



Let's port
together.
Debian fun for
everyone.

Peter De
Schrijver

Outline

Let's port together. Debian fun for everyone.

"Most civilised people are out of touch with reality because they confuse the world as it is with the world as they think about it, talk about it and describe it."

Peter De Schrijver
p2@debian.org

February 24, 2007



Overview I

Let's port
together.
Debian fun for
everyone.

Peter De
Schrijver

Outline

Portability issues

why ?

C types

Bitfields

Endianness

Alignment

Accessing peripheral hardware

Example system architectures

Trends in system design

Out of order transactions

Non-coherent I/O

Userland hardware access



Why ?

Let's port
together.
Debian fun for
everyone.

Peter De
Schrijver

Portability

Why ?

C types

Bitfields

Endianness

Alignment

Peripherals

Architectures

Trends

Out of order

I/O

Userland

- ▶ Correctness
- ▶ Debian is "The Universal Operating System"
- ▶ Debian is the most used Embedded Distribution
- ▶ Hardware advances will make Debian feasible on new platforms
- ▶ It's enlightening to see and play with other architectures/systems



C types

Let's port
together.
Debian fun for
everyone.

Peter De
Schrijver

Portability

Why ?

C types

Bitfields

Endianness

Alignment

Peripherals

Architectures

Trends

Out of order

I/O

Userland

▶ ANSI-C

- ▶ $\text{sizeof(char)} \leq \text{sizeof(short)} \leq \text{sizeof(int)} \leq \text{sizeof(long)}$
- ▶ short and int are at least 16bit
- ▶ long is at least 32bit
- ▶ $\text{sizeof(ptr)} \neq \text{sizeof(int)}$
- ▶ signedness of chars is arch dependent

▶ Tips

- ▶ use int as much as possible for computations, loop variables,...
- ▶ use ISO C99 types (`u_int8`, `u_int16`, `u_int32`, ...) for external comms
- ▶ don't abuse chars to 'save memory'
- ▶ use the latest gcc version with `-Wall`



Bitfields

typedef struct bitfields

```
{  
    unsigned char bitfield0:3;  
    unsigned char bitfield1:5;  
}
```

IA32 representation :



PowerPC representation :



Let's port together.
Debian fun for everyone.

Peter De Schrijver

Portability

Why ?

C types

Bitfields

Endianness

Alignment

Peripherals

Architectures

Trends

Out of order

I/O

Userland



Endianness

Let's port
together.
Debian fun for
everyone.

Peter De
Schrijver

Portability

Why ?

C types

Bitfields

Endianness

Alignment

Peripherals

Architectures

Trends

Out of order

I/O

Userland

```
Consider 0x12345678
```

```
Little endian : 0x78 0x56 0x34 0x12
```

```
Big endian : 0x12 0x34 0x56 0x78
```

```
PDP endian : 0x34 0x12 0x78 0x56
```

- ▶ External interfaces
- ▶ Use macros to convert between CPU and specific endianness



Alignment

Let's port
together.
Debian fun for
everyone.

Peter De
Schrijver

Portability

Why ?

C types

Bitfields

Endianness

Alignment

Peripherals

Architectures

Trends

Out of order

I/O

Userland

- ▶ Most RISC cpus require aligned accesses
- ▶ Unaligned accesses are trapped (mostly)
 - ▶ slow
 - ▶ not possible in kernel land
- ▶ Unaligned accesses are seldomly atomic with respect to SMP/other bus masters
- ▶ Better
 - ▶ avoid them
 - ▶ have the compiler generate the code



Intel style system

Let's port
together.
Debian fun for
everyone.

Peter De
Schrijver

Portability

Why ?

C types

Bitfields

Endianness

Alignment

Peripherals

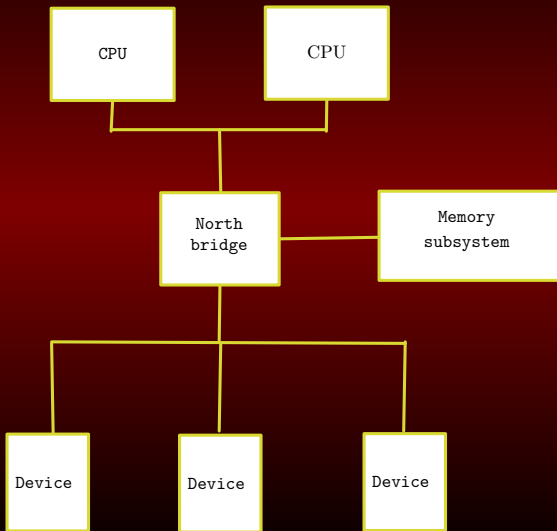
Architectures

Trends

Out of order

I/O

Userland





Intel style system

Let's port
together.
Debian fun for
everyone.

Peter De
Schrijver

Portability

Why ?

C types

Bitfields

Endianness

Alignment

Peripherals

Architectures

Trends

Out of order

I/O

Userland

- ▶ Main components
 - ▶ CPU complex
 - ▶ Northbridge
 - ▶ Southbridge
 - ▶ Memory subsystem
- ▶ Main interfaces
 - ▶ Frontside bus
 - ▶ PCI
 - ▶ AGP



Opteron style system

Let's port
together.
Debian fun for
everyone.

