



OSL UNIX Pfadfinder

Solaris auf dem PC

OSL
Gesellschaft für offene
Systemlösungen mbH

OSL – The Power of Simplicity
Informationen zu OSL Softwareprodukten (Storage–Virtualisierung, Volume–Manager, IO–Multipathing, Clustering, Anwendungs–Virtualisierung) unter <http://www.osl-it.de>

OSL **UNIX Pfadfinder**

Solaris auf dem PC

V1.01



OSL – The Power of Simplicity
Informationen zu OSL Softwareprodukten (Storage–Virtualisierung, Volume–Manager, IO–Multipathing, Clustering, Anwendungs–Virtualisierung) unter <http://www.osl-it.de>

Copyright und Handelsmarken

Copyright © OSL Gesellschaft für offene Systemlösungen mbH 2002, 2003, 2004.

Alle Rechte vorbehalten.

Eine unveränderte Nutzung dieser Dokumentation ausschließlich für private oder interne Zwecke ist gestattet. Andere Nutzungsarten, gleich welcher Form, wie z. B. die Bearbeitung, Übersetzung oder Veröffentlichung dieses Dokumentes bedürfen einer ausdrücklichen vorherigen schriftlichen Genehmigung durch OSL.

Alle verwendete Hard- und Softwarenamen sind Handelsnamen und/oder Warenzeichen oder eingetragene Warenzeichen der jeweiligen Hersteller oder Inhaber.

Beschränkungen

OSL stellt diese Dokumentation für die vorstehend beschriebene interne oder private Nutzung unentgeltlich und »wie sie ist« (»as is«) bereit. Eine Garantie auf diese Dokumentation bzw. auf die durch sie beschriebene Software, auf Code–Beispiele und beschriebene Verfahren, auf eine handelsübliche Qualität oder die Eignung für einen bestimmten Zweck ist ausgeschlossen. OSL übernimmt insbesondere keine Haftung für enthaltene Fehler, unmittelbare oder mittelbare Schäden oder Schadenersatz für Aufwendungen, die durch Auslieferung, Bereitstellung, Benutzung oder Nichtbenutzung dieses Dokumentes entstehen.

Der Erhalt dieses Dokumentes begründet keine weiteren Rechte. Alle in diesem Material enthaltenen Informationen stehen unter dem Vorbehalt einer Änderung ohne vorherige Ankündigung. Weder die beschriebene Software noch die vorliegende Dokumentation stellen Programmierschnittstellen (API's) oder Teile davon dar.

Diese Dokumentation selbst, die darin beschriebene Software und referenzierte Dokumentationen sind intellektuelles Eigentum der jeweiligen Hersteller oder Inhaber der betreffenden Rechte, das u. a. durch das Urheber-, Handels-, und Markenrecht geschützt ist. Die Benutzung, Installation, Kopie, Weitergabe oder Veräußerung solcher Software und Dokumentationen unterliegt den jeweiligen Lizenzbestimmungen.

In dieser Dokumentation enthaltene Informationen zu Produkten und Dienstleistungen Dritter sind entsprechenden Dokumentationen oder sonstigen Publikationen der jeweiligen Hersteller, sekundären oder sonstigen öffentlich zugänglichen Quellen entnommen. OSL hat diese Produkte und Dienstleistungen, Ihre Leistungsparameter und Interoperabilität – auch in Bezug auf OSL Produkte – nicht getestet und schließt folgerichtig jede Garantie oder Haftung hinsichtlich der Produkte, Dienstleistungen und Informationen Dritter aus.

Die in dieser Dokumentation enthaltenen Beispiele werden je nach Softwareständen, Hardware und sonstiger Umgebung von Ihrem System abweichen. Für die Bewertung der Korrektheit der vorliegenden Informationen, für die Auswahl und die Beurteilung der Eignung beschriebener Verfahren sowie dargestellter Hard- und Softwarekonfigurationen für einen bestimmten Zweck, für deren Anwendung oder Nichtanwendung sowie die Tauglichkeit etwaig ausgewählter Kombinationen von Hard- und Softwarekomponenten im Gesamtsystem ist allein der Anwender verantwortlich. Dies gilt auch für eine nachfolgende Installation und Konfiguration von Software, für die Nachnutzung der beschriebenen Verfahren sowie für die im Rahmen der Nutzung angestrebten Ergebnisse.

Versionen dieses Dokumentes

Version	Datum	Autor	e-mail	Inhalt / Änderungen
1.00	02.02.2004	MB	pathfinder@osl-it.de	Erste Fassung
1.01	22.06.2004	HO	pathfinder@osl-it.de	OSL Hinweis, Copyright 2004



OSL – The Power of Simplicity
Informationen zu OSL Softwareprodukten (Storage–Virtualisierung, Volume–Manager, IO–Multipathing, Clustering, Anwendungs–Virtualisierung) unter <http://www.osl-it.de>

Inhaltsverzeichnis

1. Einleitung	4
2. Partitionierung	4
2.1. Allgemeines	4
2.2. Besonderheiten mit Solaris x86	5
2.2.1. Genau eine Solaris–Partition	5
2.2.2. Wo ist der OBP?	5
2.2.3. Solaris und andere Betriebssysteme	5
3. Vorbereitungen und geeignete Hardware	5
3.1. Durchführung der Installation	7
3.1.1. Mögliches Problem mit PCI–IDE Bridges mit Funktion ungleich 1	7
3.2. Integration »nicht unterstützter« LAN–Karten	10
4. Ändern der Boot–Argumente	12
5. Solaris und Linux	13
5.1. Solaris in GRUB integrieren	14



