Error messages Explained

Graham Morrison decodes the secret meaning behind the most common Linux error messages and helps you cure the problems in the process.

Some people are scared of Linux because the error messages it produces seem to imply the coming of the apocalypse. And there's a great number of them. If you search for the word 'Error' in our forums, you get more than 150 pages of results. That's a lot of people experiencing a lot of problems.

The biggest difficulty for these users isn't the number of error messages; it's trying to get something useful out of them. What does 'Kernel Oops' mean, for example, or 'PCI Can't Allocate'? Linux error messages are obtuse, difficult to understand and rarely helpful. Which is a pity, because the vast majority of problems can be solved quite easily, and a considerable number involve the same problems recurring again and again. In business speak, these are low-hanging fruit. And it's these problems we want to target.

You shouldn't need to be a Linux expert to get your machine to boot, or a programmer to play a movie file. Yet it's this level of expertise that most error messages seem to assume of their users. We want to demystify these common errors, and provide solutions that should help ordinary Linux users side-step the problem and get their machine back on track. We've chosen areas we think are the most problematic. These include booting problems, general software usage, the filesystem, networking and distro installation.

We've picked a few of the most common errors from each, and explained what's happening along with the solution. The intention is that even if the problems don't apply to you, you can get an idea of how and why Linux error messages might seem arcane and a little intimidating. And hopefully, this will leave you with the knowledge to find a better solution that might help you to solve your own problems.

Distro installation

Every Linux distribution has a different installation routine, and each creates problems. Ubuntu might work for one machine and not for another. A machine with a working Ubuntu installation may not work with Fedora, or OpenSUSE, or Linux Mint, or Mandriva...

ERROR Can't boot from CD/DVD

If you're new to Linux, this is often your first experience of the operating system: you insert your new disc into the drive and restart the machine, only to be greeted by the same operating system you were using before. The problem is that your hard drive has a higher boot priority than your optical drive. Many modern BIOSes include a boot menu from where you can change the priority of your devices on the fly – try pressing the 'Escape' key or F12 when you first see something on the screen. From there, you can simply choose to boot from the optical drive.

Older machines might not have the same facility. You will then need to press either the F2 or 'Del' key at boot time to enter the system BIOS, and change the boot order from there. You can usually find the option under the 'Boot' menu, and you will need to save these changes to be able to boot from the optical drive. This is the same procedure you would use if you needed to boot from an external drive or USB stick, which can be just as useful if you find yourself in an internet cafe or in front of a corporate machine.

› **Occasionally, you will need to change the device boot priority to be able to boot a Live Linux distro from your optical drive.**

ERROR PCI: cannot allocate

There are many errors like this, and they mostly occur at boot time. They all share the same cause – badly behaved power management. The culprit is something called ACPI, the Advanced Configuration and Power Interface. Despite being a standard for power management, it has been causing problems for over ten years. The trouble is that hardware drivers have a habit of not fully implementing the specification. Whenever your machine's power management spins into action, such as when you turn on your machine, or resume from sleep, certain devices cause problems. Live CD installations make this problem worse, because they don't have the luxury of probing for exact hardware matches when they boot, or including every possible driver for every device, which is why this problem often occurs when installing off a Live CD.

There's only one thing you can do – turn off ACPI. You can sometimes do this from your system BIOS, but if not, you'll need to disable ACPI at boot time. Press Escape when booting to enter the *Grub* menu and select the option you normally use. Go down to the line that starts with **kernel** and press E to edit the line. At the end of this line add **acpi=off noapic**, press return and B to start the boot process. You should find that your machine boots without problems, and if you go on to install Linux, your distro should make a better job of choosing the correct drivers for the installation.

LXF109.errors Sec2:46 7/7/08 15:10:36

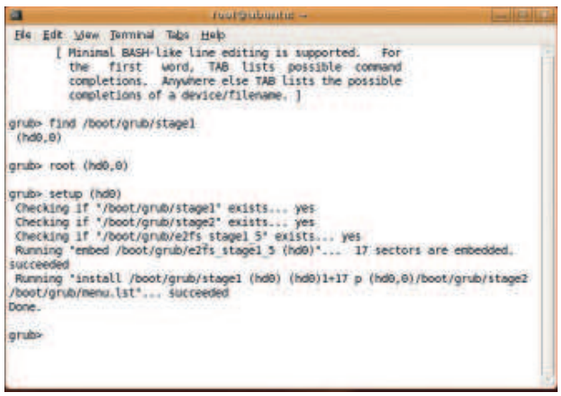
Booting problems

There's nothing worse than an error that stops your system booting – mainly because you're now without your primary problem solving tool. Yet booting problems are common. This is because we all like to install distributions, and we often run more than one on a single machine, as well as share a hard drive with Windows. Any one of these installations can mess up the boot routine, and getting a working installation back isn't always so easy.

