Debian Help System

Jonathan Eisenstein

jeisen@users.sourceforge.net

Jesse Goerz

jwgoerz@users.sourceforge.net

This document is intended to help Debian newbies learn about ways in which they can find information to solve problems they have while using Debian GNU/Linux.

Many thanks to Brian Potkin for his direct and accurate information about `/usr/share/doc`, man pages, and my error on what package zless belonged to. I would also like to thank Wayne Topa for his excellent contribution for using xman and his advice on the use of the mailing lists.

1. Introduction

Many people who use Linux say that Debian is the hardest distribution to use. This is probably true for someone who is trying out Linux for the first time. Don't worry if this describes you. After reading this document, you will learn the basic information gathering skills you will need to find solutions to most of the problems that you will encounter.

This document describes the Debian help system. Debian doesn't really have a formal help system, in that, all the information resides in one place or program. One of the great things about Debian is the multitude of resources which are readily available without cost. It will take some getting used to at first, but once you have mastered how to gather information about your problem, the solutions will come easily.

Note that many of these resources are not specific to the Debian system. Unless otherwise noted, each feature can be found on most Linux systems.

2. What's Installed?

The Debian designers have gone to great lengths to ensure the system is well documented. This pursuit has led them to include most of the necessary documentation on the Debian system for the packages you have installed. Let's take a look at some of the resources that are available on your computer.

2.1. /usr/share/doc/*

Almost every package that is installed under Debian installs some type of documentation information in the directory `/usr/share/doc`. Underneath this directory are subdirectories named after the packages which installed them. Here's what my `/usr/share/doc` directory looks like:

```
aalib1              diff             hostname         libc6-dev
abiword             docbook          iamerican        libcdparanoia0
acroread            dpkg             imlib-base       libcomerr2
adduser             dpkg-dev         info             libdb2
alien
```

Since there are so many directories in `/usr/share/doc`, I have actually cut this short so it'll be a little more readable. We'll take a look at a specific package and its documentation. Abiword is a nice word processor which imports Microsoft Word documents and quite a few other formats. Let's take a look at the abiword subdirectory.

```
README.gz  changelog.Debian.gz  copyright  examples
```

One of the really cool things about the directory `/usr/share/doc` is that many times the applications have `examples` directories in them. These subdirectories usually contain sample configuration files for programs you may be trying to use, or tips and tricks to make them work even better. You'll also notice that many of these files have the extension `.gz` on them. This means that the file is compressed in the gzip format. To read it, you must first uncompress it somehow. One way is to fully uncompress the file with the **gunzip** command. However, this can waste a lot of space, and the purpose of having the files compressed in the first place is to save space.

An easier and more logical way to read compressed files is to use the 'zless' command. This command automatically uncompresses the file using **gunzip**, loads the file in memory, and uses the **less** command to present it to you. In order to view the `README.gz` file, execute the command:

```
bash$ zless README.gz
```

If you get an error about a command not found, you may be missing the less package. **zless** is installed by default on Debian systems with the **gzip** package, but **less**, which zless depends on, may not be. If you need help installing missing packages, see the Newbiedoc "apt-get intro".

```
------> README.gz <------
```

```
----------------
Release Notes
----------------

Thank you for your interest in this Preview Release of AbiWord!

---- About AbiWord

AbiWord is an "Open Source" word processor.  (If you would like more
information about the concept of Open Source, you might start looking
at www.opensource.org.)
...
```

You can use the arrow keys to scroll the file up and down or use the space bar to advance one screen at a time. When you're done reading the `README.gz` file just simply hit the **q** key.

2.2. /usr/doc/*

The directory `/usr/doc/` is where all the documentation used to be stored, before Debian began changing to be compliant with current Linux standards. Anything located here instead of in `/usr/share/doc/` is in violation of current Debian policy and will change soon.

3. The Online Manual

3.1. The Manual

Once, in the early days of UNIX, users had the standard manuals available in print, on their shelves. 9 books were available, each one gigantic and hard to use. Someone eventually had the idea to make it standard to have all of these books on the UNIX system in the form of the **man** command, where it could be edited easily when new information became available, not to mention the added ease of having the manuals searchable and read quickly. These manuals are usually your first target when you have a problem, as nearly every aspect of your system is covered there.

Not only are most commands covered in the online manual, but often configuration files as well as functions for programming in Linux can be found. However, many users feel intimidated by man pages because they are often technical in nature and can be difficult to read for newbies. Once you get the hang of reading them, you'll find that the man pages are the greatest source of help on any system.

