



Bequem Scripten

mit Lua und LRocket

Über uns



Sewbacca

„Just a guy who likes Lua and C“

<https://codeberg.org/sewbacca>

🎓 Computer Science



leso-kn

„Design ideas, Talkative“

<https://leso.dev>

🎓 Information Science

Über uns



Sewbacca

„Just a guy who likes Lua and C“

<https://codeberg.org/sewbacca>

🎓 Computer Science



leso-kn

„Design ideas, Talkative“

<https://leso.dev>

🎓 Information Science

Irc

- Ein Compiler für die interpretierte Sprache **Lua**
- Entstanden aus einem Bundler-Nebenprojekt und der Idee QoL-Features von Go nach Lua zu bringen
- Unterstützt Executables, Shared-Libraries, Lua-Bundles und **WebAssembly**
- Open-Source (MIT): <https://codeberg.org/Irocket>

Ähnliche Projekte

- [luastatic](https://github.com/ers35/luastatic) <https://github.com/ers35/luastatic>
- [luapak](https://github.com/jirutka/luapak) <https://github.com/jirutka/luapak>
- [srlua](https://github.com/LuaDist/srlua) <https://github.com/LuaDist/srlua> (archived)

Unterscheidungen bei lrc

- Semantische Verarbeitung von Dependencies & Static Linking
- Tiefe Integration in [luarocks](#)
- Einbetten beliebiger Ressourcen-Dateien
- (bald) Cross-Compiling

Anwendung

Lua kompillieren! *Wie geht das?*



Hello World

```
-- main.lua  
print 'hello world!'
```

```
> lrc main.lua  
> ./main  
hello world!
```

- Programm kompiliert über C nach Executable
- Die Lua-Runtime ist im kompilierten Programm enthalten
 - Wahlweise kann `liblua.so` auch dynamisch verlinkt werden

Bibliotheken

```
-- main.lua  
local sfs = require 'sfs'  
print(sfs.read 'datei.txt')
```

```
> lrc main.lua  
Found Lua modules  
    sfs  
Found C modules  
    lfs  
  
> ./main  
[ Datei-Inhalt ]
```

- lrc findet Abhängigkeiten des Programms rekursiv
- C-Module können statisch einkompiliert werden
 - In diesem Fall werden Systembibliotheken dynamisch verlinkt

Luarocks Build-System

```
> luarocks make
demo scm-0 depends on sfs ~> 0.2
(0.2.0-1 installed: success)
```

```
main.lua => bin/main
```

```
Found Lua modules
```

```
    sfs
```

```
Found Native modules
```

```
    lfs (~/.luarocks/.../lfs.a)
```

```
> demo
```

```
[ Datei-Inhalt ]
```

```
-- demo-scm-0.rockspec
build = {
  type = 'lua',
  entrypoint = 'main.lua',
  output = 'bin/demo'
}
dependencies = {
  'sfs ~> 0.2'
}
```

- Durch einen Hook installiert
Luarocks native Dependencies ab
jetzt dual als (.so) und (.a)

- demo installiert nach ~/.luarocks/bin

Einbetten beliebiger Dateien

```
-- main.lua
local sfs = require 'sfs'

--lr:embed 'datei.txt'
print(sfs.read 'datei.txt')
```

- Inspiriert durch `//go:embed`
- Suchpfad-Integration in `datafile` ermöglicht Einbettung ohne Namenskonflikte

```
> luarocks make
demo scm-0 depends on sfs ~> 0.2
(0.2.0-1 installed: success)
```

