

1. 병렬처리 방식과 연관된 기술로 옳지 않은 것은?

- ① Pipelining
- ② Superscalar
- ③ VLIW(Very Long Instruction Word)
- ④ SMT(Simultaneous Multi-Threading)
- ⑤ Accumulator

정답 체크 :

(5) 누산기로서 계산에 사용하는 임시 레지스터이다.

오답 체크 :

- (1) 하나의 작업을 다수의 단계로 분할하여 시간적으로 중첩되게 실행함으로써 처리율을 높인다.
- (2) 파이프라인을 개선하기 위한 구조로 하드웨어를 추가하여 동시에 여러 개의 명령어를 처리한다.
- (3) 여러 opcode(operation code, 연산 코드) 필드가 있는 긴 명령어 하나에 독립적인 연산 여러 개를 정의하고 이들을 한꺼번에 내보내는 명령어 구조 집합의 종류이다.
- (4) 하드웨어 멀티스레딩을 지원하는 슈퍼스칼라 CPU의 전반적인 효율성을 개선하기 위한 기술이다(하나의 실행에 여러 개의 독립 스레드를 허용).

2. UDP(User Datagram Protocol)와 TCP(Transmission Control Protocol)에 대한 설명으로 옳지 않은 것은?

- ① TCP에는 흐름제어(flow control) 기능이 구현되어 있으나 UDP에는 흐름제어 기능이 구현되어 있지 않다.
- ② TCP에는 혼잡제어(congestion control) 기능이 구현되어 있으나 UDP에는 혼잡제어 기능이 구현되어 있지 않다.
- ③ TCP 계층에서는, TCP 하위계층에 의하여 전달된 패킷의 실제 전송 순서를 변경하지 않고 데이터 재조립을 수행한다.
- ④ TCP는 연결 지향적(connection oriented) 프로토콜이나 UDP는 연결 지향적 프로토콜이 아니다.
- ⑤ UDP는 실시간 데이터 스트리밍(streaming)에 더 적합하다.

정답 체크

(3) 하위계층이 아닌 TCP 계층에서 재조립을 수행한다.

오답 체크

- (1), (2) TCP에서 흐름제어와 혼잡제어를 지원하나 UDP에서는 지원하지 않는다.
- (4) TCP는 connection-oriented이고, UDP는 connectionless이다.
- (5) UDP는 고속 전송이 가능하기 때문에 스트리밍에 적합하다.

3. 서버 가상화(virtualization)에서 필요로 하는 하이퍼바이저(hypervisor)에 대한 설명으로 옳은 것만을 <보기>에서 모두 고르면?

〈보 기〉

- ㄱ. 하이퍼바이저는 운영체제(Operating System: OS) 없이 직접 하드웨어를 제어하거나 운영체제 상에서 응용 프로그램처럼 동작할 수 있다.
- ㄴ. 하이퍼바이저 상에서 실행되는 게스트(guest) OS는 반드시 특정 하이퍼바이저 상에서 실행될 수 있도록 수정되어야 한다.
- ㄷ. Type-2 하이퍼바이저는 특정 OS 상에서 응용 프로그램처럼 동작한다.
- ㄹ. 전가상화(full-virtualization)와 반가상화(para-virtualization) 기술 모두에서 하이퍼바이저를 필요로 한다.
- ㅁ. KVM(Kernel-based Virtual Machine)은 Type-1 하이퍼바이저로 분류될 수 있다.

- ① ㄱ, ㄴ, ㅁ
- ② ㄱ, ㄷ, ㄹ
- ③ ㄴ, ㄷ, ㅁ
- ④ ㄱ, ㄴ, ㄷ, ㄹ
- ⑤ ㄱ, ㄷ, ㄹ, ㅁ

정답 체크 :

(5)

ㄱ : 하이퍼바이저는 2개의 타입이 존재한다. Type-1은 운영체제가 필요 없고, Type-2는 운영체제가 필요하다.

ㄷ : Type-2는 운영체제 기반의 하이퍼바이저를 의미한다.