3.2. Man Page Layout

Manual pages have a very specific format that every official page should follow. Here is a sample, the documentation for the **uname** command. To read it, I type:

```
bash$ man uname
```

```
uname(1) uname(1)
```

```
NAME
uname - print system information

SYNOPSIS
uname [OPTION]...

DESCRIPTION
Print certain system information. With no OPTION, same as -s.

-a, --all
print all information

-m, --machine
print the machine (hardware) type

[...]

-v print the operating system version

--help display this help and exit

--version
output version information and exit

AUTHOR
Written by David MacKenzie.

REPORTING BUGS
Report bugs to <bug-sh-utils@gnu.org>.

COPYRIGHT
Copyright C 2000 Free Software Foundation, Inc.
[...]

SEE ALSO
The full documentation for uname is maintained as a Texinfo manual. If the info
and uname programs are properly installed at your site, the command

info uname

should give you access to the complete manual.
```

NAME: Here you get a brief description of the command after its official name

SYNOPSIS: The usage of the command, with optional segments listed in brackets. When [OPTION] is specified, it means that the command can have the options listed under the next section added to it.

DESCRIPTION: Detailed information about the command, including options that can be added to it. Unless specified either here or in the synopsis, you can add as many of these options as you want.

SEE ALSO: This is an important section of the man page that is often overlooked. If the information you are looking for isn't available in this page, it may be available in one of the pages listed here. Also listed here is information referencing the **info** command, if that information exists. A description of the info command is in the next section.

Using **man** works similarly to using **less**, once you're reading the page. For more information, try reading the man page for **man** itself!

3.3. xman

xman is a graphical version of the program **man** described above. If you find man pages tedious, or just really like your mouse, try typing this at the command line:

```
bash$ xman &
```

Alternatively, you could start xman from your menu system, which should look something like Figure 1:

Figure 1. Starting xman from your GUI menu

Whatever method you choose to start xman you should get a nice little window like that found in Figure 2:

Figure 2. The xman program

Just click on the "Help" button to get more information or click directly on the "Manual Page" button to look up man pages.

4. Info Pages

4.1. What are Info Pages?

While man pages are great sources of information about a command or concept, the fact that it provides one long page means that longer documents are hard to read. The **info** format was developed as a way to store more comprehensive instructions on your system. If a particular command is more complicated than a man page can document, the author commonly writes both a man page and an info page. While the man page will be a simple reference to the command and its options, the info page will be a longer reference, like a manual.

To make it easier to read longer documents, info pages are split up into sections, with links between them, much like a web site. You should feel comfortable very quickly when using info because it is so similar to a web browser.

4.2. Using Info Pages

You can start the **info** program the same way as the **man** program. For example, to get help on the **info** command itself, type

```
bash$ info info
```

You should get a screen which looks something like this:

```
File: info.info,  Node: Top,  Next: Getting Started,  Up: (dir)

Info: An Introduction
*********************

   Info is a program for reading documentation, which you are using now.

      To learn how to use Info, type the command 'h'.  It brings you to a
  programmed instruction sequence.

  * Menu:

  * Getting Started::             Getting started using an Info reader.
  * Advanced Info::               Advanced commands within Info.
  * Creating an Info File::       How to make your own Info file.
```

The text you see at the bottom of the screen followed by **::** are the links to other pages in this document. You can use your keyboard to put the cursor anywhere on the words of the link, and **enter** to select it. You may have some trouble navigating the pages, so here's a quick list of important keys:

Table 1. Important keys and their functions

Key	Result
SPACE	Page Down (To next page, if needed)
PAGE DOWN	Page Down (Until end of page)
BACKSPACE	Page Up (To last page, if needed)
PAGE UP	Page Up (Until beginning of page)
B	Beginning of page
E	End of page
TAB	Next link
N	Next page in document
P	Previous page in document

Key	Result
L	Previous page you visited
T	Beginning of document (The Menu)
Q	Quit the info program

You can also use a mouse to navigate the info pages. Just click on a link to follow it.

4.3. xinfo

If you'd rather use a graphical interface to browse your info files, you can use the program xinfo if it is installed. It is very similar to **xman**, except for info pages instead of man pages.

5. Other Help Commands

While man pages and info pages are the most reliable sources of information on a Linux system, there are other ways to obtain information if you still can't find your answer. For instance, what if you don't know the command you're looking for?

