Week 8: GPU programming in OpenCL

For this assignment, the files `vectorAdd.c` and `matrixMult.c` are required and should be provided by the zip. A `makefile` is also included that should, with emphasis on should, make compilation of the code a bit easier. The code should that you're going to write will have to compile and run on the LWP and the machines on the second floor of the Bernoulliborg.

For information on the api, have a look at `https://www.khronos.org/registry/OpenCL/sdk/2.0/docs/man/xhtml/`. Please be aware that compiling the code on the LWP will produce a depreciation warning, but it is only a warning so it will still compile.

Vector addition

The first part of the assignment is mostly meant as an introduction to OpenCL. OpenCL is a framework that needs a some set-up before it can actually offload any work to the GPU. The assignment is to write a program that sets up OpenCL appropriately and then executes a pairwise vector addition, which means that $C_i = A_i + B_i$ for $i < n$ where n is the size of the vector. This can of course be done sequentially but for large vectors there is a lot of speed-up possible by using GPUs.

The file `vectorAdd.c` already set up some groundwork and provides some insights in what sort of calls to make.

Matrix multiplication

To keep this assignment a bit simpler, we assume that only square matrices of size nxn will be used. For the test on nestor we will assume that $n = 1024$, but ensure that it also works for other values of n.

As a small refresher, matrix multiplication is computed by calculating:

$$C_{i,j} = \sum_{k=1}^{n} A_{ik} \cdot B_{kj} \text{ for } i, j \leq n$$

As one can immediatly see, this involves a lot more calculations then vector addition. `matrixMult.c` contains the same sort of framework as the previous assignment. Use the code you've written in `vectorAdd.c` to set up the OpenCL environment but now make it perform matrix multiplication.