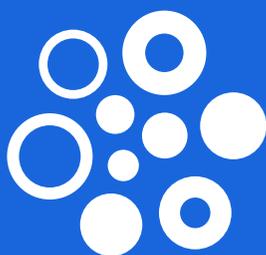




opensolaris

Try a different Unix

So you're a Linux guru? That don't impress Koen Vervloesem much. Real geeks use even more niche operating systems...



So you've been playing around with alternative OSes for a while and you reckon you've got this Linux thing mastered. Maybe you've tried Mac OS X and found it a bit too restrictive (or expensive); perhaps you've kicked the Hurd's tyres and thought you'll come back to it when it's something more than just a clever name.

If you're looking for something else to play with, we humbly suggest OpenSolaris. Like Mac OS X, which we looked at last issue, OpenSolaris is based on Unix; also like OS X, it was until recently tied to a specific type of processor (in this case Sun's SPARC architecture) but now works on a range of architectures including x86. Unlike OS X though, OpenSolaris is open source, so you can download it for free and start fiddling with it.

The source code to Solaris (well, most of it) was released to the community in June 2005, and to make sure that it became a usable home system rather than just a server OS, Sun hired Ian Murdock, founder of the Debian project, to produce OpenSolaris. The first release appeared in May 2008 and the distribution adopted an approximately half-yearly release cycle: after OpenSolaris 2008.05 we got versions 2008.11 and 2009.06.

Sun advertises its operating system as a full-blown distribution, including the Gnome desktop. But is OpenSolaris interesting enough for a Linux user? If you're happy with your favourite Linux distribution, why would you try OpenSolaris? In some aspects it's much like a

regular Linux distribution, but in other aspects it's completely different. Do the exciting features of this operating system outweigh the trouble of learning another Unix environment with other tools?

Installing OpenSolaris

Just like most Linux distributions, OpenSolaris comes with a live CD and a graphical installer that asks you for the standard information, including your location, preferred keyboard map, time/date etc. This will be familiar for Linux users, and if you're installing OpenSolaris as the sole OS on a computer you'll hardly notice the difference, but if you want to create a dual-boot system with OpenSolaris and Linux you might run into problems at the disk-partitioning stage.

The OpenSolaris installer considers all logical partitions on the disk as one extended partition, so it can't be installed on a logical partition. If you choose to install OpenSolaris on this extended partition, all enclosed logical partitions get overwritten. Second, OpenSolaris uses ZFS instead of ext3 as its filesystem. Linux has no ZFS support in the kernel because the Free Software Foundation doesn't consider it free enough to be bundled with GPL software, so if you want to get access to your OpenSolaris documents in Linux you have to mount the ZFS filesystem with Fuse as a filesystem in userland.

A third issue is that the standard *Grub* version that comes with Linux distributions doesn't understand the



› OpenSolaris' hardware support is a lot more predictable than it is in Linux, though not as extensive.

ZFS filesystem. So when you install OpenSolaris first and then your favourite Linux distribution, you can't boot into OpenSolaris anymore. The solution is to first install Linux and then OpenSolaris, and add the section for your Linux distro to *Grub's menu.lst* in OpenSolaris.

Hardware support

Linux has a big advantage over OpenSolaris in that it supports a lot more hardware, but OpenSolaris makes up for this by having a fixed device driver interface. Where the Linux kernel developers give priority to adding features even when they break compatibility with hardware drivers (which creates more work for the distro makers) OpenSolaris keeps the driver interface static, so if your printer worked with OpenSolaris 2008.05 it'll work with 2009.6 – users can even run 10-year-old drivers written for the original Solaris platform. OpenSolaris also gives you a clear overview of what is supported, rather than the suck-it-and-see approach favoured by Linux. The best way to find out whether specific

hardware components are supported is by searching the Solaris Hardware Compatibility List online.