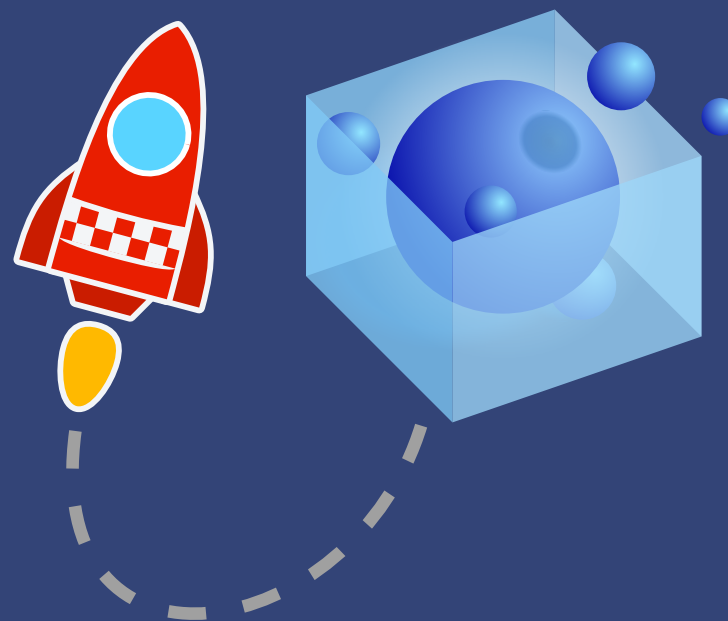
```
main.lua => bin/main
Found Lua modules
    sfs
```

```
Found Native modules
    lfs (~/.luarocks/.../lfs.a)
```

```
Found Resource files
    @<main>:datei.txt (./datei.txt)
    datei.txt (./datei.txt)
```

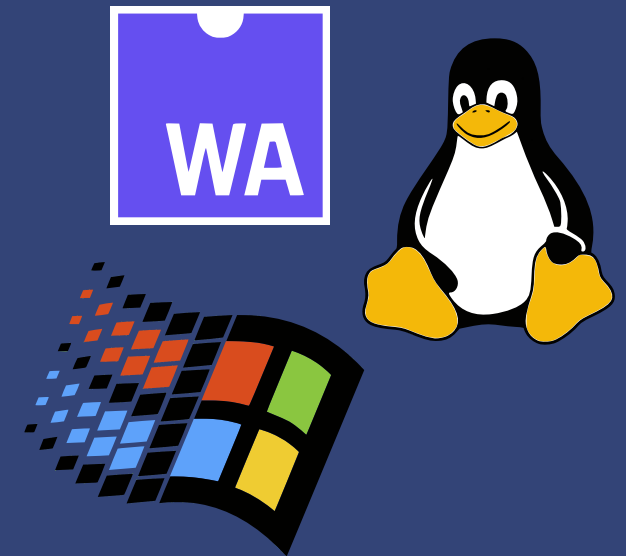
Demo Time!

3 Beispielprojekte



Aktueller Stand

- Unterstützte Plattformen: **Linux**, **Windows** und **WebAssembly**
 - WebAssembly Targets: `wasm`, `js` und `html`
- Dependency Tracing
- **Luarocks** Build Integration
- Dynamic / Static Linking
- **Einbettung von Ressourcen-Dateien**



[5] https://asd-tech.mx/wp-content/uploads/2020/11/0_YicgJSe-ALzdFwHO.png

[6] https://images.seeklogo.com/logo-png/48/1/linux-logo-png_seeklogo-482333.png

[7] Windows Logo by Microsoft · <https://www.pngall.com/wp-content/uploads/2/Windows-Logo-PNG-File-Download-Free.png>

Feature-Roadmap

- **1.0.0** [2024-04-01] – Erstveröffentlichung
- **1.0.2** [2025-06-02] – Aktuelle Release
- **1.1.0** [2025-06-27] – **Einbettung von Ressourcen-Dateien**
- **1.2.0** – *Cross-Compiling*
- **1.3.0** – *QOL / Optimierungs-Release (DWARF/PE-Sections)*

:wqa

- Projektseite: codeberg.org/lrocket 🚀
- Dokumentation: lrocket.codeberg.page 👁️
- ★ = ★ + 1



Dankeschön! :) =)

Kontakt: codeberg.org/sewbacca | [matrix] l:leso.dev

