who am I

Kyösti Mälkki

- with coreboot project since 2010
- past Google Summer of Code intern
- usbdebug, backports, refactoring
- SerialICE parser
- more interest on industrial use

what is coreboot

- coreboot is a free software project to replace proprietary firmware of a computer
- minimalistic approach on hardware initialisation
- proven to scale from tablets to clusters

early days as linuxBIOS

- started 1999 at a super-computing cluster
- remove BIOS, put linux inside the flash
- reuse existing drivers for exotic hardware
- kernel was unable to configure chipset, more early platform initialisation was required
- cache-as-RAM allows GNU compiler tools

linuxBIOS objectives

- reliable
- fast
- flexible
- unattended
- write firmware in C, please

early days as coreboot

- rename project to coreboot (+payload)
- minimal footprint, no BIOS services
- no longer limited to booting linux
- active community around AMD boards
- lack of documentation on Intel side

early days as coreboot

- funding from German government for a ruggerized laptop support, with open-source firmware
- Lenovo ThinkPad support T60/X60 triggers new wave of interest
- new objectives:
  - security by validation, quality assurance
  - power management

coreboot in ChromeOS

- x86 models with coreboot released 2012
- 2013: annual sales 2.9 million units
- 2016: forecast 8 million units
- “Chromebooks outsold Macs for the first time in the US”

  Tom Warren on May 19, 2016

coreboot in ChromeOS

- new objectives:
  - latest silicons, industry involvement
  - large-scale mass production
  - verified boot
  - forced firmware update deployment

firmware image overview

- coreboot proper
  - GPL v2 license
  - mostly C, some assembler
  - stages: bootblock, romstage, ramstage
  - build info: config, revision
- payloads

firmware image overview

- proprietary components
  - Intel ME, MRC, FSP
  - AMD SMU, PSP, binaryPI
  - USB3 xHCI, VBIOS, CPU microcode

File Edit View Search Terminal Help

```
Total Time: 224,032
kmad@kinetic:~/Desktop/coreboot-talk/lumpy$ ./cbfstool coreboot.rom print
Performing operation on 'COREBOOT' region...
Name                           Offset     Type           Size
cbfs master header             0x0        cbfs header    32
fallback/romstage              0x80       stage          34316
cpu_microcode_blob.bin         0x8700     microcode      22528
config                         0xdf80     raw            359
revision                       0xe140     raw            569
cmos_layout.bin                0xe3c0     cmos_layout    1368
spd.bin                        0xe980     spd            1536
payload_config                 0xefc0     raw            1543
payload_revision               0xf600     raw            243
(empty)                        0xf740     null           1880
mrc.cache                      0xfec0     mrc_cache      65536
fallback/ramstage              0x1ff00    stage          71863
pci8086,0116.rom               0x31800    optionrom      65536
fallback/dsdt.aml              0x41880    raw            13724
fallback/payload               0x44e80    payload        60372
(empty)                        0x53ac0    null           312280
mrc.bin                        0x9fec0    mrc            195732
(empty)                        0xcfbc0    null           195352
bootblock                      0xff700    bootblock      1952
kmad@kinetic:~/Desktop/coreboot-talk/lumpy$
```

x86 bootblock

- assembler, some C built with romcc
- enter 32-bit protected mode
- enhance boot media (SPI) access

romstage

- chipset low-level IO configurations
- enable debug log
- RAM configuration and training
- migrate away from cache-as-RAM

ramstage

- CPU: SMP, SMM, microcode update
- PCI enumeration, resource allocator
- create ACPI and SMBIOS tables
- store cache RAM training results to SPI flash

