

CURSO:	Engenharias	SEMESTRE/ANO:	01/2020
DISCIPLINA:	Programação p/ Sistemas Paralelos e Distribuídos	CÓDIGO:	206610
CARGA HORÁRIA:	60 horas	CRÉDITOS:	04
PROFESSOR:	Luiz Augusto F. Laranjeira	TURMA:	A

PLANO DE ENSINO

19 de Agosto de 2020

1 Objetivos da Disciplina

Capacitar os alunos nos conteúdos relacionados à programação de sistemas computacionais paralelos (arquitetura de memória compartilhada - shared-memory) e distribuídos (arquitetura de troca de mensagens - message-passing), incluindo computadores stand-alone de um ou mais núcleos, clusters de computadores conectados em rede, GPUs e supercomputadores.

2 Ementa do Programa

- | | |
|--|---|
| <ul style="list-style-type: none">i. Conceitos Básicosii. Modelos de Computação Paralelaiii. Multithreaded Programming - OpenMPiv. Modelo de Memória OpenMPv. Arquiteturas SIMDvi. Pipelined Programmingvii. Arquiteturas MIMD | <ul style="list-style-type: none">viii. Programação Multiprocessada em MPIix. Programação Distribuída em MPIx. Primitivas de Comunicação MPIxi. Programação de GPUs (CUDA)xii. Memórias Cachexiii. Profilersxiv. Paralelismo de Tarefas |
|--|---|

3 Horário das aulas e atendimento

AULAS: quarta e sexta-feira, das 10:00 às 11:50 hs.

ATENDIMENTO: agendado via e-mail com o professor (luiz.laranjeira@gmail.com) ou monitor. Teremos monitor(es) para a disciplina.

4 Metodologia

O método básico aplicado será o de aulas expositivas, com auxílio do quadro branco e utilização do projetor digital. A fim de fortalecer a aprendizagem da disciplina, as aulas poderão ser complementadas com atividades de exercícios e demandas extra-classe. Estas atividades serão desenvolvidas com acompanhamento do professor.

5 Critérios de Avaliação

A avaliação do curso será feita através de 4 (quatro) testes. Ainda poderão ser passadas atividades extras de caráter facultativo.

5.1 Testes

A avaliação do curso será feita através de 4 (quatro) testes, a serem realizadas em datas determinadas no cronograma. Estas datas podem ser alteradas por motivos não previstos, e tais alterações serão comunicadas com a devida antecedência em sala de aula ou através da plataforma Moodle. Cada teste terá uma ou mais questões.

Cada questão será considerada correta apenas se o resultado final, e o respectivo desenvolvimento que leva a este resultado, forem descritos corretamente pelo aluno em sua prova. Uma questão cuja solução proposta pelo aluno tiver resultado final incorreto, desenvolvimento incorreto (mesmo com a resposta correta) ou não tiver desenvolvimento, terá pontuação zerada.

Cada teste abrangerá o conteúdo dado desde o teste anterior (ou o início das aulas, no caso do primeiro teste) até a aula anterior à aplicação do teste.

5.2 Atividades Extra

Poderão ser aplicadas, a critério do professor, atividades extras que também serão levadas em conta para a nota final do aluno. A forma, data de entrega e método de avaliação de tais atividades serão divulgados em sala de aula ou na plataforma Moodle.

5.3 Menção Final

Teremos quatro (4) testes e 3 (três) programas .

Cada teste valerá 10 pontos, e a média dos testes, M_T , será dada pela média das três melhores dentre as notas dos quatro testes:

$$M_T = \frac{(\sum_{i=1}^4 T_i) - T_{min}}{3} \quad \text{onde } T_i \text{ é a nota do } i\text{-ésimo teste } (1 \leq i \leq 4) \text{ e } T_{min} \text{ é a menor nota}$$

Com o critério acima se um(a) aluno(a) perder um teste por qualquer razão a nota deste será automaticamente a nota T_{min} a ser desconsiderada no cômputo de sua média de testes.

