

Arquitectura de Computadores

Licenciatura em Engenharia Informática

Escolha múltipla dos testes 2008/2009

Q-1 [1.00 val.] Uma das características que diferencia as arquitecturas CISC e RISC é:

1. As arquitecturas CISC são load/store
2. As arquitecturas CISC possuem, por norma, mais registos do que as arquitecturas RISC
3. As arquitecturas CISC permitem a manipulação de posições de memória em instruções lógicas e aritméticas
4. As arquitecturas CISC executam aplicações mais complexas do que as RISC

Resposta 3

Q-2 [1.00 val.] Considere os seguintes excertos de código IA-32/NASM, onde **jg** (Jump if Greater) opera sobre números com sinal e **ja** (Jump if Above) opera sobre números sem sinal:

```

mov al, 255          mov al, 255
cmp al, 0            cmp al, 0
jg MaiorQueZero     ja MaiorQueZero
    
```

1. O salto para MaiorQueZero é efectuado em ambos os códigos
2. O salto para MaiorQueZero não é efectuado em ambos dos códigos
3. O salto para MaiorQueZero é apenas efectuado pela execução da instrução **jg**
4. O salto para MaiorQueZero é apenas efectuado pela execução da instrução **ja**

Resposta 4

Q-3 [1.00 val.] Considere as seguintes declarações em C:

```

struct s {
    int a; // inteiro a 4 bytes
    int b; // inteiro a 4 bytes
};
struct s v[100];
    
```

Seleccione a opção certa para carregar o valor de `a[i].b` para o registo `eax`:

1. `mov ebx, [i]`
`mov eax, [v+ebx*4+4]`
2. `mov ebx, [i]`
`mov eax, [v+ebx+4]`
3. `mov ebx, [i]`
`mov eax, [v+ebx*8+4]`
4. `mov ebx, [i]`
`mov eax, [v+ebx*8+8]`

Resposta 3

Q-4 [1.00 val.] Indique qual das seguintes afirmações é verdadeira:

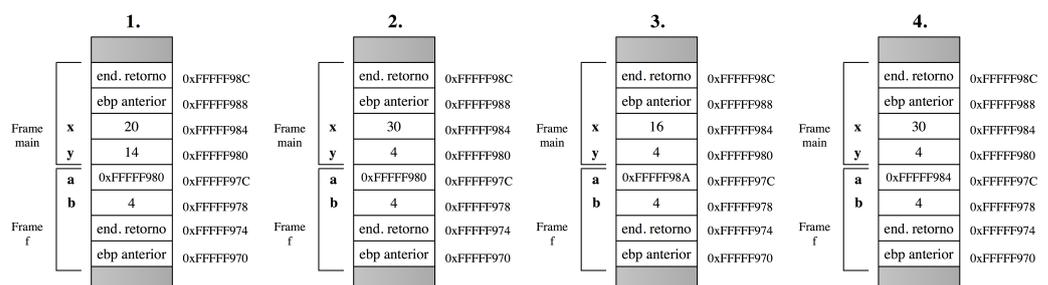
1. Cada arquitectura tem apenas um assembly para a sua programação
2. Cada componente de um computador tem uma ligação (bus) dedicada ao CPU
3. A memória endereçável por um programa é limitada pelo número de bits que compõe um endereço
4. O modelo conceptual da máquina de von Neumann introduziu o conceito de registo

Resposta 3

Q-5 [1.00 val.] Considere o seguinte código C: Indique qual das seguintes configurações da pilha de execução está correcta se pararmos a execução de programa antes da instrução apontada pela seta.

```

void f(int *a, int b) {
    a--;
    *a = b+10;
    return; ←
}
int main() {
    int x = 20, y = 4;
    f(&x, y);
    return 0;
}
    
```



Resposta 1

Q-6 [1.0 val.] Considere a FPU Intel. Qual é a diferença entre as instruções da família F, por exemplo FSTP, e as da família FI, por exemplo, FISTP?

