

2018-국가직-컴퓨터일반-가형-해설

대방고시 전산직/계리직, 하이클래스 군무원 곽후근(gobarian@gmail.com)

해설에 대한 모든 권리는 곽후근(대방고시, 하이클래스)에 있습니다.

1. 유닉스 운영체제에 대한 설명으로 옳지 않은 것은?

- ① 계층적 파일시스템과 다중 사용자를 지원하는 운영체제이다.
- ② BSD 유닉스의 모든 코드는 어셈블리 언어로 작성되었다.
- ③ CPU 이용률을 높일 수 있는 다중 프로그래밍 기법을 사용한다.
- ④ 사용자 프로그램은 시스템 호출을 통해 커널 기능을 사용할 수 있다.

정답 체크 :

(2) BSD 유닉스의 모든 코드는 C 언어로 작성되었다. 그러나 초창기 유닉스의 언어는 어셈블리 언어이다.

오답 체크 :

- (1) 유닉스는 트리 형태의 파일시스템을 가지며, 여러명의 사용자를 동시에 지원한다.
- (3) 다중 프로그래밍 기법이란 여러 개의 프로그램들을 동시에 주기억 장치에 적재하여, 한 프로그램이 입출력 등의 작업을 할 때 중앙처리 장치를 쉬게 하지 않고 다른 프로그램을 처리하게 하여 전체적인 처리 속도를 향상시키는 방식이다.
- (4) 사용자(응용) 프로그램은 시스템 호출(system call)을 통해 커널 기능을 사용할 수 있다. 예를 들어, 사용자 프로그램과 커널 사이에 데이터를 주고 받을 때 시스템 호출을 사용할 수 있다.

2. 다음에서 설명하는 해킹 공격 방법은?

공격자는 사용자의 합법적 도메인을 탈취하거나 도메인 네임 시스템(DNS) 또는 프락시 서버의 주소를 변조하여, 사용자가 진짜 사이트로 오인하여 접속하도록 유도한 후 개인정보를 훔친다.

- ① 스니핑(Sniffing)
- ② 파밍(Pharming)
- ③ 트로이 목마(Trojan Horse)
- ④ 하이재킹(Hijacking)

정답 체크 :

(2) 파밍 : Phishing(개인 정보)과 farming(대규모 피해)의 합성어이다. DNS Spoofing과 같이 인터넷 주소창에 방문하고자 하는 사이트의 URL을 입력하였을 때 가짜 사이트(fake site)로 이동시키는 공격 기법이다.

오답 체크 :

- (1) 스니핑 : 패킷을 태핑(Tapping)이나 미러링(Mirroring)을 통해 도청하는 것을 의미한다. 도청만 수행하므로 소극적 공격에 해당한다.
- (3) 트로이 목마 : 사용자가 의도하지 않은 코드를 정상적인 프로그램에 삽입한 형태이다.
- (4) 하이재킹 : TCP는 클라이언트와 서버간 통신을 할 때 패킷의 연속성을 보장하기 위해 클라이언트와 서버는 각각 시퀀스 넘버를 사용한다. 이 시퀀스 넘버가 잘못되면 이를 바로 잡기 위한 작업을 하는데, 세션 하이재킹은 서버와 클라이언트에 각각 잘못된 시퀀스 넘버를 위조

해서 연결된 세션에 잠시 혼란을 준 뒤 자신이 끼어들어가는 방식이다.

3. 다음 SQL 명령어에서 DDL(Data Definition Language) 명령어만을 모두 고른 것은?

- ㄱ. ALTER
- ㄴ. DROP
- ㄷ. INSERT
- ㄹ. UPDATE

- ① ㄱ, ㄴ
- ② ㄴ, ㄷ
- ③ ㄴ, ㄹ
- ④ ㄷ, ㄹ

정답 체크 :

(1)

(ㄱ) ALTER : DDL로 데이터베이스 객체를 변경한다.

(ㄴ) DROP : DDL로 데이터베이스 객체를 삭제한다.

오답 체크 :

(2), (3), (4)

(ㄷ) INSERT : DML로 데이터베이스 객체에 데이터를 입력한다.

(ㄹ) UPDATE : DML로 데이터베이스 객체의 데이터를 수정한다.

Tip! : SQL의 DDL, DML, DCL을 테이블로 정리하면 다음과 같다.

DDL(정의어)	CREATE(데이터베이스 객체 생성), ALTER(객체 변경), DROP(객체 삭제)
DML(조작어)	SELECT(테이블이나 뷰의 내용을 읽고 선택), INSERT(데이터 입력), UPDATE(데이터 수정), DELETE(데이터 삭제)
DCL(제어어)	GRANT(객체에 권한을 부여), DENY(사용자에게 해당 권한 금지), REVOKE(객체의 권한을 취소)

4. 다음 수식에서 이진수 Y의 값은?(단, 수식의 모든 수는 8 비트 이진수이고 1의 보수로 표현된다)

$$11110100_{(2)} + Y = 11011111_{(2)}$$

- ① 11101001₍₂₎
- ② 11101010₍₂₎
- ③ 11101011₍₂₎
- ④ 11101100₍₂₎

정답 체크 :

(2)

11110100 → 00001011 // 해당 이진수는 십진수로 -11이다.

11011111 → 00100000 // 해당 이진수는 십진수로 -32이다.

