Subject: Re: Capture division by zero
Posted by Klugier on Mon, 26 Oct 2020 22:51:46 GMT
View Forum Message <> Reply to Message

Hello Koldo,

Influential people think that divisions by zero or negative square roots are admissible in a good code. I do not agree.

This is not the problem. Error handling or problematic value handling is very important. In production software you should always do that without compromising. The problem with "CrashHandler" is that it forces error handling to all package consumers. It is not even an option - you just create global variable:

```
static CrashHandler crash;
```

Library or component provider should avoid it at all cost. You should in the documentation tell that certain function could raise exception or returns error value. If your code could read external value and then it could cause problem you are obligated to validate it and in case of problem do something with it.

At the beginning of the thread I suggest to use static analyzer to find all this divided by zero problems and handle it properly. This is the best strategy for library and professional software. I agree that minidumps are extremely helpful to identify problems on production, but it must be handled on application level - never on library level.

```
/**
 * PUBLIC API DOCUMENTATION
 * In case when divider is 0 std::runtime_error is thrown.
 */
int Calculate(int divider)
{
    if (divider == 0)
    {
        throw std::runtime_error("Calculate(): Divider can not be zero...");
    }

    return 20 / divider;
}

// Now the client could deiced what to do with the exception - it can be handled locally or globally in main function, but it is still the consumer choice what to do with it
// In case of global handler provided by library the decisions were taken from application creator.


// ... Below line may even decrese the performance, because on each new allocation NewHandler
```

will be called.
std::set_new_handler(NewHandler);
std::set_unexpected(UnexpectedHandler); // <- What if somebody defines this handler and you overriding it - you should check this at least...

To be clear my overall goal is to pick up the quality of our packages. Never decrees or compromise on error handling, so you are my ally.

Klugier