ㄹ : 전가상화(하드웨어를 완전히 가상화, guest os의 명령어 번역을 하이퍼바이저에서 처리)와 반가상화(하드웨어를 완전히 가상화하지 않음, guest os의 명령어 번역을 guest os에서 처리)는 모두 하이퍼바이저를 사용한다.

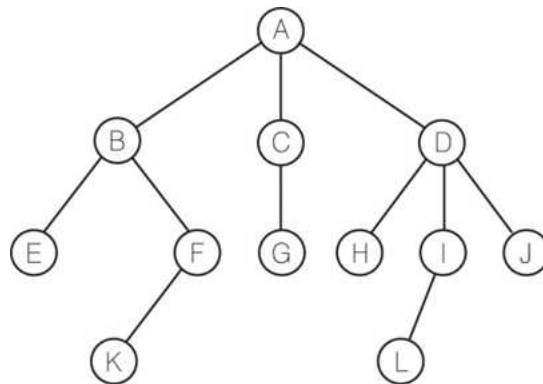
ㅁ : KVM은 하이퍼바이저 안에 완전한 리눅스 커널을 넣었기 때문에 운영체제가 필요 없는 Type-1이다.

오답 체크 :

(1), (3), (4) ㄴ : 게스트 OS는 하이퍼바이저가 관리하는 가상 머신 위에서 실행되기 때문에 특정 하이퍼바이저 상에서 실행될 수 있도록 수정할 필요가 없다. 단, 반가상화에서는 게스트 OS의 수정이 필요하다.

(2) ㅁ이 없다.

4. 깊이 우선 탐색 알고리즘을 적용하여 아래의 트리를 탐색한다고 했을 때, 방문 순서를 나타낸 것으로 옳은 것은?



- ① A-B-C-D-E-F-G-H-I-J-K-L
- ② A-B-E-F-K-C-G-D-H-I-L-J

- ③ A-B-E-K-F-C-G-D-H-I-L-J
- ④ A-E-K-F-B-G-C-H-L-I-J-D
- ⑤ A-K-L-E-F-G-H-I-J-B-C-D

정답 체크

(2) 알파벳 순으로 방문한다고 가정한다.

A -> B -> E로 방문한다. 더 이상 방문할 곳이 없으므로 B로 backtracking한다(스택을 이용).

F -> K로 방문한다. 더 이상 방문할 곳이 없으므로 F, B, A 순으로 backtracking한다.

C -> G로 방문한다. 더 이상 방문할 곳이 없으므로 C, A 순으로 backtracking한다.

D -> H로 방문한다. 더 이상 방문할 곳이 없으므로 D로 backtracking한다.

I -> L로 방문한다. 더 이상 방문할 곳이 없으므로 I, D로 backtracking한다.

J로 방문한다. 더 이상 방문할 곳이 없으므로 D, A로 backtracking한다. 더 이상 방문할 곳이 없으므로 방문을 종료한다.

5. 다음 C 프로그램의 실행 결과로 옳은 것은?

```
int fun(int n) {
    printf("%d ", n);
    if(n < 3) return 1;
    return (fun(n - 3) + fun(n - 2));
}
int main() {
    int k;
    k = fun(5);
    printf("%d\\n", k);
}
```

- ① 5 1 2 2 0
- ② 5 2 3 1 0
- ③ 5 2 3 0 1
- ④ 5 2 3 0 1 3
- ⑤ 5 2 3 0 3 1

정답 체크 :

(4) fun(5) // 5를 출력한다.

fun(2) // 2를 출력한다. 종료 조건에 의해 더 이상 출력하지 않는다(1을 반환).

fun(3) // 3을 출력한다.

fun(0) // 0을 출력한다. 종료 조건에 의해 더 이상 출력하지 않는다(1을 반환).

fun(1) // 1을 출력한다. 종료 조건에 의해 더 이상 출력하지 않는다(1을 반환).

main 함수의 printf // 3을 출력한다. fun(5)에서 3이 반환된다.

6. I/O장치에 대한 설명으로 옳지 않은 것은?

- ① 채널을 이용한 입출력 방식은 데이터의 고속성을 위해 CPU의 지속적인 간섭 없이 직접 I/O장치와 기억장치 사

이에 자료를 주고받는다.

