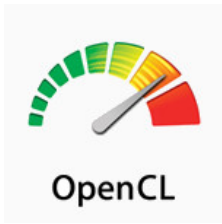


OpenCL Basics: Running multiple kernels in OpenCL



This series "[Basic concepts](#)" is based on GPGPU-questions we get via [email](#) more than once, or when the question is not clearly explained in the books. For one it is obvious, for the other just what they're missing.

They say that learning a new technique is best done by playing around with working code and then try to combine it. The idea is that when you have Stackoverflowed and Githubed code together, you've created so many bugs by design that you'll learn a lot if you make it work. When applying this to OpenCL, you quickly get to a situation that you want to run one.cl file and then another.cl file. Almost all beginner's material discuss a single OpenCL-file, so how to do this elegantly?

The answer is in [clCreateProgramWithSource](#), which takes one or more kernels. If multiple cl-files are compiled (or one big concatenated file), one Program-object is created. But how to use this?

We first need to talk about public and private as we know from C++, as this is in OpenCL cl files too. From the perspective of the host the '__kernel void' functions are public and the normal functions are private. After a Kernel-object for each kernel-function is created, all kernel-functions are callable. Now you can get an output C by running kernel1(A, B) and then kernel2(B, C).