

Porting LwIP to the GNU/Hurd

Joan Lledó (jlledom@member.fsf.org)

March 20, 2017

Abstract

The Hurd is the kernel of the GNU system. It consists on the GNU Mach microkernel and several user-space servers that offer most of the functions that usually a kernel offers, one of these functions is a TCP/IP stack. Currently, in the Hurd this stack is offered by the pfinet server, a port of the Linux 2.2 TCP/IP stack. Hurd developers think this stack should be replaced since it is obsolete and lacks important features like PPP support. In this project, we propose to port LwIP TCP/IP stack to the Hurd to replace pfinet.

About the Hurd

This section gives a little background about the Hurd and how it works that will help to understand the project. If you are familiar with the Hurd's architecture, you can skip to the next section.

Since most kernel functions in the Hurd are offered by user-space servers, instead of system calls, user programs perform RPC calls to servers to request services. For that reason, it is necessary to have an specification of which RPC calls are available and what they do. This specification is written in Matchmaker language in several *.defs files called interfaces, and RPCs declared in the interfaces are called operations. Two examples of interfaces are the io interface, for general I/O operations; and the socket interface, for TCP/IP stack operations.

In addition, a tool is needed for implementing the necessary underlying logic to use GNU Mach ports to perform remote procedure calls. MIG is the tool for that. This tool reads a *.defs file and generates C stubs for both server and user sides.

Moreover, there must be a way for a server to communicate with device drivers if needed. In the Hurd, drivers may be implemented as user-space servers or be provided by GNU Mach as in-kernel servers. In both cases, the device interface must be used to connect to the driver and send/receive data to/from it.

Why LwIP?

LwIP has been chosen to be the stack to port for several reasons. On the one hand, it is carefully designed to be portable and provides an OS abstraction layer plus a BSD sockets-like API that significantly reduce the amount of work needed to port it to a new

system. This, along with its extensive documentation and Wiki makes porting it to the Hurd a feasible goal within the GSoC calendar.

On the other hand, LwIP is an independent project that is being actively developed and releases new versions frequently. If we manage to make it work in the Hurd, future versions of LwIP could be easily adapted for the Hurd to use new features as they are implemented by the LwIP developers.

Project

Deliverable

By the end of the term, we will have a new server that will be able to replace pfinet. This new server will be divided into four modules:

1. Administration module:
It will initialize the stack and will allow it to be configured by command-line parameters and fsysopts.
2. MIG interfaces implementation:
This module will be the bridge between Glibc and LwIP. Io and socket interfaces will be implemented and we can not discard to implement some other interface's operations if needed.
3. LwIP:
The core of the TCP/IP stack will be integrated in the server.
4. Ethernet module:
It will use the device interface to enable communication between LwIP and the device driver.

The project includes the development of the modules 1,2 and 4. Regarding the project's scope, the goal is to achieve at least the same functions that pfinet offers now. That is:

- Support for IPv4 and IPv6
- Support for TCP and UDP
- Support for multiple Ethernet devices
- Support for fsysopts and command-line parameters configuration

Timeline

- [May 15 - May 28]
 - Set up the environment.

- Integrate LwIP code.
- Start coding.
- [May 29 - June 4]
 - Write a basic administration module.
 - Implement stack initialization.
- [June 5 - July 16]
 - Implement Socket interfaces.
 - Implement IO interfaces
 - Implement other interfaces' operations if needed
- [July 17 - August 6]
 - Write the Ethernet module.
 - Initialize the device interface
 - Implement output
 - Implement input
 - Implement support for more than one Ethernet interface
- [August 7 - August 20]
 - Complete the administration module
 - Add support for command-line parameters
 - Add support for fsysopts.

About me

My name is Joan Lledó and I am in the last year of my degree in CS at the Open University of Catalonia¹. My experience on programming includes working as Java EE full-stack developer for 8 years and a little experience on Free Software by contributing to Gparted² and writing Doclone³, a tool for cloning GNU/Linux systems.

I'm interested in OS programming and would like to learn and improve my skills in this field. That is why I see GSoC as a good chance to work in a real project and start making contributions to it. I will be able to work full time in this project during the GSoC term.

¹ <http://www.uoc.edu/portal/en/index.html>

² <https://git.gnome.org/browse/gparted/log/?qt=author&q=Joan+Lled%C3%B3>

³ <http://doclone.nongnu.org/>