



제129회 컴퓨터시스템응용기술사 해설집

2023.02.04



기술사 포탈 <http://itpe.co.kr> | 국내최대 1위 커뮤니티 <http://cafe.naver.com/81th>

국가기술자격 기술사 시험문제

기술사 제 129 회

제 3 교시 (시험시간: 100 분)

분야	정보통신	자격종목	컴퓨터시스템응용기술사	수험번호		성명	
----	------	------	-------------	------	--	----	--

※ 다음 문제 중 4 문제를 선택하여 설명하시오. (각 10 점)

1. 운영체제 프로세스 스케줄링 알고리즘과 관련하여 단일 큐 멀티프로세서

스케줄링 (SQMS: Single Queue Multiprocessor Scheduling) 과 멀티 큐 멀티프로세서 스케줄링 (MQMS: Multi Queue Multiprocessor Scheduling) 을 각각 설명하시오.

2. 네트워크 스위치 (Network Switch) 와 관련하여 다음을 설명하시오.

(1) 스위치 개요

(2) OSI (Open System Interconnection) 참조모델의 레이어에 따른 스위치 유형

(3) L4 스위치와 L7 스위치 비교

3. 최근 의료기관을 대상으로 사이버 침해공격이 지능화 및 다양화되고 있다. 이에 대한

대안으로 병원정보시스템 (HIS: Hospital Information System) 에 특화된

정보공유센터 (ISAC : Information Sharing Analysis Center) 의 역할이 증가하고 있다.

이와 관련하여 다음을 설명하시오.

(1) HIS 의 개요 및 구성

(2) ISAC 의 개요 및 역할

(3) 의료기관의 사이버침해공격 대응방안

4. 양자 컴퓨터(Quantum Computer)와 관련하여 다음을 설명하시오.

(1) 양자비트(Qubit, 큐비트), 양자 우월성(Quantum Supremacy),

양자 결함(Quantum Decoherence)

(2) 양자 컴퓨터 구현 방법

5. 이더넷(Ethernet) 표준에 대하여 다음을 설명하시오.

(1) 정의 및 특징

(2) IEEE 802.3 프레임(Frame) 구조

(3) 전송 프레임의 최소 크기가 64 바이트인 이유

6. DDR SDRAM(Double Date Rate Synchronous DRAM)의 갱신(Refresh)에 대하여 다음을 설명하시오.

(1) DRAM에서 갱신(Refresh)이 필요한 이유

(2) 자동 갱신(Auto Refresh)

(3) 셀프 갱신(Self Refresh)

(4) 저전력 자동 셀프 갱신(LPASR : Low Power Auto Self Refresh)

01	멀티프로세서 스케줄링		
문제	운영체제 프로세스 스케줄링 알고리즘과 관련하여 단일 큐 멀티프로세서 스케줄링(SQMS: Single Queue Multiprocessor Scheduling)과 멀티 큐 멀티프로세서 스케줄링(MQMS: Multi Queue Multiprocessor Scheduling)을 각각 설명하시오.		
도메인	운영체제	난이도	중(상/중/하)
키워드	단일 큐, 다중 큐, 워크로드 불균형, Load Balancing(Push/Pull Migration), O(1) Scheduler, CFS(Completely Fair Scheduler), BFS(Brain Fuck Scheduler)		
출제배경	멀티프로세서 환경에서의 스케줄링 방법에 대한 출제예상		
참고문헌	운영체제 : 아주 쉬운 세 가지 이야기(원유집, 박민규, 이성진)		
해설자	소원반 소민호 기술사(제 119회 정보관리기술사 / mhsope@naver.com)		

I. 멀티프로세서 스케줄링의 개념

<pre> graph LR A[멀티프로세스 스케줄링] --> B[SQMS] A --> C[MQMS] </pre>	<ul style="list-style-type: none"> - 하나의 스케줄링 큐 - BFS(Brain Fuck Scheduler) <ul style="list-style-type: none"> - 다수의 스케줄링 큐 - O(1) Scheduler, CFS(Completely Fair Scheduler)
개념	<ul style="list-style-type: none"> - 멀티 프로세서 환경에서 프로세스의 실행을 위해 어느 CPU에 프로세스를 할당할 것인지의 순서를 결정하는 정책

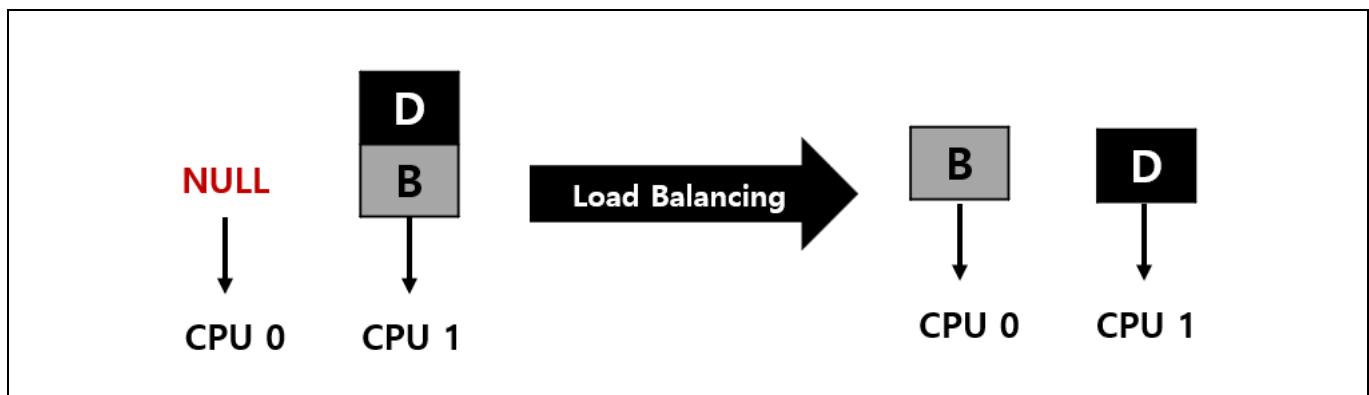
II. 단일 큐 멀티프로세서 스케줄링(SQMS: Single Queue Multiprocessor Scheduling)의 설명

구분	설명																												
개념	<ul style="list-style-type: none"> - 하나의 스케줄링 큐를 구성하여 CPU를 할당 받아 Round Robin등의 알고리즘으로 스케줄링 방식 																												
개념도	<p style="text-align: center;">Queue → A → B → C → D → E → NULL</p> <p style="text-align: center;">할당 ↓</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>CPU 0</td> <td>A</td> <td>E</td> <td>D</td> <td>C</td> <td>B</td> <td>... (repeat) ...</td> </tr> <tr> <td>CPU 1</td> <td>B</td> <td>A</td> <td>E</td> <td>D</td> <td>C</td> <td>... (repeat) ...</td> </tr> <tr> <td>CPU 2</td> <td>C</td> <td>B</td> <td>A</td> <td>E</td> <td>D</td> <td>... (repeat) ...</td> </tr> <tr> <td>CPU 3</td> <td>D</td> <td>C</td> <td>B</td> <td>A</td> <td>E</td> <td>... (repeat) ...</td> </tr> </table>	CPU 0	A	E	D	C	B	... (repeat) ...	CPU 1	B	A	E	D	C	... (repeat) ...	CPU 2	C	B	A	E	D	... (repeat) ...	CPU 3	D	C	B	A	E	... (repeat) ...
CPU 0	A	E	D	C	B	... (repeat) ...																							
CPU 1	B	A	E	D	C	... (repeat) ...																							
CPU 2	C	B	A	E	D	... (repeat) ...																							
CPU 3	D	C	B	A	E	... (repeat) ...																							
장점	<ul style="list-style-type: none"> - 하나의 스케줄링 큐밖에 없기 때문에 구현이 간단 - 기존 정책을 다수 CPU에서 동작하도록 하는 데는 많은 변경이 필요치 않음 - 워크로드의 균형을 맞추기 용이 																												
단점	<ul style="list-style-type: none"> - 동기화 오버헤드 때문에 확장성(scalability)이 결여 - 많은 개수의 프로세서에 대한 캐시 친화성이 좋지 않음 																												
사례	<ul style="list-style-type: none"> - BFS(Brain Fuck Scheduler) 																												

