

---

Subject: Re: Capture division by zero

Posted by [mirek](#) on Tue, 27 Oct 2020 08:23:49 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

koldo wrote on Fri, 16 October 2020 09:15I would like to capture floating point errors.

For now in case of error the program execution follows, but you can find in the doubles, things like INF or NAN.

This behaviour is very conservative for my apps, and I'd prefer in some situations to capture these situations stopping the execution.

I have tried it unsuccessfully using:

- signal(SIGFPE, ...)
- \_controlfp\_s()
- \_set\_se\_translator()

Any help will be acknowledged!

PD. Somebody could say "just check that all data is adequate before using it in division, sqrt, ...". That's true, but difficult and cumbersome in some situations using uncontrolled data sources where sometimes even doing lots of "if" to check the consistency of data, that's not enough.

Ah, that explains it. So you have like 2 weeks of experience with this issue

I was all hurray 3 years ago into catching all FP exceptions and avoiding all NaNs. Not so sure today....

Just consider this: Historically, FP exceptions were default and always active, divide by zero or negative sqr always stopped the code just like dereferencing NULL. Then IEEE basically replaced this behaviour with NaNs as default. Maybe they knew something?

If you want a very simple contraexample where NaN is definitely better, look at this:

<https://www.codeproject.com/Articles/1074135/Evaluating-expression-with-descend-parser-and-Uplu>

Mirek