알고리즘 실행속도 비교

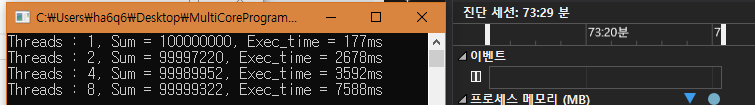
2016184017 박하연

* **실행속도 비교표**

1. **노트북**

|  |  |  |  |
| --- | --- | --- | --- |
| **스레드 수 / 프로그램** | **No mutex** | **With mutex** | **Bakery Algorithm** |
| **1** | 137ms | 1806ms | 177ms |
| **2** | 72ms | 1641ms | 2678ms |
| **4** | 53ms | 1877ms | 3592ms |
| **8** | 56ms | 2267ms | 7588ms |
| **16** | 53ms | 2372ms | X |

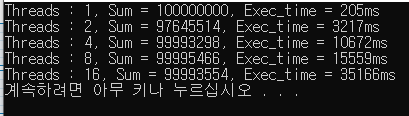
* 실습 7의 프로그램은 data race를 줄이기 위해 코드를 재작성 하기 전의 프로그램이었기 때문에 With Mutex 프로그램의 속도 차이가 많이 났다.
* No mutex 프로그램의 경우 atomic을 사용하지 않았고, thread 한 개를 사용할 때 외에는 sum이 1억보다 작게 나왔다.
* 노트북으로 Bakery Algorithm 프로그램을 실행했을 때 아래의 사진처럼 16스레드가 1시간이 지나도 결과가 나오지않아 중단하고 데스크톱으로 다시 시도하였다.



1. **데스크톱**

|  |  |  |  |
| --- | --- | --- | --- |
| **스레드 수 / 프로그램** | **No mutex** | **With mutex** | **Bakery Algorithm** |
| **1** | 158ms | 1136ms | 205ms |
| **2** | 330ms | 1882ms | 3217ms |
| **4** | 377ms | 2693ms | 10672ms |
| **8** | 417ms | 7058ms | 15559ms |
| **16** | 492ms | 12614ms | 35166ms |

* 데스크톱으로 시도하니 빵집 알고리즘의 16스레드의 결과를 볼 수 있었다.
* No Mutex, Bakery Algorithm 프로그램의 합이 1억이 안되었지만 Bakery Algorithm의 경우 오차범위 내에서 결과가 나왔다.
* 전체적으로 노트북으로 실행했을 때 보다 시간이 더 걸렸는데 Intel과 AMD의 차이라고 생각했다.



* **CPU의 종류**

|  |  |  |
| --- | --- | --- |
|  | 노트북 | 데스크톱 |
| 모델명 | Intel® Core™ i7-4720HQ | AMD Ryzen 7 1700 |
| 코어 개수 | 4 | 8 |
| 스레드 개수 | 8 | 16 |
| 클럭 | 2.60 GHz | 3.00 GHz |

* **실행 시간이 30분 이상 걸리는 원인에 대한 예측**

노트북과 데스크톱은 코어와 스레드 개수의 차이밖에 없었는데, 노트북에서는 16스레드 이상 돌아가지 않았고 데스크톱에서는 돌아갔던 것을 보아 빵집 알고리즘은 CPU의 스레드 수 이상에서는 돌아가지 않는다고 예측해보았다.