Password Cracker on GPU using CUDA

Sourabh Kulkarni, 532334

Yashas Bedre, 535744

Architecture and Programming Lab

TUHH

Details:

The submission folder consists of 3 versions of CPU and GPU (CUDA) code for 3-, 4- and 6-letter passwords.

Explanation:

The thread and block ids are used as loop indices to get a password for brute force.

The kernel generates a sequence of characters for one brute force try using the indexes of the thread and block. This generated password is passed onto the md5 algorithm which encodes the password into a 128-bit representation. The encoded bit pattern is matched against the digest of possible passwords stored in the device memory. If the password match is found, the password is stored in the global device memory which is later copied to the host for display processing using printf.

The generation of the brute force passwords help in parallelizing the for loop which generates the passwords in the case of CPU.

Results:

	Time on CPU (s)	Time on GPU (s)
3-letter	0.006155	0.000069
4-letter	0.047774	0.000244
6-letter	31.508266	0.000034

Observations:

- The code execution is faster but only matches only 4 out of 10 passwords from the digest for **3-letter version.**
- The code execution is faster but matches 9 out of 10 passwords from the digest for **4-letter version.**
- The code execution is very fast but matches no (out of 6) passwords from the digest for the **6-letter version.**