

# **SSC0146 – Sistemas Computacionais Tolerantes a Falhas**

**Prof. Jó Ueyama**

ICMC/USP – 1o. Semestre 2012

# Sejam bem-vindos!

- **Fundamentos de Sistemas Tolerantes a Falhas**
- **Trabalhos práticos envolvendo o assunto**

# Aula de Hoje

- Apresentação da disciplina
- Conteúdo programático
- Datas importantes

# Conteúdo Detalhado de Hoje

- Apresentação do professor
- Alguns esclarecimentos
- Metodologia
- Avaliação de aprendizado
- Regras de funcionamento
- Plano de Ensino
- Perguntas?

# Apresentação do professor

- PhD em Ciência da Computação

- Lancaster University - UK



- Pos-doc

- University of Kent at Canterbury - UK



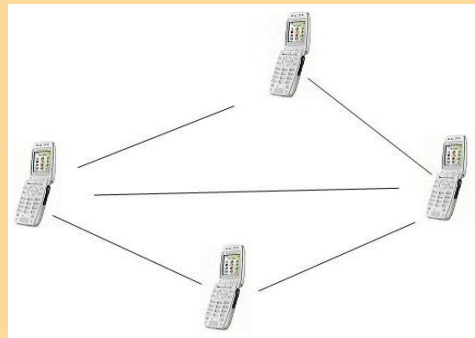
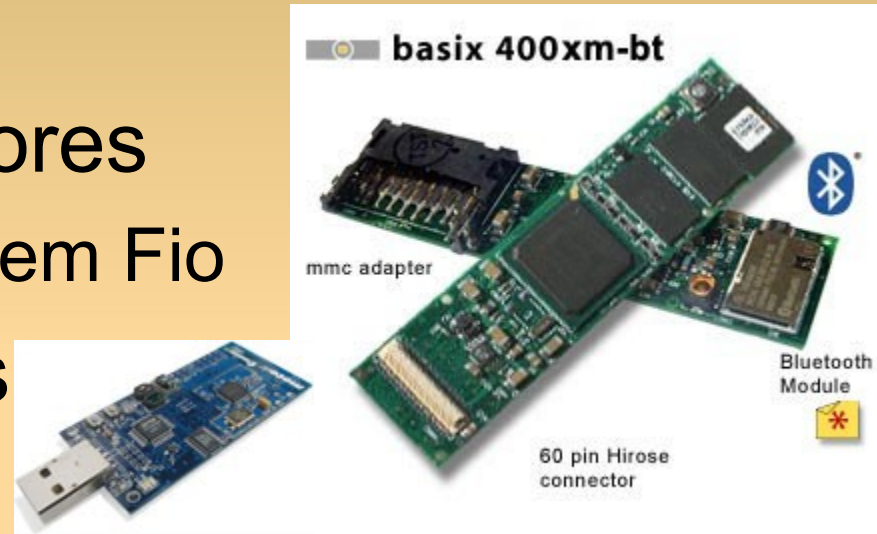
- Pos-doc

- UNICAMP – Campinas SP



# Áreas de atuação

- Redes de Computadores
  - Redes de Sensores sem Fio
- Sistemas distribuídos
  - Mobilidade
  - Programação em telefones celulares



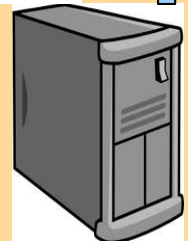
# Áreas de atuação RSSFs para o Meio-ambiente



3. emissão de alertas via SMS. Os números devem ser cadastrados previamente no servidor



1. transmissão multihop de dados via ZigBee



2. armazenamento e processamento dos dados lidos do rio



servidor com Internet<sup>7</sup>



# Áreas de atuação RSSFs para o Meio-ambiente





# Áreas de atuação RSSFs e Cloud Computing



# Alguns Esclarecimentos (1)

- Princípios básicos:
  - É uma disciplina fundamental
  - O objetivo de todos é assimilar a disciplina
    - Não ganhar uma nota
    - Não é passar no semestre seguinte
    - Não é rodar os alunos :-)
- Presença
  - Haverá chamada sistemática
  - O importante é entender os conceitos
  - Eu aconselho fortemente estar presente

# Alguns Esclarecimentos(2)

- Coteia-wiki
  - Material de apoio
  - NÃO É SUFICIENTE
- Durante a aula
  - Prestar atenção
  - Anotar
  - Perguntas interativas
  - Horários de entrada/saída
  - SILÊNCIO
- Provas
  - Provas dissertativas?
  - Escrever pouco, claramente, e JUSTIFICAR.

# Contra-exemplo de prova

- Devido ao grande número de processos para termos multitarefa e grande tamanho para mantermos as tabelas de páginas organizamos tudo como memória virtual onde teremos a sensação de que diversos processos estarão “rodando” contiguamente em nossa RAM devido ao fato de armazenarmos apenas em RAM aquelas páginas que estão sendo utilizadas pro processo e assim teremos que gerenciar o paging muito bem para obtermos essa sensação que nada mais é do que verdade visto que programas em geral têm o princípio da localidade e que nem sempre todas as rotinas e tratamentos de erros dos programas são utilizados.

