

Emacs Features

Claude TETE

mai 11, 2012

Features already in Emacs

- ▶ Presentation of several features which are not always present in other text editor or IDE

Split Screen

- ▶ split current frame as often as you want with horizontally “Ctrl+x 2”, vertically “Ctrl+x 3”, undo all split “Ctrl+x 1”
- ▶ increase/decrease size of frame (window for Emacs) with “Alt+x enlarge-window, enlarge-window-horizontally, shrink-window or shrink-window-horizontally”

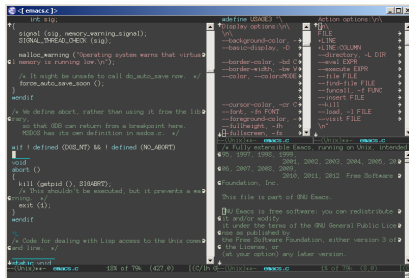


Figure: Split in four windows

No Tab

- ▶ Emacs do not use tab but history of opened files (buffers for Emacs)
- ▶ “Ctrl+x” “Ctrl+b” (for electric-buffer-list) or ECB history or “Ctrl+Left Click” to open context menu

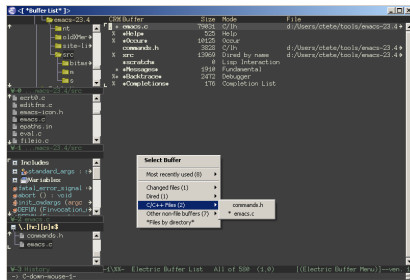


Figure: ECB History (bottom-left), Buffer List (top-right) and Context Menu

Move by paragraph/word/expression

- ▶ move from word to word with “Ctrl+LEFT/RIGHT”
- ▶ move from empty line to empty line (between paragraph/block) with “Ctrl+UP/DOWN”
- ▶ move from function beginning to ending with “Ctrl+Shift+HOME/END”
- ▶ move to preprocessing beginning with “Ctrl+c Ctrl+u”
- ▶ switch to *move from word to word* to *move from subword to subword* (place at CapitalLetterInThisWord) with “Ctrl+c Ctrl+w”
- ▶ and more about parentheses/bracket or source code block or sentence

Highlight current line

```

1  #include <unistd.h>
2
3  int sig;
4
5  signal (sig, handler_victim.signal);
6  SIGNAL_THREAD_CHECK (sig);
7
8  malloc_warning ("Operating system warns that virtual memory is running low.\n");
9
10 /* It might be unsafe to call do_auto_save now. */
11 force_auto_save_soon ();
12
13 void if
14
15 /* We define abort, rather than using it from the library,
16    so that GDB can return from a breakpoint here.
17    REDSOB has its own definition in redso.c. */
18
19 #if ! defined (DOS_NT) && ! defined (NO_ABORT)
20
21 void
22 abort ()
23 {
24     kill (getpid (), SIGABRT);
25     /* This shouldn't be executed, but it prevents a warning. */
26     exit (1);
27 }
28
29 void if
30
31 /* Code for dealing with Lisp access to the Unix cword line. */
32
33 static void
34 init_cword (argc, argv, skip_argc)
35     int argc;
36     char **argv;
37 {
38     UnixSys = emacs.c
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998

```

Figure: The highlight current line in center

Delete by word/block

- ▶ remove all empty line except one “Ctrl+x Ctrl+o”
- ▶ remove all space except one “Alt+SPACE”
- ▶ delete word left “Alt+BACKSPACE” or “Ctrl+BACKSPACE”
- ▶ delete word right “Alt+d”

Add/Remove parentheses

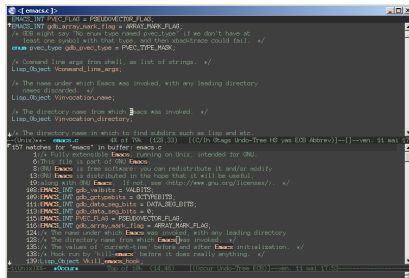
- ▶ around a selection with parentheses with “Alt+ (“
- ▶ remove a pair of parentheses with “Alt+x delete-pair ENTER”

Incremental Search

- ▶ start an forward incremental search with “Ctrl+s”, regex with “Ctrl+Alt+s”
- ▶ backward incremental search with “Ctrl+r”, regex with “Ctrl+Alt+r”
- ▶ last searched in forward with “Ctrl+s Ctrl+s” same for backward “Ctrl+r Ctrl+r”, search in history “Ctrl+s ENTER Alt+r”, etc
- ▶ during incremental search, select to the end of the word with “Ctrl+w”, paste with “Alt+y”, case sensitive with “Alt+c”, start query replace with “Alt+%”
- ▶ the incremental search highlight all matches in the buffer
- ▶ any keys to move will close incremental search
- ▶ incremental search can be very useful to move in a buffer, even during selection

