

Assembler, de eerste stappen ...

Opslaan

VOORBEELD PROGRAMMA 01

```
0100      MOV AH, 02    // SUBF 02 => één KARAKTER PRINTEN
0102      MOV DL, 41    // VUL REGISTER MET 'A'
0104      INT 21        // PRINT
0106      MOV DL, 42    // VUL REGISTER MET 'B'
0108      INT 21        // PRINT
010A      MOV DL, 43    // VUL REGISTER MET 'C'
010C      INT 21        // PRINT
010E      INT 20        // EINDE
```

OPSLAAN:

N VOORBEELD01.com

R CX
CX 0000
000E

W

```
C:\>debug
-r cx
CX 0000
:E
_
```

```
C:\>Voorbeeld.com
ABC
C:\>
```

De teller bepaald hoeveel bytes er worden opgeslaan!

Assembler, de eerste stappen ...

Opslaan

VOORBEELD PROGRAMMA 02

0100	MOV AH, 01	→ Subfunctie 01 = inlezen karakter (<i>in AL</i>)
0102	INT 21	→ <i>Trigger De Interupt (INT 21, sub01 ...)</i>
0104	MOV DL, AL	→ <i>Inhoud Van AL Naar DL</i>
0106	MOV AH, 02	→ <i>Subfunctie 02 = één karakter printen (uit DL)</i>
0108	MOV CX, 0A	// <i>TELLER OP 10</i>
010B	INT 21	
010D	LOOP 010B	// <i>REPEAT "INT 21</i>
010F	INT 21	
0111	INT 20	

De instructie INT 20 (=2bytes) staat op adres 0111
-> Einde van het programma op 113!

OPSLAAN

n VOORBEELD02.com

R CX

CX 0000

0013

W

→ Input 'q'

```
C:\>VOORBEELD02
qqqqqqqqqqqqqq
```

→ Input 'a'

```
C:\>VOORBEELD02
aaaaaaaaaaaaaa
```

Assembler, de eerste stappen ...

Opslaan

VOORBEELD PROGRAMMA 03

```
0100      MOV AH, 09          // Printen string (sub 09)
0102      MOV DX, 150        // pointer = 120
0104      INT 21
0106      INT 20
```

```
E 150 "HELLO WORLD$"
```

OPSLAAN

N VOORBEELD03.com

R CX

CX 0000

0060 → Teller waarde zodat de data startend op locatie 150 ook wordt opgeslaan. In het voorbeeld: instructies van 100 tot 108 = 8 bytes instructies. Maar de te printen karakters beginnen op locatie 150. Einde van de te printen karkaters op 105B. Minimale teller waarde is dus 5C !!

W



```
C:\>voorbeeld03
HELLO WORLD
C:\>
```

Assembler, de eerste stappen ...

Opslaan

Terug openen van een programma in debug

C:\> voorbeeld03.com

Programma wordt uitgevoerd...

C:\> DEBUG voorbeeld03.com

Debug wordt geopend met programma voorbeeld03 in het geheugen.

u 100 → = 'Unassemble' → assembler code van het programma in het geheugen (= voorbeeld03.com) wordt getoond vanaf adres 0100

```
C:\>debug voorbeeld03.com
-u 100
1481:0100 B409      MOV     AH,09
1481:0102 BA5001    MOV     DX,0150
1481:0105 CD21      INT     21
1481:0107 CD20      INT     20
```

*Opmerking: Code na adres 1481:0107 is geen programma code!
("Dit is 'random data' vertaald naar assembler instructies.")*

OPMERKINGEN, Gebruik van DOS

Gebruik van DOS

DIR → Toont de bestanden in de actieve directroy

CLS → Clear screen

CD .. → Change directory

CD XX → Change to directory XX

Help → Een overzicht van de bruikbare commando's.

Ctrl – alt F1 = qwerty

Crt – alt F2 = azerty