

VIM and browser bookmarking

NOVALUG
Doug Toppin
3-Jun-2006

note

a lot of this material was copied from
vim.org, del.icio.us, scuttle.org and
wikipedia.org

VIM

vim

- Vi IMproved
- additions over vi in general
- specific examples
- some useful things to know (that are not necessarily specific to vim)

vi

- Bill Joy wrote this program in 1976 for an early BSD release
- vi is a port of the classic BSD vi 3.7 to modern Unix systems, it uses ed as a codebase, which is BSD-style free since January 2002

additions over vi

- unlimited undo
- works on MS-Windows, Macintosh, Amiga, OS/2, VMS, QNX and other systems. And also on every Unix system.
- syntax highlighting by filetype (100s)
- GUI (uses native GUI widgets for scrolling, dividing buffers, and menuing, can also talk to the clipboard)
- vi compatibility mode

additions over vi

- tags can be preserved across sessions

vim links

- vim.org
- applications.linux.com (search vim)

sort

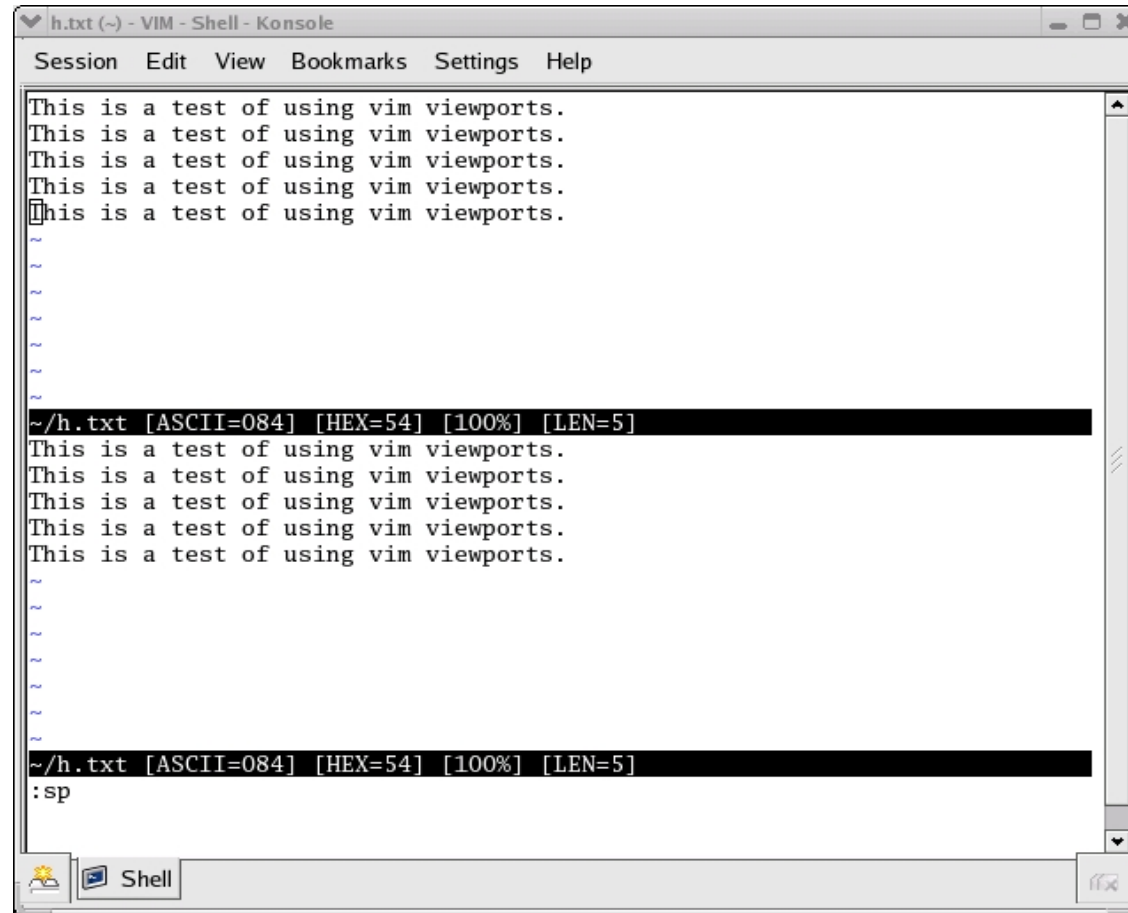
- `:sort` textual sort
- `:sort n` numeric sort

```
:[range]sor[t][!] [i][u][r][n][x][o][/{pattern}/]
```

