Alte Software auf neuer Hardware, Neue Software auf alter Hardware

Ignatios Souvatzis

Institut für Informatik **NetBSD Project**

<ignatios@cs.uni-bonn.de> <is@netbsd.org>

http://theory.cs.uni-bonn.de/~ignatios/

Einführung

- Manchmal will man alte Software auf neuen Rechnern weiterbenutzen.
  - Hersteller nicht mehr verfügbar
  - Umstellung nicht wirtschaftlich
  - historisches Interesse
- Manchmal will man neue Software auf seinen alten Rechnern haben.
  - auf bestimmte Peripherie angewiesen
  - historisches Interesse

Software neukompilieren

Voraussetzung

- Sourcen vorhanden
- Compiler vorhanden
- Betriebssystem geeignet

Kein Problem!

Nicht möglich

- Nur Binärcode vorhanden
- Betriebssystem ungeeignet für die Anwendung

Was tun?

Was tun?

- OS-Emulatoren *(faktisch: alt auf neu)*
- Emulation von alt-OS + alt-CPU mit Hardware-Unterstützung von neuer CPU
- Virtuelle Maschinen auf gleicher Hardware
- Maschinenemulation
- Portierung des Betriebssystems *(neu auf alt)*

OS-Emulation

System Call Emulation

- je nach OS des Programms andere System Call - Tabelle
- alternativer Dateibaum, um shared libraries zu finden
- schnell

OS-Emulation

Beispiele System Call Emulation

z.B.

- SunOS4, SVR4-Emulation auf NetBSD/Sparc32
- SVR4, Linux, FreeBSD auf NetBSD/i386
- HP-UX auf NetBSD/m68k

OS-Emulation

Beispiel

Netscape4 und Acroread (Linux Binary) auf NetBSD/i386

```
laplace: {16} pwd
/usr/pkg/Acrobat4/Reader/intellinux/bin
laplace: {17} file acroread
acroread: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
 for GNU/Linux 2.0.0, dynamically linked (uses shared libs), not stripped
```

OS-Emulation

virtuelle Maschine auf gleicher CPU

- z.B. IBM-Grossrechner, Wine, Wabi
- erlaubt, ganzes OS zu fahren.

Emulation von Alt-CPU mit Hardwareunterstützung

Der Vollständigkeit halber erwähnt

- schnell
- Beispiel
  - PDP11-Programme auf VAX/VMS
  - ARM26-Programme auf ARM32
    (hier nur Interupt und call/return anders)

Emulation der CPU ohne Hardwareunterstützung

- langsam, aber
- vollständige Kontrolle
- Beschleunigung durch Just In Time - Übersetzung
- ganzes OS kann ausgeführt werden
- teilweise einzelne Programme (MacOS-68k auf MacOS-PPC)

Live-Demonstration SimH-VAX (1)

```
laplace: {25} cat startup
load -r /usr/pkg/share/simh/ka655.bin
set rq0 ra92
at rq0 netbsd.dsk
boot cpu
laplace: {26} simh-vax startup

VAX simulator V2.10-0

KA655-B V5.3, VMB 2.7
Performing normal system tests.
40..39..38..37..36..35..34..33..32..31..30..29..28..27..26..25..
24..23..22..21..20..19..18..17..16..15..14..13..12..11..10..09..
08..07..06..05..04..03..
Tests completed.
>>>
```

Live-Demonstration SimH-VAX (2)

```
>>>boot dua0:
(BOOT/R5:0 DUA0)
  2..
-DUA0
  1..0..
>> NetBSD/vax boot [1.11 Wed Sep 11 10:05:08 UTC 2002] <<
>> Press any key to abort autoboot 0
nfs_open: must mount first.
open netbsd.vax: Device not configured
> boot netbsd
1517664+210032 [153168+110527]=0x1e64f4
[ using 264228 bytes of netbsd ELF symbol table ]
Copyright (c) 1996, 1997, 1998, 1999, 2000, 2001, 2002
    The NetBSD Foundation, Inc.  All rights reserved.
Copyright (c) 1982, 1986, 1989, 1991, 1993
    The Regents of the University of California.  All rights reserved.
NetBSD 1.6 (GENERIC) #0: Wed Sep 11 10:14:37 UTC 2002
...
```


Portierung des Betriebssystems

- OpenVMS: 2 Architekturen, demnächst 3
- Solaris: 2 Architekturen (oder 1) (oder doch wieder 2)
- Windows NT und verwandte: urspringl. ca. 5, jetzt 1
- Windows CE: ca. 5
- FreeBSD: ursprünglich 1, jetzt ca. 4
- Linux: 1,6,viele (je nach Definition von Linux)
- NetBSD: 13 CPUs, 39 (1.6) / 51 (-current) Architekturen, Tendenz steigend.

NetBSD-Portierung

Voraussetzungen

- CPU mit 32 bit oder 64 bit - Registern
  36 bit in Arbeit
- Paged MMU
- faktisch: `gcc`- und `binutils`- Unterstützung

Vorgehensweise

Siehe V. Ushakov, Porting NetBSD to JavaStation:
http://eurobsdcon.org/papers/ushakov.ps

- Hardware-Dokumentation besorgen
- Bootlader schreiben (Firmware-Dokumentation)
- Debug-Output über serielle Konsole (oder Firmware) schreiben und testen
- der Rest

Modularität erleichtert weiteres Vorgehen

Idealfall:

- PCI- Bustreiber schreiben
- Rest ist schon da

Notfalls:

- Bustreiber schreiben
- Busattachments für hoffentlich schon vorhandene Chiptreiber schreiben

Portierung auf PDP10

- gcc-Erweiterung auf 9-bit- ,,Bytes“ (Lars Brinkhoff)
- pcc-Portierung (Anders Magnusson)
- Filesystem umschreiben auf 512*9bit – Festplatten (to be done)
- Effizientes Testen setzt Emulator voraus — Originalhardware teuer, empfindlich
- Status: http://www.netbsd.org/Ports/pdp10

Literatur

- http://simh.trailing-edge.com/
- http://klh10.trailing-edge.com/
- http://www.netbsd.org/, .../Ports/vax , .../Ports/pdp10
- http://eurobsdcon.org/papers/ushakov.ps “Porting...”
- http://theory.cs.uni-bonn.de/~ignatios/

Ich danke für Ihre Aufmerksamkeit!