**Universidade Federal de Minas Gerais**
Instituto de Ciências Exatas
Departamento de Ciência da Computação

# Uma plataforma de tradução binária estática de programas desenvolvidos em uma arquitetura de processador de origem diferente da arquitetura de destino

Aluno: André França Braga

---

Orientadora: Mariza Andrade da Silva Bigonha

---

Belo Horizonte
Agosto de 2005

# Introdução

As tecnologias mudam. O estado da arte atual será tornado obsoleto em breve, e cada vez mais rápido. A fita cassete ficou obsoleta com a introdução do CD, o VHS ficou obsoleto com a introdução do DVD, e os dias desse já estão contados graças ao desenvolvimento dos drives ópticos de laser azul e das mídias de múltiplas camadas, que permitem a criaçao de discos compactos de alta capacidade. Porém, a transição entre a tecnologia legada e a de ponta raramente é suave.

Os avanços na Computação são diários. Novas arquiteturas são introduzidas, as existentes são refinadas e aceleradas. Normalmente, novidades em software estão sujeitas às novidades em hardware; novos estilos de programação, novos paradigmas e novas aplicações ganham aceitação quando o hardware subjacente é capaz de oferecer desempenho satisfatório a elas. Desde 1965, quando Gordon Moore enunciou sua famosa lei [GM65], a previsão de que é possível dobrar o desempenho dos computadores a cada dois anos vem sido confirmada, e novas aplicações tornaram-se viáveis. Porém, é importante salientar que desempenho puro não é tudo: de tempos em tempos acontecem descobertas que quebram paradigmas e podem potencialmente mudar de maneira radical os limites que aprendemos a aceitar.

Quebras de paradigmas e avanços radicais, no entanto, não alteram o fato de que o mundo não está pronto para mudar abruptamente, especialmente diante da diferenciação entre hardware e software. Em um primeiro momento, deve haver um modo de aproveitar as novidades sem abrir mão de tudo o que foi desenvolvido até então. Transições suaves são economicamente interessantes tanto para quem introduziu as novidades, já que seus novos produtos poderão ser consumidos imediatamente e os obsoletos podem ter sua produção interrompida assim que desejável, como para quem consome tais produtos, de forma que o investimento já feito não se torne um estorvo diante de novas necessidades de desempenho, consumo de energia elétrica, miniaturização, e até mesmo reposição de componentes que passaram a apresentar defeito.

A indústria da Computação já passou grandes transições, com diferentes níveis de sucesso, em pelo menos cinco ocasiões e com propósitos diferenciados: quando a Apple Computer, Inc. substituiu a arquitetura de seus computadores, mudando da família de processadores Motorola 68k para a família PowerPC [APPC]; quando a Digital Equipment Corporation tentou atrair clientes que usavam Windows NT em processadores Intel para sua arquitetura Alpha [FX32]; quando a Intel Corporation fez o mesmo, com relação a diversos sistemas operacionais, da arquitetura IA-32 (x86) para IA-64 (Itanium) [32EL]; quando a Transmeta Corporation anunciou seus processadores de baixo consumo Crusoe, e posteriormente Efficeon, compatível com processadores Intel mas com arquitetura interna completamente diferente [TMEO]; e, novamente, com a Apple Computer substituindo a arquitetura de seus computadores da família PowerPC para a família Intel [ARST]. Especula-se que essa última transição seja devido a uma sexta transição, nos moldes daquela feita pela Transmeta, mas dessa vez pela própria Intel: uma mudança radical na arquitetura, capaz de possibilitar novas aplicações, mas mantendo compatibilidade com o o software atual [NBTI].

Todas essas transições foram possíveis graças ao uso de técnicas e tecnologia de emulação ou tradução, seja em software separado, como no caso da Apple Computer, DEC e Intel, seja embarcado diretamente nos processadores, como é o caso da Transmeta (e, possivelmente, num futuro próximo, Intel). Tais técnicas tornaram possível continuar o uso de software legado, além de permitir o desenvolvimento de novas aplicações que tirassem proveito dos novos avanços, na mesma arquitetura de hardware.