OSL – The Power of Simplicity
Informationen zu OSL Softwareprodukten (Storage–Virtualisierung, Volume–Manager, IO–Multipathing, Clustering, Anwendungs–Virtualisierung) unter <http://www.osl-it.de>

1. Einleitung

Die Installation von Solaris auf Intel–Systemen ist nicht immer einfach. Es gibt eine Vielzahl möglicher Hardwarekombinationen, nach einer Dokumentation erfolgversprechender Vorgehensweisen sucht man oft vergeblich. Nachfolgend eine Zusammenstellung von einigen praktischen Erfahrungen.

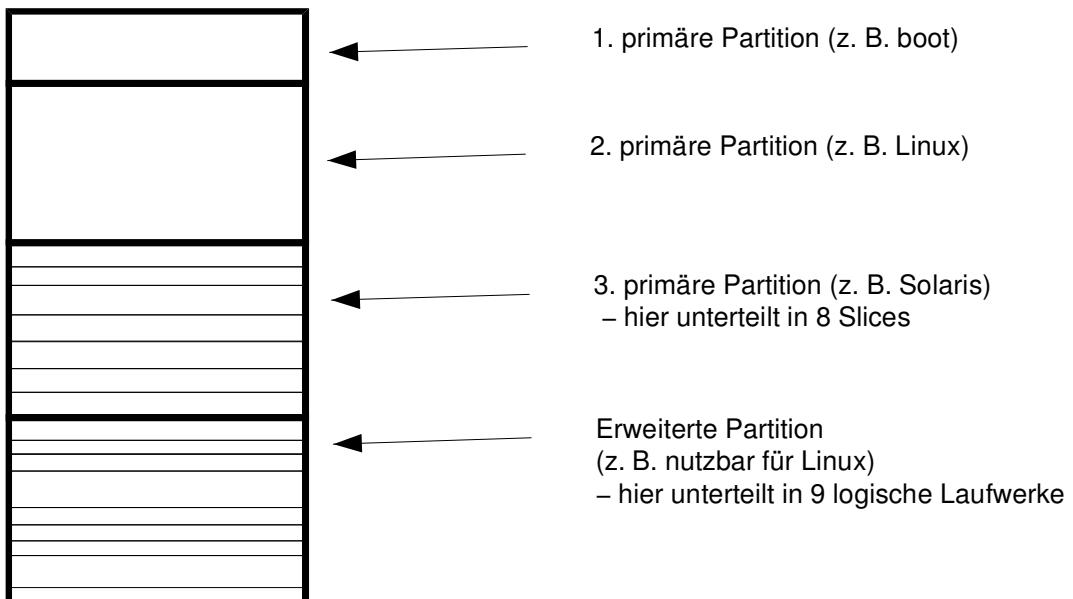
Bei Problemen kann die Suche nach deren Ursachen und Lösungsmöglichkeiten extrem zeitraubend sein, Entwickler–Support durch Sun oder Zugriff auf Hintergrundinformationen bestand nicht. Daher ist es an einigen Stellen nicht möglich, mehr als lediglich praktische Erfahrungen und Erklärungsversuche anzubieten. Fragen und eher vage Vermutungen sind im Text mit einem »?« gekennzeichnet.

Ergänzungen, weitere Erfahrungen und Hinweise sind jederzeit willkommen.

2. Partitionierung

2.1. Allgemeines

Auch auf Intel–Systemen nutzt Solaris Slices zur Einteilung der Platten. Anders als auf Sparc ist es aber möglich, die Platte für mehrere Systeme zu nutzen und den Zugriff für Solaris auf einen Teil der Platte zu beschränken, nämlich auf eine Solaris–Partition. Auf einem PC–System kann man bis zu 4 solcher primären (bootfähigen) Partitionen erstellen. Jede davon kann ein anderes Betriebssystem oder auch Software von Microsoft enthalten. Anstelle einer primären (bootfähigen) Partition kann man auch eine erweiterte Partition erstellen (allerdings nur genau eine) und diese in logische Laufwerke unterteilen (bei IDE–Platten bis zu 59). Solch eine erweiterte Partition ist von Solaris allerdings nicht nutzbar.



Pro Festplatte sind beim PC 4 primäre (bootfähige) Partitionen möglich. Alternativ kann man 1 primäre Partition durch eine erweiterte Partition ersetzen, die in eine größere Zahl logischer Laufwerke unterteilt werden kann. Nutzbar sind diese z.B. für Linux, nicht jedoch für Solaris.



OSL – The Power of Simplicity
Informationen zu OSL Softwareprodukten (Storage–Virtualisierung, Volume–Manager, IO–Multipathing, Clustering, Anwendungs–Virtualisierung) unter <http://www.osl-it.de>

2.2. Besonderheiten mit Solaris x86

2.2.1. Genau eine Solaris–Partition

Solaris kann nur in eine primäre Partition installiert werden, diese kann in maximal 8 Slices unterteilt werden. **Auf einer Platte darf es nur eine (primäre) Solaris–Partition geben** (mit allen anderen Konstellationen bewegt man sich außerhalb der Spezifikation, Trickserei lohnt nicht, es gibt an zu vielen Stellen Probleme).

