2주차

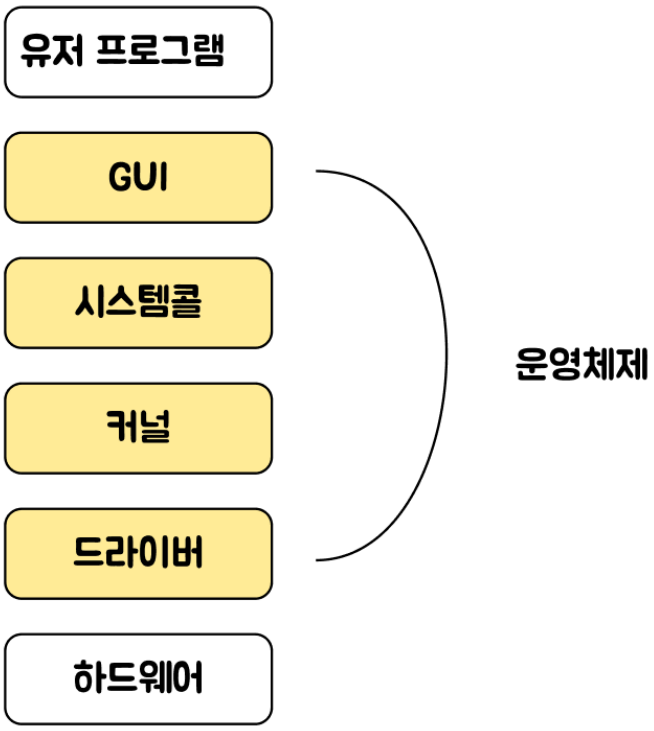
3. 운영체제

3.1 운영체제

운영체제의 역할

- CPU 스케줄링과 프로세스 관리
- 메모리 관리
- 디스크 파일 관리
- I/O 디바이스 관리

운영체제의 구조



3.1.2 컴퓨터 요소

- 컴퓨터는 CPU, DMA 컨트롤러, 메모리, 타이머, 디바이스 컨트롤러 등으로 이루어짐

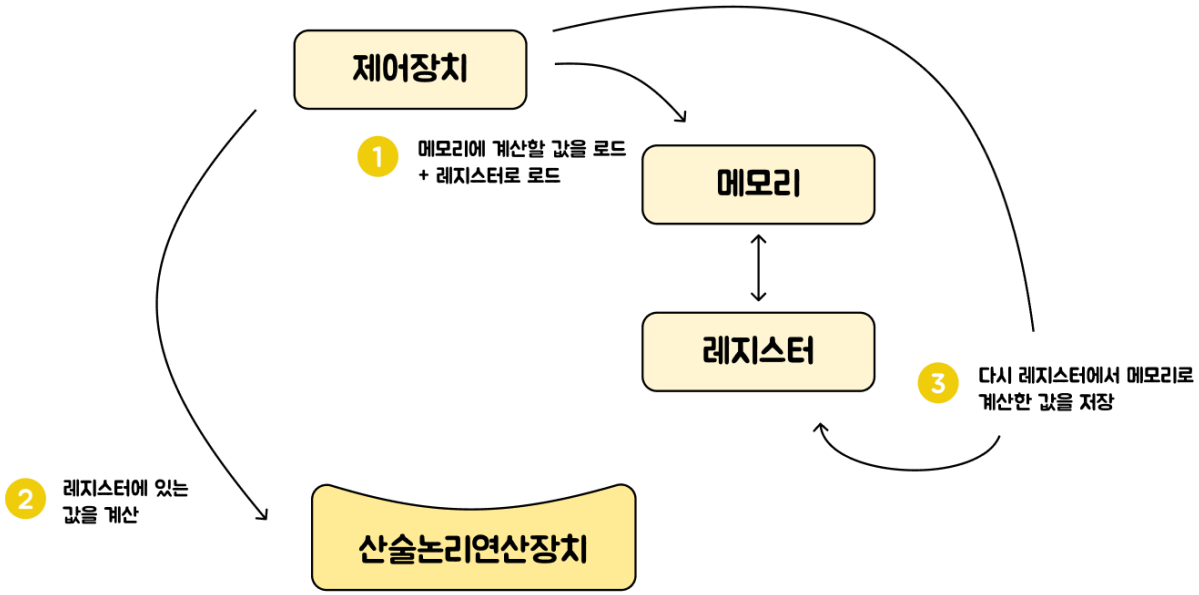
1) CPU

- **산술 논리연산장치, 제어장치, 레지스터로 구성되어 있는 컴퓨터 장치**
- 인터럽트에 의해 메모리에 존재하는 명령어를 해석하여 실행

[CPU 구성요소]

