

이 문서는 운영체제에서 메모리 가상화와 관련된 주소 변환 메커니즘을 설명하는 내용입니다. 주요 내용은 다음과 같습니다:

1. 제한적 직접 실행(Limited Direct Execution, LDE): 대부분의 프로그램은 하드웨어에서 직접 실행되지만, 시스템 콜이나 타이머 인터럽트 시 운영체제가 개입하여 제어합니다.

2. 메모리 가상화: 프로그램은 물리적 메모리를 직접 다루지 않고 가상 주소를 사용하며, 하드웨어가 이를 물리 주소로 변환합니다. 운영체제는 이를 통해 각 프로그램이 독립적인 메모리 공간을 갖는 것처럼 보이게 하며, 메모리 접근을 제어합니다.

3. 주소 변환 방식:

하드웨어-기반 주소 변환을 통해 가상 주소를 물

리 주소로 변환.

베이스와 바운드 레지스터를 사용하여 프로그램을 특정 메모리 영역에 위치시키고, 해당 영역을 벗어나면 예외가 발생합니다.

동적 재배치: 실행 중에도 프로그램의 메모리 위치를 변경할 수 있습니다.

4. 운영체제의 역할:

새로운 프로세스 생성 시 메모리 할당.

프로세스 종료 시 메모리 회수.

문맥 교환 시 베이스와 바운드 레지스터 값을 저장하고 복원하는 작업.

예외 발생 시 핸들러를 통해 적절한 처리 수행.

5. 효율성 및 문제점:

동적 재배치는 효율적이지만 내부 단편화로 인해 메모리 낭비가 발생할 수 있습니다.

이를 해결하기 위해 세그멘테이션 등의 기법이 필요합니다.

시험 대비 요점:

제한적 직접 실행이란 무엇인지, 언제 운영체제가 개입하는지 이해하세요.

주소 변환의 원리: 가상 주소가 어떻게 물리 주소로 변환되는지 설명할 수 있어야 합니다.

베이스와 바운드 레지스터의 역할과 동작 원리.

운영체제가 메모리 가상화를 어떻게 지원하는지 구체적인 예를 통해 이해하세요.

내부 단편화와 이를 해결하기 위한 기법(세그멘테이션 등)에 대한 개념을 정리하세요.