The SH4 QEMU target

A short introduction and status report

Magnus Damm

IGEL Co., Ltd.
www.igel.co.jp

September 2007

Outline

Outline

What is QEMU?

The open source community treats QEMU as a large collection of emulation software components. . .

What is QEMU?

The open source community treats QEMU as a large collection of emulation software components...

...but to the end user QEMU appears as two types of applications.

What is QEMU?

The open source community treats QEMU as a large collection of emulation software components...

...but to the end user QEMU appears as two types of applications.

The QEMU user space emulator application

- Used to execute non-native applications under Linux
  - SH4 binaries can for example be executed on x86 processors
- QEMU emulates the processor and translates the syscalls

What is QEMU?

The open source community treats QEMU as a large collection of emulation software components...
...but to the end user QEMU appears as two types of applications.

The QEMU user space emulator application

- Used to execute non-native applications under Linux
  - SH4 binaries can for example be executed on x86 processors
- QEMU emulates the processor and translates the syscalls

The QEMU system emulator application

- Used to emulate an entire hardware platform
  - Windows may for instance be installed in a virtual PC
- QEMU emulates processor, memory and I/O devices

QEMU provides...

- Efficient processor emulation using JIT techniques
  - Supports big endian and little endian hosts and targets
  - Breakpoint and single step support using built in GDB stub
  - Emulates Alpha, ARM, x86, m68k, Mips, PPC, SH4 and Sparc processors

QEMU provides...

- Efficient processor emulation using JIT techniques
  - Supports big endian and little endian hosts and targets
  - Breakpoint and single step support using built in GDB stub
  - Emulates Alpha, ARM, x86, m68k, Mips, PPC, SH4 and Sparc processors
- Multiple network emulation options
  - Simple emulated network with built in DHCP and TFTP servers
  - Virtual ethernet support using VDE
  - TAP/TUN support

QEMU provides...

- Efficient processor emulation using JIT techniques
  - Supports big endian and little endian hosts and targets
  - Breakpoint and single step support using built in GDB stub
  - Emulates Alpha, ARM, x86, m68k, Mips, PPC, SH4 and Sparc processors
- Multiple network emulation options
  - Simple emulated network with built in DHCP and TFTP servers
  - Virtual ethernet support using VDE
  - TAP/TUN support
- Multiple host architecture and operating system support
  - Both Linux and Windows hosts are supported by the system emulator

QEMU provides...

- A wide range of emulated hardware devices
  - Interrupt controllers and timers
  - Serial and parallel ports
  - Keyboard, mouse and touch screen controllers
  - VGA graphics and other frame buffers
  - IDE and SCSI disks plus CDROM/DVD drives
  - Disk controllers for IDE and SCSI
  - Ethernet controllers such as NE2K, EEPRO100 and RTL8139
  - USB OHCI and UHCI support
  - Sound cards
  - NAND and NOR Flash
  - Various devices such as RTC, CMOS, and EEPROMs

QEMU provides...

- A wide range of emulated hardware devices
  - Interrupt controllers and timers
  - Serial and parallel ports
  - Keyboard, mouse and touch screen controllers
  - VGA graphics and other frame buffers
  - IDE and SCSI disks plus CDROM/DVD drives
  - Disk controllers for IDE and SCSI
  - Ethernet controllers such as NE2K, EEPRO100 and RTL8139
  - USB OHCI and UHCI support
  - Sound cards
  - NAND and NOR Flash
  - Various devices such as RTC, CMOS, and EEPROMs
- Suspend and Resume support
- Networked user I/O using VNC

Code reuse

The QEMU project incorporates code from the following projects:

- BIOS from the Bochs project
- Network emulation code from the Slirp project
- VNC, VDE and TUN/TAP code

Code reuse

The QEMU project incorporates code from the following projects:

- BIOS from the Bochs project
- Network emulation code from the Slirp project
- VNC, VDE and TUN/TAP code

Many open source projects reuse QEMU code, among them:

- KVM - System emulation
- Xen - I/O emulation

Developer Community

- QEMU developers
  - The main author and lead developer of QEMU is Fabrice Bellard.
  - A small group of individuals also extend the QEMU code base.

Developer Community

- QEMU developers
  - The main author and lead developer of QEMU is Fabrice Bellard.
  - A small group of individuals also extend the QEMU code base.
- Discussions take place on the `qemu-devel` mailing list.

