

## PLANO DE ENSINO

Dados do Componente Curricular		
Código: GAC105	Nome: Programação Paralela e Concorrente	
Carga Horária Total: 68	Carga Horária Teórica: 34	Carga Horária Prática: 34
Dados da Oferta de Disciplina		
Semestre: 2024/1	Turma: 10A	Docente Principal: MARLUCE RODRIGUES PEREIRA
Docente Responsável: MARLUCE RODRIGUES PEREIRA		
Atividades Avaliativas: Atividade avaliativa presencial 1: 20%; Atividade avaliativa presencial 2: 20%; Projeto prático: 20%; Exercícios na aula prática: 20%; Exercícios ANP: 20%;		
Dados da Ementa		
Ementa:  Concorrência e Sincronização de Processos. Projeto de algoritmos paralelos. Programação para Plataformas de Memória Compartilhada. Programação Usando Paradigma de Passagem de Mensagens. Análise de desempenho.		
Conteúdo Programático:  1. Concorrência e Sincronização de Processos. - Conceitos de processos e threads - Interprocess Communication: pipes - Exclusão mútua, deadlock, semáforos e barreiras  2. Projeto de algoritmos paralelos. - Conceitos: decomposição, tarefa, grafo de dependência, concorrência, processo e mapeamento. - Técnicas de decomposição - Características de tarefas e interações - Técnicas de mapeamento para balanceamento de carga - Métodos para contenção de overheads de interações - Modelos de algoritmos paralelos  3. Programação para Plataformas de Memória Compartilhada - O modelo de memória compartilhada - Programação com threads - Programação com OpenMP - Programação para GPU  4. Programação Usando Paradigma de Passagem de Mensagens - O modelo de passagem de mensagens. - Operações de comunicação básicas: broadcast, redução, scatter, gather, send, receive - Interface de passagem de mensagens: MPI, PVM ou outros - Exemplos de comunicação coletiva e operações de computação - MapReduce  5. Análise de desempenho - Tempo de execução, Speedup e eficiência - Lei de Amdahl - Escalabilidade - Overhead paralelo		
Bibliografia Básica:  FARBER, Rob. CUDA application design and development. Waltham, MA: Morgan Kaufmann, c2011. xvii, 315 p.  GRAMA, A., Gupta, A., Karypis, G. and Kumar, V. Introduction to Parallel Computing. Addison Wesley. Second Edition. 2003.  MICHAEL, J. Quinn. Parallel Programming in C with MPI and OpenMP. McGraw-Hill, 2004.		
Bibliografia Complementar:  HUGHES, C. and Hughes, T. Parallel and Distributed Programming Using C++. Addison-Wesley. 2004.  KIRK, David; HWU, Wen-mei. Programming massively parallel processors: a hands-on approach. 2nd ed. Waltham, MA: Morgan Kaufmann, c2013. xx, 496 p.  PACHECO, P. Parallel Programming with MPI. Morgan Kaufman Publishers, Inc., San Francisco, California, 1997.  RAUBER, Thomas; RÜNGER, Gudula. Parallel programming: for multicore and cluster systems. 2nd ed. Berlin, DE: Springer, c2013. xiii, 516 p.  SANDERS, Jason; KANDROT, Edward. CUDA by example: an introduction to general-purpose GPU programming. Upper Saddle River, NJ: Addison-Wesley, c2011. xix, 290 p.		

