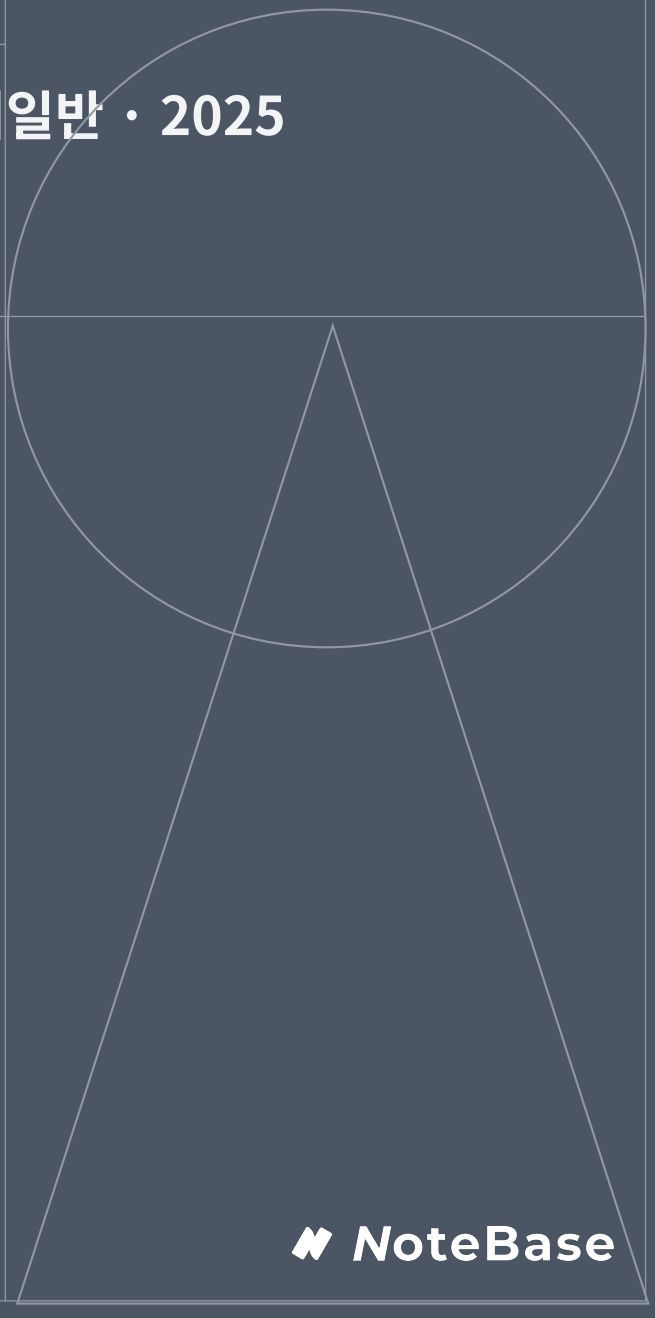
기출OX 해설집

공무원/군무원/9급 · 컴퓨터일반 · 2025



NoteBase

기출OX 해설집

공무원/군무원/9급 · 컴퓨터일반 · 2025

공무원/군무원/9급 · 컴퓨터일반 · 2025

1. 유니코드(Unicode)의 설명으로 가장 적절한 것은?
   ① 문자의 저장 공간을 줄이기 위해 8비트 구조로 설계되었으며 저장 효율이 가장 큰 문자 인코딩 방식이다.
   ② 국가별로 독립적인 문자 코드체계를 유지하는 것이 목표이며 코드 충돌을 방지하기 위해 코드페이지를 지역마다 다르게 관리한다.
   ③ 미국에서만 사용되는 특수 문자 처리를 위한 방식이며 멀티바이트 언어 표현에는 적합하지 않다.
   ④ 아스키 코드의 한계를 극복하기 위해 16비트 이상을 사용해 전 세계 대부분의 문자를 통합적으로 표현하는 유리한 통합 문자 체계이다.

정답 4번

출제 포인트

유니코드가 국가·언어별 문자 체계의 차이를 넘어 전 세계 문자를 일관되게 표현하기 위한 통합 문자 코드 표준임을 묻는 문항이다.

핵심 개념

유니코드는 플랫폼·프로그램·언어와 관계없이 각 문자에 고유한 코드 포인트를 부여하는 국제 문자 코드 표준이다. 실제 저장과 전송에는 UTF-8, UTF-16, UTF-32 등의 인코딩 형식이 사용된다. 따라서 유니코드 자체를 고정된 8비트 또는 16비트 저장 방식 하나로만 이해하면 안 된다.

NoteBase

선지별 해설

① X
틀린 표현은 '저장 공간을 줄이기 위해 8비트 구조로 설계'와 '저장 효율이 가장 큰 방식'이다. 유니코드는 전 세계 문자를 통합적으로 표현하기 위한 표준이며, 고정된 8비트 구조가 아니다. 올바르게는 UTF-8이 8비트 코드 단위를 사용하는 가변 길이 유니코드 인코딩 형식이라고 해야 한다.

② X
틀린 표현은 '국가별로 독립적인 문자 코드체계를 유지'하고 '코드페이지를 지역마다 다르게 관리'한다는 부분이다. 이는 유니코드가 해결하려는 기존 코드페이지 방식의 문제에 해당한다. 올바르게는 국가와 언어가 달라도 같은 문자에는 일관된 코드 포인트를 부여하여 코드 충돌과 호환성 문제를 줄이는다고 해야 한다.

③ X
틀린 표현은 '미국에서만 사용되는 특수 문자 처리'와 '멀티바이트 언어 표현에는 적합하지 않다'는 부분이다. 유니코드는 특정 국가만을 위한 방식이 아니며 한글을 비롯한 여러 언어와 기호를 함께 표현한다. 올바르게는 전 세계의 다양한 문자를 표현하기 위한 국제 표준이라고 해야 한다.

④ O
아스키 코드처럼 제한된 문자 집합만 다루는 방식의 한계를 넘어 전 세계 여러 문자에 고유한 값을 부여하는 통합 문자 체계라는 설명이 핵심적으로 옳다. 다만 현대 유니코드는 고정 16비트 인코딩 하나가 아니며, UTF-8·UTF-16·UTF-32 등 여러 인코딩 형식으로 표현된다.

오답 포인트

①의 '8비트'와 ④의 '16비트 이상'이라는 숫자만 보고 판단하기 쉽다. 핵심은 비트 수 자체가 아니라 유니코드의 목적이 국가별 코드페이지를 따로 유지하는 것이 아니라 전 세계 문자를 하나의 코드 체계로 통합하는 데 있다는 점이다.

다시 나오면 이렇게 판단

유니코드의 설명에서는 '전 세계 문자', '고유한 코드 포인트', '플랫폼과 언어에 독립적인 통합 표준'을 옳은 키워드로 판단한다. '미국 전용', '국가별 독립 코드페이지 유지', '고정 8비트'는 틀린 설명으로 판단한다.

암기 포인트

유니코드는 문자의 통합 번호 체계이고, UTF-8·UTF-16·UTF-32는 그 번호를 저장·전송하는 인코딩 형식이다.

NoteBase

2. 다음 파이썬 프로그램의 실행 결과로 가장 적절한 것은?

```
a = 0
b = 0
c = 0
for a in range(1, 20, 3):
        b += a
        c += b
        if b > 15:
                break
print("Result: ", c)
```

① Result: 40
② Result: 51
③ Result: 70
④ Result: 75

정답 1번

출제 포인트

range의 반복값과 누적 대입 연산을 순서대로 추적하고, break가 실행되는 시점의 출력값을 구하는 문제이다.

조건 정리

range(1, 20, 3)은 1부터 시작하여 3씩 증가하므로 1, 4, 7, 10, 13, 16, 19를 차례로 만든다.
각 반복에서 먼저 b에 a를 누적하고, 이어서 갱신된 b를 c에 누적한다.
$b > 15$인지 확인하는 시점은 $c +=b$가 실행된 뒤이므로, 조건을 처음 만족한 반복의 b도 c에 포함된다.

사용할 공식

각 반복의 갱신 관계는 $b_n = b_{n-1} + a_n$, $c_n = c_{n-1} + b_n$이다.
이 관계를 쓰는 이유는 +=가 오른쪽 값을 기존 변수에 더한 뒤 그 결과를 다시 같은 변수에 저장하는 누적 대입 연산이기 때문이다.

NoteBase

계산 과정

1. $a = 1$: $b = 0 + 1 = 1$, $c = 0 + 1 = 1$이며 b는 15보다 크지 않다.
2. $a = 4$: $b = 1 + 4 = 5$, $c = 1 + 5 = 6$이다.
3. $a = 7$: $b = 5 + 7 = 12$, $c = 6 + 12 = 18$이다.
4. $a = 10$: $b = 12 + 10 = 22$, $c = 18 + 22 = 40$이다. 이때 $22 > 15$이므로 break가 실행된다.

따라서 print문이 출력하는 값은 Result: 40이다.

선지 판단

① O — break 직전까지 c에 누적된 값이 1, 6, 18, 40이므로 Result: 40이 출력된다.
② X — Result: 51은 실제 누적 과정에서 나오지 않는다. 틀린 표현 'Result: 51'은 'Result: 40'으로 고쳐야 한다.
③ X — $a = 10$인 반복에서 $c = 40$이 된 즉시 break가 실행되므로 Result: 70까지 계산되지 않는다. 틀린 표현 'Result: 70'은 'Result: 40'으로 고쳐야 한다.
④ X — Result: 75는 첫 break를 무시하고 다음 값 $a = 13$까지 한 번 더 실행했을 때 $b = 35$, $c = 75$로 계산되는 값이다. 틀린 표현 'Result: 75'는 'Result: 40'으로 고쳐야 한다.

오답 포인트

핵심 함정은 break 조건을 확인하는 위치이다. b가 처음 15를 초과하는 반복에서도 $c + = b$가 먼저 실행되므로 22가 c에 포함된다. 반대로 break를 무시하고 다음 반복까지 진행하면 ④의 75를 잘못 고르게 된다.

다시 나오면 이렇게 판단

반복문의 생성값을 먼저 적고, 들여쓰기된 문장을 위에서 아래로 한 줄씩 실행하여 변수값을 표처럼 누적한다. break는 조건을 만족한 즉시 이후 반복만 중단하므로, break보다 앞선 문장은 해당 반복에서도 실행된 것으로 판단한다.

암기 포인트

range(시작, 끝, 간격)에서 끝값은 포함되지 않으며, break 앞의 문장은 종료되는 반복에서도 이미 실행된다.

NoteBase

3. 개체-관계(Entity-Relationship) 모델에 대한 설명으로 가장 적절한 것은?
① 약한 개체의 참여 제약 조건은 부분 참여이다.
② 개체를 구성하는 속성들 사이의 순서는 의미가 없다.
③ 관계는 자기 자신만의 속성을 가질 수 없다.
④ 개체, 속성, 관계성으로 기술하는 논리적 데이터 모델이다.

