



Lua In Embedded Linux

Glenn Edgar

Onyx Engineering, Inc.

Lua in San Diego

- Lua is or will be used in several embedded projects. Majority of these projects are Set Top related.
- The purpose of this presentation is to share our work and goals.

Current Use of Lua

- Lua is embedded in the embedded application and lua runs as a thread or task in the application. Lua has direct access to the embedded system.
- Lua connects remotely, (serial or TCP/IP) to remote user interfaces – hyperterm or telnet client.
- A modest shell interface is written to command line history and interactive command editing. (This is an important feature)

Lua Environment Continued

- C extensions tie the lua interpreter to hardware and O/S commands
- Lua Scripts are loaded at runtime and transferred from remote connection.
- The C extensions allow various levels of the driver software to be tested.
- Lua Scripts set up the environment for testing as well as provide extensive unit testing capability.

Political Problems

- Very hard to politically to incorporate scripting language in embedded system. (Even when project is bleeding from lack of testing.)
- May be very successful in one generation and taken out in the next.
- A NASA experience is documented in the following URL's (LISP SCRIPTING)
 - <http://www.flownet.com/gat/jpl-lisp.html>
 - <http://lemonodor.com/archives/000047.html>

Solution

- Lua needs to be a default tool for embedded systems.
 - Removes the political decision.
- In Embedded Linux there is a way easily do this
 - Incorporate Lua into Busybox
 - Replace the default Ash Shell with the Lua Interpreter as Command Shell

Added Benefit

- Embedded Applications are being written in scripting languages as seen from cover story in the Linux Journal.



Busy Box Economics

- Lua as shell language makes sense from memory footprint
 - Ash Shell 60K
 - Lua 130K -- (MIPS processor)
 - Full Bash Shell 900K
- With Lua's Posix , TCP/IP and string pattern matching libraries, the Busy Box environment should be smaller and more modular.

Lua Busy Box Environment

- Lua Scripting Speed
 - Will not have to create and spawn process for every command.
- Integration with Hardware
 - Shared Libraries allow hardware functions to be added to the command shell
 - Important for factory test and other applications.

Future Results

- Embedded Applications will no longer be monolithic.
- Instead embedded functions will be written as shared libraries configured by scripting languages.
 - Same code could be used for production as well as factory test
- ARINC 653 systems are showing early signs of this evolution.

Current Status

- Need to define and implement features necessary for a Lua shell language.
- Need to find applications for Lua Busy Box implementation.

Lua as a Shell Language

- Not a shell expert -- need guidance
- Need help in defining what a Lua shell language would look like.
 - Important that current shell users find it comfortable
- Need to Incorporate Streaming Capability
- Do we use the base environment or do we create a layer on top of Lua?

Need to Find Applications

- Would like to get a dialog started on incorporating Lua into an Embedded Linux distribution.
- Will happily share the work and ideas in this area to any potential users