Subject: Re: Make THISFN simpler and more powerful
Posted by Lance on Tue, 05 Nov 2024 01:25:15 GMT
View Forum Message <> Reply to Message

I had a brief discussion with regard to constexpr with Mirek a few days ago
here

Today I did a simple test with a more complicated case.

```
#include <Core/Core.h>

using namespace Upp;

CONSOLE_APP_MAIN
{
 Color c(200,50,76);
 if ( c == Color(200, 50, 76) )
 {
  RLOG("Yes");
 }else{
  RLOG("NO");
 }
}
```

Compiled to the following 338 lines of assembly code

```
 .text
 .file "TestColorConstant.cpp"
 .globl main                     # -- Begin function main
 .p2align 4, 0x90
 .type main,@function
main:                            # @main
 .cfi_startproc
# %bb.0:
 pushq %rbp
 .cfi_def_cfa_offset 16
 .cfi_offset %rbp, -16
 movq %rsp, %rbp
 .cfi_def_cfa_register %rbp
 subq $32, %rsp
 movl $0, -4(%rbp)
 movl %edi, -8(%rbp)
 movq %rsi, -16(%rbp)
 movq %rdx, -24(%rbp)
 movl -8(%rbp), %edi
 movq -16(%rbp), %rsi
```

```
 movq -24(%rbp), %rdx
 callq _ZN3Upp9AppInit__EiPPKcS2_@PLT
 leaq _Z14ConsoleMainFn_v(%rip), %rdi
 callq _ZN3Upp12AppExecute__EPFvvE@PLT
 callq _ZN3Upp9AppExit__Ev@PLT
 callq _ZN3Upp11GetExitCodeEv@PLT
 addq $32, %rsp
 popq %rbp
 .cfi_def_cfa %rsp, 8
 retq
.Lfunc_end0:
 .size main, .Lfunc_end0-main
 .cfi_endproc
                                # -- End function
 .globl _Z14ConsoleMainFn_v           # -- Begin function _Z14ConsoleMainFn_v
 .p2align 4, 0x90
 .type _Z14ConsoleMainFn_v,@function
_Z14ConsoleMainFn_v:                  # @_Z14ConsoleMainFn_v
 .cfi_startproc
# %bb.0:
 pushq %rbp
 .cfi_def_cfa_offset 16
 .cfi_offset %rbp, -16
 movq %rsp, %rbp
 .cfi_def_cfa_register %rbp
 subq $16, %rsp
 leaq -4(%rbp), %rdi
 movl $200, %esi
 movl $50, %edx
 movl $76, %ecx
 callq _ZN3Upp5ColorC2Eiii
 leaq -8(%rbp), %rdi
 movl $200, %esi
 movl $50, %edx
 movl $76, %ecx
 callq _ZN3Upp5ColorC2Eiii
 movl -8(%rbp), %esi
 leaq -4(%rbp), %rdi
 callq _ZNK3Upp5ColoreqES0_
 testb $1, %al
 jne .LBB1_1
 jmp .LBB1_2
.LBB1_1:
 callq _ZN3Upp6VppLogEv@PLT
 movq %rax, %rdi
 leaq .L.str(%rip), %rsi
 callq _ZN3UpplsERNS_6StreamEPKc
 movq %rax, %rdi
```

```
 xorl %esi, %esi
 callq _ZN3Upp6StreamlsENS_7EOLenumE
 jmp .LBB1_3
.LBB1_2:
 callq _ZN3Upp6VppLogEv@PLT
 movq %rax, %rdi
 leaq .L.str.1(%rip), %rsi
 callq _ZN3UpplsERNS_6StreamEPKc
 movq %rax, %rdi
 xorl %esi, %esi
 callq _ZN3Upp6StreamlsENS_7EOLenumE
.LBB1_3:
 addq $16, %rsp
 popq %rbp
 .cfi_def_cfa %rsp, 8
 retq
.Lfunc_end1:
 .size _Z14ConsoleMainFn_v, .Lfunc_end1-_Z14ConsoleMainFn_v
 .cfi_endproc
                                # -- End function
 .section .text._ZN3Upp5ColorC2Eiii,"axG",@progbits,_ZN3Upp5ColorC2Eiii,comdat
 .weak _ZN3Upp5ColorC2Eiii            # -- Begin function _ZN3Upp5ColorC2Eiii
