



# Embedded Systems and Gentoo Linux: An Introduction/Overview

Stephen L Arnold  
Gentoo Linux Developer

Sci, dev-tools, comm-fax herds  
CCLUG meeting, March 31, 2006

[stephen.arnold@acm.org](mailto:stephen.arnold@acm.org)  
[nerdboy@gentoo.org](mailto:nerdboy@gentoo.org)



# Embedded Devices

- What is an embedded device?
- Router-like devices: Linksys WRT54G/GL
- NAS-like devices: Kurobox/LinkStation, NSLU-2
- PDA/Phone-like devices (with or without Linux pre-installed)
  - Sharp Zaurus: 5000-D, 5500, C-x00, etc
  - Various iPaq models
  - Nokia 770
- Basically, anything with a processor and flash memory



# SBCs and Eval Kits

- Single Board Computers are simply smaller and more integrated system boards
  - Fewer I/O ports and peripheral connectors
  - Smaller footprint, lower power consumption
  - Sometimes fixed components (e.g, CPU, memory)
  - Various flash memory/EEPROM chips
  - Multiple architectures (x86, PowerPC, ARM, MIPS, SH)
- Evaluation Kits
  - An example board/CPU combination
  - Additional hardware (eg, serial I/O)
  - Embedded development kit
    - Example:  
<http://www.denx.de/wiki/DULG/ELDK>



# Hacking Embedded Devices

- Remote login or web browser interface
- Find a way in:
  - Telnet, http, etc
- Get the developer kit
  - Check Google and known vendor sites
  - Has someone already done it?
- Assimilate the developer materials
- Reverse-engineer it
- Build your own ROM (aka Linux from Scratch)
- Use your knowledge and Gentoo resources
  - Embedded stages, crossdev, portage



# Gentoo Embedded

- Stages for multiple arches, C libraries, shell environments
- Crossdev script for building custom cross-compiling toolchains (binutils, gcc, glibc/uclibc, kernel-headers)
- Leveraging portage – e.g., the gensoekris meta-ebuild
  - <http://dev.gentoo.org/~solar/embedded/local/local/net-wireless/gensoekris/gensoekris-0.0.8.ebuild>



# Install Gentoo on the Device

- Use a working linux distribution and get the patches and any required daemons in order to get Gentoo working on the device
  - Get any required source code, drivers/patches, and kernel source
  - Make some new ebuids (integrate patches, etc) or update the ones in portage to be compatible with the device
  - Make a new profile (ie, look for an appropriate parent profile and customize USE, packages, etc)
- Use crossdev and chroot to build an install image
- Adapt your Gentoo install to a reasonable size in case of limited storage space
- In many cases you can just follow a nominal



# General Caveats

- Flash memory layout
  - Where the heck do I copy my kernel and root filesystem image?
- Boot-loader support
  - How the heck do I get this thing to boot my kernel?
- Serial console
  - How the heck do I login to this thing?
- Coming soon to a portage mirror near you
  - U-boot and mkImage tools



# Resources

- Gentoo Embedded Project
  - [http://gentoo-wiki.com/Embedded\\_Gentoo](http://gentoo-wiki.com/Embedded_Gentoo)
  - <http://dev.gentoo.org/~vapier/CROSS-COMPILE-HOWTO>
- Embedded Linux
  - <http://elinux.org/wiki/FrontPage>
  - <http://www.denx.de/wiki/DULG/WebHome>
- Specific devices and boards
  - <http://openzaurus.org/wordpress/>
  - <http://www.nslu2-linux.org/>
  - <http://www.kurobox.com/>
  - <http://wiki.openwrt.org/OpenWrtDocs>
  - <http://supernova.stanford.edu/dingdong/>





# Portage features

- We provide scripts which download, patch, compile, and install packages
- Modeled on the ports-based BSD distributions
- Dependency checking, extreme customization
- Original source tarballs are downloaded
  - No need to wait for someone to make a binary package for your distribution
- The user specifies what they want, and the system is built to their specifications
  - Compiles are optimized for your specific hardware
    - E.g. AltiVec on G4 PPC chips, Pentium versus Athlon
  - Specify settings once, and all packages are built to those options



# Ebuild scripts

- Easy to read format, clear separation of phases
- KEYWORDS, DEPEND
- Stable versus testing
- CPAN/Portage integration
  - Install and manage dependencies via Portage
  - Package management for perl modules (including uninstall)
  - Ebuilds are automatically created for CPAN packages



# USE flags

- Globally defined list of features
  - Configure yours in `/etc/make.conf`
- `USE="-gnome kde qt arts -nls python perl oggvorbis opengl sdl -postgres jpeg png truetype dvd avi aalib mpeg encode fbcon mmx"`
- Each one defines specific functionality for each package to support
- USE flags generally map onto `--configure` options
- Install only what you want. No need to trim down a default installation
- “Opt-in” versus “opt-out”



