

Disciplina: Organização de Computadores
GABARITO DA AP2 2019-02

1. (2,0) Considere um sistema onde o número de ciclos de relógio para realizar cada operação de Entrada/Saída (E/S) por interrupção seja igual a 100 e o processador utiliza um relógio de 1 GHz para executar as instruções. Determine o overhead (relação entre ciclos de CPU consumidos por segundo pelo processo de E/S e número de ciclos disponíveis) que ocorre devido a operações de E/S com um mouse que transfere dados para o processador a uma taxa de 50 amostras por segundo, sendo que cada amostra corresponde a um bloco de 2 byte e que cada um desses bytes é processado por uma operação a parte, cada operação correspondendo a uma interrupção distinta.

O mouse transfere dados a uma taxa de 50 amostras por segundo. Em cada amostra, 2 bytes são transferidos. Cada byte é processado em uma operação distinta.

Total de operações = 50 amostras/seg x 2 operações/amostra = 100 operações/seg.

Total de ciclos = 100 operações/seg x 100 ciclos/operação = 10.000 ciclos/seg

O relógio é de 1GHz, que corresponde a 1×10^9 ciclos.

O overhead será obtido pela seguinte razão:

$$10.000 \text{ ciclos/seg} / (1 \times 10^9 \text{ ciclos/seg}) = 0,00001 \text{ ou } 0,001 \%$$

2. (2,0) Considere um computador, cuja representação para ponto fixo e para ponto flutuante utilize 12 bits.

S	Expoente – complemento a 2	b ₀ b ₁ b ₂ b ₃ b ₄ b ₅ b ₆
1	4 bits	7 bits

Na representação para ponto flutuante, o número a ser representado deve ser expresso na notação científica normalizada ($\pm 1, b_0b_1b_2b_3b_4b_5b_6 \times 2^{\text{expoente}}$)

O bit mais à esquerda representa o sinal e deve ser 0 para números positivos e 1 para números negativos. O expoente utiliza a representação em complemento a 2 e todos os expoentes possíveis de serem representados são utilizados para representar números expressos na notação científica normalizada.

- a. Suponha que o conteúdo dos 12 bits seja DE8₁₆ para este conjunto de bits quando considerarmos que ele está representando:

$$(DE8)_{16} = (1101 \ 1110 \ 1000)_2$$

- i. um inteiro em complemento a 2. (0,4 pontos)

$$-2^{11} + (2^{10} + 2^8 + 2^7 + 2^6 + 2^5 + 2^3) = -536$$

- ii. um número inteiro representado em sinal e magnitude. (0,4 pontos)

$$-(2^{10} + 2^8 + 2^7 + 2^6 + 2^5 + 2^3) = -1.512$$

- iii. um número em ponto flutuante utilizando a representação do enunciado. (0,4 pontos)

$$1 \ 1011 \ 1101000$$

Sinal = 1 (negativo)

Expoente = 1011 (complemento a 2) = -5

Mantissa = 1101000

$$-1,1101000 \times 2^{-5} = -(2^{-5} + 2^{-6} + 2^{-7} + 2^{-9}) = -0,5664$$

iv. **(0,4 pontos)** considere o seguinte programa que testa para verificar se um valor depositado em conta é menor que 500, e caso seja, transfere esse valor de uma conta para outra

se ($x < 500$)
 transfira(x para outra conta)

A variável x é armazenada em 12 bits. *Suponha que na primeira linha do programa acima x é tratado como um valor inteiro em complemento a dois, e na segunda linha como um inteiro sem sinal.* Um hacker informa no terminal que depositou -536 reais. Qual a representação em binário do valor inteiro em complemento a 2 informado pelo hacker? O hacker será bem sucedido na tentativa de transferir um valor maior que 500 reais para a conta destino? Ou seja, a condição ($x < 500$) será satisfeita? Caso sim, quanto será transferido de uma conta para outra? [esse ataque é conhecido como ataque do tipo “integer coercion”]

Solução:

⇒ Qual a representação em binário do valor inteiro em complemento a 2 informado pelo hacker?

*Convertendo -536 para binário na representação de complemento a 2,
Ver item (i) desta questão : $-536_{10} = 1101\ 1110\ 1000_2$*

⇒ O hacker será bem sucedido na tentativa de transferir um valor maior que 500 reais para a conta destino? Ou seja, a condição ($x < 500$) será satisfeita?