5.1. apropos

The other day a friend of mine was having a problem with a boot disk and asked me how to fix it. So my first instinct was to try:

```
bash$ man bootdisk
```

Unfortunately, this is what I got:

```
No manual entry for bootdisk
```

I knew that I had read something about boot disks before, but I couldn't seem to remember where. This is where the **apropos** command comes in handy. Apropos is kind of like saying, "I can't quite remember what that cool movie was, but I know the title started with 'The Good'" and having a friend start giving possible movies. That's what apropos does, only for topics on a Linux system. So here's what I did:

```
bash$ apropos bootdisk
mkboot (8)          - makes a bootdisk
```

Just like that, apropos gave my answer! Now all I had to do was:

```
bash$ man mkboot
```

Sometimes it will give you back more than one answer. Just go through each one until you get what you're looking for.

5.2. Using dpkg to find information

dpkg is the program which controls all the installation of .deb packages. This program can be used to help you find particular files or information about a program you may have installed but can't seem to get working.

Suppose you just installed a program but you can't find any documentation in the usual places (i.e. `/usr/share/doc/`). How would you find out where the package put the documentation? Let's pretend I didn't know where abiword installed it's documentation. I would use the dpkg command like this to find it:

```
bash$ dpkg --listfiles abiword
```

This simply tells dpkg to list all the files that were installed with the abiword package. Here's the result of that command:

```
/.
/usr
/usr/bin
/usr/bin/AbiWord_d
/usr/bin/AbiWord
/usr/bin/abiword
/usr/share
/usr/share/doc
/usr/share/doc/abiword
/usr/share/doc/abiword/examples
/usr/share/doc/abiword/examples/en-US
/usr/share/doc/abiword/examples/en-US/Latin1.abw
/usr/share/doc/abiword/copyright
/usr/share/doc/abiword/README.gz
/usr/share/doc/abiword/changelog.Debian.gz/
```

In the example above you can see the doc directory quite nicely but in reality when I executed this command all the files scrolled right off the screen and I wasn't able to read any of them. For especially large packages with lots of files we are going to need to enlist the help of another program called grep. Here's the new command to find what we are looking for:

```
bash$ dpkg --listfiles abiword | grep doc
```

The "|" character or pipe as it's called is typed by pressing **Shift**-\

This command tells dpkg to list all the files installed by abiword and then send that list to the program grep. grep then searches through the list looking for any occurrence of the word "doc". When grep finds the word doc on a line it prints that line to the computer screen. Here's what the output looks like:

```
/usr/share/doc
```

```
/usr/share/doc/abiword
/usr/share/doc/abiword/examples
/usr/share/doc/abiword/examples/en-US
/usr/share/doc/abiword/examples/en-US/Latin1.abw
/usr/share/doc/abiword/copyright
/usr/share/doc/abiword/README.gz
/usr/share/doc/abiword/changelog.Debian.gz
```

This is much easier to read then the previous example, and as you can see, it only printed out lines with the occurrence of "doc". The files didn't scroll of the screen and it's a nice short list. I can now look in those directories and see what I can find. This might seem like a silly example, but if you look closely you can use this to search for all kinds of things. You could replace "doc" with any word you are looking for. Some examples which come to mind might be "help", "ref", "html", "config", "conf", "etc", and "examples".

For more information about using **grep**, check out the NewbieDoc grep document.

6. Online resources

There are a lot of online resources for Debian. I won't be able to cover them all here, but I will cover the ones I use when I have problems or I'm just trying to learn something new.

6.1. Debian home page

The Debian home page (http://www.debian.org) is the central source for online information. There is so much to cover here that I'm going to break it out into a couple of sections. I highly recommend you go to the Debian web site and check it out. You will find more information on what Debian is about, Debian related news, security alerts and much more.

6.2. Debian mailing lists

The Debian mailing lists are the one thing which make it such an incredible distribution to use. No other distribution can claim to have such "round the clock" excellent support services for free. Debian has over 20 mailing lists covering subjects from running an ISP with Debian to issues of security. If you can think of it, Debian probably has a mailing list for it!

The best way to get familiar with the Debian mailing lists is to visit the mailing list web site (http://www.debian.org/MailingLists/). You will get some basic information about the sites and how they operate. When you're ready to subscribe to a mailing list go to the subscription (http://www.debian.org/MailingLists/subscribe) page. The most important mailing list to subscribe to for new users of Debian is the **debian-user** mailing list. Be warned though, the debian-user mailing list is a high-volume list. It's not uncommon to get 75 or more emails a day.