III. 멀티 큐 멀티프로세서 스케줄링(MQMS: Multi Queue Multiprocessor Scheduling)

구분	설명																		
개념	<ul style="list-style-type: none"> - CPU마다 별도의 스케줄링 큐를 구성하여 CPU 각각이 독립적으로 스케줄링 하는 방식 - 각각의 스케줄링 큐는 Round Robin과 같은 특정한 스케줄링 방법을 사용 																		
개념도	<p style="text-align: center;">Queue 0 → A → C Queue 1 → B → D</p> <p style="text-align: center;">Queue 0 → CPU 0 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>A</td><td>A</td><td>C</td><td>C</td><td>A</td><td>A</td><td>C</td><td>C</td><td>...</td></tr></table></p> <p style="text-align: center;">Queue 1 → CPU 1 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>B</td><td>B</td><td>D</td><td>D</td><td>B</td><td>B</td><td>D</td><td>D</td><td>...</td></tr></table></p>	A	A	C	C	A	A	C	C	...	B	B	D	D	B	B	D	D	...
A	A	C	C	A	A	C	C	...											
B	B	D	D	B	B	D	D	...											
장점	<ul style="list-style-type: none"> - 확장성이 좋음, 락(Lock)과 캐시 경합(cache contention) 문제 해결 - 캐시 친화적, 작업이 같은 CPU에서 계속 실행되기 때문에 캐시에 저장된 내용 재사용 가능 - 큐가 분리되어 있기 때문에 공유데이터에 대한 Race condition 해결 - Queue 마다 별도의 알고리즘 적용 가능 																		
단점	<p style="text-align: center;">Queue 0 Queue 1 → B → D</p> <p style="text-align: center;">Queue 0 → CPU 0</p> <p style="text-align: center;">Queue 1 → CPU 1 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>B</td><td>B</td><td>D</td><td>D</td><td>B</td><td>B</td><td>D</td><td>D</td><td>...</td></tr></table></p> <ul style="list-style-type: none"> - CPU하나가 유휴상태인 워크로드의 불균형(load imbalance) 문제 발생 가능 - 구현이 복잡함 	B	B	D	D	B	B	D	D	...									
B	B	D	D	B	B	D	D	...											
사례	<ul style="list-style-type: none"> - O(1) Scheduler, CFS(Completely Fair Scheduler) 																		

- 워크로드 불균형 문제를 해결하기 위해 **Load Balancing(Migration)** 기법 적용

IV. 워크로드 불균형 해결, Load Balancing (Migration)


구분	설명	
개념	<ul style="list-style-type: none"> - MQMS의 워크로드 불균형 문제를 해결하기 위해 비어 있는 큐에 다른 큐에 있는 프로세스를 이관하는 방법 	
종류	Push Migration	<ul style="list-style-type: none"> - 주기적으로 각 프로세서를 체크하여 작업이 많은 프로세서인 경우 다른 프로세서로 작업을 분배하는 방식
	Pull Migration	<ul style="list-style-type: none"> - Idle 상태에 놓인 프로세서가 직접 작업이 많은 프로세서로부터 Queue에 대기하고 있는 작업을 가져오는 방식

“끝”

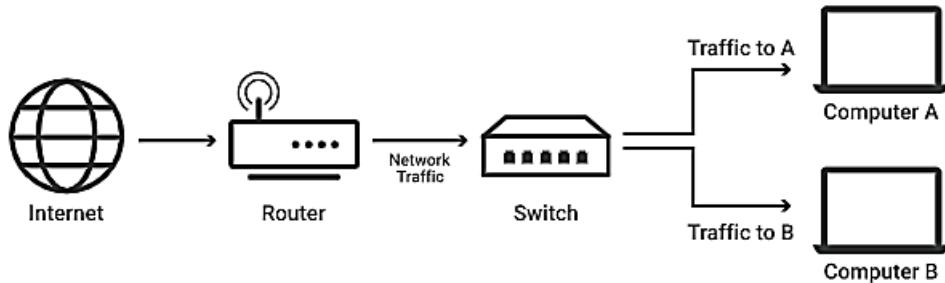
[참고] Linux 멀티프로세서 스케줄러 종류

스케줄러	큐	설명
O(1) Scheduler	멀티큐	<ul style="list-style-type: none"> - 우선순위 기반 스케줄러로서(MLFQ와 유사) 프로세스의 우선순위를 시간에 따라 변경하여 우선순위가 가장 높은 작업을 선택하는 방식
CFS (Completely Fair Scheduler)	멀티큐	<ul style="list-style-type: none"> - 프로세스의 상대적인 가중치를 이용하여 수행시간을 할당하는 방식 - 실행될 프로세스를 virtual runtime라는 값을 인덱스로 하여 레드블랙 트리(red-black tree)로 실행 큐를 구성하고, 스케줄링 시점이 되면, 레드블랙 트리 실행 큐에서 virtual runtime값이 가장 작은 프로세스를 선택하여 실행 - 2.6.23 커널 버전 이후 적용된 리눅스의 기본 스케줄러
BFS (Brain Fuck Scheduler)	단일큐	<ul style="list-style-type: none"> - 단일 이중 연결 리스트 큐를 구성하여 Virtual Deadline First 정책에 따라 실행할 프로세스 선택하는 방식

02	네트워크 스위치(Network Switch)		
문제	네트워크 스위치(Network Switch)와 관련하여 다음을 설명하시오. (1) 스위치 개요 (2) OSI(Open System Interconnection) 참조모델의 레이어에 따른 스위치 유형 (3) L4 스위치와 L7 스위치 비교		
도메인	네트워크	난이도	중(상/중/하)
키워드	전 이중 통신, MAC 주소, 소규모 통신, L7, L4, L3, L2		
출제배경	네트워크 기본 지식의 이해 확인		
참고문헌	ITPE 서브노트		
해설자	강남평일야간반 전일 기술사(제 114회 정보관리기술사 / nikki6@hanmail.net)		

I. 소규모 통신을 위한, 네트워크 스위치 개요

가. 네트워크 스위치(Network Switch) 개념



- 소규모 비즈니스 네트워크 안에서 컴퓨터, 프린터, 서버 등 모든 디바이스를 서로 연결함으로써 리소스를 쉽게 공유할 수 있도록 하는 네트워크 장치

