§ 1

Cover letter

To:
Fredrick R, Brennan
597 North Raleigh Avenue
Atlantic City, New Jersey 08401-1081

Dear Mr. Frederick *[sic]* R. Brennan,

I thank you so much for writing me. I apologize for the delay, and for it being so long.

The length is the reason for the delay in my response. I hope that OCR technologies are effective at saving you from having to type this in. Could you carefully proof the OCR results? I apologize for the burden it will be. I hope you understand the length as my taking your request seriously, and are pleased rather than dismayed. It was alot of work to write it. If you sign up for text messaging at `gettingout.com`, you can get faster responses from me. Sending a phone number will also work.

Please use your judgment in where to send this—any place that would be interested is fine.

LKML and `Slashdot.org` seem like reasonable places to send it (as of 2006). Your advice is desired.

Please let me know if you or anyone else has questions. If after sending this somewhere you still have time, could you send me info on `Reiser5`, or any interesting papers on other Filesystems, compression (especially Deep Learning based compression), etc.[1]

Gratefully,
Hans Reiser
11/26/23

[1] P. S. Someone told me this contains grammatical errors but declined to tell me where they are. `:-\`
If you are willing to edit this, please do.

From: Hans Reiser
(architect of ReiserFS V3 + Reiser4)

26 November 2023

§ 2

Introduction

I was asked by a kind Fredrick Brennan for my comments that I might offer on the discussion of removing ReiserFS V3 from the kernel. I don't post directly because I am in prison for killing my wife Nina in 2006.

I am very sorry for my crime–a proper apology would be off topic for this forum, but available to any who ask.

A detailed apology for how I interacted with the Linux kernel community, and some history of V3 and V4, are included, along with descriptions of what the technical issues were. I have been attending prison workshops, and working hard on improving my social skills to aid my becoming less of a danger to society. The man I am now would do things very differently from how I did things then.

Perhaps some might accept my apology; others might learn from my mistakes if I describe them well; some might find the design issues interesting.

I will leave it to the users to decide whether ReiserFS V3 is still useful. Users should understand that it is a burden for those who maintain VFS and the like to have to test their changes on an additional filesystem, especially given Linux filesystems are hard code at the VFS layer.

ReiserFS 4 provides a more maintainable basis for the future for those users who like the features of V3. If V3 isn't used it should go, I trust the users and the kernel maintainers to discuss whether it is used, and to make the right decision together.

§ 3

Main text

V3 had a moment in time when it was useful, and I am happy that we were able to contribute to the success of GNU/Linux for a few crucial years during which it was growing rapidly in usage. Chris[page 2 follows] Mason's contribution of journaling was the most practically useful feature of V3, and I thank him for it.

I am sad that SUSE didn't make it in the market place, I found it to be a well-crafted distro, and it was a privilege to be able to contribute to it. I am grateful for their sponsorship and support.

Reiser FS V3 was our first filesystem, and in doing it we made mistakes, because we didn't know what we were doing. I have to tell you that when I did the first benchmarks the performance was terrible, and I didn't know why. By terrible I mean that no sane person would use it for anything, there were years of dark depression before it was debugged enough to run at all, and then, 5% here and 5% there, I dragged it into being a little faster than the competition, and saving some space for traditional filesystem sizes, and more space for small files. Changing the allocation of blocks to files—simple code fortunately—yielded most of the performance gains. In V3 performance tweaking, Ext2, the existing GNU/Linux filesystem, was actually a very high performance filesystem—probably the best in the world.[1] The man I was then presented papers with benchmarks showing that ReiserFS was faster than ext2. The man I am now would start his papers [page 3 follows] crediting them for being faster than the filesystems of other operating systems, and thanking them for the years we used their filesystem to write ours. Not doing that was my first serious social mistake in the Linux community, and it was completely unnecessary.