이름	설명
제어장치	▪ 프로세스 조작을 지시하는 CPU의 한 부품 ▪ 입출력 장치 간 통신을 제어하고, 명령어들을 읽고 해석하며 데이터 처리를 위한 순서를 결정
레지스터	▪ CPU 안에 있는 매우 빠른 임시기억장치 ▪ CPU와 직접 연결되어 있으므로 연산 속도가 메모리보다 수십 배에서 수백배 빠름 ▪ CPU는 자체적으로 데이터 저장할 방법이 없기 때문에 레지스터를 거쳐서 데이터 전달
산술논리연산장치	▪ 두 숫자의 산술 연산(덧셈, 뺄셈)과 논리 연산(논리합, 논리곱)을 계산하는 디지털 회로

[CPU 연산 처리]



[인터럽트]

- 어떤 신호가 들어왔을 때 CPU를 잠깐 정지시키는 것
- 하드웨어 인터럽트
  - I/O 디바이스에서 발생하는 인터럽트 *(키보드 연결, 마우스 연결...)*
- 소프트웨어 인터럽트
  - 트랩이라고도 함
  - 프로세스 오류 등으로 프로세스가 시스템콜을 호출할 때 발동

2) DMA 컨트롤러

- I/O 디바이스가 메모리에 직접 접근할 수 있도록 하는 하드웨어 장치
- CPU 부하를 막음
- 하나의 작업을 CPU와 DMA 컨트롤러가 동시에 하는 것을 방지

3) 메모리

- 전자회로에서 데이터나 상태, 명령어 등을 기록하는 장치
- RAM을 메모리라고도 함

- CPU는 계산, 메모리는 기억을 담당

4) 타이머

- 특정 프로그램에 시간 제한을 다는 역할

5) 디바이스 컨트롤러

- 컴퓨터와 연결되어 있는 I/O 디바이스들의 작은 CPU를 말함

3.2 메모리

메모리 계층

계층	설명	휘발성	속도	기억용량
레지스터	CPU 안에 있는 작은 메모리	가장 빠름	가장 빠름	적음
캐시	L1, L2 캐시 지칭	빠름	빠름	적음
주기억장치	RAM을 가리킴	보통	보통	보통
보조기억장치	HDD, SDD를 말함	낮음	낮음	많음

캐시

[의미]

- 데이터를 미리 복사해 놓은 임시 저장소
- 빠른 장치와 느린 장치에서 속도 차이에 따른 병목 현상을 줄이기 위한 메모리
- 소프트웨어적인 대표적인 캐시에는 쿠키, 로컬스토리지, 세션 스토리지가 있음

[지역성의 원리]

- 시간 지역성
  - 최근 사용한 데이터에 다시 접근하려는 특성
- 공간 지역성
  - 최근 접근한 데이터를 이루고 있는 공간이나 그 가까운 공간에 접근하는 특성

[캐시히트와 캐시미스]

- 캐시히트
  - 캐시에서 원하는 데이터를 찾는 것
- 캐시미스
  - 해당 데이터가 캐시에 없다면 주 메모리로 가서 데이터를 찾아오는 것

[캐시매핑]

- 캐시가 히트되기 위해 매핑하는 방법
- CPU의 레지스터와 주 메모리간에 데이터를 주고받을 때를 기반으로 설명

[캐시 종류]

- 쿠키
  - 만료기한이 있는 키-값 저장소
- 로컬 스토리지
  - 만료기한이 없는 키-값 저장소
  - 10mb까지 저장
  - 브라우저를 닫아도 유지
  - HTML5를 지원하지 않는 웹 브라우저에서는 사용 X
- 세션 스토리지
  - 만료기한이 없는 키-값 저장소
  - 탭 단위로 스토리지 생성

