

Re: Great SWT Program

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2007-10/msg02875.html>

- *From:* bbound@xxxxxxxxxx
 - *Date:* Wed, 24 Oct 2007 20:26:19 -0000
-

On Oct 21, 11:47 am, blm...@xxxxxxxxxxxxxxxx <blm...@xxxxxxxxxxxxxxxx> wrote:

This is a point. In practice it doesn't feel like a bother to me, because most of the key sequences that don't show anything are so short and so automatic

Eh?

Most of the longer commands (those beginning ":", or searches) bring up a one-line "window" displaying the command so far and allowing one to move around in it and edit.

Strange behavior for a hotkey-driven interface. But it would seem to mostly solve the blind-typing problem that I definitely recall emacs has.

Except of course that you need to know some arcane command language instead of it being self-evident what a give command will do, whereas (if properly designed) a GUI's command buttons and menu items state their effects in plain English (or whatever language, modulo localization -- ah, localizability, another advantage of GUIs!)

Ah. But the thing is, in vi, insert mode -- hm, I'd almost say it's no more the "normal mode" than command mode. So a lot of the time, *all* keys behave as what you're calling "hotkeys". <shrug>

All seems like a bad idea. The ones that do something should (aside from arrow-like navigation behavior ones, which can clump with their own kind) have buffer space of unassigned keys around them so that fat-fingering one of them won't do something drastic. You don't want "save" and "delete everything" to be adjacent, for instance...

Re: Great SWT Program

And this is different from the cat jostling the mouse and then stepping on its buttons how?

It doesn't put the UI into an invisibly different state. It either does nothing or puts it into a visibly different state. And usually you can count on ctrl+Z undoing any damage in a normal Windows GUI app. If there is such a key in a unix app, it won't be Ctrl+Z and it won't be the same as in any other unix app either. :P

What evidence is needed? It's simply common sense. Likewise, if your car is capable of doing sixty and you're on the highway with little other traffic and nothing moving too slowly, why the heck would you choose to do only forty?

Speed limits? :-)

Forty? On the highway? In some repressive place like China perhaps. :P

Because I don't think it's apt. *Experienced users* of those old text-mode applications find them to be more productive than the tools you seem to be championing.

Experienced users of either that don't use the other will be more productive with what they're experienced with. I expect it goes something like this:

Inexperienced	Experienced
GUI 5	10
vi 0	8

Hell, the lower right number can even be a 10 for all it matters. Experienced GUI users see a 10 with a GUI and a 0 with vi. Experienced vi users see a whatever with vi and a 5 with a GUI. Each thinks theirs is better. But only one of them really is.

I don't know how you can refute this, given that as best I can tell you don't know any such people, other than the ones in this newsgroup, and you don't seem to believe us.

It's that whole extraordinary claims, extraordinary evidence thing. You're claiming, in essence, that you could be more productive working with hand saws, sandpaper, and a hammer in pitch darkness with a

Re: Great SWT Program

flashlight held in your teeth trying to build a house than I could be with power tools in broad daylight doing the same job, while each is proficient with his or her respective tools. That's dubious on its face.

Yes. So very likely the original vi didn't have anything resembling mouse support. What does this have to do with vim, or other vi clones, or modern emacs?

What it means is that mouse support is useless on software designed for a text terminal environment. Whereas emulation of text terminal limitations is useless on software designed for a modern workstation. If this software you champion is designed for the one environment then there is something wrong/silly with it. If it is designed for the other then there is something else wrong/silly with it. It's an inescapable trap, I'm afraid; I've got you.

That's even more perverse. Now we have two perfectly good eyes, elect to wear the eye patch for some oddball reason — and then cheat by pulling it back and peeking with the "dead" eye now and again. :)

I'd say it's closer to trying to have the best of both worlds —

The best of both worlds, where one of these is "being healthy" and the other is "being half-blind". Er, right. Isn't that achieved simply by being healthy? Well, maybe you miss out on the odd sympathy lay from someone that gets Florence Nightingale syndrome or something. Hence wearing a fake eye patch and peeking now and again?

Except that unix geeks don't ever get laid, so it's moot. :P

point-and-click for things you don't do often, and all the power of the old-style editor for things where you're willing to master a more obscure interface.

