System Programming
Assignment 2: Efficient Program Implementation

202111087 백종우

A. Implementation results

size	128	256	512	768	1024
speedup	x3.21	x3.26	x3.24	x3.21	x3.32

B. Optimization approach

A. Successful Strategies

i. **Minimization of Multiplications:** Repeated multiplications within the loop statements were converted to variables to ensure they are calculated only once. Variables that increment consistently with the operation of the loop were optimized by converting them to addition operations.

ii. **Variable Localization:** Variables such as pixels that are repeatedly accessed were localized as local variables, operated upon, and then reassigned to minimize repeated memory accesses.

iii. **Loop Unrolling:** To reduce the number of if statements repeatedly executed within the loop, the boundary parts were handled separately outside the loop, allowing the inner for-loop to operate consistently. This was unrolled to reduce unnecessary instructions.

iv. **Sliding Window Convolution:** The process of accessing overlapping data from six adjacent pixels during the convolution process was optimized to avoid repeated memory accesses by passing the data forward for use in the next operation.

B. Failed Strategies i.

i. **Cache Blocking:** Although the image was divided into smaller sizes to operate in a cache-friendly manner, it did not have a significant effect, possibly due to the small size of the image.

ii. **Inverse Filter:** Instead of referencing nine inputs for one output, the strategy was to decrease memory references by writing to the output as each input is read. However, this changed the output references to nine times per input, which slowed down the process.

C. Additional Ideas

i. **Combining Cache Blocking with Inverse Filtering:** Using cache blocking along with inverse filtering, by storing operations in a small local variable before referencing the output, seemed promising. However, due to the lack of time, it was not possible to implement and test this idea directly.