

# Lua Cookbook

By Lua Cookbook authors

## Lua Cookbook Table of Contents

### Installing Lua on Linux

Author: Alexander Gladysh [agladysh@gmail.com](mailto:agladysh@gmail.com)

Stub article

#### Installing from source

You'll need `libreadline`.

~~~[console] \$ wget http://www.lua.org/ftp/lua-5.1.4.tar.gz \$ tar -zxvf lua-5.1.4.tar.gz \$ cd lua-5.1.4.tar.gz \$~~

Check this

#### Ubuntu

To install Lua on Ubuntu:

~~~[console] \$ sudo apt-get install lua~~~

Note about LuaRocks incompatibilities (if any).

Add more Linux flavors here.

### Networking with LuaSocket

Author: Matthew Frazier [leafstormrush@gmail.com](mailto:leafstormrush@gmail.com)

The most common library for networking in Lua is LuaSocket by Diego Nehab. In addition to low-level support for communicating directly through sockets, it also includes:

- HTTP client
- FTP client
- SMTP client

- Mail processing filters
- URL manipulating functions

LuaSocket is installable via LuaRocks, as explained in :ref:‘luarocks’. Many distributions also provide LuaSocket packages.

Fix references

## Socket-Level Programming

To create a TCP socket <sup>1</sup> connected to a particular place, you can use `socket.connect`, like this:

```
require 'socket'
local sock = socket.connect("www.example.com", 80)
```

From there, you can send and receive on the socket object.

```
sock:send("GET / HTTP/1.0\n\n")
local data = sock:recv()
```

You could also use `socket.tcp("www.example.com", 80)` and then `connect` it separately, but this is a shortcut.

## Servers

Servers are *cool!* According to some guy we met:

Servers are the whole reason the Internet exists as we know it.

You can create servers by using `socket.bind`.

Here is a graph about servers.

## Concurrency

Everyone is talking about concurrency these days!

Some of the ways people have thought of for being concurrent are:

1. Multithreading.

---

<sup>1</sup>There’s also UDP, but we’re not covering it in this introduction.

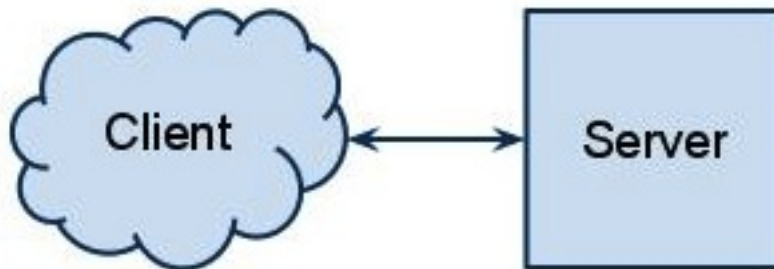


Figure 1: image

2. Having multiple processes.
3. Asynchronous programming.

Fortunately, Lua has something cool you can use for number three: Coroutines! (If you don't know about coroutines, see :ref:'using-coroutines' for more info.)

## Glossary

**client** The program run by the user that talks to clients.

**server** The program that clients talk to.

**TCP** What most people use nowadays instead of UDP.

## Lua Cookbook Alphabetic Index

Lua Cookbook is cool, everyone should read it — anonymous