2020-지방직-컴퓨터일반-B형-해설

해설에 대한 모든 권리는 곽후근(gobarian@gmail.com)에게 있습니다.

1. 인터프리터(Interpreter) 방식의 언어로 옳지 않은 것은?
① JavaScript
② C
③ Basic
④ LISP
정답 체크 :
(2) C : 대표적인 컴파일러 언어이다.
오답 체크 :
(1) JavaScript : 스크립트 언어는 인터프리터 방식으로 동작한다.
(3) Basic : 초보자도 쉽게 배울 수 있도록 만들어진 대화형 프로그래밍 언어로 1963년에 개발되었다.
(4) LISP : 인공지능 개발용 언어로 사용되며, 연결 리스트를 주요 자료 구조로 사용한다.

2. CPU 스케줄링 기법 중 라운드 로빈(Round Robin) 방식에 대한 설명으로 옳지 않은 것은?
① 선점 스케줄링 기법이다.
② 여러 프로세스에 일정한 시간을 할당한다.
③ 시간할당량이 작으면 문맥 교환수와 오버헤드가 증가한다.
④ FIFO(First-In-First-Out) 방식 대비 높은 처리량을 제공한다.
정답 체크 :
(4) 작업 길이가 다양할 때는 이전 작업을 마친 후 후보가 규정 시간량을 마치고 다음 작업으로 이동하기 때문에, 평균 대기 시간이 선입선처리보다 적다. 미완성 작업은 각 규정 시간량을 마친 후 프로세서를 기다리므로 평균 처리 시간이 높다(오랜 시간).
오답 체크 :
(1) 일정 시간 후에 다른 프로세스가 CPU를 사용하므로 선점 방식이다.
(2) time quantum or time slice을 할당한다.
(3) 시간할당량이 작으면 문맥 교환수가 증가하고, 이는 오버헤드의 증가로 이어진다.

3. 프로세서의 수를 늘려도 속도를 개선하는 데 한계가 있다는 주장으로서, 병렬처리 프로세서의 성능 향상의 한계를 지적한 법칙은?
① 무어의 법칙(Moore's Law)
② 암달의 법칙(Amdahl's Law)
③ 구스타프슨의 법칙(Gustafson's Law)
④ 폰노이만 아키텍처(von Neumann Architecture)
정답 체크 :
(2) 암달의 법칙 : 컴퓨터 시스템의 일부를 개선할 때 전체적으로 얼마만큼의 최대 성능 향상이 있는지 계산하는 데 사용된다. 병렬 컴퓨팅에서 멀티 프로세서를 사용할 때 프로그램의 성능향상은 프로그램의 순차적인 부분에 의해 제한된다.
오답 체크 :

(1) 무어의 법칙 : 반도체 집적회로의 성능이 24개월마다 2배로 증가한다는 법칙이다.
(3) 구스타프슨의 법칙 : 컴퓨터 과학에서 대용량 데이터 처리는 효과적으로 병렬화할 수 있다는 법칙이다. 구스타프슨-바시스의 법칙(Gustafson-Barsis' law)으로도 알려져 있다. 스타프슨의 법칙은 병렬화로 얻을 수 있는 프로그램의 성능 향상은 프로그램의 순차적인 부분에 의해 제한된다는 암달의 법칙에 대한 반대 개념이다. 구스타프슨의 법칙은 처음에 존 구스타프슨과 동료인 에드윈 바시스에 의해서 처음 발표되었다. S(P) = P-α(P-1). 여기서 P는 프로세서의 수이며 S는 성능향상, α는 어떤 병렬 프로세스에서 병렬화 되지 않는 순차적인 부분의 비를 말한다.
(4) 폰노이만 아키텍처 : 폰노이만은 산술 논리 장치와 프로세서 레지스터를 포함하는 처리 장치, 명령 레지스터와 프로그램 카운터를 포함하는 컨트롤 유닛, 데이터와 명령어를 저장하는 메모리, 외부 대용량 스토리지, 입출력 매커니즘을 제안하였다.