 .p2align 4, 0x90
 .type _ZN3Upp5ColorC2Eiii,@function
_ZN3Upp5ColorC2Eiii:                  # @_ZN3Upp5ColorC2Eiii
 .cfi_startproc
# %bb.0:
 pushq %rbp
 .cfi_def_cfa_offset 16
 .cfi_offset %rbp, -16
 movq %rsp, %rbp
 .cfi_def_cfa_register %rbp
 subq $32, %rsp
 movq %rdi, -8(%rbp)
 movl %esi, -12(%rbp)
 movl %edx, -16(%rbp)
 movl %ecx, -20(%rbp)
 movq -8(%rbp), %rax
 movq %rax, -32(%rbp)                 # 8-byte Spill
 movl -12(%rbp), %eax
 movb %al, %dl
 movl -16(%rbp), %eax
 movb %al, %cl
 movl -20(%rbp), %eax
                                # kill: def $al killed $al killed $eax
 movzbl %dl, %edi
 movzbl %cl, %esi
 movzbl %al, %edx
```

```
 callq _ZN3Upp3RGBEhhh
 movl %eax, %ecx
 movq -32(%rbp), %rax                # 8-byte Reload
 movl %ecx, (%rax)
 addq $32, %rsp
 popq %rbp
 .cfi_def_cfa %rsp, 8
 retq
.Lfunc_end2:
 .size _ZN3Upp5ColorC2Eiii, .Lfunc_end2-_ZN3Upp5ColorC2Eiii
 .cfi_endproc
                                     # -- End function
 .section .text._ZNK3Upp5ColoreqES0_,"axG",@progbits,_ZNK3Upp5ColoreqES0_,comdat
 .weak _ZNK3Upp5ColoreqES0_          # -- Begin function _ZNK3Upp5ColoreqES0_
 .p2align 4, 0x90
 .type _ZNK3Upp5ColoreqES0_,@function
_ZNK3Upp5ColoreqES0_:                # @_ZNK3Upp5ColoreqES0_
 .cfi_startproc
# %bb.0:
 pushq %rbp
 .cfi_def_cfa_offset 16
 .cfi_offset %rbp, -16
 movq %rsp, %rbp
 .cfi_def_cfa_register %rbp
 movl %esi, -4(%rbp)
 movq %rdi, -16(%rbp)
 movq -16(%rbp), %rax
 movl (%rax), %eax
 cmpl -4(%rbp), %eax
 sete %al
 andb $1, %al
 movzbl %al, %eax
 popq %rbp
 .cfi_def_cfa %rsp, 8
 retq
.Lfunc_end3:
 .size _ZNK3Upp5ColoreqES0_, .Lfunc_end3-_ZNK3Upp5ColoreqES0_
 .cfi_endproc
                                     # -- End function
 .section .text._ZN3UpplsERNS_6StreamEPKc,"axG",@progbits,_ZN3UpplsERNS_6StreamEPKc
,comdat
 .weak _ZN3UpplsERNS_6StreamEPKc     # -- Begin function _ZN3UpplsERNS_6StreamEPKc
 .p2align 4, 0x90
 .type _ZN3UpplsERNS_6StreamEPKc,@function
_ZN3UpplsERNS_6StreamEPKc:           # @_ZN3UpplsERNS_6StreamEPKc
 .cfi_startproc
# %bb.0:
 pushq %rbp
```

```
 .cfi_def_cfa_offset 16
 .cfi_offset %rbp, -16
 movq %rsp, %rbp
 .cfi_def_cfa_register %rbp
 subq $16, %rsp
 movq %rdi, -8(%rbp)
 movq %rsi, -16(%rbp)
 movq -8(%rbp), %rdi
 movq -16(%rbp), %rsi
 callq _ZN3Upp6Stream3PutEPKc@PLT
 movq -8(%rbp), %rax
 addq $16, %rsp
 popq %rbp
 .cfi_def_cfa %rsp, 8
 retq
.Lfunc_end4:
 .size _ZN3UpplsERNS_6StreamEPKc, .Lfunc_end4-_ZN3UpplsERNS_6StreamEPKc
 .cfi_endproc
                                # -- End function
 .section .text._ZN3Upp6StreamlsENS_7EOLenumE,"axG",@progbits,_ZN3Upp6StreamlsENS_7
EOLenumE,comdat
 .weak _ZN3Upp6StreamlsENS_7EOLenumE   # -- Begin function
_ZN3Upp6StreamlsENS_7EOLenumE
 .p2align 4, 0x90
 .type _ZN3Upp6StreamlsENS_7EOLenumE,@function