# Init scripts

- Named (not numbered) run levels
- Smart dependencies
  - Scripts can 'use' or 'depend' on others
- Start / Stop / Pause
- `/etc/runlevels/default/`
- `/etc/runlevels/boot/`



# Installation

- Doesn't it take a long time to install packages?
  - Designed for modern hardware
  - Will run on a Pentium or PPC 603 with 64mb RAM
  - Built-in ccache, distcc support
- Net connection recommended
- “Secure” by default
  - Because nothing is installed by default
- Packages are installed into a “staging” directory, then merged (portage sandbox)
- Virtual packages
- Build from source or install binary quarterly releases with optional GRP (Gentoo Reference Platform) packages



# Portage demo

- rsync
  - Local Portage mirror
- Package searching
- Installation, cleaning, removal
- Ebuild command for developers
- etc-update
  - CONFIG\_PROTECT
- /var/db/pkg



# Installation process

- Currently no graphical installer
- Just follow the detailed install documents
  - Boot from CD, setup networking, partition
  - Unpack stage 1, stage 2, or stage 3
  - Chroot, bootstrap or emerge system
  - Compile kernel, install system logger, cron daemon
  - Setup bootloader.
  - Set timezone, configure additional users



# Stage tarballs

- Stage 1 install
  - Bare-bones. Need to bootstrap, compile gcc, glibc, system (make, perl, etc.), kernel, and user environment
- Stage 2 install
  - Already bootstrapped. Compile system, kernel, and user environment
- Stage 3 install
  - Base system included. Compile kernel and boot manager
- GRP install
  - Precompiled packages with the default MAKEOPTS and USE flags for your architecture





# Binary packages

- Build your own, distribute packages to your machines
  - `emerge --buildpkg`
  - `FEATURES="buildpkg"`
  - `PORTAGE_BINHOST="http://local-server"`
- GRP (Gentoo Reference Platform)
  - Pre-built binary packages using default options
  - Including Xfree86, Mozilla, Gnome, KDE, Emacs, OpenOffice.org, Apache, MySQL, PostgreSQL, Samba



# Coming soon

- Direct GLSA/Portage integration
- Pathspec integration
- Portage-ng
- Ebuild signing
- Official AMD 64 and IA64 support - Done
- udev support - Done



# Try out Gentoo Linux

- Download from [www.gentoo.org](http://www.gentoo.org)
  - 95-600mb iso images, plus 10-85mb stage files
- Unreal Tournament 2004 demo for x86/NVIDIA
  - Does not touch your hard drive
- Live CDs for x86, PPC (NewWorld and OldWorld), Alpha, AMD 64, HPPA, Sparc
- All live CDs are also install and rescue CDs



# Documentation

- Gentoo Handbook
- Installation, FAQs
- Portage user manual
- USE flags, ENV.D, Security guide
- Desktop configuration guide, rc-scripts, ALSA, DRI, Java
- AFS, OpenMosix, Diskless/LTSP, Printing, UML, IPv6, Virtual Mailhost
- Developer documentation
  - Ebuild creation, eclass
  - Documentation guide (XML syntax)



# Gentoo Community

- Close contact with end users
- Many ebuild scripts are submitted by users
- IRC channels (on [irc.freenode.net](http://irc.freenode.net)) - #gentoo is the largest on the network with 800+ users
- Web-based forums (on [forums.gentoo.org](http://forums.gentoo.org)), 1000+ posts per day, 100,000+ topics
- Fully-public bug tracking ([bugs.gentoo.org](http://bugs.gentoo.org)), 20,000+ hits per day
- “Gentoo Bug Day”
- Linux World Expo in Boston, possibly SoCal Linux Expo in Los Angeles (Feb 2005)



# Staying informed

- News on [www.gentoo.org](http://www.gentoo.org)
- Gentoo Linux Security Announcements (GLSA)
- RDF feed of news. GLSA feed coming soon
- Multiple mailing lists (each architecture, documentation, security)
- Gentoo Weekly News (GWN)
- Informal Discussion and Announcements
  - Gentoo forums on [forums.gentoo.org](http://forums.gentoo.org)
  - Gentoo IRC channels on [irc.freenode.net](http://irc.freenode.net)



This work is an updated version of the original work below; modified by Stephen Arnold <nerdboy@gentoo.org>

<<http://dev.gentoo.org/~nerdboy>>

Portions copyright 2006 Stephen L Arnold. Some rights reserved.

<<http://dev.gentoo.org/~rajiv/IntroToGentoo/>>

Original copyright 2004 Rajiv Manglani. Some rights reserved.

The Gentoo Linux logo is Copyright 2004 Gentoo Foundation, used with permission.



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike License. To view a copy of this license, visit <<http://creativecommons.org/licenses/by-nc-sa/1.0>> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Please contact Rajiv Manglani <[rajiv@alum.mit.edu](mailto:rajiv@alum.mit.edu)> for commercial uses of this work.