N3275: fopen_s "p" and bring fopen_s's mode closer to POSIX 202x

Document #: N3275
Date: 2024-06-13
Project: Programming Language C
Reply-to: Niall Douglas
<s_sourceforge@nedprod.com>

In the June 2024 meeting, WG14 made changes to `fopen()` and requested matching changes for `fopen_s()`. These are those changes.

Contents

1 Proposed improved wording

1.1 K.3.5.2.1.5

The mode string shall be as described for `fopen`, with the addition that modes starting with the character 'w' or 'a' may be preceded by the character 'u', see the following:

~~uw truncate to zero length or create text file for writing, default permissions~~
~~uwx create text file for writing, default permissions~~
~~ua append; open or create text file for writing at end-of-file, default permissions~~
~~uwb truncate to zero length or create binary file for writing, default permissions~~
~~uwbx create binary file for writing, default permissions~~
~~uab append; open or create binary file for writing at end-of-file, default permissions~~
~~uw+ truncate to zero length or create text file for update, default permissions~~
~~uw+x create text file for update, default permissions~~
~~ua+ append; open or create text file for update, writing at end-of-file, default permissions~~
~~uw+b or uwb+ truncate to zero length or create binary file for update, default permissions~~
~~uw+bx or uwb+x create binary file for update, default permissions~~
~~ua+b or uab+ append; open or create binary file for update, writing at end-of-file, default permissions~~

`uw` truncate to zero length or create file for writing, default permissions
`ua` append; open or create file for writing at end-of-file, default permissions

The remainder of the string may contain any of the characters permitted by `fopen`.

1.2 K.3.5.2.1.6

Opening a file with exclusive mode (`'x'` as ~~the last~~ a character in the mode argument) fails if the file already exists or cannot be created. The check for the existence of the file and the creation of the file if it does not exist is atomic with respect to other threads and processes. If the implementation is not capable of performing the check for the existence of the file and the creation of the file atomically, it shall fail instead of performing a non-atomic check and creation.

1.3 K.3.5.2.1.7 (insertion)

Opening a file with private mode (`'p'` as a character in the mode argument) creates a file whose contents cannot be accessed by other means at any point in time including after program end, and whose `filename` may be ignored by implementations except to identify which storage ought to hold the file. If the implementation is not capable of creating private files, it shall fail.

2 References

[N2357] Stoughton, Nick
Change Request for fopen exclusive access
http://www.open-std.org/jtc1/sc22/wg14/www/docs/n2357.htm

[N2731] *C2x Working Draft*
http://www.open-std.org/jtc1/sc22/wg14/www/docs/n2731.pdf

[N2857] Douglas, Niall
C2x `fopen("x")` and `fopen("a")` v2
http://www.open-std.org/jtc1/sc22/wg14/www/docs/n2857.pdf

[N3059] Douglas, Niall
C23 `fopen "x"` and `"a"`
http://www.open-std.org/jtc1/sc22/wg14/www/docs/n3059.pdf

[N3060] Douglas, Niall
C2y `fopen "p"` and bring `fopen`'s `mode` closer to POSIX
http://www.open-std.org/jtc1/sc22/wg14/www/docs/n3060.pdf

[N3247] Douglas, Niall
C2y `fopen "p"` and bring `fopen`'s `mode` closer to POSIX
http://www.open-std.org/jtc1/sc22/wg14/www/docs/n3247.pdf

[POSIX.2017] *The 2017 POSIX standard*
https://pubs.opengroup.org/onlinepubs/9699919799.2018edition/functions/contents.html