

TENTAMEN ORIENTATIE INFORMATICA

6 november 2012, 9:00-11:00 uur



Dit tentamen is GEEN open boek tentamen. Je mag WEL één van de artikelen meenemen uit IEEE Computer Magazine danwel IEEE Computational Intelligence Magazine.

Voorzie de in te leveren bladen van je naam, en nummer ze. Schrijf op het eerste blad het aantal ingeleverde bladen. Je hoeft maar drie opgaven te maken. Voor iedere opgave krijg je een cijfer tussen 1 en 10. De cijfers worden opgeteld en gedeeld door drie. Als je meer maakt kijk ik slechts de eerste drie na, tenzij je een antwoord duidelijk doorstreept! Succes!

Opgave 1. NOAG-ict

De NOAG-ict bevat veel dwarsverbanden tussen de thema's, deels omdat verschillende vakgebieden in meerdere thema's aan de orde komen, deels omdat ze bij dezelfde maatschappelijke ontwikkelingen van belang zijn. Kies twee van de NOAG-ict thema's, en schrijf een essay van 350-400 woorden, waarin je in je eigen woorden uitlegt wat de dwarsverbanden zijn, zowel maatschappelijk als wetenschappelijk.

Opgave 2. Eén van de artikelen waar je geen essay over hebt geschreven!!

Neem het artikel dat je bij je hebt *maar waar je geen essay over hebt ingeleverd!!* Schrijf een samenvatting van 350-400 woorden, waarin je de belangrijkste stellingnames verwoordt, en (met name als er veel voorbeelden zijn) de meest aansprekende voorbeelden kort beschrijft.

Opgave 3. Geschiedenis

De computer kan gezien worden als een redelijke benadering van de universele Turingmachine. Is het mogelijk om ook met de techniek van de 19de eeuw een redelijke benadering van de Turingmachine te maken? Zo nee, waarom niet? Zo ja, noem enkele belangrijke verschillen tussen die 19de-eeuwse 'computer' en de huidige. Geef in beide gevallen aan welke technologische ontwikkelingen van de 20ste eeuw een belangrijke rol speelden bij het realiseren van de huidige computer.

Opgave 4. Software Engineering

Schrijf in een kort opstel van maximaal 350-400 woorden op wat voor werk je zou doen (wat je taken zijn) als je in de toekomst software architect zou zijn. Verzin een fictief product dat je moet produceren. Probeer een duidelijke relatie tussen deze baan en Software Engineering (SE) te leggen. Geef hierbij antwoord op de volgende twee vragen: Wat zullen de (SE) problemen zijn waar je mee te maken krijgt en wat zijn de uitdagingen in dit domein m.b.t. SE?

Opgave 5. Gedistribueerde systemen

Ieder onderdeel 3 punten, één punt gratis.

- Geef een definitie van een gedistribueerd systeem. Geef een voorbeeld, en leg uit waarom het een gedistribueerd systeem is.
- Wat is scalability, geef een definitie, en leg uit waarom het belangrijk is in het voorbeeld dat je net hebt gegeven.
- Wat is transparency, geef een definitie, en leg uit waarom het belangrijk is in het voorbeeld dat je net hebt gegeven.

Opgave 6. Parallel rekenen We rekenen op een computer met $p = 2000$ processoren en verkrijgen een speed-up van $s(2000) = 1000$. Eén punt gratis.

- Geef de algemene vorm van de speed-up als functie van het aantal processen, en verklaar deze functie. (2pt)

- (b) Wat is de waarde van de verhouding $p_0 = \frac{T_{par}}{T_{seq}}$ voor dit programma, met T_{par} en T_{seq} de rekestijden voor de parallesiseerbare en strikt sequentiële onderdelen op één processor? (2pt)
- (c) Wat zal de speed-up zijn op een systeem voor 100 processoren? (2pt)
- (d) Wat zal de speed-up zijn op een systeem voor 10000 processoren? (1pt)
- (e) Is het zinvol om 10000 processoren in te zetten? Licht je antwoord toe. (2pt)

Opgave 7. Visualisatie

Data visualisatie houdt zich bezig met het produceren van afbeeldingen die inzicht bieden in grote datasets. Het vakgebied is opgedeeld in twee gebieden: Wetenschappelijke visualisatie houdt zich vooral bezig met het verschaffen van inzicht in fysische processen. Informatie visualisatie houdt zich bezig met inzicht verschaffen in abstracte datasets.

