THE LINUX FOUNDATION

OPEN SOURCE SUMMIT

EUROPE

THE LINUX FOUNDATION

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
EUROPE

Accelerated Linux Build System

Jeff Shaw
Digi International

jeff.shaw@digi.com

THE LINUX FOUNDATION

Introduction

- **In the early days, it was hard work. Nothing came for free. Every install was *Roll your own.***

- **Enter the distribution. Applications were pre-compiled for easy installation.**

Source Distributions

- **Allowed custom, optimised kernel and applications.**
- **Was slow to create on machines of the day.**
- **Computers are now a lot faster!**

Source Distributions

- **Can’t we just use a ready made distribution?**

- **Of course you can, *most of the time***
  - Embedded target without a distribution tailored for the hardware
  - Target isn’t powerful enough
  - Target is low on resources
  - Need a feature not normally offered

Source Distributions

- There are several good build systems out there. Two of the most well known are:
  - YOCTO
  - Buildroot

Popular build systems

- **YOCTO**
  - Advantages
    - Well known
    - Flexible
    - Large number of supported devices
    - Industry supported
    - On-target package management

Popular build systems

- **YOCTO**
  - Disadvantages
    - Steep learning curve
    - Build times
    - Resource usage

Popular build systems

- **Buildroot**
  - Advantages
    - Simple
    - Efficient
    - Uses standard technologies like make and Kconfig
  - Disadvantages
    - The configuration is stored in a single file
    - Different targets require a new configuration file
    - Requires rebuild when configurations change

Accelerated Linux

- **Accelerated Linux (ACL)**
  - Advantages
    - Flexible
    - Large number of supported targets
    - Simple
    - Efficient
    - Standard technologies like make and Kconfig
  - Disadvantages
    - Not well known

Accelerated Linux

- **History**
  - Direct descendant of uClinux Distribution
  - Created in 1998
  - 3 years before Buildroot
- **Features**
  - MMU and non MMU support
  - Large number of targets
    - Nightly build has 147 targets
    - From ARM and MIPS to x86 and x86-64

Accelerated Linux

- **Bottom up methodology**
  - Start with nothing
  - Select a target
  - Customise the kernel
  - Choose a libc (uClibc, glibc, musl, …)
  - Configure libc, add required features
  - Build packages (busybox, …)
  - Create the firmware image

Accelerated Linux

- **Single make command**
  - Configure, compile and create the image
- **Dependency checking**
  - Kconfig
- **Customisation**
  - make config/xconfig/menuconfig

Accelerated Linux

- **ACL Optimised for:**
  - Static devices
  - Embedded applications
  - No on-target package management
  - Resource limited targets

- **Small image sizes**
  - From: 3.5Mb
  - To: >100Mb

Accelerated Linux

- **Root Directory ($ROOTDIR)**
  - Documentation
  - Quick start README file
  - Global Makefile

Accelerated Linux

- **Four main source directories**

linux	for all things kernel
lib	common libraries
user	public packages

- **Each source directory has its own Makefile**

Accelerated Linux

- **Non source directories**

vendors	target specific configuration
config	current configuration
tools	compilation and verification tools

- **Transient directories (removed with clean)**

romfs	ROM root filesystem
images	firmware image
staging	cached applications and libraries

Accelerated Linux

- **Toolchain**
  - NOT compiled as part of the build
  - ACL supplied standard builds of gnu binutils and gcc for many targets
  - Third party support
- **Advantages**
  - Reproducibility
  - Simplicity
  - Compatibility
  - Performance

Accelerated Linux

- **Open development**
  - No hardware required
  - Supported targets can be run in QEMU
  - ARM/Versatile-PB-NOMMU
  - Freescale/M5208EVB
  - X86
  - X86_64

Accelerated Linux - Building

- **Single command**
  - *make <vendor>/<product>_default*

    e.g. make Digi/WR11_default
  - Complete build process, from target clean to image creation
  - Build server friendly, no special commands required

Accelerated Linux - Building

- **Default compilation**
  - Fully parallelised with sync points (kernel/lib/user)
  - Can make finding errors difficult

- **Custom compilation**
  - Supply the parameter *single*

    e.g. *make single*

Accelerated Linux - Building

- **Incremental build**
  - Once built, rebuild the current target with *make*
  - Only modified areas will be rebuilt.
  - Updates the target filesystem
  - Create new firmware image

Accelerated Linux - Building

- **During development**
  - Specify a single area to recompile only
    e.g. *make user_only*
  - Specify a single package/directory only
    e.g. *make user/openvpn_only*
  - Doesn’t update the target filesystem
  - Doesn’t rebuild the firmware image

Accelerated Linux - Building

- **Update the target filesystem**
  - Update the area component
    e.g. *make user_romfs*
  - Update a single package component
    e.g. make *user/openvpn_romfs*
- **Regenerate the firmware image**
  - *make image*

Accelerated Linux - Config

- **Vendors and products**
  - Exist in ROOTDIR/vendors/<vendor>/<product>
    - e.g. ROOTDIR/vendors/Digi/WR11
  - Standard Makefile config files
    - config.arch
    - config.device
    - config.linux
    - config.uClibc (if uClibc is selected)
    - config.vendor
  - Contains target Makefile

Accelerated Linux - Config

- **Custom files**
  - Most target customisations, eg. /etc/rc, live in the vendor/product directory.
  - The vendor/product Makefile will define how to use these files

Accelerated Linux - Packages

- **Package types**
  - In tree source files, built in place
    - busybox
    - net-snmp
  - Automake, download, and build when required
    - ssh
    - Openvpn
    - Automatic patching via patches directory

Accelerated Linux

- **Build times**
  - ACL designed for simplicity and performance

  - Test PC: 4.2GHz i7-7700K, 32G RAM, 2Tb HDD
    - Small target, 2 minutes, 3.5Mb image
    - Medium target, 10 minutes, 14Mb image
    - Large target, 14 minutes, 24Mb image

  - Build PC (slow virtual machine)
    - 17 targets, 6 hours 6 minutes (21 minutes avg)

  - Nightly Build (10yr old PC)
    - 147 targets, 25 hours (10 minutes avg)

Accelerated Linux - Demo

- URL: https://github.com/AcceleratedLinux
- Documentation:
  - Documentation/*
- Quick start
  - README