

문제 1. 도메인 네임 구조와 계층 다음 중 도메인 네임의 계층적 구조에 대한 설명으로 가장 적절한 것은 무엇인가?

- ① 도메인 네임은 왼쪽부터 최상위 도메인(TLD)부터 하위 도메인 순으로 배열되며, 각 레이블은 점(.)으로 구분된다.
- ② 도메인 네임은 점(.)을 기준으로 분리되며, 가장 오른쪽 레이블이 최상위 도메인(TLD)로 기능한다.
- ③ 도메인 네임의 중간 레이블은 항상 국가 코드를 나타내며, 최상위 도메인은 서버 위치를 의미한다.
- ④ 도메인 네임은 계층 구조가 없으며, 모든 레이블이 동일한 중요도를 가진다.
- ⑤ 도메인 네임의 최상위 도메인은 왼쪽에 위치하며, 국제 기구에 의해 고정된 목록으로 관리된다.

문제 2. 로컬 네임 서버의 역할 다음 중 로컬 네임 서버(Local Name Server)에 대한 설명으로 가장 적절한 것은 무엇인가?

- ① 로컬 네임 서버는 클라이언트의 첫 번째 DNS 질의 대상이며, ISP에서 할당받아 해당 결과를 캐싱한다.
- ② 로컬 네임 서버는 전 세계의 도메인 정보를 직접 저장하며, 항상 최신 정보를 유지하기 위해 루트 서버와 동기화한다.
- ③ 로컬 네임 서버는 클라이언트 요청 시 무조건 루트 네임 서버로 질의를 전달하여 응답 속도를 보장한다.
- ④ 로컬 네임 서버는 자신이 관리하는 도메인 영역 내 질의만 처리하며, 외부 질의는 차단한다.
- ⑤ 로컬 네임 서버는 캐시 기능 없이 매번 권한 있는 서버에 직접 질의하여 응답의 정확성을 확보한다.

문제 3. URI 구성 요소의 역할 다음 중 URI의 각 구성 요소에 대한 설명으로 가장 적절한 것은 무엇인가?

- ① scheme은 자원 접근 방식을 지정하며, authority는 반드시 IP 주소만 포함한다.
- ② authority는 도메인 네임이나 IP 주소 등 호스트 정보를 포함하며, path는 자원의 위치를 계층적으로 표현한다.
- ③ query는 항상 JSON 형식의 데이터로 작성되며, fragment는 서버에 전달되어 자원 처리를 보조한다.
- ④ path는 절대 경로로만 표현되며, scheme은 데이터 암호화를 보장한다.
- ⑤ fragment는 클라이언트에서만 처리되며, authority는 자원의 물리적 위치를 직접 지정한다.

문제 4. HTTP 메서드와 멱등성 다음 HTTP 메서드 중, **멱등성(idempotency)**을 보장하지 않는(동일 요청을 여러 번 보내면 결과가 달라질 수 있는) 메서드는 무엇인가?

- ① GET
- ② HEAD
- ③ PUT
- ④ DELETE
- ⑤ POST

문제 5. 캐시 히트의 효과 다음 중 캐시 히트(cache hit)가 발생했을 때의 메모리 접근 및 시스템 성능에 미치는 영향에 대해 올바르게 설명한 것은 무엇인가?

- ① 캐시 히트는 데이터가 캐시에 없는 상태에서 발생하여, 주 메모리 접근을 필요로 하므로 전체 응답 시간이 증가한다.
- ② 캐시 히트는 데이터가 캐시에 존재하여, 빠른 접근을 가능하게 하고 CPU의 대기 시간을 크게 줄여 성능을 향상시킨다.
- ③ 캐시 히트는 캐시 일관성 검증 과정을 추가로 거치므로, 오히려 메모리 접근 시간이 늘어난다.
- ④ 캐시 히트는 데이터가 캐시에 저장되어 있어도, 주 메모리와의 동기화 작업으로 인해 성능 향상 효과가 미미하다.
- ⑤ 캐시 히트는 캐시 미스와 유사하게 동작하여, 데이터 접근 시 항상 동일한 응답 시간을 보장한다.

문제 6. 쿠키의 동작 원리와 보안 강화 다음 중 쿠키(cookie)의 기본 동작 원리와 보안을 강화하기 위한 조치에 대해 올바르게 설명한 것은 무엇인가?

- ① 쿠키는 서버에서 생성되며, 모든 HTTP 요청 시 암호화되어 자동 전송되므로 중간자 공격에 취약하지 않다.
- ② 쿠키는 클라이언트에 저장되어, 이후 동일 도메인에 대한 요청 시 자동으로 포함되며, HttpOnly와 Secure 속성을 사용하여 스크립트 접근 및 전송 보안을 강화할 수 있다.
- ③ 쿠키는 클라이언트의 브라우저 캐시에 저장되므로, 만료 시간이 지나면 브라우저에 의해 즉시 삭제되고 서버에서는 재설정할 수 없다.
- ④ 쿠키는 클라이언트와 서버 간 상태 정보를 유지하기 위해 사용되며, 서버는 클라이언트 쿠키의 내용을 직접 수정할 수 있다.
- ⑤ 쿠키는 오직 클라이언트 측 스크립트에서만 관리 가능하며, 서버는 쿠키 정보를 확인하거나 갱신할 수 없다.

1. 정답: 2 이유: 도메인 네임은 오른쪽 끝 레이블이 최상위 도메인(TLD)이며, 점(.)을 기준으로 계층적으로 분류됩니다. (1)은 순서가 반대이고, (3), (4), (5)는 구조적 사실과 다릅니다.
2. 정답: 1 이유: 로컬 네임 서버는 클라이언트와 가장 가까운 DNS 서버로, ISP에 의해 제공되며 질의 결과를 일정 기간 캐싱하여 응답 속도를 높입니다. 다른 선지들은 DNS 체계의 기본 원리와 다릅니다.
3. 정답: 2 이유: URI에서 authority는 도메인 네임이나 IP 주소를 포함하며, path는 자원의 위치를 계층적으로 나타냅니다. (1)은 authority의 범위를 제한하는 오류가 있고, (3), (4), (5)는 각 요소의 역할을 잘못 기술하고 있습니다.
4. 정답: 5 이유: GET, HEAD, PUT, DELETE는 멍등성을 보장하지만, POST는 요청마다 새로운 리소스를 생성하거나 상태를 변화시킬 수 있어 멍등성이 보장되지 않습니다, PUT은 덮어쓰기라 멍등성의 개념, POST는 새로운 자원 생성
5. 정답: 2 이유: 캐시 히트는 요청한 데이터가 캐시에 이미 존재할 때 발생하므로, 주 메모리 접근 없이 빠르게 데이터를 제공하여 시스템 성능 및 응답 속도를 크게 향상시킵니다.
6. 정답: 2 이유: 쿠키는 서버가 발행하여 클라이언트에 저장되고, 이후 동일 도메인 요청 시 자동 전송됩니다. 또한, HttpOnly와 Secure 속성을 지정하면 클라이언트 측 스크립트 접근을 제한하거나 암호화된 연결에서만 전송하도록 하여 보안을 강화할 수 있습니다.