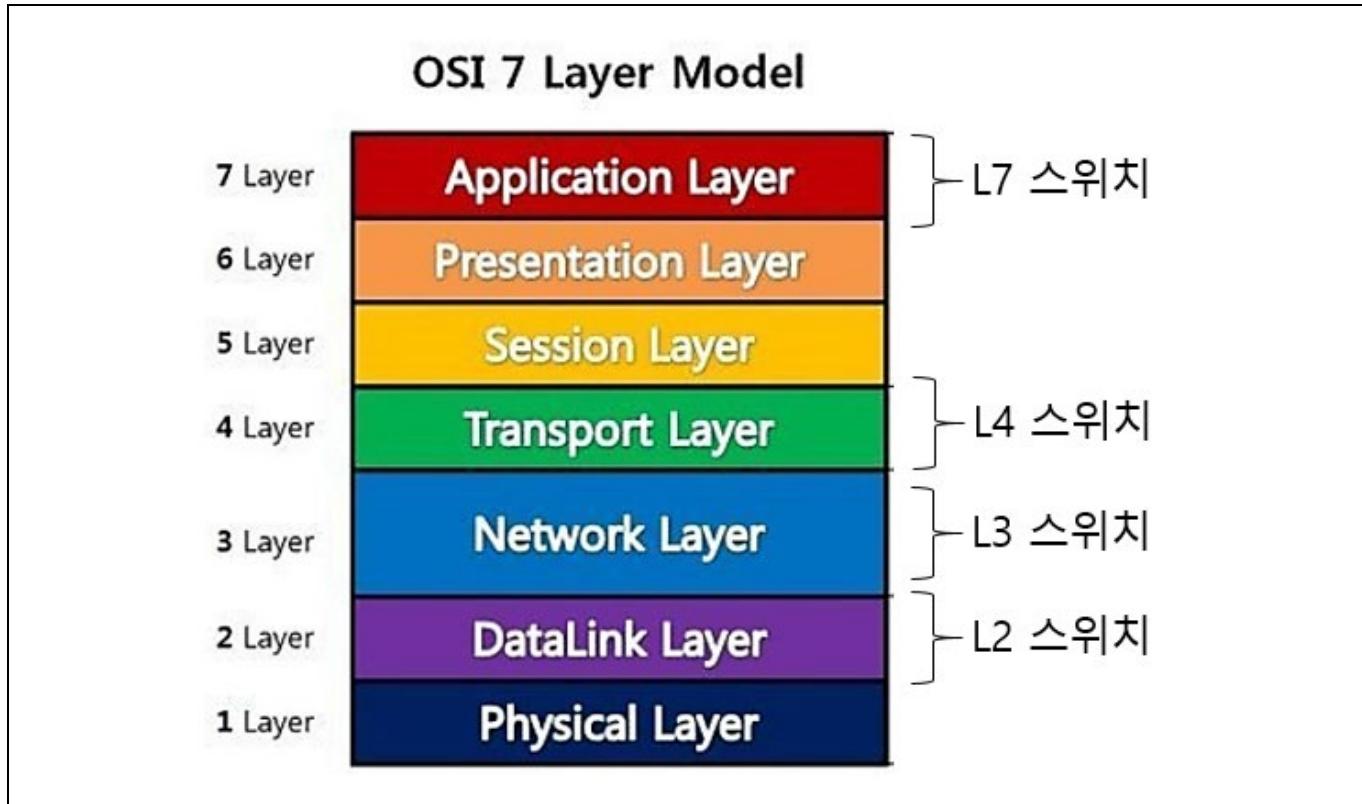
나. 네트워크 스위치(Network Switch) 특징

특징	상세 설명
향상된 네트워크 속도 제공	- 각 컴퓨터에서 주고 받는 데이터가 허브처럼 다른 모든 컴퓨터에 전송되는 것이 아니라, 데이터를 필요로 하는 컴퓨터에만 전송
전 이중 통신	- 대부분의 스위치는 전 이중 통신방식(full duplex)을 지원하기 때문에 송신과 수신이 동시에 일어나는 경우 훨씬 향상된 속도를 제공
MAC 주소 기반	- 고유한 MAC 주소를 기억하고 있어야 하며, 이 주소를 통해 어떤 데이터가 어디로 전송되어야 하는지 판단
소규모 통신 적합	- 스위치를 이용하는 경우도 대량의 브로드 캐스팅이나 스위치의 처리용량을 초과하는 데이터 흐름에 대해서는 취약할 수밖에 없으므로 커다란 네트워크의 경우는 VLAN 스위치나 라우터 등을 사용

- 네트워크 스위치라는 이름 대신 브리징 허브, MAC 브리지, 스위칭 허브, 포트 스위칭 허브라고도 사용

II. OSI(Open System Interconnection) 참조모델의 레이어에 따른 스위치 유형

가. OSI 참조모델의 레이어에 따른 스위칭 위치



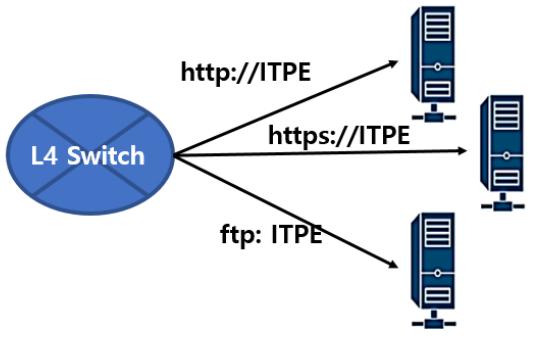
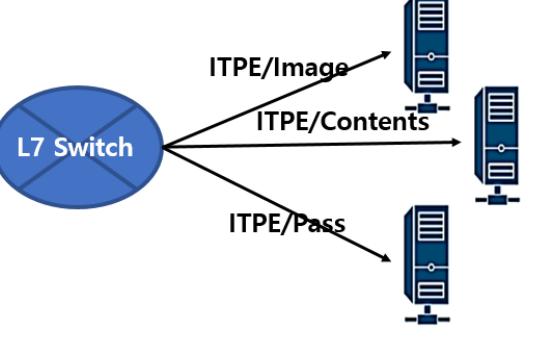
나. 스위치 유형별 상세 내용

OSI 7	스위치 유형	상세 설명
L7	- L7 스위치	- L7 layer에 해당하는 Application 정보를 바탕으로 지능적으로 트래픽 관리를 수행하는 네트워크 스위치 장비
L4	- L4 스위치	- TCP/UDP 포트 정보를 이용하여 네트워크 트래픽을 분산하고 경로를 결정
L3	- L3 스위치	- 수신된 패킷이 외부에 존재하는 IP 일 경우 그 패킷을 외부에 연결된 라우터로 보내주어 내부망 뿐 아니라 외부망에 IP기반 통신이 가능
L2	- L2 스위치	- 느린 전송속도의 브리지, 허브의 단점을 개선하기 위해서, 출발지에서 들어온 프레임을 목적지 MAC 주소 기반으로 빠르게 포워딩

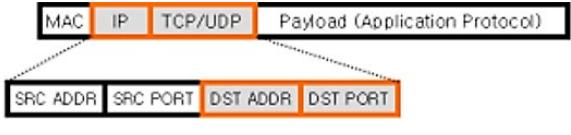
- 레이어가 높을수록 스위치 가격이 고가이며, 각각의 스위치는 자기보다 낮은 레이어 스위치 기능까지 다 할 수 있음

III. L4 스위치와 L7 스위치 비교

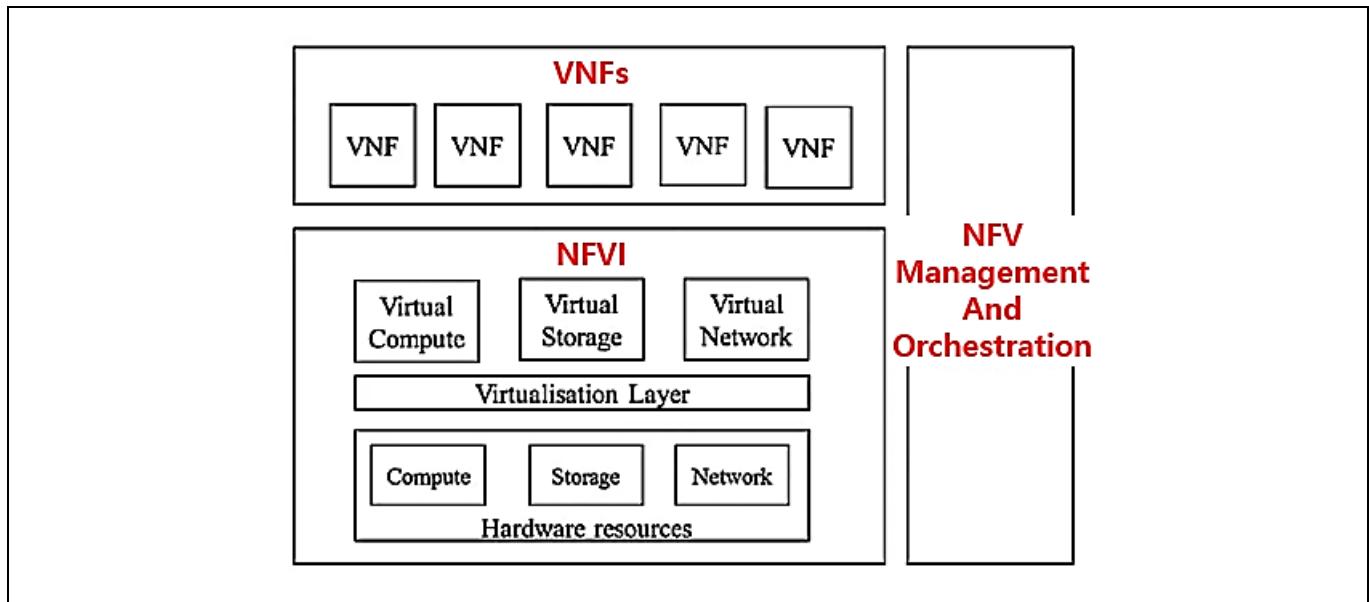
가. L4 스위치와 L7 스위치의 개념 비교

구분	L4 스위치	L7 스위치
개념도		
개념	<ul style="list-style-type: none"> - L4 스위치는 L4 주소를 가지고 스위칭을 하는 장비 - L4 주소는 포트번호를 의미 - 똑같은 IP에 대하여 포트번호가 다를 경우 다른 서버로 보낼 수 있음 	<ul style="list-style-type: none"> - L7 스위치는 URL주소로 스위칭하는 장비 - L7은 어플리케이션 영역

