

frontwheeldrive.com

[home](#) | [interviews](#) | [reviews](#) | [mentors](#) | [about](#) | [contact](#)



Anno Dominic Dominic Giampaolo

[by Tom Georgoulas]

Dominic Giampaolo isn't exactly a household name, but chances are high you've actually seen the benefits of his work. Dominic was the guy who tracked down a bug in the graphics driver during his days at SGI that was inhibiting a digital effects studio from getting their work completed on the movie Speed. He was also responsible for tracking down a gnarly bug in an eight processor system that only occurred within a window of 12 nanoseconds. Dominic worked on various portions of the RealityEngine before moving to Be Inc. as a kernel engineer and file system architect for the BeOS. I met Dominic several years ago through a mutual friend and we've been corresponding ever since, discussing topics like music, traveling, and software on a weekly basis. Dominic recently left Be Inc. for a senior project management position at Google, so I felt it was time to put him under the frontwheeldrive spotlight and get his take on various events such as the rise of Linux and the Microsoft anti-trust trial.

frontwheeldrive: As I understand it, after your graduation you basically drove from Worcester, MA to San Jose, CA and landed a job at SGI. How did you like it there?

Dominic Giampaolo: I loved working at SGI. It was like having a big extended family due to the way some of the managers in my group ran things. I came out here and knew absolutely no one and they took me in and made me part of their group. Of course I worked my butt off but when you're fresh out of school, there is no other way. I was very sad when I left because I was leaving a lot of very good friends.

What kind of stuff were you working on?

I worked on a variety of things there: I did bring-up for the RealityEngine graphics driver on 64-bit Irix; I made OpenGL 64-bit clean; fixed a ton of bugs in RealityEngine; found way too many hardware bugs in one of the support ASICs for the R8000; helped with some hardware design and verification for the InfiniteReality product and did a little bit of code in the driver for it.

After you left SGI and moved over to Be Inc., you arrived early enough to have a great bit of influence on the design of the BeOS. I know you were heavily involved in the Be File System (BFS), amongst other things. Tell us more about the work you did and why you took the positions for implementation that you did.

I worked on loads of things at Be. When I got there the kernel was lacking a lot of features, the posix sub-system was a mess and the command line environment was really deficient. I set about trying to fix as much stuff as I could. And then I got to work on the file system which was by far the hardest (and most rewarding) project I've ever done.



further reading:



metamedia:

[Google](#)
Easily the best search engine out there.

[BeDope Interview](#)
TA short but entertaining interview from BeDope, the Be Inc. fan site that makes up it's own news.

[Inside the BeOS](#)
Take a stroll through the BeOS with Dominic and other Be engineers.

You also wrote a book about the BFS called *Practical File System Design with the Be File System*. What was it like to write the book?

Writing the book was tons of work. I estimated that it took me about 400 hours to write it. You definitely don't do it for the money :-) I had a lot of fun doing it though - while you writing it you start to realize things you could have done better or differently and it gives you ideas for the next time around.

You've got some pretty entertaining stories about debugging. Share a few with us.

I'm not sure how entertaining they are... I know I've spent way too many hours in front of a monitor trying to figure bugs out.. One time at SGI the folks from Discreet Logic (makers of film/video editing software) came to us and said "the machines in the production studios for some films are crashing every three hours and these films aren't to get done if these bugs aren't fixed". The bug had been around for years but no one had been able to fix it. I said "I won't leave until it's fixed". Three days later I went home and the bug was fixed. The sad part of the story is that the bug fix wasn't some incredibly arcane problem, it was two lines of code that had been `#ifdefined` out accidentally and no one remembered/knew. It took so long to find because it was an obscure case so the code path wasn't exercised very often (and the bug wouldn't happen every time the code was hit) but still it was just two lines of code that were commented out...

At Be I can't think of any particularly hairy debugging sessions... there it was it just more of beating at a problem until you figured it out. Some bugs are like that... it takes a good 18 straight hours to get into it and understand and devise good test cases for (but then again I may not be the sharpest tack in the box :-) I know one time while trying to debug the code for writing and reading large files I was pretty lost so I printed out about 11 feet of code, taped it together and put it out on a large table in one of the conference rooms so that I could see it all at once and annotate it. That's not particularly funny though... more depressing than anything :-)

Now for a question posed in nearly every tech interview I've read this year: What's your take on Linux and open source software?

