History of the development of Lua

daurnimator

Thursday, October 13, 2016
About this talk
Conceived in 1993 as successor to SOL
Conceived in 1993 as successor to SOL
PhD project by Waldemar Celes, supervised by Roberto Ierusalimschy
The start of Lua

- Conceived in 1993 as successor to SOL
- PhD project by Waldemar Celes, supervised by Roberto Ierusalimschy
- Luiz Henrique de Figueiredo (lhf) helps out
Development model

- Done in private
Development model

- Done in private
- No release schedule
Development model

- Done in private
- No release schedule
- The public’s interaction with team is primarily via mailing list
Development model

- Done in private
- No release schedule
- The public’s interaction with team is primarily via mailing list
- Patches not accepted as is: usually rewritten
Before there was git there was svn (Subversion)
Before there was git there was svn (Subversion)
Before there was svn there was CVS
Before there was git there was svn (Subversion)
Before there was svn there was CVS
Before there was CVS there was RCS
Version Control System

- Before there was git there was svn (Subversion)
- Before there was svn there was CVS
- Before there was CVS there was RCS
- Since 1993 the Lua team has used RCS for development.
Before there was git there was svn (Subversion)
Before there was svn there was CVS
Before there was CVS there was RCS
Since 1993 the Lua team has used RCS for development.
  Not used from the very beginning
Before there was git there was svn (Subversion)
Before there was svn there was CVS
Before there was CVS there was RCS
Since 1993 the Lua team has used RCS for development.
  Not used from the very beginning
  Hence there is some history lost to the sands of time
About RCS

- RCS is not a distributed version control system.
About RCS

- RCS is not a distributed version control system.
- RCS is not even a collaborative version control system.
About RCS

- RCS is not a distributed version control system.
- RCS is not even a collaborative version control system.
- Work has been split up into separate repositories.
About RCS

- RCS is not a distributed version control system.
- RCS is not even a collaborative version control system.
- Work has been split up into separate repositories
- RCS tracks individual files, not directories
About RCS

- RCS is not a distributed version control system.
- RCS is not even a collaborative version control system.
- Work has been split up into separate repositories
- RCS tracks individual files, not directories
  - if multiple files are changed at once, you need to coalesce together
About RCS

- RCS is not a distributed version control system.
- RCS is not even a collaborative version control system.
- Work has been split up into separate repositories
- RCS tracks individual files, not directories
  - if multiple files are changed at once, you need to coalesce together
  - no tracking of file deletion
Repository structure

There are multiple ‘top level’ repositories

- oldlua: lua code up to 1997
There are multiple ‘top level’ repositories

- oldlua: lua code up to 1997
- lua: the code from 1997 until now (maintained by Roberto)
There are multiple ‘top level’ repositories

- oldlua: lua code up to 1997
- lua: the code from 1997 until now (maintained by Roberto)
- oldluac: luac code up till 1998
There are multiple ‘top level’ repositories

- oldlua: lua code up to 1997
- lua: the code from 1997 until now (maintained by Roberto)
- oldluac: luac code up till 1998
- luac: luac code and bytecode experiments + other things (maintained by Luiz)
There are multiple ‘top level’ repositories

- oldlua: lua code up to 1997
- lua: the code from 1997 until now (maintained by Roberto)
- oldluac: luac code up till 1998
- luac: luac code and bytecode experiments + other things (maintained by Luiz)
- tests: the lua test suite (maintained by Luiz)
Repository structure

There are multiple ‘top level’ repositories

- oldlua: lua code up to 1997
- lua: the code from 1997 until now (maintained by Roberto)
- oldluac: luac code up till 1998
- luac: luac code and bytecode experiments + other things (maintained by Luiz)
- tests: the lua test suite (maintained by Luiz)
- manual: the manual as a single file in custom format (maintained by Roberto)
But that’s not what a release looks like

Inside of lua-5.3.3.tar.gz:

doc/contents.html
doc/index.css
doc/logo.gif
doc/lua.1
doc/luac.1
doc/lua.css
doc/manual.css
doc/manual.html
doc/osi-certified-72x60.png
doc/readme.html
src/l*.{c,h}
src/Makefile
Makefile
README
How is a release made?

