

# MINISTÉRIO DA EDUCAÇÃO

## CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA CELSO SUCKOW DA FONSECA CAMPUS PETRÓPOLIS

CURSO SUPERIOR DE ENGENHARIA DE COMPUTAÇÃO

|                          |              |                               |                            |  |  |
|--------------------------|--------------|-------------------------------|----------------------------|--|--|
| DEPARTAMENTO             |              | PLANO DE CURSO DA DISCIPLINA  |                            |  |  |
| ENGENHARIA DE COMPUTAÇÃO |              | COMPUTAÇÃO DE ALTO DESEMPENHO |                            |  |  |
| CÓDIGO                   | PERÍODO      | ANO                           | SEMESTRE                   | PRÉ-REQUISITOS   |  |
| GCOM9057PE               | 9            | 2015                          | 1                          |  |  |
| CRÉDITOS                 | AULAS/SEMANA |                               |                            | 1. Algoritmos e Estruturas de Dados II<br>2. Sistemas Distribuídos |  |
| 4                        | TEÓRICA      | PRÁTICA                       | ESTÁGIO                    |  |  |
|                          | 2            | 2                             | 0                          |  |  |
|                          |              |                               | TOTAL DE AULAS NO SEMESTRE |  |  |
|                          |              |                               | 72                         |  |  |

### EMENTA

1. Computadores paralelos: CPUs multinucleadas (multi-core); multiprocessamento paralelo; computação distribuída: aglomerados computacionais (clusters) e grades computacionais (grids); Computação e Ambientes computacionais paralelos; Algoritmos Paralelos.
2. Ambientes de programação paralela (MPI, OpenMP). Avaliação experimental de programas paralelos.
3. Configuração de compiladores e otimizadores para processadores uni (single-core) e multinucleados; Programação paralela com threads; Programação de propósito geral em unidades de processamento gráfico (GPGPU, OpenCL, CUDA); Programação em hardware reconfigurável-FPGA (Field-Programmable Gate Array).

### BIBLIOGRAFIA

#### Básica:

1. COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. Sistemas distribuídos: conceitos e projeto. 4 a edição. Porto Alegre: Bookman, 2007.
2. TANENBAUM, A.S.; STEEN, M.v. Distributed systems: principles and paradigms. 2nd. edition. Noida, Índia: Pearson, 2015.
3. TANENBAUM, A.S. Redes de Computadores. Rio de Janeiro: Campus, 2003.

#### Complementar:

1. COMER, D.E. Interligação de Redes com TCP/IP. 5 o edição. Rio de Janeiro: Elsevier, 2006.
2. TANENBAUM, A.S. Sistemas Operacionais Modernos. São Paulo: Pearson.
3. COUTURIER, R. Designing scientific applications on GPUs. Boca Raton, FL: CRC/Taylor & Francis, 2014.
4. KIRK, D.; HWU, W. Programando para processadores paralelos: uma abordagem prática à programação de GPU. Rio de Janeiro: Elsevier, 2011.
5. KILTS, S. Advanced FPGA design: architecture, implementation, and optimization. New York: IEEE; Hoboken, N.J. : John Wiley & Sons, c2007.

|  |
|--|
|  |
|--|

|                         |
|-------------------------|
| <b>OBJETIVOS GERAIS</b> |
|-------------------------|

|  |
|--|
| O objetivo da disciplina é apresentar ao aluno as diferentes técnicas de alto desempenho, paralelismo e otimização de programas. Além disso, o aluno poderá reconhecer os diferentes tipos de arquiteturas de alto desempenho e as soluções mais eficientes para elas. |
|--|

|                    |
|--------------------|
| <b>METODOLOGIA</b> |
|--------------------|

|   |
|---|
| Cada aula consistirá na combinação adequada de: |
|---|

- |  |
|--|
| <ul style="list-style-type: none"><li>• Exposição detalhada de elementos necessários para a criação, execução e análise de tempo de programas.</li><li>• Exercícios, atividades e estudos de casos, de forma individual ou em grupo.</li><li>• Trabalhos práticos de implementação das técnicas ensinadas.</li><li>• Provas individuais.</li></ul> |
|--|

|                              |
|------------------------------|
| <b>CRITÉRIO DE AVALIAÇÃO</b> |
|------------------------------|

|  |
|--|
| A avaliação será feita através de duas provas teórica, correspondendo a 80% da nota final e dois trabalhos práticos, juntamente com seu relatório correspondendo a 20% dos pontos da disciplina. |
|--|

|                              |
|------------------------------|
| <b>CHEFE DO DEPARTAMENTO</b> |
|------------------------------|

| NOME                 | ASSINATURA |
|----------------------|------------|
| Laura Silva de Assis |            |

|  |
|--|
| <b>PROFESSOR RESPONSÁVEL PELA DISCIPLINA</b> |
|--|

| NOME                       | ASSINATURA |
|----------------------------|------------|
| Pedro Carlos da Silva Lara |            |

|  |
|--|
| <b>APROVADO PELO CONSELHO DEPARTAMENTAL EM:</b> ____/____/____ |
|--|

|                              |
|------------------------------|
| <b>CONTEÚDO PROGRAMÁTICO</b> |
|------------------------------|

- |   |
|---|
| <ol style="list-style-type: none"><li>1. Apresentação da disciplina:<ol style="list-style-type: none"><li>1. Programa;</li><li>2. Avaliação;</li><li>3. Datas importantes;</li><li>4. Onde encontrar informação, prazos, carga horária;</li><li>5. Bibliografia.</li></ol></li><li>2. Computadores paralelos<ol style="list-style-type: none"><li>1. Tipo de Processadores SIMD, MIMD MISD, SISD</li><li>2. CPUs multinucleadas (multi-core);</li><li>3. Multiprocessamento paralelo<ol style="list-style-type: none"><li>2.3.1. Tipos de Paralelismo</li><li>2.3.2.Paralelismo de dados</li><li>2.3.3.Paralelismo de tarefas</li></ol></li></ol></li></ol> |
|---|

4. Computação distribuída: aglomerados computacionais (clusters)
5. Grades computacionais (grids);
6. Computação e Ambientes computacionais paralelos;
7. Algoritmos Paralelos.
  - 2.7.1. Dependência Sequencial
  - 2.7.2. Regiões Críticas
  - 2.7.3. Uso de Mutex, Semáforo
  - 2.7.4. Otimização de Algoritmos Paralelos
3. Ambientes de programação paralela
  1. OpenMP
    - 3.1.1. Uso de diretivas de pré-compilação #pragma
    - 3.1.2. Implementando regiões paralelas
    - 3.1.3. Implementando regiões críticas
    - 3.1.4. Uso de locks
    - 3.1.5. Loop for paralelo
  2. MPI
    - 3.2.1. Ambientes de memória distribuída
    - 3.2.2. API para message passing
    - 3.2.3. Funções Send e Recv
    - 3.2.4. Sincronização de Processos
4. Avaliação experimental de programas paralelos
  1. Profiling de Programa
  2. Avaliação do uso de memória
5. Configuração de compiladores e otimizadores para processadores uni (single-core) e multinucleados;
  1. Programação paralela com threads;
  2. Programação de propósito geral em unidades de processamento gráfico (GPGPU)
    - 2.1. OpenCL
    - 2.2. CUDA
  3. Programação em hardware reconfigurável-FPGA (Field-Programmable Gate Array).