Occurrences

- ▶ search each occurrences of a regex with “Alt+s o”
- ▶ you can navigate to the next or previous match with “Alt+g Alt+n” and “Alt+g Alt+p” (if you want use them I recommend to rebind them like “F3” “Shift+F3”)
- ▶ to refresh with “g” when in * Occur* buffer



```
@ [ emacs.c ]-
EMACS_INT PREC_FLAG = PRECOMVECTOR_FLAG;
EMACS_INT GLOBAL_PREC_FLAG = GLOBAL_PREC_FLAG;
/* OS might say "no such type named exec_type" if we don't have an
exec_type symbol with that "type" and then libstdc++ could fail. */
#define exec_type gbl_exec_type + PREC_TYPE_MARK;

/* Command line args from shell, as list of strings. */
Lisp_Object Vcommand_line_args;

/* The name under which Emacs was invoked, with any leading directory
names discarded. */
Lisp_Object Vinvocation_name;

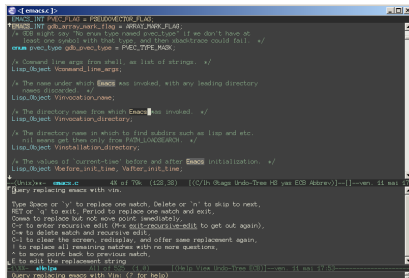
/* The directory name from which Emacs was invoked. */
Lisp_Object Vinvocation_directory;

/* The directory name in which to find subdirs such as lisp and etc.
(Units-- Emacs.c 43 of 78 (129.2) [C:/In-Charge Undo-Tree ID yes EOB Abbrev]---)---en. 11 was 18
157 matches for "emacs" in buffer: emacs.c
0 This file is part of GNU Emacs.
8 GNU Emacs is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.
100 GNU Emacs is distributed in the hope that it will be useful,
101 but WITHOUT ANY WARRANTY; without even the implied warranty of
102 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
103 GNU General Public License for more details.
104 You should have received a copy of the GNU General Public License
105 along with GNU Emacs.  If not, see <http://www.gnu.org/licenses/>.  */
106
107 EMACS_INT gbl_wildbits = WILDBITS;
108 EMACS_INT gbl_getenvbits = GETENVBITS;
110 EMACS_INT gbl_data_squbits = DATA_SQUBITS;
111 EMACS_INT gbl_data_squbits = DATA_SQUBITS;
112 EMACS_INT PREC_FLAG = PRECOMVECTOR_FLAG;
113 EMACS_INT gbl_array_mark_flag = ARRAY_MARK_FLAG;
114
115 /* The name under which Emacs was invoked, with any leading directory
116 names discarded. */
117 /* The directory name from which Emacs was invoked. */
118 /* The value of current-time before and after Emacs initialization. */
119 /* How run by kill Emacs before it does really anything. */
120 Lisp_Object Vlll Emacs_hook;
121
122 [129.21] *Occur* 11 of 18 (129.21) [C:/In-Charge Undo-Tree ID yes EOB Abbrev]---)---en. 11 was 18
```

Figure: Search occurrences of "emacs"

Replace query

- ▶ query replace without regex with “Alt+%”
- ▶ query replace with regex with “Ctrl+Alt+%”



```
emacs.c:
#define PVEC_FLAGS = PVEC_FLAGS
#define PVEC_ARRAY_MARK_FLAG = PVEC_ARRAY_MARK_FLAG
/* ICS might say "No enum type named pvec_type" if we don't have it
 * but one symbol with that type, and then backtrace could fail. */
#define pvec_type glib_pvec_type = PVEC_TYPE_MASK,

/* Command line args from shell, as list of strings. */
Lisp_Object Vcommand_line_args;

/* The name under which Emacs was invoked, with any leading directory
 * names discarded. */
Lisp_Object Vinvocation_name;

/* The directory name from which Emacs was invoked. */
Lisp_Object Vinvocation_directory;

/* The directory name in which to find subdirs such as lisp and etc.
 * nil means get this only from PATH_C:\WINDOWS\ */
Lisp_Object Vinstallation_directory;

/* The values of "current-time" before and after Emacs initialization. */
Lisp_Object Vbefore_init_time, Vafter_init_time;

(Unix)xx- emacs.c 4K of 70K (128.38) [(C) In Emacs Undo-Tree H8 yes EDS Abbrev]--|---vim: it was 17
Query replacing emacs with vim.

Type Space or 'y' to replace one match, Delete or 'n' to skip to next,
RET or 'q' to exit. Period to replace one match and exit,
Comma to replace but not move point immediately,
C-r to enter recursive edit (the quit-recurse=edit to get out again),
C-d to delete match and recursive edit,
C-l to clear the screen, redisplay, and offer same replacement again,
* to replace all remaining matches with no more questions,
* to move point back to previous match,
E to edit the replacement string.

vim: --vim: it was 17
Query replacing emacs with Vim: (? for help)
```