4. 교착상태 발생의 조건에 대한 설명으로 옳지 않은 것은?
① 상호 배제 조건: 최소한 하나의 자원이 비공유 모드로 점유되며, 비공유 모드에서는 한 번에 한 프로세스만 해당 자원을 사용할 수 있다.
② 점유와 대기 조건: 프로세스는 최소한 하나의 자원을 점유한 채, 현재 다른 프로세스에 의해 점유된 자원을 추가로 얻기 위해 반드시 대기해야 한다.
③ 비선점 조건: 프로세스에 할당된 자원은 사용이 끝날 때까지 다른 프로세스가 강제로 빼앗을 수 없다.
④ 순환 대기 조건: 대기 체인 내 프로세스들의 집합에서 이전 프로세스는 다음 프로세스가 점유한 자원을 대기하고, 마지막 프로세스는 자원을 대기하지 않아야 한다.
정답 체크 :
(4) 순환 대기 조건 : 아래 그림에서 보는 바와 같이 마지막 프로세스는 제일 처음 프로세스가 점유한 자원을 대기한다.

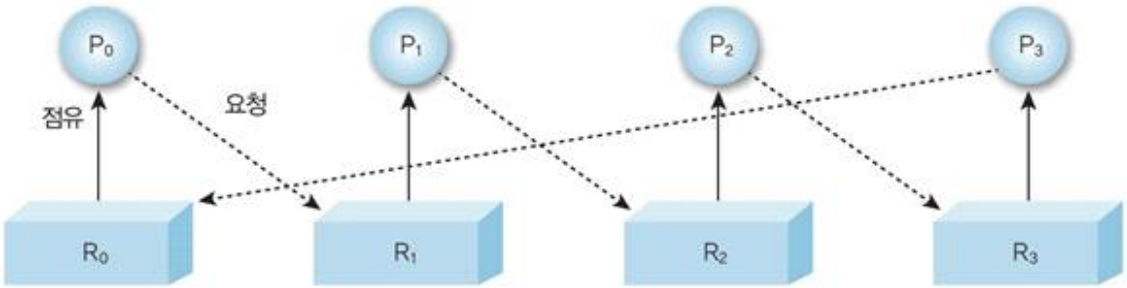


그림 5-5 순환 대기의 교착 상태

오답 체크 :
(1) 상호 배제 조건 : 자원을 최소 하나 이상 비공유한다. 즉, 한 번에 프로세스 하나만 해당 자원 사용할 수 있어야 한다. 사용 중인 자원을 다른 프로세스가 사용하려면, 요청한 자원 해제될 때 까지 대기한다.
(2) 점유와 대기 조건 : 자원을 최소한 하나 정도 보유하고, 다른 프로세스에 할당된 자원 얻으려고 대기한다.
(3) 비선점 조건 : 자원 선점 불가이다. 즉, 자원은 강제로 빼앗을 수 없고, 자원 점유하고 있는 프로세스 끝나야 해제한다.

5. CPU(중앙처리장치)의 성능 향상을 위해 한 명령어 사이클 동안 여러 개의 명령어를 동시에 처리할 수 있도록 설계한 CPU구조는?
① 슈퍼스칼라(Superscalar)

② 분기 예측(Branch Prediction)
③ VLIW(Very Long Instruction Word)
④ SIMD(Single Instruction Multiple Data)
정답 체크 :
(1) 슈퍼 스칼라 : 파이프라인을 개선하기 위한 구조로 하드웨어를 추가하여 동시에 여러개의 명령어를 처리한다.
오답 체크 :
(2) 분기 예측 : 파이프라인의 제어 해저드를 해결하기 위한 방법이다. 분기를 예측해서 다음에 실제로 처리되는 명령어를 가지고 온다.
(3) VLIW : 여러 opcode 필드가 있는 긴 명령어 하나에 독립적인 연산 여러개를 정의하고 이들을 한꺼번에 내보내는 명령어 구조 집합의 종류이다.
(4) SIMD : 하나의 명령어 스트림이 다수개의 데이터 스트림을 만들어내는 구조이다. 벡터 프로세서 또는 Array 프로세서에 해당하는 구조이다.

6. 캐시기억장치 접근시간이 20 ns, 주기억장치 접근시간이 150 ns, 캐시기억장치 적중률이 80 %인 경우에 평균 기억장치 접근시간은? (단, 기억장치는 캐시와 주기억장치로만 구성된다)
① 32 ns
② 46 ns
③ 124 ns
④ 170 ns
정답 체크 :
(2) 평균 기억장치 접근시간 = 적중률 x 캐시기억장치 접근시간 + (1-적중률) x 주기억장치 접근시간
= 0.8x20 + 0.2x150 = 16 + 30 = 46ns

7. 아날로그 컴퓨터에 대한 설명으로 옳지 않은 것은?
① 입력형식은 부호, 코드화된 숫자, 문자, 기호이다.
② 출력형식은 곡선, 그래프 등이다.
③ 미적분 연산방식을 가지며, 정보처리속도가 빠르다.
④ 증폭회로 등으로 회로 구성을 한다.
정답 체크 :
(1) 입력 형식은 전류, 전압, 온도, 속도, 압력과 같은 연속된 자료이다.
오답 체크 :
(2) 출력 형식은 곡선, 그래프와 같은 연속된 자료이다.
(3) 아날로그 컴퓨터는 미적분 연산을 사용하고, 디지털 컴퓨터는 사칙과 논리 연산을 사용한다.
(4) 아날로그 컴퓨터는 증폭회로를 사용하고, 디지털 컴퓨터는 논리회로를 사용한다.