② DMA(Direct Memory Access)를 이용한 입출력 방식은 기억장치와 입출력 모듈 간의 데이터 전송을 DMA 제어기가 처리하고 CPU가 그 동작을 관리한다.

③ I/O장치는 발생하는 이벤트를 CPU에 알리기 위해 주로 인터럽트를 발생시킨다.

④ I/O장치는 주로 인간 혹은 컴퓨터 외부와의 인터페이스(interface) 역할을 한다.

⑤ Memory-mapped I/O장치는 메모리접근 명령어를 사용해 관리한다.

정답 체크 :

(2) CPU는 DMA 제어기에게 데이터 전송 명령을 내리고 더 이상 관여하지 않는다. 즉, DMA 제어기가 동작을 관리한다.

오답 체크 :

(1) 채널(IOP)은 CPU의 간섭 없이 I/O 장치와 기억장치 사이에 자료를 주고 받는다.

(3) 폴링 방식(CPU가 I/O 장치를 계속 체크함)을 단점을 개선하기 위해 인터럽트를 사용한다.

(4) I/O 장치는 키보드, 마우스, 모니터, 프린터 등으로 인터페이스 역할을 한다.

(5) Memory-mapped I/O는 주기억장치를 입출력을 위한 기억장치로 사용하기 때문에 기존의 메모리 접근 명령어를 그대로 사용할 수 있다.

7. 다음과 같이 중위 표기법(infix notation)으로 된 식의 후위 표기법(postfix notation)으로 옳은 것은?

$$(7 + 6 / 2) / 2 + 9 * 4 / 3$$

①  $6 / 2 + 7 / 2 9 * 4 / 3 +$

②  $6 2 / 7 + 2 / 9 4 3 * / +$

③  $7 6 / 2 + 2 / 9 4 * 3 / +$

④  $7 6 2 / + 2 / 9 4 3 / + *$

⑤  $7 6 2 / + 2 / 9 4 * 3 / +$

정답 체크 :

(5)

( : 스택에 넣는다. (스택 : ()

7 : 출력한다. (결과 : 7)

+ : 스택에 넣는다. (스택 : ( +)

6 : 출력한다. (결과 : 7 6)

/ : 스택의 top(+)가 우선순위를 비교한다. /가 더 높기 때문에 스택에 넣는다. (스택 : ( + /)

2 : 출력한다. (결과 : 7 6 2)

) : 왼쪽 괄호 위의 모든 연산자를 출력한다. (결과 : 7 6 2 / +)

/ : 스택에 넣는다. (/)

2 : 출력한다. (결과 : 7 6 2 / + 2)

+ : 스택의 top(/)가 우선순위를 비교한다. top이 더 높기 때문에 top을 출력한다. (출력 : 7 6 2 / + 2 /) 더 이상 비교 대상이 없기 때문에 +를 스택에 넣는다. (스택 : +)

9 : 출력한다. (출력 : 7 6 2 / + 2 / 9)

\* : 스택의 top(+)가 우선순위를 비교한다. \*가 더 높기 때문에 스택에 넣는다. (스택 : + \*)

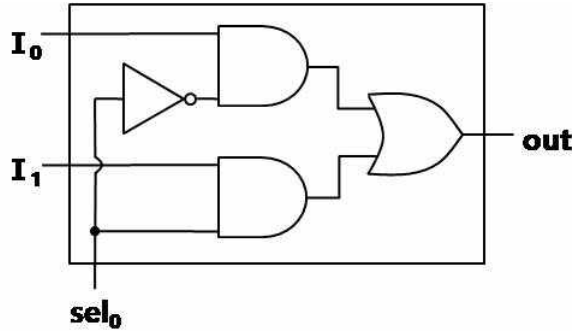
4 : 출력한다. (출력 : 7 6 2 / + 2 / 9 4)

/ : 스택의 top(\*)가 우선순위를 비교한다. top가 동일하기 때문에 top을 출력한다. (출력 : 7 6 2 / + 2 / 9 4 \*) 그 다음 top(+)가 비교한다. /가 더 높기 때문에 스택에 넣는다. (스택 : + /)

3 : 출력한다. (출력 : 7 6 2 / + 2 / 9 4 \* 3)

더 이상의 수식이 없으므로 스택에 있는 모든 연산자를 출력한다. (출력 : 7 6 2 / + 2 / 9 4 \* 3 / +)

8. 다음 논리회로의 기능에 해당하는 것은?