People keep rabbiting on about this mythical power that lurks in the keyboard and is only accessible by mastering the magical meta key somehow, and is somehow impossible to implement in a modern UI. I have yet to see any evidence that it exists, or hear of any feature of either vi or emacs that I would a) actually find helpful and b) couldn't achieve just about as easily with my preferred tools, albeit perhaps without setting any speed records with some of the ones I wouldn't be using on a more often than hourly basis or thereabouts anyway; plus my preferred tools don't have huge complication, but a

Re: Great SWT Program

small number of orthogonalized features (or even orthogonal separate apps that interoperate with copy–paste) that can be used in a large number of combinations without the associated learning curve suffering a combinatorial explosion (unlike, especially, with emacs).

So maybe you can help me understand better When you say "how is that possible?" in response to some claim I make, what *are* you saying? if you're not implying that I'm mistaken, or deliberately lying?

That you're describing something highly implausible, and if it actually does work as advertised, it probably does so only for people with memorization and related skills bordering on savantism.

Uh–huh. And quietly setting an example with your own posts of mentioning no–cost tools and sources of information, rather than jumping on any mention of anything commercial, unless it's clearly identified as such and accompanied by a list of no–cost alternatives Well, I'm sure that's different.

Strange. I don't recall any such matter arising in and therefore being relevant to this particular thread.

And regardless, playing bait–and–switch with hyperlinks is a rather different matter. I also don't care for some cases where Google has been known to abet such behavior on the part of web site operators -- I've run into more than a few cases where Google's search has a hit with excerpt text that is highly relevant, but isn't on the destination page because the destination page is a login page. If they want free traffic from organic search referrals, then it has to be, IMO, from indexing free content. Letting googlebot past a registerwall or even a paywall is some species of cheating and Google not doing much about this "cloaking" behavior is a form of bait–and–switch. Thankfully, I haven't noticed any instances of this in the past few months so maybe Google's gotten its act together and visits sites occasionally with a non–Google–looking probe to check that what their bot indexes at a page is really representative of what a human will get if they click the corresponding search result.

This sort of pay–link–masquerading–as–free–info thing, whether really intentional or not, strikes me as dishonest and rude.

The last time I checked, it's a) difficult and b) useless. The bottleneck is in the brains in the person behind the keyboard, these days.

Re: Great SWT Program

Sure. I still find that the ability to type without thinking about where all the keys are is useful. You don't?

I do too. Why, did you think I had to think about where all the keys were for some reason?

Typing k, which is "right there", to move up a line. (I had to pause a minute to think about whether it's j or k that moves up a line.

See? That doesn't happen with the arrow keys!

I guess I didn't make my point clear: My brain says "move up a line", and a 'k' is typed. I'm not thinking about it being a 'k'; I'm thinking about moving up a line. I only have to think about what key is involved if I want to tell someone else.

That does not make sense. There's clearly an extra translation step involved, and that has to have a cost in cycles and memory use. You just may not be consciously aware of it except when you're discussing the subject itself at the meta-level as we are here and therefore are consciously paying attention to it in a way that you normally don't.

Of course it's simple. What it isn't -- for me anyway -- is something I can do without thinking about what keys I need to press.

It doesn't take long to get used to, surely? It's pretty clever, making a) any navigation key also able to move a selection endpoint with a single modifier so as to orthogonalize things and cut down on learning curve and b) extending this to even include mouse clicks.

Surely, though, simple typing consumes the vast majority of the time you spend using a text editor.

Maybe that's true for some people. I apparently am unable to turn my thoughts into prose without substantial amounts of revision, however [*], and I'm fairly sure that the vast majority of the time I spend in a text editor is *not* simple text entry. Add to that

Re: Great SWT Program

the fact that I'm apt to use text editors to write things other than simple prose

Strange. I mainly revise as I go. Sometimes I delete a chunk of what I just typed and then retype some, but I rarely go way back in e.g. a usenet post and change things near the top. And of course if I'm not editing prose, or just notes of some sort, I use a more specialized tool. For code that would be my IDE, for instance.

Same under Windows, and the lack of a less-subtle visual indicator of the mode change is troubling here. But we were discussing a text editor, not a Web browser, once again. Hitting tab should insert a tab at the insertion point when in insert mode. If you're typing in text such as a filename presumably you're in insert mode;

Ah, is **that** where the confusion comes in No, if you're typing in text meant to be interpreted as a filename, you're almost certainly in the "enter an 'ex' command" mode.

OK, OK, time out.

Just how many fucking modes does this monstrosity have, anyway? :P

In "real" vi, in insert mode, pressing a key other than "ESC" inserts a character into the document. (vim can emulate this behavior, but normally doesn't quite, instead allowing the arrow and backspace keys to do what most people these days expect them to do, even in insert mode.)

