



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Organização de Computadores

AD2 1º semestre de 2023

OBS: As questões marcadas como Bônus não valem nota nem ponto extra e não são corrigidas. Entretanto, são importantes preparatórios para a AP e para aprimorar o conhecimento no curso.

- 1) (2,0 pontos) Ao consultar o ChatGPT sobre "qual arquitetura gera código objeto maior? RISC ou CISC?" o ChatGPT (versão de 14 de março de 2023) respondeu o seguinte:

Em geral, a arquitetura CISC (Complex Instruction Set Computing) tende a gerar código objeto maior do que a arquitetura RISC (Reduced Instruction Set Computing). Isso ocorre porque a arquitetura CISC tem um conjunto maior de instruções complexas que podem executar várias operações em uma única instrução, enquanto a arquitetura RISC usa um conjunto menor de instruções simples que executam apenas uma operação básica por vez.

Como resultado, programas escritos para arquiteturas CISC podem exigir mais memória para armazenar o código objeto, além de serem mais complexos e difíceis de otimizar. Por outro lado, a arquitetura RISC é projetada para maximizar o desempenho e a eficiência, e programas escritos para essa arquitetura podem ser mais simples e compactos, o que pode levar a um código objeto menor. No entanto, é importante notar que a eficiência e o desempenho também dependem de outros fatores, como a qualidade do compilador e a implementação específica da arquitetura.

Entretanto, no site de Stanford encontramos o seguinte:

<https://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/riscisc/>

Uma das principais vantagens do CISC é que o compilador precisa fazer muito pouco trabalho para traduzir uma instrução de linguagem de alto nível em código de montagem. Como o comprimento do código é relativamente curto, é necessária muito pouca RAM para armazenar as instruções. O foco é colocar instruções complexas diretamente no hardware.

Existe uma aparente contradição entre a resposta do ChatGPT e a informação provida pelo site de Stanford? Caso exista, como resolver essa contradição? Como é possível que o código objeto gerado pelo CISC ocupe mais RAM (segundo ChatGPT) e, ao mesmo tempo, o comprimento do código seja mais curto (segundo site de Stanford)? A que código se refere o conteúdo do material de Stanford? Código-fonte? Ou código objeto? Explique claramente sua resposta, enfatizando as diferenças e semelhanças entre as arquiteturas CISC e RISC.

2) (3,0 pontos) Considere os seguintes três pseudo-códigos.

Pseudo-código 1:

1. Início
2. Configurar o dispositivo de entrada
3. Iniciar a leitura de dados do dispositivo
4. Aguardar até que a leitura seja concluída
5. Armazenar os dados lidos em um buffer de memória
6. Processar os dados lidos
7. Fim

Pseudo-código 2:

1. Início
2. Configurar o dispositivo de entrada
3. Configurar a rotina de tratamento de interrupção para a entrada do dispositivo
4. Quando a interrupção ocorrer, o processador pausa a execução do programa atual
5. A rotina de tratamento de interrupção é executada para lidar com os dados de entrada
6. A rotina de tratamento de interrupção armazena os dados em um buffer de memória
7. A rotina de tratamento de interrupção retorna o controle para o programa principal
8. Processar os dados lidos
9. Fim

Pseudo-código 3:

1. Início
2. Configurar o controlador para a transferência de dados entre o dispositivo e a memória
3. Configurar o dispositivo de entrada para a transferência de dados
4. Configurar os endereços de origem e destino da transferência de dados
5. Configurar o tamanho da transferência de dados
6. Habilitar o início da transferência de dados
8. O controlador inicia a transferência de dados, sem a intervenção do processador
9. O controlador emite uma interrupção para o processador quando a transferência for concluída
10. Tratar a interrupção do controlador, se necessário
11. Processar os dados transferidos, quando desejado
12. Fim

- a) (1,0 ponto) Qual o nome da abordagem de E/S usada pelo pseudocódigo 1, pelo pseudocódigo 2, e pelo pseudocódigo 3? Qual a vantagem de cada uma delas em comparação com a anterior?
- b) (2,0 pontos) Em cada uma das situações abaixo, indique qual pseudocódigo, ou quais pseudocódigos, são mais aconselháveis, justificando claramente sua resposta em cada caso.
- (i) Imagine uma aplicação lendo dados de um dispositivo de armazenamento em um momento crítico do processamento, em que qualquer interrupção durante a leitura pode causar problemas de sincronização. A aplicação deseja controlar o momento exato da leitura dos dados.
- (ii) Considere uma aplicação que envolve operações de E/S em um ambiente de tempo real, em que a latência do sistema é um fator crítico e que precisa ser minuciosamente controlado. Como no item anterior, não se deseja usar nenhum tipo de interrupções.
- (iii) Considere uma aplicação que necessite de E/S de forma assíncrona, ou seja, sem esperar que a operação termine antes de continuar a execução. Isto ocorre, por exemplo, num laptop, ao ler dados do mouse.
- (iv) Considere uma aplicação de alto desempenho, que precisa realizar transferências de grandes volumes de dados entre dispositivos de E/S e a memória do sistema de forma eficiente e sem sobrecarga de processamento. Isto ocorre, por exemplo, em aplicações de processamento de vídeo e áudio.

