



**Universidade Federal de Santa Catarina
Centro Tecnológico
Departamento de Informática e Estatística
Plano de Ensino 2013.2**

1) Identificação

Disciplina: INE5645 - Programação Paralela e Distribuída

Turma(s): 05238B

Carga horária: 72 horas-aula Teóricas: 36 Práticas: 36

Período: 2º semestre de 2013

Professor: João Bosco Manguiera Sobral

2) Cursos

- Sistemas de Informação (238)

3) Requisitos

- INE5611 - Sistemas Operacionais

4) Ementa

Histórico da Programação Concorrente. Processos e Threads. 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 (Histórico) [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

AT (Aula Teórica), AP (Aula Prática), TP (Tarefa Prática), TT (Tarefa Teórica)

Unidade I - Introdução à Programação Paralela e Distribuída (AT)

Unidade II - Programação Paralela (AT, AP, TP)

Unidade III - Monitor, Locks e Semáforos (AT, AP, TP)

Plano de Ensino - INE5645 - Programação Paralela e Distribuída - 14/08/2013 - Página: 1/3

Unidade IV - Programação Distribuída com Sockets (AT, AP, TP)

Unidade V - Programação Distribuída com RMI (AT, AP, TP)

Unidade VI - Eventos e Notificações - Sockets ou RMI (AT, AP, TP)

8) Avaliação

Os alunos serão avaliados com base em seu desempenho nas seguintes atividades: 2 Provas teóricas (P1 e P2) realizadas em sala de aula; 4 Atividades de laboratório (LAB1 a LAB5); 1 Projeto final (PROJ); e mais pequenas tarefas práticas realizadas em laboratório, preliminarmente às tarefas LAB1, LAB2, LAB3, LAB4 e LAB5.

A média teórica será calculada aplicando a fórmula: $MT = (P1 + P2)/2$.

Caso o aluno não alcance média teórica 6,0 (seis), uma prova de recuperação (PR) será aplicada. A média MT, para esse caso, será $((P1+P2)/2)+PR)/2$.

A média prática será atribuída aplicando a fórmula $MP = (0,20 (LAB1) + 0,20 (LAB2) + 0,20 (LAB3) + 0,10 (LAB4) + 0,10 (LAB5) + 0,20*PROJ)/6$.

Para ser aprovado, o aluno deve possuir médias teórica e prática iguais ou superiores a 6,0 (seis). Nesse caso, a média final será calculada aplicando a fórmula $MF = (MT + MP)/2$. Caso contrário, a média final será igual ao valor mais baixo dentre as médias teórica e prática.

Observações:

Será atribuída uma nota para a equipe (dupla de alunos) que para cada trabalho de laboratório. Se um aluno não responder a uma pergunta do trabalho, a equipe terá desconto na nota do trabalho. Em caso de cópia de trabalhos de qualquer fonte, todos os alunos envolvidos terão nota igual a zero. Será utilizada uma ferramenta de detecção de plágio (ex. SIM - Software and text similarity tester) para comparação dos trabalhos.

Trabalhos entregues com atraso terão desconto automático de pontos. Após a segunda semana (14 dias após o fim do prazo original), o trabalho não mais será aceito, ou seja, terá nota zero.

Em caso de "cola" nas avaliações, os alunos envolvidos estarão automaticamente reprovados com média zero, e o caso poderá ser reportado ao colegiado do curso para que sejam tomadas medidas disciplinares.

Para efeito de avaliação, o prazo para entrega de tarefas deverá ser respeitado, não aceitando tarefas fora do prazo.

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

12/8 Unidade I - Plano de Ensino, Histórico da Programação concorrente (AT)

14/8 Unidade II - Multithreading: Threads, Estados de threads, Prioridades de threads, Implementação de threads (AT, AP)

19/8 Ambiente de execução Multithreading (AT)

21/8 Sincronização de threads: Produtor-Consumidor sem sincronização (AP)
26/8 Unidade III - Monitores (AT)
28/8 Produtor-Consumidor com sincronização de Monitor (AP)
02/9 Atividade de Laboratório 1 (Lab 1) (TP)
04/9 Atividade de Laboratório 1 (Lab 1) (TP)
09/9 Unidade III - Locks (AT)
11/9 Produtor-Consumidor com sincronização de Locks (AP)
16/9 Atividade de Laboratório 2 (Lab 2) (TP)
28/9 Atividade de Laboratório 2 (Lab 2) (TP)
23/9 Unidade III - Semáforos (AT)
25/9 Atividade de Laboratório 3 (Lab 3) (TP)
30/9 Atividade de Laboratório 3 (Lab 3) (TP)
02/10 Questionário 1 (TT)
07/10 [Prova 1](#) - Assunto: Unidades I, II e III
09/10 Correção da Prova 1
14/10 Unidade IV - Sockets - Aplicações Distribuidas (AT)
16/10 Datagram Sockets (AP)
21/10 TCP Stream Sockets (AP)
23/10 Multicast Socket (AP)
28/10 Atividade Laboratório 4 (Lab 4) (TP)
30/10 Atividade Laboratório 4 (Lab 4) (TP)
04/11 Unidade V - Java RMI - Aplicações Distribuidas (AP)
06/11 Exemplo básico RMI (AP)
11/11 Atividades do Laboratório 5 (Lab 5) (TP)
13/11 Atividades do Laboratório 5 (Lab 5) (TP)
18/11 Unidade VI - Eventos e Notificações (AT)
20/11 Atividades do Laboratório 6 (AP)
25/11 Questionário 2 (TT)
27/11 [Prova 2](#) - Unidades IV, V e VI
02/12 Correção da Prova 2
04/12 Atividades do Laboratório 6 (TP)
09/12 Atividades do Laboratório 6 (TP)

Recuperação da Aprendizagem: [Realizada no decorrer da disciplina, logo após as provas 1 e 2.](#)

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, 2007.
- 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.