viewports

- `:sp`
- `control-w control-w`
- `control-w -`, `control-w +`

viewports – after :sp

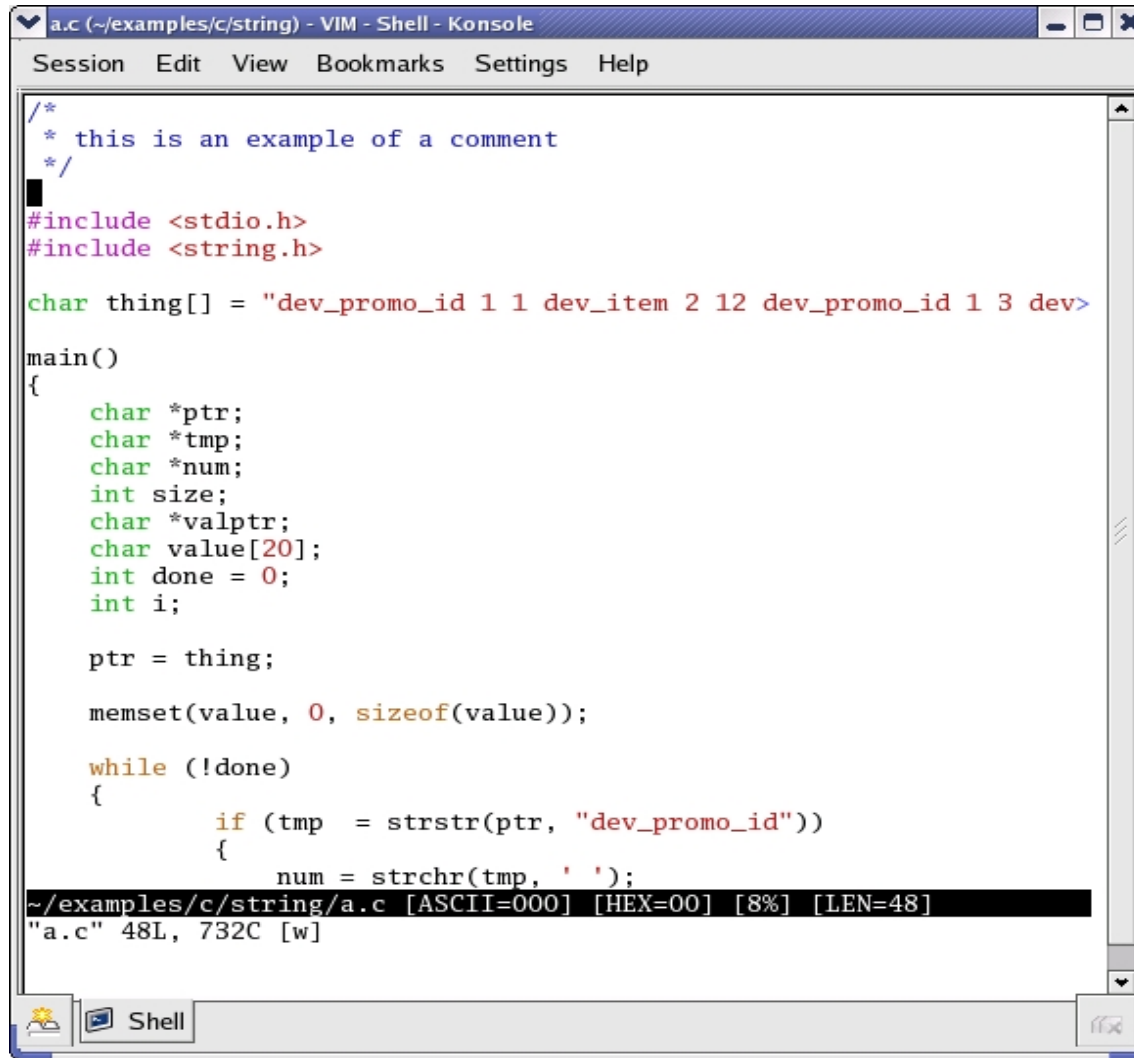


The screenshot shows a terminal window titled "h.txt (~) - VIM - Shell - Konsole". The window contains two viewports. The top viewport shows the first five lines of a file: "This is a test of using vim viewports." followed by four tilde characters. The bottom viewport shows the same five lines, with the first line highlighted in black. The status bar at the bottom of the window displays "~ /h.txt [ASCII=084] [HEX=54] [100%] [LEN=5]" and the command ":sp" is entered in the command line.

```
h.txt (~) - VIM - Shell - Konsole
Session Edit View Bookmarks Settings Help

This is a test of using vim viewports.
This is a test of using vim viewports.
This is a test of using vim viewports.
This is a test of using vim viewports.
This is a test of using vim viewports.
~
~
~
~
~
~
~/h.txt [ASCII=084] [HEX=54] [100%] [LEN=5]
This is a test of using vim viewports.
This is a test of using vim viewports.