8. RAID(Redundant Array of Inexpensive Disks)에 대한 설명으로 옳지 않은 것은?
① RAID-0은 디스크 스트라이핑(Disk Striping) 방식으로 중복 저장과 오류 검출 및 교정이 없는 방식이다.
② RAID-1은 디스크 미러링(Disk Mirroring) 방식으로 높은 신뢰도를 갖는다.
③ RAID-4는 데이터를 비트(bit) 단위로 여러 디스크에 분할하여 저장하는 방식이며, 별도의 패리티(parity)

디스크를 사용한다.
④ RAID-5는 별도의 패리티 디스크 대신 모든 디스크에 패리티 정보를 나누어 기록하는 방식이다.
정답 체크 :
(3) RAID-4 : 미리 정해진 블록 단위로 데이터를 분할하고 패리티를 계산한다.
오답 체크 :
(1) RAID-0 : 디스크 스트라이핑 방식을 사용한다.
(2) RAID-1 : 디스크 미러링 방식을 사용한다.
(4) RAID-5 : 모든 디스크에 패리티 정보를 나눈다.

9. 다음 재귀 함수를 동일한 기능의 반복 함수로 바꿀 때, ㉠과 ㉡에 들어갈 내용을 바르게 연결한 것은?

```
int func (int n) {                    //재귀 함수
        if (n == 0)
                return 1;
        else
                return n * func (n - 1);
}
int iter_func (int n) {               //반복 함수
        int f = 1;
        while ( ____㉠____ )
              ____㉡____
        return f;
}
```

㉠ ㉡
① n < 0 f = f * n--;
② n < 0 f = f * n++;
③ n > 0 f = f * n--;
④ n > 0 f = f * n++;
정답 체크 :
(3) n > 0 // 주어진 n을 감소시킬 것이기 때문에 0보다 크다는 조건을 가진다.
f = f * n--; // n을 하나씩 감소하면서 자신과 곱해주는 연산을 수행한다.

10. 데이터의 종류 및 처리에 대한 설명으로 옳지 않은 것은?
① 크롤링(Crawling)을 통해 얻은 웹문서의 텍스트 데이터는 대표적인 정형 데이터(Structured Data)이다.
② XML로 작성된 IoT 센서 데이터는 반정형 데이터(Semi-structured Data)로 분류할 수 있다.
③ 반정형 데이터는 데이터 구조에 대한 메타 데이터(Meta-data)를 포함한다.
④ NoSQL과 Hadoop은 대규모 비정형 데이터(Unstructured Data) 처리에 적합하다.
정답 체크 :
(1) 크롤링을 통해 얻은 웹문서의 텍스트 데이터는 대표적인 비정형 데이터이다.
오답 체크 :
(2) 반정형 데이터의 예는 URL 형태로 존재하는 HTML, 로그형태인 웹로그, IOT에서 제공하는 센서 데이터

등을 들 수 있다.
(3) 반정형 데이터는 데이터 내부에 정형데이터의 스키마에 해당되는 메타데이터를 갖고 있으며, 일반적으로 파일 형태로 저장된다.
(4) NoSQL과 Hadoop은 빅데이터 처리를 위한 플랫폼으로 대규모 비정형 데이터 처리에 적합하다.
Tip! 정형, 반정형, 비정형
정형 데이터(Structured Data) : 관계형 데이터베이스 시스템의 테이블과 같이 고정된 컬럼에 저장되는 데이터와 파일, 그리고 지정된 행과 열에 의해 데이터의 속성이 구별되는 스프레드시트 형태의 데이터도 있을 수 있다. 예는 RDBMS의 테이블들(단일 테이블 혹은 조인한 테이블 포함), 스프레드시트 등을 들 수 있다.
반정형 데이터(Semi-structured Data) : 데이터 내부에 정형데이터의 스키마에 해당되는 메타데이터를 갖고 있으며, 일반적으로 파일 형태로 저장된다. 예는 URL 형태로 존재(HTML), 로그형태(웹로그), IOT에서 제공하는 센서 데이터 등을 들 수 있다.
비정형 데이터(Unstructured-Data) :
데이터 세트가 아닌 하나의 데이터가 수집 데이터로 객체화돼 있다. 언어 분석이 가능한 텍스트 데이터나 이미지, 동영상 같은 멀티미디어 데이터가 대표적인 비정형 데이터다. 예는 이진 파일 형태(동영상, 이미지), 스크립트 파일 형태(소셜 데이터의 텍스트) 등을 들 수 있다.