나. L4 스위치와 L7 스위치의 장단점 비교

구분	L4 스위치	L7 스위치
패킷 구조		
장점	<ul style="list-style-type: none"> - Virtual IP 제공으로 성능확장 및 가용성 확보 용이 - 성능향상과 장애대처(Fail-over) 통한 서비스 품질 보장 	<ul style="list-style-type: none"> - Payload의 분석으로 더 지능적인 로드 밸런싱 이 가능 - 보안성 기능 강화(악성 코드 등 바이러스 필터 링)
단점	<ul style="list-style-type: none"> - 단순 포트레벨의 스위칭으로 보안상의 이슈 가 있을 수 있음 - Sticky Session 적용 시 회사 내부에서 외부로 나가는 경우 여러 사람이 timeout 시간 내에 접속하는 경우 한 서버에만 로드가 집중 됨 	<ul style="list-style-type: none"> - 컨텐츠 추가 분석으로 인한 Performance 저하 발생 가능 - L4 스위치보다 기타 Cost가 추가로 발생할 수 있음

- 최근 트래픽 및 비용의 폭발적 증가로 인해 기존 스위칭 장비보다 유연성이 뛰어나고 추가 장비 도입 없이 네트워크 장비를 재활용할 수 있는 NFV 기술 등장

IV. 네트워크 기능 가상화, NFV(Network Function Virtualization)

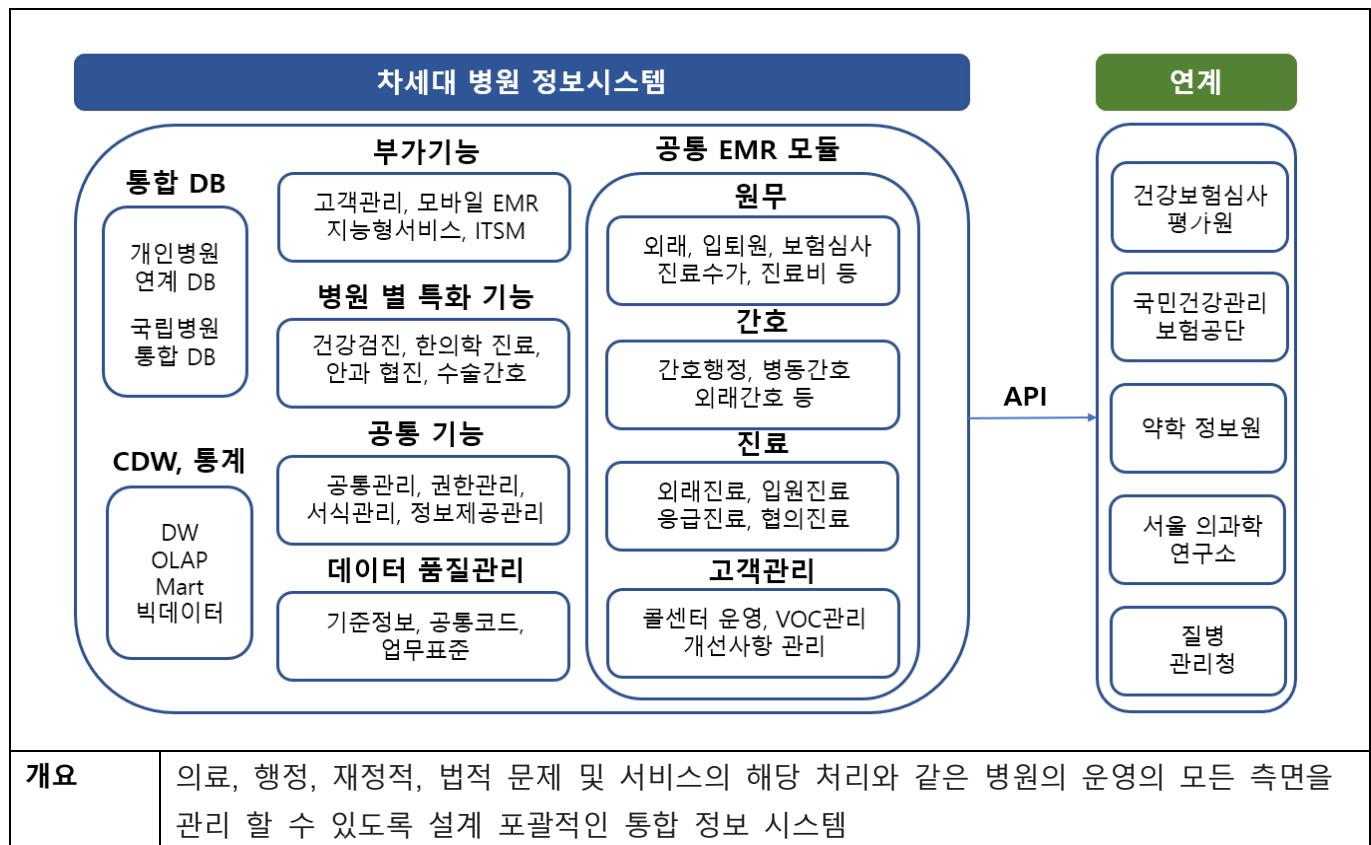
- 기존 Hardware Appliances들로 구현된 Route, NAT, Firewall, IDS, IPS, DNS, Caching등의 다양한 기능을 Software 형태의 Virtual Appliances로 구현하여 운용하는 가상화 기술
- 표준 IT 가상화 기술을 활용하여 기존의 단일 Appliance로 동작하던 장비들을 Software형식으로 Virtual Appliance로 구현

“글”

03	의료기관 사이버 침해		
문제	<p>최근 의료기관을 대상으로 사이버 침해공격이 지능화 및 다양화되고 있다. 이에 대한 대안으로 병원정보시스템(HIS:Hospital Information System)에 특화된 정보공유센터(ISAC : Information Sharing Analysis Center)의 역할이 증가하고 있다. 이와 관련하여 다음을 설명하시오.</p> <p>(1) HIS의 개요 및 구성 (2) ISAC의 개요 및 역할 (3) 의료기관의 사이버침해공격 대응방안</p>		
도메인	디지털서비스/보안	난이도	상(상/중/하)
키워드	PACS, OCS, EMR, HER, 사이버 위협, 보안, 침해 위험 분석		
출제배경	의료기관 정보공유기술 관련 이해도 확인과 사이버 공격에 대한 의견 제시		
참고문헌	<p>차세대 국립병원 정보시스템 구성_보건복지부 의료분야 랜섬웨어 예방, 대응 퀴메뉴얼_보건복지부, 한국사회보장정보원 2022년 의료기관 진료정보보호 추진방향_보건복지부</p>		
해설자	정상 기술사(제 12X회 정보관리기술사 / jeongsang_pe@naver.com)		

