

**Universidade Federal de Santa Catarina  
Atividades de Pesquisa  
Formulário de Tramitação e Registro**

Situação: **Aprovação/Depto Coordenador**  
Protocolo nº: **2013.1141**

Título:	Infraestrutura para Sistemas de Tempo Real em Processadores Multicore
Resumo:	Com a diminuição a possibilidade de aumentar a frequência e com isso, o superaquecimento dos processadores, a maioria dos fabricantes opta por adotar a tecnologia de processadores com múltiplos núcleos (multicore). Com isso, foi possível aumentar o desempenho e diminuir o consumo de energia. Para se utilizar essa nova tecnologias nos próximos projetos, será necessário o desenvolvimento de metodologias, padrões, arquiteturas e ferramentas para auxiliar no desenvolvimento sistemático desta classe especial de sistemas embarcados e de tempo real. De modo que, essas novas abordagens possam continuar a manter ou aumentar a a previsibilidade de tarefas de tempo real. Para isso, esse projeto visa a automação das tarefas de teste e verificação dos modelos de alto nível de sistemas embarcados e de tempo real, melhorias na verificação e testes das tarefas modeladas quanto à escalonabilidade e, por fim, o desenvolvimento de técnicas de software/hardware para aumentar a previsibilidade de tarefas de tempo real executadas em um processador multicore. Espera-se nesse projeto contribuições nas áreas de: (i) verificação e testes de modelos de sistema de tempo real; e (ii) suporte de sistema operacional de tempo real para processadores multicore (escalonadores, gerência de recursos, sincronização, arquitetura, etc.).
Palavras chave: (máximo 5)	real time; multicore; operating systems; simulation; verification and test
Grande Área do conhecimento:	Ciências Exatas e da Terra
Área do conhecimento:	Software Básico
Nome do Grupo de Pesquisa: (CNPq - Diretório)	Laboratório de Integração de Software e Hardware
Está vinculado a outro projeto de pesquisa?	
Período de realização:	01/10/2013 a 31/07/2016
A atividade receberá algum aporte financeiro?:	Não
Propriedade Intelectual (o resultado do projeto é ou poderá ser protegido por):	

■ ■ ■ **Envolvidos neste projeto de pesquisa**

<b>Coordenador</b>	
Nº do SIAPE:	2047937
Nome do Coordenador:	Gian Ricardo Berkenbrock
CPF do Coordenador:	2864114909
Departamento:	CAMPUS DE JOINVILLE
Centro:	CAMPUS DE JOINVILLE
Regime de trabalho:	DE
Fone de contato:	48-3721-6417
E-mail:	gian.rb@ufsc.br
Carga horária semanal nesta atividade:	20 horas
Receberá remuneração nesta atividade de pesquisa?	Não

Você gostaria de participar do guia de fontes da UFSC?	Não
--	-----

Outros prof. ou servidores da UFSC envolvidos?	Sim
--	-----

Alunos da UFSC envolvidos?	Não
----------------------------	-----

Pessoas externas à UFSC envolvidas?	Não
-------------------------------------	-----

<b>Participantes</b> Participante: Giovani Gracioli CAMPUS DE JOINVILLE
--

Outras Considerações
----------------------

Nº do Processo: 2013.1141
---------------------------



**Universidade Federal de Santa Catarina**  
**Atividades de Pesquisa**  
**Formulário de Tramitação e Registro - Participante**

Situação: **Aprovação/Depto Coordenador**

Titulo da Atividade:	Infraestrutura para Sistemas de Tempo Real em Processadores Multicore
Objetivos e Metodologia:	Com a diminuição a possibilidade de aumentar a frequência e com isso, o superaquecimento dos processadores, a maioria dos fabricantes opta por adotar a tecnologia de processadores com múltiplos núcle
Grande Área:	Ciências Exatas e da Terra
Área:	Software Básico
Nome do Grupo de Pesquisa:	Laboratório de Integração de Software e Hardware
Período Realização:	01/10/2013 a 31/07/2016
Orçamento Total:	R\$