File Edit View Search Terminal Help

```
  0:1st timestamp                                    510
  1:start of rom stage                               38,387 (37,876)
  2:before ram initialization                        39,263 (875)
  3:after ram initialization                         602,490 (563,227)
  4:end of romstage                                  603,667 (1,177)
  8:starting to load ramstage                        605,097 (1,429)
 15:starting LZMA decompress (ignore for x86)        605,106 (8)
 16:finished LZMA decompress (ignore for x86)        624,114 (19,007)
  9:finished loading ramstage                        624,132 (18)
 10:start of ramstage                                624,176 (43)
 30:device enumeration                               624,210 (33)
 40:device configuration                             626,903 (2,693)
 50:device enable                                    636,434 (9,530)
 60:device initialization                            636,964 (529)
 70:device setup done                                755,718 (118,754)
 75:cbmem post                                       755,722 (3)
 80:write tables                                     1,060,113 (304,391)
 90:load payload                                     1,066,585 (6,472)
 15:starting LZMA decompress (ignore for x86)        1,066,701 (115)
 16:finished LZMA decompress (ignore for x86)        1,089,941 (23,240)
 99:selfboot jump                                    1,089,962 (20)

Total Time: 1,089,440
kmad@kinetic:~/Desktop/coreboot-talk/lumpy$
```

first boot with RAM training

File Edit View Search Terminal Help

```
   0:1st timestamp                                    512
   1:start of rom stage                               34,195 (33,682)
   2:before ram initialization                        35,066 (870)
   3:after ram initialization                         95,236 (60,170)
   4:end of romstage                                  96,421 (1,184)
   8:starting to load ramstage                        97,861 (1,440)
  15:starting LZMA decompress (ignore for x86)        97,870 (9)
  16:finished LZMA decompress (ignore for x86)        119,505 (21,634)
   9:finished loading ramstage                        119,523 (18)
  10:start of ramstage                                119,567 (43)
  30:device enumeration                               119,599 (32)
  40:device configuration                             122,331 (2,731)
  50:device enable                                    131,860 (9,529)
  60:device initialization                            132,387 (527)
  70:device setup done                                192,222 (59,834)
  75:cbmem post                                       192,225 (3)
  80:write tables                                     195,482 (3,256)
  90:load payload                                     201,230 (5,747)
  15:starting LZMA decompress (ignore for x86)        201,344 (114)
  16:finished LZMA decompress (ignore for x86)        224,534 (23,190)
  99:selfboot jump                                    224,554 (19)

Total Time: 224,032
kmad@kinetic:~/Desktop/coreboot-talk/lumpy$
```

boot with cached RAM training

File Edit View Search Terminal Help

```
kmad@kinetic:~/Desktop/coreboot-talk/lumpy$ cat resume-ts
15 entries total:

   0:1st timestamp                                  512
   1:start of rom stage                             34,189 (33,677)
   2:before ram initialization                      35,074 (884)
   3:after ram initialization                       43,758 (8,683)
   4:end of romstage                                94,484 (50,726)
   8:starting to load ramstage                      96,020 (1,536)
  15:starting LZMA decompress (ignore for x86)      96,029 (8)
  16:finished LZMA decompress (ignore for x86)      115,222 (19,193)
   9:finished loading ramstage                      115,240 (18)
  10:start of ramstage                              115,272 (31)
  30:device enumeration                             115,306 (33)
  40:device configuration                           117,998 (2,692)
  50:device enable                                  127,536 (9,537)
  60:device initialization                          128,066 (529)
  70:device setup done                              172,979 (44,913)
  98:ACPI wake jump                                 179,334 (6,355)

Total Time: 178,815
kmad@kinetic:~/Desktop/coreboot-talk/lumpy$
```

resume from S3 suspend

payload stage

- license of your choice
- user interaction
- boot media
- filesystem
- OS bootloader

payload choices

- SeaBIOS
- GRUB2
- iPXE
- memtest86+
- TianoCore

more payloads choices

- DepthCharge
- linux-as-a-payload
- FILO
- tint, bayou, coreinfo, nvramcui
- libpayload support library

acquiring coreboot

- strictly coupled with a hardware platform
- purchase with new hardware
- after-market and refurbished laptops
- do it yourself, with attitude

