

Assembler cursus opgaven deel III

6 oktober 2006

Opgave 9

Opgave 9.1 Maak een Assembler-programma met een sub-routine. Deze sub-routine moet zichzelf 100 keer aanroepen. Druk na het starten en voor het verlaten van elke sub-routine af dat de routine danwel gestart of gestopt is. Druk ook het nummer van de desbetreffende sub-routine af. Uiteindelijk krijg je ongeveer de volgende uitvoer:

```
Routine 1 is gestart.  
Routine 2 is gestart  
Routine 3 .....  
.....  
Routine 100 is gestart  
Yeah dat waren 100 geneste sub-routines  
Routine 100 is succesvol gestopt  
Routine 99 .....  
.....  
Routine 1 is succesvol gestopt
```

Maak gebruik van de routine van opdracht 8 of SWI OS_ConvertCardinal1 om het nummer van de routine af te drukken. Zorg dat elke sub-routine de PSR onveranderd laat.

Opgave 9.2 maak het hele programma 26/32 bits neutraal.

SWI OS_ConvertCardinal1

Deze SWI converteert een integer naar een string

Invoer:

R0 = waarde die geconverteerd moet worden

R1 = pointer naar buffer waarin de string opgeslagen moet worden

R2 = buffergrootte

Uitvoer:

R0 = pointer naar buffer (zelfde waarde als R1 was meegegeven)

R1 = pointer naar de afsluitende nul in buffer

R2 = aantal vrije bytes in buffer

Opgave 10

Opgave 10.1 Maak een utility `Savesprite <spritenaam> <padnaam>` die de opgegeven sprite in de Wimpsprite-poel zoekt en vervolgens als sprite-bestand bewaart. Maak hierbij gebruik van de routines die je gemaakt heb bij opdracht 6 en 7 en zorg dat foutmeldingen van SWI's goed worden afgehandeld. Geef natuurlijk ook zelf een foutmelding wanneer de sprite niet bestaat. Gebruik `SWI OS_Module` om geheugen te alloceren of vrij te geven. Het formaat van een sprite-bestand is bijna identiek aan de sprite-poel alleen ontbreekt het 1e word (grootte van sprite-poel).

Opgave 10.2 Zorg dat je utility 26/32 bits neutraal is.

OS_Module 6

Alloceer een blok geheugen

Invoer:

R0 = 6

R3 = grootte geheugenblok

Uitvoer:

R0 onveranderd

R2 = pointer naar gealloceerde geheugenblok

R3 onveranderd

OS_Module 7

Geef geheugen vrij

Invoer:

R0 = 7

R2 = pointer naar geheugenblok

Uitvoer

R0 onveranderd

R2 onveranderd

OS_File 10

Bewaart een blok geheugen in een bestand.

Invoer:

R0 = 10

R1 = pointer naar bestandsnaam

R2 = bestandstype (bits 0 - 11)

R4 = start-adres van geheugenblok

R5 = eind-adres van geheugenblok

Uitvoer:

Registers blijven onveranderd