Linux is an interesting system. Despite the arrogance and religious zealotry surrounding Linux kernel development, they have done some things very well. Unfortunately though they've also got Unix on the brain which means they often dismiss out-of-hand any solution which is not the "unix way". They seem to eventually come around but it often takes a long time.

Overall I think that Open Source development model insures that Linux is reliable, works great on many hardware platforms and is well optimized. What it does not insure is that the end-user gets a good experience.

I recently got a Linux box set up. The stock RedHat 6.1 installation left me in tears with how pitifully broken the end-user experience was. Even simple things weren't done, like making sure that some application will open and play an mp3 after you click on a link on web page. And then there's the brain-dead behaviour of the UI in so many instances (e.g. the file panels, all 37 of them). All these sorts of things conspire to make me terribly frustrated with Linux as a whole (even though Linux is really just the kernel). I'm hopeful that the folks at Eazel will make a difference here (I know they've got some top-notch folks working on the problem so it's a good bet they will).

As for Open Source apart from Linux, I think it works great to help insure that a piece of code is the most optimized or that it works on the most variants of hardware or that it's secure. But it's hard to build a business around Open Source. I know everyone will start yelling "RedHat" and "Cygnus" but they're missing the point. I have yet to see a company that is turning a profit on a

good, easy-to-use end-user application that they give away under the terms of OpenSource. If a company can't make money then they aren't going to spend the time to develop something. And if a concentrated group of top-notch engineers don't spend the time to develop an end-user application, you're going to wind up with a poor excuse for an application with a bunch of half-completed features, missing functionality and an overall ".org" feel for the user (i.e. not professionally done).

OpenSource works great for something like gcc where the knowledge to be able to modify it for a new platform is sufficiently arcane that not many people in the entire world can do it. That means that Cygnus (now RedHat) just has to hire the bulk of those people and can then charge a hefty premium (some might say extortion) to make those changes for you.

OpenSource doesn't work well for applications that don't require lots of support and customization. For example, let's say you develop an open source e-mail application. If you've done your job, people will be able to install it and start using it without any extra help. If they don't come to you for help and they don't need it customized (since the base version has the features they need) then you don't make very much money. Sure you can charge for support but will that generate enough revenue to support an entire company? I doubt it (and I have yet to see a company that is doing well with this approach). It takes time, money and good engineers to develop a GUI app and I'm sorry but a couple of kids in college can't do it in their spare time.

I like sharing code and giving things away and I truly believe that the world would be a better place if we could all leverage off of each other's work but cold hard economic realities get in the way... maybe someday it won't be like that but for now it sure is.

How about the anti-trust ruling for Microsoft? Way off base or well deserved? What do you think should happen next?

I've been on the receiving end of their strong-arm tactics. I think they need to be spanked, hard. I don't care if they bundle their browser or not but their anti-competitive tactics go too far. I don't know if it would be effective, but an advisory board that oversees their business practices might be one solution (however it could probably be circumvented too easily). Breaking them up will probably do more harm to consumers than good... the status quo is not good though.

Will we ever see your homepage return, more specifically the "fire, explosions, and antics" section? I'd be glad to put a mirror of it on the extra space I have in my home account...

I'm not sure... maybe but honestly it was really out of date. It was kind of funny originally but it's over 4 years old now...

Now you're a Senior Project Manager at Google. What are you working on?

I'm working on managing a few projects that are mostly internal and not externally visible. Things like log analysis and performance.

What plans does Google have that frontwheeldrive readers might be interested in?

Hmmm, I think the big thing is that Google wants to be the premiere search service on the web. We're working very hard to make sure that we give the best results of anyone on the web. It's fun to be a part of that.

04182000

[home](#) | [interviews](#) | [reviews](#) | [mentors](#) | [about](#) | [contact](#)

© 2002 frontwheeldrive.com

