



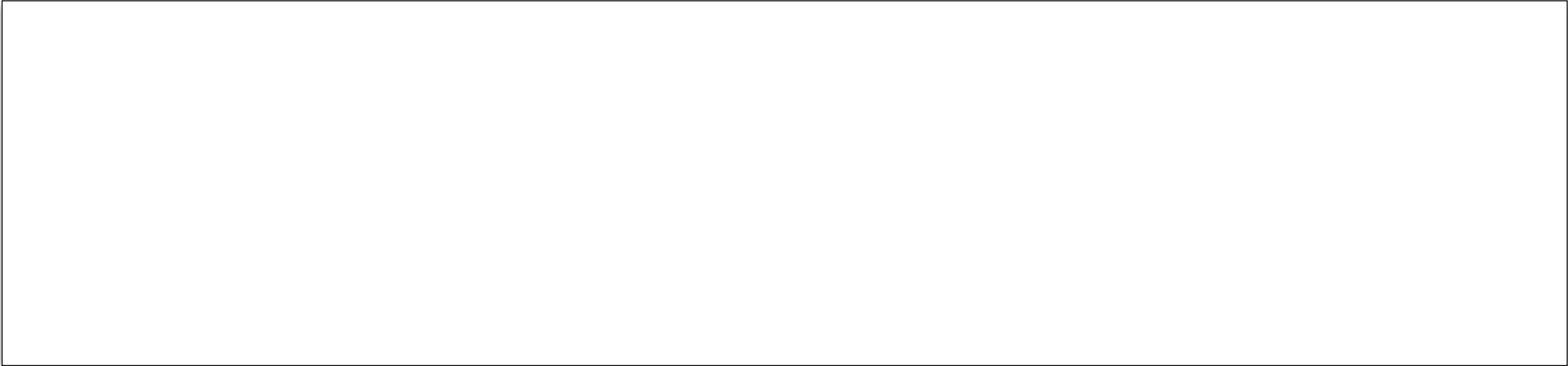


# Lightweight Data Race Detection

1

1

0











































































































































W





































```
void read_started (check_state& state)
{
    ++state.num_readers; // must be first

    if (state.is_writing)
        std::terminate();
        // read during active write
}

void write_started (check_state& state)
{
    // must be first
    if (state.is_writing.exchange (true))
        std::terminate();
        // write during active write

    if (state.num_readers > 0)
        std::terminate();
        // write during active read
}
```

```
void read_ended (check_state& state)
{
    --state.num_readers;
}
```

```
void write_ended (check_state& state)
{
    state.is_writing = false;
}
```

```
struct check_state
{
    std::atomic<size_t> num_readers { 0 };
    std::atomic<bool> is_writing { false };
};
```





enum class check\_type





template<check\_type>

read,

elise

```
if constexpr(type == check_type::read)
```



writes

read\_started(state);

structured copycheck



```
read_ended(state);
```

write\_end(state);

`write_started(`state`);`

state(check\_state)



stoped\_check(check\_state & check\_state)

checkstate & state;

scope\_dependent\_check()



Lightweight Data Race Detection





```
std::atomic<size_t> num_readers{0};
```

```
std::atomic<bool> is_writing{false};
```

struct check\_state

---







`void write_state (check_state & state)`

`++state.num_readers; // must be first`

~~/~~~~/~~~~/~~ must be first

//reading active write

**if** (state.is\_writing)

`void read_start(check_state & state)`



std::terminate();

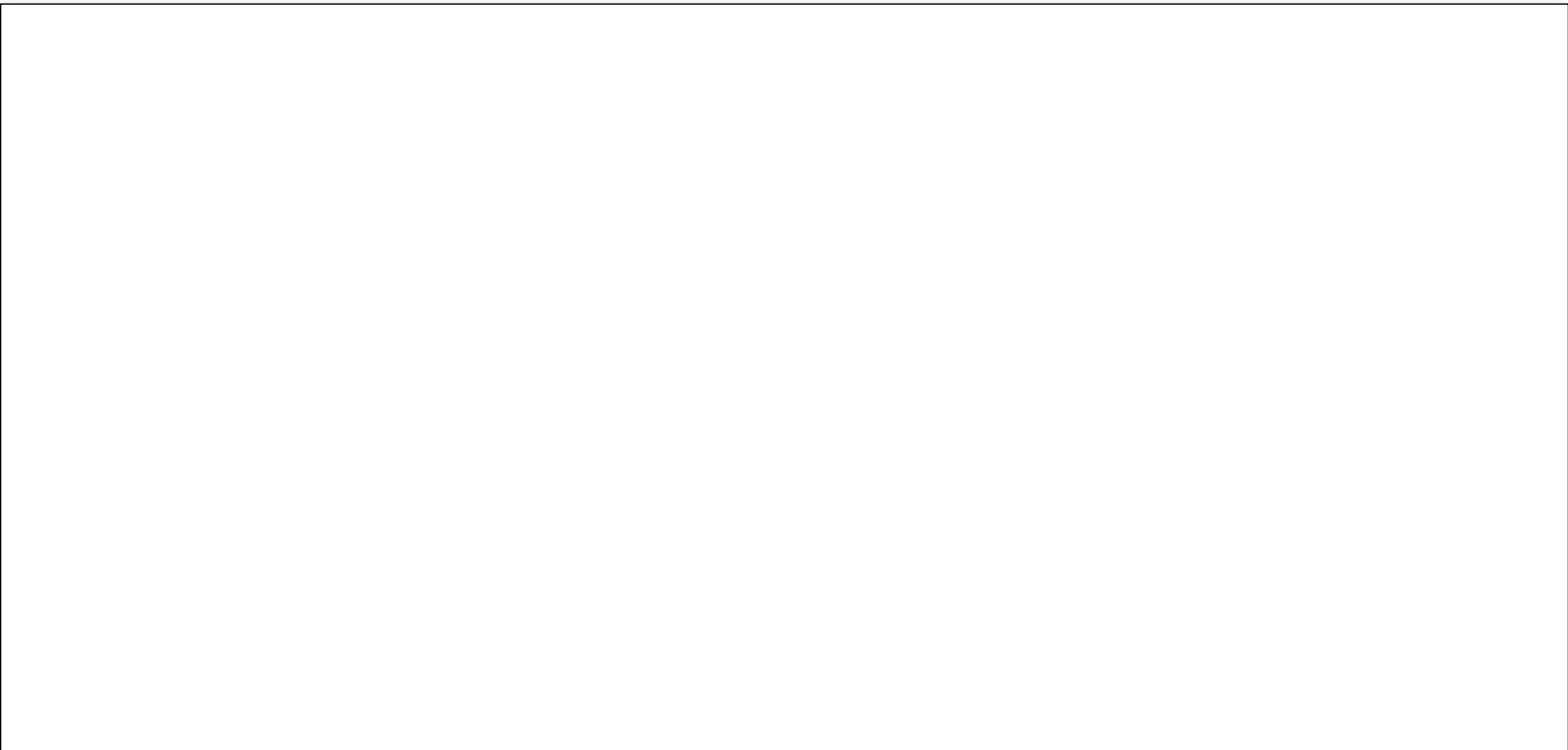


~~/~~~~/~~~~/~~write during active write

```
if(state.is_writing_exchange(true))
```

//write during active read

**if** (state.num\_readers > 0)



void read\_endded(check\_state& state)







`-state.num_readers;`

`void write_endded (checked_state& state)`

```
state.is_writing = false;
```

