```
; =============== B E G I N N I N G   O F   P R O C E D U R E ===============

; Variables:
;    saved_fp: 0
;    var_10: -16
;    var_14: int32_t, -20
;    var_18: int32_t, -24
;    var_20: int64_t, -32
;    var_24: int32_t, -36
;    var_28: int32_t, -40
;    var_2C: int32_t, -44
;    var_30: int32_t, -48
;    var_34: int32_t, -52
;    var_38: int32_t, -56
;    var_48: int64_t, -72
;    var_50: int64_t, -80
;    var_54: int32_t, -84
;    var_58: int32_t, -88
;    var_60: int64_t, -96
;    var_68: int64_t, -104
;    var_6C: int32_t, -108
;    var_78: int64_t, -120
;    var_80: int64_t, -128
;    var_88: int64_t, -136
;    var_90: int64_t, -144
;    var_98: int64_t, -152
;    var_A0: int64_t, -160
;    var_A8: int64_t, -168
;    var_B0: int64_t, -176
;    var_B8: int64_t, -184
;    var_BC: int32_t, -188
;    var_C0: int64_t, -200
;    var_D0: int64_t, -208
;    var_D8: int64_t, -216
;    var_E0: int64_t, -224


_main:
sub     sp, sp, #0x110
stp     x28, x27, [sp, #0xf0]
stp     x29, x30, [sp, #0x100]
add     x29, sp, #0x100
stur    wzr, [x29, var_14]
stur    w0, [x29, var_18]
stur    x1, [x29, var_20]
adrp    x8, #0x100003000 ; 0x1000034c8@PAGE
add     x8, x8, #0x4c8 ; 0x1000034c8@PAGEOFF, "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n"
mov     x0, x8
str     x8, [sp, #0x100 + var_80]
bl      imp___stubs__printf ; printf
adrp    x8, #0x100003000 ; 0x1000034f1@PAGE
add     x8, x8, #0x4f1 ; 0x1000034f1@PAGEOFF, "Start inside Main\n"
mov     x0, x8
bl      imp___stubs__printf ; printf
ldur    x8, [sp, #0x100 + var_80]
mov     x0, x8
bl      imp___stubs__printf ; printf
adrp    x8, #0x100003000 ; 0x100003504@PAGE
add     x8, x8, #0x504 ; 0x100003504@PAGEOFF, "Process: \\x1B[1;31mFuzzing M1 S3_6_c15_c1_5\\x1B[0m\\n"
mov     x0, x8
bl      imp___stubs__printf ; printf
sub     x8, x29, #0x40
mov     x0, x8
str     x8, [sp, #0x100 + var_90]
bl      imp___stubs__time ; time
ldr     x8, [sp, #0x100 + var_90]
mov     x0, x8
bl      imp___stubs__ctime ; ctime
adrp    x8, #0x100003000 ; 0x100003530@PAGE
add     x8, x8, #0x530 ; 0x100003530@PAGEOFF, "Today is \\x1B[0;35m%s\\x1B[0m\\n"
str     x0, [sp, #0x100 + var_98]
mov     x0, x8
mov     x8, sp
ldr     x9, [sp, #0x100 + var_98]
str     x9, [x8]
bl      imp___stubs__printf ; printf
ldr     x8, [sp, #0x100 + var_90]
mov     x0, x8
bl      imp___stubs__localtime ; localtime
stur    x0, [x29, var_40]
ldur    x8, [x29, var_40]
ldr     w10, [x8, #0x8]
stur    w10, [x29, var_24]
ldur    x8, [x29, var_40]
ldr     w10, [x8, #0x4]
stur    w10, [x29, var_28]
ldur    x8, [x29, var_40]
ldr     w10, [x8]
stur    w10, [x29, var_2C]
ldur    x8, [x29, var_40]
ldr     w10, [x8, #0xc]
stur    w10, [x29, var_30]
ldur    x8, [x29, var_40]
ldr     w10, [x8, #0x10]
add     w10, w10, #0x1
stur    w10, [x29, var_34]
ldur    x8, [x29, var_40]
ldr     w10, [x8, #0x14]
add     w10, w10, #0x76c
stur    w10, [x29, var_38]
ldur    w10, [x29, var_38]
mov     x8, x10
ldur    w10, [x29, var_34]
mov     x1, x10
ldur    w10, [x29, var_38]
mov     x2, x10
adrp    x8, #0x100003000 ; 0x100003548@PAGE
add     x8, x8, #0x548 ; 0x100003548@PAGEOFF, "Run Date (D/M/Y) = %02d/%02d/%d\n"
str     x8, [sp, #0x100 + var_A0]
mov     x0, x8
mov     x8, sp
ldr     x9, [sp, #0x100 + var_A0]
str     x9, [x8]
str     x1, [x8, #0x8]
str     x2, [x8, #0x10]
bl      imp___stubs__printf ; printf
ldur    w10, [x29, var_24]
cmp     w10, #0xc
b.ge    loc_100002bb0
```

```
ldur    w8, [x29, var_24]
