

Ongoing Optimization with CppMem

[CppMem](#) is an interactive tool for exploring the behavior of small code snippets using the C++ memory model.

- CppMem
 - verifies the behavior of small code snippets.
 - considers all possible interleaving of threads, visualizes each of them in a graph, and annotates the graph with additional details.
 - uses a simplified C/C++ syntax for the programs.
 - is a precious tool to deepen your knowledge about the C++ memory model.

Ongoing Optimization with CppMem

```
// ongoingOptimization.cpp

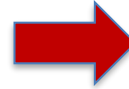
#include <iostream>
#include <thread>

int x = 0;
int y = 0;

void writing(){
    x = 2000;
    y = 11;
}

void reading(){
    std::cout << "y: " << y << " ";
    std::cout << "x: " << x << '\n';
}

int main(){
    std::thread thread1(writing);
    std::thread thread2(reading);
    thread1.join();
    thread2.join();
}
```



```
File Edit View Bookmarks Settings Help
rainer@suse:~> ongoingOptimisation
y: 0 x: 2000
rainer@suse:~> ongoingOptimisation
y: 0 x: 2000
rainer@suse:~> ongoingOptimisation
y: 11 x: 2000
rainer@suse:~> ongoingOptimisation
y: 0 x: 2000
rainer@suse:~> ongoingOptimisation
y: 11 x: 2000
rainer@suse:~> ongoingOptimisation
y: 0 x: 0
rainer@suse:~> ongoingOptimisation
y: 0 x: 2000
rainer@suse:~> ongoingOptimisation
y: 11 x: 2000
rainer@suse:~> █
rainer : bash
```

Ongoing Optimization with CppMem

Use the program `ongoingOptimization.cpp` as your starting point.

- Here are various variations of the program:
 - `ongoingOptimization.cpp`
 - `ongoingOptimizationLock.cpp`
 - `ongoingOptimizationSequentialConsistency.cpp`
 - `ongoingOptimizationAcquireRelease.cpp`
 - `ongoingOptimizationAcquireRelease2.cpp`
 - `ongoingOptimizationRelaxedSemantic.cpp`
- Transform each `cpp` program into its CppMem pendant.
- Determine for each CppMem pendant if it is thread-safe and, if so, which outcomes are possible.

Ongoing Optimization with CppMem

- CppMem Pendants

- `ongoingOptimization.txt`
- `ongoingOptimizationSequentialConsistency.txt`
- `ongoingOptimizationAcquireRelease.txt`
- `ongoingOptimizationAcquireRelease2.txt`
- `ongoingOptimizationRelaxedSemantic.txt`