Ludicrous that "real vi" didn't --- nobody in their right mind would code a text editor on the assumption that people hitting up-arrow want to insert it literally into the text, especially as it wouldn't be an up-arrow symbol, but some unprintable control character. Worse, displaying it would throw the whole screen out of whack and overprint part of the previous line with subsequent text, hiding text that is actually part of the document...left-arrow would behave like backspace except that the "deleted" text is still buried in there wasting memory but not being displayed...and so forth.

If people are inserting control characters into something, then that something is a binary file and they should be using a hex editor, or better yet generating it via some kind of compilation process from plain-ascii source code of some sort. If they aren't, then up arrow means move up. Hell, up arrow ALWAYS means move up. Nobody hits that key for any other reason, ever. Sheesh.

Re: Great SWT Program

I guess. When I'm editing code, I have things set up so that tab inserts an appropriate number of spaces instead; I don't like having literal tab characters in code. <shrug>

Must be a pain when it comes time to de-indent a section then. Eight times as many del/bksp keypresses per line. Ouch!

Worse, of course, is "Missing separator. Stop." I'm guessing make drives you batty when you need to use it and you forgot to turn tab behavior back to normal when editing the makefile way back when and, as a result, it got b0rked?

And, as noted above, filenames would normally only be entered as part of one of the "'ex' commands", which really is a mode distinct from both insert and command modes. I thought I mentioned it earlier. Perhaps not, or perhaps you don't remember. <shrug>

Frankly, I'd rather go on blissfully imagining that this beast has only the three modes, and you'll keep your knowledge of the other umpteen to yourself. ;P

(Yesterday, that number had been two instead of three.)

since I don't think you'd actually lie about something like that.

Well, that's a small success!

Well ... I'll have to consult with a psychologist or similar expert, but I don't *think* a delusional person is actually technically lying, as there's no deceptive intent, no matter how outrageous the claim (e.g. they profess to be the emperor of France, in the canonical example).

:)

":" to enter "'ex' command mode.

Won't that insert a ":" in insert mode? Great -- either we have modes within modes, or at least some modes that can't be jumped to directly from some other modes. What a catastrophic mess!

Re: Great SWT Program

Re: Great SWT Program

"r" to say I want to copy something into the file being edited.

" " to indicate that I want to copy from a file.

"myL" followed by <tab> to give the filename.

<enter> to end the command.

Of course, if you have myLongFileName001.jpg through myLongFileName999.jpg it won't know which of these you mean ... saving the ".jpg" by reducing it from four keystrokes to one isn't a big gain percentage-wise in this instance. :)

Well, I don't suppose you'd be inserting JPEG files anyway, but I think you get the picture.

Insert mode not involved. As soon as I press ":" I get a little one-line "window" in which the rest of what I type appears, so I can review the result of the autocompletion before pressing <enter> to actually do anything.

Sounds to me like the display ends up in an ambiguous state. If you walk away for a moment and come back how do you interpret it? There's your document, with a half-completed line and blinking cursor at the bottom. Is it a half-done "ex command" or a half-done line of the file in insert mode, or a half-done line of the file but the thing's in command mode? And that's considering just the three modes you've mentioned so far...

So much for easily deleting away full indent levels at a time then. :P
Having to hit backspace 8 times in a row to move stuff left one indent must get pretty tiresome pretty fast.

So I'm glad I don't have to do that.

Except that you do. You have eight spaces to delete instead of one tab character. A child can do this math correctly!

(1) Use search-and-replace to remove a selected number of spaces from the start of each line, applied to a selected range of lines.

Re: Great SWT Program

That's not possible. Search and replace for spaces either a) won't work or b) will mangle the hell out of everything. If you have e.g. 24 spaces and do a replace of nine spaces with one it will match the first nine and reduce the 24 to 16, then match the first nine of those and reduce them to 8, then match some block of nine spaces in a string literal somewhere...

Of course, if it were possible to enter a newline into a search query field you could limit it to the start of line and not mangle the string literal. Most searches I've seen wisely don't attempt to match pure-whitespace queries. And of course most treat enter as "do the search" rather than inserting it literally into the query (and if it did the latter, then how would you launch the freaking search??)

Secondly, this is a crummy workaround for the fact that you've got a bunch of spaces that really represent a single logical entity that you should really be able to backspace or delete as such.

(2) Use the Probably the simplest way to say this is "the equivalent of control-I in Eclipse". (Yes, vim has something like that. So does emacs.)

