



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

LUA INTERPRETER

DOCUMENTO DE ESCOPO DE PROJETO

Alunos

Fernando Benbassat
Guilherme Cavalcanti
Kamilla Cardoso

{fbc@cin.ufpe.br}
{gjcc@cin.ufpe.br}
{krfc@cin.ufpe.br}

RECIFE, MAIO DE 2014.

1 – Introdução

Lua é uma linguagem de programação criada em 1993 no Tecgraf, um laboratório de pesquisa e desenvolvimento da PUC-Rio. Lua é uma linguagem de script dinâmico, semelhante a outras linguagens de script como Python, Ruby e PHP. Lua destaca-se pela sua simplicidade, portabilidade, rapidez e pela facilidade com que se pode embutir um interpretador Lua em uma aplicação C. Além disso, Lua é única linguagem criada em um país em desenvolvimento a ganhar relevância global.

Outras características importantes de Lua são:

- O seu tamanho pequeno. O núcleo da linguagem somado às suas bibliotecas padrões ocupa menos de 200K.
- Lua tem uma merecida reputação de ótimo desempenho. Outras linguagens de script aspiram ser "tão rápidas quanto Lua". Vários benchmarks mostram Lua como a linguagem mais rápida dentre as linguagens de script interpretadas. Lua é rápida não só em programas específicos para benchmarks, mas no dia-a-dia também. Porções substanciais de aplicações grandes são escritas em Lua.
- Lua é distribuída via um pequeno pacote e compila sem modificações em todas as plataformas que têm um compilador C padrão. Lua roda em todos os tipos de Unix e Windows, e também em dispositivos móveis (usando Android, iOS, BREW, Symbian, Windows Phone), em microprocessadores embutidos (como ARM e Rabbit, para aplicações como Lego MindStorms), e até mainframes IBM.
- Lua é uma *engine* rápida e pequena que você pode facilmente embutir na sua aplicação. Lua tem uma API simples e bem documentada que permite uma integração forte com código escrito em outras linguagens. É simples estender Lua com bibliotecas escritas em outras linguagens. Também é simples estender programas escritos em outras linguagens com Lua. Lua é usada para estender programas escritos não só em C e C++, mas também em Java, C#, Smalltalk, Fortran, Ada, Erlang, e mesmo outras linguagens de script, como Perl and Ruby.
- Um conceito fundamental no projeto de Lua é fornecer *meta-mecanismos* para a implementação de construções, em vez de fornecer uma multidão de construções diretamente na linguagem. Por exemplo, embora Lua não seja uma linguagem puramente orientada a objetos, ela fornece meta-mecanismos para a implementação de classes e herança. Os meta-mecanismos de Lua trazem uma economia de conceitos e mantêm a linguagem pequena, ao mesmo tempo que permitem que a semântica seja estendida de maneiras não convencionais.

Lua é uma linguagem de propósito geral, que pode ser utilizada tanto para escrever pequenos scripts, com algumas poucas linhas de código, como sistemas complexos, com dezenas de milhares de linhas de código. Atualmente a linguagem é utilizada principalmente em jogos, como World of Warcraft, The Sims e Sim City. Contudo, Lua não é uma linguagem específica para jogos, sendo utilizada também em muitas outras

aplicações, como o software Adobe Photoshop Lightroom e o middleware Ginga do Sistema Brasileiro de TV Digital.

Um outro uso de Lua é no desenvolvimento de aplicações web, como o Publique! , um gerenciador de conteúdo para web desenvolvido pela Fábrica Digital,e Sputnik , um wiki extensível disponível como software livre.

2 – Objetivo

O objetivo deste projeto é desenvolver um interpretador para a linguagem Lua, suportando seus conceitos básico, e extensível o suficiente para implementações futuras de conceitos mais avançados.

Aqui está a sintaxe completa de Lua em BNF estendido. (Ela não descreve as precedências dos operadores.)

```
chunk ::= block

block ::= {stat} [retstat]

stat ::= ';' |
        varlist '=' explist |
        functioncall |
        label |
        break |
        goto Name |
        do block end |
        while exp do block end |
        repeat block until exp |
        if exp then block {elseif exp then block} [else block] end |
        for Name '=' exp ',' exp [',' exp] do block end |
        for namelist in explist do block end |
        function funcname funcbody |
        local function Name funcbody |
        local namelist ['=' explist]

retstat ::= return [explist] [';']

label ::= '::' Name '::'

funcname ::= Name {'.' Name} ['::' Name]

varlist ::= var {',' var}

var ::= Name | prefixexp '[' exp ']' | prefixexp '.' Name

namelist ::= Name {',' Name}

explist ::= exp {',' exp}
```

```

explist ::= exp {',' exp}

exp ::= nil | false | true | Number | String | '...' | functiondef |
       prefixexp | tableconstructor | exp binop exp | unop exp

prefixexp ::= var | functioncall | '(' exp ')'

functioncall ::= prefixexp args | prefixexp ':' Name args

args ::= '(' [explist] ')' | tableconstructor | String

functiondef ::= function funcbody

funcbody ::= '(' [parlist] ')' block end

parlist ::= namelist [',' '...'] | '...'

tableconstructor ::= '{' [fieldlist] '}'

fieldlist ::= field {fieldsep field} [fieldsep]

field ::= '[' exp ']' '=' exp | Name '=' exp | exp

fieldsep ::= ',' | ';'

binop ::= '+' | '-' | '*' | '/' | '^' | '%' | '..' |
         '<' | '<=' | '>' | '>=' | '==' | '~=' |
         and | or

unop ::= '-' | not | '#'

```

3 – Referências

- [1] www.lua.org
- [2] Ierusalimschy, R. (2013). Uma Introdução à Programação em Lua.
- [3] Celes, W., & Figueiredo, L. H. e Ierusalimschy, R., 2004. “A Linguagem Lua e suas Aplicações em Jogos”.