

Gentoo Prefix

a unified experience
ON TOP OF
another distribution

Fabian Groffen
<grobian@gentoo.org>





About me

- Fabian Groffen <grobian@gentoo.org>
- Gentoo user since 2002
- Developer since 2005 (mostly Prefix)
- Software developer, origin in research, PhD Computer Science UvA 2009
- I need Vim and Mutt to survive :)
- My employer has no affiliation with Gentoo Prefix in any way!



Why?

- Having to work in an env that is out of date
 - often work-related
 - sometimes because nothing else exists
- Just wanting to prolong the life of hardware
 - having recent (dev) tools is key
- Well ... Just because I can™ :)



The Begin:



- Gentoo for Mac OS X (yes, Jaguar 10.2 ...)
- Complementing the host system, adding stuff in /usr/bin, etc.
- Quickly went wrong:
 - annoying for other devs
 - annoying for us, cannot replace tools (vim is out of date!)



Gentoo Prefix

- No more conflicts, install everything *in an offset*, a directory anywhere
- Do not require root privileges (no need)
- Provide any (system) tool possible (get consistent/known behaviour)
- Provide a *toolchain* (compiler, linker)



The Toolchain

- Mostly compiler + linker
 - FSF GCC, Apple GCC, LLVM/Clang
 - GNU binutils, Apple cctools+ld64
- The trick is **not** to just provide these tools
 - compiler and linker are configured in a special way



Typical Dev Setup

 gcc -o myapp myapp.c

 Oops, I need libyaml

 (sudo) yum/apt-get/brew libyaml

 (or whatever the package's name is here)

 gcc -o myapp myapp.c -lyaml

 ...



Typical Dev Setup

- `gcc -o myapp myapp.c -lyaml`
- just works
- compile fail `#include <yaml.h>`
- link fail `-lyaml` not found
- compile fail something not in this version



Typical Dev Setup

➤ Solution: compile and install manually

```
➤ gcc -o myapp myapp.c \  
    -I/path/to/myyaml/include \  
    -L/path/to/myyaml/lib \  
    -lyaml
```

➤ Yay! ./myapp

➤ Trap: cannot find libyaml.so (ELF)



Typical Dev Setup

- ▶ Ok, let's fix it: `LD_LIBRARY_PATH`
- ▶ yuk! (`DYLD_LIBRARY_PATH` on macOS)
- ▶ results in random problems with more complex programs
- ▶ Use `RPATH` to encode runpath
- ▶ works (ELF only)
- ▶ Use `ld.so.conf` (on Linux, requires root)



Prefix Toolchain

- ▶ Instructs the compiler to consider the offset (Prefix) include path by default
- ▶ Instructs the linker to resolve libraries from the offset library paths (yes plural)
- ▶ Instructs the linker to record RPATHs for given -L paths (including those mentioned above) (ELF only)



Prefix Toolchain Darwin/macOS

- ▶ Compiler to consider current configured SDK as system headers/libraries (recent Macs don't have `/usr/include` or `/usr/lib`)
- ▶ Linker to use `-search_paths_first`, to get same behaviour as seen with GNU ld
- ▶ Mach-O records full paths (`install_names`) instead of just object name (`soname`) so RPATH injection isn't necessary



Prefix Development

- ▶ emerge yaml
- ▶ gcc -o myapp myapp.c -lyaml
- ▶ ./myapp runs!
- ▶ linked to libyaml.so/libyaml.dylib from Prefix
- ▶ resolves to Prefix lib at runtime without extra tricks



Toolchain as Base

- ▶ With this toolchain, most packages compile/link (they simply don't see a difference)
- ▶ Build systems need special care, e.g. cmake not to invent counter-productive paths
- ▶ Ordinary porting remains necessary, e.g. macOS, Solaris, etc.



About the Host

- ▶ Base perspective on the host system:
 - ▶ accept its kernel, libc
 - ▶ this means, per Solaris/macOS release configuration profiles
- ▶ Linux perspective:
 - ▶ no real releases exist, kernel and libc not tied together
 - ▶ libc is open source, so can provide it too



Linux Prefix + libc

- ▶ RAP: Prefix with its own glibc
- ▶ Nice for ancient systems (RHEL) that have a sufficient kernel
- ▶ Allows to ditch RPATH linker stuff, and use ld.so.conf, from offset's own ld.so
- ▶ More “native” approach, preferred if possible

