

Installation von Debian Linux auf IBM PSeries Cross Compiler

Markus Schaber markus.schaber@ulm.ccc.de

Volker Birk dingens@bumens.org

Ulrich Dangel udangel@gmx.de



Überblick

- Warum überhaupt ein Cross Compiling?
- Überblick über das Cross Compilen
- Debian Paket
- Vorgehensweise
- Details
- Benutzte Programme



Warum Cross Compiling?

- Kernel muss PPC64-Code sein
- Keine fertigen Bootdisks oder ähnliches außer kommerzielle Distribution SuSE Enterprise Server
- Keine bekannte und öffentlich zugängliche Linux-PPC64 Maschine vorhanden
- Es können jedoch weiterhin für 32-Bit PPC kompilierte Programme benutzt werden



Überblick über das Cross Compilen

- Im allgemeinen 4 Schritte zum Ergebnis
 - Binutils für die neue Plattform erstellen
 - gcc ohne shared libraries kompilieren
 - GnuLibc kompilieren
 - gcc mit shared libraries kompilieren



Debian Paket - Toolchain Source

- Ein fertiges Debian Paket names toolchain-source existiert
- Laut Doku und Paketbeschreibung das perfekte Tool für das Cross Compilen
- Leider funktionierte das ganze zufälligerweise für ppc64 nicht :(
- Damit ist die schnellste Lösung selber die Pakete zu übersetzen.



Vorgehensweise

- Als Ausgangsplattform Mac G3 mit PPC Prozessor
- Neuste Binutils aus dem CVS kompiliert ohne Patches
- gcc mit entsprechenden Patches kompiliert
- Fertig, da jetzt Kernel kompiliert werden kann, da der Kernel keine Shared Libraries und auch keine Glibc benötigt. :)



Details - Kompilierer Durchlauf 1

- Binutils kompilieren.
 - *-enable-targets=powerpc64-linux -prefix=/opt/ppc64 -disable-nls -disable-shared*
- gcc kompilieren.
 - *-target=powerpc64-linux -enable-languages=c -disable-shared -disable-nls -disable-threads -enable-__cxa_atexit -prefix=/opt/ppc64*
- glibc kompilieren
 - *-target=powerpc64-linux -without-cvs -enable-add-ons -enable-shared -with-headers=<PPC64 Kernel headers> -disable-profile -disable-sanity-checks -prefix=/opt/ppc64/powerpc64-linux*



Details - Kompilier Durchlauf 2

- gcc mit Shared Libraries kompilieren
 - *-target=powerpc64-linux*
-enable-languages=c,c++,f77 -enable-shared
-enable-__cxa_atexit -disable-nls
-prefix=/opt/ppc64
- glibc wird nun auch als Shared Libraries kompiliert
 - *-target=powerpc64-linux -without-cvs*
-enable-add-ons -enable-shared
-with-headers=<PPC64 Kernel headers>
-disable-profile -prefix=/opt/ppc64/powerpc64-linux



Benutzte Programme und Sonstiges

- Programme

- Sehr Sehr viele
- Die wichtigsten: patch, gcc, wget und vim :)

- Informationen

- <http://linuxppc64.org>
- <http://www.google.com>



Diese HTML-Datei enthält spezielle CSS2-Anweisungen, die für den [Präsentationsmodus](#) des Web-Browsers [Opera](#) erstellt wurden. Sofern sie diesen Browser benutzen, können sie mit der F11- Taste in diesen Modus (und wieder zurück) schalten. Die Präsentation wurde für eine Auflösung von 1280x1024 Bildpunkten erstellt.

Debian Installation auf IBM pSeries

Ein Überblick



- Vortragsreihe "Chaos-Seminar"
- Veranstalter: CCC, Erfa-Kreis Ulm
 - <http://www.ulm.ccc.de/>
 - mail@ulm.ccc.de
 - Montagstreff: Infos und Anfahrtsskizze auf der Webseite
- Referenten:
 - Ulrich Dangel - udangel@gmx.de
 - Volker Birk - dingens@bumens.org
 - Markus Schaber - markus.schaber@ulm.ccc.de
- Vortrag kommt online
 - <http://www.ulm.ccc.de/~schabi/pseries/>

Inhaltsübersicht

(So in etwa...)

