

〈용어체크〉

병행 vs 병렬 처리

복수 사용자 환경에서 여러 프로세스들을 처리하기 위한 방식으로 병행처리는 1 CPU 시스템에서 시스템의 시간을 나누어 프로세스들이 번갈아 가면서 처리되는 방식이고 병렬처리는 복수개 CPU 시스템에서 프로세스 별로 CPU를 할당하여 처리하는 방식이다.

시스템 로그 파일

로그에는 데이터베이스 항목에 관련된 트랜잭션의 모든 연산들을 기록하며 이 정보는 트랜잭션 실패를 회복하는데 사용된다.

트랜잭션

DBMS에서 데이터를 다루는 논리적인 작업의 단위이며, ACID의 특징을 갖는다.

〈학습내용〉

트랜잭션 시스템 개념

트랜잭션의 특징

병행제어의 필요성

〈학습목표〉

트랜잭션 시스템 개념을 설명할 수 있다.

트랜잭션의 특징에 대해 설명할 수 있다.

병행제어의 필요성을 설명할 수 있다.

Q. 다중 사용자, 인터리빙 시스템에서 병행 제어가 필요한 이유는 무엇일까요?

: 병행 제어를 하지 않으면 여러 가지 문제가 생길 수 있습니다. 갱신 손실 문제, 오손 읽기/임시 갱신 문제, 부정확한 요약 문제, 반복할 수 없는 읽기 문제가 그러한 문제점들입니다.

트랜잭션 시스템 개념

트랜잭션(Transaction)은 데이터를 다루는 일련의 연산들의 집합으로 DBMS에서의 논리적 작업 단위이다. ACID의 특징을 갖는다.
트랜잭션의 데이터베이스의 접근 연산으로는 `read_item(X)`는 읽기, 검색 연산이고 `write_item(X)`는 삽입, 삭제, 변경 연산이다.
읽기 전용 트랜잭션은 데이터를 검색만 하는 트랜잭션이다.
트랜잭션의 읽기 집합은 그 트랜잭션이 읽는 모든 항목들의 집합이다.
트랜잭션의 쓰기 집합은 그 트랜잭션이 기록하는 모든 항목들의 집합이다.

트랜잭션의 특징

트랜잭션의 ACID 성질

- ✓ 원자성(Atomicity)은 한 트랜잭션은 하나의 원자적 수행 단위로, 트랜잭션은 완전히 수행되거나 전혀 수행되지 않아야 한다. (All or Nothing)
- ✓ 일관성 유지(Consistency Preservation)는 트랜잭션을 완전히 실행하면 데이터베이스를 하나의 일관된 상태에서 또 다른 일관된 상태로 바꿔야 한다.
- ✓ 고립성(Isolation)은 하나의 트랜잭션은 다른 트랜잭션들과는 독립적으로 실행되는 것처럼 보여야 한다. 즉 하나의 트랜잭션의 실행은 동시에 실행 중인 다른 트랜잭션의 간섭을 받아서는 안된다.
- ✓ 지속성(Durability)은 일단 한 트랜잭션이 데이터베이스를 변경시키고 그 변경이 완료되면 그 변경은 이후의 어떠한 고장에도 손실되지 않아야 한다.

병행제어의 필요성

적절한 제어 없이 트랜잭션들을 동시에 실행하였을 때 여러 가지 문제가 발생할 수 있다.

- ✓ 갱신 손실 문제 (Lost Update Problem)
- ✓ 오순 읽기/임시 갱신 문제(Dirty Read/Temporary Update Problem)
- ✓ 부정확한 요약 문제(Incorrect Summary Problem)
- ✓ 반복할 수 없는 읽기 문제(Unrepeatable Read Problem)