Figure: Query replacing "emacs" with "vim"

Replace on a selected region

- ▶ replace without regex with “Alt+x replace-string ENTER”
- ▶ replace with regex with “Alt+x replace-regexp ENTER”
- ▶ it replace only in the selected region and do not interact with rest of the buffer

Alignment

- ▶ align declaration, parameters of functions with “Alt+x align ENTER” (following the current languages)
- ▶ align anything with regex with “Alt+x align-regex ENTER”

```
{
  register int i;
  Lisp_Object name, dir, tem;
  int count = SPEC_PDL_INDEX ();
  Lisp_Object raw_name;
}
initial_argv = argv;
initial_argc = argc;

raw_name = build_string (argv[0]);
```

Figure: Before

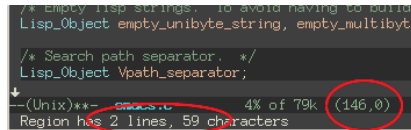
```
{
  register int i;
  Lisp_Object name, dir, tem;
  int count = SPEC_PDL_INDEX ();
  Lisp_Object raw_name;
}
initial_argv = argv;
initial_argc = argc;

raw_name = build_string (argv[0]);
```

Figure: After align declarations

Information

- ▶ Count number of character and line in a selected region with “Alt+ =”
- ▶ Information about a character with “Ctrl+x =”
- ▶ line and column number in status bar of current buffer (modeline in Emacs)



The screenshot shows the Emacs status bar at the bottom of the window. It contains the following text: `-(Unix)**- emacs.c 4% of 79k (146,0)`. Below this, a line of text reads `Region has 2 lines, 59 characters`. Two red circles are drawn on the image: one around the text `Region has 2 lines, 59 characters` and another around the coordinates `(146,0)` in the status bar.

Figure: Information about a selected region and line and column number of cursor

undo

advanced clipboard (kill ring)

- ▶ “Ctrl+k” to cut rest of line

rectangle

cua selection

macro

minor column mode

buffer point register

integrated explorer

bookmarks

command completion/search... (minibuffer)

highlight

indent (line/region following languages and settings)

folding (hide/show block)

version control integration (csv/svn by default)

calculator (on region/rectangle, hextodec. . .)

help (not welcoming)

help about a unknown shortcut/function. . .

compile

shortcuts with Windows and Menu keys

full customization (font/decoration/color/new
function/shortcuts)

settings and customization in simple text file (lisp code)

completion (dumb)

calendar

transparency

parentheses match

mouse avoidance

word wrap (truncate line to show all line but on multiple line)

backup file

dictionnary

up/down case word/region

see only one selected region without interact with rest
(narrow)

acceleration mouse scroll

remove all useless space (trailing space)

history in command (like shell)

fill (justify plain text)

find file (recursively with regex)

Mode

cedet/semantic/completion (smart for oriented object)/ede
(manage project)

ecb (quake like window but bottom for
compile/grep/custom...) source list/ directory list/ method
list (with filter and group)

gtags (search file in project)

which function

dired+ / dired sort (sort like in MS Windows Explorer +
open with associated application)

packages repository

home end

syntax coloration for every languages (just search on internet)

highlight current symbol

google calendar

column indicator

produce document, wiki, pdf, presentation... (muse mode)

show undo like a tree (undo tree mode)

org mode

parentheses match not visible

color theme

Function

memory ring for tags with semantic

check missing mode

incremental search at point

My Function

manage setting profile (my configuration)

select word at point

occur at point

toggle width of ecb window

search tab

graphical clearcase integration

new theme color

better integration in MS Windows

Customize

ignore insert key

never dialog box

never tooltips

add filesize, date and time (modeline)

french calendar

filter/group ecb windows (source/directory/method...)