Vladimir Saveliev, who had pity on me and came back after everyone else had quit, was the man who made the code work well enough that anyone would want to use it for more than a benchmark. After he came back, I left the debugging to him. He is one of the most good, mild-mannered, and

[1] I have never bench marked Apple's FS though.

hard working men that exists, anyone who has ever worked with him will agree with that. He didn't say much about it, but he believed in the Free Software Movement. Through force of will, and hard work, he made himself into a programmer of extraordinary skill, the effects of which you can see manifested in our Reiser4 code. He went from being the must junior of the programmers at the start of V3 to being the lead programmer, earning it through hard work all the way,

Assuming that the decision is to remove V3 from the kernel, I have just one request: that for one last release the README be edited to add Mikhail Gilula, Konstantin Shvachko, and Anatoly Pinchuk to the credits, and to delete anything in there I might have said about why they were not credited. It is time to let go.[page 4 follows]

In prison I have been working quite hard on developing my social skills, especially my conflict resolution and conflict avoidance skills. There is a lot of conflict in prison, as you can imagine, and it is quite a good place to learn those skills. Nothing like lots of practice, and the groups they let us take if we want to have a quite well developed curriculum, Repetition helps, at least for me. It has changed me.

I had a tendency to see people in extremes. That I am working on by being mindful of it, and by being around people who it would be easy to see in extremes, Many of them have become very good persons *[sic]* since their crimes.

I look back, with the advantage of the passing of years helping to improve my vision, and I see that while I intended to be helpful to researchers in Russia living on less than a hundred dollars a month, I could not be as helpful as a job working in America earning a Western salary. I put everything I had into the project, working 40+ hours a week at a day job, most employers not happy that I wanted to work only 40 hours if I could, and then I spent 15-20 hours a week arguing over algorithms, architecture, code, etc., by email. I learned to cut out everything in my life besides the[page 5 follows] project because otherwise my dream just would not make it.

I had more dream than experience. With Mikhail there running things in Russia it went pretty well because we had an ability to understand each other, I believe Mikhail Gilvla was the brightest mind in his generation of computer scientists, and his talent was wasted. First it was wasted by Russia's economy making his success impossible, and then in America it was wasted by most of the database field not being able to understand that he was right in wanting to rewrite the basics of their field in the ways be wanted to because he was just so much brighter than they were. I don't think they could even understand why it mattered, what he wanted to do.

We both believed that relational algebra was a special case of something more general, something that was needed for "semi-structured data". He called his approach "set-theoretic", and implemented it in the form of a database. I had my own syntax that expanded the filesystem namespace, instead. We came up with our ideas independently, before we met.[page 6 follows]

I had the idea that the place to implement it was in a filesystem, and that the motivation for implementing it there was to evolve it into a namespace that would allow unifying all of the namespaces of the operating system into one namespace. I thought this would be the most important refactoring of code ever, and would increase the expressive power of everything in those namespaces. I drew analogies between the effect of roads and waterways on the development of civilization according to Adam Smith, and the effects of free trade on specialization thus wealth according to Adam Smith again, and the effects of unifying the namespaces on the expressive power of the operating system.

That was my dream.

I was not the only one with this dream. The term "namespace" is pretty much only used by people who share in that dream enough to use a term that implies an equivalency between databases, filesystems, DNS, etc. Rob Pike and Plan 9 are examples of work to increase the Filesystem namespace, My particular flavor of the dream was that I had a syntax that could expand the filesystem hierarchical namespace to handle "semi-structured data". That means queries that are constructed from primitives that can be combined into the equivalent of searche engine queries (unstructured associations), database queries (unordered ordered pairs to search tables), or filesystems (fully[page 7 follows] ordered names to search hierarchies), or things that are richer than any of those special cases. Now that was my dream! Sigh, It probably won't surprise you that I was a teenager when I came up with this dream, and its syntax for searching semi-structured data.

I apologize to the users that I never got to that dream because of my crime and my going to prison, and they never got to see any semantic enhancements to Reiser 4.

The most technically difficult task that was a prerequisite to someday implementing my syntax, a task that had to be implemented first to do things right, was to make the filesystem efficient for small files.

I was very sensitive from the start to the power of net work effects to drown efforts to shift paradigms, to introduce better ways to do things. Network effects were abstract, so I understood them better than the importance of believing that I can make friends and allies of people who start out hostile because I had not made them feel included.

Awareness of network effects was why I decided that the path to unifying the namespaces of the operating system started with enhancing filesystem semantics which started with creating a storage layer that could emulate a filesystem faster than any existing filesystem but could also be[page 8 follows] effective for storing small objects like databases are. Yes, it was arrogant to attempt that, but it seemed like it should be possible to do it. I had no idea how long it would take to get it right.

