
Linux+Solaris HOWTO

Max Berger <max@berger.name>

Abstract

Describes how to use Linux (x86) and Solaris (x86) together on one machine

Table of Contents

Introduction.....	1
Reasons to use Solaris.....	1
Things covered in this HOWTO.....	2
Referenced versions.....	2
Warning.....	2
Preparing your hard drive.....	2
Traditional PC partitions.....	2
Solaris partition labels.....	3
Hard disk space.....	3
Quick check list.....	3
Preparing Linux.....	3
Building a new kernel.....	3
Preparing your boot loader.....	4
Installing Solaris.....	5
Solaris interactive.....	5
Web-Start.....	5
Getting GNU-tools.....	5
Boot managing.....	5
Using the Solaris boot manager.....	5
Using lilo.....	5
Using grub.....	6
Sharing data.....	6
Using a shared partition.....	6
Alternative ways.....	6
Running foreign binaries.....	7
Linux binaries on Solaris.....	7
Solaris binaries on Linux.....	7
A. Finding additional info.....	7
B. Credits and legal information.....	7
Credits.....	7
Questions, comments, suggestions.....	8
License.....	8

Introduction

Reasons to use Solaris

This is actually a good question. Why use Solaris, when you can have Linux? Solaris is a commercial system and the hardware support is scarce. Nevertheless, if you are reading this you have probably already decided on using Solaris. If not, here are some reasons:

- I had a part-time job administrating Solaris machines. It is of no use that I learning all the neat command parameters and nifty Linux tricks at home, when I cannot use them on my Solaris-Box.
- Sun was giving away Solaris for what they call "Shipping and handling cost only" for some time. They might do that again. Moreover, almost-free-of-charge is always a good deal.
- Another reason might be that you like CDE.

Things covered in this HOWTO

This HOWTO tries to help you install Solaris 7, 8, 9 or 10 on an x86 computer already running Linux. The main points are

- not to destroy any data, see the section called "Preparing your hard drive"
- to make Linux and Solaris coexist (boot manager), see the section called "Boot managing"
- to share data between Linux and Solaris, as described in the section called "Sharing data"

Referenced versions

The reference versions used while creating this document where:

- Solaris 7 x86 (= Solaris 2.7 / SunOS 5.7)
- Solaris 8 x86 (= Solaris 2.8 / SunOS 5.8)
- Linux 2.2.14, 2.3.99-pre3
- fdisk v2.10f

I have updated the information for Solaris 10 based on received feedback.

The information may apply to other versions of the software, or it may be different.

Warning

Please back up any data before messing with your system! Many of the procedures described in this document may result in excessive data loss! I am not responsible for any data loss! See also the section called "License"

Preparing your hard drive

To prepare your hard drive for Solaris, you need to know about standard PC partition tables and about Solaris disk slices. In this HOWTO, I talk about single-disk systems only, but all the information should also apply to a multi-disk environment.

Traditional PC partitions

The standard partition-table has only four entries. The entries important for us are the following:

A Primary Partition	Takes up one entry, and contains exactly one partition. A waste of resources, but you can only boot from this type of partition!
An extended Partition	Takes up one entry, but can contain multiple DOS, Linux, and other partitions

A Solaris Partition Takes up one entry, but can contain multiple Solaris Partitions

To find out what partitions are present on your system, use the `fdisk` program. Partitions numbers 1 to 4 (`hda1...4`, `sda1...4` ...) are the ones in your partition table.

Solaris partition labels

Solaris has its own partitioning scheme. It uses one entry in the partition table, and this entry is and acts as this partition would be the entire disk.

This virtual disk is divided in up to 8 slices. The third slice, `s2`, covers the complete virtual disk, so you actually have up to 7 slices for Solaris.

Unfortunately, the Solaris partition entry has the same type as a Linux Swap partition (82). Therefore, you should not have any Linux swap partitions as primary partitions. Linux does not care about this, but who knows what Solaris does?

The partition identifier has been changed in Solaris 10 to provide better compatibility with Linux. It is now `0xbf`. Solaris still supports the old identifier for compatibility reasons. When installing a new system you should use the new identifier.

Although the Linux `fdisk` program has some "Sun disklabel" support, this does not seem to help any.

Hard disk space

Of course, Solaris needs disk space. The minimum installation of Solaris 8 is about 300 MB. For the normal tools, it is about 700 MB, and for a "developer-system" about 1 GB.

However, this is only the space required for the base installation. You might want to add a lot of GNU-Tools, and other file. In addition, if you want to share data between Solaris and Linux, this has to happen on the Solaris partitions.