mov     x0, x8
ldur    w8, [x29, var_28]
mov     x1, x8
ldur    w8, [x29, var_2C]
mov     x2, x8
adrp    x9, #0x100003000 ; 0x100003569@PAGE
add     x9, x9, #0x569 ; 0x100003569@PAGEOFF, "MS Timer Start at %02d:%02d:%02d am\n"
str     x0, [sp, #0x100 + var_A8]
mov     x0, x9
mov     x9, sp
ldr     x10, [sp, #0x100 + var_A8]
str     x10, [x9]
str     x1, [x9, #0x8]
str     x2, [x9, #0x10]
bl      imp___stubs__printf ; printf
b       loc_100002f4
```

```
loc_100002bb0:
ldur    w8, [x29, var_24] ; CODE XREF=_main+320
subs    w8, w8, #0xc
ldur    w9, [x29, var_28]
mov     x0, x9
ldur    w9, [x29, var_2C]
mov     x1, x9
adrp    x10, #0x100003000 ; 0x100003585@PAGE
add     x10, x10, #0x585 ; 0x100003585@PAGEOFF, "MS Timer Start at %02d:%02d:%02d pm\n"
str     x0, [sp, #0x100 + var_B0]
mov     x0, x10
mov     x10, sp
mov     x2, x8
str     x2, [x10]
str     x1, [sp, #0x100 + var_B0]
str     x11, [x10, #0x8]
str     x1, [x10, #0x10]
bl      imp___stubs__printf ; printf
```

```
loc_100002f4:
adrp    x0, #0x100003000 ; 0x100003503@PAGE, argument "format" for method imp___stubs__printf, CODE XREF=_main+388
add     x0, x0, #0x503 ; 0x100003503@PAGEOFF, "~~~~~~~~~~~~~~~~~~~~~~~~~~\\n"
bl      imp___stubs__printf ; printf
bl      imp___stubs__clock ; clock
stur    x0, [x29, var_50]
adrp    x0, #0x100003000 ; 0x100003cfb@PAGE, argument "format" for method imp___stubs__printf
add     x0, x0, #0x5cf ; 0x100003cfb@PAGEOFF, "Now hitting main() struct sigaction\n"
bl      imp___stubs__printf ; printf
adrp    x8, #0x100003000 ; 0x100003f54@PAGE
add     x8, x8, #0x5f4 ; 0x100003f54@PAGEOFF, "Now hitting main() sigfillset(&sa.sa_mask)\n"
mov     x0, x8
bl      imp___stubs__printf ; printf
sub     x8, x29, #0x60
movn    w9, #0x0
stur    w9, [x29, var_58]
adrp    x10, #0x100003000 ; 0x100003620@PAGE
add     x10, x10, #0x620 ; 0x100003620@PAGEOFF, "Now hitting main() sa.sa_sigaction = bus_handler\n"
mov     x0, x10
str     x8, [sp, #0x100 + var_B8]
str     w9, [sp, #0x100 + var_BC]
bl      imp___stubs__printf ; printf
adrp    x8, #0x100002000 ; 0x100002c50@PAGE
add     x8, x8, #0xc50 ; 0x100002c50@PAGEOFF, _bus_handler
stur    x8, [x29, var_60]
adrp    x8, #0x100003000 ; 0x100003652@PAGE
add     x8, x8, #0x652 ; 0x100003652@PAGEOFF, "Now hitting main() sa.sa_flags = SA_RESTART | SA_SIGINFO\n"
mov     x0, x8
bl      imp___stubs__printf ; printf
movz    w9, #0x42
stur    w9, [x29, var_54]
adrp    x8, #0x100003000 ; 0x10000368c@PAGE
add     x8, x8, #0x68c ; 0x100003068c@PAGEOFF, "Now hitting main() sigaction SIGBUS, &sa, 0\n"
mov     x0, x8
bl      imp___stubs__printf ; printf
movz    w9, #0xa
mov     x0, x9
ldr     x1, [sp, #0x100 + var_80]
movz    x8, #0x0
mov     x2, x8
str     x8, [sp, #0x100 + var_C0]
bl      imp___stubs__sigaction ; sigaction
adrp    x8, #0x100003000 ; 0x100003760@PAGE
add     x8, x8, #0x669 ; 0x100003669@PAGEOFF, "Now hitting main() sa.sa_sigaction = sev_handler\n"
mov     x0, x8
bl      imp___stubs__printf ; printf
adrp    x8, #0x100002000 ; 0x100002c98@PAGE
add     x8, x8, #0xc98 ; 0x100002c98@PAGEOFF, _sev_handler
stur    x8, [x29, var_68]
adrp    x8, #0x100003000 ; 0x100003eeb@PAGE
add     x8, x8, #0x6eb ; 0x100003eeb@PAGEOFF, "Now hitting main() sigaction SIGSEGV, &sa\n"
mov     x0, x8
bl      imp___stubs__printf ; printf
movz    w9, #0xb
mov     x0, x9
ldr     x1, [sp, #0x100 + var_80]
ldr     x2, [sp, #0x100 + var_C0]
bl      imp___stubs__sigaction ; sigaction