1. Nas segundas está implícita uma conversão de vírgula flutuante para inteiro ou vice-versa
2. As primeiras operam sobre números com sinal e as segundas sobre números sem sinal
3. As segundas permitem o uso de endereçamento imediato
4. Nenhuma

Resposta 1

Q-7 [1.0 val.] A estruturação do motor de execução de um processador num pipeline:

1. Diminui o tempo de execução de cada instrução
2. Aumenta o número de instruções executadas por unidade de tempo
3. Aumenta o número de acessos a memória
4. Diminui o número de acessos a memória

Resposta 2

Q-8 [1.0 val.] As arquitecturas VLIW (Very Large Instruction Word):

1. Delegam a responsabilidade do escalonamento das instruções a executar nos compiladores
2. Reordenam as instruções em tempo de execução para evitar as dependências de dados
3. Dividem as instruções em vários pequenos estágios
4. Não podem executar várias instruções ao mesmo tempo

Resposta 1

Q-9 [1.0 val.] Considere uma arquitectura com endereços de 32 bits e uma cache de mapeamento directo com 1024 linhas, cada um com um bloco de 64 bytes. Divida o endereço 0100 1111 0010 1011 0111 1110 1011 0100 nas suas componentes.

- | | | |
|-------------------------------|----------------------------|----------------------------|
| 1. Deslocamento: 11 0100 | Linha: 01 1111 1010 | Chave: 0100 1111 0010 1011 |
| 2. Deslocamento: 10 1011 0100 | Linha: 01 1111 | Chave: 0100 1111 0010 1011 |
| 3. Deslocamento: 11 0100 | Linha: 1010 1101 1111 1010 | Chave: 01 0011 1100 |
| 4. Deslocamento: 10 1011 0100 | Linha: 1100 1010 1101 1111 | Chave: 01 0011 |

Resposta 1

Q-10 [1.0 val.] No conceito de paginação utilizado na gestão da memória virtual:

1. Cada processo tem associada uma tabela de páginas
2. Cada tabela de página tem associado um TLB (Translation Look-aside Buffer)
3. Todos os processos partilham a mesma tabela de páginas
4. Cada processo tem associado uma TLB (Translation Look-aside Buffer)

Resposta 1

Q-11 [1.0 val.] Numa arquitectura com memory-mapped I/O:

1. Existe um único espaço de endereçamento que engloba a memória e os controladores de periféricos
2. É necessário usar instruções dedicadas (por exemplo **in** e **out**) para interagir com controladores de periféricos
3. Não há interrupções
4. Existem um espaço de endereçamento para cada controlador de periférico

Resposta 1

Q-12 [1.0 val.] A programação de entradas/saídas por interrupções:

1. É menos eficiente do que a espera activa porque usa buffers
2. Consiste na transmissão de blocos de bytes de/para memória sem intervenção do CPU
3. É mais eficiente do que a espera activa porque é executada em modo supervisor
4. Gere pedidos simultâneos a múltiplos periféricos de forma mais eficaz do que a espera activa

Resposta 4

Q-13 [1.0 val.] No contexto da programação de entradas/saídas por interrupções seja R uma subrotina de atendimento de uma interrupção com prioridade i . Caso seja recebida uma interrupção de nível j :

1. A rotina R é sempre interrompida para atender a nova interrupção
2. Se $j < i$ então a rotina R é sempre interrompida para atender a nova interrupção
3. A rotina R só pode ser interrompida se tiver ligado explicitamente as interrupções
4. A rotina R nunca é interrompida para atender a nova interrupção

Resposta 3

Q-14 [1.0 val.] Comparando os volumes de discos RAID 1 e RAID 5.

1. Uma operação de escrita no RAID 5 é mais rápida no que no RAID 1
2. O espaço gasto em redundância no RAID 5 é superior ao gasto no RAID 1
3. Ao contrário do RAID 1, operações de leitura no RAID 5 tornam-se mais rápidas com o aumento do número de discos
4. O RAID 5 tem suporte para tolerância a falhas e o RAID 1 não

Resposta 3

Q-15 [1.0 val.] Nos processadores multi-core:

1. O motor de execução (ALUs, FPUs, etc) é partilhado por todos os cores
2. Cada core tem a sua própria memória central
3. Cada core tem o seu próprio conjunto de registos de uso geral (no IA-32: EAX, EBX, ...)
4. O processador no seu todo segue uma arquitectura SISD (Single Instruction Single Data)

Resposta 3