

VIM Rocks



Adam Lee
Kernel-QE, Red Hat Software

@adam8157

<http://adam8157.info>

adam8157@gmail.com

```
$ echo -n adam |md5sum |cut -c 29-32  
8157
```

Forked from

VIM Hacks

林佑安 Yo-An Lin

Creative Commons 3.0 License
<http://creativecommons.org/licenses/by/3.0/>



The **worst** way to edit
最**糟糕**的编辑方式

```
#!/usr/bin/env perl  
my $happiness = USTCLIP->join();
```

```
#!/usr/bin/env perl
```

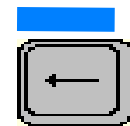
```
my $happiness = USTCLIP->join();_
```

**Oops!**

lost "a"

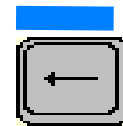
```
#!/usr/bin/env perl
```

```
my $happiness = USTCLIP->join();
```



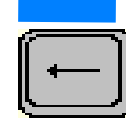

```
#!/usr/bin/env perl
```

```
my $happiness = USTCLIP->join();
```



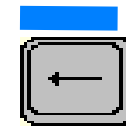
```
#!/usr/bin/env perl
```

```
my $happiness = USTCLIP->join();
```



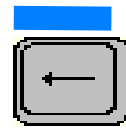
```
#!/usr/bin/env perl
```

```
my $happiness = USTCLIP->join();
```



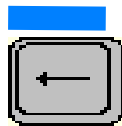
```
#!/usr/bin/env perl
```

```
my $happiness = USTCLIP->join();
```



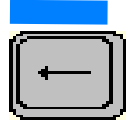
```
#!/usr/bin/env perl
```

```
my $happiness = USTCLIP->join();
```



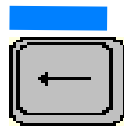
```
#!/usr/bin/env perl
```

```
my $happiness = USTCLIP->join();
```



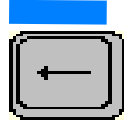
```
#!/usr/bin/env perl
```

```
my $happiness = USTCLIP->join();
```



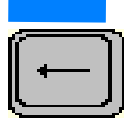
```
#!/usr/bin/env perl
```

```
my $happiness = USTCLIP->join();
```



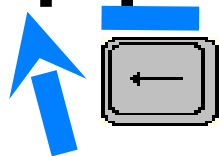

```
#!/usr/bin/env perl
```

```
my $happiness = USTCLIP->join();
```



```
#!/usr/bin/env perl
```

```
my $happiness = USTCLIP->join();
```



```
#!/usr/bin/env perl
```

```
my $hppiness = USTCLIP->join();
```

```
#!/usr/bin/env perl
```

```
my $happiness = USTCLIP->join();
```

The **VIM** way...

VIM:

```
#!/usr/bin/env perl
```

```
my $happiness = USTCLIP->join();
```



VIM:

```
#!/usr/bin/env perl
```

```
my $happiness = USTCLIP->join();
```

 append 

So What is **VIM** ?

