Laborator 09

Exercițiile din acest lab se vor rezolva folosind OpenCL, rezultatele vor fi adăugate în README.txt.

1. Rulați programul helloWorld.c compilând din linia de comandă.
2. Rulați programul helloWorld.c din VSCode. E posibil să fie nevoie să modificați path-urile din tasks.json.
3. Modificați codul dat astfel încât să efectueze adunarea a valorii 10 fiecărui element din vectorul inițial.
4. Modificați codul anterior astfel încât să efectueze adunarea valorii 10 doar elementelor cu valori pare din vectorul inițial.
5. Implementați adunarea a doi vectori. Vor trebui alocați și transmiși de pe host pe device.
6. Rulați exercițiul anterior cu un N (număr de elemente și de work-item-uri – thread-uri) de 10 ori mai mare decât numărul de core-uri disponibile. Printați id-ul thread-ului direct de pe device.
7. Implementați adunarea a două matrice folosind un singur thread (work-item). **ATENȚIE**: Între host și device pot fi transmiși doar vectori.
8. Implementați adunarea a două matrice folosind câte un thread (work-item) pentru fiecare linie a matricei.
9. Implementați adunarea a două matrice folosind câte un thread (work-item) pentru fiecare element al matricei.
10. Implementați adunarea a două matrice folosind câte un thread (work-item) pentru fiecare element al matricei. Dimensiunile work-ului vor fi bidimensionale și identice cu dimensiunile matricei. Va trebui să calculați poziția din matrice în funcție de ID-ul thread-ului.
11. Alocați câte o variabilă de tip __local, __private, __global și __constant. Descoperiți care din acestea sunt share-uite între thread-uri și la ce nivel (fiecare e diferită, între toate toate, în interiorul unui work-group). Va trebui să folosiți bariere.
12. Pentru fiecare element dintr-un vector **V** identificați câte numere prime sunt în intervalul $[\ V_i, V_{i+1}\ ]$. Câte un thread pentru fiecare interval.

Resurse:

- Doc OpenCL
- Terminologie
- Workgroups