C Scoped Enum Enhancements

ALEXANDER CHRISTENSEN

Contents

1 INTRODUCTION

2 MOTIVATION AND SCOPE

The initial motivation for this proposal was the lack of a good way in the standard library to convert an enum value to a proper string representation. Very often, I have found a need to log an enum for various purposes, and every time I create a new enum type I have to write such a function again. An example:

```
enum class GraphicsApiType { none, opengl, vulkan };

const char* get_api_type_string(GraphicsApiType apiType) {
    switch (apiType) {
    case GraphicsApiType::none:   return "none";
    case GraphicsApiType::opengl: return "opengl";
    case GraphicsApiType::vulkan: return "vulkan";
    default: return "<unrecognized>";
    }
}
```

This is cumbersome to maintain, for every time a value is added or modified inside the enum type, this other function has to be modified as well. But besides this, a potential *run-time* error may be introduced when an invalid enum is provided.

A session of browsing around sites such as StackOverflow revealed sometimes quite vividly imaginary answers for how to circumvent this limitation in the language:

(1) https://stackoverflow.com/questions/11421432/how-can-i-output-the-value-of-an-enum-class-in-c11
(2) https://stackoverflow.com/questions/1390703/enumerate-over-an-enum-in-c
(3) https://stackoverflow.com/questions/6281461/enum-to-string-c

Author's address: Alexander Christensen, alex_c007@hotmail.com.

(4) https://stackoverflow.com/questions/201593/is-there-a-simple-way-to-convert-c-enum-to-string

3 IMPACT ON THE STANDARD

4 DESIGN DECISIONS

5 TECHNICAL SPECIFICATIONS

6 ACKNOWLEDGEMENTS

7 REFERENCES