

Com consulta de *cheat sheet* manuscrita. Duração: 2h00m.

**Recomendação:** As suas respostas devem estar bem estruturadas, ser completas e estar escritas de forma clara. Podemos avaliá-las **unicamente** tendo em conta o que foi escrito, e não o que pensa que podemos inferir.

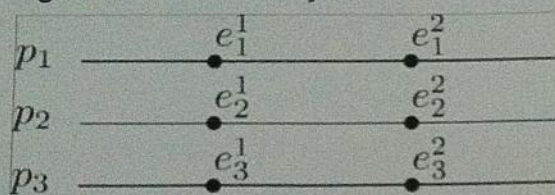
**Nota importante:** Responda às questões 1, 2 e 3 numa folha de exame e às questões 4, 5 e 6 noutra folha de exame.

1. **[RPC, 3val]** Suponha que quer competir com a Google e Microsoft na criação de uma aplicação distribuída de grande dimensão que irá executar na *cloud*. Nas aulas abordou o tema RPC, e está a considerar usar essa técnica.
  - a. Um dos pontos chave de qualquer sistema baseado em RPC é a geração dos *stubs*. Descreva, sucintamente, o processo de geração do *stub*, e como ele facilita o desenvolvimento da aplicação distribuída. Que tipo de problemas são evitados com a automatização deste processo?
  - b. Confiabilidade é uma grande preocupação em sistemas distribuídos. Qual é o modelo de falha RPC? O que pode falhar, e como são reportadas as falhas?
2. **[Segurança, 3val]** A autenticação em Kerberos usa uma cifra simétrica baseada numa chave partilhada entre cada utilizador e o *Key Distribution Center* (a KDC).
  - a. Qual a razão do protocolo Kerberos separar o processo de obtenção de um "ticket" inicial dos "tickets" de comunicação com os serviços?
  - b. Suponha que alguém pode fazer "sniffing" à rede de um sistema que usa o protocolo Kerberos. Será possível obter acesso aos serviços como sendo um utilizador autenticado? Se sim, explique porquê. Se não, explique sucintamente como o protocolo previne este tipo de ataque.
  - c. Supondo que considera que biometria, nomeadamente impressões digitais, será a forma natural de fazer *log in* num sistema, ao invés de usar a tradicional *username/password*. Poderá este tipo de *log in* ser acomodado com uma variação do protocolo Kerberos? Em caso afirmativo, como?
3. **[Sistemas de Ficheiros Distribuídos, 4val]** O *Google File System* (GFS) foi desenvolvido com preocupações de escalabilidade. Na aula teórica abordamos as opções de desenho e motivações no desenvolvimento desse sistema de ficheiros.
  - a. Quais são os principais escolhas em termos de desenho do sistema? Quais as motivações e consequências?
  - b. Descreva como pode o GFS escalar para sistemas com centenas de máquinas (ou mais), apesar de ter um único *master*.
  - c. Como é que o GFS lida com falhas no *master server*?
4. **["End-to-end argument", 3val]** Ilustre o "end-to-end argument" usando um projeto/trabalho que tenha realizado. A sua resposta será avaliada de acordo com o exemplo que der (trabalhos no âmbito de SDIS são mais valorizados) e da subtilidade da sua análise. (**Sugestão:** use o serviço de "backup" como exemplo.)



Com consulta de *cheat sheet* manuscrita. Duração: 2h00m.

5. [Logical Clocks, 4val] Considere o seguinte diagrama temporal onde os círculos representam eventos **não relacionados com a troca de mensagens**. (Nota: Embora alguns eventos estejam alinhados verticalmente, não deve inferir que são simultâneos.)



- a. Acrescente (desenhe) uma mensagem (enviada) por cada processo e os respectivos eventos de recepção/transmissão, de modo a que os seguintes pares de eventos pertençam à relação **happened before** definida por Lamport:  $(e_1^1, e_2^1)$ ,  $(e_1^1, e_3^2)$ ,  $(e_3^2, e_2^2)$ . Justifique a sua resposta.
- b. Tanto as *timestamps* de Lamport como os vetores de *timestamps* foram concebidos para representar a relação happened before. Qual é a vantagem dos vetores de *timestamps*? Ilustre-a usando o diagrama que desenhou na sua resposta à pergunta anterior.
- c. Considere o protocolo de multicast com ordem total que usa *timestamps* de Lamport discutido nas aulas. Este protocolo também garante entrega por ordem causal? Justifique.
6. [Paxos, 3val] - Considere o algoritmo Paxos e o seu uso na implementação de máquinas de estado replicadas.
- a. Qual é a diferença entre um *proposer* e um *distinguished proposer*? Explique a reação para usar um *distinguished proposer*, usando como exemplo uma execução dum configuração com 3 processos. (Sugestão: Desenhe um diagrama temporal com uma troca de mensagens e explique-o.)
- b. Para escolher um *distinguished proposer* é necessário usar um algoritmo de eleição. Explique porque o algoritmo do *bully* de Garcia-Molina não é apropriado para Paxos. (Sugestão: Considere os pressupostos de cada algoritmo.)
- c. Lamport afirma que Paxos pode ser usado na implementação de máquinas de estado replicadas de forma a que o custo de execução dum comando da máquina de estados é essencialmente o custo de execução da 2ª fase de Paxos. Explique porque assim é. (Nota: Uma resposta completa deve mencionar a principal característica de Paxos que permite esta otimização.)