- ① 멀티플렉서
- ② 플립플롭
- ③ 덧셈기
- ④ 곱셈기
- ⑤ 감산기

정답 체크 :

(1) 멀티플렉서는  $2^n \times 1$  (입력 $\times$ 출력)의 구조를 가진다.

오답 체크 :

- (2) 플립플롭은 래치(상태를 유지)와 클럭으로 구성되어, 1비트를 저장한다.
- (3) 덧셈기는 XOR과 AND로 구성된다.
- (4) 곱셈기는 여러개의 덧셈기로 구성된다.
- (5) 감산기는 XOR, AND, NOT으로 구성된다.

9. GPGPU(General-Purpose computing on Graphics Processing Units) 기술에 대한 설명으로 옳지 않은 것은?

- ① GPU에서 그래픽 연산 이외의 목적을 가진 프로그램을 실행할 수 있도록 해주는 기술을 지칭한다.
- ② CPU 기기에서의 실행을 위해 컴파일 된 모든 응용 프로그램은 GPGPU 기기에서 실행될 수 있다.
- ③ 하나의 GPGPU 기기에는 많은 수의 단순 ALU(Arithmetic Logic Unit)가 있어 높은 수준의 병렬처리가 가능하고, 이로 인해 일반 CPU에 비하여 프로그램의 병렬 처리속도가 높아진다.
- ④ OpenCL(Open Computing Language)을 이용하여 프로그래밍할 경우 다양한 제조사의 GPGPU 기기에서 실행 가능한 프로그램을 작성할 수 있다.
- ⑤ 병렬처리가 불가능한 어떤 프로그램을 GPGPU에서 실행하려 한다면 성능향상을 기대하기 어렵다.

정답 체크 :

(2) CPU 기기에서 동작하는 것이 GPGPU에서 동작하지 않는 경우가 존재한다. 예를 들어, 64비트 부동소수점 값은 CPU에서는 일반적이지만 GPGPU 중에는 지원하지 않는 경우가 있다.

오답 체크 :

- (1) 일반적으로 컴퓨터 그래픽스를 위한 계산만 맡았던 그래픽 처리 장치(GPU)를 전통적으로 중앙 처리 장치(CPU)가 맡았던 응용 프로그램들의 계산에 사용하는 기술이다.
- (3) 일반 CPU에 비해 병렬도가 높다(높은 수준의 병렬 처리가 가능).
- (4) 개방형 범용 병렬 컴퓨팅 프레임워크로서 CPU, GPU, DSP 등의 프로세서로 이루어진 이종 플랫폼에서 실행

되는 프로그램을 작성할 수 있게 해 준다.

(5) (3)번과 같은 개념으로 병렬도가 낮은 프로그램은 일반 CPU에 비해 성능향상을 기대하기 어렵다.

10. 데이터베이스의 특성에 대한 설명으로 옳은 것만을 <보기>에서 모두 고르면?

〈 보 기 〉

- ㄱ. 실시간 접근성: 데이터의 검색이나 조작을 요구하는 수시적이고 비정형적인 질의에 대하여 즉시 응답할 수 있어야 한다.
- ㄴ. 계속적인 변화: 데이터베이스의 상태는 정적이 아니고 동적이므로 현재의 정확한 데이터를 유지해야 한다.
- ㄷ. 동시공유: 데이터베이스는 동시에 여러 사용자가 접근할 수 있어야 한다.
- ㄹ. 주소에 의한 참조: 데이터베이스 내에 있는 데이터 레코드들은 주소에 의해 참조된다.

- ① ㄱ, ㄴ
- ② ㄴ, ㄷ
- ③ ㄷ, ㄹ
- ④ ㄱ, ㄴ, ㄷ
- ⑤ ㄴ, ㄷ, ㄹ

정답 체크 :

(4)

ㄱ : 사용자의 데이터 요구에 실시간으로 응답한다.