A média dos programas, M_P , será dada por:

$$M_P = \frac{P_1 + P_2 + P_3}{3} \quad \text{onde } P_i \text{ é a nota do } i\text{-ésimo programa } (1 \leq i \leq 3)$$

A nota final do curso, N_F , será dada por: $N_F = 0.3M_T + 0.7M_P$

A menção final do curso será atribuída de acordo com a tabela abaixo:

N_F	Menção	Descrição
0,0	SR	Sem Rendimento
de 0,1 a 2,9	II	Inferior
de 3,0 a 4,9	MI	Médio Inferior
de 5,0 a 6,9	MM	Médio
de 7,0 a 8,9	MS	Médio Superior
9,0 ou maior	SS	Superior

5.4 Critérios de Aprovação

Obterá aprovação no curso o aluno que cumprir as duas exigências abaixo:

1. Ter presença em 75% ou mais das aulas;
2. Obter média de testes, M_T , igual ou superior a 5,0.
3. Obter menção final igual ou superior a MM.

6 Cronograma

Semana	Aula / Data	TÓPICOS
1ª	1ª - 19/08	Apresentação do Curso – Conceitos Básicos - 1
	2ª - 21/08	Conceitos Básicos - 2
2ª	3ª - 26/08	Modelos de Computação Paralela Disponibilização do Programa Básico
	4ª - 28/08	Multithreaded Programming OpenMP - 1
3ª	5ª - 02/09	Multithreaded Programming OpenMP - 2
	6ª - 04/09	Multithreaded Programming OpenMP - 3
4ª	7ª - 09/09	Multithreaded Programming OpenMP - 4
	8ª - 11/09	Modelo de Memória OpenMP
5ª	16/09	Atividade de Programação
	18/09	Atividade de Programação
6ª	23/09	Atividade de Programação Entrega do Programa em OpenMP (Multithreaded)
	9ª - 25/09	Arquiteturas SIMD
7ª	10ª - 30/09	Arquiteturas SIMD – Pipeline – Arquiteturas MIMD
	11ª - 02/10	Algoritmos Paralelos Básicos
8ª	12ª - 07/10	Multiprocessed Programming MPI
	13ª - 09/10	Distributed Programming - MPI
9ª	14ª - 14/10	Comunicações MPI - 1
	15ª - 16/10	Comunicações MPI - 2
10ª	21/10	Atividade de Programação
	23/10	Atividade de Programação
11ª	28/10	Atividade de Programação Entrega do Programa em MPI (Distributed)
	16ª - 30/10	Programação GPU - 1
12ª	17ª - 04/11	Programação GPU - 2
	18ª - 06/11	Programação GPU - 3
13ª	19ª - 11/11	Programação GPU - 4
	20ª - 13/11	Memórias Cache
14ª	21ª - 18/11	Otimização para Caches – Divisão em Blocos
	22ª - 20/11	Profilers
15ª	23ª - 25/11	Paralelismo de Tarefas
	27/11	Atividade de Programação
16ª	02/12	Atividade de Programação
	04/12	Atividade de Programação Entrega do Programa em CUDA (GPU)
17ª	09/12	
	11/12	
	16/12	Publicação de Menções Finais
	18/12	Revisão de Notas
	21/12	Submissão de Menções Finais no Sistema Acadêmico

7 Bibliografia

- Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar: *Introduction to Parallel Computing*, Addison Wesley, 2003.
Princípios Gerais de Paralelismo; Visão Algorítmica:
- John L. Hennessy, David A. Patterson: *Computer Architecture: A Quantitative Approach*, 4th edition, Morgan Kaufmann, 2007.
Bíblia de Arquitetura de Computadores:
- Jack Dongarra, Ian Foster, Geoffrey Fox, William Gropp, Ken Kennedy, Linda Torczon, Andy White: *Sourcebook of Parallel Computing*, Morgan Kaufmann, 2003.
Um dos melhores livros em aplicações:
- David E. Culler, Jaswinder Pal Singh, Anoop Gupta: *Parallel Computer Architecture, a Hw/Sw Approach*, Morgan Kaufmann, 1999.
Ênfase em hardware e seus impactos em software:
- Michael Wolfe: *High Performance Compilers for Parallel Computing*, Addison-Wesley, 1996.
Técnicas de otimização e de compilação para paralelismo:
- Livros e padrões de OpenMP, MPI e CUDA (citados nas respectivas aulas)