VFS-style Union Mount under GNU/Hurd

Sergiu IVANOV
Technical University of Moldova
unlimitedscolobb@gmail.com

SCIENTIFIC ADVISOR: Mihail KULEV, PhD

***Abstract* – GNU/Hurd is a microkernel operating system with a unique architecture based on translators. A translator is a (user-space) server which is attached to a file system node, thus complying with the "everything is a file" concept. Translators can both modify the contents of the node they were set on and publish virtual file systems. Under normal circumstances, once a translator is mounted on a node, the contents of this node are completely obscured. The purpose of `unionmount` project is to allow mounting translators so that the virtual files systems will be merged with the contents of the underlying node. Also, this project supposes implementation of a set of rules which will allow for a more complex (merging) functionality.**

***Index Terms* – GNU/Hurd, merging rule, modularity, translator, union mount.**

I. INTRODUCTION. GNU/HURD

GNU/Hurd [1] is a microkernel operating system which, unfourtunately, has been in a rather experimental position recently. However, the fact that this operating system is not yet production ready does not prevent it from hosting implementations of a number of interesting design ideas which are meant to solve a wide variety of problems.

As it could be deduced from it's name, GNU/Hurd is a GNU project, using the GNU utilities. This implies the fact that a superficial user may not feel any difference in using these utilities from GNU/Linux or other GNU/* platforms.

A lot of work is being put into Debian GNU/Hurd flavour of GNU/Hurd now. This project makes the use of GNU/Hurd possible for users without extraordinary experience in operating systems. Of course, users are encouraged to endeavour to study the "hurdish" (of or pertaining to GNU/Hurd) approach to solving problems, but it is equally feasible to keep to the GNU/Linux style at the beginning.

One of the most attractive features of Debian GNU/Hurd as compared to pure GNU/Hurd (besides the possibility to use the enormous Debian package database, of which a large part has already been ported to GNU/Hurd) is the installation process. One needs to download the K16 installation CDs and go through a series of steps very much similar to those of installing a BSD or a GNU/Linux system. Advanced computer users have the possibility to cross-compile GNU/Hurd to get a better idea of how this operating system works.

GNU/Hurd runs on top of GNU/Mach microkernel, which eventually goes back to CMU's (Carnegie Mellon University) Mach 3.0. One specific trait of this microkernel is that it incorporates drivers, as different from MINIX3, for instance.

Going back to GNU/Hurd as a concept, it is essential to mention that, being a microkernel operating system, it has the concept of modularity at the foundation. When one speaks of a "hurdish" approach, ones mainly means modular approach combined with some specific features, dwelt on below. GNU/Hurd users are mostly advocates of modularity and the present article will also often focus on modularity.

II. TRANSLATORS

The feature which GNU/Hurd is renowned for is translators. A translator is a (user-space) server providing some a part of the POSIX-required functionality, as well as some specific possibilities. The most important difference from other microkernel platforms (like MINIX3) is that these servers are attached to some file system locations, thus constructing the familiar "everything is a file" UNIX approach.

Translators are mostly created by compiling against a set of libraries [3], among which are `libtrivfs`, `libnetfs`, and `libdiskfs`. These three (most well-known) libraries define the three general use-cases for translators. A `libtrivfs`-based translator has the simplest use-case: in most cases it operates on the contents of the file system node it is attached to (the *underlying node*) by applying some transformations to it and publishing the result to the client. A `libnetfs`-based translator will publish a virtual file system, consisting of virtual nodes, managed by the translator itself. The most complicated (though not most general) use-case pertains to `libdiskfs`-based translators: these are meant to operate on block devices and provide support for different types of file systems (for example, translator `ext2fs` which supports the `ext2` filesystem).

There have been attempts to provide translators on other platforms, but so far GNU/Hurd is the only platform possessing this feature. GNU/Hurd community has a mildly approving outlook on these attempts, considering that the *real* Hurd will be where the "hurdish" approach will be implemented best.

III. UNION MOUNT

A union mount is a mount in which several file systems (or file system locations) are merged and mounted on a single node. This concept has originated with Plan 9 and its concept of union directories and is implemented as the `UnionFS` file system for Linux, FreeBSD and NetBSD.

Just as in a large number of similar problems, such functionality is achieved via translators under GNU/Hurd. The Hurd Extra repository contains the corresponding `unionfs` translator, which not only implements the basic union mount, but also adds some features, like tracking the contents of the union-mounted directories and updating the merged file system.

The purpose of the `unionmount` project is to implement a different idea. Under normal circumstances, once you have mounted something on a file system node, the contents of that node (be that a directory or a file) are completely obscured by the mounted file system. However, there are occassons in which it might prove necessary to still have access to the underlying file system. This is the goal of the `unionmount` project: to mount translators in such a way, that the contents of the underlying node (directory) remain in view.

To prevent possible objections to the practical usefulness of this idea, a short description of a real use-case will be given next. Suppose you want to merge a real file system location and a virtual file system published by a translator. It can be implemented via `unionfs` by mounting the translator somewhere and then mounting `unionfs` on a different node. Having the possibility to union-mount this translator directly is a very handy feature, especially if one takes into consideration the fact that having a translator publish a virtual file system is a very common task under GNU/Hurd.

Nevertheless, the feature mentioned above is not the main one. Union-mounting is not by any means restricted to be a simple merging operation. Right now the author of this article is working on ideas of implementation of *rules* for merging the filesystems, which will range from deciding which nodes shall have priority (real ones or the ones published by the union-mounted translator) to automatically starting some translators when a node is accessed (thus mimicking the functionality of *passive* translators, which are automatically started by the file system when nodes with passive translators are accessed).

For advanced technical implementation details (oriented mainly towards people with GNU/Hurd programming experience) visit [4].

REFERENCES

[1] GNU/Hurd official web page http://hurd.gnu.org/

[2] Ivanov S. "Introduction to GNU/Hurd". October 2008.

[3] Ivanov S. "Namespace-based Translator Selection under GNU/Hurd". October 2008.

[4] GNU/Hurd article at Wikipedia http://en.wikipedia.org/wiki/Hurd#Development_history

[5] The GNU Hurd Reference Manual http://www.debian.org/ports/hurd/reference-manual/hurd.html

[6] bug-hurd Mailing List Archives http://lists.gnu.org/archive/html/bug-hurd/2009-04/msg00007.html