I. 의료 정보 관리의 효율화, HIS의 개요 및 구성

가. HIS의 개요



- 다양한 IT 기술을 의료 산업에 적용하여 효율화를 위한 병원정보시스템 구축 진행 중

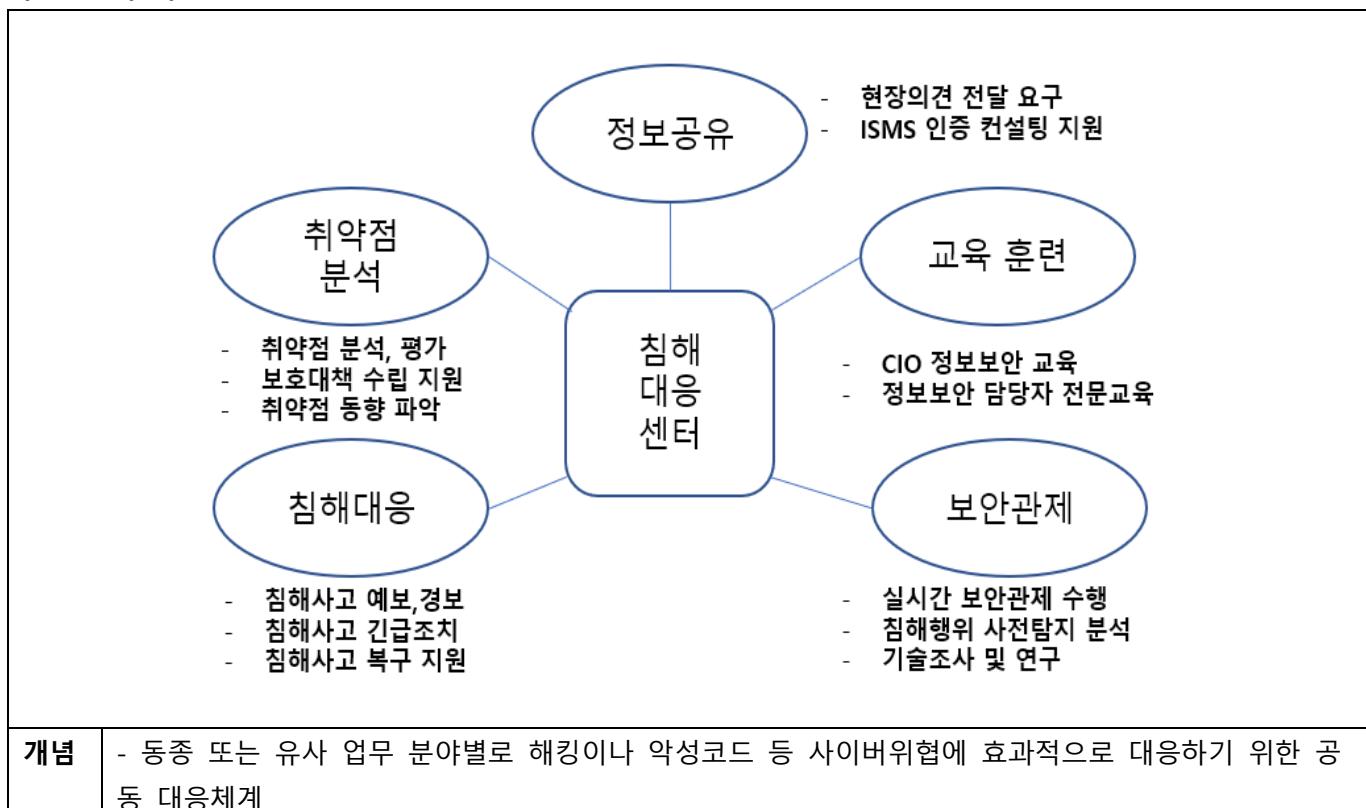
나. HIS의 구성요소

구분	핵심 기술	설명
의료기술	PACS	- 의료영상을 저장 및 전송하기 위한 시스템. DICOM 준수 영상의 획득 장비, 저장 전송 장비, 출력 장비로 구성
	OCS	- 처방(Order)을 전산처리하기 위한 진료, 진료지원(영상, 약재, 병리, 재활치료, 식이), 원무행정부서간 전산 시스템
	EMR	- 진료 관련 모든 기록을 전산 데이터화(Paperless System) 시켜서 임상의학연구 핵심 기반(DW, KDB) 구축. 의료사고 예방 또는 조기발견이 핵심적 목표 기능
	EHR	- 진료기록뿐만 아닌 의료소비자 평생건강정보 전산화된 자료 임상용어/코드 표준화, 컨텐츠 모델 개발하고 이를 통한 개인 건강기록
기반기술	클라우드	- 클라우드 기반의 확장으로 고객 데이터 분석 관리
	인공지능	- 비지도, 지도, 강화학습을 통한 환자 정보 및 해당 병 관련 솔루션 제공
	데이터베이스	- NoSQL 과 같은 데이터 베이스 구축으로 비정형 데이터 처리 저장
	보안	- 외부 침입에 대한 방화벽, 접근제어등의 보안 기술 적용

- 병원정보시스템은 다양한 신기술을 적용된 의료 핵심 시스템

II. ISAC 개념과 역할

가. ISAC의 개념



- 사이버 위협에 대한 대응을 목표로 함

나. ISAC의 역할

역할	설명
기반시설 보호	- 국가의 중요 정보통신기반시설에 대한 보호를 위한 주기적인 분석
보안구조 확보	- 관리주체로서 영역 내의 주요기반 시설에 대한 보안 구조 확보
신뢰성 확보	- 대외적인 상호의존적 영역과 회원사의 만족도 신뢰성 최대화
위협 분석	- 모든 위협들에 대한 신속 정확한 보안 위협 분석

- ISAC에서 지속적인 보안 위협에 대한 부분 신뢰성 확보 노력

III. 의료기관의 사이버침해공격 대응방안

가. 의료기관의 사이버침해공격 대응방안



- 의료기관의 시스템에서도 다양한 사이버 침해 지속 발생으로 대응방안 마련 시급

나. 의료기관의 사이버침해공격 대응방안 상세설명

구분	대응방안	설명
상시 보안 관제	보안로그 분석	- 보안장비(방화벽, IDS/IPS, 웹방화벽 등)로부터 실시간으로 보안위협 로그를 취합, 분석 및 대응
	네트워크 트래픽 분석	- 네트워크 트래픽에 대해 실시간으로 위협 요소 및 이상 행위를 탐지하고 분석 및 대응
	웹페이지 모니터링 (위변조 /웹쉘/ 악성코드)	- 웹페이지에 대한 위변조 탐지, 웹쉘 삽입, 악성코드 유포
침해행위 탐지, 분석	의심 분석, 점검요청	- 보안관제로 탐지된 이벤트 중 침해행위로 의심되는 건에 대해 분석을 통해 점검(또는 조치) 요청 또는 권고
사이버 위협 분석	탐지 이벤트 연관분석	- 내·외부로부터 신규 취약점 또는 위협 정보를 수집하고 보안관제로 탐지된 이벤트와의 연관분석 수행

“끝”

04	양자 컴퓨팅		
문제	양자 컴퓨터(Quantum Computer)와 관련하여 다음을 설명하시오. (1) 양자비트(Qubit, 큐비트), 양자 우월성(Quantum Supremacy), 양자 결잃음(Quantum Decoherence) (2) 양자 컴퓨터 구현 방법		
도메인	디지털서비스	난이도	상(상/중/하)
키워드	이온 트랩, 초전도체 기반		
출제배경	양자 컴퓨팅 관련 기술이 지속 발전함에 따라 최신 기술에 대한 지식 확인을 위한 출제		
참고문헌	위키피디아		
해설자	서경석 기술사(제119회 정보관리기술사 / akslempf@naver.com)		

I. 컴퓨팅 기술의 퀀텀 점프, 양자 컴퓨터의 개요

가. 양자 컴퓨터의 정의

- 기존 컴퓨터가 0과 1만 구분할 수 있는 반면, 0과 1을 동시에 공존시키며 양자 얹힘, 중첩, 이동 등 양자 성질을 기반으로 연산을 진행하는 차세대 컴퓨터

나. 기존 컴퓨터와 양자 컴퓨터의 비교

구분	기존 컴퓨터	양자 컴퓨터
연산 개념	<ul style="list-style-type: none"> - 정보를 0과 1로 표현 - 1개씩 순차적으로 계산 진행 	<ul style="list-style-type: none"> - 정보를 0과 1을 중첩 시켜 표현 - 중첩된 내용 기반 한 번에 연산 처리
기본 단위	- Bit(0 또는 1)	- Qubit
연산 방법	- 논리 표에 의한 계산	- 행렬 함수에 의한 계산
외부 잡음	- 오류 정정 용이	- 오류 정정 어려움
n비트의 정보량	- $0 \sim 2^n - 1$ 중 1개 값만 기억	- 2^n 의 모든 값을 중첩 기억
연산 동작 처리량	<ul style="list-style-type: none"> - n bit ALU는 1번 연산 동작 - 3비트의 경우 처리를 위해 8번 연산 	<ul style="list-style-type: none"> - n Qubit는 2^n 연산 동작 - 3 Qubit의 경우 1번 연산

- 무어의 법칙 한계 극복, 에너지 비효율성 및 비정형 데이터 처리 성능 향상을 위해 양자 컴퓨터 연구 지속

II. 양자 비트, 양자 우월성, 양자 결맞음 설명

가. 양자 비트(Qubit)의 개념 및 설명

구분	설명	
개념	- 2개의 구분 가능한 기저 상태가 중첩된 양자 상태로 존재할 수 있는 양자 정보 시스템에서 사용 되는 최소 정보의 단위	
양자 비트의 상태	<ul style="list-style-type: none"> - 고전 비트는 0과 1 사이의 선택 - 상태 $0\rangle$과 상태 $1\rangle$의 중첩 상태로 존재 - 상태 표현 수식 : $\alpha 0\rangle + \beta 1\rangle$ 	
양자 비트 규격	<ul style="list-style-type: none"> - 양자 비트를 측정할 경우 $\alpha ^2$의 확률로 0의 값을, $\beta ^2$의 확률로 1의 값을 - 특별한 경우가 아닌 이상 $\alpha ^2 + \beta ^2 = 1$을 만족하는 규격화된 형태로 활용 	
블로흐 구		<ul style="list-style-type: none"> - 정의 : 임의의 양자 비트를 나타내는 반지름이 1인 구 - 반지름이 1인 구에서 북극을 $0\rangle$, 남극을 $1\rangle$이라고 가정 - 하나의 양자 비트는 θ를 극각, ϕ를 구면 좌표의 방위각으로 하는 구면 상의 한 점으로 표현 가능