ERROR Grub...

If this is all you see when you turn on your machine, the *Grub* boot menu has been corrupted. This is the part of your Linux installation that's responsible for booting the operating system. And the only thing you can do is boot Linux off some other media, preferably a Live Linux CD. When you get to the desktop, open a command line terminal, switch to the administrator account and type **grub**. This next step will also work if your *Grub* menu entries no longer point to your Linux partition.

Type **find /boot/grub/stage1**. This is searching for the location of the original boot drive, and it should return something along the lines of **(hd0,0)** – this is *Grub*'s own syntax for the location of the hard drive, and this is dependent on your own installation, so don't assume it's going to be (**hd0,0**). You should now type **root (hd0,0)** (or your equivalent) to tell *Grub* which partition is being used to boot from, followed by **setup (hd0)** to reinstall the boot loader into the disk's master boot record. You should then be able to restart your machine and boot normally.



› **Knowing how to re-install the *Grub* bootloader onto your main hard drive can get you out of all kinds of tricky problems.**

ERROR Out of range
ERROR Fatal server error: no screens found

This occurs when the preconfigured screen mode is incompatible with your monitor. Press Ctrl+Alt+F1 to switch to a console view, and log in as root (or use *sudo* from your normal user account in Ubuntu). Users of Debian-based distros can type **dpkg-reconfigure xserver-xorg** to reconfigure the screen.

Other users will have to fix their settings manually as follows. First, type **cd /etc/X11** followed by **cp xorg.conf xorg.lxf** to make a backup of your configuration file. Now open this file with whichever command line editor you're most comfortable with. Type **nano xorg.conf** if you're not sure. If you know your screen's specification, scroll down the configuration file and look for 'Section Monitor'. You then need to hand-edit the horizontal and vertical refresh rates. If you don't know your screen's resolution, scroll even further down the file until you find the Screen section. You need to delete all the high screen resolutions here, as we're looking for the lowest common denominator (we'd suggest removing any resolution larger than 1024x768). You'll be able to increase the resolution from the desktop when you get your screen working. If neither of these methods work, the last failsafe option is to change the 'Device' driver to "vesa", sidestepping your graphics drivers entirely.

ERROR Kernel panic!

A 'Kernel panic' or 'Kernel oops!' message is the closest us Linux users get to the Blue Screen Of Death that still haunts Windows users. And like the Windows equivalent, there's very little you can do when one occurs other than hold down the power button. The kernel is at the heart of your Linux system, and a panic is usually caused by misbehaving hardware forcing the kernel into uncharted areas of your system's memory. The best solution is a kernel upgrade, as the hardware problem may have been solved in a newer version. But you may need to revert to an older version of the kernel from your boot menu to be able to use Linux, to then install the upgrade.

The other option is to identify the offending hardware. If you've just made a hardware change or installed a new driver, this is likely to be the culprit. Otherwise, you might have to resort to removing each piece of hardware in turn and seeing if your machine boots. Despite the reams of output from a kernel panic error, there's usually very little the average user can understand, as the original error has sent the kernel in a completely random direction.

ERROR Incorrect username or password

You'd be surprised at just how many readers phone us to say they've forgotten their password, or even worse, promise they've never been asked for one. Fortunately, all is not lost. You need to boot your machine into single-user or recover mode. To do this, press the Escape key as soon as your computer leaves the BIOS screen on startup. This will show the *Grub* boot menu.

If there isn't the option to boot into either a single user or recovery mode, choose the kernel that normally boots (usually top of the list), and press E to edit the boot parameters. Move to the line that begins **kernel**, then press E again to edit the text on that line. Make sure the cursor is at the end of the line and add the following: **rw init=/bin/bash**. Press Enter, followed by B to boot. We've just changed the default boot option to open a *Bash* terminal rather than the normal session. As with the safe and recovery modes, all you now need to do is type **passwd** followed by the name of the user whose password you need to change. Without a username, the **passwd** command will change the root password. Just restart the machine to use your new password.

»

› **If you've got physical access to a machine, you can easily change any user passwords by booting into recovery mode.**

LXF109.errors Sec2:47 7/7/08 15:10:57

Error messages explained

Filesystem

The filesystem is the part of your Linux installation responsible for reading and writing files, including those on external devices. It's usually quite robust, but power cuts and badly behaved hardware can occasionally cause problems. And problems with your filesystem are usually tricky to solve. For this reason, the solutions we list normally use the command line.