ㄴ : 데이터의 계속적인 삽입, 삭제, 수정을 통해 현재의 정확한 데이터를 유지한다.

ㄷ : 서로 다른 데이터의 동시 사용뿐만 아니라 같은 데이터의 동시 사용도 지원한다.

오답 체크 :

(1) ㄷ이 없다.

(2) ㄱ이 없다.

(3), (5) ㄹ : 주소에 의한 참조가 아닌 내용에 의한 참조를 사용한다.

11. 스택(stack)은 자료구조에서 말하는 구조체 외에도, 메모리에 지정된 특정 영역을 말하며 프로그램 실행에서 항상 사용된다. 이러한 스택에 대한 설명으로 옳지 않은 것은?

- ① CPU마다 스택포인터 용도로 사용하는 레지스터가 있다.
- ② 스택은 LIFO(Last-In First-Out)구조를 가지고 있다.
- ③ 버퍼 오버플로우(buffer overflow) 보안공격에 이용되기도 한다.
- ④ 운영체제의 문맥교환(context switching)을 할 때 사용된다.
- ⑤ CPU 레지스터의 내용을 스택에 저장할 수 없다.

정답 체크 :

(5) PC 등의 레지스터의 내용을 스택에 저장할 수 있다.

오답 체크 :

(1) 스택의 주소를 가리키는 레지스터가 존재한다.

(2) 스택은 LIFO 구조이다.

- (3) 스택 버퍼 오버플로우 공격에 이용된다.
- (4) 문맥교환을 위해서는 프로세스의 정보를 주기억장치와 스택에 저장해야 한다.

12. 관계형 데이터 모델(relational data model)에서 릴레이션에 대한 설명으로 옳지 않은 것은?

- ① 릴레이션을 구성하는 속성 값은 서브속성으로 다시 세분화할 수 있다.
- ② 데이터베이스 스키마(database schema)는 정적이고, 데이터베이스 인스턴스(database instance)는 동적이다.
- ③ 릴레이션 안에는 똑같은 튜플이 존재할 수 없다.
- ④ 릴레이션의 속성 사이에 순서는 의미가 없다.
- ⑤ 릴레이션의 속성의 명칭은 유일하나, 속성을 구성하는 값은 같을 수 있다.

정답 체크 :

- (1) 속성은 원자성을 가진다. 즉, 서브속성으로 세분화할 수 없다.

오답 체크 :

- (2) 스키마는 고정되었고, 인스턴스는 입력에 따라 계속 바뀐다.
- (3) 튜플의 유일성이다.
- (4) 속성의 무순서이다.
- (5) 속성은 도메인을 의미하고, 도메인을 구성하는 값은 동일한 값이 존재할 수 있다.

13. IPv4 주소체계 기반의 어떤 네트워크 상에서 두 컴퓨터 A, B가 각각 192.168.0.1과 192.168.0.65의 주소를 사용할 때, 이 두 컴퓨터가 서로 다른 서브넷(subnet) 상에 존재하기 위해 사용해야 하는 서브넷 마스크(subnet mask)로 옳은 것은?

- ① 0.0.0.0
- ② 255.255.255.0
- ③ 255.255.255.192
- ④ 255.255.255.128
- ⑤ 255.255.128.0

정답 체크 :

- (3) 서브넷을 255.255.255.192로 사용하면 각 서브넷에 64개의 IP를 사용할 수 있다. 그러므로 192.168.0.0부터 192.168.0.63은 같은 서브넷을 사용하고, 192.168.0.64부터 192.168.0.127은 같은 서브넷을 사용한다.

14. 스래싱(thrashing)에 대한 설명으로 옳은 것은?

- ① 두 개 이상의 작업이 서로 상대방의 작업이 끝나기만을 기다리고 있기 때문에 결과적으로 아무것도 완료되지 못하는 현상을 의미한다.
- ② CPU 버스트가 짧은 프로세스에게 우선순위를 항상 부여한다면, 상대적으로 CPU 버스트가 긴 프로세스가 계속해서 지연되는 것을 의미한다.
- ③ CPU 버스트가 긴 프로세스가 CPU를 양보할 때까지 다른 모든 프로세스들이 기다리는 것을 의미한다.
- ④ 페이지 부재가 너무 자주 일어나 프로세스가 실행에 보내는 시간보다 페이지 교체에 더 많은 시간을 소비하는 현상을 의미한다.
- ⑤ CPU가 프로그램을 실행하고 있을 때, 입출력 하드웨어 등의 장치나 예외 상황이 발생하여 처리가 필요한 경우 CPU에게 알려 처리할 수 있도록 하는 것을 의미한다.