    - 5mb까지 저장
    - HTML5를 지원하지 않는 웹 브라우저에서는 사용 X
    - 클라이언트에서만 수정가능

메모리 관리

[가상 메모리]

- 메모리 관리 기법
- 실제로 이용 가능한 메모리 자원을 추상화하여 이를 사용하는 사용자들에게 매우 큰 메모리로 보이게 만드는 것
- 가상적으로 주어진 주소를 가상주소, 실제 메모리상에 있는 주소를 실제 주소

[스와핑]

- 가상 메모리에 존재, 실제 메모리인 RAM에는 현재 없는 데이터나 코드에 접근 → 페이지 폴트 발생
- 스와핑을 통해서 방지
- 당장 사용하지 않는 영역을 하드디스크로 옮겨 필요할 때 다시 RAM으로 불러와 올리고, 사용하지 않으면 다시 하드디스크로 내림을 반복하여 RAM을 관리하는 것

[스레싱]

- 메모리의 페이지 폴트율이 높은 것을 의미
- 컴퓨터의 심각한 성능 저하
- 메모리에 너무 많은 프로세스가 동시에 올라가게 되면 스와핑이 많이 일어나서 발생

[메모리 할당]

- 시작 메모리 위치, 메모리의 할당 크기를 기반으로 할당

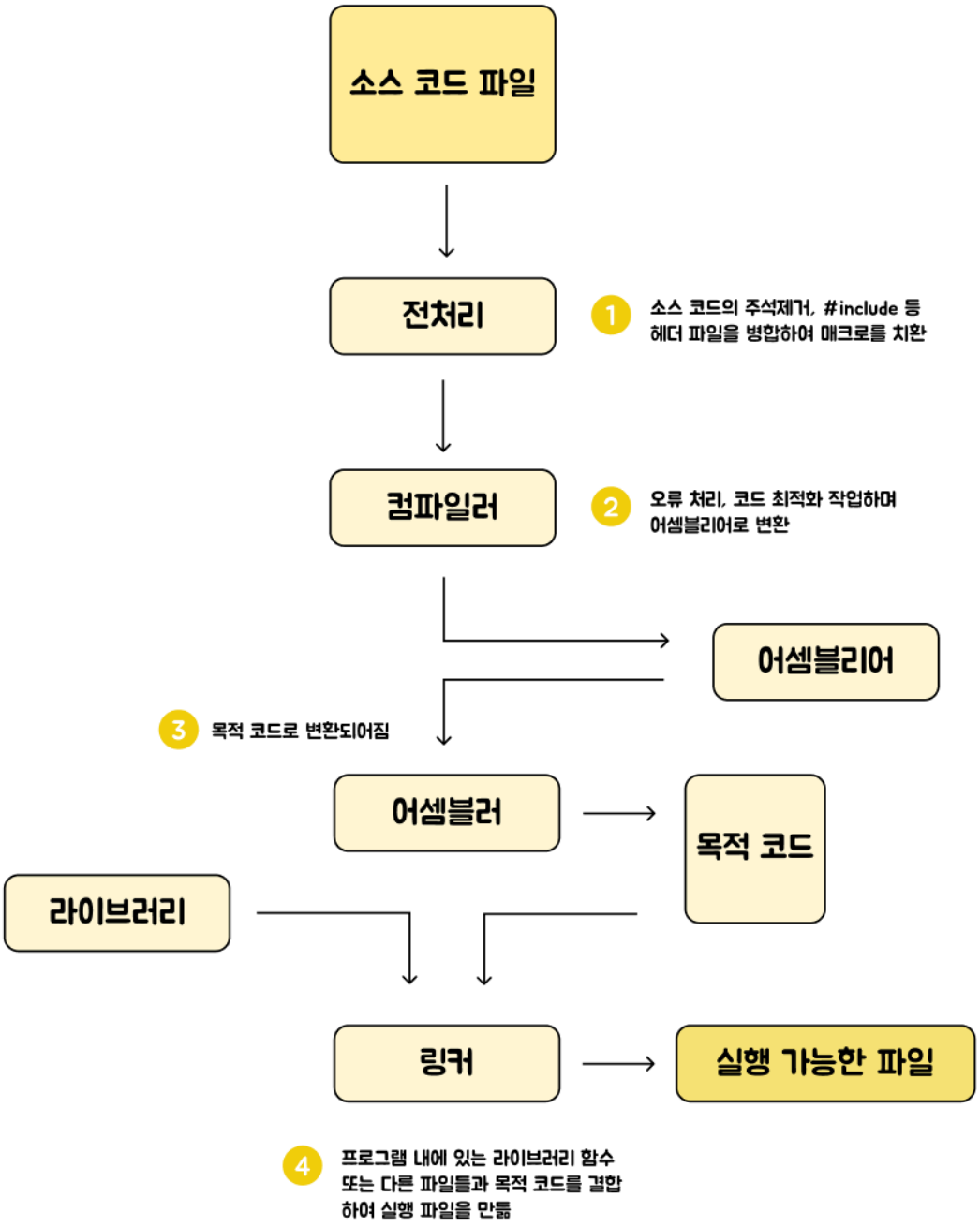
이름	설명
연속할당	▪ 연속적으로 공간을 할당 ▪ 고정 분할 방식 (메모리를 미리 나누어 관리, 내부단편화 O) ▪ 가변 분할 방식 (매 시점 프로그램의 크기에 맞게 동적으로 메모리를 나눔, 내부 단편화 X, 외부단편화 O)
불연속 할당	▪ 연속적으로 공간을 할당 X ▪ 페이징 (동일한 크기의 페이지 단위로 나누어 메모리의 서로 다른 위치에 프로세스를 할당) ▪ 세그멘테이션 (페이지 단위가 아닌 의미 단위인 세그먼트로 나누는 방식) ▪ 페이지드 세그멘테이션 (공유나 보안을 의미 단위의 세그먼트로 나누고, 물리적 메모리는 페이지로 나누는 것)