ERROR Run fsck manually

There are dozens of variations on the basic filesystem error. These errors commonly occur while booting your machine, and often result in a 'Read only' warning for your root filesystem. This means that if your machine manages to boot, you won't be able to do anything. The solution is to boot from a Live CD, as this ensures your damaged drive isn't touched by the boot process and a filesystem repair tool will be able to make the necessary changes to fix any problems. The command you need to run is **fsck -f /dev/drive**, but you need to replace **drive** with your root partition device. This is dependent on your installation. The first partition on the primary drive will be **sda1**, for example. The original error should contain this information. You also need to run **fsck** as the system administrator, which means Ubuntu users need to create an account from the Ubuntu Live CD by typing **sudo passwd root**, followed by **sudo bash**.

ERROR Device is busy

Many of us use USB sticks and external hard drives, but sometimes these devices refuse to dismount themselves from the filesystem. And you can't simply disconnect them either, as there's a possibility you'll lose locally cached data that hasn't yet been written to the device. You can solve this problem by typing **sync** on the command line, forcing any cached data to be immediately written to the device, but this won't solve the unmountable problem. To solve this, you need to use a command called **lsof**, which may need to be installed separately. Typing **lsof mountpoint** will list the system processes currently accessing files on the device, and these will need to be killed before the system can unmount the drive.

This is also a handy thing to know if you're having trouble unmounting a CD or DVD drive, as the technique is the same (only without the **sync** issues, as they're read-only devices). Here's an example session:

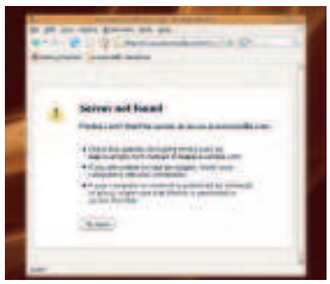
```
> umount /mnt/content
umount: /mnt/content: device is busy
> lsof /mnt/content
COMMAND  PID USER  FD  TYPE DEVICE SIZE NODE NAME
smbd  23222 root cwd  DIR  8,33 4096  2 /mnt/content
> kill -9 23222
> umount /mnt/content
```

Networking

Very few people enjoy troubleshooting network connections. But in our wired world, they're unavoidable. Fortunately, there are a handful of errors that account for a significant proportion of problems, and we've solved them for you.

ERROR Server not found

This is the classic network error. You turn on your machine, wait a minute for it to boot, and click on the link that loads your favourite web page. Except it doesn't load, and you're greeted with a server error instead. The problem is that there is no connection to the internet, and there are many probably causes. The best way to solve this problem is to work back from main connection. Is your router powered on and working? Is your broadband connection working on the router? If you're using wireless, you obviously need to check the wireless connection on your Linux machine. If you're using wired Ethernet, you need to check that both LED indicators that surround the cable are lit. An illuminated orange LED indicates a working connection, while the green LED flickers with any activity.

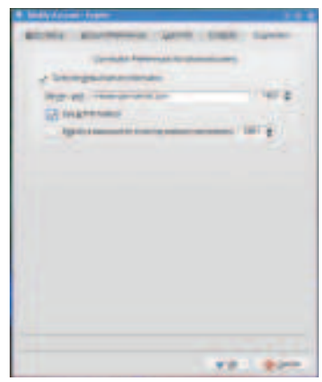


› This kind of error message can be caused by a virtually limitless number of problems.

If all these are working, the problem is with your Linux box. If you've checked your distribution's network configuration panel, and everything seems to be working as it should, you need to try a couple of command line tools. **ifconfig** generates a lot of output, but it's the quickest way of making sure your network connection has been assigned an IP address. Look for either **eth0** for a wired connection, or either **ath0** or **wlan0** for wireless, and make sure there's a sensible **inet** address for your network. If not, try **ifconfig eth0 down** followed by **ifconfig eth0 up**. You might also want to try the **route** command to make sure there's only a single defined gateway address. If you find two, remove one by typing **route del gateway_address**.

ERROR MSN won't connect

It doesn't matter which messenger client you're using – *Pidgin*, *Kopete*, *KMess* and *AMSN* will all occasionally refuse to connect to the server. This problem is usually down to a change in the server protocol, which means that each client needs to be updated. But it could also be down your local network. MSN is sensitive when it comes to firewalls and port forwarding. The solution is to use HTTP, which is normally an option in your account window. As HTTP is the same protocol used by web traffic, you shouldn't have any difficulty making the connection.



› If the connection options for MSN aren't working, switch to HTTP in the account settings page.

LXF109.errors Sec2:48 7/7/08 15:11:03

Software

This is one area of Linux use we all get frustrated with. OS X and Windows users are often amazed when they find Linux users can't simply download a package from the internet, double-click on it and install the application without any further problem. They can get the latest versions of applications like *Gimp*, *Inkscape* and *OpenOffice.org* by simply downloading a file and running it. Linux users have no such luck, and the problem is compounded by the way most distributions use a different installation method.