**Participante**

Nro do SIAPE:	2056505
Participante:	Giovani Gracioli
CPF do Participante:	1083806025
Departamento:	CAMPUS DE JOINVILLE
Centro:	CAMPUS DE JOINVILLE
Regime de Trabalho:	DE
Fone:	48-3721-6417
Período de participação nesta Atividade:	01/Outubro/2013 a 31/Julho/2016
Carga Horária nesta Atividade:	20 (Ex.: 7,5)
Receberá remuneração nesta atividade de pesquisa?	Não
Valor TOTAL da remuneração:	

**Histórico**

09/19/2013 06:59:14 PM - Criado por Gian Ricardo Berkenbrock 2047937

# Infraestrutura para Sistemas de Tempo Real em Processadores *Multicore*

Equipe:

Gian Ricardo Berkenbrock `gian.rb@ufsc.br`

Giovani Gracioli `giovani@lisha.ufsc.br`

18-09-2013

# Conteúdo

<b>1 Projeto</b>	<b>2</b>
1.1 Resumo . . . . .	2
1.2 Contextualização . . . . .	3
1.3 Objetivos . . . . .	5
1.4 Período de realização . . . . .	7
1.5 Carga horária . . . . .	7
1.6 Considerações Finais . . . . .	8
<b>Bibliografia</b>	<b>9</b>

# 1 Projeto

## 1.1 Resumo

Com a diminuição a possibilidade de aumentar a frequência e com isso, o superaquecimento dos processadores, a maioria dos fabricantes opta por adotar a tecnologia de processadores com múltiplos núcleos (*multicore*). Com isso, foi possível aumentar o desempenho e diminuir o consumo de energia. Para se utilizar essa nova tecnologias nos próximos projetos, será necessário o desenvolvimento de metodologias, padrões, arquiteturas e ferramentas para auxiliar no desenvolvimento sistemático desta classe especial de sistemas embarcados e de tempo real. De modo que, essas novas abordagens possam continuar a manter ou aumentar a a previsibilidade de tarefas de tempo real. Para isso, esse projeto visa a automação das tarefas de teste e verificação dos modelos de alto nível de sistemas embarcados e de tempo real, melhorias na verificação e testes das tarefas modeladas quanto à escalonabilidade e, por fim, o desenvolvimento de técnicas de software/hardware para aumentar a previsibilidade de tarefas de tempo real executadas em um processador *multicore*. Espera-se nesse projeto contribuições nas áreas de: (i) verificação e testes de modelos de sistema de tempo real; e (ii) suporte de sistema operacional de tempo real para processadores *multicore* (escalonadores, gerência de recursos, sincronização, arquitetura, etc.).

## 1.2 Contextualização

Com a diminuição do potencial de aumento da frequência e devido ao superaquecimento dos processadores, a maioria dos fabricantes adotou a tecnologia de processadores com múltiplos núcleos (*multicore*). Com isso, foi possível aumentar o desempenho e diminuir o consumo de energia. Como resultado, até sistemas de baixo custo e vídeo *games* passaram a utilizar processadores *multicore* (TAM *et al.*, 2007).

Neste sentido, processadores *multicore* estão sendo cada dia mais usados no contexto de Sistemas Embarcados e Tempo Real (SETR) devido à evolução e integração de funcionalidades nesses sistemas. Por exemplo, em um automóvel moderno, novas funções de segurança como “parada automática de emergência” e “auxílio de visão noturna” devem ler dados dos sensores, processar o vídeo e exibir avisos preventivos quando um obstáculo é detectado na via em tempo real (MOHAN *et al.*, 2011). Além disso, a adição de novas funcionalidades ao sistema, geralmente implementadas em hardware dedicado, custa em termos de consumo de energia e dissipação de calor e espaço (e.g., cabeamento) (CULLMANN *et al.*, 2010). Portanto, processadores *multicore* tornam-se uma boa alternativa para diminuir esses custos e integrar as diversas funções em uma única unidade de processamento, ao contrário de se ter diversas unidades de controle eletrônica dedicadas espalhadas no veículo.

