

UndoDB 2.2



Graham Morrison finds *UndoDB* the perfect tool for finding exactly how his own convoluted and obfuscated code works – now to debug *Brain Party*...

In brief...

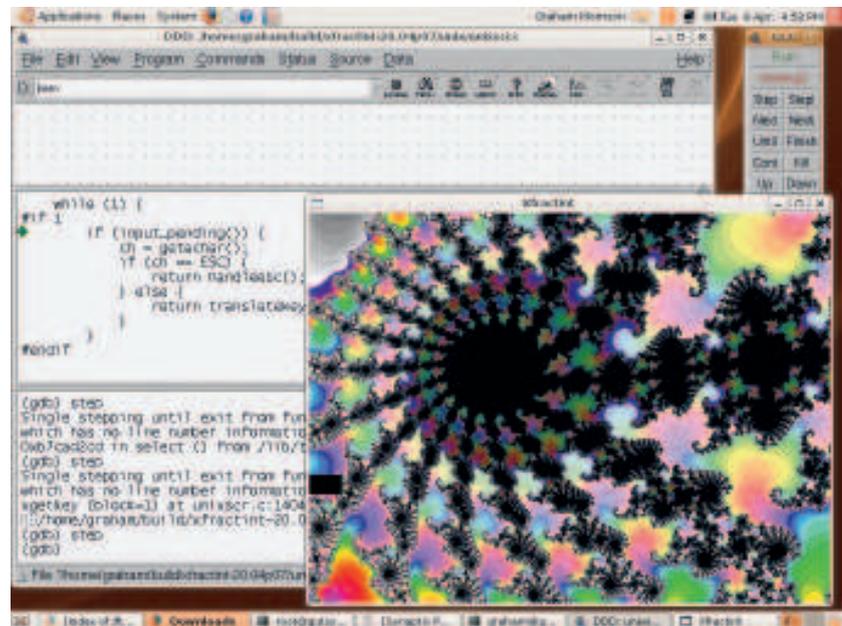
» Debugger capable of going backwards through program execution. See also: *GDB*.

We've given quite a bit of coverage to *UndoDB* over the last couple of years. We wrote a feature on the technology in **LXF86**, and reviewed version 2.0 in **LXF95**. Version 2.2, which promises a dramatic speed boost along with the ability to attach *UndoDB* to a running process. *UndoDB* has always been cripplingly slow, which isn't surprising. It performs the programmer's equivalent to alchemy, letting them turn back the clock on a program's execution.

Debugging software always adds considerable strain to your system – the CPU is coping with a glut of extra code attached to the program executable, as well as running a debugger that clings to each line of execution like a parasite. Running a graphical application through *GDB*, the open source debugger used by almost everyone, will typically halve that app's performance. When we looked at *UndoDB 2.0*, we found that its slowdown was closer to a factor of 40.

The reason for this massive difference in performance is workload. *UndoDB* does a lot more than the average debugger. For example, if we were debugging a program with *GDB* and the variable 'number' has a value of '123' on line 7, changing to a value of '321' by line 12, it would be impossible to find the previous value without restarting the program. It's a one-way street, and this can makes certain bugs very difficult to track down. With *UndoDB* you can work your way backwards through the program and probe the state of your program as if you'd run it linearly. It can do this because it stores the state of your program at each step of execution, and that takes some

» *UndoDB* is built around *GDB* and offers a superset of the original command list, making it an ideal drop-in replacement.



» Thanks to the impressive speed boosts in the new version, we found that we could interact with graphical applications for the first time while running *UndoDB*.

serious processing. But the trade-off is worth it, because without this facility, many bugs would take half a lifetime to solve.

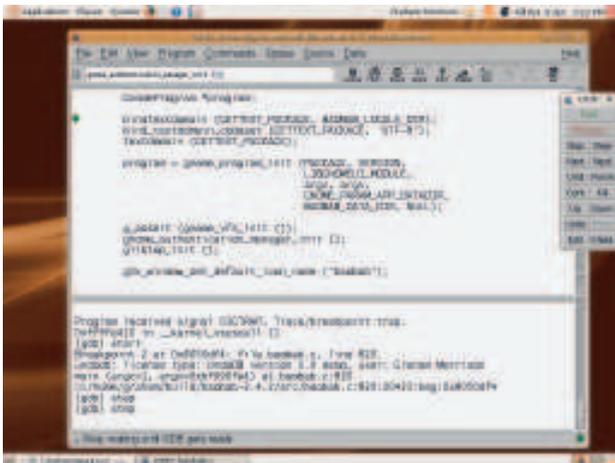
Speed++

To test the promised speed improvements in the new version, we ran it against the last release of the previous version. We pushed a variety of software through both, from a handful of command line tools to full-blown applications, and we did see a dramatic difference. Tools that don't use up too much CPU power, such as the average command line utility, ran about the same. The old version was even slightly faster in a couple of instances.

But the more resource-hungry the application, the more *UndoDB 2.2* impressed. We tried *XFractInt*, a basic *X Windows* interactive fractal utility. With *UndoDB 2.0*, it was barely usable. Rendering a small fractal image took around 11 seconds, and the interactive part was almost non-existent. The toll on the CPU made the mouse very difficult to use, and while none of this got in the way of debugging, it was a real test of patience. It was a different story with *UndoDB 2.2*. The same fractal took less than a second to render, and the user interface remained interactive at all times. We only ran into problems when the window border needed

to be redrawn for some reason. Otherwise, *UndoDB 2.2* was easily 10 times faster than the older version.

It's still slower than *GDB*, but it's getting harder to see the difference between the two. This will make *UndoDB* far more viable to people debugging graphical tools, and worth a second look if you found the old version too slow. *UndoDB* has always been about the trade-off between its unique take on debugging, and the extra resources it consumes. With this release, that trade-off looks much better value. **LXF**



LINUX
FORMAT
Verdict

UndoDB 2.2

Developer: Undo Software
Web: <http://undo-software.com>
www.techradar.com/284286

Price: Free for non-commercial use. £150 ex VAT for 1 year, £250 perpetual

Features	9/10
Performance	8/10
Ease of use	7/10
Value for money	9/10

» Not worth the upgrade for debugging small apps; otherwise essential.

Rating 8/10