- 고전 비트와 전혀 다른 특성과 상태를 보유하여 양자 컴퓨터의 기본 연산 단위로 사용

나. 양자 우월성(Quantum Supremacy)의 개념 및 설명

구분	설명	
개념	- 양자 컴퓨터가 기존 슈퍼 컴퓨터의 연산 능력을 혁신적으로 뛰어 넘은 상태	
동향	<ul style="list-style-type: none"> - 2019년 구글이 시커모어 칩을 활용한 양자 컴퓨터 개발 - 시커모어 칩은 53개 큐비트로 구성 	
달성 주장	- 기존 컴퓨터 연산 시 만년 이상의 시간이 소요 되는 pseudo-random quantum circuit problem 연산을 200초 내에 해결하였다고 주장	
연산 환경	<ul style="list-style-type: none"> - 50개 이상의 양자 비트의 결맞음 시간을 수백초 단위로 0.2% 이하의 오류 수준 유지 - 결맞음 상태 제어를 통한 양자 컴퓨터 연산의 기본 환경 달성 	
논란	- pseudo-random quantum circuit problem 문제를 기준 슈퍼 컴퓨터로 연산 시 2.5일 이내에 연산 가능 하다는 반박	
향후	<ul style="list-style-type: none"> - 근본적으로 고전 컴퓨터 구조에서 연산이 어려운 문제의 해결을 통해 확인 필요 - 사례 : 보손 샘플링, P 대 NP 문제 	

- 양자 우월성 달성을 기술적 가능성에 대해 논란이 있는 상태로 지속적 발전 필요

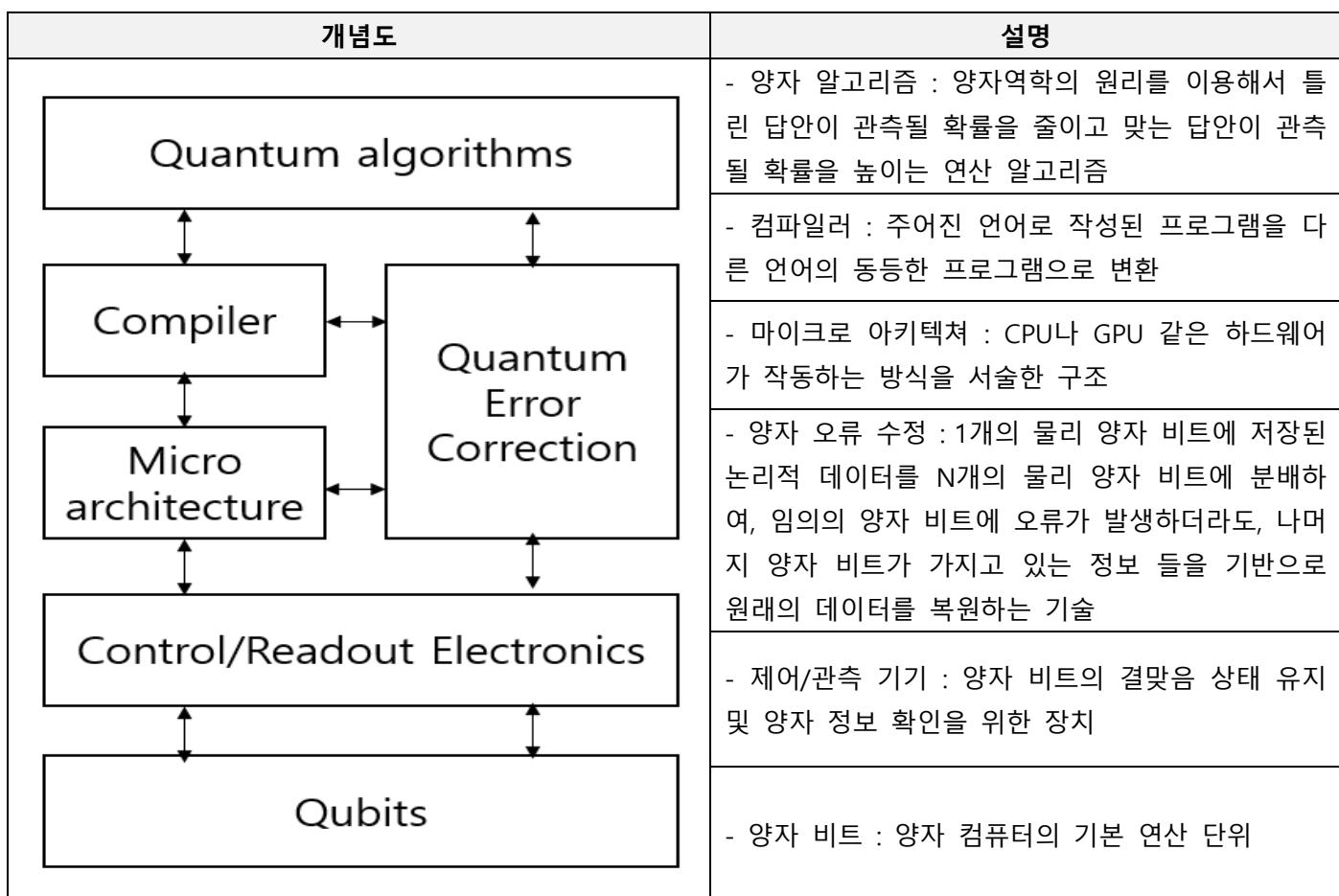
다. 양자 결잃음(Quantum Decoherence)의 개념 및 설명

구분	설명
개념	- 한 양자계가, 그 자체만으로 간섭(interference) 현상을 일으킬 수 있는 상태를 결맞음이라고 하며, 결잃음이란 외부와의 상호작용을 통하여 결맞음을 손실한 상태
결잃음 발생 절차	1. 중첩 상태의 에너지 레벨 a, b를 동시에 보유한 원자가 존재 2. 관찰 발생 시 랜덤하게 둘 중 하나의 상태로 귀결(슈뢰딩거의 고양이) 3. 원자는 고립 존재가 불가하여 다른 원자와 상호 작용 발생에 따른 중첩 상태 깨짐 4. 처음에 원자는 a, b 중첩 상태 가능하지만 다른 원자와의 교류(결잃음)로 인해 결잃음 상태로 전환 5. 중첩 상태 유지 불가, a, b 둘 중 하나의 상태로 고정
결잃음 상태 분포	$\rho = \sum_{i=1}^d p_i i\rangle \langle i $ - 임의의 p_i 확률 분포에 대해 위와 같이 정의되며, 이를 제외한 모든 상태가 결잃음

- C60 문자를 활용해 이중 슬릿 실험 수행 결과 간섭 무늬가 확인 되어 인슐린 문자 대상으로 실험 진행 중

III. 양자 컴퓨터의 개념도 및 구현 방법

가. 양자 컴퓨터의 개념도



- 양자 비트의 결맞음 상태 유지를 위해 다양한 구현 방법의 연구가 진행 중

나. 양자 컴퓨터 구현 방법

구분	설명
초전도 양자 비트	- 초전도체에 흐르는 초전도 전류의 공진 회로를 구성하고, 이 회로에 마이크로파를 가해 초전도 전류의 중첩 상태를 이용하는 구현 방식
중성 원자	- 중성 원자의 경우 광격자를 이용하여 단일 원자 양자 비트 생성 - 포획된 원자에서 선택적으로 양자 비트를 생성하거나, 주변 원자와의 게이트 조작 이용하여 양자 시뮬레이션 구현 방식
이온 트랩	- 동적인 전자기장을 이용하여, 전하를 띤 입자들을 포획하여 이온의 양자 상태를 제어하고 측정 하여 이온들의 양자 얹힘을 구현하는 방식
광자	- 특별히 상호작용하는 물질과 접촉하지 않는 이상 환경과 상호작용을 잘 하지 않아 결맞음이 오래 유지 되는 광자의 속성을 이용하여 구현하는 방식
반도체	- 반도체 내에서 양자점을 형성하는 게이트를 제작하여 개별 전자를 포획한뒤 게이트 전위 차 등을 이용하여 전기적으로 포획 전자의 양자 상태를 제어, 구현하는 방식
NMR	- 분자 내에 존재하는 핵스핀의 상태를 양자 비트로 이용하고, 핵스핀 간의 상호작용을 활용 하여 주위 분자들과의 상호작용에 의해 구분 가능한 양자 비트 확장 방식
위상 양자	- 구분 불가능한 입자가 갖는 통계적 특성에 기인한 2 차원 준입자인 애니온을 활용하여 양자정보를 저장하고 시뮬레이션하는 방식