정답 체크 :

(4) 페이지 교환이 계속 일어나는 현상이다.

오답 체크 :

- (1) 교착 상태를 나타낸다.
- (2) 기아 상태를 나타낸다.
- (3) 호위 효과를 나타낸다.
- (5) 인터럽트를 나타낸다.

15. 다음 중 양자컴퓨팅(quantum computing)에 대한 설명으로 옳은 것만을 <보기>에서 모두 고르면?

〈 보 기 〉

- ㄱ. 양자 얽힘(entanglement), 중첩(superposition)과 같은 양자역학 현상을 이용한다.
- ㄴ. 0, 1 또는 0과 1의 상태를 동시에 가질 수 있는 큐비트(Qbit)가 계산의 기본 단위이다.
- ㄷ. QKD(Quantum Key Distribution) 시스템에서는 양자 얽힘 현상을 이용하지 않는다.
- ㄹ. 이산로그(discrete logarithm) 문제를 다항시간의 복잡도로 풀 수 있는 방법을 제공한다.
- ㅁ. 소인수분해(integer factorization) 문제를 다항시간(polynomial time)의 복잡도로 풀 수 있는 방법이 존재하여 기존의 모든 NP(Non-deterministic Polynomial)문제를 다항시간 내에 풀 수 있다.

- ① ㄱ, ㄴ, ㄷ
- ② ㄱ, ㄴ, ㄹ
- ③ ㄴ, ㄷ, ㅁ
- ④ ㄱ, ㄴ, ㄹ, ㅁ
- ⑤ ㄴ, ㄷ, ㄹ, ㅁ

정답 체크 :

(2)

ㄱ : 양자컴퓨팅은 양자역학 현상을 이용한다.

ㄴ : 양자컴퓨팅은 큐비트를 기본 단위로 사용한다.

ㄹ : 이산 로그 문제(P 문제)를 다항시간의 복잡도로 풀 수 있다(일정 시간 안에 풀 수 있다).

오답 체크 :

(1), (3), (4), (5)

ㄷ : QKD는 양자 암호로서 양자역학 현상(양자 얽힘)을 이용한다.

ㅁ : 소인수분해 문제(P 문제)를 다항시간의 복잡도로 풀 수 있지만, 모든 NP 문제를 다항시간 내에 풀 수 있다는 것은 확실히 증명되지 않았다.

Tip! P 문제와 NP 문제

P 문제는 결정 문제들 중에서 쉽게 풀리는 것을 모아 놓은 집합이다. 어떤 결정 문제가 주어졌을 때, 다항식(Polynomial) 시간 이내에 그 문제의 답을 YES와 NO 중의 하나로 계산해낼 수 있는 알고리즘이 존재한다면, 그 문제는 P 문제에 해당된다. NP 문제는 결정 문제들 중에서 적어도 검증은 쉽게 할 수 있는 것을 모아 놓은 집합으로도 정의할 수 있다. 정확히 말하면, 어떤 결정 문제의 답이 YES일 때, 그 문제의 답이 YES라는 것을 입증하는



힌트가 주어지면, 그 힌트를 사용해서 그 문제의 답이 정말로 YES라는 것을 다항식 시간 이내에 확인할 수 있는 문제가 바로 NP 문제에 해당된다. NP에는 NP-hard와 NP-complete가 존재한다. NP-hard는 모든 경우의 수를 전부 확인해보는 방법 이외에는 정확한 답을 구할 수 있는 방법이 없는 문제들을 뜻한다. NP-hard는 NP에 속하는 모든 판정 문제를 다항 시간에 다대일 환산할 수 있는 문제들의 집합이므로 다항 시간내에 해결이 가능하다. NP-complete는 NP-hard이지만 다항 시간 내에 해결할 수 없는 문제를 나타낸다. 예를 들어, Travelling Salesman Problem 등을 들 수 있다. NP-complete 문제를 해결하기 위해 근사 알고리즘을 사용한다.

