SeaBIOS in a virtualized environment

Kevin O'Connor
kevin@koconnor.net

So what is SeaBIOS?

- The QEMU/KVM “firmware” on x86
- First software executed in guest
- Implements 16bit BIOS functionality

History of SeaBIOS

- Based on “Bochs BIOS”
- Wanted to run BIOS on real hardware (coreboot)
- Began in early 2008
- Adopted as default BIOS for QEMU in late 2009

Why replace Bochs BIOS?

- Uses bcc
- Has lots of 16bit assembler
- rombios.c - ~11,000 lines - 1/3rd inline assembly
- Difficult to add new functionality
- However, great reference of real-world BIOS interfaces

Initial goals

- Use modern tools (gcc, ld, gas)
- Replace assembler with C code where possible
- Standardize entry points
- Run as much code in 32bit “flat” mode as possible
- Support real hardware
  - Real hw delays
  - Full optionrom support

Gcc and 16bit mode

- Uses “.code16gcc” GNU assembler feature
- Segmented memory is a headache
- Gcc does use more stack
  - Bad for old 16bit programs

Code example

- Most code is regular 32bit “flat” C code.
- Code for 16bit and 32bit segmented mode need to wrap non-stack memory accesses with macros.

```
void handle_1588(struct bregs *regs)
{
    u32 rs = GET_GLOBAL(RamSize);
    if (rs > 64*1024*1024)
        regs->ax = 63 * 1024;
    else
        regs->ax = (rs - 1*1024*1024) / 1024;
    regs->flags &= ~F_CF;
}
```

Recent features

- Improved optionrom support
  - BIOS Boot Specification (BBS)
  - Post Memory Manager (PMM)
- Virtio disk support
- Bootsplash JPEG support
- Expanded fw_cfg interface

Features (cont)

- USB support
  - UHCI / OHCI / EHCI (basic support)
  - Keyboard / mouse
  - Disk booting
- Fast booting (Parallelize hardware init)
- Multiple PCI buses
- Boot from program / floppy image in flash

Next Steps?

- More customized ACPI tables and protocol for it between SeaBIOS / QEMU
- Managing boot order between SeaBIOS and QEMU
- Use SeaBIOS in Xen?
- Use gcc for “LGPL vgabios” too?

Questions???