_ZN3Upp6StreamlsENS_7EOLenumE:         # @_ZN3Upp6StreamlsENS_7EOLenumE
 .cfi_startproc
# %bb.0:
 pushq %rbp
 .cfi_def_cfa_offset 16
 .cfi_offset %rbp, -16
 movq %rsp, %rbp
 .cfi_def_cfa_register %rbp
 subq $32, %rsp
 movq %rdi, -8(%rbp)
 movl %esi, -12(%rbp)
 movq -8(%rbp), %rdi
 movq %rdi, -24(%rbp)               # 8-byte Spill
 callq _ZN3Upp6Stream6PutEolEv
 movq -24(%rbp), %rax               # 8-byte Reload
 addq $32, %rsp
 popq %rbp
 .cfi_def_cfa %rsp, 8
 retq
.Lfunc_end5:
 .size _ZN3Upp6StreamlsENS_7EOLenumE, .Lfunc_end5-_ZN3Upp6StreamlsENS_7EOLenumE
 .cfi_endproc
                                # -- End function
```

```
 .section .text._ZN3Upp3RGBEhhh,"axG",@progbits,_ZN3Upp3RGBEhhh,comdat
 .weak _ZN3Upp3RGBEhhh               # -- Begin function _ZN3Upp3RGBEhhh
 .p2align 4, 0x90
 .type _ZN3Upp3RGBEhhh,@function
_ZN3Upp3RGBEhhh:                    # @_ZN3Upp3RGBEhhh
 .cfi_startproc
# %bb.0:
 pushq %rbp
 .cfi_def_cfa_offset 16
 .cfi_offset %rbp, -16
 movq %rsp, %rbp
 .cfi_def_cfa_register %rbp
 movb %dl, %al
 movb %sil, %cl
 movb %dil, %dl
 movb %dl, -1(%rbp)
 movb %cl, -2(%rbp)
 movb %al, -3(%rbp)
 movzbl -1(%rbp), %eax
 movzbl -2(%rbp), %ecx
 shll $8, %ecx
 orl %ecx, %eax
 movzbl -3(%rbp), %ecx
 shll $16, %ecx
 orl %ecx, %eax
 popq %rbp
 .cfi_def_cfa %rsp, 8
 retq
.Lfunc_end6:
 .size _ZN3Upp3RGBEhhh, .Lfunc_end6-_ZN3Upp3RGBEhhh
 .cfi_endproc
                                    # -- End function
 .section .text._ZN3Upp6Stream6PutEolEv,"axG",@progbits,_ZN3Upp6Stream6PutEolEv,comdat
 .weak _ZN3Upp6Stream6PutEolEv       # -- Begin function _ZN3Upp6Stream6PutEolEv
 .p2align 4, 0x90
 .type _ZN3Upp6Stream6PutEolEv,@function
_ZN3Upp6Stream6PutEolEv:            # @_ZN3Upp6Stream6PutEolEv
 .cfi_startproc
# %bb.0:
 pushq %rbp
 .cfi_def_cfa_offset 16
 .cfi_offset %rbp, -16
 movq %rsp, %rbp
 .cfi_def_cfa_register %rbp
 subq $16, %rsp
 movq %rdi, -8(%rbp)
 movq -8(%rbp), %rdi
 movl $10, %esi
```

```
 callq _ZN3Upp6Stream3PutEi
 addq $16, %rsp
 popq %rbp
 .cfi_def_cfa %rsp, 8
 retq
.Lfunc_end7:
 .size _ZN3Upp6Stream6PutEolEv, .Lfunc_end7-_ZN3Upp6Stream6PutEolEv
 .cfi_endproc
                                # -- End function
 .section .text._ZN3Upp6Stream3PutEi,"axG",@progbits,_ZN3Upp6Stream3PutEi,comdat
 .weak _ZN3Upp6Stream3PutEi           # -- Begin function _ZN3Upp6Stream3PutEi
 .p2align 4, 0x90
 .type _ZN3Upp6Stream3PutEi,@function
_ZN3Upp6Stream3PutEi:                 # @_ZN3Upp6Stream3PutEi
 .cfi_startproc
# %bb.0:
 pushq %rbp
 .cfi_def_cfa_offset 16
 .cfi_offset %rbp, -16
 movq %rsp, %rbp
 .cfi_def_cfa_register %rbp
 subq $32, %rsp
 movq %rdi, -8(%rbp)
 movl %esi, -12(%rbp)
 movq -8(%rbp), %rcx
 movq %rcx, -24(%rbp)              # 8-byte Spill
 movq 24(%rcx), %rax