VIM

The editor

VI

Improved

Features

1. Mode

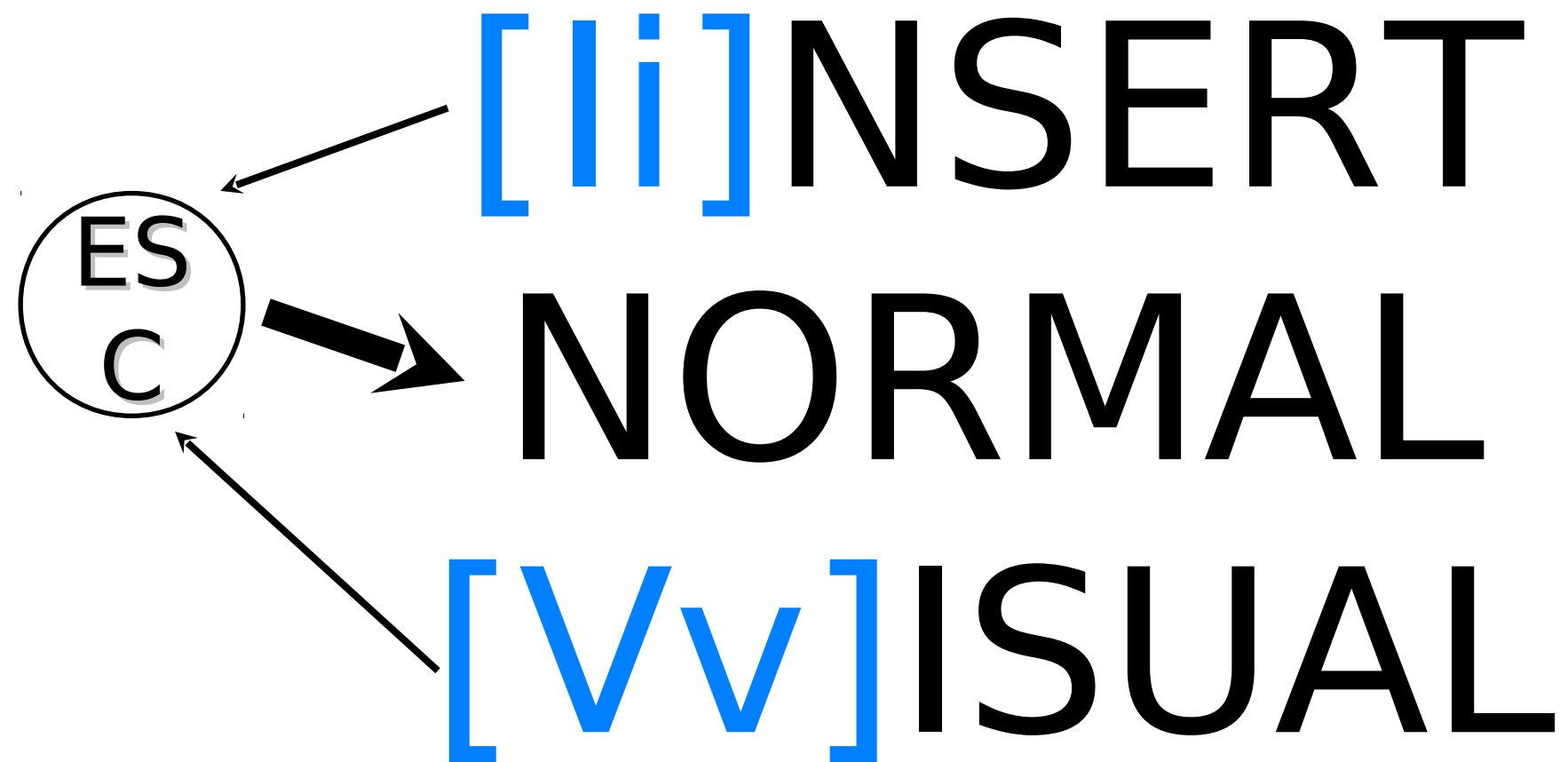
More Than
4 Edit Mode

INSERT

NORMAL

VISUAL

SELECT



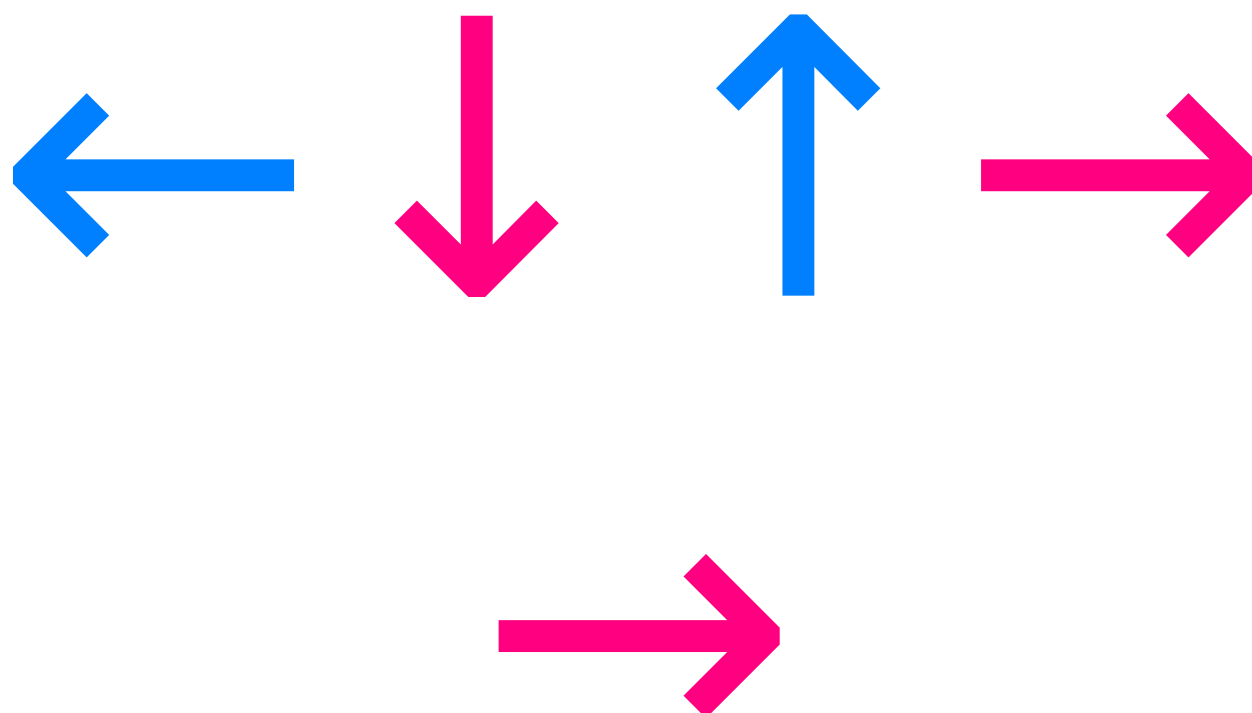
1.1 Normal Mode

Motion





HIKL



h , j , k , l

H , M , L

w , e , b

f[x] , t[x]

[{ ,]} , %

:h motion.txt

1.2 Insert Mode

Editing text

Insert Mode

- i : Insert text before the cursor

Insert Mode

- i : Insert text before the cursor
- I : Insert text before the first non-blank in the line

Insert Mode

- i : Insert text before the cursor
- I : Insert text before the first non-blank in the line
- a : Append text after the cursor

Insert Mode

- i : Insert text before the cursor
- I : Insert text before the first non-blank in the line
- a : Append text after the cursor
- A : Append text at the end of the line

1.3 Visual Mode

Select region

Visual Mode

- v : start Visual mode per character.

Visual Mode

- v : start Visual mode per character.
- V : start Visual mode linewise.

Visual Mode

- `v` : start Visual mode per character.
- `V` : start Visual mode linewise.
- `Ctrl-v` : start Visual mode blockwise.

Visual Mode

- v : start Visual mode per character.
- V : start Visual mode linewise.
- Ctrl-v : start Visual mode blockwise.