Entretanto, como SETR devem garantir que os *deadlines* das tarefas sejam respeitados, o uso de processadores *multicore* torna-se um desafio, principalmente devido ao compartilhamento de diferentes recursos físicos, tais como a memória cache (SUHENDRA; MITRA, 2008; APARICIO *et al.*, 2011; SARKAR; MUELLER; RAMAPRASAD, 2012; SHEKHAR *et al.*, 2012; MANCUSO *et al.*, 2013; GRACIOLI *et al.*, 2013), barramentos e periféricos (BOYD-WICKIZER *et al.*, 2010; BETTI *et al.*, 2013), que afetam a previsibilidade para estimar o *Worst-Case Execution Time* (WCET) das aplicações (WEHMEYER; MARWEDEL, 2005; ZHURAVLEV; BLAGODUROV; FEDOROVA, 2010).

Com isso, o Sistema Operacional de Tempo Real (SOTR) deve ser capaz de corre-

tamente monitorar quando a disputa por recursos compartilhados influencia o tempo de execução das tarefas e, conseqüentemente, a perda de *deadlines*, para que possa mitigar os efeitos do compartilhamento de recursos e aumentar a previsibilidade do sistema. Adicionalmente, o projeto e a implementação das estruturas internas do Sistema Operacional (SO) também são importantes para diminuir o sobrecusto em tempo de execução e aumentar a escalonabilidade do sistema.

Muitos destes sistemas embarcados encontram-se distribuídos, necessitando cooperar para atingir um objetivo comum. Para atender a grande demanda da indústria existe um engajamento global de pesquisadores no sentido de desenvolver metodologias, padrões, arquiteturas e ferramentas para auxiliar no desenvolvimento sistemático desta classe especial de sistemas embarcados distribuídos, cooperativos e de tempo real.

Neste sentido, a ideia de utilizar modelos como os elementos principais para projetar sistemas complexos vem ganhando força nos últimos anos. Modelos ajudam a entender um problema complexo e suas potenciais soluções através de níveis mais abstratos de especificação (SELIC, 2003). Seguindo o preceito de que modelos são essenciais nos projetos de engenharia tradicionais (e.g. a construção de um edifício, o projeto aerodinâmico de um avião, ou construção de um motor eletromecânico), muitos pesquisadores e profissionais da indústria defendem que os modelos produzidos durante o projeto devem passar a exercer o papel principal durante todo o ciclo de projeto.

A chamada Engenharia Guiada por Modelos (do inglês Model-Driven Engineering ou MDE) (SCHMIDT, 2006) define que o projeto deve focar principalmente na criação de modelos ao invés da escrita de linhas de código. Portanto, neste paradigma de projeto, modelos são os artefatos mais importantes, pois são fáceis de especificar, entender e manter, além de serem menos sensíveis às trocas de tecnologia de implementação (i.e. são independentes das plataformas de implementação). Para apoiar essa ideia, uma premissa fundamental é a de que os artefatos executáveis (i.e. código fonte a ser compilado) devem ser gerados automaticamente a partir dos modelos do sistema, evitando inconsistências entre a informação dos modelos e a implementação real do sistema.

Esse projeto é um passo a mais em direção de: (i) a automação das tarefas de teste e verificação dos modelos de alto nível de sistemas embarcados e de tempo real; (ii) verificação das tarefas modeladas quanto à escalonabilidade; e (iii) desenvolvimento de técnicas de software/hardware para aumentar a previsibilidade de tarefas de tempo real executadas em um processador *multicore*.

## 1.3 Objetivos

O principal objetivo desse projeto é desenvolver uma infraestrutura para o desenvolvimento e execução de sistemas de tempo real em processadores *multicore*. De modo a atingir esse objetivo, são definidos os seguintes objetivos específicos, cada um associado à uma meta.

### 1. Investigar o estado da arte

Esse objetivo será cumprido pela pesquisa bibliográfica realizada conjuntamente pelos colaboradores do projeto, bem como estudos de casos relevantes. Os principais tópicos de interesse incluem, mas não estão limitados a:

- Uso de estruturas de dados como listas e heaps e estruturas de dados não bloqueantes em SOTR;
- Gerenciamento de tempo nos SOTR em processador *multicore*;
- Sincronização nas estruturas de dados internas do SOTR entre os diferentes *cores*;
- Análise de algoritmos de substituição de linhas da cache para tempo real;
- Algoritmos de escalonamento *multicore* tempo real consciente de recursos compartilhados;
- Técnicas de modelagem e verificação de STR e SOTR;
- Abordagens de simulação dos modelos especificados com restrições temporais;

- Integração do tarefas de teste e verificação dos modelos de alto nível com o SOTR.

