

# TRABALHO 0: PROGRAMAÇÃO PARALELA VIA BMF DRAFT 0

RAUL H.C. LOPES

## 1. PRELIMINARES

Neste trabalho você implementará um conjunto de algoritmos que podem que têm o fato de que podem ser facilmente derivados usando os formalismos de *Bird-Meertens* e *powerlists*. O trabalho consiste de exercícios sobre ordenação paralela e exercícios de cálculo de seqüências.

Algumas dicas importantes sobre a execução do trabalho seguem.

- (1) Não deixe de ler este documento por completo antes de iniciar o trabalho.
- (2) Siga estritamente as especificações deste documento: qualquer desvio delas pode significar a anulação de um exercício ou de todo o trabalho.
- (3) Comece a trabalhar de imediato: o trabalho foi concebido para ser realizado em quatro semanas. Depois disso, você terá outro trabalho e nova prova.

O trabalho pode ser executado em grupos, mas, como estabelecido na seção 4.1, grupos maiores recebem menos créditos.

## 2. CÁLCULO DE SEQÜÊNCIAS

**Exercício 1.** *Implemente o algoritmo para cálculo de todos os primos menores do que um inteiro, apresentado em sala no dia 14/05/06.*

**Exercício 2.** *Apresente testes sobre desempenho, escalabilidade e eficiência da sua solução.*

**Exercício 3.** *Implemente o algoritmo de Reid-Miller, veja em *List Ranking and List Scan on the CRAY C-90*, para cálculo dos prefixos de uma lista encadeada de inteiros.*

**Exercício 4.** *Implemente o algoritmo de Helman-Jájá, veja em *Prefix Computations on Symmetric Multiprocessors*, para cálculo dos prefixos de uma lista encadeada de inteiros.*

Fator	Multiplicador
Grupos de 1 aluno	110/100
Grupos de 2 alunos	100/100
Grupos de 3 alunos	75/100
Grupos de 4 alunos	60/100
Grupos de mais 4 alunos	0
Plágio	0

TABELA 1. Tabela de multiplicadores

**Exercício 5.** *Apresente testes sobre desempenho, escalabilidade e eficiência da suas soluções para os ex. 3 e ex. 4.*

### 3. ORDENAÇÃO

**Exercício 6.** *Implemente o algoritmo de radix sort paralelo apresentado em sala para ordenar inteiros positivos.*

**Exercício 7.** *Implemente uma versão paralela não recursiva do algoritmo bitonic sort.*

**Exercício 8.** *Implemente uma versão não recursiva do algoritmo de merge sort paralelo de Cole, veja em Parallel merge sort.*

**Exercício 9.** *Apresente testes sobre desempenho, escalabilidade e eficiência os programas de ordenação acima descritos.*

### 4. CRÉDITOS

A definição dos critérios de avaliação A atribuição de nota para o trabalho ocorrerá em duas fases:

- (1) Atribuição de créditos pelo trabalho submetido, de acordo com critérios apresentados na sc. 4.1.
- (2) Definição do multiplicador de acordo com a tabela 1.

**4.1. Definição do número de créditos.** Veja neste documento no dia 11/06.

**4.2. Definição de multiplicador.** A nota de cada trabalho é dada pelo número de créditos obtidos pelo trabalho multiplicado pelo fator da tabela 1.

Em caso de plágio todos os trabalhos de grupos envolvidos são anulados. Poderá ser enquadrado como plágio qualquer trabalho em que um mais alunos do grupo não tenham conhecimento de qualquer item

do trabalho. Esse conhecimento deverá ser demonstrado em prova e/ou entrevista.

## 5. A SUBMISSÃO PARA CORREÇÃO

Submeta sua trabalho por e-mail entre os dias 06/06/05 e/ou 13/07/05. Trabalhos submetidos fora dessa data serão considerados nulos.

A mensagem de submissão do seu trabalho deverá conter:

- **Subject:** **pp.tp0**
- **Attachment:** **tp0.tar.bz2**

O corpo da mensagem será desprezado. O arquivo anexo deverá ser obrigatoriamente denominado

**tp0.tar.bz2**

e conterá todos os fontes necessários para compilar seu trabalho prático, incluindo fontes em **L<sup>A</sup>T<sub>E</sub>X**, **Haskell**, **Makefile** arquivo de identificação.

As seguintes regras devem ser rigorosamente seguidas:

- Nenhum compilado deve ser enviado.
- O *tarball* do seu trabalho conterá a seguinte estrutura:

- arquivo de **id**

Conterá uma linha inicial com *e-mail* de contato do grupo e, depois, uma linha de identificação para elemento do grupo, contendo número de matrícula e nome completo separados por ':' (dois pontos.) Por exemplo:

```
jolero@gmail.com
99900089:Ze' do lero
```

- arquivo **Makefile**

- diretório **doc**

Conterá os fontes de toda a documentação do trabalho.

- seu *tarball* será aberto e compilado com as seguintes linhas:

```
tar -jxf tp0.tar.gz
make all
```