Creating an automatic-reset event from WaitOnAddress

devblogs.microsoft.com/oldnewthing/20170615-00

June 15, 2017

Raymond Chen

Last time, we created a manual-reset event from `WaitOnAddress`. Today, it's an automatic-reset event.

```
struct ALT_AEVENT
{
  LONG State;
};

void InitializeAltAutoEvent(ALT_AEVENT* Event,
                            bool InitialState)
{
  Semaphore->State = InitialState;
}

void SetAltAutoEvent(ALT_AEVENT* Event)
{
 if (InterlockedCompareExchange(&Event->State,
                                true, false) == false) {
  WakeByAddressSingle(&Event->State);
 }
}

void ResetAltAutoEvent(ALT_AEVENT* Event)
{
 InterlockedCompareExchange(&Event->State,
                            false, true);
}

void WaitForAltAutoEvent(ALT_AEVENT* Event)
{
 while (!InterlockedCompareExchange(&Event->State,
                                    false, true)) {
  LONG Waiting = 0;
  WaitOnAddress(&Event->State,
                &Waiting,
                sizeof(Waiting),
                INFINITE);
 }
}
```

Most of this code is the same as with manual-reset events. One difference is that when setting the event, we use `WakeByAddressSingle` because signaling an auto-reset event releases at most one thread.

The interesting change is in thw code that waits. Instead of merely checking the state, we try to transition it from `true` to `false`, which simultaneously checks and claims the token.

Okay, next time, we're going to put together what we've been learning this week to solve an actual problem.

Raymond Chen

Follow