Debian GNU/kFreeBSD

Aurelien Jarno

What?

Why?

Status
Architectures
Toolchain
Integration in Debian
Available packages
Missing packages
Infrastructure

The future

How to help?
Trying it
As a Debian maintainer
Portable packaging
Portable programs

Additional information

Aurelien Jarno
aurel32@debian.org

FOSDEM

26/02/2006
What is Debian GNU/kFreeBSD?

- Debian port
- FreeBSD kernel (kFreeBSD for short)
  - kFreeBSD 5.4
  - experimental version of kFreeBSD 6.0
- GNU userland
- GNU libc
- Cool Debian tools (dpkg, apt, ...)

A Gentoo port has been started recently.
Why would you prefer Debian GNU/kFreeBSD to GNU/Linux?

- Because you like the FreeBSD kernel
  - Jails
  - UFS 2+
  - IPv6 statefull firewalling
  - Stable kernel API
- Better or worse device support
- To add diversity among your machines
- To be able to run FreeBSD and Linux binaries
- Debian is the "universal OS"
Why would you prefer Debian GNU/kFreeBSD to FreeBSD?

- Because you don’t like FreeBSD ports system (or just because you like the Debian package system :-)
- Because you prefer the GNU userland to the BSD one
- ext2fs, reiserfs, upcoming xfs and upcoming Xbox ethernet drivers enabled by default (GPL contaminated kernel)
- 100% free system according to the DFSG
Currently x86 only (kfreebsd-i386.deb)
An alpha port of the GNU libc has existed.
An AMD64 port is underway. The toolchain is ported, but there are still bugs to fix.
Status of the toolchain

- **GCC**
  - Support in upstream, except for Ada
  - Support all languages but Java
  - Version 3.3, 3.4, 4.0 and 4.1 are available

- **Binutils**
  - Support in upstream

- **GNU libc**
  - GNU libc ported by Bruno Haible in 2002
  - Not yet integrated in upstream, set of patches
  - Patches to remove "linuxisms" are being merged
  - Sysdeps/ directory will go to glibc-ports
  - Uses linuxthreads (how about the future?)
Integration in Debian

- Follows unstable and experimental
- "unreleased" repository for patched packages
- 77% of the packages have been ported (4700 source packages)
- The most advanced Debian non-Linux port
- popcon.debian.org shows 17 machines
- A few machines using it in production (web server, database)
Some available packages

- Xorg, KDE 3.5, most parts of Gnome, WindowMaker, ...
- Apache, PHP, MySQL, Postgresql, Samba...
- Abiword, KOffice, The Gimp, ...
- Frozen Bubble, GTetrinet, Pingus, ...
- Totem, Kaffeine, XMMS, ...

MPPlayer (not a Debian package) also works out of the box
Some missing packages

- Gecko based browsers
- Mozilla Thunderbird
- Openoffice.org
Infrastructure

- Primary mirror (ftp.gnuab.org) + a few mirrors
- Packages are searchable on http://pdo.debian.net
- 2 autobuilders
- Build status on http://buildd.net
- Build logs on http://experimental.ftbs.de
- Developer accessible machine (io.debian.net)
The future

- Reach 95% of source packages built
- Port integrated into the archive
- Release architecture
- Debian installer (currently it uses the FreeBSD installer)
- Other architectures

We need help!
How to try it?

- Installing it: [http://glibc-bsd.alioth.debian.org/install](http://glibc-bsd.alioth.debian.org/install)
- Debian Developer machine: [http://io.debian.net/ssh.html](http://io.debian.net/ssh.html)

Report bugs, and tell friends about it!
How to help as a Debian maintainer

- Try to build your packages on io.debian.net
- Try to port them
  - Out of date config.guess and config.sub
  - Out of date libtool
  - http://glibc-bsd.alioth.debian.org/PORTING
  - FreeBSD CVS
- Don’t ignore our patches sent to the BTS

Our policy is to send a bug report only when we have a patch.
A few tricks for portable packages

