

Recent Developments

Bram Moolenaar SFI Krakow April 2018

Vim popularity



Vim popularity



Vim code size



src/*.[ch]

One has only limited time available to write code. What to work on next to be most impactful?



One has only limited time available to write code. What to work on next to be most impactful?

1. Work on a feature for 2-3 months tops

One has only limited time available to write code. What to work on next to be most impactful?

- 1. Work on a feature for 2-3 months tops
- 2. Build one feature on top of another one

One has only limited time available to write code. What to work on next to be most impactful?

- 1. Work on a feature for 2-3 months tops
- 2. Build one feature on top of another one
- 3. Understand the user

One has only limited time available to write code. What to work on next to be most impactful?

- 1. Work on a feature for 2-3 months tops
- 2. Build one feature on top of another one
- 3. Understand the user

So, how did this work for Vim?

Vim first releases

1991: Vim 1.14 distributed on Fish disk

Impactful changes in Vim 1.27:

- Port to Unix
- Port to MS-DOS

More supported systems == more users



Vim 2.0

Impactful changes:

• :make and error parsing Please users: Efficient edit - build - fix cycles

Vim 2.0

Impactful changes:

• :make and error parsing Please users: Efficient edit - build - fix cycles

• Vi compatibility

Please distributors (Vim on every Linux and Mac system)



Vim 3.0

Impactful changes:

Multiple windows and buffers
 Make use of larger screens and more memory

Vim 3.0

Impactful changes:

Multiple windows and buffers
 Make use of larger screens and more memory

• Swap file Reliability, user trust



Vim 4.0

Impactful changes:

• Help in a window

More complexity requires more support

Vim 4.0

Impactful changes:

• Help in a window More complexity requires more support

• Autocommands

Extensibility, you can't build everything yourself

Vim 4.0

Impactful changes:

• Help in a window More complexity requires more support

Autocommands
 Extensibility, you can't build everything yourself

• MS-Windows port Reality: most users are there



Vim 5.0

Impactful changes:

• Syntax highlighting Making use of faster computers and better screens

Vim 5.0

Impactful changes:

• Syntax highlighting Making use of faster computers and better screens

• Vim script Extensibility, you can't build everything yourself



Vim 6.0

Impactful changes:

• Unicode support ASCII is no longer the standard

Vim 6.0

Impactful changes:

• Unicode support ASCII is no longer the standard

• Automatic indenting Users are lazy

Vim 6.0

Impactful changes:

• Unicode support ASCII is no longer the standard

• Automatic indenting Users are lazy

• Plugins

Extensibility, you can't build everything yourself



Vim 7.x

Impactful changes:

• Persistent undo

Don't worry, you can go back in time (and back to the future)



Impactful changes:

• Jobs, Channels and Timers

Extensibility, you can't build everything yourself Make use of more powerful computers



Impactful changes?



Impactful changes:

Terminal window Why?



Impactful changes:

Terminal window
 Why?

I want to debug Vim over ssh, which requires:

- 1. Window running gdb
- 2. Window running program being debugged
- 3. Window(s) to edit source code

