

# 119 회 기출풀이

## 컴퓨터시스템응용기술사

- KPC 기술사회 -



### 교육 문의 및 상담 : 한승연



- Tel : 02) 724-1831/1223
- Fax : 02) 724-1875

- Email : syhan@kpc.or.kr

- Web Site : [www.kpc.or.kr](http://www.kpc.or.kr)

[cafe.naver.com/81th](http://cafe.naver.com/81th)





## 120 회 합격대비 심화반 신청 안내

### [토요일 명품심화반]

- 단합반(SPP 반) (안경환 PE @ KPC) : 정규심화 9. 28. 개강
- FB(Future Builders) (강희석 PE @ KPC) : 정규심화 9. 28. 개강
- 열 정 반 (박상욱 PE @ KPC) : 정규심화 9. 28. 개강
- 정 주 행 (서정훈 PE @ KPC) : 정규심화 9. 28. 개강
- ITPE Makers (박제일 PE @ KPC) : 정규심화 9. 28. 개강
- MP 필통반 (구환회 PE @ KPC) : 정규심화 9. 28. 개강
- 공 감 반 (공수재 PE @ KPC) : 조기심화 8. 31. 개강

### [일요일 명품심화반]

- T.O.P 반 (유술사 PE @ KPC) : 조기심화 8. 25. 개강
- NS 반 (강정배 PE&박주형 PE @ 강남아지트): 조기심화 9. 1. 개강

### [평일 명품심화반]

- 강남평일야간반 (강정배 PE&전일 PE&박찬렬 PE @ 강남아지트/화,금):  
조기심화 9. 3. 개강

~~ KPC 홈페이지에서 신청 가능합니다. ~~



## 국가기술자격 기술사 시험문제

기술사 제 119 회

제 1 교시 (시험시간: 100 분)

분야	정보통신	종목	컴퓨터시스템응용기술사	수험 번호		성명	
----	------	----	-------------	-------	--	----	--

※ 다음 문제 중 10 문제를 선택하여 설명하시오. (각 10 점)

1. SVM (Support Vector Machine)
2. 인공 신경망의 오류 역전파 (Backpropagation) 알고리즘
3. 오버라이딩(Overriding)으로 함수를 재정의 하는 이유
4. HTTP 3.0
5. Biba Integrity Model
6. 모니터(Monitor) 프로세스 동기화 기법
7. Ad-hoc Network Routing Protocol
8. OAuth 2.0
9. 포그 컴퓨팅 (Fog Computing)
10. 어플라이언스(Appliance) 서버
11. FeRAM/STT-MRAM/PRAM
12. SCSI RDAM (Rmote Direct Memory Access)
13. WAF(Web Application FireWall)

Notes

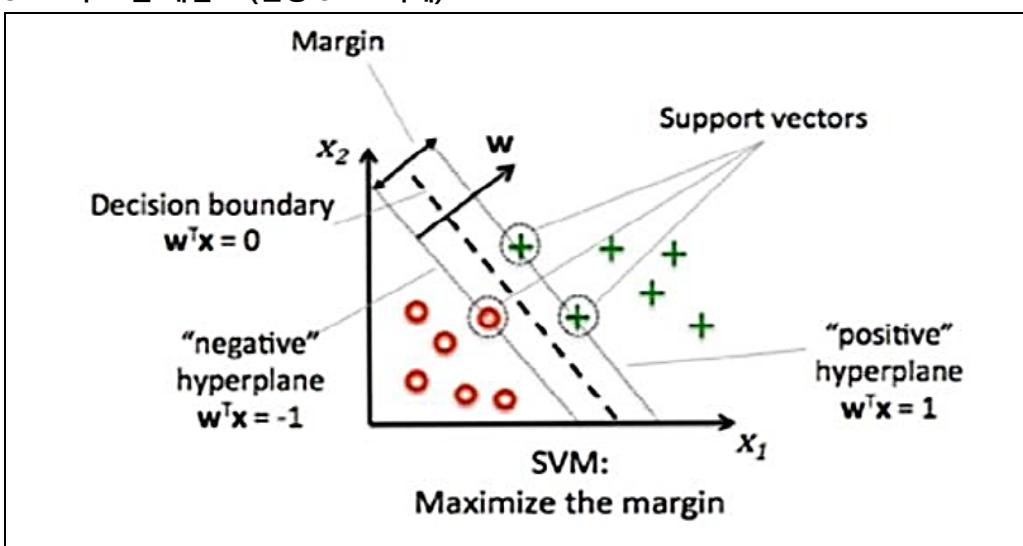
1	SVM(Support Vector Machine)
문제	SVM(Support Vector Machine)
도메인	인공지능
정의	데이터가 사상된 공간에서 경계선과 가장 근접한 데이터(Support Vector)간의 거리가 가장 큰 경계를 식별하는 지도학습 기반 분류 알고리즘
키워드	커널 평선, 서포트 벡터, 마진, 초평면 ( <a href="#">커서마초</a> )
출제의도분석	기계학습/인공지능 활용의 증대에 따라 지도학습 기반 분류 기법의 대표적인 기법인 SVM에 대한 이해
답안작성 전략	키워드 위주의 핵심 설명. SVM 활용 제시+ $\alpha$
참고문헌	113 대비 합숙 4 일차 KT AI 아카데미, "머신러닝을 위한 기초 수학" <a href="https://m.blog.naver.com/PostView.nhn?blogId=laonple&amp;logNo=220845107089&amp;p_referrer=https%3A%2F%2Fwww.google.com%2F">https://m.blog.naver.com/PostView.nhn?blogId=laonple&amp;logNo=220845107089&amp;p_referrer=https%3A%2F%2Fwww.google.com%2F</a>
풀이 기술사님	김강용 PE (제 118 회 정보관리기술사 / taoloan@nate.com)

## 1. 초평면을 이용한 데이터의 분류, SVM(Support Vector Machine) 개요

- 데이터가 사상된 공간에서 경계선과 가장 근접한 데이터(Support Vector)간의 거리가 가장 큰 경계를 식별하는 지도학습 기반 분류 알고리즘
- 특징 : 지도학습, 분류기법, 회귀분석 활용, 차원의 저주 회피

## 2. SVM의 모델 개념도 및 구성요소

### 가. SVM의 모델 개념도 (선형 SVM 사례)



- 초평면(Optimal Hyperplane)을 구하여 최대 마진 기준으로 데이터를 2개의 그룹으로 분류

### 나. SVM(Support Vector Machine) 구성요소

구성요소	설 명	특징
Optimal Hyperplane	데이터 분류의 기준이 되는 분류 경계선	Positive, Negative
Support	학습 데이터 중에서 분류 경계에 가장 가까운 곳에	초평면과 가장

Notes

Vector	위치한 데이터	가까운 위, 아래 점
Margin	분류 경계에 가장 가까운 데이터로부터 분류 경계까지의 거리	hard-margin, soft-margin
Kernel Function	비선형 SVM에서 고차원 특징 공간으로 사상하는 함수	차원 축소

- 슬랙 변수의 적용여부에 따라 하드 마진과 소프트 마진 방법 존재
- 선형 SVM과 비선형 SVM이 존재하며, 다양한 분야의 분류 기법으로 활용됨

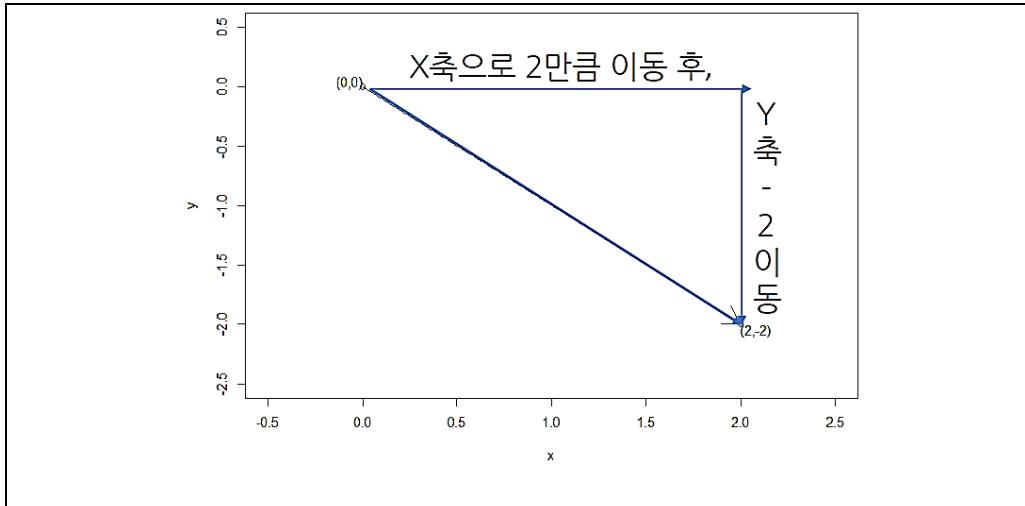
### 3. SVM의 활용 방안

활용	설명
텍스트 분류	하이퍼텍스트 분류, 스팸 분류, 긍정/부정어 분류
이미지 분류	손글씨 인식 및 이미지 특징 인지
의료 정보	DNA 단백질 효소 정보 분류

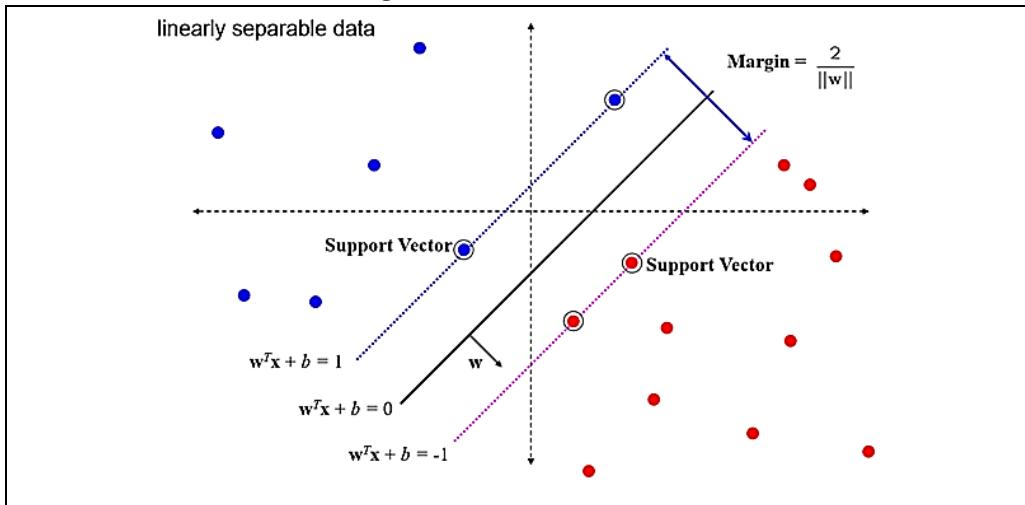
"끝"

## ※ [참고자료] 벡터의 길이(Norm)의 개념과 SVM의 Margin 계산 활용

- ✓ 벡터  $X$ 의 노름(Norm) = 벡터  $X$ 의 길이  $\|X\|$  : 피타고라스의 정리 사용( $a^2 + b^2 = c^2$ )  
 벡터  $\begin{bmatrix} 2 \\ -2 \end{bmatrix}$ 를  $A$ 라고 두고,  $A$ 의 길이  $\|A\|$ .  
 $\|A\| = \sqrt{(2^2 + 2^2)} = \sqrt{8}$



- ✓ Norm을 이용하여 SVM에서 Margin 계산



- 마진의 폭은  $2/\|w\|$  이며, 마진이 최대가 되려면  $\|w\|$ 가 최소가 되도록 최적화 수행 필요하며  $2/\|w\|$  최소화는 오차제곱 기법을 주로 사용함.

