Лабораторная работа №2. OpenCL

# Задание 1. Стеганография (+4 балла).

***Примечание****: вам не обязательно рассматривать этот массив байт именно как картинку. И тем более, не обязательно использовать соглашение BigEndian. Пример из папки Example\_1 лишь показывает реальное применение.*

1. На вход подаётся файл бинарного формата и строка текста (char). Первые четыре байта файла задают натуральное число N, следующие четыре байта — число M. Далее следуют 4NM байт данных. Ваша задача — зашифровать строку данных, используя последние биты каждого байта (см. схему). Изменённый бинарный файл сохранить по заданному пути (в начале прописать N, M, а затем байты).

Задача должна быть распараллелена при помощи технологии OpenCL.

Исходные байты:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| a0 | a1 | a2 | a3 | a4 | a5 | a6 | a7 |
| b0 | b1 | b2 | b3 | b4 | b5 | b6 | b7 |
| c0 | c1 | c2 | c3 | c4 | c5 | c6 | c7 |
| d0 | d1 | d2 | d3 | d4 | d5 | d6 | d7 |
| e0 | e1 | e2 | e3 | e4 | e5 | e6 | e7 |
| f0 | f1 | f2 | f3 | f4 | f5 | f6 | f7 |
| g0 | g1 | g2 | g3 | g4 | g5 | g6 | g7 |
| h0 | h1 | h2 | h3 | h4 | h5 | h6 | h7 |

Байт текста:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| t0 | t1 | t2 | t3 | t4 | t5 | t6 | t7 |

Зашифрованный текст:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| a0 | a1 | a2 | a3 | a4 | a5 | a6 | t0 |
| b0 | b1 | b2 | b3 | b4 | b5 | b6 | t1 |
| c0 | c1 | c2 | c3 | c4 | c5 | c6 | t2 |
| d0 | d1 | d2 | d3 | d4 | d5 | d6 | t3 |
| e0 | e1 | e2 | e3 | e4 | e5 | e6 | t4 |
| f0 | f1 | f2 | f3 | f4 | f5 | f6 | t5 |
| g0 | g1 | g2 | g3 | g4 | g5 | g6 | t6 |
| h0 | h1 | h2 | h3 | h4 | h5 | h6 | t7 |

Путь ко входному файлу, путь для выходного файла и строка для зашифровки передаются как аргументы командной строки. (2 балла)

1. Решить обратную задачу: для заданного бинарного файла извлечь зашифрованную строку. Путь к зашифрованному файлу передаётся через аргумент командной строки. Расшифрованная строка выводится в консоль. Задача должна быть распараллелена при помощи технологии OpenCL. (2 балла).

# Задание 2. Умножение матриц (+7 баллов).

1. Напишите ядро для умножения double матриц размера NxM и MxL. Вторую матрицу передавайте транспонированной. Исходные матрицы заполняйте при помощи генератора случайных чисел (для упрощения проверки можете использовать только числа -1, 0, 1). Исходные матрицы и результат умножения пишите в файл. (1 балл)
2. Напишите вторую версию ядра так, чтобы использовать векторные типы длины 8. Можете предполагать, что размеры матрицы кратны 8. (1 балл).
3. Снимите ограничение на размеры матриц (1 балл).
4. Напишите непараллельную версию кода С++ для умножения матриц. Сравните время выполнения умножения на непараллельной версии, на ядре со скалярными типами, на ядре с векторными типами. Сделайте вывод. (1 балл.)
5. Напишите генератор кода ядра для векторных типов длины N (N=2,3,4,8,16). Сгенерированный код сохраните во временный файл и затем запустите его. Учтите, что размеры матриц могут быть не кратны N. (2 балла)
6. Сравните время выполнения кода при разной длине векторных типов. Сделайте выводы. (1 балл)