Mikhail advised me to use balanced trees instead of extensible hashing (what `dcache` uses). I would come to understand that locality of reference is the *sina qua non* of performance, and that balanced trees are the best of all tools for implementing clever ways of maximizing locality of reference, Mikhail had tried extensible hashing, and learned the hard way that trying to fix its problems leads to using balanced trees. Locality of reference doesn't just affect hard drive performance, it affects compressed data performance (that means SPRAM performance), and even affects CPU cache performance (and thus DRAM stored data performance). He told me he was saving me 2 years by telling me to use balanced trees, and that's true, and I thank him for that,

I never told Mikhail that Oracle had tried implementing a filesystem using balanced trees, and its performance was terrible leading to most insiders in the industry concluding that balanced trees performed poorly for filesystem File size patterns. I didn't tell Mikhail; I didn't tell any of the programmers; I didn't tell anyone involved in the project. I couldn't see why they should be slower so I disregarded it, Vladimir would[page 9–1 follows] have good cause to be angry with me if he reads this.

I think part of their performance problems were how they did their journaling. I've observed that most balanced tree implementations are too synchronous, and most filesystems aren't very synchronous for performance reasons. `fsync()` is like using a sledge hammer to turn a screw with repeated sideways blows on the hoad of the screw. What is needed to have high data integrity guarantees with no unnecessary performance losses is a new API that allows intelligence in doing it by allowing the different layers to share their greater intelligence with the other layers. We need to allow the user to share their intelligence with the application, the application to share its intelligence with the filesystem, the filesystem and process scheduler to share their intelligence with the I/O scheduler, the I/O scheduler to share its intelligence with the I/O scheduler and other aspects of the hard drive, and then share intelligence back up the other direction of that stack, and then also tie in the memory manager in there too. [2] It requires not just ordering of I/O's, but specifying groupings of I/O's that should commit or not commit as a group, to communicate that intelligence, and not be more synchronous than necessary. Other features relating to priorities, fairness, guaranteeing I/O rates, etc, would also be desirable. If more details are wanted by anyone, ask, please, This is an area where Reiser 4 should do more[page 9-2 follows] than we had time to do.

I wish I had had the money to retain Joshua MacDonald when Orade hired him away from me, and learn more from him about logical journaling ideas he had then, I hope he is doing well—he was simply a brilliant young

[2] `fsync()` being used for data integrity is API ossification setting in in the 70's, and no one having the social courage to bring people together to figure out how to best support each other.

man, I'll just say that block aligning journaling isn't necessarily optimal for all needs that userspace has, and refer the reader to him for more. What I didn't know when we were arguing over the V3 design is that the FFS bench marks are misleading, and fail to highlight something that also hurts database performance, FFS is the BSD filesystem, and it was regarded as the best of its day. It used 4k blocks except for the tail end of files for which it used 1k blocks, and it would combine those 1k blocks from different files into the same 4k, thereby allowing them to increase block size but still conserve space.

Sounds clearly correct, yes? Alas, the performance cost of pulling the tails of files out of line with the rest of the file to combine it with the tails of other files is a seek plus a rotational delay (20 ms), That's hugely expensive compared to the tail size divided by the transfer rate, Seeks dominate filesystem performance unless the layout is perfect. You can see the importance of avoiding adding any seeks by having the tail of a[page 10 follows] file located inline with the rest of the file,

Databases with their "BLOBS", ReiserFS V3, and FFS all combine tails in a way that adds seeks and hurts performance substantially. For Databases and Reiserts V3 it's even worse though, because BLOBS make it not really a balanced tree, and the ability to effectively cache all the internal nodes of the tree is lost. Google "Reiser 4 twigs", and hopefully that will find a longer discussion of that for interested readers, including how to fix it,

That performance failing drove me crazy—I felt the design was just wrong, and I resolved to redo everything from scratch with Reiser 4, The plugins, and the modularity of the code they created, are the most important feature of Reiser 4, It is far easier to add features to Reiser 4 than to add them to other filesystems, because you just add new plugins. The hard stuff is done, and new features are a downhill ride. Well, except that I am in prison, and so I must leave all that to others.

There is a problem that filesystems have, and that is that format changes are unwanted by many for good reasons, That is the primary reason that filesystems stagnate, Well, that, and stagnating is easy...[page 11 follows]

I just had to fix all these flaws, fix them and make a filesystem that was done right. It's hard to explain why I had to do it, but I just couldn't rest as long as the design was wrong and I knew it was wrong.