Subscribing to mailing lists does bring with it some responsibilities. Here's a quick list of some of the things you ought to do when posting to the list:

- Read the list for a few days and take note of how people ask questions. Pay particular attention to those who seem to get their questions answered quickly.

- Gather as much information as possible about the problem you are having before you post to the list. You'll get help faster and it'll be more accurate. Check out the section called Information gathering programs for tools that can help you do this.
- Make sure the subject line of your email is accurate and descriptive. Subjects like "I need help!!!" may be overlooked by just the expert who could answer your question. A smart thing to do is to list the package name and the problem in the subject. Many maintainers browse the mailing lists, and usually have their mail program configured to highlight subjects about their packages.
- Tell the truth (to the best of your knowledge) about what you've done to try and fix it, what has happened as a result, and what your experience level is.
- Realize that some of the answers may do damage to your system or may be inherently risky.
- Get familiar with the mailing list archives. You can search them here (http://lists.debian.org/search.html). It's a good idea to check there first, because your question may have already been answered. Answering the same question over and over can get very annoying; just ask anyone who has a 3 year old. (And remember, they don't love you like they do their 3 year old.)
- If your problem was complicated, and took some time to solve, post a summary of the solution, so others can benefit as well. Use something like this in your email subject line: "Solution: problem description".

Here are some things you should **NOT** do when posting to the mailing lists:

- Don't post to the list using html mail formats. This is by far the worst thing you can do on the mailing lists. It irritates everyone. 99.9% of the people who use Linux and don't use email programs that support html format. They'll tell you that's what web browsers are for (and they're right!). Make your posts in plain text only!
- Don't post unrelated questions in one mail message. It's better to split it into multiple emails.
- Don't give sensitive information to anyone on the list. There is absolutely no reason anyone on the list will ever need your root password, dial-up password, or any other type of private information whatsoever.
- When posting configuration files make sure you delete or change any passwords or other sensitive information which may appear in them. When people on the list ask for configuration files and tell you to "munge" the passwords, this is what they mean.
- Don't expect the first response to your question to always work. The Debian organization is made up of volunteers, and the person answering your question may not be an expert. What worked for others may not always work for you, but it's usually worth a try.
- Don't flame or ridicule people for giving you incorrect advice (if you know it is incorrect). Remember, this is a volunteer list of people just like you, trying to find answers. Simply correct them (if you know you're right) and move on.

6.3. Debian GNU/Linux FAQ

The Debian GNU/Linux FAQ is a very extensive document which covers a lot of (go figure) Frequently Asked Questions. You can check it out here. (http://www.debian.org/doc/FAQ/)

6.4. Finding packages

Need to find out what a specific package does? Looking for a specific package? Trying to install a specific program but you don't know the name of the package it's in? Try here. (http://www.debian.org/distrib/packages)

6.5. Debian documentation project

The Debian Documentation Project (http://www.debian.org/doc/ddp) is an ongoing project to provide quality documentation on the Debian GNU/Linux operating system for free. This is a great place to start looking when you have questions that haven't been answered elsewhere. You may also wish to browse this area in your free time, as there is a great wealth of knowledge in these guides and manuals. This is a great place to start looking when you have questions that haven't been answered elsewhere. You may also wish to browse this area in your free time, as there is a great wealth of knowledge in these guides and manuals.

6.6. NewbieDoc Project

Of course, there's always the NewbieDoc Project (htto://newbiedoc.sourceforge.net), which aims to provide answers for commonly encountered problems that new users of Debian may encounter.

