

Trabalho Prático 1 - Linguagens de Programação

DCC024 - Linguagem de Programação - 2024.1

Professor: Marcio Costa Santos

Linguagem sorteada: Lua

Integrantes do Grupo:

Daniel Oliveira Barbosa	Matrícula
Diane Fenzi Goncalves	Matrícula
Frederico Dolher Souza Baker	2021031513
Rubia Alice Moreira de Souza	Matrícula
Victor Yuji Yano	2022043337
Wallace Eduardo Pereira	Matrícula

Data de Entrega: 20/05/2024

Conteúdo

1	Introdução	1
2	Histórico da Linguagem Lua	1
3	Características Básicas da Linguagem	1
3.1	Estruturas Básicas	2
3.1.1	Declaração de Variáveis e Tipos	2
3.1.2	Controle de Fluxo	2
3.1.3	Definição de Funções	4
4	Análise de Usabilidade e Eficiência	4
4.1	Facilidade de Aprendizado e Uso	4
4.2	Desempenho em Execução	4
4.3	Extensibilidade e Integração	5
4.4	Usabilidade em Diferentes Domínios	5
5	Pontos Fortes e Fracos	5
6	Conclusão	6

1 Introdução

Este relatório apresenta os resultados do estudo sobre a linguagem de programação Lua, conforme solicitado no Trabalho Prático 1 da disciplina DCC024 - Linguagem de Programação, ministrada pelo professor Marcio Costa Santos. A linguagem Lua foi sorteada para nosso grupo, e neste documento exploraremos seu histórico, características principais, usabilidade, eficiência, pontos fortes e fracos. Além disso, desenvolvemos dois códigos na linguagem: um de busca binária e um de multiplicação de matrizes, ambos disponíveis no GitHub. Também criamos um vídeo sobre a linguagem, cujo link está disponível neste relatório.

Os links para os códigos no GitHub[1] e para o vídeo[2] podem ser encontrados nas referências ao final do documento.

2 Histórico da Linguagem Lua

A linguagem Lua foi concebida em 1993 por Roberto Ierusalimsky, Luiz Henrique de Figueiredo e Waldemar Celes, membros do Computer Graphics Technology Group na PUC-Rio. Na época, o Brasil enfrentava desafios econômicos significativos, e a criação de uma linguagem de programação inovadora no país foi um marco importante. A PUC-Rio, uma instituição de ensino superior de renome, desempenhou um papel crucial ao proporcionar o ambiente acadêmico necessário para o desenvolvimento de Lua.

A princípio, a linguagem foi criada para ser usada em um projeto da Petrobrás. Devido à sua eficiência, clareza e facilidade de uso, passou a ser utilizada em várias aplicações da programação, como o desenvolvimento de jogos, controle de robôs, processamento textual, entre outros [4].

O uso da linguagem Lua entrou no mundo dos jogos com o lançamento de Grim Fandango em 1998. Descoberta por um funcionário da Lucas Arts através de um artigo na revista americana Dr Dobbs's, Lua foi gradualmente adotada em projetos internos e logo se tornou uma escolha comum em videogames de sucesso como World of Warcraft, Angry Birds e Roblox, o que popularizou a linguagem. Sua versatilidade não se limitou apenas aos jogos, sendo também utilizada em programas como Photoshop e Lightroom. Vale ressaltar que Lua é uma das poucas linguagens criadas em um país em desenvolvimento a ser usada globalmente, o que rendeu a seus criadores uma medalha Pedro Ernesto, destacando sua contribuição significativa para a indústria tecnológica [5].

Atualmente, Lua continua a ser amplamente utilizada e mantida. A versão mais recente, Lua 5.4, lançada em 2020, trouxe várias melhorias e novas funcionalidades, como coletores de lixo incrementais e constantes to-be-closed. A comunidade de desenvolvedores de Lua permanece ativa, contribuindo com bibliotecas e ferramentas, garantindo que a linguagem continue relevante e suportada para diversas aplicações modernas.

3 Características Básicas da Linguagem

Lua é uma linguagem interpretada, o que significa que não são criados arquivos executáveis diretamente, mas suas instruções são executadas por meio de outros programas, conhecidos como interpretadores. Essa característica torna a Lua uma linguagem de extensão, pois ela geralmente é incorporada em outros programas que a utilizam. Essa

flexibilidade permite que a Lua seja integrada de forma eficiente em uma variedade de aplicativos e ambientes de desenvolvimento.

Ela também se caracteriza por ser do tipo dinâmico, atribuindo automaticamente um tipo a uma variável, como um número, um texto ou uma estrutura de dados. O gerenciamento de memória é automatizado, liberando o programador da preocupação com a alocação e desalocação de memória [6].

Além das características mencionadas, Lua atrai programadores por várias outras vantagens. Reconhecida por sua robustez, é amplamente empregada em uma gama diversificada de aplicações industriais, sistemas embarcados e jogos. Sua velocidade e desempenho a tornam uma escolha comum em várias áreas. Lua é altamente portátil, compilando facilmente em diversas plataformas e dispositivos. Sua facilidade de incorporação em outras aplicações, com uma API simples e bem documentada, permite uma integração eficiente com outras linguagens. Além disso, sua poderosa e flexível natureza oferece meta-mecanismos para implementar construções, mantendo-se compacta em tamanho. E como se isso não bastasse, Lua é uma opção gratuita e de código aberto [3].

