

이 자료는 운영체제의 가상 메모리 개념에 대해 설명합니다. 주요 내용은 다음과 같습니다:

1. 초기 시스템: 초기의 운영체제는 간단한 구조로, 하나의 프로그램이 메모리에 상주하고 다른 프로그램과 공유하지 않았습니다. 메모리 가상화는 존재하지 않았습니다.

2. 멀티프로그래밍과 시분할: 컴퓨터가 고가의 장비였기 때문에, 여러 프로그램이 메모리를 공유하는 멀티프로그래밍이 도입되었습니다. 운영체제는 시분할 시스템을 통해 여러 사용자가 동시에 컴퓨터를 사용할 수 있도록 했습니다.

3. 주소 공간: 가상 주소 공간의 개념이 도입되어, 각각의 프로세스는 자신만의 독립된 주소 공간을 가지게 됩니다. 이 주소 공간에는 코드, 스택, 힙이 포함되며, 운영체제는 이를 관리합니다.

#### 4. 가상 메모리의 목표:

투명성: 프로그램은 가상 메모리의 존재를 인지하지 않고, 자신이 물리 메모리를 전용하는 것처럼 동작해야 합니다.

효율성: 운영체제는 가상화가 시간적, 공간적으로 효율적으로 동작하도록 해야 합니다.

보호: 프로세스 간의 메모리 보호가 필요합니다. 다른 프로세스나 운영체제의 메모리에 접근하지 못하게 해야 합니다.

5. 결론: 운영체제는 메모리 가상화 기능을 통해 프로세스들이 독립적으로 동작할 수 있도록 하며, 이를 위해 하드웨어와 협력하여 가상 주소를 물리 주소로 변환합니다.

시험 대비로는 가상 메모리의 개념과 멀티프로그래밍, 시분할, 그리고 운영체제가 어떻게 가상 메모리의 목표(투명성, 효율성, 보호)를 달성하는지에 대한 이해가 중요합니다.