# Problemas já identificados

- Falta às aulas;
- “professor, eu trabalho..., moro longe...”;
- Laptops (**NÃO É PERMITIDO O USO DURANTE A AULA**);
- celular;
- conversas;
- menosprezar o assunto;
- superestimar a própria inteligência.

# Metodologia (1)

- Exposição em aula pelo docente dos temas do curso.
- Realização de exercícios em aula e no laboratório.
- Realização de avaliações escritas (provas e atividades práticas) pelos alunos.
- Apresentação do progresso do trabalho prático pela equipe de alunos

# Metodologia (2)

- Atendimento de dúvidas dos alunos pelo professor:
  - Horário: quartas das 18:00 às 19:00hs. Caso ninguém compareça nos primeiros 15 minutos, o atendimento será finalizado naquele dia.
  - Local: sala 3-115
  - Não haverá atendimento na véspera das provas (até uma semana antes)
- Não haverá atendimento por email!
- Contagem de frequência APENAS no horário de atendimento



# Metodologia (3)

- Um trabalho prático a ser apresentado no final do semestre.
- Tópico: tolerância a falhas em redes de sensores sem fio
  - Uso do middleware OpenCom em *motes* Sun SPOT
- O trabalho será realizado por uma equipe de no máximo TRÊS alunos
- Atenção: plágio é nota zero na DISCIPLINA

# Avaliação do aprendizado

- Os alunos deverão freqüentar 70% ou mais das aulas.
- Procurar Seção da Graduação para as faltas.
- Duas provas (P1 e P2) e pelo menos uma atividade prática será realizada.
- As atividades práticas poderão ser realizadas em duplas.
- Será realizada uma prova substitutiva **SOMENTE** para os alunos entrarem com o requerimento na Seção da Graduação
- A matéria da prova substitutiva envolve todo o conteúdo ministrado na disciplina.

# Avaliação do aprendizado

- A média de provas (MP) é calculada da seguinte maneira:  $MP = (P1+P2)/2$ .
- Idem para a média das atividades práticas (MT)
- A média final (MF) é calculada da seguinte forma:
  - $MF = (MT + MP)/2$
  - $MF \geq 5,0$ , aluno aprovado;  $MF < 5,0$ ; aluno reprovado.

# Avaliação do aprendizado

- Requisitos:
  - $\text{freq} \geq 70\%$
  - $3 \leq \text{MF} < 5$
- Para ser aprovado:
  - $\text{MR} = (\text{MF} + \text{Rec}) / 2 \geq 5.0$
- A REC será realizada na primeira segunda-feira do período da mesma (no horário da aula).

# Regras de Funcionamento

- Cola e cópia não serão toleradas e implicarão em nota zero na **DISCIPLINA!!**
- Conversas não serão toleradas durante a aula e implicarão em perda de pontos nas provas.

# Regras de Funcionamento

- A programação de aulas é preliminar e pode estar sujeita a mudanças, que serão informadas em aula.
- Os slides da aula funcionam como “guia”, e o uso dos livros é parte essencial do aprendizado.
- O aluno poderá chegar no máximo com 30 minutos de atraso para as aulas.
- Verificação/recontagem de frequência apenas nos horários de atendimento.

# Plano de Ensino (I)

1. INTRODUÇÃO AOS SISTEMAS TOLERANTES A FALHAS
  - 1.1. Definições
  - 1.2. Falhas e mecanismos de prevenção
  - 1.3. Exclusão (*removal*) de falhas
  - 1.4. Tipos de recuperação
  - 1.5. Redundância e seus tipos
2. Estruturando a redundância para sistemas tolerantes a falhas
  - 2.1. Definições
  - 2.2. Sistemas robustos
  - 2.3. Projetos de redundância
  - 2.4. Níveis de redundância
  - 2.5. Redundância de dados
  - 2.6. Redundância temporal
  - 2.7. Arquitetura de softwares para prover redundância
  - 2.8. Estudos de casos



# Plano de Ensino (II)

3. Técnicas de programação para sistemas tolerantes a falhas
  - 3.1 *Assertions*
  - 3.2 *Checkpointing*
  - 3.3 Ações atômicas
  - 3.4 Paradigmas de software *N-version*
  
4. Projeto de redundância de software
  - 4.1 *Recovery blocks*
  - 4.2 Operações e exemplos de *recovery blocks*
  - 4.3 Operações e exemplos de *N-version*
  - 4.4 *Recovery blocks* distribuídos
  - 4.5 Operações e exemplos de *recovery blocks* distribuídos
  - 4.6 Programação *N-self checking*
  - 4.7 Operações e exemplos de programação *N-self checking*
  - 4.8 *Consensus recovery block*
  - 4.9 Operações e exemplos de *consensus recovery block*
  - 4.10 Comparações entre os diferentes tipos de projeto

# Datas Importantes

- Aulas
  - Quinta-feira
- Apresentação do trabalho prático
  - 21/06
- Provas
  - 03/05 e 14/06

# Bibliografia

## **Livros texto:**

1) Software Fault Tolerance – Techniques and implementation, Laura L. Pullum

2) Fault-Tolerant Systems

Israel Koren and C. Mani Krishna

# Perguntas?