LinuxTesting.org

SMG: Current State and Future Work

Anton Vasilyev

ISP RAS

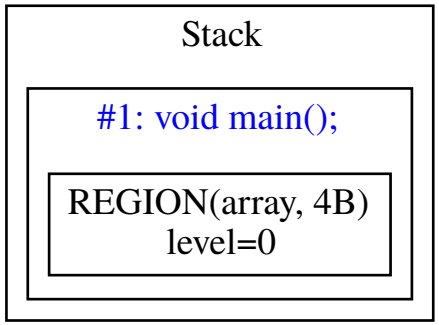
Ivannikov Institute for System Programming of the RAS

SMG

- Shape analysis for memory faults verification
  - Overflow
  - Over-read
  - Dangling pointer
  - Null pointer dereference
  - Uninitialized variables
  - Memory leak
  - Double free
  - Invalid free
  - Use after free

Symbolic Memory Graph

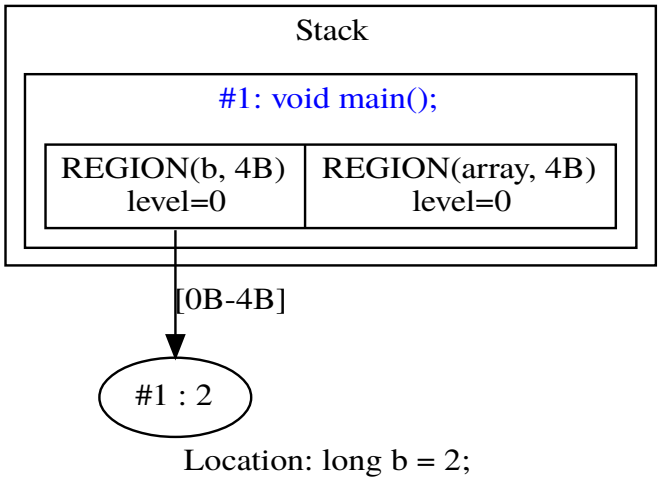
```
void main() {
    void *array;
    long b = 2;
    long c = 3;
    array = calloc(1, 16);
    memcpy(&array[4], &b, 4);
    memcpy(&array[5], &c, 4);
}
```



Location: void *array;

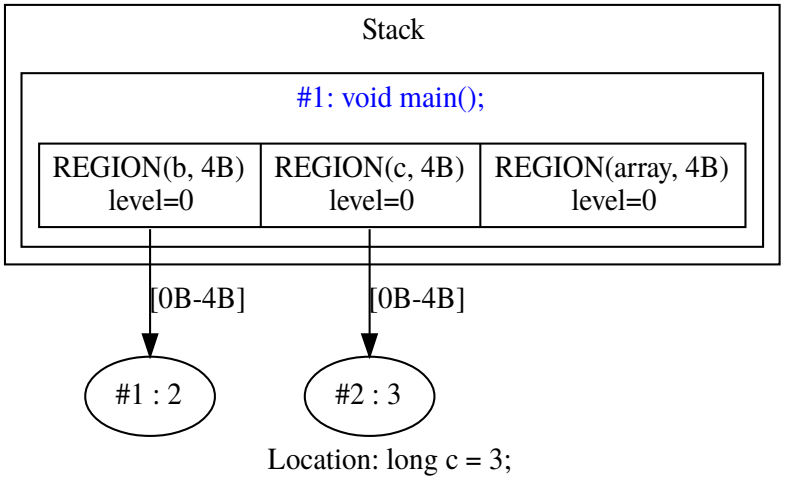
Symbolic Memory Graph

```
void main() {
    void *array;
    long b = 2;
    long c = 3;
    array = calloc(1, 16);
    memcpy(&array[4], &b, 4);
    memcpy(&array[5], &c, 4);
}
```



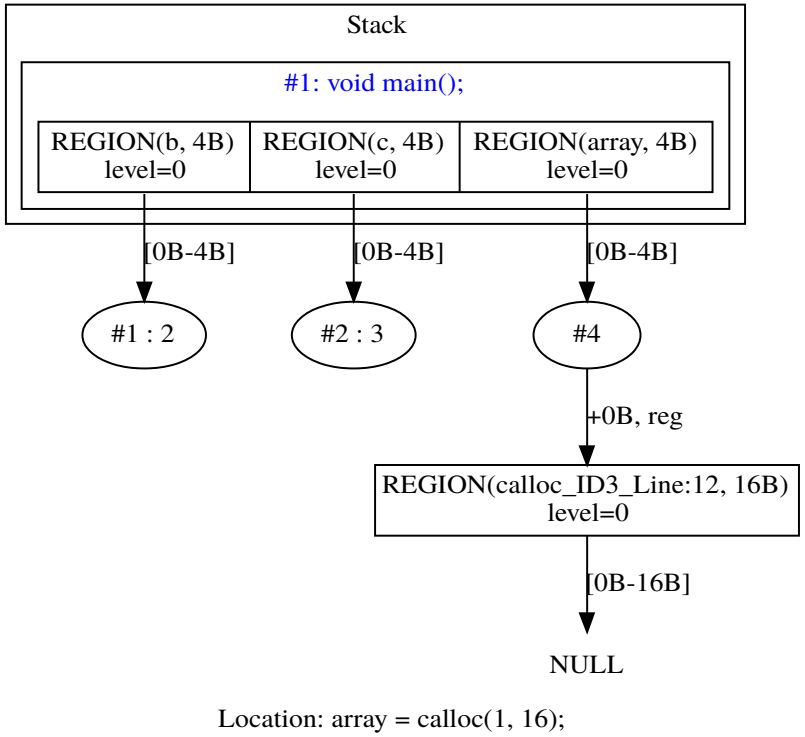
Symbolic Memory Graph

```
void main() {
    void *array;
    long b = 2;
    long c = 3;
    array = calloc(1, 16);
    memcpy(&array[4], &b, 4);
    memcpy(&array[5], &c, 4);
}
```