 cmpq 40(%rcx), %rax
 jae .LBB8_2
# %bb.1:
 movq -24(%rbp), %rdx              # 8-byte Reload
 movl -12(%rbp), %eax
 movb %al, %cl
 movq 24(%rdx), %rax
 movq %rax, %rsi
 addq $1, %rsi
 movq %rsi, 24(%rdx)
 movb %cl, (%rax)
 jmp .LBB8_3
.LBB8_2:
 movq -24(%rbp), %rdi              # 8-byte Reload
 movl -12(%rbp), %esi
 movq (%rdi), %rax
 callq *(%rax)
.LBB8_3:
 addq $32, %rsp
 popq %rbp
 .cfi_def_cfa %rsp, 8
```

```
 retq
.Lfunc_end8:
 .size _ZN3Upp6Stream3PutEi, .Lfunc_end8-_ZN3Upp6Stream3PutEi
 .cfi_endproc
                                        # -- End function
 .type .L.str,@object                   # @.str
 .section .rodata.str1.1,"aMS",@progbits,1
.L.str:
 .asciz "Yes"
 .size .L.str, 4

 .type .L.str.1,@object                 # @.str.1
.L.str.1:
 .asciz "NO"
 .size .L.str.1, 3

 .section ".linker-options","e",@llvm_linker_options
 .ident "Ubuntu clang version 18.1.3 (1ubuntu1)"
 .section ".note.GNU-stack","",@progbits
 .addrsig
 .addrsig_sym _ZN3Upp9AppInit__EiPPKcS2_
 .addrsig_sym _ZN3Upp12AppExecute__EPFvvE
 .addrsig_sym _Z14ConsoleMainFn_v
 .addrsig_sym _ZN3Upp9AppExit__Ev
 .addrsig_sym _ZN3Upp11GetExitCodeEv
 .addrsig_sym _ZNK3Upp5ColoreqES0_
 .addrsig_sym _ZN3UpplsERNS_6StreamEPKc
 .addrsig_sym _ZN3Upp6VppLogEv
 .addrsig_sym _ZN3Upp6StreamlsENS_7EOLenumE
 .addrsig_sym _ZN3Upp3RGBEhhh
 .addrsig_sym _ZN3Upp6Stream3PutEPKc
 .addrsig_sym _ZN3Upp6Stream6PutEolEv
 .addrsig_sym _ZN3Upp6Stream3PutEi
```

While the if constexpr version

```
#include <Core/Core.h>

using namespace Upp;

CONSOLE_APP_MAIN
{
 constexpr Color c(200,50,76);
 if constexpr ( c == Color(200, 50, 76) )
 {
  RLOG("Yes");
 }else{
```

```
  RLOG("NO");
 }
}
```

compiles to the following 214 lines of assembly code

```
 .text
 .file "TestColorConstant.cpp"
 .globl main                             # -- Begin function main
 .p2align 4, 0x90
 .type main,@function
main:                                    # @main
 .cfi_startproc
# %bb.0:
 pushq %rbp
 .cfi_def_cfa_offset 16
 .cfi_offset %rbp, -16
 movq %rsp, %rbp
 .cfi_def_cfa_register %rbp
 subq $32, %rsp
 movl $0, -4(%rbp)
 movl %edi, -8(%rbp)
 movq %rsi, -16(%rbp)
 movq %rdx, -24(%rbp)
 movl -8(%rbp), %edi
 movq -16(%rbp), %rsi
 movq -24(%rbp), %rdx
 callq _ZN3Upp9AppInit__EiPPKcS2_@PLT
 leaq _Z14ConsoleMainFn_v(%rip), %rdi
 callq _ZN3Upp12AppExecute__EPFvvE@PLT
 callq _ZN3Upp9AppExit__Ev@PLT
 callq _ZN3Upp11GetExitCodeEv@PLT
 addq $32, %rsp
 popq %rbp
 .cfi_def_cfa %rsp, 8
 retq
.Lfunc_end0:
 .size main, .Lfunc_end0-main
 .cfi_endproc
                                         # -- End function
 .globl _Z14ConsoleMainFn_v              # -- Begin function _Z14ConsoleMainFn_v
 .p2align 4, 0x90
 .type _Z14ConsoleMainFn_v,@function