## 2. Sistemas operacionais de tempo real

As atividades relacionadas com a meta 2 devem produzir uma série de melhorias em aspectos de como o SOTR é projetado e implementado para prover previsibilidade em aplicações com restrições temporais executadas em um processador *multicore*. Espera-se também com essa meta, a produção de algoritmos de escalonamento *multicore* tempo real conscientes de recursos compartilhados e a integração de técnicas de gerenciamento de recursos com os escalonadores.

## 3. Questões arquiteturais dos processadores *multicore* atuais

As atividades relacionadas à meta 3 devem produzir uma série de documentos descrevendo a integração de questões arquiteturais dos processadores *multicore* com o SOTR. Eventualmente, novos mecanismos de hardware, como barramentos compartilhados, podem ser projetados e implementados em FPGA para melhorar o suporte de tempo real.

## 4. Modelagem, Verificação, Teste e Simulação de modelos com restrições temporais

As atividades vinculadas nessa meta devem produzir melhorias nas abordagens de modelagem, verificação, teste e simulação de modelos de alto nível de abstração. Deve-se levar em consideração que esses modelos produzidos terão especificação temporais.

## 5. Integração das tarefas de desenvolvimento dos STR com especificidades do SOTR

Nessa meta as atividades serão concentradas em, dado um conjunto de tarefas modeladas, analisar suas escalonabilidade quanto as algoritmos de escalonamento disponíveis no SOTR. Também irá lidar com outras análises para seleção de algoritmos específicos do SOTR.

### 6. **Cooperação com grupos de pesquisa**

Esta meta visa a escrita de projetos de cooperação com outros grupos de pesquisa nacionais e internacionais com o intuito de trocar experiências e aumentar a qualidade dos resultados da pesquisa.

### 7. **Apoiar a formação de alunos de graduação, mestrado e doutorado**

Um dos principais objetivos deste projeto de pesquisa é a capacitação de alunos em fase de desenvolvimento de trabalhos de final de curso (graduação, mestrado e doutorado) através do envolvimento direto na execução do presente projeto.

### 8. **Disseminação do conhecimento**

Através dos trabalhos de conclusão de curso mencionados acima e também de artigos publicados em periódicos e congressos científicos, objetiva-se disseminar o conhecimento produzido durante a execução desse projeto.

## 1.4 Período de realização

Esse projeto será realizado no período entre os dias 01 de outubro de 2013 até 31 de julho de 2016.

## 1.5 Carga horária

Os seguintes participantes desse projetos estão listados abaixo:

<b>Participante</b>	<b>Siape</b>	<b>Carga Horária</b>
Gian Ricardo Berkenbrock	2047937	20
Giovani Gracioli	2056505	20

## **1.6 Considerações Finais**

A equipe de execução do projeto será composta pelos pesquisadores/professores e por alunos de graduação e pós-graduação, além de outros pesquisadores que por ventura possam colaborar. Os recursos financeiros para alcançar as metas propostas serão buscados através de projetos enviados para agências de fomento, como CAPES e CNPq, e através de projetos de pesquisa e desenvolvimento com a indústria, vinculados à uma fundação para gerir os recursos. Saliente-se a atualidade do tema de pesquisa proposto e os vários desafios de pesquisa existentes.