Peter De
Schrijver

Portability

Why ?

C types

Bitfields

Endianness

Alignment

Peripherals

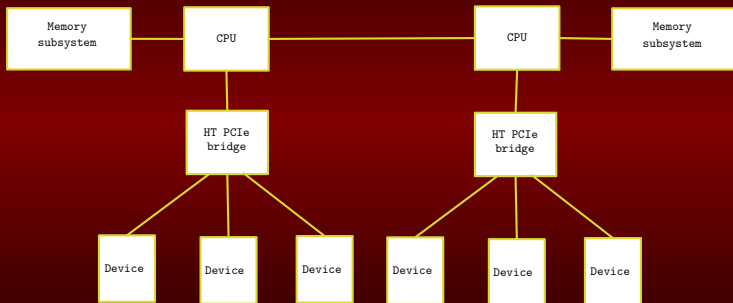
Architectures

Trends

Out of order

I/O

Userland





Opteron style system

Let's port
together.
Debian fun for
everyone.

Peter De
Schrijver

Portability

Why ?

C types

Bitfields

Endianness

Alignment

Peripherals

Architectures

Trends

Out of order

I/O

Userland

▶ Main components

- ▶ CPU
- ▶ Hypertransport - PCIe bridge
- ▶ PCIe - PCI bridge

▶ Main interfaces

- ▶ Hypertransport
- ▶ PCIe
- ▶ PCI



Trends in system design

Let's port
together.
Debian fun for
everyone.

Peter De
Schrijver

Portability

Why ?

C types

Bitfields

Endianness

Alignment

Peripherals

Architectures

Trends

Out of order

I/O

Userland

▶ Observations

- ▶ CPUs became much faster than memory
- ▶ bus and memory bandwidth have gone up faster than latencies
- ▶ parallel busses become very hard at high speeds

▶ Solutions

- ▶ Caches
- ▶ Burstmode transfers
- ▶ Advanced DMA
- ▶ multiple highspeed serial links



Out of order transactions

Let's port
together.
Debian fun for
everyone.

Peter De
Schrijver

Portability

Why ?

C types

Bitfields

Endianness

Alignment

Peripherals

Architectures

Trends

Out of order

I/O

Userland

- ▶ examples
 - ▶ CPU
 - ▶ bus bridges
- ▶ use read/write barriers
 - ▶ CPU instructions
 - ▶ "magic" reads

possibly out of order :

```
stw r20,0x20(r21)
```

```
stw r22,0x24(r21)
```

always in order :

```
stw r20,0x20(r21)
```

```
eieio
```

```
stw r22,0x24(r21)
```



Non-coherent I/O

Let's port
together.
Debian fun for
everyone.

Peter De
Schrijver

Portability

Why ?

C types

Bitfields

Endianness

Alignment

Peripherals

Architectures

Trends

Out of order

I/O

Userland

- ▶ Some systems do not support "bus snooping"
- ▶ Invalidate cachelines
 - ▶ network traffic
 - ▶ disk buffers
 - ▶ other kinds of streaming I/O
- ▶ non-cacheable memory
 - ▶ microcode
 - ▶ ring buffers



Addressing

Let's port
together.
Debian fun for
everyone.

Peter De
Schrijver

Portability

Why ?

C types

Bitfields

Endianness

Alignment

Peripherals

Architectures

Trends

Out of order

I/O

Userland

- ▶ Virtual addresses
- ▶ Physical addresses
- ▶ Bus addresses
- ▶ Translation Physical to Bus addresses
 - ▶ identity mapped
 - ▶ fixed offset
 - ▶ page based translation
 - ▶ not memory mapped
 - ▶ IA32 I/O ports
 - ▶ PowerPC DCB
- ▶ Always access hardware via special functions



Transaction atomicity

Let's port
together.
Debian fun for
everyone.

Peter De
Schrijver

Portability

Why ?

C types

Bitfields

Endianness

Alignment

Peripherals

Architectures

Trends

Out of order

I/O

Userland

- ▶ Multiple CPUs
- ▶ other busmasters (eg. on PCI)
- ▶ reads and writes are atomic only if aligned
- ▶ atomic read/modify/write is CPU specific
 - ▶ ia32: lock prefix on specific instructions
 - ▶ mips: ll/sc
 - ▶ arm: swap
 - ▶ ppc: lwarx/stwcx
- ▶ bridges may break locks



Userland hardware access

Let's port
together.
Debian fun for
everyone.

Peter De
Schrijver

Portability

Why ?

C types

Bitfields

Endianness

Alignment

Peripherals

Architectures

Trends

Out of order

I/O

Userland

- ▶ Hardware access from userland is problematic
- ▶ Seperate command transport from driver logic
 - ▶ Firewire : libraw1394
 - ▶ USB : libusb
 - ▶ SCSI and ATAPI : scsi generic like ioctl
 - ▶ ...
- ▶ Provide abstraction layer for accessing hardware