ChromeOS hardware

- latest silicons and chipsets
- affordable, available
- most have proprietary chipset initialisation
- limitations on RAM, SSD, connectivity
- somewhat hacker-friendly hardware

libreboot and MiniFree

- libreboot is in GNU Project since May 2016
- built from coreboot codebase with no proprietary components present in the firmware
- pre-built and tested firmware images available
- MiniFree sells hardware with libreboot
- FSF RYF certified systems

invest in new hardware

- PC Engines apu1 and apu2
  (also under pfSense brand)
- Purism (not yet!)
- RISC-V and lowRISC
- Power8, IBM OpenPower and TALOS

DIY preparing the build

- use of coreboot reference toolchain for compiling is highly recommended
- check repository of last-known-good builds
- recent coreboot images are reproducible
- utils: cbfstool, ifdtool, flashrom

DIY menuconfig

[live demo]

DIY write attempt

- always make backup of your original flash device
- flashrom -p internal -w coreboot.rom

  !!! VERIFY FAILED !!!
  Please do not reboot or power off
- external programming is a must to break away from vendor UEFI or BIOS today

DIY hardware tools

- laptop disassembly required
- IC clip for SOIC-8 or SOIC-16 package
- soldering skills for WSON8
- external programmer with flashrom support
  - BeagleBone Black, Raspberry Pi
  - FTDI FT232H, Bus Pirate

DIY Hey it works!

- congratulations!
- future firmware programming happens without effort from commandline, unless you secure it
- adjust look-and-feel of the payload
- consider fallback/normal recovery layout

DIY frustration strikes?

- FreeNode IRC channels:
  #flashrom, #coreboot and #libreboot
- confusion about pinouts
- mechanical issues with IC clips
- insufficient power supply
- it's rarely permanently bricked ...

end of part 1

[demo time after Q & A]

on firmware security

- firmware is everywhere and it is connected
- perfect hiding place, mask ROMs
- hard to analyze
- should be root-of-trust in the platform

on root-of-trust

- cannot trust the application if you cannot trust the OS
- cannot trust the OS if you cannot trust the bootloader
- cannot trust the bootloader if you cannot trust the firmware
- cannot trust the firmware if it can be modified without long-term physical access

SPI flash protection

- write-protection lock pin
- restricted SPI opcode set
- SMM write-protect
- Intel Flash Descriptor, Protected Ranges
- consequences: flashrom cannot backup or overwrite contents of SPI flash

towards verified boot

- bootblock is signed with vendor's private key
- custom firmware cannot boot if the device manufacturer decides so
- Intel Boot Guard / Verified Boot
- AMD Platform Security Processor

towards verified boot

- public key contained in RO firmware
- two copies of RW firmware, one slot for update
- trusted RO firmware verifies RW firmware
- RW firmware verifies kernel
- key revocation, anti-rollback (with TPM)

for students

- Google Summer of Code 2017
  (likely but unconfirmed)
- a paid internship from June to August, we have ideas what to do but convince us with your own project plan
- you need to pass some simple tasks before sending in application (early March, unconfirmed)
- get some of the best in the industry to act as your mentor

where to meet us

- FOSS events and plugfests
  33rd CCC, Hamburg Dec 2016
  FOSDEM, Brussels Feb 2017

- coreboot convention
  Denver, June 2017

more information

Questions?

https://www.coreboot.org
https://www.flashrom.org
https://libreboot.org

coreboot presentations (YouTube):

http://bit.ly/2cTK1j2	2016 coreboot conference in San Fransisco
http://bit.ly/2cSHolu	2015 coreboot conference in Bonn
http://bit.ly/2cnBepJ	2014 ChromeOS Firmware summit

Thank You

[don't leave, demo follows]

e-mail: kyosti.malkki@gmail.com
IRC: kmalkki on #coreboot FreeNode