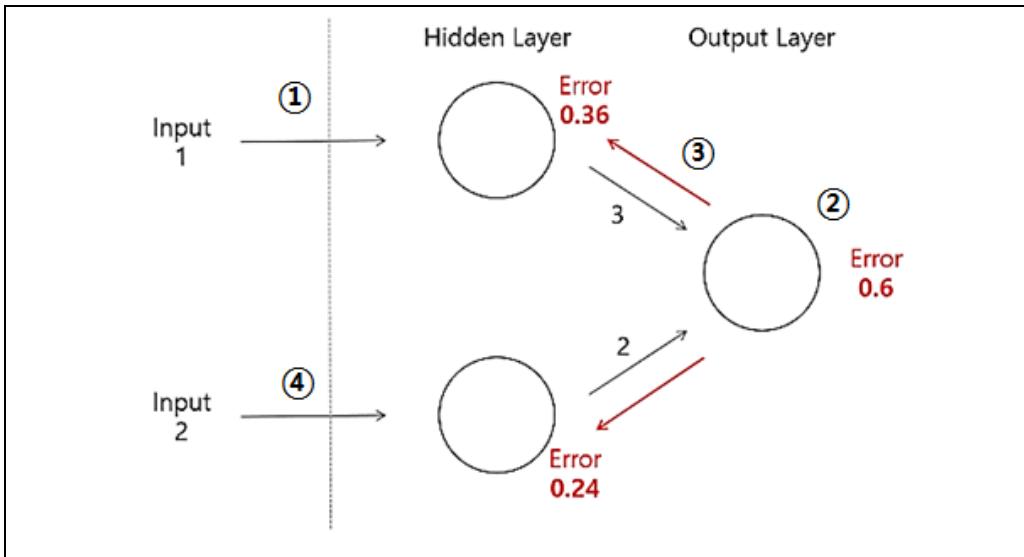
2	역전파(Backpropagation)
문제	인공신경망의 오류 역전파(Backpropagation) 알고리즘
도메인	인공지능
정의	인공신경망의 출력값이 원하는 출력과 다를 경우, 가중치를 갱신하여 오차를 최소화시키도록 반복 수행하여 신경망을 학습시키는 알고리즘
키워드	다층신경망, 입력층, 은닉층, 출력층, 가중치, 경사하강
출제의도분석	기계학습/딥러닝 확산 적용에 따른 인공신경망 학습의 핵심 이론에 대한 이해
답안작성 전략	"알고리즘"을 물어본 것이므로 학습 절차를 설명, 개선 방안 설명시+ $\alpha$
참고문헌	<a href="http://www.incodom.kr/기계학습/오차역전파">http://www.incodom.kr/기계학습/오차역전파</a>
풀이 기술사님	김강용 PE (제 118 회 정보관리기술사 / taoloan@nate.com)

## 1. 인공신경망의 가중치 학습 최적화, 역전파(Backpropagation) 개요

- 인공신경망의 출력값이 원하는 출력과 다를 경우, 가중치를 갱신하여 오차를 최소화시키도록 반복 수행하여 신경망을 학습시키는 알고리즘
- 특징 : 반복 수행, 다층신경망(MLP), 감독학습, 역방향 계산

## 2. 오류 역전파(Backpropagation) 알고리즘의 학습 설명

### 가. 오류 역전파 알고리즘의 학습 개념도



- 출력층의 결과를 비교하여 오차가 있을 경우 역전파하여 은닉층의 가중치를 조정하여 갱신

### 나. 오류 역전파 알고리즘의 학습 절차

단계	설명	특징
①피드포워드	입력층 -> 출력층으로 순전파 수행	가중치 초기화
②오류 역전파 계산	출력층 오류 최소화하는 가중치 탐색 (0.6) 출력층에서부터 역방향 진행	미분(예상값-실제값)
③가중치 조정	학습률만큼 수정한 가중치로 조정 오차값 0.6을 3.2로 배분하여 0.36, 0.24로 갱신	에러 최소평균제곱의 미분
④반복 수행	목표 도달시까지 위 과정 반복	N 회 epoch 수행

- 다층신경망에서 경사하강(Gradient Descent)을 수행하는 핵심 알고리즘
- 국소적 계산과 연쇄법칙을 통한 미분을 통해 계산량을 간소화

### 3. 오류 역전파 알고리즘의 문제점과 해결 방안

구분	항목	설명
문제점	sigmoid 함수의 문제	<ul style="list-style-type: none"> <li>- 계단형식 함수를 미분가능하도록 곡선화</li> <li>- 기울기 문제(vanishing gradient problem) 발생</li> </ul>
해결 방안	ReLU 사용	<ul style="list-style-type: none"> <li>- x 값이 0 이하이면 0 을 출력, 0 이상이면, 비례 함수 적용</li> <li>- <math>\max(0, x)</math> 함수 사용</li> </ul>

"끝"

#### ※ [참고자료] 계산그래프, 국소적 계산, 연쇄법칙 개념

- ✓ 계산그래프 : 계산과정을 그래프로 표현. 복수의 노드와 에지로 구성
- ✓ 계산그래프 흐름 : 왼쪽에서 오른쪽으로 진행 (순전파)  
오른쪽에서 왼쪽으로 진행 (역전파)
- ✓ 국소적 계산 : 각 노드에서의 단순한 계산을 수행하고 결과를 전달 (연산 단순화)
- ✓ 연쇄법칙 : 합성함수의 미분 법칙을 이용하여 국소적 미분을 전달
  
- ✓ 합성함수의 미분
  - 함수  $f$  가  $x_0$  에서 미분 가능하며, 함수  $g$  가  $f(x_0)$ 에서 미분 가능하다고 하면  $f \circ g$  의  $x_0$ 에서도 미분할 수 있음.

$$(f \circ g)'(x_0) = f'(g(x_0))g'(x_0)$$

$y = f(u), u = g(x)$  를 사용하여 다시 쓰면 다음과 같다.

$$\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dx}$$

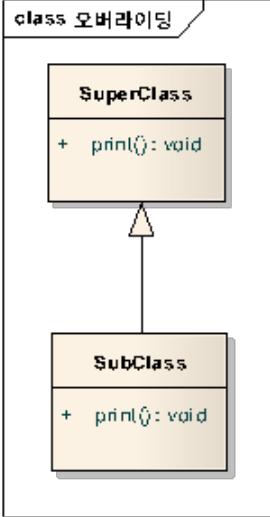
3	오버라이딩(Overriding)
문제	오버라이딩(Overriding)으로 함수를 재정의하는 이유
도메인	소프트웨어공학
정의	자식 클래스가 부모 클래스에 의해 이미 제공된 메소드를 재정의할 수 있도록 제공하는 객체지향 언어 특성
키워드	다형성, 상속, 확장, 재정의, 상위, 하위, 슈퍼
출제의도분석	객체지향 언어 특성의 기본 지식 이해
답안작성 전략	질문에 집중하여 재정의하는 이유 충실히 작성, 다이어그램과 코드 제시+α
참고문헌	107 대비 합숙 2 일차 <a href="https://hyoje420.tistory.com/14">https://hyoje420.tistory.com/14</a> <a href="https://gmlwjd9405.github.io/2018/07/13/template-method-pattern.html">https://gmlwjd9405.github.io/2018/07/13/template-method-pattern.html</a>
풀이 기술사님	김강용 PE (제 118 회 정보관리기술사 / taoloan@nate.com)

## 1. 상위 클래스 메소드를 재정의, 오버라이딩(Overriding) 개요

- 자식 클래스가 부모 클래스에 의해 이미 제공된 메소드를 재정의할 수 있도록 제공하는 객체지향 언어 특성
- 특징 : 다형성 제공, 객체지향 특성, 동일한 인자와 리턴 타입

## 2. 오버라이딩으로 함수를 재정의하는 이유 설명

### 가. 상위클래스의 기능을 하위클래스에서 확장

클래스 다이어그램	샘플코드
 <pre> classDiagram     class SuperClass {         +print(): void     }     class SubClass {         +print(): void     }     SuperClass &lt; -- SubClass   </pre>	<pre> public class SuperClass {     void print() {         System.out.println("상위클래스입니다.")     } }  public class SubClass extends SuperClass {     void print() {         System.out.println("하위클래스입니다.")     } }   </pre>
SubClass는 SuperClass로부터 상속함	SubClass에서 SuperClass의 print 메소드를 재정의

- 다형성을 통해 상위클래스의 메소드를 하위클래스에서 재정의하여 상위클래스의 기능을 확장
- 동일한 Object 타입의 상위클래스 단위로 객체 관리를 용이하게 처리

#### 나. 추상클래스의 메소드를 구현하도록 강제하여 표준 적용

클래스 다이어그램	설명
<pre> classDiagram     class AbstractClass {         +TemplateMethod()         +PrimitiveOperation1()         +PrimitiveOperation2()     }     class ConcreteClass {         +PrimitiveOperation1()         +PrimitiveOperation2()     }     AbstractClass &lt; -- ConcreteClass     Note over ConcreteClass: ...         PrimitiveOperation1()         ...         PrimitiveOperation2()     end   </pre>	AbstractClass : 알고리즘의 기본 오퍼레이션 골격 구조 정의
	ConcreteClass : 서브클래스의 특성에 맞도록 오퍼레이션을 구현

- 디자인 패턴의 행위 패턴 중, 템플릿 메소드 패턴의 예시

- 추상클래스에서 기본 골격을 제시하고 구체클래스에서 구현하도록 유도하여 표준 적용 강제화
- 구체클래스 특성별로 알고리즘을 구현하도록 하여 구현의 유연성 제공

