

- 1a Geef argumenten vóór het gebruik van threads bij het ontwerpen van servers in gedistribueerde systemen. Welke nadelen zijn er verbonden aan het toepassen van threads? 5pt
- 1b Als een thread een *blocking system call* aanroept, kan het voorkomen dat het gehele proces waarbinnen die thread geëxecuteerd wordt, blokkeert. Beschrijf twee oplossingen om dit probleem te voorkomen, en geef voor- en nadelen van elke oplossing. 10pt
- 1c Leg het principe uit van *scheduler activations*, en geef aan voor welk probleem ze een oplossing bieden. 5pt
- 2a Leg uit wat *hierarchical failure masking* en *group failure masking* is. Geef van elk een voorbeeld. 5pt
- 2b Leg het verschil uit tussen een *fail-stop* model, en een model dat uitgaat van *Byzantine failures*. Geef in beide gevallen aan hoe groot een zogeheten *k-fault tolerante* server group moet zijn. Licht je antwoord toe. 10pt
- 2c Transacties dienen zogeheten *durable* te zijn. Leg uit wat dat betekent, en hoe dit gerealiseerd kan worden door middel van *stable storage*. 5pt
- 2d Leg uit wat een *rollback* inhoudt bij een transactie en wanneer het toegepast wordt. Idem voor *roll forward*. 5pt
- 3a Leg kort de volgende begrippen uit: *atomic multicast*, *causal message ordering*, *total ordered message delivery*. 5pt
- 3b Een probleem bij *News* is dat je soms reacties eerder ontvangt dan de daarbij behorende vraag. Leg uit hoe *causal message ordering* gebruikt zou kunnen worden om dit probleem op te lossen. Bedenk tevens een veel eenvoudiger oplossing. 5pt
- 3c Leg uit wat *PRAM consistency* betekent en waarom dit model zo eenvoudig te implementeren is. 5pt
- 3d Sequentiële consistentie bij *Distributed Shared Memory* wordt doorgaans op een van twee verschillende niveau's van operaties ondersteund. Geef aan welke niveau's dat zijn, en waarom het onderscheid gehanteerd wordt. 10pt
- 4a In Jade kunnen bestaande filesystemen die toegankelijk zijn over een WAN, gemount worden in een logische *name space*, waarvan er een per gebruiker is. Beschrijf hoe dit principe werkt, en hoe vervolgens files gelezen en geschreven kunnen worden. 10pt
- 4b Als elke gebruiker zijn eigen (*file*) *name space* heeft, dan lijkt het principieel onmogelijk voor twee gebruikers om gemeenschappelijke files te hebben. Leg uit wat de gedachte achter deze redenering is, en hoe dit probleem opgelost kan worden. 5pt
- 4c Verklaar waarom *server caching* voor files in de praktijk doorgaans beperkt succesvol is, en juist *client caching* veel resultaat heeft. Hoe valt dit te verklaren met het succes van zogeheten Web proxies waarbij Web pagina's in een cache opgeslagen worden? 5pt