

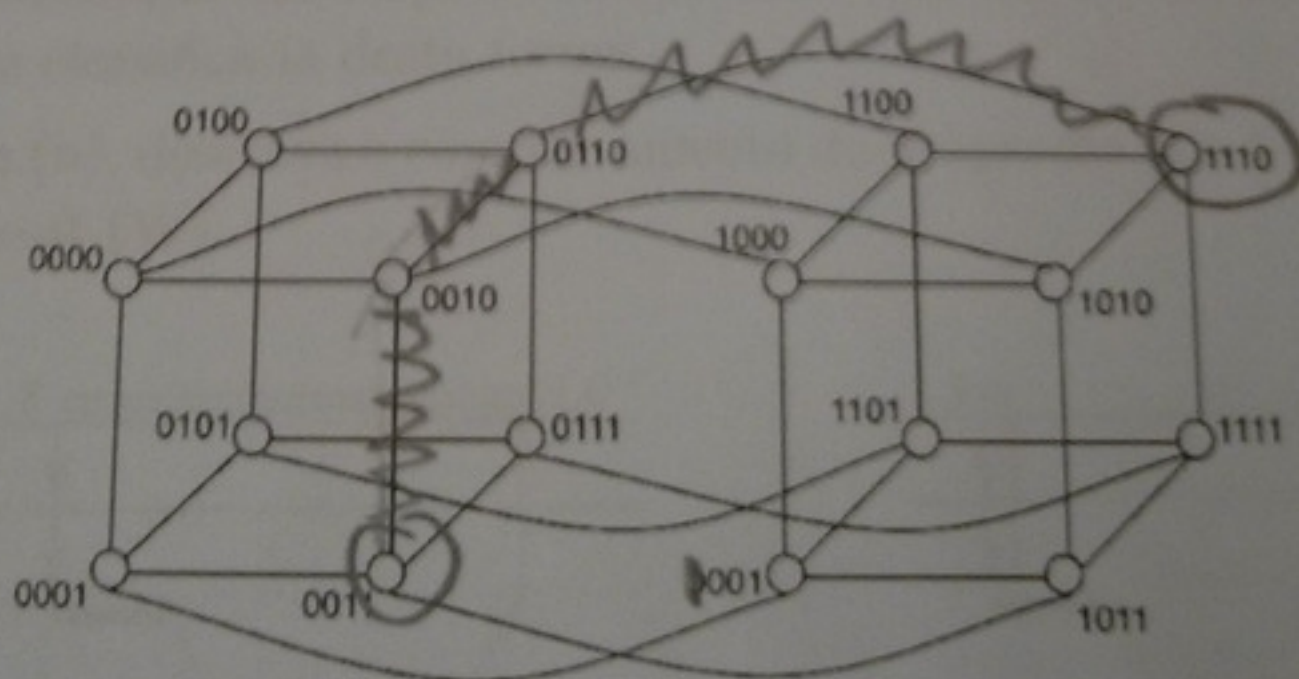
MC714: Sistemas Distribuídos  
Instituto de Computação - UNICAMP  
Prof. Luiz Fernando Bittencourt  
1ª Prova

Nome:

RA:

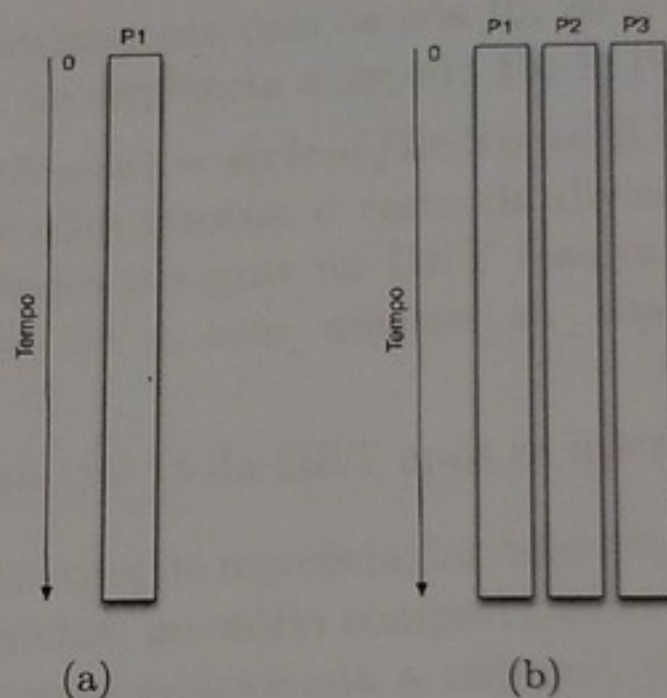
Instruções: A duração da prova é de 120 minutos. Não é permitida consulta a qualquer material. Em caso de fraude, todos os envolvidos receberão nota zero. Você pode fazer a prova a lápis (desde que o resultado final seja legível). Boa prova!