정답 2번

출제 포인트

개체-관계 모델의 구성 요소와 약한 개체의 참여 제약 조건을 정확히 구분하는 문제이다.

핵심 개념

개체-관계 모델은 현실 세계의 데이터를 개체, 속성, 관계로 표현하는 개념적 데이터 모델이다. 속성은 개체나 관계의 특성을 나타내며, 속성 사이의 순서는 의미가 없다. 약한 개체는 독립적인 기본키가 없어 소유 개체와의 식별 관계에 의존하며, 그 식별 관계에 전부 참여한다.

선지별 해설

① X - 약한 개체는 소유 개체와의 식별 관계에 반드시 참여해야 하므로 참여 제약 조건은 부분 참여가 아니라 전부 참여이다. 틀린 표현인 '부분 참여'는 '전부 참여'로 고쳐야 한다.
② O - 개체를 구성하는 속성은 순서가 있는 목록이 아니라 특성을 나타내는 집합이므로 속성 사이의 배열 순서에는 의미가 없다.
③ X - 관계도 관계 자체의 특성을 설명하는 속성을 가질 수 있다. 틀린 표현인 '가질 수 없다'는 '가질 수 있다'로 고쳐야 한다.
④ X - 개체-관계 모델은 개체, 속성, 관계성으로 현실 세계를 표현하지만 데이터 모델의 단계로는 논리적 데이터 모델이 아니라 개념적 데이터 모델이다. 틀린 표현인 '논리적 데이터 모델'은 '개념적 데이터 모델'로 고쳐야 한다.

오답 포인트

①의 '부분 참여'와 '전부 참여'를 뒤바꾸거나, ④에서 구성 요소에 관한 설명만 보고 '논리적'이라는 잘못된 분류를 놓치기 쉽다. 또한 속성은 개체에만 붙는다고 오해하면 ③도 잘못 판단하게 된다.

NoteBase

다시 나오면 이렇게 판단

ER 모델 문항에서는 구성 요소와 모델 단계를 나누어 판단한다. 개체와 관계는 속성을 가질 수 있고 속성의 순서는 무의미하며, 약한 개체는 식별 관계에 전부 참여하고, ER 모델은 개념적 데이터 모델이라는 기준을 적용한다.

암기 포인트

ER은 개념적 모델, 속성은 무순서, 관계도 속성 보유 가능, 약한 개체는 식별 관계에 전부 참여이다.

NoteBase

4. 프로토콜 구조가 다른 네트워크 환경들을 연결할 수 있는 기능을 제공하고 여러 계층의 프로토콜 변환 기능을 수행하는 장비로 가장 적절한 것은?
① 게이트웨이(Gateway)
② 리피터(Repeater)
③ 브리지(Bridge)
④ 라우터(Router)

정답 1번

출제 포인트

프로토콜 구조가 서로 다른 네트워크를 연결하면서 여러 계층의 프로토콜을 변환하는 장비를 구별하는 문제이다.

핵심 개념

게이트웨이는 서로 다른 프로토콜이나 구조를 사용하는 네트워크 사이에서 통신 형식과 프로토콜을 변환하여 연결하는 장비이다.
리피터는 물리 계층에서 약해진 신호를 재생·증폭하고, 브리지는 데이터 링크 계층에서 MAC 주소를 기준으로 LAN 구간을 연결하며, 라우터는 네트워크 계층에서 IP 주소를 기준으로 패킷의 경로를 선택한다.
결정적인 구분 기준은 '서로 다른 프로토콜의 변환' 여부이다.

선지별 해설

① O — 게이트웨이는 프로토콜 구조가 다른 네트워크를 연결하고 여러 계층에서 필요한 프로토콜 변환을 수행할 수 있으므로 문제의 설명에 해당한다.
② X — 리피터는 프로토콜을 변환하는 장비가 아니라 물리 계층에서 신호를 재생·증폭하여 전송 거리를 늘리는 장비이다. 틀린 표현 '여러 계층의 프로토콜 변환'은 '물리 계층의 신호 재생·증폭'으로 고쳐야 한다.
③ X — 브리지는 서로 다른 프로토콜을 변환하는 장비가 아니라 데이터 링크 계층에서 MAC 주소를 바탕으로 LAN 구간을 연결하고 프레임을 전달·필터링하는 장비이다. 틀린 표현 '여러 계층의 프로토콜 변환'은 '데이터 링크 계층의 프레임 전달·필터링'으로 고쳐야 한다.
④ X — 라우터는 여러 계층의 프로토콜을 변환하는 장비가 아니라 네트워크 계층에서 IP 주소와 라우팅 정보를 바탕으로 패킷의 전달 경로를 선택하는 장비이다. 틀린 표현 '여러 계층의 프로토콜 변환'은 '네트워크 계층의 패킷 경로 선택'으로 고쳐야 한다.

NoteBase

오답 포인트

네 장비가 모두 통신 범위를 이어 주는 것처럼 보이지만 핵심 동작이 다르다. 특히 ④ 라우터의 '서로 다른 네트워크 연결' 기능만 보고 고르기 쉽지만, 문제의 결정적 표현은 '여러 계층의 프로토콜 변환'이며 이는 게이트웨이에 해당한다.

다시 나오면 이렇게 판단

장비를 기능과 계층으로 구분한다. 신호 재생은 리피터, MAC 주소에 따른 LAN 구간 연결은 브리지, IP 경로 선택은 라우터, 서로 다른 프로토콜의 변환은 게이트웨이로 판단한다.

암기 포인트

리피터는 신호, 브리지는 MAC, 라우터는 경로, 게이트웨이는 프로토콜 변환이다.

NoteBase

5. 시스템 소프트웨어로 가장 적절하지 않은 것은?
① 운영체제(Operating System)
② 워드 프로세서(Word Processor)
③ 부트 로더(Boot Loader)
④ 장치 드라이버(Device Driver)

정답 2번

출제 포인트

시스템 소프트웨어와 응용 소프트웨어를 기능에 따라 구분하는 문제이다.

핵심 개념

시스템 소프트웨어: 컴퓨터 하드웨어를 관리하고 응용 소프트웨어가 실행될 환경을 제공하는 소프트웨어이다. 운영체제, 부트 로더, 장치 드라이버가 이에 해당한다.
응용 소프트웨어: 사용자의 특정 업무를 처리하기 위한 소프트웨어이다. 문서 작성을 위한 워드 프로세서가 이에 해당한다.
구분 기준은 컴퓨터 시스템의 운영·제어를 담당하는지, 사용자의 구체적인 작업을 수행하는지이다.

선지별 해설

① O — 운영체제(Operating System)는 하드웨어 자원을 관리하고 프로그램의 실행 환경을 제공하므로 시스템 소프트웨어이다.
② X — 워드 프로세서(Word Processor)는 문서 작성이라는 사용자의 특정 업무를 수행하므로 시스템 소프트웨어가 아니다. 틀린 분류인 '시스템 소프트웨어'는 '응용 소프트웨어'로 고쳐야 한다.
③ O — 부트 로더(Boot Loader)는 컴퓨터가 시작될 때 운영체제를 메모리에 적재해 실행을 준비하는 시스템 소프트웨어이다.
④ O — 장치 드라이버(Device Driver)는 운영체제가 프린터나 그래픽 카드 같은 하드웨어 장치를 제어하고 통신하도록 돕는 시스템 소프트웨어이다.

오답 포인트

②의 워드 프로세서도 컴퓨터에서 실행되는 소프트웨어이지만, 컴퓨터 시스템 자체를 운영하는 것이 아니라 사용자의 문서 작성 업무를 처리한다. 또한 문제의 '적절하지 않은'이라는 부정형을 놓치면 시스템 소프트웨어인 ①, ③, ④를 고르기 쉽다.

NoteBase

다시 나오면 이렇게 판단

하드웨어 관리, 부팅, 장치 제어, 실행 환경 제공이 핵심이면 시스템 소프트웨어로 판단하고, 문서 작성처럼 사용자의 특정 작업 수행이 핵심이면 응용 소프트웨어로 판단한다.

암기 포인트

운영체제·부트 로더·장치 드라이버는 시스템 운영, 워드 프로세서는 사용자 문서 작업이다.

NoteBase

6. 스택의 응용 분야로 가장 적절하지 않은 것은?
① 작업 대기열 관리
② 재귀함수 호출
③ 편집 실행 취소(undo)
④ 후위 표기식 계산

정답 1번

출제 포인트

스택의 후입선출 특성과 큐의 선입선출 특성을 구별하고, 각 자료구조의 대표적인 응용 분야를 판단하는 문제이다.

핵심 개념

스택은 나중에 들어온 자료를 먼저 꺼내는 후입선출(LIFO) 자료구조이다. 재귀함수 호출, 실행 취소, 후위 표기식 계산처럼 가장 최근에 저장된 항목을 먼저 처리할 때 사용한다.
큐는 먼저 들어온 자료를 먼저 꺼내는 선입선출(FIFO) 자료구조이다. 작업 대기열처럼 도착한 순서대로 처리할 때 사용한다.

선지별 해설

① X — '작업 대기열 관리'는 스택의 응용 분야라는 점이 틀리다. 작업은 일반적으로 먼저 들어온 순서대로 처리하므로, 올바르게는 큐의 대표적인 응용 분야이다.
② O — 재귀함수를 호출할 때 복귀 주소와 지역 변수 등의 호출 정보가 스택에 쌓이고, 가장 나중에 호출된 함수부터 반환된다.
③ O — 편집 실행 취소(undo)는 가장 최근에 수행한 작업부터 역순으로 되돌리므로 스택을 이용한다.
④ O — 후위 표기식 계산은 피연산자를 스택에 넣고 연산자를 만날 때 필요한 피연산자를 꺼내 계산하므로 스택을 이용한다.

오답 포인트

①의 '대기열'도 자료를 임시로 저장한다는 점 때문에 스택과 혼동하기 쉽다. 그러나 작업의 도착 순서를 유지해야 하므로 후입선출이 아니라 선입선출인 큐가 적합하다.

다시 나오면 이렇게 판단

가장 최근 항목을 먼저 처리하면 스택, 가장 먼저 도착한 항목을 먼저 처리하면 큐로 판단한다. 재귀 호출·실행 취소·후위 표기식은 스택, 작업 대기열은 큐에 대응한다.

NoteBase

암기 포인트

스택은 최근 것부터(LIFO), 큐는 먼저 온 것부터(FIFO) 처리한다.

NoteBase

7. 운영체제의 역할에 대한 설명으로 가장 적절하지 않은 것은?
① CPU, 메모리, 입출력 장치 등 하드웨어 자원을 효율적으로 할당하고 배분한다.
② 사용자에게 컴퓨터 프로그램을 효율적으로 실행할 수 있는 환경을 제공한다.
③ 사용자가 작성한 원시 프로그램을 번역하여 시스템에서 실행할 수 있게 한다.
④ 시스템 오류를 감지하고 처리하여 시스템이 정상적으로 작동하도록 유지한다.

