

HAIKU[®]: ARM & PPC Support

François Revol
revol@free.fr



Haiku

- Free Software Operating System
- Not a GNU/Linux distro
 - Our own kernel
 - Our own GUI
- Soon in R1/beta1
- Officially supports x86 (32bit)
- X86_64 mostly working



PPC Machines

- G3, G4... vs embedded cpus (Book-E)
 - Same userland, but different MMU & other things
 - Book-E MMU = TLBs and that's it
- OpenFirmware but also now U-Boot, CFE
- Usually have a PCI bus
 - But also internal SoC busses on Book-E



PPC Targets

- G3/G4 Macs (OF + PCI)
- Acube Sam460ex (AMCC460 Book-E, U-Boot)



ARM Boards

- Always different
 - Each arch version has different MMU & insns...
 - SoC components (IRQ, timer) and memory mapping
 - No enumerated system bus like PCI
 - SD, SPI, NAND, USB... at hardcoded addresses
- Usually U-Boot as firmware
 - But sometimes not (Qi, Raspberry Pi blob...)
 - **Old** and **forked** version of U-Boot (not upstreamed)



ARM Targets

- Initially ARMv5 (Gumstix Verdex)
- Now ARMv7 (Raspberry Pi 2, CubieBoard 4)
 - More standardized



Haiku loader

- Similar to NetBSD loader
- Needs a BIOS to call (for video, disk I/O)
 - Or an “initrd”-like tgz of kernel+modules
- Displays the splash
- Text menu (boot options, volume selection)
- Sets up custom MMU mappings
- Loads the kernel and modules



Das U-Boot

- Firmware for embedded hardware
- Can support booting several kernels
 - But usually only Linux support compiled in ☹️
- Old Linux protocol used “atags”
- New protocol passes an FDT
- They added a BIOS-like API for NetBSD
 - But it's never compiled in either ☹️



Flattened Device Tree

- Similar to OpenFirmware's device tree
- But serialized to a file
 - DTS (source)
 - DTB (blob)
- Libfdt provides a parser
 - Dual license: BSD/GPL so we can use in the kernel
- Allows sharing work
 - In practice Linux & BSD DTS are different 🤖



Flattened Device Tree Source

```
/dts-v1/;

#include/ "bcm2835.dtsi"

/ {
    model = "Raspberry Pi (BCM2835)";
    compatible = "raspberrypi,model-a", "raspberrypi,model-
b",
                "broadcom,bcm2835-vc", "broadcom,bcm2708-
vc";

    memreserve = <0x08000000 0x08000000>; /*Set by
VideoCore*/

    cpus {
        #address-cells = <1>;
        #size-cells = <0>;
        cpu@0 {
            compatible = "arm,1176jzf-s";
            device_type = "cpu";
            reg = <0>; /* CPU ID=0 */
            Clock-frequency = <700000000>; /* 700MHz */
        };
    };

    memory {
        device_type = "memory";
        reg = <0 0x8000000>; /* 128MB, Set by VideoCore */
    };

    aliases {
        uart0 = &uart0;
    };

    chosen {
        bootargs = ""; /* Set by VideoCore */
        stdin = "uart0";
        stdout = "uart0";
    };
};
```



The Kernel

- Some things not easy to put in modules
 - IRQ and timer support
 - MMU
 - CPU feature emulation (FPU)
- Need to select at runtime



Modules

- Organized in a filesystem tree
- Drivers in a subtree
- Old-style (BeOS) drivers & bus_managers
- New-style modules
- `~/config/add-ons/kernel/bus_manager/pci`
binary exports `/bus_manager/pci/v1` and `.../v2`
- Modules explicitly declare dependencies
 - No need for udev 😊



PPC Status

- G3/G4: Used to work long ago
 - VM code rewritten to match x86 style
 - Fails in QEMU
 - OpenBIOS maps PCI at kernel address
 - And fails to declare it in translations
- Sam460ex
 - VM setup tricky in loader: TLB only + no real mode
 - Almost there
 - No splashscreen yet ☹️



ARM Status

- Splashscreen works 😊
- VM setup in loader needs some cleanup
 - Some board-specific mappings hardcoded
- Usually fails at mounting /boot
 - But there's a NOR Flash driver



Common Roadmap

- Implement FDT bus_managers
- Implement required SoC drivers
 - IRQ
 - Timer
 - USB
- Start userland from a USB stick



References

- <http://www.denx.de/wiki/U-Boot/UBootFdtInfo>
- Ithamar Adema, *ARM roadmap* (BeGeistert 28)
- `docs/develop/ports/` in the source tree
- *Book-E Linux port paper*

- Questions?