[페이지 교체 알고리즘]

이름	설명
오프라인 알고리즘	먼 미래에 참조되는 페이지와 현재 할당하는 페이지를 바꾸는 알고리즘
FIFO	가장 먼저 온 페이지를 교체
LRU	참조가 가장 오래된 페이지 교체
LFU	가장 참조 회수가 적은 페이지 교체

3.3 프로세스와 스레드

프로세스와 컴파일 과정


소스 코드 파일
전처리
1
소스 코드의 주석제거, #include 등
헤더 파일을 병합하여 매크로를 치환
컴파일러
2
오류 처리, 코드 최적화 작업하며
어셈블리어로 변환
어셈블리어
3
목적 코드로 변환되어짐
어셈블러
목적 코드
라이브러리
링커
실행 가능한 파일
4
프로그램 내에 있는 라이브러리 함수
또는 다른 파일들과 목적 코드를 결합
하여 실행 파일을 만듦

프로세스의 상태

상태	설명
생성 상태	프로세스가 생성된 상태
대기 상태	메모리 공간이 충분하면 메모리를 할당받고 아니면 아닌 상태로 대기하고 있으며 CPU 스케줄러로부터 CPU 소유권이 넘어오기를 기다리는 상태
대기 중단 상태	메모리 부족으로 일시 중단된 상태
실행 상태	CPU 소유권과 메모리를 할당받고 인스트럭션을 수행 중인 상태를 의미
중단 상태	어떤 이벤트가 발생한 이후 기다리며 프로세스가 차단된 상태
일시 중단 상태	중단된 상태에서 프로세스가 실행되려고 했지만 메모리 부족으로 일시 중단된 상태
종료 상태	메모리와 CPU 소유권을 모두 놓고 가는 상태

프로세스의 메모리 구조

- 스택
  - 지역변수, 매개변수, 함수가 저장
  - 컴파일 시 크기가 결정
  - 동적인 특징
- 힙
  - 동적할당 할 때 사용
  - 런타임시 크기 결정
  - 동적인 특징
- 데이터 영역
  - 전역변수, 정적변수가 저장
  - 정적인 특징을 갖는 프로그램이 종료되면 사라지는 변수가 들어 있는 영역
- 코드 영역