Use the what? Control-I is 0x09, the tab character. Are you saying you have it set up to convert tab keypresses into spaces, then do an end-run around this to insert literal tabs anyway? This seems to be even more seriously broken behavior, and this time it's not software behavior that violates least surprise I'm talking about, either, but behavior of the component that goes between the keyboard and the chair...

I think there are other ways too, which I don't use much and so don't remember.

None of which are necessary except as workarounds for a user purposely crippling other functionality, as far as I can tell.

Whoever designed this thing was on crack. Where are the narcs when you need 'em?

Yeah Well, I don't like eight-character indents — four seems more reasonable — so if I used literal tab characters, I'd have to set all the tools I use for dealing with source code to use something other than the default expansion, and anyone else who looked at the code would also have to <shrug>
Another religious-war topic, I think.

Re: Great SWT Program

No—one else who looked at the code would have to do anything of the sort — everyone could use their own preferred indent–per–tab when viewing and editing, which is another argument strongly in favor of real tabs.

So the explanation is that you don't like configuring things.

It looks like I've nailed you on three more counts here.

- a) All that hand–hacking of configuration files does take extra effort and translates into a reduced willingness to use non–default settings, even when they'd only have to be set once per app.
- b) Also, you're missing having a nice fat 200+–character–wide GUI text box to edit in even if you claim you aren't.
- c) Finally, there's the little matter of there not apparently being a single global "tab width" setting that everything uses. Isn't that the sort of thing you claimed your preferred OS was good at, and isn't this the sort of problem that you associate with evil Windoze and dumbed–down MacOS? ;)

Yes, that **is** nice. Fortunately, vim and emacs provide similar functionality.

Feep! Feep! More IDE–specific features in a supposedly general–purpose text editor! Feep! Feep!

There's not a lot of overlap between old–style [*] vi keybindings and old–style emacs keybindings, that's true. In my experience, most other old–style text–mode programs were apt to adopt either vi–style keybindings or emacs–style keybindings.

Impossible, if they weren't text editors.

No other tool involves text entry, or the need to move the cursor around? or the need to search for text?

No other tool revolves around it, because a tool that revolves around text entry is, by definition, a text editor. Well, IDEs and word processors might be considered subtypes or entirely separate, but whatever. But lots of things have no use for such. Consider a paint program. What text do you ever enter in one of those? The odd filename

Re: Great SWT Program

and the odd word or, at most, paragraph of literal text to embed in the image graphically. You don't need anything fancy in this case — the alphanumeric and punctuation doing the expected along with the arrow keys, bksp, del, home, and end will suffice. Page up and page down are superfluous, as is search. Copy, cut, and paste of course are going to be used everywhere, and longer passages can be typed up and edited in something else and then pasted in anyway, and, for long enough passages, the extra time it takes to use an external tool here will be small compared to the time spent typing. In filename input boxes you again have no use for page up, page down, in-text search, or much else besides arrows and cut, copy, paste; plus of course selecting a file in the selection box with the directory listing should replace the input field's contents with the appropriate filename, and you might want drop-down MRU functionality or similarly.

About the same as the CUA keybindings, isn't it?

Which have the distinct extra bonus feature of being familiar to 99% of the computer-using population.

Well, at least now you know where "s/old/new/" comes from — though I'm not sure you should be using it given your opinion of the tools where it's still used.

```
cat usenet_posts_telling_me_what_I_should_and_should_not_write/* > /dev/null
cat first_amendment.doc > etc/motd
```

;P

No, I do not. What I do say is that I don't know of any tools that are novice-friendly and also provide the set of expert-friendly features I'm accustomed to and like.

"The set ... you're accustomed to and like". Perhaps they provide a set, but it's as alien to you as the ONLY set of features provided by your tools are to me? The typical Windoze app has zillions of key-bindings; you just can get away with knowing almost none of them, because there are multiple ways to invoke a given command. Every menu item, or at least a substantial fraction, has a three-character (sometimes four) sequence of separate keypresses (no chords) starting with hitting either alt key, then several letters one by one. Several common commands will have a ctrl+X type chord, usually as well as a menu-navigation sequence. Plus some way of triggering it by mouse. And there will be all sorts of commands. The editors are more specialized

Re: Great SWT Program

though. You won't find IDE-type functionality oriented toward editing program code in Notepad or Word, but you will find such in other editors and especially in IDEs. You won't find word processing type functionality in Notepad or an IDE but you will find such in Word and WordPerfect. And so forth.