Lua possui uma sintaxe simples e clara, semelhante a outras linguagens de script como Python. A seguir, apresentamos algumas das estruturas básicas de Lua, incluindo controle de fluxo, definição de funções e declaração de variáveis.

3.1 Estruturas Básicas

3.1.1 Declaração de Variáveis e Tipos

Em Lua, as variáveis são declaradas usando a palavra-chave ‘local’, que limita o escopo da variável ao bloco onde foi declarada, evitando interferências acidentais com outras partes do programa. Ademais, nota-se, como citado anteriormente, que não é necessário declarar os tipos das variáveis.

```
local numero = 10
local texto = "Olá, mundo"
local tabela = {1, 2, 3}
```

3.1.2 Controle de Fluxo

O controle de fluxo em Lua inclui estruturas ‘if’, ‘for’, e ‘while’, que permitem a execução condicional e a repetição de blocos de código.

Estrutura Condicional ‘if’

A estrutura ‘if’ é usada para executar um bloco de código se uma condição for verdadeira. Ela pode incluir as palavras-chave ‘elseif’ e ‘else’ para tratar múltiplas condições:

```
local numero = 5

if numero > 0 then
    print("O número é positivo")
elseif numero < 0 then
    print("O número é negativo")
else
```

```
    print("0 número é zero")
end
```

- 'if' inicia a condição.
- 'then' indica o início do bloco de código a ser executado.
- 'elseif' adiciona condições adicionais.
- 'else' define o bloco a ser executado se nenhuma das condições anteriores for verdadeira.
- 'end' finaliza a estrutura condicional.

Laço 'for'

O laço 'for' em Lua é usado para repetir um bloco de código um número específico de vezes. Ele usa a palavra-chave 'for', seguida de um contador, a faixa de valores e a palavra-chave 'do':

```
for i = 1, 5 do
    print(i)
end
```

- 'for' inicia o laço.
- 'i' é o contador que varia de '1' a '5'.
- 'do' indica o início do bloco de código a ser repetido.
- 'end' finaliza o laço.

Laço 'while'

O laço 'while' repete um bloco de código enquanto uma condição for verdadeira. Ele usa a palavra-chave 'while', seguida da condição e da palavra-chave 'do':

```
local contador = 1
while contador <= 5 do
    print(contador)
    contador = contador + 1
end
```

- 'while' inicia o laço.
- 'contador <= 5' é a condição a ser verificada.
- 'do' indica o início do bloco de código a ser repetido.
- 'end' finaliza o laço.

3.1.3 Definição de Funções

Funções em Lua são definidas com a palavra-chave ‘function’, permitindo a criação de blocos reutilizáveis de código. Uma função pode receber parâmetros e retornar valores:

```
local function saudacao(nome)
    return "Olá, " .. nome
end

print(saudacao("Mundo"))
```

- ‘function’ inicia a definição da função.
- ‘saudacao(nome)’ define o nome da função e seus parâmetros.
- ‘return’ especifica o valor a ser retornado pela função.
- ‘end’ finaliza a definição da função.

Lua é uma linguagem extensível, permitindo a criação de bibliotecas e módulos personalizados. A linguagem suporta programação funcional e orientada a objetos, embora não seja uma linguagem orientada a objetos pura. Além disso, Lua é leve e rápida, adequada para sistemas embarcados e aplicações onde o desempenho é crítico.

Nessa seção, detalhamos as características fundamentais de Lua, abordando sua sintaxe, tipagem dinâmica, uso como linguagem de script e sua extensibilidade, que tem sido um dos principais atrativos da linguagem.

4 Análise de Usabilidade e Eficiência

Devido à simplicidade e flexibilidade da sintaxe, a linguagem Lua é acessível para programadores iniciantes e experientes. Ela é conhecida por sua eficiência e por ter uma das implementações mais rápidas entre as linguagens interpretadas.

4.1 Facilidade de Aprendizado e Uso

Lua possui uma curva de aprendizado suave, graças à sua sintaxe simples e clara. Programadores iniciantes podem rapidamente se familiarizar com os conceitos básicos, enquanto programadores experientes podem se beneficiar da sua flexibilidade. A documentação oficial de Lua é abrangente e bem estruturada, proporcionando uma excelente referência para desenvolvedores de todos os níveis [6].

4.2 Desempenho em Execução

Lua é conhecida por sua alta eficiência e desempenho. A implementação padrão de Lua, chamada PUC-Rio Lua, é otimizada para velocidade e uso mínimo de memória. Em benchmarks, Lua frequentemente apresenta desempenho superior a outras linguagens interpretadas, como Python e Ruby.