정답 3번

출제 포인트

운영체제가 담당하는 자원 관리·실행 환경 제공·오류 처리 기능과 컴파일러가 담당하는 원시 프로그램 번역 기능을 구별하는 문제이다.

핵심 개념

운영체제: 컴퓨터 하드웨어 자원을 관리하고, 사용자와 응용 프로그램이 컴퓨터를 사용할 수 있는 실행 환경을 제공하는 시스템 소프트웨어이다.
컴파일러: 사용자가 작성한 원시 프로그램을 컴퓨터가 처리할 수 있는 목적 프로그램으로 번역하는 언어 번역 프로그램이다.
결정 차이축은 '자원과 시스템 운영을 관리하는가'와 '프로그래밍 언어로 작성된 코드를 번역하는가'이다.

선지별 해설

① O — 운영체제는 CPU의 처리 시간, 메모리 공간, 입출력 장치 등의 하드웨어 자원을 효율적으로 할당하고 관리한다.
② O — 운영체제는 프로그램의 실행을 관리하고 사용자에게 컴퓨터를 편리하고 효율적으로 이용할 수 있는 환경을 제공한다.
③ X — 틀린 부분은 '운영체제가 원시 프로그램을 번역한다'는 내용이다. 원시 프로그램을 목적 프로그램 등으로 번역하는 것은 컴파일러·인터프리터·어셈블러와 같은 언어 번역 프로그램의 역할이다. 올바르게는 '언어 번역 프로그램이 사용자가 작성한 원시 프로그램을 번역하여 시스템에서 실행할 수 있게 한다'이다.
④ O — 운영체제는 하드웨어와 소프트웨어의 오류를 감지하고 적절히 처리하거나 복구하여 시스템이 안정적으로 작동하도록 관리한다.

NoteBase

오답 포인트

③의 '시스템에서 실행할 수 있게 한다'는 결과만 보면 운영체제의 프로그램 실행 관리 기능처럼 보일 수 있다.
그러나 핵심 동작인 '원시 프로그램을 번역'하는 주체는 운영체제가 아니라 언어 번역 프로그램이다.

다시 나오면 이렇게 판단

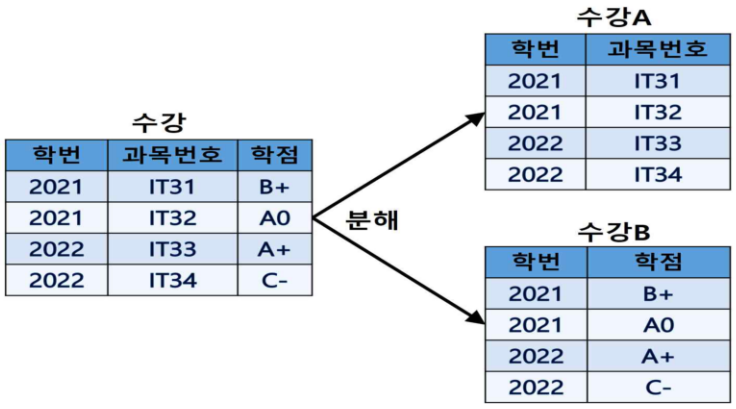
CPU·메모리·입출력 장치의 자원 관리, 프로그램 실행 환경 제공, 오류 처리는 운영체제의 기능으로 판단한다.
원시 프로그램의 번역이 나오면 컴파일러·인터프리터·어셈블러 같은 언어 번역 프로그램의 기능으로 판단한다.

암기 포인트

운영체제는 자원과 실행을 관리하고, 컴파일러는 원시 프로그램을 번역한다.

NoteBase

8. 다음 그림은 수강 릴레이션을 수강A와 수강B 릴레이션으로 분해한 결과이다. 릴레이션 수강A, 수강B가 다시 자연 조인되었을 때 가지고 올 수 있는 현상으로 가장 적절한 것은?



① 수정 이상(Modification Anomaly)
② 가짜 튜플(Spurious Tuple)
③ 삽입 이상(Insertion Anomaly)
④ 정보 중복(Information Redundancy)

정답 2번

출제 포인트

릴레이션을 분해한 뒤 자연 조인할 때 원래 릴레이션에 없던 튜플이 생성되는 현상을 구별하는 문제이다.

핵심 개념

가짜 튜플은 분해된 릴레이션을 다시 조인할 때 원래 릴레이션에 없던 속성값의 조합이 새로 생긴 튜플이다. 이 그림에서는 학번만으로 자연 조인하므로, 같은 학번에 속한 모든 과목번호와 모든 학점이 서로 조합된다. 예를 들어 학번 2021에서는 원래 없던 (2021, IT31, A0)과 (2021, IT32, B+)가 생성된다. 따라서 이 분해는 원래 릴레이션을 정확히 복원하지 못하는 손실 분해이다.

NoteBase

선지별 해설

① X — '수정 이상'이라는 명칭이 틀리다. 수정 이상은 중복 저장된 같은 정보를 일부만 변경하여 값이 서로 불일치하는 현상이다. 이 그림에서 자연 조인으로 원래 없던 행이 생기는 현상은 수정 이상이 아니라 가짜 튜플이다.
② O — 수강A와 수강B를 학번으로 자연 조인하면 같은 학번의 과목번호와 학점이 모두 교차 결합되어 원래 없던 튜플이 생성된다. 이는 가짜 튜플이다.
③ X — '삽입 이상'이라는 명칭이 틀리다. 삽입 이상은 필요한 정보를 넣으려 할 때 관련 없는 다른 정보가 없어 삽입할 수 없거나 불필요한 값까지 함께 넣어야 하는 현상이다. 이 그림의 현상은 삽입 이상이 아니라 가짜 튜플이다.
④ X — '정보 중복'이라는 명칭이 틀리다. 정보 중복은 동일한 정보가 여러 튜플에 반복 저장되는 상태이다. 이 그림에서 문제 되는 것은 반복 저장 자체가 아니라 자연 조인으로 원래 없던 조합이 만들어지는 가짜 튜플이다.

오답 포인트

분해가 데이터 중복을 줄이기 위한 정규화 과정이라는 점만 보고 수정 이상·삽입 이상·정보 중복을 고르기 쉽다. 그러나 문제는 분해의 목적이 아니라 '다시 자연 조인했을 때'의 결과를 묻는다. 원래 없던 과목번호와 학점의 조합이 생긴다는 표현이 ②를 가리킨다.

다시 나오면 이렇게 판단

분해된 릴레이션을 공통 속성으로 조인한 결과를 원래 릴레이션과 비교한다. 조인 결과에 원래 없던 속성값 조합이 추가되면 가짜 튜플이며 손실 분해이고, 원래 튜플만 정확히 복원되면 무손실 분해이다.

암기 포인트

분해 후 조인에서 없던 행이 생기면 가짜 튜플, 원본만 복원되면 무손실 조인이다.

NoteBase

9. 교착 상태가 발생하기 위한 네 가지 필요조건을 나열한 것으로 가장 적절한 것은?
① 상호 배제, 비선점, 우선순위 무시, 순환 요청
② 상호 배제, 점유와 대기, 비선점, 환형 대기
③ 점유와 대기, 자원 분할, 우선순위 할당, 순환 대기
④ 공유 자원, 동시 접근, 비선점, 병렬 대기

정답 2번

출제 포인트

교착 상태가 발생하기 위한 네 가지 필요조건을 정확히 구별하는 문제이다.

핵심 개념

교착 상태는 여러 프로세스가 서로 상대방이 가진 자원을 기다리면서 더 이상 진행하지 못하는 상태이다.
상호 배제: 한 자원은 한 번에 하나의 프로세스만 사용할 수 있다.
점유와 대기: 프로세스가 이미 자원을 가진 채 다른 자원을 기다린다.
비선점: 다른 프로세스가 점유한 자원을 강제로 빼앗을 수 없다.
환형 대기(순환 대기): 프로세스들이 원형으로 연결되어 다음 프로세스가 가진 자원을 기다린다.
이 네 조건이 모두 성립해야 교착 상태가 발생할 수 있으며, 하나라도 제거하면 교착 상태를 예방할 수 있다.

NoteBase

선지별 해설

① X
상호 배제와 비선점은 필요조건이지만, ‘우선순위 무시’는 교착 상태의 필요조건이 아니다. 또한 표준적인 조건명은 ‘순환 요청’이 아니라 ‘환형 대기(순환 대기)’이다. 올바르게는 상호 배제, 점유와 대기, 비선점, 환형 대기이다.

② O
상호 배제, 점유와 대기, 비선점, 환형 대기는 교착 상태 발생의 네 가지 필요조건을 모두 정확히 나열한 것이다.

③ X
‘자원 분할’과 ‘우선순위 할당’은 교착 상태의 필요조건이 아니며, 상호 배제와 비선점이 빠져 있다. 올바르게는 점유와 대기에 상호 배제, 비선점, 환형 대기를 더해야 한다.

④ X
‘공유 자원’, ‘동시 접근’, ‘병렬 대기’는 네 필요조건의 명칭이 아니다. 특히 동시 접근은 한 번에 하나의 프로세스만 자원을 사용한다는 상호 배제와 반대되는 표현이다. 올바르게는 상호 배제, 점유와 대기, 비선점, 환형 대기이다.

오답 포인트

①의 ‘순환 요청’처럼 뜻이 비슷해 보이는 표현이나 ④의 ‘공유 자원’처럼 교착 상태와 관련 있어 보이는 일반 용어를 공식 필요조건으로 착각하기 쉽다. 네 조건의 표준 명칭이 모두 들어 있는지를 확인해야 한다.

다시 나오면 이렇게 판단

선지에서 자원의 독점 사용, 자원을 가진 채 추가 대기, 강제 회수 불가, 원형 대기 관계가 각각 나타나는지 확인한다. 이 네 의미와 일치하지 않는 용어가 하나라도 포함되면 오답으로 판단한다.

암기 포인트

교착 상태의 네 조건은 ‘상점비환’, 즉 상호 배제·점유와 대기·비선점·환형 대기이다.

NoteBase

10. 음성 인식 기술의 화자 종속과 화자 독립 개념에 대한 설명으로 가장 적절한 것은?

① 화자 독립 음성 인식 시스템은 하나의 사용자 음성에 최적화되어 정확도는 높지만 다른 사용자의 음성은 일반적으로 인식하지 못한다.

② 화자 종속 시스템은 다양한 사용자에 대해 일반화되어 있어 학습 없이 누구의 음성이든 잘 인식할 수 있도록 설계된다.

③ 최신 음성 인식 기술은 대부분 화자 종속 방식만을 채택하며 사용자별 음성 프로파일 구축이 필수적이다.

④ 화자 독립 시스템은 다수 사용자에게 적용 가능하도록 훈련된 일반화 모델을 사용하고 화자 종속 시스템은 특정 사용자에 대해 별도로 학습된 모델을 사용하는 방식이다.