[*] Unless you include the graphical front ends developed in recent years for some of the old tools, emacs and vim for example.

Worst of both worlds. Keybindings are nutso, and the GUI is usually a) not only an afterthought but *clearly* an afterthought and b) a cheap imitation of the real McCoy anyway. To me, running into a port of a unix tool that's trying to fit in on a Windoze box is like coming across a cheap imported knockoff of a useful tool that a) has wonky and not-quite-right knobs and switches that stick or break off easily and b) there are all kinds of semantic differences. Picture a television that from a distance looks normal, with English writing where there's writing and normal-looking controls, only a) the remote is flaky and a bit odd, with the channel up and down buttons widely separated and a little dial for the volume control instead of another up/down arrow pair; b) the screen's a bit oddly proportioned and someone seems to've been screwing with the brightness, contrast, and tint settings; c) it's bulky CRT; d) the front panel has 1970s-style faux wood paneling and knobs instead of buttons, knobs that fall off easily and have to be put back on their little metal posts, and then end up pointing at the wrong things; and e) to top it off, even on channel 2 it's trying to tune in something in the upper UHF band and interpret it as a PAL signal, and as a result it tunes between local channels 17 and 18, and doesn't even get a ghostly overlap of the all-news channel and KBNY but rather some sort of complete hash and squealing that looks like you tuned to the "24 hour video feedback channel" during technical difficulties that make the picture skip and roll.

This when the state of the art is a flat-panel widescreen rear-projection, LCD, or plasma display with 1080p and digital signal handling, connected to a satellite or digital-cable service, with 500 or so channels (and nothing much on), P in P and matrix functionality, and you can't remember the last time you saw snow let alone an actual rolling picture.

(Funnily enough, back in the late 80s some Hollywood types almost perfectly guessed what the entertainment system of the future would look like. Only thing they got wrong was that they apparently thought the flat-panel widescreen TV picking up 500 channels would not become commonplace until 2015! The movie, of course, was the second Back to the Future flick...a few other things they got bang on included the old black and white Macs being antiques, teleconferencing from home and getting insta-fired by irate boss mouse-click, and those little

Re: Great SWT Program

pizzas you rehydrate and nuke that are ready in 30 seconds. Also with the timing off by eight to ten years. Which leaves us to look forward to ... flying cars and the hoverboard, and, I think, telescoping baseball bats. Oh, and the Cubs winning the series. That still hasn't happened either. But I digress...)

Long story short: trying to bolt a GUI onto something that was not designed for one in terms of its set of UI operation semantics -- bad. Trying to emulate a modern GUI when it's clearly not your main area of expertise UI-wise -- doomed to failure. Subtle things will behave differently and unexpectedly. A common mistake is for the focus to change on hover, without waiting for a mouse click, or else to only change on a *double* click. Drag and drop is often missing or woefully broken to anyone who ever actually uses it routinely. Scrollbars and selection are most critical and also not immune to being misunderstood by imitators. The most common complaint is crummy Windows 3.1 era behavior such as not scrolling the content pane in real-time and the thumb not being sized proportionally to the proportion of the document in view. (I see this in ALL attempts by cack-handed amateurs to emulate a window interface without using the native windowing API. Pros, such as Windows, Mac, Java Swing, and such copying each other, get things right. Amateurs, such as coders clearly mainly familiar with text-mode Unix and anyone working in the execrable medium known as Macromedia Flash, get so many things wrong wrong WRONG.) Tabbed dialogs don't work right or they never heard of tabs. Focus tab-order is broken or wonky, and may actually form a disjoint graph such that you can't get to some controls from other controls on the same dialog. Menus are usually conspicuously off, for example conventions like file and edit and help menus and what to put in them are flagrantly violated, and the mechanics are almost invariably botched, with one of the commonest errors being to retract menus if the mouse moves so much as a pixel outside their bounding boxes, or worse, if the mouse button is not actually held down until the desired item is selected. The absolute worst offender, though, ironically, is typically the lack of keyboard shortcuts! Often the alt, this, that ones are broken, missing, or spottily implemented, and whatever control key shortcuts there are are nonstandard, not documented on the menu items, or (worst of all) both. Or they're documented abnormally, e.g. C-x instead of Ctrl+X (I saw this gem in some sort of half-assed graphical emacs port for Windows; to add injury to insult, the offending menu item was not Edit/Cut).

