



Actors: Problems

• In practice many need different pools

- Serialises all actors on to a single thread

• Could use "Annotations for Reflection" P3394

• Different pods together

• Different scheduler types

105


```
auto get_scheduler()  
{  
    static exec::static_thread_pool pool(1);  
    return pool.get_scheduler();  
}
```

```
struct [[=LowPriority]] person(actor)  
//...
```

```
struct LowPriority_tag;
```

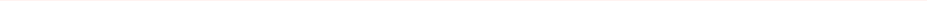
```
template<typename PoolType>  
auto get_scheduler()  
{  
    static exec::static_thread_pool pool(1);  
    // init low-priority  
    return pool.get_scheduler();  
}
```

```
struct [[=MainActor]] person(actor)  
//...
```

```
template<typename PoolType>  
auto get_main_scheduler()  
{  
    static exec::run_loop loop {};  
    // Needs to be dispatched by main thread  
    return loop.get_scheduler();  
}
```













Actors: Problems

- In practice may need different pools
 - Serialises all actors on to a single thread
- Could use “Annotations for Reflection” P3394
 - Different pool tags
 - Different scheduler types

```
struct [[=MainActor]] person(actor)
//...

template<typename PoolType>
auto get_main_scheduler()
{
    static exec::run_loop loop {};
    // Needs to be dispatched by main thread
    return loop.get_scheduler();
}
```

```
auto get_scheduler()
{
    static exec::static_thread_pool pool(1);
    return pool.get_scheduler();
}
```

```
struct [[=LowPriority]] person(actor)
//...

struct LowPriority_tag;

template<typename PoolType>
auto get_scheduler()
{
    static exec::static_thread_pool pool(1);
    // init low-priority
    return pool.get_scheduler();
}
```



Actors: Problems