### 3. 오버라이딩의 전제 조건과 제약 사항

구분	항목	조건
전제 조건	메소드 이름	상위클래스와 동일해야 함
	파라미터 개수/자료형	상위클래스와 동일해야 함
	리턴 타입	상위클래스와 동일해야 함
제약 사항	접근 제한자	하위클래스에서 상위보다 좁게 정의 불가
	예외 처리	하위에서 상위보다 큰 범위의 예외 처리 불가
	static 메소드	static 메소드를 인스턴스 메소드로 변경 불가 인스턴스 메소드를 static 메소드로 변경 불가

- 오버로딩은 비슷한 기능을 하는 메소드를 동일한 이름으로 여러 개 정의하는 다형성 특성이며, 파라미터 개수 또는 자료형이 달라야 함

"끝"

Notes

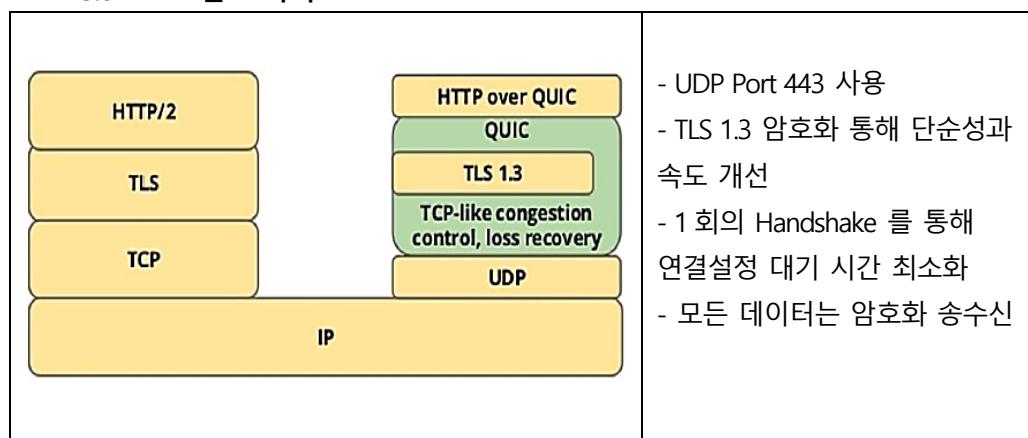
4	HTTP 3.0
문제	HTTP 3.0
도메인	디지털서비스, 네트워크
정의	HTTP 연결 및 전송 지연을 최소화하기 위해, TCP+TLS 조합 대신 UDP+TLS 조합을 이용하는 보안 및 향상된 성능을 제공하는 UDP 기반 전송 계층 프로토콜
키워드	UDP, TLS 1.3, 1-RTT, 0-RTT, QUIC, HOL 블로킹 해결
출제의도분석	5G 초연결 시대의 차세대 HTTP 표준 부각
답안작성 전략	프로토콜 구조와 동작과정 설명, 5G 연계 시사점 제시+α
참고문헌	118 대비 합숙 2 일차 주간기술동향-1901 호, "웹페이지 로딩시간 개선을 위한 IETF HTTP 표준화 개발 동향"
풀이 기술사님	김강용 PE (제 118 회 정보관리기술사 / taoloan@nate.com)

## 1. HTTP 의 차세대 표준, HTTP-over-QUIC, HTTP 3.0 개요

- HTTP 연결 및 전송 지연을 최소화하기 위해, TCP+TLS 조합 대신 UDP+TLS 조합을 이용하는 보안 및 향상된 성능을 제공하는 UDP 기반 전송 계층 프로토콜  
(2018년 12월 HTTP-over-QUIC 프로토콜이 HTTP3.0으로 변경)
- 특징 : UDP 기반, TLS 1.3, RTT 최소화, TCP의 HOL(Head Of Line) 블로킹 문제 해결

## 2. HTTP 3.0 프로토콜 스택 구조와 동작과정

### 가. HTTP 3.0 프로토콜 스택 구조



- UDP+TLS 통한 성능과 보안성 향상 및 QUIC 통한 연결 및 전송 지연 최소화

#### 나. HTTP 3.0 프로토콜의 동작과정

동작 과정	설명
<p>1-RTT (normal)</p> <p>Client: It's been a long time since I connected you. I've got a new STK and SCID. I send a full CHLO.</p> <p>Server: You STK is expired. I give you a new Source Address Token and Server Config.</p>	<p>1) Initial 1-RTT Handshake 주소 토큰, 서버 인증서를 gzip 압축 파일형태로 client로 전달</p>
<p>0-RTT (repeated resumption)</p> <p>Client: I remember the server address token and server config. It's a repeated access in a short time.</p> <p>Server: You have a right STK and SCID. So let's begin to use the previous shared master secret.</p>	<p>2) Successful 0-RTT Handshake 캐시된 자격증명 사용해서 해드쉐이크를 skip하고 암호화된 요청을 즉시 서버로 전송</p>
	<p>3) Rejected 0-RTT Handshake 클라이언트의 캐싱된 정보가 오래된 경우 1-RTT Handshake를 재수행</p>

- 최초 핸드쉐이킹후, 재 연결시 캐싱된 정보를 사용하여 Zero-RTT 처리

### 3. HTTP 3.0의 제약 사항과 5G 연계 시사점

구분	항목	설명
제약 사항	UDP 속도 제한	DDoS 공격 대응책으로 기업내 속도 제한 추세
	프로토콜 제약	중폭공격 대응책으로 초기 패킷과 응답 패킷 조건 제약
5G 연계 시사점	IoT 초연결	HTTP 기반 연결 및 전송 지연 최소화
	성능, 보안 향상	UDP와 TLS 기반의 성능 및 보안성 향상 서비스 확산

"끝"

#### ※ [참고] HOL(Head Of Line) 블로킹

- 패킷을 대기 행렬에 큐잉하여 처리하며, FIFO 처리함 (대기열의 머리에 있는 패킷은 대기열의 끝에 있는 패킷보다 먼저 전달)
- 순차 처리 제약으로 인해 머리가 처리 되지 않으면 후속 패킷은 대기하게 됨
- 동일한 송신 포트 자원에 대한 처리량 경쟁으로 인해 처리량 지연 및 프레임 손실 발생을 유발

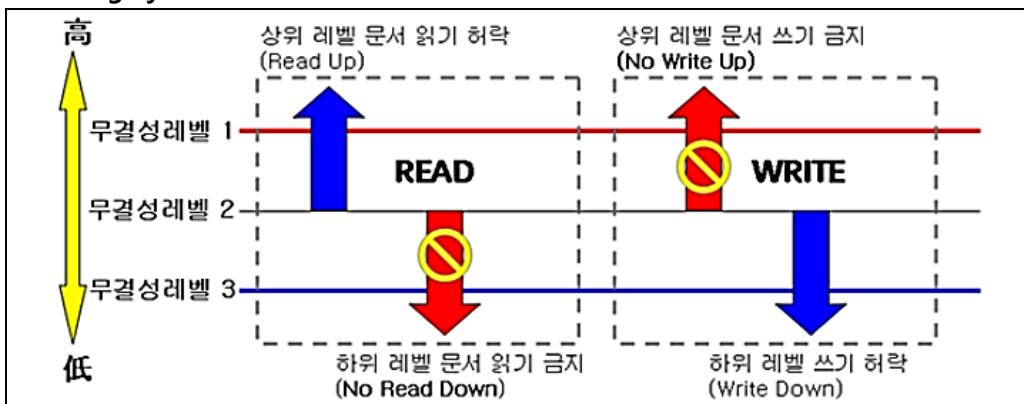
5	Biba 보안모델
문제	Biba Integrity Model
도메인	보안
정의	기밀성을 강조하는 벨-라파둘라(BLP) 모델의 단점인 무결성을 보장할 수 있도록 보완한 접근 통제 모델
키워드	무결성, No Read Down, No Write Up, 단순 속성, 스타 속성
출제의도분석	접근 통제를 실현하는 통제 모델의 대표적인 모델의 이해
답안작성 전략	키워드 위주의 작성, 보안 속성 도식화, BLP 와의 비교 또는 Biba 개선방안+α
참고문헌	117 대비 합숙 5 일차 <a href="https://blog.naver.com/PostView.nhn?blogId=sdug12051205&amp;logNo=221575582613&amp;categoryNo=2&amp;parentCategoryNo=08&amp;viewDate=&amp;currentPage=1&amp;postListTopCurrentPage=1&amp;from=search">https://blog.naver.com/PostView.nhn?blogId=sdug12051205&amp;logNo=221575582613&amp;categoryNo=2&amp;parentCategoryNo=08&amp;viewDate=&amp;currentPage=1&amp;postListTopCurrentPage=1&amp;from=search</a>
풀이 기술사님	김강용 PE (제 118 회 정보관리기술사 / taoloan@nate.com)

## 1. 무결성을 강조하는 접근통제 보안모델, Biba Integrity Model 개요

- 기밀성을 강조하는 벨-라파둘라(BLP) 모델의 단점인 무결성을 보장할 수 있도록 보완한 접근 통제 모델
- 특징 : 선택적 정책 적용 가능, BLP 모델의 비밀 등급에 대한 무결성 등급 보유,  
비인가자의 임의 수정/변조 방지,  
인가된 사용자에 의한 변조는 고려하지 않음

## 2. Biba Integrity Model 의 개념도와 보안속성

### 가. Biba Integrity Model 의 개념도



- 상위레벨 문서에 대한 쓰기 금지를 통해 보안 등급에 따른 무결성을 보장

### 나. Biba Integrity Model 의 보안속성

보안속성	규칙	설명
단순 속성	No Read Down	주체는 자신보다 낮은 등급의 객체 정보를 읽을 수 없음
*(스타) 속성	No Write Up	주체는 자신보다 높은 등급의 객체에 정보를 쓸 수 없음

- Biba 모델은 무결성을 강조하는 반면, BLP 모델은 기밀성을 강조함 (No Read Up/No Write Down)

