Template for newbieDoc SGML-format documentation

Jon Eisenstein

jeisen@mindspring.com

An introduction to the very useful **grep** program.

1. Introduction

Some introductory stuff.

2. Main stuff

Eventually you may want to split this up into sections.

Grep. It sounds like a rude noise made by some bodily function. In fact, in some languages, I believe it is just that. Of course, we're talking about the grep command in Linux here, and so we can safely assume that we're referring to the Global Regular Expression Print.

Grep is a powerful tool to search files for some phrase, or, as the name suggests, any expression. Using grep, you could discover exactly which file contains the definition of that word you're looking for, or where you typed in that complex math formula. Simply give grep an expression, a place to look, and it will return any matches, in context.

The basic syntax of grep is as follows:

```
grep [options] expression [file(s)]
```

For example, I could ask grep to find the word "distribution" in Debian's Basic Definitions document:

```
$ grep distribution ch-basic_defs.html
```

Grep would then print on the screen the following lines:

```
Debian GNU/Linux is a particular distribution of the Linux
operating membership or payment required to participate in its  distribution and
value-added Linux distributions can be built.  By providing a reliable,
compatibility, and allows Linux distribution creators to eliminate
...
```

.
.

As you can see, after finding any instance of the word "distribution" in the document, it told me where, in context, I could find that word if I searched myself. It gave me a basic idea of how the word was used each time it was found, so I could decide whether or not each one was the phrase I was searching for.

“Big deal!” you may say. “I can search for words more easily using my text editor!” That’s true, but can you search an entire directory for that word? If you wanted to, could you search your entire system? With grep, you can.

I recently had a problem compiling a kernel (../kernel/compilation.html) to use my relatively ancient sound card. The kernel documentation (located in /usr/src/linux/Documentation if you have installed the source) has an entire directory dedicated to sound card issues. However, none of the files that seemed to be relevant seemed to contain the information I needed. As you may have guessed, I called up my good friend grep and asked for help.

```
$ grep "sound blaster 16" sound/*
```

I used the wildcard (../definitions/wildcard.html) symbol to have grep search every file in the sound directory, but unfortunately came up with nothing. I asked myself what had gone wrong, and I realized that grep is case sensitive (../definitions/case_sensitive), and was being too literal to be useful here. I tried again, this time with the case insensitive option (-i), to see if might have more success.

grep -i "sound blaster 16" sound/*

This time, I was given a list of files in which the phrase "Sound Blaster 16" appeared, and found the answer to my question in a file which I had not thought of to search manually.

I have, in fact, used this method to locate a file which I had written and misplaced. I made sure to search recursively (../definitions/recursive) by using the -r option, and issued the command:

```
$ grep –r "my phrase" /
```

This searched my entire system to find one short phrase. Be warned, though, that if you do try this, searching every file will take a long time, regardless of the speed. Once you see the file that you were looking for and grep continues to look for more, you may use ^C (../process/stopping) to stop grep from searching any more and continue with your work.

There are many more useful things you can do with grep than I can describe here. Hopefully, I have convinced you that it is a tool worth using. I suggest you take a look at the manual page (../documentation/manpages), as well as other resources that might be available.