Symbolic Memory Graph

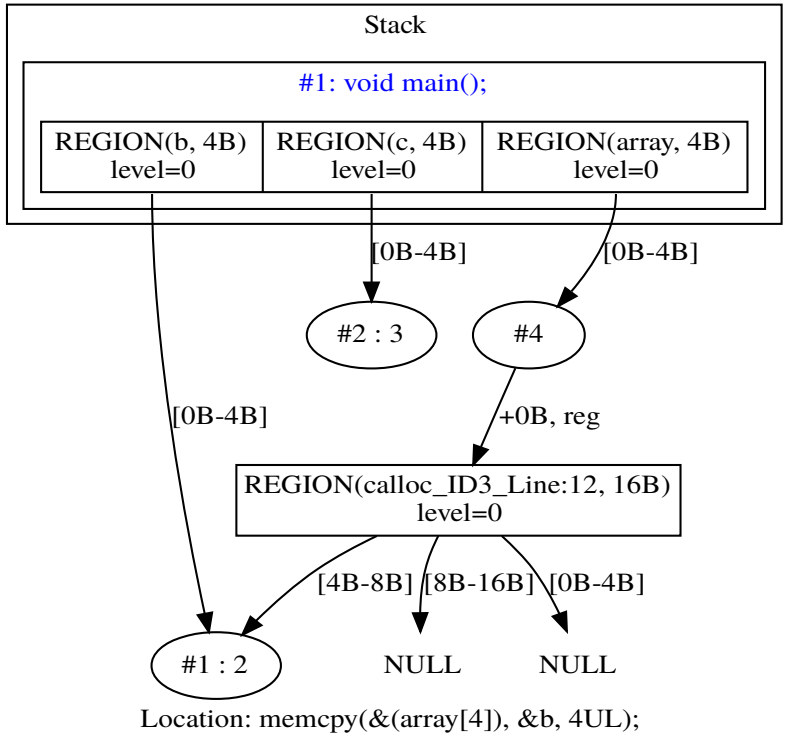
```
void main() {
    void *array;
    long b = 2;
    long c = 3;
    array = calloc(1, 16);
    memcpy(&array[4], &b, 4);
    memcpy(&array[5], &c, 4);
}
```



Location: array = calloc(1, 16);

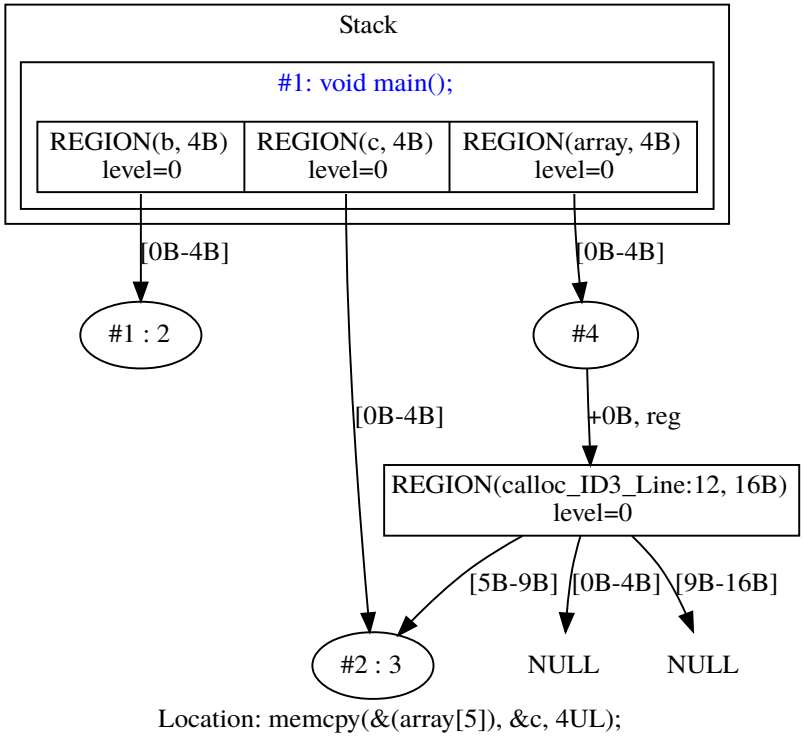
Symbolic Memory Graph

```
void main() {
    void *array;
    long b = 2;
    long c = 3;
    array = calloc(1, 16);
    memcpy(&array[4], &b, 4);
    memcpy(&array[5], &c, 4);
}
```