# Bibliografia

- APARICIO, L. C.; SEGARRA, J.; RODRIGUEZ, C.; VIÑALS, V. Improving the wcet computation in the presence of a lockable instruction cache in multitasking real-time systems. **J. of Sys. Arch.**, v. 57, n. 7, p. 695 – 706, 2011. ISSN 1383-7621.
- BETTI, E.; BAK, S.; PELLIZZONI, R.; CACCAMO, M.; SHA, L. Real-time i/o management system with cots peripherals. **IEEE Trans. Comput.**, IEEE Computer Society, Washington, DC, USA, v. 62, n. 1, p. 45–58, jan. 2013. ISSN 0018-9340.
- BOYD-WICKIZER, S.; CLEMENTS, A. T.; MAO, Y.; PESTEREV, A.; KAASHOEK, M. F.; MORRIS, R.; ZELDOVICH, N. An analysis of linux scalability to many cores. In: **Proceedings of the 9th USENIX conference on Operating systems design and implementation**. Berkeley, CA, USA: USENIX Association, 2010. (OSDI'10), p. 1–8.
- CULLMANN, C.; FERDINAND, C.; GEBHARD, G.; GRUND, D.; MAIZA, C.; REINEKE, J.; TRIQUET, B.; WEGENER, S.; WILHELM, R. Predictability considerations in the design of multi-core embedded systems. **Ingénieurs de l'Automobile**, v. 807, p. 36–42, September 2010. ISSN 0020-1200.
- GRACIOLI, G.; FRÖHLICH, A. A.; PELLIZZONI, R.; FISCHMEISTER, S. Implementation and evaluation of global and partitioned scheduling in a real-time OS. **Real-Time Systems**, v. 49, n. 6, p. 669–714, nov. 2013. ISSN 1573-1383.
- MANCUSO, R.; DUDKO, R.; BETTI, E.; CESATI, M.; CACCAMO, M.; PELLIZZONI, R. Real-time cache management framework for multi-core architectures.

In: **Proc. of the RTAS '13**. [S.l.]: IEEE, 2013.

MOHAN, S.; CACCAMO, M.; SHA, L.; PELLIZZONI, R.; ARUNDALE, G.; KEGLEY, R.; NIZ, D. de. Using multicore architectures in cyber-physical systems. In: **Workshop on Developing Dependable and Secure Automotive Cyber-Physical Systems from Components**. Michigan, USA: [s.n.], 2011.

SARKAR, A.; MUELLER, F.; RAMAPRASAD, H. Static task partitioning for locked caches in multi-core real-time systems. In: **Proc. of the CASES '12**. NY, USA: ACM, 2012. (CASES '12), p. 161–170. ISBN 978-1-4503-1424-4.

SCHMIDT, D. C. Guest editor's introduction: Model-driven engineering. **IEEE Computer**, v. 39, n. 2, Feb. 2006.

SELIC, B. The pragmatics of model-driven development. **IEEE Software**, IEEE Computer Society Press, Los Alamitos, v. 20, n. 5, p. 19–25, 2003. ISSN 0740-7459.

SHEKHAR, M.; SARKAR, A.; RAMAPRASAD, H.; MUELLER, F. Semi-partitioned hard-real-time scheduling under locked cache migration in multicore systems. In: **Proc. of the ECRTS'12**. [S.l.]: IEEE, 2012. ISBN 978-0-7695-3298-1.

SUHENDRA, V.; MITRA, T. Exploring locking & partitioning for predictable shared caches on multi-cores. In: **Proc. of the DAC'08**. [S.l.]: ACM, 2008. p. 300–303. ISBN 978-1-60558-115-6.

TAM, D.; AZIMI, R.; SOARES, L.; STUMM, M. Managing shared l2 caches on multicore systems in software. In: **In WIOSCA'07**. [S.l.: s.n.], 2007.

WEHMEYER, L.; MARWEDEL, P. Influence of memory hierarchies on predictability for time constrained embedded software. In: **DATE '05: Proceedings of the conference on Design, Automation and Test in Europe**. Washington, DC, USA: IEEE Computer Society, 2005. p. 600–605. ISBN 0-7695-2288-2.

ZHURAVLEV, S.; BLAGODUROV, S.; FEDOROVA, A. Addressing shared resource

contention in multicore processors via scheduling. In: **Proceedings of the 15th edition of ASPLOS on Architectural support for programming languages and operating systems**. New York, NY, USA: ACM, 2010. (ASPLOS '10), p. 129–142.

ISBN 978-1-60558-839-1.