- 프로그램에 내장되어 있는 소스 코드가 들어가는 영역

PCB

[의미]

- 운영체제에서 프로세스에 대한 메타데이터를 저장한 데이터

[구조]

- 프로세스 스케줄링 상태
- 프로세스 ID
- 프로세스 권한
- 프로세스 카운터
- CPU 레지스터
- CPU 스케줄링 정보
- 계정 정보
- I/O 상태 정보

[컨텍스트 스위칭]

- PCB를 교환하는 과정
- 한 프로세스에 할당된 시간이 끝나거나 인터럽트에 의해 발생

멀티프로세싱

[의미]

- 멀티프로세스를 통해 동시에 두 가지 이상의 일을 수행할 수 있는 것을 말함
- 하나의 이상의 일을 병렬로 처리
- 특정 프로세스의 메모리, 프로세스 중 일부에 문제 발생되더라도 다른 프로세스를 이용해서 처리 가능함 → 신뢰성 높음

[IPC]

- 멀티 프로세스는 IPC가 가능
- IPC는 프로세스끼리 데이터를 주고받고 공유 데이터를 관리하는 메커니즘을 뜻함
- IPC 종류로는 공유 메모리, 파일, 소켓, 익명 파이프, 명명 파이브, 메시지 큐 존재함
- 메모리가 완전히 공유되는 스레드 보다 속도 ↓

스레드와 멀티스레싱

[스레드]

- 스레드는 프로세스의 실행 가능한 가장 작은 단위

[멀티스레싱]

- 프로세스 내 작업을 여러 개의 스레드, 멀티스레드로 처리하는 기법
- 스레드끼리 서로 자원을 공유하여 효율성이 높음
- 동시성
- 한 스레드에 문제가 생기면 다른 스레드에도 영향을 끼쳐 스레드로 이루어져 있는 프로세스에 영향을 줄 수 있는 단점 존재

공유자원

- 시스템 안에서 각 프로세스, 스레드가 함께 접근할 수 있는 자원이나 변수 등을 의미
- 공유 자원을 두 개 이상의 프로세스가 동시에 읽거나 쓰는 상태 → 경쟁상태

임계영역

[의미]

- 공유 자원에 접근할 때 순서 등의 이유로 결과가 달라지는 영역
- 해결방법에는 뮤텍스, 세마포어, 모니터 존재
- 위 방법 모두 상호 배제, 한정 대기, 융통성이라는 조건 만족

[해결방법]

- 뮤텍스
  - 공유 자원을 사용하기 전에 설정하고 사용한 후에 해제하는 잠금
- 세마포어
  - 일반화된 뮤텍스
  - 간단한 정수 값과 두 가지 함수 wait 및 signal로 공유자원에 대한 접근을 처리
- 모니터
  - 둘 이상의 스레드나 프로세스가 공유 자원에 안전하게 접근할 수 있도록 공유자원을 숨기고 해당 접근에 대해 인터페이스만 제공

교착상태

- 두 개 이상의 프로세스들이 서로가 가진 자원을 기다리며 중단된 상태

[교착 상태의 원인]

- 상호 배제
  - 한 프로세스가 자원을 독점하고 있으며 다른 프로세스들은 접근이 불가능
- 점유 대기
  - 특정 프로세스가 점유한 자원을 다른 프로세스가 요청하는 상태
- 비선점
  - 다른 프로세스의 자원을 강제적으로 가져올 수 없음
- 환형 대기
  - 서로가 서로의 자원을 요구하는 상황

[해결방법]

- 자원을 할당할 때 조건이 성립되지 않도록 설계
- 교착 상태 가능성이 없을 때만 자원 할당
- 은행원 알고리즘 → 프로세스당 요청할 자원들의 최대치를 통해 자원할당 가능 여부를 파악
- 교착 상태가 발생하면 사이클이 있는지 찾아보고 이에 관련된 프로세스를 한 개씩 지움

3.4 CPU 스케줄링 알고리즘

비선점형 방식

- 프로세스가 스스로 CPU 소유권을 포기하는 방식
- 강제로 프로세스 중지 X

종류	설명
FCFS	가장 먼저 온 것을 가장 먼저 처리
SJF	실행 시간이 가장 짧은 프로세스를 가장 먼저 실행
우선순위	오래된 작업일수록 우선순위를 높이는 방법