-11 + Y = -32 // Y를 계산하면 -21이 나온다.

00001011 → 11101010 // -21을 1의 보수로 계산하면 보기의 (2)가 된다.

5. 다음 진리표를 만족하는 부울 함수로 옳은 것은? (단, ·은 AND, ⊕는 XOR, ⊙는 XNOR

연산을 의미한다)

입력			출력
A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

- ① $Y = A \cdot B \oplus C$
- ② $Y = A \oplus B \odot C$
- ③ $Y = A \oplus B \oplus C$
- ④ $Y = A \odot B \odot C$

정답 체크 :

(2)

문제를 해결하는 2가지 방법이 존재한다.

첫 번째 방법은 부울식을 이용해서 푸는 것이다.

$$Y = A'B'C' + A'BC + AB'C + ABC' = (A'B' + AB)C' + (A'B + AB')C = (A'B + AB')C' + (A'B + AB')C = (A'B + AB') \odot C = A \oplus B \odot C$$

두 번째 방법은 ABC = 000을 대입하는 것이다. 주어진 조건을 보면 해당 입력에 대해 출력이 1이 나와야 한다.

- (1) $AB \oplus C = 0 \oplus 0 = 0$
- (2) $A \oplus B \odot C = 0 \odot 0 = 1$
- (3) $A \oplus B \oplus C = 0 \oplus 0 = 0$
- (4) $A \odot B \odot C = 1 \odot 0 = 0$

Tip! : 해당 문제에서는 두 번째 방법이 빠른 것처럼 보이지만 이 경우는 운이 좋은 케이스이다. ABC = 000에서 답을 찾을 수 없는 경우 계속해서 대입을 해봐야하기 때문에 첫 번째 방법보다 시간이 더 걸릴 수도 있다.

6. 스레싱(Thrashing)에 대한 설명으로 옳지 않은 것은?

- ① 프로세스의 작업 집합(Working Set)이 새로운 작업 집합으로 전이 시 페이지 부재율이 높아질 수 있다.
- ② 작업 집합 기법과 페이지 부재 빈도(Page Fault Frequency) 기법은 한 프로세스를 중단(Suspend)시킴으로써 다른 프로세스들의 스레싱을 감소시킬 수 있다.
- ③ 각 프로세스에 설정된 작업 집합 크기와 페이지 프레임 수가 매우 큰 경우 다중 프로그래밍 정도(Degree of Multiprogramming)를 증가시킨다.
- ④ 페이지 부재 빈도 기법은 프로세스의 할당받은 현재 페이지 프레임 수가 설정한 페이지 부재율의 하한보다 낮아지면 보유한 프레임 수를 감소시킨다.

정답 체크 :

(3) 큰 크기의 작업 집합은 페이지 부재 비율을 감소시켜 다중 프로그래밍 정도를 증가시킬 수 있으나 작업 집합이 매우 큰 경우에는 지역성의 원리가 깨지게 되고 페이지 부재 비율이 증가되어 다중 프로그래밍 정도가 감소한다.

오답 체크 :

(1) 작업 집합 모델은 프로세스가 많이 참조하는 페이지 집합을 메모리 공간에 계속 상주시켜 빈번한 페이지 대치 현상을 줄이는 것이다(지역성의 원리). 만약 새로운 작업 집합으로 전이되면 많이 참조된 페이지가 아니라 새로운 페이지가 작업 집합 내에 존재하므로 페이지 부재율이 높아질 수 있다.

(2) 프로세스를 중단시키면 프레임을 회수할 수 있으므로 다른 프로세스들의 스레싱을 감소시킬 수 있다.

(4) 페이지 부재 빈도 기법은 아래 그림과 같이 현재 페이지 프레임 수가 설정한 페이지 부재율의 하한보다 낮아지면 프레임이 너무 많다는 의미이므로 보유한 프레임 수를 감소시킨다. 반대로 페이지 부재율이 높다는 것은 더 많은 프레임이 필요하다는 의미이므로 프레임 수를 증가시킨다.

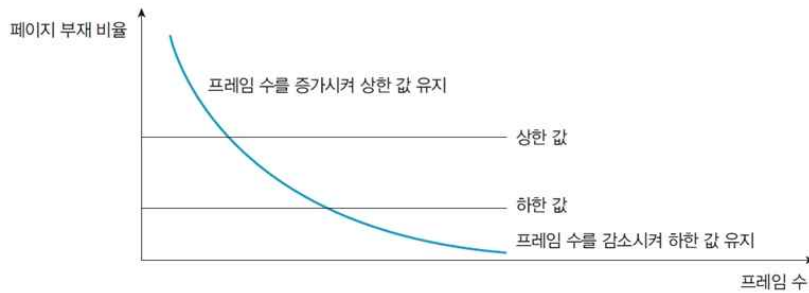


그림 8-41 페이지 부재 비율

Tip! : 스레싱이란 페이지 교환이 계속 일어나는 현상이다. 어떤 프로세스에 프레임이 충분하지 않다면 할당된 프레임을 최소 프레임 수까지 줄일 수 있다 하더라도 실제 사용하는 프레임 수만큼 갖지 못하면 빈번하게 페이지 부재가 발생 가능하다.

7. 인공신경망에 대한 설명으로 옳은 것만을 모두 고른 것은?

