

컴퓨터일반

문 1. UP(Unified Process)의 네 단계 중 아키텍처 결정을 위한 설계 작업과 분석 작업의 비중이 크고, 시스템 구성에 관련된 위험 요소를 식별하고 이를 완화하는 데 중점을 두는 단계는?

- ① 도입(inception) ② 상세(elaboration)
③ 구축(construction) ④ 이행(transition)

문 2. 문법 G가 다음과 같을 때 S_1 으로부터 생성할 수 없는 것은?

$$G: S_1 \rightarrow 0S_2 \quad S_1 \rightarrow 0 \\ S_2 \rightarrow 0S_2 \quad S_2 \rightarrow 1$$

- ① 0 ② 00
③ 01 ④ 001

문 3. 데이터 통신의 표준참조모델인 OSI모델의 각 계층에 대한 설명으로 옳지 않은 것은?

- ① 물리 계층은 송수신 시스템의 연결에서 전송 매체의 종류, 송수신되는 신호의 전압 레벨 등을 정의한다.
② 네트워크 계층은 송수신 컴퓨터의 응용프로그램 간 송수신되는 데이터의 구문과 의미에 관련된 기능으로 변환, 암호화, 압축을 수행한다.
③ 전송 계층은 연결된 네트워크의 기능이나 특성에 영향을 받지 않고 오류제어와 흐름제어 기능을 수행하여 신뢰성 있는 데이터 전송을 보장하는 것으로, 프로토콜은 TCP, UDP 등이 있다.
④ 응용 계층은 최상위 계층으로 프로토콜은 FTP, HTTP 등이 있다.

문 4. 컴퓨터 이미지에 대한 설명으로 옳지 않은 것은?

- ① 벡터 방식은 이미지의 크기가 커지면 저장 용량도 커진다.
② GIF와 JPG는 비트맵 방식의 파일 형식이다.
③ 상세한 명암과 색상을 표현하는 사진에 적합한 방식은 비트맵 방식이다.
④ 벡터 방식은 이미지를 확대, 축소, 회전하더라도 이미지의 품질에 영향을 주지 않는다.

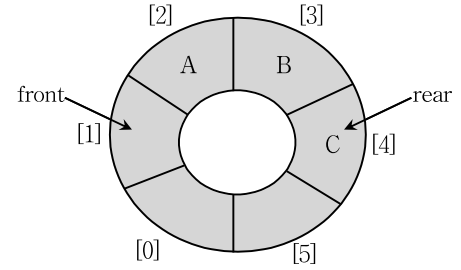
문 5. 음수를 2의 보수로 표현할 때, 십진수 -66을 8비트 이진수로 변환한 값은?

- ① 10111101₂ ② 10111110₂
③ 11000010₂ ④ 01000001₂

문 6. 디자인 패턴에 대한 설명으로 옳지 않은 것은?

- ① 일반적으로 디자인 패턴을 이용하면 좋은 설계나 아키텍처를 재사용하기 쉬워진다.
② 패턴은 사용 목적에 따라서 생성 패턴, 구조 패턴, 행위 패턴으로 분류할 수 있다.
③ 생성 패턴은 빌더(builder), 추상 팩토리(abstract factory) 등을 포함한다.
④ 행위 패턴은 가교(bridge), 적응자(adapter), 복합체(composite) 등을 포함한다.

문 7. 다음 그림과 같은 원형 큐에 한 객체를 입력하는 알고리즘에 대해 의사코드(pseudo code)를 순서대로 바르게 나열한 것은? (단, 객체는 rear 쪽에 입력되고 front 쪽에서 출력되며, M은 큐의 크기를 나타내는 정수이다)



- ㄱ. 큐가 공백 상태인지 검사: (front == rear)
ㄴ. front 값을 1 증가: front = (front+1)%M
ㄷ. 큐가 포화 상태인지 검사: (front == rear)
ㄹ. 객체를 rear 위치에 입력
ㅁ. rear 값을 1 증가: rear = (rear+1)%M

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

문 8. 다음 내용에 적합한 매체 접근 제어(MAC) 방식은?

- IEEE 802.11 무선 랜에서 널리 사용된다.
○ 채널이 사용되지 않는 상태를 감지하더라도 스테이션은 임의의 백오프 값을 선택하여 전송을 지연시킨다.
○ 수신 노드는 오류 없이 프레임 수신하면 수신 확인 ACK 프레임을 전송한다.

- ① GSM ② CSMA/CA
③ CSMA/CD ④ LTE

문 9. 다음 C 프로그램의 출력 값은?

```
#include <stdio.h>

int main()
{
    int darr[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int sum1, sum2;

    sum1 =>(*darr + 1) +>(*darr + 2);
    sum2 = *darr[1] + *darr[2];
    printf("%d, %d", sum1, sum2);
}
```

- ① 3, 5 ② 5, 5
③ 5, 11 ④ 11, 5

문 10. 스레드(thread)에 대한 설명으로 옳지 않은 것은?

- ① 스레드는 자기만 접근할 수 있는 스레드별 데이터를 갖지 않는다.
- ② 단일 프로세스에 포함된 스레드들은 프로세스의 자원을 공유할 수 있다.
- ③ 멀티프로세서 환경에서는 각각의 스레드가 다른 프로세서에서 수행될 수 있다.
- ④ Pthread는 스레드 생성과 동기화를 위해 POSIX가 제정한 표준 API이다.

