



Klaus Knopper is the creator of Knoppix and co-founder of the LinuxTag expo. He currently works as a teacher, programmer, and consultant. If you have a configuration problem, or if you just want to learn more about how Linux works, send your questions to: klaus@linux-magazine.com

# ASK KLAUS!



with it. I haven't found a setting in the US that seems to give me fewer xrns than others, and I don't understand why AGNULA would be so much better in this regard.

Is it possible with any of the newer versions of ALSA to do this without a complete uninstall, recompile, and reinstallation? I know that the second card will have to be slaved and connected to the first one via the SMPTE or Word Clock jacks. I need to find out how to get both of these cards working as one because I run a recording studio and need the extra inputs. Any suggestions?



In short, your question is answered here: <http://www.sound-man.co.uk/linuxaudio/ice1712multi.html>. With this configuration, you basically tell the ALSA-library which hardware inputs and which card to use for which (virtual) audio input for applications, and couple mixers so that the two cards behave like one. Because this is done via software, it is a timesharing-critical task, which explains why you are likely to get overruns/underruns when not preparing your system especially for this purpose. Just tuning system performance and process

priorities is not enough. You will most likely need the real time kernel extension (<http://rt.wiki.kernel.org/>). The reason why you had the best audio performance using the AGNULA Live CD may be related to the fact that they, too, used a kernel modified for real-time applications. Most "mainstream" distributions do not (yet) feature this kernel patch because it is not part of the vanilla kernel, adds some overhead, and modifies a lot of internal stuff such as interrupt handling of devices. For real-time audio synthesizers and sound processing, it is can be essential to have this kernel extension, regardless of how fast the computer is otherwise.

## Linux RAID



I have been searching for some solution on best practice in Software RAID 1. I was planning on using Ubuntu Server 7.10. My first setup was with two disks. On both disks I configured 500MB SWAP and the rest as RAID. I formatted the RAID to RAID 1 and EXT 3 mount point /. So everything went on the RAID partition. It worked, but what is it worth if I can't break it and still boot?

If I remove the second disk, I can't boot at all. If I remove the first disk, I get BusyBox after a while. I did the test on VMware Workstation 6 to be sure I could do it, but with no luck. How can I set up RAID so I can be assured it will actually recover from a disaster?



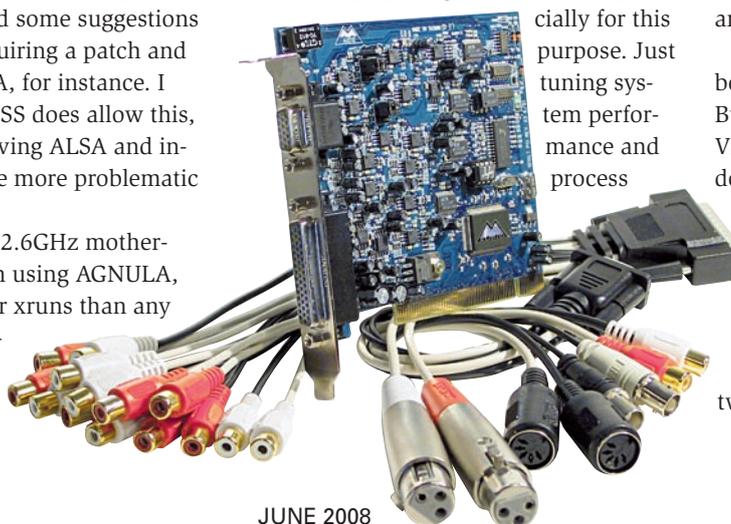
A RAID 1 (mirroring) configuration is supposed to ensure that data is always stored in two independent places so that it is easy

## Multiple Sound cards



I've got two M-Audio Delta 1010LT cards that I want to make work together as one large card so I can have 16 audio ins on my machine. I've read several descriptions of how to do this, and some suggestions are just painful – requiring a patch and reinstallation of ALSA, for instance. I have also read that OSS does allow this, but I think that removing ALSA and installing OSS would be more problematic than practical.

I'm using an AMD 2.6GHz motherboard, and have been using AGNULA, which gives me fewer xrns than any other of the multimedia distros I've tried, so I've stuck





## Take Ubuntu Further

Ubuntu Live is the official, trusted event for the Ubuntu community, welcoming all the different segments within the Ubuntu ecosystem—experts, long-time users, established companies, and newcomers alike in a passionate, energizing environment.

