

목차

1. 실행 결과.....	2
Lab1-1. Hello OpenMP!.....	2
Lab1-2. Encrypted Image	2
2. 소감	3

1. 실행 결과

Lab1-1. Hello OpenMP!

```
H:\Dev\koreatech-assignment\MulticoreProgramming\Assignment1\Lab1-1\cmake-build-debug\Lab1-1.exe
[Thread 0/4] Hello OpenMP!
[Thread 1/4] Hello OpenMP!
[Thread 3/4] Hello OpenMP!
[Thread 2/4] Hello OpenMP!

Process finished with exit code 0
```

"#pragma omp parallel num_threads(4)" 구문을 이용하여, 4개의 스레드가 자신의 스레드 ID와 전체 스레드 수를 출력하는 프로그램을 완성하고, 예상한 것과 동일한 결과를 확인하였다.

Lab1-2. Encrypted Image

```
H:\Dev\koreatech-assignment\MulticoreProgramming\Assignment1\Lab1-2\cmake-build-debug\Lab1-2.exe ../InputData/inputA.out ../InputData/inputB.out 7680 4320 output.out
The results is correct - Good job!
Your computer has 24 logical cores

*      DS_timer Report      *
* The number of timer = 2, counter = 2
**** Timer report ****
Serial Algorithm : 195.86220 ms (195.86220 ms)
Parallel Algorithm : 18.16960 ms (18.16960 ms)
**** Counter report ****
*      End of the report      *
The decryption result was written to output.out

Process finished with exit code 0
```

배열을 스레드 개수로 나눈 뒤, 반복문에 스레드 ID를 기반으로 하여 나뉜 배열 구역을 각 스레드가 작업하도록 코드를 구성하였다.

24개의 스레드를 사용하여, 10.78배 (195.86MS / 18.16MS) 정도의 시간 단축이 이루어진 것을 확인할 수 있었다.



제공된 Viewer.exe를 이용하여 복호화한 이미지를 저장한 뒤 확인해보니 “아름다운 기회의 땅 병렬처리의 세계에 오신 것을 환영합니다! 한학기 동안 병렬처리의 세계를

힘껏 탐험해보요 :)” 문자열을 확인할 수 있었다.

2. 소감

이번 과제를 통해 OpenMP를 활용한 병렬 프로그래밍의 기초적인 개념을 이해하고, 주어진 문제를 해결하는 데 적용해 볼 수 있어서 유익한 경험이었다. 앞으로도 병렬 처리를 이용한 프로그래밍 기법을 계속 연습하고, 다양한 문제에 적용할 수 있는 능력을 키우고 싶다.