

Integrating Intel into a PowerPC Network

Graham Lee
UNIX Systems Manager,
University of Oxford Physics Practical Course
leeg@teaching.physics.ox.ac.uk

Computing lab. and support infrastructure comprising 39 PowerPC systems; mostly iMac G5 but two Xserves, some other client models. Until recently...



Why I'm here

One of these Macs is an Intel one; my goal is to make using it no different from the PowerPC ones

We've been here before

- Mac already migrated M68k - PPC
- Apple's Unix is Darwin, based on Mach OS
- Mach so far ported to ≥ 10 arch's
- Already use "fat/universal" binaries for G5
- Mach-O makes this easy(!?)

Apple have already had to switch processors once; this is just mentioned for completeness as the implementation was nothing like the Universal Binary mechanism in place today. Darwin derived from NeXTSTEP, supported on 68k, i386, sparc, PA-RISC (via rhapsody on i386/ppc) Mach also available on ppc, ppc64, 88k, alpha, VAX... It's been in use for a while on Mac; dual-architecture ppc/ppc64 libs and bins to support 64bit-specific code on G5, but still allowing G3/G4s to run 32-bit versions here since 10.2.8 With ELF, would have different versions of SOs in different locations and manipulate LD_LIBRARY_PATH to find the appropriate ones; c.f. Debian universal. Mach-O means we can store a single bin/lib with code appropriate for all arch's, less to maintain.

```
$ file uname-q
uname-q: Mach-0 universal binary with 4
architectures
uname-q (for architecture i386):\
    Mach-0 executable i386
uname-q (for architecture ppc):\
    Mach-0 executable ppc
uname-q (for architecture ppc64):\
    Mach-0 64-bit executable ppc64
uname-q (for architecture m68k):\
    Mach-0 executable for m68k architecture
```

This file contains executable code for different architectures (in fact for different OS as well). When executed, the loader will decide which is the most appropriate code to run. It does make the binary bigger (on disk anyway; the kernel only loads the appropriate version so the resident size is unchanged), but then most of the storage consumed by an app is in holding resources like images, localised text etc....Unrelated to Rosetta.

Still a way to go...

- No OS X Server/i386
- OS X Server $\geq 10.4.4$
- Installation/NetBoot images arch-dependent



<http://www.bombich.com>

Depending on needs, it may not be possible to go for a completely OS X/i386 environment. Some Server functions can be carried out by client ver (or even other UNIX) but not all. Some server installations still using Panther, can't support Intel clients fully.

Not supported to have a single boot image for both architectures at present, although allegedly possible (and certainly is in principle). Means generating and maintaining images for each arch. NetRestore is a useful tool for such imaging, and supports both platforms. Have to indicate that each image is for each platform, OS X Server then serves up the right image (using BSDP)

Distribution

- Fink, darwinports assume host arch only
- Maintain/distribute separate trees
- Or combine and distribute single PPC/i386 tree

```
$ lipo gnuplot_ppc gnuplot_i386 \  
-create -output gnuplot_u  
$ file gnuplot_u  
gnuplot_u: Mach-0 universal binary  
with 2 architectures [...]
```

Usual suspects for distributing FLOSS to Macs are Fink and Darwinports. Both work well in a networked environment but at time of writing, neither make it easy to generate unibins of their packages. PPC-only distribution is one consideration, but really doesn't work "universally" (some stuff can't be Rosettad). We could build and maintain distinct hierarchies, but then we'd essentially split the network down the arch line - exactly what we want to avoid. Instead we can perform some "lipostiction" - build the binaries separately and glue them together. This produces the same file as if we'd compiled both arch's in from the start - and it's scriptable (exercise for reader). Some gotchas include ensuring both segments can read the same config file, if we're using the same prefix (i.e. keep versions in sync); remembering to distribute universal dylibs where appropriate too (see next slide)

```
$ otool -L gnuplot_u
gnuplot_u:
    /System/Library/Frameworks/
CoreServices.framework/Versions/A/CoreServices
(compatibility version 1.0.0, current version
18.0.0)
    /System/Library/Frameworks/
ApplicationServices.framework/Versions/A/
ApplicationServices (compatibility version
1.0.0, current version 22.0.0)
    /usr/lib/libz.1.dylib (compatibility
version 1.0.0, current version 1.2.3)
    /sw/lib/libgd.2.dylib (compatibility
version 3.0.0, current version 3.0.0) [...]
```

How do we know which libraries to lipo (or otherwise provide universal versions of)? otool: it's like ldd on ELF platforms. It's also like objdump, and a lot of other cool stuff too. dylibs similar to SOs.

Absolute final last resort (ever)

```
$ arch
```

```
i386
```

```
$ export PATH=/usr/local/`arch`/bin:
```

```
[...]
```

If it all really does go pear-shaped, you could always configure each system differently depending upon its architecture. However as that's what we're trying to avoid I'm really not going to explain too much about it; it's wrong. Wrong wrong wrong wrong wrong. I've not had to do this *touches wood, kisses lucky rabbit foot, picks four-leafed clover*

One more thing...

- Rosetta works ppc -> i386, not i386 -> ppc
- Xcode emits host-arch-only binaries by default
- /Library/Application Support/Apple/Developer Tools/Project Templates/
- insert “ARCHS=(ppc,i386,);”



Some applications may need tweaking to work well across both architectures, I'll choose Xcode as an example as I know about it. In default state, a "Standard Tool" builds in Debug mode which is host arch only; if I build on i386 then log in at PPC my tool won't run. Create new project templates which always emit universal binaries. These can go in e.g. /Network/Library/Application Support/...

More Info

NetRestore: <http://www.bombich.com>

Fink: <http://fink.sourceforge.net>

Darwinports: <http://darwinports.opendarwin.org>

OS X admin info: <http://www.afp548.com>

Eric's talk :-)

Slides will be available at <http://users.ox.ac.uk/~wadh1342>

That was a short description of what happens when an Intel Mac joins a PPC network, and some of the considerations to be made. Much more info is out there on the web, AFP548 is a great resource for Mac sysadmins.