SUSE didn't want a format change, they wanted incremental improvements to V3. That's the way it is for a lot of filesystem architects. Incrementally changing things they know are deeply wrong to their cores, but stuck in that misery of doing so. I said no to that, which I won't apologize for and don't regret, because I have a soul, and I had a dream. What I do apologize for is how utterly inarticulate and unsociable I was in explaining that to SUSE. What I essentially said to SUSE was that the code was unmaintainable terrible code that needed to be rewritten from scratch, trust me, it needed to be done, we didn't know what we were doing then and now we have learned what we should have known, but we will fix every bug in V3 that is found.

I could have said that I hear them, and I hear them well, so well that Reiser 4 has node format plugins that solve the format change problem. I could have told them of our plans for a repacker for Reiser 4, so that partitions could be shrunk, and layout perfectly optimized, and …spent a day visiting them and making Reiser 4 our dream not my dream. Instead I[page 12 follows] communicated that I couldn't pursue my dream without a format change. Both SUSE and I wanted what was best for Linux and SUSE, so the reason for my failure was that I had failed to socially connect by reaching past the initial hostility to the format change to I make my (and DARPA's) dream into our[3] dream. I would repeat this failure style with the Linux kernel community when Reiser 4 was submitted for inclusion, only worse.

Let me go back in time though to the early days of V3. Not long after the computers were purchased, and then the project really got going, Mikhail got a job in America, Mikhail was the reason I had hired this group. Now I no longer had somebody running things on site that I was intellectually compatible with, and frictions developed. I had not chosen them, and they had not chosen me, nor had they chosen my dream or the Free Software Movement. Alas, I was inexperienced, and they didn't respect. that, which is understandable, yes? I had a feel for abstractions that was stronger than my feel for people, and a casual delight for tossing socially established algorithms that made them even less comfortable than my wearing a cowboy hat instead of a tie, None of this was a problem with Mikhails but Mikhail had left.

It was to be expected that there would be problems, but I was too young and hopeful to see that. There was a cost to going from system administrator to architect without first spending a few years as a coder.[page 13 follows]

An example of this was their use of BLOBS in V3. I didn't know that putting unformatted nodes on a level of the tree below the putative leaves of the tree was the standard way of doing things in the database world. When I saw it in the code I objected to it. I was asked to just let them try it their way, and if it didn't work it could be removed , I was too inexperienced as a software project manager to know that that was an argument to only entertain after the first version of a product has shipped, because whatever gets written before product shipment is not likely to get rewritten until after product shipment, especially if you know the design is wrong. That's especially true if you know the design is wrong. What I didn't know was just how wrong the design was, The performance cost was higher than I feared it would be, I should also say that the coding cost of doing it right was higher than I thought it would be, To the young architects out there, let

[3] including SUSE's

me say that if you know something is wrong, don't lot it happen just to be agreeable or to accept the consensus, Yes, listen to all the arguments, but if you don't have a better feel for algorithms, you should not be in the job of architect. Believe in yourself, I paid the price for not knowing when to be firm, when the benchmarks came in, and then again when fixing it required a format change that lost me SUSE. [4] [page 14 follows]

Then Mikhail's employer hired the rest of them, and left me with a pile of poorly commented code that was not able to run. It would have been wiser for me to write it again from scratch, but that would have meant admitting that all I had invested into the project was a total loss.[5]

Now, with the distance of time, I can see that their leaving was simply the reasonable thing from their point of view, There were some unkindnesses in how they left, but that's to be expected, says the distance of time, Then I felt so betrayed. Now I wish to let go of that. They were just ordinary people caught in economic forces, and not enjoying working for the guy with more dream than experience. They too were inexperienced: none of us had worked on a filesystem before. We were all young, and impatient. I never did get to where we made enough money that I could pay people well, and I am truly sorry for that. If I had not committed my crime, that day when Reiser 4 was a worthy use of the extraordinary talent of the programmers working on it probably would have come. I was callous and indifferent to their needs and dreams when I committed my crime, and victimized them financially and ruined their dreams that I had talked them into.