operator 如 y (yank) , d
(delete) , c (change) ... etc

2. Syntax Highlight Support

```
$ ls -l /usr/share/vim/vim73/syntax/ | wc -l  
534
```

more than 500 syntax
files

```

18  * async buffer flushing, 1999 Andrea Arcangeli <andrea@suse.de>
19  */
20
21 #include <linux/kernel.h>
22 #include <linux/syscalls.h>
23 #include <linux/fs.h>
24 #include <linux/mm.h>
25 #include <linux/percpu.h>
26 #include <linux/slab.h>
27 #include <linux/capability.h>
28 #include <linux/blkdev.h>
29 #include <linux/file.h>
30 #include <linux/quotaops.h>
31 #include <linux/highmem.h>
32 #include <linux/module.h>
33 #include <linux/writeback.h>
34 #include <linux/hash.h>
35 #include <linux/suspend.h>
36 #include <linux/buffer_head.h>
37 #include <linux/task_io_accounting_ops.h>
38 #include <linux/bio.h>
39 #include <linux/notifier.h>
40 #include <linux/cpu.h>
41 #include <linux/bitops.h>
42 #include <linux/mpage.h>
43 #include <linux/bit_spinlock.h>
44 #include <linux/cleancache.h>
45
46 static int fsync_buffers_list(spinlock_t *lock, struct list_head *list);
47
48 #define BH_ENTRY(list) list_entry((list), struct buffer_head, b_assoc_buffers)
49
50 inline void
51 init_buffer(struct buffer_head *bh, bh_end_io_t *handler, void *private)
52 {
53     bh->b_end_io = handler;
54     bh->b_private = private;
55 }
56 EXPORT_SYMBOL(init_buffer);
57
58 static int sleep_on_buffer(void *word)
59 {
60     io_schedule();
61     return 0;
62 }

```

可定制 Syntax

:help syntax.txt

3. key mapping

:map

:nmap

:vmap

:imap

:smap

:xmap

... more

:map (*all*)

:nmap (*normal mode*)

:vmap (*visual mode*)

:imap (*insert mode*)

:smap (*select mode*)

:xmap (*visual , select mode*)

... more

:h map.txt

4. Text Object

Text Object

- word
 - string
 - paragraph
 - block
- ⇒ action
(yank, delete, change ...etc)

Operator Mapping

v | c | d

i | a

{ | [| (| " | ' |

visual
change
delete

Inner Object
An Object

Region

{ }

[]

()

" "

' '

Operator

va{

```
4 sub func1 {
5     "di" delete character (alias
6     for ( 1 .. 10 ) {
7         "ddw" delete d word
8     }
9     "dow" delete a WORD (see |WOP
10 }
    "dd" delete one line
    "dce" delete inner contents
```

```
0V 4 sub func1 {
5     "dl" delete character (a
6     for ( 1 delete 10 ) nge
7     "daw" delete a word
8     "dLW" delete inner WORD (
9     "daw" delete a WORD (see
10    "dd" delete one line
    "dis" delete inner senten
```



sub(blah , blah)

ci(

sub()



sub(new_args)

“Hello World”

di”

“ ”



“Hello New World”

5. FOLDS

FOLD IS

```
42 #include <linux/mpage.h>
43 #include <linux/bit_spinlock.h>
44 #include <linux/cleancache.h>
45
46 static int fsync_buffers_list(spinlock_t *lock, struct list_head *list);
47
48 #define BH_ENTRY(list) list_entry((list), struct buffer_head, b_assoc_buffers)
49
50 inline void
51 init_buffer(struct buffer_head *bh, bh_end_io_t *handler, void *private)
52 +-- 4 lines: {-----
56 EXPORT_SYMBOL(init_buffer);
57
58 static int sleep_on_buffer(void *word)
59 +-- 4 lines: {-----
63
64 void __lock_buffer(struct buffer_head *bh)
65 +-- 4 lines: {-----
69 EXPORT_SYMBOL(__lock_buffer);
70
```

Fold Methods

Fold Methods

Syntax Fold

`:set foldmethod=syntax`

set fold method as syntax , check out more
options in:

`$VIMRUNTIME/syntax/*.vim`

Fold Methods

Syntax Fold
Marker Fold

:set foldmethod=marker

fold region by markers ,
the default marker is
“{{{“ , “}}”


```
# fold this {{{  
sub do_something {
```

```
    # a lot of work ...
```

```
}
```

```
# }}}}
```

```
# fold this {{{
sub do_something {

    # a lot of work ...
    # foldlevel 2 {{{2
        # foldlevel 4 {{{4

        # }}}4
        # foldlevel here is 3
    # }}}2
}
# }}}}
```