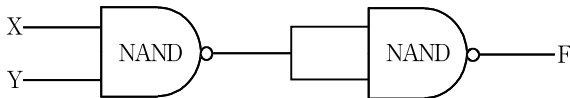
문 11. RAID에 대한 설명으로 옳은 것은?

- ① RAID 레벨 1은 패리티를 이용한다.
- ② RAID 레벨 0은 디스크 미러링을 이용한다.
- ③ RAID 레벨 0과 RAID 레벨 1을 조합해서 사용할 수 없다.
- ④ RAID 레벨 5는 패리티를 모든 디스크에 분산시킨다.

문 12. IoT(Internet of Things)기기의 확산 등으로 예상되는 인터넷 주소의 고갈 문제를 해결하기 위한 것은?

- ① HTTPS
- ② IPv4
- ③ IPv6
- ④ Common Gateway Interface

문 13. 다음 논리회로의 부울식으로 옳은 것은?



- ① $F = XY$
- ② $F = (XY)'$
- ③ $F = X'Y$
- ④ $F = XY + (XY)'$

문 14. 네트워크의 전송 데이터 오류 검출에 대한 설명으로 옳지 않은 것은?

- ① 체크섬(checksum)은 1의 보수 방법을 사용한다.
- ② 순환중복검사(CRC)는 모듈로-2 연산을 주로 사용한다.
- ③ 전송할 데이터에 대한 중복 정보를 활용하여 오류를 검출한다.
- ④ 단일 패리티 비트를 사용하는 패리티 검사는 홀수 개의 비트에 오류가 발생하면 오류를 발견할 수 없다.

문 15. 유닉스 운영체제의 커널에 속하지 않는 것은?

- ① 스케줄러
- ② 파일 관리자
- ③ 메모리 관리자
- ④ 윈도우 관리자

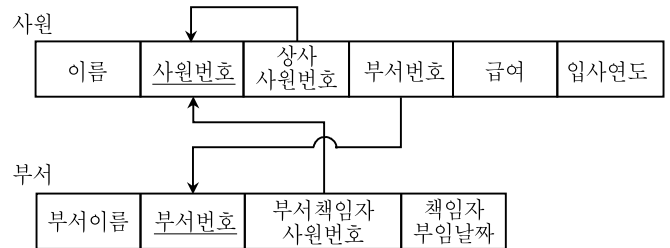
문 16. 안드로이드에 대한 설명으로 옳지 않은 것은?

- ① 안드로이드는 구글이 중심이 되어 개발하는 휴대 단말기용 플랫폼이다.
- ② 일반적으로 안드로이드 애플리케이션의 네 가지 구성요소는 액티비티, 방송 수신자, 서비스, 콘텐츠 제공자이다.
- ③ 보안, 메모리 관리, 프로세스 관리, 네트워크 관리 등 핵심 서비스는 리눅스에 기초하여 구현되었다.
- ④ 콘텐츠 제공자는 UI 컴포넌트를 화면에 표시하고, 시스템이나 사용자의 반응을 처리할 수 있다.

문 17. 컴퓨터 시스템에 대한 설명으로 옳은 것은?

- ① 임베디드 시스템은 특정 기능을 수행하기 위해 설계된 컴퓨터 하드웨어와 소프트웨어 및 추가적인 기계 혹은 기타 부품들의 결합체이다.
- ② 클러스터 컴퓨팅 시스템에 참여하는 컴퓨터들은 다른 이웃 노드와 독립적으로 동작하고 상호 연결되어 협력하지 않는다.
- ③ 불균일 기억장치 액세스(NUMA) 방식은 병렬 방식 중 가장 오래되었고, 여전히 가장 널리 사용된다.
- ④ Flynn의 분류에 따르면, MISD는 여러 프로세서들이 서로 다른 명령어들을 서로 다른 데이터들에 대하여 동시에 실행하는 것이다.

문 18. 다음 데이터베이스 스키마에 대한 설명으로 옳지 않은 것은? (단, 밑줄이 있는 속성은 그 릴레이션의 기본키를, 화살표는 외래키 관계를 의미한다)



- ① 외래키는 동일한 릴레이션을 참조할 수 있다.
- ② 사원 릴레이션의 부서번호는 부서 릴레이션의 부서번호 값 중 하나 혹은 둘이어야 한다는 제약조건은 참조무결성을 의미한다.
- ③ 신입사원을 사원 릴레이션에 추가할 때 그 사원의 사원번호는 반드시 기존 사원의 사원번호와 같지 않아야 한다는 제약조건은 제1정규형의 원자성과 관계있다.
- ④ 부서 릴레이션의 책임자부임날짜는 반드시 그 부서책임자의 입사연도 이후이어야 한다는 제약조건을 위해 트리거(trigger)와 주장(assertion)을 사용할 수 있다.

문 19. 명령어 파이프라이닝의 4단계에 속하지 않는 것은?

- ① 인터럽트
- ② 명령어 실행
- ③ 명령어 인출
- ④ 명령어 해독

문 20. 다음 C 프로그램의 출력 값은?

```
#include <stdio.h>

int recur(int a, int b)
{
    if (a <=1)
        return a * b;
    else
        return a * recur(a-1, b+1) + recur(a-1, b);
}

int main()
{
    int a=3, b=2;

    printf("%d\n", recur(a, b));
}
```

- ① 24
- ② 30
- ③ 41
- ④ 52