One of my great regrets is that I let go of Mikhail as a friend. I hope be is alive, and doing well.[page 15 follows]

I read about a DARPA request for proposals for open source software on a Friday on Slashdot, and it had a Monday submission deadline. I went into a writing Frenzy, and proposed what became Reiser 4, DARPA was very good to us, very and I learned a lot about accounting and security both from them. I'd like to apologize to DARPA for two things:

1. All we were able to complete was the infrastructure that would enable the security features I proposed, and that was despite my putting several times the money they gave us into the project from every where I could get it, including using 29% interest credit card cash advances to make payroll near the end. [6] Now that Reiser 4 is stable, the plugin infrastructure would make it so easy to add extra features, we could

[4] *[struck through:]* They wanted to use BLOBS because everyone else did, so it seemed safer than trusting in my feel for algorithms that was saying "don't do that."

[5] This, by the way, is why I am a comment the code extremist, which you can see if you read Reiser4.

[6] 29% means that a debt doubles in 2.5 years. I wonder if I will get out in time that the ≈$984,000 debt will be only $20 million, or if it will be over $100 million by the time I get out.

probably have done the encryption plug in and `stat` data inheritance in just 6 months.[7]

2. At the time I made the Reiser4 proposal it looked like democracy had won in Russia, I believe we were a point of light in US–Russian relations, The problem was that what was needed was a thousand points of light, and we were one of a very few.[page 16 follows]

Russian Intelligence/Law Enforcement, after doing a very thorough job of making sure we weren't a CIA cover operation,[8] was very nice to us, and quite supportive of us and the Free Software Movement, It might not have been a coincidence that we had no Mafia problems, The time that I got scammed for $300, they got my money back in 45 minutes. I know that one is not supposed to say good things about them, but my own personal experience is that they were very effective, kind, and even wisely guiding. In important ways,

Russian culture teaches a better understanding of people. As a for instance, one of them told me that my wife was in a lot of pain. Now I can see clearly that that was exactly correct, and perfectly illuminated the path I should have taken. Alas, I lacked the wisdom to understand the words with just one repetition.

*9

In prison I have learned that alienation, which I intend to use to protect myself, is often a cage I bulld around myself instead. A fear of being naïve can often be as dangerous in its reality distortions as being naïve,[page 17 follows] Now is not the time to try to be a light in US-Russian relations. I hope that there will again be such a time, and an end to all the death and dying. I hope the time will come again for reaching past the alienation, and finding friendship[page 18 follows] and love, between Russia and Ukraine and the U.S, and between my children and I. Call me a fool, or simply someone who thinks the pain is too much for us all, I leave that to you.

Back to Reiser4: We[10] built a beautiful filesystem that embedied everything we had figured out we should have done the first time, called Reiser 4,

[7] `stat` data inheritance is important to small files being actually small, file body inheritance plus stat data inheritance allows using the filesystem namespace to access objects within a file as either a whole or individually, and to merge the likes of config files into the filesystem namespace in a backwards compatible way, For more, consult your imagination, `:-)`

[8] When I read the book *Body of Secrets*, I realized we looked exactly like a CIA cover operation.

[9] *[struck through]*: If I could wave a magic wand to change the American divorce process, I would require that each person getting a divorce be asked at the start; "Is it possible that your spouse is in a let of pain, and could that explain their actions?" I call that "inviting empathy", and now that I know to try it, I use it a lot in conflict resolution and mediation,

[10] A whole team of us: Vladimir Saveliev, Alexander Zarochentcev, Nikita Danilov, Vladimir Demidov, testers, Alexander Lyamin being our networking sysadmin, and more.

and it smoked, What's more, it had this plugin architecture that I had proposed to DARPA in that writing frenzy, but that Nikita Danilov had made even better than what I had proposed (he made the balancing operations plugins too, and I am so glad that he was smarter than me and showed that the performance costs were acceptable).

With most filesystems, adding a feature is 80% altering the existing code and 20% writing the new feature, and then you will be likely adding a Format change that someone not even working in the filesystem group will say no to. With Reiser 4 we made every aspect of it that we could imagine doing so for into a plugin. If you want to add a new feature, you just spend your time writing the new plugins for it, and 90% of the time that is all you will have to write, or failing that, it will be 80% of what you will have to write, You get to spand your time on your clever idea instead of on why it was so much harder than it should have been to write it. I think you'll find it is more than 3 times faster to add it to Reiser 4 than to any[page 19 follows] other filesystem, with all the features required of a filesystem for compatibility (an enormous burden to write—if you have a clever filesystem idea save yourself that burden and add a plugin to Reiser 4 instead—you can be one programmer doing a 6 month plugin instead of having to fund a team for 5 years to do a filesystem) completed, it was going to be all downhill from there. We would have been theones implementing a substantive new feature per programmer per 6 months, The code that Alexander Zarochentcev, Nikita Danilov, and Vladimir Saveliev wrote (all three of them contributing equally, working together like the three musketeers) was beautiful code, and then the junior programmer Edward Shishkin came along as the fourth musketeer and his compression plugin doubled performance again for compressible files. I cannot remember ever finding anything I could improve in Alexander Zarochentcev's code: it was always perfectly written: "read it and learn how things should be written" kind of code.

