

※ 다음 문제를 읽고 알맞은 것을 골라 답안카드의 답란(가, 나, 다, 라)에 표기하십시오.

제1과목 데이터베이스

1. 탐색 방법 중 키 값으로부터 레코드가 저장되어 있는 주소를 직접 계산하여, 산출된 주소로 바로 접근하는 방법으로 키-주소 변환 방법이라고도 하는 것은?

- 가. 이진 탐색
- 나. 피보나치 탐색
- 다. 해싱 탐색
- 라. 블록 탐색

2. SQL의 DROP 문은 어떠한 목적으로 사용되는가?

- 가. 스키마, 테이블 및 뷰의 제거시에 사용된다.
- 나. 스키마, 테이블 및 뷰의 정의시에 사용된다.
- 다. 데이터베이스의 무결성을 체크하는데 사용된다.
- 라. 데이터베이스를 최적화하는데 사용된다.

3. 데이터의 중복으로 인하여 관계연산을 처리할 때 곤란한 현상이 발생하는 것을 무엇이라 하는가?

- 가. 이상(Anomaly)
- 나. 제한(Restriction)
- 다. 종속성(Dependency)
- 라. 변환(Translation)

4. 관계 데이터 모델링 중 BCNF(Boyce-Codd Normal Form)에 대한 옳은 설명으로만 짝지어진 것은?

- ㄱ. BCNF에 속하는 릴레이션은 반드시 제3정규형(Third Normal Form)에 속한다.
- ㄴ. 제3정규형에 속하지만 BCNF에 속하지 않는 릴레이션이 있다.
- ㄷ. 복합 속성을 허용하지 않는다.
- ㄹ. 완전 함수적 종속성 개념에 기반을 두었다.

- 가. ㄱ, ㄴ, ㄷ
- 나. ㄴ, ㄷ
- 다. ㄱ, ㄴ, ㄷ
- 라. ㄱ, ㄴ, ㄷ, ㄹ

5. 물리적 데이터 독립성에 대한 설명으로 가장 적합한 것은?

- 가. 기존 응용 프로그램에 영향을 주지 않고 데이터의 물리적 구조를 변경할 수 없는 것을 말한다.
- 나. 기존 응용 프로그램에 영향을 주지 않고 데이터의 물리적 구조를 변경할 수 있는 것을 말한다.
- 다. 기존 응용 프로그램에 영향을 변경하면 데이터의 물리적 구조도 이에 따라 변경되는 것을 말한다.
- 라. 데이터의 물리적 구조를 변경할 때, 자동적으로 데이터의 논리적 구조도 변경되는 것을 말한다.

6. 관계 데이터 모델에서 릴레이션(Relation)에 포함되어 있는 튜플(Tuple)의 수를 무엇이라고 하는가?

- 가. Degree
- 나. Cardinality
- 다. Attribute
- 라. Cartesian Product

7. 3계층 스키마 중 개념(Conceptual) 스키마에 대한 설명으로 옳지 않은 것은?

- 가. 한 기관 전체에서 필요로 하는 데이터베이스의 전체적인 논리적 구조이다.
- 나. 물리적 저장 장치의 입장에서 본 데이터베이스 구조이다.
- 다. 개체간의 관계와 유지해야 할 제약 조건을 나타낸다.
- 라. 접근 권한, 보안 정책, 무결성 규칙을 명세한다.

8. 직접 접근 방식(DAM : Directed Access Method)에 대한 설명으로 거리가 먼 것은?

- 가. 데이터의 입출력이 빈번히 발생하는 곳에 응용하는 것이 좋다.
- 나. 해싱 함수를 이용하여 레코드의 저장 위치를 결정한다.
- 다. 다른 레코드를 참조하지 않고 어떤 레코드를 접근할 수 있다.
- 라. 기억 공간의 효율성이 매우 좋다.

9. 데이터베이스 무결성과 보안의 차이점에 대한 설명 중 옳은 것은?

- 가. 무결성은 권한이 있는 사용자로부터 데이터베이스를 보호하는 것이고, 보안은 권한이 없는 사용자로부터 데이터베이스를 보호하는 것이다.
- 나. 무결성은 권한이 없는 사용자로부터 데이터베이스를 보호하는 것이고, 보안은 권한이 있는 사용자로부터 데이터베이스를 보호하는 것이다.
- 다. 무결성과 보안은 모두 권한이 있는 사용자로부터 데이터베이스를 보호하는 것이지만, 보안은 사용자 계정과 비밀번호를 관리한다.
- 라. 무결성과 보안은 모두 권한이 없는 사용자로부터 데이터베이스를 보호하는 것이지만, 무결성은 DBMS가 자동적으로 보장해 준다.

10. 다음의 수식을 후위 순회(Postorder Traversal)한 결과는?

$$A / B * C * D + E$$

- 가. + ** / A B C D E
- 나. A / B * C * D + E
- 다. A B / C * D * E +
- 라. A B C D E / * * +

11. 데이터베이스 설계 단계와 그 단계에서 수행되는 결과의 연결이 잘못된 것은?

- 가. 개념적 설계 단계 - 트랜잭션 모델링
- 나. 논리적 설계 단계 - 목표 DBMS에 독립적인 논리 스키마 설계
- 다. 물리적 설계 단계 - 목표 DBMS에 맞는 물리적 구조 설계
- 라. 구현 단계 - 목표 DBMS DDL로 스키마 작성

12. 물리적 데이터베이스 설계를 수행할 때 결정할 사항으로 거리가 먼 것은?

- 가. 어떤 인덱스를 만들 것인지에 대한 고려
- 나. 성능 향상을 위한 개념 스키마의 변경 여부 검토
- 다. 빈번한 질의와 트랜잭션들의 수행속도를 높이기 위한 고려
- 라. 개념스키마와 외부스키마 설계

13. 릴레이션이 가지는 특성으로 옳지 않은 것은?

- 가. 한 릴레이션에 나타난 속성값은 논리적으로 분해 가능한 값이어야 한다.
- 나. 튜플은 서로 다른 값을 갖는다.
- 다. 각 속성은 릴레이션 내에서 유일한 이름을 가진다.
- 라. 하나의 릴레이션에서 튜플의 순서는 없다.

14. 시스템 카탈로그에 관한 설명으로 옳지 않은 것은?

- 가. 가상테이블이며 메타데이터라고도 한다.
- 나. 시스템 카탈로그 내의 각 테이블은 DBMS에서 지원하는 개체들에 관한 정보를 포함한다.
- 다. 시스템의 사용자들에 관한 정보를 포함하고 있다.
- 라. DBMS가 스스로 생성하고 유지하는 데이터베이스 내의 특별한 테이블들의 집합체이다.

15. 뷰(VIEW)에 관한 설명 중 잘못된 것은?

- 가. 뷰는 SQL에서 CREATE VIEW 명령으로 정의된다.
- 나. 뷰는 하나 이상의 기본 테이블로부터 유도되어 만들어지는 가상 테이블이다.
- 다. 뷰는 INSERT, DELETE, UPDATE 등을 이용한 삽입, 삭제, 갱신 연산이 항상 허용된다.
- 라. 뷰의 정의는 ALTER 문을 이용하여 변경할 수 없다.

16. 데이터베이스 무결성에 관한 설명으로 옳지 않은 것은?

- 가. 개체 무결성 규정은 한 릴레이션의 기본 키를 구성하는 어떠한 속성 값도 널(NULL) 값이나 중복 값을 가질 수 없음을 규정하는 것이다.
- 나. 무결성 규정에는 규정이름, 검사시기, 제약조건 등을 명시한다.
- 다. 도메인 무결성 규정은 주어진 튜플의 값이 그 튜플이 정의된 도메인에 속한 값이어야 한다는 것을 규정하는 것이다.
- 라. 트리거는 트리거 조건이 만족되는 경우에 취해야 하는 조치를 명세한다.

17. 데크(Deque)에 대한 옳은 설명으로만 짝지어진 것은?

- ① 양끝에서 노드의 삽입과 삭제가 모두 가능하다.
- ② 하나의 포인터를 사용한다.
- ③ Double Ended Queue의 약자이다.
- ④ 선형 구조이다.

- 가. ①, ②
- 나. ①, ②, ③
- 다. ①, ③, ④
- 라. ①, ④

18. 분산 데이터베이스 시스템의 설명으로 옳지 않은 것은?

- 가. 소프트웨어 개발비용이 감소한다.
- 나. 지역 자치성이 보장된다.
- 다. 시스템의 확장이 용이하다.
- 라. 신뢰도가 향상된다.

19. Which of the following does not belong to the DML statement of SQL?

- 가. SELECT
- 나. DELETE
- 다. CREATE
- 라. INSERT

20. 괄호 속 내용으로 가장 적합한 것은?

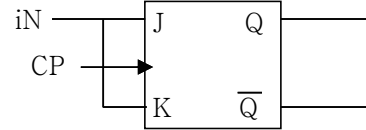
A quick method for searching an ordered, dense list for a particular record by successively looking at that half of the remaining portion of the list in which the record is known to be, is called a ().

- 가. tree search
- 나. block search
- 다. binary search
- 라. sequential search

21. 등각속도(CAV) 방식의 특징이 아닌 것은?

- 가. 모든 트랙의 저장 밀도가 같다.
- 나. 디스크 저장 공간이 비효율적으로 사용된다.
- 다. 회전 구동장치가 간단하다.
- 라. 디스크 평판이 일정한 속도로 회전한다.

22. JK 플립플롭을 그림과 같이 연결하면 어떤 플립플롭과 같은 동작을 하는가?



- 가. D
- 나. RS
- 다. T
- 라. Master-slave

23. 직접 메모리 액세스(DMA)의 특징이 아닌 것은?

- 가. CPU의 도움없이 메모리와 I/O 장치 사이에서 전송을 시행한다.
- 나. CPU와 DMA 제어기는 메모리와 버스를 공유한다.
- 다. CPU의 상태 보존은 반드시 필요하다.
- 라. 사이클 스틸을 발생하여 메모리 장치와 I/O 장치 사이의 자료 전송을 수행한다.

24. 캐시(Cache) 메모리에서 특정 내용을 찾는 방식 중 매핑 방식에 주로 사용되는 메모리는?

- 가. Nano Memory
- 나. Associative Memory
- 다. Virtual Memory
- 라. Stack Memory

25. 컴퓨터의 주기억장치 용량이 8192비트이고, 워드 길이가 16비트일 때 PC(Program Counter), AR(Address Register)와 DR(Data Register)의 크기는?

- 가. PC=8, AR=9, DR=16
- 나. PC=9, AR=9, DR=16
- 다. PC=16, AR=16, DR=16
- 라. PC=8, AR=16, DR=16

26. Exclusive-OR Gate의 출력은?

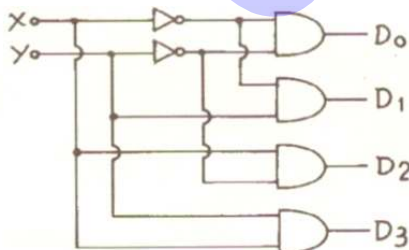
- 가. (AB)'+AB
- 나. A'B'+AB
- 다. A'B+AB'
- 라. AB'+AB'

27. 그림의 진리표에서 출력 Y를 최소화 하면?

| 입 력 | | | 출 력 |
|-----|---|---|-----|
| A | B | C | Y |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

- 가. Y=A'B
- 나. Y=AB
- 다. Y=A+B'
- 라. Y=C'

28. 컴퓨터 내부에서 시스템 순간 순간의 상태를 나타내는 것은?
 가. SP
 나. PSW
 다. Interrupt
 라. MAR
29. 주소 지정 방식(Addressing Mode) 중에서 프로그램 카운터 값에 명령어의 주소 부분을 더해서 실제 주소를 구하는 방식은?
 가. 직접 번지 방식
 나. 즉시 번지 방식
 다. 상대 번지 방식
 라. 레지스터 번지 방식
30. 컴퓨터 시스템이 작동되면 먼저 프로그램 카운터의 초기 주소값이 결정되고 주소에 의하여 명령어가 기억장치로부터 읽혀지는 것을 무엇이라 하는가?
 가. 인출(Fetch)
 나. 실행(Execute)
 다. 간접(Indirect)
 라. 인터럽트(Interrupt)
31. OP-Code가 4비트면 연산자의 종류는 몇 개가 생성될 수 있는가?
 가. 2^4-1
 나. 2^4
 다. 2^3
 라. 2^3-1
32. 0-주소 인스트럭션과 관계 있는 것은?
 가. Scratch-pad Register
 나. Accumulator
 다. Stack
 라. Instruction Buffer
33. 인터럽트 체제의 기본 요소에 속하지 않는 것은?
 가. 인터럽트 처리 기능
 나. 인터럽트 요청 신호
 다. 인터럽트 스테이트
 라. 인터럽트 취급 루틴
34. 명령어의 Operand 부분에 실제 데이터를 갖고 있는 방식은?
 가. 즉시(Immediate) 주소지정 방식
 나. 베이스(Base) 주소지정 방식
 다. 상대(Relative) 주소지정 방식
 라. 직접(Direct) 주소지정 방식
35. 다음 회로는 무엇인가?