- Manually by Luiz:
How is a release made?

- Manually by Luiz:
  - copies in Makefile, README and doc/ folder contents from previous release
  - copies contents of lua repository from Roberto to src/
  - compiles current version of luac.c and luac.1 (man page)
Manually by Luiz:
- copies in Makefile, README and doc/ folder contents from previous release
- copies contents of lua repository from Roberto to src/
Manually by Luiz:
- copies in Makefile, README and doc/ folder contents from previous release
- copies contents of lua repository from Roberto to src/
- copies in current version of luac.c and luac.1 (man page)
How is a release made?

- Manually by Luiz:
  - copies in Makefile, README and doc/ folder contents from previous release
  - copies contents of lua repository from Roberto to src/
  - copies in current version of luac.c and luac.1 (man page)
  - compiles current version of manual, outputs to doc/ folder
Converting to git

- CVS is based on RCS, and uses the same file format(s)
Converting to git

- CVS is based on RCS, and uses the same file format(s)
- This means that I could use CVS tools to work with the RCS files
Converting to git

- CVS is based on RCS, and uses the same file format(s)
- This means that I could use CVS tools to work with the RCS files
- consistency: want to be able to get the same result when run multiple times
Converting to git

- CVS is based on RCS, and uses the same file format(s)
- This means that I could use CVS tools to work with the RCS files
- consistency: want to be able to get the same result when run multiple times
  - want to be able to pull across new changes from Roberto
cvs-fast-export is a tool that converts the CVS/RCS format to a git fast-import stream
cvs-fast-export is a tool that converts the CVS/RCS format to a git fast-import stream
reposurgeon is a tool that works on fast import streams
Tools

- `cvs-fast-export` is a tool that converts the CVS/RCS format to a git fast-import stream.
- `reposurgeon` is a tool that works on fast import streams.
- `authormap` file: maps from username to name + email + timezone.
In 2003 Luiz managed to dig up an old snapshot of lua code, this was released as a historical curiosity and dubbed Lua 1.0.
In 2003 Luiz managed to dig up an old snapshot of lua code, this was released as a historical curiosity and dubbed Lua 1.0. Without the release notes added in 2003, this serves as our initial commit.
Decisions

- What git repositories should there be?
Decisions

- What git repositories should there be?
  - should the format match the releases?
What git repositories should there be?

- should the format match the releases?
  - no. that would be throwing out data
Decisions

- What git repositories should there be?
  - should the format match the releases?
    - no. that would be throwing out data
  - keep luac separate?
Decisions

- What git repositories should there be?
  - should the format match the releases?
    - no. that would be throwing out data
  - keep luac separate?
    - yes. the branches and tags don’t match up with normal repository
Decisions

- What git repositories should there be?
  - should the format match the releases?
    - no. that would be throwing out data
  - keep luac separate?
    - yes. the branches and tags don’t match up with normal repository

- The division between old and new
First few commit messages are in Portuguese.
Notable things

- First few commit messages are in Portuguese.
- Roberto keeps a ‘bugs’ file that tracks bugs + fixes
Notable things

- First few commit messages are in Portuguese.
- Roberto keeps a ‘bugs’ file that tracks bugs + fixes
- Many lines of code remain unchanged for decades
After the workshop the sources will be available for everyone. https://github.com/lua
This is not an indication that Lua’s development model has changed. However, if you do maintain a project that forks or patches Lua, I invite you to make your fork from here.
Related resources:

- The evolution of an extension language: a history of Lua
Related resources:

- The evolution of an extension language: a history of Lua
- The Evolution of Lua (Paper)
Related resources:

- The evolution of an extension language: a history of Lua
- The Evolution of Lua (Paper)
- The Evolution of Lua (Workshop talk from 2006)