Lista de Exercícios

1. Explique por que você em geral *não precisa* e *não deve* se preocupar com o desempenho dos seus programas.
2. O que é a chamada **Lei de Moore**? Indique sua importância para o desenvolvimento da tecnologia de computadores.
3. Usando o gráfico de comparação de desempenho de processadores e memórias apresentado no slide 11 do material de revisão de arquitetura de computadores, faça uma avaliação aproximada de quantas vezes cresceu o desempenho de processadores e de memórias no intervalo entre 1989 e 2000 e compare.
4. Qual a função do **barramento** (*bus*) em computadores? Qual é a principal limitação de barramentos?
5. O que é **hierarquia de memória** e qual sua importância nas arquiteturas atuais?
6. O que é **localidade**? Quais são os dois tipos de localidade? Por que ela é importante para o desempenho de sistemas computacionais?
7. De que forma uma **cache** ajuda a aproveitar a localidade existente em programas?
8. Qual a relação entre um **bloco de memória** e uma **linha de cache**?
9. O que é o *mapeamento direto* e qual sua limitação principal?
10. O que é o *mapeamento associativo* e qual sua limitação principal?
11. De que forma o *mapeamento associativo por conjuntos* resolve as limitações dos mapeamentos direto e associativo?
12. Considere um processador com frequência de operação de 2GHz e um sistema de memória com um nível de cache, sendo que o tempo de acesso à cache é de 2 ciclos, enquanto o tempo de acesso à memória é de 200 ciclos. O *hit ratio* de um programa é a fração média de acessos à memória que são servidas pela cache. Suponha um problema que precisa fazer um total de 100 milhões de acessos à memória. Calcule qual será o tempo gasto com acessos à memória se o *hit ratio* for cada um dos seguintes valores: 0.5, 0.75, 0.8, 0.9, 0.95 e 0.98.
13. O que é **multiprogramação**?
14. O que é um sistemas de **timesharing** e qual sua relação com multiprogramação?
15. Por que o uso de multiprogramação e *timesharing* leva à necessidade de lidar com **gerenciamento de memória**?
16. Explique como é realizado o gerenciamento de memória em um sistema de **memória virtual com paginação**.
17. Explique como um sistema de memória virtual com **paginação** é capaz de eliminar ou reduzir os problemas de *fragmentação de memória*.
18. Explique o conceito de *page fault*.
19. O que é um TLB e por que ele é importante?
20. De que forma um **pipeline de instruções** é capaz de aumentar a taxa de execução de operações pelo computador?
21. Quais são as principais limitações de um pipeline de instruções?
22. Explique o que é **previsão de desvio** e por que isso é importante para um sistema com pipeline de instruções.

23. O que são **dependências de dados** e qual é sua importância para processadores superescalares?

24. O que é **renomeação de registradores** e qual sua relação com execução superescalar e dependência de dados?

25. Por que razão a existência de **instruções vetoriais** em processadores aumenta ainda mais a importância de localidade no código?

26. Boa parte do crescimento de desempenho dos microprocessadores nos últimos anos se deve principalmente ao uso de paralelismo de instruções. Explique o que é paralelismo de instruções e indique por que o crescimento de desempenho baseado exclusivamente nesse tipo de paralelismo chegou ao fim.

27. Indique como o uso de paralelismo explícito pode ajudar a resolver a limitação indicada na resposta da questão anterior.

28. Uma característica importante para um sistema paralelo é sua **escalabilidade**. Explique o que é escalabilidade e por que ela é importante.

29. Compare as arquiteturas paralelas **SIMD** e **MIMD**.

30. O que é **comunicação**? Por que ela é importante em processamento paralelo.

31. Explique a diferença entre **máquinas de espaço de endereçamento compartilhado** e **máquinas de passagem de mensagem**.

32. O que é uma arquitetura **NUMA**? Por que razão arquiteturas atuais de espaço de endereçamento compartilhado são NUMA, e não UMA?

33. O que é uma **rede de interconexão**?

34. Qual a diferença entre redes de interconexão estáticas e dinâmicas?

35. O barramento (*bus*) é uma chave de interconexão. Indique suas vantagens e desvantagens na interligação de sistemas paralelos.

36. Por que razão o uso de chaves de interconexão *crossbar* era adequado para sistemas antigos mas não é adequado para sistemas modernos?

37. Quais as vantagens e desvantagens das chaves multiestágio em relação às chaves *crossbar*? E em relação a barramentos?

38. Compare as vantagens e desvantagens das seguintes topologias de redes estáticas:

    (a) Completamente conectada

    (b) Estrela

    (c) Árvore binária

    (d) Toro (malha com ligações nas bordas) 2D

    (e) Hipercubo

39. O que é **coerência de cache**, e porque isso é importante em sistemas de memória compartilhada?

40. O que é **false sharing** e por que isso pode resultar em prejuizo para o desempenho?

41. O que é o esquema de *snooping* para coerência de cache e porque ele não é utilizado em máquinas paralelas com muitos processadores?

42. O *speedup absoluto* devido a paralelismo em P processadores $S(P)$ é definido como a razão entre o tempo de execução, T_s, do programa sequencial e o tempo de execução, $T_p(P)$, do programa paralelo em P processadores, $S(P) = T_s/T_p(P)$. Suponha que um programa tenha uma quantidade fixa de trabalho a executar. Dessa quantidade, uma fração α (calculada em termos de tempo de execução) é estritamente sequencial (não pode ser paralelizada) enquanto que o restante $1 - \alpha$ pode ser perfeitamente paralelizado (pode ser dividido igualmente entre P processadores, para qualquer valor de P, sem acrescentar custos adicionais). Derive uma fórmula para o *speedup* desse programa em função de α e P. Use essa fórmula para encontrar um limite superior para o *speedup* quando se dispõe de processadores à vontade. (Este resultado é a chamada *Lei de Amdahl*.)

43. Considere um *pipeline* de 10 estágios com tempo de ciclo de $10ns$. Suponha que uma aplicação alterna a execução entre duas fases, numa das quais ela envia m operações consecutivas para serem executadas no *pipeline* e na outra ela não usa o pipeline por um tempo T (dado em ns). Encontre uma expressão para o tempo de execução do programa que executa N dessas fases.

44. Suponha que uma linha de *cache* de 32 bytes deva ser transmitida a outro processador pela rede. Suponha ainda que o tempo de partida da transmissão é de $2\mu s$ e os dados podem ser transmitidos a $100MB/s$. Qual a latência total da operação remota?

45. Suponha uma máquina com tempo de partida de $100\mu s$ e uma largura de banda assintótica de $80MB/s$. Para que tamanho de mensagem a largura de banda efetiva é a metade da largura de banda de pico? (A largura de banda efetiva é determinada pelo tamanho da mensagem dividido pelo tempo tomado em sua transmissão.)