
Subject: Re: What is the best way to create a Semaphore with timeout?

Posted by [Tom1](#) on Tue, 17 Dec 2019 10:53:20 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

I think this is now up to Mirek now... ;)

I have made the needed additions to Semaphore for implementing timeout on Wait.

Here's working code for Windows and Linux:

Mt.h:

```
...
class Semaphore : NoCopy {
#ifdef PLATFORM_WIN32
    HANDLE    handle;
#elif PLATFORM_OSX
    dispatch_semaphore_t    sem;
#else
    sem_t      sem;
#endif

public:
    bool    Wait(int timeout_ms); // Added for timeout - returns true if semaphore was signaled, false
    otherwise
    void    Wait();
    void    Release();
#ifdef PLATFORM_WIN32
    void    Release(int n);
#endif

    Semaphore();
    ~Semaphore();
};
...
```

Mt.cpp:

```
...

// For Windows:
bool Semaphore::Wait(int timeout_ms)
{
    return WaitForSingleObject(handle, timeout_ms<0 ? INFINITE : timeout_ms) ==
```

```

WAIT_OBJECT_0 ? true : false;
}

...

// For Linux:
bool Semaphore::Wait(int timeout_ms)
{
    if(timeout_ms<0){
        Wait();
        return true;
    }

    struct timespec until;
    clock_gettime(CLOCK_REALTIME, &until);

    until.tv_sec+=timeout_ms/1000;
    timeout_ms%=1000;
    until.tv_nsec+=timeout_ms*1000000;
    until.tv_sec+=until.tv_nsec/1000000000;
    until.tv_nsec%=1000000000;

    return sem_timedwait(&sem,&until)==-1 ? false : true;
}

...

```

Maybe someone with a Mac could take a look at the OSX code for the same. From what I found on the web, it looks like it could be something like this for OSX:

```

...
bool Semaphore::Wait(int timeout_ms)
{
    if(timeout_ms<0){
        dispatch_semaphore_wait(sem, DISPATCH_TIME_FOREVER);
        return true;
    }
    return dispatch_semaphore_wait(sem, dispatch_time(DISPATCH_TIME_NOW, 1000000 *
    timeout_ms)) == 0 ? true : false;
}

...

```

I hope this makes it to the Core.

Best regards,

Tom