- 가. Decoder
 나. Multiplexer
 다. Encoder
 라. Shifter
36. 하드웨어 신호에 의하여 특정 번지의 서브루틴을 수행하는 것은?
 가. Handshaking Mode

- 나. Vectored Interrupt
 다. DMA
 라. Subroutine Call

37. 다음 중 단항(Unary) 연산이 아닌 것은?
 가. Complement
 나. Rotate
 다. AND
 라. Shift
38. ROM 칩에 필요하지 않은 신호는?
 가. 쓰기 신호
 나. 주소
 다. 읽기 신호
 라. 칩 선택 신호
39. 중앙처리장치가 주기억장치보다 더 빠르기 때문에 프로그램 실행 속도를 중앙처리장치의 속도에 근접하도록 하기 위해서 사용되는 기억장치는?
 가. 가상 기억 장치
 나. 모듈 기억 장치
 다. 보조 기억 장치
 라. 캐시 기억 장치
40. 로더(Loader)의 기능 중 옳지 않은 것은?
 가. 할당(Allocation)
 나. 링킹(Linking)
 다. 재배치(Relocation)
 라. 실행(Execution)

제3과목 운영체제

41. 기억장치의 관리 전략 중 반입(Fetch) 전략의 설명으로 옳은 것은?
 가. 프로그램/데이터를 주기억장치로 가져오는 시기를 결정하는 전략
 나. 프로그램/데이터에 대한 주기억장치 내의 위치를 결정하는 전략
 다. 주기억장치 내의 빈공간 확보를 위해 제거할 프로그램/데이터를 선택하는 전략
 라. 프로그램/데이터의 위치를 이동시키는 전략
42. 데커(Dekker) 알고리즘에 대한 설명 중 옳지 않은 것은?
 가. 교착상태가 발생하지 않음을 보장한다.
 나. 프로세스가 임계영역에 들어가는 것이 무한정 지연될 수 있다.
 다. 공유 데이터에 대한 처리에 있어서 상호배제를 보장한다.
 라. 별도의 특수 명령어 없이 순수하게 소프트웨어로 해결된다.
43. 하나의 프로세스가 작업 수행 과정에서 수행하는 기억 장치 접근에서 지나치게 페이지 폴트가 발생하여 프로세스 수행에 소요되는 시간보다 페이지 이동에 소요되는 시간이 더 커지는 현상은?
 가. 스래싱(Thrashing)
 나. 워킹세트(Working Set)
 다. 세마포어(Semaphore)
 라. 교환(Swapping)
44. 운영체제 형태 중 시대적으로 가장 먼저 생겨난 것은?
 가. 다중처리 시스템
 나. 시분할 시스템
 다. 일괄처리 시스템
 라. 분산처리 시스템
45. SCAN의 무한 대기 발생 가능성을 제거한 것으로 SCAN 보다

61. 소단위명세서(Mini-Specification)에 관한 내용 중 옳지 않은 것은?
 가. 반 페이지나 한 페이지 정도의 크기로 세분된 모듈을 작성할 때 사용한다.
 나. DFD에서는 한 개의 처리공정이 그 대상이 되지만, 한 공정의 기능이 두 가지 이상이거나 더 세분화함으로써 소단위명세서를 이해하기가 쉬워진다면 더욱 세분화될 수도 있다.
 다. 소단위명세서를 작성하는 도구에는 서술문장, 의사결정나무, 의사결정표, 표, 그래프 등이 있다.
 라. 소단위명세서는 구조적 언어를 사용하지 않고, 자연어를 사용하여 이해하기 쉽고 엄밀하게 기술한다.
62. SOFTWARE Project의 비용 결정 요소와 가장 관련이 적은 것은?
 가. 개발자의 능력
 나. 요구되는 신뢰도
 다. 하드웨어의 성능
 라. 개발제품의 복잡도
63. 소프트웨어의 전통적 개발 단계 중 요구분석 단계에 대한 설명으로 옳지 않은 것은?
 가. 프로젝트를 이해할 수 있는 개발의 실질적인 첫 단계이다.
 나. 현재의 상태를 파악하고 문제를 정의한 후, 문제해결과 목표를 명확히 도출하는 단계이다.
 다. 소프트웨어가 가져야 될 기능을 기술하는 단계이다.
 라. 고품질의 소프트웨어를 개발하기 위해 소프트웨어의 내부 구조를 기술하는 단계이다.
64. DFD(Data Flow Diagram)에 대한 설명으로 거리가 먼 것은?
 가. 자료 흐름 그래프 또는 버블(Bubble) 차트라고도 한다.
 나. 구조적 분석 기법에 이용된다.
 다. 시간 흐름의 개념을 명확하게 표현할 수 있다.
 라. DFD의 요소는 화살표, 원, 사각형, 직선(단선/이중선)으로 표시한다.
65. 객체지향 개념에서 연관된 데이터와 함수를 함께 묶어 외부와 경계를 만들고 필요한 인터페이스만을 밖으로 드러내는 과정을 무엇이라고 하는가?
 가. 메시지
 나. 캡슐화
 다. 상속
 라. 다형성
66. 소프트웨어공학의 공학(Engineering)이 가지는 의미와 어울리지 않는 것은?
 가. 예술성
 나. 경제성
 다. 보편타당성
 라. 적시성
67. 현재 프로그램으로부터 데이터, 아키텍처, 그리고 절차에 관한 분석 및 설계 정보를 추출하는 과정은?
 가. 재공학(Re-Engineering)
 나. 역공학(Reverse Engineering)
 다. 순공학(Forward Engineering)
 라. 재사용(Reuse)
68. 객체 지향의 기본 개념 중 객체가 메시지를 받아 실행해야 할 객체의 구체적인 연산을 정의한 것은?

- 가. 메소드
 나. 추상화
 다. 상속성
 라. 캡슐화

69. 프로토타입 모형의 장점으로 가장 적절한 것은?
 가. 프로젝트 관리가 용이하다.
 나. 노력과 비용이 절감된다.
 다. 요구사항을 충실히 반영한다.
 라. 관리와 개발이 명백히 구분된다.
70. 하나의 프로그램을 몇 개의 작은 부분으로 분할하는 경우, 그 분할단위를 일반적으로 모듈(Module)이라고 한다. 다음 중 모듈에 대한 설명으로 옳은 것은?
 가. 모듈의 독립성을 높여주기 위해서는 각 모듈간의 관련성을 최소로 하며, 이 경우에 응집도(Cohesion)는 최소가 된다.
 나. 모듈간의 관련성을 최대한으로 하면 모듈의 독립성은 저하되며, 이 경우에 모듈의 결합도(Coupling)는 최소가 된다.
 다. 복잡성을 감소시키는 수단으로 독립성의 개념이 많이 적용되고 있으며, 모듈의 독립성 척도로서 결합도는 고려 대상이 아니며, 응집도만 적용된다.
 라. 모듈의 결합도는 자료결합도(Data Coupling)로, 모듈의 응집도는 기능적 응집도(Functional Cohesion)로 하는 것이 가장 바람직하다.
71. CASE(Computer Aided Software Engineering)에 대한 설명으로 옳은 것은?
 가. 소프트웨어 생명 주기(Life Cycle)의 전체 단계를 연결시켜 주는 통합된 도구를 제공한다.
 나. CASE 패키지의 3단계는 도식목차, 총괄 다이어그램, 상세 다이어그램으로 구분된다.
 다. 상위(Upper) CASE에서는 주로 코드를 작성하고 테스트하며, 문서화하는 작업을 지원한다.
 라. 소프트웨어 개발의 생산성과 신뢰성에 저하를 가져와 널리 사용되지 못하고 있다.
72. 프로젝트 일정 관리시 사용하는 칸트(Gantt) 차트에 대한 설명으로 옳지 않은 것은?
 가. 막대로 표시하며, 수평 막대의 길이는 각 태스크의 기간을 나타낸다.
 나. 이정표, 기간, 작업, 프로젝트 일정을 나타낸다.
 다. 시간선(Time-Line) 차트라고도 한다.
 라. 작업들간의 상호 관련성, 결정경로를 표시한다.
73. 모듈안의 작동을 자세히 관찰할 수 있으며, 프로그램 원시 코드의 논리적인 구조를 커버(Cover)하도록 테스트 케이스를 설계하는 프로그램 테스트 방법은?
 가. 블랙 박스 테스트
 나. 화이트 박스 테스트
 다. 알파 테스트
 라. 베타 테스트
74. 프로토타이핑(Prototyping) 접근방법을 채용할 때의 이익은 주로 정보문제의 본질에 대한 불확실성과 그 정보문제를 해결하기 위해 사용자가 제시하는 요구의 불확실성을 줄이는데 있다. 다음 중 불확실성 결정 요인에 해당하지 않는 것은?
 가. 지원이 필요한 일로부터의 요구연역(要求演繹)
 나. 사용자와 분석자의 지식과 경험의 수준
 다. 커뮤니케이션 문제가 일어날 가능성

- 다. 통계적 시분할 다중화
- 라. 코드 분할 다중화

91. 다음 전송제어 문자 중 부정적 응답에 해당하는 전송제어 문자는?

- 가. NAK(Negative Acknowledge)
- 나. ACK(Acknowledge)
- 다. EOT(End of Transmission)
- 라. SOH(Start of Heading)

92. OSI 계층의 네트워크 계층에 해당하는 X.25의 계층은?

- 가. 패킷 계층
- 나. 프레임 계층
- 다. 응용 계층
- 라. 세션 계층

93. 컴퓨터를 이용한 정보통신 시스템에서 정확한 데이터를 주고 받기 위해서는 컴퓨터 간의 미리 정해진 약속이 필요하다. 이러한 약속을 무엇이라 하는가?

- 가. Topology
- 나. Protocol
- 다. OSI 7 Layer
- 라. DNS

94. 통계적 TDM에서 다중화된 회선의 데이터 전송율과 접속장치들의 데이터 전송율의 합과의 일반적인 관계는?

- 가. 다중화된 회선의 데이터 전송율 < 접속장치들의 데이터 전송율의 합
- 나. 다중화된 회선의 데이터 전송율 > 접속장치들의 데이터 전송율의 합
- 다. 다중화된 회선의 데이터 전송율 ≥ 접속장치들의 데이터 전송율의 합
- 라. 다중화된 회선의 데이터 전송율 = 접속장치들의 데이터 전송율의 합

95. 패킷 교환망에서 유통되는 패킷의 수를 적절히 조절해 통신망을 효율적으로 사용하고자 하는 제어 기법이 트래픽(Traffic) 제어 기법이다. 다음 중 트래픽 제어 기법에 해당되지 않는 것은?

- 가. 에러 제어(Error Control)
- 나. 흐름 제어(Flow Control)
- 다. 혼잡 제어(Congestion Control)
- 라. Dead-Lock 방지 기법

96. 베이직 데이터 전송제어 절차가 아닌 것은?

- 가. SOH
- 나. STX
- 다. ETX
- 라. FCS

97. 다음 중 대역폭(Bandwidth)에 관한 설명으로 옳은 것은?

- 가. 최고 주파수를 의미한다.
- 나. 최저 주파수를 의미한다.
- 다. 최고 주파수의 절반을 의미한다.
- 라. 최고 주파수와 최저 주파수 사이의 간격을 의미한다.

98. 공중 통신 회선에 교환설비, 컴퓨터 및 단말기 등을 접속시켜 새로운 부가 기능을 제공하는 통신망은?

- 가. VAN
- 나. LAN
- 다. ISDN
- 라. MAN

99. 다음 프로토콜 중 네트워크 계층 구조의 하나인 트랜스포트(Transport) 계층에서 사용되는 프로토콜은?

- 가. TCP
- 나. IP
- 다. Telnet
- 라. SNMP

100. 데이터는 한쪽 방향으로만 흐르고 병목 현상이 드물지만, 두 노드 사이의 채널이 고장나면 전체 네트워크가 손상될 수 있는 단점을 갖는 토폴로지는?

- 가. 링형 토폴로지
- 나. 망형 토폴로지
- 다. 성형 토폴로지
- 라. 계층형 토폴로지

정답 및 해설

1. 다 2. 가 3. 가 4. 가 5. 나 6. 나 7. 나 8. 라 9. 가 10. 다 11. 나 12. 라 13. 가 14. 가
 15. 다 16. 다 17. 다 18. 가 19. 다 20. 다 21. 가 22. 다 23. 다 24. 나 25. 나 26. 다 27. 라 28. 나
 29. 다 30. 가 31. 나 32. 다 33. 다 34. 가 35. 가 36. 나 37. 다 38. 가 39. 라 40. 라 41. 가 42. 나
 43. 가 44. 다 45. 가 46. 다 47. 다 48. 다 49. 다 50. 라 51. 라 52. 나 53. 라 54. 다 55. 다 56. 다
 57. 나 58. 나 59. 나 60. 라 61. 라 62. 다 63. 라 64. 다 65. 나 66. 가 67. 나 68. 가 69. 다 70. 라
 71. 가 72. 라 73. 나 74. 라 75. 가 76. 라 77. 라 78. 다 79. 다 80. 라 81. 나 82. 다 83. 나 84. 나
 85. 나 86. 라 87. 다 88. 다 89. 라 90. 나 91. 가 92. 가 93. 나 94. 가 95. 가 96. 라 97. 라 98. 가
 99. 가 100. 가

1 탐색 방법 중 키 값으로부터 레코드가 저장되어 있는 주소를 직접 계산하여, 산출된 주소로 바로 접근하는 방법으로 키-주소 변환 방법이라고도 하는 것은 해싱(Hasing) 탐색입니다. 보기에 제시된 검색 방법 정도는 특징을 구분할 수 있어야 합니다.