Another option to test hardware support on a computer is simply to fire up the live CD. The Device Driver Utility icon should show up on the desktop, which detects all available hardware and lists which driver supports it, even if it is third-party. For example, when I fired up the Device Driver Utility on my Dell laptop, the program said that it didn't have a driver for my WLAN chipset (from Broadcom), but referred me to a website where I could download a third-party driver. There is also a related utility, the Device Detection Tool: this is a Java program giving the same information, which you can run on Windows, Linux and Mac OS X. So with this tool, you get a perfect overview of the hardware support before you even install OpenSolaris.

ZFS: the Zettabyte File System

One of the most important reasons to use OpenSolaris is its filesystem, ZFS. Conceptually, ZFS is simple: disks are assigned to pools, and data sets are made of pools. Filesystems and volumes are two types of data sets. For pools, OpenSolaris has the administration command *zpool* and for datasets OpenSolaris has the administration command *zfs*. We'll focus here on pools. This is how you list the available pools:

```
$ zpool list
NAME SIZE USED AVAIL CAP HEALTH ALTROOT
rpool 15.9G 1.28G 14.6G 8% ONLINE -
```

Rpool is the default pool created by the distribution's installer. With the **zpool status** command, you can display the status of all pools:

```
$ zpool status
pool: rpool
state: ONLINE
scrub: none requested
config:
NAME STATE READ WRITE CKSUM
»
```

Quick tip

The simplest way to find a device's name is with the *format* command, which shows the available disks. Just be sure to exit the *format* command with Ctrl+C, or you may end up inadvertently formatting one of your drives!

Some differences under the hood

Under the hood OpenSolaris is very different from a Linux distribution. You can see this in a different filesystem layout and different base commands. For many tools, OpenSolaris has two versions: the Solaris ones are in **/usr/bin** and the GNU ones are in **/usr/gnu/bin**. Because OpenSolaris wants to ease the transition for Linux users, **/usr/gnu/bin** comes first in the **PATH** environment variable. Other tools

are available only in a Solaris variant, which can be confusing. *lconfig* is such an example: as a Linux user, you'll soon find out that the syntax is different from what you expect. Even the network interfaces are called differently: the standard interface is not called eth0, but for example nge0 or bge0, reflecting the driver used. All in all, you have to relearn a lot of commands, which requires some effort.

Some filesystem differences

Linux directory	OpenSolaris directory
/home	/export/home
/var/log	/usr/adm, /var/adm, /var/log
/tmp	/var/tmp
/sys	/devices
/dev	/dev
/lib/modules/foo/*	/kernel/drv/*
/boot/grub	/rpool/boot/grub

Some command-line differences

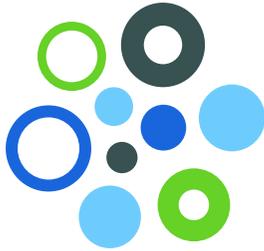
Linux command	OpenSolaris command	Aim
sudo	pfexec	Execute a command as root
apt-cache search foo	pkg search -r foo	Search for a package containing foo
apt-get install foo	pkg install SUNWfoo	Install package foo
apt-get dist-upgrade	pkg image-update	Upgrade all packages that have updates available
lsmod	modinfo	List loaded drivers
insmod	modload	Load a driver
rmmod	modunload	Unload a loaded driver
top	prstat	List the running processes
free	vmstat	List the free memory
cat /proc/cpuinfo	psrinfo -v	List processor info
ifconfig	ifconfig -a	List all network interfaces
parted	format	Format a disk

```

»  rpool      ONLINE  0      0      0
   c7d0s0    ONLINE  0      0      0
errors: No known data errors

```

One of the outstanding features of OpenSolaris is related to the filesystem: snapshots. A snapshot is a point-in-time copy of a ZFS filesystem or volume, saving the state for later reference or recovery. You can create a snapshot with the `zfs snapshot` command, but OpenSolaris 2009.06 has integrated this functionality in *Nautilus* and the Gnome Administration panel Time Slider.



