Windows Threads

We repeat the last example of two counting threads. Now we give the two threads different priority levels using the functions:

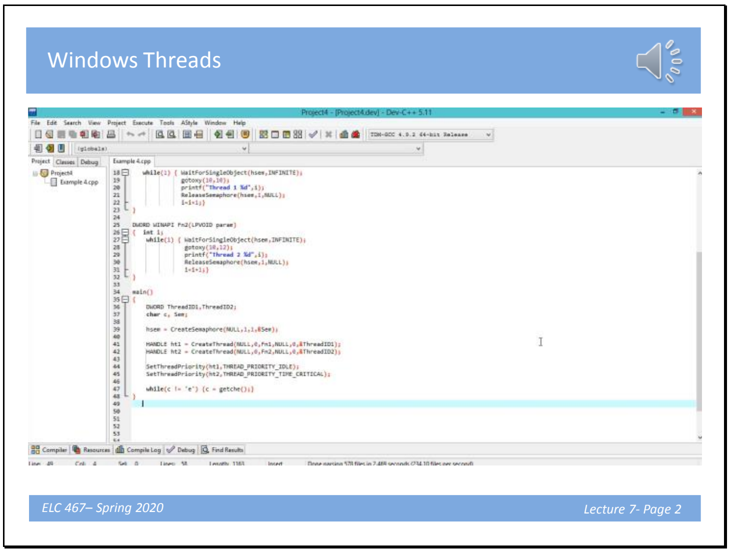
```
SetThreadPriority(ht1,THREAD_PRIORITY_IDLE);
SetThreadPriority(ht2,THREAD_PRIORITY_TIME_CRITICAL);
```

Will this have an effect on the speed at which the threads count?

ELC 467– Spring 2020 Lecture 7 Page 1

Slide 1

We give one of the threads the highest priority level in its class, and give the other the lowest level. This is done using the API function SetThreadPriority applied to the handle of each thread. We expect that the second thread will be given more CPU time than the first thread, causing it to have a faster counting rate. Run the video on the next slide to see if this happens.



Slide 2

As seen here, changing the priority levels of treads has no effect on the counting speed. Try to find why this happened before going to the next slide. Hint: on older hardware the change in counting speed is observable.

Windows Threads

The higher priority thread did not cause a delay in the low priority thread. Possible causes are:

- Current architectures have multiple cores. The two threads may ran on different cores, thus not delaying each other.
- The threads in our example are very short, not consuming long CPU times. Thus, on fast processor the delay they cause on each other is not noticeable.

ELC 467– Spring 2020 *Lecture 7 Page 3*

Windows Threads

To test if this is true, we force the two threads to run on the same core using the API functions:

```
SetThreadAffinityMask(ht1,1);
SetThreadAffinityMask(ht2,1);
```

We also increase the thread CPU times by adding a delay loop in each thread:

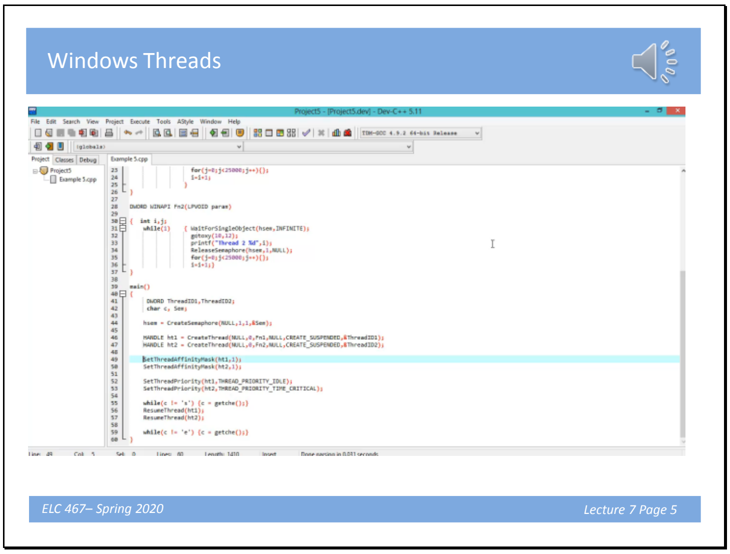
```
for(j=0;j<25000;j++){};
```

The resulting code is in file Example5.cpp.

ELC 467– Spring 2020 *Lecture 7 Page 4*

<u>Slide 4</u>

The function SetThreadAffinityMask forces the system to run a thread on a particular core. Here we cause the two threads to interfere with each other by running them on the same core. Note that we use this function here for testing only, but normally users will let the operating system select the core on which threads run to have the best possible performance.



Slide 6

As we can see, the modifications made resulted in observed difference in the speed at which the two threads count. Try to experiment with the effect of each modification alone.