

**Fabio Mascarenhas de
Queiroz**

**Integração entre a Linguagem
Lua e o Common Language
Runtime**

DISSERTAÇÃO DE MESTRADO

**DEPARTAMENTO DE INFORMÁTICA
Programa de Pós-graduação em
Informática**

Rio de Janeiro
Março de 2004

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO



Fabio Mascarenhas de Queiroz

**Integração entre a Linguagem Lua e o
Common Language Runtime**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática da PUC

Orientador: Prof. Roberto Ierusalimsky

Rio de Janeiro
Março de 2004



Fabio Mascarenhas de Queiroz

**Integração entre a Linguagem Lua e o
Common Language Runtime**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico Científico da PUC. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Roberto Ierusalimschy

Orientador
Departamento de Informática — PUC

Prof. Noemi Rodriguez

PUC–Rio

Prof. Renato Cerqueira

PUC–Rio

Prof. José Eugenio Leal

Coordenador Setorial do Centro Técnico Científico —
PUC

Rio de Janeiro, 19 de Março de 2004

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Fabio Mascarenhas de Queiroz

Graduou-se Bacharel em Ciência da Computação pela Universidade Federal da Bahia (Salvador, Bahia).

Ficha Catalográfica

Queiroz, Fabio Mascarenhas de

Integração entre a Linguagem Lua e o Common Language Runtime/ Fabio Mascarenhas de Queiroz; orientador: Roberto Ierusalimschy. — Rio de Janeiro : PUC, Departamento de Informática, 2004.

v., 68 f: il. ; 29,7 cm

1. Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui referências bibliográficas.

1. Informática – Teses. 2. Common Language Runtime. 3. Compiladores. 4. Linguagem Lua. 5. Linguagens de programação. 6. Microsoft .NET. 7. Máquinas virtuais. I. Ierusalimschy, Roberto. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Agradecimentos

Ao meu orientador, Professor Roberto Ierusalimschy, pelo apoio ao trabalho, a disponibilidade para responder minhas dúvidas e a paciência para revisar os meus textos e apontar os meus erros. Agradeço também a ele e ao Professor Renato Cerqueira pela inspiração para este trabalho.

À CAPES e à PUC-Rio, pelo auxílio concedido, sem o qual a realização deste trabalho seria bem mais difícil.

À minha família, pelo apoio incondicional à minha decisão de me mudar para outro estado, para continuar a minha formação.

Resumo

Queiroz, Fabio Mascarenhas de; Ierusalimschy, Roberto. **Integração entre a Linguagem Lua e o Common Language Runtime**. Rio de Janeiro, 2004. 68p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

O Common Language Runtime (CLR) é uma plataforma criada com o objetivo de facilitar a interoperabilidade entre diferentes linguagens de programação, através de uma linguagem intermediária (a Common Intermediate Language, ou CIL) e um sistema de tipos comum (o Common Type System, ou CTS). Lua é uma linguagem de script flexível e de sintaxe simples; linguagens de script são frequentemente usadas para juntar componentes escritos em outras linguagens, para construir protótipos de aplicações, e em arquivos de configuração.

Este trabalho apresenta duas abordagens de integração entre a linguagem Lua e o CLR, com o objetivo de permitir que scripts Lua instanciem e usem componentes escritos para o CLR. A primeira abordagem é a de criar uma ponte entre o interpretador Lua e o CLR, sem modificar o interpretador. Os recursos e a implementação desta ponte são mostrados, e ela é comparada com trabalhos que seguem a mesma abordagem.

A segunda abordagem é a de compilar as instruções da máquina virtual do interpretador Lua para instruções da Common Intermediate Language Do CLR, sem introduzir mudanças na linguagem Lua. A implementação de um compilador de instruções Lua para CIL é mostrada, e o desempenho de scripts compilados por ele é comparado com o desempenho dos mesmos scripts executados pelo interpretador Lua e com o de scripts equivalentes compilados por outros compiladores de linguagens de script para o CLR.

