



Plano de Ensino

1) Identificação

| | | | |
|-----------------------|--|--------------|--------------|
| Disciplina: | INE5645 - Programação Paralela e Distribuída | | |
| Turma(s): | 05238A | | |
| Carga horária: | 72 horas-aula | Teóricas: 36 | Práticas: 36 |
| Período: | 1º semestre de 2017 | | |

2) Cursos

- Sistemas de Informação (238)

3) Requisitos

- INE5611 - Sistemas Operacionais

4) Ementa

Modelos de interação entre processos. Compartilhamento de memória e mecanismos de sincronização. Troca de mensagens e mecanismos de comunicação. Problemas clássicos. Princípios de implementação. Programação em redes de computadores. Programação distribuída. Linguagens paralelas e distribuídas.

5) Objetivos

Geral: Apresentar conceitos e técnicas de programação paralela e distribuída.

Específicos:

- Apresentar e exercitar a programação paralela com sincronização e troca de mensagens.
- Apresentar exemplos de interfaces de programação e linguagens paralelas e distribuídas

6) Conteúdo Programático

- 6.1) Introdução à Programação Paralela e Distribuída [4 horas-aula]
 - Vantagens e Dificuldades
 - Plataformas de Execução
 - Suporte Computacional
- 6.2) Programação Paralela [16 horas-aula]
 - Processos
 - Threads
 - Paralelismo em Java
- 6.3) Controle de Concorrência [18 horas-aula]
 - Monitores
 - Locks
 - Semáforos
 - Concorrência na API Java
- 6.4) Programação Distribuída [8 horas-aula]
 - Modelos de sistemas distribuídos
 - Elementos básicos da comunicação
 - Comunicação por passagem de mensagem
- 6.5) Comunicação entre Processos [26 horas-aula]
 - Pipes
 - Sockets
 - RMI

7) Metodologia

Pela característica prática da disciplina, todas as aulas serão realizadas em laboratório de ensino, enquanto que as provas escritas serão realizadas na sala de aula destinada à disciplina.

O aluno deve realizar as tarefas, aproveitando o tempo destinado para a aula. Deve mostrar ao professor, ao término de cada tarefa, antes de postar no Moodle.

Unidade 0 - Introdução à Programação Concorrente, Paralela e Distribuída (AT). Revisão: Processos. Threads, Ciclo de Vida de Threads. Prioridades de threads. Métodos de mudança de estado de threads. Tarefas práticas em Programação concorrente sem sincronização. Com sincronização de threads . Formas de Escalonamento da Linguagem Java Lab 0 (TP)

Unidade I – Tarefas práticas em Programação Concorrente, Mecanismos de sincronização entre threads, usando Monitor. Lab 1.

Unidade II e III – Tarefas práticas usando ambientes de Programação Paralela possíveis (OpenMp, OpenCL) - Lab II e III (a, b).

Unidade IV – Tarefas práticas em Programação Distribuída com Sockets (AT, TP) - Lab IV (a,b,c,d).

Unidade V – Tarefas práticas usando ambientes de Programação Paralela Distribuída (MPI). Lab V.

Como complemento, para o que não nos é possível executar em computador, uma tarefa teórica é pedida, a ser entregue ao final da disciplina.

8) Avaliação

Os alunos serão avaliados com base em seu desempenho nas seguintes atividades:

- 2 Provas Teóricas (P1 e P2);
- 5 Atividades de laboratório: Lab 1, Lab 2, Lab 3 (a, b), Lab 4 (a,b,c,d), Lab 5.

A média das provas (MP) será calculada aplicando a fórmula:

$$MP = (P1 + P2)/2.$$

A média das tarefas práticas (MT) será atribuída aplicando a fórmula

$$MT = [0,10*(Lab 1) + 0,15*(Lab 2) + 0,20*(Lab 3a) + 0,20*(Lab 3b) + 0,05*Lab4a + 0,05*Lab4b + 0,05*Lab4c + 0,05*Lab4d + 0,15*Lab 5]$$

O aluno deve realizar cada tarefa no horário da aula (por si só ou em dupla) e termine a mesma, mostrando, antes de postar no Moodle, sua execução correta, na data prevista, ao professor.

O professor procurará realizar recuperações de notas das provas escritas, logo após os resultados das mesmas, durante o semestre.

$$\text{Media do semestre (Moodle)} = MF = (0.6*MP + 0.40*MT)$$

Para realização de avaliações em atraso, de acordo com a RESOLUÇÃO Nº 17/CUn/97, de 30 de setembro de 1997:

Art. 70 § 4o - Ao aluno que não comparecer às avaliações ou não apresentar trabalhos no prazo estabelecido será atribuída nota 0 (zero).

Art. 74 - O aluno, que por motivo de força maior e plenamente justificado, deixar de realizar avaliações previstas no plano de ensino, deverá comunicar ao professor ou formalizar pedido de avaliação à Chefia do Departamento de Ensino ao qual a disciplina pertence, dentro do prazo de 3 (três) dias úteis, recebendo provisoriamente a menção I.