정답 4번

출제 포인트

화자 독립 음성 인식과 화자 종속 음성 인식을 사용자별 학습 여부와 여러 사용자에 대한 일반화 가능성으로 구별하는 문제이다.

핵심 개념

화자 독립 방식은 여러 화자의 음성 자료로 학습하여 특정 사용자에게 별도로 맞추지 않아도 다양한 사용자의 음성을 인식하도록 설계한 방식이다. 키워드는 다수 사용자와 일반화이다.
화자 종속 방식은 특정 사용자의 음성 특성에 맞추어 학습하거나 적응하여 그 사용자의 음성을 더 잘 인식하도록 설계한 방식이다. 키워드는 특정 사용자와 사용자별 음성 프로파일이다.

NoteBase

선지별 해설

① X — '화자 독립'이라는 표현이 틀렸다. 하나의 사용자 음성에 최적화되고 다른 사용자의 음성을 일반적으로 잘 인식하지 못하는 것은 화자 종속 시스템의 특징이다. 올바르게는 '화자 종속 음성 인식 시스템은 특정 사용자의 음성에 최적화된다'이다.
② X — '화자 종속'과 '학습 없이 누구의 음성이든'이라는 표현이 틀렸다. 다양한 사용자의 음성에 일반화되어 사용자별 사전 학습 없이 사용할 수 있도록 설계되는 것은 화자 독립 시스템이다. 올바르게는 '화자 독립 시스템은 다양한 사용자의 음성을 인식하도록 일반화하여 설계된다'이다.
③ X — '대부분 화자 종속 방식만을 채택'과 '사용자별 음성 프로파일 구축이 필수'라는 표현이 틀렸다. 최신 음성 인식 기술은 다수 화자 데이터로 학습한 화자 독립 방식을 널리 사용하며, 모든 시스템에서 사용자별 음성 프로파일이 필수인 것은 아니다. 올바르게는 '최신 음성 인식 기술은 화자 독립 방식도 널리 사용하며 사용자별 음성 프로파일이 항상 필수는 아니다'이다.
④ O — 화자 독립 시스템은 다수 사용자에게 적용할 수 있도록 일반화한 모델을 사용하고, 화자 종속 시스템은 특정 사용자의 음성에 맞추어 별도로 학습하거나 적응한 모델을 사용한다. 두 방식의 차이를 정확히 설명한다.

오답 포인트

①과 ②는 화자 독립과 화자 종속의 특징을 서로 바꾸어 제시한다. ③은 개인화가 정확도를 높일 수 있다는 사실을 모든 최신 시스템의 필수 조건인 것처럼 과장한다.

다시 나오면 이렇게 판단

특정 한 사람에게 맞춘 모델이면 화자 종속, 여러 사람이 별도 등록 없이 사용할 수 있도록 일반화한 모델이면 화자 독립으로 판단한다.

암기 포인트

독립은 여러 화자에 일반화, 종속은 특정 화자에 개인화이다.

NoteBase

11. 방화벽의 기능으로 가장 적절한 것은?
① 새로운 형태의 보안 공격 차단에 능동적이다.
② 내부자에 의한 공격을 차단하는 데 용이하다.
③ 로그 자료를 바탕으로 감사 기능을 제공한다.
④ 문서에 포함된 악성 코드 탐지가 용이하다.

정답 3번

출제 포인트

방화벽의 기본 기능인 접근 통제와 로그 기록을, 방화벽만으로 대응하기 어려운 내부자 공격·신종 공격·문서 악성 코드 탐지와 구별하는 문제이다.

핵심 개념

방화벽은 내부망과 외부망의 경계에서 미리 정한 보안 규칙에 따라 네트워크 통신을 허용하거나 차단하는 보안 시스템이다. 통신의 허용·차단 내역과 접속 정보를 로그로 남길 수 있으므로 사고 분석과 보안 감사에 활용된다. 반면 경계를 통과하지 않는 내부자 행위, 기존 규칙으로 식별하기 어려운 새로운 공격, 문서 파일 내부의 악성 코드 검사는 방화벽의 기본 기능만으로 대응하기 어렵다.

선지별 해설

① X — 틀린 표현은 '새로운 형태의 보안 공격 차단에 능동적이다'이다. 방화벽은 미리 설정된 규칙을 중심으로 통신을 통제하므로 알려지지 않은 새로운 공격에 항상 능동적으로 대응하는 장치라고 볼 수 없다. 올바르게는 '설정된 보안 정책에 따른 네트워크 접근 통제에 효과적이지만 새로운 공격에는 한계가 있다'이다.
② X — 틀린 표현은 '내부자에 의한 공격을 차단하는 데 용이하다'이다. 내부망 안에서 발생하여 방화벽을 지나지 않는 행위는 경계형 방화벽이 탐지하거나 차단하기 어렵다. 올바르게는 '외부와 내부 사이의 비인가 통신을 통제하는 데 주로 사용되며 내부자 공격 차단에는 한계가 있다'이다.
③ O — 방화벽은 접속 시도, 허용·차단 결과 등 통신 내역을 로그로 기록할 수 있다. 이 로그는 보안 정책 준수 여부 확인, 침해 사고 추적, 보안 감사의 근거 자료가 되므로 방화벽의 기능으로 가장 적절하다.
④ X — 틀린 표현은 '문서에 포함된 악성 코드 탐지가 용이하다'이다. 문서 내부 악성 코드의 분석·탐지는 기본적으로 백신이나 악성 코드 탐지·콘텐츠 보안 도구의 역할이다. 올바르게는 '방화벽은 네트워크 통신을 통제하며, 문서 내부 악성 코드 탐지는 전용 보안 도구가 담당한다'이다.

오답 포인트

①과 ④는 고도화된 통합 보안 장비의 부가 기능을 방화벽의 일반적인 기본 기능처럼 표현한 함정이다. ②는 방화벽이 주로 내부망과 외부망의 경계에서 작동한다는 점을 놓치게 한다.

NoteBase

다시 나오면 이렇게 판단

선지가 '정책에 따른 통신 허용·차단'이나 '접속 로그 기록·감사'를 말하면 방화벽의 기본 기능으로 판단한다. '내부망 안의 행위', '알려지지 않은 신종 공격의 능동적 차단', '파일 내부 악성 코드 분석'은 방화벽 단독의 기본 기능과 구별한다.

암기 포인트

방화벽은 경계에서 통신을 통제하고 로그를 남긴다. 내부자·신종 공격·파일 악성 코드에는 한계가 있다.

NoteBase

12. 분산 데이터베이스 시스템의 특징으로 가장 적절한 것은?
① 데이터베이스 설계가 간단하며 관리의 편의성이 우수하다.
② 중앙 집중 시스템에 비해 신뢰성과 가용성이 우수하다.
③ 사용자는 자료가 어느 사이트에 저장되어 있는지 알아야 한다.
④ 중복 데이터의 일치성 보장을 위해 추가적인 작업이 필요 없다.

정답 2번

출제 포인트

분산 데이터베이스 시스템의 장점과 단점을 중앙 집중형 시스템과 구별할 수 있는지를 묻는 문제이다.

핵심 개념

분산 데이터베이스는 논리적으로 하나인 데이터베이스를 여러 지역의 컴퓨터에 나누어 저장하고 네트워크로 통합 관리하는 시스템이다. 여러 사이트가 데이터를 분담하거나 복제하므로 장애에 강하고 가용성이 높지만, 분산된 자료와 중복 데이터의 일관성을 유지해야 하므로 설계와 관리가 복잡하다. 또한 분산 투명성을 통해 사용자가 데이터의 실제 저장 위치를 의식하지 않고 이용할 수 있도록 한다.

선지별 해설

① X — 틀린 표현은 '데이터베이스 설계가 간단하며 관리의 편의성이 우수하다'이다. 분산 데이터베이스는 여러 사이트 사이의 통신, 분산 트랜잭션, 장애 처리와 일관성 제어가 필요하므로 설계와 관리가 복잡하다.
② O — 데이터와 처리를 여러 사이트에 분산하면 한 사이트에 장애가 발생해도 다른 사이트가 서비스를 계속할 수 있으므로 중앙 집중형 시스템보다 신뢰성과 가용성을 높일 수 있다.
③ X — 틀린 표현은 '사용자는 자료가 어느 사이트에 저장되어 있는지 알아야 한다'이다. 분산 투명성이 제공되므로 사용자는 자료의 실제 저장 위치를 알지 못해도 하나의 데이터베이스처럼 이용할 수 있다.
④ X — 틀린 표현은 '추가적인 작업이 필요 없다'이다. 여러 사이트에 중복 저장된 데이터의 일관성을 보장하려면 복제 제어, 동기화, 분산 동시성 제어 등의 추가 작업이 필요하다.

NoteBase

오답 포인트

분산이라는 말만 보고 관리도 단순할 것이라고 판단하거나, 물리적으로 여러 곳에 저장된다는 사실 때문에 사용자가 저장 위치를 알아야 한다고 생각하기 쉽다. 분산 데이터베이스의 핵심은 장애 대응력과 가용성은 높아지는 반면, 내부 설계·관리와 일관성 제어는 더 복잡해진다는 점이다.

다시 나오면 이렇게 판단

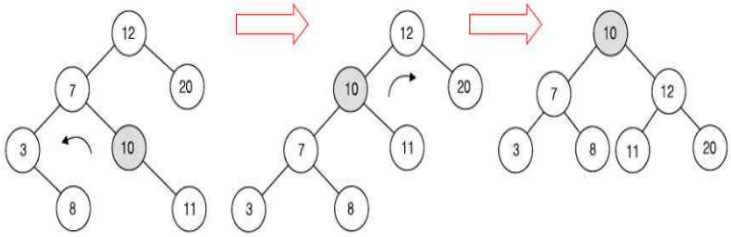
분산 데이터베이스의 선지는 '장애 분산·가용성 향상'이면 장점으로, '설계·관리·일관성 제어의 복잡성'이면 단점으로 판단한다. 사용자 관점에서는 저장 위치를 숨기는 분산 투명성이 제공되는지도 확인한다.

암기 포인트

분산 DB는 장애에는 강하고 위치는 숨기지만, 설계와 일관성 관리는 복잡하다.

NoteBase

13. 다음 이진 탐색 트리에서 특정 노드(10)를 루트로 끌어올리는 회전 과정에 대한 설명으로 가장 적절하지 않은 것은?



① 이 회전은 Splay Tree에서의 Zig-Zag 회전을 보여준다.
② 이 회전의 구조는 AVL 트리의 LR 회전 방식과 유사하다.
③ 이 회전은 AVL 트리에서의 균형 유지를 위해 수행되는 LR 회전과 그 목적과 조건이 동일하다.
④ 이 방식은 트리의 균형 유지보다 최근 접근된 노드의 접근성을 높이는 데 목적이 있다.