- Don’t suppose a GNU/Linux system:
  ./configure --host='dpkg --print-architecture'--linux
  but
  ./configure --host='dpkg-architecture -qDEB_HOST_GNU_TYPE'

- Don’t build depends on libc6-dev! The glibc development package is libc0.1-dev

- Don’t build depends on linux-kernel-headers if you don’t need kernel headers
A few tricks for portable packages (2)

- If your package supports Linux specific stuff (alsa, hal, ...), enable them for GNU/Linux only:

```
DEB_HOST_ARCH_OS := $(shell dpkg-architecture -qDEB_HOST_ARCH_OS)
ifeq (linux,$(DEB_HOST_ARCH_OS))
  linux specific stuff
endif
```

- Build-depends : field such as [linux-any], [any-i386] will be available soon

- Use DEB_HOST_ARCH_CPU to test for a specific CPU, not DEB_HOST_ARCH

- For x86 packages, add kfreebsd-i386 to the Architecture : field.
Writing portable programs

- Avoid Linux specific functions when possible
- Provide a way to bootstrap compilers
- Kernel ≠ userland
- Use autotools instead of `#ifdef` (use a recent version)
- `__FreeBSD__` = plain FreeBSD system (kernel + userland)
- `__FreeBSD_kernel__` = FreeBSD kernel only
Wrong solution:
```c
#ifdef __linux__
#include <endian.h>
#elif __FreeBSD__
#include <sys/endian.h>
#error "Unknown system"
#endif
```
Wrong solution:
```c
#ifdef __linux__
#include <endian.h>
#elif __FreeBSD__
#include <sys/endian.h>
#error "Unknown system"
#endif
```

Better solution:
```c
#ifdef __GLIBC__
#include <endian.h>
#elif __FreeBSD__
#include <sys/endian.h>
#error "Unknown system"
#endif
```
Writing portable programs (2)

- Wrong solution:
  ```c
  #ifdef __linux__
  #include <endian.h>
  #elif __FreeBSD__
  #include <sys/endian.h>
  #error "Unknown system"
  #endif
  ```

- Better solution:
  ```c
  #ifdef __GLIBC__
  #include <endian.h>
  #elif __FreeBSD__
  #include <sys/endian.h>
  #error "Unknown system"
  #endif
  ```

- Best solution: use autotools!
Writing portable programs (3)

- Wrong solution:
  
  ```c
  #include <linux/soundcard.h>
  ```

  Better solution:
  
  ```c
  #include <sys/soundcard.h>
  ```

  You can also use autotools, but don't use them to move the problem:

  ```bash
  case $host
  os in
    linux*) AC DEFINE(HAVE_LINUX_SOUNDCARD)
    freebsd*) AC DEFINE(HAVE_FREEBSD_SOUNDCARD)
  esac
  ```
Wrong solution:
#include <linux/soundcard.h>

Better solution:
#include <sys/soundcard.h>
Writing portable programs (3)

- Wrong solution:
  
  ```c
  #include <linux/soundcard.h>
  ```

- Better solution:
  
  ```c
  #include <sys/soundcard.h>
  ```

- You can also use autotools, but don’t use them to move the problem:
  
  ```
  case $host_os in
    linux*) AC_DEFINE(HAVE_LINUX_SOUNDCARD)
    freebsd*) AC_DEFINE(HAVE_FREEBSD_SOUNDCARD)
  esac
  ```
Wrong solution:

```bash
...  
  case $host_os in
    linux*) AC_DEFINE(_GNU_SOURCE)
    esac
  esac
...`

Best solution:

```bash
...  
  case $host_os in
    *-gnu) AC_DEFINE(_GNU_SOURCE)
    esac
  esac
...```
Wiki: http://wiki.debian.org/Debian_GNU/kFreeBSD
Mailing-list: debian-bsd@lists.debian.org
IRC: #gnu-kbsd@freenode
Patches, various stuff: http://glibc-bsd.alioth.debian.org/