Em uma comparação feita entre a linguagem Lua e Python em [7], observa-se que Lua supera Python em diversos casos em termos de tempo de execução. Lua também tende a consumir uma quantidade de memória similar ou menor que Python na maioria

dos testes. No entanto, há cenários específicos em que Python apresenta desempenho superior ou equivalente, destacando a eficiência de ambas as linguagens em diferentes tipos de tarefas.

4.3 Extensibilidade e Integração

Lua é altamente extensível e pode ser integrada facilmente em sistemas maiores. Sua API em C bem documentada permite que Lua seja embutida em outras aplicações, proporcionando uma linguagem de script poderosa para extensões e personalizações. Exemplos notáveis de integração incluem o uso de Lua em engines de jogos e o sistema de automação doméstica NodeMCU.

4.4 Usabilidade em Diferentes Domínios

Lua é amplamente utilizada em diversos domínios devido à sua versatilidade:

- **Desenvolvimento de Jogos:** Lua é popular em engines de jogos devido à sua velocidade e facilidade de integração.
- **Sistemas Embarcados:** A eficiência de Lua em termos de uso de memória torna-a ideal para sistemas embarcados.
- **Automação e Scripting:** Lua é frequentemente usada em scripts de automação e para adicionar funcionalidades personalizadas em software existente.

Lua continua a ser uma escolha sólida para desenvolvedores que procuram uma linguagem leve, rápida e altamente extensível, adequada para uma ampla gama de aplicações.

5 Pontos Fortes e Fracos

Os pontos fortes da linguagem Lua são suas características planejadas desde sua concepção, guiados por princípios fundamentais da linguagem. Um exemplo é sua notável simplicidade, refletida em uma sintaxe clara e acessível, sendo frequentemente comparada com Python. Em termos de eficiência, Lua foi projetada para oferecer desempenho e utilização de recursos otimizados, tornando-se uma escolha ideal para uma variedade de aplicações. A portabilidade é outra característica proeminente, graças à sua API em C bem definida, que facilita a integração em projetos maiores. Além disso, Lua se destaca pela velocidade, resultado de uma implementação altamente otimizada pelos seus criadores. [6]

Por outro lado, um dos pontos desafiadores da Lua está diretamente ligado à sua menor popularidade em comparação com outras linguagens mais estabelecidas. Como resultado, a linguagem enfrenta a falta de uma comunidade de apoio robusta e uma ampla variedade de bibliotecas disponíveis para os desenvolvedores. Exploramos os principais aspectos que fazem da Lua uma escolha preferida para determinados projetos, incluindo sua portabilidade, leveza e flexibilidade.

6 Conclusão

Neste relatório, exploramos a linguagem de programação Lua, abordando seu histórico, características principais, usabilidade, eficiência e pontos fortes e fracos. Constatamos que Lua é uma linguagem leve, eficiente e altamente extensível, ideal para uma ampla gama de aplicações, incluindo desenvolvimento de jogos e sistemas embarcados. Sua sintaxe simples e clara facilita o aprendizado e o uso tanto por programadores iniciantes quanto experientes.

Por outro lado, também ressaltamos os desafios relacionados à menor popularidade de Lua, resultando em uma comunidade de apoio e uma variedade de bibliotecas menos robustas em comparação com outras linguagens mais estabelecidas. No entanto, suas vantagens em termos de portabilidade, leveza e extensibilidade continuam a torná-la uma escolha atraente para muitos projetos.

Em suma, Lua se destaca como uma linguagem poderosa e versátil, capaz de atender a diversas necessidades de programação, ao mesmo tempo em que oferece um desempenho eficiente e uma integração fácil em sistemas maiores.

Referências

- [1] GitHub. *Códigos de Busca Binária e Multiplicação de Matrizes em Lua*. Disponível em: <https://github.com/Rubia-Souza/UFMG-LP-Sailor-Moon>.
- [2] YouTube. *Vídeo de Apresentação sobre a Linguagem Lua*. Disponível em: <https://youtube.com/>.
- [3] Lua.org. *The Official Website of the Lua Language*. Disponível em: <https://www.lua.org>. Acesso em: 30 de março de 2024.
- [4] Wikipedia. *Lua (programming language)*. Disponível em: [https://en.wikipedia.org/wiki/Lua_\(programming_language\)](https://en.wikipedia.org/wiki/Lua_(programming_language)). Acesso em: 30 de março de 2024.
- [5] Jornal PUC. *Lua (Origem e História)*. Disponível em: <http://jornaldapuc.vrc.puc-rio.br/cgi/cgilua.exe/sys/start.htm?infoid=14419&sid=29>. Acesso em: 01 de Maio de 2024.
- [6] Edu.gfc. *Lua (Características)*. Disponível em: <https://edu.gcfglobal.org/pt/conhecimentos-tecnologicos/lua-a-nova-linguagem-de-programacao/1/>. Acesso em: 14 de Maio de 2024.
- [7] The Computer Language Benchmarks Game. *Lua versus Python 3 fastest performance*. Disponível em: <https://benchmarksgame-team.pages.debian.net/benchmarksgame/fastest/python3-lua.html>. Acesso em: 15 de Maio de 2024.