Palavras-chave

linguagem Lua; compiladores; Common Language Runtime; máquinas virtuais; Microsoft .NET; linguagens de script

Abstract

Queiroz, Fabio Mascarenhas de; Ierusalimschy, Roberto. **Integrating the Lua Language and the Common Language Runtime**. Rio de Janeiro, 2004. 68p. MSc. Dissertation — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The Common Language Runtime (CLR) is a platform that aims to make the interoperability among different programming languages easier, by using a common language (the Common Intermediate Language, or CIL) and a common type system (the Common Type System, or CTS). Lua is a flexible scripting language with a simple syntax; scripting languages are frequently used to join components written in other languages, to build application prototypes, and in configuration files.

This work presents two approaches for integration between the Lua language and the CLR, with the objective of allowing Lua scripts to instantiate and use components written for the CLR. The first approach is to create a bridge between the Lua interpreter and the CLR, without changing the interpreter. The features and implementation of this bridge are shown, and it is compared with other work following the same approach.

The second approach is to compile the virtual-machine instructions of the Lua interpreter to instructions of the CLR's Common Intermediate Language, without introducing changes to the Lua language. The implementation of a Lua instructions to CIL compiler is shown, and the performance of scripts compiled by it is compared with the performance of the same scripts run by the Lua interpreter and with the performance of equivalent scripts compiled by compilers of other scripting language to the CLR.

Keywords

Lua language; compilers; Common Language Runtime; virtual machines; Microsoft .NET; scripting languages

Conteúdo

| | | |
|-----|---|----|
| 1 | Introdução | 11 |
| 1.1 | A Linguagem Lua | 13 |
| 1.2 | O Common Language Runtime | 15 |
| 2 | Interface entre o CLR e o Interpretador Lua | 18 |
| 2.1 | Recursos da Interface | 18 |
| 2.2 | Implementação da Interface | 27 |
| 3 | Compilando Bytecodes Lua para CIL | 34 |
| 3.1 | A Máquina Virtual de Lua 5.0 | 34 |
| 3.2 | Mapeamento dos Tipos de Lua para o CTS | 35 |
| 3.3 | Compilação das Instruções | 37 |
| 3.4 | Implementando Outros Recursos de Lua 5.0 | 39 |
| 3.5 | Interface com o CLR | 42 |
| 4 | Trabalhos Correlatos | 44 |
| 4.1 | Pontes entre Lua e Outras Plataformas | 44 |
| 4.2 | Pontes entre Interpretadores e o CLR | 46 |
| 4.3 | Compiladores de Linguagens Dinâmicas para o CLR | 49 |
| 5 | Conclusão | 55 |
| | Referências Bibliográficas | 59 |
| A | Instruções da Máquina Virtual de Lua 5.0 | 63 |

Lista de Figuras

| | | |
|-----|--|----|
| 4.1 | Tempos de chamada para métodos com parâmetros System.Int32 | 48 |
| 4.2 | Tempos de chamada para métodos com parâmetros de tipos de objeto | 48 |
| 4.3 | Comparação entre compiladores para o CLR | 52 |
| 4.4 | Tempos de chamada para métodos | 53 |

Lista de Tabelas

| | | |
|-----|---|----|
| 1.1 | Tipos do CTS | 16 |
| 1.2 | Tipos primitivos | 17 |
| 4.1 | Scripts para o teste de desempenho dos compiladores | 51 |

Realizing this [programming as sketching] has real implications for software design. It means that a programming language should, above all, be malleable. A programming language is for thinking of programs, not for expressing programs you've already thought of. It should be a pencil, not a pen. Static typing would be a fine idea if people actually did write programs the way they taught me to in college. But that's not how any of the hackers I know write programs. We need a language that lets us scribble and smudge and smear, not a language where you have to sit with a teacup of types balanced on your knee and make polite conversation with a strict old aunt of a compiler.

Paul Graham, *Hackers and Painters*.