

Plano de Ensino – Atividades Remotas Emergenciais – 2021/1

Código	DCC831
Disciplina	Programação Paralela
Turma	PG5
Professor	George Luiz Medeiros Teodoro

Ementa. Noções de computação paralela. Taxonomia de arquiteturas paralelas. Técnicas de paralelização e otimização. Computação paralela em ambientes de memória compartilhada, distribuída e aceleradores: principais arquiteturas, linguagens de programação e algoritmos. Modelagem de sistemas paralelos e análise de desempenho.

Programa

Aula	Data	Conteúdo previsto	Modalidade	Interação
1	17/05	Apresentação da disciplina.	Síncrona	Reunião virtual
2	19/05	Visão de Geral (Capítulos 1).	Assíncrona	Atendimento remoto
3	24/05	Conceitos de Hardware/Software Paralelos (Capítulo 2).	Assíncrona	Atendimento remoto
4	26/05	Conceitos de Hardware/Software Paralelos (Capítulo 2).	Assíncrona	Atendimento remoto
5	31/05	Programação para Memória Compartilha Utilizando Pthreads (Capítulo 4).	Assíncrona	Atendimento remoto
6	02/06	Programação para Memória Compartilha Utilizando Pthreads (Capítulo 4).	Assíncrona	Atendimento remoto
7	07/06	Discussão sobre Projeto Final.	Síncrona	Reunião virtual
8	09/06	Programação para Memória Compartilhada em OpenMP (Capítulo 5). Entrega da proposta de projeto (2 páginas).	Assíncrona	Atendimento remoto
9	14/06	Programação para Memória Compartilhada em OpenMP (Capítulo 5).	Assíncrona	Atendimento remoto
10	16/06	Otimizações para Melhoria de Localidade de Acesso à Memória. Entrega do EP 1.	Assíncrona	Atendimento remoto
11	21/06	Programação básica em MPI (Capítulo 3).	Assíncrona	Atendimento remoto
12	23/06	Comunicação coletiva em MPI, Datatypes e Avaliação de Desempenho em MPI (Capítulo 3).	Assíncrona	Atendimento remoto
13	28/06	Algoritmos de ordenação em MPI (Capítulo 3).	Assíncrona	Reunião virtual
14	30/06	Acompanhamento dos Projetos.	Síncrona	Reunião virtual
15	05/07	Acompanhamento dos Projetos.	Síncrona	Atendimento remoto
16	07/07	Projeto de Programas Paralelos (Capítulo 6).	Assíncrona	Atendimento remoto

17	12/07	Projeto de Programas Paralelos (Capítulo 6).	Assíncrona	Atendimento remoto
18	14/07	Introdução a programação em GPUs. Entrega do EP 2.	Assíncrona	Atendimento remoto
19	19/07	CUDA: modelo de execução, hierarquia de memória etc.	Assíncrona	Atendimento remoto
20	21/07	Seminários.	Síncrona	Reunião virtual
21	26/07	Seminários.	Síncrona	Reunião virtual
22	28/07	Seminários.	Síncrona	Reunião virtual
23	02/08	OpenCL	Assíncrona	Atendimento remoto
24	04/08	Intel Xeon Phi.	Assíncrona	Atendimento remoto
25	09/08	Memórias Transacionais.	Assíncrona	Atendimento remoto
26	11/08	MapReduce e Apache Spark.	Assíncrona	Atendimento remoto
27	16/08	Apresentação dos Projetos.	Síncrona	Reunião virtual
28	18/08	Apresentação dos Projetos.	Síncrona	Reunião virtual
29	23/08	Apresentação dos Projetos.	Síncrona	Reunião virtual
30	25/08	Apresentação dos Projetos.	Síncrona	Reunião virtual

Bibliografia

- An Introduction to Parallel Programming (First Edition). Peter S. Pacheco. Morgan Kaufman, 2011. [livro- texto].
- Programming Massively Parallel Processors (Second Edition) : A Hands-on Approach, David B. Kirk and Wen-mei W. Hwu. Morgan Kaufman, 2012.
- The Art of Multiprocessor Programming, Revised Reprint, Maurice Herlihy and Nir Shavit. Morgan Kaufman, 2012.
- Patterns for Parallel Programming (First Edition). Timothy G. Mattson and Beverly A. Sanders. Addison- Wesley Professional, 2004

Material de apoio

Slides e vídeo-aulas no moodle/minhaUFMG, além de referências para vídeos externos com aulas e exposições complementares, exercícios e materiais.

C - AVALIAÇÃO

A menção final será baseada nos exercícios de programação (EP), apresentação de seminário (S) e projeto final (PF). A nota final será calculada como:

$$NF = 0,3*EP + 0,2*S + 0,5*PF.$$

Os **exercícios de programação (EP)** não entregues na data combinada terão um

decrécimo de 50% por dia de atraso. Instruções para compilação e execução devem acompanhar os códigos, assim como um **relatório** descrevendo as técnicas de programação empregadas e avaliação experimental. A intenção dos trabalhos é exercitar os conceitos aprendidos em sala de aula. Todos os programas devem ser executados em plataforma Linux e utilizar bibliotecas padrão.

Os **seminários (S)** devem apresentar artigos sobre linguagens, frameworks ou técnicas de programação paralela. Artigos não apresentados nos dias agendados receberão nota zero. Além disso, a nota será baseada na apresentação, uma vez que o mesmo não será acompanhado de um relatório.

O **projeto final (PF)** consiste da implementação de uma aplicação não trivial, preferencialmente utilizando dois dos modelos de programação paralela. Os alunos devem procurar problemas que demandam paralelismo nas suas áreas de interesse. A nota do PF é dividida em proposta e acompanhamento (0,1) e relatório final (0,4).

- **Proposta:** deve se entregue uma proposta de tópico/problema, descrevendo o problema a ser implementado, sua relevância, as dificuldades do ponto de vista de paralelização. Essa proposta deverá ter até 2 páginas.
- **O acompanhamento:** corresponde a apresentação em sala de aula apresentado o problema, passos já concluídos, as dificuldades e o cronograma para o restante do trabalho.
- **Relatório final e apresentação:** deve ser entregue um relatório (formato SBC) com até 8 páginas e uma apresentação em sala de aula.