Virtualisation with OpenSolaris

OpenSolaris supports a variety of virtualisation technologies with different degrees of isolation, flexibility, performance and ease of use. Of course it runs *VirtualBox*, but it also has a port of *Xen*, called *xVM Hypervisor*. A lesser known but equally interesting virtualisation technology in OpenSolaris is Zones, a feature resembling Linux-VServer, but

with the advantage that it's not a separate patch set but supported in the official kernel.

Processes running in different Zones are completely isolated from each other. This type of virtualisation is called operating-system level virtualisation. Although each zone appears as a standalone operating system, in reality there's a single instance of the OpenSolaris kernel running behind all of them, which means that Zones are relatively light on processing power. Here's how to set up virtual instances of OpenSolaris with Zones:

1 Configure a zone

First we create a filesystem for the zones to reside in. Then we create a zone and configure the path where the root directory tree of the zone is placed. We add a virtual network interface to the zone and wire it to the physical interface `nge0`, and in the last-but-two line of code we configured this network interface with a specific IP address.

```

$ pfexec zfs create -o mountpoint=/zones rpool/zones
$ pfexec zonecfg -z myzone
myzone: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:myzone> create
zonecfg:myzone> set zonepath=/zones/myzone
zonecfg:myzone> add net
zonecfg:myzone:net> set physical=nge0
zonecfg:myzone:net> set address=192.168.1.50
zonecfg:myzone:net> end
zonecfg:myzone> exit

```

To see all zones, we execute the `zoneadm list -cv` command, which shows the status of the `myzone` zone as we've configured it:

```

$ zoneadm list -cv
ID      NAME      STATUS  PATH      BRAND  IP

```

Create a RAID mirror with ZFS

1 Create a single-device pool

```
# zpool create tank c7t0d0p0
```

We choose `tank` as the name of the pool. `c7t0d0p0` is the device name. Now the pool `tank` is available for all file system operations, and it also shows in the `zpool list` command.

If you have two drives available from the beginning, you can also create the mirror from the start in one command:

```
# zpool create tank mirror c11t0d0p0 c7t0d0p0
```

2 Attach a second pool

If you later add a second hard drive, it's very simple to convert your single-device pool to a mirror:

```
# zpool attach tank c11t0d0p0 c7t0d0p0
```

3 Replace a defective drive

Should one of your drives become defective (eg `c7t0d0p0`), you simply add a new one and replace the drive in the pool with one command:

```
# zpool replace tank c7t0d0p0 c10t0d0p0
```



Step by step: Take snapshots with ZFS



» Enable automatic snapshots

Go to System > Administration > Time Slider and click on Enable Time Slider. Now OpenSolaris will regularly take snapshots of all available ZFS filesystems. If you would like to fine-tune which mount points will get snapshotted, click on Advanced Options. You can also adapt the maximum proportion of filesystem capacity that should be used. The regular snapshots take a lot of space, so by default Time Slider reduces snapshots when storage space usage exceeds 80% of filesystem capacity.



» Restore a snapshot

If you suddenly realise that you've accidentally deleted or overwritten an important file, it's time to use the snapshots. Open *Nautilus* and navigate to the directory your file is in. If you click on the Restore button, you can navigate the snapshot history of the current location. For example, if you deleted a file you can use the slider to navigate to the time when the file still existed. You can open the file to see if it is the right version, or you can right-click on the file and select Restore To [directory]."

```
0      global  running  /          native  shared
      - myzone  configured /tank/zones/myzone ipkg
shared
```

As you can see, there's also a zone called **global**. This is the OpenSolaris installation itself.

2 Install the zone

Now that the zone has been configured, we have to install it. Essentially, this creates a second installation of OpenSolaris inside the zone:

```
$ pfexec zoneadm -z myzone install
```

This command could take several minutes to finish, after which the status of the zone will have changed from **configured** to **installed**.

3 Boot the zone and log in

Once our zone has been installed, we can boot it with the command **pfexec zoneadm -z myzone boot**. This takes only a few seconds, after which the status will change from **installed** to **running**. Now you can log in with **pfexec zlogin -C myzone**. During the first login, you have to configure the host name, time zone, root password etc, just like a regular install does.