### Tracks at Ubuntu Live

- **Ubuntu in Action:** Find Ubuntu in the community and in some unexpected places
- **Ubuntu Means Business:** How Ubuntu works for, and in, business settings
- **The Tech View:** Cool technology inside and on top of Ubuntu
- **Public Sector:** Learn how Ubuntu is changing the world, or at least the public sector and NGOs
- **Management Systems:** Tips and tools to deploy and manage Ubuntu systems

**Register Now  
and Save 15%**

Use discount code  
**ubu08lmg**

[www.ubuntu-live.com](http://www.ubuntu-live.com)

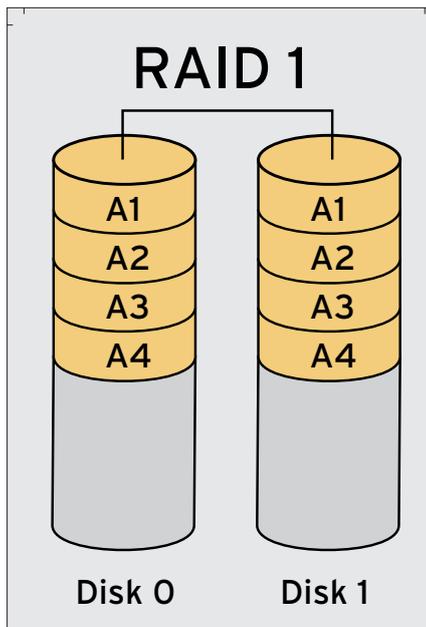
July 21 – 22, 2008 • Oregon Convention Center • Portland, OR

Co-presented by



**CANONICAL**





**Figure 1: RAID 1 with two disks (disk 0 and disk 1) over one logical volume A with all blocks replicated/mirrored from drive 0 to drive 1.**

to recover in the case of a hardware failure of one disk only. With some tricks, it is possible to use RAID 1 also for quick failover boot from the remaining “good” drive in single-drive mode, but it is not recommended to do this because you may fail to notice that you lost a drive (thus relying on a single hard drive again), and the goal of always running data duplication during normal operation is not guaranteed.

So, in case of a failure of one disk, the process would be to replace the defective disk by a good one and run the recovery procedure (i.e., starting disk replication to the new drive from the remaining drive). The last step can be done during normal operation in which the RAID array will run in “degraded” mode until data recovery is complete so there is no big downtime, and in some cases – provided that your hardware supports it – disks can even be replaced without a shutdown or reboot. What you asked for is really a “hot standby” storage that can, in case of failure, be used instead of the full array until you find time to replace the defective disk. For this, either the raid or mount configuration needs to be changed – manually or by a boot script – in order to allow temporary running with a single disk only.

For setting up RAID and changing its configuration, consider using `mdadm`.

The administration shell you are dropped into in case of a disk failure should allow you to issue the `mdadm` commands necessary to take out a disk or repair an array. The commands depend heavily on your disk configuration, so I cannot give you a good example here.

### FTP Problem

 I recently purchased the January 2008 *Linux Magazine*, which included the Fedora 8 DVD. I have successfully installed the operating system on an IBM X230 eServer. I’m trying to set up an FTP server to allow access to useful files for users who are on the road. I installed ProFTPD (included on the Fedora 8 DVD) and opened the correct ports on the Linux server and my router.

When I try to access the FTP server (`FTP://123.100.132.164`), it will ask for a user name and password, but it always fails, even with `root` as user. I have added users to the FTP group but still have no luck. I’m sure I’ve missed something simple.

 Usually, configuring ProFTPD is quite straightforward. The problems you report seem to be related to a wrong authentication mechanism or insufficient privileges for the ProFTPD server for changing user credentials or reading authentication data.

First, the ProFTPD server should be started by its init start script as root, and will drop privileges as necessary for each session:

```
sudo /etc/init.d/proftpd start
```

Failing to access the “root” account via FTP login is the default, secure behavior, otherwise it would allow anyone to probe for insecure root passwords. Plus, the password is transferred in plain text with unencrypted FTP. So, please use a different account for testing.

ProFTPD allows you to use different authentication mechanisms, and usually the Pluggable Authentication Module (PAM) is set as default. This means that the Unix login passwords set for each user are also used for FTP logins.

An exception is the “anonymous” or “ftp” user, which is kind of a “virtual” user with no password (the actual Unix

account is locked, and only used for file owner and process ID). You can enable “anonymous” ftp access in `/etc/proftpd/proftpd.conf` by uncommenting the section shown in Listing 1, which should be part of the standard ProFTPD configuration. After restarting ProFTPD, you should be able to access your ftp server with “anonymous” or “ftp” as the log-in name, and an arbitrary password (maybe containing an @). ■

### Listing 1: Enable Anonymous

```
01 <Anonymous /home/ftp>
02 User ftp
03 Group nogroup
04 # We want clients to be able
   to login with "anonymous" as
   well as "ftp"
05 UserAlias anonymous ftp
06 # Cosmetic changes, all
   files belongs to ftp user
07 DirFakeUser on ftp
08 DirFakeGroup on ftp
09 RequireValidShell
   off
10 # Limit the maximum number
   of anonymous logins
11 MaxClients
   10
12 # We want 'welcome.msg'
   displayed at login, and '.
   message' displayed
13 # in each newly chdired
   directory.
14 DisplayLogin
   welcome.msg
15 DisplayChdir
   .message
16 # Limit WRITE everywhere in
   the anonymous chroot
17 <Directory *>
18 <Limit WRITE>
19 DenyAll
20 </Limit>
21 </Directory>
22 </Anonymous>
```

Send your Linux questions to  
[klaus@linux-magazine.com](mailto:klaus@linux-magazine.com).