The one saving grace is that the main window tends to behave somewhat normally, generally because the system windowing APIs more or less physically restrain them from fucking things up *too* badly. Still, I've seen such egregious behavior as hooking the maximize button press and using it to do something funky like force the window to be 300x800 and centered instead of actually behaving normally, not to mention minimization going neither to the taskbar nor to the tray but apparently to the wonderful wizard of Oz, forcing me to find the fucking process and kill it in Task Manager as it had just decapitated

Re: Great SWT Program

itself and continued to run like a zombie chicken in some post-apocalyptic slaughterhouse in a B-rated movie so bad it should have been rated NC-17 instead on the grounds that nobody under the age of 18 should ever have their innocence so thoroughly shattered by the disillusioning realization that yes, it really is possible for a major studio to make a movie this awful *and turn a profit on it*.

Especially fun being the ones that assume you have a 1600x1200 monster SVU with 4WD and antilock brakes or some such nonsense, so they spawn a window the size of Rhode Island whose menus are excruciatingly slow to open but then stick around or, worse, stutter when you try to move to another menu or cancel the menu by clicking half a mile away. (Nearly as fun as the one that had every damn thing but the kitchen sink crammed into a single giant menu. When this was displayed the OS jumped through hoops to try to accomodate it on a merely normal 1024x768 video display and it wound up covering the whole screen in black 3-point text on light grey, even covering the taskbar. And the app somehow ate alt-tab keypresses while focused. It took some cleverness with ctrl-esc and ctrl-alt-del to get the focus elsewhere and still the menu hung around. Got it back and fortunately alt-F4 had it quietly fold up its tent and slink off in the usual manner, no mess, no "Save changes?" either unfortunately.)

And that's ignoring the "afterthought" syndrome. For example the emacs port didn't have proper MDI. If you figured out how and did whatever you do in emacs (I forget now) to spawn another concurrently-open document it split the single window's big text area in exactly the manner of a text-mode emacs. May as well have been in a fucking terminal emulator. It neither spawned a genuine second window that would respond normally according to the native environment's normal window semantics and commands, nor did it internally generate windows (like JInternalFrames) or tabs in proper MDI mode that could be navigated among in a sensible way. I don't think it even properly changed focus if you clicked the mouse in the other panel (I refuse to use the term "window" for what clearly bore no resemblance to the term as normally used on the host system whose UI it was valiantly attempting to sort of do something resembling emulate)...

On the other hand there's a strong correlation between "big complex program" and "powerful" and between "small simple program" and "no-frills"; consider Photoshop or Word vs. Paint or Notepad. Logically this means there should be a strong correlation between "GUI program" and "powerful" and between "text-mode application" and "not powerful". Yet you repeatedly and stridently assert exactly the reverse.

A lot of the power of the old tools comes from the fact that they play nice together.

Re: Great SWT Program

Except for emacs, which assimilates all the other old tools and keeps intoning "Resistance is futile" instead...

<shrug> Further, some of the "power" of the big complicated tools is --- stuff I can do without. The infamous dancing paperclip is an example.

An aberration. You shouldn't really expect much from Microsoft in that area, to be honest.

[discussion of multithreaded code, debugging, etc. snipped --- some interesting points, but no time and inclination to respond right now]

Eclipse is such a breath of fresh air, especially after reading and writing these posts!

I can think now of one use for the old unix tools. We lock them up until the 24th century as top-secret highly dangerous munitions, and when the time comes, we turn them loose on the Borg. They try to assimilate everything, so we feed them a few bait ships loaded with computers running stuff like emacs. They eat them and go bonkers. The emacs ones and the vi ones go to war. The emacs ones thrash and swap until they exhaust all available core, then self-destruct. The vi ones just go quietly mad and cease to be as big a threat as when they were big, bad, and still goal-directed. One bunch installs grub and shortly thereafter there's a cube full of infantile blank-slate Borg wetting their diapers and crying for Mommy the Borg Queen, who is soon running a full-time day care and too busy changing diapers and stuff to plot the downfall of the Federation. Then SCO sues the lot for patent infringement and makes a whole buttload of specious allegations, gets assimilated for its troubles, and suddenly the Borg are more the lawsuit-filing type than the big bad m*thaf*ckas with the big guns and the bad attitude. They keep suing the Federation for racial discrimination (and, of course, patent infringement, despite the fact that EVERY technology they use has a metric shitload of prior art) and being countersued (for --- what else? --- patent infringement, of which they are of course especially and massively guilty) while slowly going bankrupt. End of threat. :)

.