

**INSTITUTO DE MATEMÁTICA E ESTATÍSTICA**  
**UNIVERSIDADE DE SÃO PAULO**  
**Computação Paralela e Distribuída**  
**MAC 5742/MAC 0219 2018-1**  
**Prof. Alfredo Goldman**

Mini Exercício Programa: **Utilização das Memórias Caches**

Descrição

Os Processadores atuais têm diferentes níveis de caches. Tais níveis ajudam a incrementar a largura de banda da comunicação com a memória principal do computador (RAM), reduzindo os efeitos do *bottleneck de Von Neumann*.

Neste mini Exercício Programa, cada estudante deverá mostrar o aproveitamento dos níveis de cache em um programa computacional. Esse programa pode ser codificado em qualquer linguagem de programação, mas recomenda-se fortemente qualquer versão do C, C++, ou Fortran<sup>1</sup>.

Entrega:

Deverá ser entregue um pacote no sistema PACA uma pasta com nome do estudante e o sobrenome no seguinte formato: **nome.sobrenome**. Essa pasta deve conter dois itens:

1. Código fonte do programa, em conjunto com um `Makefile` que o compila e o executa.
2. Um relatório em `.txt` explicando no máximo 500 palavras o programa escolhido e as diferenças entre a versão normal e a versão otimizada para aproveitar os níveis de caches.

Na execução dos arquivos `make`, a saída deve mostrar as diferenças o tempo de execução do programa com e sem aproveitamento dos níveis de caches e o razão de melhora obtida com esta otimização.

Em caso de dúvidas, use o fórum de discussão do Paca!

A data de entrega deste mini exercício programa é **até a sexta 16 de Março**.

---

<sup>1</sup> Essas linguagens geram código nativo de máquina e assim fica mais fácil entender os possíveis resultados.

**INSTITUTO DE MATEMÁTICA E ESTATÍSTICA**  
**UNIVERSIDADE DE SÃO PAULO**  
**Computação Paralela e Distribuída**  
**MAC (0219- 5742)/2018-1**  
**Prof. Alfredo Goldman**

Desafio de Programação: **Predição de Branches**

**Descrição**

O *pipeline* permite um uso mais eficiente de diferentes partes de um processador para um melhor desempenho das aplicações. Entretanto, quando o processador encontra uma operação do tipo *if-else* (*branches* ou desvios), este *pipepine* é quebrado pois não há como ele saber de antemão qual instrução será executada em seguida.

Para amenizar as perdas de desempenho ao avaliar um *if-else*, os projetistas de hardware elaboraram um mecanismo chamado Predição de Branches, onde o processador tenta, de alguma forma, “adivinhar”<sup>2</sup> qual *branch* o fluxo de execução irá seguir ao encontrar uma instrução de tal tipo.

Nesse mini Exercício Programa cada estudante deverá evidenciar o aproveitamento dos unidades de predição de *branch* em um programa computacional. Esse programa pode ser codificado em qualquer linguagem de programação, mas recomenda-se fortemente qualquer versão do C, C++, ou Fortran.

**Entrega**

Deverá ser entregue um pacote no sistema PACA uma pasta com nome do estudante e o sobrenome no seguinte formato: **nome.sobrenome**. Essa pasta deve conter dois itens:

1. Código fonte do programa, em conjunto com um Makefile que o compila e o executa.
2. Um relatório em .txt explicando no máximo 500 palavras o programa escolhido e como ele aproveita as unidades de predição de branches.

Na execução dos arquivos Make, a saída deve mostrar o tempo de execução do programa com aproveitamento das unidades de predição de branch e o tempo de execução de outras possíveis versões da execução.

Em caso de dúvidas, use o fórum de discussão do Paca!

A data de entrega deste mini exercício programa é **até a sexta 16 de Março**.

---

<sup>2</sup> Conforme visto na aula, há duas técnicas para se "prever" a direção, uma é a execução simultânea e a outra a escolha de uma delas.