I am not known for being unable to find things to criticize, or praising easily. Somehow with the smallest budget for paying them in the filesystem world, I lucked onto the best programming team in the filesystem world. The filesystem was worthy of their talents—I wish that I as a person had been worthy of them.[page 20 follows]

The problem was that it didn't use the code that had been written by others in the kernel community, and people don't really like their code not being used. People want to feel included. I responded to their social need by, well, screwing the pooch in response (benchmarks and disputing their expertise). Imagine if I had responded by saying that I needed their help in imagining new file plugins instead?

You know how people are much more likely to read an email if it is one screen long, rather than the length of this `:-/` ? It is similar with contributing code to the kernel. It is much more social and relationship developing to contribute a screenful or two of code once every week or two over the course of years. We were dropping 90,000 lines of code on them all at once, having worked on it in total social isolation for 5 years in Moscow, Socially it was all bad. Small increments are the more social way to go. Incremental improvements to V3 would have met no opposition. We could have lived a life of being a little bit better than `ext3`, and been respected in the field as we waited for someone young to obsolete us,

Alas,[11] it had to be written from scratch to be written right, to be written the best I knew how to to design it, to fix what the benchmarks had [12] revealed to an empiricist. My leadership and project management failures needed to be atoned for. The maintainability of a plugin based filesystem was perhaps more important than the[page 21 follows] benchmarks, I had added comments, and we had added bug Fixes, but I am so glad I kept none of it for V4.

If I had had then the conflict resolution skills I have learned in prison Cognitive Behavioral Intervention classes I might would have been able to overcome all that.

Then, my attitude was, it's the fastest filesystem in the world, why aren't you happy and helpful?! Look at these benchmarks! These plugins! Why do I have to deal with these people who didn't write as fast of a filesystem ? Let them write theirs, and us write ours - VFS should allow that ! My attitude should have been, ignore the hostility, that's to be expected at first and overcome, I can overcome hostility, and the way to do that is simple : 1) make people Feel appreciated and cared about, 2) make people Feel included, 3) make people want to do things with those plugins themselves by asking them for their ideas on what plugins they could imagine.

Now I Know that it is possible to overcome such problems if I actively apply my mind to Finding an emotional or social path to making people Feel good.

The most important part of that is to believe that I can do that successfully. I used to lack the ability to imagine that I could succeed at overcoming hostility, but by doing the exercises in my Cognitive Behavioral Intervention classes in prison situations, I have started to see that I can,[page 22 follows] started to believe that I can,

If you believe you can do it, you usually can, when it comps to making things go well with others, If you focus on Finding a way past a problem, instead of on your Feelings of having been wronged, you can usually get past a relationship hiccop. If you Fail, you've lost little to nothing in making the attempt, so why Fear to make the attempt? Contrary to what young men so often Fear, if you Fail in such an attempt you don't lose Face - you gain respect From others watching you try - especially the older ones. If you

[11] *[struck through]*: I have a soul, I had a dream, and

[12] *[struck through]*: shamed me to the soul with

make a habit of always trying, you will inevitably get good at it,
*[13]

One of my dreams is to someday convince the State Legislature to teach the curriculum they teach us prisoners, in elementary school, that people like me can learn it better without having to go to prison to learn it, I'm try my to convince some people to pitch it to legislators—if any one would like to help with that please let me know. It will help with more than just avoiding prison—it will help with all relationship conflicts, and who does not have those...? The prison parenting class should be taught in High School too...[page 23 follows]

