I notice a workaround for llvm.4898 in constant_p.h:
[link](https://github.com/ruby/ruby/blob/master/include/ruby/internal/constant_p.h)

```
/* Note that __builtin_constant_p can be applicable inside of inline functions,
   * according to GCC manual. Clang lacks that feature, though.
   * @see https://bugs.llvm.org/show_bug.cgi?id=4898
   * @see https://gcc.gnu.org/onlinedocs/gcc/Other-Builtins.html */
#include "ruby/internal/has/builtin.h"
```

```
/** Wraps (or simulates) __builtin_constant_p */
#if RBIMPL_HAS_BUILTIN(__builtin_constant_p)
# define RBIMPL_CONSTANT_P(expr) __builtin_constant_p(expr)
#else
# define RBIMPL_CONSTANT_P(expr) 0
#endif
#endif
/* RBIMPL_CONSTANT_P_H */
```

and a workaround for llvm.38095 in scan_args.h:
[link](https://github.com/ruby/ruby/blob/master/include/ruby/internal/scan_args.h)

```
/* NOTE: Use `char *fmt` instead of `const char *fmt` because of clang's bug*/
/* https://bugs.llvm.org/show_bug.cgi?id=38095 */
#define rb_scan_args0(argc, argv, fmt, varc, vars)  
  rb_scan_args_set(RB_SCAN_ARGS_PASS_CALLED_KEYWORDS, argc, argv, \  
  rb_scan_args_n_lead(fmt), \  
  rb_scan_args_n_opt(fmt), \  
  rb_scan_args_n_trail(fmt), \  
  rb_scan_args_f_var(fmt), \  
  rb_scan_args_f_hash(fmt), \  
  rb_scan_args_f_block(fmt), \  
  (rb_scan_args_verify(fmt, varc), vars), (char *)fmt, varc) 
...  
```

These bugs are already marked as fixed:
[link](https://bugs.llvm.org/show_bug.cgi?id=4898)
[link](https://bugs.llvm.org/show_bug.cgi?id=38095)

Shall they be removed?

---

**History**

**#1 - 07/16/2022 03:53 PM - Chandler (Chandler Hen)**

- Subject changed from Remove workaround for some fixed bug? to Remove workaround for some fixed bug (llvm.4898 & 38095)?

**#2 - 07/19/2022 12:40 AM - shyouhei (Shyouhei Urabe)**

Well I'm not against deleting those parts someday, but our CI still have clang-3.9 coverage. It seems we still need them.

We need to define supported versions of clang first. We do so for GCC (>=4). See #18839