11. 페이지 부재율(Page Fault Ratio)과 스래싱(Trashing)에 대한 설명으로 옳은 것은?
① 페이지 부재율이 크면 스래싱이 적게 일어난다.
② 페이지 부재율과 스래싱은 전혀 관계가 없다.
③ 스래싱이 많이 발생하면 페이지 부재율이 감소한다.
④ 다중 프로그램의 정도가 높을수록 스래싱이 증가한다.
정답 체크 :
(4) 다음 그림과 같이 다중 프로그램의 정도가 높을수록 스래싱이 증가한다.

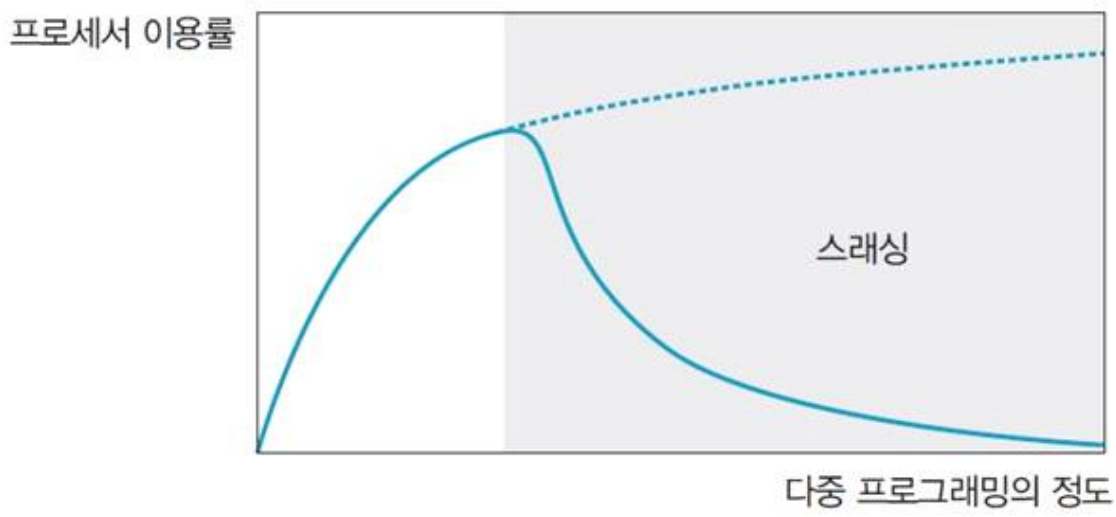


그림 8-33 프로세서의 이용률에 따른 스래싱 발생률

오답 체크 :
(1) 페이지 부재율이 크면 스래싱이 많이 일어난다.
(2) 페이지 부재율이 크면 스래싱이 많이 발생하고, 페이지 부재율이 작으면 스래싱이 적게 발생한다(다음 그

림 참고).

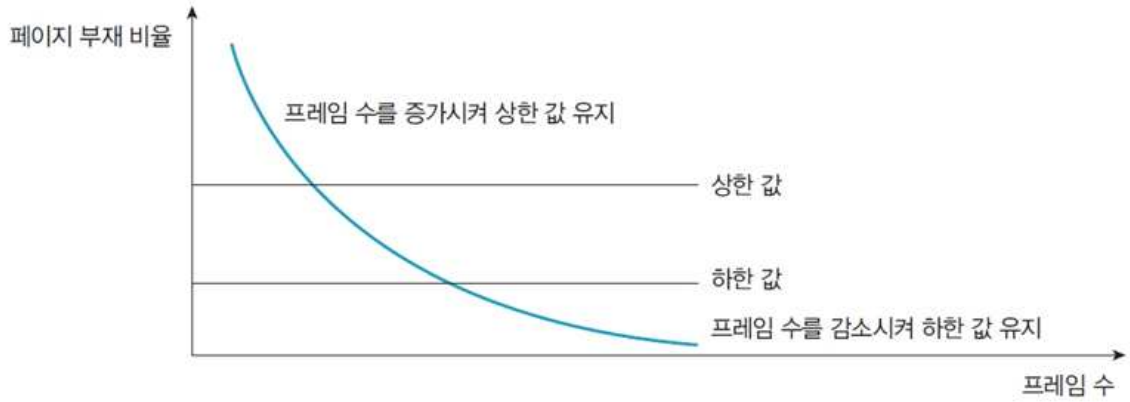


그림 8-41 페이지 부재 비율

(3) 스래싱이 많이 발생하면 페이지 부재율이 증가한다.