_Z14ConsoleMainFn_v:                     # @_Z14ConsoleMainFn_v
 .cfi_startproc
# %bb.0:
 pushq %rbp
 .cfi_def_cfa_offset 16
```

```
 .cfi_offset %rbp, -16
 movq %rsp, %rbp
 .cfi_def_cfa_register %rbp
 subq $16, %rsp
 movl .L__const._Z14ConsoleMainFn_v.c(%rip), %eax
 movl %eax, -4(%rbp)
 callq _ZN3Upp6VppLogEv@PLT
 movq %rax, %rdi
 leaq .L.str(%rip), %rsi
 callq _ZN3UpplsERNS_6StreamEPKc
 movq %rax, %rdi
 xorl %esi, %esi
 callq _ZN3Upp6StreamlsENS_7EOLenumE
 addq $16, %rsp
 popq %rbp
 .cfi_def_cfa %rsp, 8
 retq
.Lfunc_end1:
 .size _Z14ConsoleMainFn_v, .Lfunc_end1-_Z14ConsoleMainFn_v
 .cfi_endproc
                                    # -- End function
 .section .text._ZN3UpplsERNS_6StreamEPKc,"axG",@progbits,_ZN3UpplsERNS_6StreamEPKc
,comdat
 .weak _ZN3UpplsERNS_6StreamEPKc       # -- Begin function _ZN3UpplsERNS_6StreamEPKc
 .p2align 4, 0x90
 .type _ZN3UpplsERNS_6StreamEPKc,@function
_ZN3UpplsERNS_6StreamEPKc:              # @_ZN3UpplsERNS_6StreamEPKc
 .cfi_startproc
# %bb.0:
 pushq %rbp
 .cfi_def_cfa_offset 16
 .cfi_offset %rbp, -16
 movq %rsp, %rbp
 .cfi_def_cfa_register %rbp
 subq $16, %rsp
 movq %rdi, -8(%rbp)
 movq %rsi, -16(%rbp)
 movq -8(%rbp), %rdi
 movq -16(%rbp), %rsi
 callq _ZN3Upp6Stream3PutEPKc@PLT
 movq -8(%rbp), %rax
 addq $16, %rsp
 popq %rbp
 .cfi_def_cfa %rsp, 8
 retq
.Lfunc_end2:
 .size _ZN3UpplsERNS_6StreamEPKc, .Lfunc_end2-_ZN3UpplsERNS_6StreamEPKc
 .cfi_endproc
```

```
                                        # -- End function
 .section .text._ZN3Upp6StreamlsENS_7EOLenumE,"axG",@progbits,_ZN3Upp6StreamlsENS_7
EOLenumE,comdat
 .weak _ZN3Upp6StreamlsENS_7EOLenumE   # -- Begin function
_ZN3Upp6StreamlsENS_7EOLenumE
 .p2align 4, 0x90
 .type _ZN3Upp6StreamlsENS_7EOLenumE,@function
_ZN3Upp6StreamlsENS_7EOLenumE:         # @_ZN3Upp6StreamlsENS_7EOLenumE
 .cfi_startproc
# %bb.0:
 pushq %rbp
 .cfi_def_cfa_offset 16
 .cfi_offset %rbp, -16
 movq %rsp, %rbp
 .cfi_def_cfa_register %rbp
 subq $32, %rsp
 movq %rdi, -8(%rbp)
 movl %esi, -12(%rbp)
 movq -8(%rbp), %rdi
 movq %rdi, -24(%rbp)               # 8-byte Spill
 callq _ZN3Upp6Stream6PutEolEv
 movq -24(%rbp), %rax               # 8-byte Reload
 addq $32, %rsp
 popq %rbp
 .cfi_def_cfa %rsp, 8
 retq
.Lfunc_end3:
 .size _ZN3Upp6StreamlsENS_7EOLenumE, .Lfunc_end3-_ZN3Upp6StreamlsENS_7EOLenumE
 .cfi_endproc
                                        # -- End function
 .section .text._ZN3Upp6Stream6PutEolEv,"axG",@progbits,_ZN3Upp6Stream6PutEolEv,comdat
 .weak _ZN3Upp6Stream6PutEolEv         # -- Begin function _ZN3Upp6Stream6PutEolEv
 .p2align 4, 0x90
 .type _ZN3Upp6Stream6PutEolEv,@function
_ZN3Upp6Stream6PutEolEv:               # @_ZN3Upp6Stream6PutEolEv
 .cfi_startproc
# %bb.0:
 pushq %rbp
 .cfi_def_cfa_offset 16
 .cfi_offset %rbp, -16
 movq %rsp, %rbp
