Multi-threading

Dynamic Load Balancing

1. Experiment Environment

1. CPU Type: i7-6700
2. RAM: 16GB
3. OS Type: Windows
4. Clock Speed: 3.4 GHz
5. Core#: Octa-core

2. Ex4: Static/Dynamic Load Balancing

1. Static Load Balancing

A. 전체 범위 (1부터 200000)를 스레드의 개수로 나눈다.

B. i번째 스레드는 i번째 구간을 처리한다.

C. 처리가 끝나면 스레드 이름과 실행시간을 출력한다.

2. Dynamic Load Balancing

A. 전체 범위를 NUM_TASK개로 나눈다(50으로 설정)

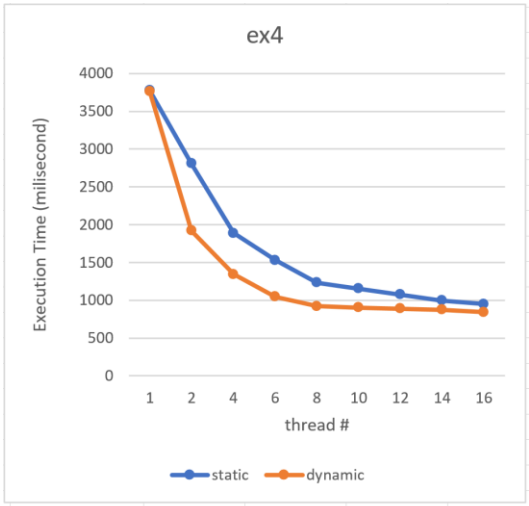
B. i번째 스레드는 먼저 i번째 구간을 처리한다.

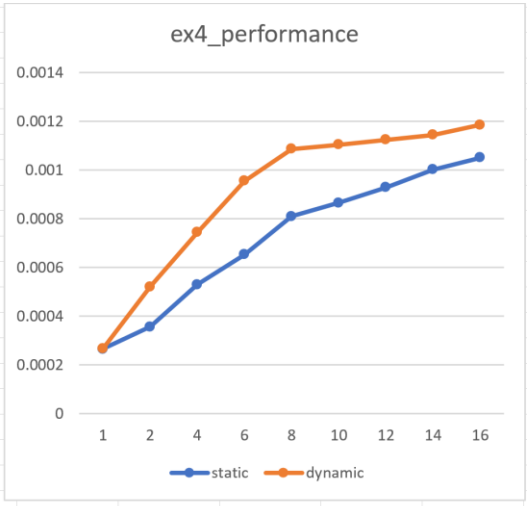
C. 처리가 끝나면 현재 남은 Task 개수를 체크하며 무한루프를 돈다.

D. 무한루프 내에서는 남은 Task를 가져오고, Task를 가져올 때 synchronized 키워드를 사용하여 같은 Task를 처리하는 것을 방지한다.

E. 남은 Task가 없으면 무한루프가 종료되고 스레드 이름과 실행시간을 출력한다.

3. 실행시간 그래프





execute time	1	2	4	6	8	10	12	14	16
static	3775	2809	1887	1530	1234	1155	1077	998	951
dynamic	3759	1920	1343	1047	920	905	889	874	843