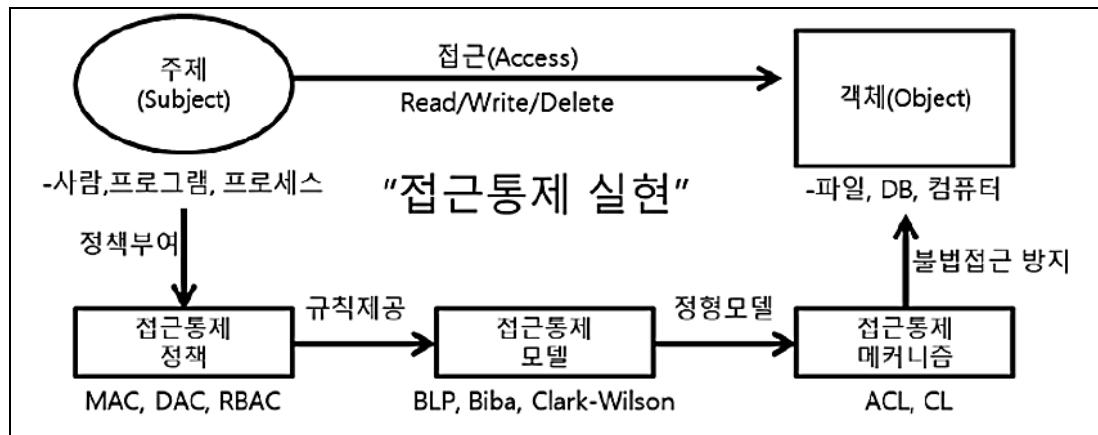
### 3. Biba Integrity Model 의 한계점과 대응 방안

구분	항목	설명
한계점	인가자의 변조	인가자에 의한 정보의 변조를 고려하지 않음
	기밀성 취약	기밀성 보다는 무결성에 치중하고 있음
대응 방안	Clark-Wilson 모델	직무 분리 모델을 적용하여 인가자의 변조를 방어
	Lipner 모델	기밀성과 무결성을 동시에 고려한 보안 모델을 적용

- 금융, 투자, 회계 등 분야에서 이해 충돌을 방지하는 Brewer-Nash(The Chinese Wall)모델도 존재

"끝"

### ※ [참고자료] 접근 통제 프레임워크



- 접근통제는 사용자(주체)의 신원을 식별/인증하여 대상정보(객체)의 접근, 사용자수준을 인가하는 절차
- 접근 통제 3 요소는 정책, 모델, 메커니즘으로 구성
- 접근통제 모델은 시스템의 보안 요구명세를 표현하는 접근통제에 관한 정형적 기능모델

6	모니터(Monitor) 동기화
문제	모니터(Monitor) 프로세스 동기화 기법
도메인	운영체제
정의	프로세스 동기화 수행시 세마포어의 단점인 타이밍 문제를 해결하기 위하여 프로그래밍 언어 수준에서 상호배제를 구현하여 동기화를 제공하는 기법
키워드	타이밍 문제, 언어수준의 상호배제, 배타동기, 조건동기, synchronized, wait, notify
출제의도분석	언어 수준에서 상호배제 기법을 구현하는 모니터 기법에 대한 이해
답안작성 전략	모니터의 사용 이유와 동작원리 설명, 소스코드 사례+α
참고문헌	<a href="https://kkimsangheon.github.io/2019/02/22/operating-system16/">https://kkimsangheon.github.io/2019/02/22/operating-system16/</a> <a href="https://about-myeong.tistory.com/34">https://about-myeong.tistory.com/34</a>
풀이 기술사님	김강용 PE (제 118 회 정보관리기술사 / taoloan@nate.com)

## 1. 프로그램 언어 수준의 동기화 기법, 모니터(Monitor) 개요

### 가. 모니터(Monitor)의 개념

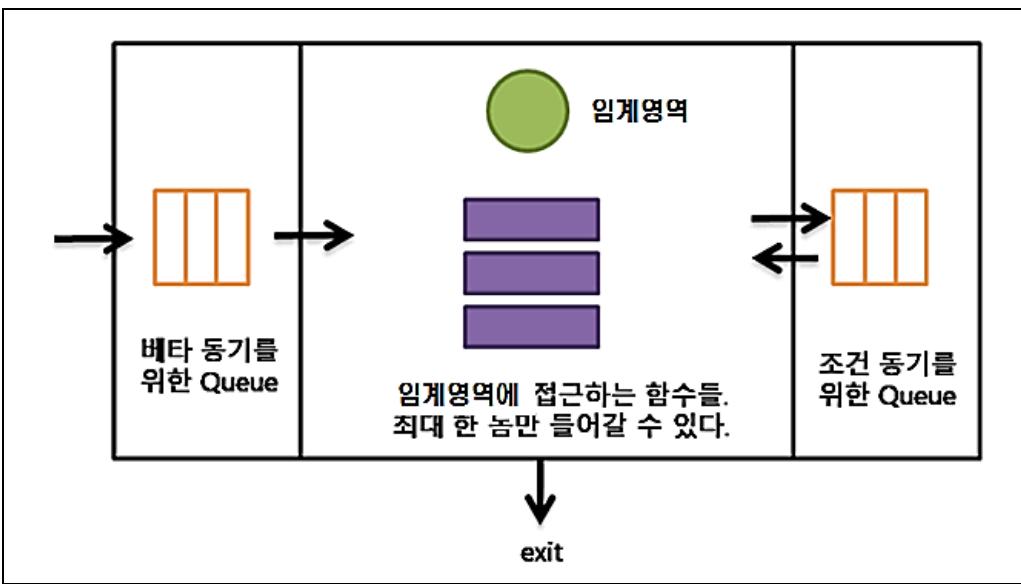
- 프로세스 동기화 수행시 세마포어의 단점인 타이밍 문제를 해결하기 위하여 프로그래밍 언어 수준에서 상호배제를 구현하여 동기화를 제공하는 기법

### 나. 모니터의 필요성

필요성	설명
타이밍 문제 해결	세마포어는 임계구역 진입 전/후에 wait 와 signal 발생 순서가 어긋날 경우에 타이밍 문제가 발생
언어 수준 편의성	카운터 변수의 관리를 캡슐화하여 보호하고 키워드를 통해 접근

## 2. 모니터의 구조와 기술요소

### 가. 모니터의 구조도



- 스레드 요청시 임계영역 접근을 위해 배타동기 큐를 사용하고 상호배제 위해 조건동기 큐 사용
- 임계영역 접근은 한 스레드만 허용하도록 counter 변수는 캡슐화하여 라이브러리에서 관리

## 나. 모니터의 기술요소

구분	기술요소	설명
동기화 요소	임계영역	프로세스와 스레드에서 접근하는 공유 자원 영역
	배타동기 큐	한 스레드만 임계영역에 접근하도록 제어하는 큐
	조건동기 큐	스레드를 조건에 따라 잠그거나 깨우는 큐
언어 요소 (JAVA 사례)	synchronized	배타동기 큐로 진입시키는 키워드
	wait()	임계영역에서 조건동기 큐로 진입시켜 스레드를 잠금
	notify(), notifyAll()	조건동기 큐에 블록된 스레드를 깨워 임계영역에 재진입

- 모니터는 언어 수준에서 상호배제를 구현하는 수단을 제공하며, Ordering 용도로도 사용 가능

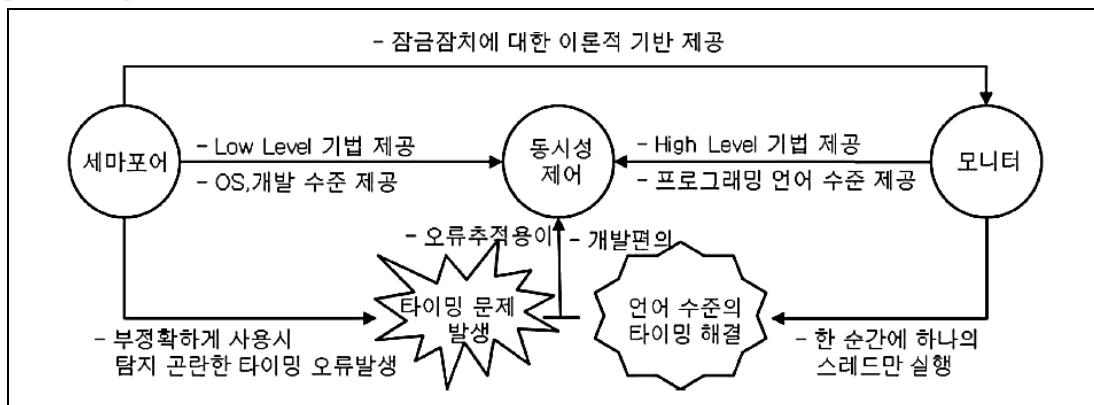
## 3. JAVA 언어에서의 모니터 사용 사례

<pre>class BankAccount {     int balance;      synchronized void deposit(int amt) {         int temp = balance + amt;         System.out.print("+");         balance = temp;     }      synchronized void withdraw(int amt) {         int temp = balance - amt;         System.out.print("-");         balance = temp;     }      int getBalance() {         return balance;     } }</pre>	<p>은행예금 클래스 계좌잔액을 공유자원으로 사용 입금메소드를 선언 - synchronized 키워드 사용하여 계좌잔액 접근시 상호배제를 수행하여 입금시 다른 메소드 접근 배제</p> <p>출금클래스를 선언 - synchronized 키워드 사용하여 계좌잔액 접근시 상호배제를 수행하여 출금시 다른 메소드 접근 배제</p> <p>공유자원인 계좌잔액을 리턴</p>
--	--

- 모니터 사용시 synchronized 키워드를 사용하여 공유자원 접근시 편리하게 사용

"끝"

## ※ [참고자료] 세마포어와 모니터의 관계



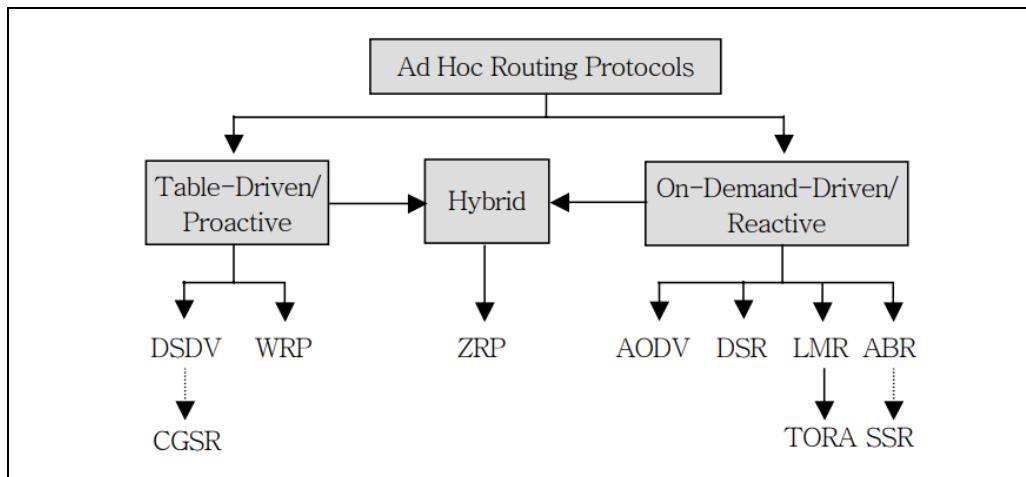
7	Ad-hoc Network Routing Protocol (핵심요약)
문제	Ad-hoc Network Routing Protocol
도메인	네트워크
정의	동적 Topology로 구성된 Ad-hoc NW 망에서 NW 참여 노드 간 통신 경로 설정 및 유지를 위한 라우팅 프로토콜
키워드	Proactive, Hybrid, Reactive(On-demand), DSDV, AODV
출제의도분석	IoT 기반 스마트 시스템(팩토리, 팜, etc) 구성 증가에 따른 Ad-hoc 기반 통신 프로토콜 출제
답안작성 전략	프로토콜 방식 분류 기반 상세 프로토콜 설명 후 추가적으로 실제 ad-hoc 라우팅 관련 시스템 구성 작성
참고문헌	- 이동 Ad Hoc 네트워크 기술 동향(전자통신동향분석 제 18 권 제 2 호), - 무선 Ad Hoc 네트워크에서 단방향 링크를 이용한 하이브리드 라우팅
풀이 기술사님	김승준 PE (제 117 회 컴퓨터시스템응용기술사 / kimsj2435@gmail.com)

