An event is an object that communicates the status of OpenCL commands.

```
size_t globalWorkSize[3] = { NUM_ELEMENT, 1, 1 };  
size_t localWorkSize[3] = { LOCAL_SIZE, 1, 1 };  
status = clEnqueueNDRangeKernel( cmdQueue, kernel, 1, NULL, globalWorkSize, localWorkSize, 0, NULL, NULL );
```

From the OpenCL Notes:

11. Enqueue the Kernel Object for Execution

```
status = clEnqueueNDRangeKernel( cmdQueue, kernel, 1, NULL, globalWorkSize, localWorkSize, 0, NULL, &waitKernelC );
```

Waiting for Events

```
cl_event waitKernelC;  
status = clEnqueueNDRangeKernel( cmdQueue, kernelC, 1, NULL, globalWorkSize, localWorkSize, 2, dependencies, &waitKernelC );
```

Creating an Execution Graph Structure

```
cl_event waitKernelC;  
status = clEnqueueNDRangeKernel( cmdQueue, kernelC, 1, NULL, globalWorkSize, localWorkSize, 2, dependencies, &waitKernelC );
```
cl_event waitKernelA, waitKernelB.

 status = clEnqueueNDRangeKernel( cmdQueue, kernelC, 1, NULL, globalWorkSize, localWorkSize, 1, &waitKernelA, NULL );

Waiting for One Event

status = clEnqueueBarrier( cmdQueue );

This does not complete until all commands enqueued before it have completed.

Placing a Barrier in the Command Queue

cl_event waitMarker;

status = clEnqueueMarker( cmdQueue, &waitMarker );

This does not complete until all commands enqueued before it have completed.

This is just like a barrier, but it can throw an event to be waited for.

Placing an Event Marker in the Command Queue

status = clWaitForEvents( 2, dependencies );

This blocks until the specified events are thrown, so use it carefully!

Waiting for Events Without Enqueuing Another Command

if( status != CL_SUCCESS )
    fprintf( stderr, "Wait: clWaitForEvents failed\n" );

Call this before starting the timer, before ending the timer, and before retrieving data from an array computed in an OpenCL program.

I Like Synchronizing Things This Way

Call this before starting the timer, before ending the timer, and before retrieving data from an array computed in an OpenCL program.

Getting Event Statuses Without Blocking

Call this before starting the timer, before ending the timer, and before retrieving data from an array computed in an OpenCL program.