- ㄱ. 단층 퍼셉트론은 배타적 합(Exclusive-OR) 연산자를 학습할 수 있다.
- ㄴ. 다층 신경망은 입력 층, 출력 층, 하나 이상의 은닉층들로 구성된다.
- ㄷ. 뉴런 간 연결 가중치(Connection Weight)를 조정하여 학습한다.
- ㄹ. 생물학적 뉴런 망을 모델링 한 방식이다.

- ① ㄱ, ㄴ, ㄷ
- ② ㄱ, ㄴ, ㄹ
- ③ ㄱ, ㄷ, ㄹ
- ④ ㄴ, ㄷ, ㄹ

정답 체크 :

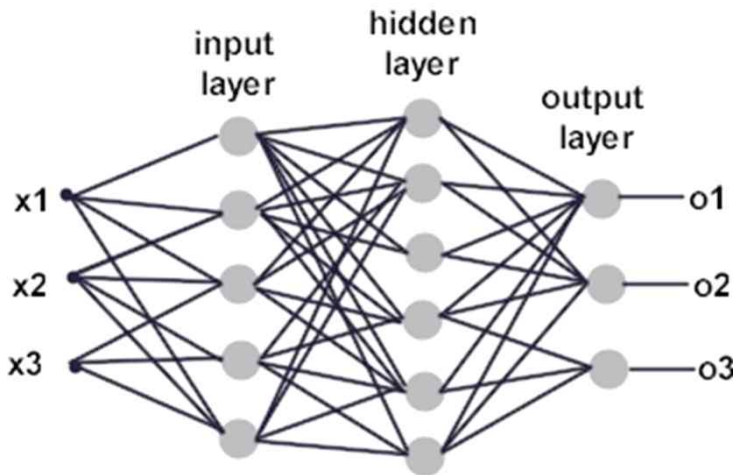
(4)

(ㄱ) : 단층 퍼셉트론(인공신경망의 한 종류)은 AND, NAND, OR 연산은 가능하지만 XOR 연산은 가능하지 않다. XOR 연산을 하기 위해서는 다층 퍼셉트론을 사용해야 한다. 여기서 연산이 가능하다는 것은 인간이 AND를 이해하는 방식으로 컴퓨터가 학습을 통해 AND 연산을 이해하고 사용할 수 있음을 의미한다.

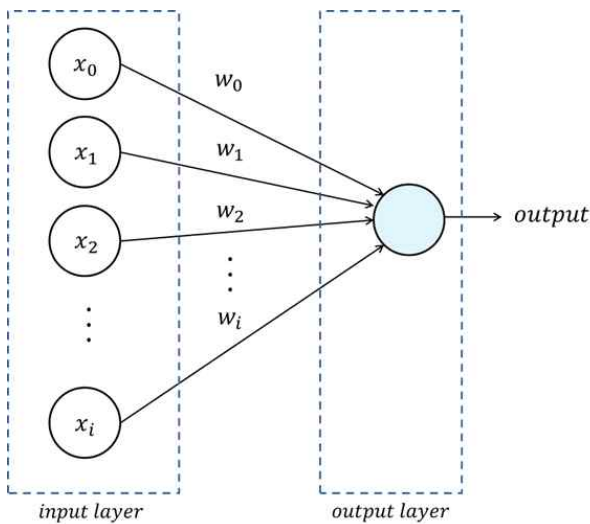
오답 체크 :

(1), (2), (3)

(L) : 다층 신경망은 아래와 같이 구성된다. 다층 신경망은 인간의 뇌구조를 컴퓨터로 구현한 것이다. 그림에서 보는 바와 같이 입력 층(input layer), 출력 층(output layer), 은닉 층(hidden layer)로 구성되는데 은닉 층은 여러 개 존재할 수 있다.



(C) : 아래 그림에서 보는 바와 같이 뉴런(신경계를 구성하는 세포) 간 연결 가중치가 존재하는 것을 알 수 있다. 그림에서 뉴런은 입력 노드와 출력 노드를 의미하고, 출력 노드의 최종 결과는 각각의 입력 노드와 가중치(w_i)를 곱하고 이들을 모두 합한 것이다. 가중치를 조정하게 되면 출력 결과가 변하므로 학습을 한다는 것은 결국 가중치가 변하는 것을 의미한다.



(E) : 인공신경망은 인간의 생물학적 뉴런 망을 컴퓨터에서 사용할 수 있도록 모델링한 것이다.

8. 네트워크 기술에 대한 설명으로 옳지 않은 것은?

- ① IPv6는 인터넷 주소 크기가 128비트이고 호스트 자동 설정 기능을 제공한다.
- ② 광대역 통합망은 응용 서비스별로 약속된 서비스 레벨 보증(Service Level Agreement) 품질 수준을 보장해 줄 수 있다.
- ③ 모바일 와이맥스(WiMAX)는 휴대형 단말기를 이용해 고속 인터넷 접속 서비스를 제공하는

무선망 기술이다.

④ SMTP(Simple Mail Transfer Protocol)는 사용자 인터페이스 구성방법을 지정하는 전송 계층 프로토콜이다.

정답 체크 :

(4) SMTP : 이메일 전송 관련 프로토콜로서 응용 계층 프로토콜이다.

