15

메모리 가상화에 관한 대화

학생 : *이제 가상화에 대해서는 모두 끝난 건가요?*

교수 : *아니야!*

학생 : *그렇게 흥분하실 이유는 없잖아요. 저는 그냥 여쭤본 것 뿐인데. 학생은 질문할 수 있지요, 그렇죠?*

교수 : *음, 교수들이 항상 그렇게 말하곤 하지. 하지만 그들이 말하는 의미는 다음과 같네. "질문을 해라,* **다만** *정말 좋은 질문이라면,* **그리고** *그에 대해서 조금이라도 생각해 보았다면".*

학생 : *음, 확실히 기를 꺾어 놓으시는군요.*

교수 : *임무 완수. 그렇다고 가상화를 끝낸 게 아니야. 오히려 우리는 단지 CPU를 가상화하는 방법만을 마친거고, 옷장 속에 진짜 큰 괴물인 메모리가 기다리고 있지. 메모리 가상화는 복잡하고 하드웨어와 운영체제의 상호작용에 관해 더 많은 상세 사항들을 이해해야만 하네.*

학생 : *멋진데요. 왜 그렇게 어렵습니까?*

교수 : *음, 많은 세부 내용들이 있고, 자네는 그러한 내용들을 머릿속에 잘 정리해야 하네. 그래야 무슨 일이 벌어지고 있는지를 이해하기 위한 모델을 만들 수가 있지. 우리는 베이스-바운드와 같은 간단하고 매우 기본적인 기술을 시작으로 나중에는 TLB와 멀티 레벨 페이지 테이블 같은 문제들도 해결할걸세. 궁극적으로 완전한 기능을 갖춘 현대 가상 메모리 관리자의 동작을 설명할 수 있을 것일세.*

학생 : *좋아요! 정보의 홍수와 수면 부족으로 고생하는 불쌍한 학생들을 위해서 팁 좀 주세요.*

교수 : *수면 부족, 그것은 간단하게 해결할 수 있지. 좀 덜 놀고 더 자도록 하게. 가상 메모리를 이해하기 위해서는 여기서 부터 시작해야 하네.* **사용자 프로그램이 생성하는 모든 주소는 가상주소이지**. *운영체제는 각 프로세스에게 단지 환상을 제공하지. 구체적으로 프로세스가 자신만의 커다란 전용 메모리를 가진다는 환상을 제공하는 것이지. 하드웨어로부터 약간의 도움을 얻어 운영체제는 이 가장된 가상 주소를 실제 물리 주소로 변환하고 원하는 정보의 위치를 찾을 수 있네.*

학생 : *그렇군요. 외울 수 있을 것 같습니다 ... (자신에게) '사용자 프로그램의 모든 주소는 가상이다. 사용자 프로그램의 모든 주소는 가상이다. 사용자 ...'*

교수 : *뭐라고 중얼거리는 건가?*

학생 : *아, 아무것도 … (어색한 침묵) … 어쨌든, 왜 운영체제는 이런 환상을 제공하길 원하는 거죠?*

교수 : **사용하기 쉬운** *시스템을 제공하기 위하여 일세. 운영체제는 각 프로세스에게 코드와 데이터를 저장할 수 있는 대용량의 연속된* **주소 공간**(***address space***)*을 가지고 있다는 시각을 제공하고자 하는거지. 프로그래머는 "이 변수를 어디에 저장해야 하는 거지?"와 같은 걱정은 하지 않아도 된다는 거야. 프로그램의 가상 주소 공간이 크고 변수 등을 위한 많은 공간을 가지고 있기 때문이지. 만일 코드와 데이터를 작고 붐비는 공간에 넣을 걱정을 해야만 했다면 프로그래머로서의 인생은 더 힘들어졌을 거야.*

학생 : *다른 이유는 없나요?*

교수 : *음, …* **고립**(***isolation***)*과* **보호**(***protection***)*도 중요한 이유지. 우리는 잘못된 프로그램이 다른 프로그램의 메모리를 읽고, 더 안 좋게는 덮어 쓸 수 있도록 만들고 싶지는 않지. 그렇지 않나?*

학생 : *아마 싫어하겠죠. 싫어하는 사람이 작성한 프로그램이 아니라면 말이죠.*

교수 : *음, … 다음 학기 시간표에 도덕과 윤리 수업을 추가해야 할 것 같구나. 운영체제 수업이 윤리를 가르치는 건 아니니까.*

학생 : *아마 그래야 할지도 모르겠어요. 하지만, 프로세스의 잘못된 행동에 대한 운영체제의 올바른 대응은 해당 프로세스를 죽이는 것이라고 제가 가르친 건 아니라는 것을 기억해 주세요.*