

Universidade Federal de Minas Gerais  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação  
Arquitetura de Computadores/Organização de computadores I- 2017/1  
Trabalho a ser realizado em grupos de 3 alunos no máximo  
Data de entrega: 08/05/2017

## Trabalho Prático 1 – Programação Assembly para MIPS

### 1. Descrição Geral

O principal objetivo deste trabalho é implementar um programa que calcula quantos números primos existem em um certo intervalo, e ver como o *pipeline* afeta o seu desempenho. Vamos usar o software DrMIPS (<http://brunonova.github.io/drmips/>) para realizar os nossos testes.

Para calcular a quantidade de números primos em um intervalo, iremos empregar a propriedade que um número é primo somente se ele somente é divisível por 1 e por ele mesmo. Assim, para testar se um número N é primo, iremos dividi-lo por todos os números primos de 1 até N-1. O intervalo pode ser grande, de forma que o seu programa deverá armazenar os números primos encontrados na memória, e não no banco de registradores.

Considere que o “main” do seu programa irá receber dois valores de limites, definidos em duas posições quaisquer de memória no seu código (os valores são somente um exemplo):

```
.data  
min: .word 10  
max: .word 550
```

Ao final da execução, o resultado (o número final de primos no intervalo) deverá estar armazenado no registrador \$v0 da máquina.

### 2. Comparação –máquina uniciclo e máquina pipeline

O trabalho ainda consiste desenvolver este código otimizado para duas máquinas diferentes, sendo uma uniciclo e outra multiciclo com *pipeline*. Assim, após terminar a versão do código para uma máquina de ciclo único, vocês deverão reescrever o código para aproveitar do paralelismo disponível em uma máquina multiciclo. A documentação deve descrever sucintamente o que foi mudado de um código para outro e porque.

Finalmente, a documentação do trabalho prático deve contar uma comparação de desempenho entre as duas versões da máquina. Usem as próprias ferramentas do DrMips para medir o tempo de execução e a CPI do código original e o otimizado quando executando na máquina com *pipeline*.

### 3. Informações Importantes

- O trabalho deve ser feito em duplas ou trios, podendo ser discutido entre os colegas desde que não haja cópia ou compartilhamento do código fonte.
- A data de entrega será especificada através de uma tarefa no Moodle.
- Os trabalhos poderão ser entregues até às 23:55 do dia especificado para a entrega. O horário de entrega deve respeitar o relógio do sistema Moodle. Haverá uma tolerância de 5 minutos de atraso, de forma que os alunos podem fazer a entrega até às 0:00. A partir desse horário, os trabalhos já estarão sujeitos a penalidades. A fórmula para desconto por atraso na entrega do trabalho prático é:

$$\text{Desconto} = 2^d / 0.32 \%$$

onde  $d$  é o atraso em dias úteis. Note que após 5 dias úteis, o trabalho não pode ser mais entregue.

- Deverá ser entregue o código fonte para as duas versões do código: otimizado para CPU sem *pipeline* e otimizado para CPU com *pipeline*.
- Além disso, deverá ser entregue uma pequena documentação contendo todas as decisões de projeto que foram tomadas durante a implementação, sobre aspectos não contemplados na especificação, assim como uma justificativa para essas decisões. Esse documento não precisa ser extenso (no máximo de 6 páginas), mas deve conter alguns testes de desempenho de ambas as versões do programa, bem como uma análise sucinta. A documentação deve indicar o nome dos alunos integrantes do grupo.
- Todas as dúvidas referentes ao trabalho serão esclarecidas por meio do fórum disponível no ambiente Moodle da disciplina.
- A entrega do trabalho deverá ser realizada pelo Moodle, na tarefa criada especificamente para tal. A entrega deverá ser feita no seguinte formato:
  - O trabalho a ser entregue deverá estar contido em um único arquivo compactado, em formato “.zip”, com o nome no formato “*tp1\_aluno1\_aluno2\_aluno3.zip*”
  - **Atenção:** Trabalhos que descumprirem os padrões definidos acima serão penalizados.