오답 체크 :

- (1) IPv6 : 주소가 128비트이고, IPv6 망에 접속하면 호스트의 IP를 자동으로 설정한다.
- (2) 광대역통합망 : 종단간 고품질 서비스가 제공 가능하도록 QoS가 보장되고, SLA(Service Level Agreement)에 따른 고객의 서비스 품질 차별화 가능 및 네트워크 전체 계층의 보안을 보장한다. 여기서 SLA란 비용을 지불한 만큼 품질을 제공함을 의미한다.
- (3) 와이맥스(WiMAX) : 건물 밖으로 인터넷 사용 반경을 대폭 넓힐 수 있도록 기존의 무선랜(WiFi) 기술을 보완한 것으로 약 30마일(48km) 반경에서 70MB/s 속도로 데이터 전송을 보장하나, 이동 때 기지국과 기지국 간 핸드오프를 보장하지 못하는 단점이 있다.

9. 다음 Java 프로그램의 출력 값은?

```
class Super {
    Super() {
        System.out.print( ' A ' );
    }
    Super(char x) {
        System.out.print(x);
    }
}
class Sub extends Super {
    Sub () {
        super();
        System.out.print( ' B ' );
    }
    Sub(char x) {
        this();
        System.out.print(x);
    }
}
public class Test {
    public static void main(String [] args) {
        Super s1 = new Super('C');
        Super s2 = new Sub('D');
    }
}
```

- ① ABCD
- ② ACBD

③ CABD

④ CBAD

정답 체크 :

(3)

Super s1 = new Super('C'); // Super 클래스의 생성자 Super(char x)가 호출되므로 'C'가 출력된다. Super()는 인자가 없으므로 호출되지 않는다. 생성자 호출 시 매개변수의 유무에 주의하여야 한다.

Super s2 = new Sub('D'); // Sub 클래스의 생성자 Sub(char x)가 호출된다(매개변수의 유무에 주의). Sub(char x)에는 this()가 있는데 이는 자신의 생성자 Sub()를 호출한다(매개변수의 유무에 주의). Sub()에는 super()가 있는데 이는 자신의 부모 클래스인 Super 클래스의 생성자 Super()를 호출한다(여기서 'A'가 출력된다). super() 후에는 'B'가 출력된다. Sub() 후에는 'D'가 출력된다.

이들을 순서대로 출력하면 "CABD"가 된다.

Tip! : 해당 프로그램에서 this는 자신의 객체를 나타내고, this()는 자신의 생성자 호출을 나타낸다. 또한 super()는 부모 클래스의 생성자를 호출하는데 매개변수의 유무에 따라 적절한 부모 클래스의 생성자를 호출하면 된다.

10. 개발자가 사용해야 하는 서브 시스템의 가장 앞쪽에 위치하면서 서브 시스템에 있는 객체들을 사용할 수 있도록 인터페이스 역할을 하는 디자인 패턴은?

① Facade 패턴

② Strategy 패턴

③ Adapter 패턴

④ Singleton 패턴

정답 체크 :

(1) Facade : 몇 개의 클라이언트 클래스와 서브시스템의 클라이언트 사이에 facade라는 객체를 세워놓음으로써 복잡한 관계를 정리(구조화)한 것이다. 모든 관계가 전면에서 세워진 facade 객체를 통해서만 이루어질 수 있게 단순한 인터페이스를 제공(단순한 창구 역할)하는 것이다.

오답 체크 :

(2) Strategy : 소프트웨어 개발에서 전략이나 전술은 알고리즘으로 구현한다. 알고리즘 군을 정의하고 같은 알고리즘을 각각 하나의 클래스로 캡슐화한 다음, 필요할 때 서로 교환해서 사용할 수 있게 해준다. 알고리즘을 사용하는 곳과, 알고리즘을 제공하는 곳을 분리시킨 구조로 알고리즘을 동적으로 교체 가능하다.

(3) Adapter : 기존 클래스를 재사용할 수 있도록 중간에서 맞춰주는 역할이다. 호환성이 없는 기존 클래스의 인터페이스를 변환해 재사용할 수 있도록 해준다.

(4) Singleton : 특정 클래스의 객체가 오직 한 개만 존재하도록 보장, 즉 클래스의 객체를 하나로 제한한다. 동일한 자원이나 데이터를 처리하는 객체가 불필요하게 여러 개 만들어질 필요가 없는 경우에 주로 사용한다.

11. 소프트웨어 모듈 평가 기준으로 판단할 때, 다음 4명 중 가장 좋게 설계한 사람과 가장 좋지 않게 설계한 사람을 순서대로 바르게 나열한 것은?

- 철 수 : 절차적 응집도 + 공통 결합도
- 영 희 : 우연적 응집도 + 내용 결합도
- 동 수 : 기능적 응집도 + 자료 결합도
- 민희 : 논리적 응집도 + 스탬프 결합도

- ① 철수, 영희
- ② 철수, 민희
- ③ 동수, 영희
- ④ 동수, 민희

정답 체크 :

(3)

응집도(Cohesion)란 모듈 내부에 존재하는 구성 요소들 사이의 밀접한 정도이다. 하나의 모듈 안에서 구성 요소들 간에 푼푼 뭉쳐 있는 정도로 평가한다. 기능적 응집은 함수적 응집으로 응집도가 가장 높은 경우이며 단일 기능의 요소로 하나의 모듈을 구성한다. 우연적 응집은 구성 요소들이 말 그대로 우연히 모여 구성된 것이다. 특별한 이유 없이, 크기가 커 몇 개의 모듈로 나누는 과정에서 우연히 같이 묶인 것이다.