Symbolic Memory Graph

```
void main() {
    void *array;
    long b = 2;
    long c = 3;
    array = calloc(1, 16);
    memcpy(&array[4], &b, 4);
    memcpy(&array[5], &c, 4);
}
```



SMG current state

- Trace:

```
assume(irq_base >= 0);
chip->irq_base = irq_base;
▸ ret = ioh_gpio_alloc_generic_chip(chip, (unsigned int)irq_base, (unsigned int)num_ports[j]);
assume(ret == 0);
j = j + 1;
chip = chip + 1;
assume(j <= 7);
irq_base = __devm_irq_alloc_descs(&pdev->dev, -1, 0U, (unsigned int)num_ports[j], -1, &__this_m
assume(irq_base >= 0);
chip->irq_base = irq_base;
▸ ret = ioh_gpio_alloc_generic_chip(chip, (unsigned int)irq_base, (unsigned int)num_ports[j]);
assume(ret == 0);
j = j + 1;
chip = chip + 1;
assume(j <= 7);
irq_base = __devm_irq_alloc_descs(&pdev->dev, -1, 0U, (unsigned int)num_ports[j], -1, &__this_m
assume(irq_base >= 0);
chip->irq_base = irq_base;
▸ ret = ioh_gpio_alloc_generic_chip(chip, (unsigned int)irq_base, (unsigned int)num_ports[j]);
assume(ret == 0);
j = j + 1;
chip = chip + 1;
assume(j <= 7);
irq_base = __devm_irq_alloc_descs(&pdev->dev, -1, 0U, (unsigned int)num_ports[j], -1, &__this_m
assume(irq_base >= 0);
chip->irq_base = irq_base;
▸ ret = ioh_gpio_alloc_generic_chip(chip, (unsigned int)irq_base, (unsigned int)num_ports[j]);
assume(ret == 0);
j = j + 1;
chip = chip + 1;
assume(j > 7);
chip = (struct ioh_gpio *)chip_save;
▸ ret = devm_request_irq(&pdev->dev, pdev->irq, &ioh_gpio_handler, 128UL, "gpio_ml_ioh", (void
assume(ret != 0);
dev_err((struct device const *)(&pdev->dev), "%s request_irq failed\n", "ioh_gpio_probe");
i = i - 1;
assume(i >= 0);
chip = chip - 1;
Field with size 8 byte can't be written at offset -560 byte of object 4672 byte size
gpiochip_remove(&chip->gpio);
```

SMG current state

- Extended witness with highlighting

Error trace

```
Global variable declarations
Entry point 'main'
Invoke PCI driver callbacks. (Relevant to 'ioh_gpio_driver')
    Probe new PCI driver. Invoke callback probe from pci_driver.
        ioh_gpio_probe
            struct ioh_gpio *chip ;
            void *chip_save ;
            chip_save = kzalloc(4672UL, 20971712U);
            chip_save = kzalloc(4672UL, 20971712U);
            chip = (struct ioh_gpio *)chip_save;
            chip = chip - 1;
            Field with size 8 byte can't be written at offset -560
            gpiochip_remove(&chip->gpio);
```

Further work

- Goal: Improve visualization
  - Task 1: Granulate highlighting
  - Task 2: Collect more information from predicates

SMG current state

- Kernel verification / GSoC
  - 2018, v.4.16.10 — 29 Bugs/22 reported bugs
  - 2017, v.4.11.6 — 49 Bugs/11 reported bugs
- Further work: Report all possible errors
  - Task 1: continue after memleak
  - Task 2: restore state after error

SMG scalability

- Current state:
  - Function summary (by Petr Melnichenko) by function models
    - Argument may be NULL
    - Argument must be not NULL
    - Function may return NULL
    - Function do not return NULL
- Further work
  - Function summary as graph transformation

SMG Absent functions

- Current state
  - “Pure” functions – return memory on-demand
  - Strict – throw an error
  - Do nothing
- Further work
  - Annotate functions by determinism and side effect tags

SMG analysis speed

- Current state: almost immutable state
- Future work: refactore predicate extension

Further work

- Task: IS_ERR(), ERR_PTR() support
  - Predicates on pointers
  - Work with undefined values

Further work

- Goal: C programs verification
  - Arrays and strings abstractions
  - Strings library functions

Further work

- Bug Fixes
  - Abstraction with memory on-demand