`http://lists.nongnu.org/mailman/listinfo/qemu-devel`

Outline

Overview

The QEMU user space emulator provides a way for the user to execute non-native binaries under Linux. For example, closed source x86 binaries may be executed on non-x86 processors using QEMU.

The name comes from the fact that only user space is emulated by the QEMU user space emulator. Kernel code is not emulated.

The core part of the QEMU user space emulator is the processor emulation code which is shared with the system emulator. This code together with syscall translation software and some glue code forms the QEMU user space emulator.

Usage Examples

How to use the QEMU user space emulator to. . .

Usage Examples

How to use the QEMU user space emulator to...

...run a static x86 binary

```
qemu-i386 busybox-i386
```

Usage Examples

How to use the QEMU user space emulator to...

...run a static x86 binary

```
qemu-i386 busybox-i386
```

...run a dynamic x86 binary

```
qemu-i386 -L /path/to/x86/root busybox-i386
```

Usage Examples

How to use the QEMU user space emulator to...

...run a static x86 binary

```
qemu-i386 busybox-i386
```

...run a dynamic x86 binary

```
qemu-i386 -L /path/to/x86/root busybox-i386
```

...run a static SH4 binary

```
qemu-sh4 busybox-sh4
```

Usage Examples

How to use the QEMU user space emulator to...

...run a static x86 binary

```
qemu-i386 busybox-i386
```

...run a dynamic x86 binary

```
qemu-i386 -L /path/to/x86/root busybox-i386
```

...run a static SH4 binary

```
qemu-sh4 busybox-sh4
```

To enable SH4 user space emulation in QEMU please configure the source using `./configure -target-list=sh4-linux-user`

Current SH4 Status

The QEMU user space emulator currently supports a subset of the SH7750 cpu core. The SH4 part of the QEMU user space emulator is functional but currently limited to static binaries only.

Current SH4 Status

The QEMU user space emulator currently supports a subset of the SH7750 cpu core. The SH4 part of the QEMU user space emulator is functional but currently limited to static binaries only.

Areas that need improvement are:

Current SH4 Status

The QEMU user space emulator currently supports a subset of the SH7750 cpu core. The SH4 part of the QEMU user space emulator is functional but currently limited to static binaries only.

Areas that need improvement are:

- FPU support - FPU flags need better emulation

Current SH4 Status

The QEMU user space emulator currently supports a subset of the SH7750 cpu core. The SH4 part of the QEMU user space emulator is functional but currently limited to static binaries only.

Areas that need improvement are:

- FPU support - FPU flags need better emulation
- DSP support - The DSP is currently not supported

Current SH4 Status

The QEMU user space emulator currently supports a subset of the SH7750 cpu core. The SH4 part of the QEMU user space emulator is functional but currently limited to static binaries only.

Areas that need improvement are:

- FPU support - FPU flags need better emulation
- DSP support - The DSP is currently not supported
- Dynamically linked binary support

Current SH4 Status

The QEMU user space emulator currently supports a subset of the SH7750 cpu core. The SH4 part of the QEMU user space emulator is functional but currently limited to static binaries only.

Areas that need improvement are:

- FPU support - FPU flags need better emulation
- DSP support - The DSP is currently not supported
- Dynamically linked binary support

All SH4 specific changes to the user space emulator are currently included in the official QEMU CVS tree.

Outline

Overview

The QEMU system emulator provides a way for the user to execute native and non-native operating systems inside of Linux. For example, closed source operating systems for x86 may be executed on non-x86 processors using the QEMU system emulator.

The QEMU system emulator emulates the entire target hardware platform including processor, memory and I/O devices. The software used inside the QEMU environment is an entire operating system or stand alone software such a boot loader or RTOS.

Usage Examples

How to use the QEMU system emulator to. . .

Usage Examples

How to use the QEMU system emulator to. . .

. . . boot x86 software from a virtual CDROM

```
qemu -boot d -cdrom winxp.iso /dev/zero
```

Usage Examples

How to use the QEMU system emulator to...

...boot x86 software from a virtual CDROM

```
qemu -boot d -cdrom winxp.iso /dev/zero
```

...boot x86 linux kernel using serial console

```
qemu -nographic -kernel bzImage /dev/zero
```

Usage Examples

How to use the QEMU system emulator to...