- **이진(Binary) 검색** : 이진(이분) 검색은 전체 파일을 두 개의 서브파일로 분리해 가면서 Key 레코드를 검색하는 방식이다.
- **피보나치(Fibonacci) 검색** : 피보나치 검색은 피보나치 수열에 따라 다음에 비교할 대상을 선정하여 검색하는 방식이다.
- **블록 검색(Block Search)** : 파일을 구성하는 레코드들을 여러 개의 Block으로 분할하여 Block 단위는 순서화시키고, Block 내의 자료는 순서화와 관계없이 저장시킨다. Index 부분을 두어, 각 Block마다 최대 레코드 키값을 가지는 레코드 번호를 저장시킨다.

2 SQL의 DROP 문은 스키마, 테이블 및 뷰의 제거시에 사용됩니다. 데이터 정의어(DDL)의 3가지 유형에 대해서 알아두세요.

데이터 정의어(DDL)의 3가지 유형

- **CREATE** : SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 정의
- **ALTER** : TABLE에 대한 정의를 변경하는 데 사용
- **DROP** : SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 삭제

3 데이터의 중복으로 인하여 관계연산을 처리할 때 곤란한 현상이 발생하는 것을 이상(Anomaly)이라고 합니다. 이상이 발생하는 원인과 이상의 종류를 꼭 기억하세요. 각 이상의 의미는 용어 그대로 이므로 한번 읽어보면 쉽게 이해됩니다.

Anomaly(이상)의 개념 및 종류

- 정규화(Normalization)를 거치지 않으면 데이터베이스 내에 데이터들이 불필요하게 중복되어 릴레이션 조작시 예기치 못한 곤란한 현상이 발생하는데 이를 이상(Anomaly)이라 하며 삽입 이상, 삭제 이상, 갱신 이상이 있다.
- **삽입 이상(Insertion Anomaly)** : 릴레이션에 데이터를 삽입할 때 의도와는 상관없이 원하지 않은 값들도 함께 삽입되는 현상
- **삭제 이상(Deletion Anomaly)** : 릴레이션에서 한 튜플을 삭제할 때 의도와는 상관없는 값들도 함께 삭제되는 연쇄 삭제 현상이 일어나는 현상
- **갱신 이상(Update Anomaly)** : 릴레이션에서 튜플에 있는 속성값을 갱신할 때 일부 튜플의 정보만 갱신되어 정보에 모순이 생기는 현상

4 정규화는 시험에 자주 출제되는 부분입니다. 이 문제를 통해 관계 데이터 모델링 중 BCNF(Boyce-Codd Normal Form)에 대한 특징 만큼은 확실히 파악하세요.

BCNF(Boyce-Codd 정규형)

- 릴레이션 R에서 결정자가 모두 후보키인 관계형이다.
- 3NF에서 후보키가 많고 서로 중첩되는 경우에 적용하는, 강한 제3정규형이라고도 한다.
- 모든 BCNF(Boyce-Codd Normal Form)가 종속성을 보존하는 것은 아니다.
- BCNF의 제약 조건
 - 키가 아닌 모든 속성은 각 키에 대하여 완전 종속해야 한다.
 - 어떤 속성도 키가 아닌 속성에 대해서는 완전 종속할 수 없다.

5 물리적 데이터 독립성이란 기존 응용 프로그램에 영향을 주지 않고 데이터의 물리적 구조를 변경할 수 있는 것을 말합니다.

6 관계 데이터 모델에서 릴레이션(Relation)에 포함되어 있는 튜플(Tuple)의 수를 카디널리티(Cardinality) 또는 기수, 대응수라고 합니다. 관계형 데이터베이스에서 릴레이션을 구성하는 용어들은 매우 중요합니다.

- **디그리(Degree)** : 속성의 수를 의미하며, 차수라고도 함
- **속성(Attribute)** : 데이터베이스를 구성하는 가장 작은 논리적 단위, 릴레이션의 각 열
- **Cartesian Product** : 일반 집합 연산자의 하나로 교차곱이라고 함

7 물리적 저장 장치의 입장에서 본 데이터베이스의 구조는 내부 스키마(Internal Schema)입니다. 스키마에서 출제되는 문제 대부분이 스키마의 3계층을 구분할 수 있는가에 대해 묻고 있습니다. 어떤 경우에도 3계층을 구분할 수 있을 정도로 3계층 각각의 개념을 명확히 하세요.

외부 스키마(External Schema) = 서브 스키마 = 사용자 뷰(View)

- 외부 스키마(External Schema)는 사용자나 응용 프로그램이 각 개인의 입장에서 필요로 하는 데이터베이스의 논리적 구조를 정의한 것이다.
- 외부 스키마(External Schema)는 전체 데이터베이스의 한 논리적인 부분으로 볼 수 있으므로 서브스키마(Subschema)라고도 한다.
- 하나의 데이터베이스 시스템에는 여러 개의 외부 스키마가 존재할 수 있으며, 하나의 외부 스키마를 여러 개의 응용 프로그램이나 사용자가 공유할 수도 있다.
- 같은 데이터베이스에 대해서도 서로 다른 관점을 정의할 수 있도록 허용한다.
- 일반 사용자는 질의어(SQL)를 이용하여 DB를 쉽게 사용할 수 있다.
- 응용 프로그래머는 COBOL, C 등의 언어를 사용하여 DB에 접근한다.

개념 스키마(Conceptual Schema) = 전체적인 뷰(View)

- 개념 스키마는 데이터베이스의 전체적인 논리적 구조로서,

모든 응용 프로그램이나 사용자들이 필요로 하는 데이터를 종합한 조직 전체의 데이터베이스로 하나만 존재한다.

- 개념 스키마는 개체간의 관계와 제약 조건을 나타내고 데이터베이스의 접근 권한, 보안 및 무결성 규칙에 관한 명세를 정의한다.
 - 단순히 스키마(Schema)라고하면 개념 스키마를 의미한다.
 - 기관이나 조직체의 관점에서 데이터베이스를 정의한 것이다.
 - 데이터베이스 관리자(DBA)에 의해서 구성된다.
- 내부 스키마(Internal Schema)**
- 내부 스키마는 데이터베이스의 물리적 구조이다.
 - 내부 스키마는 데이터의 실제 저장 방법을 기술한다.
 - 물리적인 저장장치와 밀접한 계층이다.
 - 시스템 프로그래머나 시스템 설계자가 보는 관점의 스키마이다.

8 직접 접근 방식(DAM ; Directed Access Method)은 기억공간이 많이 요구되므로 공간의 효율성은 낮습니다. 직접 접근 방식의 장·단점을 구분하여 알아두세요.

장점

- 직접 접근 기억장치(DASD)의 물리적 주소를 통하여 파일의 각 레코드에 직접 접근하거나 기록할 수 있으며, 접근 및 기록의 순서에는 제약이 없다.
- 접근 시간이 빠르고 레코드의 삽입, 삭제, 갱신이 용이하다.
- 어떤 레코드라도 평균 접근 시간(Access Time) 내에 검색이 가능하다.

단점

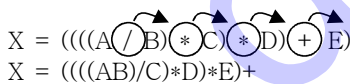
- 레코드의 주소 변환 과정이 필요하며, 이 과정으로 인해 시간이 소요된다.
- 기억공간의 효율이 저하될 수 있다.
- 기억장치의 물리적 구조에 대한 지식이 필요하고, 프로그래밍 작업이 복잡하다
- 충돌이 발생할 염려가 있으므로, 이를 위한 기억공간의 확보가 필요하다.

9 무결성은 권한이 있는 사용자로부터 데이터베이스를 보호하는 것이고, 보안은 권한이 없는 사용자로부터 데이터베이스를 보호하는 것입니다.

10 후위 순회(Postorder Traversal)는 다음에 제시된 방법대로 수행하면 쉽습니다.

- ① 연산 우선순위에 따라 괄호로 묶는다.

$$X = (((A / B) * C) * D) + E$$
- ② 연산자를 해당 괄호의 뒤로 옮긴다.



- ③ 괄호를 제거한다.

$$X = AB/C * D * E +$$

11 논리적 설계 단계에서는 목표 DBMS에 맞는(종속적인) 논리적 스키마를 설계합니다. 각 단계는 구분할 수 있을 정도로 대표적인 특징을 알아두고, 데이터베이스 설계 순서는 반드시 암기하세요.

데이터베이스 설계 순서

- ① **요구 분석** : 요구 조건 명세서 작성
- ② **개념적 설계** : 개념 스키마, 트랜잭션 모델링, E-R 모델
- ③ **논리적 설계** : 목표 DBMS에 맞는 논리적 스키마 설계
- ④ **물리적 설계** : 목표 DBMS에 맞는 물리적 구조의 데이터로 변환

⑤ **구현** : 목표 DBMS의 DDL로 스키마 기술 후 데이터베이스 생성

12 개념 스키마는 개념적 설계 단계에서 수행하고 외부 스키마는 데이터베이스를 작성한 후 운영중에 작성합니다. 물리적 설계 시 고려해야 할 사항에 대해 정리하세요.

물리적 설계시 고려 사항

- 인덱스의 구조
- 레코드 크기
- 파일에 존재하는 레코드 개수
- 파일에 대한 트랜잭션의 갱신과 참조 성향
- 시스템 운용시 파일 크기의 변화 가능성
- 데이터의 무결성, 일관성, 효율성, 보안, 회복, 데이터베이스의 확장성

13 한 릴레이션에 나타나는 속성값은 더 이상 쪼갤 수 없는 원자값만을 저장합니다.

14 독자적으로 존재하지 못하고, 기본 테이블로부터 유도된 이름을 가진 가상 테이블을 뷰라고 합니다. 시스템 카탈로그의 의미와 특징은 시험에 자주 출제됩니다. 보기에 제시된 의미를 포함해 특징까지 모두 숙지해야 합니다.

카탈로그의 특징

- 카탈로그 자체도 시스템 테이블로 구성되어 있어 일반 사용자도 SQL을 이용하여 내용을 검색해 볼 수 있다.
- INSERT, DELETE, UPDATE문으로 카탈로그를 갱신하는 것은 허용되지 않는다.
- 카탈로그는 DBMS가 스스로 생성하고, 유지한다.
- **카탈로그의 갱신** : 사용자가 SQL문을 실행시켜 기본 테이블, 뷰, 인덱스 등에 변화를 주면 시스템이 자동으로 갱신한다.

15 뷰는 삽입, 삭제, 갱신 연산에 제약이 따릅니다. 뷰의 의미, 특징, 장점과 단점 모두 중요합니다. 보기의 내용들과 더불어 뷰의 장점과 단점을 정리하세요.

장점

- 논리적 데이터 독립성을 제공한다.
- 동일 데이터에 대해 동시에 여러 사용자의 상이한 응용이나 요구를 지원해 준다.
- 사용자의 데이터 관리를 간단하게 해준다.
- 접근 제어를 통한 자동 보안이 제공된다.

단점

- 독립적인 인덱스를 가질 수 없다.
- ALTER VIEW문을 사용할 수 없다. 즉 뷰의 정의를 변경할 수 없다.
- 뷰로 구성된 내용에 대한 삽입, 삭제, 갱신 연산에 제약이 따른다.

16 도메인 무결성은 튜플이 아니라 특정 속성의 값이, 그 속성이 정의된 도메인에 속한 값이어야 한다는 규정입니다. 보기를 통해 무결성을 다시한번 정리하고, 무결성 규정은 가볍게 한번 읽어보고 넘어가세요.

- 검증 프로그램이 무결성을 검증하기 위해 무결성 규정을 사용한다.
 - **규정 이름** : 무결성 규정을 참조할 때 사용하는 식별자
 - **트리거(Trigger) 조건** : 트랜잭션의 접근 유형 및 데이터, 검사할 시기 명시
 - **프레디캣(제약 조건)** : 무결성을 위한 검사 조건
 - **위반 조치** : 검사 결과 무결성 위반이 발견되었을 때 처리할 조치

17 데크(Deque)는 Double Ended Queue의 약자로 삽입과 삭제가 리스트의 양쪽 끝에서 모두 발생할 수 있는 선형 구조입니다. 삽입과 삭제가 리스트의 양쪽 끝에서 모두 발생할 수 있으므로 2개의 포인터가 사용됩니다.

18 분산 데이터베이스 시스템은 소프트웨어 개발 비용이 증가한다는 단점이 있습니다. 분산 데이터베이스의 정의와 목표를 학습한 후 장점과 단점을 읽어보면 어렵지 않게 이해할 수 있습니다. 분산 데이터베이스의 장점과 단점을 간단하게 정리하세요.

장점

- 지역 자치성이 높다.
- 자료의 공유성이 향상된다.
- 분산제어가 가능하다.
- 시스템 성능이 향상된다.
- 효율성과 융통성이 높다.
- 신뢰성 및 가용성이 높다.
- 점증적 시스템 용량 확장이 용이하다.

단점

- DBMS가 수행할 기능이 복잡하다.
- 데이터베이스 설계가 어렵다.
- 소프트웨어 개발 비용이 증가한다.
- 처리 비용이 증가한다.
- 잠재적 오류가 증가한다.

19 SQL 중 DML에 속하지 않는 것은 CREATE입니다. CREATE는 DDL에 속합니다. DML의 4가지 명령에 대해서 알아두세요.

데이터 조작용어(DML)의 4가지 유형

- **SELECT** : 테이블에서 조건에 맞는 튜플을 검색
- **INSERT** : 테이블에 새로운 튜플을 삽입
- **DELETE** : 테이블에서 조건에 맞는 튜플을 삭제
- **UPDATE** : 테이블의 조건에 맞는 튜플의 내용을 변경

20 지문에 제시된 내용은 이진(Binary) 검색에 대한 내용입니다. 보기에 제시된 다른 검색 방법의 개념에 대해서도 정리하고 넘어가세요.