12. 전자상거래 관련 기술 중 고객의 요구에 맞춰 자재조달에서부터 생산, 판매, 유통에 이르기까지 공급사슬 전체의 기능통합과 최적화를 지향하는 정보시스템은?

① ERP(Enterprise Resource Planning)

② EDI(Electronic Data Interchange)

③ SCM(Supply Chain Management)

④ KMS(Knowledge Management System)

정답 체크 :

(3) SCM : 기업 내에 부문별 최적화나 개별 기업단위의 최적화에서 탈피하여 공급망의 구성요소들 간에 이루어지는 전체프로세스 최적화를 달성하고자 하는 경영혁신기법이다.

오답 체크 :

(1) ERP : 생산, 판매, 자재,인사, 회계 등 기업 전 부문에 걸쳐 있는 인력, 자금, 정보 등 모든 경영자원을 하나의 체계로 통합, 계획, 관리함으로써 기업 생산성을 높이는 종합경영 관리시스템이다.

(2) EDI : 독립된 조직 간에 정형화된 문서를 표준화된 자료표현 양식에 준하여 전자적 통신매체를 이용해 교환하는 방식이다.

(4) KMS : 인적 자원이 소유하고 있는 비정형 데이터인 지적 자산을 기업 내에 축적·활용할 수 있도록 하자는 것이다.

13. 프로토콜과 이에 대응하는 TCP/IP 프로토콜 계층 사이의 연결이 옳지 않은 것은?

① HTTP – 응용 계층

② SMTP – 데이터링크 계층

③ IP – 네트워크 계층

④ UDP – 전송 계층

정답 체크 :

(2) SMTP는 응용 계층이다.

오답 체크 :

(1) HTTP, (3) IP, (4) UDP는 다음 그림(OSI 7 Layer)을 참고한다.

7	응용 계층	HTTP, SMTP, SNMP, FTP, 텔넷, SSH & Scp, NFS, RTSP
6	표현 계층	XDR, ASN.1, SMB, AFP
5	세션 계층	TLS, SSL, ISO 8327 / CCITT X.225, RPC, 넷바이오스, 애플토크
4	전송 계층	TCP, UDP, RTP, SCTP, SPX, 애플토크
3	네트워크 계층	IP, ICMP, IGMP, X.25, CLNP, ARP, RARP, BGP, OSPF, RIP, IPX, DDP
2	데이터 링크 계층	이더넷, 토큰링, PPP, HDLC, 프레임 릴레이, ISDN, ATM, 무선랜, FDDI
1	물리 계층	전선, 전파, 광섬유, 동축케이블, 도파관, PSTN, 리피터, DSU, CSU, 모뎀

14. 관계 데이터베이스 스키마 STUDENT(SNO, NAME, AGE)에 대하여 다음과 같은 SQL 질의 문장을 사용한다고 할 때, 이 SQL 문장과 동일한 의미의 관계대수식은? (단, STUDENT 스키마에서 밑줄 친 속성은 기본키 속성을, 관계대수식에서 사용하는 관계대수 연산자 기호 π는 프로젝트 연산자를, σ는 셀렉트 연산자를 나타낸다)

<SQL 질의문>

```
SELECT SNO, NAME
FROM STUDENT
WHERE AGE > 20;
```

① $\sigma_{SNO,NAME}(\pi_{AGE>20}(STUDENT))$

② $\pi_{SNO,NAME}(\sigma_{AGE>20}(STUDENT))$

③ $\sigma_{AGE>20}(\pi_{SNO,NAME}(STUDENT))$

④ $\pi_{AGE>20}(\sigma_{SNO,NAME}(STUDENT))$

정답 체크 :

(2) 다음의 표에서 알 수 있듯이 셀렉트는 조건을 만족하는 투플들을 반환하고, 프로젝트는 주어진 속성들의 값으로만 구성된 투플들을 반환한다.