adrp    x8, #0x100003000 ; 0x100003716@PAGE
add     x8, x8, #0x716 ; 0x100003716@PAGEOFF, "Now hitting main() uint32_t *ptr = mmap(NULL, 0x4000, PROT_READ | PROT_WRITE | PROT_EXEC, MAP_PRIVATE | MAP_ANONYMOUS | MAP_JIT, -1, 0)\\n\\n"
mov     x0, x8
bl      imp___stubs__printf ; printf
ldr     x8, [sp, #0x100 + var_C0]
mov     x0, x8
movz    x1, #0x4000
movz    w2, #0x7
movz    w3, #0x1002
ldr     w4, [sp, #0x100 + var_BC]
movz    x10, #0x0
mov     x5, x10
bl      imp___stubs__mmap ; mmap
stur    x0, [x29, var_68]
adrp    x0, #0x100003000 ; 0x1000037a0@PAGE, argument "format" for method imp___stubs__printf
add     x0, x0, #0x7a0 ; 0x1000037a0@PAGEOFF, "Now hitting main() write_sprr_perm(0x3333333333333333)\n\n\n"
bl      imp___stubs__printf ; printf
orr     x8, xzr, #0x3333333333333333
mov     x0, x8
bl      _write_sprr_perm ; _write_sprr_perm
adrp    x0, #0x100003000 ; 0x1000037d9@PAGE, argument "format" for method imp___stubs__printf
add     x0, x0, #0x7d9 ; 0x1000037d9@PAGEOFF, "Just executed main() write_sprr_perm(0x3333333333333333)\n\n\n"
bl      imp___stubs__printf ; printf
adrp    x8, #0x100003000 ; 0x100003814@PAGE
add     x8, x8, #0x814 ; 0x100003814@PAGEOFF, "Now in main() hitting ptr[0] 0x065f03c0 RET\n\n\n"
mov     x0, x8
bl      imp___stubs__printf ; printf
ldur    x9, [x29, var_68]
movz    w0, #0x3c0
movk    w0, #0x065f, lsl #16
str     w9, [x0]
adrp    x8, #0x100003000 ; 0x100003842@PAGE
add     x8, x8, #0x842 ; 0x100003842@PAGEOFF, "Now in main() hitting for (int i = 0; i < 4; ++i)\n\n\n"
mov     x0, x8
bl      imp___stubs__printf ; printf
stur    wzr, [x29, var_6C]
```

```
loc_100002b68:
ldur    w8, [x29, var_6C] ; CODE XREF=_main+900
cmp     w8, #0x4
b.ge    loc_100002bb0
```

```
ldur    x8, [x29, var_68]
ldur    w0, [x29, var_6C]
and     w8, w8, #0xff
str     x8, [sp, #0x100 + var_D8]
mov     x0, x8
bl      _make_sprr_val ; _make_sprr_val
ldr     x9, [sp, #0x100 + var_D8]
str     x8, [sp, #0x100 + var_D8]
mov     x0, x9
ldr     x1, [sp, #0x100 + var_D8]
bl      _sprr_test  ; _sprr_test
ldur    w8, [x29, var_6C]
add     w8, w8, #0x1
stur    w8, [x29, var_6C]
b       loc_100002b68
```

```
loc_100002bb0:
bl      imp___stubs__clock ; clock, CODE XREF=_main+840
stur    x0, [x29, var_78]
ldur    x8, [x29, var_78]
ldur    x9, [x29, var_50]
subs    x8, x8, x9
ucvtf   d0, x8
adrp    x8, #0x100003000
ldr     d1, [x8, #0x4c0] ; double_value_1000
fmul    d0, d0, d1
adrp    x8, #0x100003000
ldr     d1, [x8, #0x4b8] ; double_value_1E06
fdiv    d0, d0, d1
str     d0, [sp, #0x100 + var_80]
ldr     d0, [sp, #0x100 + var_80]
adrp    x0, #0x100003000 ; 0x100003876@PAGE
add     x0, x0, #0x876 ; 0x100003876@PAGEOFF, "Finished... Total Elapsed Time in ms: %f\n\n\n"
mov     x8, sp          ; argument "format" for method imp___stubs__printf
str     d0, [x8]
bl      imp___stubs__printf ; printf
sub     x8, x29, #0x40
mov     x0, x8
bl      imp___stubs__ctime ; ctime
adrp    x8, #0x100003000 ; 0x1000038a1@PAGE
add     x8, x8, #0x8a1 ; 0x1000038a1@PAGEOFF, "End Time %s"
str     x0, [sp, #0x100 + var_E0]
mov     x0, x8
mov     x8, sp
ldr     x9, [sp, #0x100 + var_E0]
str     x9, [x8]
bl      imp___stubs__printf ; printf
adrp    x8, #0x100003000 ; 0x1000038ad@PAGE
add     x8, x8, #0x8ad ; 0x1000038ad@PAGEOFF, "Done.......\n\n\n\n"
mov     x0, x8
bl      imp___stubs__printf ; printf
ldur    w10, [x29, var_14]
mov     x0, x10
ldp     x29, x30, [sp, #0x100]
ldp     x28, x27, [sp, #0xf0]
add     sp, sp, #0x110
ret
```