85.13.7 Win32: sysopen, sysread, syswrite

As of the 5.8.0 release, sysopen()/sysread()/syswrite() do not behave like they used to in 5.6.1 and earlier with respect to "text" mode. These built-ins now always operate in "binary" mode (even if sysopen() was passed the O_TEXT flag, or if binmode() was used on the file handle). Note that this issue should only make a difference for disk files, as sockets and pipes have always been in "binary" mode in the Windows port. As this behavior is currently considered a bug, compatible behavior may be re-introduced in a future release. Until then, the use of sysopen(), sysread() and syswrite() is not supported for "text" mode operations.

85.14 Future Directions

The following things **might** happen in future. The first publicly available releases having these characteristics will be the developer releases Perl 5.9.x, culminating in the Perl 5.10.0 release. These are our best guesses at the moment: we reserve the right to rethink.

- PerlIO will become The Default. Currently (in Perl 5.8.x) the stdio library is still used if Perl thinks it can use certain tricks to make stdio go **really** fast. For future releases our goal is to make PerlIO go even faster.
- A new feature called *assertions* will be available. This means that one can have code called assertions sprinkled in the code: usually they are optimised away, but they can be enabled with the `-A` option.
- A new operator `//` (defined-or) will be available. This means that one will be able to say

  ```
   $a // $b
  ```

  instead of

  ```
  defined $a ? $a : $b
  ```

  and

  ```
  $c //= $d;
  ```

  instead of

  ```
  $c = $d unless defined $c;
  ```

  The operator will have the same precedence and associativity as ||. A source code patch against the Perl 5.8.1 sources will be available in CPAN as *authors/id/H/HM/HMBRAND/dor-5.8.1.diff*.
- `unpack()` will default to unpacking the `$_`.
- Various Copy-On-Write techniques will be investigated in hopes of speeding up Perl.
- CPANPLUS, Inline, and Module::Build will become core modules.
- The ability to write true lexically scoped pragmas will be introduced.
- Work will continue on the bytecompiler and byteloader.
- v-strings as they currently exist are scheduled to be deprecated. The v-less form (1.2.3) will become a "version object" when used with `use`, `require`, and `$VERSION`. $^V will also be a "version object" so the printf("%vd",...) construct will no longer be needed. The v-ful version (v1.2.3) will become obsolete. The equivalence of strings and v-strings (e.g. that currently 5.8.0 is equal to "\5\8\0") will go away. **There may be no deprecation warning for v-strings**, though: it is quite hard to detect when v-strings are being used safely, and when they are not.
- 5.005 Threads Will Be Removed
- The `$*` Variable Will Be Removed (it was deprecated a long time ago)
- Pseudohashes Will Be Removed