new-ui mi /dev/pts/15	Step Next Finish Cont Stop Eval
Reading symbols from vimdone.	* ":help": open a read-only window on a help file
(qdb) new-ui mi /dev/pts/15	*/
New UI allocated	void
(gdb) break ex help	ex help(exarg T *eap)
Breakpoint 1 at 0x99848: file ex cmds.c. line 6194.	1
(adh) run	char u *arg:
Starting program: /home/mool/vim/vim80/src/vim	char u *tag
[Thread debugging using libthread dh enabled]	ETLE *helnfd: /* file descriptor of heln file */
Using best libthroad dh library "/lib/y86 64-linuy-gnu/li	htbroad dh co int n'
41 All All All All All All All All All Al	int int int
1.	Line Ly
Prestraint 4 or help (con-0x7ffffffffd4f0) at av amda av6	win_r wp;
Breakpoint 1, ex_neip (eap=0x/iiiiiiiu4i0) at ex_cmus.c.o	194 Inc num_macches;
6194 {	Char_u th
(gdb)	char_u *p; eap: 0x/TTTTTTTd4T0 = {
	int emp arg = 0x5555550cb303 "gu1",
	int alt nextcmd = 0x0,
	buf_T *bu cmd = 0x555555dcb300 "he gui",
	#ifdef FEAT_MULTI_L cmdlinep = 0x7fffffffd710,
CARLES THE PLAN AND A DECEMBER OF A	int len cmdidx = CMD_help,
!gdb [running] 0,0-	1 All char_u *la argt = 2054,
11 2003 20 10 10 00 00 00 00 00 00 00 00 00 00 00	#endif skip = 0,
	#ifdef FEAT_FOLDING forceit = 0,
~	int old addr_count = 0,
	#endif line1 = 1,
VIM - Vi IMproved	line2 = 1,
Marcari Bardari Bardari I	if (eap $!=$ NULL addr type = 0,
version 8.0.1615	f = 0,
 by Bram Moolenaar et al. 	do ecmd cmd = 0×0 .
Vim is open source and freely distributable	* A "thelp do ecmd lnum = 0. at a 'l' that is
	<pre>* followed append = 0. following comman</pre>
Sponsor Vim development!	d usefilter = 0.
type 'help sponsor <enter> for information</enter>	*/
cype meth shouse encors for the macter	for (arg = ean-barg; *arg; ++arg)
type :acEnters to exit	r (aly - cap-raly, aly, raly)
type . yeller or . [1. for on line hel	if (ford 1) of 11 ford 1) rt
two thelp version9 <enters for="" inf<="" th="" version=""><th>$\frac{1}{1} \left(\frac{dy}{dy} - \frac{1}{1} \right) \left[\frac{dy}{dy} - \frac{1}{1} \right] = \frac{1}{1} \left[\frac{dy}{dy} - \frac{1}{1} \right]$</th></enters>	$\frac{1}{1} \left(\frac{dy}{dy} - \frac{1}{1} \right) \left[\frac{dy}{dy} - \frac{1}{1} \right] = \frac{1}{1} \left[\frac{dy}{dy} - \frac{1}{1} \right]$
cype :netb versions/circles for version in	$ (arg = - \alpha \alpha arg[\mathbf{I}] := NUL \alpha \alpha arg[\mathbf{I}] := -)$
ene gui	*arg++ = NUL;
	eap->nextcmd = arg;
gdb program [[No Name] - VIM] 0,0-	1 All ~/vim/vim80/src/ex_cmds.c 6223,6-13 73%
:Over	

Parts needed:

- 1. Terminal emulator: libvterm
- 2. Jobs and channels
- 3. Window toolbar for Step/Next/Continue...
- 4. Balloon to show variable values
- 5. Popup menu



How the parts are put together:





Debugger demo

new-ui mi /dev/pts/16	Step Next Finish Cont Stop Eval
Reading symbols from vimdone.	Read of Readers Readers Readers I Readers II Readers II
(gdb) new-ui mi /dev/pts/16	
New UI allocated	
(gdb)	
lgdb [running]	
gdb program [active] 0,0-1	All [No Name] 0,0-1 All

new-ui mi /dev/pts/17	Step Next Finish Cont Stop Eval
Reading symbols from vimdone.	return FALSE;
(gdb) new-ui mi /dev/pts/17	}
New UI allocated	Hondif
(gub)	#eligit
	/*
	* ":help": open a read-only window on a help file
	*/
	void
	ex_help(exarg_T *eap)
	{
	char_u *arg;
	char_u *tag;
	FILE *helpfd; /* file descriptor of help file */
	int n;
	int 1;
	Win_1 "Wp; Undo matchos:
ladh [running]	1 All atches:
rgub [ruming]	Paste
	ty fnum = 0:
	Select Word fnum = 0;
	Select Sentence f;
	# Select Paragraph ANG
	Select Line ;
	Select Block ng;
	# Select All
	#
	Set breakpoint _KeyTyped = KeyTyped;
	# Clear breakpoint
	Evaluate
	1 /*
	* A ":help" command ends at the first LE, or at a 'l' that is
	* followed by some text. Set nextcmd to the following command.
	*/
	for (arg = eap->arg; *arg; ++arg)
	{
gdb program [active] 0,0-:	1 All ~/vim/vim80/src/ex_cmds.c 6205,1 73%

