

Booting over HTTP

Zoltan Arnold NAGY
zoltan@NetBSD.org



Eötvös Loránd University,
Budapest, Hungary
November 20, 2010

Agenda

- ▶ Who am I?
- ▶ Google Summer of Code and NetBSD
- ▶ General overview
- ▶ Challenges
- ▶ Demo
- ▶ Questions and Answers



Who am I?

- ▶ BSc in CS (minor in math), working towards MSc
- ▶ Full-time software engineer, mostly Java EE
- ▶ Long time Linux user
- ▶ NetBSD developer since 1 Oct



Google Summer of Code and NetBSD



- ▶ Google's global program for student developers
- ▶ Main goal is to get them involved in doing open-source software
- ▶ NetBSD has been participating since the beginning (2005)
- ▶ "Implementing HTTP support for libsa" has been selected



General overview

- ▶ The boot process:
 - ▶ use DHCP to get an IP and gateway address
 - ▶ use TFTP to load the NetBSD bootloader
 - ▶ the bootloader decides how to proceed
- ▶ Why do we need HTTP support?
 - ▶ corporate networks are usually heavily firewalled, yet HTTP is allowed
 - ▶ HTTP is very widespread, TFTP isn't so
- ▶ Requirements
 - ▶ Should be architecture agnostic
 - ▶ Internal APIs are available (libsa)
 - ▶ `netif_driver` for defining NIC drivers (probe, get, put, ...)
 - ▶ `fs_ops` for defining filesystem operations (read, write, ...)
- ▶ For amd64/i386, PXE is available, and had a `netif_driver` implementation
- ▶ Adding it to a new architecture is easy: just write the `netif_driver`!



Challenges

- ▶ The previous implementation was UDP only
 - ▶ The IP and UDP code wasn't separated → had to refactor most of the code
- ▶ Had to implement PXE's Universal Network Driver Interface (UNDI)
 - ▶ Very messy
 - ▶ Polling required
 - ▶ Almost direct access to the NIC's ringbuffer
- ▶ We needed a minimalistic TCP stack
 - ▶ Could have reused an existing BSD-licensed TCP stack, but they were too full-featured ;)
- ▶ HTTP is easy, base64 encoder needed a bit of magic



What we can, might do, can't do

- ▶ Can:
 - ▶ boot the kernel from an HTTP address
 - ▶ authenticate with HTTP basic auth
- ▶ In the future:
 - ▶ IPv6 support
 - ▶ DNS support
- ▶ Might do HTTPS support in the future



Demo



Questions and answers!