선점형 방식

- 현대 운영체제가 쓰는 방식
- 지금 사용하고 있는 프로세스를 알고리즘에 의해 중단시켜 버리고 강제로 다른 프로세스에 CPU 소유권을 할당하는 방식

종류	설명
라운드 로빈	각 프로세스는 동일한 할당 시간을 주고 그 시간 안에 끝나지 않으면 다시 준비 큐의 뒤로 가는 알고리즘
SRF	중간에 더 짧은 작업이 들어오면 수행하던 프로세스를 중지하고 해당 프로세스를 수행하는 알고리즘
다단계 큐	우선순위에 따른 준비큐를 여러 개 사용하고, 큐마다 라운드 로빈이나 FCFS 등 다른 스케줄링 알고리즘을 적용한 것

4. 데이터 베이스

4.1 데이터베이스의 기본

용어 정리

엔티티: 사람, 장소, 물건 등 여러 개의 속성을 지닌 명사

릴레이션: 데이터 베이스에서 정보를 구분하여 저장하는 기본 단위

속성: 릴레이션에서 관리하는 구체적이며 고유한 이름을 갖는 정보

도메인: 릴레이션에 포함된 각각의 속성들이 가질 수 있는 값의 집합

관계

- 1:1 관계 ex) 유저와 유저 이메일
- 1:N 관계 ex) 유저와 상품
- N:M 관계 ex) 학생과 강의

키

- 기본키 (PK)
  - 유일성과 최소성 만족
- 외래키 (FK)
  - 다른 테이블의 기본키를 그대로 참조하는 값
  - 개체와의 관계 식별
- 후보키
  - 기본키가 될 수 있는 후보
  - 유일성과 최소성을 동시에 만족
- 대체키
  - 후보키가 두개 이상일 경우에 어느 하나를 기본키로 지정하고 남은 후보키
- 슈퍼키
  - 각 레코드를 유일하게 식별할 수 있는 유일성을 갖춘 키

4.2 ERD와 정규화 과정

정규화

정규화	설명
제 1정규형	릴레이션의 모든 도메인이 더 이상 분해될 수 없는 원자 값만으로 구성
제 2정규형	릴레이션이 제1정규형이면서 부분함수의 종속성을 제거한 상태 (부분 함수의 종속성 제거란 기본키가 아닌 모든 속성이 기본키에 완전 함수 종적인 것)
제 3정규형	제2정규형이고 기본키가 아닌 모든 속성이 이행적 함수 종속을 만족시키지 않는 상태 (이행적 함수 종속이란 A->B와 B->C 존재하면 A->C 성립, 이때 집합C가 집합A에 이행적으로 함수 종속)
보이스/코드 정규형	제3정규형이고, 결정자가 후보키가 아닌 함수 종속 관계를 제거하여 릴레이션의 함수 종속 관계에서 모든 결정자가 후보키인 상태

4.3 트랜잭션과 무결성

트랜잭션

[정의]

데이터베이스에서 하나의 논리적 기능을 수행하기 위한 작업의 단위

[특징]

- 원자성
  - 트랜잭션과 관련된 일이 수행되었거나 되지 않았거나를 보장하는 특징
- 일관성
  - 허용된 방식으로만 데이터를 변경해야 하는 것을 의미
- 격리성
  - 트랜잭션 수행 시 서로 끼어들지 못하는 것
- 지속성

- 성공적으로 수행된 트랜잭션은 영원히 반영되어야 하는 것을 의미

무결성

[정의]

데이터의 정확성, 일관성, 유효성을 유지하는 것

[특징]

이름	설명
개체 무결성	기본키로 선택된 필드는 빈 값을 허용하지 않음
참조 무결성	서로 참조 관계에 있는 두 테이블의 데이터는 항상 일관된 값을 유지
고유 무결성	특정 속성에 대해 고유한 값을 가지도록 조건이 주어진 경우 그 속성 값은 모두 고유한 값을 가짐
NULL 무결성	특정 속성 값에 NULL이 올 수 없다는 조건이 주어진 경우 그 속성 값은 NULL이 될 수 없다는 제약 조건

4.4 데이터베이스의 종류

관계형 데이터베이스

- 행과 열을 가지는 표 형식 데이터를 저장하는 형태의 데이터 베이스
- SQL이라는 언어를 써서 조작
- MySql, PostgreSQL, 오라클, SQL server, MSSQL 등....