An example of what I could have hand led differently was when Viro announced that there was a race condition in our code, but that he would not tell us where it was because if we could not audit our code for race conditions we should not be allowed into the kernel, Viro was this guy whose career focus was lacking, I didn't know any thing about auditing code for race conditions, and disappointingly Google didn't get me any where in looking for how does one audit for race conditions. My attitude at the time was that it sounded like necessary but tedious work, really tedious, that hopefully the guys who were doing the debugging would figure out while I tried to get the money for their next pay check. [14] I should have done more than communicate the problem, I should have asked "what is the audit for race conditions plan, and help me find literature on how to do it, and then let's divide up the relevant code and do it, all code to be audited by two different people". Race conditions are very expensive to debug when they are rare—auditing would have saved money. This was an area where we all needed to improve our methodology. [15]

I Failed to communicate by Failing to ask if this was an area where my guys didn't know how to do it and were too shy to say so, and in this were just like me.[page 24 follows]

What I should have done was ask Viro instead of Google, and invite him to come to Moscow, stay in my spare room, sample the world's wildest nightlife (Moscow), and give our team a seminar on auditing for race conditions and supervise us as we each took a part of the code and did the audit per his instructions, We had as much motivation to eliminate race conditions, no, much more because it was our code and business, as Viro, There was no good reason for us to not be allies in this. There was only

[13] *[struck through]*: These are the things I have learned in Cognitive Behavioral Intervention exercises and applying them to prison situations, as I seek For a possibility of redemption someday.

[14] We were using 29% interest credit card cash advances to pay salarios at the end, but I had some deals that would have made us prosperous again about to take effect except that I committed my crime and it all fell apart.

[15] Has anyone semi-automated that auditing? GPT? I remain curious about how to do that audit right.

a bad reason I was responding to the initial hostility rather than reaching past it to make an ally of someone who had a lot of potential to be very helpful to us. Now of course he might well have refused the invitation, said he was too busy, told us to get better at using Google, etc., but even if he refused the invitation he would probably have felt mollified somewhat. If he took us up on the invitation all of us could have made valuable Friendships likely to have been useful for the rest of our lives, as we fed him shi *[sic]* (щи, Russian soup, one of the best aspects of Russian cuisine is their soups) and pelmenie (პელმენი, Georgian spicy pot stickers), and took him to dance the night away when the race conditions were gone.

I could have similarly approached other Key persons in the Kernel community who expressed unhappiness at our contribution offering, if only I had had then the social confidence, and belief that it is possible to overcome initial touches of alienation, that I have now developed.[page 25 follows]

Instead I responded to hostility with my own hostility.

In prison, on MLK day, I learned of MLK's words:

Only love can fight hate.”

I have come to appreciate, and fully understand those words, I wish I had understood them then.

There were a bunch of such situations that I handled in ways that did not make people Feel appreciated or included, and I want to take this opportunity to apologize For those.

I especially want to apologize to the other members of our team who invested so much of their lives into our dream only to be failed by me, and by my alienating others in the Linux kernel community.

The Linux kernel is not about benchmarks, it is about a community of people who enjoy working together in the Christmas Spirit to give to the users all year long. Now that I have changed who I am I can better see that.

I don't know what is in Reiser 5—I haven't been told, and I cannot go on the Internet. Edward Shishkin is a very bright man though, and one of my regrets is that I didn't spend more time with him, I am confident he has done some thing nice in Reiser 5. Who knows, maybe he has done some nice plugins that I would never have imagined. The compression plugin Edward coded was the one thing yielding the biggest performance boost of all the things we did in Reiser 4. Chances are high that I won't be[page 26 follows] released anytime soon. I encourage people to allow those who worked so hard to build a beautiful filesystem for the users to escape the effects of my reputation. I invite you to empathize with what this has been like for them for a minute.

Let their dreams escape from the harm I have done, if that feels right to you.

§ 4

Conclusion

I wish I had learned the things I have been learning in prison about talking through problems, and believing I can talk through problems and doing it, before I had married or joined the LKML. I hope that day when they teach these things in Elementary School comes.

I thank Richard Stallman for his inspiration, software, and great sacrifices,

It has been an honor to be of even passing value to the users of Linux. I wish all of you well.

Hans
11/26/23

P.S. Letters are welcome, Please send them to:

Hans Reiser
CHCF G31008
P.O. BOX 213040
Stockton, CA, 95213
U.S.A.

You can also send texts. or video chat or phone about, if you go to `gettingout.com`, or phone chat if you send a phone number, for me to call and accept the phone call.