0	
	1	×	128	=	128	
Totaal					167	decimaal

We zien dat er in dit geval geen sprake is van eenheden, tientallen en honderdtallen maar van eenheden, tweetallen, viertallen, achttallen enzovoorts. Net als bij het cijfer in het tientallig stelsel wordt de waarde bepaald door de positie in het getal. Alleen hier wordt de waarde van die positie niet berekend met behulp van het getal tien maar met het getal twee. Naar links opschuivend wordt de waarde telkens een macht van twee hoger.

Het binaire getal, dat we zojuist ontleed hebben zou je ook kunnen zien als acht schakelaartjes die ’aan’ of ’uit’ staan. Deze schakelaartjes worden bits genoemd. Bit is een afkorting van binary digit (tweetallig cijfer). Ik heb niet toevallig een getal van acht cijfers als voorbeeld genomen. Een groep van acht bits wordt een byte genoemd.

De IBM-PC is een bytegeadresseerde computer. Dit wil dus zeggen dat iedere byte die in het geheugen van de computer staat een eigen adres heeft en via dit adres terug te vinden is. Een adres is een met een getal aangeven plaats in het geheugen.

Natuurlijk kun je ook een decimaal getal vertalen naar een binair getal.

We nemen weer het getal 167.

167	:	2	=	83	rest	1
83	:	2	=	41	rest	1
41	:	2	=	20	rest	1
20	:	2	=	10	rest	0
10	:	2	=	5	rest	0
5	:	2	=	2	rest	1
2	:	2	=	1	rest	0
1	:	2	=	0	rest	1

We komen via deze berekening weer op 10100111.

Misschien is het voorbeeld nog wat duidelijker als we het getal op deze manier bewerken met het tientallig stelsel. We krijgen dan:

167	:	10	=	16	rest	7
16	:	10	=	1	rest	6
1	:	10	=	0	rest	1

Zo ontstaat dan weer 167.

3.3 Het hexadecimale stelsel

Een serie binaire enen en nullen wordt ook wel een bitpatroon genoemd. Het werken als programmeur met dit soort grote reeksen nullen en enen vraagt om fouten. Het telkens omrekenen van het binaire getal naar het voor ons vertrouwde decimale stelsel is echter nogal omslachtig. Een bitpatroon wordt daarom meestal weergegeven in het zestientallig stelsel. Dit stelsel wordt het hexadecimale stelsel genoemd.

Als we een byte in twee delen splitsen van ieder vier bits, dan passen in zo’n groepje bits 16 verschillende waarden. Dit zijn de waarden nul tot en met vijftien. Zo’n groepje van vier bits heet een nibble. Als we iedere mogelijke waarde van een nibble zouden willen afbeelden in één cijfer, dan zouden we dus moeten werken in een zestientallig- ofwel het hexadecimaal stelsel. In dat geval hebben we dus zestien verschillende cijfersymbolen nodig. In het decimale stelsel beschikken we slechts over tien cijfersymbolen namelijk nul tot en met negen. Om hexadecimaal te kunnen werken hebben we dus zes extra symbolen nodig. Hiervoor zijn gekozen de letters A tot en met F. Deze letters representeren in het hexadecimale stelsel de waarden tien tot en met vijftien. Dus A = 10, B= 11, C 12, D = 13, E = 14 en F = 15.

Als we ons decimale getal 167 nu omzetten naar het hexadecimale stelsel dan krijgen we:

167	:	16	=	10	rest	7
10	:	16	=	0	rest	A

Een hexadecimale schrijfwijze voor het decimale getal 167 is dus A7.

Welk verband is er nu met het binaire getal 10100111? Als we deze byte nu eerst eens splitsen in twee nibbles dan krijgen we:

1010 0111

De eerste nibble is: 1010

0	×	1	=	0

	1	×	2	=	2	
	0	×	4	=	0	
	1	×	8	=	8	
totaal					10	decimaal = A hexadecimaal.

De tweede nibble is: 0101

	1	×	1	=	1	
	1	×	2	=	2	
	1	×	4	=	4	
	0	×	8	=	0	
totaal					7	decimaal = 7 hexadecimaal

Samen vormen deze twee nibbles dus het hexadecimale getal A7:

	7	×	1	=	7	
	A	×	16	=	160	
totaal					167	decimaal

Met een beetje oefening kun je nu zeer snel ingewikkelde bitpatronen omzetten naar een korte notatie volgens het hexadecimale stelsel.

3.4 Opgaven:

1. Zet de volgende decimale getallen om naar het binaire stelsel en maak er vervolgens hexadecimale getallen van:

   a) 73
   b) 183
   c) 255
   d) 23
   e) 205

2. Zet de volgende hexadecimale getallen om naar binaire getallen en naar decimale getallen.

   a) 11
   b) CA
   c) 3B
   d) F0
   e) 3E