This is a test of using vim viewports.
This is a test of using vim viewports.
This is a test of using vim viewports.
~
~
~
~
~
~/h.txt [ASCII=084] [HEX=54] [100%] [LEN=5]
:sp
```


syntax



```
a.c (~/examples/c/string) - VIM - Shell - Konsole
Session Edit View Bookmarks Settings Help

/*
 * this is an example of a comment
 */
#include <stdio.h>
#include <string.h>

char thing[] = "dev_promo_id 1 1 dev_item 2 12 dev_promo_id 1 3 dev>
main()
{
    char *ptr;
    char *tmp;
    char *num;
    int size;
    char *valptr;
    char value[20];
    int done = 0;
    int i;

    ptr = thing;

    memset(value, 0, sizeof(value));

    while (!done)
    {
        if (tmp = strstr(ptr, "dev_promo_id"))
        {
            num = strchr(tmp, ' ');
~/examples/c/string/a.c [ASCII=000] [HEX=00] [8%] [LEN=48]
"a.c" 48L, 732C [w]
```

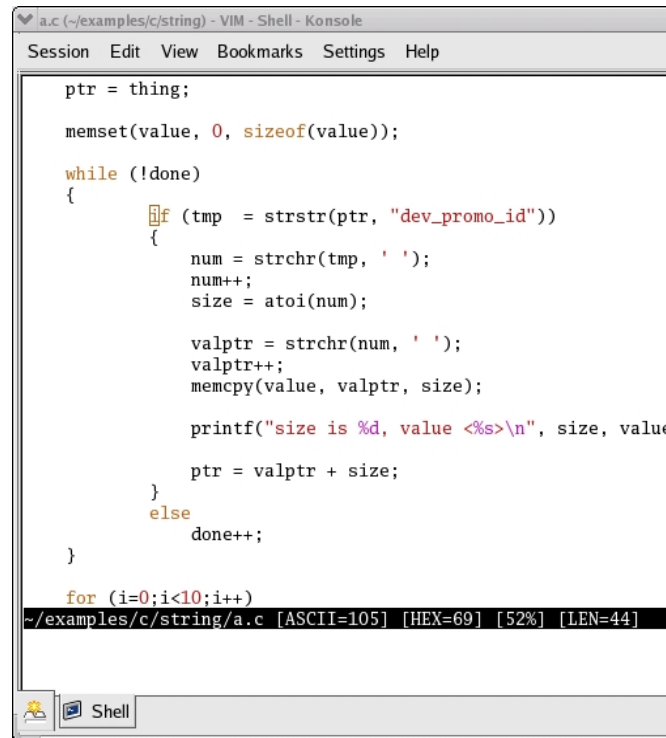
folding

- language recognition
- `:syntax on`
- `:set foldmethod syntax`
- 481 syntax files in `share/vim/vim70/share/`
- `c.vim`, `sh.vim`, `make.vim`, `fetchmail.vim`, `sql.vim`, ...

