Google Summer of Code 2016

LabLua
IO API for NetBSD kernel Lua

Guilherme Salazar
Instituto de Informática
Universidade Federal de Goiás

April 20, 2016

This is a proposal to the LabLua Google Summer of Code 2016 organization; the idea of choice is to *Develop an IO API for NetBSD Kernel Lua.* Section Basics contains the applicant's personal information, Section Experience approaches his experience, Section Project encompasses his project proposal, and GSoC is about his participation in previous GSoC editions.

1 Basics

1.1 What is your preferred email address?

gmesalazar@gmail.com

1.2 What is your Web Page, Blog, and GitHub?

`https://github.com/gmesalazar`

1.3 What is your academic background?

CV in EN-US: `http://inf.ufg.br/~guilhermesilva/arquivos/cv.pdf`
CV in PT-BR: `http://lattes.cnpq.br/5243220898815496`

1.4 What other time commitments, such as school work, another job (GSoC is a full-time activity), or planned vacations will you have during the period of GSoC?

My final academic semester is scheduled to begin on March 30 and end on July 29; I'm enrolled to take the remaining course to complete for graduation. At the end, I will present my senior project, which will not require much time commitment. I do not have other formal commitments or planned vacations.

2 Experience

2.1 What programming languages are you fluent in? Which tools do you normally use for development?

I code mostly in C, Lua, and Java, but I have implementation-specific experience with Python and a few other languages—please check my CV for more. I work with GNU/Linux—currently Arch Linux, but I've used Debian and Ubuntu for quite some time. I use Git for revision control, have used

Subversion, and am learning CVS. I have used Eclipse and Android Studio for side projects. I use Vim for most of my editing—used Emacs for some time. Other tools I commonly use are GNU Make, GCC, GDB, and others in the GNU toolchain. I have recently started using Vagrant to build and manage my development environments and am loving it.

2.2 Are you familiar with the Lua programming language? Have you developed any projects using Lua?

I was introduced to Lua a few years ago (in 2011) in a research project and implemented a few programs using the language—nothing too complex, most of them were to teach myself about the language. I've had a short experience with ALua, Lua's framework for event-driven parallel and distributed programming. As a LabLua GSoC 2015 student, I worked on porting Lua test suite to the NetBSD kernel; at the time, I ported the test suite for Lua 5.3.0. I'm now improving the port; test scripts were already adapted to the 5.3.2 test suite and I'm polishing the kernel modules. I hope to be able to commit the port to NetBSD in the coming month or so.

2.3 Have you developed software in a team environment before? Any projects with actual users?

All projects I participated in so far were developed in small groups, in which my work was not largely affected by others' work.

2.4 What kinds of projects/software have you worked on previously? (anything larger than a class project: academic research, internships, freelance, hobby projects, etc). In particular, are you (or have you been) involved with any open source development project? If so, briefly describe the project and the scope of your involvement.

Development of a Google App Engine EMR (Electronic Medical Records) application for the Medical School of the University of South Carolina (while I was at USC as an exchange student). The backend was developed with GAE for Python; the frontend was based off of Twitter Bootstrap. The frontend part was kind of challenging; I lack the skills needed to design 'good' graphical interfaces.

Development of an Android app which helps couriers to manage their workflow, keeping a list of packages they have to deliver along with their respective information (e.g., sender, address, etc). The app integrates with an existing software—the allocation of packages is done there and the app retrieves the list by means of a RESTful interface. It has grown considerably— it has as of now about 11k lines of code—which does not mean a lot per se.

I've also worked on last year's GSoC, LabLua. After that, I got accepted as a NetBSD developer and am working to bring kernel Lua test suite to the project.

3 Project

3.1 Did you select a project from our list? If yes, which project did you select? Why did you choose this project? If you are proposing a project, give a description of your proposal, including the expected results.

The project I chose from the list is to *Develop an IO API for NetBSD Kernel Lua*; in GSoC 2015, I ported parts of the *io* standard library to the NetBSD kernel: *open*, *close*, *input*, *output*, *write*, and *read* were ported in a way that preserves much of the userspace behavior (though improvements can still be made). Lua base *dofile* and *loadfile* functions were also ported. Not closely related to the goals of the project herein proposed are bits of the *math* and *os* libraries that were also ported. Refer to [1] for the kernel test suite.