2.2.2. Wo ist der OBP?

Sparc–Systeme verfügen über einen Open Boot Prom (OBP), in dem wichtige Einstellungen, z.B. zum Boot–Verhalten des Systems gespeichert werden können. PC's kennen eine solche Einrichtung nicht. Solaris für Intel behilft sich damit, daß diese Einstellungen auf der Platte gespeichert werden. Dies erfolgt im Verzeichnis /boot, der Zugriff auf die Einstellungen kann – wie auch auf Sparc–Systemen – später vom OS aus über den Befehl eeprom(1M) erfolgen. Legt man bei der Installation nur eine primäre Partition an, so werden die Daten im Root–Filesystem untergebracht. Das gibt spätestens dann Probleme, wenn man mehrere OS–Instanzen (z. B. Solaris 8 auf Slice 0, Solaris 9 auf Slice 3) installieren und wahlweise booten möchte. Hier ist es fast sicher, daß früher oder später Daten (z. B. unter /boot) zerstört werden.

Bei intensiverer Nutzung von Solaris einschl. eeprom–Funktionen, unbedingt aber bei Installation mehrerer Solaris–Instanzen sollte unbedingt eine x86–Boot–Partition (primäre Partition) während der Installation angelegt werden (Größe ca. 12 Mbyte). Diese Partition enthält dann alle Daten aus /boot in einem eigenen, kleinen Filesystem, das von jeder OS–Instanz montiert wird und bereits vor dem Laden des Kernels und dem Montieren des Root–Filesystems in der Bootshell zugreifbar ist.

2.2.3. Solaris und andere Betriebssysteme

Prinzipiell sollte es möglich sein, mehrere Betriebssysteme in jeweils einer eigenen Partition zu installieren. Der Teufel liegt jedoch im Detail. Oft wird bei der Installation der Master Boot Record (MBR) überschrieben, bei einer Installation von Windows sogar so, daß nur noch dieses geladen werden kann. Dies ist nur mit Ausprobieren der richtigen Installationsreihenfolge lösbar (in diesem Fall Windows vor Solaris). Unbedingt ratsam ist auch ein Backup der Daten **und** der Partitionstabellen nebst MBR vor jedem Installationsversuch.

Mindestens für Linux gilt, daß mehrere Solaris–Programme darauf hinweisen, daß eine gleichzeitige Installation von Linux und Solaris auf ein und derselben Festplatte nicht unterstützt wird. Es geht trotzdem, natürlich auf eigene Gefahr. Probleme liegen nach unserer Auffassung in einer womöglich unterschiedlichen Abbildung der Plattengeometrien in beiden Betriebssystemen (Probleme an Partitionsgrenzen) und darin, daß Linux Swap und Solaris gleiche Partition ID's benutzen (0x82). Installation von Solaris

3. Vorbereitungen und geeignete Hardware

Auf jedem einigermaßen aktuellen PC mit gängigen Komponenten ist ein Installationsversuch aussichtsreich (oft geht mehr, als man denkt). Neben den allgemeinen Mindestanforderungen (CPU, MEM, Plattenplatz) sollte man folgende Komponenten genauer prüfen:

- LAN–Karten
- SCSI–Controller und –Platten
- Multimedia–Devices
- USB etc.
- IDE–Controller
- Grafikkarten
- PCMCIA

Dafür gibt es auf den SUN–Seiten HCL's (Host Compatibility Lists) und Kurzberichte, von Anwendern, die Solaris bereits auf bestimmter Hardware erfolgreich installiert haben (Rubrik BigAdmin).

OSL – The Power of Simplicity

Bei **LAN-Karten** muß mdst. der eingebaute Chip von vorhandenen Treibern unterstützt werden, Unterverisionen kann man u. U. integrieren. Exzellente Unterstützung im Betriebssystem gibt es für die Intel Pro Boards. Mindestens bei Notebooks sollte man – sofern man Solaris einsetzen möchte – sich schon vor der Anschaffung darüber im Klaren sein, ob das Onboard-LAN unterstützt wird. Man kann hier meist nicht ohne weiteres nachrüsten. Sofern nicht exakt das verwendete Modell in der HCL steht, muß man also wissen, was für ein Chip benutzt wird, die klangvollen Bezeichnungen in Prospekten reichen dafür meist nicht. Bei Desktop-PC und Servern ist es nicht ganz so kritisch, hier kann man meist problemlos eine unterstützte PCI-Karte nachrüsten (z. B. eine 100Mbps-Karte mit Realtek-Chip wie z. B. von SITECOM, die es für 5–10 EUR beim Discounter gibt; Treiber von Realtek herunterladbar).

SCSI– oder gar **FC-Controller** sind sicher ein Thema für sich. Dazu liegen bei OSL bislang keine Erfahrungen vor.