정답 3번

출제 포인트

이 문제는 접근 노드 10이 부모 7의 오른쪽 자식이고, 부모 7은 조부모 12의 왼쪽 자식인 도식에서 Splay Tree의 Zig-Zag 회전을 식별하고, 이를 AVL 트리의 LR 회전과 목적·수행 조건 면에서 구별하는 문제이다.

선지별 해설

① O — 노드 10과 부모 7이 서로 반대 방향으로 연결된 Zig-Zag 형태이므로, 먼저 7을 기준으로 10을 올리고 이어서 12를 기준으로 10을 올리는 Splay Tree의 Zig-Zag 회전이다.
② O — 왼쪽 자식의 오른쪽 부분에서 꺾이는 구조를 두 번의 단일 회전으로 바로잡는 모양은 AVL 트리의 LR 회전 방식과 유사하다.
③ X — '목적과 조건이 동일하다'는 표현이 틀리다. 올바르게는 회전의 구조는 유사하지만, AVL 트리의 LR 회전은 균형 인수에 따른 불균형을 해소하는 것이 목적이고 Splay Tree의 Zig-Zag 회전은 접근한 노드를 루트 쪽으로 올리는 것이 목적이므로 수행 목적과 조건은 서로 다르다.
④ O — Splay Tree는 최근 접근한 노드를 루트로 이동시켜 같은 노드나 그 주변 노드가 다시 접근될 때의 접근성을 높이는 데 목적이 있다.

NoteBase

오답 포인트

②의 '회전 방식이 유사하다'와 ③의 '목적과 조건이 동일하다'를 혼동하기 쉽다. 두 회전은 겉모양은 같을 수 있지만, AVL 트리는 높이 균형 회복을 위해 회전하고 Splay Tree는 접근 노드를 루트로 끌어올리기 위해 회전한다.

다시 나오면 이렇게 판단

접근 노드와 부모가 조부모에서 서로 반대 방향으로 꺾이면 Zig-Zag, 같은 방향으로 이어지면 Zig-Zig으로 판단한다. 이어서 회전 모양이 비슷하다는 사실과 자료구조별 회전 목적이 같다는 주장을 분리하여 판단한다.

암기 포인트

Splay의 회전은 최근 접근 노드를 위로, AVL의 회전은 불균형을 해소하는 방향으로 수행된다. 모양은 같아도 목적과 조건은 다르다.

NoteBase

14. 인터럽트에 대한 설명으로 가장 적절한 것은?
① 여러 장치에서 인터럽트가 발생하는 경우 발생 순서에 따라 처리한다.
② 일반적으로 하드웨어 인터럽트보다 소프트웨어 인터럽트의 우선순위가 높다.
③ 일반적으로 내부 인터럽트보다 외부 인터럽트의 우선순위가 높다.
④ Daisy Chain 방식은 우선순위 판별을 위한 소프트웨어 방법이다.

정답 3번

출제 포인트

인터럽트가 여러 개 발생했을 때의 처리 기준과 하드웨어·소프트웨어 인터럽트, 내부·외부 인터럽트 사이의 일반적인 우선순위, 그리고 Daisy Chain의 구현 방식을 묻는 문제이다.

핵심 개념

인터럽트는 CPU가 현재 작업을 잠시 중단하고 긴급한 사건을 먼저 처리하도록 하는 기능이다. 여러 인터럽트가 겹치면 우선순위가 높은 요청부터 처리하며, 일반적인 우선순위는 하드웨어 인터럽트가 소프트웨어 인터럽트보다 높고 내부 인터럽트가 외부 인터럽트보다 높다. Daisy Chain은 장치들을 직렬로 연결하여 앞쪽 장치부터 인터럽트 승인 신호를 전달하는 하드웨어 우선순위 판별 방식이다.

선지별 해설

① O — 여러 장치에서 인터럽트가 발생하면 정해진 우선순위에 따라 처리하고, 우선순위가 같은 대기 요청은 일반적으로 발생 순서에 따라 처리할 수 있다.

② O — 하드웨어 고장이나 입출력 완료처럼 즉시 대응해야 하는 하드웨어 인터럽트는 프로그램 명령에 의해 발생하는 소프트웨어 인터럽트보다 일반적으로 우선순위가 높다.

③ X — 틀린 표현은 '외부 인터럽트의 우선순위가 높다'이다. 외부 인터럽트보다 CPU 내부의 예외 상황에서 발생하는 내부 인터럽트의 우선순위가 일반적으로 더 높다고 본다. 올바르게는 '일반적으로 외부 인터럽트보다 내부 인터럽트의 우선순위가 높다'이다.

④ O — Daisy Chain 방식은 인터럽트 승인 신호가 직렬로 연결된 장치들을 차례로 통과하게 하여 물리적 연결 순서에 따라 우선순위를 정하는 하드웨어 방식이다.

NoteBase

오답 포인트

③의 '내부'와 '외부'를 뒤바꾼 표현이 핵심 함정이다. 또한 Daisy Chain을 프로그램이 우선순위를 정하는 소프트웨어 방식으로 오해하기 쉽지만, 장치의 직렬 연결과 승인 신호 전달을 이용하는 하드웨어 방식이다.

다시 나오면 이렇게 판단

인터럽트 우선순위 문제에서는 먼저 하드웨어와 소프트웨어를 구분하고, 다음으로 내부와 외부를 구분한다. 일반적인 비교는 '하드웨어 > 소프트웨어', '내부 > 외부'이며, Daisy Chain은 직렬 연결을 이용한 하드웨어 방식으로 판단한다.

암기 포인트

인터럽트 우선순위는 하드웨어가 소프트웨어보다 높고, 내부가 외부보다 높다. Daisy Chain은 직렬 연결형 하드웨어 우선순위 방식이다.

NoteBase

15. 부울식 $xy + xy' + x'z + x'z'$를 최소화한 결과로 가장 적절한 것은?

① x ② y

③ 0 ④ 1

정답 4번

출제 포인트

이 문제는 부울대수의 분배 법칙과 상보 법칙을 이용하여 논리식을 최소화하는 문제이다.

조건 정리

주어진 식: $xy + xy' + x'z + x'z'$

구해야 할 값: 주어진 부울식을 최소화한 결과

주의할 조건: $x + x' = 1, y + y' = 1, z + z' = 1$인 상보 법칙을 적용한다.

사용할 공식

분배 법칙: $ab + ac = a(b + c)$

상보 법칙: $a + a' = 1$

항등 법칙: $a \cdot 1 = a$

같은 인수를 가진 항끼리 묶으면 상보 관계인 두 변수가 나타나므로 이 법칙들을 사용한다.

계산 과정

1. 앞의 두 항과 뒤의 두 항을 각각 묶는다.

$xy + xy' + x'z + x'z' = x(y + y') + x'(z + z')$

2. 상보 법칙을 적용한다.

$x(y + y') + x'(z + z') = x \cdot 1 + x' \cdot 1 = x + x'$

3. 다시 상보 법칙을 적용한다.

$x + x' = 1$

따라서 최소화한 결과는 1이다.

NoteBase

선지 판단

① X — 첫 두 항만 묶으면 $x(y + y') = x$가 되지만, 뒤의 두 항에서 얻는 x'도 더해야 한다. 틀린 표현 x는 전체 식의 결과가 아니며, 올바르게는 $x + x' = 1$이다.

② X — y는 $y + y' = 1$로 소거되므로 최종 결과에 남지 않는다. 틀린 표현 y 대신 올바른 최소화 결과는 1이다.

③ X — x와 x' 중 하나는 항상 1이므로 $x + x' = 1$이다. 틀린 표현 0 대신 올바른 결과는 1이다.

④ O — $xy + xy' = x$, $x'z + x'z' = x'$이므로 전체 식은 $x + x' = 1$로 최소화된다.

오답 포인트

①의 x는 $xy + xy'$만 계산하고 $x'z + x'z'$에서 나오는 x'를 누락할 때 선택하기 쉽다. 모든 항을 두 묶음으로 정리한 뒤 마지막 상보 관계까지 확인해야 한다.

다시 나오면 이렇게 판단

부울식 최소화 문제에서는 공통 인수를 묶은 뒤 괄호 안에 $a + a' = 1$이 생기는지 확인하고, 남은 항들에도 상보 관계가 있는지 한 번 더 판단한다.

암기 포인트

$ab + ab' = a$, $a + a' = 1$이다. 이 문항은 두 번 묶으면 $x + x' = 1$이 된다.

NoteBase

16. 디스크 스케줄링 기법에 대한 설명으로 가장 적절한 것은?

① FCFS 스케줄링은 탐색 시간을 최소화하기 위해 항상 현재 헤드 위치에서 가장 가까운 요청을 먼저 처리하는 방식으로 응답 시간의 편차를 줄이기 위한 목적에 최적화되어 있다.

② SSTF 스케줄링은 디스크 헤드를 일정 방향으로 왕복시키며 모든 요청을 순차적으로 처리하는 방식이며 헤드가 끝에 도달하면 반대 방향으로 전환한다.

③ SCAN 스케줄링은 응답 시간의 공정성을 보장하지 않으며 서비스 요청의 우선순위와 무관하게 헤드가 도착한 순서에 따라 FIFO로 처리한다.

④ N-단계 SCAN 스케줄링은 요청 큐를 일정 단위로 나누어 처리하므로 과도한 요청이 발생하더라도 각 단계별로 제한된 요청만 처리함으로써 응답 지연을 완화하고 스케줄링의 예측 가능성을 향상시킨다.

정답 4번

출제 포인트

디스크 스케줄링 기법인 FCFS, SSTF, SCAN, N-단계 SCAN의 요청 선택 기준과 동작 방식을 구별하는 문제이다.

핵심 개념

FCFS: 디스크 요청이 도착한 순서대로 처리하는 방식이다. 키워드는 도착 순서이다.
SSTF: 현재 헤드 위치에서 탐색 거리가 가장 짧은 요청을 다음에 처리하는 방식이다. 키워드는 최단 탐색 거리이다.
SCAN: 헤드가 한 방향으로 이동하면서 경로상의 요청을 처리하고, 끝에 도달하면 방향을 바꾸는 방식이다. 키워드는 엘리베이터 방식이다.
N-단계 SCAN: 대기 요청을 최대 N개씩 묶은 부분 큐로 나누고 각 부분 큐에 SCAN을 적용하는 방식이다. 처리 중 새 요청이 현재 부분 큐에 계속 끼어드는 현상을 제한하여 응답 지연의 변동을 줄인다.

NoteBase

선지별 해설

① X
틀린 표현은 'FCFS가 현재 헤드 위치에서 가장 가까운 요청을 먼저 처리한다'는 부분이다. 이는 SSTF의 설명이다. FCFS는 헤드와의 거리에 관계없이 요청이 도착한 순서대로 처리한다.