Now you can execute commands in the zone just like you do on the global OpenSolaris install. You can also halt, reboot and shut down the zone from within it, but you can also halt a zone from within the **global** zone with the command **pfexec zoneadm -z myzone halt**.

OpenSolaris on the desktop

Because OpenSolaris is advertised as a desktop distribution, it's fair to compare it with current Linux distributions. However, the first thing you notice is that the operating system is much slower than Ubuntu on the same

hardware, so don't think about installing it on older hardware. For the rest it looks like a fairly standard Gnome desktop, although *NetworkManager* is replaced by an application called *Network Auto Magic*, which does more or less the same thing but has fewer features. While most high-profile applications (including *OpenOffice.org*, *Firefox*, *Thunderbird* and *Rhythmbox*) are installed or available via the package manager, you have less luck if your favourite application is less known. For many Linux users this will be a showstopper, although packages can be ported.

And finally... YouTube!

We told you OpenSolaris is a proper desktop OS, so naturally you'll want to get Flash installed and stimulate your left brain. To get Adobe Flash Player, go to <http://get.adobe.com/flashplayer> and click the Agree And Install Now button (the website will automatically detect that you're using

Solaris). Select Save File in the window that pops up, then click OK, which saves it to your Downloads directory. Then

open a terminal window and execute the following commands:

```
cd ~/Downloads
mkdir ~/.mozilla/plugins
bunzip2 flash_player_10_solaris_x86.tar.bz2
tar -xvf flash_player_10_solaris_x86.tar
mv flash_player_10_solaris_r22_87_x86/* ~/.mozilla/plugins/
```

When you next restart *Firefox*, Flash 10 should be working. There – an advanced filesystem, easy virtualisation, *Firefox*, *OOo* and more. Could OpenSolaris be the new Linux? **LXF**

Quick tip

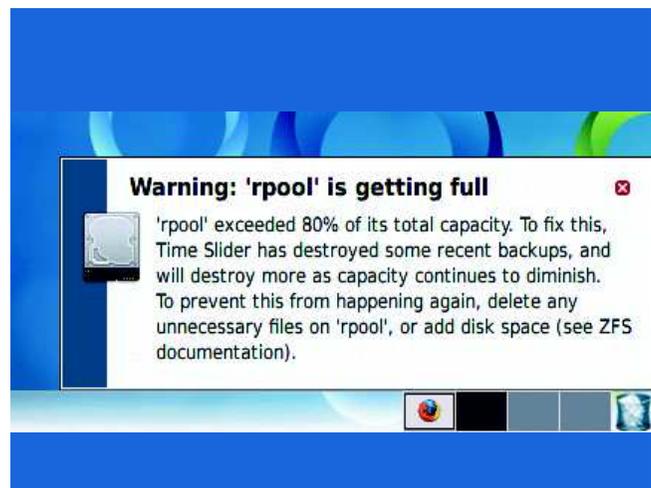
A simple and cheap way to experiment with ZFS pools is to use some USB sticks. Just plug some sticks in your computer, create a ZFS RAID pool on them, remove one stick and see what happens.

“Most high-profile applications are installed or available.”



» Compare different snapshots of a file

If you want to browse the file history of a given file, right-click on it and select Explore Versions (this option is only visible when there are different versions of the file to choose from). The Time Slider File Version Explorer window now shows you the different versions of the selected file. If it's a text file, such as code or plain text, you can compare the differences: just select an older version and click the compare button. If the file is an image, you'll be able to see the differences immediately.



» Take or delete snapshots manually

To take a snapshot of a directory manually, just click on the camera icon on the right of the time slider in the directory. If you've moved the slider to a previous snapshot, you can delete this one by clicking on the Delete button. If after a time you discover you have a lot of space wasted by snapshots you don't need anymore, just open the System > Administration > Time Slider panel again and click on Delete Snapshots, which presents you a list with all available snapshots. Select the ones you want to get rid of and click on Delete.