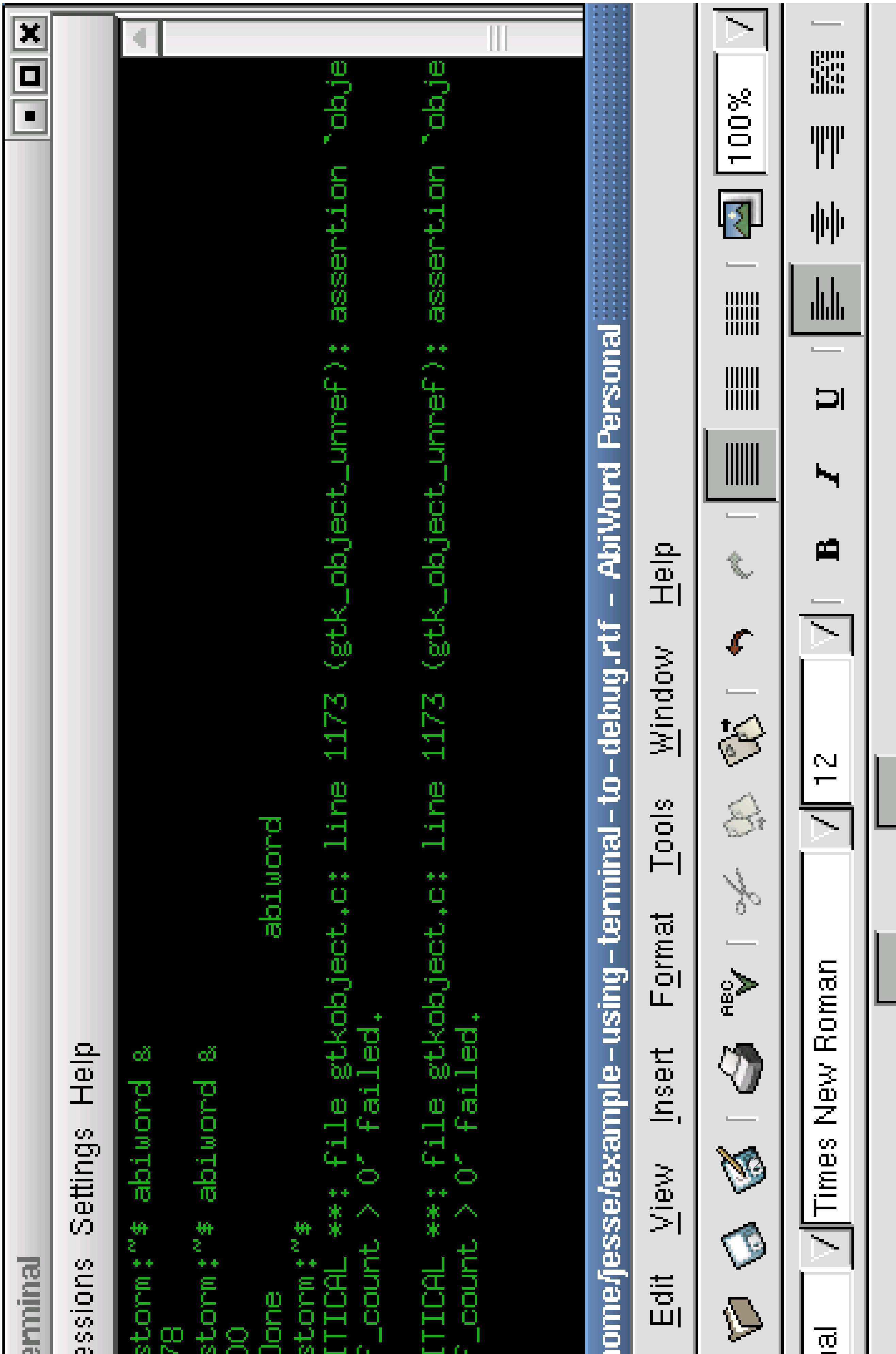
7. Information gathering programs

I'm covering these programs here just so you have a quick reference and know that they are available. Most of these programs are much more powerful and can do many more things than I'll explain. You may not ever have to use these but they may come in handy when trying to gather information so you can post an accurate question to a mailing list.

7.1. Using a terminal

Using a terminal can sometimes help you get information about a problem program. If you're running X-windows this is really nice. Let's say that my word processing program is acting weird and won't work the way I'd like to. All I need to do is open up a terminal and execute the command to start abiword from within the terminal. Any errors that aren't shown when starting a program from a menu are displayed here. Figure 3 demonstrates this.

Figure 3. Using a terminal to gather information about a malfunctioning program



As you can see in the terminal window I used the following command to open abiword:

```
bash$ abiword &
```

You can make this work with any X program. Just make sure you use the "&" (Ampersand) after the actual programs command name. Then you can just cut and paste the information in the terminal window into a file or email and use it to ask questions.