② X
틀린 표현은 'SSTF가 헤드를 일정 방향으로 이동시키고 끝에서 방향을 전환한다'는 부분이다. 이는 SCAN의 설명이다. SSTF는 매 처리 시점마다 현재 헤드에서 가장 가까운 요청을 선택한다.

③ X
틀린 표현은 'SCAN이 헤드에 도착한 순서에 따라 FIFO로 처리한다'는 부분이다. SCAN은 요청의 도착 순서가 아니라 헤드의 현재 이동 방향과 트랙 위치에 따라 경로상의 요청을 처리한다.

④ O
N-단계 SCAN은 요청 큐를 최대 N개 단위의 부분 큐로 나누어 순차적으로 처리한다. 현재 부분 큐를 처리하는 동안 새 요청이 계속 추가되는 것을 제한하므로 특정 구간에 요청이 몰릴 때의 과도한 지연을 완화하고 응답 특성의 예측 가능성을 높인다.

오답 포인트

①은 SSTF의 '가장 가까운 요청'을 FCFS에 붙였고, ②는 SCAN의 '한 방향 이동 후 반대 방향 전환'을 SSTF에 붙였다. ③은 SCAN도 요청을 차례로 처리한다는 점만 보고 이를 도착 순서 기반의 FIFO로 오해하게 만든다.

다시 나오면 이렇게 판단

선지에서 다음 요청을 정하는 기준을 확인한다. 도착 순서이면 FCFS, 현재 헤드와의 최단 거리이면 SSTF, 일정 방향으로 훑고 되돌아오면 SCAN, 요청을 N개씩 묶어 SCAN을 적용하면 N-단계 SCAN이다.

암기 포인트

FCFS는 먼저 온 순서, SSTF는 가장 가까운 순서, SCAN은 방향대로 왕복, N-단계 SCAN은 N개씩 나누어 왕복이다.

NoteBase

17. 소프트웨어 유지보수의 유형과 그 설명이 가장 적절한 것은?

① 완전형 유지보수 - 소프트웨어의 성능 향상이나 자원 최적화를 위해 기존 기능을 개선하거나 내부 구조를 변경하는 작업

② 수정형 유지보수 - 시스템의 성능 향상을 위해 알고리즘을 최적화하거나 코드 구조를 개선하는 작업

③ 예방형 유지보수 - 하드웨어 교체나 운영체제 변경 등 환경 변화에 적응하기 위한 소프트웨어 수정 작업

④ 적응형 유지보수 - 향후 기능 확장 가능성을 고려하여 소프트웨어 구조를 변경하거나 기능을 미리 추가하는 작업

정답 1번

출제 포인트

소프트웨어 유지보수를 변경 목적에 따라 완전형, 수정형, 적응형, 예방형으로 구분할 수 있는지를 묻는 문제이다.

핵심 개념

완전형 유지보수: 성능 향상, 기능 개선, 자원 최적화 등 소프트웨어의 품질을 더 좋게 만드는 작업이다.
수정형 유지보수: 운용 중 발견된 오류나 결함을 바로잡는 작업이다.
적응형 유지보수: 하드웨어, 운영체제, 업무 환경 등 외부 환경의 변화에 맞추어 소프트웨어를 수정하는 작업이다.
예방형 유지보수: 장래의 장애를 예방하고 유지보수성·신뢰성을 높이기 위해 구조를 미리 개선하는 작업이다.

NoteBase

선지별 해설

① O — 기존 기능을 개선하거나 내부 구조를 변경하여 성능을 높이고 자원을 최적화하는 것은 완전형 유지보수에 해당한다.
② X — 틀린 표현은 '수정형 유지보수'이다. 알고리즘 최적화나 코드 구조 개선으로 성능을 높이는 작업은 완전형 유지보수이다. 수정형 유지보수는 발견된 오류나 결함을 고치는 작업이다.
③ X — 틀린 표현은 '예방형 유지보수'이다. 하드웨어 교체나 운영체제 변경과 같은 환경 변화에 맞추는 작업은 적응형 유지보수이다.
④ X — 틀린 표현은 '적응형 유지보수'이다. 향후 기능 확장과 유지보수를 고려하여 구조를 미리 변경하는 작업은 예방형 유지보수이다. 적응형 유지보수는 하드웨어나 운영체제 등 외부 환경의 변화에 대응하는 작업이다.

오답 포인트

②는 '성능 향상'이라는 목적을 오류 수정과 혼동하게 하고, ③과 ④는 적응형과 예방형의 명칭을 서로 바꾸어 제시한다. 변경 내용보다 변경 목적이 오류 수정인지, 환경 적응인지, 성능 개선인지, 미래 문제 예방인지를 구분해야 한다.

다시 나오면 이렇게 판단

오류·결함을 고치면 수정형, 운영 환경 변화에 맞추면 적응형, 성능·기능을 개선하면 완전형, 미래의 장애와 유지보수 문제를 미리 줄이면 예방형으로 판단한다.

암기 포인트

수정은 오류, 적응은 환경, 완전은 성능·기능, 예방은 미래 대비이다.

NoteBase

18. OSI 7 계층에 대한 설명으로 가장 적절한 것은?
① 네트워크 계층은 MIME 인코딩이나 데이터 암호화의 기능을 지원한다.
② 데이터링크 계층은 사용자 장비와 네트워크 종단 장비 간의 물리적 인터페이스를 규정한다.
③ 전송 계층에서 다루는 데이터 단위를 프레임이라고 한다.
④ 세션 계층은 통신 시스템 간의 상호 대화를 설정하고 유지하고 동기화한다.

정답 4번

출제 포인트

OSI 7계층별 핵심 기능과 각 계층에서 사용하는 데이터 단위를 구분하는 문제이다.

핵심 개념

OSI 7계층은 네트워크 통신 기능을 물리, 데이터링크, 네트워크, 전송, 세션, 표현, 응용 계층으로 나눈 참조 모델이다.
물리 계층은 전기적·기계적 인터페이스와 비트 전송을, 데이터링크 계층은 프레임 단위의 인접 노드 간 전송을, 네트워크 계층은 논리 주소와 경로 설정을 담당한다. 전송 계층은 종단 간 전송과 세그먼트를, 세션 계층은 대화의 설정·유지·동기화를, 표현 계층은 데이터 형식 변환·인코딩·암호화·압축을 담당한다.

NoteBase

선지별 해설

① X
틀린 표현은 '네트워크 계층은 MIME 인코딩이나 데이터 암호화의 기능을 지원한다'이다. MIME 인코딩과 같은 데이터 표현 형식의 변환이나 데이터 암호화는 표현 계층의 기능이다. 네트워크 계층은 논리 주소 지정과 패킷의 경로 설정을 담당한다.

② X
틀린 표현은 '데이터링크 계층은 사용자 장비와 네트워크 종단 장비 간의 물리적 인터페이스를 규정한다'이다. 장비와 전송 매체 사이의 전기적·기계적·기능적 인터페이스는 물리 계층이 규정한다. 데이터링크 계층은 프레임 구성, 물리 주소 지정, 오류 및 흐름 제어 등을 통해 인접 노드 간 전송을 담당한다.

③ X
틀린 표현은 '전송 계층에서 다루는 데이터 단위를 프레임이라고 한다'이다. 프레임은 데이터링크 계층의 데이터 단위이고, 전송 계층의 데이터 단위는 세그먼트라고 한다.

④ O
세션 계층은 통신 시스템 사이의 대화 세션을 설정하고 유지하며 종료하고, 대화의 동기화를 담당하므로 적절한 설명이다.

오답 포인트

계층 이름보다 기능의 키워드를 보아야 한다. ①의 '인코딩·암호화'는 표현 계층, ②의 '물리적 인터페이스'는 물리 계층, ③의 '프레임'은 데이터링크 계층에 해당한다. 서로 이웃하거나 관련된 계층의 기능과 데이터 단위를 바꾸어 놓은 것이 함정이다.

다시 나오면 이렇게 판단

선지의 핵심어를 계층과 연결한다. 물리적 인터페이스·비트는 물리 계층, 프레임은 데이터링크 계층, 경로 설정·패킷은 네트워크 계층, 종단 간 전송·세그먼트는 전송 계층, 대화 설정·동기화는 세션 계층, 인코딩·암호화·압축은 표현 계층으로 판단한다.

암기 포인트

물리-비트, 데이터링크-프레임, 네트워크-패킷, 전송-세그먼트, 세션-대화, 표현-변환·암호화로 연결한다.

NoteBase

19. 개인정보보호를 위한 비식별화 기술에 대한 설명으로 가장 적절한 것은?

① 가명 처리는 정보를 공백이나 노이즈 등으로 변환해 개인 식별을 방지하는 방법으로 이름, ID 등에 사용된다.

② 범주화는 정보를 일정 범위로 구간화하여 숨기는 방식으로 나이, 거주 지역 등에 사용된다.

③ 데이터 마스킹은 식별 가능한 내용을 삭제하는 방식으로 주민등록번호, 날짜 정보 등에 사용된다.

④ 데이터값 삭제는 일부 혹은 전체 정보를 집계하여 개인의 특성을 알아볼 수 없도록 하는 기법으로 나이, 몸무게 등에 사용된다.

정답 2번

출제 포인트

개인정보 비식별화 기술인 가명처리, 범주화, 데이터 마스킹, 데이터 값 삭제의 처리 방식과 적용 예를 구분하는 문제이다.

핵심 개념

가명처리는 식별자를 다른 값으로 대체하여 추가 정보 없이는 특정 개인을 알아보기 어렵게 하는 방식이다.
범주화는 구체적인 값을 구간이나 범주로 바꾸어 정보의 정밀도를 낮추는 방식이다.
데이터 마스킹은 값의 전부 또는 일부를 임의의 문자나 기호 등으로 가려 원래 값을 드러내지 않는 방식이다.
데이터 값 삭제는 식별에 이용될 수 있는 값이나 레코드의 전부 또는 일부를 제거하는 방식이다.

NoteBase

선지별 해설

① X — 가명처리는 이름이나 ID 같은 식별자를 다른 값으로 대체하는 방식이다. 틀린 표현은 '공백이나 노이즈 등으로 변환'이며, 올바르게는 '식별자를 가명·코드 등 다른 값으로 대체한다'이다.
② O — 범주화는 연속적이거나 세부적인 값을 일정 구간으로 묶어 정밀도를 낮춘다. 나이를 연령대로, 상세 거주지를 넓은 지역 단위로 바꾸는 것이 이에 해당한다.
③ X — 데이터 마스킹은 식별 가능한 내용을 삭제하는 것이 아니라 값의 일부 또는 전부를 다른 문자나 기호로 가린다. 틀린 표현은 '삭제하는 방식'이며, 올바르게는 '주민등록번호나 날짜 정보 등의 일부 또는 전부를 가려 원래 값을 노출하지 않는 방식'이다.
④ X — 일부 또는 전체 정보를 집계하는 것은 데이터 값 삭제의 정의가 아니다. 틀린 표현은 '집계하여'이며, 올바르게는 '식별에 이용될 수 있는 데이터 값이나 레코드의 전부 또는 일부를 제거한다'이다. 여러 사람의 값을 합쳐 개인별 특성을 알기 어렵게 하는 방식은 집계 처리에 해당한다.