Fold Methods

Syntax Fold
Marker Fold
Indent Fold

:set foldmethod=indent

use indent to fold

Fold Methods

Syntax Fold

Marker Fold

Indent Fold

Manual Fold

`:set foldmethod=manual`

create folds
manually

:h folding

6. QuickFix

:grep

```
:grep [pattern] [filepath]
```


Result \Rightarrow QuickFix
Window

```

59
60     /* Don't overwrite an existing error. This preserves the first
61     * error, which is the most significant. */
62     _cairo_status_set_error (&font_face->status, status);
63
64     return _cairo_error (status);
65 }
66
67 void
68 _cairo_font_face_init (cairo_font_face_t *font_face,
69                       const cairo_font_face_backend_t *backend)
70 {
71     CAIRO_MUTEX_INITIALIZE ();
72
73     font_face->status = CAIRO_STATUS_SUCCESS;
74     CAIRO_REFERENCE_COUNT_INIT (&font_face->ref_count, 1);
75     font_face->backend = backend;

```

cairo-font-face.c

59,0-1

```

5 cairo-font-face.c|69| const cairo_font_face_backend_t *backend)
6 cairo-font-face.c|81| * cairo_font_face_reference:
7 cairo-font-face.c|82| * @font_face: a #cairo_font_face_t, (may be %NULL in which case this
8 cairo-font-face.c|87| * cairo_font_face_destroy() is made.
9 cairo-font-face.c|89| * The number of references to a #cairo_font_face_t can be get using
10 cairo-font-face.c|90| * cairo_font_face_get_reference_count().
11 cairo-font-face.c|92| * Return value: the referenced #cairo_font_face_t.
12 cairo-font-face.c|94| cairo_font_face_t *
13 cairo-font-face.c|95| cairo_font_face_reference (cairo_font_face_t *font_face)
14 cairo-font-face.c|109| slim_hidden_def (cairo_font_face_reference);

```

[Quickfix List]

10,1

QuickFix Window

```
:set grepprg=/path/to/grep
```

:copen

:cclose

:cnext

:cprevious

7. Helpful Settings

" Set fileencodings

set fileencodings=ucs-bom,utf-8,gbk,big5

" Keep more backups for one file

```
autocmd BufWritePre * let &backupext = strftime(".%m-%d-%H-%M")
```

" Save & Make

```
nnoremap <F5> :w<CR>:make!<CR>
```

```
nnoremap <F6> :w<CR>:make! %< CC=gcc CFLAGS="-Wall -g -O2"<CR>:!./%<<CR>
```

8. Plugins

supertab.vim

Do all your insert-mode
completion with Tab!

The_NERD_TREE.vim

Directory Tree Explorer

The NERD Commenter

Comment Helper

taglist.vim

Use with ctags
display macro , function , variable

4 buffer.c /u/i/l/fs.h

.. (up a dir)

/home/adam/Projects/linux/fs/

+9p/

+adfs/

+affs/

+afs/

+autofs4/

+befs/

+bfs/

+btrfs/

+cachefiles/

+ceph/

+cifs/

+coda/

+configfs/

+cramfs/

+debugfs/

+devpts/

+dlm/

+ecryptfs/

+efs/

+exofs/

+exportfs/

+ext2/

+ext3/

+ext4/

+fat/

+freevxfs/

+fscache/

+fuse/

+gfs2/

+hfs/

+hfsplus/

+hostfs/

+hpfs/

+hppfs/

+hugetlbfs/

+isofs/

+jbd/

+jbd2/

+jffs2/

+jfs/

+lockd/

22 #include <linux/syscalls.h>

23 #include <linux/fs.h>

24 #include <linux/mm.h>

25 #include <linux/percpu.h>

26 #include <linux/slab.h>

27 #include <linux/capability.h>

28 #include <linux/blkdev.h>

29 #include <linux/file.h>

30 #include <linux/quotaops.h>

31 #include <linux/highmem.h>

32 #include <linux/module.h>

33 #include <linux/writeback.h>

34 #include <linux/hash.h>

35 #include <linux/suspend.h>

36 #include <linux/buffer_head.h>

37 #include <linux/task_io_accounting_ops.h>

38 #include <linux/bio.h>

39 #include <linux/notifier.h>

40 #include <linux/cpu.h>

41 #include <linux/bitops.h>

42 #include <linux/mpage.h>

43 #include <linux/bit_spinlock.h>

44 #include <linux/cleancache.h>

45

46 static int fsync_buffers_list(spinlock_t *lock, struct list_head *list);

