

Textual help system based on vim

someonesdad1@gmail.com 12 Aug 2012

I describe a reference system I use to index and find textual information quickly. I wanted the following things for my reference system:

1. Minimize coding
2. Use hyperlinks
3. Highlighting by color coding
4. Fast navigation
5. Self-contained
6. Run on many platforms

I'd been mulling this reference system over for a year or two and I was convinced I was going to have to write it in C, with two different console modules, one for Linux and one for 32 bit Windows. I knew I was going to have to spend a lot of time writing it and I wasn't looking forward to it.

One morning about 8 to 10 years ago it struck me that I was already using a good tool for the job, vim, an improved version of the vi editor. I realized that I could adapt vim's help system to meet my needs. And navigation would be easy because I would be using a tool I was already familiar with.

In an hour or so, I had my reference system set up and working. I already had around 20000 lines of reference text already set up. I was viewing that information with hyperlinks after another hour or two of editing and putting in the hyperlink information. Now I could focus on content rather than generating the tool.

This turned out to be such a handy tool that I feel it's worth sharing with other folks.

Some of the advantages of this solution are:

1. There was no C coding needed to make the core tool. (However, I wrote some utilities to help me).
2. Once the hyperlinks are set up (tags), navigation is fast.
3. The tool can do a grep-type search for what I type in on the command line and I'll see all files and locations where there's a match. For example, if I want to see all references to printf, I'll find it under awk, C libraries, gdb, perl, snippets of useful code, the printf(1) command, and some utilities I have on my system. One of the nice features of vim is that it can highlight the result of such searches so you can quickly see what you searched for.
4. My help files are free form. I can put in whatever textual information I want.
5. vim allows you to customize your solution through its many options and scripting capabilities.

There are some disadvantages too:

1. Since I'm using a text editor to display my reference information, I'm limited to, well, text. If you were weaned on fancy GUI tools, then this tool may not be appropriate for you.
2. A GUI tool could probably display more information more densely than is possible in a console window. But the coding effort would be a lot more than this textual solution. You could encode all your information in HTML form and you could view it with a browser, but browser navigation is slow if you're used to using a tool like vi where you don't have to remove your hands from the keyboard.
3. Hyperlinks must be identifiers in the C programming language. This means they can only contain letters, numbers, and the underscore character.

How to use

The zip file that contains this package contains a [mk](#) script. This shell script is used to build the help files whenever I make any changes to the help text. The comments will explain the basic structure and methods. The script indicates it doesn't work on a Linux system, but I haven't worked on a Linux box in a few years, so I haven't fixed it (it should be easy to get running).

The script's output is put into the [h1d](#) directory (I don't remember how I picked that name). Then its copied *en masse* to a directory for tools under my home directory. Obviously, you may not want this behavior, so you're free to change it to a process that works for you.

I've included the textual help files that I've written for myself over the past few decades. It's a somewhat arbitrary mish-mash of stuff and if you plan to use it seriously, I suggest you check the veracity of the information. Some of it I harvested from elsewhere.

To access the help system, I type [h](#) at the command line. I'm presented with the following text, each item of which is a hyperlink to a documentation file:

python	matplotlib	scipy	Korn_Shell	make_	
numpy	mpmath	hg	darcs	Subversion	cvs
C	STL	perl5	perl4	basic	
sympy	Utilities	Snippets	awk	make	
g	find	fld	flex	Cpp	
bash	sed	PCL5	PostScript		
mysql	gdb	HTML			
shop	materials	sizes			
Electrical	Physics	Astronomy	Biology		
Units	Chemistry				
Math	constants	mortgage			
scipy	NSF				
Mathematica					

Some of the links may be broken for you because I didn't bother including the textual help files, as they are too specialized (e.g., the NSF link was for a medical coding format used in the electronic transmission of Medicare information over modems). The perl stuff is probably woefully out of date and incomplete, as I stopped using perl around 1998.

Note: when you run the [mk](#) script to build the help system, you will encounter an error because an environment variable won't exist. Thus, you'll have to modify the script's [CopyToRepository](#) function to make it work for you.

I use this help system with vim 7.3 on a cygwin system running under Windows XP.