new-ui mi /dev/pts/17	Step Next Finish Cont Stop Eval
Reading symbols from vimdone.	return FALSE;
(gdb) new-ui mi /dev/pts/17	}
New UI allocated	
(gdb) run	#endif
Starting program: /home/mool/vim/vim80/src/vim	
[Thread debugging using libthread_db enabled]	
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.	/*
so.1".	* ":help": open a read-only window on a help file
	*/
Breakpoint 1, ex_help (eap=0x7fffffffd4f0) at ex_cmds.c:6215	void
6215 if (eap != NULL)	ex_help(exarg_T *eap)
(gub)	t than the targe
	char u *tag:
	TIE *helpfd: /* file descriptor of help file */
	int n'
	int i
	win T *wn
	int num matches:
ladh [running] 0.0-1 All	char u **matches:
	char u *n'
	$\rho_{int} = \rho_{int}$
	int alt frum = 0
	buf T *buf
VTM - Vi TMproved	Hifdef EFAT MULTI LANG
	int len
version 8.0.1615	char u *lang:
 by Bram Moolenaar et al. 	#endif
Vim is open source and freely distributable	#ifdef FFAT FOLDING
ville is open bourde and freezy discribitions	int old KevTyped = KevTyped:
Become a registered Vim user!	#endif
type :help register <enter> for information</enter>	
sipe insep isgester and a sub-	>> if (eap != NULL)
type :g <enter> to exit</enter>	{
type :help <enter> or <f1> for on-line help</f1></enter>	
type :help version8 <enter> for version info</enter>	* A ":help" command ends at the first LF, or at a 'l' that is
- Pre-	* followed by some text. Set nextcmd to the following command.
~	*/
ehe gui	<pre>for (arg = eap->arg; *arg; ++arg)</pre>
gdb program [[No Name] - VIM] 0,0-1 All	~/vim/vim80/src/ex_cmds.c 6215,5 73%
:Break	