- Warum das ganze
- Was ist pSeries
- Cross-Compiler
- Kernel
- Openfirmware
- Die eigentliche Installation
- Was noch zu tun wäre...
- Post-Bier-Fragen bei Parasco

Was ist pSeries

- <http://www-1.ibm.com/servers/de/eserver/pseries/p630/>
- Bis zu 4 Power4 Prozessoren
- Optimiert für IO
- Preis: Zwischen 18kEUR und 50kEUR (natürlich netto)
- AIX standardmäßig (aber: es saugt!)
- Linux: SuSE Enterprise Server, ca. [3kEUR](#)
- Inzwischen auch RedHat
- Unter "Wichtige Allianzen" auch SCO :-)
- DVD-Brenner brennt wirklich *nur* DVDs!

Openfirmware

- Entspricht BIOS beim PC
- Ursprünglich von SUN
- IEEE Std 1275-1994
- Hat auch Apple im Programm
- Hat einen eigenen [Song](#) :-)

Openfirmware (2)

- Initialisiert die ganze Hardware
- Lädt und startet OS
- Implementiert in Forth :-)
- Ist frei Programmierbar
- Bietet (umständliche) Kommandozeile
- Plug & Play Plattformunabhängig

Netz-Installation

- Howto für 32-Bit PPC-Maschinen existiert.
- Wie immer: Doku passt nur fast!
- Voraussetzungen:

- Selbes Subnetz
- DHCP/Bootp-Server
- TFTP-Server
- Passendes Boot-Image auf dem Server
- Theorie: Dann geht alles ganz einfach...
- Praxis: Werbe-Vertrag mit Aspirin? :-)

Die Praxis: Openfirmware 1

- Laut Doku:
 - beim Booten im SMS-Menü mit F8 Openfirmware-Menü aktivieren
- Beim Booten:
 - Die Stelle mit dem Menü kommt nicht!
- Lösung:
 - Bei "ausgeschaltetem" Rechner
 - Über serielle Schnittstelle
 - Aktivieren des Menüs in Management-Konsole

Die Praxis: Openfirmware 2

- Soll:
 - Openfirmware holt Bootloader
 - bootloader holt kernel und initrd über OF
- Ist:
 - laut ML-Archiv ist Openfirmware buggy
 - OF wieder Bootloader statt Kernel
- Lösung:
 - mittels "make zImage.initrd" alles in ein Image
- Nächstes Problem:
 - dokumentierter Pfad funktioniert nicht
- Fazit:
 - Trial & Error führt zur Lösung...

Die Praxis: Openfirmware 3

- Soll:

- Openfirmware holt Boot-Image
- Ist:
 - Image relativ groß, braucht einige Sekunden
 - TFTP-Server ARP eintrag expired
 - Openfirmware buggy: Bricht bei ARP-Request ab.
- Lösung:
 - Lösung: "arp -s clientip clientmac" auf Server
 - ... und hoffen :-)

Die Praxis: Ich seh schwarz

- Openfirmware-Framebuffer-Treiber funktioniert nicht.
- Bildschirm schwarz, nachdem Kernel erfolgreich angesprungen
- --> Serielle Konsole, VT-Kompatibles Terminal
- Erfolg: Debian Install Disk Menü meldet sich!

Die Praxis: Treiber / Module

- Debian-Installer meldet: Keine Platte gefunden
- Ursache: die SCSI-Controller werden nicht erkannt.
- Problem: initrd-Image ist vom 32-Bit Debian
- Modutils und Module gehen nicht
- --> neu compilieren, alle "verdächtigen" Treiber statisch in den Kern
- Erfolg: Debian Installation "fast" wie auf normalem PC möglich

Die Praxis: 64Bit-Kern unterjubeln

- initrd-Image will PPC-Mac-Kern installieren
- Das würde nicht laufen :-)
- Zusätzlich schlägt wohl http-Install-Methode fehl
- Lösung:
 - eigenes Kernel-Image auf Webserver
 - Zielpartition mounten
 - Kernel und basedebs.tar per wget holen

- Installation damit fortsetzen
- Erfolg: Installation funktioniert

Die Praxis: Kaputter Boot-Loader

- Die YaBoot-Version der initrd ist buggy
- Symptom: neu installiertes Debian startet nicht
- Lösung:
 - Boot mit Kern übers Netz und "root=/dev/sda3 noinitrd"
 - Neue YaBoot-Version installieren
 - YaBoot werkeln lassen
- Erfolg:
 - Rechner läuft
 - Schabi hat Kopfweh
 - VB schießt Partitionstabelle mit fdisk wieder ab.