Versão: 1ª	Data de Cadastro: 08/02/2024 - 15:31:11	
Objetivos:		
Ao final dessa disciplina pretende-se que o aluno tenha desenvolvido as seguintes competências: resolver problemas usando ambiente de programação paralela; conhecer as bibliotecas de paralelização existentes e como funcionam; avaliar criticamente projetos de sistemas de computação (avaliação de desempenho); especificar, projetar, implementar, manter, avaliar e inovar sistemas de computação, empregando teorias, práticas e ferramentas adequadas à infraestrutura de hardware disponível.		
Metodologia de Ensino e Formas de Aferição da Frequência:		
As aulas serão ministradas utilizando os conceitos de Sala de Aula Invertida e Aprendizagem Baseada em Projetos. No Campus Virtual são disponibilizadas as vídeo aulas, exercícios e o conjunto de slides, que deverão ser vistos e analisados antes das aulas presenciais. Dessa forma, o conteúdo teórico é ministrado extraclasse. As aulas práticas serão utilizadas para fixação de conceitos, realização de atividades práticas e resolução de dúvidas. Ressalta-se que a disciplina possui ênfase em programa de algoritmos paralelos. Os exemplos apresentados estão em C/C++ voltado para o Sistema Operacional Linux. As avaliações serão baseadas em participações, atividades realizadas em sala de aula, projeto prático e avaliações individuais. O calendário acadêmico prevê 17 semanas letivas com atividades em sala de aula presencial e não presencial O aprendizado é concretizado através da resolução de exercícios disponibilizados no Campus Virtual ( <a href="https://campusvirtual.ufla.br/presencial/">https://campusvirtual.ufla.br/presencial/</a> ) As avaliações serão baseadas em atividades realizadas em sala de aula, projeto prático e avaliações individuais. Será realizada a aferição de frequência em todas as aulas práticas por meio de chamada oral e nas aulas não presenciais (ANP) através da realização das atividades propostas no Campus Virtual.		
Estratégias de Recuperação de Estudantes de Menor Rendimento:		
Os estudantes com menor rendimento devem agendar atendimento com o(s) docente(s) para elaboração de estratégia individual de recuperação, que pode envolver o desenvolvimento de atividades diversificadas, aplicadas em momentos presenciais. Essas atividades não necessariamente implicam em mudança das notas anteriores.		
Cronograma de Atividades		
Dia	Data	Descrição
1	04/03/2024	Dia letivo sem ministração de aula
2	09/03/2024	ANP Processos e threads. Exclusão mútua, deadlock, semáforos e barreiras. Programação de processos com fork. (2 pontos)
3	11/03/2024	Prática Programação de processos com fork. (3 pontos)
4	16/03/2024	ANP: Interprocess Communication: conceito de pipes, programação de pipes.(2 pontos)
5	18/03/2024	Prática de programação de pipes. (3 pontos)
6	23/03/2024	ANP Modelos de memória compartilhada e distribuída. Modelo PRAM, Cluster. Projetos de algoritmos paralelos. (2 pontos)
7	25/03/2024	Prática - exercícios (2 pontos)
8	30/03/2024	ANP Programação para memória compartilhada com Posix Threads (Pthreads)(1 ponto)
9	01/04/2024	Prática de programação para memória compartilhada com Posix Threads (Pthreads)
10	06/04/2024	ANP: Exercícios
11	08/04/2024	Atividade avaliativa presencial 1 - Pthreads (20 pontos)
12	13/04/2024	ANP: Programação para memória compartilhada com OpenMP - parte 1. (2 pontos)
13	15/04/2024	Prática: Programação para memória compartilhada com OpenMP.
14	20/04/2024	ANP: Programação para memória compartilhada com OpenMP - parte 2.
15	22/04/2024	Prática: Programação para memória compartilhada com OpenMP. (3 pontos)
16	27/04/2024	ANP: Avaliação de desempenho: tempo de execução,speedup, eficiência, Lei de Amdahl, escalabilidade - parte 1 (2 pontos)
17	29/04/2024	Prática de análise de desempenho (3 pontos)
18	04/05/2024	ANP: Avaliação de desempenho: tempo de execução,speedup, eficiência, Lei de Amdahl, escalabilidade - parte 1 (2 pontos)
19	06/05/2024	Prática de análise de desempenho - parte 2
20	11/05/2024	ANP: Exercícios
21	13/05/2024	Prática OpenMP e análise de desempenho
22	18/05/2024	ANP: Programação por passagem de mensagens com MPI - parte 1 (2 pontos)
23	20/05/2024	Prática Programação por passagem de mensagens com MPI - parte 1
24	25/05/2024	ANP: Exercícios Programação por passagem de mensagens com MPI - parte 2
25	27/05/2024	Atividade avaliativa presencial 2 -MPI (20 pontos)
26	01/06/2024	ANP: Programação para GPU - parte 1 (2 pontos)
27	03/06/2024	Prática de Programação para GPU
28	08/06/2024	ANP: Programação para GPU - parte 2 (1 ponto)
29	10/06/2024	Prática de Programação para GPU (3 pontos)

30	15/06/2024	ANP: Programação para grandes volumes de dados. Map Reduce (2 pontos)
31	17/06/2024	Prática de Programação para grandes volumes de dados. (3 pontos)
32	22/06/2024	ANP: Entrega e apresentação do projeto prático
33	24/06/2024	Prática: Entrega e apresentação do projeto prático (20 pontos)
34	01/07/2024	Semana de Avaliação adicional nos termos do art. 126 da Resolução CEPE 473/2018 (data sugerida, passível de alteração)