Лабораторная работа №2. Основы OpenCL

## Стеганография (*5,5 баллов*)

* 1. Разработать консольное приложение, на вход которому подаётся путь к бинарному файлу специального формата и текстовая строка. Требуется, используя параллельные вычисления на OpenCL, зашифровать текст в файле. Путь к обрабатываемому файлу и текст передаются как аргументы командной строки (*2,0 баллов*).
  2. Приложение-шифровщик должно наиболее эффективно использовать OpenCL (*+1,0 балла*).
  3. Написать консольное приложение, которое расшифровывает текстовую строку из бинарного файла специального формата. Путь к файлу задаётся через аргументы командной строки. Расшифрованный текст вывести в консоль (*1,5 балла*).
  4. Приложение-расшифровщик должно наиболее эффективно использовать OpenCL (*+1,0 балла*).

### Формат бинарного файла

* В начале два числа int32 N и M, указывающих высоту и ширину картинки.
* Далее — N\*M чисел int32, задающих цвета пикселей.
* Каждый пискель описывается четырься байтами: ARGB, где байт A отвечает за прозрачность, байты R, G и B — за красную, зелёную и синюю компоненты цвета.

Здесь int32 обозначает целочисленное число, занимающее 32 бита.

Для текста можно использовать любую кодировку; самая тривиальная — char (ASCII). Можете также попробовать задать свою таблицу кодов.

Для шифровки используйте последние биты байтов ARGB. Например, просто замените их на биты текста. Конечно, возможно использование и других алгоритмов шифровки.

### Пример шифровки одного байта текста

Исходные байты (8 байт изображения):

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 (8 байт модифицированного изображения):

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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** |

### *Примечание*.

Современный компьютеры могут использовать разные подходы для хранения многобайтовых данных в бинарном формате.

* **Big-endian**. Порядок байтов от старшего к младшему. Например, число

2 712 847 316 = 0xA1B2С3D4 будет записано в виде следующих четырёх байт:

0xA1 0xB2 0xC3 0xD4

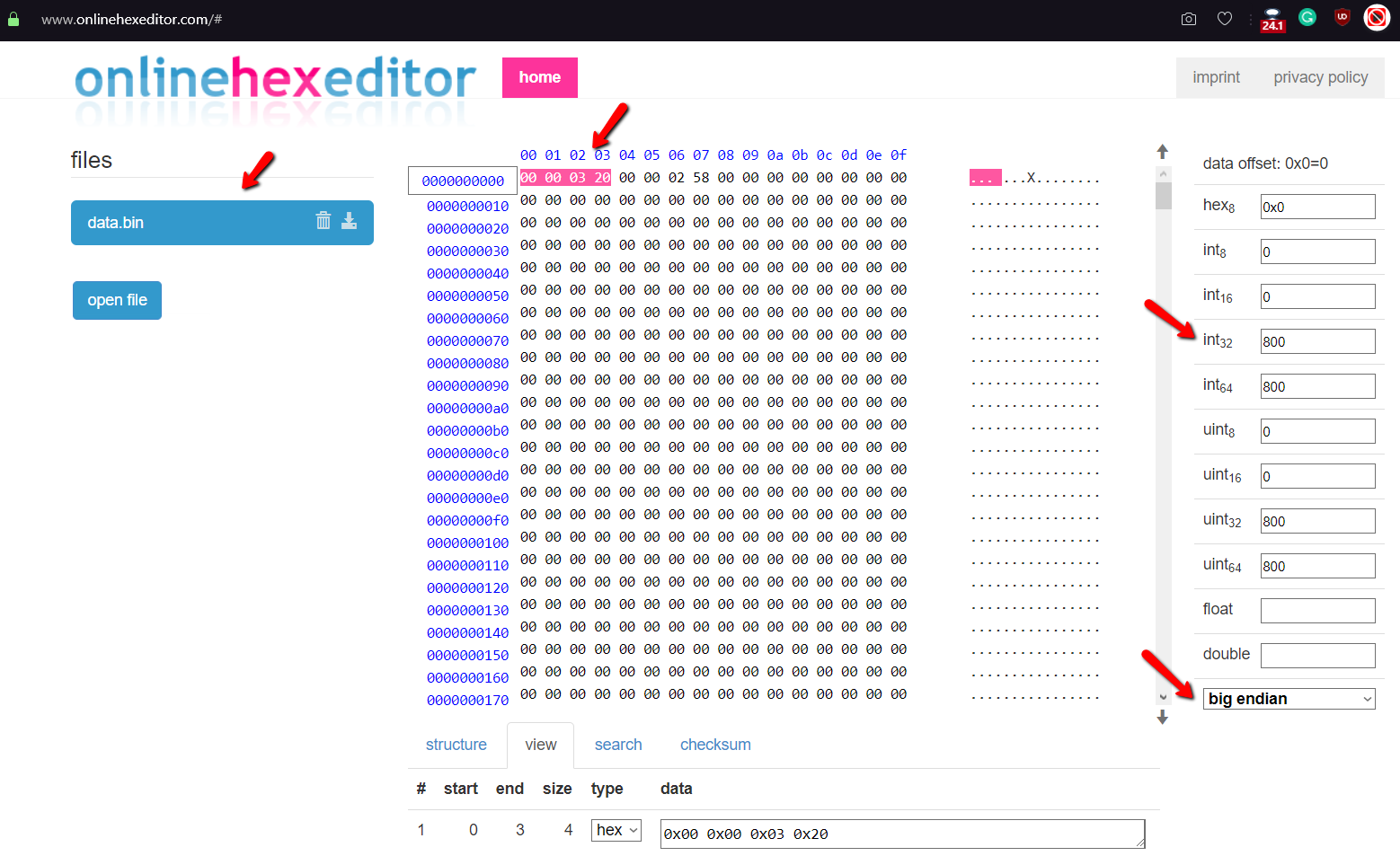
Этот порядок используется в протоколе TCP/IP и во многих стандартах форматах файлов (в частности PNG).