Was noch zu tun wäre:

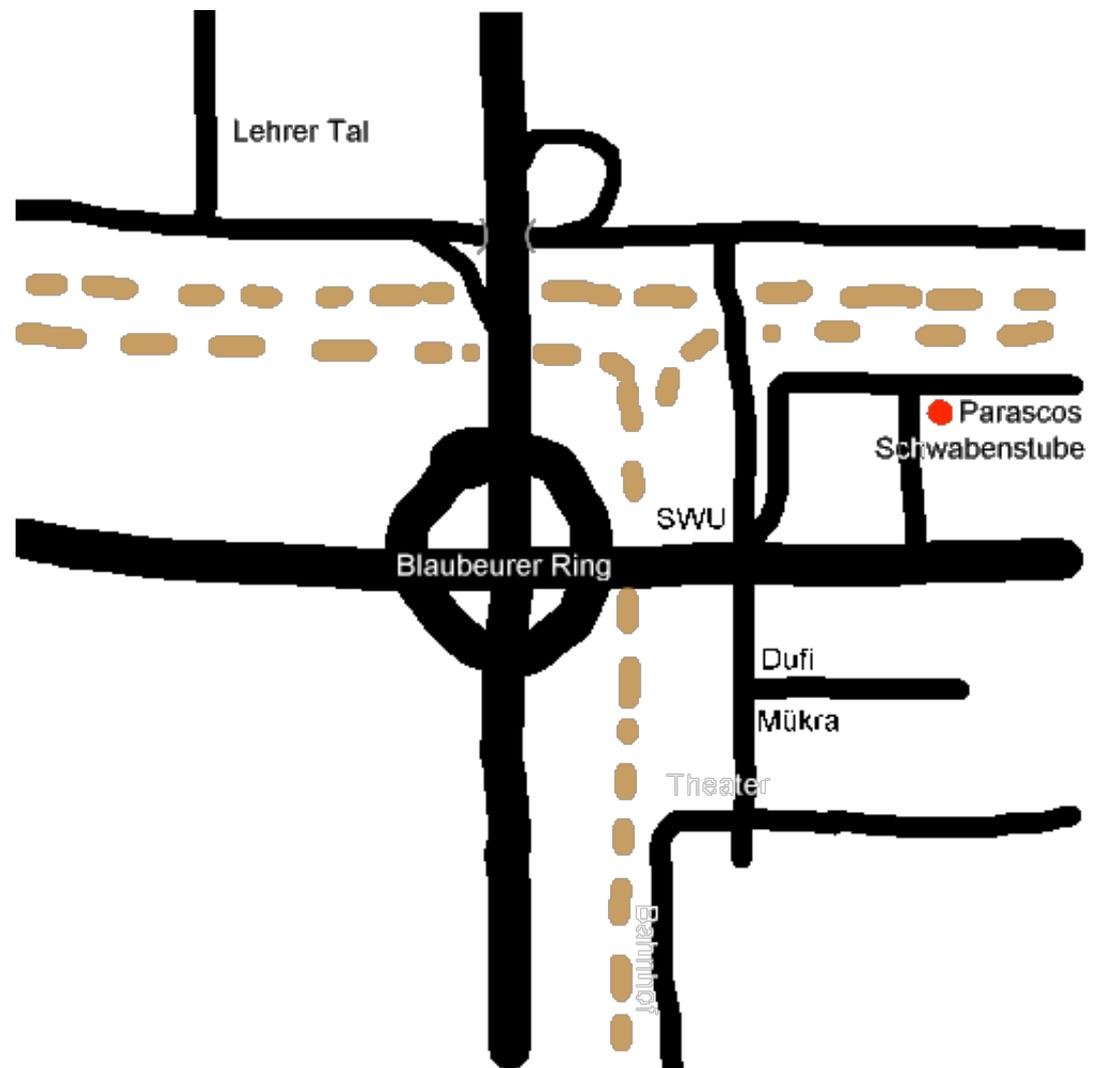
- Sauberes Build-System bauen
- 64Bit-glibc übersetzen
- modutils anpassen
- Debian Basis-System auf 64Bit ziehen
- 64Bit-Install-Image basteln
- CD/Floppy Install basteln
- Debian-Flavour gründen
- Compile/Develop-Maschine ans Debian Projekt sponsorn

Mein persönliches Fazit:

- Viel Chaos
- Aber: Alles "halb so schlimm"
- Viel gelernt, viel Spaß
- Sponsoren gesucht!

- Vielleicht sollten wir uns bei IBM bewerben?
- Jederzeit wieder

Ende



- <http://www.ulm.ccc.de/>
- mail@ulm.ccc.de
- zweiter Montag: 20:00 Uhr Vortrag hier
- sonst: ab ca 19:30, Cafe Einstein (Uni)
- Verwendete Software:
 - Opera
 - Debian GNU/Linux
 - nedit
- Ab zu Parasco?