## 1. 동적 Topology NW 내 기기간 통신, Ad-hoc NW 라우팅 프로토콜 정의

- 동적 Topology로 구성된 Ad-hoc NW 망에서 NW 참여 노드 간 통신 경로 설정 및 유지를 위한 라우팅 프로토콜
- (특징) Multi-hop Routing, 동적 Topology 구성, Peer-to-Peer 통신

## 2. Ad-hoc NW Protocol 분류 및 설명

### 가. Ad-hoc NW Protocol 분류



- RIP/OSPF 라우팅 프로토콜은 동적 변화가 빈번한 Ad-hoc NW에서는 부적합

### 나. Ad-Hoc NW Protocol 설명

분류	프로토콜	설명
Table- Driven Proactive (테이블 관리 방식)	DSDV	- 주기적 혹은 NW 변화 시, 라우팅 테이블 Broadcasting - 항상 최신 경로 정보 유지, 경로 탐색 지연 없이 트래픽 라우팅 - Broadcasting Overhead 존재, 메시지 크기 최소화 필요(소규모 NW에서 사용)
		- Destination Sequenced Distance Vector(벨만포드 알고리즘) - Sequence Number 통해 최신 메시지 판단, 브로드캐스팅 시, 루프 문제 방지

Notes

	WRP	<ul style="list-style-type: none"> <li>- Wireless Routing Protocol</li> <li>- 거리/라우팅/링크비용/메시지 재전송 리스트 테이블 기반 라우팅 수행</li> <li>- 메시지 재전송 리스트 테이블 기반 테이블 정보 송/수신 통해 라우팅 정보 유지</li> <li>- 이웃 노드에게만 라우팅 정보를 전송</li> </ul>
	CGSR	<ul style="list-style-type: none"> <li>- Clustered GW Switching Routing</li> <li>- 클러스터 단위로 NW 구성 후, GW 역할 수행 클러스터 헤드 노드 선출, 이를 기반으로 라우팅 수행 프로토콜</li> <li>- 클러스터 헤드 선출 Overhead 최소화 위해 LCC(Least Cluster Change) 알고리즘 기반 헤드 변경</li> </ul>
On-Demand-Driven Reactive (요구 기반 방식)		<ul style="list-style-type: none"> <li>- 패킷 라우팅 필요한 순간(On-demand) 라우팅 경로 계산, 라우팅 수행</li> <li>- 이동 노드 NW 경로 상시 유지하지 않고 필요할 때만 경로 계산, 유지</li> <li>- 테이블 기반 라우팅의 단점 해결(Board Casting Overhead)</li> <li>- 경로 계산으로 인한 지연 시간 발생하지만, 최신 경로로 라우팅 가능</li> </ul>
	DSR	<ul style="list-style-type: none"> <li>- Dynamic Source Routing</li> <li>- 경로발견, 경로유지 메커니즘 기반 구성</li> <li>- 다중 NW 흡 기반 소스 루트 발견, 리스트 기반 전송</li> </ul>
	AODV	<ul style="list-style-type: none"> <li>- Ad Hoc On-demand Distance Vector</li> <li>- 라우팅 경로 발견 시, 이웃 노드에 RREQ(Route Request) 브로드캐스팅, 이를 반복하여 목적지까지 경로 탐색</li> <li>- 목적지 도달 시, RREP(Route Response) 통해 응답, 이를 기반으로 경로 탐색</li> </ul>
Hybrid		<ul style="list-style-type: none"> <li>- 일정 영역 내의 경로 데이터를 Proactive 방식으로 유지, 영역 외의 경로는 Reactive 방식으로 라우팅 수행하는 방식</li> <li>- 기존 2 방식의 장/단점을 보완하기 위한 프로토콜 방식</li> </ul>
	ZRP	<ul style="list-style-type: none"> <li>- Zone Routing Protocol</li> <li>- 동일 영역(Zone) 내에서 Intra-zone Routing Protocol(IARP) 기반 Proactive 방식 라우팅</li> <li>- 외부 영역 탐색 시 Inter-zone Routing Protocol(IERP)기반 Reactive 방식 라우팅</li> </ul>

- NW 규모 및 노드 간 통신 요구사항(실시간성, 지연 등)에 따른 라우팅 프로토콜 선택 필요

[참고 - 1. 리눅스 시스템에서 ad-hoc NW 구성 방법(Ubuntu 기준)]

- 리눅스 시스템의 NW 설정 파일에 "ad-hoc" 모드 set 설정 후, network 데몬 재시작 시 반영

- Ubuntu 18.04 이전: /etc/network/interfaces.d 파일에 "wireless-mode ad-hoc" 추가

- Ubuntu 18.04 이후: /etc/netplan/adhoc.yaml 파일에 "mode: adhoc" 추가

[참고 -2. 실제 DSR/AODV 소스코드 구현]

- select() 함수 기반 event-driven 방식으로 UDP 기반 구현

- NW 관련 프로세스 시작 시, HELLO 메시지를 주기적으로 송/수신하며 NW 참여 노드상태 체크

"끝"

Notes

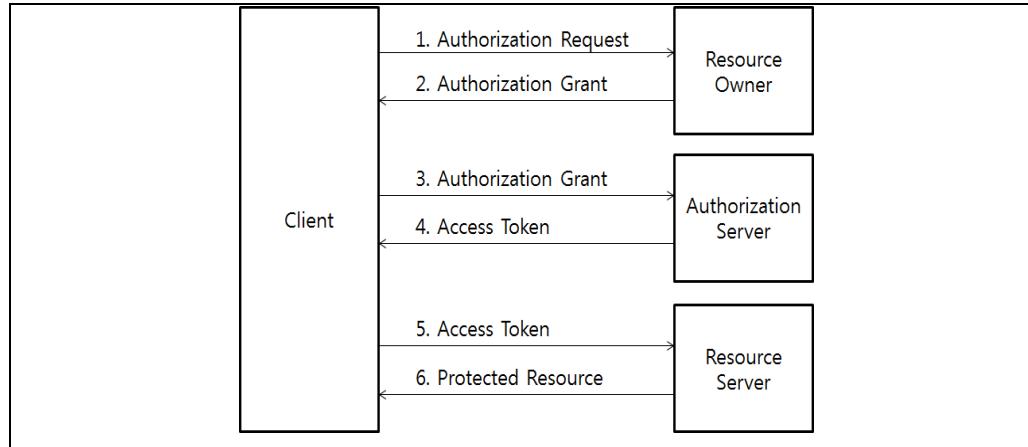
8	OAuth2.0
문제	OAuth2.0
도메인	보안
정의	서버 자원 사용 위해 권한 획득 및 Access Token 기반 인증 및 접근 제어 수행하는 인증 표준
키워드	(절차) 권한획득, 토큰획득, 자원획득 (표준) RFC 6749, RFC 6750 (구성요소) 클라이언트, 자원 소유자, 권한 서버, 자원 서버
출제의도분석	클라우드 서비스 사용의 증가, OAuth 기반 SSO 인증 시스템 구성의 증가로, OAuth2.0 문제 출제
답안작성 전략	OAuth1.0 과의 차이, OAuth2.0 인증 절차, 추가로 OAuth2.0 보안 문제 작성
참고문헌	- The OAuth 2.0 Authorization Framework(RFC6749) - OAuth2.0 개요 및 보안 고려사항(금융보안원)
풀이 기술사님	김승준 PE (제 117 회 컴퓨터시스템응용기술사 / kimsj2435@gmail.com)

## 1. 인증을 위한 Open Standard Protocol, OAuth 2.0 정의

- 서버 자원 사용 위해 권한 획득 및 Access Token 기반 인증 및 접근 제어 수행하는 인증 표준
- 제 3의 앱이 자원 소유자에게 서비스 이용자를 대신하여 서비스를 요청할 수 있도록 자원 접근 권한을 위임
- (예시) 인스타그램에서 SNS 친구 찾기 위해 페이스북 오픈 API 기반 자원소유자 친구 목록 접근

## 2. OAuth2.0 수행 절차 및 절차 설명

### 가. OAuth2.0 수행 절차



- 사용자(자원 소유자)가 권한 인증 후 토큰을 획득하여 토큰 기반 자원 서버에 접근

### 나. OAuth2.0 수행 절차 설명

No.	Client 수행	설명
1	권한 요청	- Resource Owner에게 권한 인가 요청
2	권한 인가	- Resource Owner(사용자 및 서버)는 Client 식별 후, 권한 인가(Grant)
3	권한 기반 토큰 요청	- 권한 기반 인증 서버로 토큰 요청
4	토큰 반환	- 자원 접근 권한 식별 후, 해당 요청 자원 접근 가능 토큰 발급 후 반환

Notes

5	токен 기반 자원 접근	- 자원 요청 시 해당 자원 요청 권한을 가진 토큰 기반 요청
6	자원 반환	- 토큰 식별 후, 자원 반환

- OAuth1.0 과 달리 HTTPS 를 기본으로 수행되며, 자원 서버가 별도 서명 요구하는 경우 서명사용

### 3. OAuth1.0 과 OAuth2.0 비교

특징	OAuth 1.0	OAuth 2.0
역할 명칭	이용자(User)	- 자원 소유자(Resource Owner)
	소비자(Consumer)	- 클라이언트(Client)
	서비스 제공자(Service Provider)	- 자원서버(Resource Server) - 권한서버(Authorization Server)
API 호출 인증/보안	서명 기반 인증 및 보안	- HTTP(SSL/TLS) default - 별도 서명 기능 추가 가능
클라이언트 지원	웹 어플리케이션	- 웹 어플리케이션 - 이용자 에이전트 어플리케이션 - 네이티브 어플리케이션

