

# Guido Araújo

Início

## Introdução à Programação Paralela (MC970/MO644)

<h3>Notícias</h3> <p><b>2/3 Seja bem-vindo(a)!!</b></p>	<h3>Agradecimentos</h3> <p><b>Este Ano teremos Cloud!</b></p> <p>Empresas que apoiaram o curso com acesso às suas respectivas nuvens (AWS e Azure).</p>  <p>The image shows the logos for Microsoft and Amazon. The Microsoft logo is a 2x2 grid of colored squares (red, green, blue, yellow) with the word 'Microsoft' below it. The Amazon logo is the word 'amazon' in a lowercase, sans-serif font with a curved arrow underneath it.</p>
---	--

## Ementa

Neste curso nós iremos explorar técnicas de programação paralela e estudar algumas metodologias de programação para Multicore, GPUs e Clouds: OpenMP, PThreads, CUDA, MapReduce e Spark (AWS e Azure). O curso terá um enfoque prático, cujo objetivo é fazer o aluno entender os conceitos estudados através de uma série de (pequenas) tarefas de programação. Ao final do curso espera-se que o aluno tenha uma boa noção das técnicas utilizadas e dos ambientes de programação comumente empregados nesta área.

## Horários

Aulas: Ter. e Qui.

Sala: 352 (IC3.5)

Horários: 8:00 - 10:00

Instrutor: Prof. Guido Araújo

Atendimento: agende via e-mail.

## Monitor

Maicol Zegarra e Hervé Yviquel

Atendimento: Ter. e Qui. após as aulas.

## Bibliografia

O curso será baseado nos livros abaixo. Adotaremos um livro-texto e todas as aulas, slides e exercícios serão baseados neste.

An Introduction to Parallel Programming by Peter S. Pacheco. Morgan Kauffman. [[livro-texto](#)]

Programming Massively Parallel Processors: A Hands-on Approach, David B. Kirk and Wen-mei W. Hwu. Morgan Kauffman. [[curso on-line](#)]

The Art of Multiprocessor Programming, Revised Reprint, Maurice Herlihy and Nir Shavit. Morgan Kauffman.

## Slides

As aulas utilizarão um conjunto de slides disponíveis no meu [Slides](#) ao lado. Se necessário, notas de aula adicionais, bem como artigos discutidos em classe serão disponibilizados.

## Dúvidas Exercícios

Usaremos uma ferramenta adequada para esclarecimento de dúvidas e solução de exercícios.

## Grupo

O grupo desta disciplina pode ser encontrado aqui.

## Avaliação

Esta disciplina utilizará uma avaliação, majoritariamente, baseada na implementação de programas. O objetivo é fazer com que o conhecimento do aluno seja solidificado através de pequenos exemplos práticos. As atividades desta disciplina serão compostas de tarefas de programação (Tis), projeto (P) e exame (E), sendo que a média final será calculada da seguinte forma:

(a) **MC970**: Para alunos de graduação, a média final será calculada como  $M = 0.7 * \text{Média (Ti)} + 0.3 * P$ , onde Média (Ti) é a média das notas das tarefas Ti ( $i = 1 \dots 10$ ) e P a nota do projeto. Exame Final (F) será aplicado se  $M < 5.0$ , e a média final após o exame final computada como  $MF = (M + F)/2$ . O aluno com frequência  $< 75\%$  não terá direito ao Exame Final.

(b) **MO644**: para alunos de pós-graduação, a média final será calculada como  $M = 0,6 * \text{Média (Ti)} + 0.4 * P$ , onde Média (Ti) é a média das notas das tarefas Ti ( $i = 1 \dots 10$ ), e P a nota do projeto. O conceito final será atribuído de acordo com as faixas abaixo, normalizada em relação à turma.

A:  $M \geq 8.5$

B:  $7 \leq M < 8.5$

C:  $5.0 \leq M < 7.0$

D:  $M < 5.0$

E: se frequência  $< 75\%$

(c) Alunos matriculados em ambas as disciplinas serão avaliados em MC970 pelo critério de (a) e em MO644 pelo critério de (b).

(d) As datas de entrega de todas as tarefas de do projeto estão marcadas no calendário abaixo em azul.

(e) As notas das disciplinas estarão aqui.

## Tarefas

Ao longo do curso serão solicitadas tarefas práticas de programação. Os exercícios deverão ser entregues, através do sistema Susy, nos prazos especificado no calendário abaixo. A página com as tarefas e instruções sobre a submissão das tarefas podem ser encontradas [aqui](#). A submissão de tarefas pode ser feita [aqui](#).

## Projetos

O projeto da disciplina deverá ser especificado na seguinte tabela que deve incluir o RA e nome de participantes (no máximo dois por projeto), e uma pequena descrição de um parágrafo que deve estar disponível para discussão em sala em dia a ser definido com o professor e monitor. O prazo para submissão desta descrição é às 23:59:59 hrs. de Qua. **09/05/2016**. A submissão do projeto deve seguir as instruções [aqui](#). A submissão do projeto será feita aqui, sendo que no dia da apresentação será entregue um relatório neste [formato](#). Exemplos de projetos anteriormente realizados nesta disciplina podem ser encontrados [aqui](#).

## Cronograma

2/3 Visão geral, Cap. 5 - Introdução à OpenMP

7/3 Cap. 5 - OpenMP: cláusula schedule e diretivas static, dynamic, runtime

9/3 Cap. 5 - OpenMP: critical, atomic, lock (T1)

14/3 Perfilamento - Intel VTune e Perf

15/3 Paralelismo de tarefas - task graph, dependencias e runtime (T2)

16/3 Cap. 4 - Introdução Pthreads

21/3 Cap.4 - Pthreads: Seção crítica (T3)

23/3 Cap.4 - Pthreads: Mutexes e semáforos

28/3 CUDA - Basics (T4)

30/3 CUDA - Arquitetura e Modelo de Programação

4/4 CUDA - Histograma e operações atômicas (T5)

6/4 CUDA - Tiled matrix multiplication

11/4 CUDA - Divergência de controle e coalescing (T6)

13/4 Feriado

18/4 Paralelismo usando Map-Reduce (Spark)

20/4 Programando Amazon AWS

25/4 Programando Microsoft Azure

27/4 Modelo OpenMP para aceleradores (T7)

2/5 Aceleração via Cloud usando OmpCloud

4/5 Aceleração via OpenCL e FPGA

9/5 Discussão de propostas dos grupos I e II (T8)

11/5 Discussão de propostas dos grupos III e IV

16/5 Cap. 2 - Hardware e software paralelo (T9)

18/5 Arquiteturas multicore e coerência

23/5 Cap. 2 - Hardware e software paralelo (T10)

25/5 Arquiteturas multicore e coerência

30/5 Modelos de sincronização

01/6 Acompanhamento do grupo I

06/6 Acompanhamento do grupo II

08/6 Acompanhamento do grupo III

13/6 Acompanhamento do grupo IV

15/6 Feriado

20/6 Apresentação do grupo I

22/6 Apresentação do grupo II

27/6 apresentação do grupo III

29/6 Apresentação do grupo IV

11/7 Exame Final (MC970)

---