You might even think of sharing your home directories between Solaris and Linux. As the time of this writing: Forget it! I messed up my home directory doing so and I was *very* happy about my backup. See also the section called "Sharing data"

Quick check list

Here is the quick checklist. Make sure you:

- have used no more than 3 entries in your partition table
- have no Linux swap partitions as primary partitions
- Have at least 1 Linux ext2 partition as primary
- Have at least 1 GB unpartitioned space

Preparing Linux

Building a new kernel

Some people do not like building a custom kernel. For those of you: you *must* build your own kernel if you want to share data between Solaris and Linux on one hard disk. If, however, you do not want to share data, you are safe with your old kernel.

Please note that these instructions were for Linux 2.2 and 2.4 kernels. UFS support has probably improved in newer Linux versions.

Here are your kernel options for "Solaris compatibility":

- Under code maturity:

```
Prompt for development drivers,      Unfortunately, UFS writing is still experimental.
```
- In file systems:

```
UFS file system support      UFS is the Solaris file system.
```

```
UFS file system write support      The only way to pass data to Solaris.
```

```
partition types / Solaris (x86) partition table support      This is needed for the slices within your Solaris partition to be found.
```

Additional info about kernel building can be found in the Kernel-HOWTO.

Preparing your boot loader

Before messing with your boot loader, please make sure that you have an emergency disk ready that enables you to boot your old system.

Unfortunately, the Solaris install overwrites the master boot record (MBR) and overwrites your old boot manager.

Fortunately, the Solaris boot manager is able to chain-boot. For this to work you have to put your old boot loader (e.g. lilo) into the boot sector of one primary Linux partition.

For lilo, this means: look at your `/etc/lilo.conf`. Look for the `boot=` line. If it is something like `boot=/dev/hda1` everything is fine. However, if it is something like `boot=/dev/hda` it points to the MBR. Please change it to point to a primary Linux partition (e.g. `boot=/dev/hda1`)

A way to check if your boot-loader is chain-loadable is installing lilo in to the MBR and trying to chain load your other boot-manager. I used the following file called `lilo.conf.mbr` for this:

```
#lilo.conf.mbr
#Lilo in Master Boot Record doing nothing but chain-loading another lilo
boot=/dev/hda
root=/dev/hda5
install=/boot/boot.b
map=/boot/map
vga=ask
delay=50
other=/dev/hda1
label=lilochain
```

You can then run `lilo -C lilo.conf.mbr` to install lilo into your MBR. In addition, do not forget to run `lilo` to have lilo in the boot sector of your hard-drive.

If you reboot now, you should have a lilo-prompt. When you select `other` this chain loads the old lilo, which in turn loads Linux.

Installing Solaris

Solaris interactive

Solaris 7 Solaris Interactive is on the first CD. Boot from the CD and select Solaris interactive.

Solaris 8 On the Solaris 8 CDs the interactive installer is on the first software CD. You do not need the install CD at all.

The Solaris installation itself is quite straightforward. One thing that might be confusing is the partitioning. Solaris will automatically use all the free space on your hard disk, and create its own virtual disk in it. Moreover, when it comes to slicing up your virtual disk, do not forget to add space for Solaris/Linux exchange. A good place for this is an extra `/export/share` partition.

Please Note: If you abort during the Solaris installation, the MBR of your hard disk may already be overwritten and you may not be able to boot from the hard disk at all.

Web-Start

I could not get Web-Start to work at all. Please mail me if you have any experience using it.

However, once Solaris was installed, I had no problems adding web-start-packages

Getting GNU-tools

Solaris standard installation does not even have a `less` command. Some people recommend getting the Redhat Package Manager (RPM) for Solaris, but the standard packaging tool will usually do it. You can find many packages at <http://www.sunfreeware.com> The standard packages can be installed with `pkgadd <package>`, and the web-start packages with `java <packagewithoutclassextension>`

Boot managing

Using the Solaris boot manager

The Solaris boot manager gives you a choice to select any of the four primary partitions for booting. If you had your boot loader installed on the boot sector of a primary partition, everything should be fine. Boot from this partition, and Linux should come up.

Using lilo

Now you have lilo as a secondary boot loader, but you want it to be your primary boot loader again? This is no Problem. Lilo can easily chain boot Solaris. Just add the appropriate `other=` lines to your `lilo.conf`

For testing, you should add the Solaris chain-loading to the lilo in your boot sector first. Then try chain loading lilo, Solaris, lilo, Solaris ... until you have enough of it. If this works, you should be able to put lilo back into the MBR again.

