

## [C208] Atividade – Revisão

1- Quais as etapas necessárias à geração de um código executável a partir de um código fonte escrito em linguagem de baixo nível? Apresente-as na sequência.

**Resposta:**

Etapas 1 – O código fonte é preparado para a compilação (pré-processado);

Etapas 2 – Compilador: código fonte é traduzido para código de montagem (Assembly);

Etapas 3 – Montador: código fonte de linguagem de baixo é traduzido em código objeto. Ou seja, o montador traduz símbolos Assembly para respectivos códigos de máquina (op-codes);

Etapas 4 – Ligador: O código objeto é ligado (link-editor) á outros códigos objetos, bibliotecas, recursos do SO para se tornar um código executável;

2- Com relação as arquiteturas RISC e CISC dos processadores, em nível de conjunto de instruções, são feitas as seguintes afirmações:

I. Cada instrução RISC é decodificada sem a necessidade de um microcódigo, ou seja, via hardware.

II. O tempo de execução de uma sequência de instrução é maior na arquitetura RISC uma vez que as instruções são complexas.

Classifique-as como verdadeira ou falsa e justifique todas as suas respostas.

**Resposta:**

I – Afirmação verdadeira.

II – Afirmação falsa. Na arquitetura CISC, o tempo de processamento de execução da instrução é mais demorado, pois a complexidade está na interpretação do opcode (micro-código).

3- Como sabemos, no set de instruções do MIPS há instruções básicas e pseudo-instruções. Esta última não está implementada em hardware. Por este motivo são substituídas por instruções básicas no processo de montagem. Cite um exemplo de pseudo-instrução e sua instrução básica equivalente.

**Resposta:** Pseudo-instrução "li" (load immediate) usada para alocar um valor imediato em algum registrador. A "li" não é uma instrução básica do MIPS e é implementada como uma pseudo-instrução, porque não existe uma instrução única de máquina que faça isso diretamente.

4- Considerando que a=\$t1, b=\$t2, c=\$t3, d=\$t4 e x=\$t5, transcreva o trecho de código em um correspondente em linguagem de alto nível :

```

.text

li $t1, 1
li $t2, 2
li $t3, 3
li $t4, 4
li $t5, 10
blt $t2,$t1, Exit
bgt $t3,$t4,c2
addi $t5, $t5, -1
j Exit
c2: addi $t5,$t5,1
Exit:

```

```

int main() {
    int a = 1;
    int b = 2;
    int c = 3;
    int d = 4;
    int x = 10;

    if( b < a ) {
        goto Exit;
    }
    if( c > d ) {
        x = x + 1;
    }
    else {
        x = x - 1;
    }
    Exit: return 0;
}

```