Министерство образования и науки Российской Федерации

Санкт-Петербургский Политехнический Университет Петра Великого

—

Институт кибербезопасности и защиты информации

**ЛАБОРАТОРНАЯ РАБОТА №3**

«Программирование на GPU»

по дисциплине «Языки программирования»

1. Выполнил
2. студент гр. 4851004/10001 Пудалева З.А.

<*подпись*>

1. Выполнил
2. студент гр. 4851004/10001 Ледовской И.П.

<*подпись*>

1. Преподаватель
2. асс. преподавателя Орёл Е.М.

<*подпись*>

1. Санкт-Петербург
2. 2023

**Цель работы.**

Получить начальные навыки программирования на графическом процессоре (GPU).

**Ход работы.**

В качестве варианта работы был выбран брутфорс хэша пароля (алгоритм шифрования – MD5). Далее была написана программа, осуществляющая перебор паролей для хэшей неизвестных заранее заданных паролей следующим образом:

1. Считываются все пароли из файла, содержащего список наиболее употребимых, когда-либо употребимых и т.д. паролей. Назовем список таких паролей известным.
2. Последовательно (на CPU) (один за другим) / параллельно (на GPU) (каждому потоку соответствует свой пароль) для каждого такого известного пароля вычисляется MD5 хэш.
3. Затем сравнивается с хэшами неизвестных паролей. Если два хэша совпадают, то пароли, соответствующие хэшам, совпадают (можно считать, что любому паролю соответствует только один MD5 хэш, и никакой другой). Следовательно, удалось найти неизвестный (зашифрованный) пароль соответствующего MD5 хэша.

На рисунке 1 изображен пример работы программы на GPU.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 – Пример работы программы на GPU.

Для одного неизвестного хэша и базы известных паролей в количестве 10000000, время выполнения на GPU составило 3.545 сек., а на CPU составило 9.582 сек.

На рисунке 2 изображены графики зависимости времени выполнения функций перебора паролей от количества паролей из списка известных паролей на CPU и GPU.

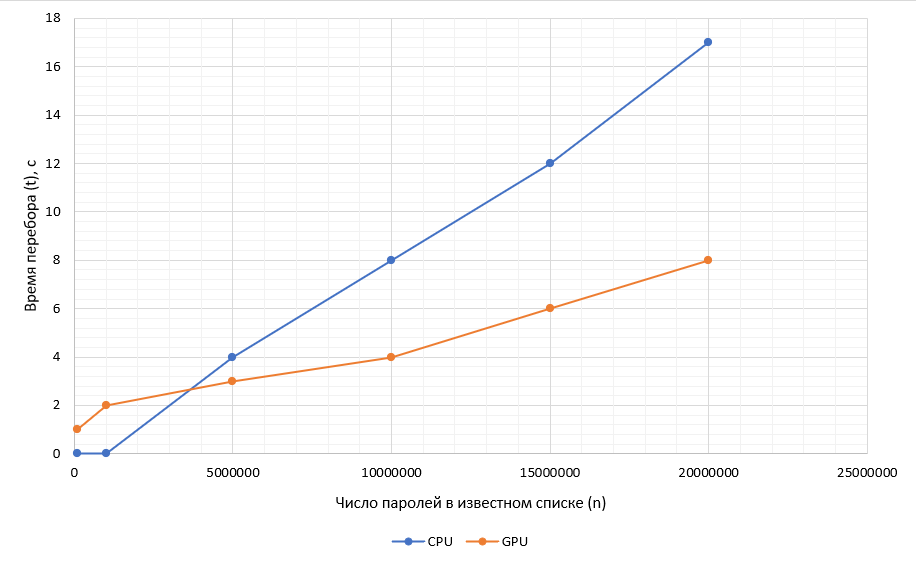


Рисунок 3 – График зависимости времени перебора от количества известных паролей в списке для перебора.

Как видно, оба графика имеют ~линейную зависимость. Причем график функции перебора на CPU растёт существенно быстрее графика для перебора на GPU. Однако, при малом количестве паролей из известного списка, время работы функции на GPU больше, чем на CPU, что можно объяснить долгой инициализацией и подготовкой ядра перед запуском.

**Вывод**

В ходе выполнения работы были получены базовые теоретические и практические навыки программирования на GPU. Разработанная программа для перебора на GPU позволяет существенно ускорить перебор на больших значениях числа известных паролей (), однако на малых значениях n перебор на GPU хуже, чем на CPU.