*A condição será atendida visto que $-536 < 500$.
Sim, o hacker será bem sucedido !*

⇒ Caso sim, quanto será transferido de uma conta para outra?

Será transferido o valor de $1101\ 1110\ 1000_2$ em decimal, que na segunda linha segue a representação de inteiro sem sinal.

O valor a ser transferido em decimal = $2^{11} + 2^{10} + 2^8 + 2^7 + 2^6 + 2^5 + 2^3 = 3.560$

b. Qual o menor e o maior valor positivos (na base 2) na representação em ponto flutuante para números normalizados, descrita no enunciado, para este computador? (0,4 pontos)

Maior valor positivo : $0\ 0111\ 1111111 = 1,1111111 \times 2^{+7} = +11111111 = +255$

Menor valor positivo : $0\ 1000\ 0000000 = 1,0000000 \times 2^{-8} = +0,00391$

3. (2,0) Verdadeiro ou falso? Justifique claramente suas respostas.

a) **(0,5)** Endereçamento direto é muito útil para acessar vetores na memória.

Falso, o mais indicado para acesso à vetores é o modo indexado. Quando este não está disponível, temos como opção o modo indireto.

b) **(0,5)** Arquiteturas paralelas, com várias unidades de processamento, não tem nenhuma utilidade prática, e por isso não são encontradas nos computadores atuais.

Falso, é fato de que a utilização de arquiteturas paralelas leva ao aumento do poder computacional e a redução do tempo de processamento em função da divisão de tarefas entre os

processadores. Ao contrário do afirmado no enunciado, os computadores e dispositivos que tenham processadores (como celulares) já implementam algum tipo de arquitetura paralela.

Anos atrás, para introduzirem processadores mais rápidos, designers projetavam unidades que funcionassem com clocks cada vez maiores. Entretanto, com o passar do tempo, as limitações físicas do semicondutor ficaram em evidência e o aumento acentuado de clock tornou-se inviável. A opção foi de projetar processadores com mais de um núcleo (multicore).

- c) **(0,5)** A compilação é vantajosa em relação à interpretação na medida em que a compilação não envolve a geração de um código objeto.

Falso, ao contrário do que é afirmado, é a interpretação que não envolve a geração de um código objeto.

- d) **(0,5)** Endereçamento indireto envolve dois acessos à memória para operação com o dado.

Verdadeiro.

4. **(2,0)** Explique os processos de compilação e interpretação, descrevendo algumas vantagens e desvantagens de cada um desses métodos de tradução.

A compilação consiste na análise de um programa escrito em linguagem de alto nível (programa fonte) e sua tradução em um programa em linguagem de máquina (programa objeto).

Na interpretação cada comando do código fonte é lido pelo interpretador, convertido em código executável e imediatamente executado antes do próximo comando.

A interpretação tem como vantagem sobre a compilação a capacidade de identificação e indicação de um erro no programa-fonte (incluindo erro da lógica do algoritmo) durante o processo de conversão do fonte para o executável.

A interpretação tem como desvantagem o consumo de memória devido ao fato de o interpretador permanecer na memória durante todo o processo de execução do programa. Na compilação o compilador somente é mantido na memória no processo de compilação e não utilizado durante a execução. Outra desvantagem da interpretação está na necessidade de tradução de partes que sejam executadas diversas vezes, como os loops que são traduzidos em cada passagem. No processo de compilação isto só ocorre uma única vez. Da mesma forma pode ocorrer para o programa inteiro, em caso de diversas execuções, ou seja, a cada execução uma nova interpretação.

5. **(2,0)** Crie um conjunto de instruções de um operando (o outro implícito na instrução é o registrador Acumulador, ACC), definidas em Linguagem Assembly, necessárias para a realização de operações aritméticas e elabore um programa para o cálculo da seguinte equação:

CONJUNTO DE INSTRUÇÕES PARA 1 OPERANDO:

ADD X => (ACC) <- (ACC) + (X)

SUB X => (ACC) <- (ACC) - (X)

MUL X => (ACC) <- (ACC) * (X)

DIV X => (ACC) <- (ACC) / (X)

LOAD X => (ACC) <- (X)

STORE X => (X) <- (ACC)

$$Y = ((A + B) * C) - A$$

LOAD A => (ACC) <- (A)

ADD B => (ACC) <- (ACC) + (B)

MUL C => (ACC) <- (ACC) * (C)

SUB A => (ACC) <- (ACC) - (A)

STORE Y => (Y) <- (ACC)