

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DA EDUCAÇÃO SUPERIOR

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA
CELSO SUCKOW DA FONSECA
Campus Nova Friburgo**

CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

| DEPARTAMENTO |
|--|
| Coordenação do Curso de Bacharelado em Sistemas de Informação |

| PLANO DE CURSO DA DISCIPLINA |
|---|
| PROGRAMAÇÃO PARALELA E CONCORRENTE |

| CÓDIGO |
|------------------|
| GSI9601NF |

| CRÉDITOS |
|----------|
| 4 |

| PERÍODO |
|-----------|
| 6º |

| ANO |
|-------------|
| 2016 |

| SEMESTRE |
|-----------|
| 2º |

| PRÉ-REQUISITOS |
|---|
| Algoritmos em Grafos Sistemas Operacionais Programação Orientada a Objetos |

| DISTRIBUIÇÃO DA CARGA HORÁRIA (AULAS/SEMANA) | | | | |
|---|---------|---------|---------------------|-----------------------|
| PRESENCIAL | | | SEMI- PRESENCIAL | TOTAL AULAS/SEMANA |
| TEÓRICA | PRÁTICA | ESTÁGIO | | |
| 2 | 2 | 0 | 1 | 5 |

| TOTAL DE AULAS NO SEMESTRE |
|-------------------------------|
| 90 |

| EMENTA |
|---|
| Princípios e técnicas de programação concorrente. Memória compartilhada e preempção. Exclusão mútua. Semáforos. Monitores. Threads. Projeto de programas paralelos. Bibliotecas de programação paralela. Comunicação por troca de mensagens. Análise de desempenho. Distribuição de trabalho e recursos. Invocação remota de métodos. |

| OBJETIVOS GERAIS |
|---|
| Conhecer conceitos e modelos de programação paralela e concorrente, tendo-os experimentado na prática ao longo do seu estudo. |

| BIBLIOGRAFIA |
|---|
| BIBLIOGRAFIA BÁSICA <ol style="list-style-type: none">SCHMIDT, Douglas et al. Pattern-oriented software architecture, v.2: patterns for concurrent and networked objects. John Wiley & Sons, 2001.TANENBAUM, Andrew S.; STEEN, Maarten Van. Sistemas distribuídos: princípios e paradigmas. 2.ed. São Paulo: Pearson, Prentice Hall, 2007.TANENBAUM, Andrew S. Sistemas operacionais modernos. 3.ed. São Paulo: Pearson: Prentice Hall, 2010. BIBLIOGRAFIA COMPLEMENTAR <ol style="list-style-type: none">DEITEL, Harvey M. et al. Sistemas operacionais. 3.ed. São Paulo: Prentice Hall, 2005.GOETZ, Brian et al. Java concorrente na prática. Rio de Janeiro: Alta Books, 2008.KIRK, David; HWU, Wen-Mei W. Programando para processadores paralelos: uma abordagem prática à programação de GPU. São Paulo: Elsevier, 2011.SILBERSCHATZ, Abraham et al. Fundamentos de sistemas operacionais. 6.ed. Rio de Janeiro: LTC, 2009.TANENBAUM, Andrew S.; Woodhull, Albert S. Sistemas operacionais: projeto e implementação. 3.ed. Porto Alegre: Bookman, 2008. |

METODOLOGIA

Aulas expositivas com a utilização de recurso audiovisual (data show). Aulas práticas em laboratório.

CRITÉRIO DE AVALIAÇÃO

Aulas expositivas, dialogadas e práticas, ilustradas, com o auxílio de recursos audiovisuais. Todas as aulas serão realizadas em laboratório devidamente equipado com ferramentas de apoio.

PROGRAMA

1. Princípios e técnicas de programação concorrente
 - 1.1. Multiprocessadores
 - 1.2. Memória compartilhada
 - 1.3. Troca de mensagens
 - 1.4. Programação paralela e programação concorrente
 - 1.5. Taxonomia de Flynn
 - 1.6. Fluxos (theads) paralelos
 - 1.7. Fluxos (threads) independentes
 - 1.8. Processos
 - 1.9. Preempção
 - 1.10. Threads cooperativas
 - 1.11. Exemplos de programas
2. Memória compartilhada e preempção
 - 2.1. Funcionamento
 - 2.2. Impasse (deadlock)
3. Exclusão mútua
 - 3.1. Eventos
 - 3.2. Transições
 - 3.3. Visão como máquinas de estado
 - 3.4. Estado da thread
 - 3.5. Estado do sistema
 - 3.6. Travamento (locks)
 - 3.7. Modelos de travamento
 - 3.8. Região Crítica
 - 3.9. Algoritmos (Peterson, Filtros, Padaria, etc.)
4. Semáforos
 - 4.1. Notação de Andrews (atomicidade e await)
 - 4.2. Sincronização
 - 4.3. Problema do produtor/consumidor
 - 4.4. Cooperação
 - 4.5. Buffer limitado
 - 4.6. Outros exemplos (filósofos, canibais e cozinheiros, etc.)
 - 4.7. Técnica de passagem de bastão
5. Monitores
 - 5.1. Conceitos
 - 5.2. Exemplos – exclusão mútua
 - 5.3. Cooperação
 - 5.4. Sinalização
 - 5.5. Políticas de sinalização
 - 5.6. Exemplos
 - 5.7. Cuidados
 - 5.8. Implementação de monitores
6. Threads
7. Projeto de programas paralelos
 - 7.1. Oportunidades de paralelização
 - 7.2. Particionamento
 - 7.3. Comunicação

- 7.4. Aglomeração
- 7.5. Mapeamento
- 8. OpenMP
- 9. Comunicação por troca de mensagens
 - 9.1. Troca de mensagens
 - 9.2. Identificação de pares
 - 9.3. Semântica de envio e recebimento
 - 9.4. Formatos e tipos de dados
 - 9.5. Operações coletivas
 - 9.6. Exemplos (MPI/OpenMP)
- 10. Análise de desempenho
 - 10.1. Modelos de desempenho
 - 10.2. Tempo de computação
 - 10.3. Tempo de comunicação
 - 10.4. Eficiência
 - 10.5. Aceleração
 - 10.6. Escalabilidade
 - 10.7. Projeto de experimentos
 - 10.8. Métricas
 - 10.9. Erros comuns de análise de desempenho
 - 10.10. Medidas de tempos
- 11. Distribuição de trabalho e recursos
 - 11.1. Alocação de trabalhos a processadores
 - 11.2. Distribuição e balanceamento
 - 11.3. Causas de desbalanceamento
 - 11.4. Distribuição sob demanda
 - 11.5. Espaço de tuplas
 - 11.6. Pool de threads
 - 11.7. Transferência e migração de processos
- 12. Invocação remota de métodos (RMI)
- 13. Futuros
- 14. Memória Transacional

| CHEFE DO DEPARTAMENTO | |
|------------------------------|-------------------|
| NOME | ASSINATURA |
| DACY CÂMARA LOBOSCO | |

| PROFESSOR RESPONSÁVEL PELA DISCIPLINA | |
|--|-------------------|
| NOME | ASSINATURA |
| BRUNO POLICARPO TOLEDO FREITAS | |