Awaiting a set of handles in C++/WinRT

devblogs.microsoft.com/oldnewthing/20240429-00

April 29, 2024

Raymond Chen

C++/WinRT provides the `resume_on_signal` awaiter that allows you to await until a kernel handle is signaled. What if you have a bunch of these handles, and you want to await until *all* of them are signaled?

It turns out that this is easier than it sounds. We can use the same trick as we used in our basic `when_all` function: Just await each handle in sequence.

```
winrt::Windows::Foundation::IAsyncAction
    when_all_signaled(std::vector<HANDLE> handles)
{
    for (auto handle : handles) {
        co_await winrt::resume_on_signal(handle);
    }
}
```

If you want to accept the handles varadically, then it's the one-liner we saw before:

```
template<typename... Handles>
winrt::Windows::Foundation::IAsyncAction
    when_all_signaled(Handles... handles)
{
    (co_await winrt::resume_on_signal(handles), ...);
}
```

Things get more complicated if we want to await a set of handles with a timeout. That's what we'll be looking at for the next few days.