표 6-2 순수 관계 연산자의 종류

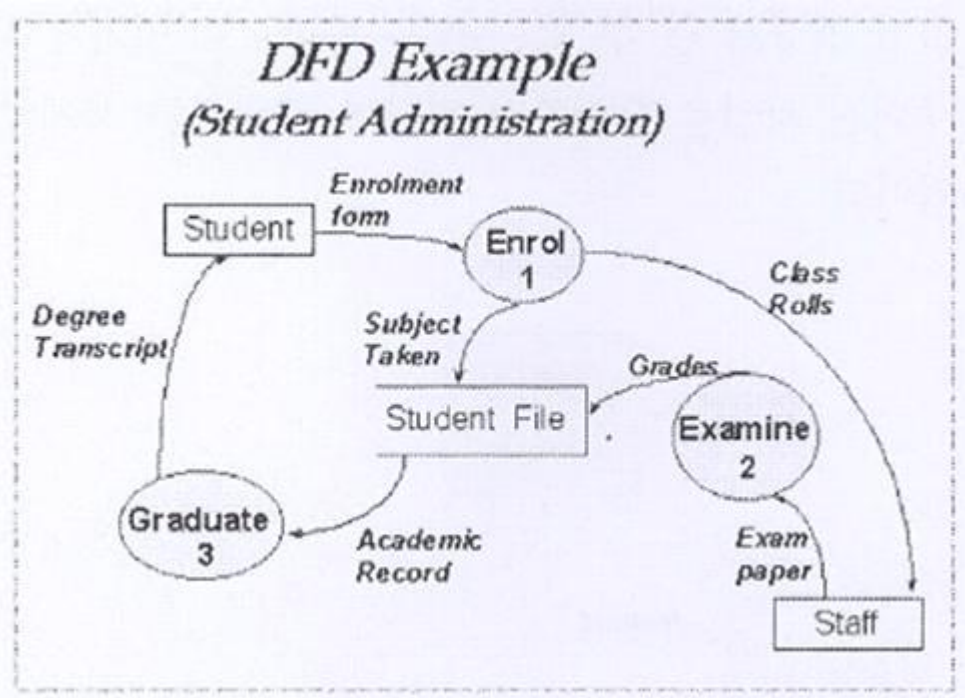
연산자	기호	표현	의미
셀렉트	σ	$\sigma_{조건}(R)$	릴레이션 R에서 조건을 만족하는 투플들을 반환
프로젝트	π	$\pi_{속성리스트}(R)$	릴레이션 R에서 주어진 속성들의 값으로만 구성된 투플들을 반환
조인	⋈	R ⋈ S	공통 속성을 이용해 릴레이션 R과 S의 투플들을 연결하여 만들어진 새로운 투플들을 반환
디비전	÷	R ÷ S	릴레이션 S의 모든 투플과 관련이 있는 릴레이션 R의 투플들을 반환

15. 두 프로토콜 개체 사이에서 흐름제어와 오류제어 및 메시지 전달 등의 기능을 수행하며, 연결성과 비연결성의 두 가지 운용모드를 제공하는 OSI 참조 모델 계층은?
① 데이터링크 계층(Datalink Layer)
② 네트워크 계층(Network Layer)
③ 전송 계층(Transport Layer)
④ 응용 계층(Application Layer)
정답 체크 :
(3) 전송 계층 : 메시지 종단(End-to-End) 전달, 포트 주소 지정, 단편화와 재조립, 연결 제어(관리), 흐름 제어, 혼잡 제어 등을 수행한다.
오답 체크 :
(1) 데이터링크 계층 : 패킷 노드(Node-to-Node or Peer-to-Peer) 전달, 물리적인(MAC) 주소 기정, 접근 제어(MAC filtering), 흐름 제어(stop-and-wait, sliding window), 오류 처리(ARQ) 등을 수행한다.
(2) 네트워크 계층 : 패킷 종단(End-to-End) 전달, 논리적인(IP) 주소 지정, 경로 지정(Routing), 주소 변환(ARP) 등을 수행한다.
(4) 응용 계층 : 네트워크 상의 소프트웨어 사용자에게 사용자 인터페이스 제공한다. 전자우편(X.400), 원격파일 접근과 전송(FTAM), 공유 데이터베이스 관리 및 여러 종류의 분산 정보 서비스(X.500) 제공한다.

