

이 문서는 운영체제에서의 페이징 성능 향상을 다루며, 특히 ****TLB(변환 색인 버퍼)****를 통해 주소 변환 속도를 높이는 방법에 대해 설명하고 있습니다. 주요 내용을 요약하고 시험 대비용으로 정리하면 다음과 같습니다:

1. TLB(Translation Lookaside Buffer)란?

페이징은 메모리 성능 저하를 유발하는데, TLB는 주소 변환 정보를 캐싱하여 페이징 성능을 개선하는 하드웨어 캐시입니다. TLB가 있어야만 페이징이 실질적으로 사용 가능해집니다.

2. TLB 기본 알고리즘

가상 주소에서 ****VPN(가상 페이지 번호)****를 추출하고, TLB에서 해당 VPN을 찾습니다.

TLB 히트 시, 변환 정보를 바로 사용해 빠르게 물리 주소를 찾을 수 있습니다.

TLB 미스 시, 페이지 테이블에 접근해 변환 정보를 가져와 TLB를 갱신한 후 다시 시도합니다.

3. 예제: 배열 접근

배열의 원소를 접근할 때, 페이지 경계에 따라 TLB 미스가 발생할 수 있습니다. 한 페이지에 여러 배열 원소가 포함되면 TLB 히트가 발생하고, 다른 페이지에 걸쳐 있으면 TLB 미스가 발생합니다.

TLB 미스가 적을수록 성능이 좋아지며, TLB 히트율이 중요합니다.

4. TLB 미스 처리

TLB 미스는 하드웨어 또는 **소프트웨어(운영체제)**에서 처리됩니다. 하드웨어로 처리되면

TLB 갱신 후 명령어를 재실행합니다.

RISC 구조에서는 소프트웨어로 처리하며, 운영 체제가 TLB 미스를 처리하고 다시 실행하는 구조입니다.

5. TLB의 문제: 문맥 교환

문맥 교환 시 TLB는 이전 프로세스의 변환 정보를 가지고 있어야 하는데, 이를 해결하기 위해 **주소 공간 식별자(ASID)**가 도입됩니다.

ASID를 통해 여러 프로세스의 TLB 정보를 구분하여 성능을 유지할 수 있습니다.

6. TLB 교체 정책

새로운 변환 정보를 추가할 때, 기존 정보를 교체해야 하는데, 일반적으로 **LRU(가장 최근에 사

용되지 않은 항목 교체)**나 랜덤 정책을 사용합니다.

7. 실제 TLB

TLB는 일반적으로 32, 64, 128개의 항목을 가지며, 완전 연관 방식으로 설계됩니다. 변환 정보는 TLB 전체에서 병렬적으로 검색됩니다.

시험 대비용 핵심 정리

1. TLB 역할: 주소 변환 속도 향상, 페이징 성능 개선.
2. TLB 히트와 미스: 히트는 빠른 변환, 미스는 페

이지 테이블 참조로 성능 저하.

3. 문맥 교환 문제: ASID로 해결, 여러 프로세스 간 변환 정보 충돌 방지.

4. 교체 정책: LRU나 랜덤으로 TLB 항목 교체.

5. TLB 성능 최적화: 페이지 크기, ASID, 교체 정책 등이 중요.

이 요점을 중심으로 시험 대비를 하면 좋습니다. TLB의 구조와 동작을 이해하고, 문제 해결 과정에서 어떤 방식으로 최적화를 이루는지가 중요합니다.