오답 포인트

①의 '변환', ③의 '삭제', ④의 '집계'처럼 비슷해 보이는 처리 동사를 서로 바꾸어 놓은 것이 함정이다. 특히 마스킹은 값을 가리는 것이고 삭제는 값을 제거하는 것이므로 구별해야 한다.

다시 나오면 이렇게 판단

처리 동사를 기준으로 판단한다. 다른 값으로 바꾸면 가명처리, 구간으로 묶으면 범주화, 문자·기호로 가리면 마스킹, 값을 없애면 데이터 값 삭제이다.

암기 포인트

가명은 대체, 범주는 구간화, 마스킹은 가리기, 삭제는 제거이다.

NoteBase

20. 애자일 모델의 특징에 대한 설명으로 가장 적절한 것은?

① 시스템 개발 시 위험을 최소화하기 위해 점진적으로 완벽한 시스템으로 개발해 나가는 방법이다.

② 팀 구성원과 이해 관계자 간의 협업과 의사소통이 중요하며 계획, 위험 분석, 엔지니어링 및 평가의 4단계로 구성된다.

③ 일정한 주기를 가지고 끊임없이 프로토 타입을 만들어내며 적시에 필요한 요구를 더하고 수정하여 소프트웨어를 개발해 나간다.

④ 각 단계에서 계획 및 문서화가 강조되며 요구사항이 잘 정의된 간단한 프로젝트에 적합하다.

정답 3번

출제 포인트

애자일 모델의 반복적·점진적 개발 방식과 폭포수 모델, 나선형 모델의 특징을 구분하는 문제이다.

핵심 개념

애자일 모델은 짧은 개발 주기를 반복하면서 실행 가능한 소프트웨어를 점진적으로 만들고, 고객의 피드백과 변경 요구를 신속하게 반영하는 개발 방식이다.
애자일은 반복·점진 개발, 변화 수용, 협업과 의사소통을 중시한다. 반면 폭포수 모델은 단계별 계획과 문서화를 중시하고, 나선형 모델은 반복 과정에서 위험 분석을 핵심으로 삼는다.

NoteBase

선지별 해설

① X — 틀린 표현은 '점진적으로 완벽한 시스템으로 개발'한다는 부분이다. 애자일은 처음부터 완벽한 시스템을 목표로 고정된 계획을 따르는 방식이 아니라, 짧은 주기마다 실행 가능한 결과물을 만들고 피드백을 반영하여 점진적으로 개선함으로써 위험을 줄이는 방식이다.
② X — 팀 구성원 및 이해 관계자의 협업과 의사소통을 중시한다는 부분은 애자일의 특징이지만, '계획, 위험 분석, 엔지니어링 및 평가의 4단계'는 나선형 모델의 반복 주기에 해당한다. 올바르게는 애자일이 짧은 반복 주기에서 개발·검토·피드백을 수행한다고 표현해야 한다.
③ O — 일정한 짧은 주기를 반복하여 결과물을 계속 만들고, 필요한 요구를 추가하거나 수정하면서 소프트웨어를 발전시킨다는 설명은 변화 수용과 반복·점진 개발을 중시하는 애자일 모델의 특징에 부합한다.
④ X — 틀린 표현은 '각 단계에서 계획 및 문서화가 강조'되고 '요구 사항이 잘 정의된 간단한 프로젝트에 적합'하다는 부분이다. 이는 단계가 순차적이고 요구 사항의 사전 확정을 중시하는 폭포수 모델의 특징에 가깝다. 올바르게는 애자일이 요구 사항의 변경 가능성이 크고 지속적인 피드백이 필요한 프로젝트에 적합하다고 표현해야 한다.

오답 포인트

②는 협업과 의사소통이라는 애자일 키워드 뒤에 나선형 모델의 4단계를 붙였고, ④는 애자일과 반대되는 폭포수 모델의 계획·문서화 중심 특징을 제시했다. 선지의 앞부분만 보고 판단하면 모델이 뒤섞인 표현을 놓치기 쉽다.

다시 나오면 이렇게 판단

'짧은 반복, 점진적 결과물, 고객 피드백, 요구 변경 수용'이 보이면 애자일로 판단한다. '순차적 단계, 사전 계획, 문서화, 확정된 요구'는 폭포수 모델이고, '반복적 위험 분석'은 나선형 모델로 구분한다.

암기 포인트

애자일은 짧게 반복하며 변화를 반영하고, 폭포수는 단계별로 내려가며, 나선형은 위험을 분석하며 반복한다.

NoteBase

21. 다음 C 프로그램의 실행 결과로 가장 적절한 것은?

```
#include <stdio.h>
int main()
{
    int *ptr, x = 9;
    ptr = &x;
    *ptr += 1;

    printf("%d, %d\n", *ptr, x);
}
```

① 9, 9　　② 9, 10
③ 10, 9　　④ 10, 10

정답 4번

출제 포인트

포인터에 변수의 주소를 저장한 뒤 역참조 연산으로 값을 변경하면, 포인터가 가리키는 원래 변수의 값도 함께 변경된다는 점을 묻는 문제이다.

조건 정리

초깃값: x는 9이다.
주소 연결: ptr = &x에 의해 ptr은 x의 주소를 가리킨다.
출력 대상: *ptr과 x를 차례로 출력한다.

사용할 원리

*ptr은 ptr이 가리키는 메모리 공간의 값이다. ptr이 x의 주소를 가리키므로 *ptr과 x는 서로 다른 값이 아니라 같은 저장 공간의 값을 나타낸다. 따라서 *ptr += 1은 x의 값을 1 증가시키는 것과 같다.

실행 과정

1. int *ptr, x = 9;에서 x를 9로 초기화한다.
2. ptr = &x;에서 x의 주소를 ptr에 저장한다.
3. *ptr += 1;은 ptr이 가리키는 x의 값을 9에서 10으로 증가시킨다.
4. printf에서 *ptr과 x는 모두 같은 저장 공간의 값 10을 읽으므로 10, 10이 출력된다.

NoteBase

선지 판단

① X — 9, 9는 *ptr += 1의 실행 결과를 반영하지 않은 값이다. 올바르게는 *ptr과 x가 모두 10이므로 10, 10이다.
② X — 9, 10은 *ptr과 x가 같은 저장 공간을 참조한다는 점을 놓친 값이다. *ptr도 증가된 x를 읽으므로 올바르게는 10, 10이다.
③ X — 10, 9는 역참조를 통한 변경이 원래 변수 x에는 반영되지 않는다고 잘못 판단한 값이다. *ptr += 1이 x를 직접 10으로 바꾸므로 올바르게는 10, 10이다.
④ O — *ptr과 x가 동일한 메모리 공간의 증가된 값 10을 각각 출력하므로 10, 10이 맞다.

오답 포인트

핵심 함정은 *ptr과 x를 별개의 값을 가진 변수처럼 보는 것이다. ptr에는 x의 주소가 저장되어 있으므로 *ptr을 변경하면 x 자체가 변경된다.

다시 나오면 이렇게 판단

포인터 문제에서는 먼저 ptr이 어느 변수의 주소를 가리키는지 연결한 뒤, *ptr에 대한 연산을 그 변수에 대한 연산으로 바꾸어 실행 순서대로 값을 추적한다.

암기 포인트

ptr = &x이면 *ptr은 x이며, *ptr의 변경은 곧 x의 변경이다.

NoteBase

22. 레드-블랙 트리에 대한 설명으로 가장 적절하지 않은 것은?

① 동일한 키에 대해 여러 경로를 통해 삽입 또는 삭제가 가능하다.

② 삽입 및 삭제 시 색상 조건만으로 회전 필요 여부를 결정할 수 있다.

③ 기존 이진 탐색 트리(BST)의 탐색 기능을 그대로 유지할 수 있다.

④ 레드-블랙 트리는 2-3-4 트리의 구조를 단순화하여 이진 탐색 트리로 표현할 수 있다.

정답 1번

출제 포인트

레드-블랙 트리가 이진 탐색 트리의 탐색 규칙을 유지하면서 색상 규칙과 회전으로 균형을 조정한다는 점을 묻는 문제이다.

핵심 개념

레드-블랙 트리는 각 노드에 적색 또는 흑색을 부여하고 정해진 색상 규칙을 유지하여 높이가 한쪽으로 지나치게 치우치지 않도록 하는 균형 이진 탐색 트리이다.
탐색은 일반 이진 탐색 트리처럼 키 비교 결과에 따라 왼쪽 또는 오른쪽 자식으로 진행하므로, 특정 키에 이르는 탐색 경로는 하나로 결정된다. 삽입·삭제 뒤 색상 규칙이 깨지면 재색칠과 회전으로 복구한다.
레드-블랙 트리는 2-3-4 트리의 여러 키를 가진 노드를 적색 연결로 표현한 이진 탐색 트리 관점으로 이해할 수 있다.

선지별 해설

① X — '동일한 키에 대해 여러 경로를 통해 삽입 또는 삭제가 가능하다'는 표현이 틀렸다. 이진 탐색 트리에서는 키 비교에 따라 탐색 방향이 정해지므로 특정 키를 찾는 경로는 하나이다. 올바르게는 삽입·삭제 대상의 위치를 하나의 탐색 경로로 찾은 뒤, 필요한 경우 재색칠과 회전으로 균형을 복구한다고 해야 한다.
② O — 삽입·삭제 후 노드·부모·삼촌 또는 형제 등의 색상 관계를 검사하면 레드-블랙 규칙 위반과 회전 필요 여부를 판단할 수 있다. 실제 회전 방향은 해당 노드들의 위치 관계까지 고려해 정한다.
③ O — 레드-블랙 트리는 이진 탐색 트리의 대소 관계를 그대로 유지하므로 기존 BST와 같은 방식으로 키를 탐색할 수 있다.
④ O — 2-3-4 트리의 한 노드에 함께 들어 있는 키 관계를 적색 연결로 펼치면 이진 형태의 레드-블랙 트리로 표현할 수 있다. 따라서 레드-블랙 트리는 2-3-4 트리와 대응되는 이진 탐색 트리 표현으로 이해할 수 있다.

NoteBase

오답 포인트

①의 '여러 경로'는 삽입·삭제 뒤 여러 종류의 재조정 사례가 있다는 사실을 탐색 경로와 바꾸어 표현한 함정이다. 재색칠이나 회전 방식은 여러 가지일 수 있지만, 특정 키를 찾아 내려가는 BST 탐색 경로 자체는 하나이다.

다시 나오면 이렇게 판단

레드-블랙 트리 관련 선지는 먼저 BST의 대소 관계와 단일 탐색 경로를 유지하는지 확인하고, 그다음 색상 규칙 위반을 재색칠·회전으로 복구하는지를 확인하면 판단할 수 있다.