- **이진(Binary) 검색** : 전체 파일을 두 개의 서브파일로 분리해 가면서 Key 레코드를 검색하는 방식이다.
- **이진 트리 검색(Binary Tree Search)** : 파일을 이진 검색 트리로 구성하여 검색하는 방식
- **블록 검색(Block Search)** : 파일을 구성하는 레코드들을 여러 개의 Block으로 분할하여 Block 단위는 순서화시키고, Block 내의 자료는 순서화와 관계없이 저장시킨다. Index 부분을 두어, 각 Block마다 최대 레코드 키값을 가지는 레코드 번호를 저장시킨다.
- **선형 검색(Linear Search)** : 순서화되어 있지 않은 파일에서 순차적으로 검색하는 방식으로, 찾고자 하는 Key값을 첫 번째 레코드 Key값부터 차례로 비교하여 검색하는 방식 (=Sequential Search)

21 디스크에서 등각 속도(CAV) 방식을 사용할 경우 트랙의 위치에 따라 데이터 저장 밀도가 달라집니다.

22 JK 플립플롭을 그림과 같이 두 입력선 J와 K를 하나로 묶어서 (iN) 처리하면 T 플립플롭과 같이 입력된 값의 보수가 출력됩니다. JK 플립플롭의 특징에 대해 알아두세요.

- JK 플립플롭은 RS FF에서 S=R=1일 때 동작되지 않는 결점을 보완한 플립플롭으로 모든 플립플롭의 기능을 포함합니다.

■ JK 플립플롭의 특성표는 다음과 같습니다.

| J | K | Q _(n+1) |
|---|---|--------------------|
| 0 | 0 | Q _(n) |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | Q _{(n)'} |

23 DMA는 입·출력 장치가 직접 주기억장치를 접근(Access)하여 Data Block을 입·출력하는 방식으로, 입·출력 전송이 CPU의 레지스터를 경유하지 않고 수행되는 것으로 CPU의 상태 보존이 필요하지 않습니다. 사이클 스틸에 대해 한번 정리하고 넘어가세요.

사이클 스틸(Cycle Steal)

- 데이터 채널(DMA 제어기)과 CPU가 주기억장치를 동시에 Access할 때 우선순위를 데이터 채널에게 주는 방식이다.
- Cycle Steal은 한번에 한 데이터 워드를 전송하고 버스의 제어를 CPU에게 돌려준다.
- Cycle Steal을 이용하면 입·출력 자료의 전송을 빠르게 처리해 주는 장점이 있다.

24 캐시(Cache) 메모리에서 특정 내용을 찾는 방식 중 매핑 방식에 주로 사용되는 메모리는 연관 메모리(Associative Memory)입니다. 캐시(Cache) 메모리는 문제가 다양한 형태로 많이 출제됩니다. 개념과 특징을 꼭 숙지하세요.

캐시의 개념과 특징

- 캐시 메모리는 CPU의 속도와 메모리의 속도 차이를 줄이기 위해 사용하는 고속 Buffer Memory이다.
- 캐시는 주기억장치와 CPU 사이에 위치한다.
- 캐시 메모리는 메모리 계층 구조에서 가장 빠른 소자이며, 처리 속도가 거의 CPU의 속도와 비슷할 정도이다.
- 캐시를 사용하면 주기억장치를 접근(Access)하는 횟수가 줄어들므로 컴퓨터의 처리 속도가 향상된다.
- 캐시 주소표는 검색시간을 단축시키기 위해 주로 연관기억장치(CAM, Associative Memory)를 사용한다.
- 캐시의 크기는 보통 수십 KByte~수백 KByte이다.

25 주기억장치의 용량이 8192비트이고 1워드의 길이가 16Bit이므로 용량을 워드로 환산하면 512워드가 됩니다. 즉 512×16=2⁹×16으로 표현할 수 있으므로 주소를 지정하는 PC와 AR의 크기는 0~511를 지정할 수 있는 9Bit가 필요하고, 워드의 길이가 16Bit이므로 내용을 담는 DR은 16Bit 입니다.

26 Exclusive-OR 즉 XOR 게이트를 논리식으로 풀어쓰면 다음과 같습니다.

$$Y = A \oplus B \qquad Y = A'B + AB'$$

$$Y = (A + B)(A' + B')$$

27 출력을 1로 만드는 입력에 대해서만 가르노도에 표현하여 간소화하면 다음과 같습니다.

| | | | | |
|---|----|----|----|----|
| | BC | | | |
| A | 00 | 01 | 11 | 10 |
| 0 | 1 | | | 1 |
| 1 | 1 | | | 1 |

- ① A와 B는 0, 1 모두 묶음에 포함되므로 의미가 없다.
- ② C는 0만 해당되므로 묶음 부분은 C'이다.

28 컴퓨터 내부에서 시스템 순간 순간의 상태를 나타내는 것은 PSW(Program Status Word)입니다.

- **인터럽트(Interrupt)** : 프로그램을 실행하는 도중에 예기치 않은 상황이 발생할 경우, 현재 실행 중인 작업을 즉시

중단하고 발생한 상황을 우선 처리한 후 실행 중이던 작업으로 복귀하여 계속 처리하는 것을 말한다. 일명 ‘끼어들기’라고도 한다.

- **메모리 주소 레지스터(MAR)** : 기억장치를 출입하는 데이터의 번지를 기억하는 레지스터

29 주소 지정 방식(Addressing Mode) 중에서 프로그램 카운터 값에 명령어의 주소 부분을 더해서 실제 주소를 구하는 방식은 상대 번지 방식입니다. 계산에 의한 주소 3가지는 꼭 암기하세요.

상대주소(Relative Mode)

- **유효주소** : 명령어의 주소 부분 + PC(프로그램 카운터)
- 명령어 자신의 기억장소를 기준으로 하여 데이터의 위치를 지정하는 방식이다.

베이스 레지스터(Base Register Mode)

- **유효주소** : 명령어의 주소 부분 + Base Register
- 프로그램을 재배치(Relocation)할 때 이용한다.

인덱스 레지스터(Index Register Mode)

명령어의 주소 부분 + Index Register

30 컴퓨터 시스템이 작동되면 먼저 프로그램 카운터의 초기 주소 값이 결정되고 주소에 의하여 명령어가 기억장치로부터 읽혀지는 것을 인출(Fetch)이라고 합니다. 각 스테이트의 특징을 모두 암기하지는 못해도 다른 스테이트와 비교하여 구분할 수 있을 정도로는 알아두세요.

인출 단계(Fetch Cycle)

- Fetch Cycle은 명령어를 주기억장치에서 중앙처리장치의 명령 레지스터로 가져와 해독하는 단계이다.
- 읽어와 해석된 명령어가 1 Cycle 명령이면 이를 수행한 후 다시 Fetch Cycle로 변천한다.

간접 단계(Indirect Cycle)

- Fetch 단계에서 해석된 명령의 주소부가 간접주소인 경우 수행된다.
- 이 사이클에서는 Fetch 단계에서 해석한 주소를 읽어온 후 그 주소가 간접주소이면 유효주소를 계산하기 위해 다시 Indirect 단계를 수행한다.

실행 단계(Execute Cycle)

- Fetch 단계에서 인출하여 해석한 명령을 실행하는 단계이다.
- Execute 단계에서는 플래그 레지스터의 상태 변화를 검사하여 Interrupt 단계로 변천할 것인지를 판단한다.
- Execute 단계에서는 Interrupt 요청신호를 나타내는 플래그 레지스터의 변화가 없으면 Fetch 단계로 변천한다.

인터럽트 단계(Interrupt Cycle)

- 인터럽트 발생시 복귀주소(PC)를 저장시키고, 제어 순서를 인터럽트 처리 프로그램의 첫 번째 명령어로 옮기는 단계이다.
- 인터럽트 단계를 마친 후에는 항상 Fetch 단계로 변천한다.

31 OP-Code가 4비트면 연산자의 종류는 $2^4=16$ 개가 생성될 수 있습니다.

32 0-주소 인스트럭션과 관계 있는 것은 스택(Stack)입니다. 각 명령어 형식에서 자료의 저장소로 사용하는 요소들은 꼭 기억하세요.

- **3주소 명령어** : GPR(범용 레지스터)
- **2주소 명령어** : GPR(범용 레지스터)
- **1주소 명령어** : ACC(누산기)
- **0주소 명령어** : 스택(Stack)

33 인터럽트 체제의 기본 요소에 속하지 않는 것은 인터럽트 스테이트입니다. 인터럽트는 다음과 같은 과정을 통해 수행됩니다. 인터럽트의 처리 순서를 묻는 문제도 많이 출제됩니다. 차분히 생각해 보면서 이해하면 어렵지 않습니다.

인터럽트 처리 순서

- 명령을 실행하던 중 인터럽트 요청신호가 발생한다.
- **프로그램 실행을 중단한다** : 현재 실행 중이던 명령어는 끝까지 실행한다.
- **현재의 프로그램 상태를 보존한다** : 프로그램 상태는 다음에 실행할 명령의 번지를 말하는 것으로서 PC가 가지고 있다. 즉 PC가 가지고 있는 값을 보관한다.
- **인터럽트 처리 루틴을 실행한다** : 인터럽트 처리 루틴을 실행하여 인터럽트를 요청한 장치를 식별한다.
- **인터럽트 서비스 취급 루틴을 실행한다** : 실질적인 인터럽트를 처리한다.
- **상태복구** : 인터럽트 요청신호가 발생했을 때 보관한 PC의 값을 다시 PC에 저장한다.
- **중단된 프로그램 실행 재개** : PC의 값을 이용하여 인터럽트 발생 이전에 수행 중이던 프로그램을 계속 실행한다.

34 명령어의 Operand 부분에 실제 데이터를 갖고 있는 방식은 즉시(Immediate) 주소지정방식입니다. 주소지정방식의 종류는 반드시 암기하고 각 주소지정방식의 특징은 이름을 보고 이해하세요.

즉치(즉시)적 주소지정방식(Immediate Mode)

- 즉치적 주소지정방식은 명령어 자체에 오퍼랜드(실제 데이터)를 내포하고 있는 방식이다.
- 별도의 기억장소를 액세스하지 않고 CPU에서 곧바로 자료를 이용할 수 있어서 실행 속도가 빠르다는 장점이 있다.
- 명령어의 길이에 영향을 받으므로 표현할 수 있는 데이터 값의 범위가 제한적이다.

직접 주소지정방식(Direct Mode)

- 직접 주소지정방식은 명령의 주소부(Operand)가 사용할 자료의 번지를 표현하고 있는 방식이다.
- 명령의 Operand부에 표현된 주소를 이용하여 실제 데이터가 기억된 기억장소에 직접 사상시킬 수 있다.
- 기억용량이 2ⁿ개의 Word인 메모리 시스템에서 주소를 표현하려면 nBit의 Operand부가 필요하다.

간접 주소지정방식(Indirect Mode)

- 간접 주소지정방식은 명령어에 나타낼 주소가 명령어 내에서 데이터를 지정하기 위해 할당된 비트(Operand 부의 비트)수로 나타낼 수 없을 때 사용하는 방식이다.
- 명령의 길이가 짧고 제한되어 있어도 긴 주소에 접근 가능한 방식이다.
- 명령어 내의 Operand부에 실제 데이터가 저장된 장소의 번지를 가진 기억장소의 번지를 표현함으로써, 최소한 주기억장치를 두 번 이상 접근하여 데이터가 있는 기억장소에 도달한다.

계산에 의한 주소지정방식

- 계산에 의한 주소지정방식은 Operand부와 CPU의 특정 레지스터의 값이 더해져서 유효주소를 계산하는 방식이다.
- 사용하는 레지스터의 종류에 따라 상대, 베이스, 인덱스 주소지정방식으로 구분한다.

35 문제에 제시된 그림은 디코더입니다. 보기에 제시된 각 회로들의 특징들만 간단히 정리하세요.

- **멀티플렉서(Multiplexer)** : 2ⁿ의 입력선 중 1개를 선택하여 그 선으로부터 입력되는 값을 1개의 출력선으로 출력시키는 회로
- **인코더(Encoder)** : 2ⁿ개의 입력선으로 입력된 값을 n개의 출력선으로 코드화해서 출력하는 회로

- **쉬프터(Shifter)** : 특정 비트의 자리를 이동하는 회로
- 36 하드웨어 신호에 의하여 특정 번지의 서브루틴을 수행하는 것은 벡터(Vectored) 인터럽트입니다.
 - **인터럽트 벡터** : 중앙처리장치는 인터럽트가 발생한 장치 번호를 받은 후에는 해당되는 인터럽트 서비스(취급) 루틴으로 분기하게 된다. 이때 기억장치 내의 특정한 곳에는 인터럽트 취급 루틴으로 분기하는 명령어들만을 기억하는 영역이 있는데, 이를 인터럽트 벡터라고 한다. 인터럽트 벡터에는 인터럽트가 발생했을 때 프로세서의 인터럽트 서비스가 특정의 장소로 점프하도록 점프할 분기번지가 기억되어 있다.
- 37 AND는 연산할 때 피연산자가 2개 필요한 이항연산자입니다. 이항연산자와 단항연산자를 구분하는 문제도 자주 출제됩니다.
 - **단항연산자(Unary Operator)** : NOT, Complement, Shift, Rotate, MOVE 등
 - **이항연산자(Binary Operator)** : 사칙연산, AND, OR, XOR, XNOR 등
- 38 ROM은 기억된 내용을 읽을 수만 있는 기억장치로서 일반적으로 쓰기는 불가능합니다. 그러므로 쓰기 신호는 필요하지 않습니다.
- 39 중앙처리장치가 주기억장치보다 더 빠르기 때문에 프로그램 실행 속도를 중앙처리장치의 속도에 근접하도록 하기 위해서 사용되는 기억장치는 캐시 기억 장치입니다. 24번 해설을 통해 캐시의 개념 및 특징을 다시한번 정리하세요.
 - **가상기억장치(Virtual Memory)** : 기억용량이 작은 주기억장치를 마치 큰 용량을 가진 것처럼 사용할 수 있도록 하는 운영체제의 메모리 운영 기법이다.
- 40 다음에 제시된 내용을 통해 로더(Loader)의 기능을 구분할 수 있을 정도로 파악하세요.