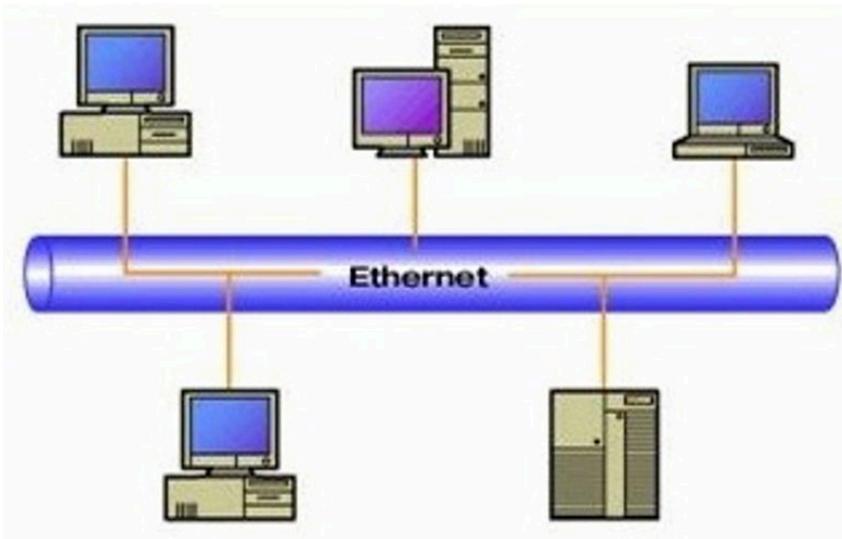
- 현재까지 완벽하게 구현된 양자 컴퓨터는 없는 상태로 다양한 구현 방식의 연구, 개발 진행 중

“끝”

05	이더넷(Ethernet)		
문제	이더넷(Ethernet) 표준에 대하여 다음을 설명하시오. (1) 정의 및 특징 (2) IEEE 802.3 프레임(Frame) 구조 (3) 전송 프레임의 최소 크기가 64바이트인 이유		
도메인	네트워크	난이도	중(상/중/하)
키워드	CABLE, CSMA/CD, 이더넷 프레임		
출제배경	네트워크 layer1, 2 기본 지식 확인		
참고문헌	ITPE 기술사회 자료		
해설자	이상용 기술사(제 124회 정보관리기술사 / orangeday77@gmail.com)		

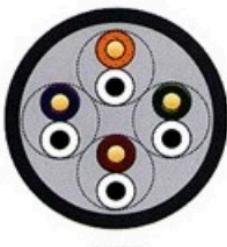
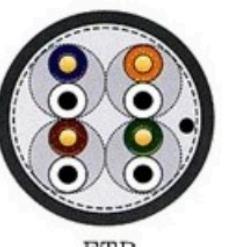
I. 네트워크 연결 핵심 기술, 이더넷(Ethernet)의 정의 및 특징

가. 이더넷(Ethernet)의 정의

개념도	
정의	<ul style="list-style-type: none"> - 48비트 길이의 고유 MAC 주소를 기반으로 네트워크에 연결된 기기들이 상호간 데이터를 교환할 수 있는 기술

- CSMA/CD 매체접근제어를 사용한 네트워크 기술

나. 이더넷(Ethernet)의 특징

케이블			
	<ul style="list-style-type: none"> - UTP(Unshielded Twisted Pair): 차폐 기능이 없는 기본적인 랜선 - FTP(Foil screened Twisted Pair): 피복 안쪽을 호일로 한 번 감싸고 그 호일 안쪽에 구리접지선을 추가한 랜선 - STP(Shielded Twisted Pair): 구리선을 한 묶음으로 나누어 한 번 더 호일을 감싸 준 랜선 		
속도	<ul style="list-style-type: none"> - 1 Gbps - 100 Mbps - 10 Mbps 		
방식	<ul style="list-style-type: none"> - 10BASE-T (초당 10메가비트를 지원하는 이더넷) - 100BASE-TX (초당 100메가비트를 지원하는 이더넷) - 100BASE-FX (광케이블을 이용해 초당 100메가비트를 구현하는 이더넷) - 1000BASE-T (초당 1기가비트를 지원 이더넷) - 1000BASE-SX (멀티모드 광케이블을 이용해 초당 1기가비트를 전송 이더넷) 		

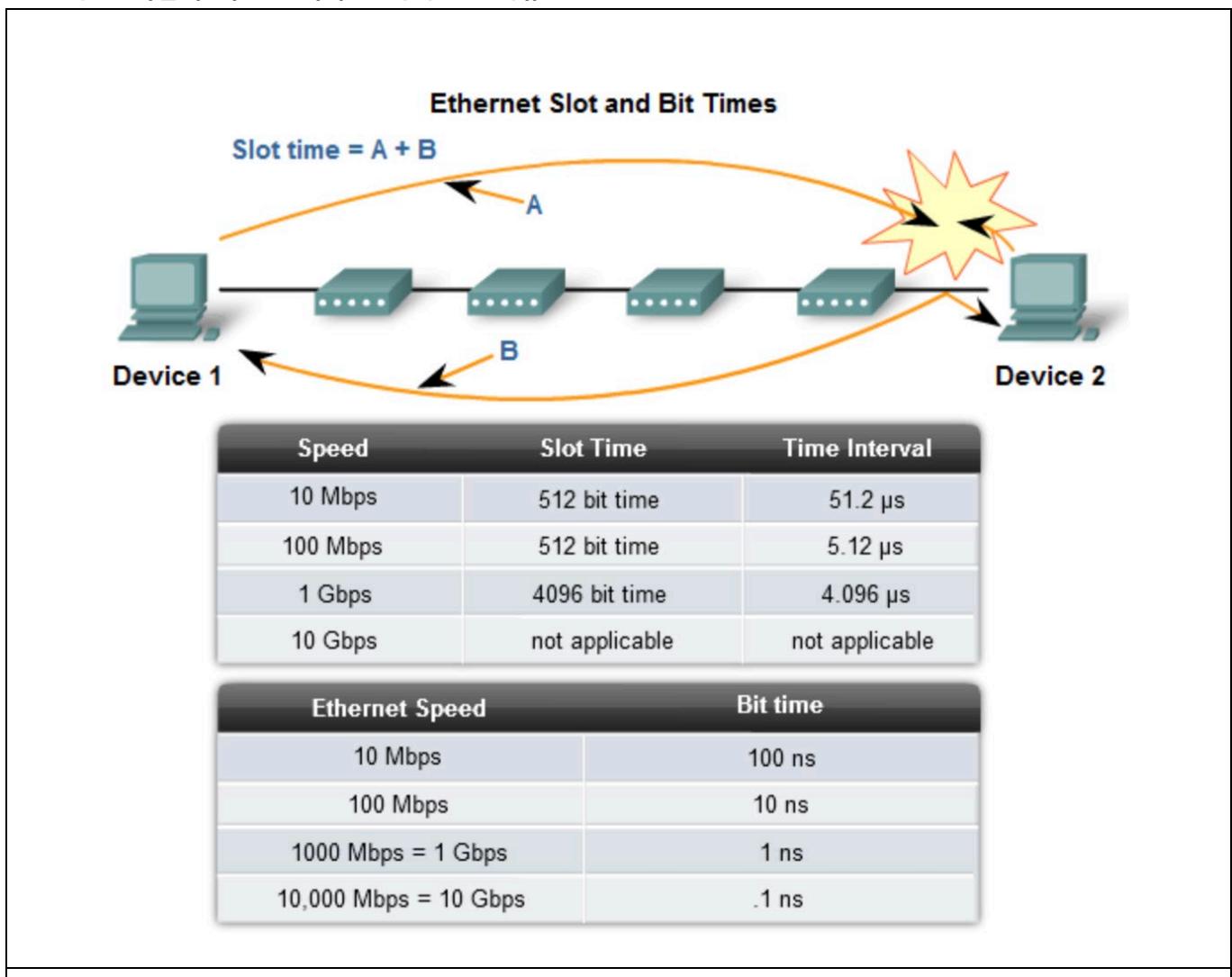
- 현재 최신 표준은 IEEE 802.3bz (2.5G, 5G 이더넷)