16. IEEE 802.11 무선랜 표준에 대한 설명으로 옳지 않은 것은?

- ① MIMO(Multi-Input Multi-Output) 등과 같은 다중 안테나 기술이 적용된 경우, 일반적으로 사용하는 안테나 개수가 증가함에 따라 전송 속도가 높아진다.
- ② 3GPP LTE 표준보다 전송 속도(data rate)가 빠를 수 있다.
- ③ OFDM(Orthogonal Frequency-Division Multiplexing) 기법은 IEEE 802.11b 규격의 기기에도 사용이 가능하며, 최고 전송 속도를 11Mbps까지 높일 수 있다.
- ④ 특정 기기가 사용하는 채널의 대역폭(bandwidth)은 가변적일 수 있다.
- ⑤ 다수의 채널(channel)을 사용하여 근접한 거리에 있는 서로 다른 두 쌍의 기기가 서로 간섭을 받지 않고 동시에 통신할 수 있다.

정답 체크 :

(3) IEEE 802.11b는 DSSS 기법을 사용한다.

오답 체크 :

- (1) 다중 안테나 기술을 사용하면 안테나 개수가 증가함에 따라 속도가 높아진다.
- (2) 최신 무선랜은 10Gbps를 지원한다(LTE의 속도가 1Gbps라고 가정).
- (4) 최고 속도는 정해져있지만 전송 속도(대역폭)는 상황에 따라 가변적이다.
- (5) 무선랜은 여러 개의 채널이 존재하여 근접한 거리의 기기가 서로 간섭을 받지 않는다.

17. 객체지향(object-oriented) 언어인 C++에 대한 설명으로 옳지 않은 것만을 <보기>에서 모두 고르면?

〈 보 기 〉

- ㄱ. 다중 상속(multiple inheritance)이 불가능하다.
- ㄴ. 함수 오버로딩(overloading)이 가능하나 한 클래스(class) 내에서 동일한 이름을 가진 다수의 함수는 정의될 수 없다.
- ㄷ. 클래스를 통해 데이터 캡슐화(encapsulation)와 추상화(abstraction)를 지원한다.
- ㄹ. 다형성(polymorphism)과 상속(inheritance)을 지원한다.
- ㅁ. 다형성(polymorphism)의 지원을 위해 동적 바인딩(dynamic binding)을 사용한다.

- ① ㄱ, ㄴ
- ② ㄱ, ㄴ, ㅁ
- ③ ㄱ, ㄷ, ㄹ
- ④ ㄴ, ㄹ, ㅁ
- ⑤ ㄷ, ㄹ, ㅁ

정답 체크 :

(1)

ㄱ : 다중 상속이 가능하다. (자바는 다중 상속이 불가능하다)

ㄴ : 오버로딩을 통해 한 클래스 내에서 동일한 이름을 가진 다수의 함수를 정의할 수 있다.

오답 체크 :

(2), (3), (4), (5)

ㄷ : C++은 객체지향 언어의 특성인 캡슐화와 추상화를 지원한다.

ㄹ : C++은 객체지향 언어의 특성인 다형성과 상속을 지원한다.

ㅁ : 다형성의 지원을 위해 동적 바인딩(실행 시간에 함수 호출을 결정)을 사용한다.

18. 다음의 C 언어로 작성된 프로그램에서 calc함수의 실행시간(또는 실행 단계 수)를 점근 표기법으로 표시했을 때 옳은 것은?

```
int calc(int n) {
    int i, j, ret = 0;
    if (n <= 0)
        return 0;
    for (i = 1; i <= n; i++)
        ret += 1;
    return ret + calc(n - 1);
}
```

①  $O(\log_2 n)$

②  $O(n \log_2 n)$

③  $O(n)$

④  $O(n^2)$

⑤  $O(n^3)$

정답 체크 :

(4) 실행 시간은 다음과 같이 계산이 가능하다.

calc(n) // n번 연산이 수행된다고 가정하자(for 문에서 발생하는 연산만 계산해 포함).

calc(n-1) // n-1번 연산이 수행된다고 가정하자.