로더(Loader)의 기능

 - **할당(Allocation)** : 실행 프로그램을 실행시키기 위해 기억장치 내에 옮겨놓을 공간을 확보하는 기능
 - **연결(Linking)** : 부프로그램 호출시 그 부프로그램이 할당된 기억 장소의 시작 주소를 호출한 부분에 등록하여 연결하는 기능
 - **재배치(Relocation)** : 디스크 등의 보조기억장치에 저장된 프로그램이 사용하는 각 주소들을 할당된 기억 장소의 실제 주소로 배치시키는 기능
 - **적재>Loading** : 실행 프로그램을 할당된 기억 공간에 실제로 옮기는 기능
- 41 기억장치의 관리 전략 중 반입(Fetch) 전략은 프로그램/데이터를 주기억장치로 가져오는 시기를 결정하는 전략입니다. 기억장치의 관리 전략 중 배치 전략과 교체 전략의 개념에 대해서도 정리해두세요.
 - **배치(Placement) 전략** : 새로 반입되는 프로그램이나 데이터를 주기억장치의 어디에 위치시킬 것인지를 결정하는 전략이다.
 - **교체(Replacement) 전략** : 주기억장치의 모든 영역이 이미 사용 중인 상태에서 새로운 프로그램이나 데이터를 주기억장치에 배치하려고 할 때, 이미 사용되고 있는 영역 중에서 어느 영역을 교체하여 사용할 것인지를 결정하는 전략이다.
- 42 데커(Dekker) 알고리즘은 소프트웨어적으로 상호 배제 기

법을 구현하는 방법입니다. 임계 구역은 여러 개의 프로세스가 공유하는 데이터 및 자원에 대하여 어느 한 시점에서는 하나의 프로세스만 자원 또는 데이터를 사용하도록 지정된 공유 자원으로, 임계 영역은 특정 프로세스가 독점할 수 없기 때문에 임계 영역에 들어가는 것이 무한정 지연될 수는 없습니다. 데커 알고리즘과 임계 영역의 개념만 간단히 기억하고 넘어가세요.

- 43 하나의 프로세스가 작업 수행 과정에서 수행하는 기억 장치 접근에서 지나치게 페이지 폴트가 발생하여 프로세스 수행에 소요되는 시간보다 페이지 이동에 소요되는 시간이 더 커지는 현상을 스래싱(Thrashing)이라고 합니다. 스래싱을 막는 문제는 자주 출제되므로 스래싱의 의미를 정확히 알고 있어야 합니다. 나머지 용어도 함께 정리해두세요.
 - **워킹세트(Working Set)** : 프로세스가 일정 시간 동안 자주 참조하는 페이지들의 집합
 - **세마포어(Semaphore)** : '신호기', '깃발'을 뜻하며, 각 프로세스에 제어 신호를 전달하여 순서대로 작업을 수행하도록 하는 기법
 - **스왑핑(Swapping) 기법** : 하나의 프로그램 전체를 주기억장치에 할당하여 사용하다 필요에 따라 다른 프로그램과 교체하는 기법
- 44 운영체제 형태 중 시대적으로 가장 먼저 생겨난 것은 일괄처리 시스템입니다. 운영체제 형태의 발전 순서를 묻는 문제가 종종 출제됩니다. 발전 순서 정도는 기억해두세요.

운영체제 운용 기법의 발달 과정

 - **1세대** : 일괄 처리 시스템
 - **2세대** : 다중 프로그래밍 시스템, 다중 처리 시스템, 시분할 시스템, 실시간 처리 시스템
 - **3세대** : 다중 모드
 - **4세대** : 분산 처리 시스템
- 45 SCAN과 같이 진행 방향상의 요청을 서비스하지만, 진행중에 새로이 추가된 요청은 서비스하지 않고 다음 진행시에 서비스하는 디스크 스케줄링 기법은 N-step SCAN 스케줄링입니다. 스케줄링 기법에서는 각 스케줄링을 사용할 때의 이동 순서와 총 거리를 묻는 문제, 개념을 구분하는 문제가 출제됩니다. 각 스케줄링의 개념을 구분하여 알아두세요.
 - **C-SCAN 스케줄링** : 항상 바깥쪽에서 안쪽으로 움직이면서 가장 짧은 탐색 거리를 갖는 요청을 서비스하는 기법으로, 헤드는 트랙의 바깥쪽에서 안쪽으로 한 방향으로만 움직이며 서비스하여 끝까지 이동한 후, 안쪽에 더 이상의 요청이 없으면 헤드는 가장 바깥쪽의 끝으로 이동한 후 다시 안쪽으로 이동하면서 요청을 서비스한다.
 - **SSTF 스케줄링** : SSTF는 탐색 거리(Seek Distance)가 가장 짧은 트랙에 대한 요청을 먼저 서비스하는 기법
 - **FCFS 스케줄링** : 각 페이지가 주기억장치에 적재될 때마다 그때의 시간을 기억시켜 가장 먼저 들어와서 가장 오래 있었던 페이지를 교체하는 기법
- 46 워킹세트(Working Set)는 프로세스가 일정 시간(t-w 시간부터 t까지) 동안 참조하는 페이지들의 집합을 의미하는 것으로, 참조된 페이지가 {2, 3, 5, 6, 7}이므로 워킹 셋은 {2, 3, 5, 6, 7}이 됩니다.
- 47 UNIX 운영체제는 대부분의 코드가 C 언어로 기술되어 있습니다. UNIX 운영체제의 특징을 묻는 문제가 자주 출제됩니다. UNIX 운영체제의 특징을 꼭 알아두세요.

UNIX 운영체제의 특징

- 시분할 시스템(Time Sharing System)을 위해 설계된 대 화식 운영체제로, 소스가 공개된 개방형 시스템(Open System)이다.
 - 대부분 C 언어로 작성되어 있어 이식성이 높으며 장치, 프로세스 간의 호환성이 높다.
 - 크기가 작고 이해하기가 쉽다.
 - Multi-User, Multi-Tasking을 지원한다.
 - 많은 네트워킹 기능을 제공하므로 통신망(Network) 관리 용 운영체제로 적합하다.
 - 트리 구조의 파일 시스템을 갖는다.
 - 전문적인 프로그램 개발에 용이하다.
 - 다양한 유틸리티 프로그램들이 존재한다.
- 48 분산 처리 시스템의 설계 이유에는 자원 공유, 연산 속도 향상, 신뢰도 향상, 컴퓨터 통신이 있습니다. 분산 시스템을 설계하는 4가지 이유를 숙지하세요. 또한 분산 시스템의 설계 목적이 나 장점을 묻는 문제에 '보안의 향상'이 보기로 제시되어 있다면 이것이 답이 됩니다. 분산 시스템은 지리적으로 떨어진 시스템을 연결하여 사용하는 것이므로 보안에 문제가 발생할 수 있습니다.
- 49 스레드(Thread)는 프로세스 내에서의 작업 단위로서 시스템의 여러 자원을 할당받아 실행하는 프로그램의 단위로, 동일 프로세스 환경에서 서로 독립적인 다중 수행이 가능합니다. 문제를 통해 스레드의 특징에 대해 정리하세요.
- 50 직접 파일은 파일을 구성하는 레코드를 임의의 물리적 저장 공간에 기록하는 것으로, 주로 임의 접근이 가능한 자기 디스크나 자기 드럼을 사용합니다. 자기 테이프 장치를 사용하는 것은 순차 파일입니다. 문제를 통해 직접 파일의 특징을 정리하고, 순차 파일의 특징도 함께 기억하세요.
- 순차 파일**
- 레코드를 논리적인 처리 순서에 따라 연속된 물리적 저장 공간에 기록하는 것이다.
 - 파일의 레코드들이 순차적으로 기록되어 관독할 때도 순차적으로 접근하기 때문에 순차 접근 방식(SAM, Sequential Access Method)이라고도 한다.
 - 급여 업무처럼 전체 자료를 처리 대상으로 일괄 처리하는 업무에 사용된다.
 - 순차 접근이 가능한 자기 테이프를 모형화한 구조이다.
- 51 파일 손상을 막기 위한 파일 보호 기법에는 파일 명명(File Naming), 접근 제어(Access Control), 암호화(Password/Cryptography)가 있습니다. 파일 보호 기법의 종류 뿐만 아니라 개념까지도 알아두세요.
- **파일의 명명(Naming)** : 접근하고자 하는 파일 이름을 모르는 사용자를 접근 대상에서 제외시키는 기법
 - **비밀번호(Password, 암호)** : 각 파일에 관독 암호와 기록 암호를 부여하여 암호를 아는 사용자에게만 접근을 허용하는 기법
 - **접근 제어(Access Control)** : 사용자에게 따라 공유 데이터에 접근할 수 있는 권한을 제한하는 방법, 즉 각 파일마다 접근 목록을 두어 접근 가능한 사용자와 동작을 기록한 후 이를 근거로 접근을 허용하는 기법
- 52 스푼링은 다중 프로그래밍 환경 하에서 용량이 크고 신속한 액세스가 가능한 디스크를 이용하여 각 사용자 프로그램이 입·출력할 데이터를 직접 I/O 장치로 보내지 않고 디스크에 모았다가 나중에 한꺼번에 입·출력함으로써 입·출력 장치의 공유 및 상대적으로 느린 입·출력 장치의 처리 속도를 보완하는 기법입니다. 스푼링은 입·출력할 데이터를 디스크에 모은다는 것을 반드시 기억해두세요. 스푼링에 대한 특징은 각 보기를 통해 정리하세요.
- 53 제어 프로그램 중 문제에 설명된 프로그램은 데이터 관리 프로그램(Data Management Program)입니다. 운영체제는 제어 프로그램과 처리 프로그램으로 구분할 수 있습니다. 제어 프로그램은 다시 감시 프로그램과 작업 제어 프로그램, 자료 관리 프로그램으로 구분할 수 있습니다. 제어 프로그램의 종류별 특징을 읽어보고, 정리하세요.
- **감시 프로그램(Supervisor Program)** : 제어 프로그램 중 가장 중요한 역할을 담당하는 것으로, 각종 프로그램의 실행과 시스템 전체의 작동 상태를 감시·감독하는 프로그램
 - **작업 제어 프로그램(Job Control Program)** : 어떤 업무를 처리하고 다른 업무로의 이행을 자동으로 수행하기 위한 준비 및 그 처리에 대한 완료를 담당하는 프로그램으로, 작업의 연속 처리를 위한 스케줄 및 시스템 자원 할당 등을 담당한다.
 - **자료 관리 프로그램(Data Management Program)** : 주 기억장치와 보조기억장치 사이의 데이터 전송과 보조기억 장치의 자료 갱신 및 유지 보수 기능을 수행하는 프로그램
- 54 Flynn이 제안한 4가지 병렬처리 방식 중에서 이론적일 뿐 실질적인 처리방식으로 사용되지 않는 구조는 MISD 구조입니다. 플린(Flynn)의 병렬 컴퓨터의 분류는 2과목 전자 계산기 구조에 있는 내용입니다. 자주 출제되는 내용은 아니지만 각 분류의 특징 정도는 알고 넘어가세요.
- **SISD(Single Instruction stream Single Data stream)** : 현재의 보통 컴퓨터 구조로, 명령 하나가 자료 하나를 처리하는 구조이다.
 - **SIMD(Single Instruction stream Multi Data stream)** : 한 개의 명령으로 여러 Data를 동시에 처리하는 구조로, 다수의 처리기가 한 개의 제어장치에 의해 제어된다. 즉 모든 처리기는 제어장치로부터 같은 명령을 수행하도록 제어하지만 처리기는 각각 다른 자료를 사용한다.
 - **MISD(Multi Instruction stream Single Data stream)** : 다수의 처리기에 의해 각각의 명령들이 하나의 Data를 처리하는 구조로, 실제로는 사용되지 않는 구조이다.
 - **MIMD(Multi Instruction stream Multi Data stream)** : 다수의 처리기가 각각 다른 명령 흐름과 자료 흐름을 가지고 여러 개의 자료를 처리하는 구조이다.
- 55 UNIX에서 프로세스를 복제하는 명령어는 fork입니다. 각 보기에 제시된 UNIX 명령어의 기능도 모두 알고 있어야 합니다.
- **getpid** : 자신의 프로세스 아이디를 얻는다.
 - **getppid** : 부모 프로세스 아이디를 얻는다.
 - **exec** : 새로운 프로세스를 수행한다.
- 56 교착상태 해결 방안으로 교착상태 발생 가능성을 인정하고 교착 상태가 발생하려고 할 때, 교착상태 가능성을 피해가는 방법을 회피(Avoidance)라고 합니다. 대표적인 교착상태 회피 기법은 '은행원 알고리즘'이라는 것도 알고 있어야 합니다. 나머지 보기에서 제시하고 있는 기법의 특징도 함께 정리해두세요.
- **예방 기법(Prevention)** : 교착 상태가 발생하지 않도록 사전에 시스템을 제어하는 방법으로, 교착 상태 발생의 4가지 조건 중에서 어느 하나를 제거(부정)함으로써 수행된다. 자원의 낭비가 가장 심한 기법이다.