II. IEEE 802.3 프레임(Frame) 구조와 구성요소

구분	설명
프리엠블 (Preamble)	<ul style="list-style-type: none"> - 7바이트 크기로, 수신 호스트가 송신 호스트의 클록과 동기를 맞출 수 있도록 시간 여유를 제공하는 것이 목적 - 각 바이트는 10101010 비트 패턴을 포함
시작구분자 (Start Delimiter)	<ul style="list-style-type: none"> - 프레임의 시작 위치를 나타냄 - 프리엠블 필드와 구분해 값이 10101011
송신 호스트 주소 (Source Address)	<ul style="list-style-type: none"> - 맥 계층에서는 호스트를 구분하는 고유의 맥 주소

수신 호스트 주소 (Destination Address)	- 맥 계층에서는 호스트를 구분하는 고유의 맥 주소
길이 (Length)	- 데이터 필드에 포함된 가변 길이의 전송 데이터 - 최대값은 1,500
체크섬 (Checksum)	- 데이터 전송 과정에서 데이터 변형 오류의 발생 여부를 수신 호스트가 확인할 수 있도록 송신 호스트가 값을 기록

III. 전송 프레임의 최소 크기가 64바이트인 이유



- Ethernet의 최초 케이블 10BASE-5: 대역폭 (10Mbps), 거리 (500M)
- Ethernet half-duplex로서 충돌(Collision) 발생하는 구조 -> CSMA/CD 방식
- 왕복 최대 전송 지연 시간 : 51.2usec
- 신호의 전체 왕복시간이 51.2usec -> 512bit(=64byte) 필요

“끝”

06	SDRAM 갱신(Refresh)		
문제	DDR SDRAM(Double Date Rate Synchronous DRAM)의 갱신(Refresh)에 대하여 다음을 설명하시오. (1) DRAM에서 갱신(Refresh)이 필요한 이유 (2) 자동 갱신(Auto Refresh) (3) 셀프 갱신(Self Refresh) (4) 저전력 자동 셀프 갱신(LPASR : Low Power Auto Self Refresh)		
도메인	CA	난이도	중 (상/중/하)
키워드	전하량과 온도는 비례, Refresh, 휘발성 메모리, 속도		
출제배경	고전 토픽의 DRAM에 대한 기본기능인 Refresh에 대한 숙지 확인		
참고문헌	Micron사의 DDR_DRAM 원본 문서(https://www.micron.com/-/media/client/global/documents/products/data-sheet/dram/ddr4/8gb_ddr4_sdram.pdf)		
해설자	장건환 기술사(제 126회 정보관리기술사 / jkh556@naver.com)		

I. DDR4 SDRAM의 개념 및 DRAM에서 갱신(Refresh)이 필요한 이유

SDRAM	- DDR3 보다 2배 높은 3.2Gbps 속도, 2배 Density향상, 공정/아키텍처 진화로 20% 전력 소비 감소한 JEDEC에서 표준화한 고속의 DRAM
Refresh	- 자연적인 캐패시터의 전하 방전 현상에 대한 대책으로 일정 주기마다 전하량을 회복하는 DRAM의 주요 기능
tRC	- ROW가 Refresh하는데 소요되는 시간
tREF	- SDRAM의 Refresh 주기
Refresh의 필요성	데이터 복원 - 전하 손실이 있기 전에 주기적으로 Cell에 동일한 데이터를 복원
	전류 흐름 보존 - 전하 방전시 전압이 낮아지는 방향으로 흐르게 되는 문제를 방지
	회로 구조 단순 - Refresh를 활용함에 따라 회로구조가 단순해지고 동작속도가 빠름
	비용 - 전하 회복으로 인해 용량을 효율적으로 활용할 수 있어 효율이 좋음

II. DRAM의 자동 갱신(Auto Refresh)의 설명

구분	설명					
정의	<ul style="list-style-type: none"> - SDRAM의 컨트롤러 명령에 주행되는 다른 Refresh와 다르게 SDRAM이 IDLE 상태일 때 SDRAM 내부에서 자동으로 카운트되며 수행되는 Refresh 					
개념도						
특징	<table border="1"> <tr> <td>내부 Controller 사용</td><td>- IDLE 상태에서 외부 Controller 명령없이 내부에서 사용</td></tr> <tr> <td>Row를 동시 사용</td><td>- Row를 동시 갱신하기 때문에 좀더 경제적인 방법</td></tr> </table>	내부 Controller 사용	- IDLE 상태에서 외부 Controller 명령없이 내부에서 사용	Row를 동시 사용	- Row를 동시 갱신하기 때문에 좀더 경제적인 방법	
내부 Controller 사용	- IDLE 상태에서 외부 Controller 명령없이 내부에서 사용					
Row를 동시 사용	- Row를 동시 갱신하기 때문에 좀더 경제적인 방법					

- CBR 리플레쉬라고도 불리며 'IDLE' 상태에서만 실행 가능

III. 셀프 갱신(Self Refresh)의 설명

구분	설명					
정의	<ul style="list-style-type: none"> - 장시간 사용할 일이 없을 경우 전력소모를 줄이기 위해 사용되며, 'POWER DOWN'(대기모드) 상태에서 주기적으로 수행되는 Refresh 					
개념도						
특징	<table border="1"> <tr> <td>CKE가 'LOW' 상태</td><td>- 외부 명령없이 주기적으로 자동 Refresh</td></tr> <tr> <td>CKE가 'HIGH' 상태</td><td>- Self Refresh는 종료되고 이후 READY('IDLE') 상태가 되기 위해 tRC시간 필요</td></tr> </table>	CKE가 'LOW' 상태	- 외부 명령없이 주기적으로 자동 Refresh	CKE가 'HIGH' 상태	- Self Refresh는 종료되고 이후 READY('IDLE') 상태가 되기 위해 tRC시간 필요	
CKE가 'LOW' 상태	- 외부 명령없이 주기적으로 자동 Refresh					
CKE가 'HIGH' 상태	- Self Refresh는 종료되고 이후 READY('IDLE') 상태가 되기 위해 tRC시간 필요					

IV. 저전력 자동 셀프 갱신(LPASR : Low Power Auto Self Refresh)의 설명

구분	설명	
개념	<ul style="list-style-type: none"> - DDR4에 온도센서를 내장하여 Self Refresh 모드에서 온도에 따라 Refresh 빈도를 줄여 대기모드의 전력소모를 줄여주는 기능 - 모바일 기기에 대한 DDR4에 적용하기 위해 고려한 기능 - DRAM 셀에서 방전되는 전하량과 온도는 비례 	
개념도		
조건	85도 이상일 때	- tREF가 $3.9\mu s$ 로 줄어 2배 자주 Refresh
	45도 이하일 때	- $7.8\mu s$ 보다 길게 할 수 있음

- JEDEC 스펙 기준이며 DRAM 제조사의 재량 따라 시간은 차이가 있을 수 있음

- Refresh의 기본 전하량은 방전시간은 $7.8\mu s$

“끝”



ITPE

ICT 온라인, 오프라인 융합 No 1

PMP 자격증 정보관리기술사/컴퓨터시스템응용기술사
IT전문가과정 정보시스템감리사
정보통신기술사 애자일

오프라인 명품 강의

ITPE 기술사회

제129회 정보처리기술사 기출문제 해설집

대상 정보관리기술사, 컴퓨터시스템응용기술사, 정보통신기술사, 정보시스템감리사 시험

발행일 2023년 02월 04일

집필 강정배PE, 소민호PE, 전일PE, 정상PE, 석PE, 이상용PE, 장건환PE

출판 **ITPE(Information Technology Professional Engineer)**

주소 ITPE 대치점 서울시 강남구 선릉로 86길 17 선릉엠티빌딩 7층

ITPE 선릉점 서울시 강남구 선릉로 86길 15, 3층 IT교육센터 아이티피이

ITPE 강남점 서울시 강남구 테헤란로 52길 21 파라다이스벤처타워 3층 303호

ITPE 영등포점 서울시 영등포구 당산동2가 하나비즈타워 7층 ITPE

연락처 070-4077-1267 / itpe@itpe.co.kr

본 저작물은 **ITPE(아이티피이)**에 저작권이 있습니다.

저작권자의 허락없이 **본 저작물을 불법적인 복제 및 유통, 배포**하는 경우

법적인 처벌을 받을 수 있습니다.