ERROR Permission denied

This error is a result of system security, and is a common problem when trying to execute applications on the command line and edit certain files. Linux locks down certain files and directories so that even if a user account is compromised, that user can't run system critical applications. This system is far more practical on a server, or a Linux box hosting hundreds of user accounts. And while it's also important on a single-user system, there's nothing wrong with side-stepping the precaution and giving yourself permission to run or open the file in question.

You can do this from either your desktop or the command line, but you need to be using the system administrator's account to be able to change the required permissions. From the command line, this means typing **sudo bash**, or just **su** on non-Debian systems. You can change ownership of a file by typing **chown username filename**; adding the **-R** argument will recursively change file ownership in a directory. But this won't help other users of your computer, as they'll still encounter the permission problem. The answer is to change the executable permission on the file so that anyone can run it. This is achieved using the *chmod* command. Type **chmod +x filename** to grant every user on your machine permission to execute the file. Similarly, **chmod +rw filename** should also grant everyone read and write access to the file.

ERROR Downloads won't run!

A couple of months ago, we included a *Runes of Avalon 2* demo on our DVD. It was tucked away inside a **tar.gz** file. Most of us, hardened to the eccentricities of Linux, don't even notice. But we received a few calls from new Linux users (the people we need to win over!), asking why the **tar.gz** file didn't run. The answer, of course, is that **.tar.gz** is an archive. We explained that it's like a **zip** file, so it needs to be decompressed into a directory, and the demo run from there. You can do this from most desktops with a right-click, or you can do the same thing on the command line by typing **tar xvf filename.tar.gz**, but there's no need for newbies to know this. You then need to look for either a **.bin** or **.sh** file, and click on this to execute the demo. If you're unlucky, you might need to type **./install.sh** on the command line from within the newly created directory. On behalf of Linux advocates everywhere, we apologise for this inconvenience.

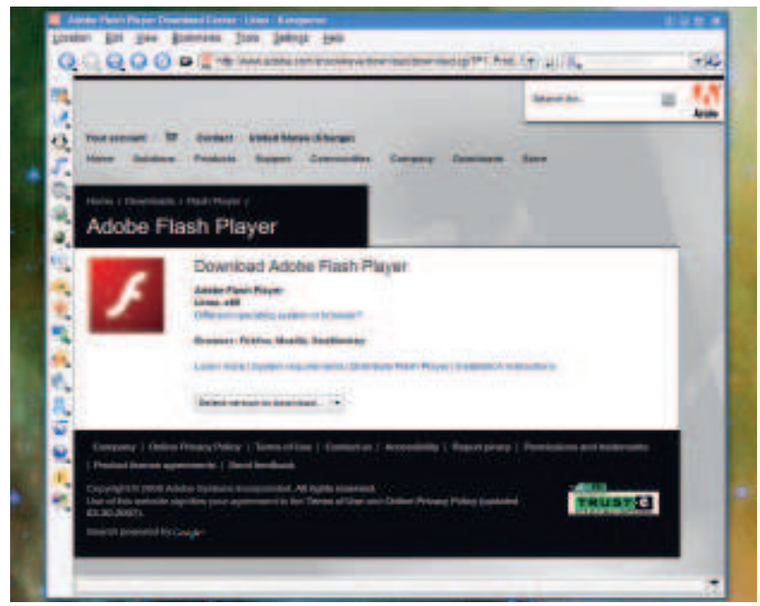
› Unless you're used to Linux, running demos and downloads isn't as straightforward as on other operating systems.

ERROR Flash movies don't move

Pity those new Linux users who boot up their fresh installation, only to find they can't waste the day on YouTube. Yes, very few Linux distributions include Flash playback support by default. What's worse is that your browser's hopelessly optimistic claim that installation is only a few clicks away is usually a lie. However, here's a failsafe way to get Flash support to work.

Search for 'adobe flash download' in Google and click on the top hit. From the 'version to download' drop-down menu on the new page, choose 'tar.gz for linux' and click on the agreement. The file will now be saved, and you need to remember the location where the browser has stored the resulting file. Next, open up a command line terminal and type **cd** followed by a space and the destination directory of the file you've just downloaded (this is normally going to be **~/Desktop** for your desktop. When you get there, type **tar xvf install_flash*** to uncompress the download and **cd** into the new directory. You now need to execute the installer by typing **./flashplayer-installer** and follow the on-screen prompts. A browser restart later, you will have a working Adobe Flash installation. LXF

› Even if Adobe Flash support isn't part of your distribution, a working installation is only two minutes away.



Over to you

There are thousands of different errors, and we've only highlighted a handful. But we'd like this to be the beginning of something like an FAQ, where we can help deconstruct some of the more mystifying messages. But we need your help. We need to know which messages you find the most troubling, and the most likely to put new users off Linux before they've even given it a chance. If you've got any brilliant examples of obtuse Linux errors, send them our way and we'll collate the best and let you know the results.

LXF109.errors Sec2:49 7/7/08 15:11:25