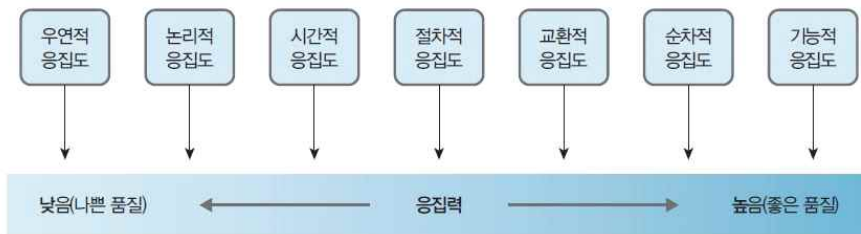


그림 6-2 모듈 내 구성 요소 간의 응집도

결합도(coupling)는 모듈과 모듈 사이의 관계에서 관련 정도를 의미한다. 하나의 모듈 안에서 구성 요소들 간에 푼푼 뭉쳐 있는 정도로 평가한다. 데이터(자료) 결합은 모듈들이 매개변수를 통해 데이터만 주고받음으로써 서로 간섭을 최소화하는 관계이다. 모듈 간의 독립성 보장한다. 내용 결합은 모듈 간에 인터페이스를 사용하지 않고 직접 왔다 갔다 하는 경우의 관계이다. 상대 모듈의 데이터를 직접 변경할 수 있어 서로 간섭을 가장 많이 하는 관계이다.

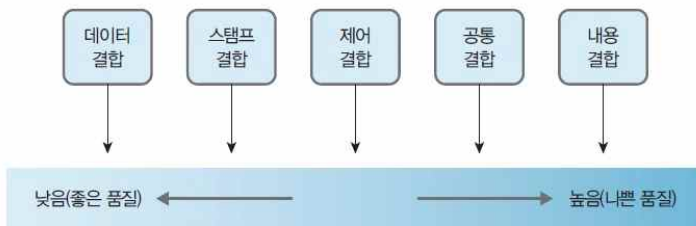


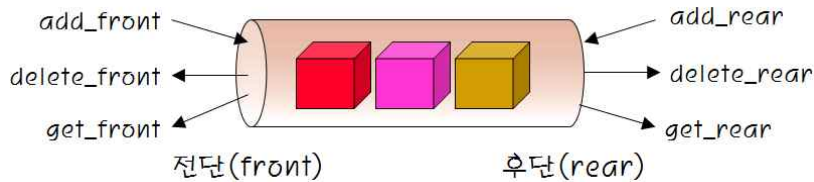
그림 6-11 모듈 간의 결합도

12. 자료 구조에 대한 설명으로 옳지 않은 것은?

- ① 데크는 삽입과 삭제를 한쪽 끝에서만 수행한다.
- ② 연결리스트로 구현된 스택은 그 크기가 가변적이다.
- ③ 배열로 구현된 스택은 구현이 간단하지만 그 크기가 고정적이다.
- ④ 원형 연결리스트는 한 노드에서 다른 모든 노드로 접근이 가능하다.

정답 체크 :

(1) 데크는 삽입과 삭제가 양쪽 끝에서 수행된다. 다음 그림은 데크(deque)를 나타낸다.



오답 체크 :

- (2) 연결리스트로 구현된 스택은 필요시 메모리를 할당해서 사용가능하므로 그 크기가 가변적이다(실행 시에 메모리를 할당할 수 있다).
- (3) 배열로 구현된 스택은 구현과 조작이 간단하지만 컴파일 전에 미리 그 크기를 고정한다.
- (4) 원형연결리스트는 한 노드에서 다른 모든 노드로 접근이 가능하지만 양방향성이 아니어서 바로 옆의 노드에 가기 위해 한 바퀴를 돌아야 하는 경우가 발생한다. 이를 위해 이중연결노드를 사용한다.

13. IPv4가 제공하는 기능만을 모두 고른 것은?

- | |
|--|
| ㄱ. 혼잡 제어
ㄴ. 인터넷 주소지정과 라우팅
ㄷ. 신뢰성 있는 전달 서비스
ㄹ. 패킷 단편화와 재조립 |
|--|

- ① ㄱ, ㄴ
- ② ㄴ, ㄷ
- ③ ㄴ, ㄹ
- ④ ㄷ, ㄹ

정답 체크 :

(3)

- (ㄴ) : 논리적인 주소 지정(IP 주소)과 라우팅 기능(Best Route)를 지정한다.
- (ㄹ) : 메시지가 1500바이트보다 큰 경우에 패킷 단편화(Fragmentation) 기능을 제공한다.

오답 체크 :

(1), (2), (4)

- (ㄱ) : 혼잡 제어는 TCP에서 제공한다. 혼잡 제어란 네트워크로 유입되는 패킷의 양을 조절하여 네트워크가 혼잡(네트워크의 중간이 라우터 등에서 혼잡)해지지 않게 조절하는 것을 말한다. 예를 들어, 패킷의 양이 많은 것을 감지하여 패킷을 적게 보내면 혼잡 현상이 일어나는 것을 막을 수 있다.
- (ㄷ) : TCP는 보낸 패킷에 대한 응답(ACK)를 받으므로 속도는 UDP에 비해 느리지만 신뢰성 있는 전달 서비스를 제공한다.