- **발견 기법(Detection)** : 시스템에 교착 상태가 발생했는지 점검하여 교착 상태에 있는 프로세스와 자원을 발견하는 것을 의미한다.
- **회복 기법(Recovery)** : 교착 상태를 일으킨 프로세스를 종료하거나 교착 상태의 프로세스에 할당된 자원을 선점하여 프로세스나 자원을 회복하는 것을 의미한다.

57 4개의 페이지를 수용할 수 있는 주기억장치이므로 다음 그림과 같이 표현할 수 있습니다. 참조 페이지가 페이지에 없을 경우는 페이지 결함(부재)이 발생합니다. 또한 LRU 기법은 최근에 가장 오랫동안 사용하지 않은 페이지를 교체하는 기법이므로 참조 페이지 5를 참조할 때에는 3을 제거한 후 5를 가져오게 됩니다. 그러므로 총 페이지 결함 발생수는 5번입니다.

| | | | | | | | | | |
|--------|---|---|---|---|---|---|---|---|---|
| 참조 페이지 | 1 | 2 | 3 | 1 | 2 | 4 | 1 | 2 | 5 |
| 페이지 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | | | 3 | 3 | 3 | 3 | 3 | 3 | 5 |
| 부재 발생 | ● | ● | ● | | | ● | | | ● |

58 운영체제를 자원 관리자(Resource Manager)라는 관점으로 보았을 때, 자원들을 관리하는 과정을 순서대로 옳게 나열하면 '어떤 프로세스에게 언제, 어떤 자원을 할당할 것인가를 결정하는 분배 정책 수립 과정 → 시스템 내 모든 자원들의 상태를 파악하는 과정 → 자원을 배당하고 운영함으로써 수립된 정책을 수행하는 과정 → 프로세스에 배당된 자원을 회수하는 과정' 순입니다. 지문의 내용을 잘 읽어보면 순서를 쉽게 찾을 수 있습니다.

59 프로세스 관리는 커널이 수행하는 역할입니다. 커널과 셸의 기능을 구분하는 문제가 자주 출제됩니다. 커널과 셸의 역할을 반드시 구분해줘야 합니다.

커널(Kernel)

- UNIX의 가장 핵심적인 부분이다.
- 컴퓨터가 부팅될 때 주기억장치에 적재된 후 상주하면서 실행된다.
- 하드웨어를 보호하고, 프로그램과 하드웨어 간의 인터페이스 역할을 담당한다.
- 프로세스(CPU 스케줄링) 관리, 기억장치 관리, 파일 관리, 입·출력 관리, 프로세스간 통신, 데이터 전송 및 변환 등 여러 가지 기능을 수행한다.

셸(Shell)

- 사용자의 명령어를 인식하여 프로그램을 호출하고 명령을 수행하는 명령어 해석기이다.
- 시스템과 사용자 간의 인터페이스를 담당한다.
- DOS의 COMMAND.COM과 같은 기능을 수행한다.
- 주기억장치에 상주하지 않고, 명령어가 포함된 파일 형태로 존재하며 보조기억장치에서 교체 처리가 가능하다.
- 공용 Shell(Bourne Shell, C Shell, Korn Shell)이나 사용자 자신이 만든 Shell을 사용할 수 있다.

60 NUR은 최근에 사용하지 않은 페이지를 교체하는 기법으로, 최근의 사용 여부를 확인하기 위해서 각 페이지마다 2개의 비트, 즉 참조 비트(Reference Bit)와 변형 비트(Modified Bit, Dirty Bit)가 사용됩니다. 참조 비트와 변형 비트에 따른 교체 순서는 다음과 같습니다. NUR은 2개의 비트를 사용한다는 것과 참조 비트와 변형 비트에 따른 교체 순서를 정확하게

알아두세요.

| 참조 비트 | 변형 비트 | 교체 순서 |
|-------|-------|-------|
| 0 | 0 | 1 |
| 0 | 1 | 2 |
| 1 | 0 | 3 |
| 1 | 1 | 4 |

61 소단위 명세서(Mini-Specification)는 세분화된 자료 흐름도에서 최하위 단계 버블(프로세스)의 처리 절차를 기술한 것으로 자연어가 아닌 구조적 언어, 의사 결정표(판단표) 등을 이용하여 기술합니다. 소단위 명세서의 개념과 특징은 문제의 보기를 통해 숙지하세요.

62 SOFTWARE Project의 비용 결정 요소와 가장 관련이 적은 하드웨어의 성능입니다. 하드웨어의 성능이 비용과 완전히 무관한 것은 아니지만 가장 관련이 적은 것을 고르는 문제이므로 하드웨어의 성능이 답이 됩니다.

63 고품질의 소프트웨어를 개발하기 위해 소프트웨어의 내부 구조를 기술하는 단계는 설계 단계입니다. 문제의 보기를 통해 요구 분석 단계에서의 특징을 정리하고, 설계 단계의 특징도 함께 알아두세요.

설계

- 요구사항 분석 단계의 산출물인 요구사항 분석 명세서의 기능이 실현되도록 알고리즘과 그 알고리즘에 의해 처리될 자료 구조를 문서화하는 것이다.
- 좋은 소프트웨어를 개발하기 위한 가장 핵심이 되는 기술로, 소프트웨어의 품질 평가를 위한 지침이 된다.

64 자료 흐름도(DFD, Data Flow Diagram)는 시간 흐름이 아니라 자료의 흐름 및 변환 과정과 기능을 도형 중심으로 기술하는 방법입니다. 자료 흐름을 중심으로 한다는 것을 기억해야겠죠! 자료 흐름도의 개념 뿐만 아니라 자료 흐름도에서 사용되는 4가지 구성 요소의 기호와 의미도 알고 있어야 합니다.

자료 흐름도의 구성 요소

| | |
|--------------------|--|
| 프로세스(Process) | <ul style="list-style-type: none"> ■ 자료를 변환시키는 시스템의 한 부분(처리 과정)을 나타내며 처리, 기능, 변환, 버블이라고도 한다. ■ 원이나 둥근 사각형으로 표시하고 그 안에 프로세스 이름을 기입한다. |
| 자료 흐름(Flow) | <ul style="list-style-type: none"> ■ 자료의 이동(흐름)을 나타낸다. ■ 화살표 위에 자료의 이름을 기입한다. |
| 자료 저장소(Data Store) | <ul style="list-style-type: none"> ■ 시스템에서의 자료 저장소(파일, 데이터베이스)를 나타낸다. ■ 도형 안에 자료 저장소 이름을 기입한다. |
| 단말(Terminator) | <ul style="list-style-type: none"> ■ 시스템과 교신하는 외부 개체로, 입력 데이터가 만들어지고 출력 데이터를 받는다(정보의 생산자와 소비자) ■ 도형 안에 이름을 기입한다. |

65 객체지향 개념에서 연관된 데이터와 함수를 함께 묶어 외부와 경계를 만들고 필요한 인터페이스만을 밖으로 드러내는 과정을 캡슐화라고 합니다. 캡슐화의 개념 뿐만 아니라 보기에 제시된 용어의 개념도 구분할 수 있어야 합니다.

- **메시지(Message)** : 객체들 간에 상호작용을 하는데 사용되는 수단으로, 객체의 메소드(동작, 연산)를 일으키는 외부의 요구 사항이다.

- **상속성** : 이미 정의된 상위 클래스(부모 클래스)의 모든 속성과 연산을 하위 클래스가 물려받는 것
 - **다형성** : 메시지에 의해 객체(클래스)가 연산을 수행하게 될 때 하나의 메시지에 대해 각 객체(클래스)가 가지고 있는 고유한 방법으로 응답할 수 있는 능력을 의미함
- 66 소프트웨어 공학의 공학(Engineering)이 가지는 의미와 어울리지 않는 것은 예술성입니다. 이 문제는 점수를 얻게 하기 위한 문제입니다. 이런 문제는 틀리면 안되겠죠! 다양한 소프트웨어 공학의 개념을 한번 읽어보고 넘어가세요.
- 소프트웨어 공학의 개념**
- IEEE의 소프트웨어 공학 표준 용어사전 : 소프트웨어의 개발, 운용, 유지보수, 폐기 처분에 대한 체계적인 접근 방안이다.
 - Fairley : 지정된 비용과 기간 내에 소프트웨어를 체계적으로 생산하고 유지보수 하는 데 관련된 기술적이고 관리적인 원리이다.
 - Boehm : 과학적인 지식을 소프트웨어 설계와 제작에 응용하는 것이며 이를 개발, 운용, 유지보수하는 데 필요한 문서 작성 과정이다.
 - 소프트웨어 공학은 제품을 단지 생산하는 것이 아니라 가장 경제적인 방법으로 양질의 제품을 생산하는 것이다.
 - 소프트웨어 공학은 안정적이며 효율적으로 작동하는 소프트웨어를 생산하고, 유지보수 활동을 체계적이고 경제적으로 수행하기 위해 계층화 기술을 사용한다.
- 67 현재 프로그램으로부터 데이터, 아키텍처, 그리고 절차에 관한 분석 및 설계 정보를 추출하는 과정을 역공학이라고 합니다. 각 용어의 개념도 함께 정리하세요.
- **재공학(Re-Engineering)** : 새로운 요구에 맞도록 기존 시스템을 이용하여 보다 나은 시스템을 구축하고, 새로운 기능을 추가하여 소프트웨어 성능을 향상시키는 것
 - **재사용(Reuse)** : 이미 개발된 인정받은 소프트웨어의 전체 혹은 일부분을 다른 소프트웨어 개발이나 유지에 사용하는 것
- 68 객체 지향의 기본 개념 중 객체가 메시지를 받아 실행해야 할 객체의 구체적인 연산을 정의한 것은 메소드입니다. 보기에 제시된 객체 지향의 기본 개념은 시험에 자주 출제되는 내용입니다. 각 개념을 구분할 수 있도록 알아두세요.
- **추상화(Abstraction, 개념화)** : 문제의 세부 사항을 먼저 설계하기보다는 전체적이고 포괄적인 개념을 설계한 후 차례로 세분화하여 구체화시켜 나가는 설계 방법
 - **상속성(Inheritance)** : 이미 정의된 상위 클래스(부모 클래스)의 모든 속성과 연산을 하위 클래스가 물려받는 것
 - **캡슐화(Encapsulation)** : 데이터(속성)와 데이터를 처리하는 함수를 하나로 묶는 것
- 69 프로토타입 모형(Prototype Model, 원형 모형)은 사용자의 요구사항을 정확히 파악하기 위해 실제 개발될 소프트웨어에 대한 건본(시제품)을 만들어 최종 결과물을 예측하는 모형으로, 이 모형의 가장 큰 장점은 요구사항을 충실히 반영할 수 있고, 요구사항의 변경이 용이하다는 것입니다. 프로토타입 모형과 폭포수 모형에 대한 문제가 자주 출제됩니다. 폭포수 모형의 특징도 숙지해두세요.
- 폭포수 모형(Waterfall Model)**
- 폭포에서 한번 떨어진 물은 거슬러 올라갈 수 없듯이 소프트웨어 개발도 각 단계를 확실히 매듭짓고 그 결과를 철저히 검토하여 승인 과정을 거친 후에 다음 단계를 진행하며 이전 단계로 넘어갈 수 없는 방식이다.
 - 폭포수 모형은 소프트웨어 공학에서 가장 오래되고 가장 폭넓게 사용된 전통적인 소프트웨어 생명 주기 모형으로, 고전적 생명 주기 모형이라고도 한다.
 - 소프트웨어 개발 과정의 앞 단계가 끝나야만 다음 단계로 넘어갈 수 있는 선형 순차적 모형이다.
 - 제품의 일부가 될 매뉴얼을 작성해야 한다.
 - 다음 단계를 수행하기 위해 각 단계가 끝난 후에는 결과물이 명확하게 산출되어야 한다.
 - 두 개 이상의 과정이 병행하여 수행되지 않는다.
- 70 모듈의 독립성은 결합도(Coupling)와 응집도(Cohension)에 의해 측정되며, 독립성을 높이려면 모듈의 결합도를 약하게 하고 응집도를 강하게 해야 합니다. 결합도 중 가장 약한 것은 자료 결합도이고, 응집도 중 가장 강한 것은 기능적 응집도입니다. 나머지 보기가 잘못된 이유도 알아두세요.
- 가. 모듈의 독립성을 높여주기 위해서는 각 모듈간의 관련성을 최소로 하며, 이 경우에 결합도가 최소가 된다.
나. 모듈간의 관련성을 최대로 하면 모듈의 독립성은 저하되며, 이 경우에 모듈의 결합도(Coupling)는 최대가 된다.
다. 복잡성을 감소시키는 수단으로 독립성의 개념이 많이 적용되고 있으며, 모듈의 독립성 척도로서는 결합도와 응집도가 모두 사용된다.
- 71 CASE(Computer Aided Software Engineering)에 대한 설명으로 옳은 것은 가 번입니다. 나머지 보기가 잘못된 이유를 꼭 찾을 수 있어야 합니다.
- 나. 도식목차, 총괄 다이어그램, 상세 다이어그램으로 구분되는 것은 HIPO입니다.
다. 상위(Upper) CASE에서는 문제를 기술(Description)하고 계획하며 요구 분석과 설계 단계를 지원하는 한다. 코드를 작성하고 테스트하며, 문서화하는 작업을 지원하는 것은 하위(Lower) CASE이다.
라. CASE는 소프트웨어 개발 과정에서 사용되는 요구 분석, 설계, 구현, 검사 및 디버깅 과정 전체 또는 일부를 컴퓨터와 전용 소프트웨어 도구를 사용하여 자동화하는 것이므로 생산성과 신뢰성을 향상시킨다.
- 72 간트(Gantt) 차트는 프로젝트의 각 작업들이 언제 시작하고 종료되는지에 대한 작업 일정을 막대 도표를 이용하여 표시하는 프로젝트 일정표입니다. 작업들간의 상호 관련성, 결정 경로를 표시하는 것은 PERT/CPM 네트워크입니다. 가, 나, 다 번의 보기를 숙지하여 간트(Gantt) 차트의 기능을 확실히 이해하세요.
- 73 모듈안의 작동을 자세히 관찰할 수 있으며, 프로그램 원시 코드의 논리적인 구조를 커버(Cover)하도록 테스트 케이스를 설계하는 프로그램 테스트 방법은 화이트 박스 테스트입니다. 블랙 박스 테스트와 화이트 박스 테스트를 구분하는 문제가 출제됩니다. 블랙 박스 테스트의 특징도 함께 정리해두세요.
- 블랙 박스 테스트(Black Box Test)**
- 소프트웨어가 수행할 특정 기능을 알기 위해서 각 기능이 완전히 작동되는 것을 입증하는 검사로서, 기능 검사라고도 한다.
 - 부정확하거나 누락된 기능, 인터페이스 오류, 자료 구조나 외부 데이터베이스 접근에 따른 오류, 행위나 성능 오류, 초기화와 종료 오류 등을 발견하기 위해 사용되며 테스트 과정의 후반부에 적용된다.
 - 소프트웨어 산물의 각 기능별로 적절한 정보 영역(입-출력)을 정하여 적합한 입력에 대한 출력의 정확성을 점검한다.
 - 블랙 박스 테스트의 종류에는 동치 분할 검사, 경계값 분석,