However, Solaris might have messed up your partition table. This has happened on my system after installation of Solaris#8. Lilo does not like the new partition, nor does `fdisk`. I had no problems with Solaris#7

Using grub

According to feedback I have received about grub, booting Solaris with grub works fine.

```
title Solaris 9 - sdb
rootnoverify (hd1,0)
makeactive
chainloader +1
```

This will boot Solaris, in this case from hd1,0. You will have to adjust this for your system.

Sharing data

Using a shared partition

If you have build your kernel with Sun disk label support and UFS support as mentioned in the section called “Building a new kernel”, you are now able to mount your Solaris partitions. During boot up, you should get something similar to:

```
hda: [PTBL] [523/255/63] hda1 hda2 < hda5 hda6 hda7 hda8 > hda3 <Polaris: [s0]
hda9 [s1] hda10 [s2] hda11 [s3] hda12 [s6] hda13 [s7] hda14 >
```

Meaning (in this case): the third partition (hda3) is a Solaris partition with six slices (s0, s1, s2, s3, s6, s7). They are mapped to Linux devices hda9 to hda14.

Try mounting your Solaris partitions. When mounting UFS partitions, you always have to add an `-oufstype=` argument, in this case `-oufstype=sunx86`. Therefore, the command to mount a partition is something like:

```
mount -oufstype=sunx86 /dev/hda14 /mnt
```

Now, test your partition. *Please note: The write support on UFS partitions is very experimental. Please do not trust any data you write on your UFS partitions!*

If you want your Solaris partitions automatically mounted at boot time, you can add a line like

```
/dev/hda14 /solaris ufs ufstype=sunx86 0 0
```

to your `/etc/fstab`

Alternative ways

In addition, of course there are always other ways of sharing data:

Floppy disk Solaris supports UFS on floppies. That is, you can format a floppy and make an ufs file system on it with `mkfs`, as for any disk. Then, you can directly mount the floppy on the virtual file system. Such an ufs floppy might be mountable also on Linux, with a properly configured kernel.

A standard DOS floppy cannot be directly mounted: Solaris kernels up to SunOS 5.7 do not support FAT, and Solaris 8 might neither. Nevertheless, it can be accessed using the standard `mtools`, like under Linux. `mtools` do not come in bundle with Solaris, but they can be downloaded from [sunfreeware](#).

Network Solaris knows NFS. Linux knows NFS. This might actually be the best way of sharing data

Running foreign binaries

Linux binaries on Solaris

Is supposed to work using a program called "lrun". I have not got around to testing it yet. See the Solaris web site for more information.

Solaris binaries on Linux

Are as far as I know unsupported.

A. Finding additional info

Here are some URI that might help you:

- Sun Freeware [<http://www.sunfreeware.com>] My first stop for Solaris resources. It has many GNU-tools in binary format ready to install.
- "Community Software for Solaris" (CSW) [<http://www.blastwave.org/>] has many softwarepackages as well.
- Sun Microsystems [<http://www.sun.com>] Suns official web page.
- Solaris Homepage [<http://www.sun.com/software/solaris/>]
- Solaris Hardware Compatibility List (HCL) [<http://soldc.sun.com/support/drivers/hcl/index.html>]
- Solaris Online Documentation [<http://docs.sun.com>]
- My Website [<http://max.berger.name/>] You will always find the newest version of this document somewhere on this page.

B. Credits and legal information

Credits

Thanks to Marcel Meyer for beta testing this HOWTO while installing Solaris on his machine.

Thanks to Marino Miculan for the floppy disks section.

Thanks to Mike Khosraviani, John Misino, Ted Remillard, Ryoji Kawagishi, and Markus Wagner for information about grub.

Thanks to Ferenc Veres for the link to CSW.

An addition, many thanks to Baptiste M  l  s for translating this HOWTO into french, KIMURA Tomoaki for the tranlastion into japanese, and to all other translators that I do not know about!

Questions, comments, suggestions

Questions, comments, and suggestions are always welcome. You can reach me via e-mail at <mailto:max@berger.name>

License

This document is copyrighted © by Max Berger. You may use and/or modify it according to the Linux Documentation Project License (LDPL) found at <http://en.tldp.org/COPYRIGHT.html>.

According to the license you do not have to tell me, but I would very much like to know when you modify and/or republish this document.

Although the information given in this document is believed to be correct, the author will accept no liability for the content of this document. Use the hints and examples given herein at your own risk.