문서의 23장: 페이징 - 더 작은 테이블의 핵심 내용을 요약하면 다음과 같습니다.

1. 문제점: 페이지 테이블 크기

페이지 테이블이 크면 메모리 낭비가 심해집니다. 예를 들어, 32비트 주소 공간에서 페이지 테이블은 수백 MB의 메모리를 차지할 수 있습니다. 이 문제를 해결하기 위한 방법들이 논의됩니다.

2. 해결책 1: 더 큰 페이지

페이지 크기를 증가시키면 페이지 테이블 항목 수가 줄어들어 테이블 크기가 감소하지만, 내부 단편화(메모리 낭비)가 발생할 수 있습니다.

3. 해결책 2: 하이브리드 접근 - 페이징과 세그멘트 결합

세그멘테이션과 페이징을 결합하여 주소 공간을 효율적으로 관리하는 방식입니다. 각 세그먼트에 별도의 페이지 테이블을 두어 메모리 낭비를 줄일 수 있습니다.

4. 해결책 3: 멀티 레벨 페이지 테이블

페이지 테이블을 트리 구조로 나누어 필요한 부분만 할당하

는 방식입니다. 이 방법을 통해 사용되지 않는 공간을 줄일 수 있습니다. 다만, 페이지 테이블을 참조할 때 더 많은 메모리 접근이 필요하다는 단점이 있습니다.

5. 역 페이지 테이블

시스템 전체에 하나의 페이지 테이블만 사용하는 방식으로, 물리 메모리를 기준으로 페이지를 관리합니다. 전체 메모리를 검색해야 하므로 탐색 속도가 느릴 수 있지만 해시 테이블을 통해 이를 개선할 수 있습니다.

6. 페이지 테이블 스와핑

페이지 테이블을 물리 메모리뿐 아니라 디스크로 스와핑할 수 있는 방법을 설명합니다. 메모리 부족 시 페이지 테이블을 디스크로 옮기는 방법으로, 일부 시스템에서 사용됩니다.

시험 대비 포인트:

1. 페이지 테이블의 크기를 줄이기 위한 다양한 방법 (더 큰 페이지, 멀티 레벨, 하이브리드 등)

2. 멀티 레벨 페이지 테이블 구조와 장단점