 .cfi_def_cfa_register %rbp
 subq $16, %rsp
 movq %rdi, -8(%rbp)
 movq -8(%rbp), %rdi
 movl $10, %esi
 callq _ZN3Upp6Stream3PutEi
 addq $16, %rsp
```

```
 popq %rbp
 .cfi_def_cfa %rsp, 8
 retq
.Lfunc_end4:
 .size _ZN3Upp6Stream6PutEolEv, .Lfunc_end4-_ZN3Upp6Stream6PutEolEv
 .cfi_endproc
                                        # -- End function
 .section .text._ZN3Upp6Stream3PutEi,"axG",@progbits,_ZN3Upp6Stream3PutEi,comdat
 .weak _ZN3Upp6Stream3PutEi            # -- Begin function _ZN3Upp6Stream3PutEi
 .p2align 4, 0x90
 .type _ZN3Upp6Stream3PutEi,@function
_ZN3Upp6Stream3PutEi:                  # @_ZN3Upp6Stream3PutEi
 .cfi_startproc
# %bb.0:
 pushq %rbp
 .cfi_def_cfa_offset 16
 .cfi_offset %rbp, -16
 movq %rsp, %rbp
 .cfi_def_cfa_register %rbp
 subq $32, %rsp
 movq %rdi, -8(%rbp)
 movl %esi, -12(%rbp)
 movq -8(%rbp), %rcx
 movq %rcx, -24(%rbp)                  # 8-byte Spill
 movq 24(%rcx), %rax
 cmpq 40(%rcx), %rax
 jae .LBB5_2
# %bb.1:
 movq -24(%rbp), %rdx                  # 8-byte Reload
 movl -12(%rbp), %eax
 movb %al, %cl
 movq 24(%rdx), %rax
 movq %rax, %rsi
 addq $1, %rsi
 movq %rsi, 24(%rdx)
 movb %cl, (%rax)
 jmp .LBB5_3
.LBB5_2:
 movq -24(%rbp), %rdi                  # 8-byte Reload
 movl -12(%rbp), %esi
 movq (%rdi), %rax
 callq *(%rax)
.LBB5_3:
 addq $32, %rsp
 popq %rbp
 .cfi_def_cfa %rsp, 8
 retq
.Lfunc_end5:
```

```
 .size _ZN3Upp6Stream3PutEi, .Lfunc_end5-_ZN3Upp6Stream3PutEi
 .cfi_endproc
                                        # -- End function
 .type .L__const._Z14ConsoleMainFn_v.c,@object # @__const._Z14ConsoleMainFn_v.c
 .section .rodata.cst4,"aM",@progbits,4
 .p2align 2, 0x0
.L__const._Z14ConsoleMainFn_v.c:
 .long 4993736                          # 0x4c32c8
 .size .L__const._Z14ConsoleMainFn_v.c, 4

 .type .L.str,@object                   # @.str
 .section .rodata.str1.1,"aMS",@progbits,1
.L.str:
 .asciz "Yes"
 .size .L.str, 4

 .section ".linker-options","e",@llvm_linker_options
 .ident "Ubuntu clang version 18.1.3 (1ubuntu1)"
 .section ".note.GNU-stack","",@progbits
 .addrsig
 .addrsig_sym _ZN3Upp9AppInit__EiPPKcS2_
 .addrsig_sym _ZN3Upp12AppExecute__EPFvvE
 .addrsig_sym _Z14ConsoleMainFn_v
 .addrsig_sym _ZN3Upp9AppExit__Ev
 .addrsig_sym _ZN3Upp11GetExitCodeEv
 .addrsig_sym _ZN3UpplsERNS_6StreamEPKc
 .addrsig_sym _ZN3Upp6VppLogEv
 .addrsig_sym _ZN3Upp6StreamlsENS_7EOLenumE
 .addrsig_sym _ZN3Upp6Stream3PutEPKc
 .addrsig_sym _ZN3Upp6Stream6PutEolEv
 .addrsig_sym _ZN3Upp6Stream3PutEi
```

Without going into details of assembly code, we know if constexpr indeed provides additional optimization opportunities.

Note in oder for the second example to compile, certain modifications are required in <Core/Color.h>, and a minimum std=c++20 might be required.
