

**UNIVERSIDADE DO VALE DO ITAJAÍ**

**ESCOLA DO MAR, CIÊNCIA E TECNOLOGIA - EMCT**

**CURSO DE ENGENHARIA DE COMPUTAÇÃO**

**DISCIPLINA DE ARQUITETURA DE COMPUTADORES II**

**Prof. Thiago Felski Pereira**

LUCAS JOSÉ DA CUNHA

LUIZ ALBERTO ZIMMERMANN ZABEL MARTINS PINTO

**Desempenho**

Avaliação 01

12/09/2018

Itajaí – Santa Catarina

**ATIVIDADE**

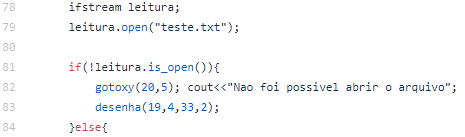
Elaborar um programa que, mediante a entrada de um arquivo de memória de instrução (ROM) em linguagem de máquina (hexadecimal), apresente a quantidade total de ciclos gastos e número de ciclos por instrução (CPI) médio para o programa.

O programa deve conseguir classificar todas as instruções utilizadas nas atividades de programação do MIPS na disciplina de Arquitetura de Computadores I, compreendendo as instruções diretas e as geradas por pseudo-instruções.

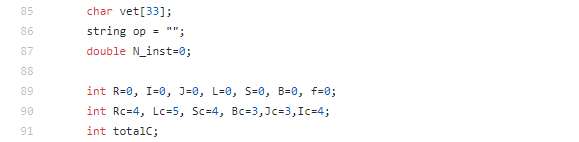
Considere a execução sequencial do programa em um processador MIPS com organização multiciclo. Após o levantamento dos ciclos, apresentar um relatório com classificando as instruções conforme o número de ciclos

**► Etapas**

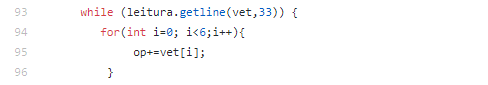
* Após a criação de um tipo “ifstream” para a leitura de arquivo, é feito uma verificação para se o arquivo existe ou não.



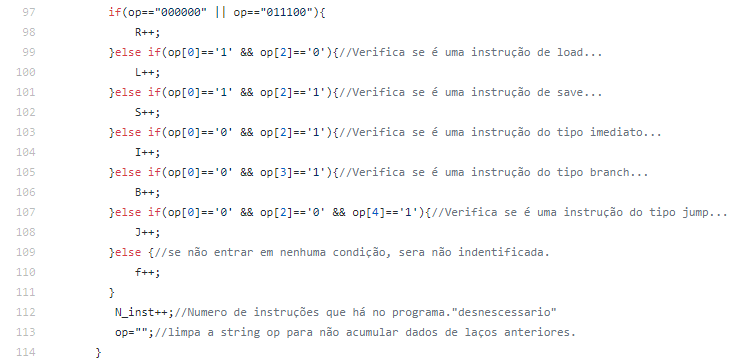
* Criado vetor de char para verificar os bits da instrução, contadores de formatos de instruções, e os ciclos de cada tipo de formato.



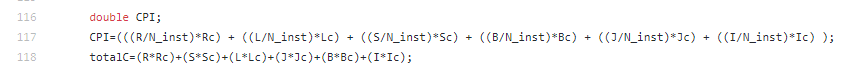
* Fazendo uma leitura de cada linha (que seria cada instrução), foi verificado as 6 primeiras posições para analisar primeiramente o formato da instrução.



* Analisando alguns bits, adiciona-se aos contadores seus específicos formatos.

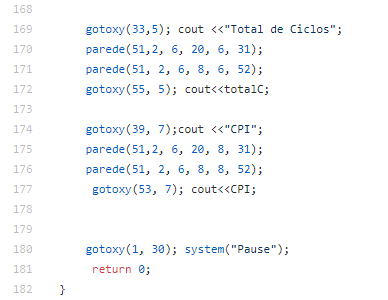


* Tendo os valores dos ciclos, conclui-se com o cálculo de CPI (Ciclos por Instrução).



* Imprime na tela os valores registrados.





**► Exemplo de entrada e saída**

**Entrada:**

**Exercício 1 (.asm) :**

.text # segmento de código (programa)

main:

addi $s1, $zero, 4

addi $s2, $zero, 3

addi $s3, $zero, 2

addi $s4, $zero, 1

add $t0, $s1, $s2 # $t0 = g + h

add $t1, $s3, $s4 # $t1 = i + j

sub $s0, $t0, $t1 # f = $t0 - $t1

**Dump File :**

**00100000000100010000000000000100**

**00100000000100100000000000000011**

**00100000000100110000000000000010**

**00100000000101000000000000000001**

**00000010001100100100000000100000**

**00000010011101000100100000100000**

**00000001000010011000000000100010**

**Saída:**