NoSQL 데이터베이스

- SQL을 사용하지 않는 데이터베이스
- MongoDB, redis 등...

4.5 인덱스

B-트리

- 인덱스는 B-트리라는 자료구조로 이루어짐
- 루트 노드, 리프노드, 브랜치 노드로 나뉨

인덱스가 효율적인 이유 및 대수확장성

- 인덱스가 효율적인 이유는 효율적인 단계를 거쳐 모든 요소에 접근할 수 있는 균형 잡힌 트리 구조와 트리 깊이의 대수확장성 때문
- 대수확장성은 트리 깊이가 리프 노드 수에 비해 매우 느리게 성장하는 것을 의미

인덱스 최적화

- 테스트
  - 인덱스를 만들고 쿼리를 보낸 이후에 테스팅을 하며 걸리는 시간을 최소화
- 복합 인덱스는 같은, 정렬, 다중 값, 디널리티 순
  - 인덱스를 생성할 때는 순서가 있고 생성 순서에 따라 인덱스 성능이 달라짐

4.6 조인의 종류

- 하나의 테이블이 아닌 두 개 이상의 테이블을 묶어서 하나의 결과물을 만드는 것

내부조인

- 두 테이블간의 교집합

```
SELECT * FROM TABLEA A
INNER JOIN TABLEB B ON
A.KEY = B.KEY
```

왼쪽 조인

- 테이블 B에서 일치하는 부분의 레코드와 함께 테이블 A를 기준으로 완전한 레코드 집합을 생성

```
SELECT * FROM TABLEA A
LEFT JOIN TABLEB B ON
A.KEY = B.KEY
```

오른쪽 조인

- 테이블 A에서 일치하는 부분의 레코드와 함께 테이블 B를 기준으로 완전한 레코드 집합을 생성

```
SELECT * FROM TABLEA A
RIGHT JOIN TABLEB B ON
A.KEY = B.KEY
```

합집합 조인

- 양쪽 테이블에서 일치하는 레코드와 함께 테이블 A와 테이블 B의 모든 레코드 집합 생성

```
SELECT * FROM TABLEA A
FULL OUTER JOIN TABLEB B ON
A.KEY = B.KEY
```

4.7 조인의 원리

중첩 루프 조인

- 중첩 for문과 같은 원리로 조건에 맞는 조인을 하는 방법
- 랜덤 접근에 대한 비용이 증가 → 대용량 테이블에서 사용 X

정렬 병합 조인

- 각각의 테이블을 조인할 필드 기준으로 정렬하고, 정렬이 끝난 이후에 조인 작업을 수행하는 조인

해시 조인

- 해시 테이블을 기반으로 조인하는 방법

질문

CPU의 구성요소에 대해 설명해주세요

제어장치, 레지스터, 산술논리연산장치로 구성되어있습니다. 제어장치는 프로세스 조작을 지시하는 CPU의 한 부품입니다. 레지스터는 CPU 안에 있는 빠른 임시기억장치입니다. 산술논리연산장치는 두 숫자의 연산을 계산하는 디지털 회로입니다.

교착 상태의 원인은 무엇인가요

원인으로는 상호 배제, 점유 대기, 비선점, 환형대기가 존재합니다.

트랜잭션의 특징에 대해 말해주세요

트랜잭션은 원자성, 일관성, 격리성, 지속성이라는 특징을 갖고 있습니다. 원자성은 트랜잭션과 관련된일이 수행되거나 되지 않는 것을 보장하는 특징입니다. 일관성은 허용된 방식으로만 데이터를 변경해야 하는 것을 의미합니다. 격리성은 트랜잭션 수행시 서로 끼어들지 못하는 것을 말합니다. 지속성은 성공적으로 수행된 트랜잭션은 영원히 반영되어야 한다는 것을 의미합니다.