The experience acquired with last year's project would be of utmost relevance to the current projct, especially that which is result of the work in the *io* kernel Lua library and the underlying *iolib*, which handles the underlying use of kernel APIs. The idea proposed by Lourival in [3] is to implement an IO API that allows NetBSD kernel Lua port to access filesystem and sockets capabilities provided by the NetBSD kernel. My proposal to accomplish these goals is structured as follows:

- Implementation of full-support in kernel space to Lua standard library *io*. This front of work would deal with porting the existing *io* standard library to kernel space; part of the work done porting the test suite [1] (and the experience therein acquired) would be a valuable starting point in this task. The *iolib* library [2] that was developed in that occasion to provide a layer on top of which *io* could be implemented covers basic operations, such as opening, closing, reading, and writing to files. It does not provide access, for instance, to buffering capabilities, as required in certain functions of the *io* library (e.g., *io.flush* and *io.setvbuf*). I propose doing this work in-place in the *io* library source code, using pre-compiler directives where appropriate, instead of building an independent kernel module; doing so would avoid the additional *modload* and a *require* on the module, and also avoid repetition of code.

- Implementation of a sockets library to NetBSD kernel Lua. The goal of this front of work would be to implement a loadable kernel module to provide a sockets API to kernel Lua. I propose setting the scope of the implementation in terms of "domain" and "type", as follows:

  - Domain: relative to the domain within which the communication is to take place
    - local-domain sockets
    - inet (IPv4) sockets
    - inet6 (IPv6) sockets
  - Type: the semantics of communication
    - stream-oriented communication
    - datagram-oriented communication

  The library would provide the following API:

  - *socket (domain, type)*: creates a socket and returns a file descriptor
  - *bind (socket, sockaddr)*: binds a socket to the given address
  - *listen (socket, backlog)*: puts a socket to accept connections and sets the maximum size of its waiting queue
  - *accept (socket, sockaddr)*: accepts a connection on a socket
  - *connect (socket, sockaddr)*: connects 'socket' to an endpoint specified by 'sockaddr'
  - *send (socket, buff)*: writes the contents of 'buff' to 'socket'
  - *read (socket, buff, sz)*: reads 'sz' bytes from the socket
  - *shutdown (socket, how)*: shuts down the connection of 'socket'; stop receiving or transmitting, according to 'how'

  This front of work would require significantly more work than the first; as such, a larger window of development should be allocated to its implementation and tests.

3.2 Please provide a schedule with dates and important milestones/deliverables (preferably in two week increments).

Below is a rough tentative schedule, stating from a pre-announcement period, which is not entirely related to this proposal.

- Pre-announcement (until April 22): continue work on the test suite and NetBSD; I'm integrating the kernel modules developed in GSoC 2015 into Lua implementation. The way it is done may have implications on how the first front of work of this proposal would be accomplished, since it involves Lua *io* library

- Community bonding (April 23 - May 23): investigate kernel APIs that would be used in the implementation and do simple experiments to study their behavior; start working on the *io* library. The deliverable must be, at the very least, a report containing a description of the kernel APIs studied and where/how they fit in the libraries to be developed
- *IO* library (May 24 - Jun 7): by the end of this time frame, Lua *io* library must be fully ported to kernel space; the corresponding tests in the test suite should be enabled as well. Therefore, the deliverables are the port and the corresponding tests
- *Sockets* library Sprint 1 (Jun 8 - Jun 22): implementation of the API for datagram-oriented sockets; the deliverable is the working-API for UDP sockets, plus simple test programs
- *Sockets* library Sprint 2 (Jun 23 - Jul 7): implementation of the API for stream-oriented sockets; the deliverable is the working-API for TCP sockets and a few simple programs
- Benchmarks (Jul 8 - Jul 31): benchmark the libraries against their user space counterparts; this could be done by means of simple scripts implementing common use cases
- Polishing and documenting (Aug 1 - Aug 23): refactoring of code, writing of documentation; this is also a good time to fix issues, such as inconsistencies exposed by the benchmarks

3.3 What will be showable two months into the project?

The core of the project—*io* library, along with its tests and the *sockets* library—must be implemented by this point; benchmarks must also be getting in shape, with a few test cases already implemented.

4 GSoC

4.1 Have you participated to GSoC before? If so, How many times, which year, which project?

I participated in past year's Google Summer of Code, LabLua, under Lourival's mentorship. That was my only participation in GSoC.

4.2 Have you applied but were not selected? When?

No.

4.3 Did you apply this year to any other organizations?

No, I did not apply to other organizations.

References

[1] *NetBSD Kernel Lua Test Suite GitHub Repository*
`https://github.com/gmesalazar/luatests`

[2] *iolib on NetBSD Kernel Lua Test Suite GitHub Repository*
`https://github.com/gmesalazar/luatests/blob/master/kernel/modules/iolib/`

[3] *Ideas List - Google Summer of Code 2016*
`http://www.lua.inf.puc-rio.br/gsoc/ideas2016.html#netbsd`