

Tentamen Computerarchitectuur, 12 juni 2012

=====

Beantwoord de vragen over computerarchitectuur kort en bondig. Lange, uitgebreide antwoorden kunnen een reden zijn om een opgave als 'onvoldoende' te beoordelen.

Geef duidelijk aan bij welke vraag je antwoord hoort.

Succes!

=====

=====

1. Wat zijn de niveaus die door Tanenbaum in de computerarchitectuur worden onderscheiden. Geef bij elk niveau kort aan wat de functie van het niveau is.
2. Een CPU bevat zelf een aantal componenten. Bekende componenten zijn de `Program Counter`, het `Instruction Register`, de `ALU` en de `cache`. Geef van deze onderdelen kort aan wat hun functie is.
3. De volgende 20 bytes geven de inhoud van de bytes van een binaire file die door een 32-bits big-endian computer zijn geschreven (de linker kolom zijn de byte-indices, de overige kolommen geven de inhoud van de bytes); letters zijn ascii-characters, getallen zijn numerieke byte-waarden). De bytes op adres 16 t/m 19 moeten worden geïnterpreteerd als een 32 bits waarde.

0000 h e l l o 32 w o

```
0008   r   l   d   0   0   0   0   0
0016   0   0   4   3
```

Iemand schrijft een programma dat de tekst uit de eerste 16 bytes en vervolgens de binaire waarde die in de volgende 4 bytes is opgeslagen inleest.

Vraag:

- Wat is de numerieke waarde (decimale of hexadecimale waarde is OK,

indien hexadecimaal laat de waarde dan beginnen met 0x) die in de bytes op adressen 16 t/m 19 is opgeslagen (NB: dus 1 numerieke waarde, niet 4 waarden)

- Wijzig de file zodanig dat een 64-bits little endian computer dezelfde tekst en dezelfde waarde zal lezen. Omdat de little endian computer een 64 bits architectuur heeft dient de waarde dus niet alleen omgezet te worden naar een little-endian architectuur, maar ook naar een 64-bits architectuur.

4. Teken een decoder voor vier uitgangssignalen.

5. Teken de opbouw (en inhoud) van de stack van de IJVM

a) direct na de aanroep van van een functie f1 met vier lokale variabelen b1, b2, b3 en b4. De aanroep gebeurt m.b.v een call-instructie (1 byte), waarna het adres van de aan te roepen functie is vermeld. Adressen en elementen op de stack bestaan uit 4 bytes. De variabelen kunnen in de stack worden aangegeven met hun naam; hun waarde hoeft niet te worden vermeld.

b) direct nadat het stack-frame is opgebouwd.

De stack begint op een laag adres en 'groeit' naar hogere adressen. De beginsituatie van de stack, net voorafgaand aan de aanroep van de

functie

f1 is:

```
-----  
                                stack  
-----  
register      element  adres  
-----  
    SP        ->    a3     108  
                a2     104  
                a1     100  
    LV        ->    80     96  
                2013    92  
-----
```

De instructie om f1 aan te roepen staat op adres 1000.

6.

a) Waarom hoeft een programmeur (of een vertaler, zoals de C vertaler),

die een computer gebruikt waarin een besturingssysteem is geïnstalleerd, niet zelf complexe opdrachten te programmeren, waarbij

hij/zij alleen kan beschikken over de standaard instructies van de

instructie-set architectuur? (zulke complexe opdrachten zijn bijvoorbeeld 'schrijf een tekstregel naar het beeldscherm', 'lees the

character van het keyboard', 'lees een sector van de harde schijf'.)

b) Beschrijf (in algemene termen, of met behulp van een code-voorbeeld)

hoe dergelijke complexe opdrachten in een programma worden uitgevoerd.