원인-효과 그래프 검사, 오류 예측 검사, 비교 검사 등이 있다.

74 사용자의 요구 사항에 대한 불확실성 결정 요인에 해당하지 않는 것은 '프로토타이핑시에 소요되는 비용 문제'입니다. 사용자의 요구 사항의 변동이 왜 발생되느냐를 중심으로 생각하면 답을 찾는 데 어려움이 없습니다.

75 객체 모형(Object Model), 동적 모형(Dynamic Model), 기능 모형(Functional Model)의 3개 모형으로 구성되어 있는 객체 지향 분석 기법은 럼바우 분석 기법(Rumbaugh Method)입니다. 럼바우의 분석 기법은 자주 출제되는 내용입니다. 각 모형의 의미까지 모두 알고 있어야 합니다.

럼바우의 분석 기법

- 객체 모델링 : 시스템에서 요구되는 객체를 찾아내어 속성과 연관 식별 및 객체들 간의 관계를 규정하여 객체 다이어그램으로 표시하는 것
- 동적 모델링 : 상태 다이어그램(상태도)을 이용하여 시간의 흐름에 따른 객체들 사이의 제어 흐름, 상호 작용, 동작 순서 등의 동적인 행위를 표현하는 모델링
- 기능 모델링 : 자료 흐름도(DFD)를 이용하여 다수의 프로세스들 간의 자료 흐름을 중심으로 처리 과정을 표현한 모델링

76 소프트웨어 비용 중 유지보수 비용은 개발비용 보다 많습니다. 유지보수는 개발된 소프트웨어의 품질을 항상 최상의 상태로 유지하기 위한 것으로, 소프트웨어 개발 단계 중 가장 많은 노력과 비용이 투입되는 단계입니다.

77 소프트웨어 유지보수 작업의 목적에는 하자 보수, 환경 적응, 예방 조치, 완전화 보수가 있습니다. 소프트웨어 유지보수의 4가지 작업의 종류와 각각의 의미까지 기억하세요.

- 수정(Corrective) 보수 = 수리·교정·정정·하자 보수 : 시스템을 운영하면서 검사 단계에서 발견하지 못한 오류를 찾아 수정하는 활동
- 적응(Adaptive) 보수 = 환경 적응, 조정 보수 : 소프트웨어의 수명 기간 중에 발생하는 환경의 변화(하드웨어, 운영체제 등)를 기존의 소프트웨어에 반영하기 위하여 수행하는 활동
- 완전화(Perfective) 보수 = 기능 개선, 기능 보수 : 소프트웨어의 본래 기능에 새로운 기능을 추가하거나 성능을 개선하기 위해 소프트웨어를 확장시키는 활동으로, 유지보수 활동 중 가장 큰 업무 및 비용을 차지하는 활동이다.
- 예방(Preventive) 보수 : 미래에 유지보수를 용이하게 하거나 기능을 향상시키기 위해 소프트웨어를 변경하는 활동으로, 예방 유지보수를 소프트웨어 재공학이라고도 한다.

78 민주주의적(분산형) 팀 구성은 팀원 모두가 의사 결정에 참여하는 비이기적인 구성 방식으로, 팀 구성원 간의 의사교류를 활성화시키므로 팀원의 참여도와 만족도를 증대시킵니다. 가, 나, 라번의 내용은 중앙 집중형 팀 구성에 대한 설명입니다. 민주주의적(분산형) 팀 구성의 다른 특징에 대해서도 한번 읽어보고 넘어가세요.

분산형 팀 구성

- 팀원 모두가 의사 결정에 참여하는 비이기적인 구성 방식으로, 민주주의적 팀 구성이라고도 한다.
- 의사 결정을 민주주의적으로 하며 팀 구성원의 참여도와 작업 만족도를 높이고 이직률을 낮게 한다.
- 팀 구성원 각자가 서로의 일을 검토하고 다른 구성원이

- 일한 결과에 대하여 같은 그룹의 일원으로서 책임을 진다.
- 여러 사람의 의사를 교류하므로 복잡하고 이해되지 않는 문제가 많은 장기 프로젝트 개발에 적합하다.
- 링 모양의 구조를 가지며 이는 모든 구성원이 동등한 레벨에 있음을 보여준다.
- 팀 구성 방법 중 가장 많은 의사 소통 경로를 갖는 구조이다.

79 개발 기간(시간)은 '노력/투입 인원'입니다. 노력은 'LOC/1인당 월평균 생산 코드 라인 수'로 구하므로 25,000/500, 즉 50입니다. 그렇다면 개발 기간은 50/5이므로 10개월이 됩니다. 공식만 알고 대입해서 풀면 어렵지 않습니다.

80 문제가 요구하는 의도를 잘 파악해보세요. 소프트웨어 프로젝트 계획자가 프로젝트를 시작하기 전에 추정해야 할 항목에 대해 물어봅니다. '프로젝트 시작하기 전'이 포인트가 되겠죠! 유지보수 비용은 프로젝트가 완성된 후에 사용되는 비용이므로 문제의 답이 됩니다.

81 흐름 제어 방식에서 일반적으로 한번에 여러 개의 프레임 전송할 경우 효율적인 기법은 슬라이딩 윈도우 기법입니다. 흐름 제어 방식 중 프레임의 개수에 따른 종류를 확인하세요.

정지-대기(Stop-and-Wait)

- 수신측의 확인 신호(ACK)를 받은 후에 다음 패킷을 전송하는 방식
- 한 번에 하나의 패킷만을 전송할 수 있다.

슬라이딩 윈도우(Sliding Window)

- 수신측의 확인 신호를 받지 않더라도 미리 정해진 패킷의 수만큼 연속적으로 전송하는 방식으로, 한 번에 여러 개의 패킷을 전송할 수 있어 전송 효율이 좋다.
- 송신측은 수신측으로부터 확인 신호(Ack) 없이도 보낼 수 있는 패킷의 개수를 미리 약속받는데, 이 패킷의 개수가 윈도우 크기(Window Size)를 의미한다.
- 윈도우 크기(Window Size)는 상황에 따라 변한다. 즉, 수신측으로부터 이전에 송신한 패킷에 대한 긍정 수신 응답(ACK)이 전달된 경우 윈도우 크기는 증가하고, 수신측으로부터 이전에 송신한 패킷에 대한 부정 수신 응답(NAK)이 전달된 경우 윈도우 크기는 감소한다.

82 데이터 통신에서 전송 제어 절차는 "데이터 통신 회선의 접속 → 데이터 링크의 설정(확립) → 정보 메시지 전송 → 데이터 링크의 종결 → 데이터 통신 회선의 절단"입니다.

83 LAN은 확장성과 재배치가 쉬우며, 공유 매체를 사용하므로 경로 선택 없이 매체에 연결된 모든 장치로 데이터를 전송합니다. LAN의 정의와 특징에 대한 문제가 출제됩니다. 짧은 거리의 제한된 지역 내에 설치된 통신망이라는 것에 초점을 맞춰 이해하세요.

LAN의 특징

- 단일 기관의 소유, 제한된 지역 내의 통신이다.
- 광대역 전송 매체의 사용으로 고속 통신이 가능하다.
- 공유 매체를 사용하므로 경로 선택 없이 매체에 연결된 모든 장치로 데이터를 전송한다.
- 오류 발생률이 낮다.
- 네트워크에 포함된 자원을 공유한다.
- 네트워크의 확장이나 재배치가 쉽다.
- 전송 매체로 꼬임선, 동축 케이블, 광섬유 케이블 등을 사용한다.
- 망의 구성 형태에 따라서 성형, 버스형, 링형, 계층형으로 분류할 수 있다.

84 데이터 신호 속도를 계산하는 문제가 자주 출제됩니다. 확실

히 구분하여 기억하세요.

- 데이터 신호 속도(bps) = 변조 속도(baud) × 변조시 상태 변화 수
- 변조 속도(baud) = 데이터 신호 속도(bps) / 변조시 상태 변화 수
- 8위상 편이 변조 방식일 경우 변조시 상태 변화 수는 3비트 이므로 $2400 \times 3 = 7,200\text{bps}$

85 컴퓨팅, 교환, 디지털 전송 장치간의 구분이 없어지고, 음성, 데이터 및 이미지 전송에 동일한 디지털 기술이 적용된 통합 시스템은 ISDN입니다. ISDN의 개념과 더불어 ISDN에서 제공되는 서비스를 구분하여 알아두세요.

베어러 서비스(Bearer Service)

- ISDN 사용자와 망 간의 인터페이스에서 단말 장치가 전송하는 정보를 변형 없이 그대로 전달만 하는 서비스
- 회선 교환, 패킷 교환 등 하위 계층의 기능만을 제공

텔레 서비스(Teleservice)

- 베어러 서비스를 기본으로 통신망, 단말 장치 등에서 이용하는 고도의 기능을 부가하여 제공하는 서비스
- 실제로 단말 장치를 조작하고 통신하는 이용자 측에서 본 서비스
- 상위 계층의 전화, 팩스, 텔레텍스, 비디오텍스, 텔렉스, 원격 회의 등의 서비스와 하위 계층의 정보 전송, 액세스 서비스 등을 모두 제공

부가 서비스

베어러 서비스나 텔레 서비스에 발신 번호 표시, 수신자 부담, 통화 대기 등의 기능을 부가하여 서비스 이용률을 높이는 서비스

86 다중화기(Multiplexer) 중 변복조 기능도 포함하는 기기는 주파수 분할 다중화기입니다. 주파수 분할 다중화기의 특징을 묻는 문제가 출제되고, 특히 보호 대역(Guard Band)을 사용하는 이유를 묻는 문제가 자주 출제됩니다. 주파수 분할 다중화기의 특징을 확실히 알아두세요.

주파수 분할 다중화기(FDM, Frequency Division Multiplexer)

- 통신 회선의 주파수를 여러 개로 분할하여 여러 대의 단말기가 동시에 사용할 수 있도록 한 것이다.
- 전송 신호에 필요한 대역폭보다 통신 회선의 유효 대역폭이 큰 경우에 사용한다.
- 다중화기 자체에 변복조 기능이 내장되어 있어 모뎀을 설치할 필요가 없다.
- 시분할 다중화기에 비해 구조가 간단하고 가격이 저렴하다.
- 저속(1,200bps 이하)의 비동기식 전송, 멀티 포인트(Multi-Point) 방식에 적합하다.
- 아날로그 신호 전송에 적합하다.
- 대역폭을 나누어 사용하는 각 채널들 간의 상호 간섭을 방지하기 위한 보호 대역(Guard Band)이 필요하다.
- 보호 대역(Guard Band) 사용으로 인한 대역폭의 낭비가 초래된다.

87 VAN의 기능을 묻는 문제가 자주 출제됩니다. 특히 통신 처리 기능에 대해서는 자세히 알아두세요.

VAN의 계층 구조 및 기능

- **전송 기능(기본 통신 계층)** : 사용자가 단순히 정보를 전송할 수 있도록 물리적 회선을 제공하는 VAN의 가장 기본적인 기능이다.
- **교환 기능(네트워크 계층)** : 가입된 사용자들을 서로 연결시켜 사용자 간의 정보 전송이 가능하도록 제공하는 서비스이다.

스이다. 패킷 교환 방식을 이용한다.