47

48 #define BH_ENTRY(list) list_entry((list), struct buffer_head, b_assoc_buffers)

49

50 inline void

51 init_buffer(struct buffer_head *bh, bh_end_io_t *handler, void *private)

52 {

53 bh->b_end_io = handler;

54 bh->b_private = private;

55 }

56 EXPORT_SYMBOL(init_buffer);

57

58 static int sleep_on_buffer(void *word)

59 {

60 io_schedule();

61 return 0;

62 }

63

64 void __lock_buffer(struct buffer_head *bh)

65 {

cont_write_begin

create_empty_buffers

discard_buffer

do_thaw_all

do_thaw_one

drop_buffers

emergency_thaw_all

end_bio_bh_io_sync

end_buffer_async_read

end_buffer_async_write

end_buffer_read_nobh

end_buffer_read_sync

end_buffer_write_sync

free_buffer_head

free_more_memory

fsync_buffers_list

generic_block_bmap

generic_cont_expand_simple

generic_write_end

grow_buffers

grow_dev_page

init_buffer

init_page_buffers

inode_has_buffers

invalidate_bdev

invalidate_bh_lru

invalidate_bh_lrus

invalidate_inode_buffers

link_dev_buffers

ll_rw_block

lookup_bh_lru

mark_buffer_async_read

mark_buffer_async_write

mark_buffer_async_write_end

mark_buffer_dirty

mark_buffer_dirty_inode

nobh_truncate_page

nobh_write_begin

nobh_write_end

nobh_writepage

osync_buffers_list

page_zero_new_buffers

quiet_error

recalc_bh_state

/home/adam/Projects/linux/fs buffer.c 51,9 0% Tag_List 144,5 91%

1 buffer.c[56] <<global>> EXPORT_SYMBOL(init_buffer);

2 /home/adam/Projects/linux/include/linux/buffer_head.h[144] <<global>> void init_buffer(struct buffer_head *, bh_end_io_t *, void *);

3 /home/adam/Projects/linux/drivers/media/video/cx23885/cx23885-video.c[649] <<buffer_prepare>> int rc, init_buffer = 0;

4 /home/adam/Projects/linux/drivers/media/video/cx23885/cx23885-video.c[669] <<buffer_prepare>> init_buffer = 1;

5 /home/adam/Projects/linux/drivers/media/video/cx23885/cx23885-video.c[673] <<buffer_prepare>> init_buffer = 1;

6 /home/adam/Projects/linux/drivers/media/video/cx23885/cx23885-video.c[679] <<buffer_prepare>> if (!init_buffer) {

7 /home/adam/Projects/linux/drivers/media/video/cx25821/cx25821-video.c[562] <<cx25821_buffer_prepare>> int rc, init_buffer = 0;

8 /home/adam/Projects/linux/drivers/media/video/cx25821/cx25821-video.c[585] <<cx25821_buffer_prepare>> init_buffer = 1;

9 /home/adam/Projects/linux/drivers/media/video/cx25821/cx25821-video.c[589] <<cx25821_buffer_prepare>> init_buffer = 1;

10 /home/adam/Projects/linux/drivers/media/video/cx25821/cx25821-video.c[597] <<cx25821_buffer_prepare>> dprintk(1, "init_buffer=%d\n", init_buffer);

[Quickfix List] :scs find s init buffer

:q

1 2 3 4 5 6 7 8 9 buffer.c (~/Projects/linux/fs) - VIM

Mon Dec 05, 13:30

vimrc

<https://github.com/adam8157>

Thank You

Q & A