...boot x86 software from a virtual CDROM

```
qemu -boot d -cdrom winxp.iso /dev/zero
```

...boot x86 linux kernel using serial console

```
qemu -nographic -kernel bzImage /dev/zero
```

...boot SH4 linux kernel

```
qemu-system-sh4 -M r2d -kernel zImage
```

Usage Examples

How to use the QEMU system emulator to...

...boot x86 software from a virtual CDROM

```
qemu -boot d -cdrom winxp.iso /dev/zero
```

...boot x86 linux kernel using serial console

```
qemu -nographic -kernel bzImage /dev/zero
```

...boot SH4 linux kernel

```
qemu-system-sh4 -M r2d -kernel zImage
```

To enable SH4 system emulation in QEMU please configure the source using `./configure -target-list=sh4-softmmu`

Current SH4 Status

The QEMU system emulator currently supports emulation of a subset of the SH7750 cpu. The emulator is not yet able to fully run a Linux kernel, but basic support for MMU, timers and serial ports are in place.

Current SH4 Status

The QEMU system emulator currently supports emulation of a subset of the SH7750 cpu. The emulator is not yet able to fully run a Linux kernel, but basic support for MMU, timers and serial ports are in place.

A subset of the R2D-PLUS board is currently emulated.

Current SH4 Status

The QEMU system emulator currently supports emulation of a subset of the SH7750 cpu. The emulator is not yet able to fully run a Linux kernel, but basic support for MMU, timers and serial ports are in place.

A subset of the R2D-PLUS board is currently emulated.

- No boot loader software is currently in place
  - U-boot may be a good candidate
- RTL8139 emulation already provided by QEMU

Current SH4 Status

The QEMU system emulator currently supports emulation of a subset of the SH7750 cpu. The emulator is not yet able to fully run a Linux kernel, but basic support for MMU, timers and serial ports are in place.

A subset of the R2D-PLUS board is currently emulated.

- No boot loader software is currently in place
  - U-boot may be a good candidate
- RTL8139 emulation already provided by QEMU

The CVS version of the SH4 system emulator is currently lacking many features. Patches will be sent to `qemu-devel` in the near future.

Current SH4 Status

Areas and their current status:

Current SH4 Status

Areas and their current status:

- CPU core - 80% done
  - Supervisor and user instructions separation needed
  - Exceptions need more attention to work with delay slots

Current SH4 Status

Areas and their current status:

- CPU core - 80% done
  - Supervisor and user instructions separation needed
  - Exceptions need more attention to work with delay slots
- Timers - 95% done
  - TMU block emulation supports SH7750 and SH7780
  - Capture and RTC modes not supported

Current SH4 Status

Areas and their current status:

- CPU core - 80% done
  - Supervisor and user instructions separation needed
  - Exceptions need more attention to work with delay slots
- Timers - 95% done
  - TMU block emulation supports SH7750 and SH7780
  - Capture and RTC modes not supported
- Interrupt controller - 25% done
  - INTC kernel code should be ported to QEMU
  - Current interrupt controller code is very primitive

Current SH4 Status

Areas and their current status:

- CPU core - 80% done
  - Supervisor and user instructions separation needed
  - Exceptions need more attention to work with delay slots
- Timers - 95% done
  - TMU block emulation supports SH7750 and SH7780
  - Capture and RTC modes not supported
- Interrupt controller - 25% done
  - INTC kernel code should be ported to QEMU
  - Current interrupt controller code is very primitive
- Serial ports - 25% done
  - Need to rewrite SCI and SCIF emulation code
  - Current code does not support interrupts

Current SH4 Status

Areas and their current status:

- CPU core - 80% done
  - Supervisor and user instructions separation needed
  - Exceptions need more attention to work with delay slots
- Timers - 95% done
  - TMU block emulation supports SH7750 and SH7780
  - Capture and RTC modes not supported
- Interrupt controller - 25% done
  - INTC kernel code should be ported to QEMU
  - Current interrupt controller code is very primitive
- Serial ports - 25% done
  - Need to rewrite SCI and SCIF emulation code
  - Current code does not support interrupts
- PCI - 0% done
  - Needs basic r2d board support code for interrupts
  - Nothing done yet

Summary

- The QEMU user space emulator for SH4 is useful *now*.
- The QEMU system emulator for SH4 needs more work.