Casos de recuperação (regras da UFSC): Vide no que segue.

Conforme parágrafo 2º do artigo 70 da Resolução 17/CUn/97, o aluno com frequência suficiente (FS) e média final no período (MF) entre 3,0 e 5,5 terá direito a uma nova avaliação ao final do semestre (REC), sendo a nota final (NF) calculada conforme parágrafo 3º do artigo 71 desta resolução, ou seja: $NF = (MF + REC) / 2$.

9) Cronograma

O cronograma aproximado, pelas experiências em períodos passados:

06/03 Unidade 0 - Plano de Ensino, Histórico da Programação Concorrente, Paralela e Distribuída (AT)

08/03 Unidade 0 - Breve Revisão: Processos e Threads. Ciclo de Vida das Threads. Métodos de mudança de estado de threads. Prioridades de threads. Implementação de threads. Multithreading. Exemplo sem sincronização; Formas de escalonamento em Java. (Lab 0, sem contar na avaliação)

13/03 Unidade I - Atividade de Laboratório 1 (Lab 1) Monitor (TP).

15/03 Unidade I - Atividade de Laboratório 1 (Lab 1) Monitor (TP)

20/03 Unidade I - Atividade de Laboratório 1 (Lab 1) Monitor (TP)

22/03 Unidade II- Atividade de Laboratório 1 (Lab 2) Monitor (TP)

27/03 Unidade II - Atividade de Laboratório 2 (Lab 2) OpenMP (TP)

29/03 Unidade II - Atividade de Laboratório 2 (Lab 2) OpenMP(TP)

03/04 Unidade II - Atividade de Laboratório 2 (Lab 2) OpenMP (TP)

05/04 Unidade II - Atividade de Laboratório 2 (Lab 2) OpenMP (TP)

10/04 Unidade III - Atividade de Laboratório 3 (Lab 3) OpenCL (TP)

12/04 Unidade III - Atividade de Laboratório 3 (Lab 3) OpenCL (TP)

17/04 Unidade III - Atividade de Laboratório 3 (Lab 3) OpenCL (TP)

19/04 Unidade III - Atividade de Laboratório 3 (Lab 3) OpenCL (TP)

24/04 Prova 1 – Unidades I, II e III. (escrita)

26/04 Unidade IV - Atividade de Laboratorio 4 (Lab 4a) Sockets UDP (AT)

03/05 Unidade IV - Atividade de Laboratorio 4 (Lab 4a) Sockets UDP (TP)

08/05 Unidade IV - Atividade de Laboratorio 4 (Lab 4b) Sockets TCP (TP)

10/05 Unidade IV - Atividade de Laboratorio 4 (Lab 4b) Sockets TCP (TP)

15/05 Unidade IV - Atividade de Laboratorio 4 (Lab 4c) Multicast Socket (TP)

17/05 Unidade IV - Atividade de Laboratorio 4 (Lab 4c) Multicast Socket (TP)

22/05 Unidade IV - Atividade de Laboratorio 5 (Lab 4d) CallBack c/Sockets (TP)

24/05 Unidade IV - Atividade de Laboratorio 5 (Lab 4d) CallBack c/Sockets (TP)

29/05 Unidade V - Atividade de Laboratorio 5 (Lab 5) MPI (TP)

31/05 Unidade V - Atividade de Laboratorio 5 (Lab 5) MPI (TP)

05/06 Unidade V - Atividade de Laboratório 5 (Lab 5) MPI (TP)

07/06 Unidade V - Atividade de Laboratório 5 (Lab 5) MPI (TP)

12/06 Outros Modelos de Programação Paralela (TT)

14/06 Outros Modelos de Programação Paralela (TT)

19/06 Outros Modelos de Programação Paralela (TT)

21/06 Outros Modelos de Programação Paralela (TT)

26/06 Prova 2 – Unidades IV, V (escrita)

28/06 Entrega de Tarefas Atrasadas

03/07 Avaliação Final

05/07 Entrega das Notas Finais

10) Bibliografia Básica

- ANDREWS, G. R., Concurrent Programming, Benjamin-Cummings, 1991.
- COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim. Distributed Systems: -Concepts and Design. 3rd Edition. Addison-Wesley, 2001.
- DEA, Doug “Concurrent Programing in Java”, 2nd Ed., Addison-Wesley, 2000.
- DEITEL, Harvey M.; DEITEL, Paul J. Java: Como Programar. 4a Edição. Bookman, 2002.
- HORSTMANN, Cay S.; CORNELL, Gary. Core Java 2. Vol I e II. Makron Books, 1999.

11) Bibliografia Complementar

- ORFALI, Robert; HARVEY, Dan. Client/Server Programming with Java and CORBA. 2nd Edition. John Wiley, 1998.
- STEVENS, W. R., Unix Network Programming, vols. 1 e 2, Prentice-Hall, 1998.
- TANENBAUM, Andrew. Sistemas Operacionais Modernos. Prentice-Hall, 2003.