- **통신 처리 기능(통신 처리 계층)** : 축적 교환 기능과 변환 기능을 이용하여 서로 다른 기종 간에 또는 다른 시간 대에 통신이 가능하도록 제공하는 서비스이다.
- **정보 처리 기능(정보 처리 계층)** : 온라인 실시간 처리, 원격 일괄 처리, 시분할 시스템 등을 이용하여 급여 관리, 판매 관리 데이터베이스 구축, 정보 검색, 소프트웨어 개발 등의 응용 소프트웨어를 처리하는 기능이다.

통신 처리 기능

- **축적 교환 기능**
 - **전자 사서함** : 상대방이 부재중이더라도 나중에 확인해볼 수 있도록 정보 메시지를 축적하고 검색 서비스를 제공하는 기능
 - **데이터 교환** : 데이터 축적 기능을 이용하여 송·수신자 사이의 데이터 교환을 수행하는 기능
 - **동보 통신** : 한 단말기에서 여러 단말기로 같은 내용을 동시에 전송하는 기능
 - **정시 수집** : 정해진 시간에 VAN이 송신자로부터 정보를 받는 기능
 - **정시 배달** : 정해진 시간에 VAN에 축적되어 있는 정보를 수신자에게 송신하는 기능
- **변환 기능**
 - **속도 변환** : 축적 기능을 이용하여 속도가 빠른 컴퓨터로부터 데이터를 받아들여 축적한 후 상대방 단말기의 속도에 맞추어 보내주는 기능
 - **프로토콜 변환** : 서로 다른 네트워크 간에 또는 서로 다른 기종 간에 통신이 가능하도록 통신 절차를 변환하는 기능
 - **코드 변환** : 송신측과 수신측이 사용하는 문자 코드 체계를 맞추는 기능
 - **데이터 형식(Format) 변환** : 서로 다른 데이터 표현 형식을 일치시키는 기능
 - **미디어 변환** : 텍스트, 음성, 영상 등을 상호 변환하는 기능

88 문제의 지문에 제시된 잡음은 충격성 잡음에 대한 설명입니다. 보기에 제시된 오류들에 대해서도 알아두세요.

백색 잡음(White Noise)

- 가우스 잡음(Gaussian Noise), 열 잡음(Thermal Noise)이라고도 한다.
- 전송 매체 내부에서 온도에 따라 전자의 운동량이 변화함으로써 생기는 잡음으로, 완벽하게 제거하기는 힘들다.

상호 변조(간섭) 잡음(Intermodulation Noise)

- 서로 다른 주파수들이 하나의 전송 매체를 공유할 때 주파수 간의 합(合)이나 차(差)로 인해 새로운 주파수가 생성되는 잡음
- 통신 시스템의 비선형성 때문에 생긴다.

누화 잡음 = 혼선(Cross Talk Noise)

- 인접한 전송 매체의 전자기적 상호 유도 작용에 의해 생기는 잡음
- 신호의 경로가 비정상적으로 결합된 경우 나타난다.

89 OSI 7계층 중 장치와 전송매체 간의 인터페이스 특성 규정, 전송 매체의 유형 규정, 전송로의 연결, 유지 및 해제를 담당하는 계층은 물리 계층입니다. OSI 7 계층 가운데 어떤 계층을 설명한 것인지를 구분하는 문제가 출제됩니다.

물리 계층(Physical Layer)

- 물리 계층은 전송에 필요한 두 장치 간의 실제 접속과 절단 등 기계적, 전기적, 기능적, 절차적 특성에 대한 규칙을 정의한다.

- 물리적 전송 매체와 전송 신호 방식을 정의하며, RS-232C, X.21 등의 표준이 있다.

데이터 링크 계층(Data Link Layer)

- 데이터 링크 계층은 두 개의 인접한 개방 시스템들 간에 신뢰성 있고 효율적인 정보 전송을 할 수 있도록 한다.
- 송신측과 수신측의 속도 차이를 해결하기 위한 흐름 제어 기능을 한다.
- 프레임의 시작과 끝을 구분하기 위한 프레임의 동기화 기능을 한다.
- 오류의 검출과 회복을 위한 오류 제어 기능을 한다.
- 프레임의 순서적 전송을 위한 순서 제어 기능을 한다.
- HDLC, LAPB, LLC, LAPD 등의 표준이 있다.

네트워크 계층(Network Layer, 망 계층)

- 네트워크 계층은 개방 시스템들 간의 네트워크 연결을 관리 하는 기능과 데이터의 교환 및 중계 기능을 한다.
- 네트워크 연결을 설정, 유지, 해제하는 기능을 한다.
- 경로 설정(Routing), 데이터 교환 및 중계, 트래픽 제어, 패킷 정보 전송을 수행한다.
- 관련 표준으로는 X.25, IP 등이 있다.

전송 계층(Transport Layer)

- 전송 계층은 논리적 안정과 균일한 데이터 전송 서비스를 제공함으로써 종단 시스템(End-to-End) 간에 투명한 데이터 전송을 가능하게 한다.
- OSI 7 계층 중 하위 3 계층과 상위 3 계층의 인터페이스(Interface)를 담당한다.
- 종단 시스템(End-to-End) 간의 전송 연결 설정, 데이터 전송, 연결 해제 기능을 한다.
- 주소 설정, 다중화, 오류 제어, 흐름 제어를 수행한다.
- TCP, UDP 등의 표준이 있다.

세션 계층(Session Layer)

- 세션 계층은 송수신측 간의 관련성을 유지하고 대화 제어를 담당하는 계층이다.
- 대화(회화) 구성 및 동기 제어, 데이터 교환 관리 기능을 한다.
- 송수신측 간의 대화(회화) 동기를 위해 전송하는 정보의 일정한 부분에 체크점을 두어 정보의 수신 상태를 체크하며, 이때의 체크점을 동기점이라고 한다.
- 동기점은 오류가 있는 데이터의 회복을 위해 사용하는 것으로, 종류에는 소동기점과 대동기점이 있다.

표현 계층(Presentation Layer)

- 표현 계층은 응용 계층으로부터 받은 데이터를 세션 계층에 보내기 전에 통신에 적당한 형태로 변환하고, 세션 계층에서 받은 데이터는 응용 계층에 맞게 변환하는 기능을 한다.
- 서로 다른 데이터 표현 형태를 갖는 시스템 간의 상호 접속을 위해 필요한 계층이다.
- 코드 변환, 데이터 암호화, 데이터 압축, 구문 검색, 정보 형식(포맷) 변환, 문맥 관리 기능을 한다.

응용 계층(Application Layer)

- 응용 계층은 사용자(응용 프로그램)가 OSI 환경에 접근할 수 있도록 서비스를 제공한다.
- 응용 프로세스 간의 정보 교환, 전자 사서함, 파일 전송 등의 서비스를 제공한다.

- 90 전송 매체상의 전송 프레임마다 해당 채널의 시간 슬롯이 고정적으로 할당되는 다중화 방식은 동기식 시분할 다중화기입니다. 보기에 제시된 나머지 기기들의 특징도 알아두세요.
- **주파수 분할 다중화** : 통신 회선의 주파수를 여러 개로 분할하여 여러 대의 단말 장치가 동시에 사용할 수 있도록 하는 다중화 기법이다.
 - **통계적 시분할 다중화** : 마이크로프로세서를 이용하여 접

속된 단말기 중 전송할 데이터가 있는 단말기에만 시간폭(Time Slot)을 제공한다.

- **코드 분할 다중화** : 주파수나 시간을 모두 공유하면서 각 데이터에 특별한 코드를 부여하는 방식

- 91 전송 제어 문자 중 부정적 응답에 해당하는 전송 제어 문자는 NAK(Negative Acknowledge)입니다. 전송 제어 문자의 기능은 꼭 암기하세요.

- **SYN(SYNchronous idle)** : 문자 동기
- **SOH(Start Of Heading)** : 헤딩의 시작
- **STX(Start of TeXt)** : 본문의 시작 및 헤딩의 종료
- **ETX(End of TeXt)** : 본문의 종료
- **ETB(End of Transmission Block)** : 블록의 종료
- **EOT(End Of Transmission)** : 전송 종료 및 데이터 링크의 해제
- **ENQ(ENquiry)** : 상대방에 데이터 링크 설정 및 응답을 요구
- **DLE(Data Link Escape)** : 전송 제어 문자 앞에 삽입하여 전송 제어 문자임을 알림
- **ACK(ACKnowledge)** : 수신된 메시지에 대한 긍정 응답
- **NAK(Negative Acknowledge)** : 수신된 메시지에 대한 부정 응답

- 92 OSI 계층의 네트워크 계층에 해당하는 X.25의 계층은 패킷 계층입니다. X.25의 개념 및 특징도 정리하세요.

X.25의 특징

- X.25는 ITU-T에서 제정(1976년 승인)한 국제 표준 프로토콜로, 우수한 호환성을 가진다.
- 강력한 오류 체크 기능으로 신뢰성이 높다.
- 한 회선에 장애가 발생하더라도 정상적인 경로를 선택하여 우회 전송이 가능하다.
- 디지털 전송을 기본으로 하므로 전송 품질이 우수하다.
- 가상 회선 방식을 이용하여 하나의 물리적 회선에 다수의 논리 채널을 할당하므로 효율성이 높다.
- 축적 교환 방식을 사용하므로, 전송을 위한 처리 지연이 발생할 수 있다.

- 93 컴퓨터를 이용한 정보통신 시스템에서 정확한 데이터를 주고받기 위해서는 컴퓨터 간의 미리 정해진 약속이 필요합니다. 이러한 약속을 프로토콜(Protocol)이라고 합니다.

- 94 통계적 TDM에서 다중화된 회선의 데이터 전송율과 접속장치들의 데이터 전송율의 합과의 일반적인 관계는 다중화된 회선의 데이터 전송율이 접속장치들의 데이터 전송율의 합보다 작습니다.

- 95 패킷 교환망에서 유통되는 패킷의 수를 적절히 조절해 통신망을 효율적으로 사용하고자 하는 제어 기법이 트래픽(Traffic) 제어 기법입니다. 트래픽 제어 기법에서는 특히 흐름 제어의 기능에 대해서 자세히 알아두세요.

트래픽 제어 기법

- 트래픽 제어(Traffic Control)는 네트워크의 보호, 성능 유지, 네트워크 자원의 효율적인 이용을 위해 전송되는 패킷의 흐름 또는 그 양을 조절하는 기능이다.
- 흐름 제어
 - 흐름 제어(Flow Control)란 네트워크 내의 원활한 흐름을 위해 송수신측 사이에 전송되는 패킷의 양이나 속도를 규제하는 기능이다.
 - 송신측과 수신측 간의 처리 속도 또는 버퍼 크기의 차이에 의해 생길 수 있는 수신측 버퍼의 오버플로우(Overflow)를 방지하기 위한 기능이다.

- 폭주(혼잡) 제어
흐름 제어(Flow Control)가 송-수신측 사이의 패킷 수를 제어하는 기능이라면, 폭주 제어(Congestion Control)는 네트워크 내의 패킷 수를 조절하여 네트워크의 오버플로우(Overflow)를 방지하는 기능을 한다. 즉 폭주 제어는 네트워크 내의 모든 단말 장치들의 패킷 수를 제어한다.
 - 교착 상태 방지
 - 교착 상태(Dead Lock)란 교환기 내에 패킷들을 축적하는 기억 공간이 꽉 차 있을 때 다음 패킷들이 기억 공간에 들어가기 위해 무한정 기다리는 현상을 말한다.
 - 패킷이 같은 목적지를 갖지 않도록 할당하고, 교착 상태 발생시에는 교착 상태에 있는 한 단말 장치를 선택하여 패킷 버퍼를 폐기한다.
- 96 FCS(Frame Check Sequence Field, 프레임 검사 순서 필드)는 프레임 내용에 대한 오류 검출을 위해 사용되는 부분으로, 일반적으로 CRC코드가 사용됩니다. 91번 해설을 통해 전송 제어 절차에 사용되는 문자들에 대해 다시 한번 정리하세요.
- 97 대역폭(Bandwidth)은 최고 주파수와 최저 주파수 사이의 간격을 의미합니다.
- 98 공중 통신 회선에 교환설비, 컴퓨터 및 단말기 등을 접속시켜 새로운 부가 기능을 제공하는 통신망은 부가가치 통신망(VAN)입니다. 87번 해설을 통해 부가가치 통신망의 계층 구조 및 기능에 대해서도 다시 한번 정리하세요.
- 99 네트워크 계층 구조의 하나인 트랜스포트(Transport) 계층에서 사용되는 프로토콜에는 TCP, UDP 등이 있습니다. 89번 해설을 통해 OSI 7 계층의 각 계층별 특징에 대해서 다시 한번 정리하세요.
- 100 데이터는 한쪽 방향으로만 흐르고 병목 현상이 드물지만, 두 노드 사이의 채널이 고장나면 전체 네트워크가 손상될 수 있는 단점을 갖는 토폴로지는 링형 토폴로지입니다. 보기에 제시된 각 토폴로지의 특징 정도는 알아두세요.
- **망형(Mesh)** : 모든 지점의 컴퓨터와 단말 장치를 서로 연결한 형태로, 노드의 연결성이 높다.
 - **성형(Star)** : 중앙에 중앙 컴퓨터가 있고, 이를 중심으로 단말 장치들이 연결되는 중앙 집중식의 네트워크 구성 형태
 - **계층형(Tree)** : 중앙 컴퓨터와 일정 지역의 단말 장치까지는 하나의 통신 회선으로 연결시키고, 이웃하는 단말 장치는 일정 지역 내에 설치된 중간 단말 장치로부터 다시 연결시키는 형태