Este projeto pretende desenvolver uma plataforma para tradução de programas na forma binária entre arquiteturas cujos conjuntos de instruções sejam diferentes entre si. O objetivo é permitir que programas desenvolvidos para arquiteturas legadas, e que por algum motivo não podem ser transportados - isto é, recompilados - para arquiteturas modernas, como por exemplo, pela perda de código-fonte, possam ser usados nestas de maneira transparente. Algumas razões para isso já foram enumeradas: aumento de desempenho, diminuição no consumo de energia elétrica, redução do espaço ocupado, ou mesmo oferecer uma alternativa quando não existem peças de reposição para componentes defeituosos.

# Detalhamento do problema

Neste projeto orientado em computação pretende-se desenvolver um tradutor binário estático capaz de traduzir programas simples de uma arquitetura para outra. A simplicidade deve-se às restrições de tempo de execução de um projeto orientado, ou seja, aproximadamente 10 meses. Assim, o sistema operacional em ambas arquiteturas será o mesmo, a fim de manter o mesmo conjunto de chamadas de sistema, e as arquiteturas escolhidas serão relativamente modernas, a fim de facilitar acesso ao hardware em questão ou simuladores adequados dos mesmos. Esse projeto envolve as disciplinas Software Básico, Compiladores e Sistemas Operacionais.

A proposta inicial é que a arquitetura de conjunto de instruções de origem seja o MIPS32 [MIPS] e a de destino, Intel IA-32 (conforme implementado na primeira versão do processador Pentium) [IA32]. O sistema operacional em ambas arquiteturas será escolhido entre duas possibilidades: Linux ou NetBSD. Não é feita qualquer garantia sobre eficiência; o objetivo é a correção.

Esse Projeto será mantido suficientemente flexível para que o tradutor entre um par de arquiteturas específicas possa ser estendido, futuramente, para uma plataforma eficiente de tradução entre quaisquer pares de arquitetura, possivelmente envolvento técnicas de otimização através de tradução dinâmica ou quaisquer outras técnicas cujas implementações não sejam viáveis em um projeto de final de curso. A idéia é criar condições iniciais para que um

projeto de maior porte seja desenvolvido dentro desta Universidade, já que, como pode ser visto, há oportunidades comerciais claras para o uso da tecnologia da emulação e tradução binária. É muito importante que nossa Universidade e nosso País domine tais tecnologias, por questões econômicas e estratégicas.

Algumas soluções existentes, certamente mais sofisticadas que a aqui proposta, podem ser encontradas em [FX32], [32EL], [UQBT], [ARST], [RSTT], [QEMU], [TSTV].

# Plano de trabalho

## Pesquisa

### Parte I

- Duração: 1o. de setembro de 2005 a 30 de setembro de 2005

• Primeira fase: estudo de viabilidade
- Duração aproximada: 1o. a 10 de setembro
- Estudo preliminar para determinar a viabilidade das escolhas iniciais de arquitetura de instruções e sistemas operacionais. Nela será feita uma pesquisa sobre possíveis emuladores de processadores e sistemas completos capazes de executar um sistema operacional, e também testes correspondentes. A suposição é que o sistema operacional NetBSD seja uma escolha viável.

*Assumindo que as escolhas iniciais não apresentem problemas, segue este cronograma tentativo:*

• Segunda fase: estudo da arquitetura MIPS
- Duração aproximada: 11 a 20 de setembro

• Terceira fase: determinação de possíveis linguagens intermediárias
- Duração aproximada: 21 a 25 de setembro
- Será estudada a possibilidade de abstrair a arquitetura de destino através de uma linguagem intermediária, que, por sua vez, será traduzida para o conjunto de instruções em questão. Inclui-se aqui o estudo do uso de um compilador já existente ou a necessidade de desenvolver uma linguagem e compilador correspondente específicos para essa aplicação.

• Elaboração da apresentação oral
- Duração aproximada: 25 a 29 de setembro

*Encerramento da parte I*
- Apresentação de resultados parciais
- Data: 30 de setembro

### Parte II

- Duração: 1o. de outubro de 2005 a 5 de dezembro de 2005

◦ *Opcional*: estudo da arquitetura IA-32
- Duração aproximada: 1o. a 20 de outubro
- Caso a segunda hipótese levantada na terceira fase seja determinada verdadeira.