...

cal(1) // 1번 연산이 수행된다고 가정하자.

이들의 연산을 모두 더하면 다음과 같다.

$$n + (n-1) + (n-2) + \dots + 1 = n(n+1)/2 = (n^2+n)/2$$

빅오에서는 최고차 항만을 고려하므로  $O(n^2)$ 이 된다.

19. 초기 페이지 프레임은 비어 있다고 가정할 때, 3개의 페이지 프레임을 가진 기억장치에서 다음과 같이 페이지 참조열이 주어졌다. LRU(Least Recently Used) 알고리즘을 적용했을 때 발생하는 페이지 부재의 수는?

```
2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 5, 3
```

① 8

② 9

③ 10

④ 11

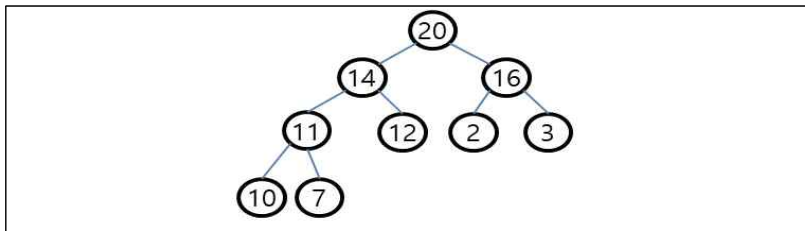
⑤ 12

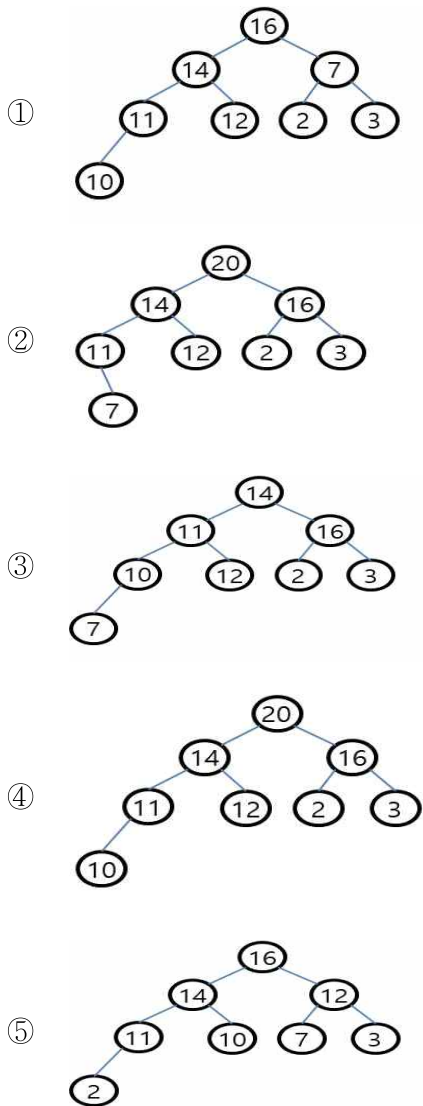
정답 체크 :

(3) 주어진 조건으로 LRU 알고리즘을 적용하면 다음과 같이 10번의 페이지 부재가 발생한다.

2	3	4	2	1	5	6	2	1	2	3	5	3
2	2	2	2	2	2	6	6	6	6	3	3	3
	3	3	3	1	1	1	2	2	2	2	2	2
		4	4	4	5	5	5	1	1	1	5	5
fault	fault	fault		fault	fault	fault	fault	fault		fault	fault	

20. 아래의 최대힙(max heap)에서 노드를 한 개 삭제하는 연산을 실행하였을 때의 결과로 옳은 것은?





정답 체크 :

(1) 최대힙에서 삭제 연산은 다음과 같다.

루트 20을 삭제한다.

7을 루트로 올린다.

7의 자식 중 큰 자식(16)가 최대힙 조건(부모가 크거나 같아야 함)을 비교한다.

7과 16을 교환한다.

7은 자식(2 또는 3)보다 크므로 최대힙 조건을 만족한다.

삭제 연산을 마친다.