- 다양한 클라이언트 지원 및 성능 확장, 암호화 요구사항을 만족하기 위해 OAuth2.0 지원

[참고 1 – OAuth2.0 보안 위협]

분류	공격 유형	설명
통신구간 보안위협	중간자 공격	- 자격증명 Hand Shaking 을 NW 계층에서 도청 및 변조
클라이언트 유형 별 위협	웹 토큰 캐싱	- 모든 클라이언트 토큰 유지하여 악의적인 공격
	이용자 단말 공격	- 악성 코드 감염 단말기에서 Access Token 획득
자원 소유자 공격	- Pharming, Click-Jacking 통해 인증 정보 탈취, 권한 승인 유도 * Click-Jacking) 정상/악성 웹 페이지 중첩하여 공격자 의도대로 마우스 클릭 유도하는 공격 기법	

[참고 2 – 금융권에서의 OAuth2.0 역할]

< OAuth 2.0의 역할 및 금융권 예 >

역할	설명	금융권 예
자원 소유자 (Resource owner)	보호 자원에 대한 접근 권한을 부여할 수 있는 개체(일반적으로 이용자)	핀테크 업체의 금융 서비스를 사용하는 금융고객
클라이언트 (Client)	자원 서버에 보호 자원을 요청하고 관련 서비스를 제공하는 어플리케이션	금융 오픈 API를 사용하여 금융 서비스를 제공하는 핀테크 어플리케이션
자원 서버 (Resource server)	보호된 자원에 대한 서비스 API를 제공하는 서버	금융 오픈 API(예금 조회/이체, 결제 등)를 제공하는 금융사의 서버
권한 서버 (Authorization server)	클라이언트가 보호된 자원에 대한 제한된 접근할 수 있도록 자원 접근 권한을 위임 및 관리하는 서버	금융 서비스에 대한 제한된 접근 권한을 핀테크 어플리케이션에게 위임 및 관리하는 금융사의 서버

"끝"

9	포그 컴퓨팅(Fog Computing)
문제	포그 컴퓨팅(Fog Computing)
도메인	디지털 서비스
정의	기존 중앙 집중식 데이터 수집 및 처리 서버-클라이언트 모델의 트래픽 부하를 분산하기 위해 엣지 부분에서 데이터를 선 처리하는 아키텍처
키워드	IEEE 1394, Location Awareness, Low Latency, Heterogeneity Support
출제의도분석	107 회 관리 기출 토픽으로, 현재 MEC(Mobile Edge Computing)과 더불어 클라우드 시스템 관련 토픽으로 출제
답안작성 전략	기출 토픽이므로 상세한 아키텍처 및 기술 요소 작성으로 차별화 필요
참고문헌	- Fog Computing Architecture: Survey and Challenges(Article) - Fog Computing Overview(Cisco)
풀이 기술사님	김승준 PE (제 117 회 컴퓨터시스템응용기술사 / kimsj2435@gmail.com)

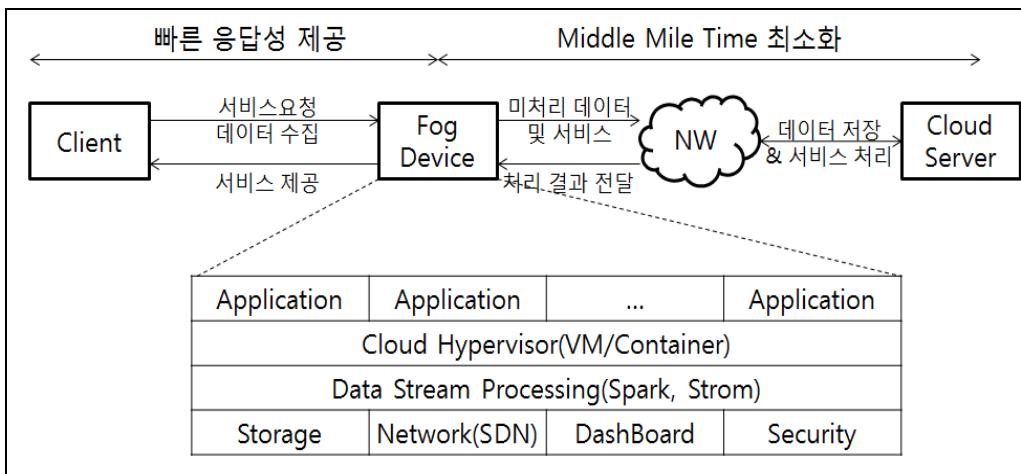
## 1. 근거리 대용량 데이터 처리 아키텍처, 포그 컴퓨팅 정의



- 기존 중앙 집중식 데이터 수집 및 처리 서버-클라이언트 모델의 트래픽 부하를 분산하기 위해 데이터 발생 지역에서 데이터를 선 처리하는 아키텍처
- (특징) Middle Mile Time 최소화, Low Latency, Fog 노드 기반 처리

## 2. 포그 컴퓨팅 아키텍처 및 구성요소

### 가. 포그 컴퓨팅 아키텍처



- Client 서비스 요청 및 데이터 관련 1차 처리는 Fog Device에서 수행, 이후 Cloud SVR로 전달

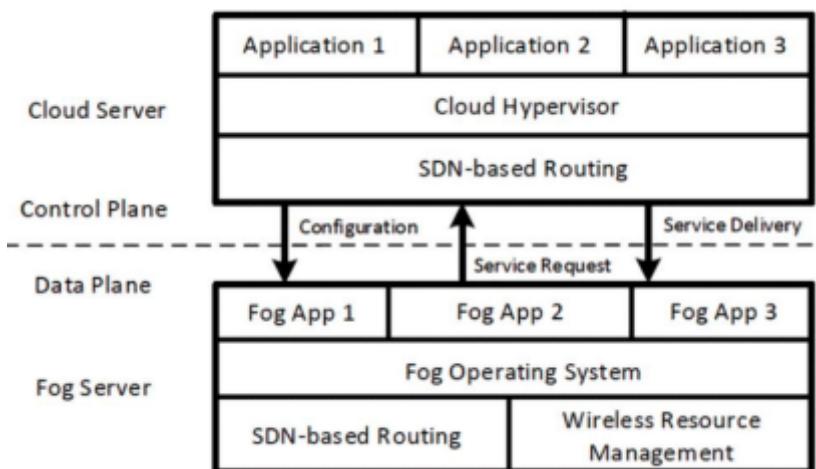
### 나. 포그 컴퓨팅 구성요소

Notes

분류	구성요소	설명
End Point (Client)	스마트 디바이스	- 사용자 요청 서비스 및 데이터 Fog 노드로 전달
	IoT 및 센서	- 실시간 발생 데이터 수집 및 처리 관련 Fog 노드 전달
Fog Node	서비스 처리부	- 컴퓨팅에 필요한 메모리, 저장 기능 보유 - 간단한 서비스 및 데이터 즉각적인 분석 가능
	가상화	- 클라우드 서비스 아키텍처 따라 가상화 기술 적용 - 최근 MSA 기반 서비스 아키텍처로 쿠버네티스 기반 컨테이너
Cloud Server	IDC 센터	- 대용량 데이터 또는 Fog 노드 처리 불가 데이터 처리

- 클라우드 서비스가 Fog 및 엣지 컴퓨팅 기반 Middle Mile Time 최소화로 빠른 응답성을 제공

[참고 – SDN 기반 포그 컴퓨팅 아키텍처]



- 클라우드 서버에서 SDN 기반 Fog Node NW Traffic 제어 및 분산 처리하는 아키텍처 구성 가능

"끝"

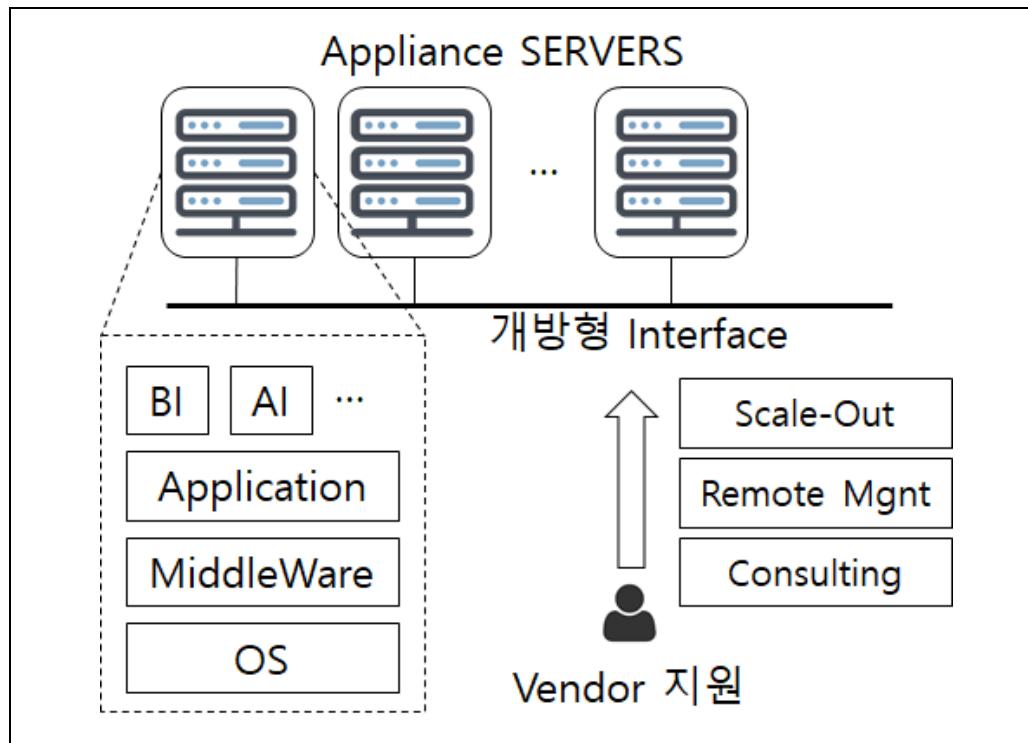
10	어플라이언스(Appliance) 서버
문제	어플라이언스(Appliance) 서버
도메인	디지털 서비스
정의	사용자 별도의 OS 및 SW 설치 없이 벤더가 사전에 통합, 최적화하여 제공되어 전원 연결 시 즉시 사용이 가능한 서버
키워드	어플라이언스 디바이스, OS/SW 벤더 제공, Scale-out/관리/업데이트 용이
출제의도분석	On-premise 클라우드 서버 구축 수요의 증가로 어플라이언스 형태의 IT 시스템 구축 관련 토픽 출제
답안작성 전략	클라우드 데이터 센터 구축 증가에 따라 빠른 Scale-out, SW 업데이트, 이중화가 가능한 어플라이언스 기기 관련 토픽 출제
참고문헌	- 보안 관리와 비용 절감의 스마트한 3C(AhnLab) - Cisco Network Appliance Server White Paper(Cisco)
풀이 기술사님	김승준 PE (제 117 회 컴퓨터시스템응용기술사 / kimsj2435@gmail.com)