3) (2,0 ponto) Divisão de inteiros.

a) (1,0 ponto) Escreva o processo realizado para dividir um número inteiro por dois apenas com o deslocamento de bits. Atenção! Assuma que estamos interessados apenas no quociente, e não no resto, da divisão. Por exemplo: 5 dividido por 2 é igual a 2.

b) (1,0 ponto) O procedimento descrito no item a) se altera caso o número esteja em complemento a dois. Se sim, o que se altera?

4) (2,0 pontos) Considere dois números inteiros A e B representados em complemento a 2, com 8 bits de tamanho cada um. Seja $A = 01011110$ e $B = 10011101$, ambos em binário (complemento a 2). Responda os itens a seguir. O resultado das operações **deve ser indicado em decimal (pode deixar as contas indicadas) e na representação em complemento a 2**. Considere, em todos os itens, que todas as operações são executadas em variáveis de 8 bits. Ou seja, a resposta também será de 8 bits e também estará representada em complemento a 2.

a) (0,5) Indique o resultado de $A + B$.

b) (0,5) Indique o resultado de $B - A$.

c) (0,5) Mostre o processo de divisão do número A por dois até que o resultado seja zero.

d) (0,5) Mostre o processo de divisão do número B por dois até que o resultado seja zero.

5) (1,0) Considere um número em ponto flutuante, representado na notação IEEE-754, com sinal igual a 0 (positivo), mantissa igual a 1100100, em binário, e expoente igual a -3, em decimal. Qual é o valor desse número em decimal?

B1) (bônus) Qual é a classificação de Flynn que descreve os seguintes casos? Para cada caso, responda SISD, SIMD, MISD, MIMD, e ilustre um exemplo de um sistema prático que se enquadre nesta categoria.

a) um sistema com múltiplas unidades de controle e múltiplas unidades de processamento de dados

b) um sistema com múltiplas unidades de controle e uma única unidade de processamento de dados

c) um sistema com uma única unidade de controle e uma única unidade de processamento de dados

d) um sistema com uma única unidade de controle e múltiplas unidades de processamento de dados

B2) (bônus) Assista o seguinte vídeo sobre supercomputadores

<https://www.youtube.com/watch?v=qflyuEPIzvU>

e responda às seguintes questões.

- Explique como a solução do vídeo emprega a ideia de clusters com placas Raspberry Pi.
- Faça uma pesquisa e descreva como o Google emprega clusters em suas aplicações.
- Escreva um pseudocódigo para realizar a soma de duas matrizes quadradas. Esse pseudocódigo poderia se beneficiar de um cluster computacional, para tornar mais rápida sua execução?
- Nem todas as aplicações podem se beneficiar de um cluster, pois são difíceis de se paralelizar. Cite uma delas e explique o porquê de sua dificuldade de paralelização.

(B3) (bônus) Considere a seguinte interação entre um usuário e o ChatGPT.

Usuário	Considere o número 11b, já representado em binário, ou seja, 11 já é o valor de 2 bits com o qual desejamos trabalhar. Como dividir 11 por dois, usando o deslocamento de bits?
ChatGPT, Resposta 1	Se o número 11b já está representado em binário, então podemos dividir por 2 usando o deslocamento de bits fazendo um shift binário para a direita de 1 posição, o que equivale a uma divisão inteira por 2. O número 11 em binário é representado por 1011. Fazendo um shift binário para a direita de 1 posição, obtemos 0101, que é igual a 5 em decimal. O bit mais à direita é descartado e um novo bit mais significativo com valor "0" é adicionado na posição mais significativa. Portanto, para dividir o número 11b por 2 usando o deslocamento de bits, basta fazer um shift binário para a direita de 1 posição, resultando em 0101b (que é 5 em decimal).
Usuário	tem certeza?
ChatGPT, Resposta 2	<p>Peço desculpas, cometi um engano novamente em minha resposta anterior. Se o número 11 está representado em binário com apenas 2 bits, então não é possível fazer um shift binário para a direita de 1 posição, pois isso resultaria em um número com apenas 1 bit, o que não é suficiente para representar o número 5 em decimal.</p> <p>Nesse caso, podemos dividir o número 11 por 2 usando o seguinte procedimento: como o número 11 tem apenas 2 bits, podemos simplesmente dividir por 2 por meio da operação de divisão inteira, que nos dá o quociente 5 e o resto 1. Portanto, dividir o número 11b por 2 resulta em 0101b como quociente e 0001b como resto.</p>

