37

병행성을 정리하는 대화

교수: *어때, 머리가 슬슬 아파오지?*

학생: *(해열진통제 두 알을 먹으며) 음, 조금이요. 쓰레드가 서로 얽히는 모든 경우를 생각하기는 어려운 것 같아요.*

교수: *사실 그렇지. 병행되는 코드 몇 줄이 이해가 안되는 걸 보면 놀라곤 하지.*

학생: *저도 그래요! 컴퓨터 과학자로서 다섯 줄의 코드도 이해할 수 없다는 것이 당혹스럽기도 해요.*

교수: *오, 너무 기분 상해하지 말게. 병행 알고리즘에 대한 초기 논문들을 살펴보면 잘못된 것들이 가끔 있다네! 그리고 저자들은 보통 교수들이지!*

학생: *(헉) 교수님들이 ... 어... 틀릴 수 있다고요?*

교수: *그래, 그게 사실이긴 하지만 누구에게도 말하면 안 되는 영업 비밀 중에 하나야.*

학생: *비밀을 지킬게요. 하지만 만약 병행 코드가 그렇게 어렵다면, 그리고 제대로 구현하기 어렵다면, 올바른 병행 코드를 어떻게 작성할 수 있죠?*

교수: *그것이 제대로 된 질문이 되겠군, 그렇지? 간단한 몇 가지에서 시작할 수 있다고 생각하네. 먼저 간단하게 만들어야 하네! 쓰레드 간의 복잡한 상호 동작을 피하고, 잘 알려진 그리고 확실하게 검증된 기법을 사용하여 쓰레드들의 동작을 관리해야 하네.*

학생: *간단한 락이나 생산자-소비자 큐와 같은 것을 말씀하시는 건가요?*

교수: *정확하네! 그런 것들이 일반적인 패러다임이야. 자네가 배운 것들을 기반으로 동작 가능한 해법을 만들어 낼 수 있을 것이네. 둘째는 정말 필요할 때 병행성을 사용하되 피할 수 있으면 전적으로 피해야 하네. 어설프게 최적화된 프로그램처럼 나쁜 것은 없다네.*

학생: *그렇군요. 그러면 쓰레드가 필요 없는 데 사용하는 이유는 무엇인가요?*

교수: *바로 그거야. 세 번째는 정말 병렬화가 필요하다면 다른 단순화된 방식을 찾아봐야해. 예를 들어 병렬 데이터 분석 코드를 위한 맵리듀스 기법은 락이나 컨디션 변수 또는 다른 우리가 다뤘던 골치아픈 녀석들이 갖는 복잡성을 전혀 사용하지 않고도 병렬성을 달성하는 훌륭한 예제이지.*

학생: *맵리듀스요? 흥미진진하게 들리는군요. 찾아 읽어봐야겠어요.*

교수: *훌륭해! 꼭 읽어보게. 결국에는 그런 것을 아주 많이 해야 할 거야. 우리가 배운 주제들은 소개에 지나지 않아. 읽고, 또 읽은 후에 좀 더 읽어보게! 그리고 무언가*

시도해 보고 코드도 작성해 보고 그리고 나서 좀 더 작성해 보게나. Gladwell이 "아웃라이어(Outlier)"에서 말했듯이 대략 10,000 시간을 투자해야 그 분야에 진정한 전문가가 된다네. 수업시간만으로는 그 시간을 다 채울 수가 없지.

학생 : *와우, 우울하게 받아 드려야 할지 희망적으로 받아드려야 할지 잘 모르겠어요. 하지만 후자라고 믿고 시작해 보겠습니다! 병행 코드를 좀 더 작성해봐야 할 시간이 되었네요...*