## 1. 전원 연결로 바로 사용이 가능한 서버, 어플라이언스 서버 개요

- 사용자 별도의 OS 및 SW 설치 없이 벤더가 사전에 통합, 최적화하여 제공되어 전원 연결 시 즉시 사용이 가능한 서버
- (특징) 모든 구성요소 단일 벤더에서 제공, 고성능 확장/고가용성 목표 이중화 구성

## 2. 어플라이언스 서버 아키텍처 및 특장점

### 가. 어플라이언스 서버 아키텍처



- Client 필요 어플리케이션 및 Configuration 을 vendor에서 제공하여 빠른 인프라 구성 가능

#### 나. 어플라이언스 서버 특장점

유형	상세	설명
고성능	응용 최적화 설정	- HW 구성부터 BIOS 옵션, OS 계층부터 응용 서비스 최적화 위한 vendor 사 설정 제공
	응용 기반 최적화 연산	- 데이터 처리 등과 같은 고 연산 처리 시, 최적화된 성능 제공
저비용	Total 솔루션	- 한 개의 벤더에서 total 솔루션 제공
	개방형 Interface	- 다양한 BI 어플리케이션과의 개방형 Interface 제공
	관리 비용 절감	- SA/DBA/컨설팅 등 인력 감소
편의성	설치 시간 최소화	- 실제 사용 어플리케이션 확장 설치 시간 최소화
	Scale-out 용이	- 통일된 인터페이스로 시스템 확장 방안 및 간편 업데이트 가능

- 클라우드 트랜스포메이션 수요 증가에 따라 On-premise 구축 시, 어플라이언스 서버 사용

### 3. 어플라이언스 서버의 기대효과

- HW 유지 비용 및 SW 업데이트에 대한 벤더 사 지원으로 문제 해결
- Scale-out 및 이중화 구성 용이하여 고가용성 제공
- HW 일체형으로 시스템 자원 효율적 사용 및 구성 가능

"끝"

Notes

11	FeRAM/STT-MRAM/PRAM
문제	FeRAM/STT_MRAM/PRAM
도메인	컴퓨터 구조
정의	<p>FeRAM) 외부 전기장 없이도 스스로 분극을 가지는 강유 전체 기반 비휘발성 메모리</p> <p>STT-MRAM) 기존 MRAM 의 Shrink 한계 극복, 고집적화 달성을 위해 Switching 방식에 변화를 준 차세대 비휘발성 메모리</p> <p>PRAM) 비정질/결정질 상태에 따라 전기적 비저항 차이를 기반으로 정보 기록하는 비휘발성 메모리</p>
키워드	FeRAM) 강유 전체 STT-MRAM) 고집적, 자성체, 스피드 PRAM)
출제의도분석	반도체 시장이 외교적인 이슈로 화두가 되고 있고, 삼성전자의 STT_MRAM, SK하이닉스의 PRAM 등의 차세대 메모리 반도체 관련 토Pic으로 출제
답안작성 전략	차세대 반도체 기술이므로, 각 기술의 특징 및 기존 기술과의 차이점 언급
참고문헌	<ul style="list-style-type: none"> <li>- 신 메모리 소자의 개발 현황 및 전망(정홍식)</li> <li>- 전극형 상변화 메모리(PRAM) 기술(김수경, 이현, 홍성훈)</li> </ul>
풀이 기술사님	김승준 PE (제 117 회 컴퓨터시스템응용기술사 / kimsj2435@gmail.com)

## 1. 차세대 반도체 정보기억장치, FeRAM/STT\_MRAM/PRAM 개요

분류	정의
FeRAM	<p>[Ferroelectric(강유 전체) RAM]</p> <p>강유 전체의 이력현상(히스테리시스) 기반 유전체막의 전기적 잔류 분극 극성에 따라 정보 기록하는 비휘발성 메모리</p> <p>[추가 설명]</p> <ul style="list-style-type: none"> <li>* 강유 전체?) 외부의 전기장이 없이도 스스로 분극(+V/-V)을 가지는 재료           <ul style="list-style-type: none"> <li>&gt; DRAM 에 강유 전체를 포함하여 외부 전원 없이도 정보 비휘발</li> <li>&gt; (분극)+V/-V 에 따라서 비트 0/1 구분하여 정보 저장</li> </ul> </li> <li>* 이력 현상?) 전기장 세기에 따라 분극 값이 변하고, 전기장의 세기가 작아져도 원점으로 돌아가지 않고 일정한 분극 값을 가지는 현상           <ul style="list-style-type: none"> <li>&gt; 전기장이 작아져도 일정한 분극 값을 유지하기에 비휘발성</li> </ul> </li> </ul>
STT_MRAM	<p>[Spin Transfer Torque Magnetoresistive RAM]</p> <p>MRAM 의 Shrink 한계를 극복하고 고집적화를 달성을 위해 Switching 방식에 변화를 준 비휘발성 메모리</p> <p>[추가 설명]</p> <ul style="list-style-type: none"> <li>* MRAM ?) 자성체(마그네틱) 소자를 이용한 비휘발성 메모리           <ul style="list-style-type: none"> <li>&gt; Flash 메모리 대비 쓰기 속도 1000 배 빠르고 전력 소모가 적음</li> <li>&gt; DRAM 의 단점인 Refresh 동작 없이 오랜 시간 데이터 보관 가능</li> </ul> </li> <li>* Shrink ?) 칩사이즈 축소 기술 &gt; 기존 MRAM 은 집적 기술에 한계 존재</li> </ul>
PRAM	<p>[Phase Change(전극형 상변화) RAM]</p> <p>비정질 상태(Amorphous)와 결정질 상태(Crystal)에서의 전기적 비저항 차이를 기반으로 정로를 기록하고 읽는 비휘발성 메모리</p>

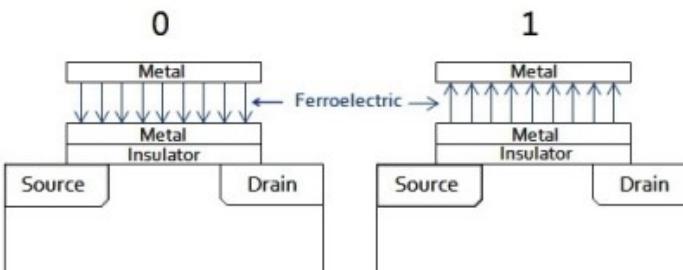
## [추가설명]

결정질 ?) 재료에 원자 간 반복적인 혹은 주기적인 배열이 존재하는 재료  
 비정질 ?) 결정화되지 않는 재료로, 장범위의 원자 규칙성이 미존재  
 > 이러한 원자 배열에 따라 전자들의 상태도 상이하므로 이런 성질 이용

- 클라우드 시스템 활성화로, 데이터 센터 구성에 필요한 차세대 메모리 반도체 기술 필요

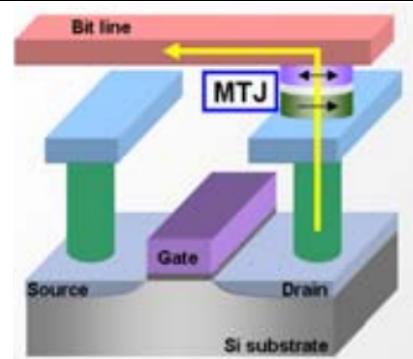
## 2. FeRAM/MRAM/PRAM 설명

## 가. 강유 전체 기반 비휘발성 메모리, FeRAM 설명

FeRAM	설명
	<ul style="list-style-type: none"> <li>- 자기장의 세기에 따라 분극의 값이 변경</li> <li>- 강유 전체 성질로 인해 외부 자기정이 없어도 해당 분극 값이 일정하게 유지</li> <li>- 비휘발성</li> </ul>

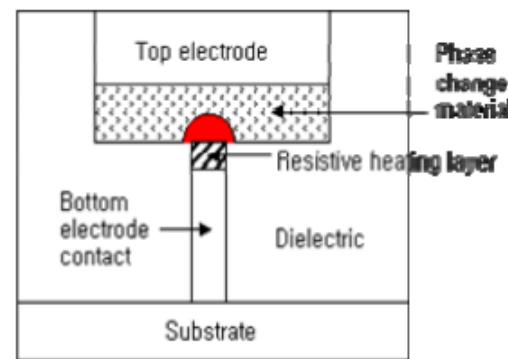
- 상대적 낮은 전압으로 정보를 재기록 가능

## 나. MRAM 의 Shrink 한계 극복, STT-MRAM 설명

STT-MRAM	설명
	<ul style="list-style-type: none"> <li>- 스핀전류에 의한 자화 반전</li> <li>- 전류 통해 선택된 Cell 만 스위칭</li> <li>- 소자가 작아질수록 스위칭 전류 감소(저전력)</li> <li>- 단순 구조로 고집적 가능</li> </ul>

- 작용/반작용과 비슷한 원리로, 자화 방향이 전류에 영향을 주어 cell 스위칭

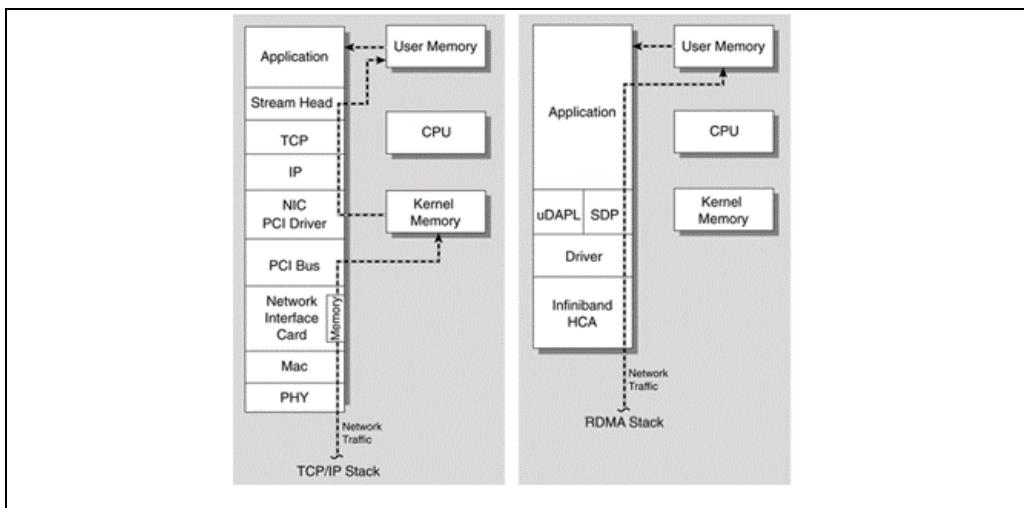
## 다. 비정질/정질의 전자 상태 차이 기반 메모리, PRAM 설명

PRAM	설명
	<ul style="list-style-type: none"> <li>- 하부전극, 가열층(Heating Layer), 상변화 물질, 상부전극으로 구성</li> <li>- 가열층과 상변화 물질에서 열에 의한 비정질, 결정질 간의 상변화 발생</li> </ul>