암기 포인트

탐색은 BST처럼 한 경로, 균형은 색상과 회전, 구조는 2-3-4 트리와 대응한다.

NoteBase

23. TCP/IP의 전송 계층에 해당하는 프로토콜로 가장 적절한 것은?

① ARP ② DNS

③ FTP ④ UDP

정답 4번

출제 포인트

TCP/IP의 각 계층에서 동작하는 대표 프로토콜을 구분할 수 있는지를 묻는 문제이다.

핵심 개념

응용 계층은 사용자의 네트워크 서비스를 제공하며 DNS와 FTP가 해당한다. 전송 계층은 종단 간 데이터 전송을 담당하며 TCP와 UDP가 해당한다. 인터넷 계층은 논리 주소 지정과 패킷 전달을 담당하며 IP와 ARP 등이 관련된다.
UDP는 연결 설정 없이 데이터그램을 전송하는 비연결형 전송 계층 프로토콜이다.

선지별 해설

① X — ARP는 전송 계층 프로토콜이 아니다. IP 주소에 대응하는 MAC 주소를 알아내는 주소 변환 프로토콜로, TCP/IP 계층 구분에서는 인터넷 계층에 관련된다. 올바르게는 'ARP는 IP 주소를 MAC 주소로 대응시키는 프로토콜이다'라고 해야 한다.
② X — DNS는 전송 계층 프로토콜이 아니라 도메인 이름을 IP 주소로 변환하는 응용 계층 프로토콜이다. 올바르게는 'DNS는 응용 계층에 해당한다'라고 해야 한다.
③ X — FTP는 전송 계층 프로토콜이 아니라 파일을 송수신하는 응용 계층 프로토콜이다. 올바르게는 'FTP는 응용 계층에 해당한다'라고 해야 한다.
④ O — UDP는 TCP와 함께 전송 계층에 속하며, 연결을 설정하지 않고 데이터그램 단위로 데이터를 전송한다.

오답 포인트

ARP, DNS, FTP도 모두 TCP/IP에서 사용하는 프로토콜이므로 계층을 구분하지 않으면 혼동하기 쉽다. 특히 DNS가 이름을 IP 주소로 바꾼다는 이유만으로 인터넷 계층이라고 판단하면 안 된다. DNS는 이름 변환 서비스를 제공하는 응용 계층 프로토콜이다.

다시 나오면 이렇게 판단

종단 간 전송 방식을 정하는 TCP·UDP는 전송 계층, 사용자에게 이름 변환이나 파일 전송 서비스를 제공하는 DNS·FTP는 응용 계층으로 구분한다. ARP는 IP 주소와 MAC 주소를 대응시키는 기능으로 판단한다.

NoteBase

암기 포인트

전송 계층은 TCP·UDP, 응용 계층은 DNS·FTP, 주소 변환은 ARP이다.

NoteBase

24. 소프트웨어 구현 시 응집도(Cohesion)와 결합도(Coupling)의 설명으로 가장 적절한 것은?

① 모듈 내의 응집도는 최대화되어야 하며 모듈 간의 결합도는 최소화되어야 한다.
② 모듈 내의 응집도는 최대화되어야 하며 모듈 간의 결합도 또한 최대화되어야 한다.
③ 모듈 내의 응집도는 최소화되어야 하며 모듈 간의 결합도는 최대화되어야 한다.
④ 모듈 내의 응집도는 최소화되어야 하며 모듈 간의 결합도 또한 최소화되어야 한다.

정답 1번

출제 포인트

이 문제는 모듈 내부의 응집도와 모듈 사이의 결합도를 어떤 방향으로 설계해야 하는지를 묻는 문제이다.

핵심 개념

응집도는 한 모듈 안의 구성 요소들이 하나의 목적을 위해 얼마나 밀접하게 관련되어 있는지를 나타내며, 높을수록 좋다.
결합도는 서로 다른 모듈이 얼마나 강하게 의존하는지를 나타내며, 낮을수록 좋다.
좋은 모듈 설계의 기본 원칙은 높은 응집도와 낮은 결합도이다.

선지별 해설

① O — 모듈 내부의 응집도를 최대화하고 모듈 간의 결합도를 최소화한다는 설명은 좋은 모듈 설계의 원칙과 일치한다.
② X — 틀린 표현은 '모듈 간의 결합도 또한 최대화'이다. 모듈 사이의 의존성을 줄여야 하므로 결합도는 최소화되어야 한다. 올바르게는 '모듈 내의 응집도는 최대화하고 모듈 간의 결합도는 최소화해야 한다'이다.
③ X — '응집도는 최소화'와 '결합도는 최대화'가 모두 반대로 서술되었다. 올바르게는 '모듈 내의 응집도는 최대화하고 모듈 간의 결합도는 최소화해야 한다'이다.
④ X — 결합도를 최소화한다는 부분은 맞지만 '응집도는 최소화'가 틀렸다. 올바르게는 '모듈 내의 응집도는 최대화하고 모듈 간의 결합도도 최소화해야 한다'이다.

오답 포인트

응집도와 결합도를 모두 낮추거나 모두 높여야 한다고 생각하기 쉽다. 그러나 응집도는 모듈 내부의 관련성이므로 높이고, 결합도는 모듈 사이의 의존성이므로 낮춰야 한다.

NoteBase

다시 나오면 이렇게 판단

‘모듈 내부’가 나오면 응집도는 높게, ‘모듈 사이의 의존’이 나오면 결합도는 낮게 설계하는지를 확인한다.

암기 포인트

모듈 안은 끈끈하게: 응집도는 높게. 모듈 사이는 독립적으로: 결합도는 낮게.

NoteBase

25. 운영체제의 단편화(Fragmentation) 문제에 대한 설명으로 가장 적절한 것은?

① 단편화는 프로세스 간의 문맥 교환 속도를 높이기 위해 의도적으로 발생시키는 메모리 최적화 전략이며 이를 방지하기 위해 페이지 크기를 줄이는 것이 효과적이다.

② 내부 단편화는 주기억장치 전체가 비어 있음에도 불구하고 외부 단편화가 해결되지 않아 프로세스가 적재되지 못하는 현상을 의미한다.

③ 단편화는 주기억장치에서 사용되지 않는 비연속적인 작은 조각들이 생기는 현상이며 이를 해결하기 위해 인접한 자유 공간을 하나로 합치는 통합(Coalescing) 기법이나 메모리의 데이터를 재배치하여 연속된 공간을 확보하는 압축(Compaction) 기법이 사용된다.

④ 단편화를 해결하기 위해 가장 일반적으로 사용하는 방법은 고정된 위치에만 프로세스를 할당하여 주소 공간의 이동을 최소화하는 방식이며 이는 동적 메모리 할당보다 유연성이 높다.

정답 3번

출제 포인트

운영체제의 단편화가 발생하는 위치와 내부·외부 단편화의 차이, 외부 단편화를 줄이는 통합과 압축 기법을 구별하는 문제이다.

핵심 개념

단편화: 메모리의 일부가 낭비되어 사용 가능한 공간을 효율적으로 활용하지 못하는 현상이다.
내부 단편화: 할당된 메모리 블록의 크기가 요청한 크기보다 커서 블록 내부에 남는 공간이 생기는 현상이다.
외부 단편화: 전체 빈 공간의 합은 충분하지만 빈 공간이 여러 곳에 흩어져 있어 필요한 크기의 연속 공간을 확보하지 못하는 현상이다.
통합은 서로 인접한 빈 공간을 하나로 합치고, 압축은 사용 중인 영역을 재배치하여 흩어진 빈 공간을 하나의 큰 연속 공간으로 모은다.

NoteBase

선지별 해설

① X
단편화는 문맥 교환 속도를 높이려고 의도적으로 만드는 최적화 전략이 아니라 메모리 할당과 해제 과정에서 발생하는 공간 낭비 현상이다. 또한 페이지 크기를 줄이는 것은 페이지 내부의 남는 공간을 줄여 내부 단편화를 완화할 수 있지만 단편화 전체를 방지하는 일반적 해법이라고 할 수 없다. 틀린 표현인 '의도적으로 발생시키는 메모리 최적화 전략'은 '메모리 관리 과정에서 비의도적으로 발생하는 공간 낭비 현상'으로 고쳐야 한다.

② X
주기억장치 전체가 비어 있으면 빈 공간이 여러 곳에 흩어질 수 없으므로 외부 단편화 때문에 프로세스를 적재하지 못하는 상황이 발생하지 않는다. 외부 단편화는 전체 빈 공간의 합은 충분하지만 빈 공간이 여러 곳에 분산되어 필요한 크기의 연속 공간을 확보하지 못하는 현상이다. 따라서 이 선지는 '주기억장치 전체가 비어 있음'을 '전체 빈 공간의 합은 충분함'으로, '내부 단편화'를 '외부 단편화'로 각각 고쳐야 한다.

③ O
주기억장치에 작고 비연속적인 빈 공간이 생기는 것은 외부 단편화의 전형적인 모습이다. 인접한 빈 공간은 통합으로 합칠 수 있고, 사용 중인 영역을 재배치하는 압축으로 흩어진 빈 공간을 큰 연속 공간으로 모을 수 있으므로 적절한 설명이다.

④ X
프로세스를 고정된 위치에만 두는 방식은 주소 이동을 제한하므로 유연성이 높은 해결책이 아니며, 흩어진 빈 공간을 하나로 모으는 압축에도 프로세스의 동적 재배치가 필요하다. 틀린 표현인 '고정된 위치에만 프로세스를 할당하며 동적 메모리 할당보다 유연성이 높다'는 '재배치가 가능한 방식으로 프로세스를 이동시켜 연속된 빈 공간을 확보하며, 고정 위치 할당보다 유연하다'로 고쳐야 한다.

오답 포인트

②는 '주기억장치 전체가 비어 있음'과 '내부 단편화'가 모두 틀린 표현이다. 주기억장치 전체가 비어 있으면 빈 공간이 분산된 상태가 아니므로 외부 단편화로 적재가 막히지 않는다. 전체 빈 공간의 합은 충분하지만 빈 공간이 블록 밖 여러 곳에 흩어져 큰 연속 공간을 만들지 못하는 현상이 외부 단편화이고, 할당된 블록 안에 공간이 남는 현상이 내부 단편화이다.

다시 나오면 이렇게 판단

낭비 공간의 위치를 먼저 본다. 할당된 블록 안에 남으면 내부 단편화, 블록 밖의 빈 공간이 여기저기 흩어져 있으면 외부 단편화로 판단한다. 이어서 인접한 빈 공간만 합치면 통합, 사용 중인 영역까지 옮겨 큰 연속 공간을 만들면 압축으로 구별한다.

NoteBase

암기 포인트

내부는 할당 블록 안의 낭비, 외부는 블록 밖에 흩어진 빈 공간이다. 인접 공간은 통합, 흩어진 공간은 재배치 후 압축이다.

NoteBase