Auch **Grafikkarten** waren in der Vergangenheit fast immer ein Problem. Hier kommen neue Modelle fast im Monatstakt auf den Markt. Sofern man einen aktuellen PC hat, ist es extrem unwahrscheinlich, daß die eingebaute Grafikkarte in der vorliegenden Solaris-Version bekannt ist. Auswege sind:

- Ausprobieren, ob die ähnliche Treiber funktionieren (da bei Fehlkonfiguration das System meist neu von CD gestartet werden muß, extrem zeitraubend),
 - Einsatz von XFree386 (unter Solaris erst in letzter Zeit für den Nicht–Guru beherrschbar),
 - Nutzung eines generischen VESA–Treibers, was i. d. R. gut brauchbare Ergebnisse liefert und für kompatible Karten funktioniert (lief bei uns auf allen getesteten PC's). Leider ist dieser Treiber erst ab Solaris 9 enthalten (bei Solaris 8 wohl aber als Patch verfügbar),
 - Nutzung eines normalen VGA–Modus (640x480), was aber wohl nur dann akzeptabel ist, wenn die Maschine ausschließlich über das Netz genutzt wird (Server).

Bei **IDE-Controllern** und Platten ist es heute in der Regel üblich, daß auf dem Mainboard ein »Primary« und ein »Secondary« Controller integriert sind (außer Notebooks ?). Mit dieser Konfiguration hatten wir auf **allen** getesteten PC's Probleme, für die auch kein Workaround gefunden werden konnte. **Sowohl Installation als auch Betrieb waren nur möglich, wenn ausschließlich der »Primary« Controller in Betrieb war.** Übliche Konfiguration der PC's »von der Stange« ist heute:

Es hilft also nach unserer Erfahrung nur ein Umbau, empfohlene Konfiguration wie folgt:

Primary IDE	Master: Slave:	Festplatte CD-ROM oder DVD
Secondary IDE		unbenutzt

Das CD-ROM (DVD) ist prinzipiell ja nur in der Installationsphase erforderlich. Wenn man also zwei Festplatten wünscht, ersetzt man CD-ROM/DVD einfach nach Installation von Solaris (nebst vollständiger Netzwerk-Konfiguration) einfach durch die Platte. Zumindest für den professionellen Nutzer findet sich sicher eine zweite Maschine, deren CD-ROM man später bei Bedarf über NFS montiert um etwa von CD weitere Software zu installieren. Bei CD-ROM-freiem PC ist natürlich eine sichere Boot-Umgebungen mit einem alternativen Root-FS dringend zu empfehlen, do sonst bei Problemen in jedem Falle wieder ein Umbau fällig ist.

Die Performance ist übrigens beeindruckend: Mit 2 Platten am »Primary« konnten wir parallel von Master und Slave mit jeweils 40 MByte/s lesen (in Summe als 80 Mbyte/s).

Gemessen mit einem 2,6 GHz Intel Low-End PC, SiS-Mainboard.

Alle anderen Komponenten sind für uns entbehrlich, wir haben darauf verzichtet, hier weitere Zeit zu investieren. Mdst. für **Sound** gibt es wohl aber privat entwickelte generische Treiber im Web. Nicht genutzte Hardware (Game-Port etc.) deaktivieren wir vorsichtshalber vor der Installation bereits im BIOS-Setup. Das scheint nicht unbedingt erforderlich zu sein, spart aber ggf. problematische Situationen.



OSL – The Power of Simplicity
Informationen zu OSL Softwareprodukten (Storage–Virtualisierung, Volume–Manager, IO–Multipathing, Clustering, Anwendungs–Virtualisierung) unter <http://www.osl-it.de>

3.1. Durchführung der Installation

Hier empfiehlt sich natürlich zunächst eine Lektüre der Dokumentation von Sun. Abgesehen von den »exotischen« Wegen wie Installation über Netz (was wenigstens beim PC eher etwas für den Profi ist) gibt es drei einfache Alternativen (Einlegen und Anfangen):

- Installation von **DVD**
wohl erst ab Solaris 9, schnell, spart Wechsel Medium, Web–Start (s. CD »Installation«)
- Installation von **CD »Installation«**
Web–Start mit netter Oberfläche, bei Problemen mit Hardware (Grafikkarte) besser anderen Weg gehen, legt u. U. (Solaris 8) immer (?) eine boot–Partition an.
- Installation von **CD »Software 1 of 2«**
unser bevorzugter Weg, relativ gute Möglichkeiten, auf Probleme zu reagieren

Sofern das BIOS beim Systemstart eine Übersicht über verwendete Interrupts, IO–Adressen, PCI–Devices und –Funktionen ausgibt, sollte man spätestens nach dem ersten erfolglosen Installationsversuch einige Notizen machen, diese Informationen können später ggf. Weiterhelfen. Also CD (DVD) einlegen und los...

Prinzipiell wird bei der Installation zunächst der Device Configuration Assistant (**DCA**) geladen, der zunächst vorhandene Hardware sucht. Man hat hier die Chance, über den Punkt »F4_Device_Tasks« bestimmte Geräte einzutragen, Konflikte zu beseitigen etc. Wirklich entscheidende Ergebnisse bringt das aber im allgemeinen nicht. Entweder der DCA erkennt die Geräte oder nicht, es sind keine »Wunder« von diesem Tool zu erwarten. Einzige Ausnahme: Beseitigung von Adreß–Konflikten im Memory durch Löschen einzelner Geräte. Dies steht dann aber bei jedem Boot an.