• Quinta fase: especificação do tradutor de conjunto de instruções
- Duração aproximada: 1o. de outubro a 7 de novembro

• Sexta fase: estudo das chamadas de sistema do sistema operacional escolhido
- Duração aproximada: 8 de novembro a 17 de novembro
- Nessa fase serão estudadas as diferenças nas convenções de chamadas de sistema dependentes de arquitetura, como o mecanismo de passagem de parâmetros, interrupções de software etc.

• Sétima fase: especificação do tradutor de chamadas de sistema
- Duração aproximada: 18 de novembro a 30 de novembro

• *Encerramento da parte II*: apresentação oral do trabalho do POC I
- Data: 2 de dezembro

• Entrega do relatório final de POC I

- Data: 05 dezembro
- O relatório será elaborado ao longo de todo o curso de POC I.

# Implementação

• Primeira fase: “prova de conceito”
- Duração aproximada: 3 a 4 semanas
- Consistirá em fazer um tradutor extremamente simples que converta a linguagem de montagem do μ-RISC, que é bastante utilizado nos cursos de Organização de Computadores, para a linguagem de montagem da arquitetura IA-32 conforme implementada na primeira versão do Pentium.

• Segunda fase: Da linguagem de montagem do MIPS32 para linguagem de montagem da IA-32
- Duração aproximada: 3 a 5 semanas
- Com base no trabalho desenvolvido na primeira fase, a segunda fase irá fazer um tradutor que converta a linguagem de montagem da arquitetura MIPS32 (versão R2000) para a da arquitetura IA-32 (Pentium).

• Terceira fase: “por onde os ovos são quebrados”
- Duração aproximada: 3 a 5 semanas
- IA-32 é uma arquitetura “little endian”. MIPS32 pode funcionar tanto em modo “little endian” como em “big endian”. Essa fase tratará das traduções necessárias entre os dados dos programas. Ela está para os dados assim como a segunda fase está para as instruções. O resultado dessa fase será combinado com o da segunda para formar um tradutor completo de instruções e dados brutos.

• Quarta fase: Programa de NetBSD MIPS32 para programa de NetBSD IA-32
- Duração aproximada: 3 a 5 semanas
- Nessa fase, será feita a tradução do mecanismo de chamada de sistema de uma arquitetura para a outra. O resultado dessa fase será combinado com o da terceira, e o produto do projeto estará praticamente concluído.

• Quinta fase: testes abrangentes
- Duração aproximada: 3 a 5 semanas
- Nessa fase, alguns aplicativos reais de linha de comando serão traduzido. Os problemas encontrados serão corrigidos. A possibilidade de traduzir um programa com interface gráfica será investigada.

• Sexta fase: entrega do produto.

# Referencias bibliograficas iniciais


[GM65]
ftp://download.intel.com/museum/Moores_Law/Articles-press_Releases/Gordon_Moore_1965_Article.pdf

[APPC]
http://lowendmac.com/orchard/05/0801.html

[FX32]
http://www.usenix.org/publications/library/proceedings/usenix-nt97/full_papers/chernoff/chernoff.pdf

[MIPS]
http://www.mips.com/content/Documentation/MIPSDocumentation/ProcessorArchitecture/doclibrary

[IA32]
http://developer.intel.com/design/pentium/MANUALS/241430.htm

[32EL]
http://portal.acm.org/citation.cfm?id=956417.956550
http://www.intel.com/design/itanium/downloads/254318.htm
http://news.com.com/Intel+plans+Itanium+course+correction/2100-1006_3-997936.html?tag=nl

[TMEO]
http://www.transmeta.com/efficeon/codemorphing.html

[NBTI]
http://www.theinquirer.net/?article=25496

[PTOW]
http://www.program-transformation.org/Transform/BinaryTranslation

[UQBT]
http://www.itee.uq.edu.au/~cristina/uqbt.html

[ARST]
http://developer.apple.com/documentation/MacOSX/Conceptual/universal_binary/universal_binary_exec_a/chapter_7_section_1.html

[RSTT]
http://www.wired.com/wired/archive/13.08/start.html?pg=12

[QEMU]
http://fabrice.bellard.free.fr/qemu/qemu-tech.html#SEC7

[TSTV]
http://www.transitive.com/technology.htm