new-ui mi /dev/pts/17	Step Next	Finish	Cont Sto	op Eval	
Reading symbols from vimdone.	/*				
(gdb) new-ui mi /dev/pts/17	<pre>* ":help": op</pre>	en a read	I-only window o	on a help file	
New UI allocated	*/				
(qdb) run	void				
Starting program: /home/mool/vim/vim80/src/vim	ex help(exarg	T *eap)			
[Thread debugging using libthread db enabled]	{				
Using host libthread db library "/lib/x86 64-linux-gnu/libthread db.	char u	*arg:			
so 1"	char u	*tag:			
30.1 .	ETLE	*helnfd	• /* fi	le descriptor of	heln file */
Presknoint 1 ev heln (ean-0v7fffffffd/f0) at ev omds c:6215	int	n'	'' ···	the description of	neth itte i
6245 if (con 1- WILL)	int	1.			
0215 II (eap := NULL)	Line T	1,			
(gab)	WIII_I	wp,			_
	int	hum_	0		
	char_u	**ma ea	ip: UX/TTTTTTTT	1410 = {	
	char_u	*p;	arg = 0x555555	dcb463 "gu1",	
	int	empt	nextcmd = $0x0$,	0	
	int	alt_	cmd = 0x5555555	dcb460 "he gui",	
	buf_T	*buf	cmdlinep = 0x7	/ffffffd710,	
CARLON TO THE ALL AND A CARLON AND A	#ifdef FEAT_MU	LTI_LA	cmdidx = CMD_h	nelp,	
!gdb [running] 0,0-1 All	int	len;	argt = 2054,		
	char_u	*lan	skip = 0,		
~	#endif	SCHWOM 1	forceit = 0,		
-	#ifdef FEAT_FO	LDING	addr count = @	ð.	
	int	old	line1 = 1,		
VIM - Vi IMproved	#endif	1000	1ine2 = 1,		
 A THE SET OF THE SET	and the second second		addr type = 0 .		
version 8.0.1615	if (eap !=	NULL)	flags = 0 .		
~ by Bram Moolenaar et al.	{	nore)	do ecmd cmd =	0x0	
Vim is onen source and freely distributable	· /*		do ecmd lnum =	- 0	
VIII 13 Open Source and Treety discributante	* ^ "	help"	append - 0	. 0,	at a 111 that is
Recome a registered Vim user!	* fol	lowed	usofiltor - 0		following command
ture thelp register (Enters) for information	*/	TOwen	userincer - 0,		TOTTOWING COmmand.
cype .neth register <citer> for intermation</citer>	For In		Seral toral H	(000)	
tune instators to ovit		rg - eap-	sary, ary, th	rarg)	
 type :qcEnter> to exit type :qcEnter> or cF1> for on line boln 	1	Itora	Uni II tora		
 type :netp<enter> or <f1> for on-line netp</f1></enter> 	-11	("ary ==	(terr lul		
type :neip version8 <enter> for version info</enter>		11	(*arg == 1 @	arg[1] != NUL 6	& arg[1] != '['))
	1				
		*arg++	= NUL;		
he gui		eap->ne	extcmd = arg;		
A		break;	S2545		700
gdb program [[No Name] - VIM] 0,0-1 All	~/vim/vim80/src/	ex_cmds.c			6223,6-13 /3%
:Over					

What else can you use it for?



Run make in a terminal window

	!/bin/tcsh [running] - VIM 📃 🖨 🖲 🙆
-I/usr/include/pixma sr/include/libpng16 e/pango-1.0 -I/usr/i I/usr/include/freety c -Wunreachable-code TFREE -DABORT_ON_INT gcc -c -IIproto -I/usr/lib/x86_64-J nclude/cairo -I/usr/ -I/usr/include/pixma sr/include/libpng16 e/pango-1.0 -I/usr/i I/usr/include/freety c -Wunreachable-code TFREE -DABORT_ON_INT	1-1 -I/usr/include/libpng16 -I/usr/include/gdk-pixbuf-2.0 -I/ ·I/usr/include/pango-1.0 -I/usr/include/harfbuzz -I/usr/include/ plib-2.0 -I/usr/lib/x86_64-linux-gnu/glib-2.0/include pe2 -g -Wall -Wextra -Wshadow -Wmissing-prototypes -Wpedant -Wunused-result -U_FORTIFY_SOURCE -D_FORTIFY_SOURCE=1 -DEX ERNAL_ERROR -0 objects/quickfix.0 quickfix.c DHAVE_CONFIG_H -DFEAT_GUI_GTK -pthread -I/usr/include/gtk-2. inux-gnu/gtk-2.0/include -I/usr/include/gi0-unix-2.0/ -I/usr/ include/pango-1.0 -I/usr/include/atk-1.0 -I/usr/include/cairo 0-1 -I/usr/include/libpng16 -I/usr/include/gdk-pixbuf-2.0 -I/ ·I/usr/include/pango-1.0 -I/usr/include/harfbuzz -I/usr/include cude/glib-2.0 -I/usr/lib/x86_64-linux-gnu/glib-2.0/include pe2 -g -Wall -Wextra -Wshadow -Wmissing-prototypes -Wpedant -Wunused-result -U_FORTIFY_SOURCE -D_FORTIFY_SOURCE=1 -DEX ERNAL_ERROR -0 objects/regexp.0 regexp.c
<pre>!/bin/tcsh [running]</pre>	1,1 To
/* don't use	VTERM_KEY_ENTER, it may do an unwanted conversion */
case K_BS:	<pre>/* don't use VTERM_KEY_BACKSPACE, it always * becomes 0x7f DEL */ c = term_backspace_char; break;</pre>
0000 FCC	KON - VIEDN KEY ESCADE, brook
case ESC:	key = VIERM_KEY_ESCAPE; DIEAK;
case K DOWN	key - VTERM KEY DOWN: break;
case K S DOM	mod = VTERM_MOD_SHIET
	key = VTERM KEY DOWN: break:
case K END:	key = VTERM KEY END: break:
case K S END	mod = VTERM MOD SHIFT;
	key = VTERM_KEY_END; break;
case K_C_END	mod = VTERM_MOD_CTRL;
	<pre>key = VTERM_KEY_END; break;</pre>
case K_F10:	<pre>key = VTERM_KEY_FUNCTION(10); break;</pre>
terminal.c	1089,14-21 21