Nach dem Device–Scan und Ausgabe einer Geräteliste werden dann die möglichen Boot–Devices angeboten. Bei der Installation wählt man hier natürlich das CD–ROM (DVD), sonst in der Regel die Festplatte. Mit »F2_Continue« wird dann vom ausgewählten Medium das Root–FS montiert und Solaris gestartet.

3.1.1. Mögliche Problem mit PCI–IDE Bridges mit Funktion ungleich 1

Auf einigen PC's, speziell mit Motherboards von SiS und MEGABYTE scheint hier häufig die Installation bereits am Ende zu sein, wenn folgende Meldung erscheint:

```
/pci@0,0/pci-ide@2,1/ide@0/sd@1,0:a: can't open - invalid boot interface
Mountroot failed.
```

Press Enter to Continue.

Folgende Interpretationshilfe zu Device–Pfaden wie dem obigen:

pci-ide@2,1	PCI–IDE Bridge als PCI–Controller Nr. 2, Funktion 1.
pci-ide@2,5	PCI–IDE Bridge als PCI–Controller Nr. 2, Funktion 5.
ide@0	Primary IDE Controller (1. Kabel).
ide@1	Secondary IDE Controller (2. Kabel).
sd@0	Über sd–Treiber angesprochenes 1. Gerät am Bus (Master). CD–ROM wird über sd–Treiber angesprochen!
sd@1	Über sd–Treiber angesprochenes 2. Gerät am Bus (Slave). CD–ROM wird über sd–Treiber angesprochen!
cmdk@0	Über cmdk–Treiber angesprochenes 1. Gerät am Bus (Master). IDE–Platten werden über den cmdk–Treiber angesprochen.
cmdk@1,0:a	2. Festplatte am Bus (Slave), 1. Partition (0), Slice 0 (a).

Prüft man die tatsächliche Konfiguration nach, stellt man fest, daß das angegebene Gerät überhaupt nicht existiert. Entweder der DCA oder die geladenen Treiber gehen immer von Geräte–Funktion 1 aus und liefern so u. U. ein falsches Gerät, so daß der Root–Mount fehlschlägt. Dieses Problem existiert unter Solaris 7 bis 9 (zwischenzeitlich soll es wohl aber einen Patch geben ?). Im obigen Beispiel hatte die PCI–IDE–Bridge in Wirklichkeit Funktion 5, wie man aus dem BIOS und sogar aus der Device–List des DCA entnehmen konnte.



OSL – The Power of Simplicity
Informationen zu OSL Softwareprodukten (Storage–Virtualisierung, Volume–Manager, IO–Multipathing, Clustering, Anwendungs–Virtualisierung) unter <http://www.osl-it.de>

Problem–Umgehung

Glücklicherweise werden die Geräteknoten dennoch korrekt angelegt, lediglich der Mount–Aufruf wird mit einem falschen Argument gestartet. Folgende Umgehung wird damit möglich:

1. Nach Erscheinen der obigen Fehlermeldung nicht »Enter« sondern stattdessen »Ctrl–C« drücken, um in die Boot–Shell zu gelangen. In der Boot–Shell hat man nur eine englische Tastaturbelegung. Hilfe gibt es über Eingabe von »help«.
Hier kann man sich wie folgt die gefundenen Geräte anzeigen lassen:

```
> show-devs
+boot/memory@0,0
...
pci@0,0/pci-ide@2,5/ide@0
pci@0,0/pci-ide@2,5/ide@1
...
```

Diese Ausgabe ist gekürzt, entscheidend ist die Zeile »pci@0,0/pci-ide@2,5/ide@0« aus der wir erfahren, daß unser tatsächlich vorhandenes Gerät 2, Funktion 5 wirklich erkannt wurde.

2. Wir montieren jetzt das korrekte Root–Device (das CD–ROM als Slave am »Primary«), nachdem wir es zunächst (flüchtig) in das Environment aufgenommen haben.

```
> printenv
...
bootpath='/pci@0,0/pci-ide@2,1/ide@0/sd@1,0:a'
> setprop bootpath /pci@0,0/pci-ide@2,5/ide@0/sd@1,0:a
> getprop bootpath
/pci@0,0/pci-ide@2,5/ide@0/sd@1,0:a
mount $bootpath /
```

Wenn man dies falsch macht, erscheint wieder die bereits aus dem DCA bekannte Fehlermeldung. Sobald der Mount erfolgreich war, mit »Ctrl–D« die Boot–Shell wieder verlassen, der DCA wird dann allein den Kernel laden (der fortgeschrittene User kann natürlich auch gleich aus der Boot–Shell ggf. etwas anderes starten, in der Regel ist der automatische Start durch den DCA jedoch einfacher).

Hinweis zum Listing: printenv zeigt immer die (an dieser Stelle unveränderlichen) eeprom–Settings an. Entscheidend ist jedoch der tatsächliche Wert eines Properties, den man in der Boot–Shell mittels »setprop« verändern kann. Diese Änderung bleibt nur für diesen Bootvorgang erhalten (flüchtig) und kann nur mittels »getprop« angezeigt werden.