7.2. plog

plog is used to find problems when setting up or trying to fix your ppp connection. plog shows you the last few lines of `/var/log/ppp.log`. If that file doesn't exist, it shows you the last few lines of `/var/log/syslog` file, but excluding the lines not generated by pppd. It is common to start a terminal and then issue the command:

```
bash$ plog -f
```

Then in another terminal try and start your ppp connection. The -f command causes plog to "force" the file `/var/log/ppp.log` to stay open so as new lines are printed in the file they will appear on the screen. Here's what mine looks like when I shutdown my ppp connection.

You'll probably have to be root to use this program.

```
storm:/home/jesse# plog -f
Apr  8 23:15:14 nexus pppd[22001]: sent [LCP EchoReq id=0x28 magic=0xca942c68]
Apr  8 23:15:15 nexus pppd[22001]: rcvd [LCP EchoRep id=0x28 magic=0x55aae757]
Apr  8 23:15:44 nexus pppd[22001]: sent [LCP EchoReq id=0x29 magic=0xca942c68]
Apr  8 23:15:44 nexus pppd[22001]: rcvd [LCP EchoRep id=0x29 magic=0x55aae757]
Apr  8 23:16:11 nexus pppd[22001]: Terminating on signal 15.
Apr  8 23:16:11 nexus pppd[22001]: Script /etc/ppp/ip-down started (pid 22411)
Apr  8 23:16:11 nexus pppd[22001]: sent [LCP TermReq id=0x2 "User request"]
Apr  8 23:16:12 nexus pppd[22001]: rcvd [LCP TermAck id=0x2]
Apr  8 23:16:12 nexus pppd[22001]: Connection terminated.
Apr  8 23:16:12 nexus pppd[22001]: Connect time 21.0 minutes.
Apr  8 23:16:12 nexus pppd[22001]: Sent 23101 bytes, received 177378 bytes.
Apr  8 23:16:12 nexus pppd[22001]: Hangup (SIGHUP)
Apr  8 23:16:12 nexus pppd[22001]: Waiting for 1 child processes...
Apr  8 23:16:12 nexus pppd[22001]:   script /etc/ppp/ip-down, pid 22411
Apr  8 23:16:12 nexus pppd[22001]: Script /etc/ppp/ip-down finished (pid 22411), status = 0x
Apr  8 23:16:12 nexus pppd[22001]: Exit.
```

If you're having problems with your ppp connection this can really help in finding the answer.

7.3. /var/log/syslog

`/var/log/syslog` works on much the same principal as plog. The only difference is you'll need to use the **tail** command to make it work. Also, as the name implies, this log file contains all the systems logs so there is usually a lot of information in it. Again, open up a terminal and do the tail command. Watch the log file print to screen and try and figure out what's happening that's causing you problems. Here's the command to make it work:

You'll probably have to be root to use this program.

```
bash# tail -f /var/log/syslog
```

7.4. tcpdump

Need.

7.5. dump

Need.

7.6. lspci

lspci is a utility for displaying information about all PCI buses in the system and all devices connected to them. The information this program outputs can sometimes be very helpful to developers who are writing device drivers and modules which make your PCI cards run. PCI cards can be sound cards, video cards, special controllers, and any number of other things. I was once asked for the output of this program when my video card wasn't recognized by the beta version of Progeny. The two most common options are -s and -n. Don't worry if you don't understand the output, you'll be sending this information to someone who (hopefully) does. Here's what they both look like on my machine:

```
jesse@storm:~/test$ /sbin/lspci -n
00:00.0 Class 0600: 8086:122d (rev 02)
00:07.0 Class 0601: 8086:122e (rev 02)
00:07.1 Class 0101: 8086:1230 (rev 02)
00:0e.0 Class 0300: 5333:8811

=========divider========================

jesse@storm:~/test$ /sbin/lspci -v
00:00.0 Host bridge: Intel Corporation 430FX - 82437FX TSC [Triton I] (rev 02)
        Flags: bus master, medium devsel, latency 64

00:07.0 ISA bridge: Intel Corporation 82371FB PIIX ISA [Triton I] (rev 02)
```

```
        Flags: bus master, medium devsel, latency 0

00:07.1 IDE interface: Intel Corporation 82371FB PIIX IDE [Triton I] (rev 02) (prog-if 80 [M
        Flags: bus master, medium devsel, latency 64
        I/O ports at ffa0

00:0e.0 VGA compatible controller: S3 Inc. 86c764/765 [Trio32/64/64V+] (prog-if 00 [VGA])
        Flags: medium devsel, IRQ 11
        Memory at ff000000 (32-bit, non-prefetchable)
```