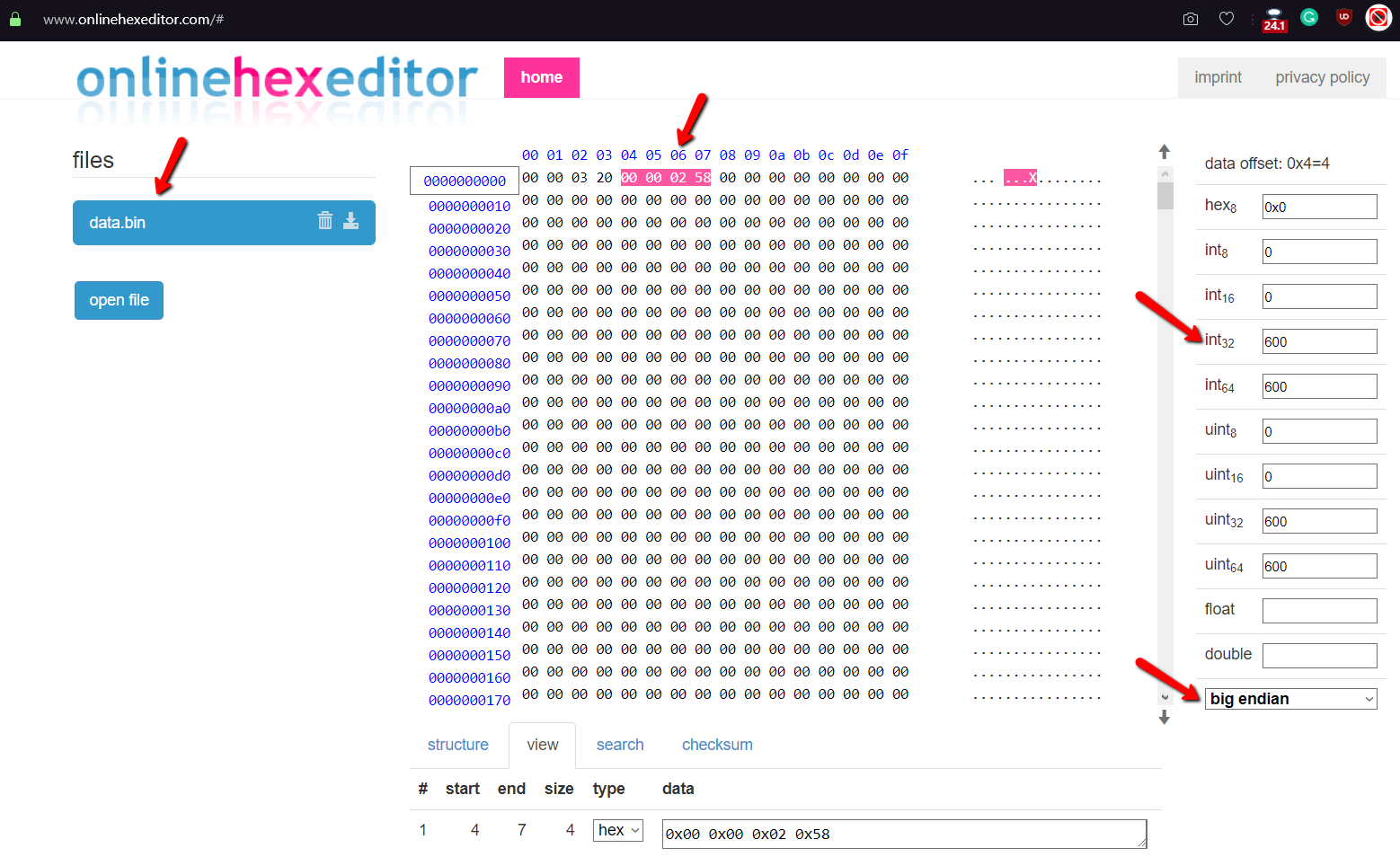
* **Little-endian**. Порядок байтов от младшего к старшему. Число 0xA1B2С3D4 будет записано как

0xD4 0xC3 0xB2 0xA1

Обратите внимание, что биты внутри байтов порядок не меняют.

Этот порядок используется при работе с памятью кмопьютеров архитектуры x86.





## Умножение матриц (*5,0 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. (*1,5 балла*)
  2. Используя методы OpenCL, выведите в консоль число *K* — оптимальный размер вектора для данной задачи (*0,5 балла*).
  3. Напишите второе ядро, в котором разверните цикл по *K* итераций. Используйте векторные типы. Можно считать, что N, M и L делятся на K. (1,0 балла)
  4. Снимите ограничение на делимость N, M и L (*1,0 балла*).
  5. Засеките и сравните скорость работы первого и второго ядра на больших матрицах. Сделайте выводы (*1,0 балла*)