Schrijf een essay (350-400 woorden) waarin je deze twee gebieden onderling vergelijkt. Geef van ieder gebied tenminste één voorbeeld en leg uit

- wat het op te lossen probleem is (wat wil de gebruiker zien?)
- welke soort data een rol speelt
- wat de belangrijkste uitdagingen zijn (met name omdat de gebruikers niet informatici zijn).
- welke aspecten informatie-visualisatie moeilijker maken dan wetenschappelijke visualisatie
- wat de hoeveelheden data zijn die je moet verwerken
- wat de rol van interactie is.

Opgave 8. Computer Vision

In het volgende mag je uitgaan van binaire beelden. Ieder onderdeel 3 punten, één punt gratis.

- (a) In connected filtering spelen z.g. “connected components” (samenhangende componenten) een belangrijke rol. Leg in je eigen woorden uit wat een connected component is. Teken een simpel voorbeeld waarin je aangeeft wat je bedoelt.
- (b) Attributen kunnen gebruikt worden om componenten te beschrijven. Leg in je eigen woorden uit wat een attribuut-filter doet. Geef een mogelijke toepassing.
- (c) Het is mogelijk om z.g. clusterende connectiviteit te gebruiken in attribuut filters. Leg uit wat dit doet, en wanneer dit nuttig is. Geef ook een toepassing als voorbeeld.

Opgave 9. Wederzijdse uitsluiting

Ieder onderdeel 3 punten, één punt gratis.

- (a) Er zijn een aantal processen die tegelijk een niet-eindigende lus uitvoeren:

```
while (true) {
    NCS // andere activiteit
    intro
    CS // kritische sectie
    exit
}
```

Het probleem van wederzijdse uitsluiting is de commando's `intro` en `exit` zo te programmeren dat er nooit meer dan één proces tegelijk in CS is. We mogen hiertoe variabelen gebruiken die door alle processen gelezen en geschreven worden.

Beschrijf in één of enkele alinea's het belang van wederzijdse uitsluiting voor computerprogramma's en computersystemen.

(b) Over implementaties.

Bij het toilet gebruikt men gewoonlijk een deur die van binnen op slot gedaan kan worden. We kunnen dit modelleren met een gedeelde variabele

```
boolean open = true ; // initieel

while (true) {
    NCS // andere activiteit
    while (! open) skip ; // wacht tot de deur open is
    open = false ; // doe de deur achter je op slot
    CS // kritische sectie
    open = true ; // doe de deur weer open
}
```

Dit werkt helaas niet voor processen, want twee of meer processen zouden net na elkaar `open == true` kunnen lezen en dan allebei na elkaar `open = false` zetten en CS in kunnen gaan. Wel correct is:

```
while (true) {
    NCS // onbelangrijk
wacht: < if (! open) goto wacht ; else open = false ; >
    CS // kritische sectie
    open = true ;
}
```

De hoekjes `<S>` om een commando `S` drukken uit dat het commando `S` atomair uitgevoerd moet worden, dus zonder interferentie van andere processen. De opdracht "goto wacht" is een sprongopdracht (wacht is het label waarnaartoe gesprongen wordt).

Leg uit waarom dit process niet eerlijk is.

(c) Om oneerlijkheid te voorkomen voert men, b.v. bij de apotheek, een tellertje in. Dit is een gedeelde variabele

```
int teller = 0 ; // initieel
```

die in één atomaire opdracht gelezen en opgehoogd kan worden volgens

```
< eigenTeller = teller ; teller = teller + 1 ; >
```

Programmeer nu eerlijke wederzijdse uitsluiting met deze teller. Zorg dat elk proces keurig op zijn beurt wacht. Argumenteer dat aan wederzijdse uitsluiting voldaan wordt en dat elk wachtend proces ooit aan de beurt komt. Je mag zelf desgewenst andere gedeelde variabelen en privévariabelen invoeren, maar houd het eenvoudig. Zorg dat alle variabelen goed geïnitieerd zijn.