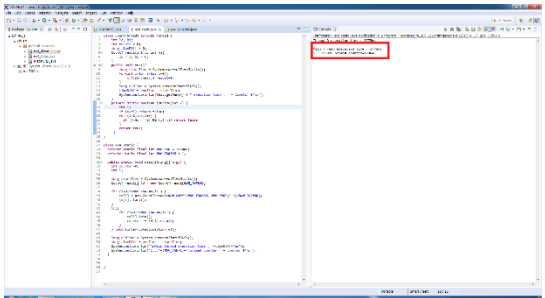
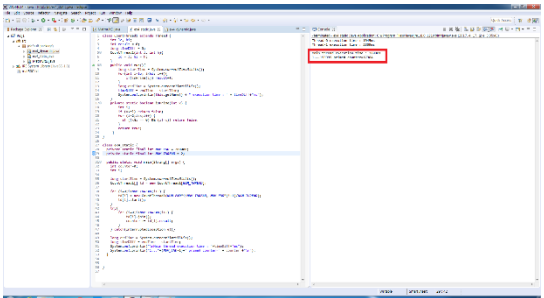
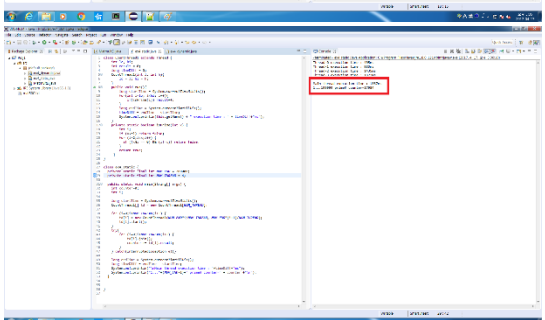
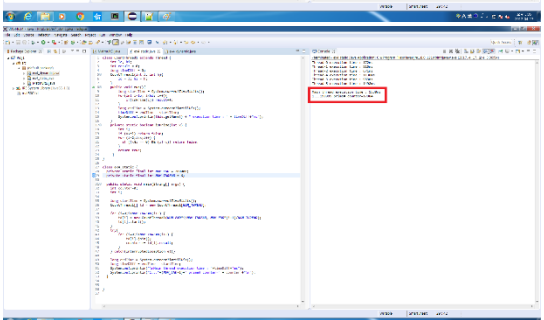
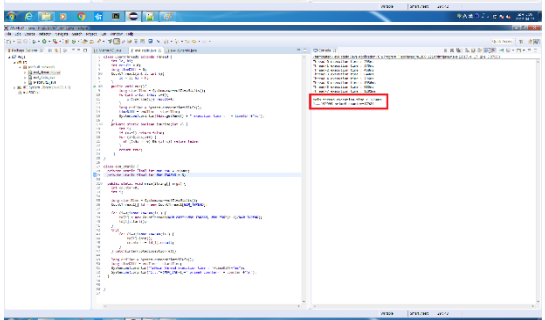
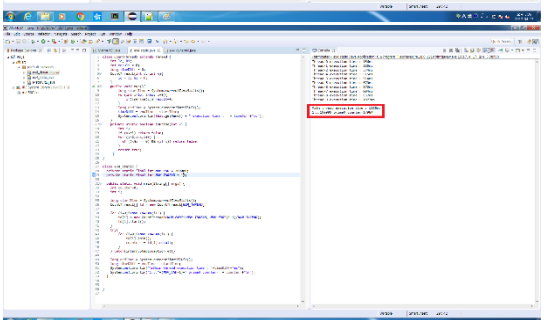
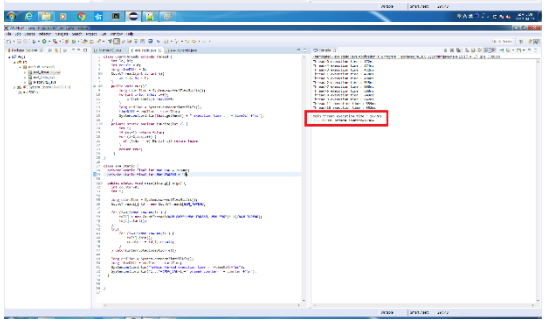
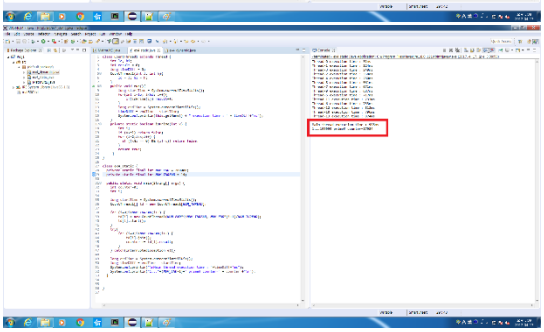
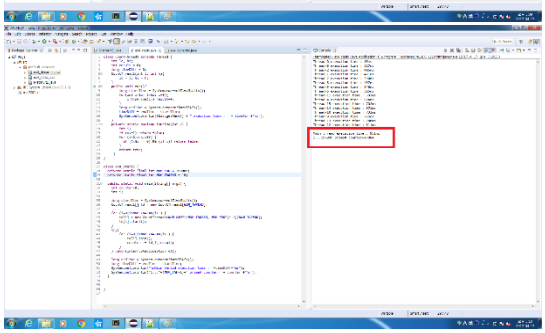
Performance	1	2	4	6	8	10	12	14	16
static	0.000265	0.000356	0.00053	0.000654	0.00081	0.000866	0.000929	0.001002	0.001052
dynamic	0.000266	0.000521	0.000745	0.000955	0.001087	0.001105	0.001125	0.001144	0.001186

- ex4 그래프: x축은 스레드 개수, y축은 실행 시간을 의미한다.
- ex4_performance 그래프: x축은 스레드 개수, y축은 성능(1/실행시간)을 의미한다.

1. Static/Dynamic 둘 다 실행시간이 점점 감소(성능이 증가)함을 알 수 있다.
2. 스레드가 어느정도 커지면 실행 시간 감소량이 줄어드는(성능 향상이 더디어지는) 것을 볼 수 있다. 이는 뒤에 나오는 수가 소수인지 판별하는 데에 오래 걸리고, 소수 자체도 드문드문 존재하기 때문이다.
3. 스레드 개수가 2배 늘어난다고 실행 시간이 2배 줄어드는 것은 아님을 확인할 수 있다. 이는 멀티 스레딩을 통해 성능을 향상시키는 데에는 한계가 있고, 나머지 부분 (메인 스레드 혼자 처리하는 부분)이 성능 향상에 많은 영향을 미침을 알 수 있다.

4. 별첨(실행 결과)

1. ex4_static (스레드 1부터 순서대로 증가, 확대해서 자세히 확인할 수 있습니다).

2. ex4_dynamic (스레드 1부터 순서대로 증가, 확대해서 자세히 확인할 수 있습니다).