Running external command in the GUI

File Edit	Tools Syntax	Buffers Win	dow Plugin	Help				
🗎 🙆 (🔒 😫 🤸	🧀 📈 🖻	i 🛍 📿	> < 🖻		\$	3 0	0
ca ca ca ca ca ca ca ca ca ca ca ca ca c	ISE ESC: ISE K_DEL: ISE K_DOWN: ISE K_S_DOWN: ISE K_S_DOWN: ISE K_S_END: ISE K_S_END: ISE K_F10: ISE K_F12: ISE K_F12	key key mod key mod key mod key key key key key key key key key key	 VTERM_KEY 	Y_ESCAPE; break; Y_DEL; break; Y_DOWN; break; D_SHIFT; Y_DOWN; break; Y_END; break; D_SHIFT; Y_END; break; D_CTRL; Y_END; break; Y_END; break; Y_END; break; Y_FUNCTION(10) Y_FUNCTION(11) Y_FUNCTION(12) Y_FUNCTION(2); Y_FUNCTION(2); Y_FUNCTION(3); Y_FUNCTION(4); Y_FUNCTION(5); Y_FUNCTION(6);	ak; ; break; ; break; ; break; ; break; ; break; ; break; ; break; ; break; ; break;			
ANSI color cube								
Press ENTE	R or type co	mmand to cont	inue			Į		

Testing with a screenshot diff



Testing old style

.in file:

```
STARTTEST
:so small.vim
:set belloff=all
/Start cursor here
vaBiBD:?Bug?,/Piece/-2w! test.out
/^- Bug
:s/u/~u~/
:s/i/~u~/
:s/i/~u~/
:s/o/~~~/
:.w >>test.out
```

.ok file:

_	Bug	in '	'vPPPP"	on	this	text	(Webk):	
		{							
		}							
_	Bug	uuur	vPPPE מי	י "נ	่นนนนน	uuun	this	text	(Webb):

Testing new style

.vim file:

```
func Test move cursor()
 new
 call setline(1, ['aaa', 'bbb', 'ccc', 'ddd'])
 call cursor([1, 1, 0, 1])
 call assert equal([1, 1, 0, 1], getcurpos()[1:])
 call cursor([4, 3, 0, 3])
 call assert equal([4, 3, 0, 3], getcurpos()[1:])
 call cursor(2, 2)
  call assert_equal([2, 2, 0, 2], getcurpos()[1:])
  " line number zero keeps the line number
 call cursor(0, 1)
 call assert_equal([2, 1, 0, 1], getcurpos()[1:])
```

Testing with a screenshot

.vim file:

```
call term_sendkeys(buf, "GA\<C-N>")
call VerifyScreenDump(buf, 'Test_popup_position_01', {'rows': 8})
```

Testing with a screenshot diff



Release: "in a few weeks"





Questions?