파일처리

파일 사용의 형식

일괄처리 형식

* 시스템의 부하를 줄이기 위하여 트랜잭션을 일정 기간 모아서 한꺼번에 처리하는 방식 ex) 급여계산, 재고관리, 순차화일의 갱신 등
* 마스터 파일 효율적으로 접근하도록 트랜잭션들을 구성함
* 트랜잭션들을 그룹화하여 처리하는 성능이 주요 관심사

대화 형식 -> 실시간 처리 방식

* 트랜잭션이 발생하는 대로 구성하고 처리함 ex) 예약시스템, 은행업무처리
* 개별 트랜잭션의 처리 성능이 주요 관심사

파일에 대한 기본 연산

연산의 종류

생성, 기록(갱신, 삽입, 삭제), 판독, 개방과 폐쇄

Disk fragment(단편화) -> 디스크 조각모음

단편화의 원인

* 기억장치는 데이터가 입력되면 이를 순차적으로 배열해 내부의 기록 영역에 배치
* 그 중 일부 데이터가 지워지면 해당 영역은 데이터가 지워진 용량만큼 중간이 비게 됨
* 추후 새로운 데이터가 입력되면 빈 영역부터 채우게 됨
* 이러한 과정이 계속되면 많은 조각이 생김
* 제거(조각 모음)

주기억 장치

* 비교 연산 등 주요연산의 수행 횟수로 평가

알고리즘의 평가척도

복잡도 – 알고리즘에서 수행되는 주요연산의 회수

데이터 접근시간은 모두 일정한 것으로 가정

보조 저장 장치

* 데이터 접근 시간이 메인 메모리에 비해 매우 길다
* 메인 메모리에서의 연산수행시간은 무시될 정도로 작음
* 보조 저장 장치의 접근 횟수가 프로그램 성능 평가 요소
* 파일 구조 선정의 중요성

파일 구조 선정 요소

1. 가변성

파일을 성격

* 내용이 변하지 않는 정적 파일
* 주로 과거의 기록
* 내용이 자주 변하는 동적 파일
* 현재의 상황 데이터
* 전체 레코드 수에 대해 추가되거나 삭제되는 레코드 수
* 가변성이 높은 동적 파일은 빠른 접근과 갱신이 필요 ex) 직접 파일
* 가변성이 낮은 정적 파일은 일괄처리가 유리 ex) 순차 파일

1. 활동성

* 주어진 기간 동안에 파일의 총 레코드 수에 대해 접근한 레코드 수의 비율
* 활동성이 높으면 순차화일 구조가 유리
* 낮으면 직접화일이 유리

1. 사용 빈도수

* 일정 기간 동안의 파일의 사용 빈도수
* 가변성과 활동성에 밀접히 관련
* 사용 빈도수와 파일 구조
* 낮으면 순차
* 높으면 직접

1. 응답 시간

* 검색이나 갱신에 대해 요구하는 지연 시간
* 응답시간은 파일구조와 요구되는 질의에 종속적임
* 임의 접근
* 직접 파일
* 순차 접근

1. 파일 크기

* 레코드 수와 각 레코드 길이가 파일 크기 결정
* 시간이 지남에 따라 파일 크기 성장
* 레코드 길이 확장
* 레코드 수 증가
* 성장을 유연하게 수용할 수 있는 구조 필요
* 정적해싱 vs 동적해싱

1. 파일 접근 유형

파일 접근 유형과 파일 구조

* 연산의 유형과 접근 형식에 따라 파일 구조 결정
* 연산의 유형에 따라
* 판독 위주 접근 vs 갱신 위주 접근
* 접근 형식에 따라
* 순차 접근 vs 임의 접근