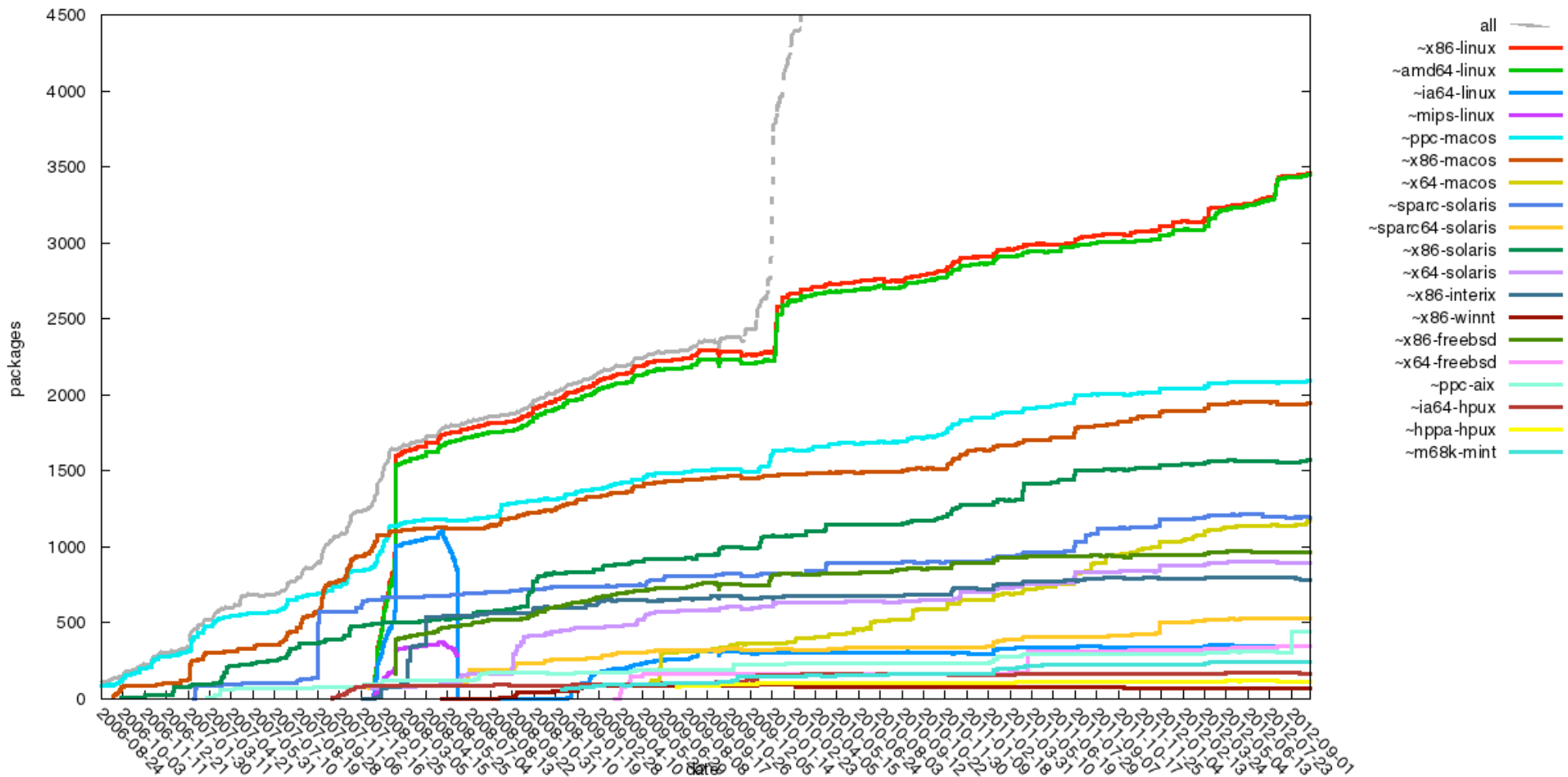


Prefix Today

- Fairly good support for macOS, Solaris and Linux
- Linux support is mostly inherited from Gentoo Linux, e.g. support for its arches
- No more support for things like AIX, IRIX, Atari, ...



What once was 2006





Support Today

- Support what we can
(as opposed to what we think we could)
- Need more (automated) testing and fixes
- <https://bootstrap.prefix.bitzolder.nl/results/>
keeps track of some semi-automated tests



Status within Gentoo

- ▶ Prefix is an official sub-project
- ▶ Separate infra for non-RAP targets due to need for patches
- ▶ Prefix is effective part of Gentoo since its specification in EAPI=3
- ▶ Long standing target to fully integrate with Gentoo Linux infra



Prefix Project

- ▶ <https://prefix.gentoo.org>
for current members, resources, contact
- ▶ Most contributions/issues via
<https://bugs.gentoo.org>
 - ▶ Gentoo Alt ▶ Prefix Support



Questions?