Hinweis zur Tastaturbelegung: Da nur die englische Belegung funktioniert folgende Hilfe für die deutsche Tastatur:

@	Shift–2	–	ß
/	–	:	Shift–Ö

3. Der Kernel startet jetzt, wird die Devices konfigurieren und man kann normal mit der Installation fortfahren.

Nach dem Ende der Installation ist natürlich das Problem mit dem Bug der Boot–Umgebung nicht beseitigt, d. h. man wird bei jedem Boot immer wieder im DCA landen. Um dies nicht übermäßig aufwendig zu gestalten, gehen wir bis zur Verfügbarkeit eines Bugfixes folgenden Weg:

1. Bereits **beim ersten Reboot nach der Systeminstallation**, der von Platte erfolgt, wird man analog zum Booten bei der Installation selbst das Root–FS montieren müssen, jetzt aber unter Nutzung des cmdk–Treibers. Wenn die Platte wie empfohlen als Master am »Primary« hängt also wie folgt:

```
> printenv
...
bootpath='/pci@0,0/pci-ide@2,5/ide@0/cmdk@0,0:a'
> setprop bootpath /pci@0,0/pci-ide@2,5/ide@0/cmdk@0,0:a
> getprop bootpath
/pci@0,0/pci-ide@2,5/ide@0/cmdk@0,0:a
```



OSL – The Power of Simplicity
Informationen zu OSL Softwareprodukten (Storage–Virtualisierung, Volume–Manager, IO–Multipathing, Clustering, Anwendungs–Virtualisierung) unter <http://www.osl-it.de>

```
> mount $bootpath /
```

2. Da der unter 1. beschriebene Schritt nun bei jedem Boot notwendig wäre verfahren wir wie folgt: Wir booten das System ganz normal bis zum Multiuser–Mode. In einem Terminal setzen wir jetzt den korrekten Wert für »bootpath«. Im nachfolgenden Beispiel haben wir eine zweite Solaris–Instanz auf Slice 3 (d) installiert. Um diese ohne die Eingabe des gesamten kryptischen Devicepfades einfach laden zu können, ist dafür auch ein entsprechender Alias definiert worden. Siehe dazu auch eeprom(1M).

```
# eeprom bootpath="/pci@0,0/pci-ide@2,5/ide@0/cmdk@0,0:a"
# eeprom altroot="/pci@0,0/pci-ide@2,5/ide@0/cmdk@0,0:d"
# eeprom
auto-boot?=true
auto-boot-cfg-num=-1
auto-boot-timeout=5
boottimeout=0
bshfirst=false
output-device=screen
input-device=keyboard
boot-file=kernel/unix
target-driver-for-scsi=sd
target-driver-for-direct=cmdk
target-driver-for-csa=cmdk
target-driver-for-dsa=cmdk
target-driver-for-smartii=cmdk
pcide=true
prealloc-chunk-size=0x2000
ata-dma-enabled=1
kbd-type=German
kbd-wkeys=true
probed-arch-name=i86pc
probed-compatible=i86pc
bootpath=/pci@0,0/pci-ide@2,5/ide@0/cmdk@0,0:a
altroot=/pci@0,0/pci-ide@2,5/ide@0/cmdk@0,0:c
```

Damit ist der Wert für bootpath beim Systemstart korrekt versorgt. Damit wir nun nicht nach dem fehlgeschlagenen Mount des DCA jedesmal den Mountbefehl manuell absetzen müssen, kann man in /boot/solaris/boot.rc den Mountbefehl einfügen (einzufügende Zeile im Listing fett hervorgehoben)

```
#ident  "@(#)boot.rc    1.14    00/07/17 SMI"
#
# Copyright (c) 1998, by Sun Microsystems, Inc.
# All rights reserved.
...
#
# Confirm success or failure of configuration assistant and behave
# appropriately.
#
cd /options
if .streq ( ${root-is-mounted}X , falseX )
    echo
    echo 'The root filesystem is not mounted and the configuration'
    echo 'assistant has exited prematurely. Booting is unlikely to'
    echo 'succeed. CTL-ALT-DEL may be used to reset the machine.'
    echo
    echo 'Failover to boot interpreter - type ctrl-d to resume boot'
    console
    mount $bootpath /
endif
...
```

Damit wird jetzt beim Booten der DCA interaktiv starten. Man fährt zweimal mit F2 fort bis zum Boot–Versuch, der natürlich fehlschlägt. Jetzt drückt man »Ctrl–C«, gelangt in die Boot–Shell und beendet diese sofort wieder mit »Ctrl–D«. Der Boot läuft jetzt nach dem Mount von »/« normal weiter und die ganze Aktion braucht lediglich 5 Sekunden zusätzlich. Leider haben wir keinen Weg für einen vollautomatischen Boot bei Vorliegen dieses Fehlers gefunden.



OSL – The Power of Simplicity
Informationen zu OSL Softwareprodukten (Storage–Virtualisierung, Volume–Manager, IO–Multipathing, Clustering, Anwendungs–Virtualisierung) unter <http://www.osl-it.de>

3.2. Integration »nicht unterstützter« LAN–Karten

Wenn eine LAN–Karte tatsächlich nicht unterstützt wird, d. h. für den verwendeten Chip oder kompatible Chips kein Treiber existiert geht natürlich nichts. Häufig werden aber lediglich neuere Modelle, die sich anders identifizieren, nicht erkannt, obwohl für einen kompatiblen Chip bereits ein Treiber existiert.