14. 결정 명령문 내의 각 조건식이 참, 거짓을 한 번 이상 갖도록 조합하여 테스트 케이스를 설계하는 방법은?

- ① 문장 검증 기준(Statement Coverage)
- ② 조건 검증 기준(Condition Coverage)

- ③ 분기 검증 기준(Branch Coverage)
- ④ 다중 조건 검증 기준(Multiple Condition Coverage)

정답 체크 :

(2) 조건 검증 : 두 개의 개별 조건식이 존재할 때 개별 조건식의 T와 F를 최소한 한 번은 테스트할 수 있도록 테스트 케이스 선정한다. 분기 검증 기준에서 발견하지 못한 오류(개별 조건식에 존재하는 오류)를 발견할 수 있는 더 강력한 테스트이다.

오답 체크 :

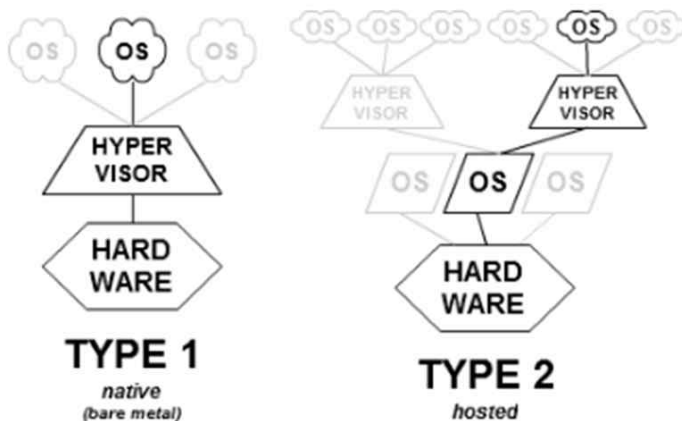
- (1) 문장 검증 : 프로그램 내의 모든 문장이 최소한 한 번은 실행될 수 있는 테스트 데이터를 갖는 테스트 케이스(입력값에 대한 출력값을 미리 만들어 놓고 테스트)를 선정한다.
- (3) 분기 검증 : 조건문에 대해 T(ture), F(false)가 최소한 한 번은 실행되는 입력 데이터를 테스트 케이스로 사용한다. 분기 시점 또는 합류 위치에서 조건과 관련된 오류를 발견할 가능성이 높다.
- (4) 다중 조건 검증 : 마스크 문제까지 해결한 테스트 케이스에 해당하는 테스트 데이터를 생성하는 기준이다. 마스크 문제란 and의 경우 두 식 중 하나가 F인 경우 나머지 식이 F이든 T이든 상관없이 결과가 F인 것이고, or인 경우 두 식 중 하나가 T인 경우 나머지 식은 F이든 T이든 상관없이 결과가 T라는 것이다.

15. 가상 머신(Virtual Machine)에 대한 설명으로 옳지 않은 것은?

- ① 단일 컴퓨터에서 가상화를 사용하여 다수의 게스트 운영체제를 실행 할 수 있다.
- ② 가상 머신은 사용자에게 다른 가상 머신의 동작에 간섭을 주지 않는 격리된 실행환경을 제공한다.
- ③ 가상 머신 모니터(Virtual Machine Monitor)를 사용하여 가상화하는 경우 반드시 호스트 운영체제가 필요하다.
- ④ 자바 가상 머신은 자바 바이트 코드가 다양한 운영체제 상에서 수행될 수 있도록 한다.

정답 체크 :

(3) 다음 그림과 같이 호스트 운영체제가 사용되는 경우와 사용되지 않는 경우라 나뉜다. 여기서 가상 머신 모니터는 하이퍼바이저(Hypervisor)라고도 하며 게스트 운영체제를 관리해준다.



오답 체크 :

- (1) 단일 컴퓨터에 오라클의 VirtualBox 여러 개를 사용하여 다수의 운영체제(리눅스, 유닉스 등)를 설치할 수 있다. 이들은 호스트 운영체제(메인 운영체제) 위에서 동작하므로 게스트 운영체제라고 부른다.

(2) 격리된 실행 환경을 제공하여 프로그램 등을 테스트하는 환경으로 사용할 수 있다. 그러나 가상 머신간 해킹은 불가능한 것은 아니다.

(4) 자바는 하이브리드 컴파일 방식을 이용하여 일단 컴파일을 수행하여 자바 바이트 코드로 만든 후 각각의 컴퓨터에 설치된 자바 가상 머신(JVM)에서 인터프리터를 이용해서 해석되어 실행된다. 이는 한번의 컴파일로 다양한 운영체제에서 실행될 수 있도록 해준다(이식성과 호환성). 그러나 JVM 위에서 동작하므로 속도가 느린 것은 감수해야 한다.

16. IEEE 802.11 무선랜에 대한 설명으로 옳은 것은?

- ① IEEE 802.11a는 5GHz 대역에서 5.5Mbps의 전송률을 제공한다.
- ② IEEE 802.11b는 직교 주파수 분할 다중화(OFDM) 방식을 사용하여 최대 22Mbps의 전송률을 제공한다.
- ③ IEEE 802.11g는 5GHz 대역에서 직접 순서 확산 대역(DSSS) 방식을 사용한다.
- ④ IEEE 802.11n은 다중입력 다중출력(MIMO) 안테나 기술을 사용한다.

