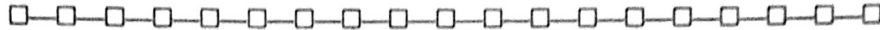


TENTAMEN ORIENTATIE INFORMATICA

11 november 2010, 9:00-12:00 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. Met elk van de 10 vragen is 3 punten te verdienen. **Je hoeft dus maar drie opgaven te maken.** Als je meer maakt kijk ik slechts de eerste drie na, tenzij je een antwoord duidelijk doorstreept! Je krijgt 1 punt gratis. Succes!

Opgave 1. NOAG-ict

Kies één van de NOAG-ict thema's, en schrijf een essay van 350-400 woorden, waarin je in je eigen woorden uitlegt wat het thema omvat, wat de belangrijkste uitdagingen zijn, en wat de belangrijkste maatschappelijke belangen zijn. Noem ook een aantal vakgebieden binnen de informatica die een rol spelen bij het thema.

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 verwoord, en (met name als er veel voorbeelden zijn) de meest aansprekende voorbeelden kort beschrijft. Geef ook aan welke NOAG-ict thema's een rol spelen.

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 een kort opstel (350-400 woorden), waarin je aangeeft hoe een bedrijf efficiënt goede software kan ontwikkelen. Leg ook uit waarom je oplossingen werken! Behandel ten minste drie van de volgende vijf aspecten: Analysis, Design, Verification and Validation, Processes, Professional Practice.

Opgave 5. Gedistribueerde systemen

Astronomen koppelen verschillende radiotelescopieën verspreid over de gehele wereld tot één gigantische telescoop, om zo fijnere details te kunnen observeren. Dit uiteraard een gedistribueerd systeem. Schrijf een essay van 350-400 woorden over de uitdagingen in het ontwerpen en onderhouden van een dergelijk gedistribueerd informatie systeem (b.v., heterogeneity, scalability, openness, etc.). Geef van ieder term die je gebruikt ook aan wat het betekent en maak een afweging van het belang in deze context (b.v. door ze te prioriteren).

Opgave 6. Parallel rekenen

We rekenen op een computer met $p = 2000$ processoren en verkrijgen een speed-up van $s(2000) = 1000$.

- Wat is de waarde van de ratio $p_0 = \frac{T_{par}}{T_{seq}}$ voor dit programma?
- Wat zal de speed-up zijn op een systeem for 100 processoren?
- Wat zal de speed-up zijn op een systeem for 10000 processoren?

Opgave 7. Non-photorealistic rendering

In computer graphics was in het algemeen het doel om beelden te maken die zo goed mogelijk op echte

foto's lijken. Pas recent zijn onderzoekers technieken gaan ontwikkelen die zich van dit ene doel "losweken" en hebben het vakgebied "non-photorealistic rendering" ontwikkeld.

Bespreek in 350-400 woorden wat deze nieuwe technieken te bieden hebben, wat toepassingsgebieden zijn, en welke uitdagingen er zijn die niet in de traditionele computer graphics te vinden zijn. Bespreek ook hoe interfaces een rol spelen binnen de computer graphics en in het creëren van non-fotorealistische beelden.

Opgave 8. Visualisatie

Visualisatie van grote data sets is een ingewikkeld probleem, en er zijn vele oplossingen voor bedacht. Kies een aantal voorbeelden gegeven op het college, en geef in een essay van 350-400 woorden in grote lijnen aan wat ze doen, en waarom ze voor bepaalde toepassingen geschikt zijn. Neem hierbij tenminste één voorbeeld uit de "scientific visualisation" en één uit de "information visualisation". Geef aan wat deze termen betekenen en hoe de visualisatie technieken verschillen.

Opgave 9. Computer Vision

In het volgende mag je uitgaan van binaire beelden. Ieder onderdeel 1 punt

- 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.
- Attributen kunnen gebruikt worden om componenten te beschrijven. Leg in je eigen woorden uit wat een attribuut-filter doet. Geef een mogelijke toepassing.
- 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 10. Wederzijdse uitsluiting

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 één of enkele alinea's het belang van wederzijdse uitsluiting voor computerprogramma's en computersystemen.
- 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 // critische 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, zoals we die in ons programmeeronderwijs jullie nooit leren (wacht is het label waarnaartoe gesprongen wordt).

b1) Leg uit waarom dit proces niet eerlijk is.

Om oneerlijkheid te voorkomen voert men (bv) 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 ; >
```

b2) 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.