- 줄가열(Joule Heating) 방법 따라 PRAM 소자의 포어형 및 전극형으로 구분 가능 “끝”

12	SCSI RDMA(Remote Direct Memory Access)
문제	SCSI RDMA(Remote Direct Memory Access)
도메인	네트워크
정의	다른 기기에 연결되어 있는 SCSI 디바이스에 RDMA 프로토콜 기반 접근하는 SCSI Remote 프로토콜
키워드	CPU 미개입, NW 스택이 아닌 RDMA 프로토콜, 저지연, Zero-Copy,
출제의도분석	최근 클라우드, 빅데이터 기술 발전과 더불어 스토리지/디바이스 원격 제어 네트워크 관련 기술 토픽 출제, SAN과 더불어 함께 사용되는 기술
답안작성 전략	TCP 와 RDMA 프로토콜의 차이점을 언급하고, 이를 기반으로 실제 중심 작성
참고문헌	- SCSI RDMA Wikipedia, Linux SRP Initiator(Linux Foundation) - An SRP target mode to improve read performance of SRP-based IB-SAN
풀이 기술사님	김승준 PE (제 117 회 컴퓨터시스템응용기술사 / kimsj243@gmail.com)

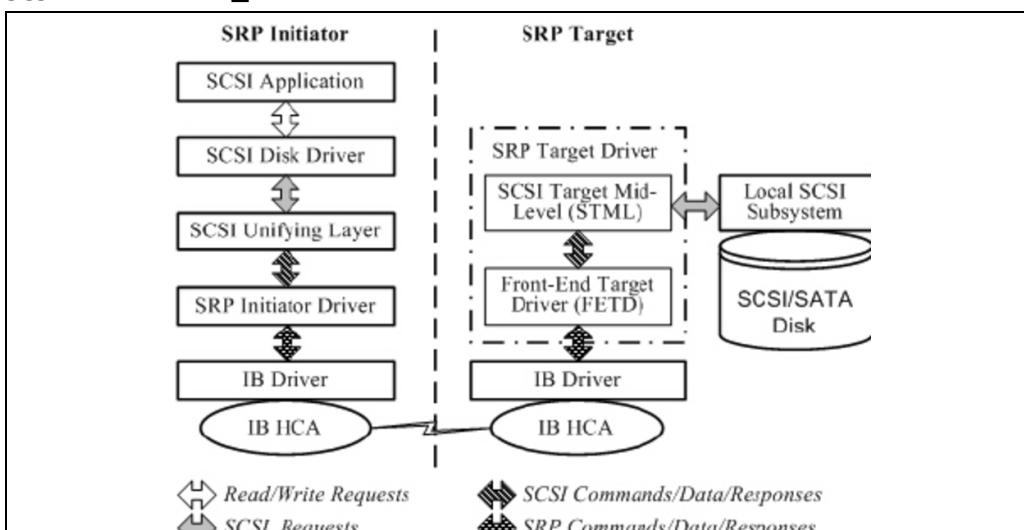
## 1. RDMA 기반 원격지 SCSI 디바이스 접근 프로토콜, SCSI RDMA 정의



- 다른 기기에 연결되어 있는 SCSI 디바이스에 CPU 개입 없이 응용계층에서 직접 RDMA 프로토콜 기반 접근하는 SCSI Remote 프로토콜
- (특징) 인피니 밴드 기반, iSCSI 와 같은 원격 접근 프로토콜, RDMA 기반

## 2. SCSI RDMA 프로토콜 및 설명

### 가. SCSI RDMA 프로토콜



Notes

- Linux에서는 SRP(SCSI RDMA Protocol) Initiator 커널 모듈로 해당 기능 제공

#### 나. SCSI RDMA 프로토콜 상세

분류	프로토콜	설명
SRP Initiator	- 원격 SCSI 디바이스에 접근하는 노드	
	SCSI Application	- SCSI 사용 어플리케이션(SCSI 디바이스 read/write)
	SCSI Unifying Layer	- 여러 SCSI 디바이스 통합 인터페이스 - 각 SCSI 디바이스 별 기능 동작 구현
	SRP Initiator Driver	- SRP 기능 수행 디바이스 드라이버 - 인피니 밴드 HW 연결 및 command 송/수신
	IB Driver/IB HCA	- SRP 프로토콜 송/수신
SRP Target	- 원격에서 접근하는 SCSI 디바이스가 위치하는 노드	
	Front-end Target Driver	- SCSI RDMA 프로토콜 송/수신
	SCSI Mid target Driver	- SCSI RDMA 프로토콜 Local SCSI command 변경
	Local SCSI Sub-System	- SCSI read/write command 수행

- SAN NW에서 고속 연산 및 데이터 송/수신을 위해 SCSI RDMA 프로토콜 사용

"끝"

Notes

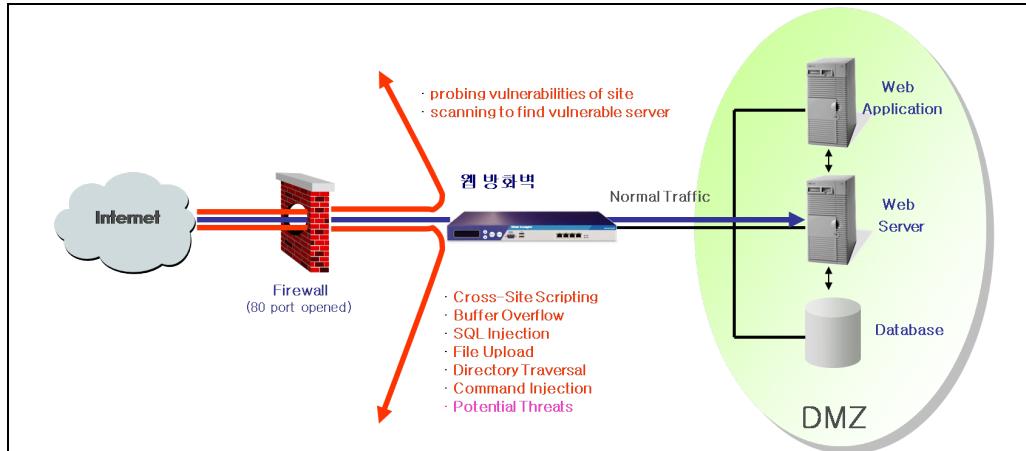
13	WAF(Web Application Firewall)
문제	WAF(Web Application Firewall)
도메인	보안
정의	SQL Injection, XSS 와 같은 웹 공격 및 보안을 위협하거나 과도한 리소스 사용 공격을 탐지하고 차단하는 방화벽 기능
키워드	접근 제어, 로깅 및 감시, Authentication
출제의도분석	웹 기반 서비스 증가로 웹 방화벽 기능에 대한 고전 토픽 출제
답안작성 전략	웹 방화벽 구성도 및 상세 기능 기반 작성 후, 추가적으로 클라우드 시장에서 WAF 제공 및 기능 구현에 대한 내용 언급
참고문헌	- AWS/AhnLab/Google Cloud WAF Solution Overview - 펜타시큐리티 WAF Overview ( <a href="https://www.pentasecurity.co.kr/web-application-firewall/">https://www.pentasecurity.co.kr/web-application-firewall/</a> )
풀이 기술사님	김승준 PE (제 117 회 컴퓨터시스템응용기술사 / kimsj2435@gmail.com)

## 1. 웹 어플리케이션 특화 보안, 웹 방화벽 정의

- SQL Injection, XSS 와 같은 웹 공격 및 보안을 위협하거나 과도한 리소스 사용 공격을 탐지하고 차단하는 방화벽 기능
- (공격유형) 홈페이지 내용 위/변조, 거래 정보 변조, 중요 정보 유출, 악성 코드 유포, XSS, SQL Injection

## 2. 웹 방화벽 구성도 및 기능

### 가. 웹 방화벽 구성도



- XSS, SQL Injection과 같은 웹 공격 패턴 인지, 특정 트래픽 차단 등을 통한 웹 어플리케이션 보안 효율적

### 나. 웹 방화벽 기능

분류	기능	설명
사용자 요청 검사	어플리케이션 접근 제어	- 서비스 사용 및 자원 접근 시, 사용자 권한 식별 기반 접근 제어
	Web Dos 제어	- 과도한 리소스 사용 차단

Notes

	업로드 파일/요청 형식 검사	- 바이러스 및 악성 파일 업로드 검사
	버퍼오버플로우/SQL/스 크립트 차단	- XSS, SQL Injection 등의 웹 공격 차단
컨텐츠 보호	정보 유출 차단	- 신용카드, 주민등록번호 등 유출 차단
	웹 변조 방지	- 응답 형식 검사 및 코드 노출 차단 기반 변조 식별
보안	URL 및 서버 위장	- 사용자 제공 정보(URL/서버정보 등) 일부 변조
	SSL/TLS 지원	- HTTPS 웹 트래픽 암호화 기능 제공

- 접근제어, 사용자 요청 보안 기능 및 패턴 인식 통해 웹 공격 방지

### 3. 퍼블릭 클라우드 환경에서의 WAF 기능 제공

분류	기능	설명
퍼블릭 클라우드	AWS	- 사용자 등록 Rule 기반 WAF 서비스 기능 제공 - Default All-Deny, 사용자 규칙 기반 트래픽 허용
	Google Cloud	- 여러 보안 업체 솔루션(AlterLogic, CloudFlare, ETC)의 보안 솔루션 서비스 제공
	NCloud	- 보안 강화 인프라(Secure Zone)을 두어 중요 정보 보관 및 WAF 기능 제공
프라이빗 클라우드	어플라이언스 서버	- WAF 기능 포함 어플라이언스 서버 기반 증설
	WAF 솔루션 및 컨설팅	- CloudFlare, Akamai 등 WAF 솔루션 기반 구축

- 다양한 웹 공격에 대한 방어로 서비스 안전성 제고 및 AI 기반 다양한 웹 공격 패턴 인식

"끝"