	Peço desculpas novamente por qualquer confusão causada pela minha resposta anterior.
O usuário abre uma nova aba, e pergunta novamente	Considere o número 11b, já representado em binário, ou seja, 11 já é o valor de 2 bits com o qual desejamos trabalhar. Como dividir 11 por dois, usando o deslocamento de bits?
ChatGPT, Resposta 3	<p>Para dividir 11 por 2 usando o deslocamento de bits, podemos simplesmente deslocar todos os bits de 11 uma posição para a direita. Isso é equivalente a uma divisão por 2, porque o deslocamento de bits para a direita por uma posição é o mesmo que a divisão por 2 na base 10.</p> <p>Assim, temos:</p> <p>11b >> 1</p> <p>Onde ">>" representa o deslocamento de bits para a direita. Isso resultará em:</p> <p>5b</p> <p>Isso significa que dividir 11 por 2 na base 10 é o mesmo que deslocar todos os bits de 11 uma posição para a direita na base 2, resultando no número binário 101b, que representa o valor decimal 5.</p>

Alguma das respostas do ChatGPT está correta? Caso sim, qual? O que está errado em cada uma das outras respostas?

Refleta sobre o papel da inteligência artificial em nosso cotidiano, e discorra sobre as possíveis implicações éticas associadas ao uso indevido da inteligência artificial. Caso deseje, acesse o ChatGPT e coloque mais algumas respostas geradas pelo ChatGPT para esta pergunta sobre quanto vale 11 deslocado para a direita de 1 bit, indicando se as respostas do ChatGPT estão certas ou erradas.

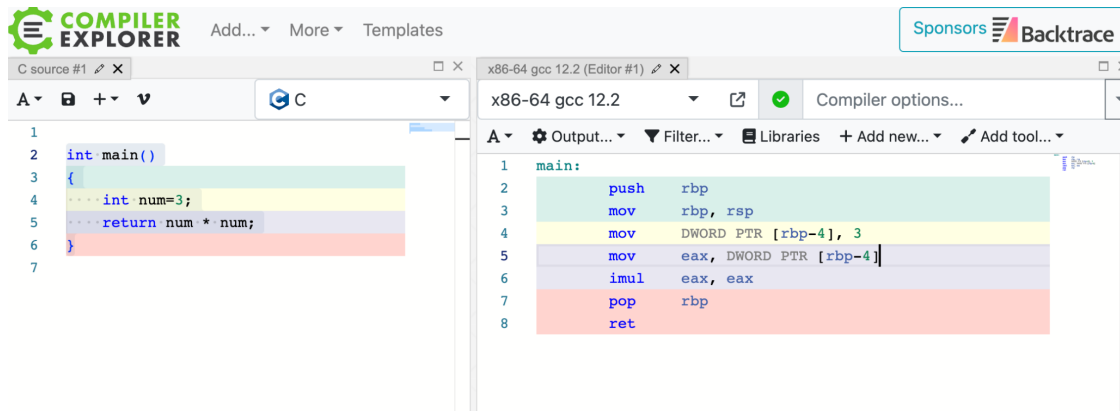
B4) (bônus) O site <https://gcc.godbolt.org/> pode ser usado para ajudar na solução de exercícios envolvendo Assembly. Copie e cole o código abaixo no site mencionado e responda às seguintes perguntas.

```
int main() {
    int num=3;
```

```

        return num * num;
    }

```



- Explique, passo a passo, cada uma das linhas de 4 a 6 em Assembly correspondentes ao código acima.
- Considere uma arquitetura alternativa, com um registrador acumulador ACC, conforme visto em aula, além dos registradores de trabalho padrões, A e B, e que o registrador A seja inicializado com o valor 3. Nessa máquina alternativa, considere o seguinte programa:

```

LD A
MUL A
STR B

```

O que será armazenado no registrador B? Relacione este programa com o programa original em Assembly acima, indicando as semelhanças e diferenças. Em particular, reflita sobre os diferentes modos de endereçamento.

- Substitua o sinal de vezes (*) pelo sinal de adição (+) e estude o código gerado. Explique o que ocorreu. O que ocorre se você trocar agora a operação de `num+num` pela operação `2*num` em C? Alguma coisa muda no Assembly? E se você trocar a operação de `num+num` pela operação `4*num` em C? Alguma coisa diferente ocorre? Explique a nova instrução introduzida no código Assembly.
- Substitua o sinal de vezes (*) pelo sinal de subtração (-) e estude o código gerado. Explique o que ocorreu.