



IBM Linux Technology Center

Porting Linux to a new architecture, done right

Arnd Bergmann

© 2010 IBM Corporation

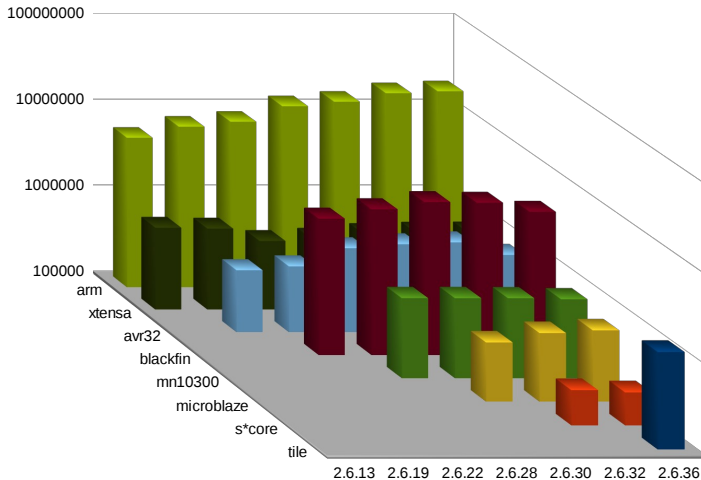
Overview

- 1 Status of new architecture ports
- 2 Typical approaches to new architectures
- 3 Lessons Learned

About me

- s390 architecture 2002-2005
- PowerPC/Cell architecture 2005-now
- 32/64 bit syscall emulation
- Maintaining include/asm-generic
- Reviewing new architectures
- Learning about ARM

Recently merged architectures



Upcoming architectures

- nios II
- lm32
- UniCore
- OpenRISC
- c64x
- MMIX
- nameless 48-bit architecture
- nameless DSP architecture

What makes architecture ports so hard?

Typical approaches to new architectures

Language barrier #1: English



Homes of the current architecture maintainers

Language barrier #2: C

C

Language barrier #2: C

GNU C99

Language barrier #2: C

GNU C99 with static annotations

Language barrier #2: C

ILP32/LP64 GNU C99 with static annotations

Language barrier #2: C

freestanding
ILP32/LP64 GNU C99 with static annotations

Language barrier #2: C

object-oriented freestanding
ILP32/LP64 GNU C99 with static annotations

Language barrier #2: C

pragmatically object-oriented freestanding
ILP32/LP64 GNU C99 with static annotations

Language barrier #2: C

pragmatically object-oriented freestanding
ILP32/LP64 GNU C99 with static annotations
and enforced coding style

Architectural challenges

- Symmetric Multiprocessing

Architectural challenges

- Symmetric Multiprocessing
- Noncoherent I/O

Architectural challenges

- Symmetric Multiprocessing
- Noncoherent I/O
- Out-of-order I/O

Architectural challenges

- Symmetric Multiprocessing
- Noncoherent I/O
- Out-of-order I/O
- Memory management units

Architectural challenges

- Symmetric Multiprocessing
- Noncoherent I/O
- Out-of-order I/O
- Memory management units
- Multiple ABIs

Architectural challenges

- Symmetric Multiprocessing
- Noncoherent I/O
- Out-of-order I/O
- Memory management units
- Multiple ABIs
- Multiple ISAs

Architectural challenges

- Symmetric Multiprocessing
- Noncoherent I/O
- Out-of-order I/O
- Memory management units
- Multiple ABIs
- Multiple ISAs
- Multiple platforms

Architectural challenges

- Symmetric Multiprocessing
- Noncoherent I/O
- Out-of-order I/O
- Memory management units
- Multiple ABIs
- Multiple ISAs
- Multiple platforms
- Timekeeping

Where to start

- Copy from x86

Where to start

- Copy from ARM

Where to start

- Copy from Tile

Where to start

- Do not copy at all!

Generic header files

- 42 versions of struct stat?

Generic header files

- 42 versions of struct stat?
- One minimal syscall list!

Generic header files

- 42 versions of struct stat?
- One minimal syscall list!
- Cover all the simple implementations

Generic Architecture template

- Early Boot code
- zImage compression
- Library functions
- device tree
- Trap handling
- Signal handling
- ptrace
- pci

User space: runtime

- glibc/eglibc
- uClibc
- klibc

User space: distro

- Full distribution
 - Debian, Fedora, OpenSUSE, Ubuntu, Gentoo, ...

User space: distro

- Full distribution
 - Debian, Fedora, OpenSUSE, Ubuntu, Gentoo, ...
- Embedded distribution
 - buildroot
 - yocto
 - emdebian

User space: distro

- Full distribution
 - Debian, Fedora, OpenSUSE, Ubuntu, Gentoo, ...
- Embedded distribution
 - buildroot
 - yocto
 - emdebian
- Busybox initramfs

Lessons Learned

Lesson #1

Start small

Lesson #2

Generalize existing code for your special case

Lesson #3

Understand the development process

Lesson #4

Follow the upstream kernel

Lesson #5

Debug your system with qemu with gdb

Lesson #6

Simplify drivers using virtio and hvc

Lesson #7

Describe SoC in a flattened device tree

Lesson #8

Clean up after sparse and checkpatch

Lesson #9

Run lockdep enabled kernels

Lesson #10

Become a git master

Legal Statement

This work represents the view of the author and does not necessarily represent the view of IBM.

IBM, IBM (logo), e-business (logo), pSeries, e (logo) server, and xSeries are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product, and service names may be trademarks or service marks of others.

Questions?