Das folgende Beispiel zeigt das Vorgehen mit einem Onboard LAN–Controller in einem Notebook.

Während der Installation sollte man bereits alle Netzwerkeinstellung ganz normal vornehmen, auch wenn die Installations–Software den Controller nicht anbietet und konfiguriert. So werden jedenfalls die wichtigsten Netzwerkeinstellungen bereits vorgenommen und einige Dateien mit sinnvollen Werten gefüllt. Nach Abschluß der Installation und Reboot wird man natürlich feststellen, daß keine Netzwerkkarte konfiguriert ist. Weiß man aber, daß man einen Chip eingebaut hat, der mit einem vorhandenen Treiber läuft (hier iprb, nachlesen muß man freilich beim Hersteller der Karte, in diesem Fall bei Intel), sucht man jetzt die LAN–Karte im Device–Tree:

```
# prtconf -pv | less
      Node 0x1a14a0
      assigned-addresses: 82014010.00000000.fcfff000.00000000.0000100
0.81014014.00000000.0000ecc0.00000000.00000040
      class-code: 00020000
      compatible: 'pci1028,2002' + 'pci1028,2002' + 'pciclass,020000'
device-id: 0000103d
      devsel-speed: 00000001
      fast-back-to-back:
      interrupts: 00000001
      max-latency: 00000038
      min-grant: 00000008
      model: 'PCI: 1028,2002 - class: Ethernet controller'
      name: 'pci1028,2002'
      power-consumption: 00000001.00000001
      reg: 00014000.00000000.00000000.00000000.00000000.02014010.0000
0000.00000000.00000000.00001000.01014014.00000000.00000000.00000000.00000040
      revision-id: 00000081
      slot: 00000000
      subsystem-id: 00002002
      subsystem-vendor-id: 00001028
      unit-address: '8'
vendor-id: 00008086
```

Wenn der Ethernet–Controller gefunden ist, interessieren insbesondere die »device–id« und die »vendor–id« (beides im Listing neben der Benutzereingabe zwecks Übersichtlichkeit fett hervorgehoben).

Sofern dieses Pärchen nicht im System bekannt ist, wird der Controller nicht konfiguriert, da unbekannt ist, welcher Treiber zu nutzen ist. Wenn wir es wissen, teilen wir das dem System durch einen Eintrag in der Datei /etc/driver_aliases mit (der neue Eintrag ist im Listing fett hervorgehoben):

```
# cat /etc/driver_aliases
...
iprb "pci8086,1029"
iprb "pci8086,1030"
iprb "pci8086,1229"
iprb "pci8086,2449"
iprb "pci8086,103d"
...
```



Vermutlich müssen wir jetzt noch für korrekte Konfiguration der Netzwerkschnittstelle sorgen. Folgendes Beispiel zeigt die Konfigurationsdateien für einen Host »birne« mit der Netzadresse 192.168.5.5 mit Default–Netzmaske (Class C):

```
# cat /etc/hostname.iprb0
birne

# cat /etc/hosts
#
# Internet host table
#
127.0.0.1      localhost          loghost
192.168.5.5    birne.garten.welt  birne

# cat /etc/netmasks
#
# The netmasks file associates Internet Protocol (IP) address
# masks with IP network numbers.
#
#      network-number  netmask
#
# The term network-number refers to a number obtained from the Internet Network
# Information Center.
#
# Both the network-number and the netmasks are specified in
# "decimal dot" notation, e.g:
#
#          128.32.0.0 255.255.255.0
#
192.168.5.0    255.255.255.0
```

Wenn jetzt alles fertig konfiguriert ist, dann sollte ein Reconfigure–Reboot gefahren werden und die Netzwerkkarte sollte konfiguriert sein:

```
# touch /reconfigure
# reboot
```



OSL – The Power of Simplicity
Informationen zu OSL Softwareprodukten (Storage–Virtualisierung, Volume–Manager, IO–Multipathing, Clustering, Anwendungs–Virtualisierung) unter <http://www.osl-it.de>

4. Ändern der Boot–Argumente

Hier hat man zwei Möglichkeiten, entweder temporär beim Booten des Systems oder dauerhaft via eeprom(1M). Setzt man im eeprom das Flag »bshfirst=true«, gelangt man noch vor dem DCA in die Boot–Shell. Der Device–Tree wird aber erst unmittelbar vor dem Root–Mount erzeugt, so daß man wahrscheinlich nach einer Änderung z. B. des bootpath–Properties zunächst den DCA starten möchte, in den man wieder mit »Ctrl–D« gelangt. Ist der DCA gelaufen, der Device–Tree angelegt und Root montiert, kann man lediglich noch Boot–Argumente ändern. Will man aus einem anderen Slice starten, so muß man zuvor im DCA das Boot–Device anpassen (F4_Boot_Tasks, Unterpunkt View/Edit Autoboot Settings). Das Boot–Device bezeichnet das Device, das als Root–Device montiert wird.