정답 체크 :

(4) 802.11n : MIMO 기술(기지국과 단말기에 여러 안테나를 사용하여, 사용된 안테나수에 비례하여 용량을 높이는 기술)을 사용한다.

오답 체크 :

- (1) 802.11a : 54Mbps의 전송률을 제공한다.
- (2) 802.11b : DSSS 방식(디지털 신호를 매우 작은 전력으로 넓은 대역으로 분산하여 동시에 송신)을 사용하여 11Mbps의 전송률을 제공한다.
- (3) 802.11g : 2.4GHz 대역이다.

Tip! : 802.11n 무선 랜을 테이블로 정리하면 다음과 같다. 여기서, OFDM은 하나의 정보를 여러 개의 반송파(캐리어)로 분할하고, 분할된 반송파 간의 간격을 최소화하기 위해 직교성을 부가하여 다중화해 전송하는 방법이다. 그리고 MU-MIMO는 제각기 하나 이상의 안테나를 갖춘 사용자나 무선 터미널들이 서로 통신하는, 무선 통신을 위한 MIMO 기술들의 집합을 나타낸다.

〈표 6-8〉 무선랜 비교

무선랜 규격	최대속도 (Mbps)	채널 대역폭 (MHz)	사용 주파수 대역 (GHz)	주요 특징
802.11b	11	20	2.4	• 초기 기술, 느린 전송속도 • DSSS
802.11a	54	20	5	• OFDM
802.11g	54	20	2.4	• OFDM, DSSS • 2.4GHz 대역 간섭
802.11n	600	20/40	2.4/5	• OFDM • MIMO
802.11ac	6900	20/40/80/160		• OFDM • MU-MIMO

17. 데이터베이스의 동시성 제어에 대한 설명으로 옳지 않은 것은? (단, T1, T2, T3는 트랜잭션이고, A는 데이터 항목이다)

- ① 다중버전 동시성 제어 기법은 한 데이터 항목이 변경될 때 그 항목의 이전 값을 보존한다.
- ② T1이 A에 배타 로크를 요청할 때, 현재 T2가 A에 대한 공유 로크를 보유하고 있고 T3가 A에 공유 로크를 동시에 요청한다면, 트랜잭션 기아 회피 기법이 없는 경우 A에 대한 로크를

T3가 T1보다 먼저 보유한다.

③ 로크 전환이 가능한 상태에서 T1이 A에 대한 배타 로크를 요청 할 때, 현재 T1이 A에 대한 공유 로크를 보유하고 있는 유일한 트랜잭션인 경우 T1은 A에 대한 로크를 배타 로크로 상승할 수 있다.

④ 2단계 로킹 프로토콜에서 각 트랜잭션이 정상적으로 커밋될 때까지 자신이 가진 모든 배타적 로크들을 해제하지 않는다면 모든 교착 상태를 방지할 수 있다.

정답 체크 :

(4) 교착상태란 트랜잭션들이 상대가 독점하고 있는 데이터에 unlock(해제) 연산이 실행되기를 서로 기다리면서 트랜잭션의 수행을 중단하고 있는 상태이다. 그러므로 정상적으로 커밋될 때까지 자신이 가진 모든 배타적 로크들을 해제하지 않으면 교착상태가 오히려 증가한다.

오답 체크 :

(1) 다중 버전 동시성 제어 기법은 트랜잭션이 한 데이터 아이템을 접근하려 할 때, 그 트랜잭션의 타임스탬프와 접근하려는 데이터 아이템의 여러 버전의 타임스탬프를 비교하여, 현재 실행하고 있는 스케줄의 직렬가능성이 보장되는 적절한 버전을 선택하여 접근하도록 하는 기법이다. 그러므로 한 데이터 항목이 변경될 때 그 항목의 이전 값을 보존하여 다중 버전을 만든다.

(2) 기본 로크(lock, 로킹, locking) 규약에 따라 공유 로크(T2)에 대해서는 공유 로크(T3)만 요청할 수 있다. 만약, 트랜잭션 기아(트랜잭션이 수행되지 못하고 계속 무한정 기다림) 회피 기법이 존재하면 선입선처리 기법(공평한 처리 방식)에 따라 먼저 요청한 T1이 A에 대한 로크를 보유한다.

(3) 로크 전환이란 항목 A가 보유하고 있는 로크를 하나의 로크 상태에서부터 다른 로크 상태로 전환하는 것을 의미한다. 로크 상승(공유 로크에서 배타 로크로 전환)과 로크 하강(배타 로크에서 공유 로크로 전환)이 가능하다. 단, 전제 조건은 로크를 보유한 트랜잭션이 하나이어야 한다.

18. 파일구조에 대한 설명으로 옳지 않은 것은?

① VSAM은 B+ 트리 인덱스 구조를 사용한다.

② 히프 파일은 레코드들을 키 순서와 관계없이 저장 할 수 있다.

③ ISAM은 레코드 삽입을 위한 별도의 오버플로우 영역을 필요로 하지 않는다.

