Notes for cscope

Author : Vikas Nagpal (https://github.com/vikasnagpaliitd)

Version 1.0

Note: This tutorial uses a small source code directory structure present at path “https://github.com/vikasnagpaliitd/linux-prog-tools/tree/master/ctags/project”

Background

Imagine a big project having 100 .c files, 50 .h files, spread across 20 source/include directories.
Given a function, how do you remember which file implements it?
Given a function, how do you find all places from which it is called?
In short, how do you navigate in the code?

Do you use grep, find, and complex Linux commands each time for this?

There are two main alternatives to manual search - ctags and cscope (There are many other tools too)

Here we discuss cscope.

cscope is basically a utility program, which allows easy search in a code base.

Installing cscope

```
$ sudo apt install cscope
```

Generating the cscope database

cscope stores its data in a file called cscope.out

It rebuilds the database incrementally if some of the files change (determined based on modification timestamp)

Below command will generate cscope.out in the current directory for all .c and .h files in the directory tree rooted at ‘.’

```
$ cscope `find . -name '*.[ch]'`
```

OR

```
$ find . -name '*.[ch]' > cscope.files
$ cscope -i cscope.files
```

If you wish to use a different filename for database, or the cscope.out is at a different path, you can use -f option

```
-f <path to cscope.out>
```

```
$ man cscope # gives lot of good information
```

When you run cscope, a menu like following appears

```
—--------------------
Cscope version 15.9                              Press the ? key for help











Find this C symbol:  // Look for a variable, macro , function, etc.
Find this global definition: // Search where is the symbol defined
Find functions called by this function: // which functions does func() call?
Find functions calling this function: // which all functions do call func()
Find this text string: // Search for a test string
Change this text string: // Search replace in the entire directory tree
Find this egrep pattern: // pattern based search
Find this file: // where is file named xyz.c located
Find files #including this file: // who are the users of my module who include module.h
Find assignments to this symbol:// where all is this symbol assigned to
```

—-----------------------

You can use up/down arrows keys to move across menu items.

Assume you invoked “Find this text string”, and many options have come.
You can navigate through the list using up-key, down-key, space(to display more), enter (to select an option)

If you select an option, that file is opened in the editor.

Or, instead of selecting option, you can press Tab to go to the menu again.

You can press ctrl-d (end of input) to quit cscope

vi with cscope

You can also load the cscope.out in vi, and invoke cscope commands from within vi.
We will not study that for now. But, do check the references to note how it looks like.

References

1. http://cscope.sourceforge.net/
2. Cscope Tutorial
3. Cscope data can integrate with vi. Vim/Cscope tutorial and How to use cscope? - Stack Overflow
4. Browse C code using cscope | Mouhsen Ibrahim Site