16. 소프트웨어 개발 언어에 대한 설명으로 옳지 않은 것은?
① C#은 마이크로소프트 닷넷 프레임워크를 지원하는 객체지향 언어이다.
② Python은 인터프리터 방식의 객체지향 언어로서 실행시점에 데이터 타입을 결정하는 동적 타이핑 기능을 갖는다.
③ Kotlin은 그래픽 요소를 강화한 게임 개발 전용 언어이다.
④ Java는 컴파일된 프로그램이 JVM상에서 인터프리터 방식으로 실행되는 플랫폼 독립적 프로그래밍 언어이다.
정답 체크 :
(3) Kotlin은 JVM에서 동작하는 프로그래밍 언어이고, 2011년 7월, 젯브레인사가 공개하였다. 코틀린은 2019년 5월 7일 이후로 구글의 안드로이드 앱 개발에서 선호하는 언어가 되었다.
오답 체크 :
(1) C#은 기존 언어들의 장점을 결합한 마이크로소프트 닷넷 프레임워크를 지원하는 객체지향 언어이다.
(2) Python은 인터프리터 방식의 객체지향 언어이고 동적 타이핑 기능(아래의 Tip 참고)을 지원한다.
(4) Java는 컴파일된 프로그램(자바 바이트코드)이 JVM 상에서 인터프리터 방식으로 실행되므로 플랫폼 독립적이다.
Tip! 동적 타이핑(dynamic typing)
자료형 검사의 대부분이 컴파일 타임이 아닌 실행 시간에 수행될 경우에 사용한다. 값은 자료형을 가지고 있지만 변수는 그렇지 않다. 즉, 변수는 모든 자료형의 값을 가질 수 있다. 동적으로 정형된 언어는 일반적으로 모든 런타임 개체가 자료형 정보를 함께 갖고 있다. 이런 런타임 자료형 정보는 함수 오버로드, 동적 디스패치, 게으른 바인딩, 다운캐스팅 등에도 사용될 수 있다.

17. 소프트웨어 시스템은 기능 관점, 동적 관점 및 정보 관점으로 분류할 수 있다. 동적 관점에서 시스템을 기술할 때 사용할 수 있는 도구로 옳지 않은 것은?
① 사건 추적도(Event Trace Diagram)

② 자료 흐름도(Data Flow Diagram)
③ 상태 변화도(State Transition Diagram)
④ 페트리넷(Petri Net)
정답 체크 :
(2) 자료 흐름도 : 다음 그림과 같이 모듈 사이의 관계를 나타내는 그래프로 동적 관점 보다는 기능 관점을 기술한다.



오답 체크 :
(1) 사건 추적도 : 일반적으로 사건이 일어나는 일종의 시나리오를 작성하여 사건을 추적한다. 사건 추적도에서는 사건의 순서와 사건을 주고받는 객체나 시스템을 표현하며, 사건의 흐름은 화살표로 표시한다.
(3) 상태 변화도 : 시스템이 가지고 있는 값을 표시하는 상태와 외부에서 가해지는 사건이다. 일반적으로 상태 변화도의 상태는 사건에 의해서 변형되고 상태에 의해 사건을 제시한다.
(4) 페트리넷 : 시스템 모델링 방법중 하나로 1962년 Petri라는 학자에 의해 만들어졌다. 페트리넷의 구성요소는 시스템내에서의 가능한 모든 "상태"와 한 상태에서 다른 상태를 이어주는 동작에 해당되는 "전이"가 있다.

18. 다음에서 설명하는 네트워크 데이터 오류 검출 방법은?

송신측: 첫 번째 비트가 1로 시작하는 임의의 n+1비트의 제수를 결정한다. 그리고 전송하고자 하는 데이터 끝에 n비트의 0을 추가한 후 제수로 모듈로-2 연산을 한다. 그러면 n비트의 나머지가 구해지는데 이 나머지가 중복 정보가 된다. 수신측: 계산된 중복 정보를 데이터와 함께 전송하면 수신측에서는 전송받은 정보를 동일한 n+1제수로 모듈로-2 연산을 한다. 나머지가 0이면 오류가 없는 것으로 판단하고, 나머지가 0

이 아니면 오류로 간주한다.

① 수직 중복 검사(Vertical Redundancy Check)

② 세로 중복 검사(Longitudinal Redundancy Check)

③ 순환 중복 검사(Cyclic Redundancy Check)

④ 체크섬(Checksum)

정답 체크 :

(3) 순환 중복 검사 : 모듈로-2 연산과 다항식을 사용하고, 2계층(데이터링크 계층)에서 사용한다.

오답 체크 :

(1) 수직 중복 검사(각 단위의 전송 방향에서 패리티 비트), (2) 세로 중복 검사(전체 블록의 전송 방향에서는 패리티 비트)는 다음 그림과 같다.