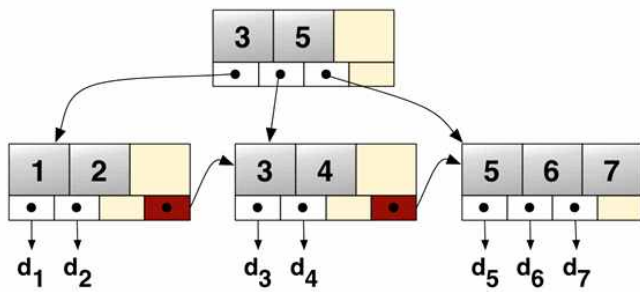
④ 순차 파일에서 일부 레코드들이 키 순서와 다르게 저장 된 경우, 파일 재구성 과정을 통해 키 순서대로 저장될 수 있다.

정답 체크 :

(3) ISAM : 자료를 순서대로 처리할 수도 있고 특정 항목을 색인으로 하여 순서에 관계없이 처리할 수도 있다. ISAM을 적용하는 파일은 색인부(자료의 색인), 주저장부(자료를 새로 저장), 오버플로부(자료의 중간에 새로운 자료를 저장) 등 3개의 영역으로 나누어 관리한다.

오답 체크 :

(1) VSAM : 가상 기억 환경에서 직접 접근과 순차 접근의 2가지 방식으로 기억 장치에 있는 데이터를 읽거나 기록할 수 있게 한다. 1차 색인이 B+ 트리 구조로 구성되고 나무의 노드에 탐색 키, 앞의 부분에 레코드가 배치되어 필요에 따라 보조 기억 장치와 주기억 장치 사이에 데이터의 페이지가 전송된다. 이를 그림으로 나타내면 다음과 같다.



(2) 히프 파일(Heap file) : 데이터가 입력되는 순서대로 저장된 파일이다. 즉, 키 순서와 관계 없다. 레코드에 대한 분석, 분류, 표준화 과정을 거치지 않는다. 필드의 순서, 길이 등에 제한이 없고, 레코드의 길이, 타입이 일정하지 않다. 파일 화일(pile file)이라고도 한다.

(4) 순차 파일(Sequential file) : 화일 내에서의 레코드는 키 필드 값에 따라 정렬된다. 모든 레코드는 똑같은 순서의 데이터 필드로 구성된다. 파일에 새로운 레코드를 삽입하거나 삭제하는 경우 파일 재구성을 수행한다.

19. 다음 C 프로그램의 출력 값은?

```
#include <stdio.h>
int a = 10;
int b = 20;
int c = 30;
void func(void)
{
    static int a = 100;
    int b = 200;
    a++;
    b++;
    c = a;
}
int main(void)
{
    func();
    func();
    printf("a = % d, b = % d, c = % d\n" , a, b, c);
    return 0;
}
```

- ① a = 10, b = 20, c = 30
- ② a = 10, b = 20, c = 102
- ③ a = 101, b = 201, c = 101
- ④ a = 102, b = 202, c = 102

정답 체크 :

(2)

printf("a = %d, b = %d, c = %d\n", a, b, c); // 일단 main 안에 지역변수 a, b, c가 없기 때문에 전역 변수 a, b, c를 출력하게 된다. func에 a, b가 있지만 해당 지역변수는 func 함수 안에서만 사용할 수 있다. 그러므로 a는 10이고, b는 20이 된다. c는 func 함수 안에서 변하기 때문에 두 번 호출되는 func 함수를 분석해야 한다.

첫 번째 func() 호출 : 정적변수 a는 10이 되고, c에 할당된다.

두 번째 func() 호출 : 정적변수 a는 두 번째 호출에도 해당 값이 유지하므로 102가 되고, c에 할당된다. 그러므로 최종 c는 102가 된다.

Tip! : 해당 문제의 경우 지역 변수와 전역 변수의 특성을 이해하면 바로 문제가 풀리므로 func 함수를 분석할 필요가 없다. 왜냐하면 직관적으로 전역변수 c는 변한다는 것을 알 수 있기 때문에 답이 정해져있다. 시간을 절약하기 위해 해당 문제의 경우에는 답을 작성하고 나중에 확인해도 된다.

20. 해싱(Hashing)에 대한 설명으로 옳지 않은 것은?

- ① 서로 다른 탐색키가 해시 함수를 통해 동일한 해시 주소로 사상 될 수 있다.
- ② 충돌(Collision)이 발생하지 않는 해시 함수를 사용한다면 해싱의 탐색 시간 복잡도는 $O(1)$ 이다.
- ③ 선형 조사법(Linear Probing)은 연결리스트(Linked List)를 사용하여 오버플로우 문제를 해결한다.
- ④ 폴딩 함수(Folding Function)는 탐색키를 여러 부분으로 나누어 이들을 더하거나 배타적 논리합을 하여 해시 주소를 얻는다.

정답 체크 :

(3) 선형 조사법 : 해당 설명은 체이닝이고, 선형조사법은 충돌이 일어난 항목을 해시 테이블의 다른 위치에 저장한다.

오답 체크 :

- (1) 해싱은 충돌이 발생할 수 있다.
- (2) 충돌이 발생하지 않으면 해시 함수를 통해 바로 탐색이 되기 때문에 시작 복잡도는 $O(1)$, 즉 탐색에 상수의 시간(고정된 시간)이 소요된다.
- (4) 폴딩 함수는 아래와 같이 계산된다.