1. (0.5) Qual a importância de se caracterizar/modelar sistemas paralelos e distribuídos?
2. (1.0) Diferencie a busca por um item em uma rede P2P estruturada da busca em uma rede P2P não estruturada. Discuta as duas buscas em termos de escalabilidade.
3. (2.0) Considere um sistema Chord e uma tabela hash distribuída (DHT – *distributed hash table*) cuja função de hash resulta em identificadores inteiros no intervalo  $[0, 31]$ .
  - (a) Suponha uma rede P2P inicialmente com os nós 10, 20 e 30. Descreva como se dá a entrada dos seguintes nós na rede, na sequência a seguir: 15, 1, 19, 31.
  - (b) Seja a função hash  $h(fileName) = strlen(fileName) \% 32$ , onde a função  $strlen(x)$  retorna o tamanho da string  $x$  e  $a \% b$  retorna o resto da divisão inteira de  $a$  por  $b$ . Descreva a inserção dos 7 itens de dados a seguir na DHT resultante após as inserções de nós feitas em (a): *archiv.doc*, *arquivo.txt*, *a*, *este\_arquivo\_eh\_super\_importante.tar.gz*, *a.out*, *a1.out*, *arquivo1.txt*.
  - (c) Descreva a remoção dos nós 1 e 15 da DHT após as inserções de itens de dados feitas em (b).
4. (1.0) Do ponto de vista da arquitetura de memória, há duas abordagens fundamentalmente distintas para o projeto de sistemas paralelos: memória compartilhada e troca de mensagens. Qual das duas arquiteturas está mais diretamente relacionada a sistemas distribuídos? Qual característica dos sistemas distribuídos levou-o a esta resposta?
5. (1.0) A figura abaixo apresenta uma topologia de hipercubo. Descreva, tão detalhadamente quanto possível, um roteamento do nó 0011 para o nó 1110.





6. (0.8) Compare algoritmos centralizados e descentralizados. Que característica dos algoritmos descentralizados os tornam mais escaláveis em relação ao tamanho do sistema?
7. (0.5) Descreva um exemplo onde threads podem ser utilizadas para melhorar o desempenho do lado do cliente, e um exemplo do lado do servidor.
8. (0.5) Qual a diferença básica entre virtualização e abstração? Dê um exemplo.
9. (1.2) Um escalonador *round robin* seleciona processos de uma fila e aloca-os em processadores disponíveis de forma circular, isto é, primeiro processo no primeiro processador, segundo processo no segundo processador, e assim por diante. Ao chegar no último processador, se há mais processos, o próximo da fila é colocado logo após o primeiro processo no primeiro processador, repetindo esse escalonamento de forma circular. Considere os dois sistemas das figuras (a) e (b) abaixo, onde cada processador (P1 a P4) tem a mesma capacidade de processamento. Considere o seguinte conjunto de processos e seus tempos de processamento:  $t_0 = 5$ ,  $t_1 = 10$ ,  $t_2 = 5$ ,  $t_3 = 20$ ,  $t_4 = 2$ ,  $t_5 = 1$ ,  $t_6 = 10$ ,  $t_7 = 15$ ,  $t_8 = 12$ ,  $t_9 = 1$ . Esses processos são independentes (não comunicam-se entre si) e não sofrem preempção. Sejam  $r_1 r_2 r_3 r_4 r_5 r_6$  os dígitos de seu RA. Escalone a fila de processos,  $t_{r_6}, t_{r_5}, t_{r_4}, t_{r_3}, t_{r_2}, t_{r_1}$  nos sistemas (a) e (b) abaixo. Qual o *speedup* de tal conjunto de processos, seguindo a alocação circular definida acima, no sistema (b) em relação ao sistema (a)?



10. (1.5) Considere a execução das primitivas *send* (S) e *receive* (R) da figura abaixo.
- (a) (1.0) Determine, em cada uma das figuras (a) e (b), se a primitiva *send()* (S) é síncrona/assíncrona e bloqueante/não bloqueante, e se a primitiva *receive()* (R) é bloqueante/não bloqueante. Justifique sua resposta descrevendo que características/comportamento de cada primitiva o fez classificá-la desta forma.
- (b) (0.5) Na figura (b), descreva o comportamento do programa nos 4 locais de possíveis chamadas da primitiva *wait* (W).