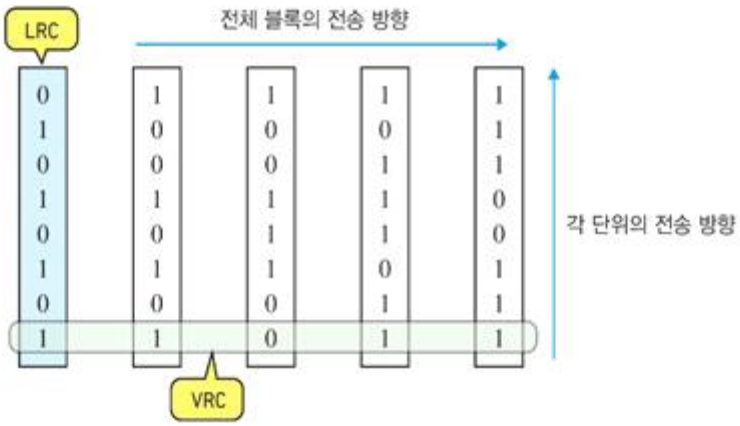
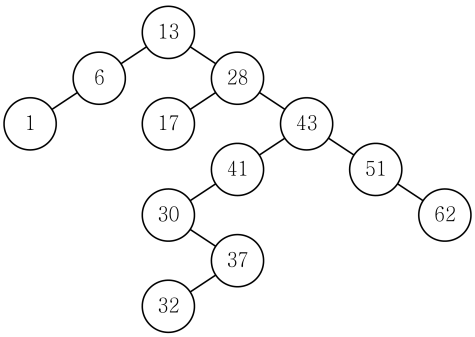


그림 9.9 수직 중복 검사와 세로 중복 검사

(4) 체크섬 : 1의 보수 또는 2의 보수를 이용하여 검사합을 만들고, 4계층(TCP, UDP)에서 사용한다.

19. 다음 이진검색트리에서 28을 삭제한 후, 28의 오른쪽 서브트리에 있는 가장 작은 원소로 28을 대치하여 만들어지는 이진검색트리에서 41의 왼쪽 자식 노드는?



① 13

② 17

③ 32

④ 37

정답 체크 :
(4) 다음의 알고리즘을 통해 37이 선택된다.
- 오른쪽 서브트리에서 후계자(30)를 찾는다.
- 후계자(30)를 찾아서 계속 왼쪽으로 이동한다.
- 후계자의 부모(41)와 후계자의 자식(37)을 연결한다. 후계자가 왼쪽에 있는 경우는 후계자의 부모(41) 왼쪽에 후계자의 오른쪽 자식(37)을 연결한다.
- 참고로 후계자가 오른쪽에 있는 경우 후계자의 부모 오른쪽에 후계자의 오른쪽 자식을 연결한다.

20. 다음은 리눅스 환경에서 fork() 시스템 호출을 이용하여 자식 프로세스를 생성하는 C 프로그램이다. 출력 결과로 옳은 것은? (단, "pid = fork();" 문장의 수행 결과 자식 프로세스의 생성을 성공하였다고 가정한다)

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/types.h>
#include<errno.h>
#include<sys/wait.h>

int main(void) {
    int i=0, v=1, n=5;
    pid_t pid;

    pid = fork();

    if( pid < 0 ) {
        for(i=0; i<n; i++) v+=(i+1);
        printf("c = %d ", v);
    } else if( pid == 0 ) {
        for(i=0; i<n; i++) v*=(i+1);
        printf("b = %d ", v);
    } else {
        wait(NULL);
        for(i=0; i<n; i++) v+=1;
        printf("a = %d ", v);
    }
    return 0;
}
```

① b = 120, a = 6
② c = 16, b = 120
③ b = 120, c = 16
④ a = 6, c = 16

정답 체크 :

(1) fork()는 자식 프로세스를 생성한다. 즉, 부모와 자식 프로세스의 분기(2개의 프로세스)가 일어난다. pid가 0보다 작으면 fork() 실패를 나타내고, 나머지는 fork() 성공을 나타낸다(0이면 자식 프로세스이고 0보다 크면 부모 프로세스이다). 부모 프로세스 부분의 wait(NULL)은 자식 프로세스가 끝날 때까지 기다리는 것을 의미하므로 b가 먼저 출력되고, 그 후에 a가 출력된다. 참고로, 해당 문제는 국회직에서 출제된 문제의 유형과 동일하다.