folding

- `zf/string` fold up to string
- `zk` move to previous fold
- `zj` move to next fold
- `zo` open fold
- `zc` close fold

folding



```
a.c (~/examples/c/string) - VIM - Shell - Konsole
Session Edit View Bookmarks Settings Help

ptr = thing;
memset(value, 0, sizeof(value));
while (!done)
{
    if (tmp = strstr(ptr, "dev_promo_id"))
    {
        num = strchr(tmp, ' ');
        num++;
        size = atoi(num);

        valptr = strchr(num, ' ');
        valptr++;
        memcpy(value, valptr, size);

        printf("size is %d, value <%s>\n", size, value);

        ptr = valptr + size;
    }
    else
        done++;
}

for (i=0;i<10;i++)
~/examples/c/string/a.c [ASCII=105] [HEX=69] [52%] [LEN=44]
```


marks

- `m[a-zA-Z]` set mark at cursor
- `'[a-zA-Z]` move to mark
- marks can be used in commands supporting address ranges
- `'a,'b co .`
- `'a,'b w newfile.txt`

jumps

- `:jump` show jump list (note across files)
- `control-o` `control-i` to move to various jump points

viminfo and vimrc

- `.viminfo` preserves info across sessions (to continue where you left off)
- holds history of commands, searches, file marks, ...
- `.vimrc` initialization commands, each executed as `ex` commands
- `:help viminfo`
- `:help vimrc`

plugins

- plug-ins are macros loaded when vim starts
- `share/vim/vim70/macros/`
- life, hanoi and maze are examples of the power of vim macros

vim (and maybe vi) tips

- reformat a paragraph (by sending the paragraph to the external 'fmt' app)
 !`}fmt` to reformat a text paragraph
- set `so=5` 5-lines top/bottom for scope

bookmarking

social/centralized bookmarking

- store your bookmarks online rather than on a specific browser
- associate tags with pages of interest
- plug-ins for browser buttons to tag current page, view tags
- search for/view tags from others
- see what tags are popular or recent

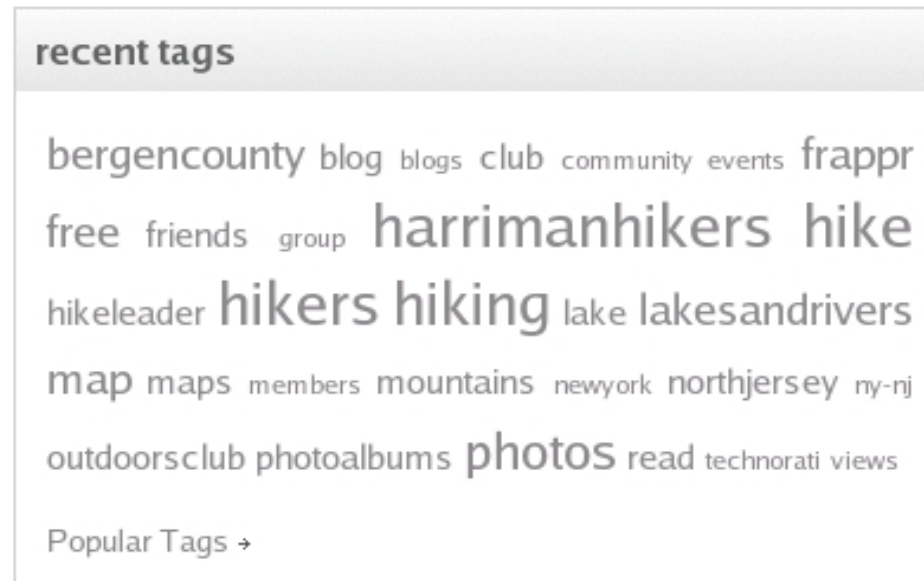
social/centralized bookmarking

- store all your favorite links in one place, accessible from anywhere
- share your bookmarks with everyone, with friends on your watchlist or just keep them private
- tag your bookmarks with as many labels as you want, instead of wrestling with folders

tag cloud

- tag cloud is a visual depiction of content tags used on a website. more frequently used tags are depicted in a larger font or otherwise emphasized, while the displayed order is generally alphabetical. Selecting a single tag within a tag cloud will generally lead to a collection of items that are associated with that tag.
- first widely known tag cloud appeared on Flickr, the photo sharing site

tag cloud



popular sites/tools

- <http://del.icio.us>
- <http://scuttle.org> – open source (run your own server)