89

```cpp
class safe_thread
{
public:
    template<typename F, send... Args>
    safe_thread (F&& f, Args&&... args)
        : thread (std::forward<F> (f), std::forward<Args> (args)...)
    {
        static_assert((is_function_pointer_v<std::decay_t<std::decay_t<F>>>
                       && ! std::is_member_function_pointer_v<std::decay_t<F>>)
                      || is_send_v<std::decay_t<F>>);
    }

    safe_thread (safe_thread&& other)
        : thread (std::move (other.thread))
    {
    }

private:
    std::jthread thread;
};
```

```
class safe_thread
{
public:
    template<typename F, send... Args>
    safe_thread (F&& f, Args&&... args)
        : thread (std::forward<F> (f), std::forward<Args> (args)...)
    {
        static_assert((is_function_pointer_v<std::decay_t<std::decay_t<F>>>
                       && ! std::is_member_function_pointer_v<std::decay_t<F>>)
                      || is_send_v<std::decay_t<F>>);
    }

    safe_thread (safe_thread&& other)
        : thread (std::move (other.thread))
    {
    }

private:
    std::jthread thread;
};
```

Send in C++: *Moved between threads*

```
template<typename F, send... Args>
safe_thread (F&& f, Args&&... args)
    : thread (std::forward<F> (f), std::forward<Args> (args)...)
{
}
```