Hat man im eeprom bereits ein alternatives Root–Device definiert (vgl. Punkt 3.2.1), kann man auch »bshfirst=true« setzen und vor dem Start des DCA (s. o.) in der Boot–Shell einfach (temporär) das Boot–Device ändern mit »setprop bootpath \$altroot« (s. o., altroot muß natürlich zuvor bekannt sein), dann weiter mit »Ctrl–D« zum DCA. (der läßt sich auch mit »run /boot/solaris/bootconf.exe« starten, was aber leicht zum wiederholten Ladeversuch von Treibern führen kann, d. h. auch der Kernel sollte dann ebenfalls explizit mit run geladen werden).

Sofern dann das Root–Device bereits montiert ist, kann man dann noch Boot–Argumente ändern (z. B. interaktiv booten, um das Laden bestimmter Treiber zu verhindern oder /dev/null als alternative system–Datei anzugeben, wenn /etc/system falsche Werte enthält):

```
Type      b [file-name] [boot-flags] <ENTER>      to boot with options
or       i <ENTER>                         to enter boot interpreter
or       <ENTER>                         to boot with defaults

<<< timeout in 5 seconds >>>"

Select (b)ooot or (i)nterpreter: b kernel/unix -ras
```

Die wichtigsten Schalter sind:

- r reconfigure boot (entspricht Anlegen von /reconfigure mit nachfolgendem Reboot), Geräteknoten für neue Devices werden angelegt.
- a ask for some parameters, hier kann zum Beispiel /dev/null für /etc/system angegeben werden, um mit Default–Werten zu starten, oder man kann den Suchpfad für Kernel–Module ändern (z. B. um problematische Treiber am Laden zu hindern).
- s Boot to Single User Mode.



OSL – The Power of Simplicity
Informationen zu OSL Softwareprodukten (Storage–Virtualisierung, Volume–Manager, IO–Multipathing, Clustering, Anwendungs–Virtualisierung) unter <http://www.osl-it.de>

5. Solaris und Linux

Wie bereits im Abschnitt 2.2 »Solaris und andere Betriebssysteme« dargestellt, ist die Installation von Linux und Solaris auf ein und derselben Festplatte wohl nicht durch Sun vorgesehen (?). Neben der Tatsache, daß Linux Swap und Solaris gleiche Partition ID's benutzen (0x82), kann es vermutlich (?) auch Probleme dadurch geben, daß beide Systeme die Plattengeometrien unterschiedlich im OS abbilden könnten. Das kann Probleme an Partitionsgrenzen geben. Es ist daher ein guter Plan, zwischen der Solaris–Partition und einer durch Linux genutzten besser einige Zylinder Platz zu lassen (?). Erfolgreich durchgeführt (aber natürlich nicht wirklich abgesichert, d. h. man geht schon ein höheres Risiko ein) wurde folgendes Verfahren (mit Solaris 9 und SuSE Linux 8.2):

1. Man bootet zunächst von CD ein SuSE Rescue System. Dann legt man mit fdisk an (Bsp. Notebook mit 40 GB HD, C/H/S = 4864/255/63, Größen exemplarisch):

– Primäre Linux Boot–Partition	2 Cyl	mdst. 10 MB
– Primäre Linux Swap–Partition	z. B. 4000 Cyl.	für Solaris–Installation
– Erweiterte Partition	Rest	für Linux, logische Laufwerke nach Bedarf
2. System rebooten und mit Solaris–Installation beginnen, Solaris wird sich in die primäre Partition mit der ID 0x82 (Linux Swap) installieren.
3. Noch einmal SuSE Rescue System starten und mit fdisk die Partition ID der Solaris Partition zur Sicherheit auf etwas unverfängliches setzen.
4. Installation von Linux in die Boot–Partition (boot) und in die erweiterte Partition (das eigentliche Linux). Unbedingt ein Auge auf die Solaris–Partition haben!
5. Mittels SuSE Rescue System die Partition ID der Solaris–Partition wieder zurücksetzen auf 0x82.

Jetzt kann man beide Systeme starten. Für mehr Komfort empfiehlt sich die Nutzung von GRUB.



OSL – The Power of Simplicity
Informationen zu OSL Softwareprodukten (Storage–Virtualisierung, Volume–Manager, IO–Multipathing, Clustering, Anwendungs–Virtualisierung) unter <http://www.osl-it.de>

5.1. Solaris in GRUB integrieren

GRUB stellt sich in der Handhabung wesentlich einfacher und zugleich komfortabler dar als LILO, er kann insbesondere einige Dateisysteme richtig lesen (die map–Dateien entfallen).

Um den Start von Solaris über GRUB zu ermöglichen sind wir wie folgt verfahren (Beispiel):

1. GRUB muß natürlich bei der Installation mit installiert worden sein.
2. Die Datei /boot/grub/menu.lst modifizieren, z. B. gemäß nachfolgendem Beispiel:

```
color white/blue black/light-gray
default 0
gfxmenu (hd0,0)/message
timeout 8

title Solaris
rootnoverify (hd0,1)
chainloader +1

title linux
kernel (hd0,0)/vmlinuz root=/dev/hda6 vga=0x305 hdc=ide-scsi \
hdclun=0 splash=silent showopts
initrd (hd0,0)/initrd

title floppy
root (fd0)
chainloader +1

title failsafe
kernel (hd0,0)/vmlinuz.shipped root=/dev/hda6 showopts ide=nodma \
apm=off acpi=off vga=normal nosmp noapic maxcpus=0 3
initrd (hd0,0)/initrd.shipped
```