LEWG policy on handling errors with OS resource creation

Document #:	D1977R0
Date:	2019-12-28
Project:	Programming Language C++ Library Evolution

Reply-to:	Elias Kosunen <isocpp@eliaskosunen.com> Jeff Garland <jeff@crystalclearsoftware.com> Niall Douglas <s_sourceforge@nedprod.com>

1 Introduction

There are currently multiple proposals in the library evolution pipeline, including [P1031R2] and [P1883R0] *Low level file i/o library* and [P1750R1] *A Proposal to Add Process Management to the C++ Standard Library*. During the review of this papers by LEWGI in Belfast (November 2019), it became apparent that it is presently unclear how errors with creating operating system resources are to be handled. Encouraged by LEWGI, this paper attempts to establish clear guidance on this subject matter, for now and for the future.

Once decided, we propose for this policy to become a part of the future LEWG design policy standing document.

2 Overview of present proposals

2.1 Process management (P1750)

2.2 Low level file i/o (P1031 and P1883)

3 Design alternatives

3.1 Factory function returning a struct containing an `error_code`

3.2 Factory function returning an `expected` (P0323)

[P0323R9]

Includes hand-rolled error

3.3 Constructor taking an `error_code`

3.4 Throwing exceptions

4 References

[P0323R9] JF Bastien, Vicente Botet. 2019. std::expected.
https://wg21.link/p0323r9

[P1031R2] Niall Douglas. 2019. Low level file i/o library.
https://wg21.link/p1031r2

[P1750R1] Klemans Morgenstern, Jeff Garland, Elias Kosunen, Fatih Bakir. 2019. A Proposal to Add Process Management to the C++ Standard Library.
https://wg21.link/p1750r1

[P1883R0] Niall Douglas. 2019. Walkthrough of P1031s file_handle for LEWG-I.
https://wg21.link/p1883r0