LISTA 1 - MAB353 Período 2020-2 Remoto
90 pontos
Retorne seu pdf identificado como lista1-nome1-nome2.pdf

O objetivo desta lista é trabalhar manualmente com máscaras, conversão de bases e com a representação em ponto flutuante, tanto em precisão simples e como em precisão dupla. Embora existam maneiras de obter respostas através de programas ou aplicativos, a lista deve ser respondida com todos os passos feitos manualmente, **como se estivesse em prova sem uso de calculadora e/ou PC.** Somente use calculadora onde for explicitamente permitido. Qualquer outra conversão terá que ser feita manualmente, com aritmética simples.

Questão 1) (10 pontos)
Procure descobrir como listar os tamanhos dos caches disponíveis no processador dos computadores dos membros da dupla. Indique os sistemas operacionais, liste os procedimentos utilizados para cada sistema operacional, se diferentes, e apresente as capturas de telas com as informações dos caches. Verifique se há cache específico para instrução e para dados. Terão que ser apresentadas as informações relativas a cada aluno da dupla.

Questão 2) (10 pontos)
Escreva um programa C para imprimir uma máscara (*unsigned mask*) tendo apenas o MSB (bit mais significativo) ligado e todos os outros bits em zero, **sem usar o conhecimento do tamanho em bits da arquitetura**. Liste o programa feito. Numa mesma tela de terminal, compile o programa para 32 bits (opção –m32) e para 64 bits, rode os dois executáveis e capture tudo na mesma tela. A tela terá que mostrar a linha de comando de compilação, execução e saída para cada arquitetura.

Questão 3) (40 pontos)
Queremos justificar a impressão do programa abaixo, trabalhando manualmente com as representações em ponto flutuante:

```
int main (){
    float x = 5.26, y;
    printf("dump de x\n\n");
    double z = 3.1;
    y = x - z;
    printf ("y = 5.26 - 3.1 = %10.13f \n", y);
}
```

Veja que as variáveis x e y são de precisão simples, enquanto z é em precisão dupla. Para os valores envolvidos, as representações serão sempre normalizadas, no formato <s><exp><f>. É preciso mostrar separadamente o cálculo de f e de exp.

a) (5) Converta 0,26 para binário fracionário, mostrando o passo a passo e obtendo uma quantidade de bits de pelo menos 24 bits após a vírgula. Mostre o resultado em hexadecimal. Não pode simplesmente indicar o resultado. É necessário mostrar o processo de conversão, como se estivesse fazendo uma prova sem consulta. Tem que indicar na mão.

b) (5) Obtenha agora a representação normalizada de 5,26 em precisão simples. Obtenha inicialmente f, lembrando de arredondar ao truncar em 23 bits. Obtenha agora exp. Mostre a representação em hexadecimal do float x.

c) (5) Converta 0,1 para binário fracionário, mostrando o passo a passo e obtendo uma quantidade de bits de pelo menos 52 bits após a vírgula. Mostre o resultado em hexadecimal. Não pode simplesmente indicar o resultado. É necessário mostrar o processo de conversão, como se estivesse fazendo uma prova sem consulta. Tem que indicar na mão.

d) (5) Obtenha agora a representação normalizada de 3,1 em precisão dupla. Obtenha inicialmente f, lembrando de arredondar ao truncar em 52 bits. Obtenha agora exp. Mostre a representação em hexadecimal do double z.

e) (10) Como operações com ponto flutuante são realizadas internamente no processador na maior precisão, obtenha a diferença x – z, estendo a mantissa de x para 52 bits e realizando a operação como se estivesse subtraindo dois valores representados em double. Obtenha então a representação desta diferença em double. Lembre-se que para realizar a operação é preciso antes igualar os expoentes e depois subtrair as mantissas resultantes. Após, volte a normalizar o resultado obtido para obter a representação double.

f) (5) Como y é float, precisão simples, mas terá que ser impresso como um double pela rotina printf, trunque a parte fracionária da mantissa obtida acima em 23 bits, arredondando se necessário, e estenda com zeros à direita para completar os 52 bits. Indique em hexadecimal a parte alta (4 bytes superiores) e a parte baixa (4 bytes inferiores) a serem impressos por printf.

g) (5) Sabendo que o compilador imprime assumindo o valor 1073825710 (0x400147AE) para a parte alta e o valor 10737411824 (0x40000000) para a parte baixa do resultado, obtenha a representação binária do resultado (a partir de f e exp). Mostre o valor desta representação com uma soma de frações e simplifique para obter uma razão final, numerador/denominador. Indique esta fração. Com uma calculadora, faça a divisão e obtenha o que o programa irá imprimir, com 13 casas decimais após a vírgula.

Questão 4 (20 pontos)

a) (10) Como representar $0,3 \times 2^{-136}$ em ponto flutuante precisão simples na arquitetura de 32 bits? Mostre passo a passo o desenvolvimento de sua resposta e apresente a representação do resultado final em hexa.

b) (10) Represente agora o mesmo valor em precisão dupla e verifique a importância de se ter um maior número de dígitos significativos.

Questão 5 (10 pontos)

a) (5) Qual a maior magnitude real que pode ser representado com precisão dupla? Dê sua resposta indicando os 64 bits da representação e mostrando o valor em termos de potências de 2.

b) (5) Qual a menor magnitude real que pode ser representado com precisão dupla? Dê sua resposta indicando os 64 bits da representação e mostrando o valor em termos de potências de 2.